

Reconfigurable Computing A review of the technology and its architecture

Gagandeep Singh

(Electronics and Communication Engineering/ Chandigarh Group of Colleges, Mohali, India)

Abstract: *Reconfigurable computing is a computer architecture which is intended to fill the gap between the hardware and the software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware. Most commonly and widely used high speed computing fabrics deployed in reconfigurable computing are field-programmable gate arrays (FPGAs). This paper illustrates the architecture used in reconfigurable computing and also demonstrates the advantages of using reconfigurable computing design over conventional ASIC or a simple microprocessor for achieving high level of performance for a specific application.*

Keywords; *ASIC – Application Specific Integrated Circuit, FPGA- Field Programmable Gate Array, GPU- Graphics Processor Unit, LUT- Lookup Table, RC- Reconfigurable Computing*

I. INTRODUCTION

The use of Reconfigurable Computing (RC) to accelerate computation arose in the late '80's with the widespread commercial availability of Field-Programmable Gate Arrays (FPGAs). The innovative development of FPGAs whose configuration could be re-programmed an unlimited number of times spurred the invention of a new field in which many different hardware algorithms could execute, in turn, on a single device, just as many different software algorithms can run on a conventional processor.

Earlier two primary methods in conventional computing for the execution of algorithms were used, either an Application Specific Integrated Circuit (ASIC) or software programmed microprocessors.

ASICs are designed specifically to perform a given computation, and thus they are very fast and efficient when executing the exact computation for which they were designed. However, the circuit cannot be altered after fabrication. This forces a redesign and refabricating of the chip if any part of its circuit requires modification. This is an expensive process and also somewhat inflexible, frequently requiring a board redesign and replacement in the event of changes to the application.

The second method is to use software-programmed microprocessors - a far more flexible solution. Processors execute a set of instructions to perform a computation. By changing the software instruction, the functionality of the system is altered without changing the hardware. However, the processor must read each instruction from memory, decode its meaning, and only then execute it. This results in a high execution overhead for each individual operation. Additionally, the set of instructions that may be used by a program is determined during the fabrication time of the processor. Any other operations that are to be implemented must be built out of existing instructions.

Reconfigurable computing is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware. A reconfigurable computing system typically contains one or more processors and a reconfigurable fabric upon which custom functional units can be built. The processor(s) executes sequential and noncritical code, while the computational codes are mapped to the reconfigurable hardware. Like an ASIC, the functions that have been mapped to the reconfigurable fabric can take advantage of the parallelism achievable in a hardware implementation but unlike an ASIC, a new fabric need not be designed for each application. A given fabric can implement a wide variety of functional units.

There are other devices which show some of the flexibility of reconfigurable computers like graphics processing units (GPU) and application specific array processors. These devices perform well on their intended application, but cannot run more general computations, unlike reconfigurable computers and microprocessors. GPUs also need large amounts of power relative to the computations being performed. GPUs provide a high level of acceleration, but at a significant cost in power consumption. Instead, FPGAs offer a viable alternative.

In recent years reconfigurable computing systems, by virtue of their parallel structures, have provided substantial acceleration for many compute-intensive algorithms as compared to various software optimized implementation's [1]. An implementation of The Serpent Block Cipher in the Xilinx Virtex XCV1000 shows a throughput increase by a factor of over 18 compared to a Pentium Pro PC running at 200MHz [2]. Many recent

applications have been shown to exhibit significant speed-ups using reconfigurable hardware like automatic target recognition [3], striking pattern matching, and data compression [4].

Other advantages of configurable designs include a reduction in size and hence cost, improved time-to-market, improved flexibility and upgradability which are especially important for embedded application. In order to achieve these performance benefits, yet support a wide range of applications, reconfigurable systems are usually formed with a combination of reconfigurable logic and a general purpose microprocessor.

II. BACKGROUND

Researchers in The United States and France in search of flexible, high performance building blocks, envisioned a new kind of supercomputer, composed of hardware-reprogrammable components that by customizing the hardware to each application could deliver up to two orders of magnitude performance increase over convention fixed instruction set processors. The first reconfigurable computers were built by the IDA Supercomputing Research Center (SRC) in the USA and the DEC Paris Research Lab (PRL).

