

Enhancement of Digital Photo Frame Capabilities With Dedicated Hardware

A Thesis submitted in partial fulfillment of the requirements for the
degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Leo Kurians Paulose

Roll No. 108EC035

Cheedella Phani Teja

Roll No. 108EC024

Under the supervision of

Dr. Kamala Kanta Mahapatra

Professor



**Department of Electronics and Communication Engineering,
National Institute of Technology, Rourkela
Session 2011-2012**

Enhancement of Digital Photo Frame Capabilities With Dedicated Hardware

A Thesis submitted in partial fulfillment of the requirements for the
degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Leo Kurians Paulose

Roll No. 108EC035

Cheedella Phani Teja

Roll No. 108EC024

Under the supervision of

Dr. Kamala Kanta Mahapatra

Professor



Department of Electronics and Communication Engineering,

National Institute of Technology, Rourkela

Session 2011-2012



National Institute of Technology, Rourkela

C E R T I F I C A T E

This is to certify that the Thesis entitled, '**Enhancement of Digital Photo Frame Capabilities using Dedicated Hardware**' submitted by **Leo Kurians Paulose and Cheedella Phani Teja** in partial fulfillment of the requirements for the award of **Bachelor of Technology Degree in Electronics and Communication Engineering** at the **National Institute of Technology, Rourkela** is an authentic work carried out by him under my supervision. To the best of my knowledge and belief the matter embodied in the Thesis has not been submitted by him to any other University/Institute for the award of any Degree/Diploma.

Date

Prof. Kamala Kanta Mahapatra

Dept. of Electronics and Communication Engg.,

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

This project in itself is an acknowledgement to the inspiration, drive and the technical assistance contributed to it by many people. It would have never seen the light of day without the help and guidance that it received from them.

Firstly, we would like to express my sincere thanks and deepest regards to my guide

Dr. K K Mahapatra, Professor, Department of Electronics and Communication Engineering, NIT Rourkela, who has been the driving force behind this work. We thank him for giving me the opportunity to work under him by putting a trust in my credentials and capabilities, and helping me in exploring my potential to the fullest.

We are grateful to **Prof. Sukadev Meher**, Head of the Department of Electronics and Communication Engineering, for permitting me to make use of the facilities available in the department to carry out the project successfully.

We are thankful to **Mr. Vijay Sharma**, PG student in the Department of Electronics and Communication Engineering, NIT Rourkela, for his generous help and continuous encouragement in various ways towards the completion of this project.

Last but not the least we would like to thank all my friends for their support. We are thankful to our classmates for all the thoughtful and mind stimulating discussions we had, prompting us to think beyond the obvious.

Leo Kurians Paulose

Cheedella Phani Teja

ABSTRACT

Photo frames have come a long way since the typical ones that needed to have a photo printed and stuck on them. Today in this digital era we have a new concept, named digital photo frame, a modern representation of the conventional photo frame. A digital photo frame is basically a picture frame that displays photos without the need to print them. They are available in a variety of sizes and with varied configurations. A typical frame varies in size from 7 inches to 20 inches. There are also key chain sized frames available. These frames also support a variety of formats like .jpeg, .tiff, .bmp and so on. Most of the frames provide an option to run the photos in a sequential or random manner as a slideshow with an adjustable time interval. The mode of input of the photos to the frame is also multi-fold. It can be done directly via the memory card of the camera, or else various memory devices like USB drives, SD Cards, MMC Cards and so on can be used. Nowadays even Bluetooth technology is being used. Another option that is becoming quite popular is that, users can take their photos directly from the Internet from sites like Flickr, Picassa or from their e-mail. Also these frames generally come with built in speakers and with remote controls.

Our initial objective was to decide on which all features can be added to the Digital Photo Frame that we design. For this purpose we conducted simulation exercises in MATLAB so as to prove its feasibility.

This simulation exercise was divided into two parts. The first part was to perform compression and decompression and the second half dealt with the various enhancements that can be added to the frame. For our compression and decompression we considered the JPEG standard. Joint Photographic Experts Group - an ISO/ITU standard for compressing still

images. The JPEG format is very popular due to its variable compression range. A few limitations of JPEG include the fact that it is lossy and also not great for displaying text. The common extension for it include *.jpg, *.jff, *.m-jpeg,*.mpeg

The various enhancement features that we tested for feasibility include Mean Filter, Median Filter, Image Sharpening, Negative Image Extraction, Logarithmic Transformations, Power Law Correction (Gamma Correction), Contrast Stretching, Grey Level Slicing, Bit Plane Slicing, Laplace Filtering.

We then proceeded onto the hardware implementation of the above said features. We only implemented a handful of features owing to the complexity of design and lack of time. We first implemented the Compression and Decompression algorithm. The two enhancement features we implemented were Laplace Filter and Median Filter. For our implementation we used the VIRTEX 2 FPGA Board.

Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1	Motivation	01
1.2	Problem Statement	02
1.3	Literature Review	02
1.3.1	JPEG Standard	02
1.3.2	Modes of Operation	03
1.3.3	Image Enhancement	04
1.3.4	FPGA	07
1.3.5	FPGA Architecture	07
1.3.6	FPGA Design Flow	09
1.3.7	Behavioural Simulation	10
1.3.8	Synthesis of Design	11
1.3.9	Design Implementation	11
1.3.10	Advantages	13
1.3.11	FPGA Specifications	14
1.4	Patent Search	15

1.4.1 Patent 1 (US 2005/0057578 A1)	15
1.4.2 Patent 2 (US 2008/0030478 A1)	15
1.4.3 Patent 3 (US 2008/0273126 A1)	16
1.4.4 Patent 4 (US 2010/0013810 A1)	17

CHAPTER 2: FEASABILITY VERIFICATION

2.1 Approach	18
2.2 Image Enhancement	22
2.2.1 Image Negation	22
2.2.2 Logarithmic Transformation	23
2.2.3 Gamma Correction	24
2.2.4 Piece Wise Linear Transformation	26
2.2.4.1 Contrast Stretching	26
2.2.4.2 Grey Level Slicing	28
2.2.4.3 Bit Plane Slicing	29
2.2.5 Smooth Linear Filtering	30
2.2.6 Median Filtering	31
2.2.7 Laplace Filtering	32

CHAPTER 3: HARDWARE IMPLEMENTATION: COMPRESSION AND DECOMPRESSION

3.1	Discrete Cosine Transform	34
3.1.1	Introduction	34
3.1.2	Two Dimensional FDCT	35
3.1.3	DCT Module	35
3.1.4	2D DCT Architecture	38
3.1.5	RTL Schematic	39
3.1.6	DCT and Quantization Output	40
3.1.7	DCT and Quantization Design Summary	40
3.2	Normalized quantization matrix for hardware simplification	41
3.2.1	Reordering using Zigzag sequence matrix	42
3.2.2	Huffman Coding Architecture Implementation	42
3.2.3	RTL Schematic	43
3.2.4	Simulation Result	44
3.2.5	Huffman Coding Results using MATLAB	44

CHAPTER 4: HARDWARE IMPLEMENTATION: ENHANCEMENTS

4.1	Median Filtering	46
-----	------------------	----

4.1.1	RTL Schematic	46
4.1.2	Design Summary	47
4.1.3	Simulation Result	48
4.1.4	Hardware Output	48
4.2	Laplace Filter	49
4.2.1	RTL Schematic	49
4.2.2	Design Summary	49
4.2.3	Simulation Result	50
4.2.4	Hardware Output	50
CHAPTER 5: CONCLUSION AND FUTURE WORK		51
References		52

List of Figures

Figure No.	Title	Page No
1.1	FPGA Architecture	8
1.2	FPGA Design Flow	10
2.1	DCT Based Decoder Processing Steps	19
2.2	Information packing in DCT coefficients	20
2.3	Image Compression and Decompression for different Q values	21
2.4	Plot of various transformation functions	22
2.5	Negative Image Extraction	23
2.6	Logarithmic Transformation	24
2.7	Gamma Correction (1)	25
2.8	Gamma Correction (2)	26
2.9	Contrast Stretching	28
2.10	Grey Level Slicing	29
2.11	Bit Plane Slicing (1)	29
2.12	Bit Plane Slicing (2)	29
2.13	Smooth Linear Filtering (1)	31
2.14	Smooth Linear Filtering (2)	31
2.15	Median Filtering	32
2.16	Laplace Filtering	33
3.1	Adder and Subtractors Structure for 8-point DCT	38
3.2	RTL Schematic	39

3.3	DCT and Quantization Output	40
3.4	DCT and Quantization Design Summary	40
3.5	Zig Zag Order Sequence Matrix	42
3.6	RTL Schematic	43
3.7	Simulation Result	44
3.8	Huffman Coding Results using MATLAB	44
4.1	RTL Schematic (1)	46
4.2	RTL Schematic (2)	47
4.3	Design Summary	47
4.4	Simulation Result	48
4.5	Final Output	48
4.6	RTL Schematic	49
4.7	Design Summary	49
4.8	Simulation Results	50
4.9	Hardware Output	50

List of Tables

Table	Title	Page No
2.1	Compression Ratio for Different Quantization Level	21

Chapter 1

Introduction

1.1 Motivation

Photo Frames have existed for time immemorial. Whenever we wanted to remember a particular event or occasion and share it with the others, the first thought that popped in our mind was to frame the photo. Be it getting a medal or a holiday trip in Europe, photo frames have always been used. Infact we have become very accustomed to it in our daily life. However a few limitations do arise. Firstly, the space constraint. There are only a finite number of photo frames that we can place in our rooms. But our memories see no bounds. Hence there arises a conflict of interests. Secondly, the photos can get tarnished over the years because of the environmental conditions.

Hence Digital Photo Frames are the need of the hour. They can store multiple images and display them continuously. Also there is no risk of the image being destroyed by environmental factors, because the photos are digital. As soon as this product was introduced in the market, it had been selling as hot pie. Over the years the features in digital photo frames have increased, but the basic design remains the same.

A Digital Photo Frame essentially consists of a screen to display the pictures, memory to store the photos and a power supply to power the device. Many optional features like remote control, touch screens and so on are being made available nowadays. The key motivation to

work in this area is that, the development in this field is quite limited, as this is an emerging technology. Thus the quality of research in this field is at minimal compared to other sectors. Secondly, this is a highly market friendly avenue. Any considerable development in this field can be marketed, as companies are eager to add-on more features to their Photo Frames. Thirdly, this field provides a huge scope for patenting. As mentioned earlier, the research done in this field is quite primitive, thus further advances can be patented.

Infact in the course of this thesis, the work that was developed as a result of this project, was being processed for application of a patent.

1.2 Problem Statement

Most of the commercial digital photo frames available today offer only the basic of image manipulation features like red eye correction and image cropping. Our aim is to develop a library of features that can be added on to the photo frame design so as to increase its capability.

1.3 Literature Review:

1.3.1 JPEG Standard:

The primary image format that we shall consider for all our experimentation purposes is JPEG. Digital Imaging applications have increased over the years. This is mainly owing to the advances in the various aspects of digital technology, with prime focus to image acquisition, data storage, bitmapped printing and display and the like. However the cost being high becomes a major factor in making these applications specialised. With certain

exceptions like that of facsimile, digital images are not common place in general purpose computing systems like how text and other graphics are.

The mode of image acquisition remains predominantly analog in nature. One of the chief limiting factor is the vast amount of data that needs to represent digital images. A digitized version of a single, multi-colour picture contains over a million bytes in data. A higher resolution lens, which is very common today requires much more data than that. The cost involved in maintaining and storing images is increasing owing to the alarmingly increasing size of images.

Thus the advent of image compression is of high relevance. Modern technologies offer image compressions which can compress images to 1/10 to 1/50 their original value without visibly affecting the image quality. But compression does not alone suffice. A standard image compression method is required to help facilitate interoperability of equipment from different manufacturers.

This is where JPEG comes to play. JPEG is abbreviated as Joint Photographic Experts Group. This group had been working towards establishing the first international digital image compression standard for continuous tone (multilevel) still images, both grayscale and colour. The 'joint' refers to a collaboration between CCITT and ISO. JPEG convenes officially as the ISO Committee designated JTC1/SC2/WG10, but operates in close informal collaboration with CCITT SGVIII. JPEG will be both an ISO Standard and a CCITT Recommendation.

1.3.2 Modes of Operation:

JPEG has the following modes of operation:

- *Sequential Encoding*: Each image component is encoded in a single left to right, top to bottom scan.

