PAPER Special Section on Test and Verification of VLSIs

A Novel Per-Test Fault Diagnosis Method Based on the Extended *X*-Fault Model for Deep-Submicron LSI Circuits

Yuta YAMATO[†a)], ***Student Member***, **Yusuke NAKAMURA**[†], ***Nonmember***, **Kohei MIYASE**[†], **Xiaoqing WEN**[†], ***and*** **Seiji KAJIHARA**[†], ***Members***

SUMMARY Per-test diagnosis based on the *X*-fault model is an effective approach for a circuit with physical defects of non-deterministic logic behavior. However, the extensive use of vias and buffers in a deep-submicron circuit and the unpredictable order relation among threshold voltages at the fanout branches of a gate have not been fully addressed by conventional per-test *X*-fault diagnosis. To take these factors into consideration, this paper proposes an improved per-test *X*-fault diagnosis method, featuring (1) an extended *X*-fault model to handle vias and buffers and (2) the use of occurrence probabilities of logic behaviors for a physical defect to handle the unpredictable relation among threshold voltages. Experimental results show the effectiveness of the proposed method.

key words: *fault diagnosis, X-fault model, per-test, via*

1. Introduction

Fault diagnosis is the most widely used approach to help localize physical defects in a failing LSI circuit [1]. In fault diagnosis, a *fault model* in an abstract circuit model (usually a gate-level netlist) is used to represent the logical behavior of physical defects in an actual LSI circuit. A fault is considered *responsible* if the simulated response of the fault in the circuit model matches the observed response of the failing circuit under certain criteria used in a *fault diagnosis procedure* [2]. The locations of physical defects are then identified with the help of the layout information on such responsible faults. Clearly, a good fault model and a good diagnosis procedure are needed in order to obtain sufficient resolution in fault diagnosis.

A good fault model for fault diagnosis needs to closely resemble underlying physical defects and compensate for the modeling gap between logical faults and physical defects with respect to two attributes: *location* and *logical behavior*. In a gate-level circuit model, the location attribute is one or more nets or pins, and the logical behavior attribute is one or more logic values. *Fault modeling* defines these attributes in a general manner. On the other hand, Physical defects can be characterized from three aspects: *complexity* (simple or complex), *temporality* (static or dynamic), and *cardinality* (single or multiple). A simple defect forces a single site to a fixed logic value of 0 or 1. A complex defect, such as a resistive short or open, causes multiple effects around the defect site. For example, a complex defect in a fanout gate forces its output to an intermediate voltage and multiple faulty logic values may appear at its fanout branches depending on the threshold voltages of the branches [7]. A static defect shows the same behavior for all input vectors, while a dynamic defect changes its behavior for different input vectors because the strength of a signal may vary for different input conditions. Finally, a circuit may contain a single or multiple defects.

The defect complexity issue has been addressed by using a set of simple fault models or by using a realistic fault model. For example, [3] uses four fault models to cover various defects. On the other hands, various realistic fault models, such as stuck-open [3], bridging [4], [5], transistor leakage [6], and Byzantine [7], [8], better reflect actual defect mechanisms.

Recently, a new realistic fault model, called the *X*-fault model [9], [10], has been proposed for modeling complex defects, especially those with unpredictable and non-deterministic logic behavior. The *X*-fault model represents all possible behaviors of any defect or defects in a gate and/or on its fanout branches by assigning different *X* symbols to the fanout branches. This makes the *X*-fault model highly accurate since no defect information is lost in fault modeling. In addition, partial symbolic fault simulation, instead of full symbolic fault simulation [8], is used in *X*-fault simulation in order to achieve high time efficiency. Since an ever-increasing portion of physical defects in deep-submicron LSI circuits manifest themselves by unpredictable and non-deterministic logic behavior [11], using *X*-fault model in fault diagnosis is becoming more and more advantageous.

On the other hand, *Per-test fault diagnosis* is gaining popularity as an effective approach to handle the cardinality and temporality issues of complex defects [12]. The basic idea is to process failing vectors separately, one at a time, in fault diagnosis, based on the observation that only one of the multiple defects in an LSI circuit may be activated by one failing vector in some cases. This allows a single fault model to be assumed for the activated defect and a relatively easy fault diagnosis procedure based on single fault simulation to be used for multiple and/or dynamic defects.

Several per-test fault diagnosis methods have been proposed [3], [9], [12]–[14]. The single stuck-at fault model is used in [12]–[14], while a combination of stuck-at, stuck-open, net, and bridging fault is used in [3]. These methods

Manuscript received April 11, 2007.
Manuscript revised August 18, 2007.

[†]The authors are with the Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.
a) E-mail: yamato@aries30.cse.kyutech.ac.jp

DOI: 10.1093/ietisy/e91–d.3.667

Copyright © 2008 The Institute of Electronics, Information and Communication Engineers

attempt to find a minimal set of fault that explains as many failing vectors as possible. Such a fault set is called a *multiplet* in [12]. In addition, [3] calculates a score for a fault depending on the number of failing vectors explained by the fault, and [12] further extracts diagnostic information from multiplets, while [14] scores each multiplet based on probability functions.

Recently, a per-test fault diagnosis method based on the X-fault model has been proposed [9]. On top of the accuracy achieved by X-fault modeling, this fault diagnosis procedure employs a flexible matching criterion that takes matching details into consideration. The detailed diagnostic information extracted from the relation between an observed response and a simulated response is expressed as a *diagnosis value* for each X-fault and each failing test vector, and all diagnosis values form a *diagnosis table*, from which multiplets are obtained and ordered. It has been shown that such per-test X-fault diagnosis can achieve high diagnostic resolution for complex, multiple, and/or dynamic defects.

