Transitional Delay Fault Test Generation using a Genetic Algorithm

M. J. O'Dare and T. Arslan
Cardiff School of Engineering
University of Wales College Cardiff
P.O. Box 917
Cardiff
CF2 1XH,
United Kingdom
email: spetsa@cf.ac.uk

Abstract

With the continuous technological advancements in digital electronic circuit technology devices are becoming even more complex, it is essential that a high level of operational reliability is maintained which is why there is always a requirement for new and improved test methodologies. A fault condition in a digital circuit may be the result of a manufacturing problem causing physical imperfection in the device, this imperfection may be open circuit or short circuit connections, or a flaw that changes other characteristics of the device. The propagation of a gate may be affected introducing a delay defect into the circuit, often creating timing problems, which have always proved difficult to detect. This paper presents a new approach to the generation of test patterns that detect gross delay defects in combinational VLSI circuits. In O'Dare and Arslan [1] the authors presented a technique that implemented a genetic algorithm (GA) for the generation of test patterns to detect single stuck-at-faults in combinational VLSI circuits. In order to generate a test for delay faults it is necessary to create a transition in logical state at the fault site within the circuit, the only way to achieve this is by generating pairs of test patterns <P1,P2>. The first test pattern initialises the state of all nodes within the circuit, the second test pattern is then used to force the required transition at the designated test node. The problem of test generation for delay faults is considerably more complex than that of test pattern generation for single stuck-at-faults presented in O'Dare and Arslan [1], with an increase in search space size from 2^n to 2^{2n} for an n input circuit. The problem is further augmented by the necessity of applying the two test patterns in a strict ordered sequence to create the required transition. The authors have therefore developed a new genetic test technique to overcome the problem, using chromosomes-pairs to represent the test patterns. The GAs primary component is a dynamically evolving Global Record Table (GRT), which is used to guide the search towards optimal test pairs in an otherwise complex solution space, producing a compact and efficient set of test pattern-pairs as the GA evolves. The experimental results presented in this paper are compared with other research results for well known combinational benchmark circuits.

1 Introduction

Digital systems produced today are extremely intricate and are ever increasing in complexity, they are required for use in a widening range of domestic and industrial applications, Breuer [2]. In order to ensure reliability of these devices it is necessary to test their performance to identify any defects prior to using them in a fully operational environment.

In order to generate a test for a fault condition it is necessary to model the fault condition and simulate the circuit operation with that modelled fault condition present. This paper considers the generation of tests that detect gross delay faults in combinational digital circuits, Park *et al* [3]. The basic elements of such circuits are gates that produce a steady state output denoted by 0 or 1, which is purely dependant on the combination of its input values. The gates within a combinational digital circuit considered in this paper have the primitive functions NAND, AND, NOR, OR and NOT. The interconnection of the gates in a combinational circuit have no feedback loops, and therefore have outputs that are totally independent of their previous values. The most common fault model is the Single Stuck-at-fault model, Johnson [4]. which is simply used to represent an output or input node of a gate permanently held at the logic state 0 (Stuck-at-zero) or logic state 1 (Stuck-at-one). Test pattern generation for single-stuck-at faults was presented by the authors in O'Dare and Arslan [1], the stuck-at fault model however cannot be used to model delay defects, as the actual size of the defect needs to be considered, Park *et al* [3]. Recently there has been increased attention to the test and modelling of delay defects, Park *et al* [5] present a quantitative delay fault coverage model that takes into effect the delay defect size and the system clock interval, Devadas *et al* [6] present a method implementing synthesis procedures that produce networks which are fully delay testable under a non-robust fault model. In this paper the authors implement a gate delay fault model, which assumes that the output of a gate having a delay defect has a slow-to- rise or a slow-to-fall fault condition present.

2 The Gate Delay Fault Model

In the gate delay fault model the delay defect is considered to be a slow-to-rise or a slow-to-fall fault condition, occurring at a single node within the circuit due to a delay defect on the gate driving that node. A delay defect of this order may cause an error at the primary output of the circuit, figure 1 is used as a simple example of this fault condition. A series of connected elements in a combinational circuit are shown, at time t_0 the circuit is initialised by applying a signal (logic 0) to its primary inputs; in this example all nodes will reflect the logical state of the input. At time t_1 a second signal (logic 1) is applied to the primary input of the circuit, and at time t_2 the state of the primary output is observed. Consider t_0,t_1,t_2 . . . t_n to be the active edges of a system clock, and

t_n-t_{n-1} to be the clock interval τ. The clock interval has a duration greater than the sum of gate delays, i.e. if each of the gates A-D has a gate delay Δ, then $\tau > 4\Delta$ for the example circuit shown, therefore, with a circuit having g gates in it's longest path (from primary input to primary output), then $\tau > g\Delta$.

