Low Level Image Processing Algorithms Using Hardware Software Co Simulation

Dushyant Mankar **Prof. S.S.Mungona**

Abstract—**Digital image processing based applications area have found its presence in most of the today's world. It includes signal and image processing, medical field, machine vision, traffic load computation etc. These image processing algorithms can be implemented very easily by using Xilinx System Generator. The use of XSG enables us to use the concept of hardware software co-simulation which is very essential in low cost but powerful processing chips. Proposed architecture gives an most efficient way by using a graphical user interface which is developed by combining MATLAB, Simulink and XSG tool. Prototype of Application Specific Integrated Circuit [ASIC] can be obtained by FPGA based implementation of image processing algorithm. Comparative analysis using software and hardware is to be done.**

Index Terms—**XSG, ASIC, FPGA, Image Processing Algorithms.**

I. INTRODUCTION

In the last few years, the field of image processing has undergone a rapid evolution. Image processing has varied applications in medical field, computer vision, digital photography, Traffic load computation etc .The main objective of image processing is to improve the quality of the images for human interpretation .Image quality can be enhanced by creating image processing algorithms using Xilinx system generator such as image negative, contrast stretching, image segmentation, edge detection etc. This paper particularly features on developing suitable method for rapid and efficient way to perform hardware implementation for some of Basic crucial image processing algorithms that can be used in simple application specific devices. With advancement in the VLSI technology hardware implementation has become an attractive alternative. Assigning complex computation tasks to hardware and exploiting the parallelism and pipelining features yield significant speedup in running times. Implementation of image processing on re-configurable hardware, enables rapid prototyping of complex algorithm and simplifies debugging and verification. In recent years FPGAs (Field Programmable Gate Array) have become superior platform, with high-speed parallel computing capacity. FPGA is a fine grained device with large number of Input Output Blocks (IOBs) and Configurable Logic Block (CLBs) and other logic elements.

FPGA is rich source of high speed multipliers, adder, and memory; in the design process as they can be directly called.

It is requirement of an efficient rapid prototyping system to develop an environment targeting the hardware design platform. The Xilinx System Generator environment allows for the Xilinx line of FPGAs to be interface directly with Simulink. In addition there are several cost effective development boards available on the market that can be utilized for the software design development phase. Xilinx System Generator (XSG) is an integrator design environment (IDE) for FPGAs, which uses Simulink, as a development environment, it is presenting in the form of block set. It has an integrated design flow, to move directly to the configuration file (*.bit) necessary for programming the FPGA. One of the most important features of Xilinx System Generator is possessed abstraction arithmetic, which is working with representation in fixed point with a precision arbitrary, including quantization and overflow. XSG automatically generates VHDL code and a draft of the ISE model being develop. Xilinx System Generator has created primarily to deal with complex Digital signal processing (DSP) applications. The blocks in Xilinx System Generator operate with Boolean values or arbitrary values in fixed point, for a better approach to hardware implementation. The connection between blocks, Xilinx system generator and Simulink Blocks are gateway blocks.

II. LITERATURE REVIEW

An extensive work is done in the field of edge detection for real time image processing. In recent years, in September 2009[8] the algorithm implementation for FPGA based design for various applications using edge detection was proposed. The hardware processing on an FPGA allows the capture and online processing in real time on the same chip. External memory module can be saved using memory from FPGA only. The rate achieved by PC is greater than FPGA because PC clock is much higher than FPGA. In another paper [7] improvement was done by using block RAM and IO interface on the FPGA. After this in 2010 the improvement [6] was done by implementing the hardware circuit of the algorithm on ISE 9.1 and simulated in Model sim 6.0.It gave high precision edge. When the system is validated, it indicates that the video image edge detection system can detect high precision edge. This paper realizesa hardware based video edge detection system on FPGA mainly by the completion of the system IP core. Next stage was achieved by using A Moving Window Architecture in MAY 2012[5] which performs the more computational complex operations of edge detection algorithm like non maximum suppression and double thresholding. By using this design results are stored within the FPGA and eliminates the needs of large memory buffer. Improvement over this was done using FPGA Accelerated object detection using edge information. This paper proposes an FPGA based object detection that utilizes edge information to reduces the search space involved in object detection.Performance speed up and energy saving rating compared to traditional Moving

