Image Enhancement using Hardware co-simulation for Biomedical Applications

Kalyani A. Dakre
Dept. of Electronics and Telecommunications
P.R. Pote (Patil) college of Engineering and,
Management, Amravati, India
kalyanidakre@gmail.com

Prof. P. N. Pusdekar
Dept. of Electronics and Telecommunications
P.R Pote(Patil) college of Engineering and
Management Amravati, India
pusdekar.wardha@gmail.com

***Abstract*—** Digital image enhancement techniques are to improving the visual quality of images. Main objective of image enhancement is to process an image so that result is more suitable than original image for specific application. Image is one of the most fundamental and significant features. The correctness and reliability of its results affect directly the comprehension machine system made for objective world. The implementation of image enhancement algorithms on a field programmable gate array (FPGA) is having advantage of using large memory and embedded multipliers. FPGAs are providing a platform for processing real time algorithms on application-specific hardware with substantially higher performance than programmable digital signal processors (DSPs). This project focus on implementation issues of image enhancement algorithms like brightness control, contrast stretching, negative transformation, thresholding, filtering techniques on FPGA that have become a competitive alternative for high performance digital signal processing applications. This project will use System Generator tool and modular construction methods to build a image algorithm platform in MATLAB. By a brief analysis about display image and resource consumption after achieving on Spartan-3E development board, we can see the image using System Generator for FPGA algorithm design superiority, have the vast application prospects

Keywords-: *FPGA, DSP, Image Processing, Image Enhancement, MATLAB, System generator.*

*****

I. INTRODUCTION

Digital image processing plays a vital role in the analysis and interpretation of remotely sensed data. Especially data obtained from Medical and Satellite Remote Sensing, which is in the digital form, can best be utilized with the help of digital image processing. Image enhancement and information extraction are two important components of digital image processing. Image enhancement techniques help in improving the visibility of any portion or feature of the image. Image enhancement processes have different techniques to improve the visual appearance of an image In recent years, as FPGA superior performance, rich resources, high-speed parallel computing capacity, FPGA digital signal processing systems have become the core of the device, especially in digital communications, video and images FPGA has been widely used in the field. By using FPGA design become more flexibility, procedures and modules portability improved, and it can also shorten the design cycle, reduce the hardware investment risk. This project presents information about FPGA implementation for various Image Processing Algorithms using the most efficient tool called Xilinx System Generator (XSG) for Matlab. This project focus on implementation of image enhancement algorithms like

1) brightness control
2) contrast stretching,
3) negative transformation,
4) Segmentation using threshold transform
5) Range highlighting transform.
6) Parabola transform

A. XILINX SYSTEM GENERATOR

System Generator is part of the ISE® Design Suite and provides Xilinx DSP Blockset such as adders, multipliers, registers, filters and memories for application specific design. These blocks leverage the Xilinx IP core generators to deliver optimized results for the selected device. Previous experience with Xilinx FPGAs or RTL design methodologies is not required when using System Generator. Designs are captured in the DSP friendly Simulink modelling environment using a Xilinx specific Blockset. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file. Advantage of using Xilinx system generator for hardware implementation is that Xilinx Blockset provides close integration with MATLAB Simulink that helps in co-simulating the FPGA module with pixel vector provided by MATLAB Simulink Blocks.

B. DESIGN FLOW FOR IMAGE PROCESSING WITH XILINX SYSTEM GENERATOR

For performing Image processing task using Xilinx System Generator needs two Software tools to be installed. One is MATLAB Version R2009a. Or higher and Xilinx ISE 13.1. The System Generator token available along with Xilinx must be configured to MATLAB. That is addition of Xilinx Blockset to the Matlab Simulink environment which can be directly utilized for building algorithmic model. The algorithms are developed and models are built for image negative, enhancement etc. using library provided by Xilinx Blockset. The image pixels are provided to Xilinx models in Xilinx fixed point format. These models are simulated in Matlab Simulink environment with suitable simulation time and simulation mode and tested. The results can be seen on a video viewer. Once the expected results are obtained System Generator is configured for suitable FPGA board. FPGA board Spartan3E is used hereand the model is implemented for JTAG hardware co-simulation. The System generator parameters are set and generated. On compilation the netlist is generated and a draft for the model and programming file in Verilog HDL is formed which can be accessed using Xilinx ISE. The module is checked for behavioral syntax check,

synthesized and implemented on FPGA..Bitstream compilation is done which is necessary to create an FPGA bit file which is suitable for FPGA input. The Fig.1 shows the Design flow for Xilinx System Generator.

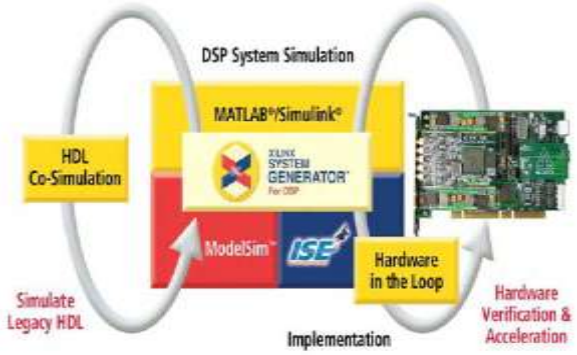


Fig-1 .Design flow of XSG system

II. PROPOSED WORK

The entire operation is proposed using Simulink and Xilinx blocks goes through three phases,

- Image pre-processing blocks.
- Image enhancement algorithm using XSG.
- Image post-processing blocks

The design flow of hardware implementation of image enhancement using XSG is given in fig 1.Image source and image viewer are simulink block sets by using these blocks image can give as input and output image can be viewed on image viewer block set. Image preprocessing and image post-processing units are common for all the image processing applications which are designed using Simulink block sets. image enhancement algorithms are different for each and every edged detection algorithms which is implemented using Xilinx System Generator block sets.

