

# A Survey on Image Implementation based on FPGA

Farah Saad Al-Mukhtar

Computer Science Department / College of Science / Al-Nahrain University/ Baghdad-Iraq

---

## ABSTRACT

The image processing is one of the most powerful fields in the modern Digital signal processing techniques; also it has a wide range of applications these days such as image compression, filtering and coloring. However, these processes required a huge data processing, so it has a problem under real time or movie, the huge data processing under real time requires special processing tools such as super parallel processing computers or special hardware systems. Field Programmable Gate Array (FPGA) technology has become a viable target for the implementation of image processing algorithms.

This paper gives a comprehensive survey to the hardware implementation on image processing and a brief historical development of image representation. Also, the paper gives a study on image implementation and processing on Field Programmable Gate Array (FPGA) technology.

**Keywords:** Image Processing, ASIC, DSP, SPLDs, CPLDs, FPGA, Xilinx, Virtex.

## HOW TO CITE THIS ARTICLE

Farah Saad Al-Mukhtar, "A Survey on Image Implementation based on FPGA", International Journal of Enhanced Research in Science, Technology & Engineering, ISSN: 2319-7463, Vol. 8 Issue 1, January -2019.

## 1. INTRODUCTION

Image processing is a form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it <sup>[1]</sup>.

Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned space crafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics software ... etc <sup>[2]</sup>.

## 2. METHODS OF IMAGE PROCESSING

There are two methods available in image processing:

- A. Analog image processing
- B. Digital image processing

### A. Analog image processing

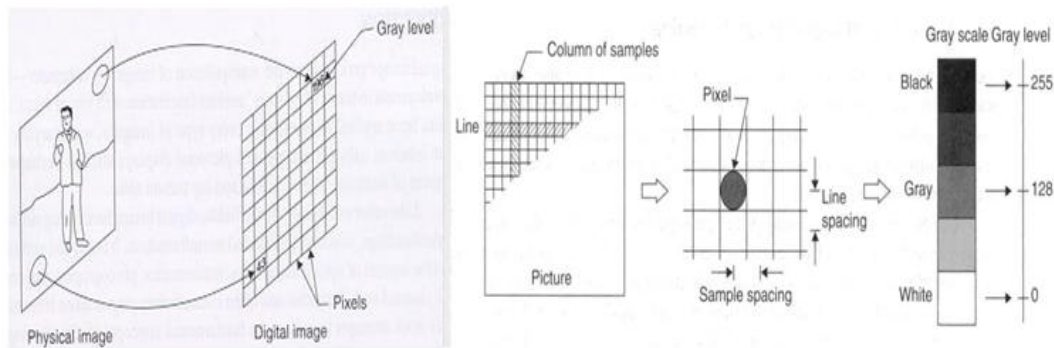
Analog image processing refers to the alteration of image through electrical means. The most common example is the television image.

The television signal is a voltage level which varies in amplitude to represent brightness through the image. By electrically varying the signal, the displayed image appearance is altered. The brightness and contrast controls on a TV set serve to adjust the amplitude and reference of the video signal, resulting in the brightening, darkening and alteration of the brightness range of the displayed image.

## B. Digital image processing

Digital image processing focuses on two major tasks: Improvement of pictorial information for human interpretation and Processing of image data for storage, transmission and representation for autonomous machine perception.

The most requirements for image processing of images are those images which are available in the digitized form, that is, arrays of finite length binary words. For digitization, the given image is sampled on a discrete grid (means measuring the value of an image at a finite number of points) and each sample or pixel is quantized (is the representation of the measured value at the sampled point by an integer) using a finite number of bits (figure 1). The digitized image is processed by a computer <sup>[3]</sup>.



**Fig.1 : image digitization**

## 3. HISTORY OF DIGITAL IMAGE PROCESSING

Early 1920s: One of the first applications of digital imaging was in the news-paper industry, as shown in (Figure 2).



**Fig. 2: Early digital image**

Mid to late 1920s: Improvements applied resulted in higher quality images.

1960s: Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing

1964: Computers are used to improve the quality of images of the moon taken by the Ranger 7 probe. Such techniques were used in other space missions including the Apollo landings (figure 3).



**Fig.3: A picture of the moon taken by the Ranger 7 probe minutes before landing**

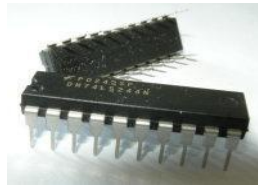
1970s: Digital image processing begins to be used in medical applications.