However, the conventional per-test X-fault diagnosis method [9] faces two serious problems, as described bellow:

Problem 1: *The existence of vias is not considered in relation to fanout branches.*

In the conventional X-fault model [9], the existence of vias is ignored. This is illustrated in Fig. 1, where the gate G is considered to have 6 fanout branches, $L_1 \sim L_6$. That is, 6 X symbols, $X_1 \sim X_6$, need to be assigned to $L_1 \sim L_6$, respectively. In reality, however, vias or buffers may exist as shown in Fig. 1. Since X-fault diagnosis cannot identify defective fanout branches, treating many lines as fanout branches from the same gate may reduce diagnostic resolution.

Problem 2: *Difference in occurrence probabilities of possible logic behaviors of a physical defect is ignored.*

In the conventional X-fault diagnosis procedure [9], all possible logic combinations at the fanout branches of a gate are considered to be equally likely. In reality, however, this is usually not true. This is illustrated in Fig. 2, where the gate G has a defective voltage V_m, which may be intermediate. In Fig. 2 (a), the gate G has two fanout branches, L_1 and L_2, corresponding to threshold voltages V_{th1} and V_{th2}, respectively. In Case-1 ($V_{th1} < V_{th2}$), $\langle 00 \rangle$, $\langle 10 \rangle$, and $\langle 11 \rangle$ may occur at L_1 and L_2. However, in Case-2 ($V_{th1} > V_{th2}$), $\langle 00 \rangle$, $\langle 01 \rangle$, and $\langle 11 \rangle$ may occur at L_1 and L_2. Note that, in a deep-submicron LSI circuit, especially one with low-voltage design, the order relation between V_{th1} and V_{th2} may be unpredictable due to process variation. That is, both Case-1 and Case-2 may show up. Under this condition, it is clear that the occurrence probabilities of $\langle 00 \rangle$, $\langle 10 \rangle$, $\langle 01 \rangle$, $\langle 11 \rangle$ are 2/6, 1/6, 1/6, and 2/6, respectively, which are clearly not equal. Therefore, treating the occurrence probabilities of all logic combinations at the fanout branches of a gate as equal in an X-fault diagnosis procedure may either reduce diagnostic resolution or produce misleading diagnostic results.

Therefore, there is a need to solve Problem 1 and Problem 2 in order to improve diagnostic resolution and diagnosis efficiency, and this is the focus of this paper.

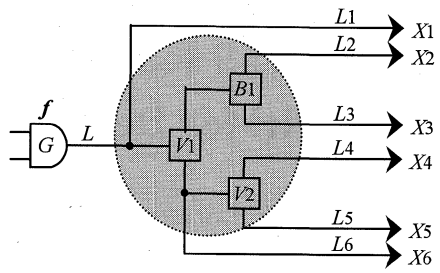
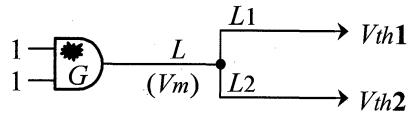
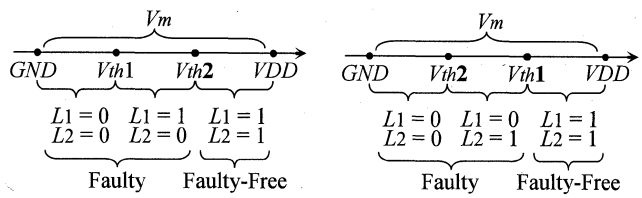


Fig. 1 Vias, buffers, fanout branches, and X-fault.



(a) Defective Voltage



(b) Case-1 ($V_{th1} < V_{th2}$) (c) Case-2 ($V_{th1} > V_{th2}$)

Fig. 2 Defective voltage and possible logic combinations.

2. Previous Per-Test X-Fault Diagnosis Method

2.1 X-Fault Model

The conventional X-fault model [9] is defined as follows:

Definition 1: A fanout gate has one X-fault, corresponding to any physical defect or physical defect or defects in the gate or on its n fanout branches. The X-fault assumes n different X symbols on the n fanout branches to represent all possible faulty logic values in fault simulation.

Figure 3 shows the X-fault for an AND gate with two fanout branches, where X_1 and X_2 denote two arbitrary faulty logic values. Clearly, $\langle X_1, X_2 \rangle$ represents any possible faulty logic combination that may appear on the fanout branches. Note that the conventional X-fault model treats all fanout branches as directly connected signal lines, without considering that vias may exist at fanout branches.

The X-fault model has the following characteristics:

- **Generality:** One single X-fault can represent all possible faulty logic combinations at a defect site, while multiple stuck-at faults are needed to cover the same defect scenario.
- **Size:** The number of X-faults in a circuit is manageable since it is equal to the number of gates in the circuit.
- **Accuracy:** The X-fault model handles the unknown faulty behavior of a complex defect, such as a

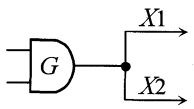


Fig. 3 Conventional X-fault model.

Byzantine defect, with different X symbols. No diagnostic information is lost since no effort is made to aggressively determine faulty logic values with an assumption, such as wired-AND, wired-OR, driving, etc., that may not be always true in reality.

