IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 2, February 2016.
www.ijiset.com
ISSN 2348 – 7968

VHDL Implementation of Evolutionary Algorithm in the Evolutionary Design of Combinational Circuits

Vidyavathi.A[1], Chidambaram.S[2]

[1] Electronics and Communication Engineering, Adhiyamaan College of Engineering Hosur,Tamilnadu

[2] Electronics and Communication Engineering, Adhiyamaan College of Engineering Hosur,Tamilnadu

Abstract

Now a days space vehicles and other electronic hardware also demands that the architectures should be small, speed in operation, low power consumption, small in area and be reconfigurable in unexpected environments. The evolvable hardware (EHW) or evolutionary algorithm used to design the desired circuit automatically**.** Genetic algorithm is the commonly used evolutionary algorithm. EHW, because of its binary representation, which matches perfectly with the configuration bits used in FPGA. The use of reconfigurable FPGA system will reduce the time required to re-design the system each and every time. The paper discuss about the Cartesian genetic programming for the proposed adder evolvable unit.

Key words: *Evolvable hardware (EHW), Adder, genetic algorithm, reconfigurable FPGA*

1. Introduction

Evolvable hardware is the combination of artificial intelligence, reconfigurable hardware, autonomous system and fault tolerance, and also it uses to improve performance of the circuits [1]. Evolution of the circuits can be achieved by using different stochastic algorithm. Genetic algorithm is most commonly used evolutionary algorithm [2]. Genetic algorithm can be applied to EHW because of its binary representation. FPGA configuration bits are perfectly matches with the EHW binary representation.

Evolvable hardware is mainly used in two areas (i) new solutions can be automatically generated (ii) autonomous adaptive devices can be implemented [3] .for case (i), in design phase we are using evolutionary algorithm and this will be applied in this paper [4]. For case (ii), evolvable hardware is automatically adopted for changing environments .EHW can be implemented by using two methods (i) Intrinsic Evolution: every configuration bits are downloaded and each bits are tested physically. (ii) Extrinsic Evolution: In this every bits are tested by using software. Intrinsic evolvable hardware is performed more accurately in real world applications.

Genetic algorithm is a solution that is point in the search space is represented by zeros and ones. Each zero's or one's called as string. The set of strings are called as the chromosomes. The combination of chromosomes called the individuals or population [5].The population size is depending upon the length of the string and problem being optimized. The evolution process is usually starts from the generation of random numbers. Objective function value is used to evaluate the strings. In GA terminology the string objective function value is referred the fitness of the string. Based on the fitness value we select the combination of random numbers [6].

Reconfigurable FPGA is used as an evolvable hardware. Now a day Xilinx is the most popular model for evolvable hardware [7]. The FPGA can be divided into two parts. i.e. (i) programmable area,(ii) non programmable area [8]. Case (i) includes the parts of configuration interface and configuration logic. Case (ii) includes the parts of routing resources, portion of IOB's and CLB's etc. In reconfiguration mode interface circuits helps to give configuration bits to configuration logic and write them into proper locations in configuration memory.

Virtual reconfigurable circuit is implemented on the top layer of FPGA for speed evolution [9]. Once VRC is uploaded on FPGA, the following units are created at specified positions. They are (i) Programmable Interconnection Network (PIN), (ii) Configuration memory, (iii) Configuration port and (iv)an Array of Programmable Elements (PE).Based on our requirement; we can use the programmable elements [10].

This paper structured as follows. The following section discuss about the GA and an Evolvable Hardware. In section 3 we will discuss about the Virtual reconfigurable circuit. Section 4 deals about proposed unit for adder. Section 5 describes about the results and discussion. In section 6 deals with the conclusion and future work.

ISSN 2348 – 7968

2. GA and an Evolvable Hardware

Genetic algorithm is first introduced by "JOHN HOLLAND" in 1975. Genetic algorithm is inspired by the biological evolution process. Generally it uses the concepts of "genetic inheritance" and "natural selection". Genetic algorithm is one of the populations of candidate solution. To solve the problem it is evolved good solutions. Each candidate solution has a different set of properties, which can be altered and mutated. The evolution process is generally starts by using the generation of random numbers for individuals, and it is an iterative process. To achieve the good solution for our problem we use the following steps.

