VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnanasangama, Macche, Santibastwada Road
Belagavi-590018, Karnataka

A
BE MINI-PROJECT (19EC6DCMPR) REPORT
On

"Implementation of image enhancement on FPGA"

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics & Communications Engineering - ECE
By

1DS20EC011	**ADITI PARVATI**
1DS20EC023	**AMBAPRASAD N**
1DS20EC024	**AMOGH R**
1DS20EC045	**CHAITANYA**

Under the guidance
of
Dr. Pushpalatha K N
Associate Professor
ECE Dept., DSCE, Bengaluru

Department of Electronics & Communication Engineering
(An Autonomous College affiliated to VTU Belgaum, accredited by NBA & NAAC, Ranked by NIRF)
Shavige Malleshwara Hills, Kumaraswamy Layout,
Bengaluru-560111, Karnataka, India

2022-23

Certificate

Certified that the mini-project work **(19EC6DCMPR)** entitled **"Implementation of image enhancement on FPGA"** carried out by **ADITI PARVATI (1DS20EC011), AMBAPRASAD N(1DS20EC023), AMOGH R(1DS20EC024) and CHAITANYA (1DS20EC045)** are bonafide students of the ECE Dept. of Dayananda Sagar College of Engineering, Bangalore, Karnataka, India in partial fulfillment for the award of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka for the VI Semester course during the academic year 2022-23. It is certified that all corrections / suggestions indicated for the mini-project work have been incorporated in the mini-report submitted to the ECE department. This Mini-Project report has been approved as it satisfies the academic requirement in respect of mini-project work prescribed for the said degree.

Mini-Project Guide Sign:
Name: **Dr. Pushpalatha K N**

Mini-Project Section Coordinator Sign:
Name: **Prof. Trupti Tagare**

Mini-Project Convener & Chief Coordinator Sign:
Name: **Dr. Shashi Raj K.**

Dr. T. C. Manjunath:
HOD, ECE, DSCE

Dr. B. G. Prasad:
Principal, DSCE

External Mini-Project Viva-Voce (SEE)

Name of the mini-project examiners (int & ext) with date:

1: Signature: _____

2: Signature: _____

Declaration

Certified that the mini-project work entitled, "**Implementation of image enhancement on FPGA**" with the course code **19EC6DCMPR** is a bonafide work that was carried out by ourselves in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics & Communication Engineering. of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2022-23 for the VI Semester Autonomous Course. We, the students of the mini-project group/batch no. A7 do hereby declare that the entire mini-project has been done on our own & we have not copied or duplicated any other's work. The results embedded in this mini-project report has not been submitted elsewhere for the award of any type of degree.

Aditi Parvati
1DS20EC011

Sign:

AmbaPrasad N
1DS20EC023

Sign:

Amogh R
1DS20EC024

Sign:

Chaitanya
1DS20EC045

Sign:

Date: /06/2023

Place: Bengaluru

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all. Chairman **Dr Hemachandra Sagar**, Vice Chairman **Dr Premachandra Sagar**, Secretary **Galiswamy**, Joint Secretary **Tintisha Sagar**,

We express profound gratitude to respected principal **Dr. B G Prasad**, Dayananda College of Engineering for providing a congenial environment to work in. Our sincere gratitude to **Dr. T. C. Manjunath**, Head of the Department, Electronics and Communication Engineering for encouraging and providing this opportunity to carry out the project in the department.

We would like to thank our guide **Dr. Pushpalatha K N**, Associate Professor, Department of ECE who helped us in all the ways to carry out the project work. Our guide has provided us expert guidance throughout the duration of the project and lent help whenever necessary without hesitation, which we are extremely grateful for.

A special mention must be made to the **staff** throughout the college and in particular, the **ECE department** for always lending a helpful hand whenever necessary and allowing us to work on the project in the respective labs which was a holistic environment to work in. Last but not least, we must thank our **friends** and **families** for the unwavering support throughout.

Abstract

In this work we demonstrated and improved the real time configurable results of Image enhancement. Image processing plays a vital role in the fields of healthcare and services based on multimedia, etc. The Image Processing methods are traditionally implemented on a digital processing software such as MATLAB or using Digital Signal Processors (DSPs). Image Enhancement using Hardware language is a whole new approach which uses the reconfigurable circuits i.e., Field Programmable Gate Arrays (FPGAs) for digital image processing due to its high processing speed. The Image Enhancement techniques used are threshold, contrast, brightness, invert to manipulate the RGB values of every pixel of the image to improve the human interpretation of image. The implementation of the image enhancement techniques on FPGA is quite different from implementing image processing in MATLAB or using DSPs due to the parallel nature.

Table of Contents

Title Sheet		i
Certificate		ii
Declaration		iii
Acknowledgement		iv
Abstract		v
Table of Contents		vi
List of Figures		vii
Nomenclature and Acronyms		viii
Chapter 1	Introduction	1
	1.1 Overview of the mini-project work	2
	1.2 Literature survey	3
	1.3 Objectives / Scope / Aim of the mini-project work	4
	1.4 Motivation & Problem Solving	5
	1.5 Proposed Methodology	6
Chapter 2	Block diagram, Circuit Diagrams and Working principle, Algorithms, Flow-Charts & DFDs	7-11
Chapter 3	Hardware / Software tools / Description / Interfacing / Working of the complete mini-project module	12-16
Chapter 4	Results and Discussions	17-20
Chapter 5	Applications, Advantages, Outcome and Limitations	21-22
Chapter 6	Conclusions and Future Work	23-25
References		26-27
Appendix		28-29

List of Figures

Fig. 2.1: Block-diagram of the proposed methodology 7

Fig. 2.2: Flow chart for Invert Operation 8

Fig. 2.3: Flow chart for Threshold Operation 9

Fig.2.4:Flow chart for Contrast Operation 10

Fig.2.5:Flow chart for Brightness Operation 11

Fig. 3: Block Diagram of Intel DE1-SoC FPGA 12

Fig4.1: Image before contrast operation and its histogram 16

Fig 4.2: Enhanced Image after contrast operation along with its histogram 16

Fig 4.3: Image before brightness operation along with its histogram 17

Fig 4.4: Enhanced Image after reducing the brightness and its histogram 17

Fig 4.5: Image before Invert operation and its histogram 18

Fig 4.6: Enhanced Image after invert operation and its histogram 18

Fig 4.7: Image before Threshold operation and its histogram 19

Fig 4.8: Enhanced Image after Threshold operation and its histogram 19

Fig. a: Circuit Diagram for Altera Cyclone V SoC of HPS 27

Fig. b: Diagram of the DE1-SoC FPGA kit 27

Fig. c: Flowchart for sequential and Parallel implementation 28

Nomenclature and Acronyms

Abbreviations (Alphabetical Order):

ASIC	Application-Specific Integrated Circuit
BMP	Bitmap
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
HPS	Hard Processor System
HDL	Hardware Description Language
IC	Integrated Circuits
IEEE	Institute of Electrical & Electronics Engineers
MATLAB	MATrix LABoratory
RGB	Red Green Blue
USB	Universal Serial Bus
VGA	Video Graphics Array

CHAPTER-1

1. Introduction

Image enhancement refers to the techniques and algorithms used to improve the visual quality of an image. It aims to enhance specific aspects of an image such as brightness, contrast, sharpness, and color balance, among others. The goal of image enhancement is to make the image more visually appealing or to highlight certain features for better interpretation or analysis. It finds applications in various fields like medical imaging, surveillance, photography, and computer vision.

Image processing, on the other hand, is a broader field that encompasses various techniques and algorithms for manipulating digital images. It involves the acquisition, representation, analysis, and manipulation of visual data. Image processing techniques can include image filtering, segmentation, edge detection, object recognition, and many more. It is widely used in applications such as image editing software, pattern recognition, image-based automated inspection, and medical imaging.