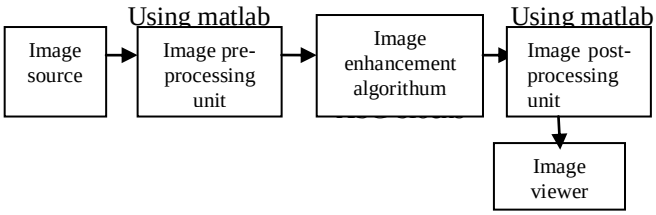


Fig -2. Design flow of hardware implementation of image enhancement.

A. Image pre-processing unit

Image preprocessing in Matlab helps in providing input to FPGA as specific test vector array which is suitable for FPGA Bitstream compilation using system generator.

Resize: Set Input dimensions for an image and interpolation i.e. bicubic it helps in preserving fine detail in an image.

Convert 2-D to 1-D: Converts the image into single array of pixels.

Frame conversion: Frame conversion block sets output signal to frame based data of particular size and provided to unbuffer block

Unbuffer: Unbuffer block which converts this frame to scalar samples output at a higher sampling rate.

The model based design used for image pre-processing is shown in Figure 2. Input images which could be color or grayscale are provided as input to the File block.

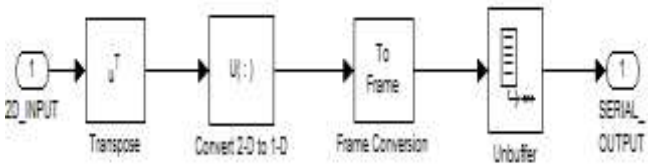


Fig- 3: Image Pre-processing

B. Image Post Processing Blocksets

The post-processing blocks which are used to convert the image output back to floating point type we get 2d image at output is shown in figure 4.3. It compare with threshold value and finally we get fine edge detected image which we can see in Simulink environment.

Post-processing uses-

Data type conversion: It converts image signal to unsigned integer format.

Buffer : Converts scalar samples to frame output at lower sampling rate

Convert 1D to 2D: Convert 1D image signal to 2D image matrix.

Sink: It is used to display the output image back on the monitor.

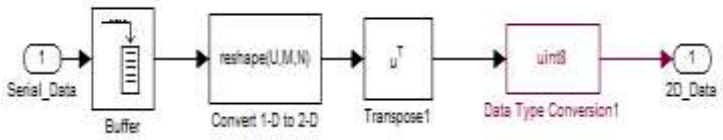


Fig-4: image post-processing blocks

III. XILINX MODELS FOR IMAGE PROCESSING ALGORITHMS

Development of models is based on algorithms used for Image Processing. Once the FPGA boundaries have been established using the Gateway blocks, the DSP design can be constructed using blocks from the Xilinx DSP block set. Standard Simulink blocks are not supported for use within the Gateway In/Gateway Out blocks.

A. Algorithm for Brightness control.

The brightness of a dark image can easily be increased by adding a constant to gray value of each pixel. This addition operation will shift the histogram towards brighter side with a constant factor. While applying this method to increase brightness of an image, we must choose the constant wisely so that the complete range of gray values lies within 0 to 255. If the final gray value of any pixel is greater than 255 then we will lose the information.This algorithm works as follows:

ISSN: 2321 - 8169
869 - 877

J(r)=I(r)+g,if I(r)+g<=255
J(r)= 255, if I(r)+g<=255

g-is a constant value (g>0). I(r)-is gray level of input pixel (r) and J(r) is the gray level of output pixel (r) after the brightness increasing process.

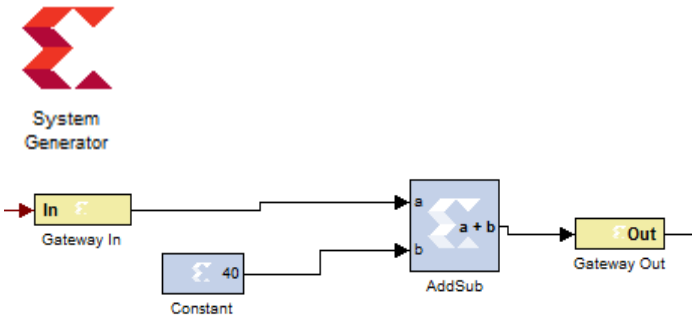


Figure-5 : Algorithm for Image Brightness control

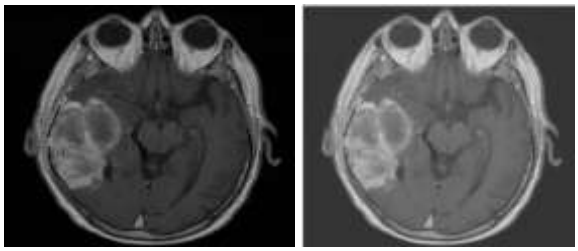


Figure-6: Result for brightness control

B. Algorithm for contrast stretching

Contrast stretching attempts to improve an image by stretching the range of intensity values it contains to make full use of possible values. Mathematically it is expressed as:

