HARDWARE SOFTWARE CO-SIMULATION FOR TRAFFIC LOAD COMPUTATION USING MATLAB SIMULINK MODEL BLOCKSET

ADHYANA GUPTA[1]

[1]DEPARTMENT OF INFORMATION TECHNOLOGY, BANASTHALI UNIVERSITY, JAIPUR, RAJASTHAN
adhyanagupta@gmail.com

ABSTRACT

Due to increase in number of vehicles, Traffic is a major problem faced in urban areas throughout the world. This document presents a newly developed Matlab Simulink model to compute traffic load for real time traffic signal control. Signal processing, video and image processing and Xilinx Blockset have been extensively used for traffic load computation. The approach used is Edge detection operation, wherein, Edges are extracted to identify the number of vehicles. The developed model computes the results with greater degrees of accuracy and is capable of being used to set the green signal duration so as to release the traffic dynamically on traffic junctions.

Xilinx System Generator (XSG) provides Simulink Blockset for several hardware operations that could be implemented on various Xilinx Field programmable gate arrays (FPGAs). The method described in this paper involves object feature identification and detection. Xilinx System Generator provides some blocks to transform data provided from the software side of the simulation environment to the hardware side. In our case it is MATLAB Simulink to System Generator blocks. This is an important concept to understand in the design process using Xilinx System Generator. The Xilinx System Generator, embedded in MATLAB Simulink is used to program the model and then test on the FPGA board using the properties of hardware co-simulation tools.

KEYWORDS

Vehicle detection, Image processing, FPGA, Xilinx System Generator, Edge Detection.

1. INTRODUCTION

This document primarily aims at the new technique of video image processing and Xilinx tool used to detect the traffic load in order to compute green signal duration for the release of that

traffic, making the implementation of real time traffic signal control possible reducing the congestion as well as the waiting time of all the users on the road.

The Video and Image Processing Blockset and Xilinx Blockset contains block that perform the Edge Detection. Edge detection is a technique for obtaining image features for object tracking and recognition. Hence, Edge Detection method can be performed on traffic images captured using CCTV camera, installed at the desired intersection point. The Edge Detection block finds edges in an image. This block finds the edges in the image.

DOI : 10.5121/ijcsity.2013.1201

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer (i.e. designer) after manufacturing and hence called field programmable. The FPGA configuration is generally specified using a hardware description language (HDL), similarly as used for an application-specific integrated circuit (ASIC). Presently, FPGAs have large resources of Look up tables (LUTs), logic gates and RAM blocks to implement any complex digital computations. The use of FPGA in image processing has a large impact on image or video processing Applications. The number of vehicles on the road has been detected using Matlab Simulink Model Blocksets. A Simulink model has been developed using different image processing blocksets from MATLAB. In developing the Simulink model, the Video and Image Processing Blockset software tool in MATLAB has been used. The Video and Image Processing Blockset software is the tool for processing images and videos in the Simulink environment, which can improve and modify the Image and video characteristics.

The remainder of the paper is organized as follows. Section 2 briefly presents the related work. Section 3 briefly describes MATLAB and Video and Image Processing and Xilinx Blockset, being used for the development of this model. Section 4 presents the experimental model and results. Section 6 draws the conclusion.