Window approach. The proposed architecture in this paper [4] are providing a platform for real time algorithm on application specific hardware with higher performance than programmable digital signal processors. In 2011 edge detection algorithm [3] for prewitt and sobel was presented and compared their result. After that A methodology [2] for real time architecture for Edge detection using sobel filter for image processing using Xilinx System Generator. In this approach a Virtex 5 FPGA kit is used to achieve higher performance than previous methods. In 2013 Paulo possa presented [1]a new flexible architecture for image and video processing with reduced latency and memory requirements supporting a variable input resolution. The proposed architecture is optimized for feature extraction such as the canny edge detector and the Harris corner detector. Also algorithm simplications are employed to reduce mathematical implementations on an FPGA based modules. It has clear advantage in low level power applications, low latency and portability are required.

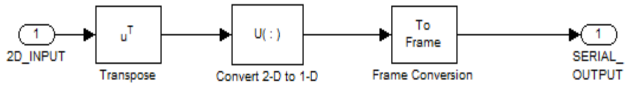
III. PROPOSED WORK

A. *Software implementation*

Image pre and post processing blocks

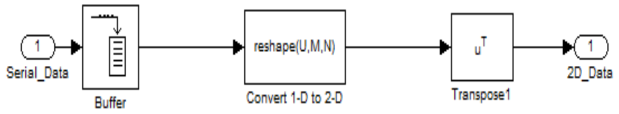
1. Initially the original digital image which may be color is first converted into Grey scale image using MATLAB.
2. This 2-D image is then converted into 1-D for further processing. This 1-D data stream is given to Frame conversion block.

Pre Processing blocks:

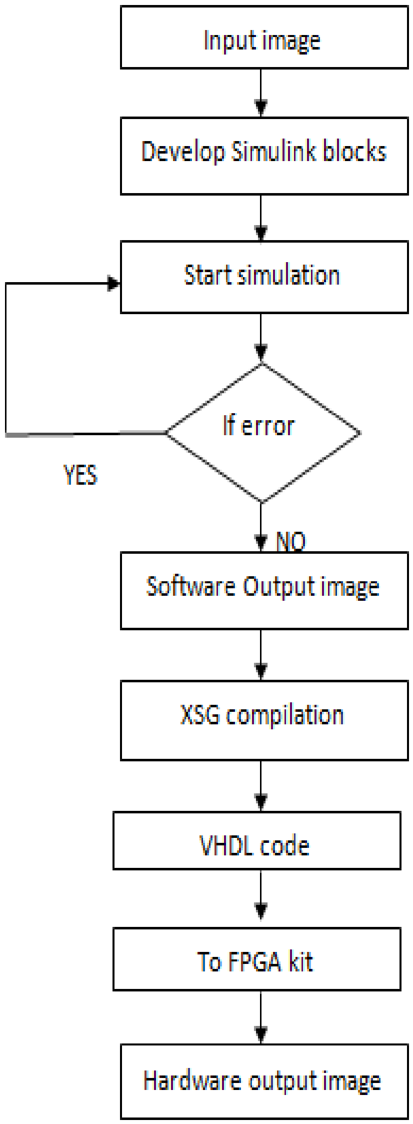


3. At this point all the image processing algorithms are done using XSG.
4. The 1-D output of XSG block set are given to image post processing blocks to obtain original 2-D image with its required features.

Post Processing Blocks:

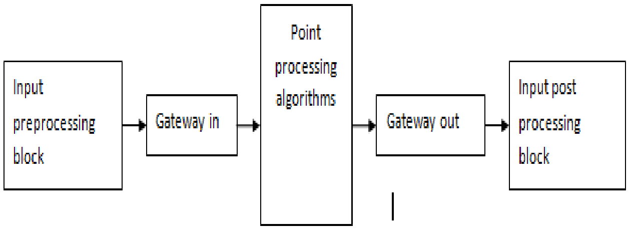


B. *Flowchart*



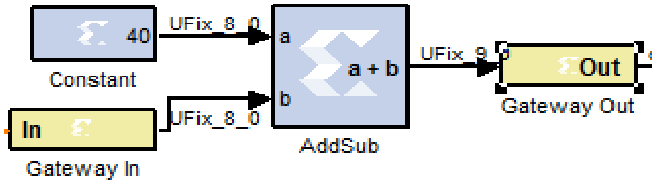
IV. IMPLEMETATION

To implement image processing algorithms Xilinx system generator is used because of its ability to generate HDL code. Various algorithms like negative generation, image enhancement, contrast stretching, image thresholding, edge detection are implemented using XSG. Before going to implement algorithms, Gateway In and Gateway Out blocks are connected in between preprocessing and post-processing blocks. These blocks act as input and output to the Xilinx portion of Simulink design.


