

# FPGA Implementation of 2D and 3D Image Enhancement Chip in HDL Environment

Priyanka Saini

M.Tech Scholar, Department of  
Electronics, Institute of  
Engineering and Technology,  
Alwar Rajasthan India

Adesh Kumar

Assistant Professor,  
Department of EEI, University  
of Petroleum and Energy  
Studies, Dehradun India

Neha Singh

Associate Professor,  
Department of Electronics,  
Institute of Engineering and  
Technology, Alwar India

## ABSTRACT

Digital image processing is an ever expanding and dynamic area with applications reaching out into everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image enhancement and object detection. Implementing such applications on a general purpose computer can be easier, but not very time efficient due to additional constraints on memory and other peripheral devices. Hardware implementation for application specific can offer much greater speed than a software implementation. With the advancement in the Very Large Scale of Integration (VLSI) technology hardware implementation has become an attractive alternative. Implementing complex computation tasks on hardware and by exploiting parallelism and pipelining in algorithms yield significant reduction in execution times. It has been observed that most of the work has been done either in C and JAVA. No work has been done in the VHDL to design and develop the chip for image enhancement algorithms. Intensity transformation is used to enhance the size of image pixels. The chip design for 2D and 3D image enhancement is done in Xilinx 14.2 software. Image enhanced values are verified with the help of waveform editor of Modelism software, Modelsim SE 10.1b.

## General Terms

Integrated circuits, Image enhancement, Functional simulation

## Keywords

Very High Speed Integrated Circuit hardware Description language (VHDL), Very Large Scale of Integration (VLSI), Field Programmable Gate Array (FPGA)

## 1. INTRODUCTION

Digital image processing is electronic data processing on a 2D array of numbers. An image is represented with 2D array. Image processing techniques are used to modify pictures and improve the quality with the different techniques such as enhancement, restoration, extract information analysis, recognition, and changing their structure composition[7][8], image editing. Images can be processed by photographic optical and electronic means, but image processing using digital computers is the most common method because digital methods are precise, fast and flexible. An image can be synthesized [8] from a micrograph of various cell organelles by assigning a light intensity value to each cell organelle. The sensor signal is “digitized” converted to an array of numerical values, each value representing the light intensity of a small

area of the cell. The digitized values are called picture elements, or “pixels,” and are stored in computer memory as a digital image. A typical size for a digital image is an array of 512 by 512 pixels, where each pixel has value in the range of 0 to 255. The digital image is processed by a computer to achieve the desired result. Image enhancement improves the quality of images for human viewing. Removing blurring and noise, increasing contrast, and revealing details are examples of enhancement operations [2][13]. For example, an image might be taken of an endothelial cell, which might be of low contrast and somewhat blurred. Reducing the noise and blurring and increasing the contrast range could enhance the image. The original image might have areas of very high and very low intensity, which mask details. An image enhancement uses adaptive algorithms adjust their operation based on the image information pixels being processed. In this case the mean intensity, contrast, and sharpness could be adjusted based on the pixel intensity statistics in various areas of the image. Image processing technology is used by planetary scientists to enhance images of Mars, Venus, or other planets. Doctors use this technology to manipulate CAT scans and MRI images [8] [2]. A graphical display can be improved with the image features enhancement such as boundaries, or contrast. It includes contrast level & gray manipulation, edge crispening, noise reduction, and sharpening, filtering, pseudo coloring, interpolation, magnification and so on. Image enhancement deals with the improvement of visual appearance of the scene, to improve the detect ability of objects to be used by either a machine vision system or a human observer. Integrated image capture, processing, and communication power on a compact, portable, hand-held device is attracting increased interest from computer vision researchers with a goal of applying a diverse collection of vision tasks on the small hand-held device. The research emphasis on computer vision to pattern enhancement technologies and the chip implementations that can be used in camera phones, mobiles, scanners etc.

## 2. IMAGE ENHANCEMENT METHODS

The 2D image  $I$  is ordinarily represented in software accessible form as an  $M \times N$  matrix containing indexed elements  $I(i, j)$ , where  $0 \leq i \leq M - 1, 0 \leq j \leq N - 1$ . The elements  $I(i, j)$  represent samples of the image intensities, usually called pixels. For simplicity, assume that it come from a finite integer valued range. Image enhancement techniques can be divided into two broad categories

*Spatial domain methods*

Which operate directly on pixels, and

*Frequency domain methods*

Which operate on the Fourier Transform of an image.

## 2.1 Spatial Domain Methods

The value of a pixel with coordinates  $(x, y)$  in the enhanced image  $\hat{F}$  is the result of performing some operation on the pixels in the neighborhood of  $(x, y)$  in the input image,  $F$ . Neighborhoods can be of any shape, but usual form is rectangular.