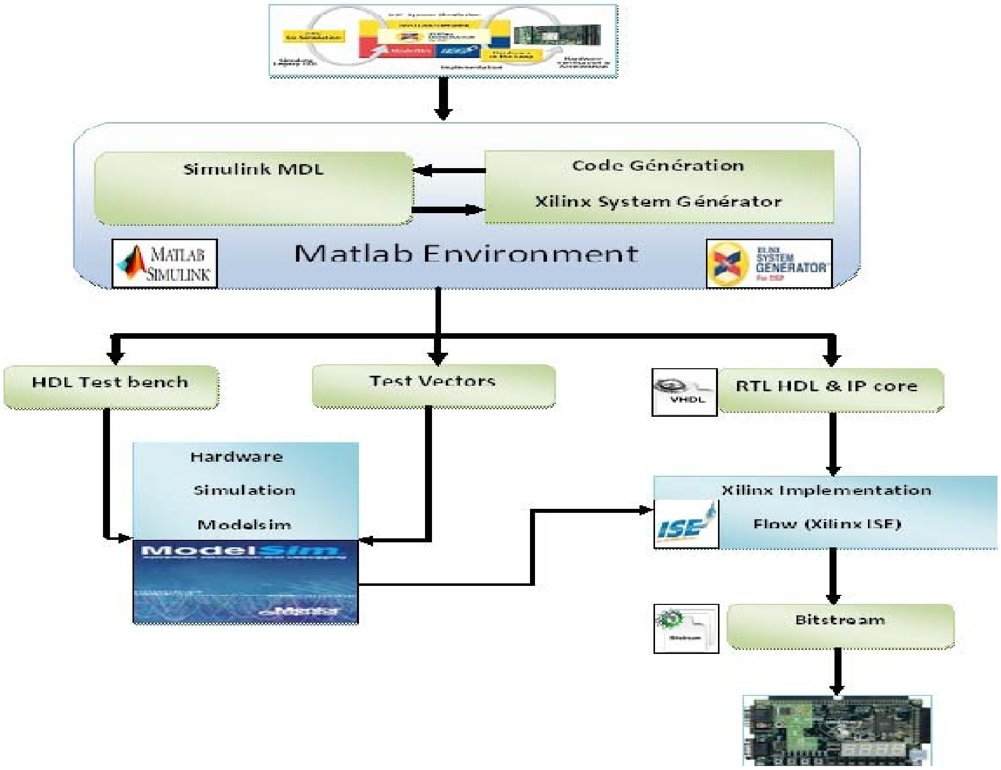


Figure 1. Design methodology with Xilinx System Generator

2. RELATED WORK

Suthar et al. [1] this paper presented the basics of image processing in model based approach and demonstrated some of the image processing application which is done under SIMULINK and implemented using Xilinx System Generator (XSG). The Xilinx System Generator tool is a new tool in image processing, and friendly design environment for processing, because we can make processing units using Xilinx block sets. This tool also supports software simulation, but it is well known for its capability to synthesize on FPGAs hardware in parallelism, robust and with speed, these features are essentials in image processing applications.

Chikkali [2] discussed Histogram is used for automatically determining the threshold for different region in image. Segmentation is done with the help of histogram and here we extract features with the help of intelligent computer.

Ali et al. [3] concluded that Xilinx system generator is a very useful tool for developing computer vision based algorithms. Image processing is used to extract picture information and then modify by changing their structure. They focused in the processing of pixel to pixel of an image and modification of pixel neighbourhoods and these transformations can be applied to the whole image or only a partial region. The need to process an image in real time, leading to the implementation on hardware, which offers parallelism and thus significantly, reduces the processing time. In this paper we have shown how to read an Image and enhance its characteristics either a gray scale or a color Image and then we have taken two test color images for the color image negative to give better idea.

Elamaran et al. [4] discussed in this paper the real time image processing algorithm that is implemented on FPGA .Implementation of this algorithm is having a advantage of using large memory and embedded multipliers that is available on FPGAs. Point processes are the simplest and basic image processing operations. Applications like background estimation in videos, image filtering both in spatial and frequency domains and digital image watermarking applications etc can be easily designed using Xilinx system generator.

Chandrashekar et al. [5] in this paper discussed enhancement of digital image to exact true image, which is very useful in many applications and known as image enhancement. Human intervention is always needed in image processing and it’s hard to do all automatically. FPGA synthesized results are compared with Matlab simulations experiments and comparisons to histogram equalization are conducted. The transformation is applied to perform both a nonlinear and a shape preserving stretch of the image histogram. This paper deals with hardware implementation of SMQT is applied for automatic image enhancement. This image enhancement results are compared with the histogram equalization.

Acharya et al. [6] discussed FPGA based hardware design for enhancement of color image and gray scale image in image and video processing. The approach used is known as adaptive histogram equalization which works very effectively for image captured under extremely dark environment as well as non-uniform lighting environment where bright regions are kept unaffected and dark object in bright background. This paper shows that reconfigurable FPGAs have both real time and parallel computing expectations for the enhancement process in Images.

Gribbon et al. [7] used FPGA as platforms for implementing real time image processing applications to exploit spatial and temporal parallelism. High level languages and compilers which automatically extract parallelism from the code are not directly compatible to hardware. Low level mapping can overcome the software mind set but they must now deal more closely with the constraints i.e. are labour intensive and are rarely reusable.

Devika et al. [8] explain FPGA is widely used in implementation of real time algorithms suited to video image processing applications. The FPGA provide basic digital blocks with flexible interconnections to achieve realization of high speed digital hardware. The FPGA consists of a system of logic blocks, such as LUTs, gates or flip flops and some amount of memory. The image is then transferred from PC to FPGA board using universal Asynchronous receiver

/transmitter (UART) serial communication. After required filtering, the result will be transferred back to PC. In PC both the results will be compared and validated.

Thakur et al. [9] explain Tonsillitis, Tumor and many more skin diseases can be detected in its early state and can be cured. Image segmentation is the processes of partitioning a digital image into multiple segments that is sets of pixels. The segmented images are more meaningful and easier to analyze. A new idea for efficient Gabor filter design with improve data transfer rate, efficient noise reduction, less power consumption and reduced memory usage is proposed.Systems provide both highly accurate and extremely fast processing of huge amount of image data.

Christe et al. [10] discussed in this paper of digital image at which the image brightness changes sharply or has discontinuities is named as Edge Detection. The points at which image brightness changes sharply or has discontinuities are typically organized into a set of curved line segments known as edges. . In other words, an edge is the boundary between an object and the background. Focuses on processing an image pixel by pixel and in modification of pixel neighbourhoods and the transformation that can be applied to the whole image or only on a partial region.

Draper et al. [11] explain that computers keep getting faster and faster but new IP cores are needed to satisfy the newly arrived applications. Examples of current high- demand applications include real-time video stream encoding and decoding, real-time biometric namely face, retina, and/or fingerprint recognition, and military aerial and satellite surveillance applications. To meet the demands new image processing techniques are needed. Simple image operators are faster on FPGAs because of their greater Input Output bandwidth to local memory, although this speed-up is not that high (a factor of ten or less). More complex tasks needs larger speed-ups, up to a factor of 800 in one experiment, which can be attained by using parallelism within FPGAs and the strengths of an optimizing compiler.

Manan [12] this paper explains the importance of digital image processing and the significance of their implementations on hardware to achieve better performance, particularly this work addresses implementation of image processing algorithms like median filter, morphological operations, convolution and smoothing operation and edge detection on FPGA using VHDL language.

