

## An Improved Genetic Algorithm for Delay-fault Test Generation

Wang Yong    Chen Guangju

(CAT Lab., UEST of China Chengdu 610054)

**Abstract** In this paper, the energy functions for hazard-free delay testing are presented, thus the problem of RRT (restricted robust test) generation for delay faults becomes an energy minimization problem. An improved genetic algorithm for RRT generation is given. This algorithm can inhibit premature convergence by modifying the population size with the degree of the evolution. The experiment shows that this method is effective.

**Key words** delay fault; test pattern generation; energy function; genetic algorithm

Correct operation of digital systems requires proper logical behavior as well as proper timing behavior. Logical defects are tested by applying tests for stuck-at-faults. Some defects do not affect the logical function, but affect the speed of a gate. These failures are called delay faults. The goal of delay fault testing is to make sure that a logical circuit meets timing specification and operates correctly at the desired clock rate.

Delay faults are described using the gate delay faults model or path delay faults model<sup>[1,2]</sup>. The former assumes isolated failures at distinct gates, while the later assumes distributed failures along a path. It is believed that path delay faults model is a better model for testing, since it considers cumulative propagation delays of gates along the path which directly determine the circuit performance. The usual method to detect delay faults is to apply an input vector  $V_1$  to the circuit at time  $t_0$ . After all signals in the circuit have been stabilized, a second input vector  $V_2$  is applied at time  $t_1$ , such a rising or falling transition is propagated from the input of the path, along the tested path, to the output of the path. By sampling the path output at time  $t_2$ , where  $(t_2 - t_1)$  is the pre-specified signal propagation delay through the circuit, one can ascertain the existence or non-existence of a delay fault. For a delay fault, ideal test pattern does not care about any arbitrary gate delay, which is known as RT (robust test). Another test is NRT (non-robust test), which can detect the delay fault in the target path if the arrival times of all off-path inputs are not late (i.e. not faulty).

In certain cases, static and dynamic hazard on off-path inputs can invalidate delay tests. To illustrate the hazard make test invalid, consider a falling transition delay fault on path  $b-d-f$  shown in Fig.1. When test is as follows:  $V_1: (a_1=1, b_1=1)$  and  $V_2: (a_2=1, b_2=0)$ , signal  $e$  maybe occur a static hazard, thus signal  $f$  would occur a dynamic hazard. In this case, test becomes invalid. So we must consider whether there exist race and hazard when applying test vectors, thus RT can be classified as general robust test (GRT) and hazard-free robust test (HFRT). The restricted robust test (RRT), which is the HFRT and guarantees a single-path propagation along the tested path, is important in the diagnosis of delay fault<sup>[3]</sup>. So, this paper only deals with the test generation of the RRT.

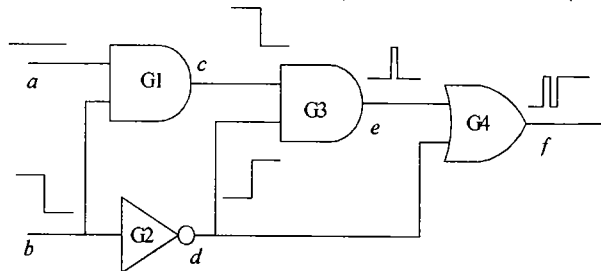


Fig1 The hazard make test invalid

Artificial neural network has

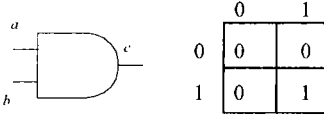


Fig2 The AND gate's truth table

learning ability, parallel processing ability and nonlinearity, thus using parallel processing ability of neural network to deal with the problem of TPG had caused extensive interests of many researchers. In 1988, Chakradhar et al. presented primitive gate's Hopfield neural network models<sup>[4]</sup>. Using these models, the problem of TPG becomes an energy minimization problem. This

method had acquired satisfying results<sup>[5]</sup>. Based on this principle, we construct energy functions for hazard-free delay testing. In the process of using these functions to generate RRT, we present an improved genetic algorithm which modifies the size of population to accelerate converge. Because this method can be parallelized, it is of positive significance.

## 1 Energy Functions for Hazard-free Delay Testing Generation

We illustrate the derivation of the energy function for an AND gate. Energy functions for the other primitive gates can be derived from the AND gate energy function.

For a digital circuit, when applying a pair of vectors ( $V_1, V_2$ ) at time  $t_0$  and time  $t_1$ , respectively, eight kinds of waveform probably exist in the circuit<sup>[6]</sup>. C. J. Lin et al. classified these eight kinds of waveform as four groups: S0, S1, U0, U1, XX (refer to Fig.3 of the Ref. [7]). We must avoid occurring hazard in the circuit to make sure the RRT is valid. For the AND gate, if a pair of input vectors is (01, 10) or (10, 01), the output of the AND gate probably occurs static hazard, when this static hazard continues to propagate in the circuit, the dynamic hazard will occur. So we must avoid using (01, 10) or (10, 01) as the input vectors for the AND gate when applying RRT.

