Real Time Hardware Co-Simulation for Image Processing Algorithms Using Xilinx System Generator

Mohammed Alareqi[1,3], Rachid Elgouri[1, 2], Khalid Mateur[1], and Laamari Hlou[1]

[1]Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, Ibn Tofaïl University, Kenitra, Morocco
[2]Laboratory of Electrical Engineering and Telecommunications Systems, National School of Applied Sciences (ENSA), Ibn Tofaïl University, Kenitra, Morocco
[3]Community College, Sana'a, Yemen
Alareqi_mohammed@yahoo.com

Abstract: The implementation of digital image processing required detailed knowledge of both hardware design and hardware description languages. This paper presents an efficient approach for the implementation of real time hardware digital image processing algorithms without requiring detailed knowledge of hardware design and hardware description languages. The purpose of this work is to achieve a real time hardware implementation with higher performance in both size and speed. It focuses on the implementation of an efficient architecture for image processing algorithms like segmentation (threshold) and contrast stretching by using the fewest possible system generator blocks for DSP tool, which integrates itself with the MATLAB based Simulink graphics environment and relieves the user of the textual HDL programming. While Past research has shown that the Image enhancement techniques on FPGA based on the Xilinx System Generator. This study connect between image histogram and Image enhancement techniques depending on the type of enhancement required. This paper describes also the methodology for implementing real-time DSP applications on FPGA and concept of hardware software co-simulation for digital image processing by using the Mathworks model-based design tool Simulink / Xilinx System Generator (XSG). Performances of efficient architectures are implemented on FPGA Virtex5 (XUPV5-LX110T).

Keywords: Image processing; Xilinx system generator; Field Programmable Gate Array (FPGA); DSP.

1. Introduction

Image enhancement techniques improve the visibility of the images. Enhancement results, the output image is more suitable for a specific application rather than input image. It is used in many image-processing applications like medical imaging, SONAR and RADAR as a pre-processing step. Hence, this paper presents the hardware co-simulation and implementation of image enhancement algorithms on FPGA.

At present, the use of FPGA in research and development of applied digital systems for specific tasks are increasing. This is due to the advantage FPGAs have over other programmable devices. The advantages include: high clock frequency, high operations per second, code portability, code library reusability, low cost, parallel processing, capability of interacting with high or low interfaces, security and intellectual property (IP) retention [1].

The availability of FPGA design tools that work at a higher level of abstraction allows users to work with FPGAs without detailed knowledge of HDLs. One of such tools is the LabVIEW FPGA. [2] Discussed the use of this tool in DSP teaching. Another tool that also works at a higher level of abstraction and is conducive for those having prior knowledge of MATLAB and Simulink, is the Xilinx System Generator for DSP. [3] Presented a basic DSP design methodology using this tool.

Received: March 21st, 2015. Accepted: December 15th, 2015
DOI: 10.15676/ijeei.2015.7.4.13

DSP functions are implemented on two primary platforms such as Digital Signal Processors (DSPs) and FPGAs [4]. Consequently, this paper presents an efficient approach for the implementation of digital image processing in real time (Image enhancement) on FPGA without requiring detailed knowledge of both hardware design and hardware description languages. The approach is based on the Xilinx System Generator for DSP tool, which integrates itself with the MATLAB based Simulink Graphics environment and relieves the user of the textual HDL programming.

FPGAs are increasingly used in modern imaging applications image filtering [5-6], medical imaging [7-8], image compression and wireless communication [9]. The rest of the paper is organized as follows. Section2 describes Xilinx System Generator. Section3 presents proposed work. Study case (algorithm for image segmentation threshold and an algorithm for image contrast stretching) displayed in section 4. Sections 5&6 describes the hardware co-simulation, Implementation, and results. Finally, the concluding remarks are given in Section 7.

2. Overview of the Xilinx System Generator

System Generator is a DSP design tool from Xilinx created to implement DSP applications on FPGA using the Mathworks model-based design tool Simulink [10]. This tool is very easy to work with because it does not require a previous knowledge of hardware design methodologies. The design by using System Generator only needs a DSP Simulink modeling environment but based on a specific block set from Xilinx. All the downstream FPGA implementation steps including synthesis, place and route are automatically performed to generate an FPGA programming file. System Generator provides a system integration platform for the design of DSP and FPGAs that allows the RTL, Simulink, MATLAB and C/C++ components of a DSP system to come together in a single simulation and implementation environment.

