FPGA Design and Implementation for Huge Data Manipulation Systems

Asmaa Hameed Rasheed
Lecturer
College of Engineering, Baghdad University
Baghdad, Iraq

ABSTRACT

The FPGAs will continue to be used for many of today's digital signal processing applications. The reasons for this are firstly FPGAs are highly configurable hardware, since it has a grid of reconfigurable gate structure. The second reason is MIPS and MMACS (Millions of instructions and Millions of Multiply-Accumulate Operations executed per Second) requirements of specific application. Although FPGAs have their limitations and processing restrictions. The main obstacle of using FPGA with huge data processing systems is the limited number of pins of the FPGA kit that restrict the number of input samples that must be processed simultaneously. This paper describes an efficient FPGA based hardware design with the assistance of XSG (Xilinx System Generator) applied for huge data systems. The used approach is a windowing technique to cut specified number of pixels from a huge data using Xilinx System Generator block sets to build the required digital system design that can be converted to a hardware co-simulation design which is defined for FPGA environments . Image filtering techniques and algorithms are used in this work for testing the proposed approach. Xilinx software ISE 14.7 with (VHDL) language for Spartan3-700A and MATLAB R2013a are the combined S/W for hardware co-simulation design.

Keywords

FPGA, Xilinx system generator (XSG), Data processing, image filtering.

1. INTRODUCTION

Image processing techniques can be chosen as an example of huge data processing systems due to large number of data samples in both horizontal and vertical axis result in huge amount of data. Filtering processing is used in many areas for taking the decisions. Common occurrence of noise in digital images is the impulse noise [1], [2]. This type of noise occurred due to the transmission of pictures through the channels, digitizing, and scanning. Impulse noise can corrupt the images where the corrupted pixel takes the minimum or maximum gray level. The most suitable filter for removing this impulse noise in digital images is the median filter. This filter is good at lower percentages of noise in images [3], [4], [5], [6] but those are not sufficient for real time implementation.

Different hardware platforms exist to implement the real time signal processing solutions. Each platform has its own strengths and weakness, FPGAs and DSPs offer unique and different options for signal processing applications. FPGA still be the most candidate choice due to its reconfigurable gate structure and MIPS requirements of operation [7], [8]. FPGAs are usually used in modern digital image applications with real time systems, since FPGA's are good at flexibility, reprogram and parallelism. Therefore, the speed of operation is acceptable and gives the fast output response compared to other H/W [9],[10].

The integrated software environment (ISE) tools offer very good environment to deal with FPGA internal components and systems [11]. The use of corresponding hardware description language (HDL) to the respected image filter technique i.e. impulse C code. Xilinx platform studio consists of mainly three panels, those are project information panel, system assembly panel and connectivity panel. The project information panel provides control and information about the project. In addition it provides the project, applications & IP Catalog tabs. The system assembly panel is where you view and configure system block elements. The connectivity panel is a graphical representation of the hardware platform interconnects. Bit stream is generated by using these three panels. Download the Executable & Linkable format file on FPGA. By using RS232 serial cable, and parallel JTAG is connected to the target board by using PC [12], [13].

2. FPGA STRUCTURE FOR DIGITAL SYSTEM DESIGN

Since a huge data cannot be treated simultaneously, so a sliding window must be used which slides over the hole input data [14]. An FPGA consists of a matrix of programmable logic cells, with a grid of interconnecting lines and switches between them. I/O cells exist around the perimeter, providing an interface between the interconnect lines and the chip's external pins. Programming (configuring) anFPGA consists of specifying the logic function of each cell and the switches in the interconnecting lines [13].

One of the main problems with FPGA programming is the limited number of pins which restrict the treatment of huge data that must be entered simultaneously to FPGA kit. In such situations, a cooperative method between MATLAB system generator and Xilinx FPGA logic be useful to solve this problem.