new_value = (old_value – 5) × contrast + 2

new_value = (old_value – 160) × 3 + 192

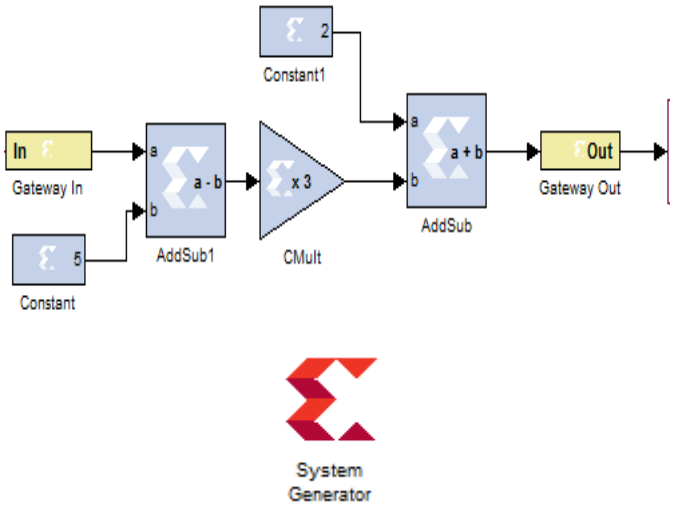


Figure -7: Algorithm for Contrast stretching grayscale

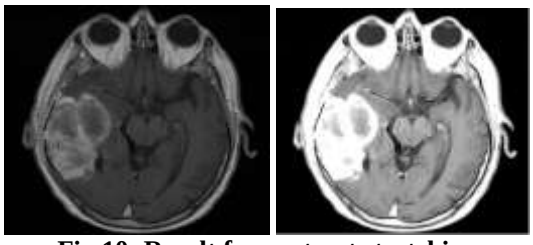


Fig 10: Result for contrast stretching

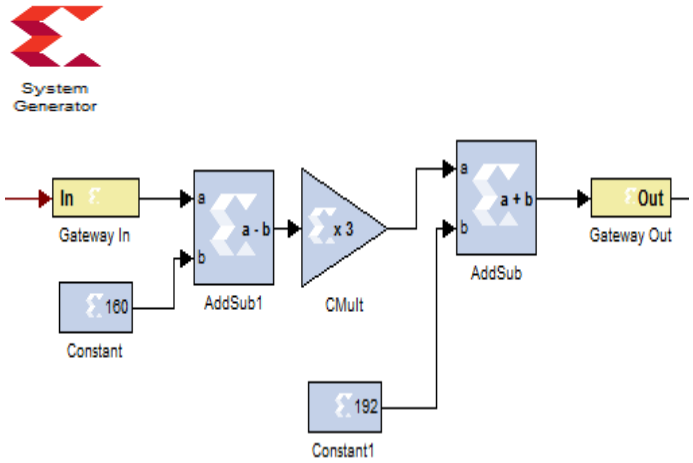


Figure-8: Algorithm for histogram stretching2

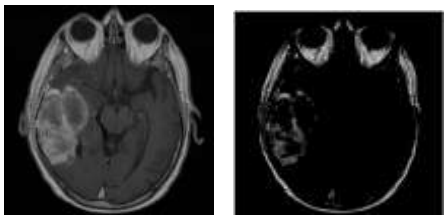


Figure-9: Results for Histogram stretching 2

C. Algorithm for Negative Transformation

The negative transform exchanges dark values for light values and vice versa. This is the complement of a grayscale image like a photographic negative. The equation is as follows :

new pixel = 255 – old pixel

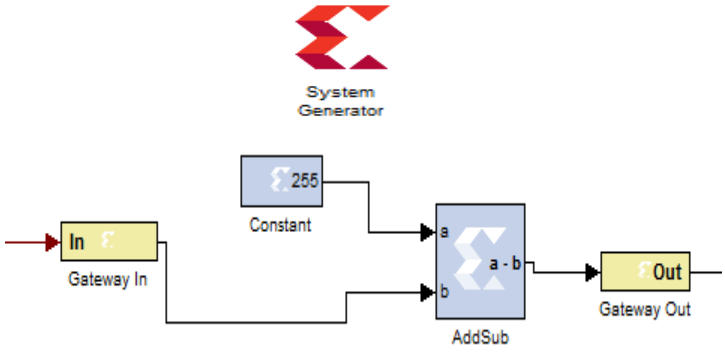


Figure-10: Algorithm for negative transform

 ISSN: 2321 - 8169
869 - 877

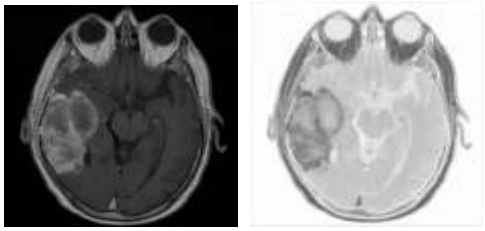


Figure 11: Result for Negative transformation

D. Algorithm for Image Segmentation using Threshold

Image segmentation can be used to separate pixels associated with objects of interest from the image background. This is an important step in many imaging applications of automated analysisand robotics. We demonstrate segmentation on a simple pixel-by-pixel basis using threshold decisions. are well-separated. We use Mcode block and use 50 as a threshold in our demonstration.