3. EXPERIMENTAL ENVIRONMENT

3.1. Matlab Introduction

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

3.2. Video and Image Processing Algorithms used in adjunct with Matlab

Simulink environment provide platform for model-Based Design out of a user-Friendly block diagram environment. (Figure 2, Figure 3, Figure 4 and Figure 5)

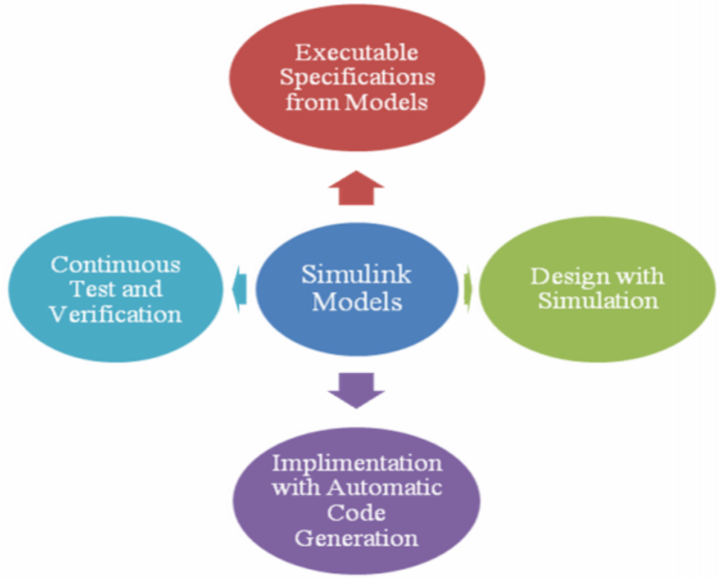

Executable Specifications from Models
Continuous Test and Verification
Simulink Models
Design with Simulation
Implimentation with Automatic Code Generation