This paper presents the concept of hardware software cooperative for image processing using Xilinx System Generator (XSG). This technique provides a set of Simulink blocks for several hardware operations that are implemented on Xilinx FPGA [12].

Digital data of speech or image represent a huge data system that need a special treatment when they are processed using FPGA kit [15], [16], [17]. Since the image needs heavy data density, (as an example image of size 250*400 i.e. 100000 pixels), then it can be a good example of huge data system, which can not be entered simultaneously to the FPGA kit and this leads to the need of a XSG to treat this huge data.

3. STEPS OF COMBINED FPGA & SG ALGORITHM

1. Inter huge data to MATLAB work space.
2. Transfer mask from original data to Xilinx FPGA by

using Xilinx System Generator (XSG).

I. Do the segmentation process to multi over-lapped windows. The window size must be determined depending on the used technique.

II. Transfer single window to FPGA kit using XSG.

3. Complete the process on FPGA kit.

4. Display the output in MATLAB package.

Flowchart of combined FPGA-SG system is illustrated in Figure 1.

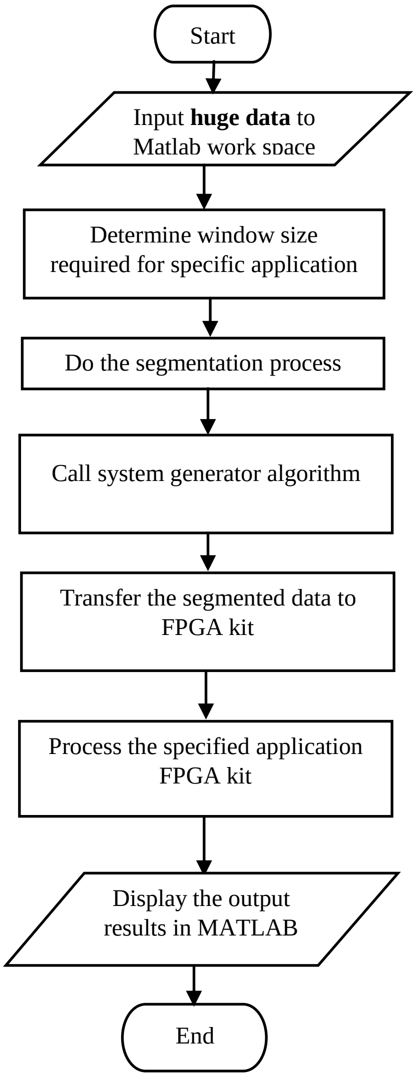

Start
Input **huge data** to Matlab work space
Determine window size required for specific application
Do the segmentation process
Call system generator algorithm
Transfer the segmented data to FPGA kit
Process the specified application FPGA kit
Display the output results in MATLAB
End

Fig 1: Flowchart of FPGA-SG system design.

4. COMBINED FPGA-SG SYSTEM DESIGN

XILINX block set consist of system generator (SG) block which is responsible of system control and black box block that implements the majority voting circuit in VHDL. Gateway in and gateway out are also parts of the XILINX block set which are used for type conversion from MATLAB data type to XILINX data type.

System design with specified number of input data samples two main steps, the first step is the design of the simulink design using SG block set, and the second one is the implementation of the H/W co-simulation design using FPGA structure.

A tested technique to be applied with our work is the image processing. The overall design of 9 input data samples system is shown in Figure 2, and Figure 3. This can be extended to any number of input data samples as shown in Figure 4 and Figure 5.