In the procedure of testing, the first input vector  $V_1$  is applied at time  $t_0$ , after the signals in the circuit have been stabilized, the second input  $V_2$  is applied at time  $t_1$ . Therefore, for the AND gate in circuit under test, the inputs  $a_1, b_1$  and its output  $c_1$  meet the truth table as shown in Fig. 2 during the time from  $t_0$  to  $t_1$ . When the circuit does not occur hazard during the time from  $t_1$  to  $t_2$ , if there isn't any delay fault at the AND gate's inputs and its output, the stabilized signal values  $a_2, b_2$  and  $c_2$  will also meet this truth table. Otherwise,  $a_2, b_2, c_2$  will not meet this truth table. So, we can construct the AND gate's energy function for delay testing as follows

$$E'_{\text{AND}} = E_1 + E_2 + E_3 \quad (1)$$

where  $E_1$  is the energy function during the time from  $t_0$  to  $t_1$ , when the inputs and output meet the truth table as shown in Fig. 2 during the time,  $E_1=0$ . Otherwise,  $E_1>0$ .  $E_2$  is the restrain condition, when a pair of input vectors are not (01, 10) or (10, 01),  $E_2=0$ . Otherwise,  $E_2>0$ .  $E_3$  is the energy function during the time from  $t_1$  to  $t_2$  when there is no delay fault at the AND gate's inputs and its output, if the signal values  $a_2, b_2$  and  $c_2$  meet the truth table as shown in Fig. 2,  $E_3=0$ . Otherwise,  $E_3>0$ .

We construct  $E_1, E_2$  and  $E_3$  as follows

$$E_1 = a_1b_1 + c_1 - 2a_1b_1c_1 \quad (2)$$

$$E_2 = a_1\bar{b}_1\bar{a}_2b_2 + a_2\bar{b}_2\bar{a}_1b_1 \quad (3)$$

$$E_3 = a_2b_2 + c_2 - 2a_2b_2c_2 \quad (4)$$

We can verify that  $E_1, E_2, E_3$  satisfy the above discussion, so new AND gate's energy function can be written as follows

$$E'_{\text{AND}} = a_1b_1 + c_1 - 2a_1b_1c_1 + \bar{a}_1\bar{b}_1\bar{a}_2b_2 + a_2\bar{b}_2\bar{a}_1b_1 + a_2b_2 + c_2 - 2a_2b_2c_2 \quad (5)$$

From the new AND gate energy function, we can derive other primitive gates energy functions. As an

example of the NAND gate, when we invert the output of an AND gate, we obtain a NAND gate. Therefore, if we replace  $c_1, c_2$  in the AND gate energy function (6) by  $\bar{c}_1, \bar{c}_2$ , respectively, we can get the energy function for NAND gate

$$E'_{\text{NAND}} = a_1\bar{b}_1 + \bar{c}_1 - 2a_1\bar{b}_1\bar{c}_1 + a_1\bar{b}_1\bar{a}_2\bar{b}_2 + a_2\bar{b}_2\bar{a}_1\bar{b}_1 + a_2\bar{b}_2 + \bar{c}_2 - 2a_2\bar{b}_2\bar{c}_2 \tag{6}$$

Energy function for the other primitive gates can be derived similarly as follows

$$E'_{\text{OR}} = \bar{a}_1\bar{b}_1 + \bar{c}_1 - 2\bar{a}_1\bar{b}_1\bar{c}_1 + a_1\bar{b}_1\bar{a}_2\bar{b}_2 + a_2\bar{b}_2\bar{a}_1\bar{b}_1 + a_2\bar{b}_2 + \bar{c}_2 - 2a_2\bar{b}_2\bar{c}_2 \tag{7}$$

$$E'_{\text{NOR}} = \bar{a}_1\bar{b}_1 + \bar{c}_1 - 2\bar{a}_1\bar{b}_1\bar{c}_1 + a_1\bar{b}_1\bar{a}_2\bar{b}_2 + a_2\bar{b}_2\bar{a}_1\bar{b}_1 + a_2\bar{b}_2 + \bar{c}_2 - 2a_2\bar{b}_2\bar{c}_2 \tag{8}$$

$$E'_{\text{Buffer}} = a_1 + \bar{c}_1 - 2a_1\bar{c}_1 + a_2 + \bar{c}_2 - 2a_2\bar{c}_2 \tag{9}$$

$$E'_{\text{NOT}} = a_1 + \bar{c}_1 - 2a_1\bar{c}_1 + a_2 + \bar{c}_2 - 2a_2\bar{c}_2 \tag{10}$$

Based on the above energy functions, we can make the problem of RRT generation be an energy minimization problem. In the next section, we use genetic algorithm to minimize energy function.