FPGA stands for Field-Programmable Gate Array. It is a type of integrated circuit (IC) that can be programmed or reprogrammed after manufacturing. Unlike application-specific integrated circuits (ASICs) that are designed for specific functions, FPGAs offer flexibility and can be configured to perform a wide range of tasks, including digital signal processing and image processing. They consist of a matrix of configurable logic blocks and programmable interconnects, which allow designers to implement custom digital circuits and systems.

FPGAs are particularly well-suited for image processing applications due to their parallel processing capabilities. They can process multiple pixels or image regions simultaneously, which enables real-time and high-performance image processing. Moreover, FPGAs can be optimized for specific image processing algorithms, resulting in efficient hardware implementations.

In summary, image enhancement involves improving the visual quality of images, image processing encompasses a broader range of techniques for manipulating digital images, and FPGA technology offers a flexible and powerful platform for implementing image processing algorithms efficiently. Together, these concepts contribute to the advancement of various applications in fields like computer vision, medical imaging, and digital photography.

1.1 Overview

The project focuses on implementing image enhancement techniques on Field Programmable Gate Arrays (FPGAs). Traditionally, image processing methods are implemented using software-based tools like MATLAB or Digital Signal Processors (DSPs). However, this project introduces a new approach by utilizing the parallel processing capabilities of FPGAs for real-time configurable image enhancement.

As we know, Image enhancement refers to the process of improving the quality, clarity, or visual appearance of an image. It involves various techniques and algorithms designed to enhance specific aspects of an image, such as brightness, contrast, sharpness, color balance, or noise reduction. The goal of image enhancement is to make the image more visually appealing, easier to interpret, or better suited for a particular application.

The specific image enhancement techniques used in the project include thresholding, contrast adjustment, brightness adjustment, and inverting. These techniques involve manipulating the RGB values of each pixel in the image, aiming to improve the human interpretation of the image.

The project highlights the advantages of FPGA-based implementation, such as high processing speed and reconfigurability. Unlike implementing image processing in MATLAB or using DSPs, the parallel nature of FPGAs offers a different approach and potential performance benefits.

The primary objective of the project is to demonstrate and improve the real-time configurable results of image enhancement on FPGAs. The project acknowledges the significance of image processing in fields like healthcare and multimedia services.

Overall, the project presents a novel approach to image enhancement by leveraging the capabilities of FPGAs, aiming to enhance the interpretability of images for human perception

1.2 Literature survey

[*1] Luliana Chiuchisan. "An Approach to the Verilog-based System for Medical Image Enhancement", https :// ieeexplore.ieee.org/ document/ 978-1-4673 - 75 45-0 /15/31.00. 2015.*

- **Advantages**

Real-time image processing: The emphasis on real-time applications indicates that the implemented image enhancement algorithms can operate in real-time

Hardware-level filter: Implementing image enhancement algorithms at the hardware level can offer high performance and throughput due to the parallel processing capabilities of FPGA. This allows for efficient processing of medical images and can potentially handle large datasets.

- **Disadvantages:**

Complexity of FPGA design: Implementing hardware-level filters for image processing on an FPGA can be a complex task. It requires expertise in FPGA design, Verilog HDL, and image processing algorithms.

Limited resources: FPGAs have limited hardware resources, including logic elements, memory, and digital signal processing blocks. Complex image enhancement algorithms or a combination of multiple filters may require a significant number of resources, potentially limiting the complexity or scale of the implemented system.

[2] S. Sowmya and Roy Paily, "FPGA Implementation of Image Enhancement Algorithms", https://ieeexplore.ieee.org/document/5739392, 2021.

- **Advantages**

High performance processing: FPGA-based implementations can leverage the parallel processing capabilities of FPGAs, allowing for high-performance image enhancement

Reduced memory requirement: FPGA-based implementations can efficiently utilize FPGA resources, including memory. By optimizing the image enhancement algorithms specifically for the FPGA architecture, the memory requirements can be reduced processing of medical images and can potentially handle large datasets.

- **Disadvantages**

Complexity of FPGA design: Implementing hardware-level filters for image processing on an FPGA can be a complex task. It requires expertise in FPGA design, Verilog and image processing algorithms.

Limited resources: FPGAs have limited hardware resources, including logic elements, memory, and digital signal processing blocks. Complex image enhancement algorithms or a combination of multiple filters may require a significant number of resources, potentially limiting the complexity or scale of the implemented system.

[3] Prof. Narayan A. Badiger, Miss.Jayamma Muragod, "FPGA Implementation of Image Enhancement using Verilog HDL", https://www.irjet.net/, June 2020

- **Advantages**

Performance Improvement: The paper presents an improved image enhancement algorithm that focuses on enhancing performance parameters such as speed, power, and resource utilization. This implies that the implemented algorithm is designed to achieve high-speed processing on FPGA, which is advantageous for real-time or time-critical applications.

- **Disadvantages**

Limitation of LoG (Laplacian of Gausian) for processing real-time processing of coloured image: limitations of using the Laplacian of Gaussian algorithm for real-time image processing of colored images. While the details of the limitations are not specified, it suggests that there may be challenges or performance trade-offs when applying the LoG algorithm to colored images, such as increased computational complexity or memory requirements.

1.3 Objectives

The objective of the mini project is to practically implement image enhancement techniques on an FPGA platform. By developing hardware-based solutions for enhancing image quality, the project aims to demonstrate the advantages and effectiveness of FPGA technology in real-time image processing. The objective involves selecting and implementing specific image enhancement algorithms, optimizing them for FPGA hardware, and evaluating the performance of the implemented system. The project aims to showcase the potential of FPGA for high-speed, powerful, and cost-effective image enhancement, while also considering factors such as resource utilization and image quality. Ultimately, the objective is to contribute to the field of FPGA-based image processing and provide valuable insights for future research and applications.

- **Aim:**

The aim of the mini project is to investigate and demonstrate the feasibility and benefits of implementing image enhancement algorithms on an FPGA platform. The project seeks to explore the potential of FPGA-based solutions for real-time image processing by designing and implementing hardware-level algorithms. The project aims to contribute to the advancement of FPGA-based image processing techniques and provide a solid foundation for further research and development in this field.

- **Advantages of using FPGA:**

1.Higher processing speeds

2.Operations performed in parallel

3.Low latency

1.4 Motivation & Problem Statement

We have chosen the project "Implementation of Image Enhancement using FPGA" due to the compelling motivations and opportunities it presents. Image enhancement is a critical component in image processing applications, serving as a foundation for a wide range of areas and finding increasing utility in fields like medicine. However, implementing image enhancement techniques can be demanding in terms of processing units, especially when dealing with high-resolution images. Traditional approaches relying on DSPs or software tools may struggle to meet the requirements of high speed, power efficiency, and cost-effectiveness. This is where FPGA (Field-Programmable Gate Array) technology comes into play.

By utilizing FPGA, we can leverage its inherent advantages to address the challenges posed by image enhancement algorithms. FPGA allows for higher processing speeds, significantly reducing the time required for image enhancement tasks. Additionally, FPGA-based solutions offer lower power consumption compared to traditional methods, making them more environmentally friendly and cost-effective. The ability to perform operations in parallel enables concurrent processing of multiple pixels, resulting in enhanced performance. Moreover, FPGA-based implementations exhibit low latency, enabling real-time image processing and responsiveness.

Traditionally, image processing tasks, including image enhancement, have been carried out using DSPs (Digital Signal Processors) or software tools. However, as the complexity of real-time systems increases, there is a growing need for more efficient ways to perform image processing. The demand for high-resolution image enhancement techniques poses challenges in terms of processing power and speed. The project aims to overcome the limitations of traditional approaches and contribute to the advancement of image processing techniques. The FPGA based implementation will offer superior performance and enable real-time image enhancement, opening up new possibilities in various applications.