System Generator supports a black box block that allows RTL to be imported to Simulink and co-simulated with either ModelSim or Xilinx ISE Simulator. Consequently, the proposed Xilinx System Generator and HDL Co-simulator platform based approaches help the designer to make fault diagnosis, complete analysis of the system, evaluation of system performance, and enhancement of the system performance prior to final implementation onto hardware. This result in reduction in design time and faults also ensures that the designs meet design specifications. Figure1 summarizes the steps of the System Generator design flow.

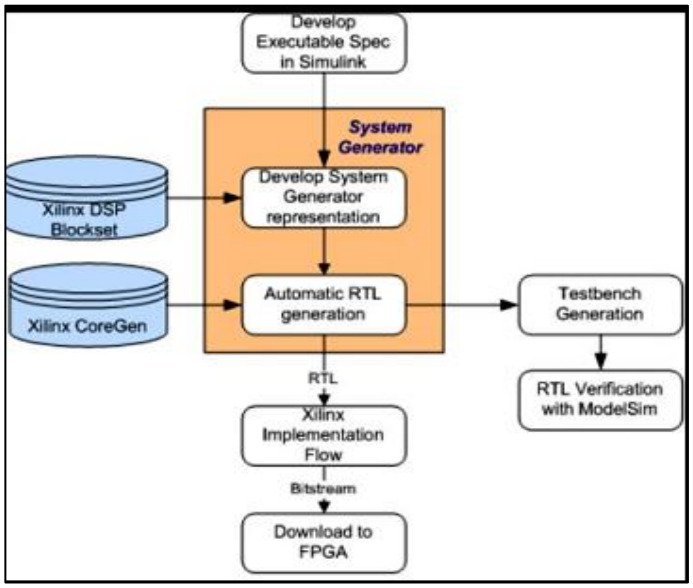


Figure 1. Design flow for xilinx system generator [10]

