Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 53(67), Fascicola 1, 2008

Test Pattern Generation Multiple-Valued Logic Circuits

Vitaly Levashenko[1]

Abstract – In this paper we present a test pattern generation tool for combinational Multi-Valued Logic (MVL) Circuits. Test generation using deterministic algorithms is highly complex and time consuming. New approaches are needed to augment the existing techniques, both to reduce execution time and to improve fault coverage. Genetic Algorithms (GA's) have been effective in solving many research and optimization problems. Since test generation is a search process over a large vector space, it is a best candidate for GA's. The GA evolves candidate test vectors and sequences, using a fault simulation to compute the fitness of each candidate test.

Keywords: Multi-Valued Logic Circuits. Test generation. Genetic Algorithms

I. INTRODUCTION

One of the essential problems of hardware development now is physical limits of the signal diffusion speed, number and square of chip interconnections. One decision of this problem is increasing of information transferring affectivity in the chip. It is provided by MVL (*m*-valued) gates usage. Using of these gates allows to process the signals with *m* stable states. Today, such gates are used in serial production of chips by the firms *Intel* and *Motorola* [5, 12]. It is possible to reduce the transistor amount in 1,5-2,0 times, square of interconnection in 5,2-6,36 times and increases performance in 1,2-1,6 times. However, the reliability parameters of these chips are becoming worst.

Investigations establishing relations between physical defects of *m*-valued gates realized by different technologies and logical fault models of these gates are described in [1, 2]. An analysis, classification and description of possible single faults are represented in [15]. Such fault variety is a serious problem for *m*-valued circuits testing. There are some difficulties in adaptation known testing algorithms for circuits on *m*-valued gates. Therefore special testing algorithms, which are essentially differ from algorithms for binary circuits, are developed. The obtained results in this area are represented more in more detail in the Proceedings of *IEEE International Symposium on Multiple-Valued Logic* (1970-1998 *years*).

The D-algorithms for *m*-valued circuits testing were generalized in [10,11,13]. Method for detection of *stuck-at* and *window* β_1-β_2 faults is proposed in paper [2]. An approach of fault detection by investigation of circuit sensitivity to signal changing on input lines is shown in [4, 6, 9].

These algorithms belong to the class of deterministic algorithm. These algorithms allow to form a test set for all detected faults. Testing of circuits including large quantity of gates by these algorithms is highly complex and time consuming. However, using of *m*-valued circuits with small complicity justifies development of the deterministic algorithms for their testing. Now complicity of the *m*-valued circuit is increasing greatly. Therefore, investigations for development of classes of testing algorithms are necessary. One of such classes is a class of random algorithms.

The main idea of these algorithms is random generation of input signal set and searching of faults detected by this set. Decision about including of input set in the test set is obtained by analysis of the group of detected faults. Efficiency of such algorithms is provided by peculiarities of circuit faults. Anyway, large numbers of tests detects essential part of faults, therefore such faults are easy to detection by random algorithms. Moreover, procedures of faults modeling in random algorithms have smaller calculation complexity than procedures of faults propagation in deterministic algorithms. One of the random testing algorithms for *m*-valued circuits is offered in [3]. This algorithm allows generating tests for *stuck-at* faults detection.

We take a pattern generation based on GA principles. The main idea is to generate a random test pattern and to evaluate its fitness function as the number of faults detected by this pattern. Next this pattern is included or not in the resulted pattern set in accordance with a number of criteria. So, the best set are chosen so that to cover the largest number of faults which can occur in all lines of the circuit.

[1] Fakulta of Management Science and Informatics, University of Zilina, ul. Univerzitna 8215/1, Zilina 01026 Slovakia, e-mail: Vitaly.Levashenko@fri.uniza.sk

II. FAULT MODELS IN MVL CIRCUITS

In this section we give a brief review the known fault models. *Stuck-at* faults are the standard model for binary logic. They are useful tools to describe some faults in MVL systems, too.

Definition. A *stuck-at-k* fault occurs in line *x*, if *x* generates the output signal *k* for all input signals [3,6,11].

