Proceedings of the International Conference on Manufacturing Systems – ICMaS
Vol. 4, 2009, ISSN 1842-3183
University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department
Bucharest, Romania

MULTIPLE STUCK-AT FAULTS AND VLSI DIAGNOSTIC TEST VECTOR GENERATION

Pavlinka RADOYSKA, Kamen FILLYOV

Abstract: *One of the most common heavy problems in fault detection and fault diagnosis is the fault masking. Multiple stuck-at fault analysis is an approach for solving the problem, but the full multiple fault combination analysis is a time consuming procedure. Selecting only the dependent faults reduces the time for analysis and test pattern generation. We suggest a method for test pattern generation with multiple fault simulations to overcome masking fault problem. The method consists of two algorithms: (1) finding the pairs for multiple fault simulation; (2) generating the test patterns, which can avoid potential masking effect.*

Key words: *digital circuits, multiple fault simulation, fault masking, diagnosis, test pattern generation.*

1. INTRODUCTION

The main purpose of manufacturing test pattern generation is to gain maximum fault coverage for a minimum number of tests. In this aspect a single test failing could be caused by a large number of faults. The main purpose of VLSI diagnostics is just the opposite – minimum number of faults for a single test failing. This process is time and hardware consuming. It could take days to find the single vector for the single fault and it could take years to test and analyze the result for all manufacturing chips. Moreover the tester storage space is limited and an infinitely number of failing patterns cannot be stored there. Typically, fewer than 100 failing patterns can be stored in the tester memory.

The single stuck-at faults and the ATPG are well researched and elaborated territory with significant test detection effect. However, the single-fault model may not be adequate for diagnosing defects in modern devices with million gates. Multiple stuck-at faults model is more realistic but more complicated for diagnostics.

Fault effect masking is a significant fault detection and fault diagnostics problem. One possible decision to overcome this effect during the tests is multiple fault simulation on test generation time. The experimental results reported in [1] indicate that activating multiple faults will increase the probability of fault convergence and fault masking. A similar experiment, reported in [2], for correcting design errors also confirms that the presence of error effect interaction grows with the number of activated errors.

The aim of the paper is to propose one method for test pattern generation with reasonable length and maximum diagnosis effect. The algorithm is z-set based and uses multiple fault simulations to improve detection and diagnostic test quality. This paper is organized as follows. In Section 2, are outlined the main fault diagnosis approaches, give some definitions and describe z-set ideology. In Section 3, are discussed the masking effect as the significant multiple faults problem, calculate the average number of multiple fault simulations and considerations for faults selection. In Section 4, are suggested the algorithm for test pattern generation. In Section 5, are given experimental results for multiple faults collapsing getting from our masking simulator. In Section 6, are made conclusions.

2. PRELIMINARIES

2.1. Contemporary fault diagnostics approaches

The existing diagnostics algorithms can be divided into two main classes: cause–effect and effect–cause. The Cause-Effect Diagnostics approach is based on some kind of fault dictionary. The faulty responses of modeled faults are pre-computed and stored in a dictionary. The observed failure responses are compared with those in the dictionary and the faults whose pre-computed failure responses have the closest match with the observed failure responses are chosen as fault candidates. This diagnosis process is fast but requires a large memory for the dictionary [3].

The Effect-Cause Diagnostics approach is based on simulations and uses a standard ATPG produced test. By back tracing from failing outputs the potential fault candidates are obtained and only these faults are simulated. Some faults are dropped from the simulation while the simulated output states get the different from those obtained under the test. This approach can achieve very high diagnosis accuracy without high memory demands but is quite slow.

2.2. Definitions

Diagnosis consists of locating the faults in a structural model of the circuit under detection (CUD).

Indistinguishable faults. In three-valued logic, two faults f_1 and f_2 in one circuit are indistinguishable if for any input pattern, either the binary values on each primary output for f_1 and f_2 are equal or one of them is unknown (X); otherwise, they are **distinguishable**.

Structural equivalent faults. Two faults $f_i \in F$ and $f_j \in F$ are structural equivalent if for any possible input pattern $pi_k \in P_{input}$ the corresponding output patterns $po_{ik} \in P_{output}$ and $po_{jk} \in P_{output}$ are equivalent ($po_{ik} \equiv po_{jk}$).