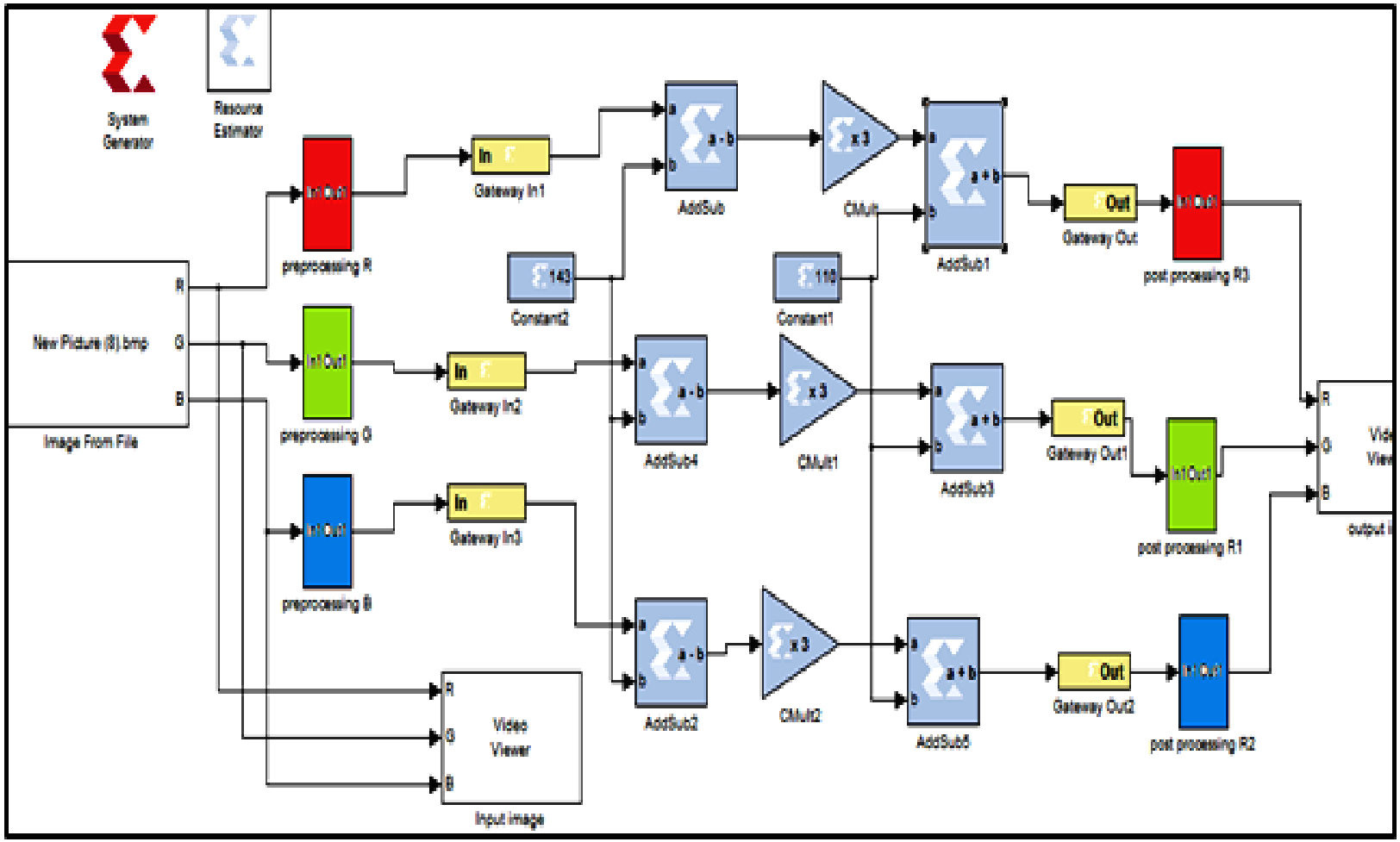


Figure 4. Algorithm for color image contrast stretching

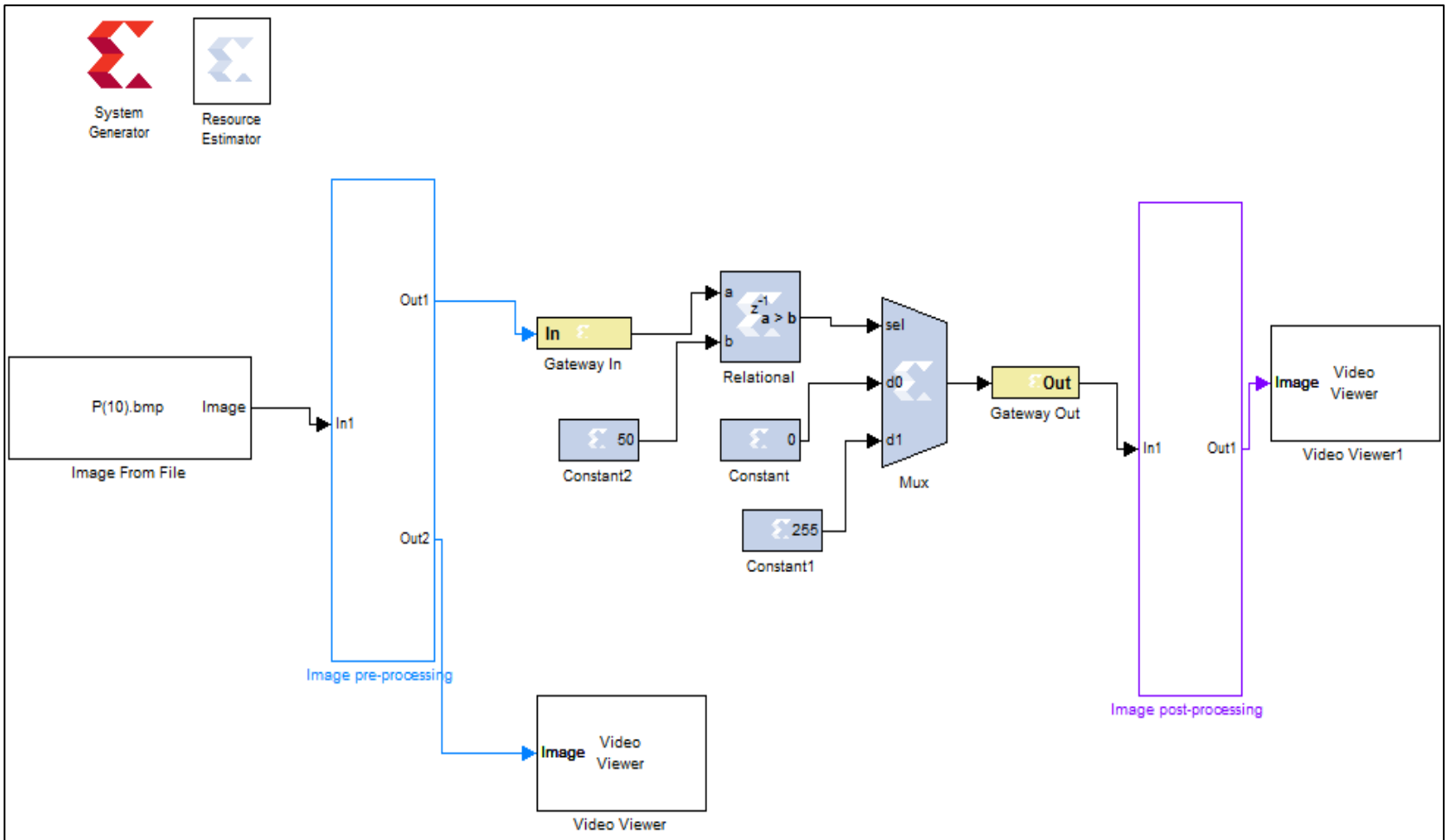
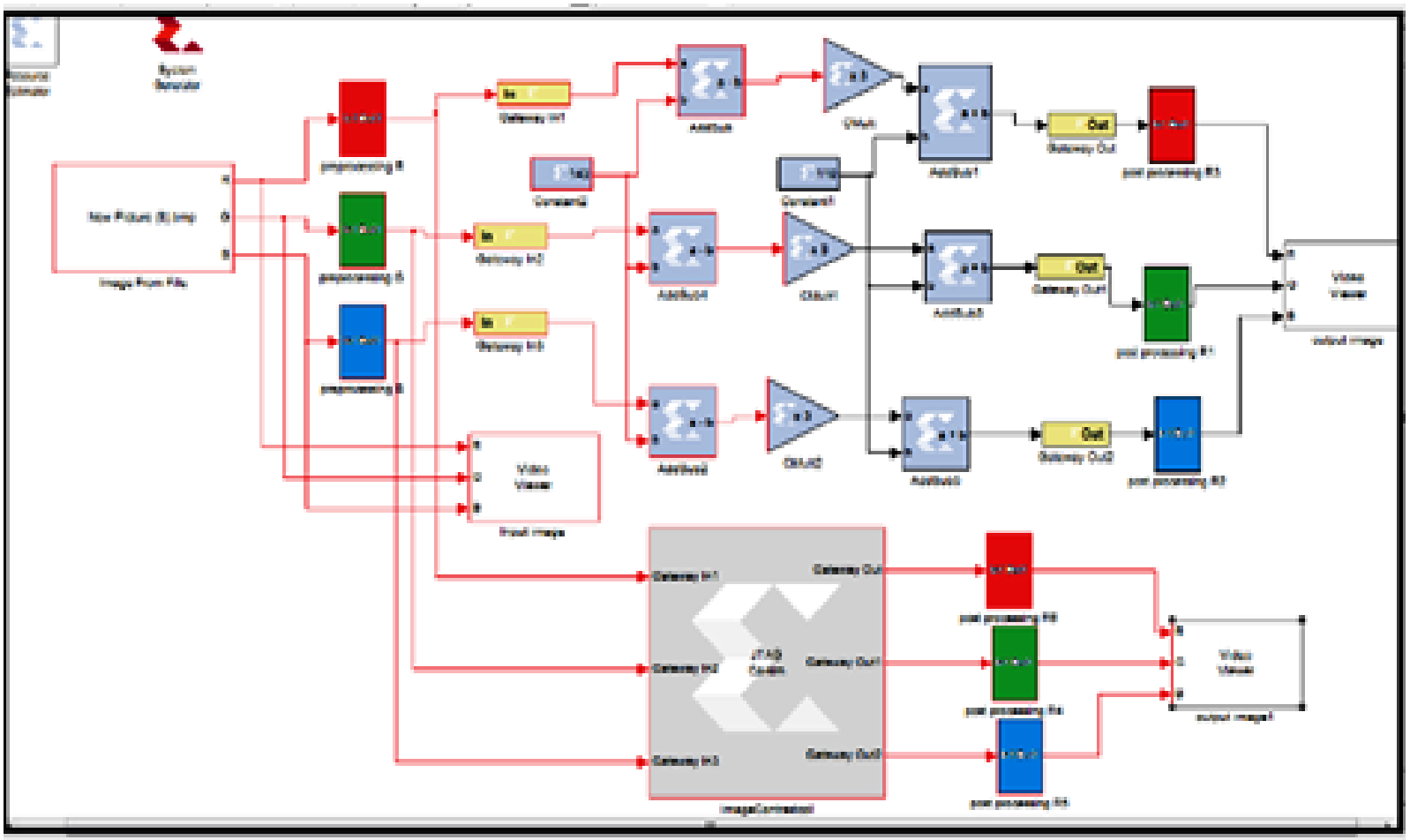


Figure 5. Algorithm for image segmentation by thresholding



b. Contrast stretching

Figure 6. Complete hardware/software co-simulation design

Figure 8 shows the software and hardware simulations for the Contrast stretching design for the input image.