Figure 2. Simulink Model

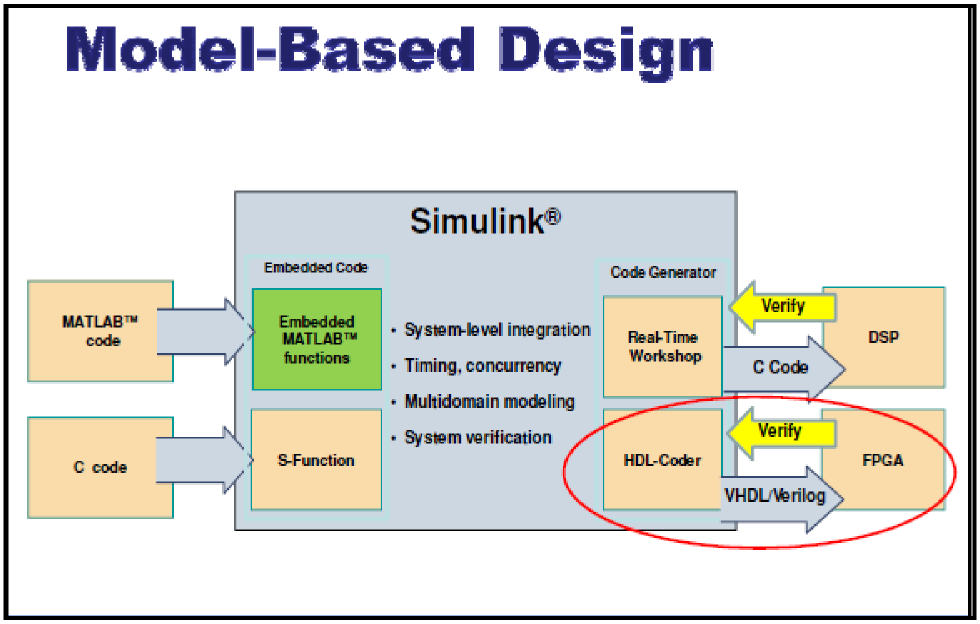


Figure 3. Model-Based Design

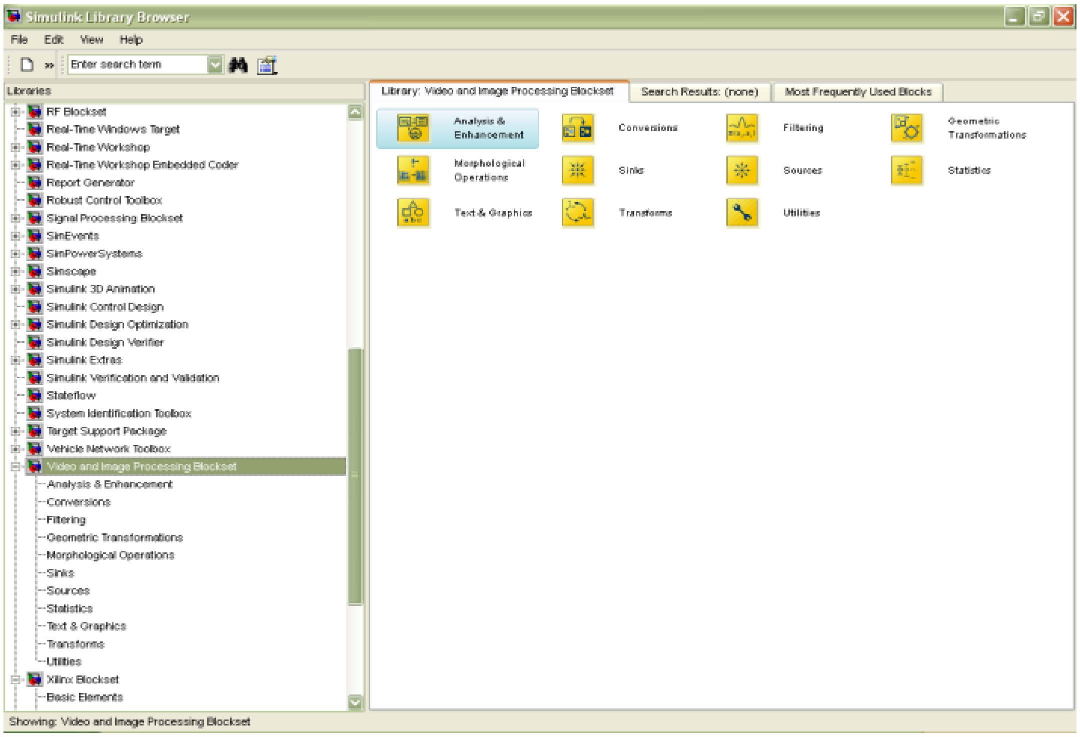


Figure 4. Video and Image Processing Library Browser

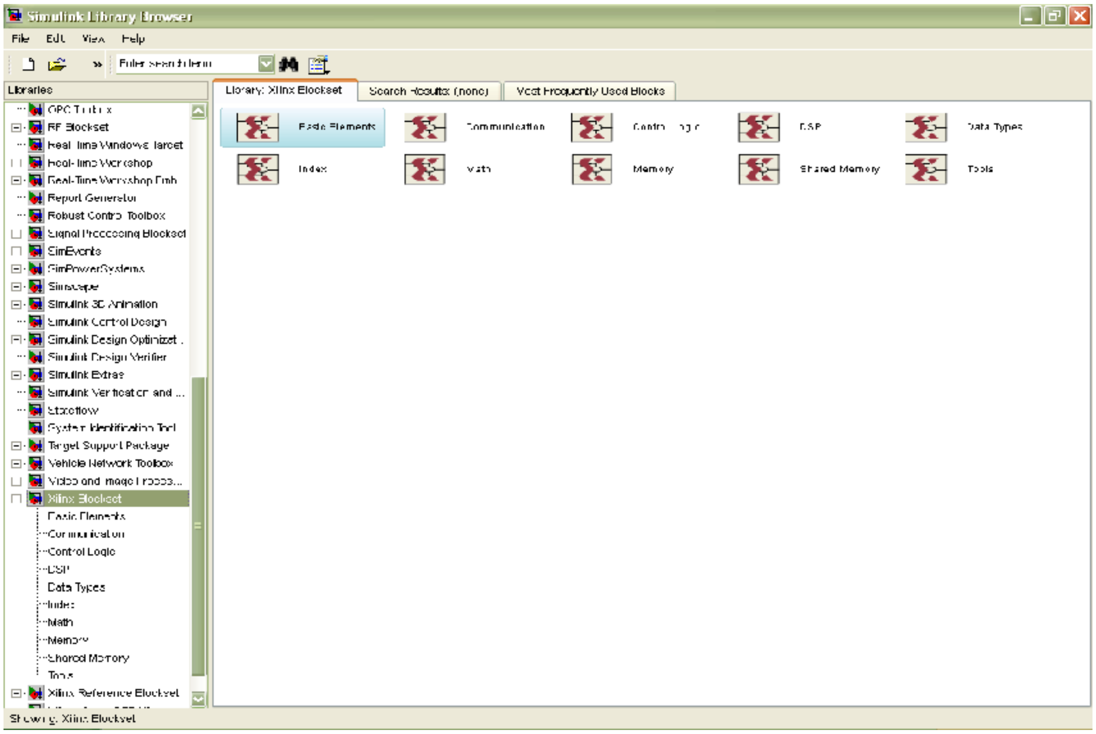


Figure 5. Xilinx Blockset Library Browser

3.3. Edge Detection

A set of mathematical methods for identifying points in a digital image at which the image brightness changes sharply or has discontinuities is named as Edge Detection. The points at which image brightness changes sharply or has discontinuities are typically organized into a set of curved line segments known as edges. This is generally done by detecting the maximal value of gradient such as Sobel, Roberts, Prewitt, Canny and so on, all of which are the known classical edge detectors.

4. System design and architecture

The Image edge detection is very powerful and used method in the field of Image processing Applications. Edge detection is used in image processing Applications, machine vision applications and computer vision applications, mainly in the areas of feature detection and feature extraction.

