iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** V **Month of publication:** May 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Hardware Co-Simulation of Sobel Edge Detection Using FPGA and System Generator

Sneha Moon[1], Prof Meena Chavan[2]
[1,2]*Department of Electronics BVUCOE Pune India*

Abstract*: *This paper implements an image processing algorithm applicable to Edge Detection for still image in a Xilinx FPGA using System Generator. We prefer sobel algorithm which is most reliable and gives us an efficient output. If we prefer to write HDL code for such algorithm in Xilinx FPGA then it's too bulky and time consuming. We design this system with use of Xilinx System Generator blocks. Its tool with a high- level graphical interface under Matlab environment Its Simulink based blocks which makes it very easy to handle with respect to other software for hardware description. In this paper we have presented new technique SimSH: Simulink Sw/Hw Co Design system. Introduced system gives a programmed way from a algorithm captured in Simulink to a heterogeneous implementation. Given an allotment and a mapping choice, the SimSH automatically synthesizes the Simulink model on heterogeneous target. SimSH also helps to detect underutilized bus and optimize Simulink allows user to concentrate

Keywords*: *Matlab, Xilinx System Generator, FPGA, Sobel Edge detection algorithm.

I. INTRODUCTION

The emerging market for video processing systems requires high-performance digital signal processing as well as low device costs appropriate for a volume application. Xilinx FPGA devices provide a platform with which to meet these two contrasting requirements. A Xilinx tool, the System Generator for DSP, offers an efficient and straightforward method for transitioning from a PC-based model in Simulink to a real-time FPGA based hardware implementation. The system model can be simulated in the Simulink environment. This higher abstraction level reduces the analysis and debugging time. For real hardware testing, Xilinx System Generator supports the possibility to perform hardware in-the-loop co-simulation. This methodology provides easier hardware verification and implementation compared to HDL based approach. The Simulink simulation and hardware-in-the loop approach presents a far more cost efficient solution than other methodologies. The ability to quickly and directly realize a control system design as a real-time embedded system greatly facilitates the design process. The goal of this project was to implement an image-processing

algorithm applicable to Edge Detection system in a Xilinx FPGA using System Generator for DSP, with a focus on achieving overall high performance, low cost and short development time. Xilinx System Generator is a DSP design tool from Xilinx that enables the use of the

Math works model-based design environment Simulink for FPGA design. It is a system-level modelling tool in which designs are captured in the DSP friendly Simulink modelling environment using a Xilinx specific block set. The advantages of using proposed model are as follows

A. Introducing a SimSH that provides an automatic path from Simulink to a heterogeneous platform, given PE allocation and mapping. The SimSH empowers algorithm developers rapidly synthesize the application avoiding tedious and errorprone manual implementation efforts.

B. The SimSH automatically inserts necessary communication and synchronization across PEs via Communication Refinement. The synthesized layered communication is influenced by the OSI standard to enable reusability and scalability over varying architectures.

C. A communication optimization is introduced which detects an underutilized bus, and increases efficiency through pack-/unpack to fully utilize the bus. We demonstrate the benefits using Sobel Edge Detection, and map it to a heterogeneous platform of Blackfin DSP and Xilinx FPGA. The results demonstrate significant benefits in terms of (a) rapid realization (within minutes), and (b) increased performance and energy efficiency (both 2.68x over SW implementation).

II. LITERATURE SURVEY

A lot of work [1]done on edge detection algorithm to detect edge of an object .On the basis of edge detection algorithm we can

improve quality of image for human interpretation. Image processing used in various field now days such as in medical application, for digital aerial image detection from satellite, for vehicle detection etc.

A. There are broadly three methods to detect edges

1) First order derivative (Gradient Method) Method. Example:

a) Robert Operator

b) Prewitt Operator

c) Sobel Operator

2) Second order derivative Method. Example:

a) Laplacian

b) Laplace of Gaussian

c) Difference of Gaussian

3) Optimal edge detection method.

a) *Canny edge detection.:* The derivative operators are used for image enhancement or to enhance the details present in the image and these derivatives operations can be used for detection of edges present in the image. In this paper [2] use System Generator tool in developing vehicle image processing edge detection algorithms which is developed by Xilinx based on MATLAB. Edge detection algorithm model and design are finished in MATLAB Simulink, preparation of top-level file in ISE 10.0 environment then achieve a System Generator functions and other modules instantiated. Import the hardware design which generate by System Generator into the paper, and then the paper should be simulated, synthesis, finally completed the hardware-based of the algorithm. And display the processing image through VGA.Simulink R2014a [3] also supports concurrent execution code generation. However, it does not specifically address communication optimization. Furthermore, Simulink only targets specific heterogeneous architectures (such as Zynq with single CPU and up to 2 FPGAs), while our work targets a general heterogeneous architecture. Different from the industry approach, SimSH reveals both design methods and usage. It allows users in the academic community to easily expand the tool to support other platforms. Synthesizing Simulink algorithm models to specifications has emerged in recent research. In [5,6], authors proposed a framework for software code generation from Simulink and validation on MPSoC architecture.