Definition. A β_1-β_2 *window* fault occurs in line *x* with if (i) *x* operates correctly for input signals $\beta_1 < t < \beta_2$, and (ii) *x* is *stuck-at*-β_1 for input signal $t < \beta_1$ and *stuck-at*-β_2 $t > \beta_2$, where $\beta_1, \beta_2 \in \{0,..., m-1\}$ [2].

Definition. A $r^{\pm}$*-order input signal variation* fault is occurred in line *x*, $r \in \{0,...,m\text{-}1\}$, if *x* generates the output signal $t' = t \pm r$ for input signal *t* . [15]

Definition. Let *Q* be a fault in a MVL circuit. Then any input vector **X** of the circuit which causes the output *t*(X) in the presence of *Q* and the different output *s*(X) in the fault-free case, is called *a test* of *Q* in *S*. A fault *Q* in *S* is called *detectable,* if a test of Q exists. Otherwise *Q* is called *undetectable*.

III. USING GA PRINCIPLES FOR TESTING ALGORITHMS DEVELOPMENT FOR MVL CIRCUITS

Using of GA principles allow to combine features of stochastic and deterministic methods of search and optimization. Application of these principles in the task of circuit testing causes keeping information received during previous tests searching and using this information at further input sets generating [8]. So, used information is formed on the base of the testing circuit peculiarities.

The main idea of GA is repeated selection the best chromosomes of the current population. Information about each population is not lost but it transforms and passes into next population during GA working.

An approach of using GA principles for MVL applications is offered in [14].

We perform some researches for effective using GA in the testing algorithms. Authors of researches show that tests detecting faults of circuit are allocated irregularity in test space. These tests are joined onto test groups. Power and allocation of mentioned groups are dependent on from used gate basis and structure of circuit. We propose to satisfy these demands by means of crossover and mutation GA procedures.

1. *Chromosome* is input pattern signal. *Gene* is one of *n* elements of chromosome (*n* is number of primary input of the circuit). All chromosomes of the initial population are formed randomly.
2. *Mutation.* The main destination of procedure is searching of new test groups. Chromosome is essential changed after mutation procedure execution. So, search in wide space of input signal sets is executed. of mutation procedure is new chromosome. The analysis of faults number detected by this chromosome shows expedience of further search in environment of analyzed chromosome. (Fig. 1a).
3. *Crossover.* The destination of procedure is searching of test detected the greatest number of fault of one test group. Chromosome is not essentially changed after crossover procedure execution (Fig.1b).

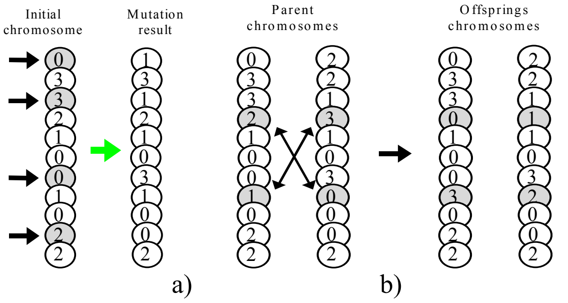

Fig. 1. Illustration of mutation (a) and crossover (b) GA procedures.

This realization of mutation and crossover procedures allows optimizing the developed algorithm work by execution time or by power of test setting. Really, let us define the main demand to algorithm as providing minimal power of test set. In this case algorithm has to select the better tests in the frame of each test group. It is achieved by predominantly using of crossover procedure.

In other case, if it is necessary to obtain test set during minimal time, algorithm has to quickly analyze the space of possible input sets m^n. During this analysis algorithm must detect the test groups and select tests from them. In this case the test analysis in the frame of one group is essentially reduced. It causes exceeding of power of test set, but performance characteristics of algorithm are improved. Note, that mutation procedure is used priority than crossover.

Mutation and crossover coefficient values control priority of using mutation and crossover procedures. User can choose necessary optimization (by time or by power of test set) by changing values of these coefficients [7].

IV. ALGORITHM DESCRIPTION

General structure of this algorithm is represented.

1. *Generation of initial GA population.* All chromosomes of initial population are formed randomly.
2. *Model process.* Weight is calculated for each chromosome of current population. This weight corresponds to number of new faults detected by this chromosome.
3. *Mutation and crossover procedures are execution and selection chromosomes to the test set.* There are following cases:
   - Chromosome is not included in test set if it has weigh less than defined value Ψ_1. Mutation operation is executed on such chromosomes. So,

chromosome will be essentially changed after this operation.