1980s - Today: The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas:

Image enhancement, Image restoration, Artistic effects, Medical visualization, Industrial inspection, Law enforcement, Human computer interfaces, GIS, and Image compression.

#### 4. Overview of the Programmable Logic

Prior to the invention of programmable logic electronic systems, designers had to use specialized integrated circuits, each of which contained just a few gates. Such chips were called discrete logic as seen in (figure 4). In order to create even a moderately complex device one had to mount a few tens of chips on one board. This led to more complex board layout and reduced performance.



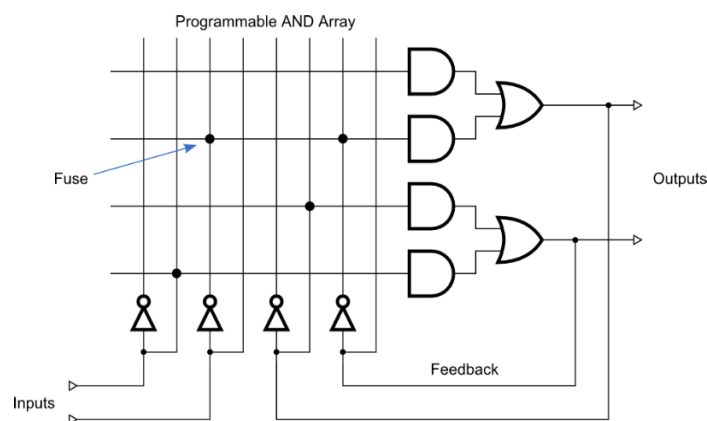
**Fig.4; Discrete logic ICs**

A general term that refers to any type of integrated circuit used for implementing digital hardware, where the chip can be configured by the end user to realize different designs is called Field-Programmable Device (FPD). Programming of such a device often involves placing the chip into a special programming unit, but some chips can also be configured “in-system”. Another name for FPDs is programmable logic devices (PLDs); although PLDs encompass the same types of chips as FPDs, the term FPD is preferred because historically the word PLD has referred to relatively simple types of devices <sup>[5]</sup>.

The three main categories of FPDs are delineated <sup>[6]</sup> :

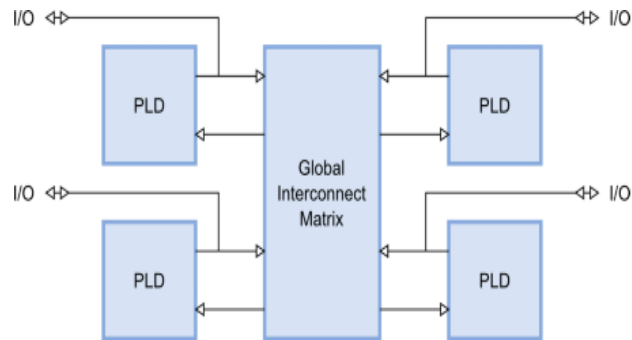
#### Simple PLDs (SPLDs), Complex PLDs (CPLDs), and Field-Programmable Gate Arrays (FPGAs).

- **Simple PLDs**(figure 5) could only handle up to 10–20 logic equations, so it couldn't fit a very large logic design into just one of them. It had to figure out how to break the larger designs apart and fit them into a set of PLDs. This was time-consuming<sup>[7]</sup>.



**Fig.5: SPLD**

- A CPLD (figure 6) contains a bunch of PLD blocks, but their inputs and outputs are connected together by a global interconnection matrix. So a CPLD has two levels of programmability: each PLD block can be programmed, and then the interconnections between the PLDs can be programmed<sup>[8]</sup>.



**Fig.6: CPLD**

- **FPGAs** have large resources of logic gates and RAM blocks to implement complex digital computations. As FPGA designs employ very fast I/Os and bidirectional data buses it becomes a challenge to verify correct timing of valid data within setup time and hold time. Floor planning enables resources allocation within FPGA to meet these time constraints. FPGAs can be used to implement any logical function that an ASIC could perform<sup>[6]</sup>.

## 5. THE ARCHITECTURE OF FPGA

Field Programmable Gate Arrays (FPGAs) were first introduced almost two and a half decades ago. Since then they have seen a rapid growth and have become a popular implementation media for digital circuits. The advancement in process technology has greatly enhanced the logic capacity of FPGAs and has in turn made them a viable implementation alternative for larger and complex designs. Further, programmable nature of their logic and routing resources has a dramatic effect on the quality of final device's area, speed, and power consumption<sup>[4]</sup>.

Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low to medium volume productions, FPGAs provide cheaper solution and faster time to market as compared to Application Specific Integrated Circuits (ASIC) which normally require a lot of resources in terms of time and money to obtain first device. FPGAs on the other hand take less than a minute to configure and they cost anywhere around a few hundred dollars to a few thousand dollars. Also, for varying requirements, a portion of FPGA can be partially reconfigured, while the rest of an FPGA is still running. Any future updates in the final product can be easily upgraded by simply downloading a new application bit stream. However, the main advantage of FPGAs i.e. flexibility is also the major cause of its drawback. Flexible nature of FPGAs makes them significantly larger, slower, and more powerful, consuming than their ASIC counterparts. These disadvantages arise largely because of the programmable routing interconnect of FPGAs which comprises of almost 90% of total area of FPGAs. But despite these disadvantages, FPGAs present a compelling alternative for digital system implementation due to their less time to market and low volume cost<sup>[4]</sup>.

Normally FPGAs comprise of:

- Programmable logic blocks which implement logic functions.
- Programmable routing that connects these logic functions.
- I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections.

In (Figure 7), where configurable logic blocks (CLBs) are arranged in a two dimensional grid and are interconnected by programmable routing resources. I/O blocks are arranged at the periphery of the grid and they are also connected to the programmable routing interconnect. The "programmable/reconfigurable" term in FPGAs indicates

their ability to implement a new function on the chip after its fabrication is complete. The re-configurability / programmability of an FPGA is based on an underlying programming technology, which can cause a change in behavior of a pre-fabricated chip after its fabrication.

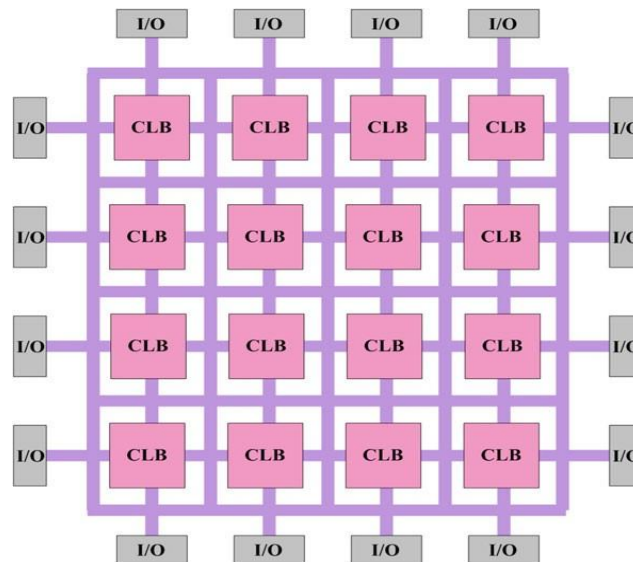


Fig. 7: Overview of FPGA architecture

## 6. PROGRAMMING TECHNOLOGIES

There are a number of programming technologies that have been used for reconfigurable architectures. Each of these technologies has different characteristics which, in turn, have a significant effect on the programmable architecture. Some of the well-known technologies include : **static memory, flash and anti-fuse** .

## 7. FPGA Vs. ASIC

Widely used computer architectures have a fixed central processing unit (CPU) operating on data stored in a memory. Programs determine the sequence of single instructions executed by the CPU. This is a disadvantage for algorithms which can be executed in parallel.

In contrast, FPGA computers have no given processor structure but offer large amounts of logic gates, registers, RAM and routing resources. These can be used for performing logical and arithmetical operations, for variable storage and to transfer data between different parts of the system. Programs do not determine the sequence of execution, but the logic structure of the reconfigurable machine. Therefore, algorithms are not only executable in parallel, but are executed by using a minimum amount of hardware. A single bit operation, for instance, is mapped on a single logical block of an FPGA (typically less than 0.01% of the machine size for currently existing architectures) instead of using about 3% of a complete 32-bit ALU like in a general purpose processor. No register-register transfers are needed to bring operands to the logical element or store the result. Typically thousands of operations can be performed in parallel on an FPGA computer during every clock cycle<sup>[9]</sup>.

Application Specific Integrated Circuits (ASICs), as the name suggests, are tailor-made on demand for specific applications, rather than intended for general-purpose use. (e.g. a chip designed solely to run a cell phone is an ASIC). In fact, often the same individual or company that designed the chip is the end user and the device is not available commercially. It is clear that this way of designing and producing circuits is extremely expensive and time consuming, but inevitable for certain high-end applications. However, for smaller designs and/or lower production volumes, ASICs have started to become a less attractive solution, as FPGAs grow larger, faster and more capable. Many companies nowadays use FPGAs during the early design phase and preproduction phases and then switch later to ASIC for volume production. For applications whose future commercial success is unknown the FPGA route offers lower risk<sup>[9][10]</sup>.