### 2.1.1 Grey scale manipulation

In gray scale manipulation the operator acts on a  $1 \times 1$  pixel neighborhood in the input image, that is  $\hat{F}(x, y)$  only depends on the value of  $F$  at  $(x, y)$ . This is a grey scale transformation or mapping. The simplest method is thresholding where the intensity profile is replaced by a step function, active at a chosen threshold value. In this case any pixel with a grey level is the threshold in the input image gets mapped to 0 in the output image. Other pixels are mapped to 255. Different types of grey scale transformations are shown in figure 1.

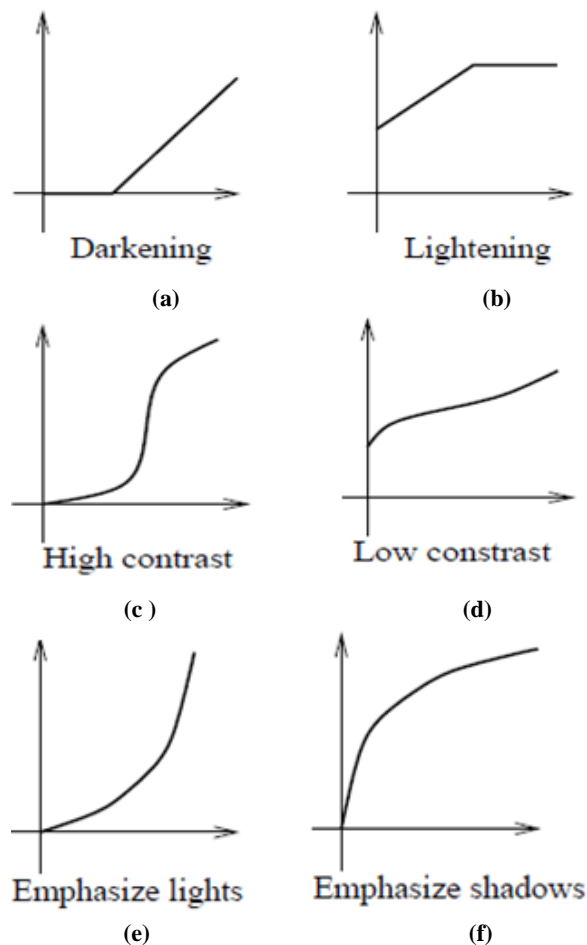


Fig 1: Tone-scale adjustments [13]

### 2.1.2 Histogram Equalization

Histogram equalization is a common technique for enhancing the appearance of images. Consider an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail is compressed into the dark end of the histogram. It is possible to stretch out the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer. Histogram equalization involves finding a grey scale transformation function that creates an output image with a uniform histogram.

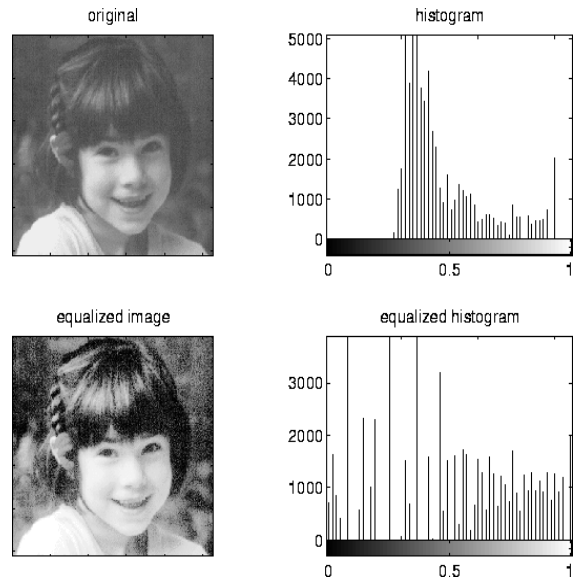


Fig 2: The original image and its histogram, and the equalized versions. Both images are quantized to 64 grey levels [13]

## 2.2 Frequency domain methods

Frequency domain is straightforward method for image enhancement. Fourier Transform is used to compute the Fourier values of the image to be enhanced, after that multiplication is done with the result by a filter instead of convoluting in the spatial domain, and take the inverse transform to produce the enhanced image. Blurring of an image can be done by reducing its high frequency components or sharpening an image by increasing the magnitude of its high frequency components. However, computationally, it is often more efficient to implement these operations as convolutions by small spatial filters in the spatial domain. Understand the frequency domain concepts are very important, leads to enhancement techniques that might not have been thought of by restricting attention to the spatial domain.

### 2.2.1 Filtering

Low pass filtering is the method which is used to eliminate the high frequency components in the image. Blurring of the image is the resultant of filtering and thus a reduction in sharp transitions associated with noise. An ideal low pass filter is the solution to retain all the low frequency components, and eliminate all the high frequency components. However, ringing and blurring are two problems associated with ideal filters. These problems are due to the shape of the associated spatial domain filter, which has a large number of undulations. Butterworth filter achieve much better results because of smoother transitions in the frequency domain filter.