2.1 Random number generation

Random number generator (RNG) is a device it is used to generate the sequence of numbers or symbols that does not have any pattern [11]. We can generate the random numbers by using two methods (i).True random number generator (ii).Pseudo random number generator. In case (i) we generate the random numbers by using the some physical phenomenon for case (ii) we use the algorithms for generation.

2.3 Selection

Roulette wheel selection is used to select the best chromosome. In this method the chromosomes or individuals are placed in roulette wheel according to their fitness. Virtual roulette wheel is stopped when we select the corresponding segment. The process is repeated until we select the desired segments.

2.4 Cross over

Crossover operator is one of the genetic algorithm operators. In this two individual chromosomes (parents) are combined and produced a new off springs (child) [12].This new off spring is one of the better one if it is take the better characteristics from chromosomes.

2.5 Mutation

Mutation is the process of random change of genes in chromosome [13].If we get the perfect solution of our problem mutation will help to stop the random number generation.

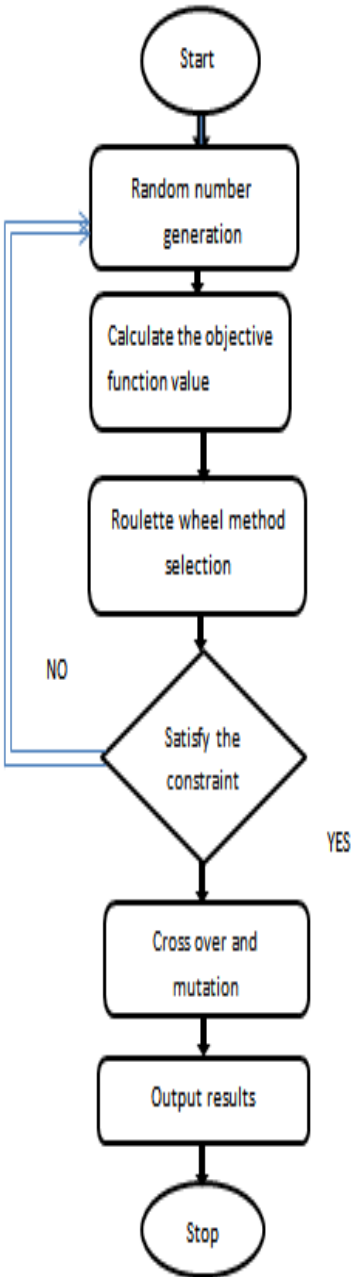

Fig. 1 Flow chart for Genetic algorithm

Reconfigurable hardware is manufactured by using the evolvable hardware .By using this hardware; we can get the efficient desired circuit automatically if the program is failure. Evolvable hardware is controlled by under the evolutionary algorithm.

To speed up the operation we implement the evolvable hardware in FPGA (Field Programmable Gate Array) [14]. We have three types of evolution to implement the evolution phase in evolvable hardware [15]. (i) Intrinsic evolution (ii) Extrinsic evolution (iii) Complete evolution. Intrinsic evolution is on–line evolution. In this design each individual tested by outside of the hardware. In

extrinsic evolution we use off-line evolution. In this each individual is tested by using the software [16]. Whole evolvable design is implemented in complete hardware evolution.
In our paper we are going to use the intrinsic evolution because it was very suitable for real world applications.