Two versions of the “Splash” systolic array were built at the SRC i.e. The Splash 1 and The Splash 2 systems. The Splash 1 system contained 32 Xilinx 3090 series FPGAs connected in a linear array and adjacent FPGA chips sharing a memory buffer. The RC was connected to a Sun workstation via VME interconnect. Splash 1 could perform DNA sequence comparison at 45Xa (1990 era) high performance workstation.

Splash 2 [5], reduced the number of FPGAs to 16. However, due to rapidly increasing density of FPGAs, Splash 2, with 16 Xilinx 4010 FPGAs contained 1.5 times the logic of Splash 1. To improve interconnect flexibility, Splash 2 augmented the linear interconnect with a crossbar, allowing any FPGA to communicate directly with any other.

Concurrently with the Splash board development, the DEC PRL built a “Programmable Active Memory” (PAM) Perle-0 [6]. This system contained a 5x5 mesh of Xilinx 3020 FPGAs with a VME interface to a host processor. Applications such as 512-bit multiplication, data compression, and image processing were benchmarked on this early board, demonstrating speed-ups of 2-100 compared to software implementations.

Following the success of these early research machines, FPGA-based reconfigurable computers became commercialized, and today there are many custom RC boards being fabricated.

III. FIELD PROGRAMMABLE GATE ARRAYS

The Field-Programmable Gate Array (FPGA) is the computational unit for RC systems. The FPGA is a regularly tiled 2-D array of logic blocks. Each logic block is a Look-up Table (LUT), a simple memory that can store an arbitrary n-input Boolean function. The logic blocks communicate through a programmable interconnection network. The periphery of the FPGA contains “I/O blocks” to interface between the internal logic blocks and the I/O pins as shown in Fig. 2a. This simple, homogeneous architecture has evolved to become much more heterogeneous, including on-chip memory blocks as well as DSP blocks.

FPGAs can also be classified as either fine-grained or coarse-grained. Fine grain reconfigurable fabrics are very flexible. They can be used to implement any sequential and combinational Boolean logic function, but are slower and physically bigger in general like Xilinx 6200 series of FPGAs. On the other hand, coarse grain reconfigurable fabrics are faster, occupy smaller areas, but are limited to implementing only one of the predefined functions for example in ADRES [7] and RaPiD architectures.

Some RFs, such as modern FPGAs, contain a mix of both fine grain and coarse grain reconfigurable units. For example, Xilinx Virtex-4 FPGAs contains dedicated blocks for digital signal processing applications. This block can be programmed by the user to perform a combination of multiplication, addition or subtraction.

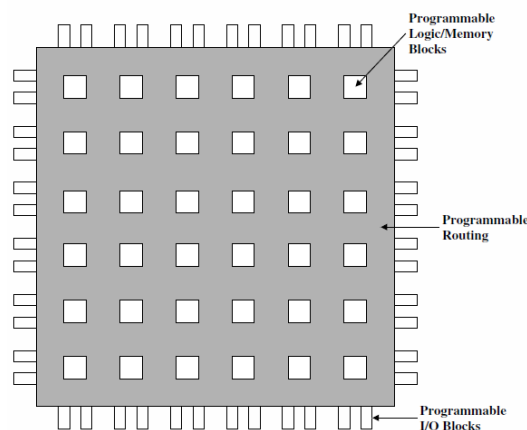


Fig. 3a A Generic FPGA Architecture

A. Programmable Logic Block

A basic programmable logic element generally contains some form of programmable combinational logic, a flip flop and some fast carry logic. The output of the block is selectable between the output of the combinational logic and the output of the flip-flop as shown in Fig. 3b. Nowadays, commercial FPGA devices generally provide a large amount of flexibility within the logic element. For instance a flip-flop can be made to operate as a simple latch, can be programmed to have several combinations of asynchronous or synchronous sets and resets, and can be negative- or positive-edge triggered.

