DESIGN AND IMPLEMENTATION IMAGE PROCESSING FUNCTIONAL UNIT USING SPATIAL PARALLELISM ON FPGA

Nada Qasim Mohammed[1], Muataz H. Salih[1], Rafikha Aliana[1], Qasim Mohammed Hussein[2] and Noor Aldeen A. Khalid[1]
[1]School of Computer and Communication Engineering Universiti Malaysia Perlis, Perlis, Malaysia
[2]College of Petroleum and Mineral Engineering, Tikrit University- Iraq
E-Mail: muataz@unimap.edu.my

ABSTRACT

According to the importance of using hardware implementation for digital image processing to gain performance with high quality; this paper produces some basic techniques to enhance images with their hardware implementation and results using the VHDL and mega Core modules. The paper focuses on exploring the parallelism features on FPGA to perform image enhancement in the spatial parallelism, by applying spatial parallelism to design and build embedded real time system. The FPGAs reconfigurable make them very suitable choice for real-time image processing to implement many enhancement algorithms to get fast computation. It utilizes of parallel on-chip registers and memory enhanced the processing time of implemented algorithms. An FPGA development board DE2-115 is used as vehicle project. The results of implemented design show that the throughput is increased in term of coarse-grain scale and reduce the power consumption. Also, the operating frequency is increased to 1GHz by configuring phase-locked loop (PLL). As well as the possibility to implement several algorithms using the same Hardwar, only reconfigurable it.

Keywords: embedded system design, FPGA system design, image enhancement, spatial parallelism, and digital image processing.

INTRODUCTION

Image enhancement is a method which is used to improve image vision and makes the image adapt to be processed by a computer. Processes of image enhancement comprise a collection of techniques to improve the appearance of image visual or the image conversion to a form better suited for analysis by machine or human [1].

Different ways and a variety of types of processing to enhance the images have been used. In the early stages, a sequential processing was used for image enhancement. This type of processing consumes a lot of time since the images have massive data. Thus, it's necessity to use parallel processing to speed up the implementation of operations on the images.

Image processing applications are suitable for using the parallelism in algorithm with high degree, since the same operations can be used for each pixel in a portion of an image or the whole image, and this will lead to a significant decreasing in the consuming of time for large image processing [2].

Many ways are used for parallel processing using software or hardware. The implementations make the use of parallel functional unit to invest the parallelism by execution the instructions in parallel in spatial domain Thus, the implementation will take less time consuming for the enhancement processing [3].

The implementation of applications with specific hardware (embedded system) provides better speed than implementation of applications with specific software. In hardware technology design, there are two classes: full custom hardware, known as application specific integrated circuits (ASIC) which its design cannot change, and semi-custom hardware design techniques like pipeline and parallelism can be developed using (FPGA) that are not possible in designing the dedicated (DSP). The FPGA has good features like; configurable, low power consumption, work in real-time, minimize the product cost, and able to perform complex algorithms. [4].

FPGAs are often used to implement platforms for real-time image processing applications because their structure can exploit spatial and temporal parallelism.

The FPGA platforms have been joined in order to bring the benefits of these schemes to improve an optimized design by having the full advantage of the hardware features such as execution, robustness, and accuracy, and to eliminate the undesired features like inflexibility. The FPGA software, in turn, has the reusability and re- configurability features.

This work provides an implementation of different image enhancement algorithms such on FPGA. The development and education board (DE2_115) will used to implement proposed system for image enhancements. The VHDL is used to programme the software

RELATED WORK

Many embedded system to perform image enhancement are proposed with different levels of performance, each one has its benefit and limitations.

The paper in [3], present a designing that utilizing from VHDL code and execute it on Cyclone EPIC6Q240C8, this device is currently not available with Quartus II version 13.0. In [5] Tsuboi produced FPGA based vision system that was developed depending on using the Xilinx Virtex2E mounted on the board of FPGA. Three algorithms of image processing were implemented for finding the image gravity centre and object oriented detection by using Hough transform in addition to radial projection. C/C++ was used to code the algorithms into the HDL language Cycle C using system compiler. Cycle C can be converted into VHDL and Verilog. A [6], present a designing that utilizing a VHDL code and image which send by mobile to FPGA and show it on VGA.

In paper [7] suggest an implementation of interface of VGA port that available on FPGA board to produce the character(s) from ASCII text characters. Paper in [8] introduces the Time comparison between multiple image processing algorithms implemented in CPU and CPU+GPU is presented; they achieved a speedup of almost 100 times when using the GPU like a coprocessor.

