Evolutionary Algorithms for Low Power Test Pattern Generator

Bagavathi Chandrasekara
Assistant Professor
Department of Electronics and Communication Engineering
K.S.R. College of Engineering
Tiruchengode – 637 215

ABSTRACT

VLSI testing has been an essential part of chip design recently. A circuit must be tested before fabricating to avoid any malfunctioning. Testing a circuit has become mandatory that the circuit must be designed by ensuring testability. In VLSI testing, the circuit for testing is embedded with the actual design itself to reduce area and it is known to be Built-In Self Test (BIST). The test patterns generated by BIST are applied to the circuit. The test patterns are to be optimized to cover all the faults, reduce testing time and consume less power. This is achieved by employing Evolutionary Algorithms in selecting the patterns such that the inputs of design switch minimally. Test pattern generator is designed using these evolutionary algorithms so that the test vectors selected can be used for reducing the switching activity in the circuit and also by maintain the fault coverage. Genetic Algorithm and Particle Swarm Optimization are concentrated and their efficiencies are explained in this work

General Terms

VLSI Testing, Evolutionary Algorithms

Keywords

Evolutionary algorithms, Genetic Algorithm, Particle Swarm optimization, Low power, Test Pattern Generation.

1. INTRODUCTION

With the increasing use of portable computing and wireless communications, power dissipation has been a major concern in today’s VLSI systems. A circuit or system consumes more power in test mode than in normal mode. It has been reported that power consumption of VLSI chip during test application can be as high as 200% of that in normal mode [1]. This extra power consumption can give rise to severe threats for circuit reliability and can even inflame instant circuit damage. Moreover, it can fashion complications such as amplified product cost, striving in performance verification, reduced autonomy of portable systems, and decline of overall yield [2]. Low power dissipation during test application is becoming progressively more imperative in today’s VLSI systems design and is a major goal in the future development of VLSI design [1]. One handy method to reduce power dissipation during testing is to reduce the circuit transition between successive test inputs by choosing weighted random patterns [3] and transition density patterns [4].

BIST design is the most recurrently used technique for testing a chip for its well-known advantages. Since BIST is an in-built testing mechanism, the power consumption of BIST has to be reduced [5]. The BIST architecture consists of a linear feedback shift register (LFSR), a clock and circular shift register. The power consumed is mainly in the test patterns generated by LFSR as they are rarely correlated [6]. But in normal mode of a circuit operation, the patterns applied as input are highly allied. For testing procedure, highly uncorrelated patterns can be avoided so that the power consumed during testing can be reduced and the test pattern set can be compacted [7]. The parameters to be considered are fault coverage and weighted switching activity for low power BIST [8]. These indiscriminate test inputs are to be selected through evolutionary processes which have proved good in VLSI testing [9].

Evolutionary algorithms are well known for their robustness and self-adaptation [10]. These algorithms are extensively used in many applications such as VLSI testing, physical design and many more fields which require coordinated, controlled way of randomness inserted in the process of finding the solution. Genetic Algorithm and Particle Swarm Optimization are the two techniques that are to be focused in this paper for Test pattern generation with main emphasis on low power dissipation.

2. BUILT-IN SELF TEST

Built-in Self Test is a design procedure in which elements of a circuit are used to test the circuit itself [6]. It is the potential of a circuit to test itself. In circuit testing, the circuit can be tested using all possible combinations of input vectors or only the necessary vectors that can find the faults exactly. Sometimes a single vector can spot more than one faults. Whenever a set of test vectors are applied to a circuit sequentially, there will be a lot of switching in the inputs and outputs. To reduce the power during testing, minimum switching must be ensured. Hence, instead of using exhaustive testing, test vector size compaction can be used to reduce the memory requirements of BIST architecture. The inputs can be grouped in such a manner that there is reduction in space needs.

The main component of BIST is Linear Feedback Shift Register (LFSR) [11]. LFSR contains a sequence of registers or D-flip flop which is independent or guarded by clock. The circuit is cyclic in the sense that when clocked repeatedly, they result in a fixed sequence of states. Consider a LFSR with n flip-flops, it goes through 2n states. The last state can be just fed back to the first stage or it can be given to a modulo-2 adder, whose another input is state of any flip-flop other than last flip-flop. A cyclic shift register is used to shift and produce 2n patterns for n inputs.

