Implementation of Genetic Algorithm for Automatic Test Pattern Generation

MrsRachna singh, Dr.Arvind Rajawat

Abstract: This paper presents, genetic based algorithm for random test pattern generation .Genetic algorithm solve many search and optimization problem, effectively. In genetic algorithm, the optimized test vector is generated, which enhances the fault coverage and improve the global search .As, a result a new random-based test pattern generation technique based on GA was presented. Experiment results showed that the genetic algorithm improved the ability of global search and increases the fault coverage. This algorithm improves the test size with a factor of about 25% in comparison with other approaches for ATPG**.**

Keywords: Automatic Test Pattern Generation (ATPG), Genetic Algorithm (GA), Design Validation, Simulation Based Approaches, Evolutionary Algorithm.

1 INTRODUCTION:

In present scenario, digital systems are extremely intricate and increasing in complexity, which are required for use in widening range of domestic and industrial application. So to ensure reliability of these digital circuits, it is necessary to test their performance to identify any defects prior to using them in a fully operational environment.

These circuits are tested by test vectors. The test vectors are generated by efficient automatic test pattern generator (ATPG).The generation of test pattern with high fault coverage rate is a very expensive process for large circuits. An efficient ATPG tool reduces the test pattern generation time and cost, beside the high fault coverage rate.

There are many approaches for ATPG,like deterministic approach, simulators etc.The aim of this technique should be both to reduce execution time and to improve fault coverage.

Genetic algorithm described by Goldberg[13] is specially suited to solve large scale combination optimization problem.GA have been successfully applied in different areas of VLSI design ,especially in test branches such as test pattern generation [5].

2 MATERIAL AND METHODS:

2.1 Automatic Test Pattern Generation (ATPG):

ATPG is a method used to find an input (or test vectors) sequence that, when applied to a digital circuit, enables tester to distinguish between the correct behavior and the faulty circuit behavior caused by defects. The efficiency of this method is measured by using fault coverage, computational time and the length of test set.

Breuer [8] uses a fault simulator to evaluate sets of random vectors and to select the best vector to apply in each time frame. Weighted random pattern are interfaced with fault simulators [15] and high coverage is obtained for combinational circuits. The test generator [16] is built around fault simulator; the next candidate vector is generated using the hamming distance with the previous test vector.

2.2 Genetic Approach for Test PatternGeneration:

In past, test generation using deterministic & fault oriented algorithm is highly complex and time consuming new approaches are needed to augment the existing techniques, to reduce execution time and to improve fault coverage.GA was first used for simulation based test generation in [17].Several

Dr.ArvindRajawat is an Associate Professor ,MANIT, BHOPAL,India email:arvind.rajawat@gmail.com

Mrs.Rachna Singh is an Assistant Professor ,SISTEC-E , BHOPAL,Indiaemail:ahana10jan@gmail.com

approaches to test generation have been proposed in [4], [9].In reference [4], [9] the fitness evaluation and population scoring is low cost and only based on the fault coverage of each test vector. The disadvantage of the technique is that if a dropping fault simulation is used, experimentally after almost 10 generation, the generated vectors stop detecting remaining faults. This method has resulted in better final test set, but it is very expensive. A new operator is used in [4], in which, after each generation, the best vector in population is put on the final test set and then rescored with a new decreasing fitness.

2.2.1 Introduction:

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. As you can guess, genetic algorithms are inspired by Darwin's theory about evolution. Simply said, solution to a problem solved by genetic algorithms is evolved.As a result, a new random based test pattern generation technique based on GA is presented. Experimental result show that this algorithm increases fault detection rate while test size decreases.

This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

2.2.2 Outline of the Basic Genetic Algorithm

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population
3. **[Replace]** Use new generated population for a further run of algorithm

3 USING GENETIC ALGORITHM IN ATPG:

In this paper, a genetic algorithm approach to ATPG is used .The set of solutions called population is the test vectors. The purpose of this algorithm is to finding optimal solution with high convergence speed. A random population of n chromosome is generated and fitness of each chromosome in the population is evaluated. New population is created by repeating selection, crossover, mutation and acceptance. The test vector’s generated by this algorithm cover’s maximum number of faults in the VLSI circuits. Results show both a reduction in test sizes and an improvement in fault coverage compared with other results for combinational benchmark circuits.