A ideal low pass filter transmits, without any distortion, all the signals of frequencies below a certain frequency  $\omega_c$  radians per second.  $\omega_c$  is the cut off frequency. The transfer function of an ideal Low Pass Filter(LPF) is given by

$$|H(\omega)| = \begin{cases} 1 & \text{for } |\omega| < \omega_c \\ 0 & \text{for } |\omega| > \omega_c \end{cases}$$

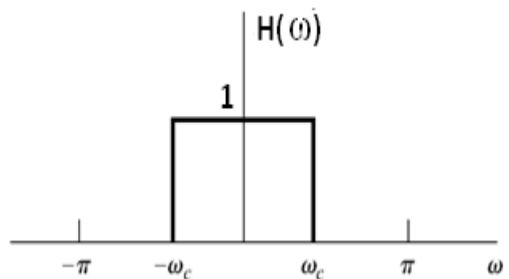


Fig 3: Ideal Low pass filter

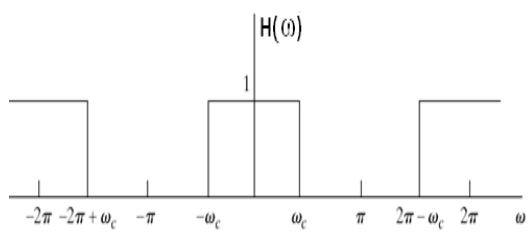


Fig 4: Low pass Filter

Impulse response can be determined by inverse Fourier transform formula

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(\omega) e^{j\omega t} d\omega$$

$$h(t) = \frac{1}{2\pi} \int_{-\omega_c}^{+\omega_c} (1) e^{j\omega t} d\omega$$

$$h(t) = \frac{1}{2\pi} \left[ \frac{e^{j\omega t}}{jt} \right]_{-\omega_c}^{+\omega_c}$$

$$h(t) = \frac{1}{\pi t} \left[ \frac{e^{j\omega_c t} - e^{-j\omega_c t}}{2j} \right]$$

$$h(t) = \frac{\sin \omega_c t}{\pi t}$$

$$h(t) = \frac{\omega_c}{\pi} \text{Sinc } \omega_c t$$

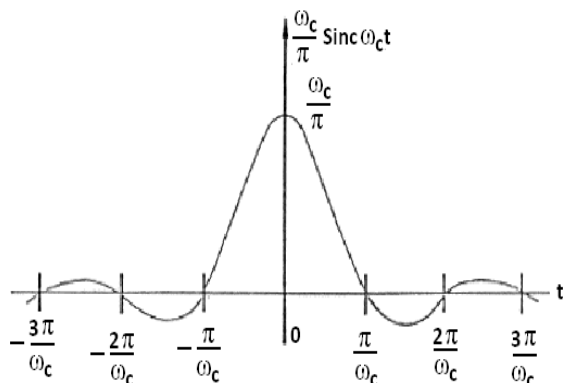


Fig 5: Impulse response of ideal low pass filter

From the impulse response it is clear that it does not have desirable characteristics. Ideal low pass filter does have perfect frequency selectivity and  $h(t)$  is not zero for  $t < 0$ .

### 2.2.2 Homomorphic Filtering

Images normally consist of light reflected from objects. The basic nature of the image  $F(x, y)$  is characterized by two components: (1) the amount of source light incident on the scene being viewed, called the illumination and (2) the

amount of light reflected by the objects in the scene called reflectance. These portions of light are called the illumination and reflectance components, and are denoted  $i(x, y)$  and  $r(x, y)$  respectively. The functions  $i$  and  $r$  combine multiplicatively to give the image function  $F$  which is given by

$$F(x,y) = i(x,y)r(x,y)$$

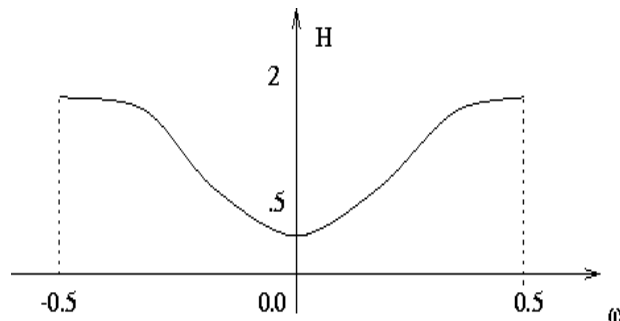


Fig 6: Transfer function for homomorphic filtering

Thus, the process of homomorphic filtering can be summarized by the following diagram

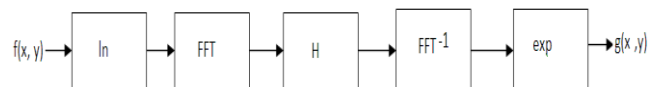


Fig 7: Homomorphic filtering process [13]

## 2.3 Image Enhancement Techniques

Image Enhancement Techniques are listed below

*Contrast Enhancement Method*

*Smoothing of an Image.*

*Brightness*

*Threshold*

*Intensity Transformation*

### 2.3.1 Contrast Enhancement Method

An Image is taken and its contrast is increased or decreased as per the enhancement requirements of the Image. The required contrast enhancement is achieved applying the Histogram Stretching of the Image.