**Table 1: FPGA and ASIC comparison**

Characteristic	FPGA	ASIC
Time-to-market	Short	Long
High volume unit cost	High	Low
Flexibility after manufacturing	High	None
Performance	Medium	Very high
Density	Medium	Very high
Power consumption	High	Low
Minimum order quantities	None	High
Design flow complexity	Medium	Very high
Complexity of test	Low	High
Turnaround time	Hours	month

### 8. RECENT FPGA DESIGN TIMELINE

Xilinx offers its Virtex family at the high end and Spartan at the low end, Altera offers Stratix at the high end and Cyclone at the low end.

**Table 2: FPGA design timeline**

	Altera	Xilinx
high-end FPGA family	Stratix	Virtex
low-end FPGA family	Cyclone	Spartan

	Altera	Xilinx
1997	APEX	
1998		Virtex
2000		Spartan II
2001	APEX II	Virtex II
2002	Stratix & Cyclone	Virtex II Pro
2003		Spartan-3
2004	Stratix II & Cyclone II	Virtex-4

### 9. FPGA APPLICATIONS IN IMAGE PROCESSING

The increasing demand for real-time and smart digital signal processing (DSP) systems, calls for a better platform for their implementation. Most of these systems (e.g. digital image processing) are highly parallelizable, memory and processor hungry; such that the increasing performance of today's general-purpose microprocessors is no longer able to handle them. A highly parallel hardware architecture, which offers enough memory resources, offers an alternative for such DSP implementations.



FPGAs are particularly well suited to meet the requirements of many video and image processing applications. Altera FPGAs have the following characteristics that make them very appealing for video and image processing architectures<sup>[11]</sup>:

- High performance: HD processing can be implemented in a single Altera FPGA.
- Flexibility: Altera FPGAs provide the ability to upgrade architectures quickly to meet evolving requirements, while scalability allows the use of FPGAs in low-cost and high-performance systems.
- Low development cost: Video development kits from Altera start as low as US\$1,095 and include the software tools required to develop a video system by using Altera FPGAs.
- Obsolescence proof: Altera FPGAs have a very large customer base who ships products for many years after introduction. Also, FPGA designs are easily migrated from one process node to the next one.
- Structured ASIC migration path to low costs: Altera structured ASICs start at US\$15 at 100ku for 1 million ASIC gates.
- Altera's Video and Image Processing Solution: This includes optimized DSP Design Flows, Altera's Video and Image Processing Suite, and interface and third-party video compression IP, and video reference designs.

## **10. HARDWARE CHALLENGES**

Image processing is often viewed as a software engineering problem. Although important in the development and implementation of algorithms, software design is often given prominence over other aspects of the system. Applied image processing is really a system engineering problem because there are other important aspects to consider such as lighting, optics and integration with supporting hardware and machinery. Design with FPGAs fits well into a system engineering context because it is performed at several different levels. These include high-level algorithmic design down to bit-level operation design<sup>[12]</sup>.

Although the flexibility is available to work at the bit level, designers do not want to spend all their time there. Schematic entry and HDLs are often too low level as design tools because they do not capture the algorithmic nature of image processing functions adequately. Design at this level is complex, tedious and error prone<sup>[13]</sup>. An alternative that aids programmer productivity and more closely matches algorithmic design are high-level languages that infer circuitry by using a hardware compiler. In this context, FPGA configurations may resemble traditional high level languages like C, but specify hardware not software<sup>[14]</sup>. One advantage with this approach is that traditional software techniques which can be co-opted to help write code. The danger is the temptation to port software algorithms to hardware because of the similarities between the languages. These similarities are often superficial in the sense that the hardware configuration has merely undergone a representational change. This leads to the implementation being 'constrained' by the algorithm because the approach assumes that good software algorithms make good hardware algorithms. This is often untrue for the following reasons:

- Optimal processing modes differ on an FPGA [14]. Random-access and pointer-based operations are efficient in software. A typical processing scenario involves grabbing a frame and moving it to main memory. The processor can then sequentially process pixels, affording random access to the image. On an FPGA, this can be highly inefficient and costly.
- Clock speeds are typically an order of magnitude slower than processors due to delay overheads through the general routing matrix. Therefore, configurations must exploit parallelism rather than relying solely upon a high rate of processing.
- Sequential processing of software code avoids contention for system resources. An FPGA's potential for massive parallelism frequently complicates arbitration and creates contention for memory and shared processors.
- Lack of an operating system complicates management of 'thread' scheduling, memory, and system devices, which must be managed manually.