The pseudo code of genetic algorithm for ATPG is shown in fig1.The description of each step of the algorithm is as follows:-

3.1 Generation of the Initial Antibodies:

In this step, the antibodies (test vectors) are createdrandomly on feasible space. Population size shouldbe large enough in order to ensure adequate diversity; however, it is a trade off between getting higher convergence rate with larger search space and less genetic operation time. Population size in algorithm of Ivask, 1998 is constant value for all circuits. Experiments have proven that required population size increases with increasing test vector length. In this paper we used a more exact population size as shown in table 1

TABLE 1

POPULATION SIZE FOR DIFFERENT VECTOR LENGTH

Population Size	Vector Length
8	<4
16	4-16
16	17-49
24	50-63
24	64-99
32	>99

3.2Calculating the Fitness:

The fitness function provides a quantification of the quality of the chromosome. It is the fitness of the chromosome that determines whether the chromosome will be selected to produce offspring and quantifies its chance for survival among the other chromosome in the population to the next generation. The fitness function is problem specific. In this paper fault simulation with fault is with fault dropping is used in order to evaluate the test vectors. The score given to each individual is equal to the number of fault it detects, the fitness function for given test vector is calculated by eq(1) given below:

$$F(x) = \frac{\text{No of detected faults by test vector}}{\text{Total no of faults}}$$

3.3Creating a New population:

New population is created by repeating the following steps until the new population is complete:

3.3.1 Selection:

The genetic algorithm uses selection operator to simulate natural evolution. In GA, individual with high fitness is inherited to the next generation with greater probability. Usually, chromosomes with high fitness are selected for crossover to converge faster to best solution. Chromosomes with high fitness should not be selected for mutation to prevent the danger of diverting from good solutions in the search space. Therefore, chromosomes with low fitness are usually selected for mutation. All selection methods are based

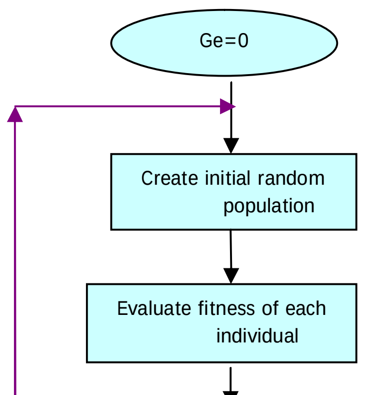
Ge=0
Create initial random population
Evaluate fitness of each individual

Fig1. Genetic Algorithm Flow Chart

on the fitness of chromosomes. The disadvantage of selecting chromosomes with high fitness is the probability of less diversity in the search space.Therefore, chromosome with low fitness is usually selected for mutation. All Selection methods are based on the fitness of the chromosome.

In this paper Roulette wheel selection is used to select the individuals.

Roulette Wheel Selection:

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where are placed all chromosomes in the population, every chromosome has its place big accordingly to its fitness function, like on the following picture.

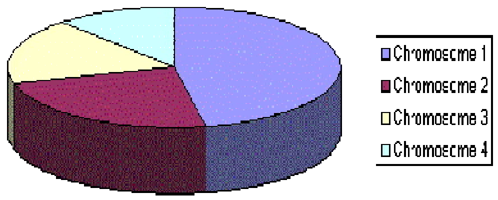

Fig 2

Then a marble is thrown there and selects the chromosome. Chromosome with bigger fitness will be selected more times.

This can be simulated by following algorithm.

1. **[Sum]** Calculate sum of all chromosome finesses in population - sum ***S***.
2. **[Select]** Generate random number from interval ***(0,S)*** - ***r***.
3. **[Loop]** Go through the population and sum finesses from ***0*** - sum ***s***. When the sum ***s*** is greater then***r***, stop and return the chromosome where you are

Of course, step **1** is performed only once for each population.

3.3.2Crossover

Crossover is the key to genetic algorithm, power that is to exchange corresponding genetic properties from the two parents, to allow useful genes on different parents to combine in their offspring. Most common crossover types are one-point, two-point, uniform crossover. In this paper, as shown in fig (3), two-point crossover is used

3.3.3Mutation:

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. It says how often will be parts of chromosome mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed.

Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to random search.

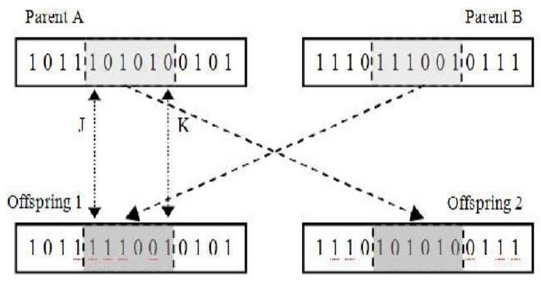

Fig3. Two-point crossover

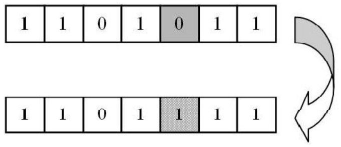

Fig4. Mutation in binary string

4.EXPERIMENT AND RESULTS:

The experiment is carried on the combinational circuits. These circuits are simulated for fault coverage ie, the number of faults covered by applying a test pattern to the circuit.

The Genetic algorithm is simulated using c language .The experiments are carried out on ISCAS'85 combinational benchmark circuits & other combinational circuits.