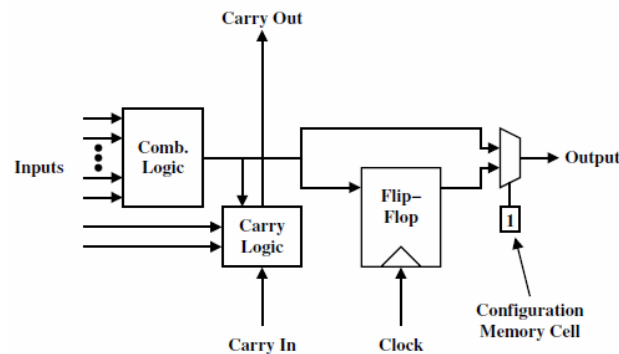


Fig. 3b A Generic Programmable Logic Block

The fig. 3b also illustrates that some form of configuration memory cell is used to control the output of the multiplexer. Reconfigurable computers have exclusively used SRAM- programmable devices as in Fig. 3c. Thus, the configuration of the FPGA, including the “object code” defining the algorithm is loaded onto the device and is stored in an on-chip SRAM. By loading different configurations into the SRAM, different algorithms can be executed. The configuration determines the Boolean function computed by each logic block and the interconnection pattern between logic and I/O blocks.

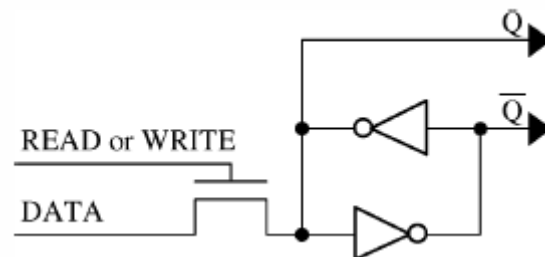


Fig. 3c A programming bit for SRAM-based FPGAs

B. Routing Resources

Interconnect resources are provided in a reconfigurable architecture to connect together the device’s programmable logic elements. These resources are usually configurable, where the path of a signal is determined at compile or run time rather than fabrication time. This flexible interconnect between logic blocks or computational elements allows for a wide variety of circuit structures, each with their own interconnect requirements, to be mapped to the reconfigurable hardware. Generally, some amount of routing is included within each logic cluster, so that the logic elements can be combined to form larger functions. External to the logic cluster is the more global routing architecture of the FPGA. To implement programmable routing, three basic switch types are used: multiplexers, pass transistors and tri-state buffers.

There are primarily four global routing architectures: Island, Cellular, Long-line, and Row. Generally island-style routing is used for FPGAs [8]. In this routing architecture, as shown in Fig.3d, logic clusters are surrounded by segmented horizontal and vertical routing channels. Each cluster connects to the routing through “connection boxes” and each segment in the routing can be connected to another segment through a “switch box”

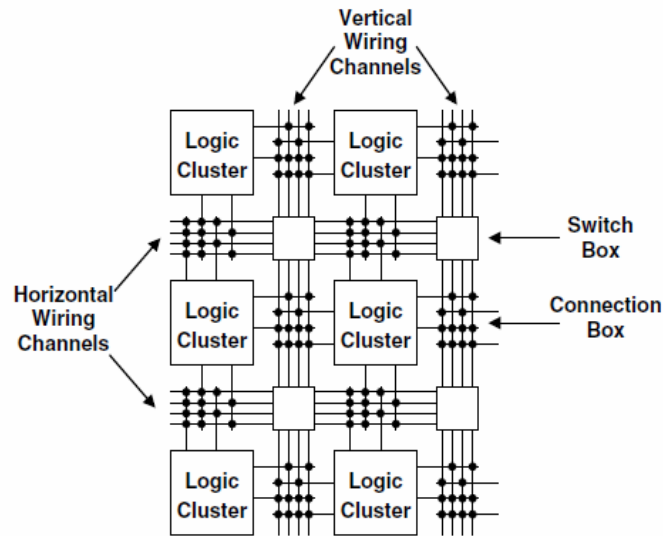


Fig. 3d A generic island-style FPGA routing architecture

IV. RC SYSTEM LEVEL ARCHITECTURE

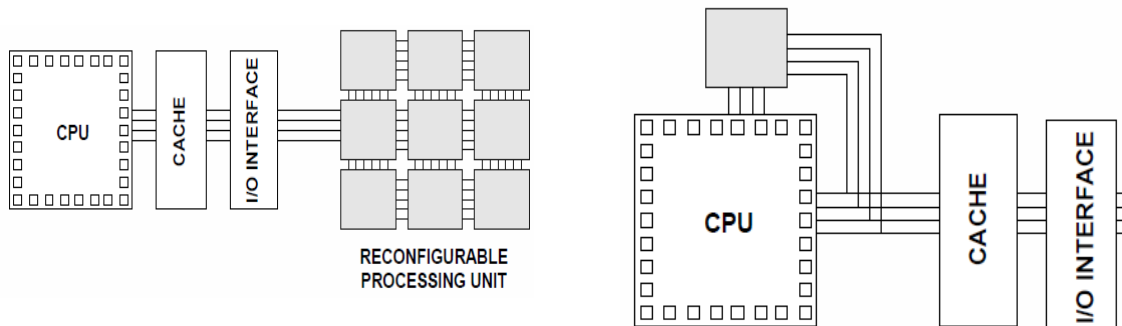
To ensure a high-quality product, diagrams and lettering MUST be either computer-drafted or drawn using India ink.