### 2.3.2 Smoothing

A noisy Image is taken and the noise removal is done by applying a smoothing technique. The noise removal is achieved by using a mask which enables neighborhood pixel processing. The aim of image smoothing is to diminish the effects of camera noise, spurious pixel values, missing pixel values etc. There are many different techniques for image smoothing, but neighborhood averaging and edge-preserving smoothing are suiting to our application.

Neighborhood Averaging [13], is an approach in which each point in the smoothed image,  $\hat{F}(x,y)$  is obtained from the average pixel value in a neighborhood of  $(x,y)$  in the input image. For example, use of a  $3 \times 3$  neighborhood around each pixel masking is done.

$$1/16 \quad 1/16 \quad 1/16$$

$$1/16 \quad 1/16 \quad 1/16$$

$$1/16 \quad 1/16 \quad 1/16$$

Each pixel value is multiplied by  $1/16$ , summed, and then the result placed in the output image. This mask is successively

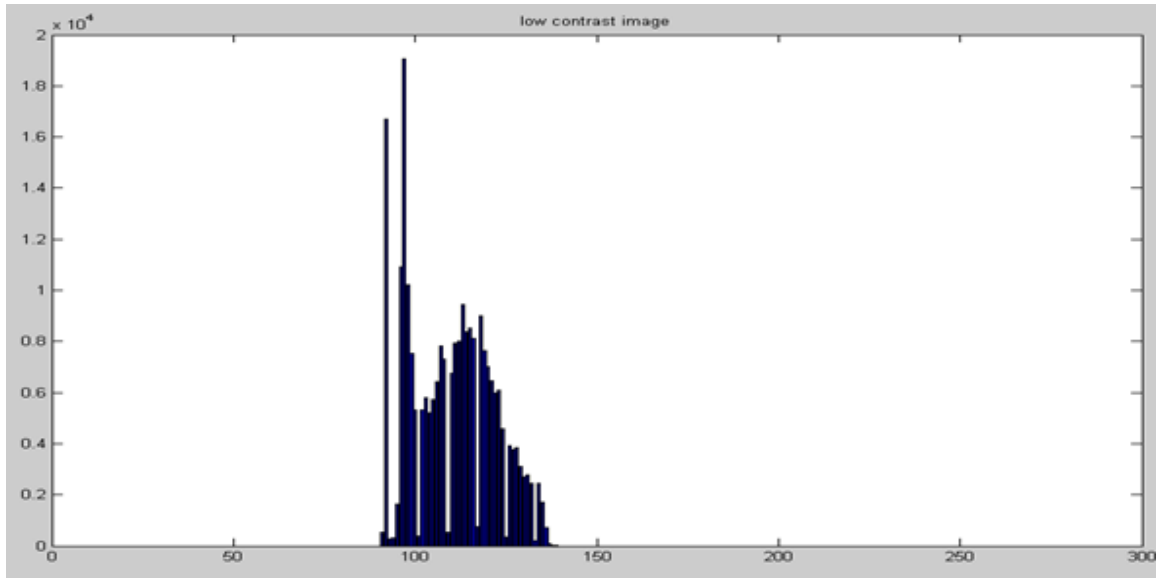


Fig. 8: after Contrast Enhancement [14]

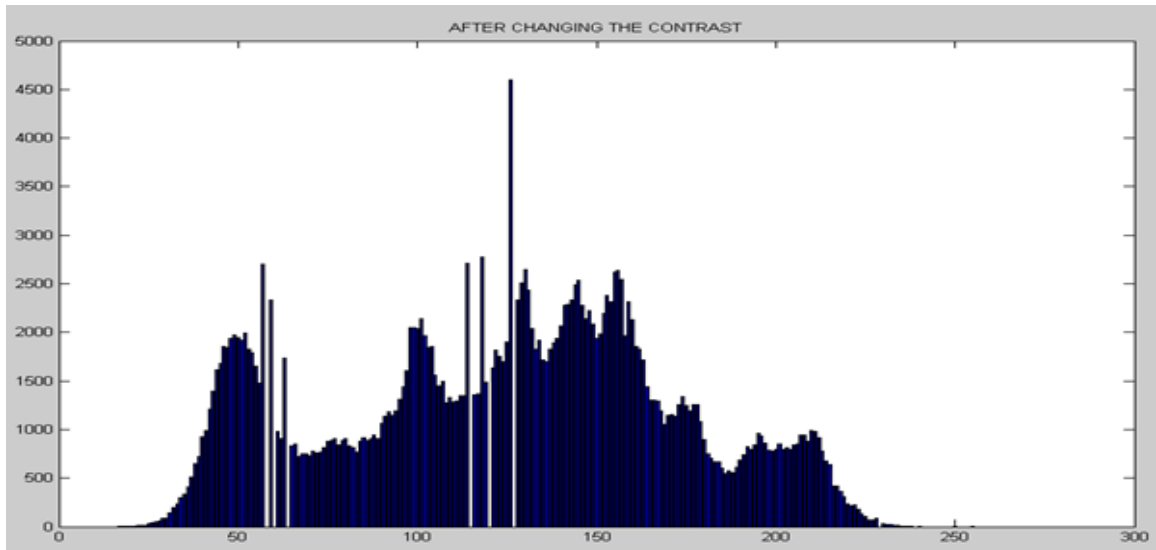


Fig. 9: before contrast [14]