1.5 Proposed Methodology

The proposed image enhancement system involves the conversion of the image file into a hex format using MATLAB, application of enhancement algorithms such as contrast adjustment, brightness correction, thresholding, and image inversion with the use of spatial parallelism for efficient processing, and the generation of an output image with enhanced pixel values and standard BMP header values. This methodology aims to improve the quality of the image by preparing the data, applying enhancement operations, and producing a visually improved output.

Chapter 2

Block diagram, Algorithms, Flow-Charts and Working principle

2.1 Block diagram

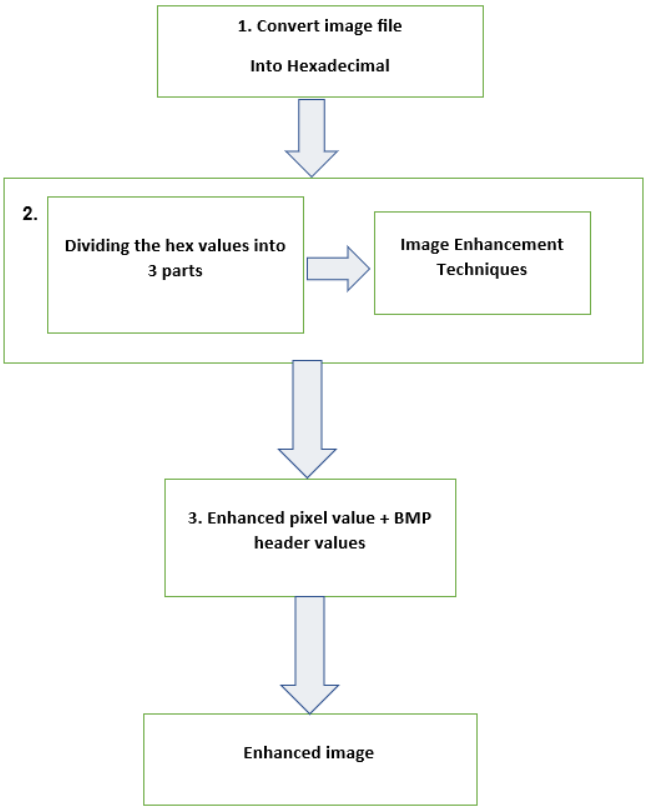


Fig. 2.1: Block-diagram of the proposed methodology

The proposed image enhancement system is implemented into three phases shown in the block Diagram.

Block 1: Concerns with how to prepare the image data, for which we have used MATLAB to convert the image file into a hex file

Block 2: Describes the spatial parallelism functional unit. During this, distribution can be executed and an algorithm on the image is applied for contrast, brightness, threshold, and invert operations.

Block 3: Deals with the enhanced pixel values and writes standard BMP header values along to see the output image.

2.2 Algorithm applied given as flowchart

2.2.1 Invert Operation Algorithm

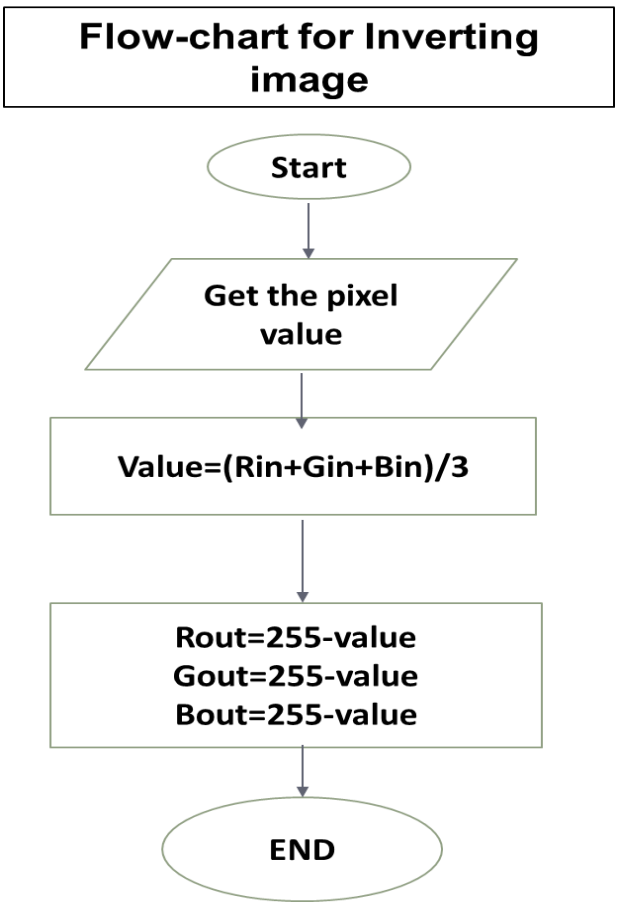


Fig. 2.2: Flow chart for Invert Operation

Invert Operation Algorithm:

Step 1: Start the program and initialize variables to store the sum and count of numbers.

Step 2: Read the hexadecimal values from the hex file, representing the pixel values.

Step 3: Divide the pixel value into its red, green, and blue components, and calculate their average.

Step 4: Subtract the average value from 255 to obtain the difference.

Step 5: Store the difference in the RGB out variable.

Step 6: Write the updated pixel values, including the inverted colors, back to the BMP image file.

2.2.2 Threshold Operation Algorithm

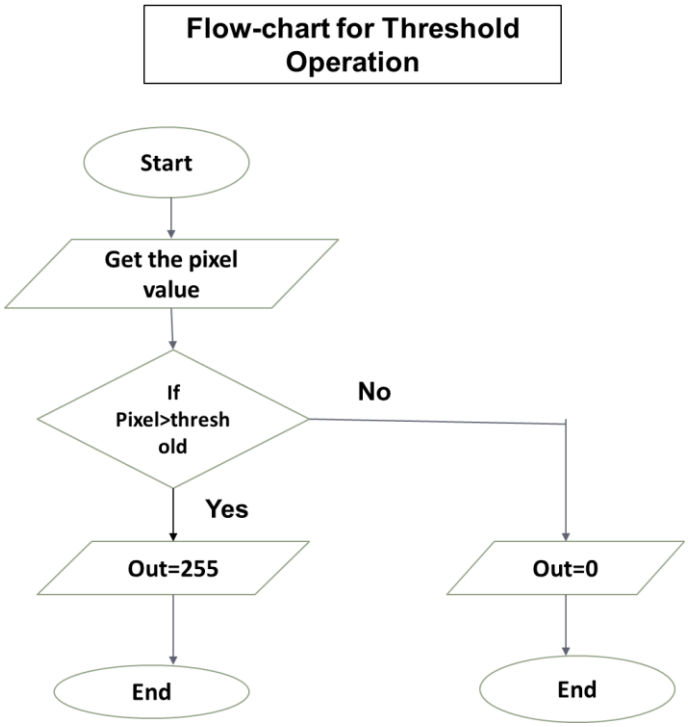


Fig. 2.3: Flow chart for Threshold Operation

Threshold Operation Algorithm:

Step 1: Start the program and initialize variables.

Step 2: Read the hex values from the hex file (pixel values).

Step 3: Divide the pixel value into RGB components.

Step 4: Calculate the average intensity of the RGB components.

Step 5: Define a threshold value.

Step 6: Compare the average intensity with the threshold.

Step 7: If the average intensity is greater than or equal to the threshold, set the RGB components to their maximum value (255); otherwise, set them to their minimum value (0).