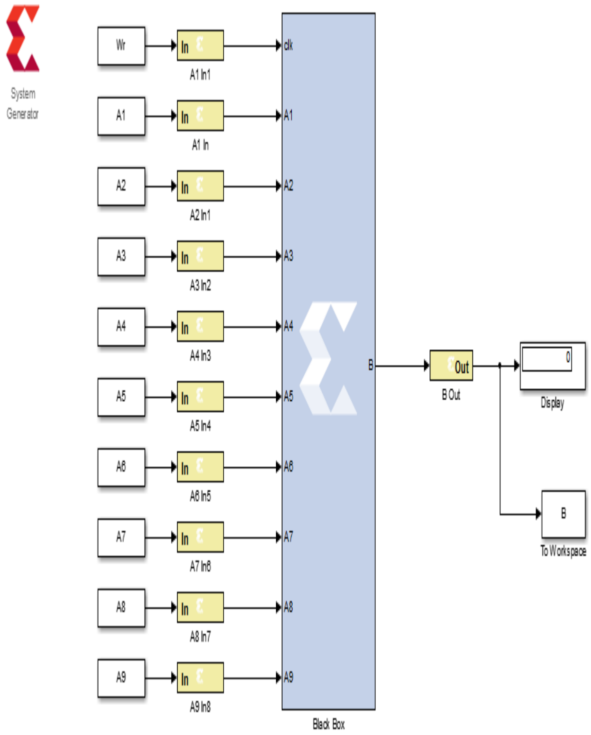


Fig 2: System generator block of 9 input data samples, windowing technique [18]