Input preprocessing block
Gateway in
Point processing algorithms
Gateway out
Input post processing block

A. *Algorithm for Arithmetic Operations*

The arithmetic operations are addition, subtraction, division and multiplication of a pixels by a constant value. Addition and subtraction changes the brightness of the image.Fig. 1 shows the XSG blocks involved while adding and subtracting 40 from the image andshows input and output respectively. Similarly, multiplication and division by different values can adjust the contrast of the images.



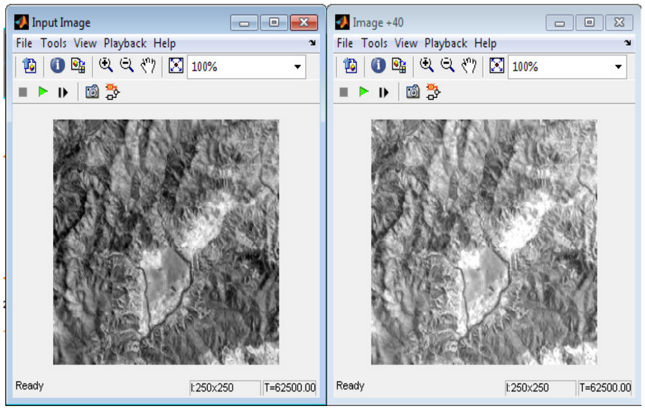
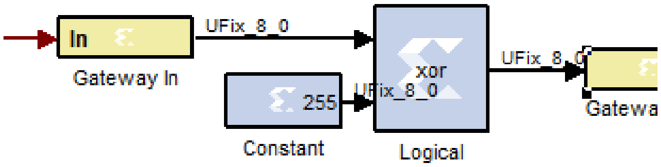


Figure1

B. *Algorithm for Logical Operation*

The Exclusive OR function sets bits that are the same in each operand to 0 and bits that are different to 1. All pixels of a certain value can be found by applying XOR function. Almost a negative image produced by XOR with 255 to the image is shown in fig. 2.



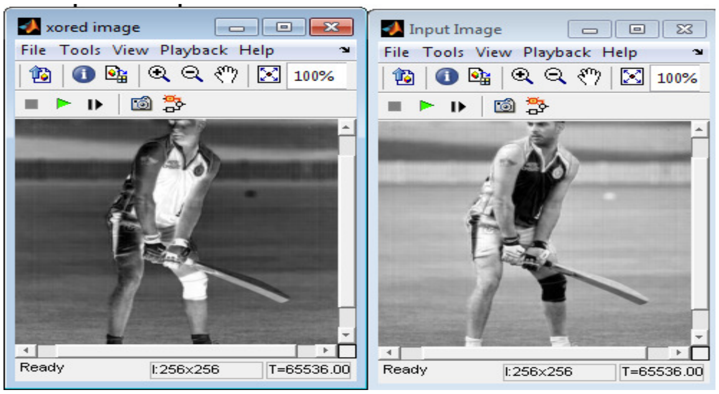
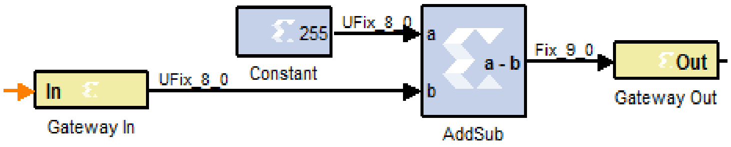


Figure.2

C. *Algorithm for Image Negative*

The algorithm for gray scale image negative is given in Fig.3.Here addsub block simply subtracts the constant 255 from pixel value to generate the negation of an input image.