Step 8: Write the updated RGB values back to the bmp image file

2.2.3 Contrast Operation Algorithm

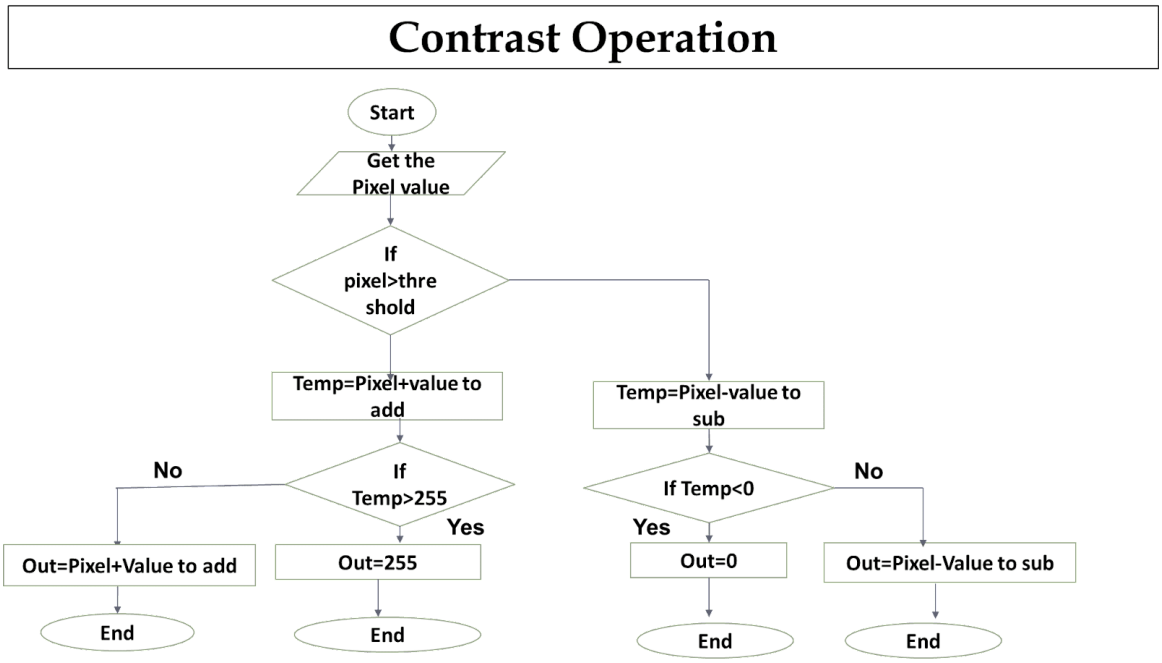


Fig.2.4: Flow chart for Contrast Operation

Contrast:

Step 1: Start the program and initialize variables.

Step 2: Read the hex values from the hex file (pixel values).

Step 3: Divide the pixel value into RGB components.

Step 4: Calculate the average intensity of the RGB components.

Step 5: Define a contrast factor.

Step 6: If the average value is greater than the predefined threshold value, add the contrast adjustment factor to the average value. If the average value is less than the predefined threshold value, subtract the contrast adjustment factor from the average value.

Step 7: Clip the RGB values to the valid range (0-255).

Step 8: Write the updated RGB values back to the bmp image file.

2.2.2 Brightness Operation Algorithm

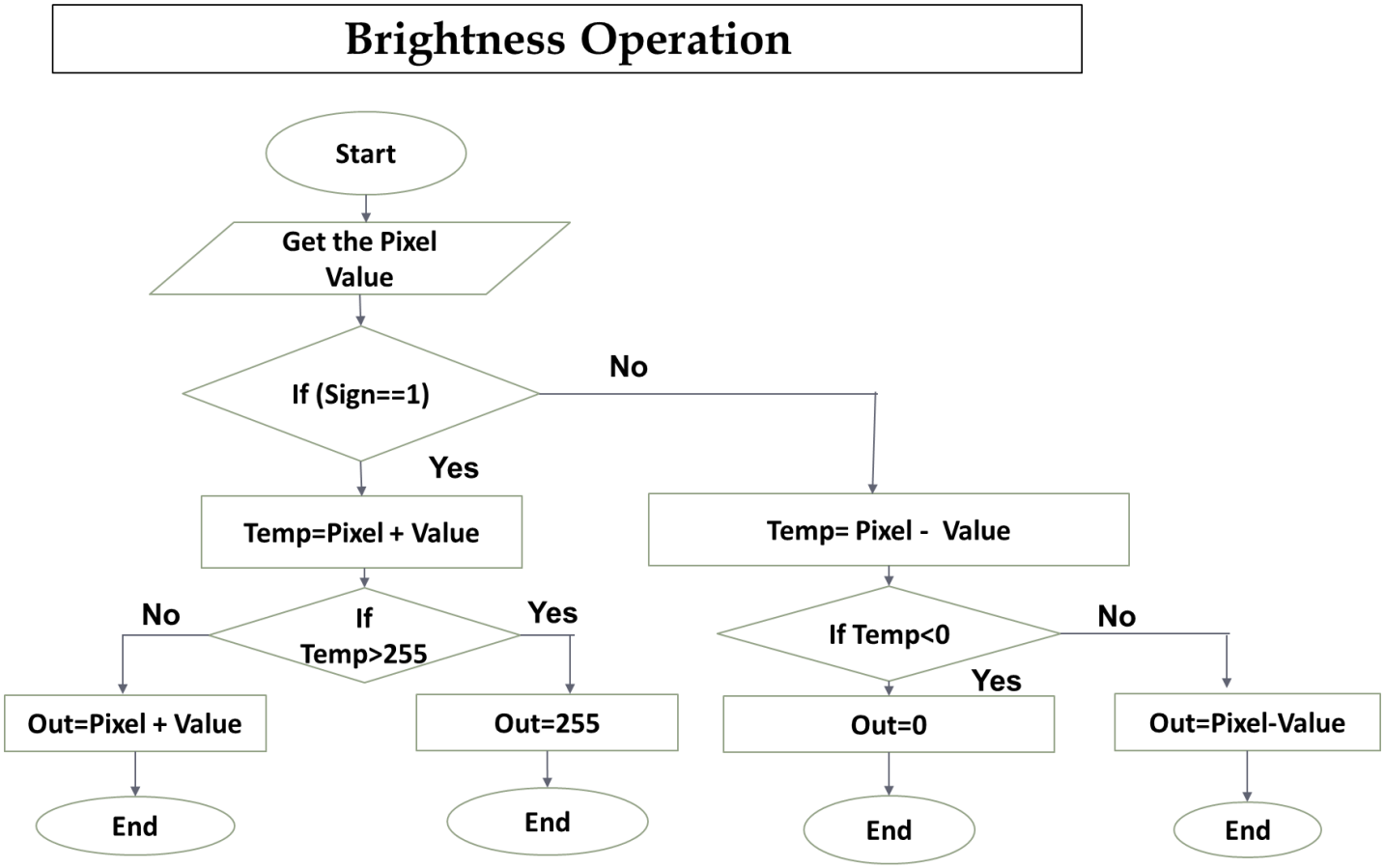


Fig. 2.5: Flow chart for Brightness Operation

Brightness Operation Algorithm:

Step 1: Start the program and initialize variables.

Step 2: Read the hex values from the hex file (pixel values).

Step 3: Divide the pixel value into RGB components.

Step 4: Calculate the average intensity of the RGB components.

Step 5: Define a brightness adjustment factor.

Step 6: If the sign flag value is 1, add the brightness adjustment factor to the average of RGB values.

If the sign flag value is 0, subtract the brightness adjustment factor from the average of RGB values.

Step 7: Clip the RGB values to the valid range (0-255).

Step 8: Write the updated RGB values back to the bmp image file.

Chapter 3.

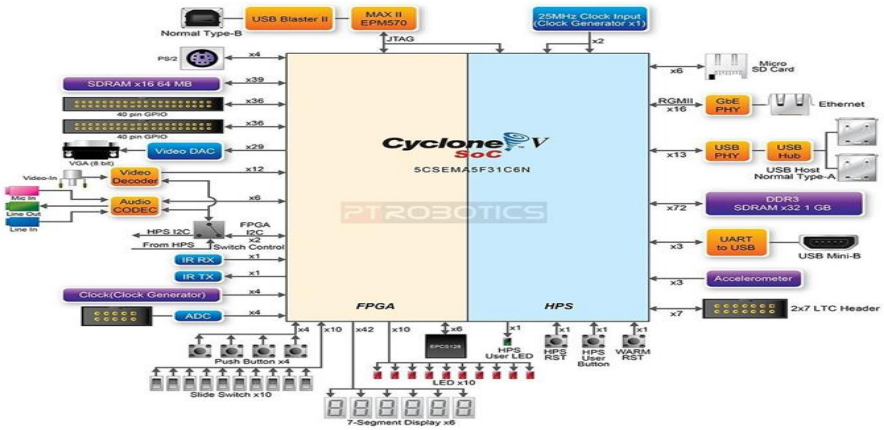
Hardware and Software Description along with Working of the complete mini-project module

- **Hardware:**

The hardware that is used for the mini-project work is Intel DE1-SoC FPGA Board.

