International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-5 Issue-4, April 2016

Comparison of Hsclone and Roulette Genetic Algorithms on the Application of Combinational Circuits

Suhas S, Gayatri Malhotra, Rajini V.H

Abstract—Future planetary and deep space exploration require robust methods of operation to operate spacecraft in the outer atmosphere without any variations or faults. The best fault tolerant method which can be used for operations of this kind is the class of Genetic Algorithms (GA) which are a sort of evolutionary algorithm. In this domain of operation, a combinational circuit is designed by the method of Cartesian Genetic Programming (CGP). The Circuit after the design is fed to the two GAs, namely, HsClone and Roulette. The main advantage in this use of GA is the likely determination of the best possible circuit within the space of a thousand circuits. The combinational circuit design is applied to both the algorithms and tested for fitness. After the required fitness is obtained, both the algorithms are compared with respect to their cumulative generational fitness and other allied aspects. The better algorithm will hence be determined to integrate it into the future design of spacecraft hardware. This is expected to help the spacecraft recover from Single Event Upsets (SEU) which usually occur due to hostile temperature conditions and outer atmospheric radiation.

Index Terms—Cartesian Genetic Programming (CGP), Genetic Algorithm (GA), Evolvable Hardware (EHW), Reconfigurable FPGA, Evolutionary Algorithm (EA)

I. INTRODUCTION

Due to the rapid development in remote sensing, communication and deep space science technology there is an important need to develop a sophisticated circuit that enhances speed, memory, faster computation and reduced latency. The circuits which satisfy these constraints are the digital circuits. To satisfy the criteria of digital circuits there requires an advanced level of optimization along with high level functional ability so that the design criteria are not compromised. The optimization is a very important factor in the design of digital circuits in the spacecraft as power and area plays a very decisive role for the embedding of the circuit on to OBC. Since optimization is playing a very important role the need to develop stochastic and probabilistic type of algorithm is very much necessary, one such type of algorithm which takes that role is evolutionary algorithm. Evolutionary algorithm works on the principle of natural Adaptation and evolution and EHW completely follows evolutionary algorithm in its operation. EHW is an active field that combines reconfigurable hardware, artificial intelligence, fault tolerance and autonomous system. EHW is a replicated evolution to search for new hardware configurations. There are different types of evolved algorithms but one which is used in large scale for hardware configurations is Genetic Algorithm. Genetic Algorithms are used widely because its binary representation matches perfectly with reconfiguration Bits of FPGA [1] [11] [12]. EHW is a reconfigurable hardware which will reconfigure by itself under the influence of evolutionary algorithm. The better search mechanism for hardware can be done through software which is extrinsic evolution type. In the general search mechanism for a better hardware configuration we will also use intrinsic evolution along with the extrinsic one. Extrinsic evolution employs circuit models and simulators to evaluate the fitness of various hardware configurations through the use of software, so that last generated data will be converted into control bits and downloaded on to the reconfigurable hardware. On the other hand intrinsic evolution meanwhile deals with downloading and physical measuring of fitness [8]. Based on the performance in unpredictable environment the intrinsic evolution is much more accurate when compared to extrinsic one. Evolvable hardware is a reconfigurable hardware that self operates under the influence of evolutionary algorithm [1] [11] [10].

Genetic algorithms are a probabilistic type of algorithm which mainly contracts on reproduction, crossover and mutation.GA mainly operates on the population of different solutions and the fittest among them is found to obtain the better approximation to the solution. At the new set of generation is created by the method of selecting of individuals according to the level of fitness in their set of problem domain and breeding together using operators obtained through natural genetics [11]. HsClone GA is half-sibling clone algorithm which mainly deals with the crossover technique which manages the assignment of fitness probabilities to chromosomes. Here each individual will be evaluated to the fitness and the best individual will be chosen based on the highest fitness. If the exact fitness is not obtained then crossover and mutation will be performed to enhance the fitness level so that it reaches the maximum value required for the particular combinational circuit [4][13]. Roulette GA is a probability type of GA where a roulette wheel will be employed to determine the fitness value. In this algorithm also, in order to improve the fitness crossover and mutation is performed compulsorily by assigning the crossover and mutation rate. In this algorithm bubble sorting technique is employed to obtain the best chromosomes with highest fitness [4] [13].