The obtained result when the algorithm is applied by using Equation (1), is shown in Figure 8-b. It shows that the histogram stretches in dark region. Another result is obtained when using Equation (2) is shown in Figure 8-c. It shows the histogram expands in bright region.

Both results revealed that, there is no difference between those obtained from MATLAB and FPGA.

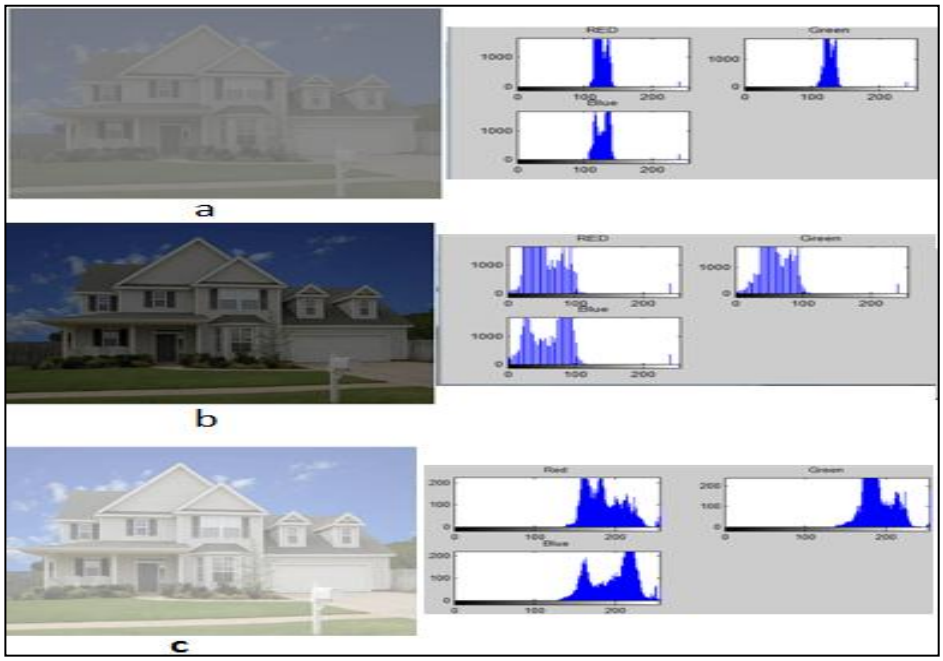


Figure 8. a) Original image b) XSG based contrast stretching by using equation1 c) XSG based contrast stretching by using equation2

The Top-level RTL schematic for the contrast stretching algorithm developed and implemented on FPGA is shown in Figure 9. This is a schematic representation of the pre-optimized design shown at the Register Transfer Level (RTL).This system blocks are designed for the Virtex-5 ML505 board.