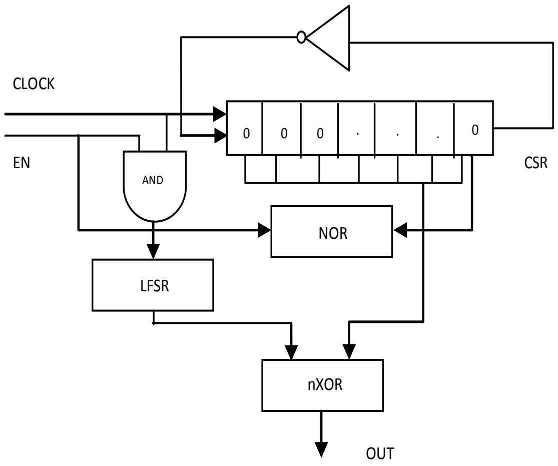

Fig 1: Low power test pattern generator (LPTPG) [6]

Consider the structure of test pattern generator shown above. It has a cyclic shift register (CSR), a linear feedback shift register (LFSR) synchronised with clock and some gates [6]. The cyclic shift register is mainly used to reduce the number of toggles between successive patterns. This results in low power architecture with less dynamic power consumption. Low Power Test Pattern Generator (LPTPG) [12] has been proved that it consumes less power compared to that of conventional pattern generator [13]. Further details of this circuit are discussed in design and implementation section.

3. LOW POWER TESTING

With increase in complexity of circuits, the testing process is becoming more complex and power consuming [14]. The chip's complexity is increasing while maintaining the size. There are so many test methods available for various different circuits. These available test techniques are evaluated using area overhead, fault coverage, test application time and test development time. Now power consumption should also be taken into account.

Power increases in test mode due to various reasons. First, test efficiency correlated with toggle rate. The test patterns are rarely correlated and hence, the switching power consumption increases. Second, DFT circuitry is used extensively while testing but usually it is idle during normal mode. In most of practical cases, inputs change very less but in test mode, inputs change is uncorrelated. Thus leading to increased power consumption.

Test power is a possible major engineering problem in the future of SoC development [15]. As both the SoC designs and the deep-submicron geometry become ubiquitous, larger designs, snuggy timing constraints, higher operating frequencies, and lower applied voltages all upset the power consumption systems of silicon devices [16].

3.1 Average power

Average power is the total allocation of power over a time period. The ratio of energy to test time gives the average power. Elevated average power increases the thermal load that must be vented away from the device under test to prevent structural damage to the silicon, bonding wires, or package. The average power of a device determines the overall capacity to handle high power for a span of time.

3.2 Instantaneous power

Instantaneous power is the power spent by a circuit at any given moment. Usually, it is defined as the power consumed right after the application of a synchronizing clock signal. Eminent instantaneous power might overload the power distribution systems of the silicon or package, causing brown-out. Instantaneous power has to be reduced to increase the life of the chip.

3.3 Peak power

The maximum power value at any given instant governs the component's electrical and thermal stability limits, system cooling and packaging requirements. If peak power exceeds a certain limit, designers can no longer guarantees that the entire circuit will function correctly. Peak power should be limited to a level, so that circuit will not burn-out on higher values.

3.4 Weighted Switching activity

Assuming a given CMOS technology and supply voltage for the circuit design, number of switching of a node i in the circuit is the only parameter that affects the energy, peak power, and average power consumption. The dynamic power consumption of the circuit is decided by number of toggles in the circuits, capacitance and supply voltage. When capacitance and supply are assumed to be constant, the number of toggles determines the power consumption. This is given by the measure weighted switching activity [5].

4. EVOLUTIONARY ALGORITHMS (EA)

Evolutionary computation has become an important problem solving methodology among many researchers. The population-based collective learning process, self-adaptation, and robustness are some of the key features of evolutionary algorithms when compared to other global optimization techniques such as heuristics. Evolutionary algorithm behavior is determined by the exploitation and exploration relationship kept throughout the run [10].

The evolutionary algorithm can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. As a result, evolutionary algorithms have recently received amplified interest, particularly with regard to the approach in which they may be applied for practical problem solving. They all share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction. The processes depend on the seeming performance of the individual structures as defined by the problem.

Compared to other global optimization techniques, evolutionary algorithms (EA) are easy to implement and very often they provide satisfactory solutions. A population of candidate solutions is initialized. New solutions are created by applying operators on the chosen parameters. The fitness or worthiness of the resulting solutions is evaluated and suitable selection strategy is then applied for the continuation of those solutions in the next iteration.

The main decisive factor of a solution is its quality as required by the fitness function. For several problems a simple Evolutionary algorithm might be good enough to find the desired solution. Some of the common Evolutionary algorithms are Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).