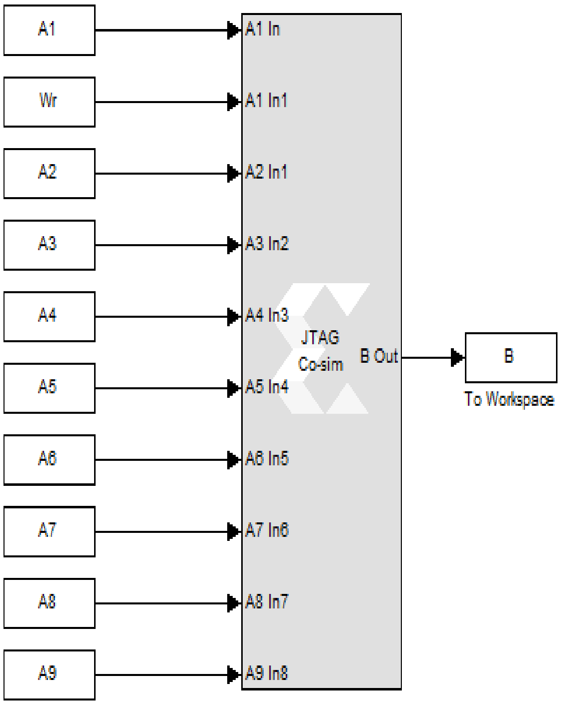


Fig 3: H/W Co-simulation design of Figure 2

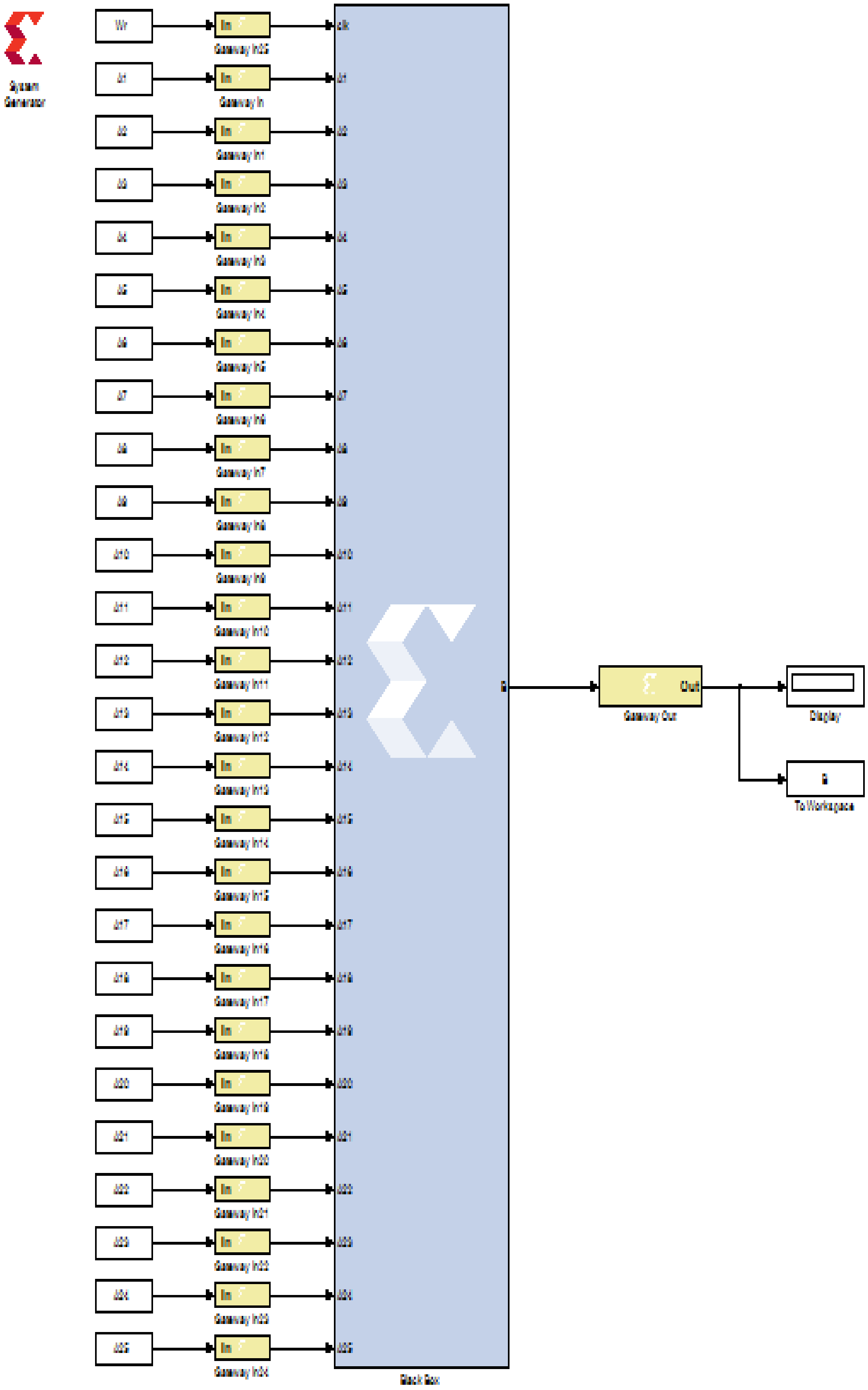


