A GENETIC ALGORITHM FOR TEST PATTERN GENERATION MVL CIRCUITS[♣]

V.Levashenko[1]*, S.Yanushkevich*[2]

[1)] *Department of Information Technologies, State Economic University, Partizan Ave 26, 220672, Minsk, Belarus, e-mail: levashen@bseu.minsk.by*

[2)] *Institute of Computer Science & Information Systems, Technical University, Zolnierska str. 49, 71210, Szczecin, Poland, e-mail: yanushki@beta.ii.tuniv.szczecin.pl*

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.

1. INTRODUCTION

One of the essential problems of hardware development now is achievement of 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 firms *Intel* and *Motorola* [Stark81, Ishizuka91]. It is possible to reduce the transistor amount 1,5-2,0 times, square of interconnection 5,2-6,36 times and increases performance 1,2-1,6 times. However, the reliability parameters of these chips are worsen [Kawahito94].

Investigations establishing relations between physical defects of *m*-valued gates realized by different technologies and logical fault models of these gates are described in [Coy79, Chiang82, Chang96]. An analysis of possible single faults, their classification and description are completely represented in [Yanushkevich97]. Such fault variety is 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 different 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 [Spillman77, Tabakow93, Shmerko97]. Method for detection of *stuck-at* and *window* β_1-β_2 faults is proposed in paper [Coy79]. An approach of fault detection by investigation of circuit sensitivity to signal changing on input lines is shown in [Giuma87, Shmerko93, Dubrova94].

These algorithms belong to the class of deterministic algorithm. These algorithms allow to form a test set for 100% 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 they testing. Now complicity of the *m*-valued circuit is increasing greatly. Therefore, investigations for development classes of testing algorithms it are necessary. One of such classes is a class of random algorithms.

[♣] This work was supported by grant T98-M130 of Fund of Fundamental Researches (Belarus).

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 test set is obtained by analysis of group 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 observing and transporting in deterministic algorithms. Therefore, random testing methods are useful in practical applications for detection approximately 80% of faults. 20% rest faults are detected by deterministic methods.

One of the random testing algorithms for *m*-valued circuits is offered in [Drechsler95]. This algorithm allows generating tests for *stuck-at* faults detection. Merit of this algorithm is reducing the number of processing faults by 20%. It is achieved by means of faults collision researches.

We propose 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" candidates (test patterns) are chosen so that to cover the larger number of faults which can occur in all lines of the circuit.

2. 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 [Rudnick97]. 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 does not lose but it transforms and passes into next population during GA working.

An approach of using GA principles for MVL applications is offered in [Wesselkamper95].

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.

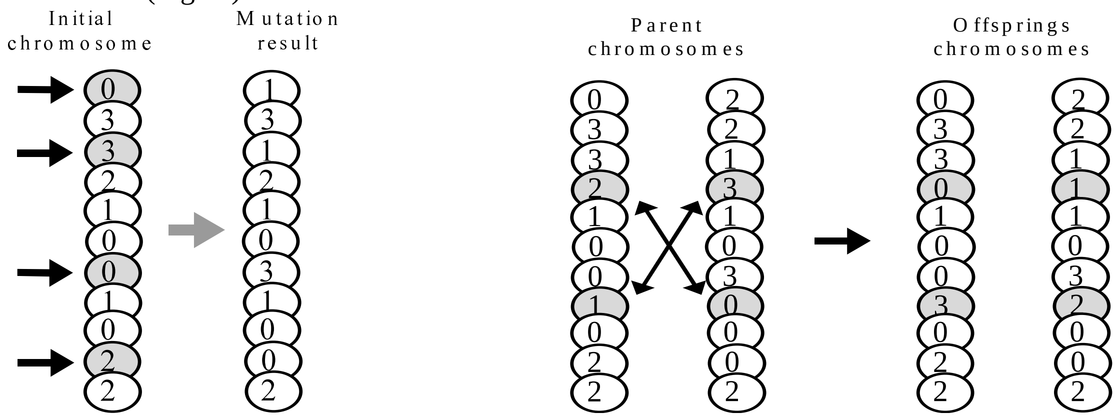
The analysis of experimental researches allows to make following demands to our algorithm:

- algorithm has to search maximum number of test groups for detection the greatest number of faults.
- algorithm has to select minimal number of the likely tests from each test group for minimization of test set power. So, number of faults detected by each selected test has to be the largest.

Authors propose to satisfy these demands by means of crossover and mutation GA procedures.