Based on these reasons, a more suitable algorithm may exist that can better exploit the available parallelism of the selected architecture. However, modifying an algorithm and designing the computational and memory architecture requires extra development effort on the part of the system designer.

Given these challenges we now present the complete design cycle which will allow deeper exploration of some of these issues.

## 11. THE COMPILATION PROCESS: FROM SCHEMATIC TO BITSTREAM

The process of the development software goes through compiling a design into a bit stream as depicted in (Figure 8). The steps are:

1. Entering a description of the logic circuit by using a hardware description language (HDL) such as VHDL or Verilog, or drawing the design by using a schematic editor, or using a combination of the two.
2. A logic synthesizer transforms the HDL into a netlist. The netlist is just a description of the various logic gates in the design and how they are interconnected. (A schematic is already pretty close to a netlist, so it doesn't need as much work done on it as the HDL code.)
3. The implementation phase employs three different tools (translate, map and place and route).  
 A translator merges together one or more netlists along with any design constraints. This is fed to a mapper that combines gates in the netlist into groups that will fit efficiently into the LUTs of the FPGA. The gate groupings are sent to the place and route tool that assigns them to LUTs at various locations in the FPGA and then determines how to connect them together by using the routing resources (wires) in the switching matrix. This part takes the most time as finding a placement that can be routed efficiently which requires a lot of computation.
4. A bit stream generator takes the output of the implementation phase, combines it with a few other configuration settings, and outputs a binary bit stream. This bitstream (which, depending upon the bit rate of the FPGA can be of many megabits in length) contains the truth-tables that will be loaded into the RAM of every LUT and the connection settings for the wiring matrix that will connect them.
5. At this point, a bit stream is just a bunch of 1s and 0s in a file on the computer. The downloader will transfer this file into a physical FPGA chip. In most cases, this chip is already mounted on a circuit board where it waits for the bit stream that will make it perform its intended function <sup>[15]</sup>.

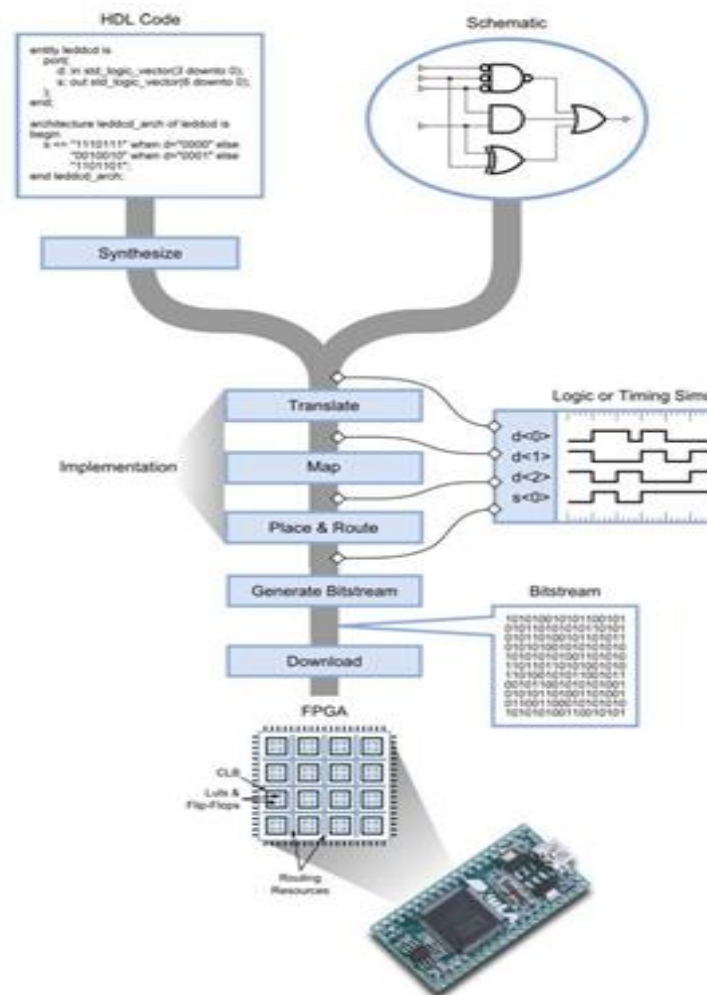


Fig. 8; compilation process



## 12. RELATED WORK

There are many projects focused on developing hardware implementations of popular image processing algorithms for use in an FPGA.