Manuscript published on 30 April 2016.
***** Correspondence Author (s)
Suhas S, Student, Department of Electrical Communication Engineering, RNS Institute of Technology, Bangalore, India.
Gayatri Malhotra, Scientist/Engineer, ISRO Satellite Centre, Bangalore, India.
Dr. Rajini V.H, Associate Professor, Department of Electrical Communication Engineering, RNS Institute of Technology, Bangalore, India.

II. EVOLUTIONARY ALGORITHM AND ITS COMPONENTS

Evolutionary algorithm is a computer based algorithm which mainly based on the principles of self-adaptation and natural evaluation. In the evolutionary algorithm each chromosomes generated by the computer produces a distinctive solution and the fitness is generated. Based on the fitness value, a best individual will be selected with highest fitness. Here the best individual will be selected to the next population that is,

$$P (t + 1) = S (E (v (p (t)))) \quad (1)$$

Here P (t) is a population of generation's t, v (.) is a random variation operator, E (.) is an evaluation operator and S (.) is a selection operator.

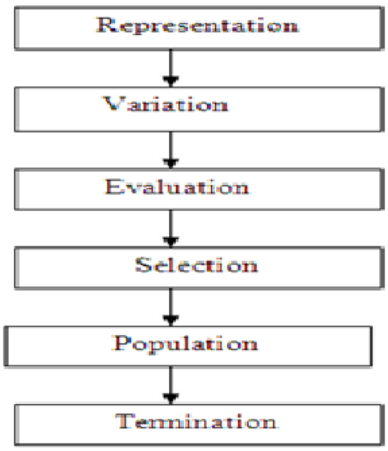

Fig (1): Components of Evolutionary Algorithm

The components of evolutionary algorithm are representation, variation, evaluation, selection, population and termination. Representation mainly indicates the encoding of the data structure needed to provide a solution. Variation indicates a process of providing a new solution to the existing one, Evaluation mainly focuses on providing a numeric value to the obtained solution and measuring its quality. Selection process mainly concentrates on obtaining the better solution among the rest using the parameters of fitness. Population is a set of solution as EA evolves and finally termination when the
required conditions are satisfied.

III. GENETIC ALGORITHMS

Genetic algorithms is a stochastic search method that operates on the chromosomes generated from the system and applies the theory of survival of fittest to determine the fittest chromosomes. Here chromosomes can be referred to as a population. The population contains a different set of individuals generated randomly and genetic algorithm applies fittest theory to determine the better individual. The main 3 operators in GA are reproduction, crossover and mutation. Reproduction is a way an individuals are copied depending on the objective function value. Crossover is a method of re-ordering of bits or interchanging bits, there are mainly two types one is one-point crossover and other is two-point crossover. One point crossover is where a bit is copied from one parent to another depending on the crossover point, whereas two point crossover is where two bits from the parents are crossover so that new off-springs are generated. Mutation mainly deals in flipping one or more bits. GA helps in direct manipulation of bits obtained from the population. GA is used in FPGA because its binary representation matches perfectly with the reconfiguration bits of FPGA.

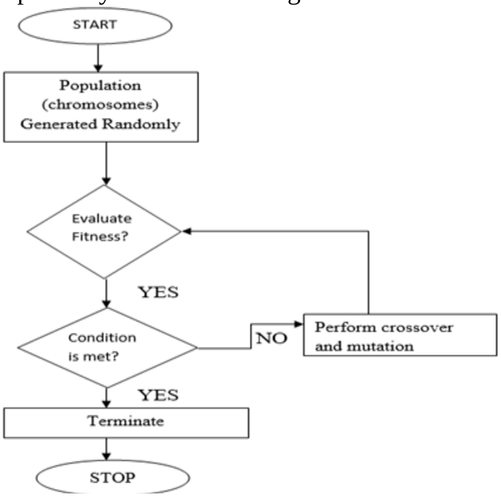

Fig (2): Flow chart of Genetic Algorithm

As shown in Fig (2) a randomly individual is generated randomly by the system and is tested for fitness, the fitness value is set according to the combinational circuit design. For example a 2-bit multiplier has 4 inputs (i.e. 16 input combinations) and 4 outputs so the fitness value will be 16x4 = (number of inputs X number of outputs) = 64, if the fitness value satisfies the condition set by the combinational circuit then the algorithm will be terminated or else recombination operation such as crossover and mutation is performed until the fitness is obtained or if the algorithm is terminated externally.