- *Progressive Encoding*: The image is encoded in multiple scans for applications in which the transmission time is long, and the viewer prefers to watch the image build up in multiple coarse to clear passes.
- *Lossless Encoding*: The image is encoded to guarantee exact recovery of every source image sample value
- *Hierarchical Encoding*: The image is encoded at the multiple resolutions so that lower-resolution versions may be accessed without first having to decompress the image at its full resolution.

1.3.3 Image Enhancement:

The principle objective of enhancement is to process an image so that the result is more suitable than the original for a specific purpose.

The various Image Features are as follows:

- *Negative Image Extraction*:

The negative of an image with gray levels in the range $[0, L-1]$ is obtained by using the negative transformation which is given by the expression:

$$s = L-1-r$$

- *Logarithmic Transformations*:

The general form of the log transformation is:

$$s = c \log (1+r)$$

Any curve having the general shape of the log functions would accomplish this spreading/compression of gray levels of an image.

- *Power Law Correction (Gamma Correction):*

Gamma correction is important if displaying an image accurately on a computer screen is of concern. Images that are not properly corrected can likely look very dark. The use of Gamma correction has increased tremendously over the past few years because of the use of digital images for various commercial purposes over the Internet. Today most the latest monitors and screens come inbuilt with the Gamma Correction Feature.

- *Piece Wise Linear Transformation:*

- *Contrast Stretching:*

It is a simple piecewise linear function. Images with low contrast can result from poor illumination, lack of dynamic range in the image sensor, or wrong aperture setting of lens while capturing the image.

- *Grey Level Slicing:*

Many a time we might require to highlight a specific range of gray levels in an image. It has a varied amount of applications including enhancing flaws in X-Rays and enhancing satellite imagery.

- *Bit Plane Slicing:*

Consider that each pixel in an image is represented by 8 bits. Thus we would like to highlight the contribution made by each respective bits. In terms of 8 bit bytes , plane 0 contains all the lowest order bits in the bytes comprising of pixels in the image and plane 7 contains all the high order bits. It is noteworthy to mention that higher order bits (especially the top four) contains the majority of the visually significant data.

The rest of the bits contribute to the more subtle details in the image. A major advantage of this method is that, we can use it, to find the significance of a particular bit.

- *Image Subtraction:*

The difference between two images can be obtained by computing the difference between all pairs of corresponding pixels from f and h. The images are represented as $f(x,y)$ and $h(x,y)$. The difference is expressed as:

$$g(x,y) = f(x,y) - h(x,y)$$

- *Smoothing Linear Filter:*

The output of a smoothing, linear spatial filter is the average of the pixels contained in the neighbourhood of the filter mask. These filters are also called averaging filters. They are also referred to as low pass filters.

- *Median Filtering*

They are non – linear spatial filters whose response is based on ordering the pixels contained in the image area encompassed by the filter, and then replacing the value of the centre pixel with the value determined by the ranking result.

- *Laplace Filtering:*

This method basically consists of defining a discrete formulation of the second order derivative and then constructing a filter mask based on that formulation.

1.3.4 FPGA:

FPGA is abbreviated as Field Programmable Gate Array. It can be programmed by the designer after manufacturing and during designing. This is also known as On Site programmable. Their structure is similar to gate array or ASIC rather than PAL. They are used to prototype ASICs or its used to substitute in places where we are confident that eventually ASIC will be used. This is penultimate interest to one, when the prime objective is to get the design to the market first. The programming of FPGA is done using HDL. The programmable logic blocks are called configurable logic blocks and reconfigurable interconnects are called switch boxes. Logic blocks can be programmed to perform complex combinational functions, or simple logic like AND and XOR. In a majority of FPGA's the logic blocks also include memory elements, which can be as simple as a flip flop or as complex as complete blocks of memory.

1.3.5 FPGA Architecture:

FPGA architecture are variations of the figure shown below, however the final architecture depends on the seller. Essentially the architecture consists of Configurable I/O blocks, Programmable Interconnects and Configurable Logic Blocks. It also has a clock circuitry to drive the clock signals to each logic block. Also other resources like ALU's, Decoders and Memory may be available.

Two basic programmable elements available are Static RAM and Anti-fuses. The number of CLBs and I/Os required can easily be determined from the design but the number of routing tracks is different even within the designs employing the same amount of logic.

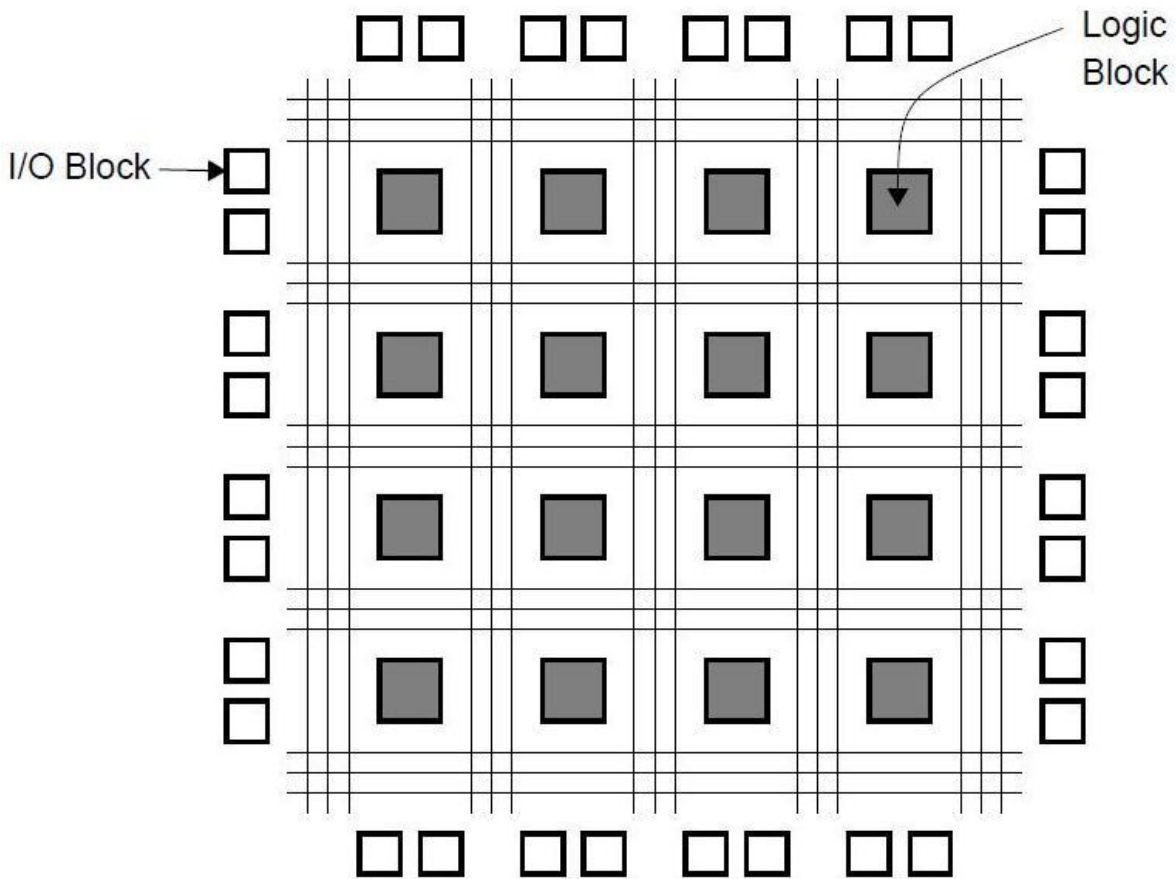


Fig 1.1 : FPGA Architecture

1. *Configurable Logic Blocks:*

They contain the logic for the FPGA. CLBs contain RAM for creating arbitrary combinatorial logic functions. It also has flip-flops for clocked storage elements, and multiplexers that route the logic within the block to/from external resources.

2. *Configurable I/O Blocks:*

Configurable I/O block is used to route signal towards and away from the chip. It comprises input buffer, output buffer with three states and open collector output controls. Pull-up and Pull-down resistors may also be present at the output. The output polarity is programmable for active high or active low output.

3. Programmable Interconnects :

FPGA interconnect is similar to that of a gate array ASIC and different from a CPLD. There are long lines that interconnect critical CLBs located physically far from each other without introducing much delay. They also serve as buses within the chip. Short lines that interconnect CLBs present close to each other are also present. Switch matrices that connect these long and short lines in a specific way are also present. Programmable Switches connect CLBs to interconnect lines and interconnect lines to each other and the switch matrix. Three-state buffers connect multiple CLBs to a long line creating a bus. Specially designed long lines called Global Clock lines are present that provide low impedance and fast propagation times.

4. Clock circuitry:

Special I/O blocks having special high-drive clock buffers, called clock drivers, are distributed throughout the chip. The buffers are connected to clock I/P pads. They drive the clock signals onto the Global Clock lines described above. The clock lines have been designed for fast propagation time and less skew time.

1.3.6 FPGA Design Flow:

The flow for the design using FPGA outlines the whole process of device design, and guarantees that none of the steps is overlooked. Thus, it ensures that we have the best chance of getting back a working prototype that will correctly function in the final system to be designed.

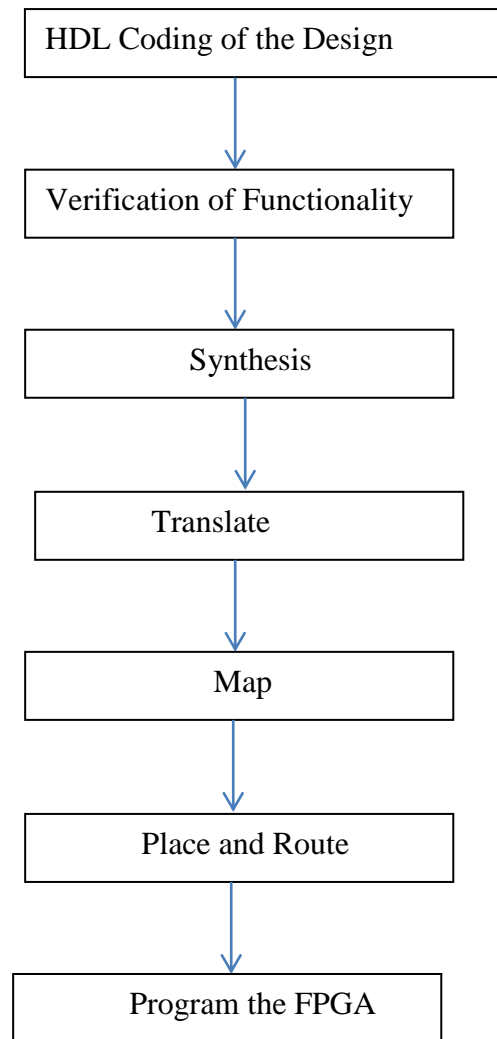


Fig 1.2 : FPGA Design Flow

1.3.7 Behavioural Simulation:

After HDL designing, the code is simulated and its functionality is verified using simulation software, e.g. Xilinx ISE or ModelSim simulator. The code is simulated and the output is tested for the various inputs. If the output values are consistent with the expected values then we proceed further else necessary corrections are made in the code. This is what is known as Behavioral Simulation. Simulation is a continuous process. Small sections of the design should be simulated and verified for functionality before

assembling them into a large design. After several iterations of design and simulation the correct functionality is achieved. Once the design and simulation is done then another design review by some other people is done so that nothing is missed and no improper assumption made as far as the output functionality is concerned.

1.3.8 Synthesis of the Design:

Post the behavioral simulation the design is synthesized. During simulation following takes place:

(i) HDL Compilation:

The Xilinx ISE tool compiles all the sub-modules of the main module. If any problem takes place then the syntax of the code must be checked.

(ii) HDL synthesis:

Hardware components like Multiplexers, Adders, Subtractors, Counters, Registers, Latches, Comparators, XORs, Tri-State buffers, Decoders are synthesized from the HDL code.

1.3.9 Design Implementation:

(i) Translation:

The translate process is used to merge all of the input net-lists and the design constraints. It outputs a Xilinx NGD (Native Information and Generic Database) file. The logical design reduced to Xilinx device primitive cells is described by this .ngd file. Here, User Constraints are defined by assigning the ports in the design to physical elements. (e.g. pins, switches, buttons, etc) for the target device as well as

specifying timing requirements. This information is stored in a UCF file which can be created using PACE or Constraint Editor.

(ii) Mapping:

After the translation process is complete the logical design described in the .ngd file to the components or primitives (Slices/CLBs) present on the .ncd file is mapped onto the target FPGA design. The whole circuit is divided into smaller blocks so that they can be appropriately fit into the FPGA blocks. The mapping is done onto the CLBs and IOBs in accordance with the logic.