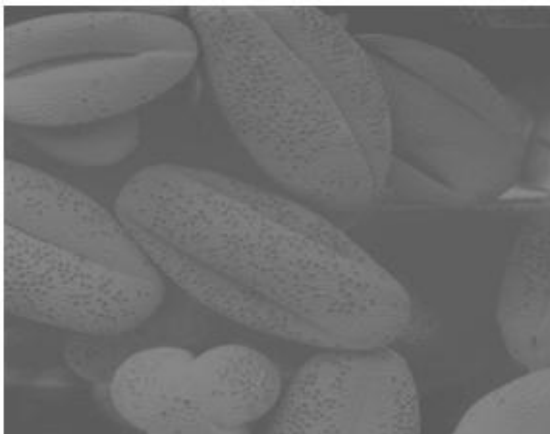


Fig. 10: before contrast [14]

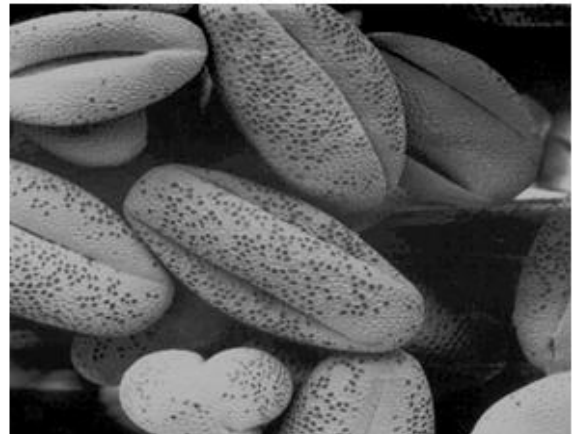


Fig. 11: after contrast [14]

moved across the image until every pixel has been covered. Edge preserving smoothing, Neighborhood averaging or Gaussian smoothing will tend to blur edges because the high frequencies in the image are attenuated.

An alternative approach is to use median filtering. Setting of the grey level is done to be median of the pixel values in the neighborhood of that pixel. The median  $m$  of a set of values is such that half the values in the set are less than  $m$  and half are greater. For example, suppose the pixel values in a  $3 \times 3$  neighborhood are (10, 20, 20, 15, 20, 20, 20, 25, and 100). With the help of sorting technique among the values (10, 15, 20, 20, 20, 20, 25, 100) median value is found 20.



Fig. 12(a): before smoothing (b) after smoothing

### 2.3.3 Brightening an Image

Brightened Image developed from original image by increasing every pixel with a constant. The values of pixels are increased in the constant ratio.



Fig. 13: Image before bright [9]



Fig. 14: Image after bright [9]

### 2.3.4 Threshold

An image can be enhanced by its threshold value. Threshold of any image is the point, after that, there is not any increment in the pixel in any direction.

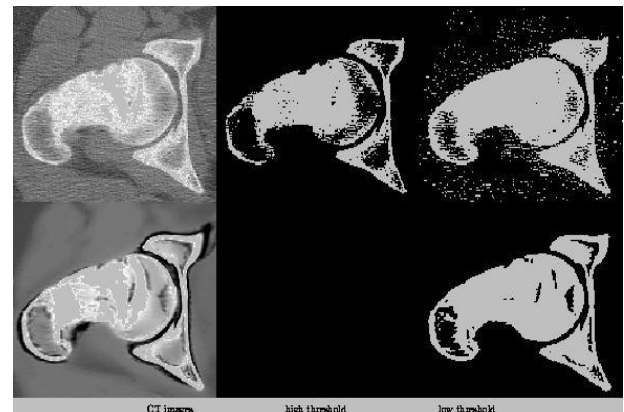


Fig. 15: Image under different threshold [2]

### 2.3.5 Intensity transformation

By increasing the pixel size of any image enhancement of image is done in  $x$ ,  $y$  and  $z$  all the three directions. Scaling concept is applied in the intensity transformation in which the value of each pixel is increased in a scale factor  $a$ , let a 2D image function if defined by  $[m(x), m(y)]$  after intensity transformation it will be  $[a.m(x), b.m(y)]$  where,  $a$  and  $b$  are the intensity factors in  $x$  and  $y$  directions respectively.



Fig.16: Image under less intensity transformation [3]



Fig. 17: Image under more intensity transformation [3]

In intensity transformation uses Discrete Wavelet Transformation (DWT) and scaling technique for image enhancement. First draw the histogram of the particular image. In the discrete values of  $X$ ,  $Y$  and  $Z$  directions add the



scaled value (intensity by which the value is going to increase), the resultant image is the desired output of enhanced image.