A simple genetic algorithm procedure is explained below

Step 1: Generate the individuals randomly which is mainly Referred to as population.

Step 2: Evaluate the fitness of each individual generated.

Step 3: If an individual satisfies the fitness then such individual is considered as best individual OR perform the recombination operation such as crossover and mutation on the individuals.

Step 4: Repeat this procedure until the individual is fit enough.

Step5: Terminate when the conditions are satisfied.

IV. CARTESIAN GENETIC PROGRAMMING

Cartesian Genetic Programming (CGP) is used for evolved gate level design as well as functional level design. In CGP the candidate solutions are of fixed length represented through direct oriented graph. Here the word Cartesian is used because it represents 2-dimensional grid of nodes. The digital circuit when represented as CGP will have n rows and m columns. The main rule of CGP will connect bits from the column to its left. In CGP all the graphs will be represented 2-D computational grid nodes [3] [5].

The circuit which is reconfigured by CGP will have columns represented by n_c and rows represented by nr. The input to the CGP circuit is represented by n_i and similarly output is represented by no [9]. CGP has n_n nodes as node input and single output. Each node in the CGP can be programmed to n_f functions defined in the F set.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The interconnectivity of the circuit in the CGP is defined by L parameters which determines how the preceding column outputs is connected to current column. The length of the chromosomes in the CGP
is denoted by

$$Len = n_r n_c(n_n + L) + n_o \quad (2)$$

The maximum number of computational nodes is $Ln = n_r.n_c$, the level back parameter is used to show the connectivity to The graph.

The maximum number of computational nodes is $Ln = n_r x n_c$, the level back parameter is used to show the connectivity to the graph. For the level back parameter L=1 to indicate that node gets the input only from the column of nodes which is to its left or through primary inputs. An example is shown in Fig (3). When L=2 a node can have its input connected to the output of any node in immediate left to the 2 columns to the primary input, when the nodes are connected only to the left then L = n_c [7].

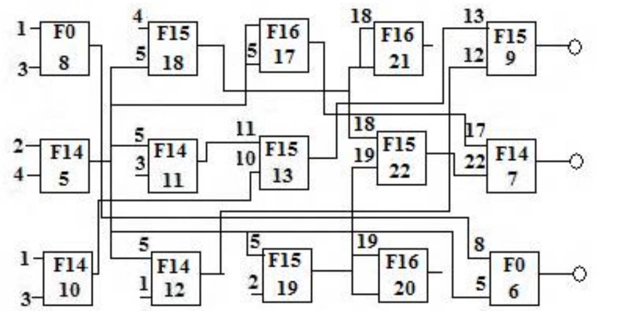

Fig (3): An example of CGP circuit of a 2-bit adder

V. HSCLONE GENETIC ALGORITHM

HsClone GA is a half-sibling clone approach with an efficient crossover technique that manages the assignment of fitness probabilities of the population. In this GA the fitness probabilities of the designed circuit is already pre-defined and the individuals generated randomly are tested for the fitness. If the fitness of the individual matches with predefined fitness then that individual is considered as the best individual, otherwise the individual will be subjected to crossover and mutation until the maximum (desired) fitness is obtained. The best fit will be according to the designed circuit (for 2-bit multiplier it will be 64), the worst fit of the circuit will be 25 percent of the best fit (for 2-bit multiplier it will be 16). The crossover and mutation will be reduced with the increase in the fitness value. This GA requires memory for storing the population and selection of the appropriate parameters. The flow chat is shown in the figure (4) below

