2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Logic Circuit Fault Models Detectable by Neural Network Diagnosis

Hisayuki Tatsumi[1], Yasuyuki Murai[2], Hiroyuki Tsuji[2],
Shinji Tokumasu[2], Masahiro Miyakawa[1]

1) Department of Computer Science, Tsukuba College of Technology
2) Department of Information and Computer Sciences, Kanagawa Institute of Technology

1) 4-12-7 Kasuga, Tsukuba, Ibaraki, 305-0821, Japan
2) 1030 Shimo-Ogino, Atsugi, Kanagawa, 243-0292, Japan

email: tatsumi@cs.k.tsukuba-tech.ac.jp

Abstract -- In order for testing faults of combinatorial logic circuit, the authors have developed a new diagnosis method : "Neural Network (NN) fault diagnosis", based on NN error back propagation functions. This method has proved the capability to test gate faults of wider range including so called SSA (single stuck-at) faults, without assuming neither any set of test data nor diagnosis dictionaries. In this paper, it is further shown that what kind of fault models can be detected in the NN fault diagnosis, and the simply modified one can extend to test delay faults, e.g. logic hazard as long as the delays are confined to those due to gates, not to signal lines.

1. Introduction

As a useful fault model, the SSA (single stuck-at) fault model has accomplished an important role for fault diagnosis of combinatorial logics. However, this model is not enough for the nanometer technologies, because it is available only for the gate defects of SA0 (stuck-at 0) and SA1 (stuck-at 1) and also, as mentioned above, it needs some sets of test data and diagnosis dictionaries beforehand [1]. So far, generation of the sets of test data and diagnosis dictionaries, e.g. "test generation" is accepted as a natural consequence of diagnosis algorithms. On the contrary, the authors' method restricts the faults to those due to gates not to signal lines, that is, only the gate faults are considered in our model. Then, it stands on the assumption that "the faults come from the defects on some gates in the logics and the defects are observed as if a defective gate were replaced by another gate with different logic function". Using the same idea that was proposed in [2], it is shown that a minor extension of the NN model could be a general gate fault tester for dealing with gate delay faults as well as SSA faults and SO (stuck-at open) faults, without assuming any set of test data nor diagnosis dictionaries.

In this paper, this new extended model, now called the SSAG (Synthetic stuck-at-gate) fault model, is applied to tests for gate delay defect and also, its advantageous features are proved by numerical experiments.

2. NN fault diagnosis

Since each type of circuit gates such as OR, AND, etc. has several fan-in lines and one fan-out line, an equivalent NN gate is constructed easily by using a set of fan-in data and the resulting fan-out data as teaching data and the error back propagation method of as its learning rule.

First of all, the feed forward NN is taken for fault diagnosis as the type of the NN gates, and has no cycles in its network. For simplicity to explain the sensitivity factor introduced by the authors [2], suppose the NN consists of the three consecutive units illustrated in Figure 1.

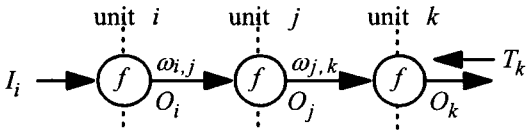


Figure 1. Simplified 3 stage model

Let I, O, f and θ be defined to the present unit as input from the preceding one, its output, the I/O function and the offset value (threshold value) respectively. Moreover, let ω, T and E as the connection weight between adjacent units, a teaching value at the final unit and the error function in the network, respectively. Here, at the final unit k, by introducing the notation DO_k and δO_k as in the eqs. (1),

$$DO_k = \partial E / \partial O_k = O_k - T_k, \quad \delta O_k = DO_k \cdot f'(I_k) \tag{1}$$

the differential coefficient of E by $\omega_{j,k}$ and that of E by θ_k are represented as in the eqs. (2) and (3).

$$\partial E / \partial \omega_{j,k} = DO_k \cdot f'(I_k) \cdot O_j = \delta O_k \cdot O_j \tag{2}$$

$$\partial E / \partial \theta_k = DO_k \cdot f'(I_k) \cdot 1 = \delta O_k \tag{3}$$

Similarly at the intermediate unit j, by introducing the notation DO_j and δO_j as in the eqs. (4),

$$DO_j = \partial E / \partial O_j = \delta O_k \cdot \omega_{j,k}, \quad \delta O_j = DO_j \cdot f'(I_j) \tag{4}$$

the differential coefficient of E by $\omega_{i,j}$ and that of E by θ_j are represented as in the eqs. (5) and (6).