RC system typically consists of one or more processors, reconfigurable fabrics and memories. Thus, RC systems are often classified according to the degree of coupling between the reconfigurable fabric and the CPU.

The design of the actual computation blocks within the reconfigurable hardware varies from system to system. Compton et al. [9] presented the four classifications shown in Fig. 4(a-d). The first four classes of systems are characterized by the physical presence of a single controlling processor. They differ in the way that the processor communicates with the reconfigurable fabric (RF) of the system. In Fig. 4(a) the RF is in the form of one or more standalone devices. The existing input and output mechanisms of the processor are used to communicate with the RF. In this configuration, the data transfer between the fabric and the processor is relatively slow, so this architecture only makes sense for applications in which significant amount of processing can be done by the fabric without processor intervention. Emulation systems often take on this sort of architecture.

Figure 4b show an intermediate structure [7]. The cost of communication is lower than that of the architecture in Fig. 4a. In fig. 4b the reconfigurable unit may be used as a coprocessor. A coprocessor is able to perform computations with the constant supervision of the host processor.

The Figure 4c shows a system that integrates RF directly into the data path of the controlling processor as a functional unit. It allows the RF to have access to all local information about the running processor, such as the register file. Such tight integration ensures maximum integration between software and hardware. However, this tight integration limits RF speedup due to the lack of instruction level parallelism [8].



(a) External stand-alone processing unit

(b) Co-processor

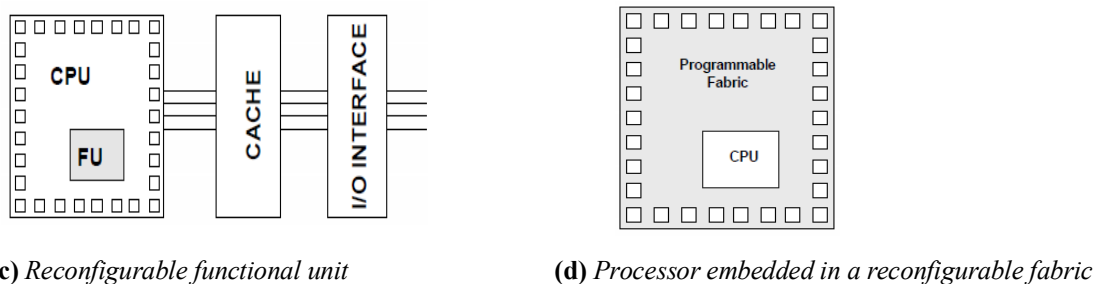


Fig. 4(a-d) Four classes of reconfigurable systems

Figure 4d, represents a new class of architecture that is made possible only with recent advances in reconfigurable hardware technologies. Instead of connecting RF to a processor system these machines embed processors within RFs. These embedded processors can either implemented physically or as soft core processors which are implemented using the resources of the RF itself [10]. Each of these styles have distinct benefits and drawback. In tighter integration there is a constant host processor intervention and amount of reconfigurable logic is quite limited. Whereas the more loosely coupled styles allows for greater parallelism in program execution but suffer from higher communication overhead.

V. RC PARALLELISM AND ITS APPLICATIONS

Instruction-level parallelism (ILP) is a measure of operations in a computer program that can be performed simultaneously. By creating more processing units a tradition microprocessor is able to execute more than one instruction in a clock cycle.

Unlike a traditional CPU, FPGAs do not run code. When we program RC systems we directly program the logic gates inside the FPGA. RC systems are tailored uniquely according to the specific application. In a conventional processor the level of parallelism is fixed at the time it is designed. On a FPGA these units are constructed from the configurable logic blocks. Their number, width and type are arbitrarily and can be optimized to the application. Moreover, FPGAs are free from instruction fetching operations because the instructions are built into the FPGA data path itself. Thus, an “instruction” as interpreted by the reconfigurable system is an arbitrary collection of related logic circuits, in which the number and type of arithmetic units is optimized to each application. Some of the important applications of RC Parallelism are Cryptography, Video image processing, DSP Applications, Network security etc.