The Intel DE1-SoC FPGA kit is a development board designed by Intel for prototyping and developing digital logic and embedded systems. The board features a Cyclone V SoC FPGA, which combines a dual-core ARM Cortex-A9 processor with programmable logic resources. The DE1-SoC provides 1GB of DDR3 memory for the ARM processor, as well as 512MB of shared memory between the FPGA and ARM. It also includes an SD card slot for additional storage.

The board offers a variety of peripherals, including VGA, HDMI, and LCD connectors for video output, audio jacks, USB ports, Ethernet, and UART for communication, and Fig. 5: Block-diagram of the proposed methodology



various switches, buttons, and LEDs for user interaction. The DE1-SoC features multiple

Fig. 3.1: Block Diagram of Intel DE1-SoC FPGA

expansion headers, including GPIO headers, an Arduino header, and HSMC and LTC connectors. With the DE1-SoC FPGA kit, you can explore various applications, such digital signal processing, image and video processing, robotics, Internet of Things (IoT) devices, and more.

➢ Software:

The software tools used for the mini-project work are MATLAB, ModelSim, and PuTTY.

MATLAB is a high-level programming language and development environment widely used for numerical computing, data analysis, and algorithm development. It stands for "MATrix LABoratory" and was developed by MathWorks.

MATLAB provides a comprehensive set of mathematical and numerical functions, making it suitable for performing a wide range of computations. MATLAB provides a comprehensive set of mathematical and numerical functions, making it suitable for performing a wide range of computations. MATLAB is commonly used for developing and testing algorithms. It provides a convenient environment for writing and debugging code, with features such as built-in debugging tools and profiling capabilities.

In our project, we are using MATLAB in order to obtain the hexadecimal equivalent value of pixels of the image on which we are supposed to perform the operations. The hexadecimal values are then fed as input to the Verilog code during simulation.

ModelSim Intel FPGA Edition (formerly known as ModelSim Altera Edition) is a simulation and verification tool specifically designed for FPGA (Field-Programmable Gate Array) design and development. It is developed and supported by Intel (formerly Altera), a leading manufacturer of programmable logic devices.

ModelSim allows you to simulate the behavior of your digital design at the register-transfer level (RTL) or higher abstraction levels. ModelSim provides a graphical waveform viewer that allows you to visualize and analyze signals and waveforms during simulation. ModelSim supports the creation of testbenches, which are used to simulate the interaction between the design under test (DUT) and the test environment. ModelSim offers coverage analysis capabilities to assess the quality and completeness of your testbench.

In our project, we are using ModelSim for simulation of the Verilog code written for performing image enhancement operations and obtaining the desired results.

PuTTY is a free and open-source terminal emulator and SSH client for Windows, widely used for remote access to servers, network devices, and other systems. It provides a

secure and reliable way to establish SSH (Secure Shell) connections and perform various remote administration tasks.

PuTTY supports SSH, allowing you to establish encrypted and secure connections to remote servers and devices. PuTTY provides a terminal emulation environment, allowing you to interact with remote systems through a command-line interface. It supports various terminal types, including xterm, VT100, and ANSI. In addition to SSH, PuTTY can also establish direct serial connections to devices that use a serial console interface. PuTTY includes utilities for file transfer over SSH, including the secure file transfer protocol (SFTP) and the secure copy protocol (SCP). PuTTY allows you to save session configurations for quick and easy access to frequently used remote connections.

In our project, we are using PuTTY for the purpose of communication between the system and the FPGA board.

- **Working:**

In this part, we are going to present a brief explanation about the working of the mini project.

In the first step, we are considering an image file in BMP format on which the desired image enhancement operation to be performed. Then we will be using MATLAB to convert the pixel values of the image file to hexadecimal values as it becomes compatible for FPGA implementation.

In the next step, we are going to perform the simulation of Verilog Code of Image Enhancement Operations in ModelSim Intel FPGA edition. Here we are going to consider the MATLAB output as input to the Verilog code. The Verilog Code consists of four parts: Image_read, Image_write, Operations and testbench. Image_read part is used to hexadecimal value from the .hex file and perform the desired operations on the pixel values of the image and Image_write part is used to reframe the pixels in its original form to get the processed image. Operations file is used to select which image enhancement operation has to be performed and the Testbench file is used for simulation. After simulation, we will get the desired output image which could be later compared with results obtained from the FPGA implementation.

Now, we will be performing the implementation of Image Enhancement operations on FPGA. Here we are using PuTTY terminal as a means of communication between the

HPS on the FPGA board and the system being used. Through PuTTY, we are dumping the code on HPS. Then the contents are being transferred to the FPGA from HPS through the FPGA bridge. Hence the final step of implementation has been performed and the results are being displayed on the VGA monitor through VGA cable connected to the FPGA board.

Chapter 4

Results and Discussions

A basic idea about histogram: The histogram of an image represents the distribution of pixel intensities. It shows the frequency of occurrence of different intensity values in the image. A typical histogram has intensity values on the x-axis and the corresponding frequency on the y-axis.

❖ *Contrast Operation Output:*

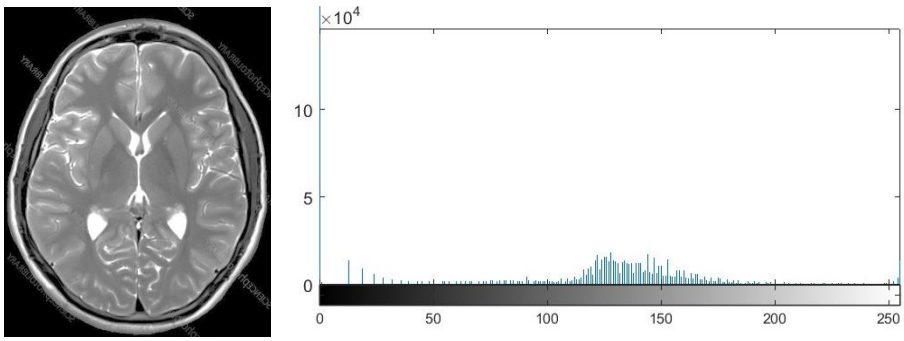


Fig 4.1: Image before contrast operation and its histogram

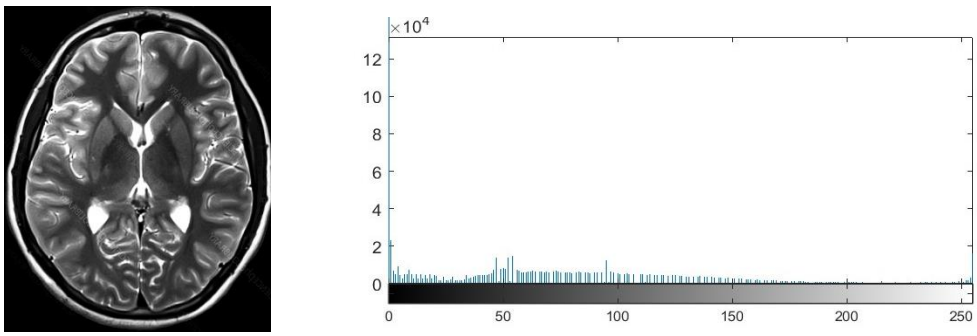


Fig 4.2: Enhanced Image after contrast operation (threshold value:80 and value to add:10) along with its histogram

In contrast operation, the distribution of intensities has been modified to accentuate the contrast. The extent of the changes depends on the specific method used for contrast enhancement.