III. METHODLOGY

A. Xilinx System Generator

System Generator is part of the ISE® Design Suite and provides Xilinx DSP Block set 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 Block set. 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 Block set provides close integration with MATLAB Simulink that helps in co-simulating the FPGA module with pixel vector provided by MATLAB Simulink Blocks. The System Generator block defines which type of FPGA board will be used, as well as provide several additional options for clock speed, compilation type and analysis. With a library of over 90 DSP building blocks, System Generator allows for faster prototyping and design from a high-level programming stand point.

Some blocks such as the M-code and Black box allow for direct programming in MATLAB M-code, C code, and Verilog to simplify integration with existing projects or customized block behaviour. System Generator projects can also easily be placed directly onto the FPGA as an executable bit stream file as well as generating Verilog code for additional optimizations or integration with existing projects within the Xilinx ISE environment[4].

Our top level design was built by using XSG (Fig. 3). Inside the Edge Filter block is the Sobel Edge Detection designed with the steps are described in the section IV by Simulink block sets. We show the comparison hardware resources and power consumption in three types of platforms of FPGA (Table I). The power parameters are obviously rather different. The quiescent power of implement on Virtex 5 is 1441mW in total. On the other hand, these are only 181mW (Spartan3A) and 79mW (Zynq AP SoC) compare with the total power. We can see that the number of power which implements on Virtex 5 is greater ten times than Zynq-7000 AP SoC. Thus, the technology and architecture of the platform prove these evidences [8].

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

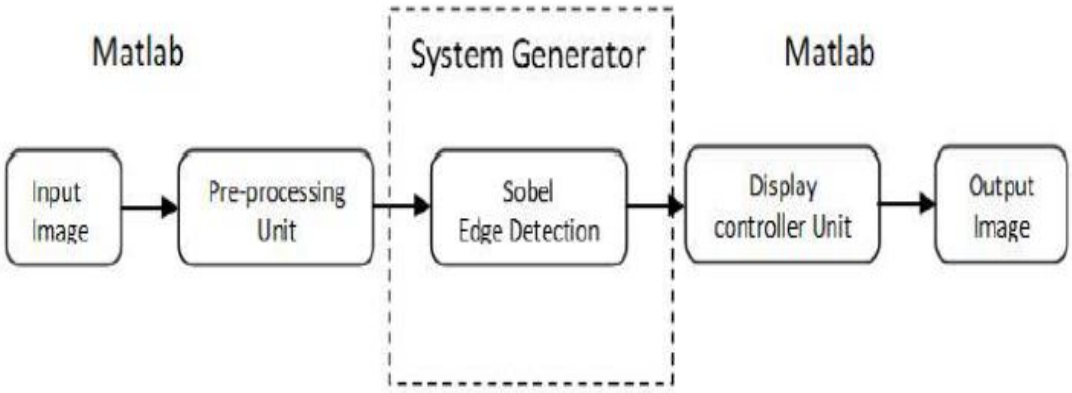


Figure 1: Sobel Edge Detection design flow based on Xilinx system generator

B. Sobel Edge Detection Algorithm

Sobel edge detection algorithms are the most commonly used techniques in image processing for edge detection [6]. In this paper 2 types of Sobel operators were used (horizontal, vertical). The operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The Sobel kernels are given by

$G_x =$, $G_y =$ (1)

Here the kernel Gx is sensitive to changes in the x direction, i.e., edges that run vertically, or have a vertical component. Similarly, the kernel Gy is sensitive to changes in y direction, i.e., edges that run horizontally, or have a horizontal component. The two gradients[12] computed at each pixel (Gx and Gy) by convolving with above two kernels can be regarded as the x and y components of gradient vector. This vector is oriented along the direction of change, normal to the direction in which the edge runs. Gradient magnitude and direction are given by:

G= (2)

An approximate magnitude is computed using:

= + (3)

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

=a (4)

C. Hardware Software Co-Design Framework

Input to SimSH system is Simulink specification model. Figure below shows detail architecture of SimSH model. Simulink model acts as input and guides the user in allocating and mapping blocks based on profiling. SimSH employs Algo2Spec to generate a SLDL specification model (in SpecC), and then profiles the specification using scprof. The profiler reports computation and traffic demands in terms of number of operations, individually for each operation and data type. The profiling exposes computational and communication hot spots of the application. Synthesis occurs in 3 phases: Front-end Synthesis, Communication Refinement, and Back-end Synthesis, yielding the SW/HW implementation.