### 3. TOOLS & HDL ENVIRONMENT

*Project navigator Application Version ISE 14.2 of Xilinx Company:* Xilinx [11] has been a semiconductor industry leader at the forefront of technology, market and business achievement. It is a tool to design the IC and to view their RTL (Register Transfer Logic) schematic. It is a tool to test the code on FPGA environment and we can get the all parameters details required to implement the Chip.

*Model SimEE 10.0 D of Mentor Graphics Company:* Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim [12] the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

#### Simulation and Design Steps

The following diagram shows the basic steps for simulating a design in ModelSim.

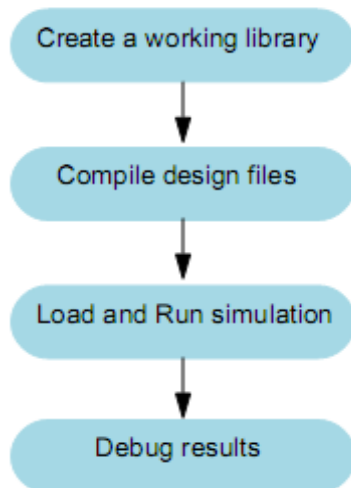


Fig. 18: Chip Design Process Flow [12]

#### Creating the Working Library

In ModelSim, all designs are compiled into a library. Typically start a new simulation in ModelSim by creating a working library called "work," which is the default library name used by the compiler as the default destination for compiled design units.

#### Compiling Design

After creating the working library, design is being compiled into it. The ModelSim library format is compatible across all supported platforms.

#### Loading and Running the Simulator with the Design

With the design compiled, we load the simulator with design by invoking the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the design loads successfully, the simulation time is set to zero, and you enter a run command to begin simulation.

#### Debugging

Modelsim's robust debugging environment is used to track down the cause of the problem.

## 4. SIMULAION RESULTS

The result obtained in Modlesim software for 2D image enhancement chip is shown below in figure 19.

*Step input 1:* reset =1, clk is used for synchronization and then run.

*Step input 2:* reset =0, same clk is used for synchronization, give image input value and scaling value to get the results.

We have taken the case of 2D Image  $6 \times 6$ .  $image\_in(x)$ ,  $image\_in(y)$  have the discrete instant values and  $sample\_in(x)$  and  $sample\_in(y)$  are the value which should be added resulted values are  $image\_out(x)$  and  $image\_out(y)$ .

The result obtained in Modlesim software for 3D image enhancement chip is shown below in figure 20.

*Step input 1:* reset =1, clk is used for synchronization and then run.

*Step input 2:* reset =0, same clk is used for synchronization, give image input value and scaling value to get the results.

We have taken the case of 3D Image  $6 \times 6 \times 6$ .  $image\_in(x)$ ,  $image\_in(y)$  and  $image\_in(z)$  have the discrete instant values and  $sample\_in(x)$ ,  $sample\_in(y)$  and  $sample\_in(z)$  are the value which should be added resulted values are  $image\_out(x)$ ,  $image\_out(y)$  and  $image\_out(z)$ .

### 4.1 Device utilization & Timing Summary of 2D and 3D image

Device utilization is the hardware and logic circuitry required to implement the design. It consist of no. of slices, flip flops, gates, combinational logic circuitry and input/output blocks.

Selected Device: 3s400pq208-4

Table 1 Hardware utilized for 2D image enhancement chip

FPGA Resources	2D chip
Slices	45.23%
Slice Registers	43%
LUTs	15.7%
IOB	15%
GCLKs	25%
Equivalent Gate Count	227,176

Table 2 Hardware utilized for 3D image enhancement chip

FPGA Resources	3D chip
Slices	57.85%
Slice Registers	52.42 %
LUTs	16.58 %
IOB	21.112%
GCLKs	25%
Equivalent Gate Count	294056

### 4.2 Timing Summary

Timing summary is the detail achieved by software with the minimum and maximum time required to implement the chip.