1. Import the target image from MATLAB and view the image. (Figure 6)

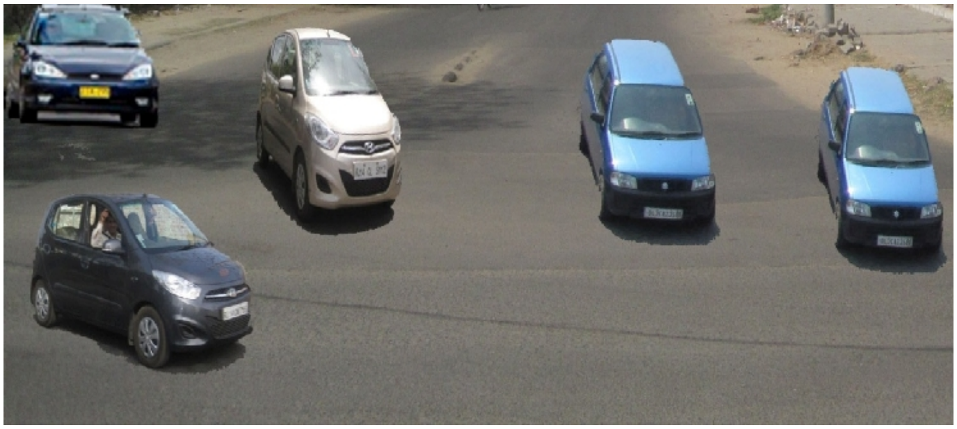


Figure 6. Original image

2. Create Model using Xilinx Blockset, signal processing Blockset, Simulink, video and image processing Blockset in Matlab Environment for Edge Detection. (Figure 7)

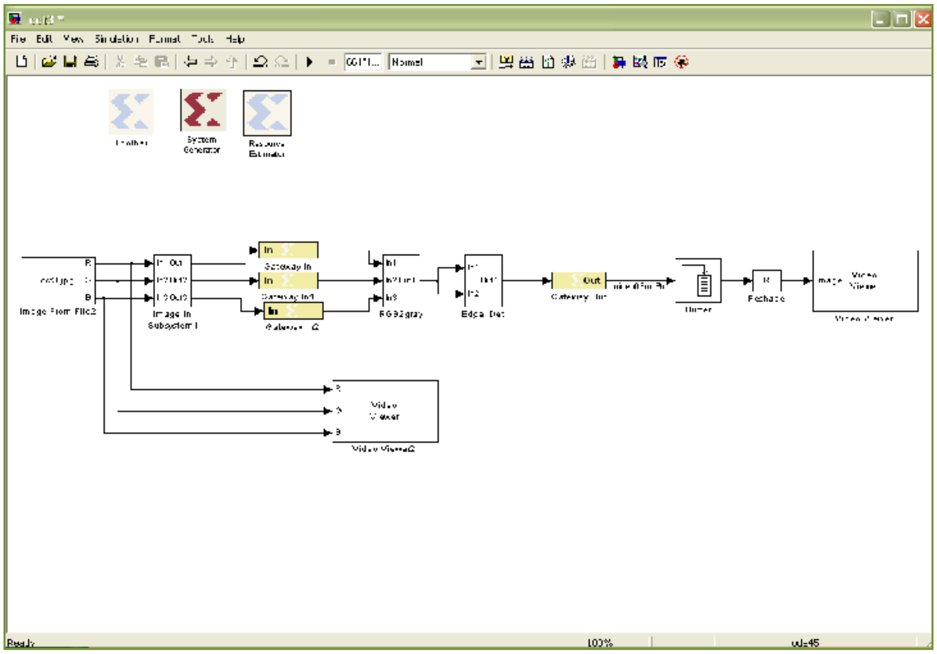


Figure 7. Edge Detection model using Xilinx, signal processing, Simulink, video and image processing block sets