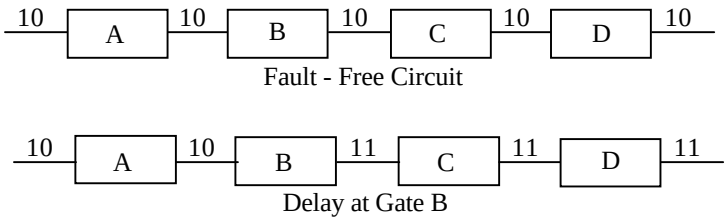

Figure 1: Delay model example.

In this example if a delay defect was present gate B with a slow to rise fault condition, it's effect will be observed at the primary output. This statement is only true if the delay defect size together with the sum of the delay sizes for each gate is greater than the system clock interval, however, if the delay defect size is greater than the system clock interval it will be detected if the fault condition is propagated a primary output, Park *et al* [3]. A delay fault with these characteristics is referred to as a gross delay defect, which is the condition considered in this work, another assumption of the model is that there is only one fault condition present at any one time, the model is therefore referred to as non-robust, Park *et al* [3]. In order to test for delay defects it is necessary to initialise the state of all nodes within the circuit, before a test pattern may be applied to the primary inputs of the circuit. A valid test for delay defects therefore comprises of a pair of test patterns <P1,P2>, applied in ordered sequence. P1 being the test pattern that initialises the circuit and P2 the test pattern that causes the required transition at the test node within the circuit. The problem is considerably more complex than that of stuck-at-fault testing, with an increase in the search space from 2^n to 2^{2n} for an n input circuit, this problem is made even more complex by the fact that the test patterns need to be applied in ordered sequence. GAs are suited to the NP - completeness of test pattern generation having already proved successful for single-stuck-at fault detection in both combinational and sequential circuits, O'Dare and Arslan [1], Rudnick *et al* [7], Prinetto *et al* [8], Hayashi *et al* [9], O'Dare and Arslan [10].

3. Genetic Implementation

Genetic algorithms (GAs) have grown in popularity following Holland's work in the field, Holland [11]. The technique adopts the mechanics of the biological genetic process, using binary strings as a substitute for *chromosomes*. In nature

the characteristics of the living organism are encoded into chromosomes. GAs have been successfully applied to many real world problems, Goldberg [12], Davis [13]. GAs have been applied successfully to other areas of VLSI design including channel routing, Buttita [14] and cell placement, Hegde [15]. Aylor *et al* [16] also employ GAs with a covering heuristic to reduce the size of test sets, that were produced by a random test pattern generation system.

3.1 Population Initialisation

The first stage of the GA is to initialise a population of chromosomes. In the single-stuck-at fault model the population of chromosomes were a direct representation of the test patterns applied to the test circuit, the length of each chromosome being equal to the number of inputs to the circuit. The initialisation of chromosomes for the delay fault model is similar, as the test patterns are directly represented as chromosomes, however, each test consist of a pair of test patterns, it was therefore necessary to generate chromosome - pairs to directly represent the potential delay test pattern - pairs.

The example circuit of Figure 2 is used to illustrate a test - pattern pair applied to a test circuit, pattern 100 is first applied in order to initialise the all circuit nodes, then the second vector is applied to cause any necessary transitions. The diagram also shows the direct chromosomal representation of the test pattern - pair applied.

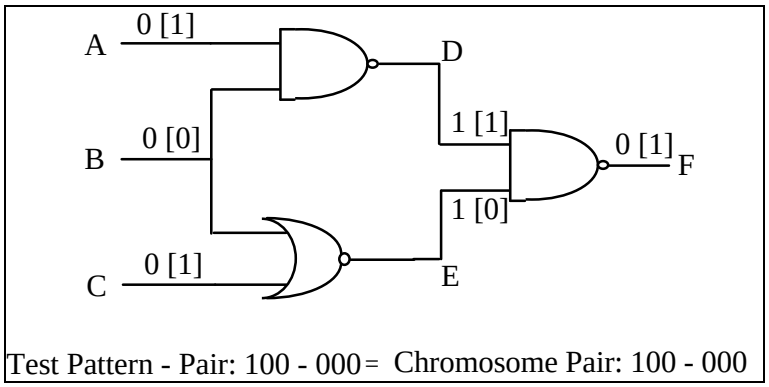

Figure 2: Test - pattern pair applied to test circuit.