According to [9] a comparison between FPGA, GPU and CPU performances is presented for different image processing algorithms (stereo vision and two-dimensional _litters). In paper [10] present Image enhancement and DE noising process using brightness manipulation, threshold operation in FPGA with On board reconfiguration and partial dynamic reconfiguration, in addition to median filtering operation to increase the performance of FPGA and decreases the resource requirement

PARALLEL PROCESSING PLATFORMS COMPARISON

Different platforms were presented and used today worldwide to design embedded system for different image processing applications. They differ in the requirements such as cost, availability of the tools. The success of any one depends on the application which is used. Selecting the right platform is a significant step in the design of any embedded system [11]. This section shall present the strength and weakness of different parallel platforms by giving a comparing between different hardware platforms.

a) FPGA vs. ASIC

An ASIC is a device specialized to implement applications of specific tasks and it builds with ICs during the production. In an ASIC, the architecture is designed using a Hardware Description Language (HDL), and its function cannot be modified. The main limitation of an ASIC is its price and the time to market, market but it has low power consumption coupled with high performance.

On other hand, FPGAs are reconfigurable device that can be reprogrammed several times based on design and logic gates and memory bits. Figure-1 shows FPGA vs. ASIC design flow comparison.

An ASIC requires less material in comparing with FPGA, so it spends low recurring costs. While FPGA is better than an ASIC when building low volume production circuits. Choosing ASIC or FPGA system is based on the cost and the need to reprogram or not. Table-1 shows a comparison between FPGA and ASIC. [17].

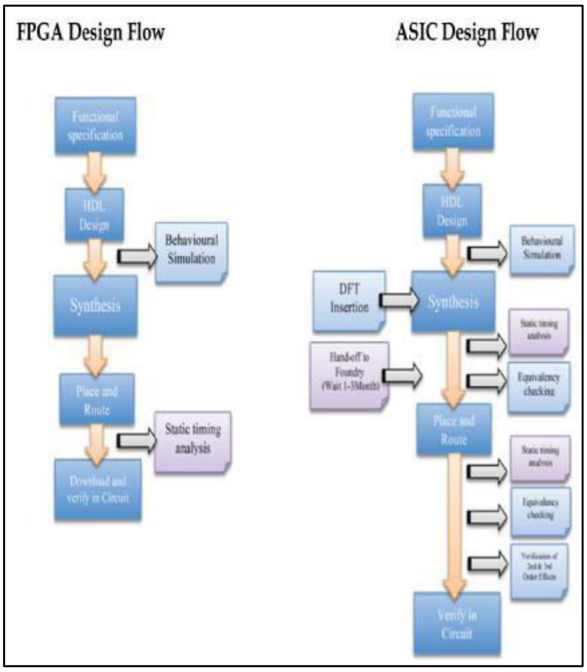

Figure-1. FPGA vs. ASIC design flow comparison.

Table-1. FPGA and ASIC compression.

Platform	Adv.	Dis.	Proram. Language
FPGA	•Allows operations to be performed faster than on CPUs or DSPs. •Ability to perform operations simultaneously on large sets of data •Perform more complex control tasks. •Fast and efficient systems •Real time applications parallel data processing	• More expensive than custom silicon • Slower • No dynamic logic • Not good for high volume applications	VHDL Verilog
ASIC	Low power consumption.	Increasing IC design costs Don't provide flexible programm-ability.	BASIC, GW-BASIC

b) FPGA vs. DSP

DSPs are a processor system that contain a specialized architecture that permit instruction level parallelism processing, known SIMD (Single Instruction Multiple Data). From the programmer point of view, a special assembler instructions included in the C program is used to run the parallel instructions. DSP has cheap digital storage.

Digital processing is not always working in real time DPS has fixed connection and hardware structure, so the operations are predefined. Whereas, the FPGA hardware structure and interconnection can be defined by the user. Therefore, the FPGA operations are not predefined.

c) FPGA vs. GPU

Another utilized platform in embedded system design is the Graphic Processing Unit (GPU). This platform consists of hundreds of small cores that can be used for graphics applications or high-performance computing. GPUs use programming models like CUDA and Open CL, based on high-level languages like C, C++ or Fortran. When working with GPUs, threading is handled automatically by the hardware thread manager. The programmer does not have direct control of the processors of the GPU; everything is done through Application Programming Interfaces (API) [17].

One of the main disadvantages of GPU's are related to cooling and energy, which are also large components, which are complicated to install in small places. GPU's are intended to be used like a co-processor, so it always works in parallel with a CPU, which limits their use in some embedded systems [8].