After performing a contrast operation, the histogram shows an increased spread of intensity values. The dark regions get pushed towards lower intensity values, and the

bright regions get pushed towards higher intensity values. This results in a histogram that is stretched or expanded across a wider range of intensities.

❖ *Brightness Operation Output:*

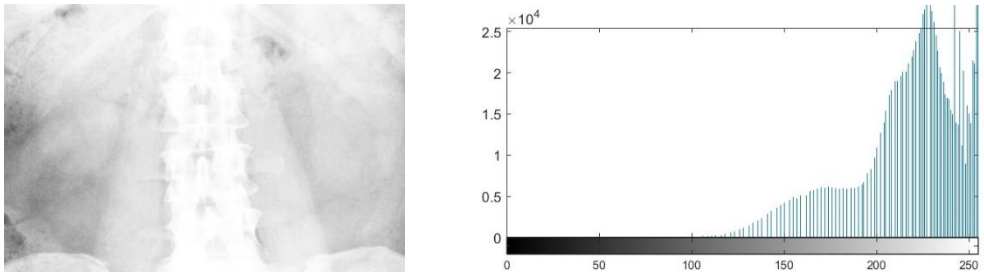


Fig 4.3: Image before brightness operation along with its histogram

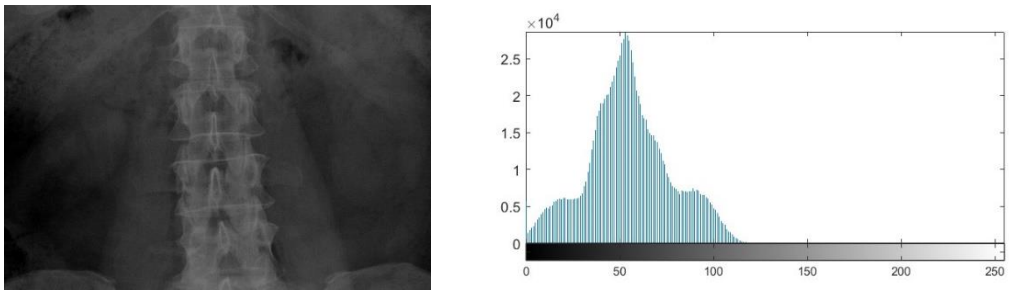


Fig 4.4: Enhanced Image after reducing the brightness(value to subtract:100) and its histogram

Before the brightness operation, the histogram had a diverse range of intensity values spread across the histogram. It includes pixels with high intensities (representing bright regions) as well as pixels with low intensities (representing dark regions). The distribution of intensities was reflective of the original image's content.

After reducing brightness, the intensities that were originally in the brighter range shifts towards lower values. Consequently, the histogram becomes compressed towards the left side, indicating a decrease in the frequency of higher intensity values

❖ *Invert Operation Output:*

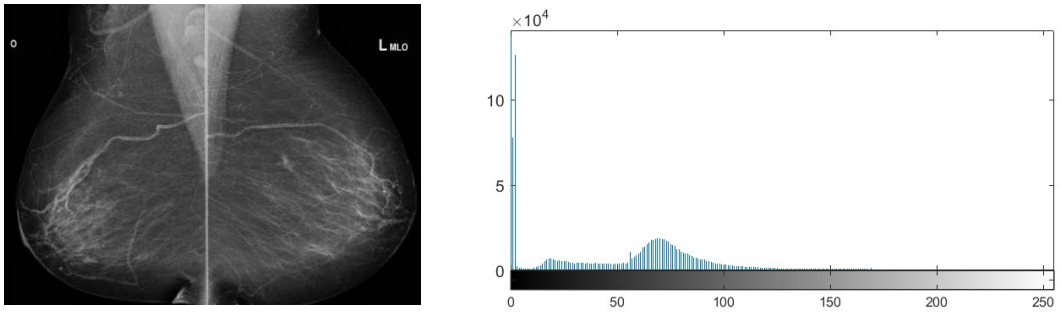


Fig 4.5: Image before Invert operation and its histogram

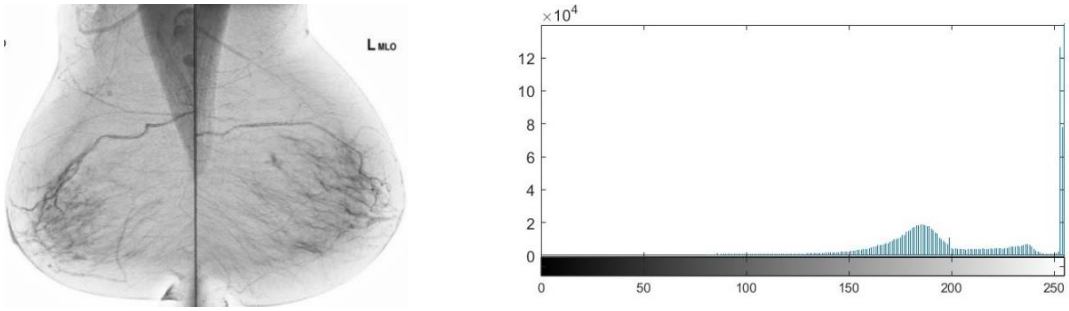


Fig 4.6: Enhanced Image after invert operation and its histogram

After performing invert operation, the intensities that were originally in the darker range get shifted towards higher values, and the intensities that were in the brighter range get shifted towards lower values. This flipping of intensities in the image gets reflected in the histogram. Histogram is mirrored along the vertical axis. The intensities that were originally more frequent in the darker regions of the histogram has become more frequent in the brighter regions, and vice versa. The overall shape and distribution of the histogram will remain the same, but the intensities will be reversed.

❖ *Threshold Operation Output:*

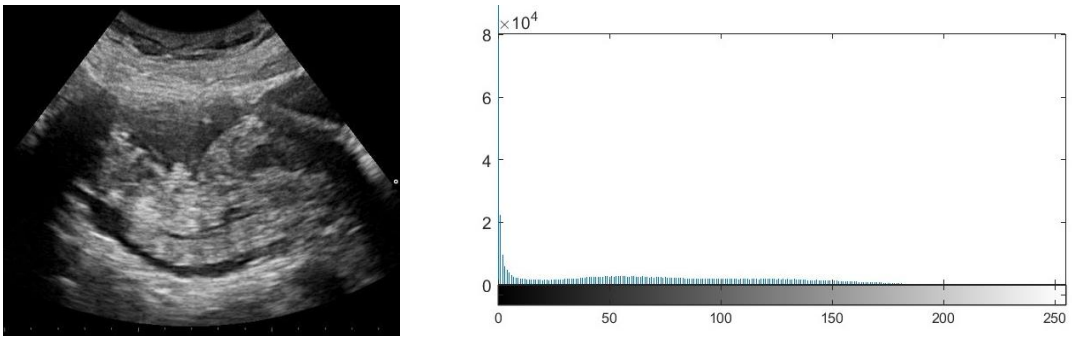


Fig 4.7: Image before Threshold operation and its histogram

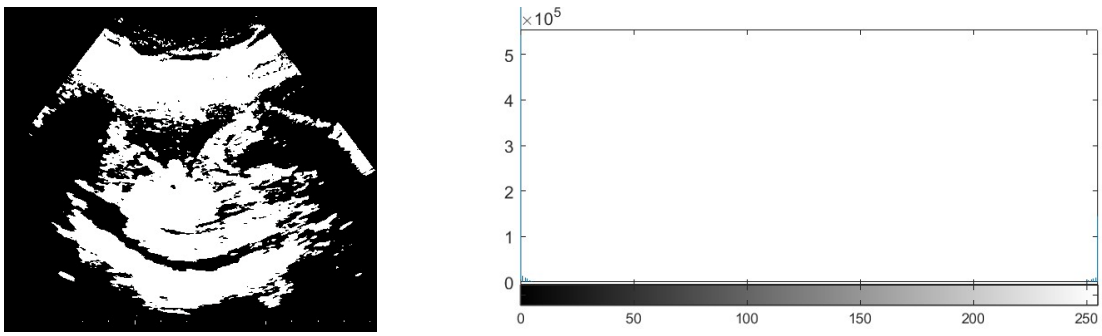


Fig 4.8: Enhanced Image after Threshold operation (threshold value:80) and its histogram

Threshold operation effectively divides the histogram into two parts: one representing pixels below the threshold and another representing pixels above the threshold. The histogram will become bimodal, with a clear separation between the two categories. The height and width of these peaks will depend on the threshold value and the number of pixels falling into each category. The overall shape of the histogram will be modified to reflect this separation into two distinct intensity ranges.