Functional equivalent faults. Two faults $f_i \in F$ and $f_j \in F$ are functional equivalent if for any test pattern $t_k \in T$ the corresponding output patterns $po_{ik} \in P_{output}$ and $po_{jk} \in P_{output}$ are equivalent ($po_{ik} \equiv po_{jk}$). Special attention should be paid to $T \subset P_{input}$.

We can divide *masking effect* definition in two aspects: fault detection and fault diagnosis. In the fault detection context fault $f_i \in F$ is masked by fault $f_j \in F$ if in the test patterns set T there is $t_i \in T$ which can detect fault f_i as a single fault but there is not test pattern $t_i' \in T$ which can detect fault f_i while fault f_j is present. The fault f_i is *masked fault* and f_j is *masking fault*.

In the fault diagnostics context fault $f_i \in F$ is masked by fault $f_j \in F$ if in the set T there is not test pattern $t_k \in T$ which can allow distinguish fault f_i from fault f_j.

2.3. Z-set theory [4, 5, 6]

Z_{all} is a collection of all primary outputs of the CUD - $Z_{all}=\{z_0, z_1, \dots, z_n\}$.

Z-set is a collection of primary outputs on which one fault could be detected. $Z(f)$ for the fault f on the line g is the set of outputs such that there is a directed path from line g to each of them. The z-sets are independent of the test set used and the type of the fault. Therefore we can write $Z(f)=Z(g)$.

For illustration, let consider the circuit of Fig. 1. The Z-sets for faults f_1, f_2 and f_3 shown in Fig. 1 are $Z(f_1) = \{z_1\}$; $Z(f_2) = \{z_1, z_2\}$; $Z(f_3) = \{z_2\}$.

Z-set is a parameter which can be used to distinguish faults. Considering the example on fig.1 the two faults f_1 and f_2 are distinguished since $Z(f_1) \cap Z(f_3) = \varnothing$.

Z-set for a line i can be described as a vector $Z_i = Z_i(0)\ Z_i(1)\ Z_i(2) \dots\ Z_i(n\text{-}1)$, where n is the number of outputs, $Z_i(j)=1$ if $z_i \in Z_i$ and $Z_i(j) = 0$ if $z_i \notin Z_i$. Consider the example $Z(f_1) = \{10\}$, $Z(f_2) = \{11\}$ and $Z(f_3) = \{01\}$.

Let us have two faults $f_i \in F$ and $f_j \in F$ and two test patterns $t_i \in T$ and $t_j \in T$, which can detect respectively f_i and f_j as a single stuck-at faults. Z-sets for the two faults are $Z(f_i)$ and $Z(f_j)$. The relationships between the two z-sets can be put in four categories:

1) $Z(f_i) \cap Z(f_j) = \varnothing$ (Fig. 2,a). This means that f_i and f_j are totally independent and if they are present in the circuit simultaneously (as a multiple stuck-at faults) they safely could be distinguished according to the failing outputs. For every t_i, $t_j \in T$, f_i and f_j are distinguishable.

2) $Z(f_i) \equiv Z(f_j)$ (Fig. 2,b). This means that the two faults can be observed on the same outputs. In this situation must be find such kind of tests t_i and t_j, so that t_i detects f_i, but it does not detect f_j and t_j detects f_j, but it does not detect f_i. f_i and f_j are distinguishable only if $\exists\ t_i, t_j (t_i \neq t_j)$.

3) $Z(f_i) \subset Z(f_j)$ (Fig. 2,c). This means that f_j and f_i could be distinguishable for $t_i = t_j$ if any output in $Zset = Z(f_j) - Z(f_i)$ is failed and any output in $Zset = Z(f_i)$ is pass. But if any output in $Zset = Z(f_i)$ failed, the two faults become undistinguishable. They could be distinguishable only if $t_i \neq t_j$.

4) $Z(f_i) \cap Z(f_j) \neq \varnothing$ but neither $Z(f_i) \not\subset Z(f_j)$ nor $Z(f_j) \not\subset Z(f_i)$ (Fig. 2,d). This means that for $t_i = t_j$ f_i can distinguished from f_j if any output on $Zset = Z(f_i) - Z(f_j)$ failed and f_j can distinguished from f_i if any output on $Zset = Z(f_j) - Z(f_i)$ failed.

In conclusion we can say that for $t_i = t_j$ the two faults f_i from f_j could be distinguishable on the outputs $z_k \in Zset = Z(f_i) - Z(f_j)$. If $Zset$ is empty we must find test vectors $t_i \neq t_j$.