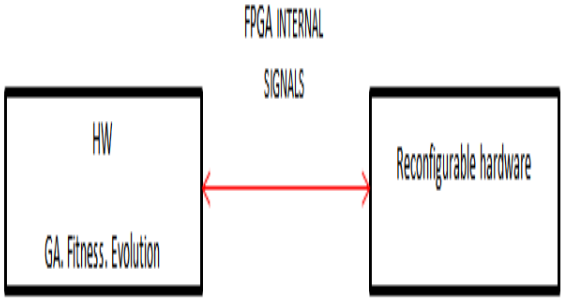

Fig. 2 Intrinsic evolution model

3. Virtual Reconfigurable Circuit

Virtual Reconfigurable circuit was introduced by Sekanina.L in 2006. VRC is the combination of software reconfigurable devices and genetic algorithm [17]. The structure of the VRC circuit is determined by downloading the binary bit strings called architecture bits. This is one of the new kinds of reconfigurable platform using the conventional FPGA. Once VRC is uploaded into FPGA the following parts are created at specified positions. (i)An array of programmable elements, (ii) Configuration port, (iii) Configuration memory and (iv) programmable interconnection network. VRC architecture is similar to the Cartesian genetic programming architecture. Multiplexers are used as routing elements. Programmable elements have a number of registers to store the information. Programmable interconnection network is used to connect the programmable elements. Bits are treated as chromosomes and these are downloading the device. Based on the bits the device structure will be change. The performance of the device is improved as the chromosomes are evolved by calculating the fitness. The process is repeated until reach the desired function. In VRC configuration memory is controlled by a genetic unit. The advantage of VRC is that the routing circuits, arrays and the configuration memory can be exactly designed as requirement of application.

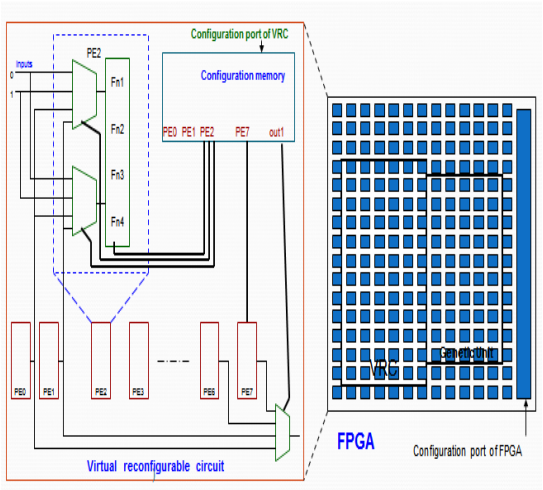

Fig. 3 Virtual Reconfigurable circuit

4. Proposed Unit for Adder

Evolvable hardware is used to increase the speed and performance. Evolvable hardware is mainly suitable for the space missions. The commonly used evolutionary algorithm is genetic algorithm. By using genetic algorithm we can get the desired circuit automatically if the circuit is failure. Evolutionary algorithm works based on the principle of natural evolution and self-adoption. Evolvable hardware is worked based on genetic algorithm because of its binary representation.

Fig.4 shows 2-bit adder circuit and its configuration In (CSG) Cartesian genetic programming with Parameter X={X1, X2, X3, X4, X5}, nc=4, nr=4, ni=4,no=3,nf=5.The configuration information's :a0,b0,1a1,b1,4,a0,b1,3,a1,b1,6,1,8,2,1,b1,1, a1,b1,2,a0,b0,8,3,3,5,4,a1,2,7,5,2,3,11,3,9,8,3,12,10,6.TheLast integers determined the connection of outputs. Gates 13, 14, 15 are not utilized.

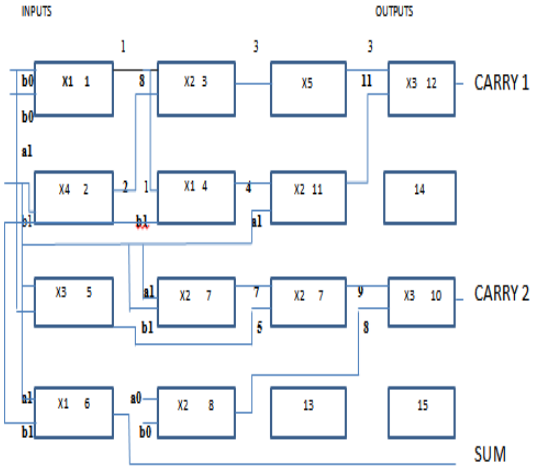

Fig.4 proposed circuit

ISSN 2348 – 7968

The length of the chromosome measured in the gene is

$$\nabla = nc.nr(nn+1) + no \quad (1)$$

Where nr is the number of rows, nc is the number of columns, no is the number of outputs, nn is the node of inputs. The length of the chromosome measured in genes, Λ =51.