Fig 4: Extended system generator block of Figure 2

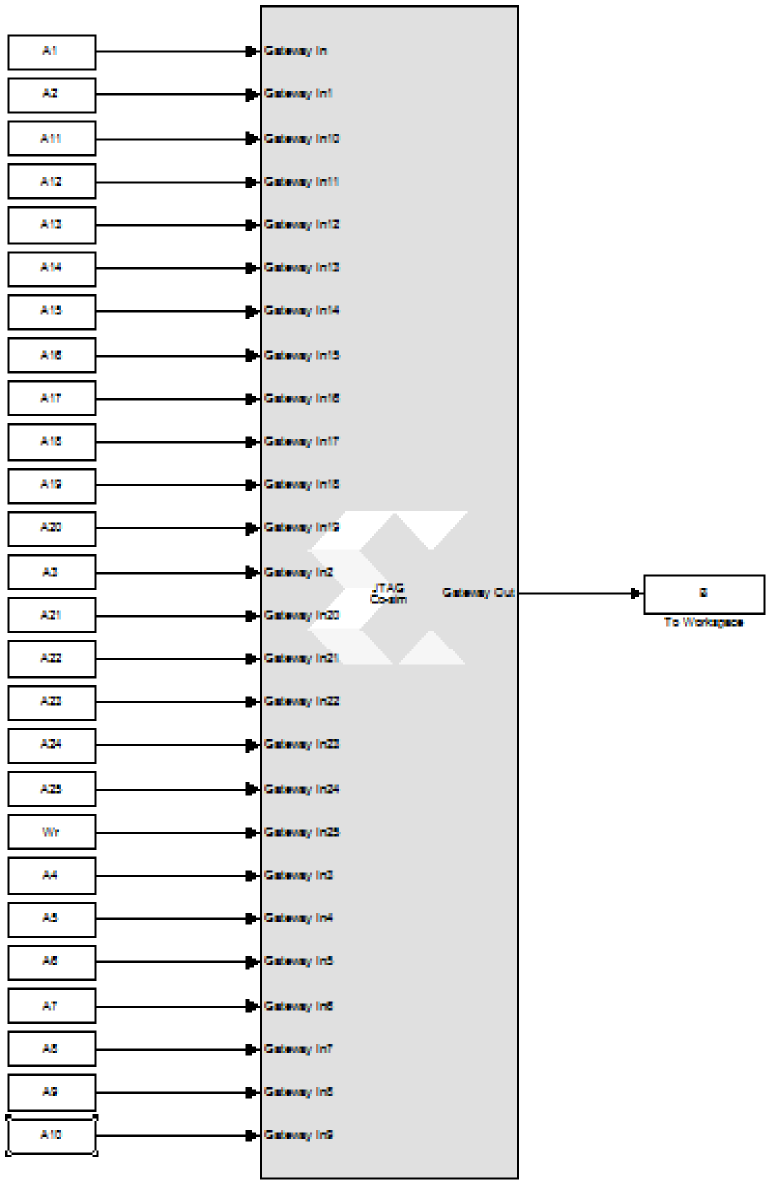


Fig 5: H/W Co-simulation Design of Figure 4.