3.2 Fitness Evaluation Function

To evaluate the efficiency of the chromosome - pairs as test patterns they are applied in an ordered sequence to the primary inputs of a compiled circuit model. Once the states of the circuit nodes have been set by the first chromosome the values are stored in the Global Record Table (GRT), and the second chromosome is then applied to the test circuit's primary inputs. A

comparison of the logical state for each node is made with the value stored in the GRT for the first chromosome, a change in state of 0 to 1 is recorded as a test for a slow-to-fall fault and a 1 to 0 is recorded as a test for a slow to rise fault condition. In order to validate these tests a full fault simulation is required to prove that the potential fault condition would be propagated to an observable primary output. If during the fault simulation the chromosome - pair detects a fault condition by causing the required transition and propagating it to an observable output, a fitness value is assigned to the chromosome - pair reflecting it's efficiency as a delay test, fitness is a biologically derived term used to measure the proficiency of individuals in the quest for producing offspring that have a greater ability to solve a given problem than their parents. The fittest chromosome - pair in the first generation is the first delay test entry into the test set, the test pattern - pair together with it's associated fault coverage is recorded in the GRT, the test pattern - pair may also be a valid test for certain single stuck-at-faults within the circuit, this information is also recorded and updated within the GRT. for subsequent delay test entries into the test set the GRT is updated with the new fault coverage for the test circuit, a test pattern - pair is only entered into the test set if it improves the overall coverage of the circuit, this procedure ensures that duplicates are not entered into the set and that the fault coverage for the circuit always improves with each entry.

The fitness evaluation function employed by the delay test system is a function of circuit size and complexity, involving a relationship between the number of primary inputs and gates within the circuit being tested. If a chromosome - pair detects delay fault condition that is previously undetected (*Pu*) a *score* is assigned to the fitness value for that pair:

$$score = 2(\sqrt{G} + I). \tag{1}$$

Where G represents the number of gates in the circuit an I represents the number of inputs to the circuit. The fitness evaluation function, F is represented by:

$$F = \sum_{j=1}^{k} Pu_j \ (score) + \sum_{j=1}^{k} Pd_j. \tag{2}$$

Where k is the number of possible delay faults present in the circuit, Pu and Pd (previously detected fault condition) are both random variables 0 or 1 based on the probability of distribution resulting form fault simulation of a chromosome - pair.

3.3 Parent Selection

Having determined fitness values for the initial population, it is necessary to select parents to engage in crossover. The technique adopted by the system is

the roulette wheel method of parent selection, this method ensures that the fitness values assigned to the test patterns, are proportional to their chances of being selected as parents, promoting the survival of the fittest concept, Goldberg [12], Davis [13].

3.4 Crossover and mutation

Two - point crossover is employed with each chromosome of the chromosome - pair having a 50% chance of selection, the two parents selected for crossover exchange information lying between two randomly generated points within the binary string. The resultant *offspring* inherit characteristics from their *parents,* this procedure is repeated until a whole new population of chromosomes have been created.

The next genetic operator to be applied to the population is mutation, employed to introduce diversity into the population. The delay test system uses a mutation function with a bit alteration probability of 1%.

4 Global Record Table

As well as holding the test pattern - pairs and their relate fault coverage the GRT also plays another important function, providing the delay fault simulation function with information necessary to the fitness allocation, by informing the simulation function of undetected delay faults existing in the circuit. The fitness scoring function uses this information to assign a fitness value to chromosome - pairs that are candidates for entry into the test set..

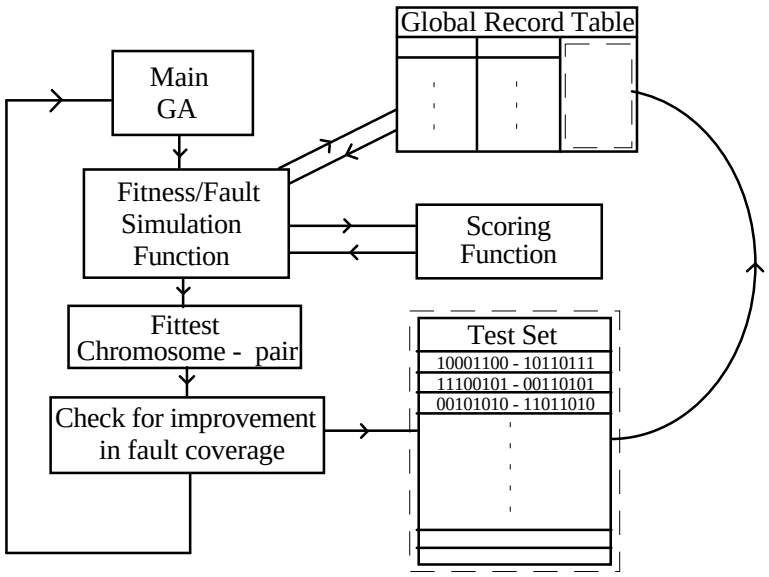

Figure 3: Interaction of Global Record Table with Main Functions.