3. Create RGB to Gray Conversion Blockset using Xilinx Blockset. (Figure 8)

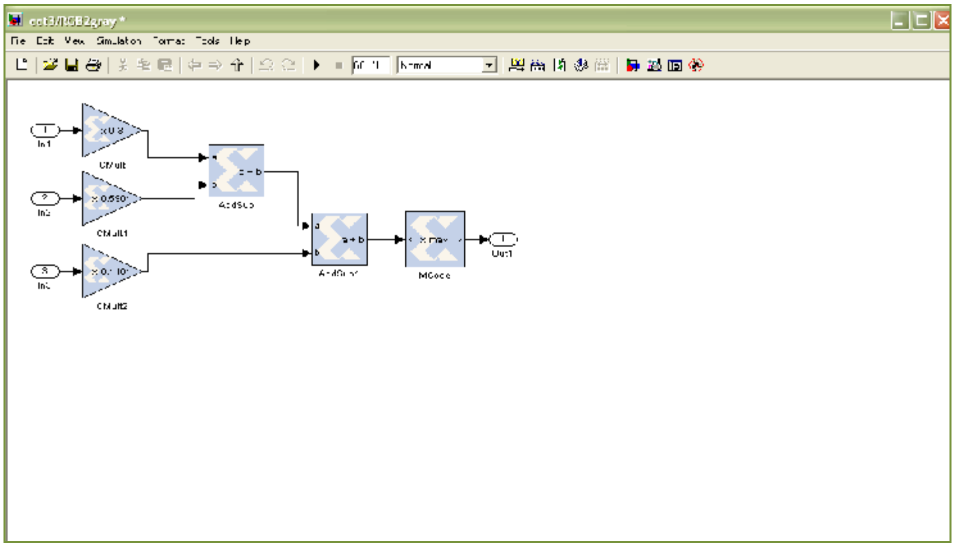


Figure 8. RGB to Gray Conversion Blockset using Xilinx Blockset

4. Create Edge detection model using Xilinx Blockset. (Figure 9)

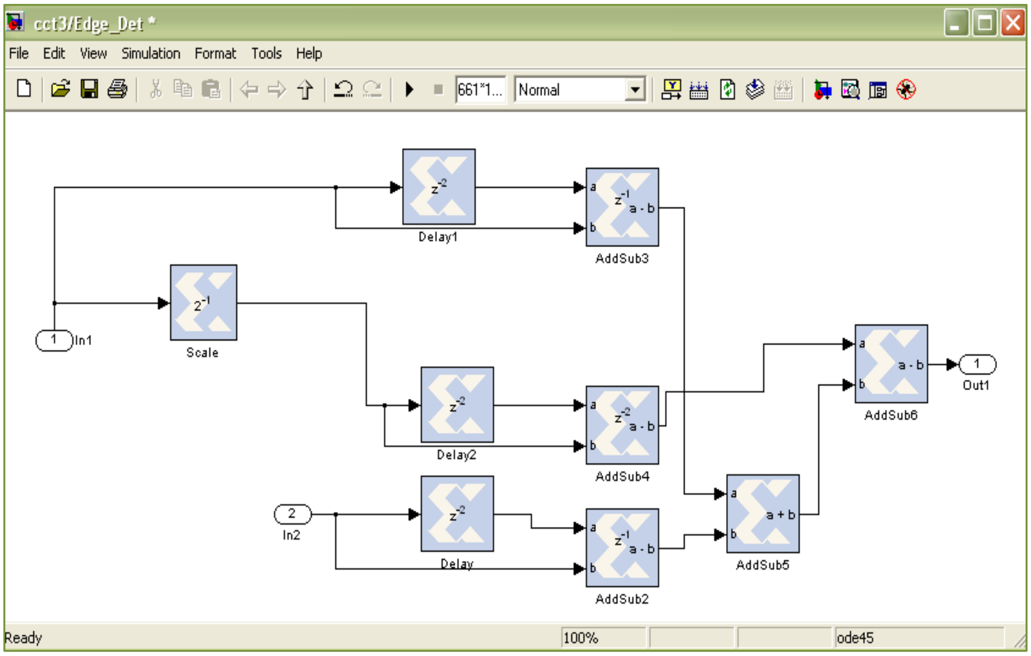
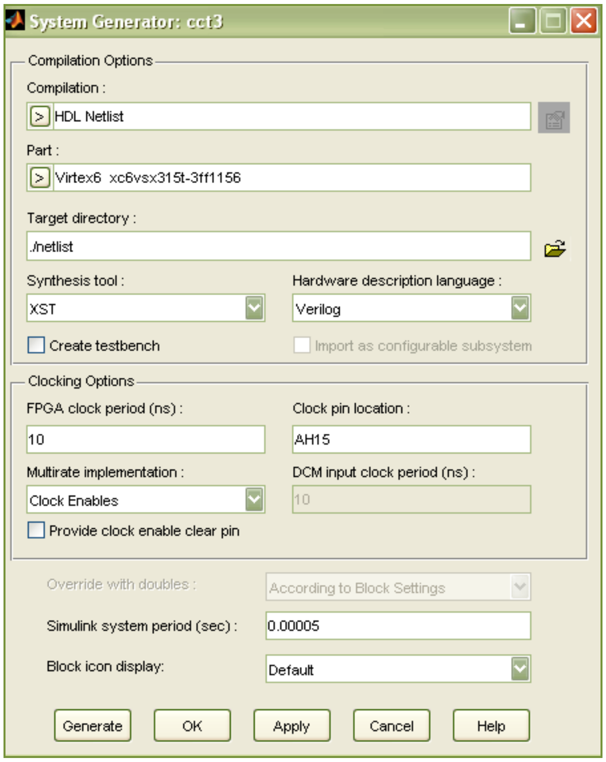


Figure 9. Edge detection model using Xilinx Blockset

5. Then compile the model using Xilinx system generator. (Figure 10 and Figure 11)


System Generator: cct3

Compilation Options

Compilation : HDL Netlist

Part : Virtex6 xc6vsx315t-3ff1156

Target directory : ./netlist

Synthesis tool : XST

Hardware description language : Verilog

□ Create testbench

□ Import as configurable subsystem

Clocking Options

FPGA clock period (ns) : 10

Clock pin location : AH15

Multirate implementation : Clock Enables

DCM input clock period (ns) : 10

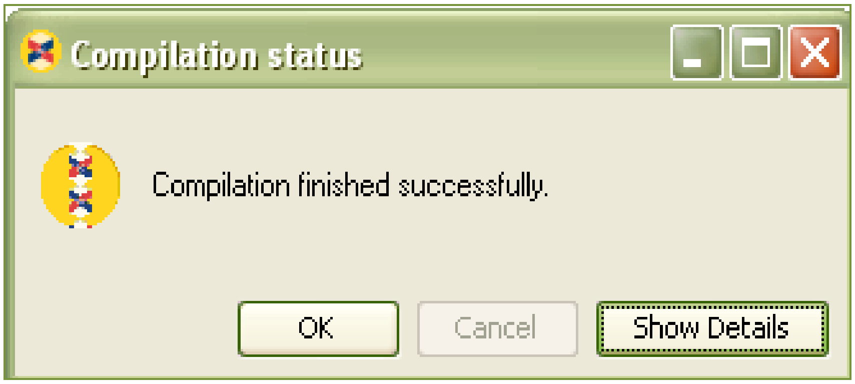
□ Provide clock enable clear pin

Override with doubles : According to Block Settings

Simulink system period (sec) : 0.00005

Block icon display: Default

Figure 10. System generator



Compilation status

Compilation finished successfully.