(iii) Placing and Routing:

After the mapping process the PAR program is used to place the sub-blocks from the map process onto the logic blocks as per the constraints and then connect these blocks. Trade-off between all the constraints is taken into account during the placement and routing process. Place process places the sub-blocks according to logic but does not provide them the physical routing. On running the Route process physical connections between the sub-blocks are made using the switch-matrices.

(iv) Bit file generation:

Bit-stream is used to describe the collection of binary data used to program the reconfigurable logic device. The 'Generate Programming File' process is run after the FPGA design has been completely routed. It runs BitGen, the Xilinx bit-stream generation program, to produce a .bit or .isc file for Xilinx device configuration.

Using this file the device is configured for the intended design using the JTAG boundary scan method. The working is then verified for different inputs.

(v) Testing

System testing is necessary to ensure that all parts of the system correctly work together after the prototype is mapped onto the system. If the system doesn't work then the problem can be fixed by making some changes in the system or the software. The problems are documented so that on the next revision or production of the chip they are fixed. When the ICs are produced it is necessary to have some sort of burnt-in self-test mechanism such that the system gets tested regularly over a long period of time.

1.3.10 Advantages:

FPGAs have become very popular in the recent years owing to the following advantages that they offer:

Fast prototyping and turn-around time-

Prototyping is defined as the building of an actual circuit to a theoretical design to verify for its working, and to provide a physical platform for debugging the core if it doesn't. Turnaround is the total time between expired between the submission of a process and its completion. On FPGAs interconnects are already present and the designer only needs to fuse these programmable interconnects to get the desired output logic. This reduces the time taken as compared to ASICs or full-custom design.

NRE cost is zero-

Non-Recurring Engineering refers to the one-time cost of researching, developing, designing and testing a new product. Since FPGAs are reprogrammable and they can be used without any loss of quality every time, the NRE cost is not present. This significantly reduces the initial cost of manufacturing the ICs since the program can be implemented and tested on FPGAs free of cost.

High-Speed-

Since FPGA technology is primarily based on referring to the look-up tables the time taken to execute is much less compared to ASIC technology.

Low cost-

FPGA is quite affordable and hence is very designer-friendly. Also the power requirement is much less as the architecture of FPGAs is based upon LUTs.

Due to the above mentioned advantages of FPGAs in IC technology and DCT in mapping of images, implementation of DCT in FPGA can give us a clearer idea about the advantages and limitations of using DCT as the mapping function. This can help in forming better image compression and restoration techniques.

1.3.11 FPGA Specifications:

The FPGA we have considered in this project has the following specifications:

- Vendor : XILINX
- Family : Virtex 2
- Synthesis Tool : VHDL
- Simulator : Xilinx ISE 10.1

1.4 Patent Search:

As this is a new emerging technology the number of patents filled in this field are quite limited. An extensive Patent Search had been conducted and the 4 most relevant patents have been given below along with their abstracts.

1.4.1 Patent 1:

Pub No : US 2005/0057578 A1

Pub Date : Mar. 17,2005

Abstract :

“A Digital Photo Frame comprises a storage unit for storing various picture data and music data. A built in control software can be used to select a matching music and a matching digital outer frame pattern for each picture. When displaying a different picture, the matching music can be played and the matching digital outer frame can be displayed automatically. Through the sound recording function, a matching music can be recorded for each picture. The alarm clock function and the radio reception function can also be added into this digital photo frame. With the control software, the time display format and the position can be set. The music to be played can also be selected for the alarm clock. Through the digital photo frame, a user can select the picture and the outer frame pattern to be displayed and the music to be played according to his mood and liking.”

1.4.2 Patent 2:

Pub No : US 2008/0030478 A1

Pub Date : Feb 7, 2008

Abstract :

“A digital photo frame includes a processing unit, a display controller, a display unit and a control panel. The processing unit modifies an original image from an original image file stored in a first storage according to modifying instructions received from the control panel, saves modifying parameters relating to the modification in an image modification file that is stored in a second storage and corresponds to the original image file. After the modification, if the original image is selected at a later time, the processing unit automatically modifies the original image according to the modifying parameters, thereby producing a modified image to be displayed on the display unit.”

1.4.3 Patent 3:

Pub No : US 2008/0273126 A1

Pub Date : Nov 6, 2008

Abstract :

“A digital photo frame apparatus comprises at least one memory card slot, at least one transmission interface port, a power supply device, a signal processing controller, a display device, a display driver module and an input control module. The memory card slot is for connecting to a memory card to read digital image data in a memory card, and the signal processing controller is for processing inputted digital image data and controlling its playback method, and the display device and the display driver module are for displaying digital images. The input control module is for controlling an input device which is connected to a port, such as a keyboard , a mouse or a microphone, so that a user can use the input device to input data to a digital image for adding or recording text data, graphic data or audio data to the digital image.”

1.4.4 Patent 4:

Pub No : US 2010/0013810 A1

Pub Date : Jan 21,2010

Abstract :

“A digital photo frame (DPF) includes a power source, a display panel, a light detector, a motion detector, a processing unit and power management unit. The light detector is configured to detect the ambient brightness. The motion detection unit is configured to detect whether anyone is around. The processing unit is connected to the light detector and the motion detector. If the motion detector detects someone is around the DPF and the light detector detects the ambient brightness is below a predetermined value, the processing unit controls the power management unit to provide power to display panel.”

Chapter 2

Feasibility Check

2.1 Approach

Our first objective is to find the feasible enhancement techniques that we wish to incorporate into our Digital Photo Frame. For checking the feasibility, we implement the enhancements in MATLAB. In the sections below, we have briefly described the enhancement feature and shown the outputs that we achieved using MATLAB.

Image Compression and Decompression

Size of the image is always a concern while dealing with limited storage. We have two types of storage on the digital photo frame. One is the internal memory and the other is the external memory which can be interfaced by MMC, SD Cards, USB Drives and so on.

However the memory is always limited. Hence our prime objective is to compress the image with at least loss in clarity. This will allow storing more photos in the limited space.

For our compression and decompression we shall consider the JPEG standard. Joint Photographic Experts Group - an ISO/ITU standard for compressing still images.

The JPEG formats are very popular owing to its variable compression range. JPEG is saved on a sliding resolution scale, depending on desired quality. A few limitations of JPEG include the fact that it is lossy and also not great for displaying text. The common extension for it include *.jpg, *.jff, *.m-jpeg, *.mpeg

We perform DCT Based coding for implementing our compression module. The figure shown in the next page details the various aspects included in the compression and

decompression. The compression method is usually lossy, meaning that some original image information is lost and cannot be restored, possibly affecting image quality.

There is an optional lossless mode defined in the JPEG standard; however, that mode is not widely supported in products. A number of alterations to a JPEG image can be performed losslessly (that is, without recompression and the associated quality loss) as long as the image size is a multiple of 1 MCU block (Minimum Coded Unit) (usually 16 pixels in both directions, for 4:2:0 chroma subsampling).

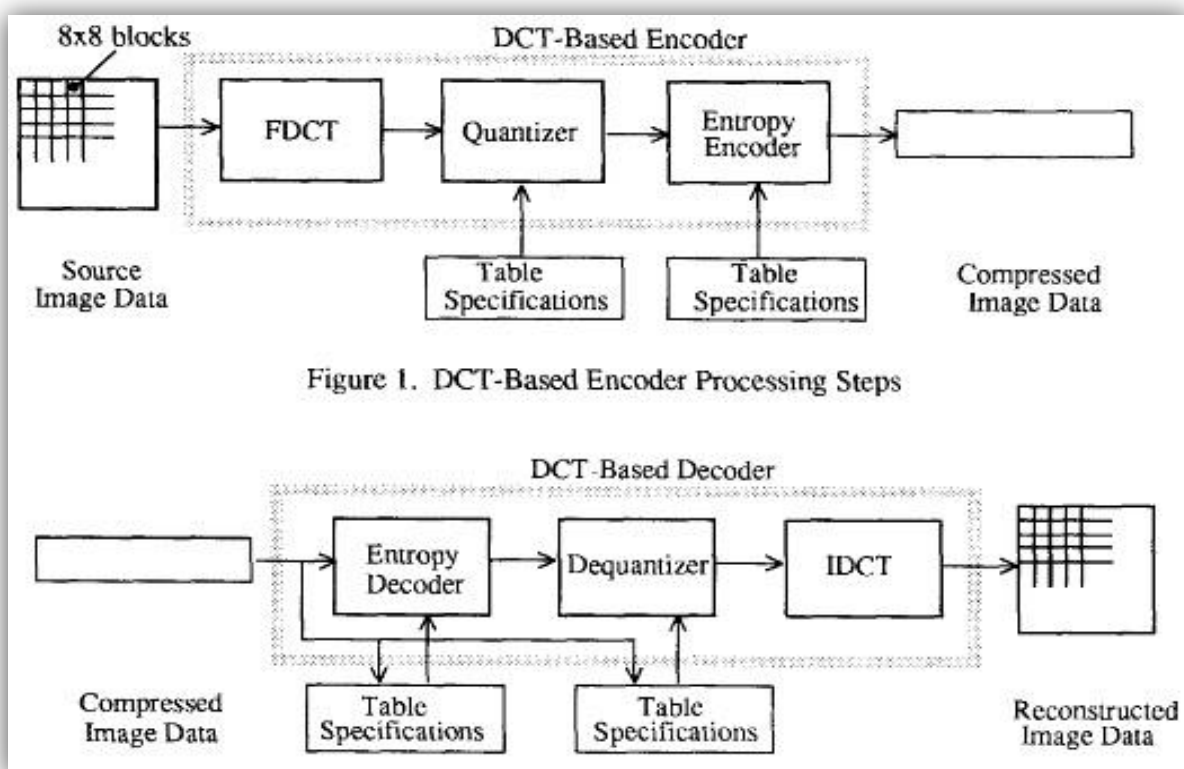


Fig 2.1 : DCT Based Decoder Processing Steps

The DCT transform:

- Break up image into 8x8 image blocks.
- Change the basis for representing the block image.
- Performing 2D DCT operation
- $F(u,v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \left(\cos \frac{(2x+1)u\pi}{16} * \cos \frac{(2y+1)v\pi}{16} \right) \right]$
- Quantization
- $FQ(u,v) = \text{Integer Round} \frac{F(u,v)}{Q(u,v)}$
- DC Coding and Zigzag sequencing
- Encoding is of two types: Arithmetic Coding and Huffman Coding

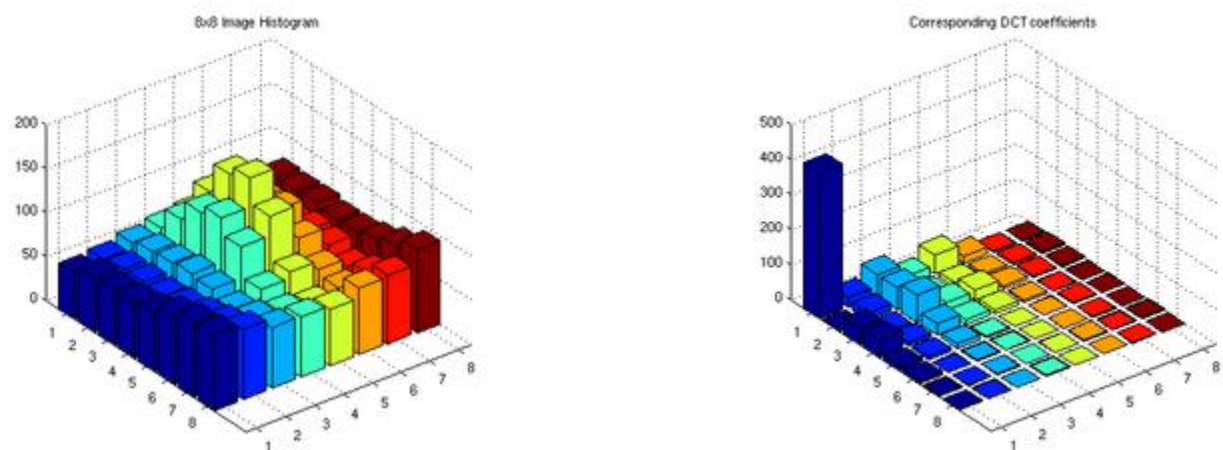
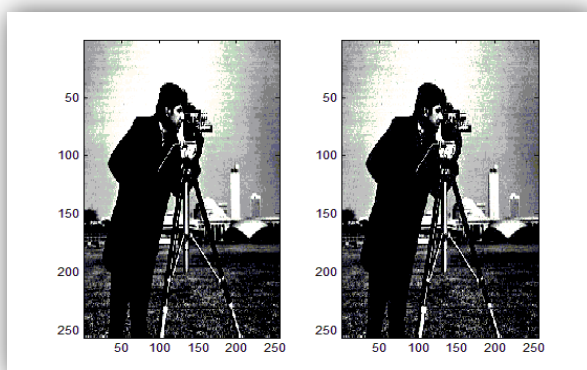
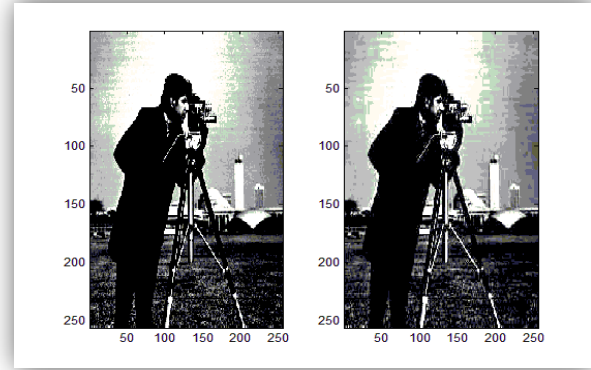