In the Front-end Synthesis, the mapped specification model is split into hardware models and software models and then synthesized into software implementation in C/C++ and hardware implementation in Hardware Description Language (HDL). In this step, the functionality of all blocks in the model is synthesized for different PEs while the communication across the PEs is missing. To address that, Proxy is inserted in the model that encapsulates the cross-PE communication which will be further refined.

In the Communication Refinement, the Proxy is refined and realized following the OSI standard. In this work, the Proxy is comprised of 4 layers:

1) the application layer for the consistent interface,
2) the transport layer for synchronization,
3) the network layer for addressing and marshaling

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

4) the physical layer for interfacing with the physical bus.

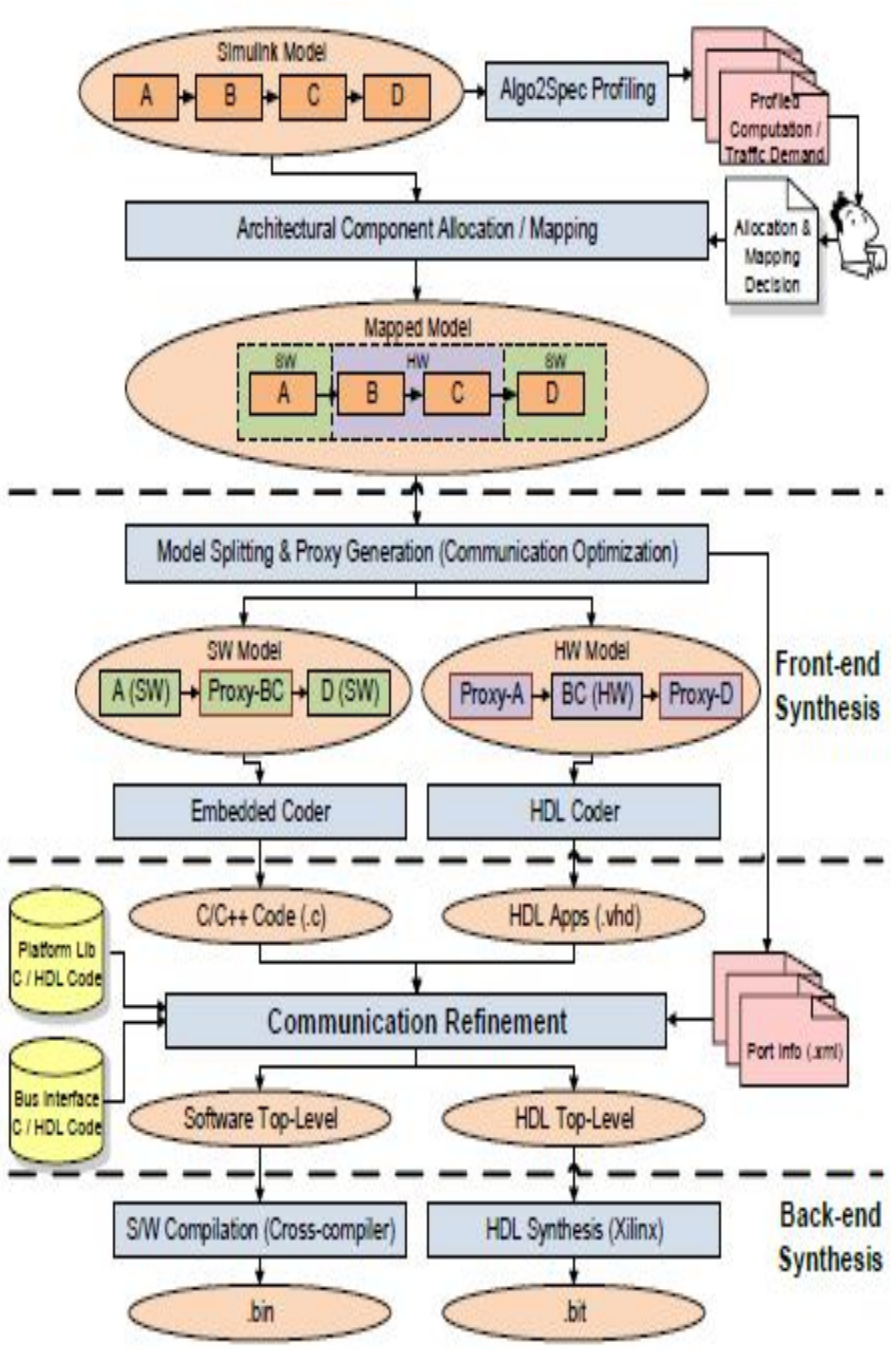


Figure 2: SimSH Flow

Then the refined communication is integrated into the software and hardware implementation. In the Back-end Synthesis, SimSH integrates the cross-compilation environment for software compilation and Xilinx ISE for high level synthesis. It finally generates software binary for processors and bit stream for FPGAs. The work in this paper makes assumptions and restrictions: (a) the user selects allocation and mapping manually. (b) it is bounded by Simulink Embedded Coder and HDL Coder restrictions and only supports discrete event models using fixed step solver.