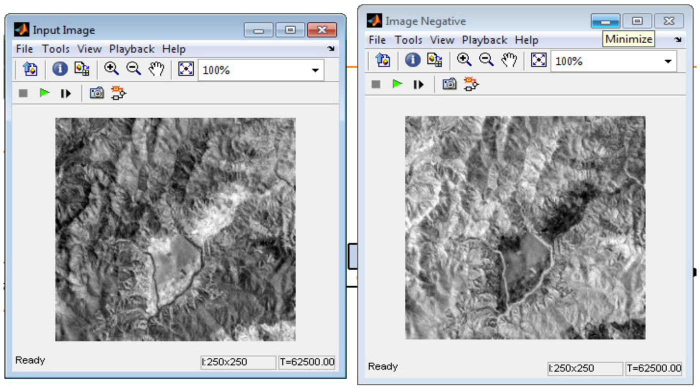


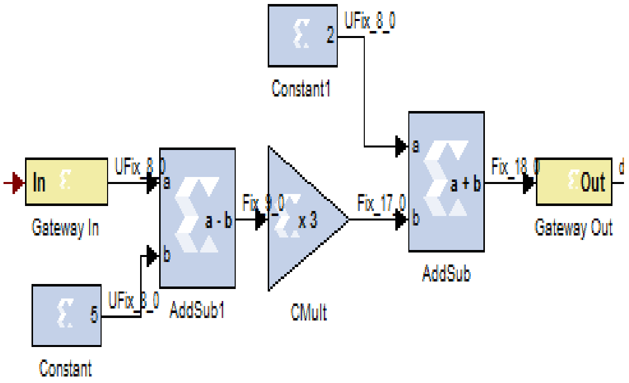
Figure.3

D. *Histogram Stretching (Contrast Stretching)*

The contrast of an image is its distribution of light and dark pixels. To stretch a histogram, contrast stretching is applied to an image to fill the full dynamic range of the image. We can stretch out the gray levels in the center of the range by applying piecewise linear function according to the equation.

new pixel = (12/4) (old pixel-5) + 2

where new pixel is its result after the transformation. Fig.4 shows the XSG blocks involved and input and output blocks after performing histogram stretching.



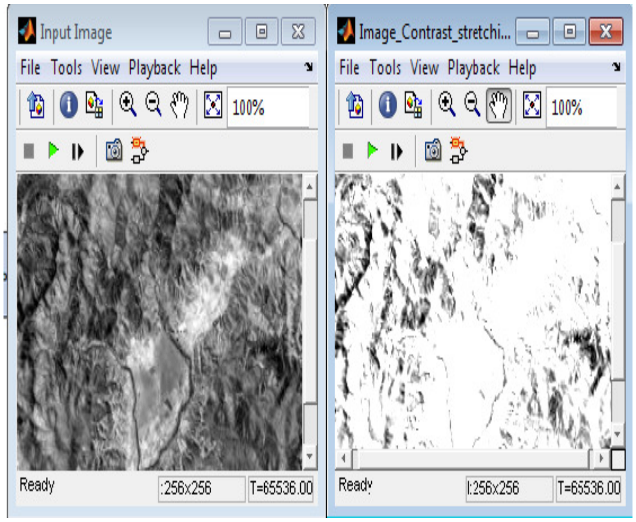
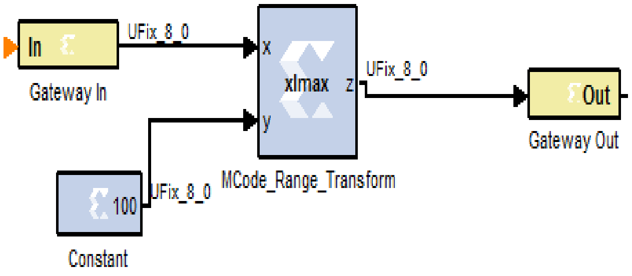

Input Image
File Tools View Playback Help
100%
Ready
256x256
T=65536.00
Image_Contrast_stretchi...
File Tools View Playback Help
100%
Ready
256x256
T=65536.00

Figure.4

E. .Image segmentation using threshold

Image segmentation can be used to separate pixels associated with objects of interestfrom the image background. This is an important step in many imaging applicationsof automated analysis and robotics.Fig.5 shows the XSG blocks involved and input and output blocks after performing image segmentation.