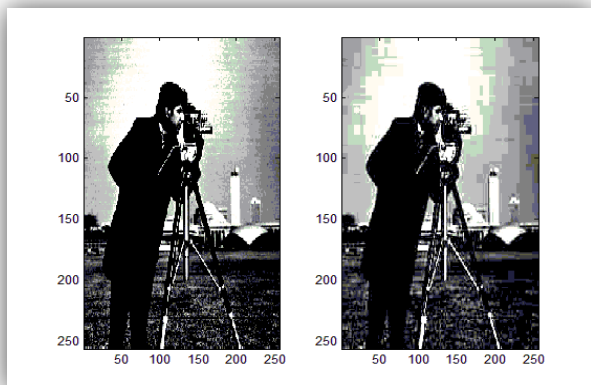
Fig 2.2 : Information packing in DCT coefficients



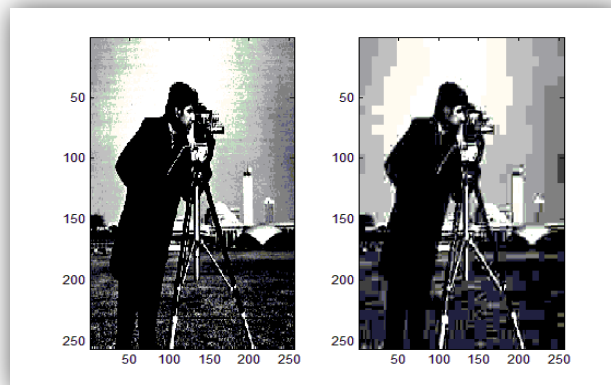
Q = 90



Q = 60



$Q = 30$



$Q = 10$

FIG : 2.3 : Image Compression and Decompression for different Q values

The table given below shows the size of the compressed and original images for various Quantisation Levels. The simulations were done in MATLAB and the outputs are shown below.

Quantization Level	Size of Compressed Image (in Bytes)	Compression Ratio
90	15878	2.72
80	9895	4.375
60	6214	6.96
30	3791	11.42
10	1853	23.36

Table 2.1 : Compression Ratio for Different Quantization Level

Original Image Size: 43299 bytes

2.2 Image Enhancement:

The second part includes deciding on which enhancements can be added to our frame. We shall try simulating various enhancements and then decide which ones shall be appropriate for our design.

Image enhancement simply means, transforming an image f into image g using T . Where T is the transformation. The values of pixels in images f and g are denoted by r and s , respectively. As said, the pixel values r and s are related by the expression,

$$s = T(r)$$

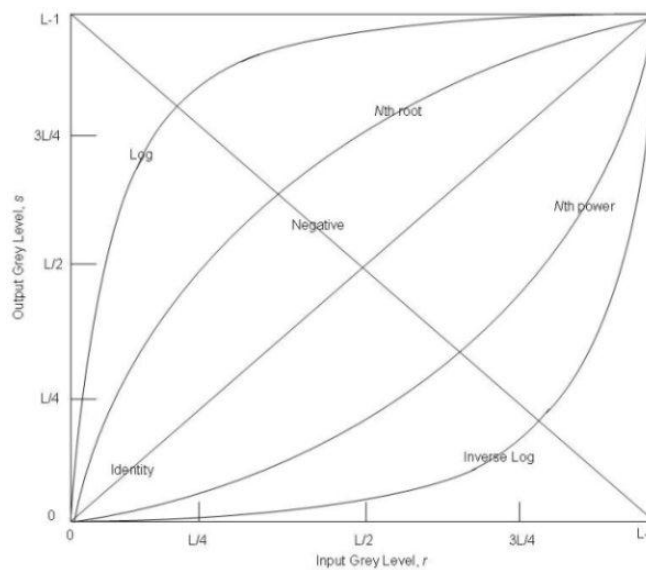


Fig 2.4 : Plot of various transformation functions

2.2.1 Image Negation:

The negative of an image with grey levels in the range $[0, L-1]$ is obtained by the negative transformation shown in figure above, which is given by the expression,

$$s = L - 1 - r$$

This expression results in reversing of the grey level intensities of the image thereby producing a negative like image. The output of this function can be directly mapped into the grey scale look-up table consisting values from 0 to

L-1.



Fig 2.5 : Negative Image Extraction

2.2.2 Logarithmic Transformation:

The log transformation curve shown in fig. A, is given by the expression,

$$s = c \log(1 + r)$$

Where c is a constant and it is assumed that $r \geq 0$. The shape of the log curve in fig. A tells that this transformation maps a narrow range of low-level grey scale intensities into a wider range of output values. And similarly maps the wide range of high-level grey scale intensities into a narrow range of high level output values. The opposite of this applies for inverse-log

transform. This transform is used to expand values of dark pixels and compress values of bright pixels.

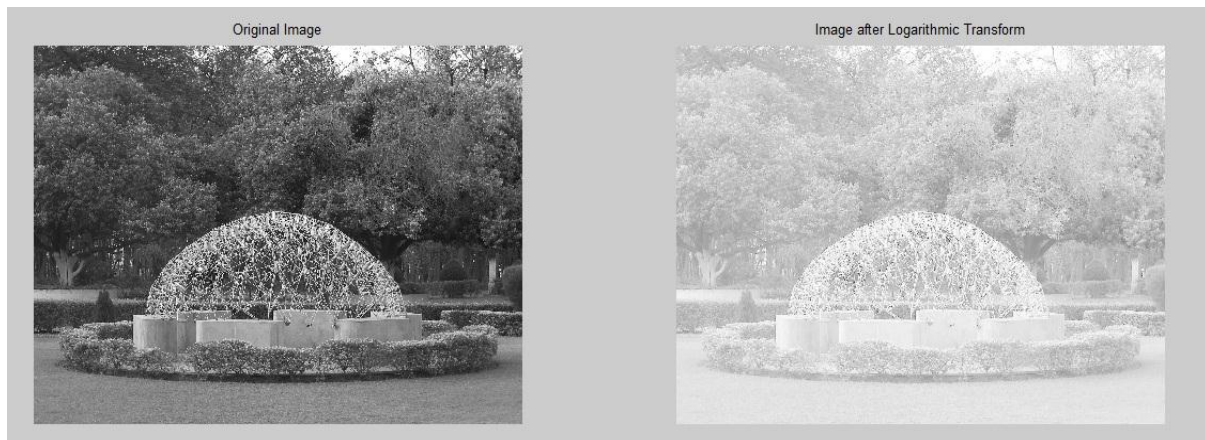


Fig 2.6 : Logarithmic Transformation

2.2.3 Gamma Correction:

The n th power and n th root curves shown in fig. A can be given by the expression,

$$s = cr^\gamma$$

This transformation function is also called as *gamma* correction. For various values of γ different levels of enhancements can be obtained. This technique is quite commonly called as *Gamma Correction*.

If you notice, different display monitors display images at different intensities and clarity. That means, every monitor has built-in gamma correction in it with certain gamma ranges and so a good monitor automatically corrects all the images displayed on it for the best contrast to give user the best experience.

The difference between the log-transformation function and the power-law functions is that using the power-law function a family of possible transformation curves can be obtained just by varying the λ .

These are the three basic image enhancement functions for grey scale images that can be applied easily for any type of image for better contrast and highlighting. Using the image negation formula given above, it is not necessary for the results to be mapped into the grey scale range $[0, L-1]$. Output of $L-1-r$ automatically falls in the range of $[0, L-1]$.

But for the Log and Power-Law transformations resulting values are often quite distinctive, depending upon control parameters like λ and logarithmic scales. So the results of these values should be mapped back to the grey scale range to get a meaningful output image.

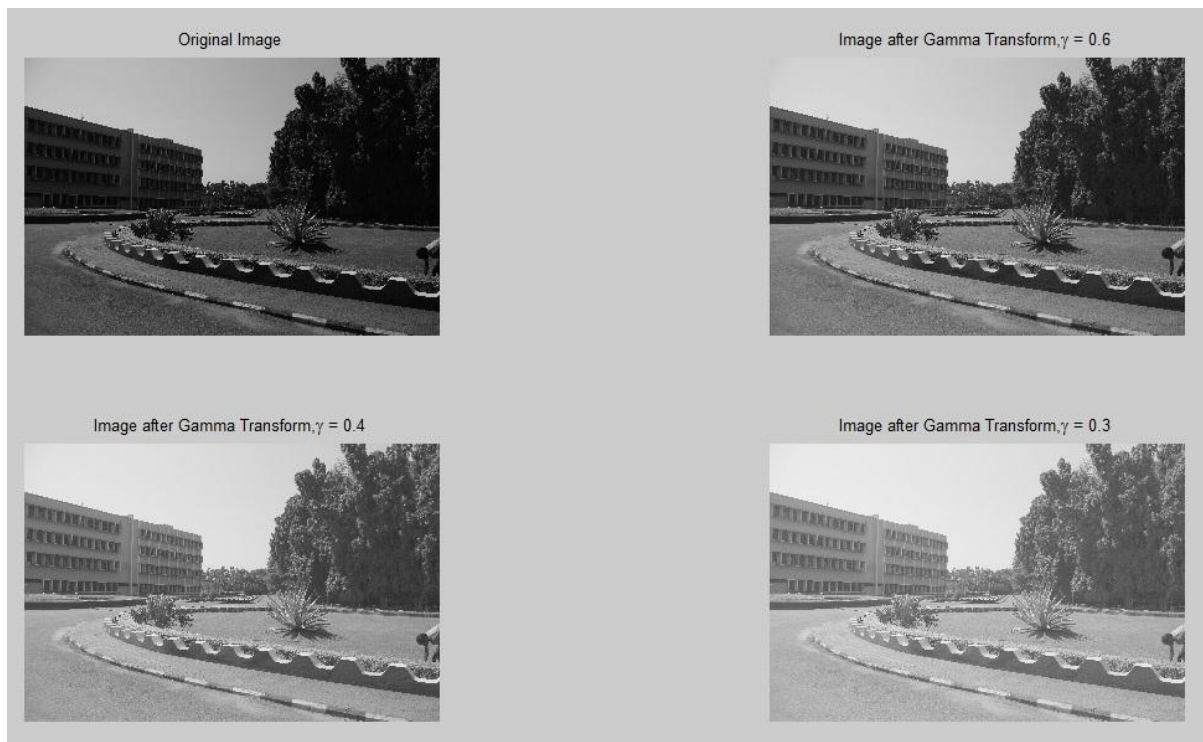


Fig 2.7 : Gamma Correction (1)



Fig 2.8 : Gamma Correction (2)

2.2.4 Piece Wise Linear Transformation:

2.2.4.1 Contrast Stretching:

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by 'stretching' the range of intensity values it contains to span a desired range of values, *e.g.* the full range of pixel values that the image type concerned allows.

It differs from the more sophisticated histogram equalization in that it can only apply a *linear* scaling function to the image pixel values. As a result the 'enhancement' is less harsh. (Most implementations accept a gray level image as input and produce another gray level image as output.) Before the stretching can be performed it is necessary to specify the

upper and lower pixel value limits over which the image is to be normalized. Often these limits will just be the minimum and maximum pixel values that the image type concerned allows.

For example for 8-bit gray level images the lower and upper limits might be 0 and 255. Call the lower and the upper limits a and b respectively. The simplest sort of normalization then scans the image to find the lowest and highest pixel values currently present in the image. Call these c and d .

Then each pixel P is scaled using the following function:

$$P_{out} = (P_{in} - c) \left(\frac{b - a}{d - c} \right) + a$$

Values below 0 are set to 0 and values about 255 are set to 255.