Figure 3 illustrates the interaction of the main system functions with the GRT. The Fitness/fault simulation function consult the GRT before the scoring function assigns a fitness value, the fittest chromosome - pair is then entered as a valid delay test if the overall fault coverage of the circuit is improved, the GRT is then updated with the new fault coverage and the chromosome - pair is recorded in the test set.

5 Results and Discussion

The delay test system has been successfully applied to a number of combinational test circuits and to the ISCAS 1985 combinational benchmark circuits Brglez [17], where it produced favourable results when compared to result presented by other researchers, detecting not only delay faults but producing compact test sets that detected a high number of single stuck-at-fault conditions. The results presented in this paper however, do not concentrate solely on the performance of the delay system, but represent also represent the characteristics of the GA in it's application to the delay test problem.

Circuit	Test Set Size	Generations	Coverage
C432	28	35	99.23%
C880	30	31	100%
C7552	121	121	95.59%

Figure 4: Table to illustrate some of the results achieved for the ISCAS Benchmark Circuits.

The results shown in figure 4 are a sample of the some of the results achieved for a range of ISCAS benchmark circuits, the fault coverage is higher in most of the considered circuits when compared to results for single stuck-at fault coverage. presented by Ayari and Kaminska [18]. Furthermore, the genetic delay system by comparison often reduces the test set size even further than the test set compaction system presented in Ayari and Kaminska [18].

In figure 5 the graph illustrates the relationship between the population size used in the genetic delay system and the number of generations required to reach a delay fault coverage of 100%, for a population size less than 100 the generations required to reach 100% fault coverage often exceeded 35, but for a population size greater than 100 a fault coverage of 100% was generally achieved in less than 20 generations.

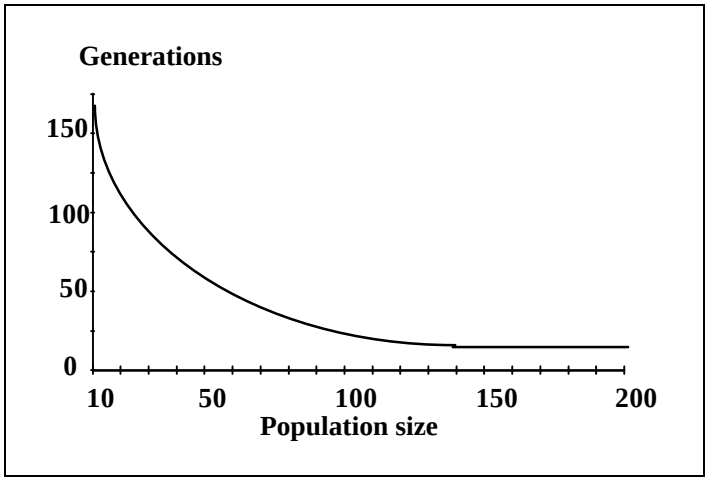

Figure 5: Graph to show population size against the number of generations of the GA.

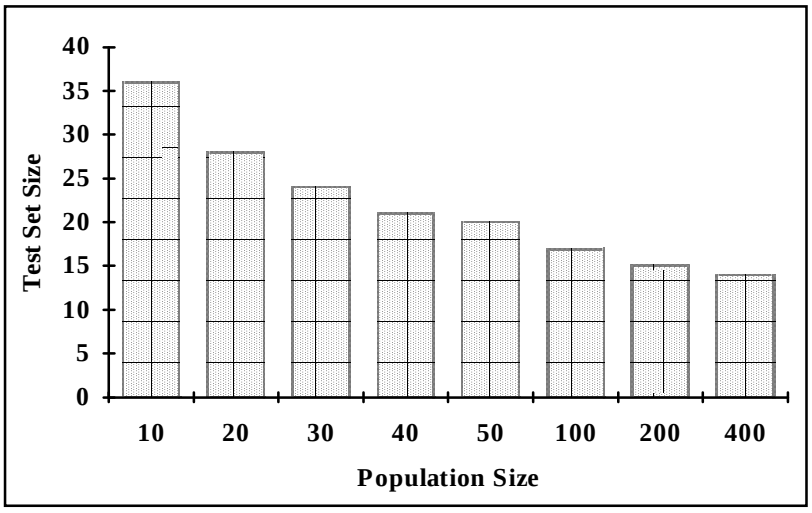

Figure 6: Graph to show population size against the number of test pattern - pairs in the test set.