4.1 Design considerations

All most all Boolean functions are designed by using Shannon's effect. The circuit is implemented by using a minimum number of gates at least $2^{ni}ni^{-1}$. The size of the design circuit is measured by following expression.

$$H = 2^{no2ni}$$

$$C = nf^{nc.nr}(ni+nr)^{nn.nr(nc-1)}ni^{nn.nr} \quad (2)$$

Where C is in the fact the size in searches

In proposed circuit logic functions are expressed in terms of functions.F1 is represents the XOR funcition.X2 shows the AND function. OR function is showed by X3. X4 represent the NOT function.X5 showed as wire. The combinations of inputs and outputs for 2 bit adder are showed below

Table.1: combinations of inputs and outputs

Case	I/P1 a0	a1	I/P2 b0	b1	C 2	C 1	Sum
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	1
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	0	1	1
7	0	1	1	1	1	0	0
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	1
10	1	0	1	0	1	0	0
11	1	0	1	1	1	0	1
12	1	1	0	0	0	1	1
13	1	1	0	1	1	0	0
14	1	1	1	0	1	0	1
15	1	1	1	1	1	1	0

5. Results and Discussion

5.1 Random number generation

Fig 5 shows Top level random number generation module.clk is an input signal giving to the module and random_num[175:0] is the output

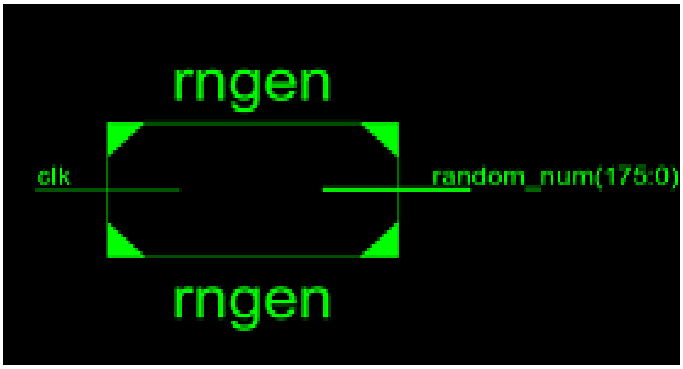

Fig.5 Top level random number generation module view

.

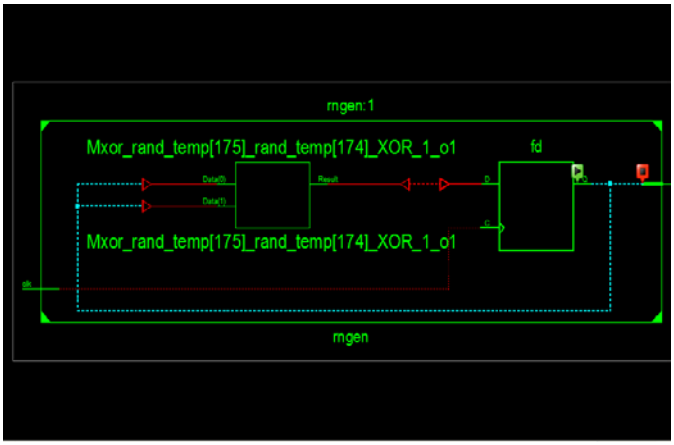

Fig.6 Schematic diagram for random number generation

The different 44x4 bits outputs are generated for every rising edge clock by using XOR logic .The simulated waveform is shown in fig.7. This random number we considered as initial population of individuals. This random number generation process until termination of Genetic Algorithm (sufficient fitness achieved).One more new reproduction generator single point Crossover is shown in fig 5.Best_indv is taken as an input & random- pattern O is an getting output for single point crossover.