The Mcode block description is as follows :

```
function z = newpixel(x,y)
if x>y
z = x;
else
z = 1;
end
```

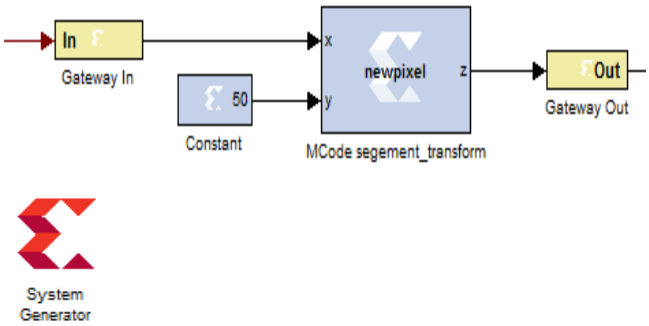


Figure12: Algorithm for Image segmentation and threshold

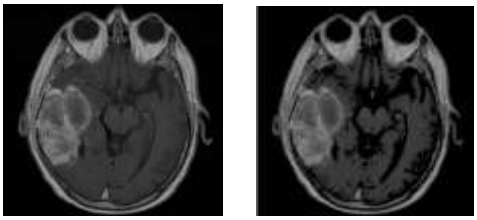


Figure- 13: Results for segmentation using threshold transform

E. Algorithm for Range highlighting Transformation

An intensity transform can also highlight a range of pixels while keeping others constant.

```
function z = newpixelone(x,y,c)
if (x > y) & (x < c)
z = x;
else
z = 1;
end
```

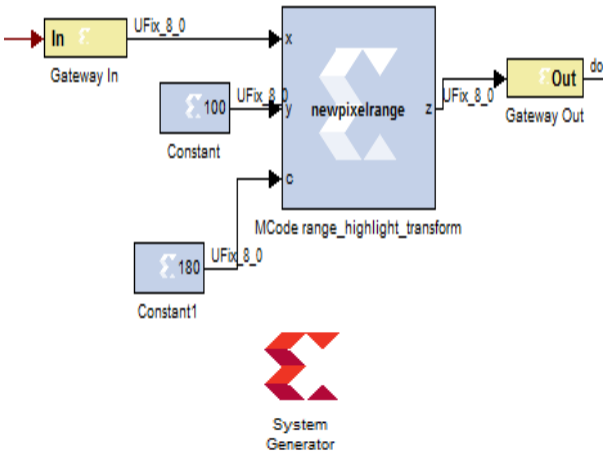


Figure-14: Algorithm for range highlight transform

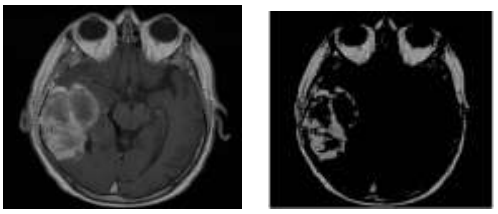


Figure-15: Results for range highlight transform

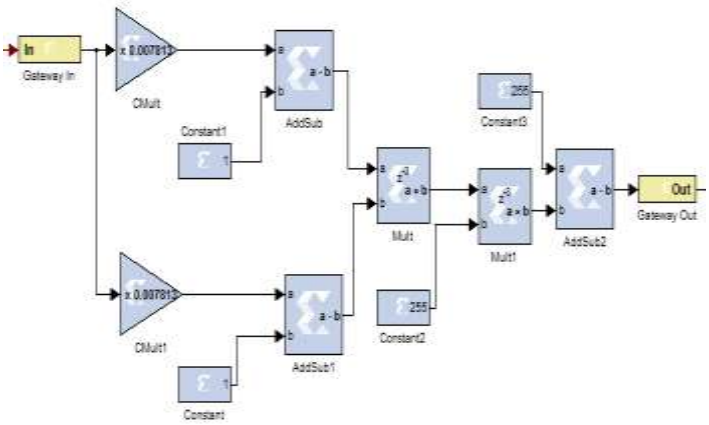
F. Parabola Transformation

The two formulas for the parabola transformation are as follows:

new pixel = 255 – 255 ((old pixel/128) -1)2 (4) and

new pixel = 255 ((old pixel/128) -1)2 (5)

Xilinx blocks are connected for the above equations and displayed in Fig.. Both the results are observed and produced.



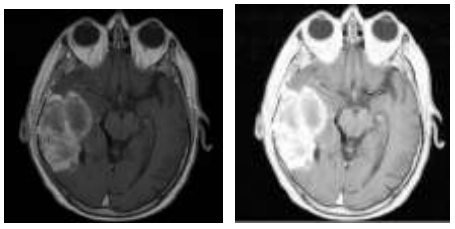


Figure-16: Results for parabola transform

 ISSN: 2321 - 8169
869 - 877

IV. HARDWARE CO-SIMULATION

For implementation of this design in a FPGA board the entire module should be converted to FPGA synthesizable format. For that purpose main module for edge detection is converted to JTAG hardware co-simulation, this is done with the help of System generator block specially its system generator token. This block is configured according to the target platform and a bit stream (*.bit) file is generated. After the bit stream file is generated, hardware co-simulation target is selected and in this work, Spartan 3E starter kit (XC3S500E-FG320) is used for board level implementation.