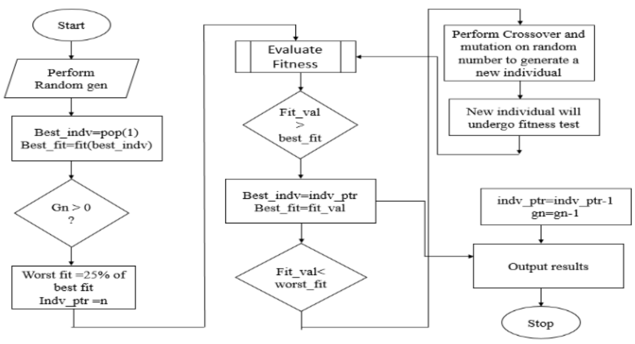

Fig (4): Flow chart of Hsclone Genetic Algorithm

VI. ROULETTE GENETIC ALGORITHM

In Roulette Genetic algorithm both the crossover and mutation is very important parameter, a population containing 100s of individuals are generated randomly and the fitness of all the individuals are determined by the fitness selection criteria. Then the probability of the fitness is calculated by

$$P_i = f_i / \sum^{\infty}_{i=1} f_i \quad (3)$$

After determining the probabilities of all the fitness choose 2 probabilities such that P1 has highest probability and P2 has the second highest probability. In the roulette genetic algorithm perform the crossover to obtain the children from the parents based on the certain conditions. The crossover rate should be ideally less than 30 %. After crossover perform mutation which is with the children, mutation rate and the size of the population, the mutation rate will be ideally 10 % of the Maximum. After mutation a new set of population will be generated and the fitness of the population will once again be determined, after which both the best and the worst fit values will be known and repeat the loop for n/2 (here n is the size of the population) times or until the required fitness is obtained.

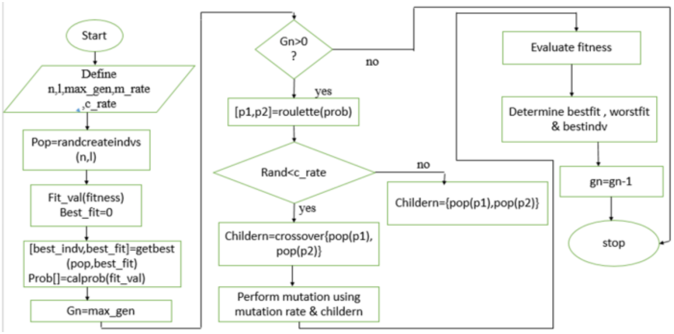

Fig (5): Flow Chart of Roulette GA

In this GA the error value is given the highest rank and repeats n times in the roulette wheel, if the error value is small then it is given the second highest rank and will be duplicated n-1 times. The ensuing size of the wheel is n (n+1)/2. In roulette GA assignment of the probability and selection of the best fitness is performed using Bubble-Sorting Method. The drawback of this GA is an additional memory of n (n+1)/2 are required in the selection of individuals for the crossover. All the probabilities which are generated can be scaled to the integer value for the implementation in FPGA for the fitness calculation.

VII. IMPLEMENTATION OF 2-BIT MULTIPLIER USING HSCLONE AND ROULETTE GENETIC ALGORITHM

The combinational circuit which is designed in this paper is a 2-bit multiplier which consists of 4 inputs (a0, a1, b0, b1) and 4 outputs (Carry3, Carry2, Carry1, Sum). A simple pictorial representation of the implementation of 2-bit multiplier using the HsClone and Roulette GA's is shown in fig (6)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

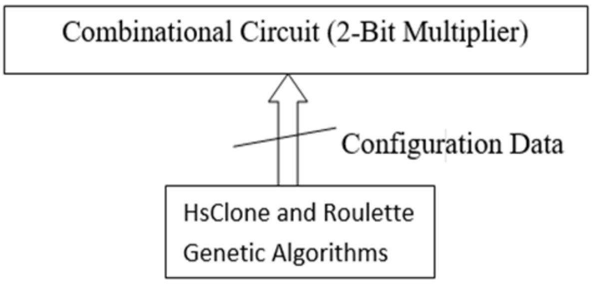

Fig (6): Implementation of HsClone and Roulette Genetic Algorithms on a combinational circuit

The gate level implementation of a 2-bit multiplier using HsClone Genetic algorithm is shown in fig 7 below. In the figure (7) for the gate level representation of a 2- bit multiplier the inputs are numbered as 0,1,2,3. The table 1 shows the numerical value and corresponding representation in CGP. The functional value representation for the designed 2-bit multiplier is indicated in the table 2 below. In Fig 7 the outputs carry3, carry2, carry1 and sum are obtained through the gate 15, gate 14, gate 13 and gate 4 respectively. The chromosome pattern for the above CGP circuit shown in fig 7 is (131, 121, 031, 021, 443, 553, 663, 931, a21, 871, bc2, 471).