ISSN 2348 – 7968

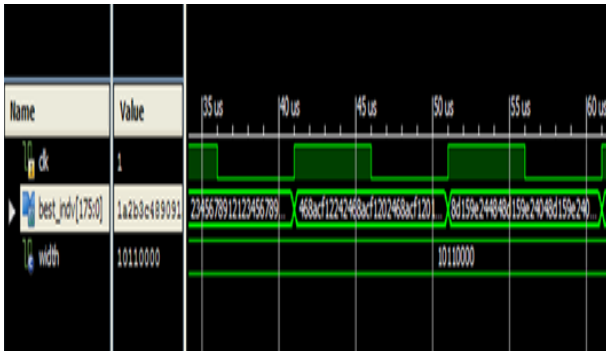

Fig.7 Simulation result for random number generation

5.2 Crossover

Crossover is an alternative method to generate random numbers. Crossover simulated result shown in fig 8. best_indv signal represents input data and random_pattern represents output data. Single point crossover technique used for to generate random numbers.

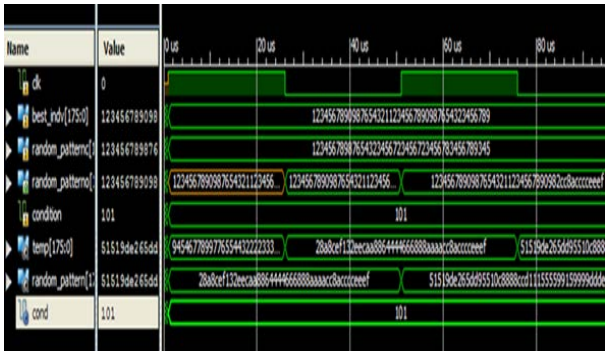

Fig .8 Simulation result for crossover

5.3 Fitness calculation

Fig 9 shows Top level fitness calculation module. Here best_indv (175 bits) is an input signal and best_fit (3 bits) is an output signal

Fig.9 Top level fitness calculation module view

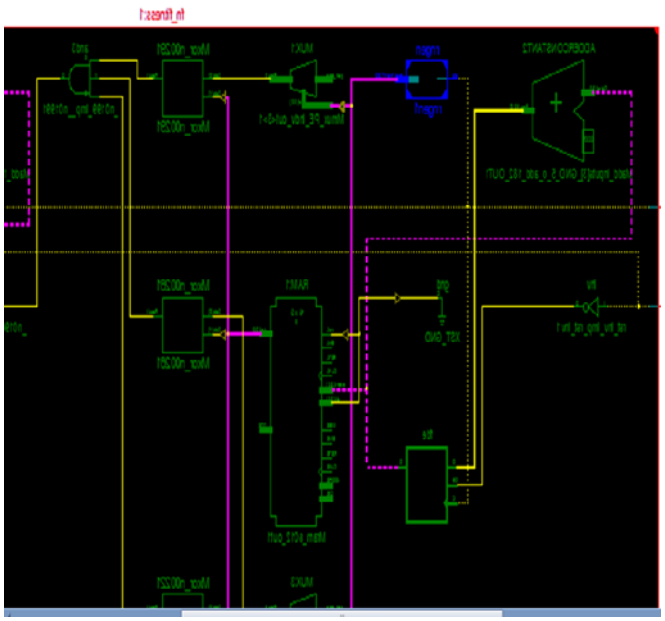

Fig.10 Schematic diagram for fitness calculation

Schematic Diagram for Fitness calculation is shown in fig 10. Fitness Calculation Simulated output window is shown in fig 11, here best_indv considered as an input and best_fit representing output, it shows each and every best_indv's fitness status. Pe_indv[15:0] representing Programmable Elements input status. pe_indv_out representing pe_indv output status, inputs[3:0] representing a0,a1,b0,b1 status. Fns [15:0] represents Logical Function condition.