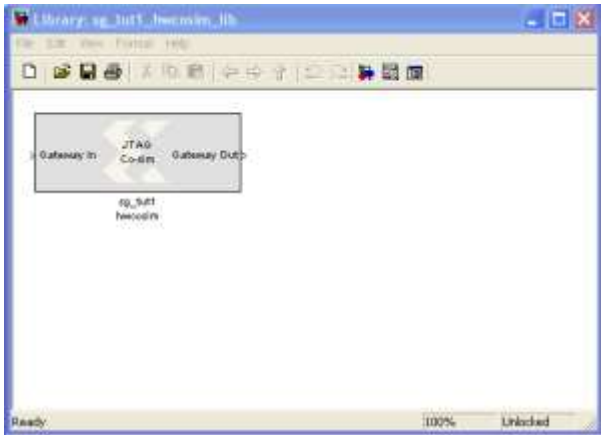
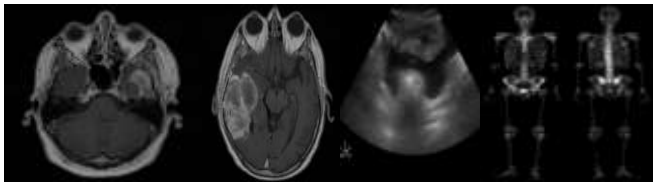


Figure-17: Hardware co-simulation block

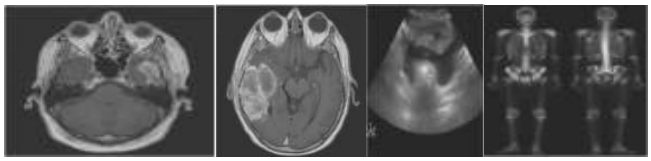
V. RESULTS

The proposed Fpga implementation of various image enhancement algorithms is done using Matlab simulink and Xilinx system generator.Both hardware and software implementation of image enhancement algorithms for various images is tabulated below.

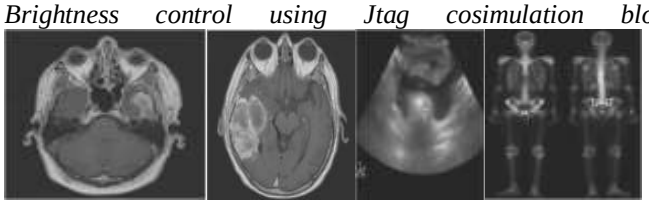
Input images



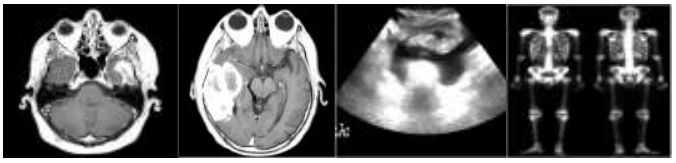
1) Brightness control using Xilinx system generator hardware block



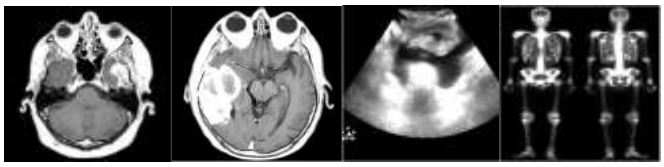
Brightness control using Jtag cosimulation block



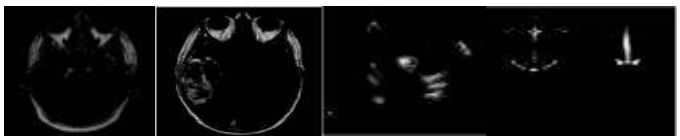
2) Contrast stretching1 using Xilinx system generator hardware block



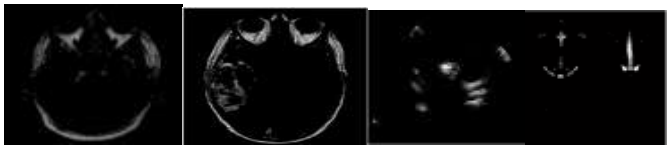
Contrast stretching using Jtag co-simulation block



3) *Histogram stretching2 using Xilinx system generator hardware block*

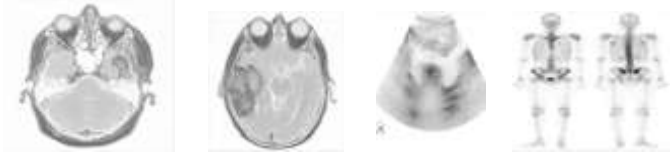


Histogram stretching2 using Jtag co-simulation block

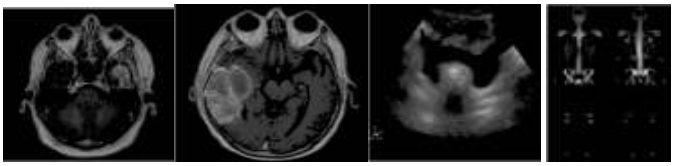


4) Negative transform using Xilinx system generator hardware block

Negative transform using Jtag co-simulation block

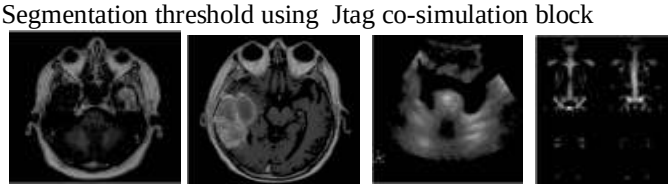


5) Segmentation threshold using Xilinx system generator hardware block

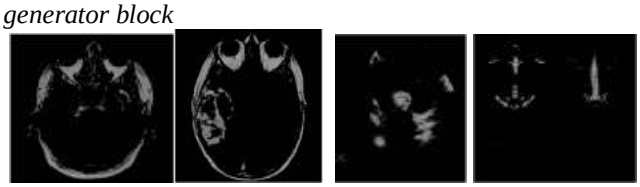


 ISSN: 2321 - 8169
869 - 877

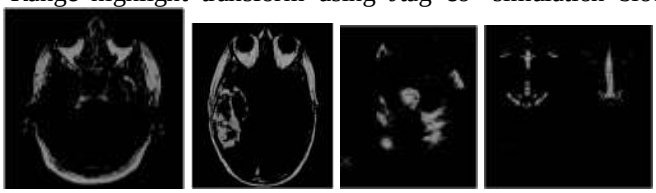
Segmentation threshold using Jtag co-simulation block



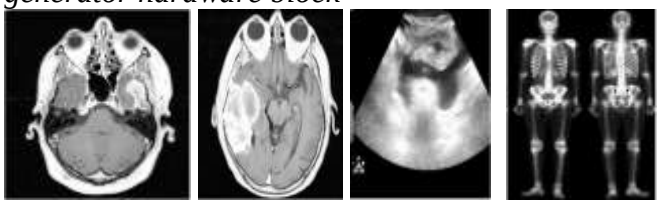
6) Range highlight transform using Xilinx system generator block



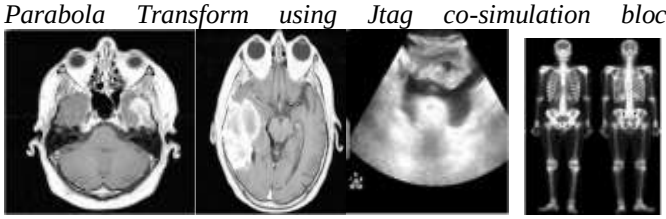
Range highlight transform using Jtag co- simulation block



7) *Parabola Transform using Xilinx system generator hardware block*



Parabola Transform using Jtag co-simulation block



Implementations of RTL schematic and view technology schematic for Brightness control, Contrast stretching, Negative transformation, segmentation using Thresholding range highlighting transform and parabola transform are given