S. Ogrenci , et al. 2000 <sup>[16]</sup> , presented an initial analysis of the iterative image restoration algorithm as well as modifications made on the algorithm during the adaptation onto reconfigurable platform. The hardware design for the image restoration algorithm and the estimations on the performance of the FPGA implementation are presented. The results show that the speedup gained for practical systems varies between 6.5 and 10.2 for different images.

A. Amira , et al. 2001 <sup>[17]</sup> , presented a novel architecture for the Fast Hadamard Transform, using distributed arithmetic techniques. The mathematical model for the algorithm was proposed, the associated design by using both a distributed arithmetic ROM and accumulator structure and a sparse matrix factorization technique together with the implementation of the algorithm on a Xilinx FPGA board are described.

T. Nakano , et al. 2003 <sup>[18]</sup> , proposed a PC system for recognition of natural scene images including human faces and various objects. Coarse region segmentation of real images with 64×64 pixels at the video rate is achieved by FPGA implementation of resistive-fuse networks. A flexible template matching based on dynamic-link architecture is performed on their PC system.

Koji Nakano, et al. 2003 <sup>[19]</sup> , created Verilog HDL for an image retrieval system using FPGAs. The created Verilog HDL source is embedded in an FPGA by using the design tool provided by the FPGA vendor.

Mohamed Nasir Bin , et al. 2007 <sup>[20]</sup> , constructed a real time hardware image processing system on Field Programmable Gate Array (FPGA). The chosen image processing algorithm is a single color filtering algorithm. The functionality of the algorithm is first verified in Matlab, simulating the expected output of the system before implementing it onto the FPGA development board. Two band-pass-filter-like algorithms have been tested and implemented. The work is currently conducted to quantify the effectiveness of the band-pass filtering algorithm on FPGA before proceeding to test and implement the triple and quadruple band-pass filtering methods.

T. Latha , et al. 2007 <sup>[21]</sup> , presented the VLSI implementation of multiresolution transform based filtering impulse noise from images. This transform has many advantages in comparison with Wavelet Transform, such as fast computation, error-free reconstruction, etc. Synthesis is performed with Xilinx Spartan-II FPGA which yields users high performance, unlimited reprogrammability, very low cost and provides system clocks up to 200MHz.

Félix Moreno, et al. 2008 <sup>[22]</sup> , proposed a hardware architecture system; using Tiny Neural Networks (TNN) specialized in image recognition. One of the most important features of Tiny Neural Networks (TNN) is their learning ability. Weight modification and architecture reconfiguration can be carried out at run time. The system performs shape identification by the interpretation of their singularities. This is achieved by interconnecting several specialized TNN. The system detects accurately a test shape in almost all the experiments performed. Simulation results show that this architecture has significant performance benefits.

Khader Mohammad, et al. 2009 <sup>[23]</sup> , presented a direct method of reducing convolution processing time by using hardware computing and implementations of discrete linear convolution of two finite length sequences ( $N \times N$ ). This implementation method is realized by simplifying the convolution building blocks. The purpose of this research is to prove the feasibility of an application specific integrated circuit (ASIC) that performs a convolution on an acquired image in real time. The efficiency of the proposed convolution circuit is tested by embedding it in a top level FPGA.

M. Chandrashekar, et al. 2009 <sup>[24]</sup> , deal with Field Programmable Gate Array (FPGA) based hardware Implementation of Infrared Image (IRI) enhancement of thermo graphic images. FPGA Implemented results compared with Matlab Experiments and comparisons to histogram equalization are conducted.

Ammar A. Hassan, et al. 2010 <sup>[25]</sup> , proposed an algorithm that colorizes each gray scaled pixel by matching chromatic value of it with each pixel of colored image and synthesis it on the Xilinx FPGA devices by using VHDL synthesizer tool. Testing and performance of this technique obtained on ISE 4.1i software implementation and comparing results with other simulator results.

M. Khalil-Hani, et al., 2010 <sup>[26]</sup> , proposed a novel approach to personal verification by using infrared finger vein biometric authentication implemented on FPGA-based embedded system. The system is prototyped on Altera Stratix II FPGA hardware board with Nios2-Linux Real Time Operating System running at 100MHz clock rate. Experiments conducted on a database of 100 images from 20 different hands show encouraging results with system acceptable accuracy of less than 1.004%. The first version of the embedded system, which is wholly in firmware, resulted in an execution time of  $1953 \times 10^6$  clock cycles or 19 seconds. The results demonstrate that our approach is valid and effective for vein-pattern biometric authentication.