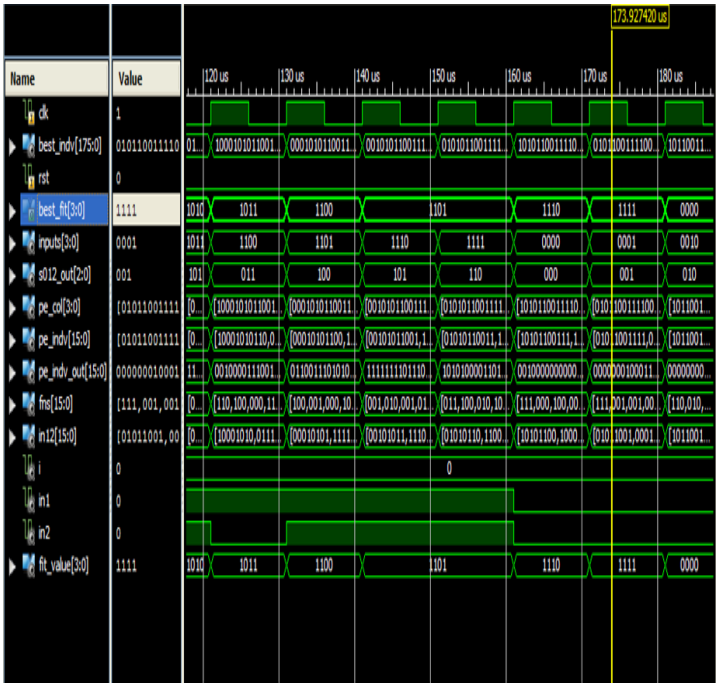

Fig 11 Simulation result for fitness calculation

ISSN 2348 – 7968

6. Conclusion

Proposed 2-bit adder is designed by using 16 number of PEs. The configuration of every PE is represented in the chromosomes as input1, input2, and function. 11 bits are needed to configure a single PE. The length of the chromosome is $\Lambda=51$ and size of the search space is up to $|C| = 10^{11}$. For dynamic adaption EHW using bio-inspired techniques like Genetic Algorithm (GA). Hsclone GA, Roulette GA, and Compact GA this are the different genetic algorithms used for implementation. Here, Hsclone GA is used for implementation. Implementation of Evolvable system using FPGA is cost-effective and flexible. There are various approaches for Xilinx FPGA to use it for re-configuration. The use of proposed intrinsic run-time configurations will allow overcoming the redesign the circuit every time if the circuit is failure. FPGA based EHW development and proposed evolution unit set-up can be applied for future space application.

Random Number generation was used to get individual genes .The coding was done in VHDL and successfully verified using Isim simulator. Fitness value is obtained using counter witch increments if selection outputs and referred outputs are equal for a given input combination of 2-bit adder. The coding was done in VHDL and successfully verified using Isim simulator. Spartan-II forms the genetic unit. The genetic operators were used for getting configuration bits. These configuration bits are sent from Spartan to Virtex. FPGA-Based Run-Time Configuration System is used for implementing EHW

6.1 Future Work

The use of proposed intrinsic run-time configurations will allow one to overcome at the certain level of the scalability problems and reduced to re-manufactured the system every time. FPGA based EHW development and proposed evolution unit set-up can be applied for future application.

Moreover with the logic complexity, the configuration bits number increases, Cartesian Genetic Algorithm structure cannot be used for representing the reconfiguration bits. Other way of representation will be searched and developed at a later stage.

Hsclone GA, roulette algorithms are used in Evolvable Hardware, for evolving Combinational Circuits. Time requirement for this Genetic Algorithms is more .The future work as carried out as to reduce time requirement.

The design of Evolvable system on FPGAs can be made easier by using CAD tools. In that case user can specify the target application at a higher level of abstraction and the design system will automatically generate the VHDL code of the application. Code can be subsequently synthesized for various target flat forms. Hence the Proposed evolvable unit can be redesign according the needs of the given application.

Once a 2-bit adder is evolved using genetic algorithm, other combinational logic and sequential logic circuits will be the next step to evolve.

References

[1] Parisa Soleimani 1, Reza Sabbaghi-Nadooshan 2 , Sattar Mirzakuchaki3,and Mahdi Bagher “Using Genetic Algorithm in the Evolutionary Design of Sequential Logic Circuits” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, may 2011.