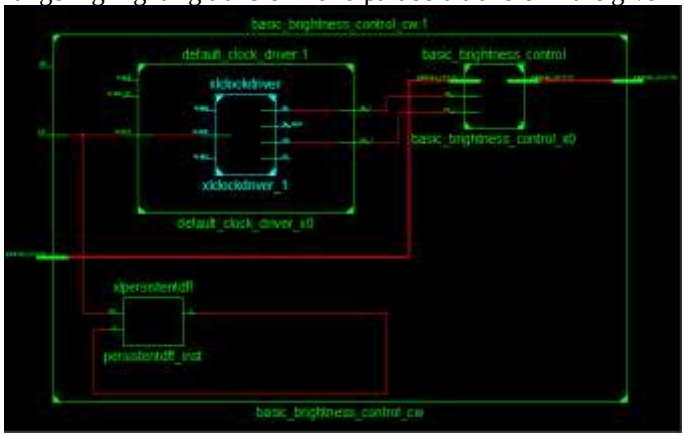


Figure-18 : RTL schematic of Brightness control

Figure-19 : View technology schematic of Brightness control

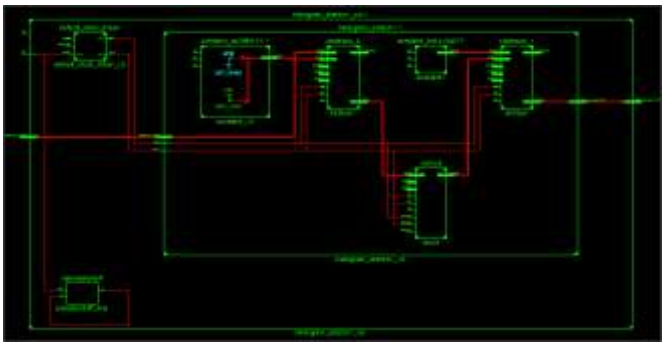


Figure-20 : RTL schematic of Histogram stretching1

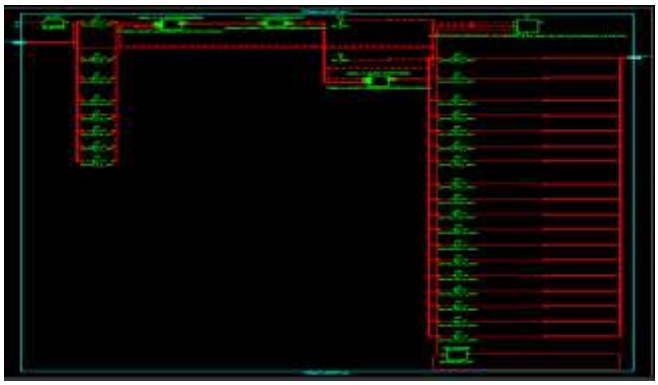


Figure-21: View technology schematic Histogram stretching1

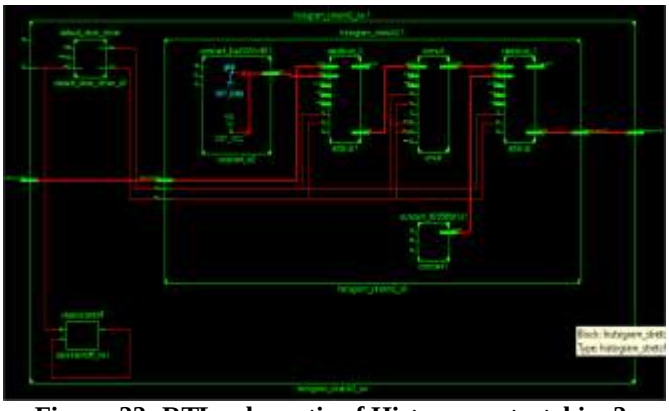


Figure-22: RTL schematic of Histogram stretching2

ISSN: 2321 - 8169
869 - 877

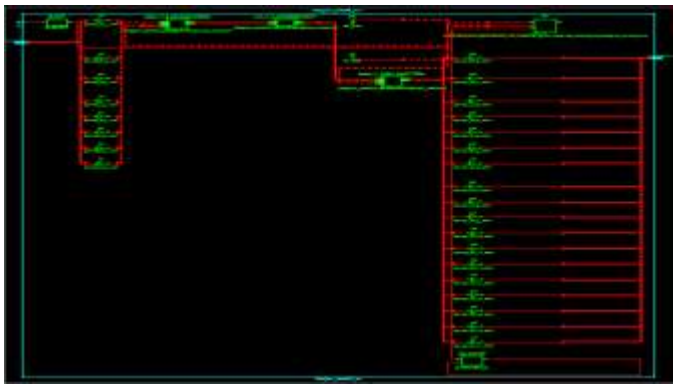


Figure-23:View technology schematic of Histogram stretching2

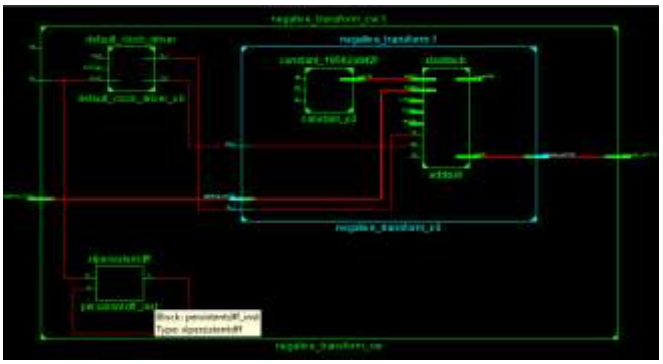


Figure-24 : RTL schematic of negative transform

Figure-24 : View technology schematic of negative transform

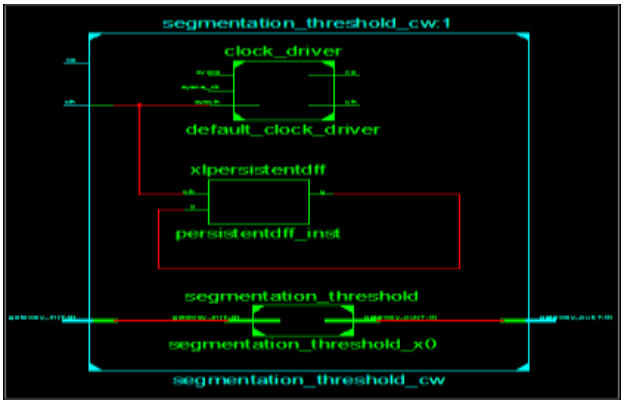


Figure-25: RTL Schematic of segmentation using thresholding

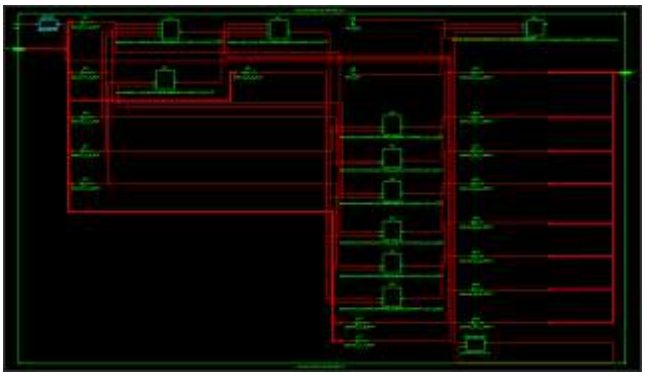


Figure-26 : View technology Schematic segmentation using thresholding

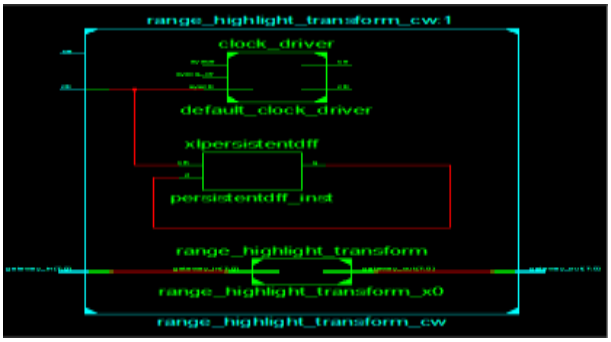


Figure-27: RTL schematic of Range highlighting transform

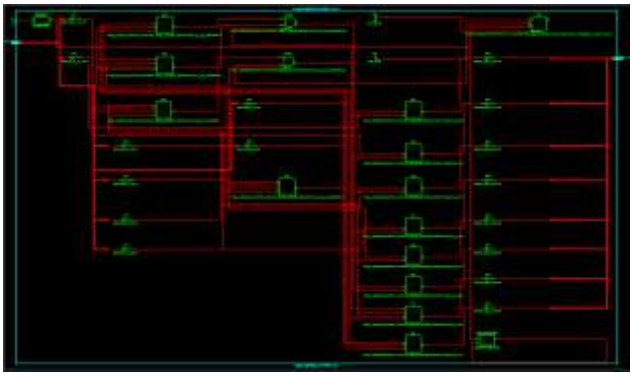


Figure-28: View technology schematic of range highlight transform

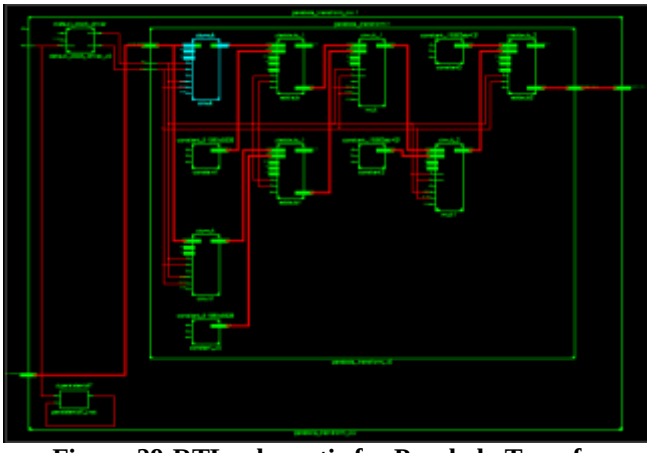


Figure-29:RTL schematic for Parabola Transform

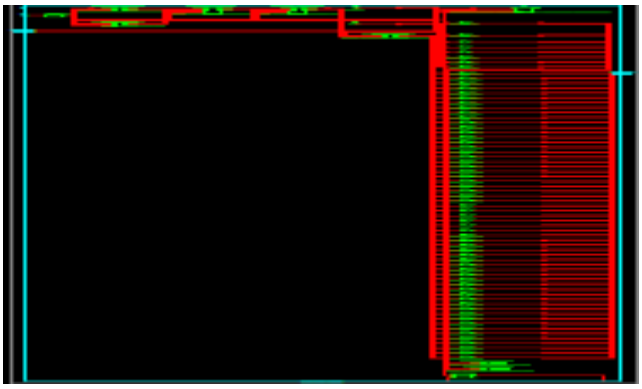


Figure-30:View Technology schematic of parabola transform

TABLES SHOWS ANALYSIS FOR RESOURCE ESTIMATION AND SIMULATION TIME FOR SPARTAN3E FPGA FOR VARIOUS IMAGE ENHANCEMENT ALGORITHMS

Device Utilization Summary for Brightness control

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of bonded IOBs	17	232	7%
Number of GCLKs	1	24	4%

Device Utilization Summary for Histogram stretching1

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of bonded IOBs	27	232	11%
Number of GCLKs	1	24	4%

Device Utilization Summary for Histogram stretching2

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of bonded IOBs	27	232	11%
Number of GCLKs	1	24	4%

Device Utilization Summary for negative transform

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of bonded IOBs	17	232	7%
Number of GCLKs	1	24	4%

Device Utilization Summary for segmentation using threshold transform

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	6	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of 4 input LUTs	9	9312	0%
Number of bonded IOBs	17	232	7%
Number of GCLKs	1	24	4%

Device Utilization Summary for Range highlighting transform

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	8	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of 4 input LUTs	12	9312	0%
Number of bonded IOBs	17	232	7%
Number of GCLKs	1	24	4%

Device Utilization Summary for parabola transform

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of Slice Flip Flops	1	9312	0%