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Figure 3: Input Image

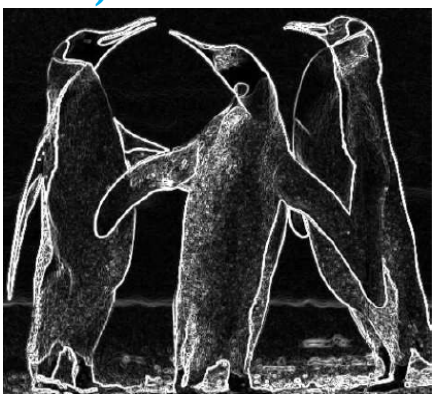


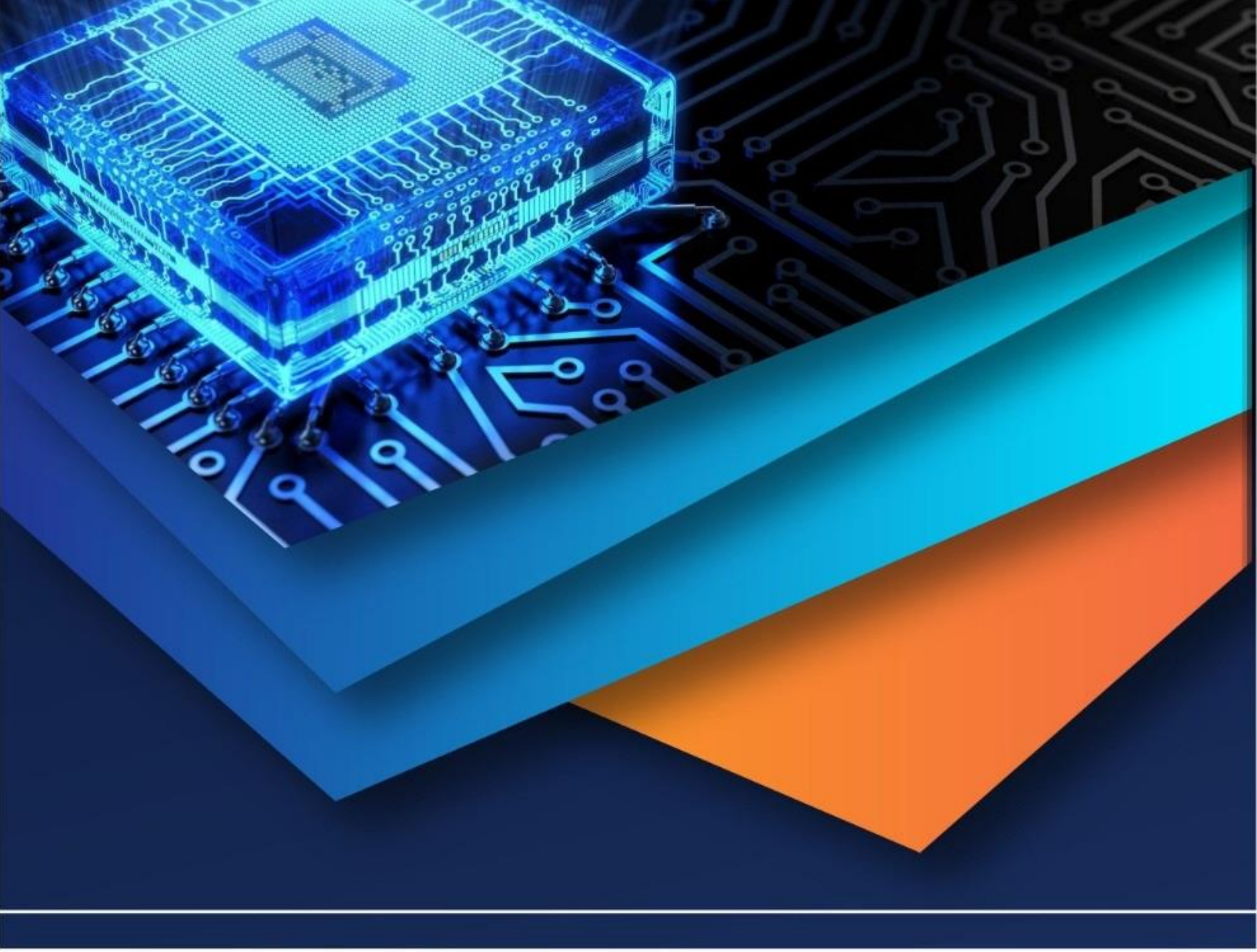
Figure 4: Output edge detection Image

The Fig 3 shows the input image which is give to system, then it convert into grey image. This image is give serialization to Sobel edge detection model. Sobel Edge detection model made with help of XSG blocks in Simulink environment. Our edge detected image finally goes through image post processing block where it again converted into 2D image. Edge detected image obtain by this approach is given below in figure 4.We use targeted board for implementation is Xilinx Spartan 3E XC3S500e.

IV. CONCLUSION

Xilinx system generator has a unique hardware in the loop co-simulation feature that allows designers to greatly accelerate simulation while simultaneously verifying the design in hardware. The purpose of this paper was to demonstrate the use of System Generator to design a system Edge Detection for image processing. Edge detection using software is not tough job but when we are going to implement it on hardware we have to face challenges like total VHDL code or Verilog code actually becomes very bulky it's near about 5000 lines. To shrink it we use Xilinx system generator. Simulation speed increase by this hardware software co-simulation technique .We can easily go for ASIC prototype by this approach. This design is implemented in the Xilinx FPGA Development kit.

[1] Yahia Said, Taoufik Saidani, Fethi Smach and Mohamed Atri "Real Time Hardware Co-simulation of Edge Detection for Video Processing System", 2012 IEEE.
[2] Zhang Shanshan, Wang Xiaohong "Vehicle Image Edge Detection Algorithm Hardware Implementation on FPGA", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).