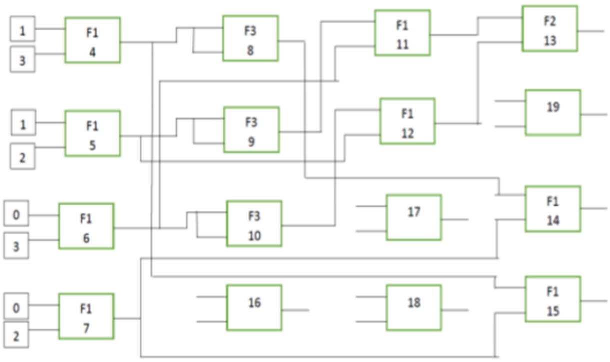

Fig (7): 2- Bit multiplier using HsClone Genetic Algorithm

TABLE (1): INPUT REPRESENTATION

Value	Representation
0	**a0**
1	**a1**
2	**b0**
3	**b1**

In the chromosome pattern first two digits indicate the input pattern and the last digit indicate the function bit. For example in the first chromosome 131, the first 2 digits means 1 and 3 are the inputs and their representation according to table 1 will be a1 and b1. The third bit is 1 and their equivalent functional value representation is and gate as per table 2. In this way all the chromosomes are represented. Similarly the gate level representation of 2-bit multiplier using roulette genetic algorithm is shown in fig (8) below

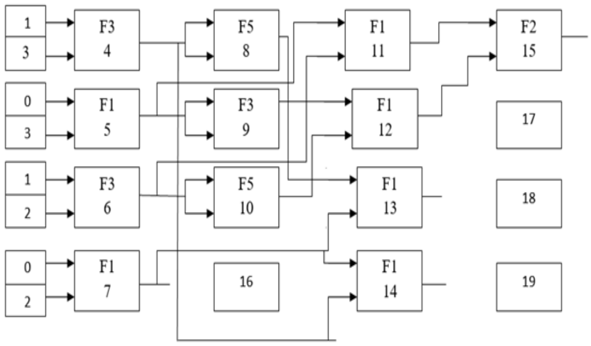

Fig (8): 2-Bit multiplier Using Roulette GA

For the above figure the input level and functional level representation is same as per table (1) and table (2). The outputs of the 2 bit multiplier (carry 3, carry 2, carry 1 and Sum) are obtained through gate 13, gate 15, gate 14 and gate 8 respectively. The comparison graph of fitness v/s maximum number of generations is shown in fig 9. The chromosome pattern for Roulette GA is (133;031,123,021,445,553,665,561,9a1,871,741,bc2) In the comparison graph the fitness value was higher for roulette when compared with respect to HsClone GA. The comparison Table (3) for HsClone and Roulette GA is shown below.

TABLE (2): FUNCTION REPRESENTATION

Value	Representation
F0	ip1 xor ip2
F1	ip1 and ip2
F2	ip1 or ip2
F3	ip1 nand ip2
F4	ip1 xnor ip2
F5	ip1 nor ip2
F6	ip1
F7	ip2
F8	not ip1
F9	not ip2
F10	not ip1 or ip2
F11	ip1 or not ip2
F12	not ip1 and ip2
F13	ip1 and not ip2
F14	not ip1 xor ip2
F15	ip1 xor not ip2

TABLE (3): COMPARISON TABLE OF HSCLONE AND ROULETTE GA

Parameter	Hsclone	Roulette
Circuit	2-bit multiplier	2-bit multiplier
Length of the individual	144	144
Type Of Crossover	Single Point	Single Point
Number of generations required	116	60
Time Required	180 ns	500 ns
Fan-Out Gates	113	187
Net Skew	0.347 ns	0.467 ns

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Crossover rate	50%	50%
Mutation rate	0.69%	0.69%
Number of global clocks	12%	12%
Number of 4 input LUT‘S	3%	1%
Number of Slice Flip flops	2%	5%
Number Of Bits Flipped In Mutation	1	1
Device Used	Spartan 3	Spartan 3

In the comparison table numbers of generations indicate at the time of maximum fitness and time required is for one generation. The comparison Graph show in fig 9 indicates that Hsclone algorithm has obtained a maximum fitness at 116th generation and Roulette algorithm has obtained a maximum fitness at 60th generation respectively. The maximum fitness is 64.