A. Cryptography

Cryptography has been one of the most important drivers of computing technology for more than six decades. It includes code creation, analysis and breaking. Cryptography ensures secure communication of information among sender and recipients over insecure computer networks. The algorithms that form the basis for this security require large amounts of computing power and can require very long runtimes to complete. When deployed on FPGAs, these algorithms can use available FPGA parallelism with extreme efficiency. FPGA-based methods can be used to crack many data encryption schemes that once appeared to be strong [11].

For example, FileVault is based on the Advanced Encryption Standard (AES) and provides an encrypted file system for the Apple Macintosh operating system. Recovering FileVault passwords requires hashing a possible password with the SHA-1 hash function thousands of times. Using a 72-FPGA SC5 cluster, the speed of the FileVault key recovery application increases 498 times when compared to the original software implementation running on an Intel Core i7 processor at 2.93 GHz. This results in the reduction of runtime from 21 hours to just 2.5 minutes [12].

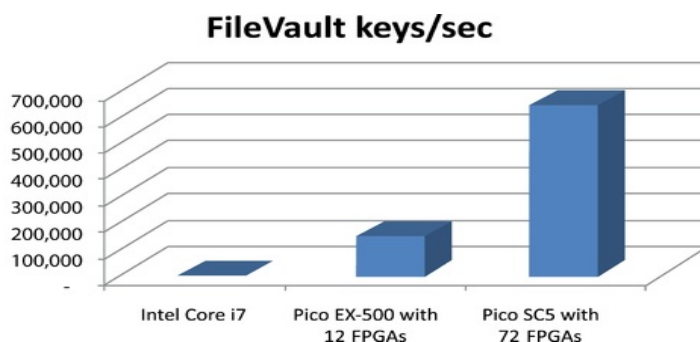


Fig. 5 FileVault cracking is accelerated 498x using an FPGA cluster

B. Video image processing

The goal of image processing is to robustly extract useful, high level information from images and videos. Reconfigurable computer are particularly well suited to meet the requirements of many video and image processing applications [12]. Reconfigurable computer combine the flexibility of a software approach found in a workstation, with the performance of a hardware solution found in an ASIC. Moreover, the low development cost makes them ideal for video and image processing applications. By using RC systems rather than general purpose processors, speed-ups are typically two orders of magnitude.

C. DSP Applications

DSP involves the representation of signals digitally as sequences of numbers or symbols and the processing of these sequences to extract information from the signals or to synthesize signals with desirable properties either as completely new signals or from existing signals. The effectiveness of RC for DSP is mainly due to the parallelism that can be exploited in DSP applications [13]. As a simple example consider how a finite-impulse response (FIR) filter can be implemented in a hardware. A FIR filter implements the following equations:

$$y(k) = \sum_{n=0}^{N-1} a(n)x(k-n)$$

Whereas in reconfigurable hardware, FIR filters can be implemented using parallelism so that N multiplications and N-1 additions are performed each cycle with a result provided every cycle [14].

D. Network security

To check for vulnerabilities such as whether the encryption methods being used are obsolete or to check for weak passwords in the network security, a method called “Network Intrusion Detection System” is used. The increase in attacks from the Internet like viruses, spam, malwares as well as other malicious activities has given rise to the need of protection methods which help protecting user’s system. Network intrusion detection system (NIDS) is one of the solutions that can deeply inspect all payloads of the packet.

String matching is used by Network Intrusion Detection Systems (NIDS) to inspect incoming packet payloads for hostile data [15]. The speed of network has reached several gigabits per second (Gbps) and could be higher in the future. Therefore, it is difficult for software NIDS solutions to handle such large amount of data and moreover every string of bytes of the traffic will be compared with a large number of rules which makes the software solutions skip packet due to the limitation in memory. String-matching speed is often the main factor limiting NIDS performance. String-matching performance can be dramatically improved by using Field-Programmable Gate Arrays (FPGAs) by making use of the true parallelism offered by the hardware [16].