Time comparison between multiple image processing algorithms implemented in CPU and CPU+GPU is presented; they achieved a speedup of almost 100 times when using the GPU like a coprocessor. In addition to energy efficient, as shown in Table-2. According to [9] a comparison between FPGA and GPU, FPFAs are considered the best in the design of Image processing unit.

Table-2. FPGA and GPU comparisons.

Platform	Adv.	Dis.	Proramm. Language
FPGA	•Allows operations to be performed faster than on CPUs or DSPs. •Ability to perform operations simultaneously on large sets of data •Perform more complex control tasks. •Fast and efficient systems •Real time applications parallel data processing	• More expensive than custom silicon • Slower • No dynamic logic • Not good for high volume applications	VHDL Verilog
GPU	•Fast and Cheap. • Energy efficient.	• Hard to program. • Not all algorithms can have theoretical speedup.	C, C++ or Fortran

FIELD PROGRAMMABLE GATE ARRAY (FPGA) WITH IMAGE PROCESSING

The system performance, power consumption, and cost are considered the key to the success or failure of any system. Increasing the system performance is a relative issue since that it can be done via multiple mechanisms and approaches [12].

The FPGAs are considered one of the solutions that increase the system performance since they have the ability to be programmed and even re-programmed for the sake of keeping up with new functionalities of the design. Field Programmable Gate Arrays (FPGA), massively parallel architectures can be developed to accelerate the execution speed of several image processing algorithms" [13]. All FPGA components can be electrically programmed by the user after manufacturing and implemented. FPGA features propose using it in implementing many complex digital circuits [14].

FPGAs have always trying to cover all the drawbacks of other schemes such as Programmable Logic Devices (PLD) or Application Specific Integrated Circuit (ASIC). Although the reprogrammable silicon has the same software flexibility running on processor-based system, it is constrained by the processing core number that is available. In comparison of FPGAs with the other processors, FPGAs have parallel nature truly. Therefore, the processing operations don’t compete for the same resources. Each independent processing task is assigned to a dedicated section of the chip, and can function autonomously without any influence from other logic blocks [15] Every dedicated section of the chip is handle an independent processing task without any influence from other logic blocks, that it can function autonomously As a result, when you use more processing, the performance of the application is not affected [16].

SYSTEM SPECIFICATIONS

We took in our consideration during designing our system to reduce the power consumption, work in real-time, minimize the product cost, and able to perform complex algorithms. Figure-2 illustrates the top-level design of the system.

The top-level design of the system. it consists of main unit. The Input Unit Manager is used to organize the data entry process, which can be organized by many sources .SRAM and On-Chip Memory, as” units of memory” which are used for data storing in order to read it by the input unit manager.

The clock generator can generate a clock signal reaches up to1 GHz by using the PLL in this design .Addresses for the input data that stored on the SRAM and on-chip memory will generate by the I/P address generation and then will be stored as an array inside the Input Data Buffer (IDB).

The presented design was used spatial parallelism technique for processing many blocks of the input data. The Output Data Buffer (ODB) is used to store the processed data in the form of an array the responsibility of generating many addresses for this data will be the function of the O/P address generator. The function of output unit manager will be processing the data exit through using a special file for introducing the data to the VGA

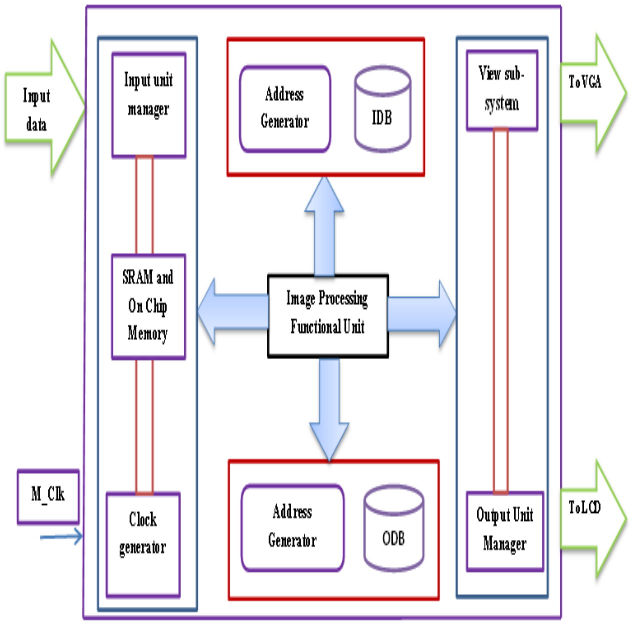

Figure-2. System top level designs.