3. MASKING EFFECT AND MULTIPLE FAULT SIMULATIONS

3.1. Multiple fault problems

Multiple fault problems are different in fault detection and fault diagnosis, but the masking problem is a significant problem in both processes. We suggest that multiple fault simulation and dynamic testing could help to overcome the masking effect and distinguish masked faults. Faults with z-set cardinality equivalent to one are easier to diagnose with a single stuck at fault presumption [4], but in real situation with multiple faults they could not be distinguished and some times if they are masked they become undetectable.

Let us look at the circuit on Fig. 3. There is an AND gate with two inputs (a and b) and an output z. Let there are two faults: stuck-at-1 on line a ($a/sa1$) and stuck-at-0 on line z ($z/sa0$). For the input vector ($a = 1$, $b = 1$) on the line z a level 0 is observed instead of 1. So, in spite of the a / $sa1$ the fault z / $sa0$ is detected. Input pattern ($a = 0$, $b = 1$) is the only vector, which could detect single *a/sa1* fault.

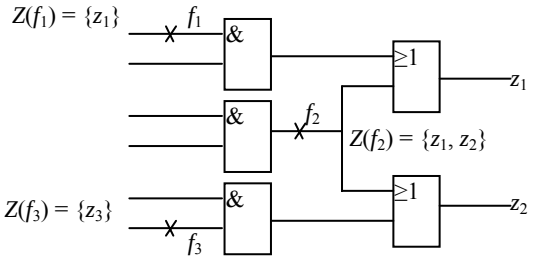

Fig. 1. Using z-sets to distinguish faults.

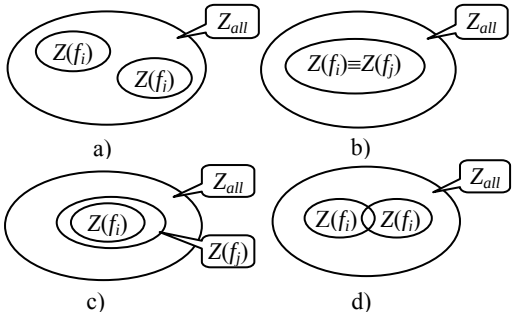

Fig. 2. Relationships between the two z-sets.

But for Fig.3 it is observed level 0 on the line *z*, the same as for the fault free circuit. For the masked faults, for which there is only one path to the some primary output true the masking fault, there is no chance to diagnosis or detect the fault. There is a possibility to detect the fault if there is another path like the one described in the Fig.4. For the input pattern ($a = 1$, $b = 0$, $c = 0$, $d = 1$), if faults *e* / *sa*1 and z_2 / *sa*0 persist in the circuit, output pattern will be ($z_1 = 1$, $z_2 = 0$), while for the fault free circuit output pattern will be ($z_1 = 0$, z_2=0) and for the circuit with single stuck-at fault *e* / *sa*0 – ($z_1 = 0$, $z_2 = 0$). Therefore we will be able to find input patterns for detecting the two faults by implementing multiple fault simulation for this pair. This solution involves two significant problems:

1. There could be a lot of masking faults and the number of combinations could become very high.
2. It is possible to brake down some other faults diagnosis (lose distinguishability).

Let us speculate about the number of multiple fault combinations. The total number of multiple fault combinations can be calculated according to the equation $\sum_{k=2}^{n} C_n^k = 2^n - n - 1$, where n is a number of faults. This number is extremely high. We can reduce it according to the *z*-set theory. The only faults which can interact are the faults with the same z-set. If the *z*-set cardinality for each fault is equal to one and there is an even distribution, then the numbers of faults in one group is about $m = n / p$, where n is an all faults number and p is the number of outputs. Then the number of multiple fault combinations can be calculated according to the equation $\sum_{k=2}^{n} C_m^k = 2^m - m - 1 = 2^{n/p} - n/p - 1$, which is lower than the total number of combinations but is still a high number. Let us look now at Fig. 5. If there are several faults on the one path from the fault line to any primary output, which can cause masking effect, they all are operating as one masking fault. We can say that faults *b* / *sa*1, *c* / *sa*1, *d* / *sa*1 and *z* / *sa*1 are equivalent masking faults and all of them can be represented with one masking fault (for example the last one – *z* / *sa*1). As a result we can simulate only one combination between the fault *a* / *sa*1 and *z* / *sa*1. Then the number of multiple simulations for one fault is the number of paths to the all failed outputs of its *z*-set. For all the faults we can get the average level by calculating the number of multiple fault combinations:

$$\#multiple\_fault\_combinations = \#faults.\frac{\#fanouts}{2.\#outputs}.$$

This is sizable minimization and it is possible to be done.