- **Flexibility:** The X-fault model allows a fault to have different faulty behaviors under different input vectors, making it suitable for the per-test fault diagnosis scheme in handling dynamic defects.

2.2 X-Fault Simulation

Given an X-fault f and an input vector v, X-fault simulation is to obtain the simulated response of f under v, denoted by $SimRes(f, v) = \{R_1, R_2, \ldots, R_k\}$, where $R_1, R_2, \ldots, R_k$ $(k \geq 1)$ are logic combinations at primary outputs, corresponding to k possible faulty logic combinations at $C_1, C_2, \ldots, C_k$, at the site of f, respectively. Generally, X-fault simulation uses a partial symbolic procedure, consisting of three steps:

(1) ***X-Injection*:** Assign different X symbols to the fanout branches of a gate.
(2) ***X-Propagation*:** Propagate X symbols to primary outputs.
(3) ***X-Resolution*:** Resolve all X symbols to primary outputs to obtain a final simulation result [9].

An example is shown in Fig. 4.

In Fig. 4 (a), *X-injection* assigns 3 initial X symbols, $X_1(b_1)$, $X_2(b_2)$, and $X_3(b_3)$, to the fanout branches, b_1, b_2, and b_3, of the gate G_1, respectively. In Fig. 4 (b), *X-propagation* is conducted by keeping inversion function but ignoring all other logic functions. For example, the output of the gate G_4 is the AND function of $X_1(b_1)$ and $X_2(b_2)$. This functional information is ignored and the result is a new X symbol $X_4(b_1, b_2)$, in which (b_1, b_2) is used to indicate that the output of G_4 only comes from branches b_1 and b_2.

In Fig. 4 (c), *X-Resolution* is conducted to remove the ambiguity due to X symbols existing in the initial simulated response. Since b_3 is not responsible, only three possible faulty logic combinations, C_1, C_2, and C_3, need to be considered at the fault site. As a result, the final simulated responses is $SimRes(f, v) = \{R_1, R_2, R_3\}$, where R_1, R_2, and R_3 correspond to C_1, C_2, and C_3, respectively. Note that in the conventional X-fault simulation procedure, C_1, C_2, and C_3, are assumed to be equally likely, i.e. each of them having an occurrence probability of 33%. In reality, this assumption is usually not true. This problem will be addressed in this paper by calculating the occurrence probability of each possible faulty logic combination at fanout branches, and by using this probabilistic information in X-fault diagnosis.

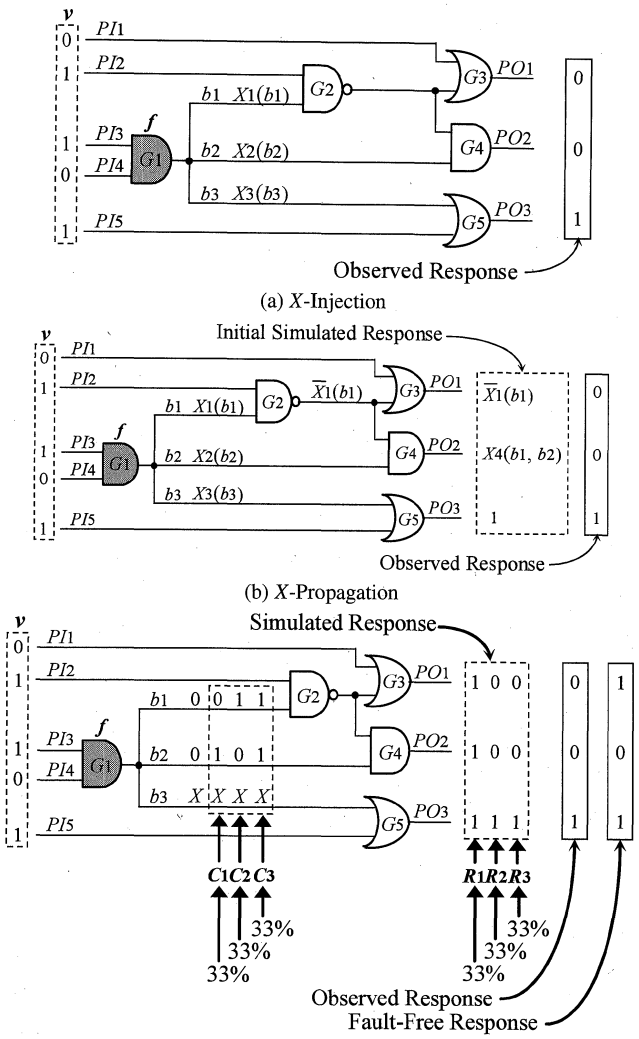


Fig. 4 X-fault simulation.

2.3 Diagnosis Value Calculation

After conducting X-fault simulation for an X-fault f under a failing vector v, the simulated response $SimRes(f, v) = \{R_1, R_2, \ldots, R_k\}$ needs to be compared with the observed response $ObvRes(v)$ to extract diagnostic information. The comparison result is represented by a so-called *diagnosis value* under v and f, denoted by $d(f, v)$, and the conventional method to calculate $d(f, v)$ is as follows [9]:

$$d(f, v, R_i) = \frac{Level(f)}{L_{max}} \times \frac{|Error_PO(v) \cap Reach_PO(f)|}{|Reach_PO(f)|}$$

if R_i is the same as $ObvRes(v)$ on $Reach_PO(f)$; otherwise, $d(f, v, R_i) = 0$. And,

$$d(f, v) = \left(\sum_{i=1}^{k} d(f, v, R_i) \right) \Big/ k$$

Table 1 Fault diagnosis table.