The design in this work deals with utilizing Evolutionary Algorithms to optimize the toggling of input vectors which is to be given to circuit under test. Slight modifications are done at the output of figure 1 to ensure the required results.

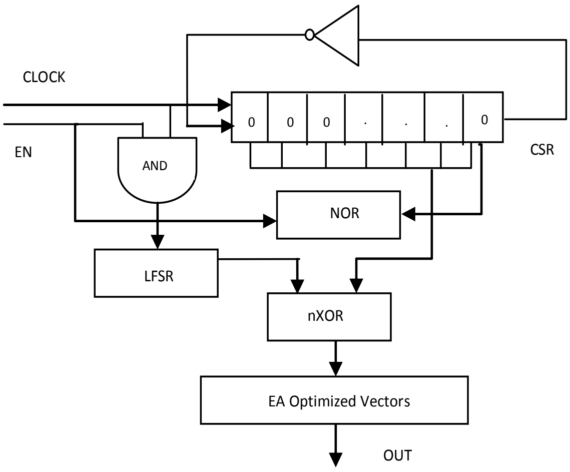

Fig 2: Modified Low Power Test Pattern Generator using Evolutionary Computation

In figure 2, the output of the n-bit XOR gate is optimized using Evolutionary algorithms to optimize the number of toggles between successive inputs. Genetic Algorithm and Particle Swarm Optimization are utilized for the goal of lower power consumption.

5. GENETIC ALGORITHM

Genetic Algorithm is a bio-mimic phenomena related to inheritance, crossover and mutation of biological evolution. This is a good searching technique used in optimisation problems [17]. A genetic algorithm (GA) is an exploration technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics [18]. Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover (also called recombination) [19]. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.

Genetic algorithms tend to thrive in an environment in which there is a very large set of candidate solutions and in which the search space is uneven and has many hills and valleys. Genetic algorithms will do well in any environment, but they will be greatly outclassed by more situation specific algorithms in the simpler search spaces. But genetic algorithms are not always the best choice. Sometimes they can take quite a while to run and are therefore not always feasible for real time use. They are, however, one of the most powerful methods with which to (relatively) quickly create high quality solutions to a problem.

This search method is used to reduce the number toggles between the successive patterns applied to circuit under test (CUT) to reduce the power consumed due to high frequency switching of input lines[20]. GA is used to group the inputs which affect the circuit in the same way as defined by a fitness function. The fitness function is designed to reduce the switching.

The most common type of genetic algorithm works with a population created with a group of individuals created randomly [21]. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. Two individuals are then selected based on their fitness, the higher the fitness, higher the chance of it being selected. These individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer. Commonly, the program terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

The majority of optimization methods move from a single point in the decision space to the next using some transition rule to determine the next point [22]. This point-to point method is dangerous as it can locate false peaks in multimodal (many-peaked) search spaces. GAs overcome this by working from a database of points simultaneously (a population of strings), climbing many peaks in parallel. The probability of finding a false peak is reduced compared to methods that go point to point. The concept of GA is easy to understand and implement. It supports multi objective optimization. They are able to solve problems knowing nothing about the problem from the start. They are inherently parallel and easily distributed.

6. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization has been a successor of Genetic Algorithm developed by James Kennedy and Russ Eberhart [23]. This optimization method takes its motivation from the physical co-ordinated movement of birds, fish and other animals with respect to the movements of their peers. It is modelled from the unpredictable group dynamics of bird social behaviour [24]. PSO combines self experience with social behaviour and hence it is more effective. The flock of birds move based on their inter-related individual distances. They tend to maintain an optimum distance between themselves and their neighbours. Compared to GA, PSO requires lesser steps to reach the destination. PSO optimizes the problem by employing a population based search method.

Each particle is associated with three vectors. The **x-vector** records the current position (location) of the particle in the search space, **p-vector** records the location of the best solution found so far by the particle, and **v-vector** contains a gradient (direction) for which particle will travel in if undisturbed. Particles can be seen as simple agents that fly through the search space and record (and possibly communicate) the best solution that they have discovered.
The new **x-vector** can be obtained by adding the current **x-vector** with **v-vector**. The new **v-vector** can be obtained by adding the current v-vector with cognitive and social behavior of the particle based on the current available nest solutions [24].

7. DESIGN AND IMPLEMENTATION

Low power test pattern generator (LPTPG) is used to reduce the power consumed during test operation. Consider the low power TPG shown in figure 1. For a n-bit cycle shifted register (CSR), only after performing one cycle (which is 2n clock cycles) and returning to all zero state can the LFSR generate the next pattern. The n-bit LFSR perform XOR operation with n-bit outputs of CSR one by one to produce the patterns of LFSR.

The LPTPG can be compared with the following circuit shown in figure 2, which uses genetic algorithm and particle swarm optimization to group the inputs that affect the circuit in the same way. The CSR contains only m registers instead of n. The m represents the number of groups of the inputs. In order to minimize power consumption, the elements in each group must be optimised. The optimisation is done through genetic algorithm and particle swarm optimization. The fitness function is fixed based on weighted switching activity measures.

$$WSA = \sum_{k}\sum_{i} S(i,k)F_i$$

Where s(i,k) is the number of toggles provoked by a test vector $TP_k = (V_k, V_{k+1})$ andF_i is the fan-out of the node.

$$WSA_{peak} = \frac{max_k WSA}{T}$$

$$WSA_{ave} = \frac{WSA}{L * T}$$

Where L is the test series length and T is the clock period. There are many varieties of genetic algorithm. In this work, two variations of GA are used. The type of genetic algorithm is called a recurrent genetic algorithm [25][26]. The fittest of the current population will be used to crossover with fittest of the next generation. This method results in faster convergence. Alternative crossover is used with two crossover points. Particle Swarm Optimization with random values of position and velocities are set. Each iteration velocities and positions are updated for all the points. The swarm is evaluated based on the switching between successive vectors. The two techniques are applied to four combinational circuits. The circuits are C499, c1908, c3540 and C6288 from ISCAS’85 benchmark. The population is set based on the number of inputs in the circuit.

8. RESULTS AND ANALYSIS

The coding was done in MATLAB and fault coverage was obtained from Atalanta 2.0. The results are analysed based on the fault coverage and weighted switching activity. Three techniques are applied to three different circuits. The weighted switching activity has been reduced in recurrent genetic algorithm compared to particle swarm optimization. The weighted switching activity is tabulated below.

Table 1.Tabulation of T (wsa-total), P (wsa-peak), A (wsa-average) for various test circuits and test methods

Total Switching Activity				
	C499	**C1908**	**C3540**	**C6288**
GA-LPTPG	81	65	99	63
PSO-LPTPG	150	118	209	77
Peak Switching Activity				
GA-LPTPG	2	2	2	2
PSO-LPTPG	14	13	14	9
Average Switching Activity				
GA-LPTPG	0.0122	0.0152	0.01	0.0156
PSO-LPTPG	1.83	1.79	2.09	1.21

The weighted switching activity has reduced drastically in Recurrent GA method compared to that of PSO method. The fault coverage is another important criterion for testing a circuit. The results show that the fault coverage improves in PSO-LPTPG for higher number of inputs. In Recurrent GA, the competent parts of chromosomes are lost when alternative crossover is used.

Table 2. Tabulation of Fault Coverage of Genetic Algorithm and Particle Swarm Optimization based Pattern Generator.

Method	**Fault Coverage**			
	C499	C1908	C3540	C6288
GA-LPTPG	68.2%	69.7%	66%	59.3%
PSO-LPTPG	65.8%	69.7%	68%	68.7%

Even the fittest chromosome from previous generation doesn’t yield good fault coverage. Thus some other methods should be developed to improve the fault coverage. The low power criterion has been well satisfied by the recurrent selection method.

9. CONCLUSION

The work of recurrent genetic algorithm and particle swarm optimization in reducing the power consumption of a test pattern generator has been presented in this report. Weighted switching activity has been reduced to a greater extent with fewer generations. In all the four circuits, the switching activity was reduced when GA method was applied. It has been observed that switching activity is greatly reduced in c432 circuit with 36 inputs. The fault coverage was not satisfying for recurrent genetic algorithm. Other methods of GA operators such as roulette selection can be employed to improve fault coverage.

10. ACKNOWLEDGMENTS

The author wishes to thank Almighty for knowledge. She is indebted to her parents N. Chandrasekara, R. Santhakumari, Brother C. Senthil Kumaran and husband K. V. Shivakumar for their moral support and strength.

11. REFERENCES

[1] Y Zorian, “A Distributed BIST Control Scheme for Complex VLSI Devices,” *Proceedings ofVLSI Test Symposium (VTS’93)*, pp. 4 – 9, 1993.

[2] F Corno, M Rebaudengo, M Reorda, G Squillero and M Violante, “Low Power BIST via Non-Linear Hybrid Cellular Automata,” *Proceedings of VLSI Test Symposium (VTS’00)*, pp. 29 – 34, 2000.

[3] G Marsaglia and L Tsay, “Matrices and the Structure of Random number Sequences”, *Proceedings of Linear algebra and its applications*, pp 147 – 156, 1985.

[4] H Rahaman, Farhana Rashid, Vishwani D Agrawal, “Power Problems in VLSI Circuit Testing”, *VDAT 2012*, LNCS 7373, pp 393 – 405, 2012.

[5] P Girard et.al, “Low Energy BIST Design: Impact of the LFSR TPG Parameters on the Weighted Switching Activity,” *Proceedings of International Symposium on Circuits and Systems*, 1999.

[6] Enmin Tan, Li Wang A, “Built-in Self-test Design with Low Power Consumption Based on Genetic Algorithm”,

Proceedings of Ninth International Conference on Electronic Measurement & Instruments, pp 526 – 529, 2009.

[7] Miron Abramovici, Melvin A Breuer, Arthur D Friedman, "Digital systems testing and testable design", *IEEE Press*, Revised edition, New York.

[8] R H He, X W Li and Y Z Gong, "A low power BIST TPG design", *Proceedings of the 5th international conference on ASIC*, Vol. 2, pp 1136-1139, 2003

[9] E M Tan, S D Song and W K Shi, "Power Reduction in BIST Design Based on Genetic Algorithm and Vector-Inserted TPG", *Proceedings of 2007 8th International Conference on Electronic Measurement & Instruments*, Vol. 4, pp 533 – 537, 2007.

[10] M H Marghny, Rasha M Abd El-Aziz and Ahmed I Taloba, "An Effective Evolutionary Clustering Algorithm: Hepatitis C: Case Study", *International Journal of Computer Applications*, Vol. 34, No. 6, November 2011.

[11] Vishwani D Agrawal and M Bushnell, "Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI circuits", *Kluwer Academic Publishers*, 2000.

[12] X D Zhang, W L Shan, and K Roy, "Low-power weighted random pattern testing", *IEEE Transactions on CAD of ICs and Systems*, Vol. 19, pp 1389-1398, 2000.

[13] N Ahmed, M H Tehranipour and M Nourani, "Low power pattern generation for BIST architecture", *Proceedings of the 2004 International Symposium on Circuits and Systems, (ISCAS '04)*, Vol. 2, pp 23 - 26, 2004.

[14] P Girard, "Survey of Low-Power testing of VLSI Circuit", *IEEE Design & Test of Computers*, Vol. 19, pp 82 – 92, 2002.

[15] Gary Yeap, "Practical low power digital VLSI design", *Springer*, 2009.

[16] S Gerstendiirfer and H J Wunderlich, "Minimized Power Consumption for Scan-based BIST", *Proceedings of IEEE International Test Conference*, September 1999.

[17] Pinaki Mazumder, and Elizabeth M Rudnick, "Genetic Algorithms for VLSI Design, Layout & Test automation", *Prentice Hall*, 1999.

[18] Melanie Mitchell, "An Introduction to Genetic Algorithms", *MIT press, Cambridge*, England.

[19] Franz Rothlauf, "Representations for Genetic and Evolutionary Algorithms", *Springer*, 2006.

[20] M J O'Dare and T Arslan, "Generating Test Patterns for VLSI circuits using a Genetic Algorithm", *Electronics Letters*, Vol. 30, No. 10, pp 778 – 779, 1994.

[21] Randy L Haupt, Sue Ellen Haupt, "Practical Genetic Algorithms", *Wiley Interscience publication*, Second edition.

[22] Dhiraj Sangwan, Seema Verma and Rajesh Kumar, "A Genetic Algorithm based Two Phase Fault Simulator for Sequential Circuit" *International Journal of Computer Applications*, Vol. 12, No. 10, November 2011.

[23] J Kennedy and R Eberhart "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

[24] Balwnder Singh, Sukhleen Bindra Narang and Arun Khosla, "Particle Swarm Optimization Framework for LowPower Testing of VLSI Circuits", *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 2, No. 3, July 2011.

[25] Adnan Fakeih and Ahmed Kattan, "Recurrent Genetic Algorithms: Sustaining Evolvability", *EvoCOP 2012*, LNCS 7245, pp. 230–242, 2012.

[26] Claudio Lopes De Souza, "Comparisons of intra-, inter population and modified recurrent selection methods Electronics Industry," *Review of Brazilian Genetics*, Vol. 16, No. 1, pp. 369-395, 1993.