Image processing functional unit

In order to present the Image processing functional unit, which is the core of system top level designs, it includes the algorithms that can be used to enhancement the image that stored in the memory through the distribution which can execute the spatial parallelism functional units synchronously and display the image enhancement on the LCD monitor.

The paper manipulates many algorithms to enhance the image: manipulation of contrast, manipulation of brightness, images inverting and operations of threshold, as shown if Figure-3.

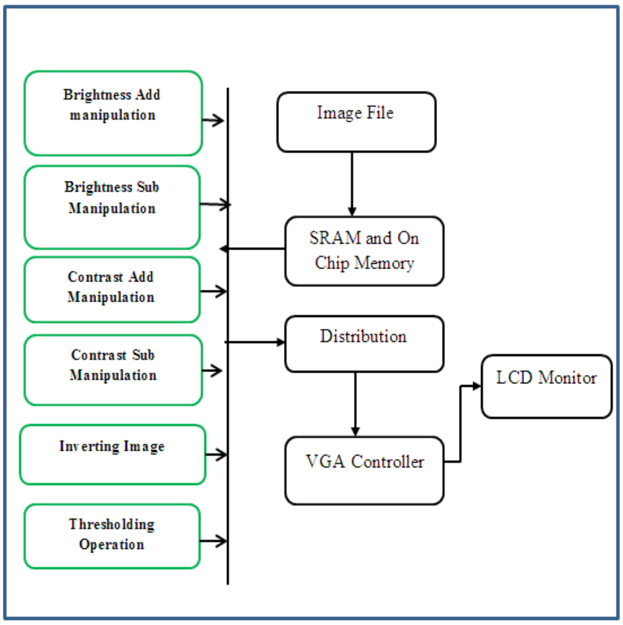

Figure-3. Image processing functional unit.

A. Manipulation of contrast

This way to expand the range of contrast in manipulated image through transfer the brightest pixel value to white, while the darkest pixel value to black.

B. Manipulation of brightness

Point operation is executed to increasing and decreasing the image brightness. Dark region in an image might get to be brighter after performing this operation. The VHDL code is used to increase or decrease the value of the picture pixel values

C. Images inverting

Doing the reverse of all the range of contrast is supporting the equivalent of a photographic negative. In point operation, the ordering of pixel values are reversed by multiplying each value with -1 and adds a constant, is known as inverting an intensity image. This method transforms the pixel value to the accepted range again [13].

D. Operations of threshold

Operations of threshold are especially interesting method for image segmentation which allows separating object of interest from its background. Thresholding an image will separate all pixels value in one of two values only. The thresholding is done using specific threshold value that select by operator

RESULT

This section highlights the performance of the system implementation (Enhance Implementation of Image Processing Functional unit using Special Parallelism on FPGA). The experimental results are obtained after implementing the proposed embedded system by using VHDL in quartus II with different input images.

In CAD tool is programmed to consists six logarithms on one block: Brightness Add manipulation, Brightness Sub Manipulation, Contrast Add Manipulation, Contrast Sub Manipulation, Inverting Image and Threshold Operation. The block will be send the project to the Field Programming Gate Array board (DE2-115), Then FPGA board is connected to the VGA. At this stage, the project is ready for operation.

The outputs obtained from the board testing will be displayed on VGA, after implementing using DE2_115 board.

After executing the project, the results will be displayed on the VGA. At this time, the researcher has to take the decision to changing its value until get the appropriate image depend on the quality of the obtained image from the execution.

Figure 4 a and b shows the result of execution the brightness algorithm. It appears that the dark region in images become brighter. So, the point operations are used to brightness increasing.

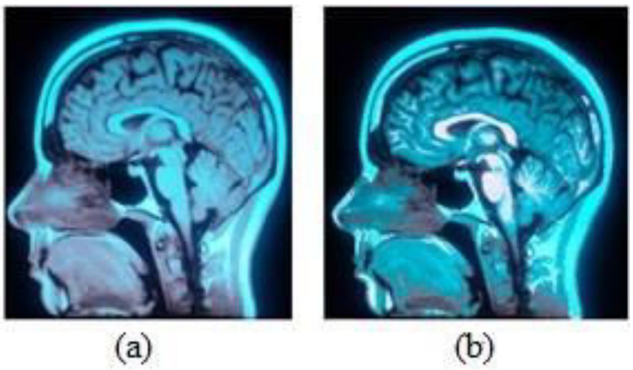

Figure-4. (a) Original image (b) Brightness image.