For the second problem we suggest to find one more test for the fault with redundant output values, therefore the faults become distinguishable. Let us return to the circuit on Fig. 4. If the single fault z_1 / *sa*1 persists in circuit, for the input pattern ($a = 1$, $b = 0$, $c = 0$, $d = 1$) output pattern is ($z_1 = 1$, $z_2 = 0$). It is the same output pattern as if the fault pair (*e* / *sa*1, z_2 / *sa*0) persists in the circuit. The faults are detected but they could not be distinguished. The new input pattern ($a = 0$, $b = 1$, $c = 1$, $d = 0$) detects fault z_1 / *sa*1 and does not detect faults *e* / *sa*1 and z_2 / *sa*0. So they become distinguishable.

3.2. Multiple faults selection

For test pattern generation we use fault reduced dictionaries with equivalent fault collapsing. For fault masking effect investigation it is useful to represent a collapsed group with the most closest to an output fault.

How to select the masking faults and pairs for multiple simulations? Let us focus on the AND gate on fig.3. We can distinguish the next four variants according to the stuck-at-1 faults on the inputs:

- fault *a* / *sa*1 can be masked by *z* / *sa*0;
- *fault b* / *sa*1 can be masked by *z* / *sa*0;
- fault *a* / *sa*1 can be masked by *b* / *sa*0;
- fault *b* / *sa*1 can be masked by *a* / *sa*0.

But *a* / *sa*0, *b* / *sa*0 and *z* / *sa*0 are structural equivalent faults. Therefore we can reduce combinations to the first two.

As mentioned before for every path without fanouts all masking faults (for this observed fault) are collapsed to the last one. The rules for collapsing are too complex for analytical description and we suggest our masking simulator for this purpose.

The algorithm for finding the pairs for multiple fault simulation contains of two stages:

1. Building the collection of potential masked faults *MdF*. In this collection we must include all single faults except dominated faults. It is not necessary to observe dominated faults for masking, as the result will be the same as the dominating fault.
2. Building the collection of masking faults MgF_i for every fault $f_i \in MdF$. As mentioned before for every path without fanouts all masking faults (for the observed fault) are collapsed to the last one. The rules for collapsing are too complex for analytical description and we suggest our masking simulator for this purpose.

Masking simulator is the simulator which can propagate one fault to every out and can keep the fault free and fault levels on every line. For every fanout and every primary output this simulator generates a masking fault which is such that the line level becomes equal to the fault free circuit. This simulator is time and memory economical because it tracks only the fault propagation path and start tracking from fault line.

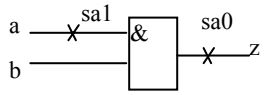

Fig. 3. Fault effect masking.

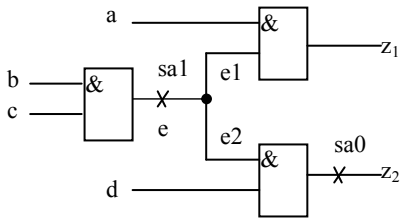

Fig. 4. Overcome fault effect masking

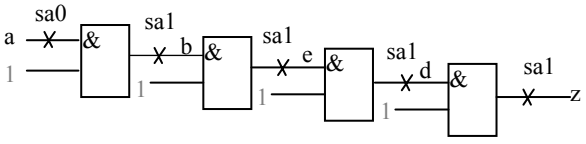

Fig. 5. Fault effect masking equivalence.

Table 1

Collapsed group for multiple fault simulations

Gate type	*Multiple fault group*
AND	• for every input *sa*1 & *sa*0 on the output; • *sa*0 & *sa*1 on the output;
NAND	• for every input *sa*1 & *sa*1 on the output; • *sa*1 & *sa*0 on the output;
OR	• for every input *sa*0 & *sa*1 on the output; • *sa*1 & *sa*0 on the output;
NOR	• for every input *sa*0 & *sa*0 on the output; • *sa*0 & *sa*1 on the output;

Table 2

The number of pairs for multiple fault simulations

Benchmark circuit	Number of full fault pairs	Number of reduced masking pairs	Collapsing ratio
c499.bench	286,903	18,244	15
c432.bench	137,026	8,409	16
c880.bench	443,211	8,128	54
c1355.bench	1,237,951	95,332	12
c1908.bench	176,4381	77,594	22
c2670.bench	3,771,631	36,623	102
c3540.bench	5,873,878	177,070	33
c5315.bench	14,308,575	71,225	200

4. TEST PATTERN GENERATION BASED ON MULTIPLE FAULT SIMULATION

Test generation procedure can be divided into two stages.

During the first stage ATPG and z-set principals are used for producing such test patterns that every single fault could be distinguishable.

1. Building single fault dictionary (SFD) by collapsing structural equivalent faults.

2. Constructing *Z*-sets and sets of test patterns for every single fault in the dictionary.

The aim of the second stage is to get over the masking effect using the multiple fault simulation and multiple tests for every fault.

1. Building masking dictionary (MD). Every record contains a masked fault and the set of reduced masking faults, which are collected according the algorithm1.

2. Performing the multiple fault simulation for every pair masked-masking fault in the masking dictionary and for every test pattern which can detect the masked fault as a single fault.

3. Compare the resulting output states with these in the SFD. If there is a duplicated results, generating a new extra test pattern for the faults pair is required to guarantee the faults distinguishability.

As a result in the MD there are records with complex structure (tree structure). For every fault $f_i \in MD$ there is test pattern t_0, which can detect and diagnose f_i as a single fault and test patterns T_i. Every test pattern $t_{i_j} \in T_i$ can detect and diagnose f_i if masking fault f_j persist in the circuit.

4. Ranking all faults in the MD by the masking level (*ml*) parameter. Masking level can be calculated by a number of pairs in which the fault persists as a masking fault.

5. Sorting the MD by *ml* in ascending direction. The fault, which influence the most of the other faults will be on the top.

5. EXPERIMENTAL RESULTS FOR MULTIPLE FAULTS COLLAPSING

We have made a tool for multiple faults collapsing which includes a masking simulator and have performed experiments on several benchmark circuits. From the experimental results shown in the Table 2 it can be seen that the fault pairs are significantly collapsed. The collapsing ratio varies from 12 to 200, depending on the structure of the circuit.

5. CONCLUSIONS

We suggest our method for masking faults diagnosis which is based on multiple fault simulation. This method consists of two algorithms. The first one is for multiple faults collapsing and getting the masking fault dictionary. The second is for diagnosis test generation with multiple fault simulation and built the rating set with records which has the tree-like test structure.

REFERENCES

[1] Liu, J.B., Veneris, A., (2005). *Incremental fault diagnosis*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Vol. 24, No. 2, Feb. 2005, pp. 240–251, ISSN: 0278–0070.

[2] Ruifeng G., Huang, Y., Cheng, W.-T., (2007). *Fault Dictionary Based Scan Chain Failure Diagnosis*, Asian Test Symposium, 2007, pp. 45–52, ISBN: 978–0–7695–2890–8, Beijing, China, 8–11 Oct. 2007, IEEE Computer Society, Beijing.

[3] Arslan, B., Orailoglu, A., (2002). *Fault dictionary size reduction through test response superposition*, Computer Design: VLSI in Computers and Processors, 2002, Proceedings, 2002 IEEE International Conference on , pp. 480–485, ISBN: 0–7695–1700–5, Sept. 2002.

[4] Pomeranz, I., Reddy, S.M., Venkataraman, S., (2007). *z-Diagnosis: A Framework for Diagnostic Fault Simulation and Test Generation Utilizing Subsets of Outputs*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, Vol. 26, No. 9, (Sept. 2007), pp. 1700–1712, ISSN: 0278–0070.

[5] Pomeranz, I., Reddy, S.M., (2009). *Same different fault dictionary: an extended pass/fail fault dictionary with improved diagnostic resolution*, Computers & Digital Techniques, IET, Vol. 3, No. 1, January 2009, pp. 85–93, ISSN: 1751–8601

[6] Seshadri, B., Yu, X., Venkataraman, S., (2006). *Accelerating diagnostic fault simulation using z-diagnosis and concurrent equivalence identification*, VLSI Test Symposium, 2006. Proceedings, 24th IEEE, ISBN: 0–7695–2514–8, Berkeley, CA, April–May 2006.

Authors:

Pavlinka RADOYSKA, Assistant Prof., TU-Sofia, KEE,
E-mail: pradoiska@abv.bg
PhD, Kamen FILLYOV, Professor, TU-Sofia, FCSU
E-mail: kfillyov@ecad.tu-sofia