	$f1$	$f2$	$f3$	$f4$	$f5$
$v1$	0.81	0.65	0	0	0
$v2$	0	0	0.61	0.17	0
$v3$	0.26	0	0.83	0	0
$v4$	0	0	0	0	0.55
ave	0.27	0.16	0.36	0.04	0.14

Here, $Error_PO(v)$ is the set of all primary outputs on which an observed response has errors, and $Reach_PO(f)$ is the set of primary outputs that is reachable from the gate with the X-fault f. In Fig. 4 (c), $Error_PO(v) = \{PO_1\}$ and $Reach_PO(f) = \{PO_1, PO_2, PO_3\}$. Moreover, $Level(f)$ is the level of the output of the gate with the X-fault f and L_{max} is the maximum level in the circuit, assuming that all primary outputs have level 1. In Fig. 4, $Level(f) = 3$ and $L_{max} = 3$.

For the X-fault simulation result shown in Fig. 4 (c), $SimRes(f, v) = \{R_1, R_2, R_3\} = \{\langle 111\rangle, \langle 001\rangle, \langle 001\rangle\}$, $ObvRes(v) = \langle 001\rangle$, and the fault-free simulation result is $\langle 101\rangle$. Clearly, $d(f, v, R_1) = 0$ and $d(f, v, R_2) = d(f, v, R_3) = (3/3) \times (1/3) = 0.33$. Therefore, $d(f, v) = (0 + 0.33 + 0.33)/3 = 0.22$.

Diagnosis values are calculated for all failing vectors and faults, and they are stored in a table called a *fault diagnosis table*. Table 1 is an example fault diagnosis table which contains the average diagnosis value for each fault that is used in scoring a fault diagnosis result. Clearly, compared with a normal fault dictionary with only 0 and 1 entries, a fault diagnosis table contains more diagnostic information.

It is obvious that diagnosis values are calculated with unique matching criteria, which take the reachable range of primary outputs, the number of matched errors, and the depth of a fault into consideration [9]. Note that, however, this calculation method follows the assumption that all possible faulty logic combinations at the fanout branches of a gate are equally likely, which may not be true in reality.

2.4 Per-Test X-Fault Diagnosis Flow

Figure 5 shows the per-test X-fault diagnosis flow [9], which consists of two stages, one for collecting diagnostic information and the other for drawing diagnostic conclusion.

In Stage-1 (*Information-Collecting*), all X-faults are simulated for each failing vector, the simulated responses are compared with observed responses, and a diagnosis table is created. In Stage-2 (*Diagnostic-Reasoning*), a diagnosis result is produced from the fault diagnosis table. The basic processing is to find a minimal set of X-faults that cover all failing vectors corresponding to non-all-zero rows in a fault diagnosis table. Such a fault set is called a *multiplet*. Then, the score of each multiplet is calculated by adding up the diagnosis values of all composing X-faults. These scores are then used to determine the order of multiplets, and the X-faults in top multiplets form the final fault diagnosis result.

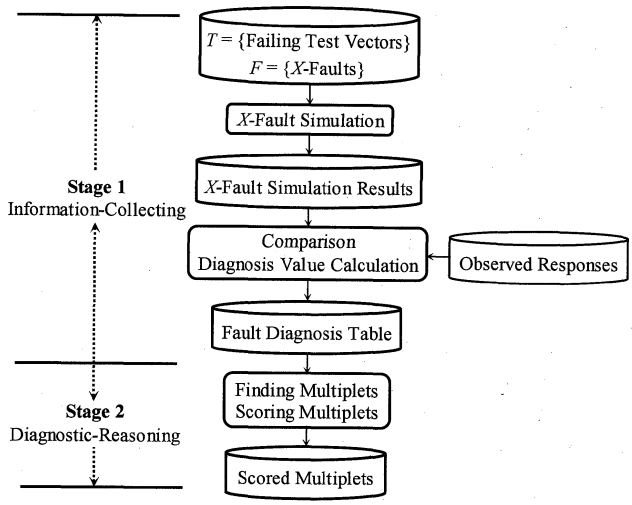


Fig. 5 Per-test X-fault diagnosis flow.

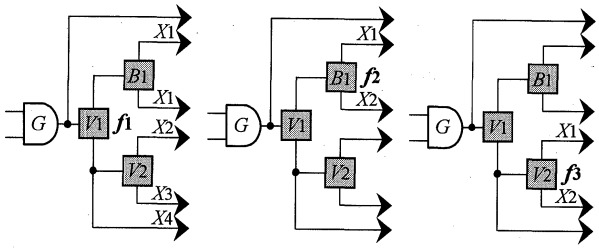
3. Improved Per-Test X-Fault Diagnosis

As described above, there are two major problems with the conventional per-test X-fault diagnosis: (1) treating all fanout branches of a gate as directly connected signal lines without considering the existence of vias, and (2) assuming the all possible logic combinations at the fanout branches of a gate have equal occurrence probabilities. The first problem is addressed in Sect. 3.1 with an extended X-fault model and the second problem is addressed in Sect. 3.2 with a method for calculating the occurrence probability of each possible logic combination. Section 3.3 defines a new diagnosis value based on occurrence probability.