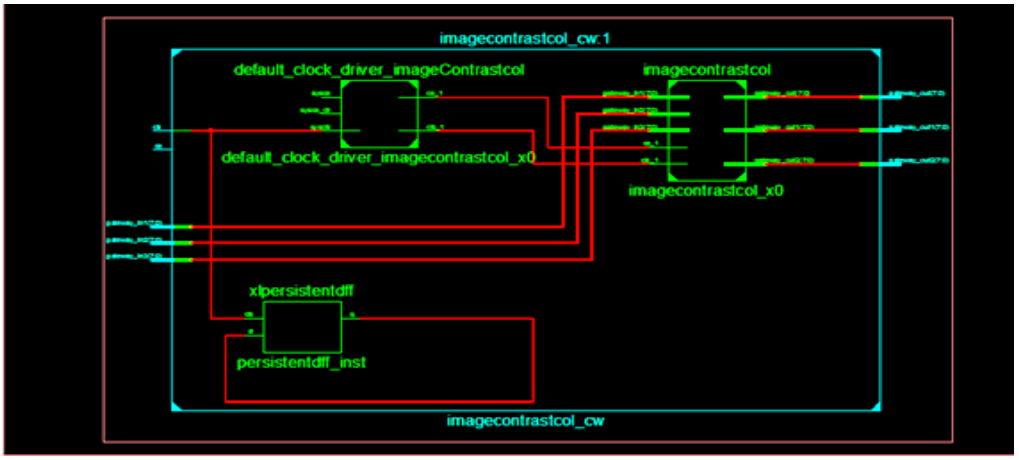


Figure 9. RTL schematic for Image contrast stretching

7. Conclusion

The objective of this paper was to demonstrate the use of System Generator for implementing the digital image processing algorithms on a FPGA. Since it does not require a previous knowledge of hardware design methodologies, also has the advantage of using large memory, and embedded multipliers. Advances in FPGA technology with the development of sophisticated and efficient tools for modeling, simulation and synthesis have made FPGA a highly useful platform. Flexible and reasonable use of DSP building blocks provided in the XSG can easily turn the flowchart of algorithm into a corresponding implement on FPGA.

8. References

[1]. A. R.Arreguína, J. C. M. Moralesa, J. M. R.Arreguín, J. C. P.Ortegaa, S. T.Arriagaa, M.A.A. Fernandeza and J. d. J. R.Magdalenob, "FPGA Open Architecture Design for a VGA Driver," *Iberoamerican Conference on Electronics Engineering and Computer Science,vol.3, pp. 324–333, 3 (2012).*

[2]. N. Kehtarnavaz and S. Mahotra, "FPGA implementation made easy for applied digital signal processing courses," Acoustics, Speech and Signal Processing (ICASSP), 2011 *IEEE International Conference on, vol., no., pp.2892-2895, 22-27 May 2011.*

[3]. M.Ownby and W.H. Mahmoud, "A design methodology for implementing DSP with Xilinx® System Generator for Matlab®," System Theory, 2003. *Proceedings of the 35th Southeastern Symposium on, vol., no., pp. 404- 408, 16-18 March 2003.*

[4]. A. Mohammed, E. Rachid, and H. Laamari, "High Level FPGA Modeling for Image Processing Algorithms Using Xilinx System Generator," *International Journal of Computer Science and Telecommunications, Vol.5, Issue 6, pp.1-8, June 2014.*

[5]. S. Hasan, A. Yakovlev, and S. Boussakta, "Performance efficient FPGA implementation of parallel 2-D MRI image filtering algorithms using Xilinx system generator," *IEEE International Conference on Communication Systems Networks and Digital Signal Processing (CSNDSP), pp. 765–769, July2010.*

[6]. R.Harinarayan, R. Pannerselvam, M. Mubarak Ali, and D. Kumar Tripathi, "Feature extraction of Digital Aerial Images by FPGA based implementation of edge detection algorithms," *IEEE International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT), pp.631 – 635, March 2011.*

[7]. C.JohnMoses, 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.*

[8]. M.F. Bin Othman , N. Abdullah, and N.A. Bin Ahmad Rusli ,"An Overview of MRI Brain Classification using FPGA Implementation," *IEEE Symposium on Industrial electronics & Applications (ISIEA),pp.623 – 628, Oct. 2010.*

[9]. H. Taha, A.N.Sazish, A.Ahmad, M. S.Sharif, and A. Amira, "Efficient FPGA Implementation of a Wireless Communication System Using Bluetooth Connectivity," *IEEE International Symposium Circuits and Systems (ISCAS), pp 1767-1770, June 2010.*

[10]. Xilinx System Generator User's Guide, www.Xilin x.com.

[11]. P.H. Pawar and R. P. Patil, "FPGA Implementation of Canny Edge Detection Algorithm," *International Journal of Engineering and Computer Science, Vol. 3, PP. 8704-8709, Issue 10, October 2014*

[12]. Kalpit R.Chandpa, Ashwini M. Jani, Ghanshyam I. Prajapati, "Comparative Study of Linear and Non-linear Contrast Enhancement Techniques," *International Journal of Research and Scientific Innovation, Vol. I,PP.37-41 Issue VI, November 2014.*

[13]. Salem Saleh Al-amri, N.V. Kalyankar and Khamitkar S.D, "Image Segmentation by Using Threshold Techniques" *Journal of Computing, vol. 2, issue.5, PP.83-86 may 2010.*