Chapter 5.

Applications, Advantages, Outcome and Limitations

- ***Application of Implementation of Image Enhancement using FPGA:***

1.**Medical Imaging:** FPGA-based image enhancement algorithms are utilized to improve the clarity, contrast, and detail of X-ray and CT images. By reducing noise, enhancing edges, and adjusting the brightness and contrast levels, FPGA implementations can enhance the visibility of anatomical structures, making it easier for medical professionals to interpret the images accurately. They are also used to enhance the image quality of MRI scans. FPGA-based image enhancement techniques can mitigate artifacts, reduce noise, and improve spatial resolution, resulting in clearer and more detailed MRI images.

2. **Surveillance and Security:** FPGA-based image enhancement can enhance the quality of images captured by surveillance cameras, aiding in object recognition and tracking. They can improve the visibility of objects in surveillance videos, making it easier to detect and track individuals or objects of interest. FPGA implementations can enhance the quality of license plate images captured by surveillance cameras, enabling reliable license plate recognition systems. It enables real-time processing of surveillance video feeds. The inherent parallelism and low-latency nature of FPGAs allow for immediate enhancement and analysis of incoming video frames.

3. **Automotive Vision Systems:** FPGA-based image enhancement can enhance the performance and reliability of automotive vision systems for applications like lane detection and pedestrian detection. By applying algorithms for image denoising, edge enhancement, and dynamic range compression, FPGA implementations can improve the clarity, detail, and realism of the virtual or augmented content. FPGA-based image enhancement can be used in broadcasting applications to improve the visual quality of transmitted content. By applying algorithms for noise reduction, color correction, and dynamic range compression, FPGA implementations can enhance the clarity, detail, and color accuracy of broadcasted images or videos.

4. **Remote Sensing:** FPGA-based image enhancement can improve the clarity and quality of remotely sensed images for better analysis and interpretation.

5. **Multimedia and Entertainment:** FPGA-based image enhancement can enhance the visual quality of images and videos for gaming, virtual reality, and video post-processing applications.

- ***Advantages of Implementation of Image Enhancement using FPGA:***

1. **Parallel Processing:** FPGAs are highly parallel devices that can perform multiple operations simultaneously. In image processing, this parallelism is advantageous because many image processing algorithms involve performing the same operation on multiple pixels of image regions simultaneously. FPGA devices enable parallel processing, leading to faster image enhancement compared to sequential processing on DSP models.
2. **Customization:** FPGAs allow for custom hardware design and optimization, resulting in improved performance and resource efficiency.
3. **Low Latency**: FPGA models offer low-latency processing, making them suitable for real-time applications. Low latency processing can also be applied on real-time applications such as video processing or object recognition.
4. **Energy Efficiency:** FPGA models can be optimized for power consumption, leading to energy-efficient image enhancement implementations. FPGAs can be more power efficient as compared to general purpose processors.
5. **Flexibility and Upgradability:** FPGA models can be reprogrammed, enabling flexibility and upgradability as new algorithms or improvements are developed. FPGAs are reconfigured to implement different image processing algorithms or adapt to changing requirements. This flexibility allows for rapid prototyping, experimentation, and optimization of image processing pipelines.
6. **Real-Time Adaptability:** FPGAs enable real-time adaptability, allowing image processing algorithms to be dynamically modified or adjusted during runtime. This capability is beneficial in applications where the processing requirements change based on varying environmental condition.

- ***Outcome of Implementation of Image Enhancement using FPGA:***

1. **Enhanced Image Quality:** FPGA-based image enhancement can improve image clarity, contrast, and visibility, leading to better visual quality and more accurate

interpretation of images. The ability to process images at high speeds and fine-grained control over the algorithms allows for precise adjustments, resulting in visually appealing and clearer images.

2. **Real-time Processing:** FPGA implementations enable real-time image enhancement, allowing for immediate feedback and response in time-critical applications. By leveraging the parallel processing capabilities of FPGAs, image enhancement algorithms can be applied to incoming image frames in real-time. This is particularly beneficial in applications such as video surveillance and medical imaging where immediate analysis and enhanced visualization of images are crucial.

3. **Increased Efficiency:** FPGA models optimize hardware resources and parallelize processing, resulting in faster and more efficient image enhancement compared to software-based DSP models.

- ***Limitations of Implementation of Image Enhancement using FPGA:***

While FPGA-based image enhancement offers significant advantages, it is essential to consider the limitations and associated development challenges when deciding to implement image enhancement using FPGA technology.

1. **Hardware Complexity:** FPGA development requires specialized knowledge and expertise in hardware description languages (HDLs) like VHDL or Verilog, making it more complex than software-based DSP development.

2. **Development Time:** FPGA-based image enhancement implementations may require more time for development and optimization compared to DSP models due to the hardware-level programming involved.

3. **High Initial Cost**: FPGAs can be relatively expensive compared to DSP processors, which may impact the initial cost of implementing image enhancement systems. Additionally, if the application requires a high volume of units, the cost per unit for FPGA-based solutions may be higher than for software-based solutions.

Chapter 6

Conclusions and Future Work

In this chapter, a brief conclusion of the proposed mini-project work that is being done undertaken is being presented here.

The implementation of image enhancement on FPGA, incorporating operations such as brightness adjustment, thresholding, contrast enhancement, and inversion, has demonstrated the potential for efficient and real-time image processing using hardware acceleration. The FPGA-based solution has allowed for parallel processing and optimized performance, resulting in improved image quality and enhanced visual appearance. We specifically chose medical images, such as those depicting the brains, kidney etc., to highlight the significance and potential use cases of image enhancement in the medical field.

Through brightness adjustment, we were able to control the overall illumination of the medical images. By carefully tuning the brightness levels, we could enhance the visibility of subtle details within the brain, making them more discernible and aiding in diagnostic processes.

Contrast enhancement played a crucial role in enhancing the distinction between different tissues and structures in the medical images. By expanding the range of pixel intensities, we effectively improved the visual quality, making it easier to differentiate areas of interest within the images.

Thresholding proved to be a powerful tool in image segmentation, where we could separate the regions of interest from the background by determining an optimal threshold value. This allowed us to isolate specific structures within the medical images, facilitating precise analysis and highlighting areas of potential concern.

The use of FPGA technology, coupled with medical image enhancement algorithms, holds immense potential in advancing medical imaging applications and ultimately improving patient care.

By leveraging the FPGA's capabilities, the project has successfully demonstrated the feasibility of implementing image enhancement algorithms on dedicated hardware, offering advantages such as low latency, high throughput, and flexibility for

customization. The use of FPGA technology enables real-time processing, making it suitable for applications where fast and responsive image enhancement is crucial, such as video processing, medical imaging, and surveillance systems.

In terms of future work, several areas can be explored to further enhance the project:

1. The implemented image enhancement algorithms can be further optimized for improved performance, resource utilization, and better visual results. Exploring advanced techniques and adapting algorithms specific to FPGA architecture can lead to better efficiency and image quality.
2. Consider integrating additional image enhancement techniques, such as noise reduction, edge enhancement, or color correction, into the FPGA implementation. This can broaden the range of improvements and provide a more comprehensive image enhancement solution.
3. Integrate the FPGA-based image enhancement system into practical applications or systems. This may involve interfacing with different image sources, such as cameras or video streams, and integrating with larger systems for real-world deployments.