## 2 Generation of Delay Testing Based on Genetic Algorithm

For restricted robust test (RRT), it must guarantee a single-path propagation along tested path, so we modify the sensitization condition as shown in the Fig.6 of the Ref. [7] as the single path sensitization condition as shown in Table 1. To combine the single path sensitization condition with the hazard-free delay testing energy, we get the algorithm of RRT for a path delay fault as follows:

**Table 1 Single path sensitization condition**

Gate	On-path Transition	
	Rising	Falling
AND or NAND	$x_1=x_2=1$	$x_1=x_2=1$
OR or NOR	$x_1=x_2=0$	$x_1=x_2=0$

- 1) Replace every gate in the circuit by its energy function.
- 2) Construct energy function for the circuit. It is the summation of individual gate energy function.
- 3) Derive off-path input values by using Table 1.
- 4) Simplify energy function by substituting off-path input values.
- 5) Minimize energy function. If the minimum is 0, we obtain a RRT. Otherwise, no RRT is possible for the path delay fault.

Genetic algorithms are evolutionary algorithms that mimic the way nature improves the characteristics of living beings. These algorithms are able to provide robust search in complex problem space and be parallel processed. Therefore, they have long been used as a possible solution for many search and optimization problems, specially for test generation using energy functions whose variables are only {1, 0}.

For the genetic algorithms, every solution (individual) is represented as a string (chromosome) of elements (genes) and is assigned a fitness value based on the value given by the fitness function. The fitness function measures how close the individual

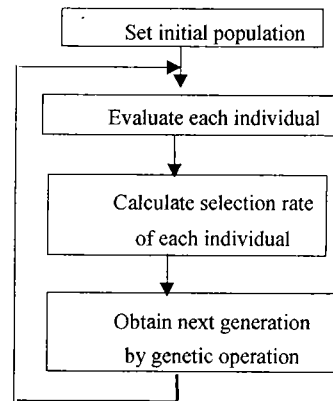


Fig3 The flow chart of genetic algorithms

is to the optimum solution. A set of the individuals constitutes a population that evolves from one generation to the next through the creation of new individuals and the deletion of some old ones. The flow

chart of genetic algorithms is shown in Fig. 3. In the procedure of genetic algorithms, evolutionary operations are most important, which include selection, crossover, and mutation. Selection is a process that selects superior chromosomes that will survive to the next generation, and also selects inferior chromosomes that will perish. Crossover generates offsprings of the new generation by exchanging genetic material between pairs of highly fitted chromosomes. After crossover, chromosomes are subjected to mutation. Mutation helps to prevent irrecoverable loss of potentially important genetic materials by increasing the variability of the population. Standard crossover and mutation operations are shown in Table 2.

**Table 2 Standard crossover and mutation operations**

operation	Old individual	new individual
crossover	$a_1 a_2 \cdots a_k a_{k+1} \cdots a_n$	$a_1 a_2 \cdots a_k b_{k+1} \cdots b_n$
	$b_1 b_2 \cdots b_k b_{k+1} \cdots b_n$	$b_1 b_2 \cdots b_k a_{k+1} \cdots a_n$
mutation	$a_1 a_2 \cdots a_p \cdots a_q \cdots a_n$	$a_1 a_2 \cdots b_p \cdots b_q \cdots a_n$

operations are most important, which include selection, crossover, and mutation. Selection is a process that selects superior chromosomes that will survive to the next generation, and also selects inferior chromosomes that will perish. Crossover generates offsprings of the new generation by exchanging genetic material between pairs of highly fitted chromosomes. After crossover, chromosomes are subjected to mutation. Mutation helps to prevent irrecoverable loss of potentially important genetic materials by increasing the variability of the population. Standard crossover and mutation operations are shown in Table 2.

Fundamental genetic algorithm only provides framework. For concrete problem, we must select appropriate GA parameters, so we consider the following respect for RRT generation.

The problem of RRT generation by using energy is a minimization problem, so we use the following map mechanism for fitness function and energy function

$$f = \frac{k}{E + k} \tag{11}$$

where  $k$  is a positive integer that is lower than 10, and should be determined by the scale of the circuit under test, the circuit's scale is greater,  $k$  is bigger.