VI. CONCLUSION

Reconfigurable computing is becoming an important part of research in computer architectures and software systems. By placing the computationally intense portions of an application onto the reconfigurable hardware that application can be greatly accelerated. This is because reconfigurable computing combines many of the benefits of both software and ASIC implementations. Like software, the mapped circuit is flexible, and can be changed over the lifetime of the system or even the lifetime of the application. Similar to an ASIC, reconfigurable systems provide a method to map circuits into hardware. Reconfigurable systems therefore, have the potential to achieve far greater performance than software as a result of bypassing the fetch-decode-execute cycle of traditional microprocessors as well as possibly exploiting a greater degree of parallelism.

Reconfigurable hardware systems come in many forms, from a configurable functional unit integrated directly into a CPU, to a reconfigurable co-processor coupled with a host microprocessor, to a multi-FPGA stand-alone unit. The level of coupling, granularity of computation structures, and form of routing resources are all key points in the design of reconfigurable systems. The use of heterogeneous structures can also greatly add to the overall performance of the final design.

The strength of Reconfigurable Computers in the field of cryptography, signal and image processing etc. have been apparent since their inception but the ongoing research and use of multi-million gates have expanded the application space and is opening doors to new avenues. Reconfigurable systems offer best of both the worlds: Run-time programmability and Hardware level performance.

REFERENCES

- [1] Guo, Z., Najjar, W., Vahid, F., and Visser, K: ‘A quantitative analysis of the speedup factors of FPGAs over processors’. Proc. Int Symp. On FPGAs (ACM Press, 2004)
- [2] Elbirt, A. J. and Paar, C., “ an FPGA implementation and performance evaluation of the serpent block cipher”, *ACM/SIGDA International Symposium on FPGAs*, 2000, 176-184

- [3] Rencher, M. and Hutchings, B. L., “ Automated target recognition on SPLASH2”, *IEEE Symposium on Field-Programmable Custom Computing Machines*, 1997, 192-200
- [4] Duncan Buell, Jeffrey Arnold, and Walter Kleinfelder, *Splash 2: FPGAs in a custom computing machine*, Wiley- IEEE Computer Society Press, 1996
- [5] P. Bertin, D. Roncin, and J. Vuillemin, Introduction to programmable active memories, *Systolic Array Processors* (J. McCanny, J. McWhirther, and E. Swartzlander Jr., eds.), *Prentice Hall*, 1989, pp. 300-309
- [6] Goldstein, S.C., Schmit, H., Budju, M., Cadambi, S., Moe, M., and Taylor, R.: ‘ PipeRench: a reconfigurable architecture and compiler’, *Computer*, 2000, 33, 940, pp. 70-77
- [7] Betz, V. and Rose, J.: ‘FPGA routing architecture: Segmentation and buffering to optimize speed and density’, *ACM/SIGDA International Symposium on FPGAs*, 1999, pp. 59-68
- [8] Mei, B., Vernalde, S., Verkest, D., De Man, H., and Lauwereins, R.: ‘ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix’, *Lect. Notes Comput. Sci.*, 2003, 2778
- [9] Compton, K., and Hauck, S.: ‘ Reconfigurable computing: a survey of systems and software’, *ACM Comput. Surv.*, 2002,34, (2), pp. 171-210
- [10] Xilinx, Inc, *Microblaze Processor Reference guide*, June 2004
- [11] David Hulton, Pico Comptuing, “Accelerating cryptography with FPGA clusters”, <http://mil-embedded.com/articles/accelerating-cryptography-fpga-clusters/>
- [12] P.M. thanas and A.L. Abbott, “Real-time image processing on a custom computing platfor,”, *IEEE Computer*, Vol. 28, pp. 16-24
- [13] Sanjit K. Mitra, *Digital signal processing : A computer-based approach*, second ed., *McGraw-Hill*, New York, NY, 2001
- [14] Russell Tessier and Wayne Burleson, *Reconfigurable computing for digital signal processing: A survey*, *The Journal of VLSI Signal Processing* 28 (2001), 7-27.
- [15] C.R. Clark and D.E. Schimmel, “ A pattern-matching co-processor for network intrusion detectino system”, *Proc. Int. Conf. on Field Programmable Technology*, IEEE, 2003
- [16] Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar: ‘FPGA Implementation of AES Engryption and Decryption’ *International Conference on “Control, automation, communication and energy conservation “*,2009