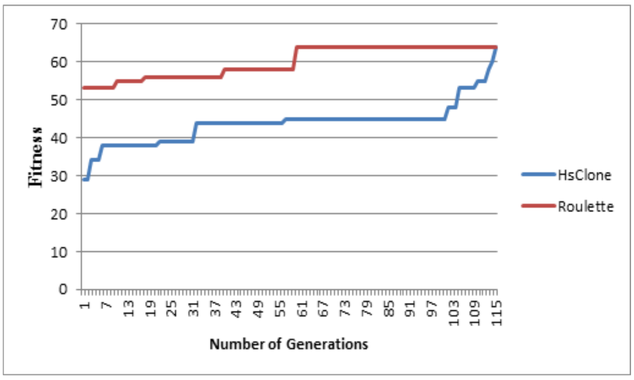

Fig (9): Comparison Graph between HsClone and Roulette GA in terms of number of generation.

In the Fig (10) the comparison is made between HsCloneand Roulette GA in terms of time. the HsClone GA requires 180ns to evaluate one generation of chromosomes whereas Roulette GA requires 500ns to evaluate one generation of chromosomes.

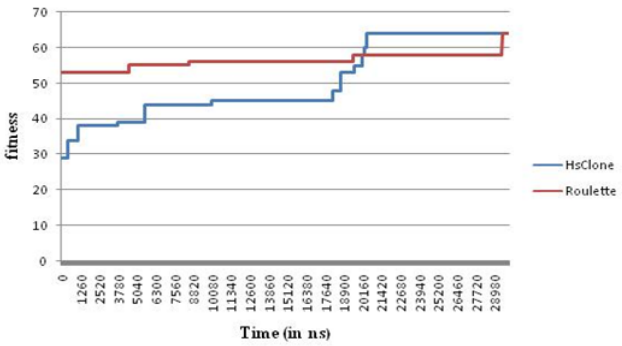

Fig (10): Comparison Graph between HsClone and Roulette GA in terms of time

VIII. RESULTS AND DISCUSSIONS

In the Fig (11) random numbers are represented as population which is of 144 bits, to generate a random number LFSR technique is employed.

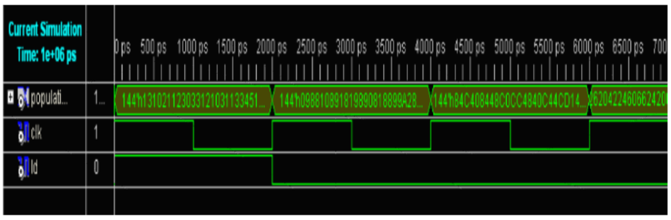

Fig (11): Simulation Waveform of Random number generation

For the crossover operation shown in Fig (12) there will be2 individuals namely indv1 and indv2 respectively and the crossover will be performed between the 2 individuals depending on the crossover rate. During this operation crossover rate was 50% with single point crossover technique being employed.

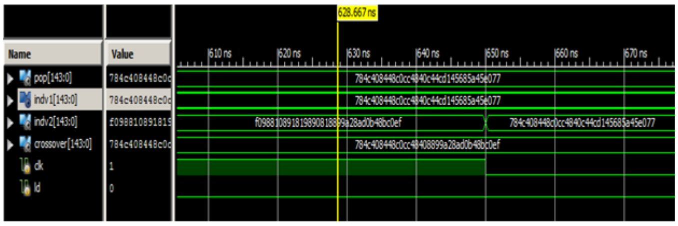

Fig (12): Simulation Waveform of Crossover operation

The simulation waveform of mutation operation is shownin Fig (13); here the random numbers (pop) will be mutated by the mutation pattern (mutp) to perform the mutation. Here mutation can be performed for single-bit flipping or multi-bit flipping. Here the words inside the bracket indicates the ones used in the simulation waveform.