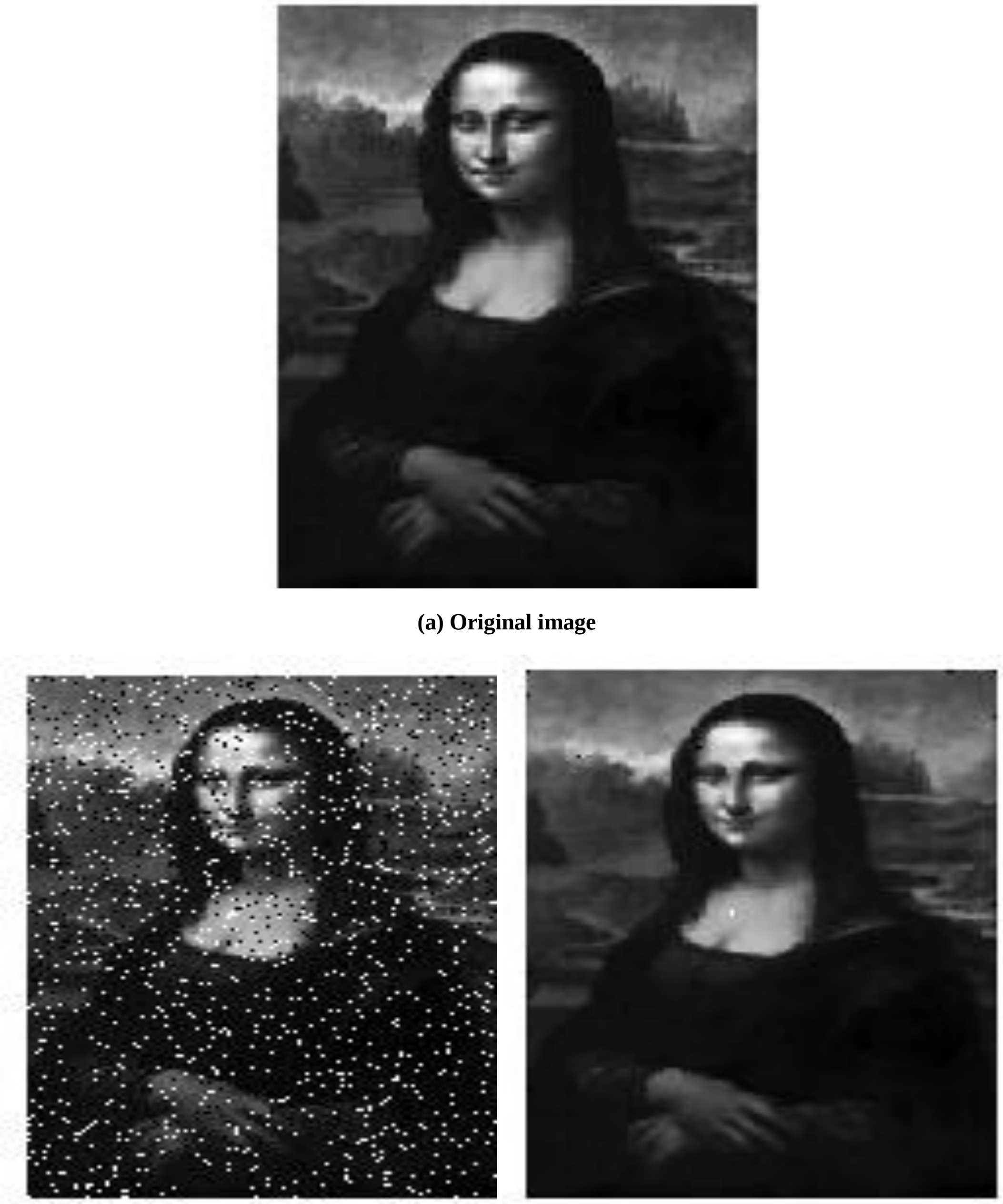
5. THE RESULTS

The image of size of 127*270 is taken to test the system performance. The chosen tested noise is the impulse noise since the median filter behaves good with this noise type. Several images are taken for median filtering to establish good filter performance with proposed technique using SG algorithm.

The original and processed images are shown in Figure 6. The noisy image was corrupted with impulse noise. The output image after applying the designed filtering process is shown in Figure 6-(c). The system performance shows that the XSG block set design operate correctly as its expected.

These huge input data may by duplicated once, twice or more depending on the application requirements.

Total number of used FPGA internal components for different specified input data samples are illustrated in Table 1 and Table 2. The number of bounded IOBs pins is 81 for 9 input data samples that takes 21% from the total available. The percentage of occupied slices is only 5% from the total available. When the number of entered data samples are extended to 25 input, the number of IOBs are increased to 212 that take 57% from the total available. For this extended input number of samples, the occupied slices are increased respectively to occupy 45% from the total number of slices.



(a) Original image

(b) Noisy image

(c) Output image

Fig. 6: (a) Original image, (b) Noisy image, and (c) Output image.

Table 1: Design Utilization of 3*3 Median Filters

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	650	11,776	5%
Number of occupied Slices	320	5,888	5%
Number of bonded IOBs	81	372	21%

Table 2: Design Utilization of 5*5 Median Filters

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	5,424	11,776	45%
Number of occupied Slices	2,700	5,888	45%
Number of bonded IOBs	212	372	57%

6. CONCLUSSION

From the proposed technique it is concluded that the problem of limited number of pins of FPGA, which restrict the processing of huge amount of data, can be overcame using system generator block set design.

The proposed technique proved its validity by checking the image filtering system performance with the impulse noise. The original image that represent the huge input data of (187 *270) i.e 50,490 data samples can be processed efficiently using XSG design. At least three images are needed to be stored (original, noisy, and output image), thus the total required memory is 151,470 cells.

The device utilization summary of 9 input data samples shown in the design utilization table is as follows: the number of 4 input LUTs is 650 among 11776 and the area utilized for it is 5% only. The number of occupied slices is 5% also. Then from the design utilization table its concluded that the used method is area efficient and the system generator succeed to be used for interfacing Simulink blocks and Xilinx FPGAs blocks.