The global searching ability for genetic algorithms is powerful, but local searching ability of the genetic algorithms is weak, thus it maybe occur premature convergence to a locally superior solution for GA searching process. In respect of ensuring diversity of population to inhibit premature, it seems that the population size should be as greater as possible, but the quantity of the computation would be increased. Therefore, we present an improved genetic algorithm which modifies the size of the population with the degree of searching. This improved genetic algorithm starts with  $N$  random initial chromosomes. Through  $M$  genetic operations, we could get  $M$  successive generations. To compare the maximum fitness in  $M$  successive generations with one of the initial population, if the increment is lower (such as less than 15%), it means that search falls into locally superior solution, thus we can break away from local searching through expanding search space at this time, i.e., inserting another  $P$  random chromosomes into successive generations, to continue genetic operations. Therefore, we get the algorithm as the following:

In the population each chromosome is represented by  $Z = \{x_1 \dots x_j \dots x_n\}$ , where  $x_j$  is only 0 or 1, the  $t$ th population is represented by  $P(t)$ .

1) Generate  $2N$  random chromosomes as initial population, it means that  $P(0) = \{a_j, j=1, \dots, 2N\}$ , set  $t=0$ ;

2) Calculate the fitness  $f_i$  of each chromosome, inserting the best chromosome into the next population  $P(t+1)$ . For each chromosome, we could calculate its selection rate as the following

$$P_{\text{select}}(i) = \frac{f_i}{\sum_j f_j} \tag{12}$$

3) According to the roulette wheel selection scheme, we select 2 chromosomes  $a_j$  and  $a_k$  from the  $P(t)$  and perform crossover operation for  $a_j$  and  $a_k$  with crossover rate, the crossover point is chosen randomly, thus we obtain chromosomes  $a'_j$  and  $a'_k$ ;

4) Perform mutation operation for  $a'_j$  and  $a'_k$  with mutation rate, and insert the new chromosomes into

the  $P(t+1)$ ;

5) If  $P(t+1)$  is not full, go to 3), otherwise  $t=t+1$ ;

6) When  $t=40$ , if  $\{\text{maximum fitness of the } P(40)\}/\{\text{maximum fitness of the } P(0)\} < 1.15$ , we insert another  $0.1N$  random chromosomes into the  $P(40)$ , then go to 2) until the minimum solution is found or after the given iteration limit is reached.

**Table 3 The results of the experiment**

Circuit name	Number of the node	number of the path under test	time of the TPG (1)	time of the TPG (2)
C17	11	5	0.35 s	0.27 s

Considering RRT for falling transition on 5 paths in the C17 benchmark circuit, we practice the above algorithm on a i486/66 computer. We chose  $k=2$  in Eq. (11), the size of the initial population is 40, crossover rate is 0.8, mutation rate is 0.01, and the given iteration limit is 300. The results of the experiment are shown in Tab.3.

In Tab.3, the time of TPG (1) is the result obtained by traditional genetic algorithm, the time of TPG (2) is the result obtained by improved genetic algorithm.

### 3 Conclusion

In this paper, we construct hazard-free delay testing energy functions and present an improved genetic algorithm to generate RRT. The experiment results shows the feasibility of this method.

Minimizing delay testing energy function by improved genetic algorithm, we can get RRT vectors. This method possesses good internal parallelism and conforms to the nowadays trend on algorithms research.

### References

- 1 Wagner K D. The error latency of delay faults in combinational and sequential circuits. Proc Int Test Conf, 1985: 334~341
- 2 Smith G L. Model for delay faults based upon paths. Proc Int Conf, 1985: 342~349
- 3 Savir J, Mcanney W H. Random pattern testability of delay faults. IEEE Trans Computer, 1985, 37(3): 291~300
- 4 Chakradhar S T. Automatic test generation using neural network. Proc ICCAD'88, 1988: 416~419
- 5 Chakradhar S T, Agrawal V D, Rothweiler S. A transitive closure algorithm for test generation. IEEE Trans CAD, 1993,12(7): 1 015~1 028
- 6 Glover C T, Merover H R. A method for delay fault test generation. Proc 25 th DAC, 1988: 90~95
- 7 Lin C, Reddy S M. On delay fault testing in logic circuits. IEEE Trans CAD, 1987: 694~703
- 8 Holland J. Adaptation in natural and artificial system. Ann Arbor: The University of Michigan Press, 1975

## 面向时滞测试生成的改进遗传算法\*

王 勇\*\* 陈光禡

(电子科技大学 CAT 室 成都 610054)

**【摘要】**在提出的无冒险的时滞测试能量函数的基础上,对传统的遗传算法进行了改进,即在搜索中根据进化程度对群体尺寸进行调整来加速收敛,用于时滞测试生成。实验证明该方法是一种较有发展前途的算法。

**关键词** 时滞故障; 测试生成; 能量函数; 遗传算法

**中图分类号** TN407

1998年12月30日收稿

\* 电子部预研基金资助项目

\*\* 男 32岁 博士生