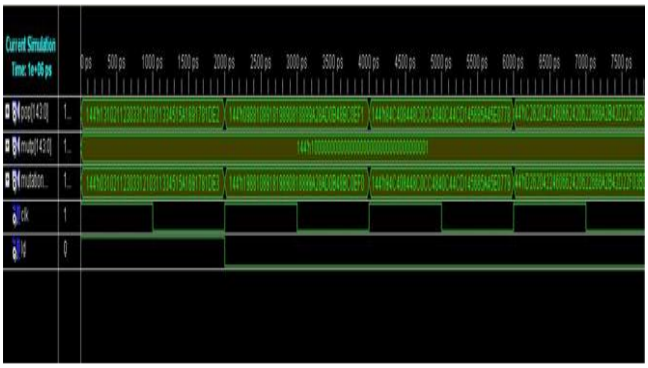

Fig (13): Simulation waveform of Mutation operation

In the HsClone Genetic Algorithm shown in Fig (14), pop 1shown in fig indicates the random number which is having a fitness of 0, this random number is crossover with the best individual to generate a new individual pop 2 after which its fitness fit 2 is raised to 28, since the fitness level is not satisfactory once again the crossover is performed with respect to random numbers generated inside the population to generate a 3rd individual as pop 3 at which the fitness level fit 3 was 14. Finally the mutation was performed to generate an individual pop 4 to generate a fitness fit 4 as 64. Finally bubble sorting technique was performed to determine the best among the fitness and since fit 4 had the highest fitness the best fit was 64 and best indv was pop 4. The HsClone operation was performed to 2-bit multiplier hencemaximum fitness was 64.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

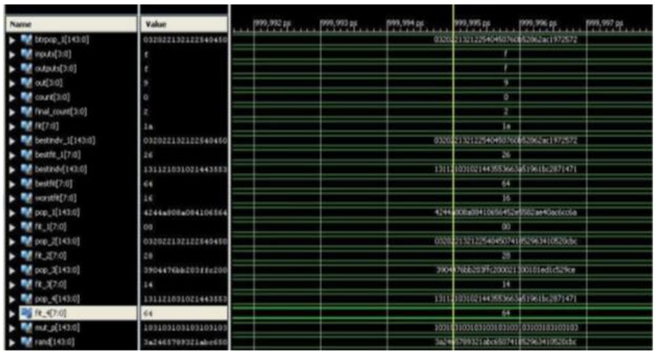

Fig (14): Simulation Waveform of HsClone Genetic Algorithm

Here the population (pop) of 100 individuals will be generated and best fit among all the individuals in the population will be selected, then roulette wheel selection is applied to determine the best among the probabilities, then in the next step crossover is performed to obtain children and mutation is performed finally to obtain the best fit. Here only for the first iteration the fitness of the population is found in later iterations the comparison of the fitness will be done on the present and the previous fitness until the required maximum fitness is obtained. The present best individual and previous best individual will be taken as P1 and P2 respectively according to Fig (15) so that the convergence to the maximum fitness will be high.