   Explore power optimization techniques to minimize power consumption while maintaining the performance and image quality. This can involve power-aware design methodologies and implementing power management strategies tailored to the FPGA platform.

References

[1] LulianaChiuchisan. “An Approach to the Verilog-based System for Medical Image Enhancement”, https :// ieeexplore.ieee.org/ document/ 978-1-4673 - 75 45-0 /15/31.00. 2015.

[2] S. Sowmya and Roy Paily, “FPGA Implementation of Image Enhancement Algorithms”, https://ieeexplore.ieee.org/document/5739392, 2021.

[4] S. C. Chan, H. O. Ngai and K. L. Ho, "A programmable image processing system using FPGA," Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94, London, UK, 1994, pp. 125-128 vol.2, doi: 10.1109/ISCAS.1994.408921.

[5] M. I. AlAli, K. M. Mhaidat and I. A. Aljarrah, "Implementing image processing algorithms in FPGA hardware," 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, 2013, pp. 1-5, doi: 10.1109/AEECT.2013.6716446.

[6] M. A. Altuncu, M. M. Kösten, M. A. Çavuşlu and S. Şahın, "FPGA-based implementation of basic image processing applications as low-cost IP core," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018, pp.1-4, doi: 10.1109/SIU.2018.8404175.

[7] P. G. Patel, A. Ahmadi and M. Khalid, "Implementing an Improved Image Enhancement Algorithm On FPGA," 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), ON, Canada, 2021, pp. 1-6,doi: 10.1109/CCECE53047.2021.9569049.

[8] T. M. Bittibssi, Y. Z. Mohasseb, G. I. Salama and A. E. Henawy, "Image Enhancement Algorithms using FPGA," 2012 8th International Computer Engineering Conference (ICENCO), Giza, Cairo, Egypt, 2012, pp. 96-99, doi: 10.1109/ICENCO.2012.6487097.

Websites:

[1]. http://www.nptel.org

[2]. https://ui.adsabs.harvard.edu/abs/2020SPIE11519E..1LX/abstract

[3]. https://www.youtube.com/watch?v=iW7oGRXX_dY

[4]. https://medium.com/image-processing-using-fpga

[5]. https://www.ecstuff4u.com/2018/08/advantages-and-disadvantages-of-fpga.html

[6]. https://www.geeksforgeeks.org/digital-image-processing-basics/

Appendix

➢ IC Pin Configurations:

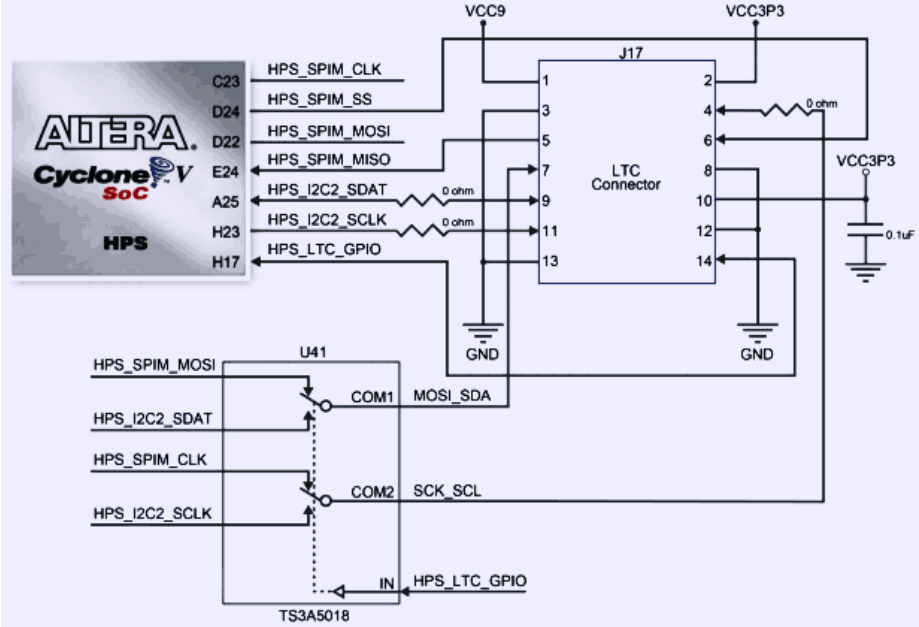


Fig. a: Circuit Diagram for Altera Cyclone V SoC of HPS

➢ Intel FPGA DE1-SoC:

Fig. b: Diagram of the DE1-SoC FPGA kit.

➢ Sequential Vs Parallel Operation:

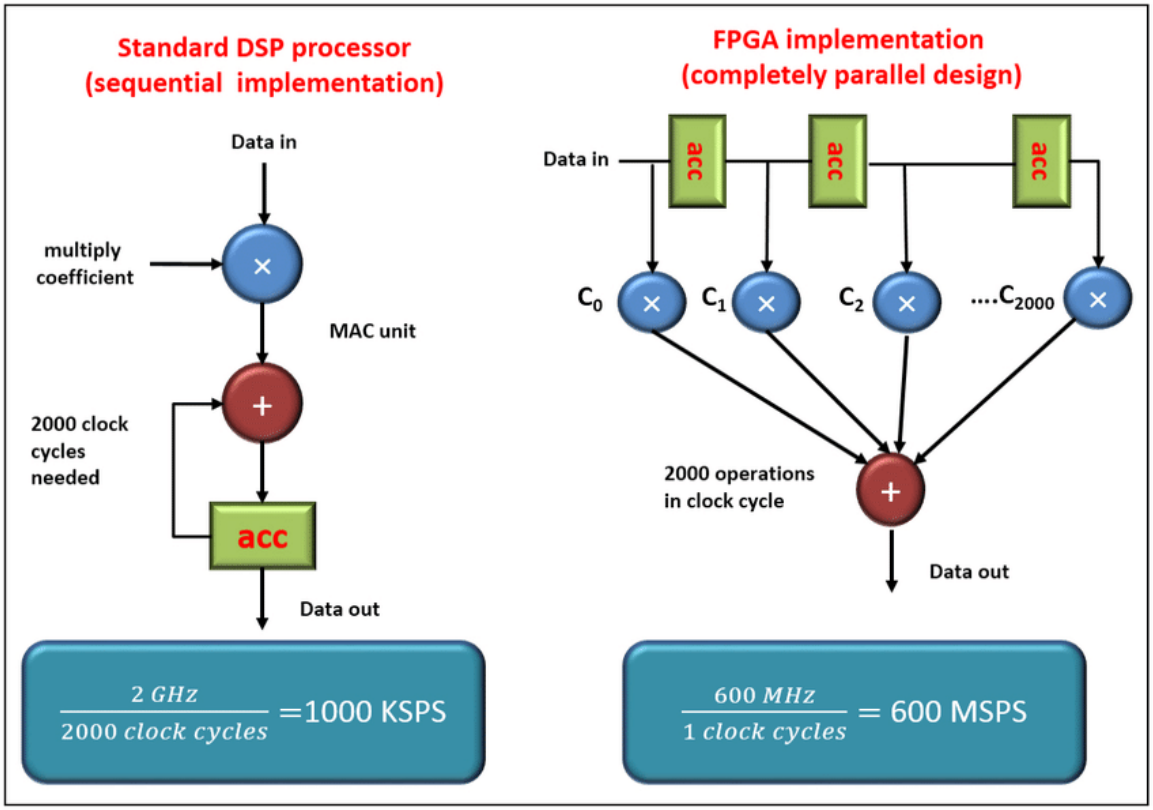


Fig. c: Flowchart for sequential implementation and Parallel implementation