Previous fault diagnosis methods [15], [16] exist for handling open-via defects. In [15], the logic uncertainty due to intermediate defective voltages caused by an open-via defect is represented by different X-symbols, in a manner similar to the X-fault model [9]. However, different from [9] that uses partial symbolic simulation, [15] uses full symbolic simulation to process X-symbols, which makes it difficult to be applied to a large circuit. In [16], physical information is used to predict the logic behavior of an open-via defect. If physical information extraction is possible and accuracy can be guaranteed, this method will generate good diagnostic results. However, its applicability is limited by the availability of physical design information and tool accuracy.

3.1 Extended X-Fault Model

The conventional X-fault model [9] assumes one X-fault for each gate that may have any number of fanout branches. This is illustrated in Fig. 1, where all fanout branches, $L_1 \sim L_6$, from the gate G are treated without any distinction from each other. Note that, when the X-fault f is assumed at G, as shown in Fig. 1, 6 X symbols, $X_1 \sim X_6$, are assigned to the branches. The disadvantage of this X-fault model is that it can only lead to the finding that a fault and its fanout



(a) X-Fault f_1 (b) X-Fault f_2 (c) X-Fault f_3

Fig. 6 Extended X-fault model.

branches may be defective, without sufficient information to identify which fanout branch is more likely to be defective.

In reality, the layout of a deep-submicron LSI circuit usually involves multiple layers, which means that vias are extensively used. In addition, buffers are used to raise driving ability when fanout elements have a lot of branches. In order to handle such via and buffer information, the conventional X-fault model [9] is extended as follows:

Definition 2: A fanout element (gate or via or buffer) has one X-fault, corresponding to any physical defect or defects in the element or on its fanout branches. The X-fault assumes different X symbols on the fanout branches of the element to represent non-deterministic faulty logic values in fault simulation.

Examples of the extended X-fault model are shown in Fig. 6 (a)~(c). Since there are two vias V_1 and V_2 and one buffer b_1. 3 extended X-faults, $f_1 \sim f_3$, are added, in addition to the conventional X-fault f shown in Fig. 1. 4, 2, and 2 different X symbols are assigned for the three extended X-faults, f_1, f_2, and f_3, respectively. Obviously, these new added X-faults are all different from the conventional X-fault f shown in Fig. 1.

Note that, when a buffer exists on the paths from an assumed X-fault to its branches, the same X symbols are assigned to all branches connected to the buffer. In Fig. 6 (a), buffer B_1 exists on path from X-fault f_1 to its two branches. In this case, X_1 is assigned to both branches.

The most significant advantage of the extended X-fault model is that it can locate defects to more detailed level, which greatly improves the diagnostic resolution. In the case of Fig. 6, diagnostic results now can be obtained with respect to V_1, V_2, and B_1, instead of only G.

3.2 Occurrence Probability Calculation

Suppose that a gate or via has n fanout branches, $L_1, L_2, \ldots$, and L_n, whose corresponding threshold voltages are $V_{th1}, V_{th2}, \ldots$, and V_{thn}, respectively. If the order of $V_{th1}, V_{th2}, \ldots$, and V_{thn} is fixed and known, there will be exactly $n + 1$ possible logic combinations at the fanout branches [17].

In a real LSI circuit, however, process variation in the deep sub-micron era and shrinking difference among threshold voltages in low-voltage design increasingly make

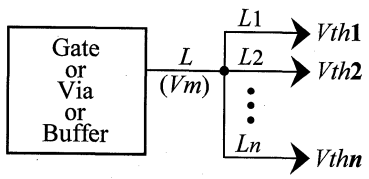


Fig. 7 Fanout branches.

it difficult to deterministically know the order of threshold voltages corresponding to the fanout branches of a gate or via or buffer. In other words, it is necessary to consider all possible orders of threshold voltages at the fanout branches. As illustrated in Fig. 2, this results in different occurrence probabilities of logic combinations at the fanout branches of a gate or via or buffer. This phenomenon can be quantitatively expressed by the following theorem.

Theorem 1: For a gate or via or buffer with n fanout branches, the total number of possible orders of threshold voltages at the fanout branches is $n!$. In addition, the probability that the fanout branches have a logic combination with p 0's ($0 \le p \le n$) is $(p! \times (n-p)!)/(n+1)!$.

Proof: Consider the general case shown in Fig. 7. Here, the n fanout branches, $L_1 \sim L_n$, whose corresponding threshold voltages are $V_{th1} \sim V_{thn}$, respectively. In addition, it is assumed that the stem L has a non-deterministic voltage V_m. Depending on the relations of L with $V_{th1} \sim V_{thn}$, different logic combinations may appear on $L_1 \sim L_n$ [17].

1st Half: When n threshold voltages are ordered, there are n choices for the first threshold voltage, $(n-1)$ choices for the second threshold voltage, ..., and 1 choice for the n-th threshold voltage. As a result, there are a total of $n!$ possible orders of n threshold voltages.

2nd Half: First, for each order of n threshold voltages, there are $(n+1)$ possible logic combinations [17], depending on which of the $(n+1)$ voltage intervals in the order the corresponding intermediate voltage V_m falls in. That is, the total number of occurrences of possible logic combinations is $(n+1) \times n! = (n+1)!$ In addition, a logic combination having p 0's ($0 \le p \le n$) means that p threshold voltages are lower than V_m and $(n-p)$ threshold voltages are higher than V_m, where V_m is the corresponding intermediate voltage. As a result, there are $p! \times (n-p)!$ logic combination with p 0's. Therefore, the occurrence probability of such a logic combination is $(p! \times (n-p)!)/(n+1)!$ □

The occurrence probabilities of logic combinations at n fanout branches are summarized in Table 2 (a), and the special case of $n = 3$ is shown in Table 2 (b). Clearly, logic combinations with different numbers of 0's may have different occurrence probabilities.