1. *Chromosome* is input signal set. *Gene* is input signal value on one input circuit line.
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 does not essentially change after crossover procedure execution (Fig.1b).



a) b)

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 the main demand to algorithm is 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 to 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 power of test set (pseudo optimal tests are selected), 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 [Levashenko98].

3. 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.* Weigh is calculated for each chromosome of current population. This weigh corresponds to number of new faults detected by this chromosome.
3. *Mutation and crossover procedures are execution and selection chromosomes to 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 essential changed after this operation.
- Chromosome is included in test set if it has weigh greater than defined value Ψ_2. All faults detected by this chromosome is excluded from circuit fault set.
- Crossover procedure is performed on chromosomes if they have weigh between values Ψ_1 and Ψ_2. Probably such chromosomes are included in one of test groups but they 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 first type optimization.

Accordingly, decreasing of this distance means using of second type optimization. Note, that values Ψ_1 and Ψ_2 is depended of 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.

4. 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. Algorithm presented in [Drechsler95] is selected as prototype for comparison analysis fulfillment. Therefore, note the following peculiarities of experimental researches:

- authors analysis only *stuck-at* faults in the frame of these researches.
- authors choose following parameters of algorithm work: power of test set (number of tests in this set), time of test set forming, percentage of detected faults.

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

Circuits			Mutation parameter μ =0,2 (minimum run time)						Mutation parameter μ =0,8 (minimum size of the test pattern set)					
Name	**IN**	**Gate**	**Num**	**Time** [s]	**Fault type**[%] *s-a-***0**	*s-a-***1**	*s-a-***2**	*s-a-***3**	**Num**	**Time** [s]	**Fault type**[%] *s-a-***0**	*s-a-***1**	*s-a-***2**	*s-a-***3**
c8_f16	16	50	25	0,21	76	78	70	68	22	0,30	72	71	70	68
			30	0,25	83	81	80	79	27	0,32	80	81	82	83
			36	0,29	97	90	93	87	32	0,34	95	87	91	88
rd53_f2	5	64	30	0,34	74	72	68	69	28	0,40	71	68	63	74
			36	0,38	80	79	84	76	35	0,45	78	75	79	83
			42	0,40	96	94	98	97	38	0,50	93	90	91	89
zx5p1_f1	7	58	25	0,28	69	68	70	71	22	0,38	68	64	69	64
			31	0,34	85	80	78	79	27	0,42	83	82	80	78
			39	0,37	98	96	90	91	33	0,46	96	93	92	94
zx5p1_f2	7	62	31	0,36	70	73	68	74	27	0,40	68	64	72	70
			35	0,38	80	76	82	84	33	0,46	81	84	82	79
			46	0,45	95	93	89	88	39	0,49	92	89	93	90
zx5p1_f3	7	94	42	0,41	65	69	73	68	34	0,58	68	69	73	71
			48	0,46	82	78	76	84	38	0,63	82	84	78	80
			54	0,48	92	89	90	91	42	0,66	94	91	89	90
bm5_17	17	258	25	0,68	79	78	80	71	22	0,78	78	74	79	74
			31	0,84	85	80	88	79	27	1,02	83	82	80	78
			39	0,97	98	93	93	92	33	1,16	99	93	96	97

- **Time**[s] is time of algorithm work in the seconds (*Pentium*-166, 32Mb).
- **Num** is test set power.

5. 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 means deterministic algorithms (for instance, algorithms showed in [Shmerko97]). Such faults are detected by single tests.

ACKNOWLEDGMENTS

The authors thank the professional support and help of Prof. Dr. Vlad. Shmerko, Fellow IEE. PhD student A.Shakirin and student V.Smotriaev have helped us for fulfilling all of experiments.

REFERENCES

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.

Chiang K.W., Vranesic Z.G. Fault Detection in Ternary *n*MOS and CMOS Circuits. *IEEE Proc.12th Int.Symp. on Multiple-Valued Logic*, 1982, pp.129-138.

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.

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.

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.

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.

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.

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.

Levashenko V., Shakirin A., Yanushkevich S., A Genetic Test Pattern Generation for MVL Circuits. *Proc. of the 7th Int.Workshop on Post-Binary ULSI Systems*. Fukuoka,Japan,1998, pp.55-56.

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.

Shmerko V., Yanushkevich S. Fault Detection in Multivalued Logic Circuits by New Type of Derivatives of Multivalued Functions. *Proc of the European Conf. on Circuit Theory and Design*, 1993, pp.643-646.

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.

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.

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

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

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.

Yanushkevich S., Levashenko V., Moraga C. Fault Models and their Classification for MVL Gates and Circuits. *Proc. of the 4th Int. Conf. Application on Computing System*, 1998, pp.309-314.