[3] The Math Works Inc. MATLAB and Simulink, 1993-2014.
[4] MS.DIPIKAS.WARKAR, DR.U.A.KSHIRSAGAR "FPGA Implementation of Point Processing Operation using Hardware Simulation", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 4, April 2015
[5] K. Popovici et al. Simulink Based Hardware-Software Code sign Flow For Heterogeneous MPSoC. In Proceedings of the 2007 summer computer simulation conference, pages 497–504. Society for Computer Simulation International, 2007.
[6] K. M. Popovici. Multilevel Programming Environment for Heterogeneous MPSoC Architectures. PhD thesis, Grenoble Institute of Technology, 2008.
[7] M. V. Fernando, C. Kohn, and P. Joshi, "Zynq all programmable SoC Sobel filter implementation using the Vivado HLS tool," vol. 890, pp. 1–16, 2012.
[8] T. Saidani, D. Dia, W. Elhamzi, M. Atri and R.Tourki, "Hardware Co-simulation for Video Processing Using Xilinx System Generator," Proceedings of the World Congress on Engineering, vol.1, Jun 2009. London, U.K
[9] A. T. Moreo, P. N. Lorente, F. S. Valles, J. S. Muro and C. F. Andres, Experiences on developing computer vision hardware algorithms using Xilinx system generator" Microprocessors and Microsystems, Vol. 29, pp.411-419 November 2005.
[10] C.John Moses, D. Selvathi, S.Sajitha Rani, "FPGA Implementation of an Efficient Partial Volume Interpolation for Medical Image Registration" IEEE International Conference on Communication Control and Computing Technologies(ICCCCT-10), pp.132– 137,Oct.2010
[11] Yahia Said, Taoufik Saidani, Fethi Smach and Mohamed Atri "Real Time Hardware Co-simulation of Edge Detection for Video Processing System", 2012 IEEE.
[12] Zhang Shanshan, Wang Xiaohong "Vehicle Image Edge Detection Algorithm Hardware Implementation on FPGA", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY