Precise Test Generation for Resistive Bridging Fault of CMOS Combinational Circuit

Toshiyuki Maeda and Kozo Kinoshita
Dept. of Applied Physics, Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871 Japan
E-mail:{maeda, kozo}@ap.eng.osaka-u.ac.jp

1 Introduction

Logic testing based on the stuck-at fault model is still a main method for VLSI testing. However, according to the use of Nano technology the test generation for non-stuck-at faults has become a new important problem for test generation. I$_{\text{DDQ}}$ testing based on the current measurement of circuit is well known as an effective method for bridging faults. But, it is considered to be difficult to apply I$_{\text{DDQ}}$ testing for deep sub-micron CMOS circuits [1], as it becomes difficult to select a threshold value of current to distinguish good circuit from faulty one due to the increase of the background leakage current. Though I$_{\text{DDQ}}$ testing has the ability to detect most bridging faults, if the application of I$_{\text{DDQ}}$ testing becomes much more difficult, a test method to detect bridging faults using the logic testing must be considered.

Usually hard-short fault having zero resistance value is assumed as the model of the bridging fault. However, as an actual case, we must consider a resistive bridging fault having non-zero resistance value. A test generation method for resistive bridging fault has been proposed in [3], where they assume that the threshold voltage of the gate is given and apply a test generation method for detectable resistance range. However, since the detectable resistance range of bridging fault depends on the input vector of bridging gate as well as the bridging resistance [4], we have to take into account them to make complete detection of bridging faults. In this paper, we propose a novel method for testing resistive bridging faults by logic testing, considering dependency of threshold values, resistance values and gate input vectors. The proposed method can be applied to sub-micron CMOS circuits as a precise test generation method for resistive bridging faults.

2 Resistive Bridging Fault

In Figure 1, the resistive bridging fault(**RBF**) between signal lines X and Y is assumed. V_x and V_y denotes the voltage of signal line X and Y, respectively. R_f denotes the resistance of the bridging fault.

According to our previous experimental results, by logic testing the resistive bridging faults can be detected, which depends on the input vector as well as the threshold value of the following gate. Thus if we have no exact information on threshold value, we have to check for all possible inputs which generate the different value at the bridging point in order to detect all detectable **RBF** by logic testing. It is not enough only to set (X, Y) = $(0, 1)$ or $(1, 0)$ at bridging fault lines (X, Y) like the traditional test generation.

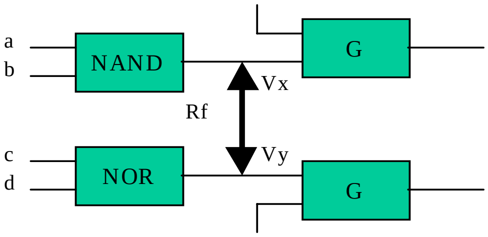

Figure 1: The resistive briding fault

Table 1: The detection table of NAND-NOR briding fault

	NAND	1	1	0
NOR		(0, 0)	(0, 1) / (1, 0)	(1, 1)
1	(0, 0)			X
0	(0, 1) / (1, 0)	X	X	
0	(1, 1)	X	X	

For example of the bridging fault between outputs of NAND and NOR gate, there are ten gate input vectors which has ability to detect the fault. But considering the symmetry feature of CMOS gate, input vector (0, 1) regarded as equivalent to input vector (1, 0). In the case of the brdging fault between NAND and NOR gates, five test input vectors must be considered as shown in table 1.

3 Test generation algorithm

We will propose test vector generation methods for resistive bridging faults. Here we assume that the resistance of **RBF** and threshold value of gate are unknown. By our previous experimental results, we use the following facts:

- Whether the **RBF** is excited or not depends on the resistance and logic threshold values of gates.
- The different gate input vector has the different ability to detect the fault.

The basic idea is to generate all possible input combinations at the input of bridging gates and is to propagate D($\overline{\text{D}}$)-cube to a primary output. The proposed test generation algorithm for **RBF** is as follows.

RBFTF-ALG

1. Assume Bridging fault B_{rf}(A, B) between signal lines A and B.

Table 2: Experimental Result

circuit	*#generated*	*#used*	*#all test*	*#tested*	test coverage	*#testable*	test efficiency	time(s)
s208.1	1112	243	12904	21530	59.93	13136	98.23	2.6
s298	1125	361	24586	37148	66.18	24603	99.93	6.8
s526	4094	1166	98344	140734	69.88			61.0
s838	8013	1268	186390	287822	64.76			257.2
s1488	5935	1590	948401	1646952	57.59	971902	99.86	439.8
s5378	16378	7848	3944873	5717546	69.00			25001.9

2. For bridging lines (A, B), gate input vector is selected as $(I_a, I_b) = (1, 0)$ or $(0, 1)$.
3. Error signal value D($\overline{\mathrm{D}}$) is inserted at signal line $I_a(I_b)$ whose value is 1(0), and we try to justify the gate input vector and propagate error value to primary outputs.
4. If justification succeeded, the justified primary input vector would be test vector for gate input vector (I_a, I_b).
5. Above procedures 2 - 4 are repeated for other gate input vector that can detect the briding fault.
6. Above procedures 2 - 5 are repeated for all considering briding faults.

In this algorithm, for justification of gate input vector and D propagation we can apply many heuristic methods that have been used by previous test generation methods for combinational circuits. In **RBFTG-ALG**, since the number of bridging faults increases according the increase of circuit size, the application of this algorithm will become time consuming as the size of circuit becomes larger. In order to generate test vector efficiently for **RBF**, we can use random test patterns, which are often used for IDDQ test.

4 Experimental Results

Table 2 summarizes the experimental result for ISCAS89 Benchmark circuits. We use the combinational part of sequential circuit. In this experiment, we use random vector. *#generated* and *#used* denote the number of generated random patterns and valid vectors for test, respectively. *#all test* denotes the denominator of test coverage,

$$\text{test coverage} = \frac{\#tested}{\#all\ test} = \frac{\#tested}{2\sum_{i=1}^{N} V(i)}. \qquad (1)$$

whose *#tested* denotes the number of test vectors which succeed in the propagation of faulty value.

Table 2 shows that it looks difficult to generate precise test vectors even for small circuits. We need take into consideration that so many test vectors are actually indispensable. In table 2, test coverage is not high. In order to clarify the reason, we tried to calculate untestable cases by means that all input vectors are applied to some small circuits. In table 2, *#testable* denotes the number of untestable cases,

$$\#testable = \#all\ test - \#untested \qquad (2)$$

Here, we define test efficiency

$$\text{test efficiency} = \frac{\#tested}{\#testable} \times 100(\%) \qquad (3)$$

Table 2 also shows that test efficiency is more than 99% by test generation using random patterns for most circuits. Untestable faults exist in the case when gate input vectors to excite the fault can not be justified or the case when the propagation of faulty value to primary output fails.

According to our experimental results, we expect test generation using random pattern is effective for lager circuits. But for larger circuits we have to consider more effective method in the practical application.

1. If threshold voltage of gate is known, gate input vectors can be restricted and the number of setting D can be fewer. As a result, *#all test* becomes smaller.
2. If type of gate is restricted, gate input vectors can be restricted. Therefore *#all test* becomes smaller.

5 Conclusion

In this paper we have presented a precise test generation method to detect **RBF** by logic testing. Some **RBF**s are missed to be detected by traditional test generation methods, and proposed the new algorithm to detect **RBF**. Because there are many bridging faults in large circuit, we proposed a heuristic method using random patterns, and presented some results. In this study, since we assume that threshold voltage of a gate are unknown. but if threshold voltage is known, we can apply this method more effectively.

Acknowledgement

The authors would like to acknowledge to Mr. Arabi Keshk for calculating bridging voltage by Spice simulation used in the examples of this paper.

References

[1] T. Williams, et al, "Iddq Test: Sensitivity Analysys of Scaling," *Proceeding of ITC*, pp. 786-792, 1996.

[2] Current Ratios, "A Self Techniques for Production IDDQ Testing," *Procceding of ITC*. pp.738-746, 1999.

[3] V. R. Sar-Dessai and D. M. H. Walker, "Resistive Bridging Fault Modeling, Simulation and Test Generation," *Proceeding of ITC*, pp. 596-605, 1999.

[4] A. Keshk, Y. Miura and K. Kinoshita, "IDDQ Current Depenency on Test Vectors and Bridging Faults," *Procceding of ATS'99*, pp. 158-163, 1999.