Figure 5 (a) and (b) shows the highlighted and dimmed region very clearly. It can be very beneficial for architects and civil engineers at the time of designing and building. The benefit can be taken from the threshold algorithm when selecting construction field. For instance, to determine how much light is important for constructing a building.

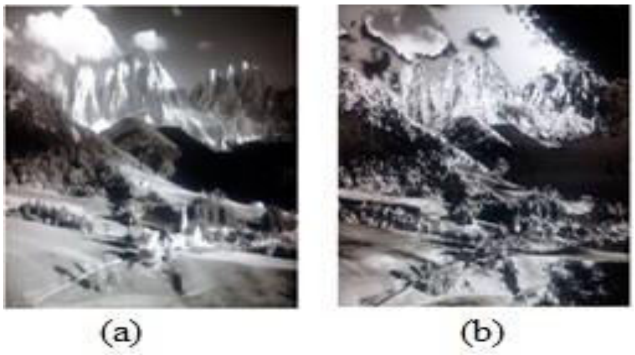

Figure-5. (a) Original image (b) Threshold image.

Figure-6 (a) and (b) shows the contrast image through the distribution. It can be observed that the new image become clearer than the original image. This result prove the visibility of features in the image which are obtained after assigning the darkest pixel value to black, as well as the brightest value to white.

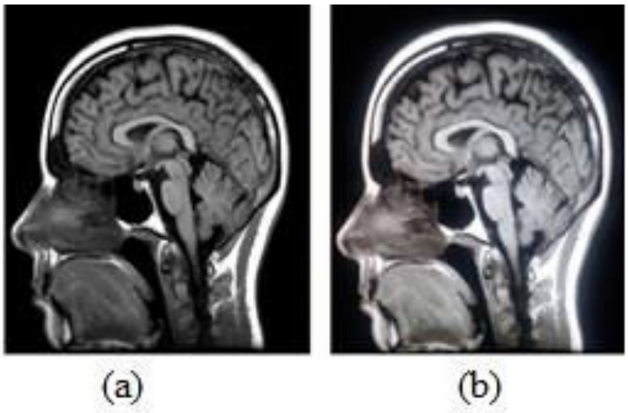

Figure-6. (a) Original image (b) Contrast image.

One of the applications of the invert image logarithm is to take image under the X-Ray type and apply the invert image logarithm on it. Thus, the result will be obtained which gives accurate details of many medical aspects, as shown in Figure 7 (a) and (b).

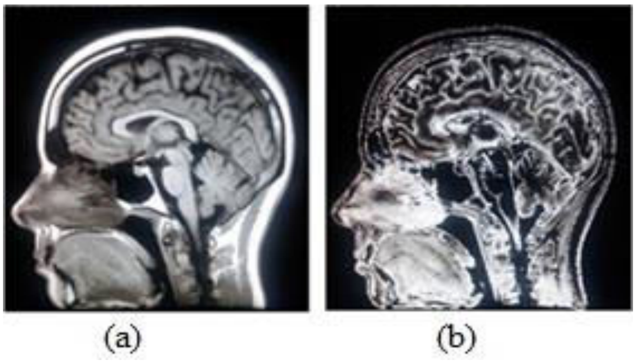

Figure-7. (a) Original image (b) Effect of invert operation.

It's very important to point out that the efficient capacity of the used board, through highlighting the amount of the used logic elements and compare it with that available logic element on the board. Table-3 shows that despite FPGA resource consumption and the design complexity that will demand large amount of logic elements; the DE2-115 board was enough for yield the design with the needed resource. Also, Table-4 shows the total logic element and max. Operating frequency utilized by proposed design.

Table-3. FPGA resources consumption.