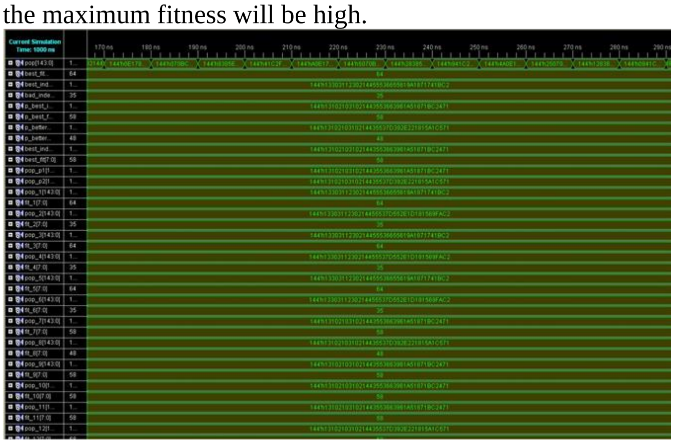

Fig (15): Simulation Waveform of roulette GA

IX. APPLICATIONS OF GENETIC ALGORITHMS ON SPACE RELEVANCE MISSIONS

The future planetary and deep space exploration requires arobust method of operation to operate on the outer atmosphere without any variations or faults. Due to the advancement in space technology and also due to the high requirement of various complex mission operations there is a stressing need to reduce the faults which are occurring mainly due to the radiation, temperature and other outer atmospheric effects. These types of faults causes bit- flipping which leads to Single Event Upset (SEU). To overcome these effects various techniques have been developed and one such technique will be the use of Evolvable Hardware (EHW). EHW is an evolutionary algorithmic method which operates on any given possible circuits and determines the best among the circuits through the imposed specifications. Genetic algorithms play a vital role in EHW in the pursuit of determining the best circuits among thousands. The overall implementation of the combinational circuit can be performed through the method of Cartesian Genetic Programming (CGP). Evolvable hardware can be obtained through the algorithms which are used after running through certain number of generations. These procedures have been used to suit the deep remote missions taken by the spacecraft and to ensure no fatal errorsoccurs in the OBC (On-Board Computer), as the spacecraft is millions of kilometres away from the earth. Therefore in the remote operations for the spacecraft since both adaptability and safety is very important evolvable hardware increases the potential and re-configurability to tackle inappropriate controls which are highly fatal and disastrous.

X. CONCLUSION AND FUTURE SCOPE

Evolutionary Algorithms is a best fault tolerant method used in the spacecraft and genetic algorithm is a part of evolutionary algorithms. Evolvable hardware is a self-reconfigurable hardware used. Genetic algorithms are mainly are of 3 types Hsclone, Compact and roulette. The combinational circuit is designed using Cartesian Genetic Programming (CGP) which is then applied to Hsclone and Roulette GA. After the analysis of both the algorithms HsClone is found to be better for any space related operations because of its less complexity in terms of implementation and the convergence to result is more faster when compared to that of Roulette and also due to other comparison parameter which is indicated in table 3. The Future Scope of operation is another Genetic algorithm called compact can also be implemented for the same combinational circuit and the comparison can be done on all the 3 operations. In this paper combinational circuit is implemented and the circuit design can also be extended for sequential circuit as well.

REFERENCES

1. D B Verneker. Gayatri Malhotra, V Colaco "Reconfigurable FPGA using Genetic Algorithm" International Conference and Workshop on emerging Technology (ICWET 2010).
2. Vedavati A, Meena K.V, Gayatri Malhotra "VHDL Implementation of Genetic Algorithm for 2 bit adder" International conference on Electronics and communication engineering, 20th May 2012, Bangalore, ISBN: 978-93-81693-29-2.
3. L Sekania, Stepan Friedl "An Evolvable Combinational unit for FPGAs Draft". Computing Informatics, Vol. 23, 2004, 461-486, 2005-July-7.
4. Zhenhuan Zhu, David Mulvaney, and Vassilios Chouliaras, "A NovelGenetic Algorithm Designed for Hardware Implementation", World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol: 1, No: 12, 2007
5. Lukas Sekanina and Stepan Friedl, "An Evolvable combinational Unit for FPGAs", Computing and Information, Vol.23, 2004, 461-486, V2005- Jul7.
6. Julian. F. Miller and P. Thomson, "Cartesian genetic programming, inGenetic Programming", Proceedings of EuroGP 2000. Springer-Verlag, 2000, pp. 121,132.
7. M. Oltean, C.Grosan, "Evolving digital circuits using Multiexpression programming", Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware, 24-26 June 2004, pages: 87-94
8. H. Mostafa, A. Khadragi, Y. Hanafi, "Hardware implementation of
9. Genetic Algorithm on FPGA". Zdenek Vasicek and Lukas Sekanina, "An evolvable hardware system in Xilinx Virtex II Pro FPGA", Int. J. Innovative Computing and Application, Vol. 1, No. 1, 2007"
10. R.S.Zebulum, A.Stoica, D.Keymeulen, M.I. Ferguson, A. Thakoor, "Evolvable, Reconfigurable Hardware For Future Space Systems", Jet Propulsion Laboratory, California Institute of Technology. K. Chong, I. Aris, S.M.Bashi, "Application of Evolutionary Algorithm in NOptimizing Digital Structure Design", AIML Journal, Vol. 6, December 2006.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

11. K. Chong, I. Aris, "Digital Circuit Structire design via Evolutionary
12. Algorithm Method". Sharawi, M.S., Quinlan, J. and Abdel-Aty-Zohdy, H.S., "A hardware implementation of genetic algorithms for measurement characterization",
13. IEEE 9th International Conference of Electronics, Circuits, and Systems, Dubrovnik, Croatia, 3, 2002, pp.1267-1270

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)