Figure 11. Compilation status

Result:

6. Run the model after Xilinx system generator compilation. (Figure 12,13 and 14)

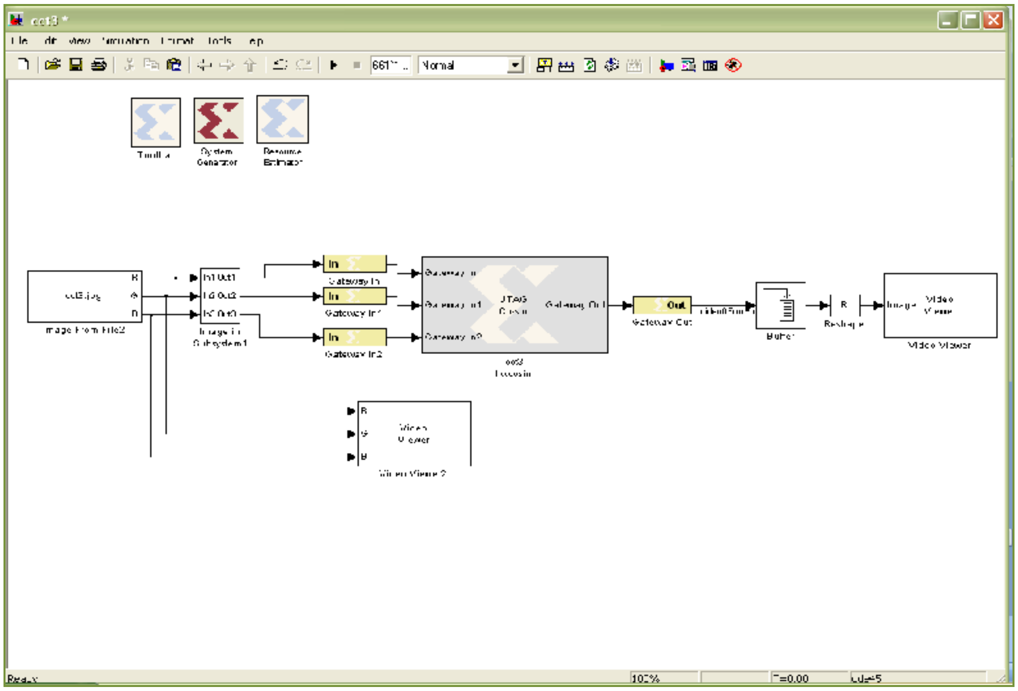


Figure 12. Edge Detection of Vehicles (Final Result)