For extended input data samples (25 samples here), the utilization summary shows that the used components and area is very large compared to 9 input data samples.

The most important thing that must be noted here is that, for this FPGA kit (Spartan3-700A) the number of input data samples for the tested technique, cannot be increased further more since the required IOBs will be out of the kit capacity that limited to 372 only. Hence, a more enlarged versions of FPGA kits must be used to extend the range of input data that must be processed simultaneously for any huge data systems.

7. REFERENCES

[1] Harsh Prateek Singh, Ayush Nigam,Amit Kumar Gautam, Aakanksha Bhardwaj,Neha Singh,"Noise Reduction in Images using Enhanced Average Filter", International Conference on Advances in Computer Engineering & Applications (ICACEA),pp. 25-28, 2014.

[2] Mr. Rohit Verma, Dr. Jahid Ali, "A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, pp. 617-622, 2013.

[3] M. Karaman, A. Atalar, "Design and Implementation of a General-Purpose Median Filter Unit in CMOS VLSI", IEEE Journal of Solid-State Circuits, Vol. 25, No. 2, April 1990.

[4] Hyeong-Soek Yu, Joon-Yeop Lee and Jun-Dong Cho, "A Fast VLSI Implementation of Sorting Algorithm for Standard Median Filters", 12th Annual IEEE International ASIC/SOC Conference Proceedings, 1999.

[5] N.Rajesh Kumar, J.Uday Kumar,"A Spatial Mean and Median Filter For Noise Removal in Digital Images", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 4, Issue 1, January 2015.

[6] A. K. Jain, "Fundamentals of Digital Image Processing",Prentice Hall of India, First Edition, 1989.

[7] Galani Tina, R.D.Daruwala, "Performance Improvement of MIPS Architecture by Adding New Features", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, pp. 423-430, 2013.

[8] Scott Hauck, Gaetano Borriello, "Pin Assignment for Multi-FPGA Systems", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 16, No. 9, pp. 956-964, 1997.

[9] Ramyashree B H,Mrs. Vidhya R, Mr. Manu DK, "FPGA implementation of contrast stretching for image enhancement using system generator", IEEE, 2016.

[10] Krishna Kishore Anumanchi, K. V. Ramana Reddy, Dr. Siva Yellampalli, " Rapid Prototyping of Image Edge Detection and Display Sub System on FPGA using Soft Processor", International Journal of Engineering Research & Technology (IJERT), Vol. 4, pp. 45-47, 2015.

[11] B. Draper, R. Beveridge, W. Böhm, C. Ross, M. Chawathe, "Implementing Image Applications on FPGAs", InternationalConference on Pattern Recognition, Quebec City, 2002.

[12] Xilinx System Generator user guide.

[13] Xilinx System Generator website (http://www.xilinx.com/video/hardware/getting-started-with-system-generator.html.html).

[14] R. Gonzalez and R.E. Woods, Digital Image Processing. Reading, MA: Prentice Hall,3rd edition, 2007.

[15] M.A. Vega-Rodriguez, J.M. Sanchez-Perez, J.A. Gomez-Pulido, "An FPGA-Based Implementation for Median Filtering Meeting the Real time Requirements of Automated Visual Inspection Systems", Proceedings of the 10th Mediterranean Conference on Control and Automation, July 2002.

[16] G.L. Bates and S. Nooshabadi, "FPGA Implementation of a MedianFilter", Proceedings of IEEE Speech and Image Technologies for Computing and Telecommunications, Vol. 2, pp. 437-440, 1997.

[17] Madhuri Gundam, Dimitrios Charalampidis, "Median Filter on FPGAs", 44th IEEE Southeastern Symposium on System Theory, pp. 83-87, 2012.

[18] Radhamadhab Dalai, "A Low Power Selective Median Filter Design", Master‟s Thesis, National Institute Of Technology, Rourkela, Department Of Computer Science And Engineering, June-2008.