#### Timing Summary for 2D image chip

Speed Grade: -4

Minimum period: 2.414ns

Maximum frequency: 414.250MHz

Minimum input arrival time before clock: 5.629ns

Maximum output required time after clock: 6.604ns

#### Timing Summary for 3D image chip

Speed Grade: -4

Minimum period: 2.414ns

Maximum frequency: 414.250MHz

Minimum input arrival time before clock: 5.028 ns

Maximum output required time after clock: 5.104ns

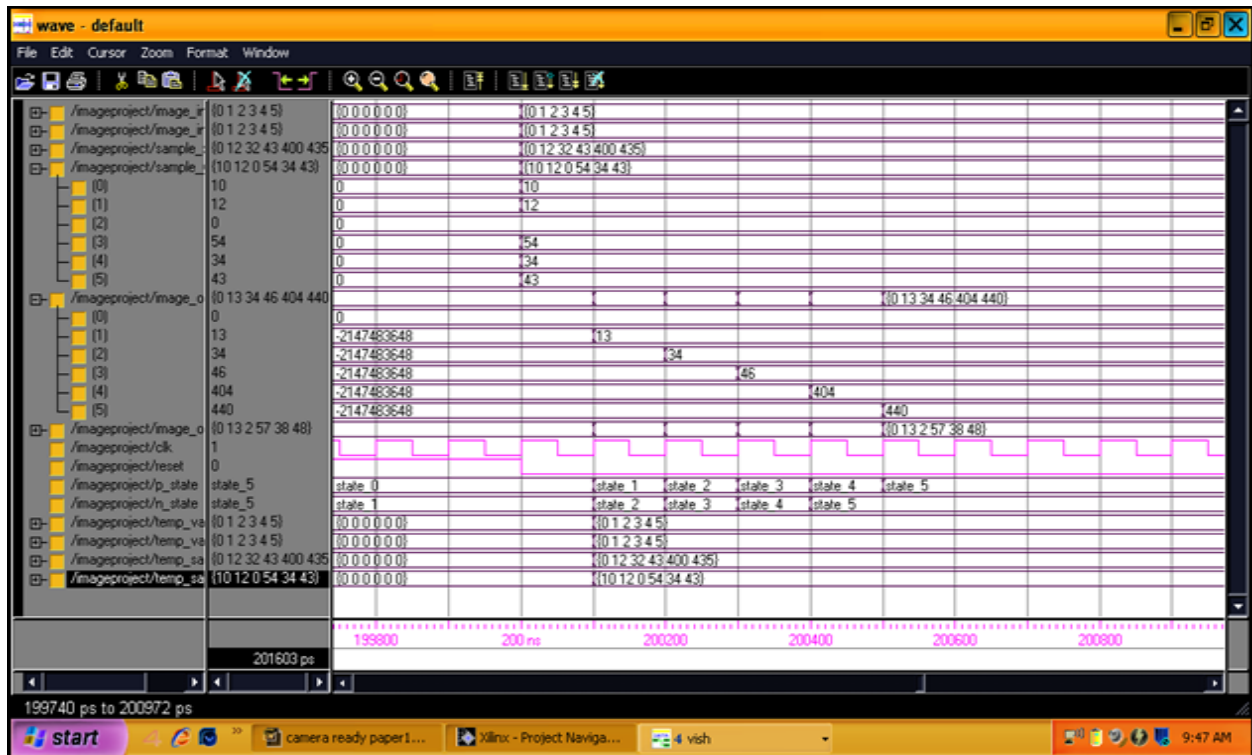


Fig 19: Wave form of 6 x 6 2D image enhancement chip

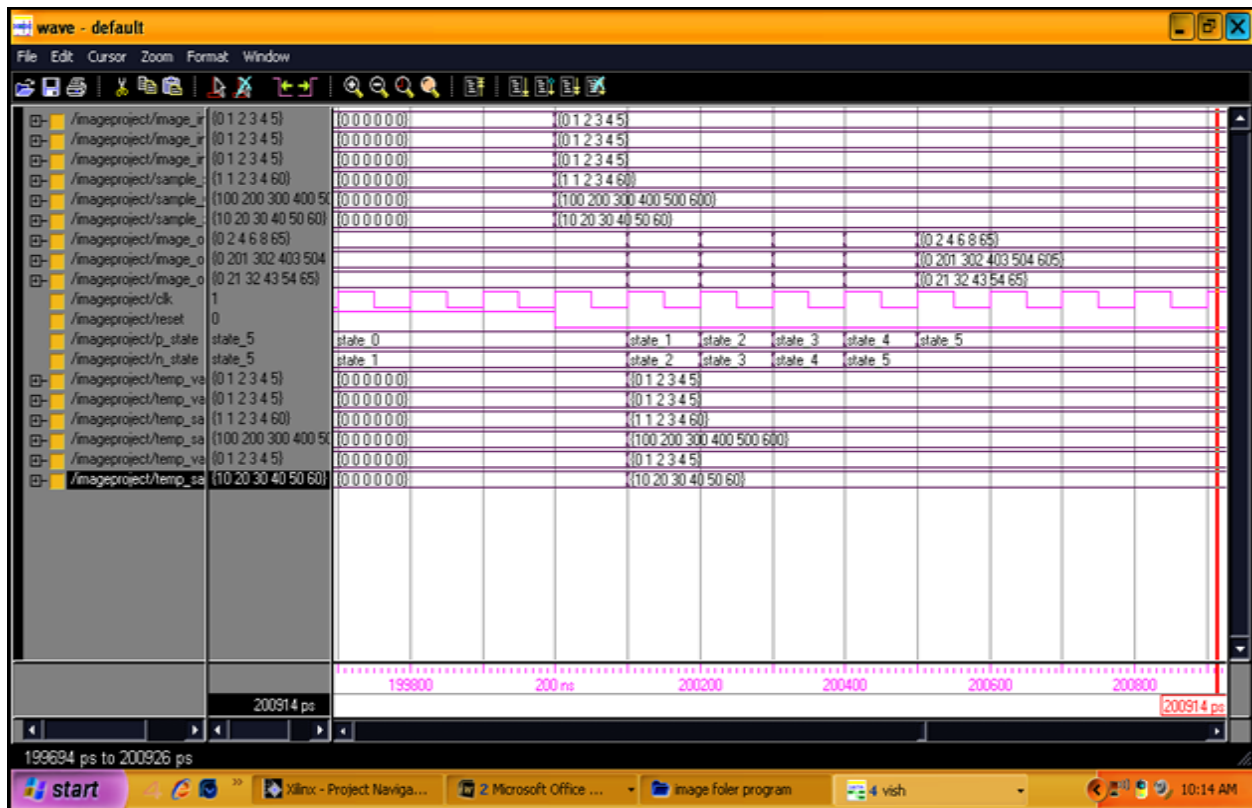


Fig 20: Wave form of 6 x 6 3D image enhancement chip

## 5. CONCLUSION

The design IC's for 2D and 3D image enhancement chip that leads the low power and optimized design is done in VHDL programming environment. In contrast the development is done to make programmable image chip to increase the efficiency, lower the power requirement, and reduce the delay with the comparison of previous work. Such chips can be used in digital cameras, digital oscilloscope etc. VHDL has been used to write all the programs for the IC's because of its user-friendly nature and thus modifications if required for further development shall not prove to be an obstacle. Thus the research focuses on simulation prior to fabrication. Burning these programs on FPGA (Field Programmable Gate Array) will help us to see the functional design of ICs. These results in addition to the systematic view generated would help us to design Application Specific ASIC. This work is a significant effort towards total digitization of image processing and would surely prove a boon for VLSI design industry.

## 6. REFERENCES

- [1] M. Chandrashekhar, U. Naresh Kumar, K. Sudershan Reddy and K. Nagabhushan Raju, "FPGA Implementation of High Speed Infrared Image Enhancement", International Journal of Electronic Engineering Research ISSN 0975-6450 Volume 1 Number 3 (2009) pp 279-285.
- [2] Deggu Venkateshwar Rao, Shruti patil, Naveen Anne Babu and V Muthukumar "Implementation and evaluation of Image processing Algorithms on Reconfigurable architecture using C-based Hardware Descriptive Languages" International Journal of Theoretical and applied Computer Sciences Volume 1 Number 1 (2006) pp 9-34.