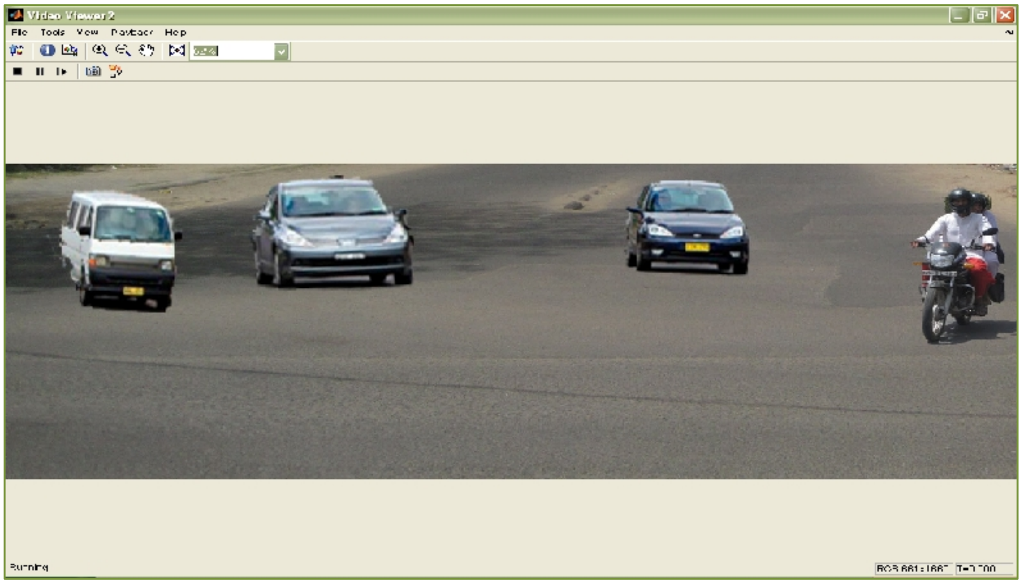


Figure 13. Original Image

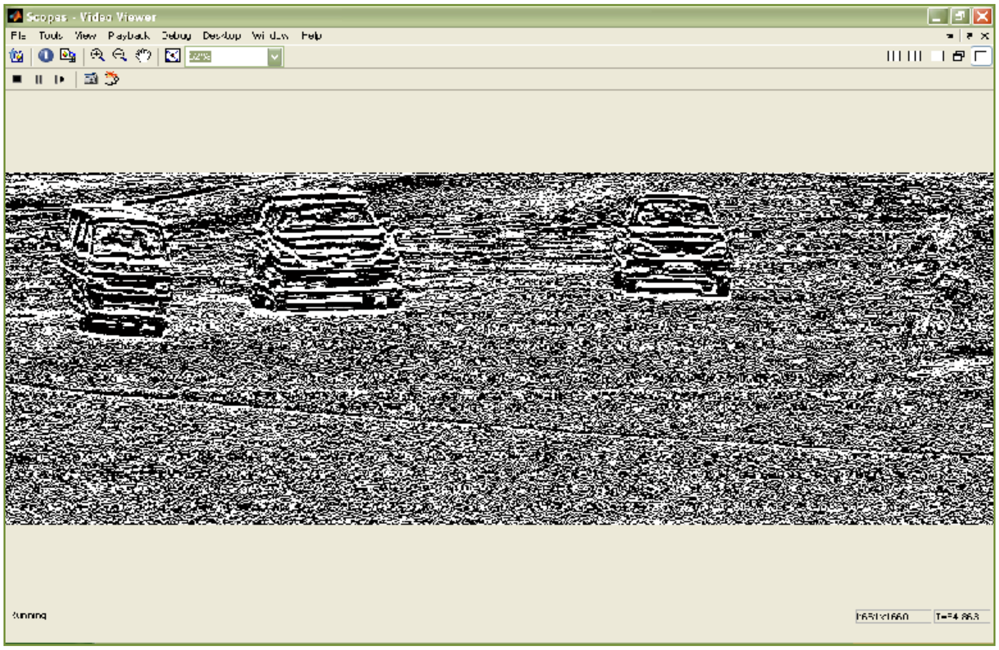


Figure 14. Edge Detection Image

5. CONCLUSION

Xilinx system generator provides a simpler and useful tool for developing computer vision based algorithms. It is a more suitable and beneficial option if compared to designing using VHDL or Verilog hardware description languages (HDLs). The result shows that the strength of applying Edge detection which makes it more sensitive in detecting edges in any image and hence more edges are detected using this method. In this paper, Edge detection technique is used for traffic load computation which proved very useful in detecting the edges in any traffic image. Edges serve to simplify the analysis of images. The developed Simulink model is reliable & can perform Edge detection for vehicles on roads.

This Model was compiled successfully in the SIMULINK environment. The Xilinx System Generator, embedded in MATLAB Simulink was used to program the model and then test on the FPGA board using the properties of hardware co-simulation tools. The proposed algorithm uses the image processing features of the MATLAB software and incorporates the time efficiency of hardware. This Simulink model will be useful to detect the traffic on road. Hence, the algorithm described, proves to be an efficient solution for real-time traffic load computation.

REFERENCES

[1] A.C. Suthar, M. Vayada, C.B. Patel, G.R. Kulkarni, "Hardware Software co-simulation for Image Processing Applications", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No.2, March 2012, ISSN (Online):1694-0814

[2] P.S. Chikkali, "FPGA based Image Edge Detection and Segmentation", Vol. No. 9, Issue No. 2, pp187 – 192.

[3] S.M. Ali, Naveen, P.A. Khayum, "FPGA Based Design and Implementation of Image Architecture using XILINX System Generator ", IJCAE, Vol. No. 3 Issue 1, July 2012, pp132- 138.

[4] V. Elamaran, G. Rajkumar, "FPGA implementation of point processes using Xilinx system generator", 31st July 2012. Vol. 41 No.2, ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195 201.

[5] M. Chandrashekar, U.N. Kumar, K. S. Reddy, K.N. Raju, "FPGA Implementation of High Speed Infrared Image Enhancement", ISSN 0975 - 6450 Vol. 1 No. 3 (2009) , pp 279–285.

[6] A. Acharya, R. Mehra, V.S. Takher," FPGA Based Non Uniform Illumination Correction in Image Processing Applications", Vol. 2, pp 349-358, http://www.ijcta.com/documents/volumes/vol2issue2/ijcta2011020219.pdf

[7] K.T. Gribbon, D.G. Bailey, C.T. Johnston, "Design Patterns for Image Processing Algorithm Development on FPGAs", TENCON2005, pp. 1-6, November 21-24, 2005, doi: 10.1109/TENCON.2005.301109.

[8] S.V. Devika, S.K. Khumuruddeen, Alekya, "Hardware implementation of Linear and Morphological Image Processing on FPGA", Vol. 2, Issue 1, Jan-Feb 2012, pp645-650.

[9] R.R. Thakur, S.R. Dixit, A.Y. Deshmukh, "VHDL Design for Image Segmentation using Gabor filter for Disease Detection", Vol.3 No.2, April 2012. http://connection.ebscohost.com/c/articles/82032456/vhdl-design-image-segmentation-using-gabor-filter-disease-detection

[10] S.A. Christe, M. Vignesh, A. Kandaswamy, "An efficient FPGA implementation of MRI image filtering and tumour Characterization using Xilinx system generator", Vol. 2 No. 4, December 2011. http://airccse.org/journal/vlsi/current2011.html

[11] Draper, B.A. Beveridge, J.R. Bohm , A.P.W. Ross, C. Chawathe , M., "Accelerated Image Processing on FPGAs", IEEE Transactions on Image Processing , Dec. 2003 Vol. 12, issue 12, pp1543-1551.

[12] A. Manan, "Implementation of Image Processing Algorithm on FPGA", Akgec Journal of Technology, Vol. 2, No.1

AUTHOR

Adhyana Gupta is an active researcher in the field of image processing, currently studying in M.Tech (IT) from Banasthali University, Rajasthan. She received M.Sc Degree in Computer Science from Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal in 2008.