3.3 Use of Occurrence Probabilities

3.3.1 New X-Resolution

Based on Theorem 1, one can determine the occurrence

Table 2 Occurrence probabilities of logic combinations.

(a) Case of n fanout branches

Number of 0's in logic combination	Number of Occurrences	Probability of Occurrence
0	$n!$	$1 / (n+1)$
1	$1! \times (n-1)!$	$1 / (n \times (n+1))$
2	$2! \times (n-2)!$	$2 / ((n-1) \times n \times (n+1))$
⋮	⋮	⋮
p	$p! \times (n-p)!$	$(p! \times (n-p)!) / (n+1)!$
⋮	⋮	⋮
n	$n!$	$1 / (n+1)$

(b) Case of 3 fanout branches

Number of 0's in logic combination	Number of Occurrences	Probability of Occurrence
0	6	25%
1	2	≈ 8%
2	2	≈ 8%
3	6	25%

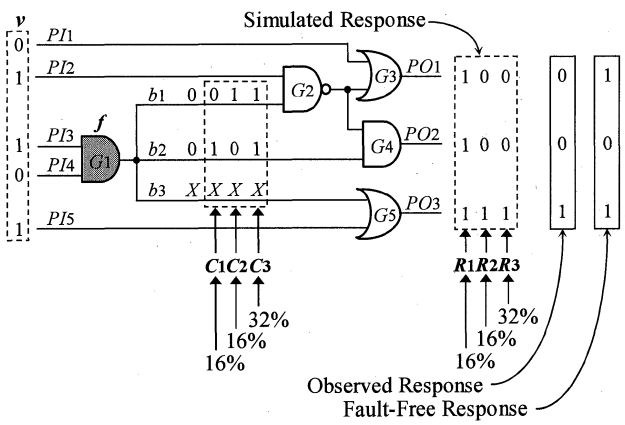


Fig. 8 New X-resolution.

probabilities for all possible faulty logic combinations at the fanout branches of a gate or via or buffer. This information is used in X-resolution during X-fault simulation, in order to better reflect the reality in deep-submicron LSI circuits.

Consider the X-resolution example shown in Fig. 8 for simulating the X-fault at the gate G_1. This gate has 3 fanout branches, b_1, b_2, and b_3, and there are 3 possible faulty logic combinations, $C_1 = \langle 01X \rangle$, $C_2 = \langle 10X \rangle$, and $C_3 = \langle 11X \rangle$. Since C_1 represents $\langle 010 \rangle$ and $\langle 011 \rangle$ that both have the occurrence probability of 8%, the occurrence probability of C_1 is 16%. Similarly, the occurrence probabilities of C_2 and C_3 can be obtained as 16% and 32%, respectively. As a result, the occurrence probabilities of C_1, C_2, and C_3 are 16%, 16%, and 32%, respectively, which are different from the conventional assumption that all of them have the same occurrence probability of 25% as shown in Fig. 4 (c).

Suppose that the possible logic combinations at the fanout branches for X-fault f under test vector v are $C_1 \sim C_n$, whose occurrence probabilities are $p(C_1), p(C_2) \ldots$, and $p(C_n)$, respectively. Note that $p(C_1), p(C_2) \ldots$, and $p(C_n)$ can be readily calculated by using Theorem 1. Also suppose that the simulated response $SimRes(f, v) = \{R_1, R_2, \ldots, R_n\}$, where $R_1, R_2, \ldots$, and R_n are resulting logic combinations at primary outputs, corresponding to $C_1 \sim C_n$, respectively. Clearly, the occurrence probabilities of $R_1, R_2, \ldots$, and R_n, denoted by $p(R_1), p(R_2) \ldots$, and $p(Rn)$, respectively, are equal to $p(C_1), p(C_2) \ldots$, and $p(C_n)$, respectively.

For example, in Fig. 8, since the occurrence probabilities of logic combinations at fanout branches, C_1, C_2, and C_3, are 16%, 16%, and 32%, respectively, the occurrence probabilities of logic combinations at primary outputs, R_1, R_2, and R_3 are also 16%, 16%, and 32%, respectively.

3.3.2 New Diagnosis Value Calculation

In the conventional definition of diagnosis value [9] described in 2.3, it is assumed that all possible faulty logic combinations at the fanout branches of a gate are equally likely, which may not be true in reality. In the following, a new definition of diagnosis value is presented to take the difference in occurrence probabilities into consideration.

Generally, the simulated response $SimRes(f, v) = \{R_1, R_2, \ldots, R_k\}$ is compared with the observed response $ObvRes(v)$ to extract diagnostic information, and the comparison result is represented by a *diagnosis value* under v and f, denoted by $d(f, v)$, and the new method to calculate $d(f, v)$ is as follows:

$$d(f, v, R_i) = \frac{Level(f)}{L_{max}} \times \frac{|Error_PO(v) \cap Reach_PO(f)|}{|Reach_PO(f)|}$$

if R_i is the same as $ObvRes(v)$ on $Reach_PO(f)$; otherwise, $d(f, v, R_i) = 0$. And,

$$d(f, v) = \sum_{i=1}^{k} (d(f, v, R_i) \times p(R_i))$$

Here, $Error_PO(v)$, $Reach_PO(f)$, $Level(f)$, and L_{max} are all as defined in Sect. 2.3.