The problem with this is that a single outlying pixel with either a very high or very low value can severely affect the value of c or d and this could lead to very unrepresentative scaling.

Therefore a more robust approach is to first take a histogram of the image, and then select c and d at, say, the 5th and 95th percentile in the histogram (that is, 5% of the pixel in the histogram will have values lower than c , and 5% of the pixels will have values higher than d). This prevents outliers affecting the scaling so much. Another common technique for dealing with outliers is to use the intensity histogram to find the most popular intensity level in an image (*i.e.* the histogram peak) and then define a *cutoff fraction* which is the minimum fraction of this peak magnitude below which data will be ignored.

The intensity histogram is then scanned upward from 0 until the first intensity value with contents above the cutoff fraction. This defines c . Similarly, the intensity histogram is then

scanned downward from 255 until the first intensity value with contents above the cutoff fraction. This defines d . Some implementations also work with colour images. In this case all the channels will be stretched using the same offset and scaling in order to preserve the correct colour ratios.

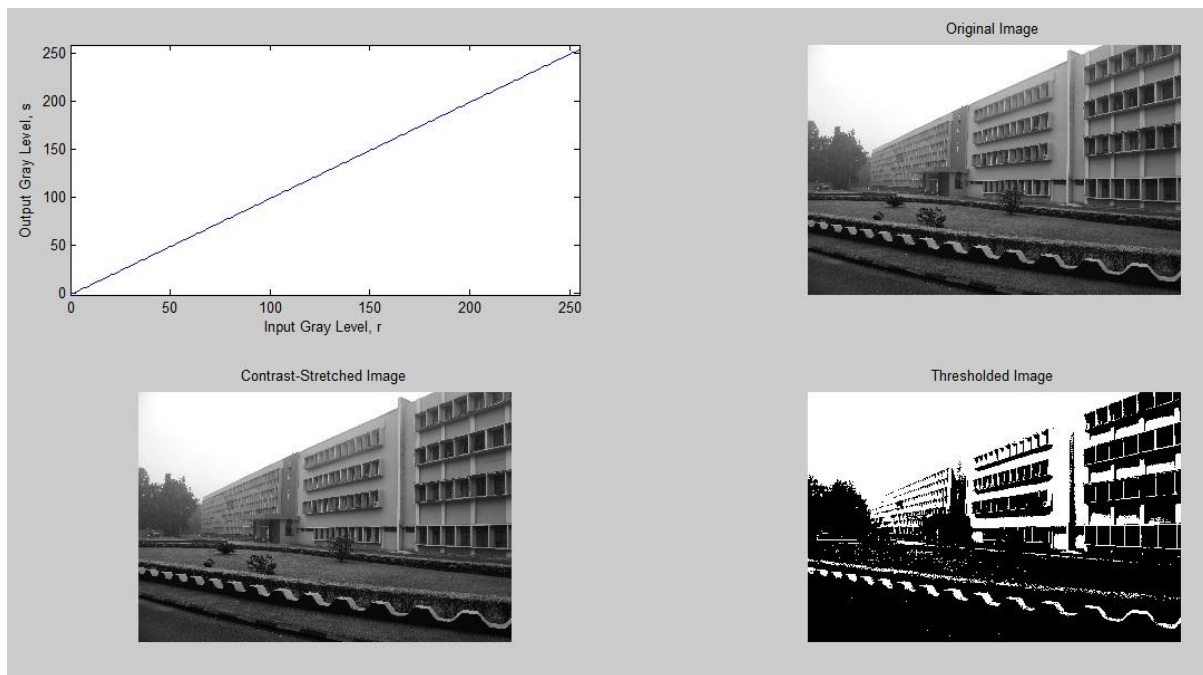


Fig 2.9 : Contrast Stretching

2.2.4.2 Grey level Slicing:

Highlighting a specific range of gray-levels in an image is often desired.

Applications include enhancing features such as masses of water, crop regions, or certain elevation area in satellite imagery. Another application is enhancing flaws in x-ray. There are two main different approaches:

- Highlight a range of intensities while diminishing all others to a constant low level.
- Highlight a range of intensities but preserve all others.

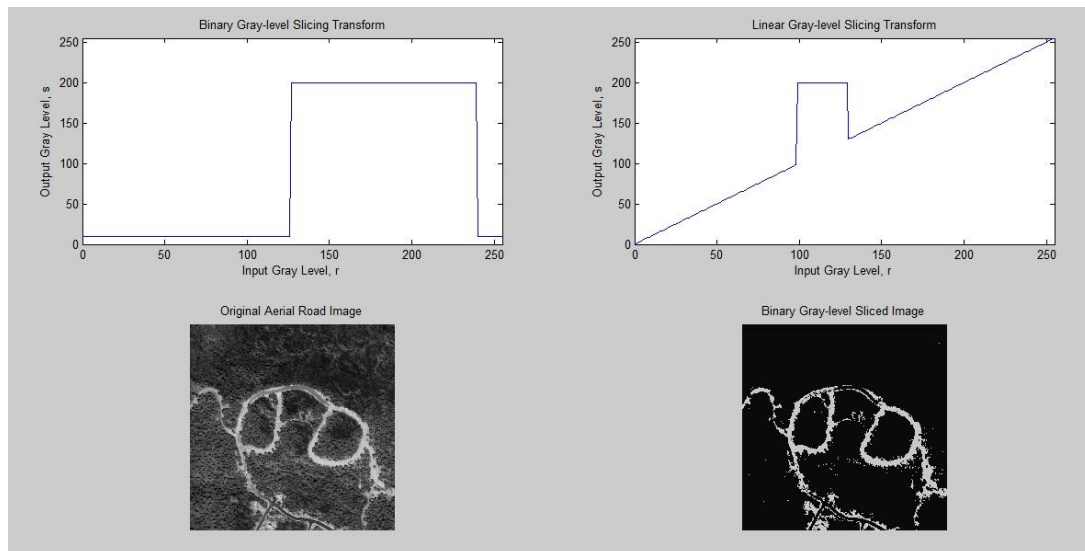


Fig 2.10 : Grey Level Slicing

2.2.4.3 Bit Plane Slicing:

Instead of highlighting intensity ranges, highlighting the contribution made to the total image appearance by specific bit might be desired. Imagine that the image is composed of eight 1-bit planes, ranging from plane 0 for least significant bit to plane 7 for the most significant bit. Bit-plane slicing reveals that only the five highest order bits contain visually significant data. Also, note that plane 7, corresponds exactly with an image threshold at gray-level 128.

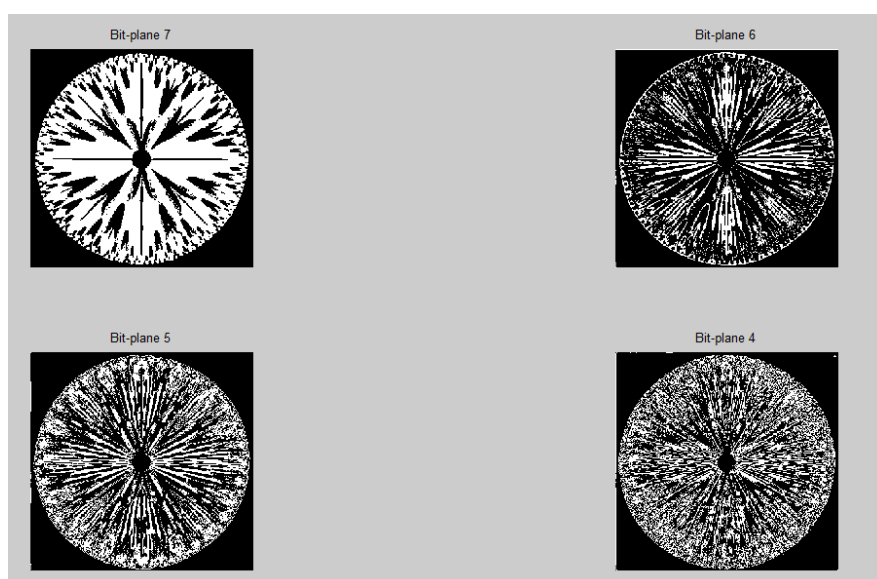


Fig 2.11 : Bit Plane Slicing (1)

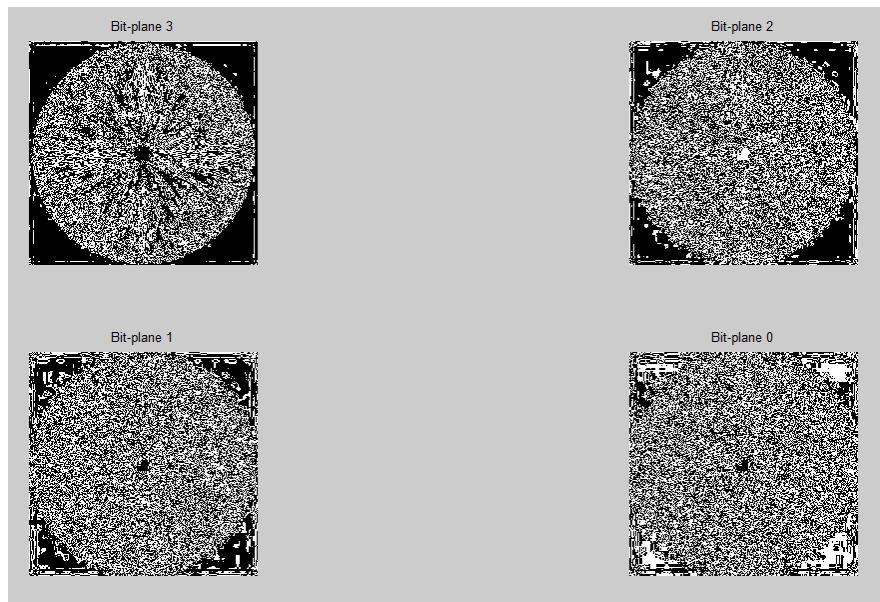


Fig 2.12 : Bit Plane Slicing (2)

2.2.5 Smooth Linear Filtering:

Here the emphasis is on:

- The definition of correlation and convolution,
- Using convolution to smooth an image and interpolate the result,
- Using convolution to compute (2D) image derivatives and gradients,
- Computing the magnitude and orientation of image gradients.

The output of a smoothing linear filter is simply the average of the pixels contained in the neighbourhood of the filter mask. These filters sometimes are called averaging filters. They are also called low pass filters.



Fig 2.13 : Smooth Linear Filtering (1)



Fig 2.14 : Smooth Linear Filtering (2)

2.2.6 Median Filtering:

In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise.

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. The pattern of neighbours is called the "window", which slides, entry by entry, over the entire signal. For 1D signal, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median, see median for more details.

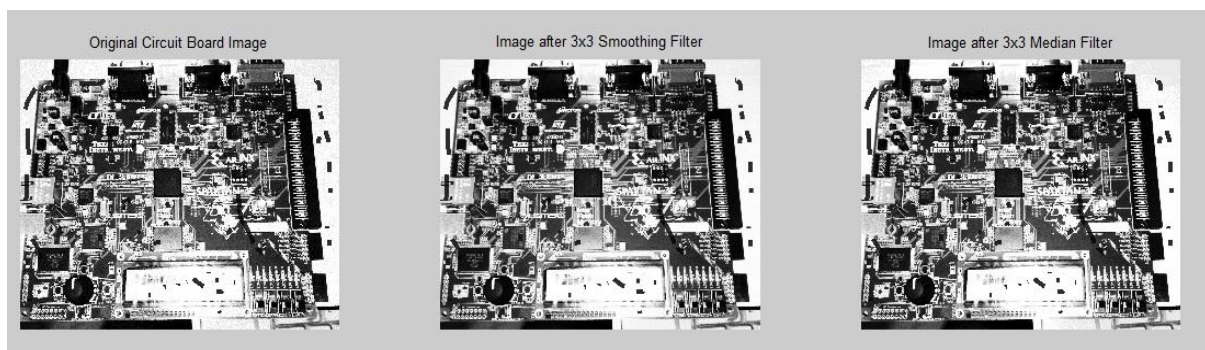


Fig 2.15 : Median Filtering

2.2.7 Laplace Filtering:

A Laplacian filter forms another basis for edge detection methods. A Laplacian filter can be used to compute the second derivatives of an image, which measure the rate at which the first derivatives change. This helps to determine if a change in adjacent pixel values is an edge or a continuous progression (see Detecting Edges for more information on edge detection).

Kernels of Laplacian filters usually contain negative values in a cross pattern (similar to a plus sign), which is centered within the array. The corners are either zero or positive values. The centre value can be either negative or positive.