Number of bonded IOBs	68	232	29%
Number of GCLKs	1	24	4%

Timing summary:

Algorithms	Maximum combinational path delay
Brightness control	3.692ns
Histogram stretching1	3.692ns
Histogram stretching2	3.692ns
Negative transform	3.692ns
Image segmentation using threshold	9.083ns
Range highlighting transform	9.459ns
Parabola Transform	3.692ns

VI. DISCUSSION

The results of proposed hardware and software implementation for various image enhancement algorithms is compared. Device utilization summary shows that the device requirement for all those image enhancement algorithms is very less. The algorithms were implemented on Spartan-3E fpga development kit. This gives digital images with good brightness/contrast and it improves quality of images. This

techniques used for different applications depends on requirement and need.

CONCLUSION

We conclude from this paper that Xilinx System Generator is a multipurpose tool to perform software and hardware image processing task. It provides rapid means to do hardware implementation of complex techniques used for processing images with minimum resource and minimum delay.

This paper implemented for high speed image enhancement applications using fpga. The image enhancement techniques such as brightness and contrast adjustment are important factors in medical images. This paper explains implementation of Brightness control, Contrast stretching, Negative transform, segmentation using thresholding,Range highlight transform,Parabola transform algorithms on fpga. These techniques used in digital x-ray, digital mammography, CT scans, MRI etc.

REFERANCES

[1] Basu, A.,T..S. Das and S.K. Sarkar, 2012. SysGen architecture for visual information hiding framework. Int. J. Emerg. Technol. Adv. Eng., 2: 32-40.
[2] Christe, S.A., M. Vignesh and A. Kandaswamy, 2011. An efficient FPGA implementation of MRI image is filtering and tumour characterization using Xilinx system generator. Int. J. VLSI Des. Commun. Syst., 2: 95-109.
[3] ZhangShanshan, WangXiaohong "Vehicle Image Edge Detection Algorithm Hardware Implementation On FPGA." (2010 International Conference on Computer Application and System Modeling (ICCASM 2010).
[4] Harinarayan, R., R. Pannerselvam, M.M. Ali and D.K. Tripathi, 2011. Feature extraction of digital aerial images by FPGA based implementation of edge detection algorithms. Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology, March 23-24, 2011, Tamil Nadu, pp: 631-635.
[5] Sudeep, K.C. and J. Majumdar, 2011. A novel architecture for real time implementation of edge detectors on FPGA. Int. J. Comput. Sci., 8: 193-202.
[6] Xilinx System Generator User's Guide, www. Xilinx.com
[7] Spartan-3E Starter Kit Board User Guide, www. Xilinx.com