In Fig. 8, for example, $SimRes(f, v) = \{R_1, R_2, R_3\} = \{\langle 111 \rangle, \langle 001 \rangle, \langle 001 \rangle\}$, $ObvRes(v) = \langle 001 \rangle$, and the fault-free simulation result is $\langle 101 \rangle$. Thus, $d(f, v, R_1) = 0$, $d(f, v, R_2) = d(f, v, R_3) = (3/3) \times (1/3) = 0.33$. Therefore, $d(f, v) = 0 \times 16\% + 0.33 \times 16\% + 0.33 \times 32\% = 0.16$. Clearly, this diagnosis value is different from the diagnosis value calculated in Sect. 2.3 for the example of Fig. 4 (c).

3.4 Improved Per-Test X-Fault Diagnosis Flow

The general flow of the improved per-test X-fault diagnosis is basically the same as shown in Fig. 5. The differences are as follows:

- The extended X-fault model described in Sect. 3.1 is used. Since via and buffer information is utilized, it becomes possible to locate defects to more detailed level, thus greatly improving the diagnostic resolution.

- The new diagnosis value described in Sect. 3.3 is used. Since the occurrence probabilities of possible faulty logic combinations are taken into consideration, the reality in a deep-submicron LSI circuit is better reflected, which contributes to the improvement of diagnostic resolution.

4. Experimental Results

Table 3 summarizes the experimental results. The number of input vectors for each circuit [18] is shown under "Vector". In each experiment, an open-via defect was randomly inserted at a fanout gate to imitate a defective chip, and the defect was assumed to cause faulty effects at multiple branches of the gate. Since there is no via information for the benchmark circuits, a hypothetical via configuration is assumed at the fanout branches of each gate. 10 experiments were conducted for each circuit, and the average number of faulty fanout branches is 2.0.

For each sample, three per-test fault diagnosis programs were run: "Old" uses the conventional X-fault model [9], "New-A" uses the extended X-fault model but the conventional method for diagnosis value calculation [9], and "New-B" uses the extended X-fault model and the proposed method for diagnosis value calculation. "SLAT" shows the average number of SLAT vectors for 10 samples. "MPLT" shows the average number of multiplets, and "Exact" shows the average percentage of exact diagnosis, i.e. a multiplet containing all inserted defects, both for 10 samples. In other words, "Exact" is the inverse of "MPLT". "FH" (First Hit) shows the average position of the first exact-diagnosis-producing multiplet in a multiplet list. "Time" shows the average run time of "New-B" (CPU: 2.6 GHz).

From the average FH results, it is clear that the use of extended X-fault model is indeed effective. Furthermore, the use of occurrence probability in diagnosis value calculation can further improve FH. However, when the extended X-fault model is used, the number of faults increases in comparison to that of the conventional X-fault model [9]. This may lead to a degrading FH value as in the case of c6288.

Table 3 Fault diagnosis results.

Circuit	Vector (#)	SLAT (#)	MPLT (#)	Exact (%)	FH(#) Old	FH(#) New-A	FH(#) New-B	New-B Time (Sec.)
c432	28	2.3	5.9	16.9	3.0	2.4	2.6	0.02
c499	52	5.6	28.2	3.5	27.3	2.2	1.0	0.05
c880	21	4.9	3.0	33.3	1.4	1.2	1.0	0.02
c1908	106	2.3	16.1	6.2	12.6	3.8	4.0	0.12
c2670	45	21.0	2.1	47.6	1.2	2.2	1.1	1.05
c6288	14	8.6	9.4	10.6	1.7	5.9	3.8	2.99
c7552	75	17.5	38.1	2.6	4.0	4.8	3.5	2.81
s1238	125	22.1	6.3	15.9	4.7	1.0	1.0	0.11
s1423	24	11.0	1.8	55.6	1.2	1.0	1.0	0.09
s5378	100	46.9	6.3	15.9	3.0	1.6	1.3	0.83
s9234	111	23.0	5.0	18.0	2.9	1.0	1.5	1.39
					5.7	**2.5**	**2.0**	—

5. Conclusions

This paper proposed a new per-test fault diagnosis method based on (1) the use of the extended X-fault model for handling vias and buffers and (2) the calculation of occurrence probabilities of possible faulty logic combinations at the fanout branches of a gate or a via. As demonstrated by experimental results, the improved per-test X-fault diagnosis method can achieve better diagnostic resolution. More experiments on larger circuits are under way to further evaluate and improve the proposed method.