Fig 2.16 : Laplace Filtering

Chapter 3

Hardware Implementation: Image Compression and Decompression

3.1 DISCRETE COSINE TRANSFORM (DCT)

3.1.1 INTRODUCTION:

The forward discrete cosine transform (DCT) converts 64 spatial samples which are arranged in 8*8 block of data to 64 samples in frequency domain where each sample is represented in frequency domain so that we can see the pixel values include low frequency DC coefficient and 63 AC coefficients.

One dimensional DCT:

The formula for forward 1-D DCT is given by

$$C(u) = \frac{s(u)}{2} * \sum_{x=0}^{N-1} s(x) \cos \frac{\pi(2x+1)u}{2*N} \quad (1)$$

$$S(u) = 1/\sqrt{2} \quad \text{for } u=0$$

$$=1 \quad \text{for } u>0$$

$C(u)$ is the DCT coefficient value and $s(x)$ is pixel value

For $u = 0, 1, 2, \dots, N-1$, and N represents the length of FDCT. In this case $N = 8$.

Similarly the inverse Transform can be calculated using

$$s(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \frac{\pi(2x+1)u}{2*N} \quad (2)$$

For $x = 0, 1, 2, \dots, N-1$.

$$\alpha(u) = 1/\sqrt{N} \quad \text{if } u = 0$$

$$\alpha(u) = \sqrt{(2/N)} \quad \text{if } u \neq 0.$$

3.1.2 Two Dimensional FDCT

The forward 2D-DCT formula is given by

$$C(u, v) = \alpha(u) \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right] \quad (3)$$

For $u, v = 0, 1, 2, \dots, M-1$.

N represents the length of DCT block. In this case $N = 8$.

Similarly the inverse Transform can be calculated by

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u) \alpha(v) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right] \quad (4)$$

For $x, y = 0, 1, 2, \dots, N-1$,

$f(x, y)$ gives the pixel value of an image.

$$\alpha(u) = 1/\sqrt{N} \quad \text{if } u = 0$$

$$\alpha(v) = \sqrt{(1/N)} \quad \text{if } v=0;$$

$$\alpha(u) = \sqrt{(2/N)} \quad \text{if } u \neq 0$$

$$\alpha(v) = \sqrt{(2/N)} \quad \text{if } v \neq 0$$

3.1.3 DCT Module

DCT is one of computational intensive algorithm and can be calculated by a large number of multiplications and additions. The use of multipliers is not advisable in hardware implementation as they require high power and large area when developed. Hence, Distributed Arithmetic (DA) approach is commonly used technique to avoid the use of

multiplications. In Distributed Arithmetic algorithm, it includes basic addition and shifting operation.

For 8-point 1D-DCT ,from equation (1)

$$F(u) = \frac{s(u)}{2} * \sum_{x=0}^{N-1} X(x) \cos \frac{\pi(2x+1)u}{2*N}$$

Where N=8.

So using periodical property, we can write

$$F(0)=[X(0) + X(1)+ X(2)+ X(3)+ X(4) + X(5) + X(6)+ X(7)]P$$

$$F(1)=[X(0)-X(7)]A +[X(1)-X(6)]B +[X(2)-X(5)]C +[X(3)-X(4)]D$$

$$F(3)=[X(0) - X(3)- X(4)+ X(7)]M +[X(1) + X(2) + X(5)+ X(6)]N$$

$$F(4)= [X(0) - X(1)- X(2)+ X(3)+ X(4) - X(5) - X(6)+ X(7)]P$$

$$F(5)=[X(0)-X(7)]C +[X(1)-X(6)](-A) +[X(2)-X(5)]D +[X(3)-X(4)]B$$

$$F(6)=[X(0) - X(3)- X(4)+ X(7)]N +[X(1) - X(2) - X(5)+ X(6)](-M)$$

$$F(7)=[X(0)-X(7)]D +[X(1)-X(6)](-C) +[X(2)-X(5)]B +[X(3)-X(4)](-A)$$

$$\text{Where } M= \frac{1}{2} \cos \frac{\pi}{8}, \quad A= \frac{1}{2} \cos \frac{\pi}{16},$$

$$N= \frac{1}{2} \cos \frac{3\pi}{8}, \quad B= \frac{1}{2} \cos \frac{3\pi}{16},$$

$$P= \frac{1}{2} \cos \frac{\pi}{4}, \quad C= \frac{1}{2} \cos \frac{\pi}{16}, D= \frac{1}{2} \cos \frac{7\pi}{16}$$

The constant cosine value can be written in binary format, for example F(1) can calculated using

$$\begin{aligned}
& \cdot \begin{bmatrix} \frac{1}{2} \cos(\pi/16) & \frac{1}{2} \cos(3\pi/16) & \frac{1}{2} \cos(5\pi/16) & \frac{1}{2} \cos(7\pi/16) \end{bmatrix} \cdot \begin{bmatrix} (X(0) - x(7)) \\ (X(1) - x(6)) \\ (X(2) - x(5)) \\ (X(3) - x(6)) \end{bmatrix} \\
& = \begin{bmatrix} -2^0 & 2^{-1} & \dots & 2^{-12} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} (X(0) - x(7)) \\ (X(1) - x(6)) \\ (X(2) - x(5)) \\ (X(3) - x(6)) \end{bmatrix}
\end{aligned}$$

F (1) can be represented as shown below,

$$F(1) = \begin{bmatrix} -2^0 & 2^{-1} & \dots & 2^{-12} \end{bmatrix} \begin{bmatrix} F^0(1) \\ F^1(1) \\ F^2(1) \\ F^3(1) \\ F^4(1) \\ F^5(1) \\ F^6(1) \\ F^7(1) \\ F^8(1) \\ F^9(1) \\ F^{10}(1) \\ F^{11}(1) \\ F^{12}(1) \end{bmatrix}$$

The powers of F denote the number of right shifting operation required, the 8 point DCT can be achieved by using the following structure in as shown in figure

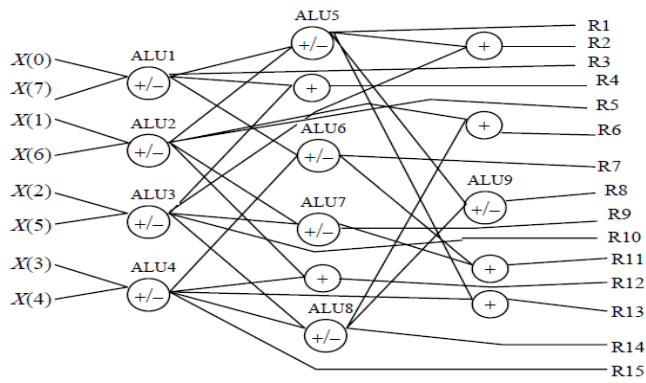


Fig 3.1: Adder and Subtractors Structure for 8-point DCT

The + and – signs in Table 3.1 indicates the function of ALU's to be performed , in Fig 3.1.Each DCT coefficient can be calculated by shifting R_i values by power of F and then summing each column.

	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)
ALU1	+	–	+	–	+	–	+	–
ALU2	+	–	+	–	+	–	+	–
ALU3	+	–	+	–	+	–	+	–
ALU4	+	–	+	–	+	–	+	–
ALU5	+	+	+	+	–	+	+	+
ALU6	+	+	–	+	+	+	–	+
ALU7	+	+	–	+	+	+	–	+
ALU8	+	+	+	+	–	+	+	+
ALU9	+	+	+	+	–	+	+	+
F^0	0	0	0	R6	0	R5	R9	R12
F^1	0	0	0	R6	0	R5	R9	R12
F^2	R8	R2	R7	R1	R8	R7	0	R10
F^3	0	R1	R11	R13	0	R15	R7	R1
F^4	R8	R7	R11	R15	R8	R10	R7	R1
F^5	R8	R13	0	R7	R8	R14	R9	R2
F^6	0	R4	R7	R5	0	R3	0	0
F^7	R8	R9	R7	R2	R8	R13	0	R14
F^8	0	R4	0	R5	0	R3	R9	0
F^9	R8	R7	R9	R14	R8	R9	R11	R13
F^{10}	0	R12	R11	R7	0	R14	R11	R2
F^{11}	0	R12	R7	R7	0	R14	R7	R2
F^{12}	0	R14	R7	R12	0	R4	R7	R1

Table 4.1

Function of each ALU to calculate the 8-point DCT

3.1.4 2D-DCT ARCHITECTURE

The 2D-DCT module can be designed using structural style of modelling in which two 1-D DCT modules as its basic components. It is implemented block wise (8 x 8 blocks). The

image pixel values are given as input to the 1-D module row wise and one row in 1 clock cycle and the calculated values are stored in registers. So, 8 clock cycles are required to complete row wise operation. After completion of row wise calculation, column wise 1-D DCT will be calculated which takes 8 more cycles. The final 2-D DCT takes 16 clock cycles (8 for row operation+8 for column operation).

3.1.5 RTL Schematic:

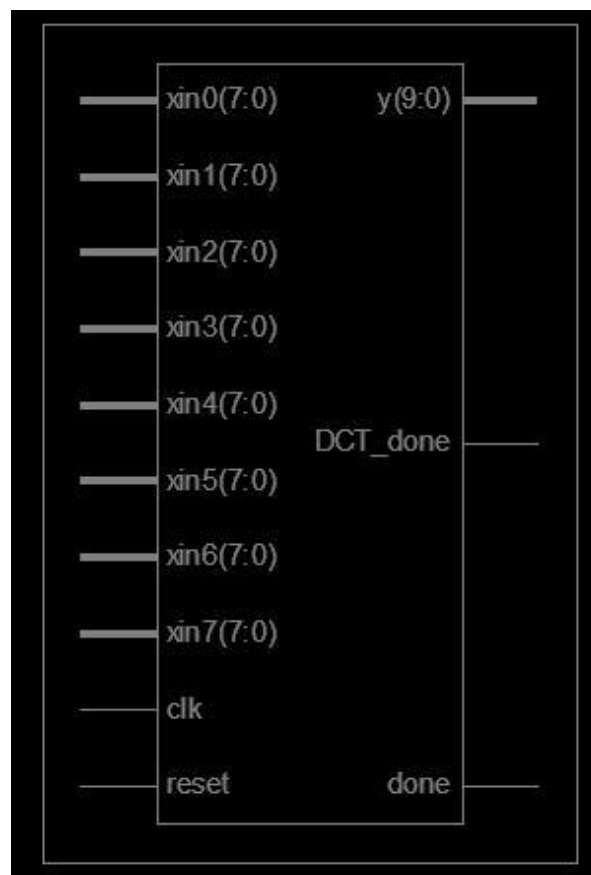


Fig 3.2: RTL Schematic

3.1.6 DCT and Quantization Output:

Current Simulation Time: 1000 ns		25 ns	50 ns	75 ns	100 ns	125 ns	150 ns	175 ns	200 ns	225 ns	250 ns	275 ns	300 ns	325 ns	350 ns	375 ns	400 ns	425 ns	450 ns	475 ns	500 ns	
clk	1																					
reset	0																					
xin0	-4	-95 108 107 104 2 -1 -3 -4																				
xin1	-7	-96 103 102 99 -3 -5 -7																				
xin2	-10	99 -2 96 -6 -8 -9 -10																				
xin3	-8	98 -3 96 -5 -6 -7 -8																				
xin4	-8	94 -7 -8																				
xin5	-19	100 80 -20 -19																				
xin6	-40	57 58 -41 60 -40																				
xin7	-57	54 39 -60 41 -58 42 -57																				
y[9:0]	-1	10hUUU X0X3X15X2X3X4X0X5X6X5X2X9X7X0X1X0X-1X-2X-1X-2X-3X0X1																				
done	0																					
state	9	2 3 4 5 6 7 8 9																				
dct_done	1																					
dct[1:64]	{...}	{10hUUU 10hUUU 10hUUU 10hUU... {... {... {... {... {10 3 15 -2 -3 -4 0 -5 -6 -5 -2 -9 -7 0 -1 0 -1 -1 -2 -1 -1 -2 -3 0 0 -1 0 0 -1 0 -2 -1 -1...																				
dct[9:0][1]	10	10hUUU X 10																				
dct[9:0][2]	3	10hUUU X 3																				
dct[9:0][3]	15	10hUUU X 15																				