Name	Brightness add	Brightness sub	Contrast add	Contrast sub	Invert	Threshold
Total logic element	1134/114.480 (1%)	1036/114.480 / (< 1%)	1011/114.480 / (< 1%)	983/114.480 (< 1%)	961/114.480 (< 1%)	925/114.480 (< 1%)
Total combinational functions	852/114.480 (< 1%)	633/ 114.480 (< 1%)	621/114.480 (< 1%)	611/114.480 (< 1%)	609/114.480 (< 1%)	597/114.480 (< 1%)
Dedicated logic registers	467/114.480 (< 1%)	428/114.480 (< 1%)	413/114.480 (< 1%)	411/114.480 (< 1%)	407/114.480 (< 1%)	456/114.480 (< 1%)
Total registers	467	428	413	411	407	456
Total pins	75/529(16%)	75/529(16%)	75/529(16%)	75/529(16)	75/529(16)	75/529(16%)
Embedded memory bit	3.673.600 / 3.981.312 (92%)					
Embedded multiplier 9 bit elements	0/532 (0%)	0/532 (0%)	0/532 (0%)	0/532 (0%)	0/532 (0%)	0/532 (0%)
Total PLLS	1/4 (25%)					

CONCLUSIONS

The implementation results of the designed system show the system ability to perform enhancement image processing functional unit in spatial parallelism. It has full benefits of the FPGA chip both in terms of the ability to reconfigure the system as well as increasing the throughput. The implementation of applications with specific hardware (embedded system) provides better speed than the implementation of applications with specific software. Also, it shows the benefits of using the on-chip PPL to control clock cycle which increases the operating frequency to IGHZ. These results are obtained for image size 640 *480. However, the approach discussed in this study can be used for images of any size.

ACKNOWLEDGEMENT

The authors would like to thank the Ministry of Education Malaysia (MOE) for providing the FRGS research grant (Grant no. 9003-00474).

REFERENCES

[1] Russ J. C. 2015. The image processing handbook. CRC press.

[2] Paul J. 2017. Image Processing on Heterogeneous Multiprocessor System-on-Chip using Resource-aware Programming (Doctoral dissertation, Technische Universität München).

[3] Zhao J. 2015. Video/Image Processing on FPGA (Doctoral dissertation, Worcester Polytechnic Institute).

[4] Al-Doori M. H., Ahmad R. B., Yahya A. & Arshad M. R. 2012. FPGA Design and Implementation of Multi-Filtering Techniques Using Flag-Bit and Flicker Clock. International Journal of Computer Science. (9): 39-48.

[5] Rao D. V., Patil S., Babu N. A. & Muthukumar V. 2006. Implementation and evaluation of image processing algorithms on reconfigurable architecture using C-based hardware descriptive languages. International Journal of Theoretical and Applied Computer Sciences. 1(1): 9-34.

[6] Wang G., Guan Y. & Zhang Y. 2009. Designing of VGA character string display module based on FPGA. In Intelligent Ubiquitous Computing and Education, 2009 International Symposium on (pp. 499-502). IEEE.

[7] Woon W. K. 2012. The Design and Implementation OF VGA Controller on FPGA.

[8] Chiuchisan I., Cerlinca M., Potorac A. D. & Graur A. 2012. Image Enhancement Methods Approach using Verilog Hardware Description Language. In Poceedings of the 11th International Conference on Development and Application Systems. tefancel Mare University of Suceava, ISSN, 5039: 144-148.

[9] Fykse E. 2013. Performance Comparison of GPU. DSP and FPGA implementations of image processing and computer vision algorithms in embedded systems.

[10] C. Kalyana Sundaram, M. Elango, P. Marichamy. 2014. Implementation of Image Processing Algorithm Using Partial Dynamic Reconfiguration in FPGA. International Journal of Innovative Research in Science, Engineering and Technology. 3(3).

[11] Chawan M. P., Patle B., Cholake V. & Pardeshi S. 2012. Parallel Computer Architectural Schemes. In International Journal of Engineering Research and Technology, 1(9). ESRSA Publications.

[12] Yousif O. F., Salih M. H., Hassnawi L. A., Albreem M. A., Seddeq M. Q. & Isam H. M. 2015. Design and implementation computing unit for laser jamming system using spatial parallelism on FPGA. In 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) (pp. 38-43). IEEE.

[13] Botero-Galeano D. 2012. Development of algorithms and architectures for driving assistance in adverse weather conditions using FPGAs (Doctoral dissertation, INSA de Toulouse).

[14] Sundararajan P. 2010. High performance computing using FPGAs. Xilinx White Paper: FPGAs, 1-15.

[15] Demler M. 2011. Dual ARM Cortex-A9 MPCore features 28-nm, low-power programmable logic for high-end embedded systems.

[16] Mellit A. & Kalogirou S. A. 2014. MPPT-based artificial intelligence techniques for photovoltaic systems and its implementation into field programmable gate array chips: Review of current status and future perspectives. Energy. 70, 1-21.

[17] Kalarot R. & Morris J. 2010, June. Comparison of FPGA and GPU implementations of real-time stereo vision. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops (pp. 9-15). IEEE.