1. Experimental result shows that the test pattern size has been reduced up to 25% in comparison with other methods like verifault simulators as shown in table2.
2. Similarly the results show that the fault coverage is increased in genetic algorithm in comparison with verifault simulator as shown in table 3.

5. CONCLUSION:

This paper uses genetic algorithm for automatic test pattern generation in VLSI circuits. ATPG tools can reduce the amount of effort and cost of test generation.

TABLE 2
REDUCTION IN TEST SIZE USING IGA

Our experimental results showed that genetic algorithm based method are more efficient in test generation in comparison with other approaches for ATPG.

The experimental results shows an

- Average 25% reduction in test size
- Increase in fault coverage.

TABLE 3
RESULT OF FAULT SIMULATION ON COMBINATIONAL CIRCUITS

Circuit	Total no of Detected fault by Verifault Simulator	Total no of Detected fault by GA	% Improvement in fault detected
C17	19	22	4.9%
1 bit adder	22	24	8.3%
2 to 1 MUX	13	14	7.2%

REFERENCES

[1].Azimipour, M. and M. Eshghi, Parallel Circular- Scan Architecture. Asian Network for Scientific Information (ANSI), Journal of applied sciences, pp: 1-8,2008.
[2].Azimipour, M. Eshghi and A. Khademzadeh,. A Modification to Circular-Scan Architecture to

Circuit	No of Nodes		Test size		Improvement
	input	*output*	*Verifault fault simulator(Rad,M.A,S.M. Eshgh,2007)*	*Genetic Algo*	
c 17	5	2	4	3	25%
1 bit adder	3	2	4	3	25%
2 to 1 MUX	3	1	5	4	20%

improve test data compression. *IEEE CS Proc. Of 15th Int. conf. on advance computing and communication*,pp: 27-33,2007.
[3].Al-Yamani, A., Erik Chmelar and Mikhail Grinchuk.Segmented Addressable Scan Architecture.*In Proc. of VLSI Test Symposium*, pp: 405-411,2005.
[4]. Rad, M.A. and S.M. Eshgh. A heuristic Algorithm with a revocation based GA for test Pattern generation of VLSI circuits. *IEEE Proc. Of Int Conf. On Electrical Engineering*, pp: 1-5,2007.
[5]. Li, Y., Y. Dai and X. Ma. A heuristic immune-genetic algorithm for multimodal function optimization. *In Proceeding of CIMCA/IAWTIC*

conference, pp: 36-40, 2005.
[6].Chattopadhyay, S. and N. Choudhary. Genetic algorithm based approach for low power combinational circuit testing. *In proceeding of the IEEE 16th Int. Conference on VLSI Design*, pp: 552-557, 2003.
[7] Jiaxin, Y., C. Baichao, T. Cuihua and S. Yu. The research of inverter's control based on immune genetic algorithm. *In Proc. Of IEEE Conference on Industrial Electronics and Applications,*pp: 1-6, 2006.
[8].Breuer, A. and A.D. Friedman. Diagnosis and reliable design of digital systems*: Computer Science Press Inc.* , 1997
[9].Arslan, T. and M.J., O'Dare. A genetic algorithm for multiple fault model test generation. *In Proc. of Second International Conference on Genetic Algorithms in Engineering Systems. Innovations and Applications*, pp: 462-466 , 1997
[10].Al-Yamani, A., Erik Chmelar and Mikhail Grinchuk. Segment Addressable Scan Architecture. In Proc. of VLSI Test Symposium, pp: 405-411, 2005
[11].Essentials Of Electronic Testing For Digital, Memory And Mixed- Signal VLSI Circuits , by Michael L. Bushnell (*Rutgers University)* Vishwani D. Agrawal **(***Bell Labs, Lucent Technologies),* Kluwer Academic Publishers, 2002.
[12] .Introduction to genetic algorithm –tutorial, by Mark obitko, www.obotiko.com\tutorial\genetic-algorithm

[13].Goldberg,D.E,"genetic algorithms in search,optimization and Machine Learning.Reading,MA:Addison-Wesley.1989.

[14].Jiao ,L. and L. Wang,2000.,"A novel genetic algorithm based on immunity".PERLINKhttp://www.informatik.unitrier.de/ley/db/journals/tsmc/tsmca30.html\1"JiaoW00"IEEE Transactions on systems,Man and Cybernetics,Parta 30(5):552-561,2000.

[15].Schnurmann,H.D,E.Lindbloom and R.G.Carpenter,"The weighted random test –pattern generator.*IEEETrans.Comput., VOL. c-24,*pp:695-700,1975.

[16].Snethen,T.J.,"Simulator Oriented fault test generator" *Inproc.Of design Automation Conf.*,pp:88-93,1977.

[17].Seshu,S. and D.N.Freeman,"The diagnosis osasyncronus sequential switching systems,*IRE Trans on electronics Computing,vol.EC-11*,pp:459-465,1962