- [3] Nathaniel J. McCaffrey, Francis P. Pantuso "Very low cost real time histogram-based contrast enhancer utilizing fixed-point DSP processing" Real-Time Imaging III (Proceedings of SPIE Volume 3303) San Jose, CA; (1998); p. 36-43;
- [4] Khurram Bukhari, Georgi Kuzmanov and Stamits Vassiliadis: In this paper "DCT and IDCT Implementation on different FPGA Technologies"
- [5] W. Sowmya, Roy PailyIn, "FPGA implementation of Image Enhancement Algorithm" International conference on Communication and signal processing (ICCSP) (2011) IEEE pp 584-588.
- [6] Tripti Jain, Prashant bansod, C. B. Singh Kushwah and mayenk Mewara, "Reconfigurable Hardware for Median Filtering for image Processing Application" 3<sup>rd</sup> International Conference on Emerging Trends in Engineering and Technology(2010) IEEE.
- [7] K. Sri Rama Krishna, A. Guruva Reddy, M. N. Giri Prasad "A Fault Tolerant FPGA based image Enhancement Filter using Self Healing Algorithm" International Journal of Engineering Science and Technology Vol2(9), 2010,pp 4922-4928.
- [8] Himanshu Singh, Ajay Kumar, S.S. Negi, "Algorithm and hardware Implementation of Real Time Automatic Gain Control Feature for Thermal Imager" Proceedings of the Fourth Indian Conference on Computer Vision, Graphics & Image Processing(2004)
- [9] John C. Ross. Image Processing Hand book, CRC Press. 1994.
- [10] Peter Mc Curry, Fearghal Morgan, Liam Kilmartin. Xilinx FPGA implementation of a pixel processor for object detection applications. In the Proc. Irish Signals and Systems Conference, Volume 3, Page(s):346 – 349, Oct. 2001.
- [11] Namrata Vaswani, Yogesh Rathi, Anthony Yezzi, and Allen Tannenbaum, "Deform PF-MT Particle Filter With Mode Tracker for Tracking Nonaffine Contour Deformations" IEEE Transactions on Image Processing, Vol. 19 ,No .4, April 2010.
- [12] B. Srinivasa Reddy and B. N. Chatterji, "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration", IEEE Transactions on Image Processing, Volume 5, No. 8, August 1996.
- [13] Computer Vision CITS4240 School of Computer Science & Software Engineering, The University of Western Australia
- [14] B.Chanda D, Dutta Majumder Digital Image processing and Analysis by, PHI Publication , 2<sup>nd</sup> edition pp, 20-121
- [15] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing. Addison-Wesley Publishing Company, 1992, chapter 4