S. Allin Christe, et al.2011<sup>[27]</sup>, presented an efficient architecture for various image filtering algorithms and tumor characterization by using Xilinx System Generator (XSG). This architecture offers an alternative through a graphical user interface that combines MATLAB, Simulink and XSG and explores important aspects concerned to hardware implementation. Performance of this architecture implemented in SPARTAN-3E Starter kit (XC3S500E-FG320) exceeds those of similar or greater resources architectures. The proposed architecture reduces the resources available on target device by 50%.

Rucha R. Thakur , et al.2012<sup>[28]</sup> , proposed a new idea for efficient Gabor filter design with improved data transfer rate, efficient noise reduction, less power consumption and reduced memory usage. The code for Gabor filter was developed in VHDL using Model SIM and then implemented on SPARTAN-3E FPGA kit. These systems provided both highly accurate and extremely fast processing of large amounts of image data.

P.Sivarama Prasad, et al.2013<sup>[29]</sup>, presented a FPGA based on hardware accelerator for extracting the information from the screen image. The Xilinx Spartan-6 FPGA board is used for realizing morphological image processing modules along with Microblaze soft core. The Microblaze software performs the control operation and provides 100 Mbps Ethernet access to PC. The image processing modules are verified working at 100 MHz clock with chipscope occupying 70% of the selected Spartan-6 LX45 device along with Microblaze soft core.

### 13. CONCLUSION

This paper has described a different approach to the implementation of digital image processing algorithms based on field programmable gate arrays (FPGAs); it discussed issues related to many phases of the design process, starting from the algorithmic modifications, throughout the hardware design, it has presented a hardware solution which primarily exploits the parallelism of the application in order to gain speed-up against software implementations.

Our problem solving is to

1. The usage of image processing algorithms by using windowing operator's on VHDL.
2. Read the image data directly from the FPGA's RAM process it, simulate, and then get the output to the FPGA's screen.
3. Also, the image could be read from file process it, simulate, and then store the outputs results on file. Read the output file in MATLAB to get the image after processing.
4. Use the pointer to reach the positions in RAM instead of using the first in first out implementation (FIFO) and this reduces the complexity of the algorithms implementation, also it reduces the size of the algorithms.
5. The development of FPGA image processing algorithms can at times, be quite tedious, but the results speak for themselves. If high -speed, windowing algorithms are desired; the FPGA technology is ideally suited to the task. In fact, with the aid of the window generator, a whole series of image processing techniques is available to the designer, many of which can be synthesized for high -speed applications.

A prediction is that over time programmable logic will become the dominant form of digital logic design and implementation. Their ease of access, principally through the low cost of the devices, makes them attractive to small firms and small parts of large companies. The fast manufacturing turn-around they provide, is an essential element of success in the market.

### REFERENCES

- [1]. Tinku Acharya, and Ajoy K. Ray , " Image Processing - Principles and Applications" , John Wiley & Sons, INC., application , 2005.
- [2]. K.M.M. Rao , "overview of image processing", Deputy Director, National Remote Sensing Agency, Hyderabad, India. Reading in image processing, 2004. [http://www.drkmm.com/resources/INTRODUCTION\\_TO\\_IMAGE\\_PROCESSING\\_29aug06.pdf](http://www.drkmm.com/resources/INTRODUCTION_TO_IMAGE_PROCESSING_29aug06.pdf)
- [3]. Emanuele Trucco, Alessandro Verri, Prentice Hall , " Introductory Techniques for 3-D Computer Vision" ,1998.
- [4]. U. Farooq, "Tree-Based Heterogeneous FPGA Architectures" , DOI: 10.1007/978-1-4614-3594-5\_2, © Springer Science+Business Media New York 2012.
- [5]. Stephen Brown and Jonathan Rose, " Architecture of FPGAs and CPLDs: A Tutorial". Department of Electrical and Computer Engineering, University of Toronto, <http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf>
- [6]. Vaughn Betz, Jonathan Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density". Toronto, Ontario, Canada M5S 3G4, 1999.
- [7]. Dave Vandebout, " FPGAs!? Now What? Learning FPGA Design with the XuLA Board". TUT001 (V1.0) Feb 22, 2013.
- [8]. Arnaud Taffanel, Peyman Pouyan, "How Does FPGA Work". Advanced Digital IC Design ,2008.
- [9]. Ognjen Šćekić, Mentor: Prof. Dr. Veljko Milutinović, "FPGA Comparative Analysis". University of Belgrade ETF – School of Electrical Engineering 2005.
- [10]. S. K. Tewksbury, " Application-Specific Integrated Circuits (ASICs)". Microelectronic Systems Research Center, Morgantown, WV 26506,(304)293-637, 1996.