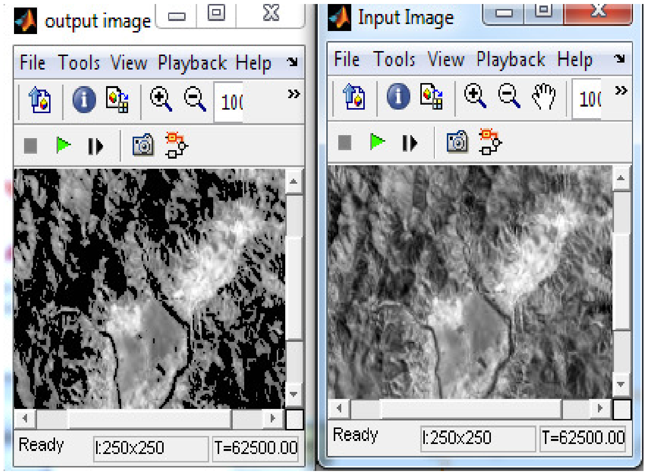


Figure.5

V. Hardware Software Co-Simulation

JTAG co-simulation is approach used for hardware implementation of the design. In this approach normal simulink blocks are executed in the matlab environment that generates operation desired. While JTAG co-simulation block load bit stream file generated (*.bit) in the FPGA in loop. And further data is accessed clock by clock basis that can be processed further. Once the data is accessed, filtering action is performed onto the image pixels. Thus once the JTAG co-simulation block is encountered in the loop processing of data is carried out in FPGAhardware. Thus desired results are generated that are again made available in matlabenvironment via JTAG cable.

1. Double click on the System Generator block. A dialog box will show up.
2. This dialog box allows to select the type of the hardware generated using system generator
3. If the board is supported it will appear and it is desired to follow the steps below. If the board is not supported then select new compilation target.
4. It is required to specify details regarding to the hardware board used.

Conclusion

Real time applications are always on hardware platform rather than software. In this paper low level digital image processing technique which are image negative, contrast stretching, arithmetical and image segmentation, edge detection algorithm over software and hardware platform is proposed. XSG tool of MATLAB provides simpler and efficient way of FPGA programming with minimum resource and delay. It provides friendly graphic interface. The Simulink model obtained here can be easily combined with other models. All the essential features required for real time image processing which are parallelism, speed, area minimization are obtained using XSG.

REFERENCES

[1] P R possa ,s a mahmoudi,cvalderrama, p mannback, "A multi resolution fpga based archtecthure for real time edge and corner detection Algorithm", IEEE transactions on computers, Jan 2013

[2] Yahid Said, TaoufikSaidani, FethiSmach and Mohamed Atri, Real Time Hardware Co-simulation of Edge Detection for Video Processing System , IEEE 2012.

[3] A.Amaricai,O.Boncalo,M.Iordate,B.Marinescu A Moving Window Architecture for HW/SW codesign Based Canny Edge Detection for FPGA, International

Conference On Microelectronics,13-16 MAY 2012 (MIEL 2012

[4] R. Harinarayan, R. Pannereselvam, M. Mubarak Ali, D. Tripathi, "FeatureExtraction of Digital Aerial Images by FPGA based implementation of edgedetection algorithms", Proceeding of ICETECT, Vol. 29, 2011, pp. 229-233.

[5] ChristosKykou,ChristosTtoffis and TheocharisTheocharides,FPGA-ACCELERATE OBJECT DETECTION USING EDGE INFORMATION, International Conference on Field Programmable Logic and Applications,2011

[6] Jinchengwu, "Design and Implementation of video image edge dtetection system based on FPGA"(CISP 2010).

[7] Mahendra ,Ajith, Ranga "FPGA based compact and flexible architecture for real time embedded vision system"(ICIIS 2009).

[8] Alfredo Gardel Vicente, Ignacio Bravo Munoz, Embedded Vision Modules for Tracking and Counting People, IEEE transaction on instrumentation and measurement, vol.58 , no 9, September 2009.

[9] www.Xilinx.com