References

[1] L.-T. Wang, C.-W. Wu, and X. Wen, (ed.), VLSI Test Principles and Architectures: Design for Testability, Elsevier, 2006.
[2] H. Takahashi, K.O. Boateng, K.K. Saluja, and Y. Takamatsu, "On diagnosing multiple stuck-at faults using multiple and single fault simulation," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.21, no.5, pp.362–368, 2002.
[3] S. Venkataraman and S. Drummonds, "POIROT: A logic fault diagnosis tool and its applications," Proc. Intl. Test Conf., pp.253–262, 2000.
[4] S.D. Millman, E.J. McCluskey, and J.M. Acken, "Diagnosing CMOS bridging faults with stuck-at fault dictionaries," Proc. Intl. Test Conf., pp.860–870, 1990.
[5] P. Maxwell and R. Aiken, "Biased voting: A method for simulating CMOS bridging faults in the presence of variable gate logic thresholds," Proc. Intl. Test Conf., pp.63–72, 1993.
[6] W. Mao and R.K. Gulati, "QUIETEST: A quiescent current testing methodology for detecting short faults," Proc. ICCAD'90, pp.280–283, 1990.
[7] D. Lavo, T. Larrabee, and B. Chess, "Beyond the Byzantine generals: Unexpected behavior and bridging fault diagnosis," Proc. Intl. Test Conf., pp.611–619, 1996.
[8] S. Huang, "Speeding up the Byzantine fault diagnosis using symbolic simulation," Proc. VLSI Test Symp., pp.193–198, 2002.
[9] X. Wen, T. Miyoshi, S. Kajiihara, L.-T. Wang, K.K. Saluja, and K. Kinoshita, "On per-test fault diagnosis using the X-fault model," Proc. Int'l Conf. on Computer-Aided Design, pp.633–640, 2004.
[10] X. Wen, H. Tamamoto, K.K. Saluja, and K. Kinoshita, "Fault diagnosis for physical defects of unknown behaviors," Proc. Asian Test Symp., pp.236–241, 2003.
[11] ITRS Roadmap: Test & Test Equipment, 2004 Edition, http://public.itrs.net/, 2004.
[12] T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski, "Diagnosing combinational logic designs using the single location at-a-time (SLAT) paradigm," Proc. Intl. Test Conf., pp.287–296, 2001.
[13] J. Waicukauski and E. Lindbloom, "Failure diagnosis of structured circuits," IEEE Des. Test Comput., vol.6, no.4, pp.49–60, 1989.
[14] D. Lavo, I. Hartanto, and T. Larrabee, "Multiplets, models and the search for meaning," Proc. Intl. Test Conf., pp.250–259, 2002.
[15] S.-Y. Huang, "Diagnosis of Byzantine open-segment faults," Proc. Asian Test Symp., pp.248–253, 2002.
[16] W. Zou, W.-T. Cheng, and S.M. Reddy, "Interconnect open defect diagnosis with physical information," Proc. Asian Test Symp., pp.203–209, 2006.
[17] X. Wen, H. Tamamoto, K.K. Saluja, and K. Kinoshita, "Fault diagnosis for static CMOS circuits," Proc. Asian Test Symp., pp.282–287, 1997.
[18] F. Brglez and H. Fujiwara, "A neutral netlist of combinational benchmark circuits and a target translator in FORTRAN," Proc. Intl. Symp. on Circuits and Systems, pp.663–698, 1985.

Yuta Yamato received his B.E. and M.E. degrees in Computer Sciences and Electronics from Kyushu Institute of Technology, Japan, in 2005 and 2007 respectively. Currently, he is studying towards his Ph.D. degree in the Creation Informatics Program at Graduate School of ComputerScience and Systems Engineering, Kyushu Institute of Technology, Japan. His research interest is fault diagnosis of VLSI circuits.

Yusuke Nakamura received his B.E. and degree in Computer Sciences and Electronics from Kyushu Institute of Technology, Japan, in 2007. Currently, he is studying towards his M.E. degree in the Creation Informatics Program at Graduate School of ComputerScience and Systems Engineering, Kyushu Institute of Technology, Japan. His research interest is fault diagnosis of VLSI circuits.

Kohei Miyase received his B.E., M.E., and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan, in 2000, 2002 and 2005, respectively. From 2005 to 2007, he was a researcher in Innovation Plaza Fukuoka, Japan Science and technology Agency, Japan. From 2007, he has been an Assistant Professor of Kyushu Institute of Technology. His research interests include test compaction, test compression, design for testability, low power test, and fault diagnosis. He received the Excellent Student Award of the IEEE Fukuoka Section in 2005. He is a member of the IEEE and the IPSJ.

Xiaoqing Wen received the B.E. degree from Tsinghua University, Beijing China, in 1986, the M.E. degree from Hiroshima University, Hiroshima, Japan, in 1990, and the Ph.D degree from Osaka University, Osaka, Japan, in 1993. From 1993 to 1997, he was a Lecturer at Akita University. He was a Visiting Researcher at University of Wisconsin, Madison, U.S.A., from Oct. 1995 to March 1996. He joined SynTest Technologies, Inc., U.S.A., in 1998, and served as its CTO until 2003. In 2004, he joined the Kyushu Institute of Technology, Iizuka, Japan, where he is currently a Professor. His research interests include VLSI test, diagnosis, and testable design. He is a member of the IEEE and the REAJ.

Seiji Kajihara received the B.S. and M.S. degrees from Hiroshima University, Japan, and the Ph.D. degree from Osaka University, Japan, in 1987, 1989, and 1992, respectively. From 1992 to 1995, he worked with the Department of Applied Physics, Osaka University, as an Assistant Professor. In 1996, he joined the Department of Computer Science and Electronics of Kyushu Institute of Technology, Japan, where he is a Professor currently. His research interest includes test generation, delay testing, and design for testability. He received the Young Engineer Award from IEICE in 1997, the Yamashita SIG Research Award from IPSJ in 2002, and the Best Paper Award from IEICE in 2005. Dr. Kajihara is a member of the IEEE and the IPSJ.