- [11]. Altera, White Paper " Video and Image Processing Design Using FPGAs"., Altera Inc.
- [12]. D. G. Bailey, "Machine Vision: a Multidisciplinary Systems Engineering Problem," in Hybrid Image and Signal Processing Orlando, Florida: SPIE 939, pp. 148-155,1988.
- [13]. J. L. Tripp, M. B. Gokhale, and K. D. Peterson, "Trident: From High-Level Language to Hardware Circuitry Computer" , vol. 40, pp.28-37, 2007.
- [14]. M. C. Herbordt, T. VanCourt, Y. Gu, B. Sukhwani, A. Conti, J. Model, and D. DiSabello, "Achieving High Performance with FPGA-Based Computing," IEEE Computer, vol. 40, pp. 50-57, 2007.
- [15]. Peter Alfke , " Creative Uses of Block RAM". White Paper: Virtex and Spartan FPGA Families, WP335 (v1.0) June 4, 2008.
- [16]. S. Ogrenici, K. Bazargan, and NI. Sarrafzadeh, " Image Analysis And Partitioning For FPGA Implementation Of Image Restoration". IEEE, 0-7803-6488,2000.
- [17]. A. Amira, A. Bouridarze and P. Milligan, " An FPGA based Walsh Hadamard Transforms ".IEEE, 0-7803-6685, 2001.
- [18]. T. Nakano, T. Morie , and A. Iwata , " A Face/Object Recognition System Using FPGA Implementation of Coarse Region Segmentation". SICE Annual Conference in Fukui. August 4-6.Fukui University, Japan, 2003.
- [19]. Koji Nakano , and Etsuko Takamichi, "An Image Retrieval System Using FPGAs ". IEEE , 0-7803-7659 , 2003.
- [20]. Mohamed Nasir Bin Mohamed Shukor, Lo Hai Hiung , Patrick Sebastian, " Implementation of Color Filtering on FPGA" . IEEE International Conference on Intelligent and Advanced Systems , 1-4244-1355, 2007.
- [21]. T. Latha , M. Sasi kumar ,and A. Albert Raj, " FPGA based digital image restoration using multiresolution transform based filtering", IET-UK International Conference on Information and Communication Technology in Electrical Sciences, pp.625-628. Dec. 20-22 ,2007.
- [22]. Félix Moreno, Jaime Alarcón, Rubén Salvador and Teresa Riesgo, "FPGAImplementation of an Image Recognition System based on Tiny Neural Networks and on-line Reconfiguration" . IEEE ,978-1-4244-1766, 2008 .
- [23]. Khader Mohammad and Sos Agaian , " Efficient FPGA implementation of convolution" . IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA , 978-1-4244-2794, October 2009 .
- [24]. U. Naresh Kumar,K. Sudershan Reddy and K. Nagabhushan Raju, "FPGA implementation of high speed infrared image enhancement". International Journal of Electronic Engineering Research ISSN 0975 - 6450 Volume 1 Number 3 pp. 279–285, 2009.
- [25]. Ammar A. Hassan, "Coloring of gray-scale image using FPGA". Journal of Engineering, Number 4, Volume 16 , December 2010 .
- [26]. M. Khalil-Hani and P.C. Eng, "FPGA-Based Embedded System Implementation of Finger Vein Biometrics". IEEE Symposium on Industrial Electronics and Applications (ISIEA 2010), October 3-5, Penang, Malaysia, 2010 .
- [27]. S. Allin Christe , Mr.M.Vignesh and Dr.A.Kandaswamy, " an efficient fpga implementation of MRI image filtering and tumour characterization using xilinx system generator".International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.4, December 2011.
- [28]. Rucha R. Thakur , Swati R. Dixit and Dr.A.Y.Deshmukh , "VHDL Design for Image Segmentation using Gabor filter for Disease Detection". International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012.
- [29]. P.Sivarama Prasad, K.Srinivasa Rao, " Hardware and Software Codesign for Computer Screen Image Processing Applications using FPGA". IJCA Proceedings on International Conference on Recent Trends in Information Technology and Computer Science 2012 ICRTITCS(6):6-11, February 2013.