- Chromosome is included in the test set if it has weight greater than defined value Ψ_2. All faults detected by this chromosome is excluded from the circuit fault set.
- Crossover procedure is performed on chromosomes if they have weight between values Ψ_1 and Ψ_2. Probably such chromosomes are included in the one of the test groups but their characteristics will be improved by means crossover procedure in the frame of this test group.

  Changing the values Ψ_1 and Ψ_2 causes choice between optimization by working time or test set power. Increasing the distance promotes fulfillment of the first type optimization. Accordingly, decreasing of this distance means using of second type optimization. Note, that values Ψ_1 and Ψ_2 is depended on mutation parameter μ.

4. *Algorithm finish criterion.* Achievement of necessary percentage of detected faults, population number, time of algorithm executing should be chosen as algorithm finish criterion.

V. EXPERIMENTALS

Efficiency of proposed algorithm is confirmed by a number of experimental researches. In the frame of these researches new software system has been developed for MVL circuits modeling and different type of faults investigations.

We used the EDIF benchmarks interpreting them as 4-valued combinational circuits by transforming AND-gates into MIN-gates, AND-gates into MAX-gates and NOT-gates into 4-valued NOT-gates (for an input signal r the output signal is $r'=3-r$).

The aim of the experiment was to evaluate the proposed GA with respect to the following parameters: (i) the size of the resulted test pattern set (**Num**); (ii) run time of forming the resulted pattern set (**Time**); (iii) the percentage of detected faults (**Fault [%]**).

The experiments were performed on *Pentium*-166, 32Mb.

Three groups of parameters are given in Table 1 for each circuit. Each group includes size of the resulted patterns set and run time required to generate the set. The first group are the parameters if not less then 70% of possible faults were detected; the second - 80% and the third - 90%.

The choice of the mutation parameter μ =0,2 allows to generate the pattern set of the lager size for the smaller time. When μ =0,8 the size of the pattern set is smaller, but the run time is increased.

Table 1. Experimental of GA (*m*=4).

Circuits			Mutation parameter μ =0,2		
Name	**IN**	**Gate**	**Num**	**Time**	**Fault** [%]
c8_f16	16	50	25 30 36	0,21 0,25 0,29	73,8 81,2 92,3
rd53_f2	5	64	30 36 42	0,34 0,38 0,40	71,1 80,2 96,5
zx5p1_f1	7	58	25 31 39	0,28 0,34 0,37	70,0 81,2 94,6
zx5p1_f2	7	62	31 35 46	0,36 0,38 0,45	71,2 81,4 91,9
zx5p1_f3	7	94	42 48 54	0,41 0,46 0,48	70,0 80,0 91,2
bm5_17	17	258	25 31 39	0,68 0,84 0,97	76,7 83,3 95,4
			Mutation parameter μ =0,8		
			Num	**Time**	**Fault** [%]
c8_f16	16	50	22 27 32	0,30 0,32 0,34	70,1 82,7 90,0
rd53_f2	5	64	28 35 38	0,40 0,45 0,50	70,3 80,6 91,7
zx5p1_f1	7	58	22 27 33	0,38 0,42 0,46	70,8 81,3 94,6
zx5p1_f2	7	62	27 33 39	0,40 0,46 0,49	70,1 82,1 92,8
zx5p1_f3	7	94	34 38 42	0,58 0,63 0,66	71,4 81,3 92,0
bm5_17	17	258	22 27 33	0,78 1,02 1,16	76,1 81,7 96,1

VI. CONCLUSION

Proposed algorithm is intended for testing the circuits included more than 10^2 MVL gates. Method of MVL circuit testing by using offered algorithm includes following stages. Firstly, tests for 90-95% faults detected are quickly generated by proposed algorithm. Further, tests for rest 5-10% faults detected are searched by the means of deterministic algorithms (for instance, algorithms showed in [11]). Such faults are detected by the single tests.

Proposed approach for logic circuits testing has another application. For example, this approach is used in reliability analysis for Multi-State System [16, 17].