Fig 3.3: DCT and Quantization Output

3.1.7 DCT and Quantization Design Summary:

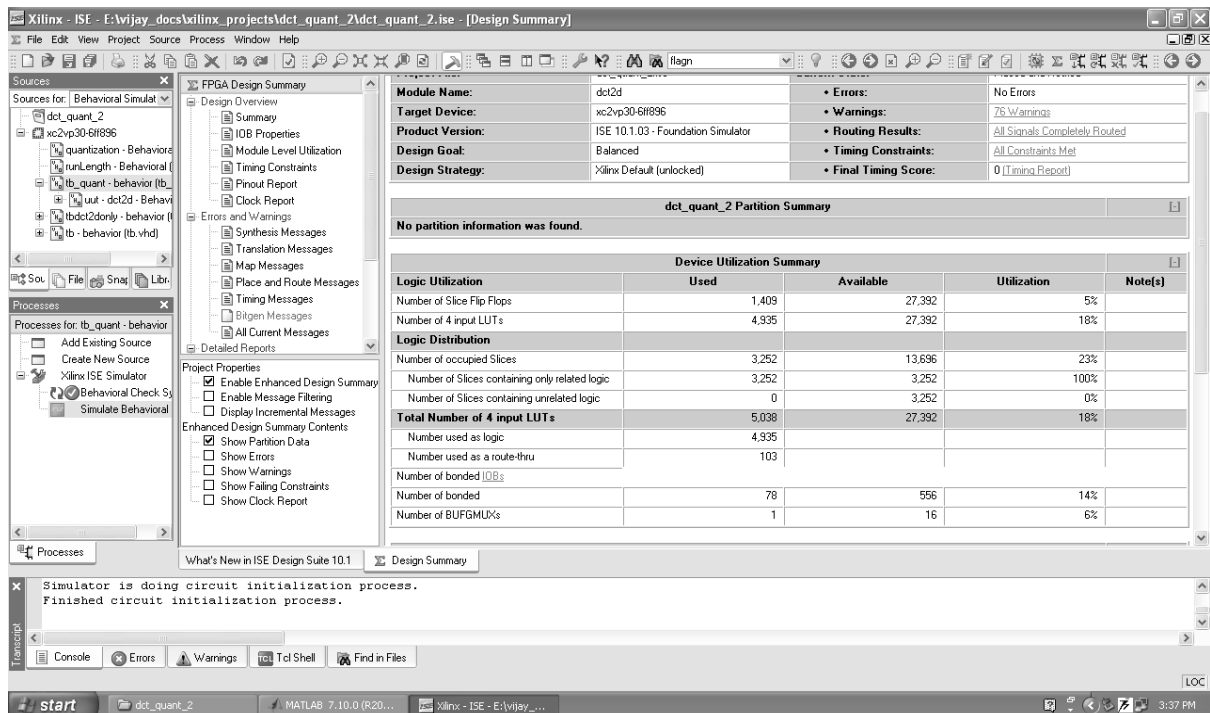


Fig 3.4: DCT and Quantization Design Summary

After completion of 2D-DCT calculation for particular block of input pixel values it stores the DCT coefficient values which are quantised using quantisation matrix and these values are stored in zigzag sequencing in which coefficients are stored in increasing in frequency component in it. Once this is completed the values are transformed in to the next module which gives the encoded values for DC coefficients and AC coefficients separately using Huffman encoding technique.

3.2 Normalized quantization matrix for hardware simplification

JPEG coding procedure has been described in previous section 2.6. 8x8 2D-DCT, quantization and Huffman coding are the three major steps followed in standard implementation. Data from 2D-DCT module output is quantized using a quantization matrix. Quantization is achieved by dividing each DCT coefficient value by a quantizer step size and followed by rounding to the nearest integer (Eq. 4.1).

where, $F(u,v)$ is the DCT coefficient output value and $Q(u,v)$ is the quantization matrix value (also called as normalization matrix). The standard matrix is given by JPEG to use implement image compression but user is free to use their own matrix to quantize the DCT coefficient values. If we use the standard quantization matrix we need 64 memory locations to store the different matrix values and we need a divider to divide the each coefficient.

So ,we can use the matrix shown below to quantizes the DCT output values with our need of any dividers by using only shifting operation in which results yield to give the same as standard method with good compression ratio.

$$Q_n = \begin{bmatrix} 16 & 16 & 16 & 16 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 16 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 32 & 32 & 64 & 64 & 64 \\ 16 & 16 & 32 & 32 & 32 & 64 & 64 & 64 \\ 32 & 32 & 32 & 64 & 128 & 128 & 128 & 128 \\ 64 & 64 & 64 & 64 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{bmatrix}$$

3.2.1 Reordering using Zigzag sequence matrix:

Zigzag sequence is used to store the 8*8 block of modified image values in a 1D array in which the values are given with increasing frequency component in it. So, we re-arrange the coefficients using the zigzag sequencing. The first pixel value in every will contain only DC value and remaining 63 coefficients contain AC frequencies.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Fig 3.5 : Zig Zag Order Sequence Matrix

3.2.2 Huffman Coding Architecture Implementation

Huffman coding is a variable length coding technique used in image compression to remove the data redundancy. It can be achieved by encoding the more frequently occurring data

For the hardware implementation of Huffman encoding , we use Huffman standard tables to make the hardware simple and high performing. JPEG uses the code table called Huffman encoding table. The architecture of Huffman coding is explained in the next section using JPEG base line image coding.

- ### 3.2.3 RTL Schematic:



3.2.4 Simulation Result:

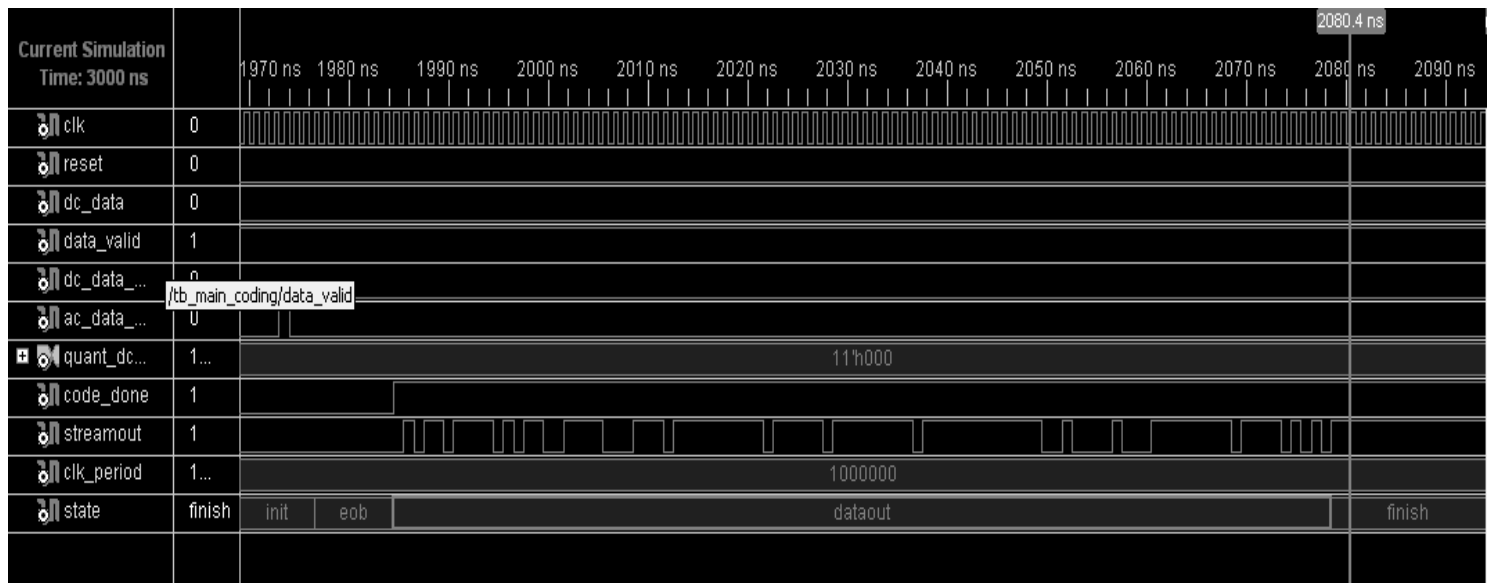


Fig 3.7 : Simulation Result:

3.2.5 Huffman Coding Results using MATLAB:

```

DC(11) 1    0    1    1    0    1    1
AC(25,59,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,...)

1    1    0    1    0    1    1    0    0    1    1    1    1    0
0    0    1    1    1    0    1    1    1    1    1    1    1    1
1    0    1    1    1    1    1    0    1    1    1    1    1    1
1    1    0    1    1    1    1    1    1    1    1    1    1    1
1    0    0    1    0    0    0    0    1    0    0    0    1    1

```

Fig 3.8 Huffman Coding Results using MATLAB

After completion of the Huffman coding we can have the final compressed data which can be used to store the image in memory.

When we want to retrieve back the image from compressed image data, we reverse the process in which the data is processed through decoding using Huffman decoding table for

standard AC and DC coefficients separately. And then each block is arranged in the correct order from the zigzag ordering and then multiply each pixel value with the quantization matrix so that the scaling can be redone to display the image. Once we get the de-quantized values we can send the values to 2D-Inverse DCT module which give the decompressed image data in 8*8 block size. By processing all the values we can get the final image data and we can display the image.

Chapter 4

Hardware Implementation: Enhancements

4.1 Median Filtering:

We obtained the following results after implementing Median Filtering.

4.1.1 RTL Schematic:

Shown below is the RTL Schematic that we were able to generate in Xilinx ISE.



Fig 4.1 RTL Schematic (1)

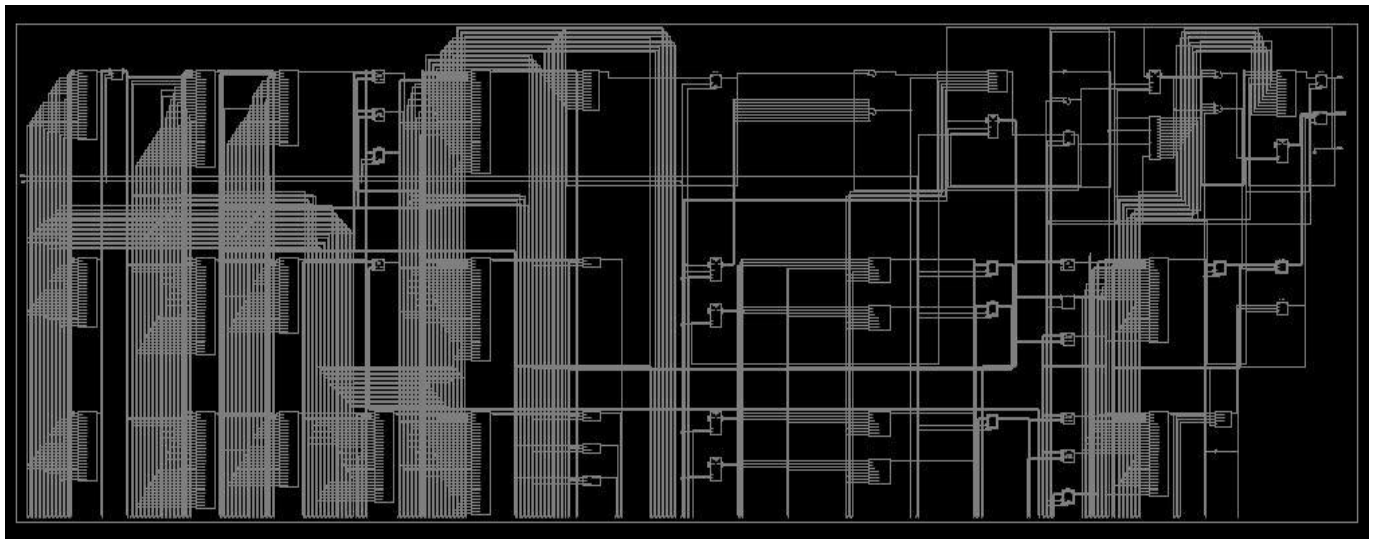


Fig 4.2 RTL Schematic (2)

4.1.2 Design Summary:

sort_hdl Project Status					
Project File:	sort_hdl.isc	Current State:	Synthesized		
Module Name:	sort	• Errors:	No Errors		
Target Device:	xc2vp30-6ff896	• Warnings:	13 Warnings		
Product Version:	ISE 10.1.03 - Foundation Simulator	• Routing Results:			
Design Goal:	Balanced	• Timing Constraints:			
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:			

sort_hdl Partition Summary			
No partition information was found.			