[2] S. Karakatic, V. Podgorelec, M. Hericko “Optimization of Combinational Logic Circuits with Genetic Programming” Elektronika Ir Elektrotechnika, ISSN 1392 1215,vol.19 Mar 2013

[3] K.H.Chong,I.B .Aris,M.A.Sinan “Digital Circuit Structure Design via Evolutionary Algorithm Method”Journal of Applied science 7(3):380-385,2007.ISSN 1812-5654

[4] Rakesh Kumar, Senior Member, IACSIT and Jyotishree, Member, IACSIT “Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms” International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.

[5] Cyrille Lambert, Tatiana Kalgovnova,and Emanuele stomea” FPGA –Based System for Evolvable Hardware”processing of world acodomy of science,Engineering and technology volume12 march 2006 ISSN 1307-688.

[6] Firas Alabsi, Reyadh Naoum “ Fitness Function for Genetic Algorithm used in Intrusion Detection System ”Vol. 2 No. 4; April 2012 .

[7] Prashant Sen , Priyanka Pateriya “ Implementation of Generic Algorithm Using VHDL on FPGA” International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011 ISSN 2229-55518.

[8] Cyrille Lambert, Tatiana Kalganova, and Emanuele Stome0“FPGA-based Systems for Evolvable Hardware” International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:1, No:12, 2007

[9] Omar E. Elnokity, Imbaby I. Mahmoud, Mohamed K. Refai, Hasan M. Farahat "Hardware Implementation of Virtual Reconfigurable Circuit for Fault Tolerant Evolvable Hardware System on FPGA" International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-2 Issue-12,OCT 2014.

[10] L.Sekanina,"Towards Evolvable IP cores for FPGAs"In:proc. Of The 2003 NASA/DoD Conference on Evolvable Hardware,Los Alamitos,US, ICSP, pp. 145-154, ISBN 0-7695-1977-6. 2003

[11] Chandra Sekhar, K. Saritha Raj "An efficient pseudo random number generator for cryptographic applications" International journal of engineering and advanced technology (IJEAT)ISSN: 2249 – 8958, Volume-4 Issue 1,Oct. 2.

[12] Zdenec Vasicaek and Lukas Sekania "An Evolvable Hardware System in Xillinx Virtex-II pro FPGA"Innovative Computation and Application, Vol. 1,No. 1 ,2007.

[13] Lukas Sekanina, Tomas Martinek "Extrinsic and Intrinsic evolution of multifunctional combinationals modules"IEEE Congress on evolutionary computation,vol.16 may 2006.

[14] Miller, J-Thomson, P:"Cartesian Genetic Programming "In proc of the 3rd European Conference on Genetic Programming, LNCS 1802, Springer verlag ,Berlin ,2000,pp. 121-132.

[15] Yang Zhange,Stephen L,Smith,AndyM.Tyrrell "Digital Circuit Design using Intrinsic Evolvable Hardware"proceeding of the 2004 NASA/DOD Conference on Evolution Hardware (EH'04)0-7695-2145-2/04 2004 IEEE.

[16]L.Sekanina,s Feriedl "On Routine Implementation of vvirtual Evolvable devices using COMBO6" In:proc. Of The 2004 NASA/DoD Conference on Evolvable Hardware,Los Alamitos,US, ICSP, pp.63-70, ISBN 0-7695-2145-2. 2004.

[17] C. Aporntewan, P. Chongstitvatana, "An On-Line Evolvable Hardware for Learning Finite-State Machine," in Proceeding of int, pp.13-15, 2000.

[18] L Sekania , Stepan Friedl "An Evolvable Combinationl unit for FPGAs –Draft".Computing Informatics,Vol. 23,2004,461-486,V 2005-july-7.

[19]Carlos A. Coello Coello,Alan D. Christian and Arturo Hernoandez Aguirre "Automated design Of Combinational Logic Circuits using G enetic Algorithms".

[20]JBits: Java based interface for reconfigurable computing Steve Guccione, Delon Levi and Prasanna Sundararajan Xilinx Inc., 2100 Logic Drive San Jose, CA 95124 (USA).