ACKNOWLEDGMENT

This work was partially supported by grant of Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Science (Vega) and grant MVTS for Belarus-Slovak cooperation.

REFERENCES

[1]. Chang Y.J, Lee C.L. Chen.J.E. Fault Models for the Multi-Valued Current Mode CMOS Circuit. *IEEE Proc. 26th Int. Symp. on Multiple-Valued Logic*, 1996.

[2]. Coy W., C.Moraga. Description and Detection of Faults in Multiple-Valued Logic Circuits. *IEEE Proc. 9th Int Symp. on Multiple-Valued Logic*, 1979, pp.74-81.

[3]. Drechsler R., Krieger R. and Becker B. Random Pattern Fault Simulation in Multi-Valued Circuits *IEEE Proc. 25th Int. Symp. on Multiple-Valued Logic*, 1995, pp. 98-103.

[4]. Dubrova E., Gurov D., Muzio J. Full Sensitivity and Test Generation for Multiple-Valued Logic Circuits. *IEEE Proc. 24th Int. Symp. on Multiple-Valued Logic*, 1994, pp.284-288.

[5]. Ishizuka O., Takarabe H., etc , Synthesis of Current-Mode Pass Transistor Circuits. *IEEE Proc. 21th Int. Symp. on Multiple-Valued Logic*, 1991, pp.139-146.

[6]. Guima T.A., Tapia M.A. Differential Calculus for Fault Detection in Multivalued Logic Circuits. *IEEE Proc. 17th Int. Symp. on Multiple-Valued Logic*, 1987, pp.99-108.

[7]. Kawahito S., Ishida M., Nakamura T., Kameyama M. High-Speed Area-Efficient Multiplier Design Using Multiple-Valued Current-Mode Circuits. *IEEE Trans. on Comp.*, 1994, vol.C-28, **1**, pp.34-41.

[8]. Rudnick E., Patel J.H., etc, A Genetic Algorithm Framework for Test Generation. *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 1997, vol.16, **9**, pp.1034-1043.

[9]. Levashenko V., Zaitseva E., Sanko A., Technique to Study Multiple-valued Logic in Courses on Modern Logic Design of Discrete Devices. *Proc of the 2-nd Int. Conference on New Information Technologies in Education*, vol.1, Minsk, Belarus, 1996, pp.329-338.

[10]. Shmerko V., Yanushkevich S., Levashenko V. Test Pattern Generation for Combinational Multi-Valued Circuits Based on Generalized D-Algorithm. *IEEE Proc. of the 27th Int. Symp. on Multiple-Valued Logic,* 1997, pp.139-144.

[11]. Spillman R.J., S.Y.H.Su. Detection of Single Stuck-Type Failures in Multivalued Combinational Circuits. *IEEE Trans.on Comp.*, 1977, vol.C-26, **12**, pp.1242-1251.

[12]. Stark M. Two Bits Per Cell ROM. *Proc. of IEEE COMPCON*, 1981, pp.209-216.

[13]. Tabakow I.G. Using D-Algebra to Generate Test for *m*-Logic Combinational Circuits. *Int.J.Electronics,* 1993*,* vol.75, **5**, pp.897-906.

[14]. Wesselkamper T.C., Danowitz J. Some News for Multiple-Valued Genetic Algorithms. *IEEE Proc. 25th Int. Symp. on Multiple-Valued Logic*, 1995, pp. 264-269.

[15]. Shmerko V., Yanushkevich S., Levashenko V., Zaitseva E., Test Generation for Multiple-Valued Logic Networks by Logic Derivatives. *Proc. of the Int. Workshop on Design Methodologies for Signal Processing*, Zakopane, Poland, 1996, pp.51-58.

[16]. Zaitseva E.N., Reliability Analysis of Multi-State System, *Dynamical Systems and Geometric Theories*, vol. 1**,** N. 2, November 2003, pp.213-222.

[17]. Zaitseva E., Puuronen S., Estimation of Multi-State system reliability depending on changes of some system component efficiencies, *Proc. of European Safety and Reliability Conference* (ESREL 2007), 25-27 June, Stavanger, Norway, 2007, pp.253-261