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slices	616	13696	4%	
Number of Slice Flip Flops	291	27392	1%	
Number of 4 input LUTs	1188	27392	4%	
Number of bonded IOBs	12	556	2%	
Number of BRAMs	64	136	47%	
Number of GCLKs	1	16	6%	

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Mon Apr 2 01:10:25 2012	0	13 Warnings	14 Infos
Translation Report					

Fig 4.3 Design Summary

4.1.3 Simulation Result:

























Current Simulation Time: 1000 ns																																																					
		15 ns		20 ns		25 ns		30 ns		35 ns		40 ns		45 ns		50 ns		55 ns		60 ns		65 ns		70 ns																													
  din[7:0]	131	0								162								163								162																											
  medianou...	131	0								162								163								162																											
  medcount...	109	0								1								2								3								4								5								6			
 reset	0																																																				
 state	9	3	4	5	6	7	8	9	10	11	12	7	8	9	10	11	12	7	8	9	10	11	12	7	8	9	10	11	12	7	8	9	10																				
 state2	2	0	1	2	0	1	2	3			0	1	2	3			0	1	2	3			0	1	2	3			0	1	2	3			0	1	2	3															
 romaddr	367	1	1	1	5	1	2	5	514			3	5	515			4	6	516			5	6	517			6	6	518			7	6	519			8	6	520														
  memstd[7:0]	129	162								163								166								162								155								160								157			
  reg[0:8]	{...	{...}																																																			
 median	131	0								162								163								162																											
 flagp	0																																																				
 flagn	0																																																				
 flag3	1																																																				
 sig_stop	0																																																				
 count255	111	1	2	3			4			5			6			7			8			9																															
 count3	0	1	2	0																																																	
 n1	112	1	2	3			4			5			6			7			8			9																															
 n2	368	257	258	259			260			261			262			263			264			265																															
 n3	623	513	514	515			516			517			518			519			520			521																															

Fig 4.4 Simulation Result

4.1.4 Hardware Output:



Original Image



Enhanced Image

Fig 4.5 Final Output

4.2 Laplace Filter:

We obtained the following results for Laplacian Filtering after implementation.

4.2.1 RLT Schematic:

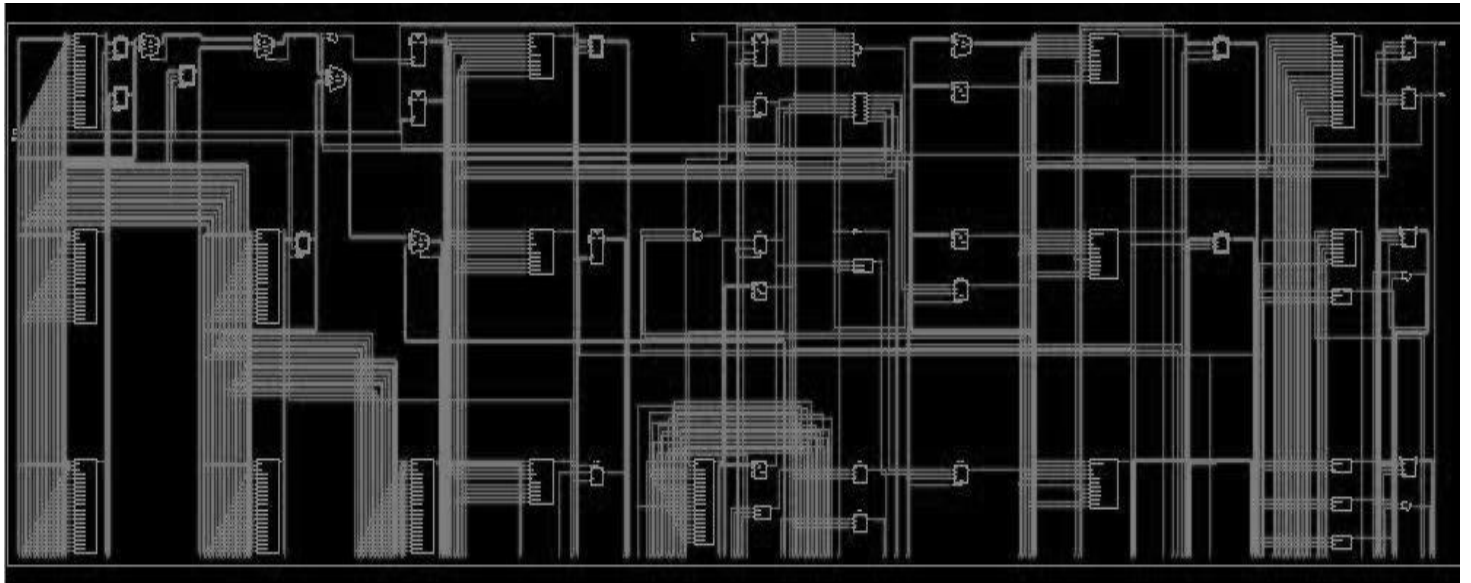


Fig 4.6 RTL Schematic

4.2.2 Design Summary:

edge_vhdl Project Status (05/03/2012 - 01:36:07)			
Project File:	edge_vhdl.isc	Current State:	Synthesized
Module Name:	edge	• Errors:	No Errors
Target Device:	xc2vp30-6ff896	• Warnings:	59 Warnings
Product Version:	ISE 10.1.03 - Foundation Simulator	• Routing Results:	
Design Goal:	Balanced	• Timing Constraints:	
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:	

edge_vhdl Partition Summary		[+]
No partition information was found.		

Device Utilization Summary (estimated values)				[+]
Logic Utilization	Used		Available	Utilization
Number of Slices	490		13696	3%
Number of Slice Flip Flops	366		27392	1%
Number of 4 input LUTs	903		27392	3%
Number of bonded IOBs	25		556	4%
Number of BRAMs	108		136	79%
Number of MULT18x18s	3		136	2%
Number of GCLKs	1		16	6%

Fig 4.7 : Design Summary

4.2.3 Simulation Result:

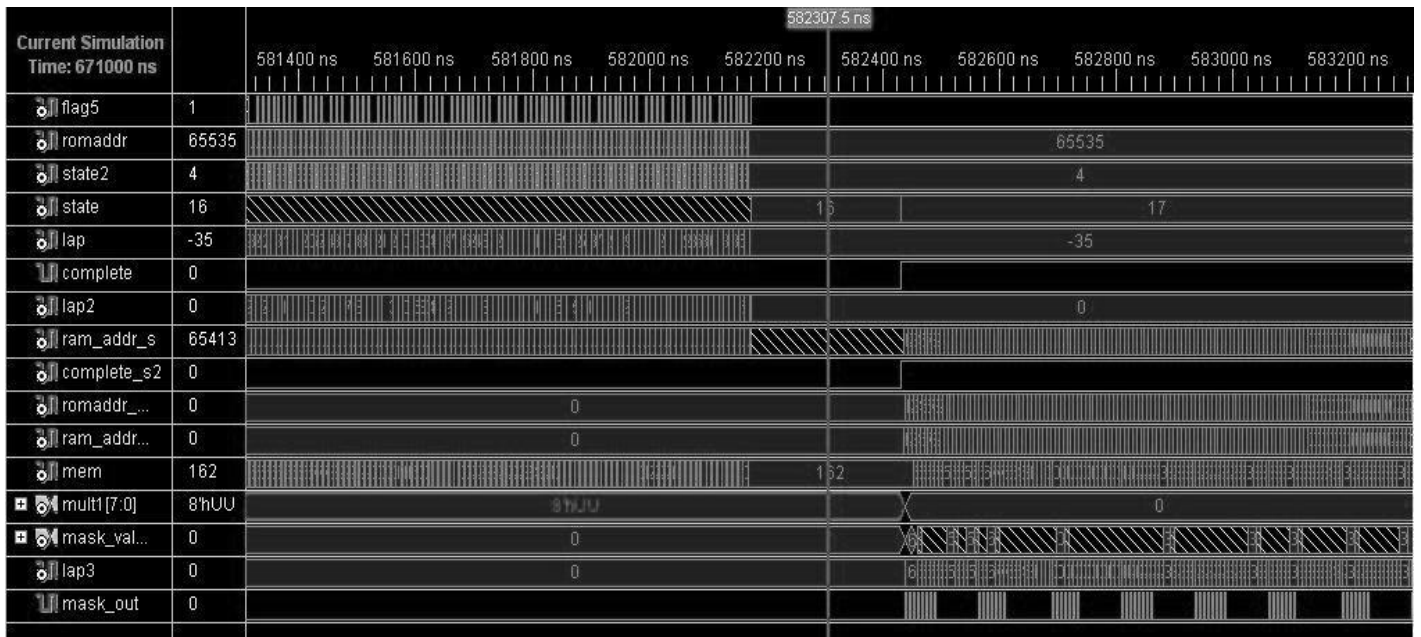


Fig 4.8 : Simulation Result

4.2.4 Hardware Output:



Original



Laplacian Mask



Output Image

Fig 4.9 : Hardware Output

Chapter 5

Conclusion and Future Work

Thus in this way we were successfully able to complete our objective. We were able to decide precisely on which features we wanted to add onto our photo frame. We were able to prove their feasibility using MATLAB. We were also able to successfully implement them in Hardware using VHDL. We were able to study the design summary of our designs and were thus able to get an idea regarding the hardware requirements and complexity.

We are currently working on developing an Soft IP CORE of the image compression and decompression method as well as the enhancements. We are planning to patent this and then provide it as a library for vendors who manufacture Digital Photo Frames. They can add the entire library or part of it, in their design, depending upon their requirement.

References:

- Rui Xi Chen, Hong Zhi Zhang, Wei Ming Yeh, Wei Min Yang, “NIOS Embedded Electronic Photo Album”, Wei Min Yang of Electrical Engineering Institute, St. John’s University (2006)
- Wallace, G.K. , "The JPEG still picture compression standard," Consumer Electronics, IEEE Transactions on , vol.38, no.1, pp.xviii-xxxiv, Feb 1992
- “The USC – SIPI Image Database”, Signal and Image Processing Institute, Ming Hsieh Department of Electrical Engineering
- Rafael C Gonzalez and Richard E Woods, “Digital Image Processing”.
- Gavin L Bates and Saeid Nooshabadi , “FPGA Implementation of a Median Filter”, School of Electrical Engineering , Northern Territory University.
- Pei-Yin Chen; Chih-Yuan Lien; Hsu-Ming Chuang; , "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.18, no.3, pp.473-481, March 2010
- Moshnyaga, V.G.; Hashimoto, K.; , "An efficient implementation of 1-D median filter," Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on , vol., no., pp.451-454, 2-5 Aug. 2009
- Chih-Yuan Lien; Pei-Yin Chen; Li-Yuan Chang; Yi-Ming Lin; Po-Kai Chang; , "An efficient denoising chip for the removal of impulse noise," Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on , vol., no., pp.1169-1172, May 30 2010-June 2 2010

- Peng Chungan; Cao Xixin; Yu Dunshan; Zhang Xing; , "A 250MHz optimized distributed architecture of 2D 8x8 DCT," ASIC, 2007. ASICON '07. 7th International Conference on , vol., no., pp.189-192, 22-25 Oct.
- Anup Sarma, Soubhagya Sutar, V.K Sharma, K.K Mahapatra, "An ASIP for Image Enhancement Applications in Spatial Domain using LISA", ECE Dept, NIT Rourkela, 2011
- Rohit Kumar Jain, "Design and FPGA Implementation of CORDIC-based 8-point 1D DCT Processor", ECE Department, NIT Rourkela, 2011
- Shan-Jang Chen, Chang Wei Lin, Pao-Chyuan Chen, "Digital Photo Frame", Pub No.: US 2005/0057578 A1
- Xiao-Guang Li, Kuan-Hong Hsieh, Hua-Dong Cheng, "Digital Photo Frame", Pub No.: US 2008/0030478 A1
- Yu-Tsung Chang, Miao-Lih Hsuan, "Digital Photo Frame Apparatus", Pub No.: US 2008/0273126 A1
- Shin-Hong Chung, Tu-Cheng Xiao-Guang Li, Han-Che Wang, Tu Cheng, Fen Li, Kuan-Hong Hsieh, Tu-Cheng, Feng Zhu, "Intelligent Digital Photo Frame", Pub No.: US 2010/0013810 A1
- Syed Ali Khayam, "The Discrete Cosine Transform (DCT): Theory and Application", ECE 802 – 602: Information Theory and Coding, 2003
- Luciano Volcan, Agostini, Ivan Saraiva Silva and Sergio Bampi, "Multiplierless and fully pipelined JPEG compression soft IP targeting FPGAs," *Microprocessors and Microsystems*, vol. 31(8), Dec. 2007, pp.487-497.
- S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine*, vol.6, no.3, Jul.1989, pp.4-19.
- Sung-Hsien Sun and Shie-Jue Lee, "A JPEG Chip for Image Compression and Decompression," *Journal of VLSI Signal Processing*, Vol.35(1), pp.43–60, 2003.