The graph illustrated in figure 6 represents the relationship between the number of test pattern - pairs in the test set, required to detect 100% of the circuits delay faults and the size of the population used by the genetic system to generate that test set. The maximum number for the population size shown on the graph is 400, beyond this there was no notable improvement in the reduction of the test set size.

6 Conclusions

The Genetically based delay test pattern generation system presented has proved successful in this complex search problem, the results achieved by the system clearly indicate it's effectiveness as a fault detection tool for the detection of both gross delay defects and single stuck-at-faults, whilst producing a compact and efficient test set for combinational test circuits. The GRT was instrumental in guiding the GAs search for optimal solutions in an otherwise complex search space.

7 References

1 O'Dare M.J. and Arslan T. "Generating test patterns for VLSI circuits using a genetic algorithm," *IEE Electronics Letters*, Vol 30, No. 10, pp. 778-779, May 1994.

2 Breuer M.A and Friedman A.D. *Diagnosis and Reliable Design of Digital Systems*, Computer Science Press Inc, Digital System Design Series, 1977.

3 Park E.S. and Mercer M.R. An Efficient Delay Test Generation System for Combinational Logic Circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 926-938, Vol 11, No. 7, July 1992.

4 Johnson B.W. *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley Publishing Company, 1989.

5 Park E.S. and Mercer M.R. The Total Delay Fault Model and Statistical Delay Fault Coverage, pp 688 - 697, Vol 41, No. 6, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, June 1992.

6 Devadas S. and Keutzer K. Validatable Nonrobust Delay-Fault Testable Circuits Via Logic Synthesis, pp 1559 - 1573, Vol. 11, No. 12, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, December 1992.

7 Rudnick, J. G. Holm, D. G. Saab and J.H. Patel, Application of Simple Genetic Algorithms to Sequential Circuit Test Generation, pp. 40-45, *Proc. European Conference on Design Automation.* 1994.

8 Prinetto P., Rebaudengo M. and Reorda S, GATTO: An Intelligent Tool for Automatic Test Pattern Generation for Digital Circuits, pp. 411-417.

Proceedings of the 6th International Conference on Tools With Artificial Intelligence, November 6-9, 1994.

9 Hayashi T, Hidehiko K. and Hatayama K. A Genetic Approach To Test Generation For Logic Circuits, pp. 101-106, *Proceedings of The 3rd Asian Test Symposium, Nara Japan,* November 1994.

10 O'Dare M.J. and Arslan T, Hierarchical Test Pattern Generation Using A Genetic Algorithm With A Dynamic Global Reference Table, *First IEE/IEEE International Conference on - Genetic Algorithms in Engineering Systems: Innovations and Applications.* pp. 517-523, No. 414, 12-14, September 1995.

11 Holland H, *Adaptation in Natural and Artificial Systems,* Univ. of Michigan Press, Ann Arbor, 1975.

12 Goldberg D.E, *Genetic algorithms in search, optimisation and machine learning,* Addison Wesley, Reading, 1989.

13 Davis L, *Handbook of genetic algorithms,* Van Nostrand Reinhold, New York, 1991.

14 Buttitta B, Orlando P, Sorbello F, and G. Vassallo, Monreale: A new genetic algorithm for the solution of the channel routing problem, *Proc. IEEE,* CH3001-5, pp. 462-466, 1991.

15 Hegde U. and Ashmore B, A feasibility study of genetic placement, *Texas Instrum. Technol. J.*, vol. 9 , pp. 72-82, 1992.

16 Aylor J.H, Cohoon J.P, Feldhousen E.L, and Johnson B.W, GATE - A genetic algorithm for Compacting randomly generated test sets, *Int. J. Computer Aided VLSI Design,* vol 3, pp. 259-272, 1991.

17 M.J. O'Dare and T.Arslan, System Design for test using a Genetically based ATPG system. (accepted for publication), IEE Proceedings of The System Design for Testability Coll. April 1995.

18 Brglez, and Fujiwara H, A neutral netlist of combinational benchmark designs, *Proc. Int. Symp. Circuits and Systems*, June 1985.

19 Ayari B and Kaminska B, A New Dynamic Test Vector Compactioin for Automatic Test Pattern Generation, *IEEE Trans. Computer-Aided Design, vol.* 13, pp. 353-358, March 1994.