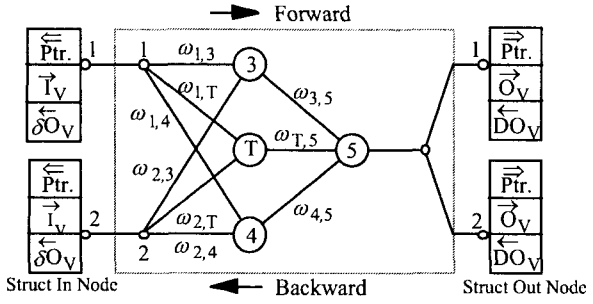
$$\partial E / \partial \omega_{i,j} = DO_j \cdot f'(I_j) \cdot O_i = \delta O_j \cdot O_i \tag{5}$$

$$\partial E / \partial \theta_j = DO_j \cdot f'(I_j) \cdot 1 = \delta O_j \tag{6}$$

Although the error back propagation method of as its learning rule is represented as the improvement to the connection weight and the offset value, the improved values are determined by DO and δO (we called these coefficients sensitivity factor).

The NN fault diagnosis uses the above idea. Namely, since a logic circuit takes only binary values 0 or 1 as I/O data on its primary signal lines or internal states on logical gates, it might be easier not only to locate fault gates, but to determinate the type of the faults than conventional ways by replacing the circuit gate with an equivalent neural gate illustrated in Figure 2 and by learning the circuit on neurons using the sensitivity factors : DO and δO. Then, for the

Forward/Backward propagation illustrated in Figure 3, if NN circuit has any fault, the sensitivity factor indicates an unusual value compared with the normal one at fault gate. Thus, the neural network circuit behaves like a diagnostic engine, and needs basically no preparation of special test patterns nor fault dictionary before diagnosing.



. Figure 2. Neural network gate

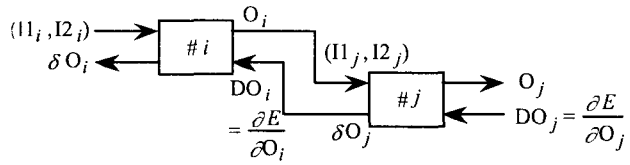


Figure 3. Forward/Backward propagation

3. Logic circuit diagnosis algorithm

The mechanism of testing diagram is formulated as shown in Figure 4, where TBox and OBox represents an objective test circuit and the corresponding sample NN circuit respectively. This sample NN circuit works as if it were a diagnostic engine feeding rating data to the succeeding fault-rating stage.

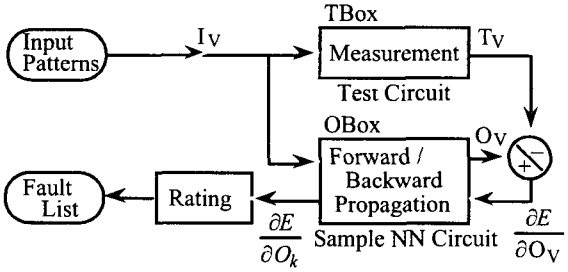


Figure 4. Testing diagram

In Figure 4, testing procedure starts, first, generating random patterns I_V's and feed them in turn to the primary signal lines of both TBox and OBox at the same time. TBox is a device and the output pattern T_V is measured by using equipments. On the other hand, since OBox is a software system, the output pattern O_V are calculated according to the forward propagation module (Forward), as well as the fan-in data I_j's and fan-out datum O_j of each NN gate $\#j$, where the O_j is normalized to 0/1 to compensate the numerical error in the NN. Then, the sensitivity factor at the primary signal lines, that is, $\partial E / \partial O_V (= O_V - T_V)$, is fed to the primary output signal lines of OBox and propagated backward to the input signal lines, according to the backward propagation module (Backward), which calculates sensitivity factor $\partial E / \partial O_j$ at the fan-out of each NN gate $\#j$ (See Figure 3).

In term of locating and determining the fault type of the gate, we assume that the number of fault gates in a test circuit is exactly one. Furthermore, diagnosis requirements are based upon the assumption that there actually existed some input-output patterns in which it is not necessarily the same between a test circuit and the corresponding sample NN circuit. If all input-output patterns of both circuits are the same, the test circuit may be regarded as being non-fault, even though both circuits are not equivalent to each other.

Under this precondition, there exist necessary conditions that the target gate to be fault. Here, an output pattern $\mathbf{O_V} = (S_j)$ for sample NN circuit and that of $\mathbf{T_V} = (T_j)$ for test circuit are obtained when the same input pattern $\mathbf{I_V} = (I_i)$ fed both circuits, and these are not the same each other, that is, $\mathbf{O_V} \neq \mathbf{T_V}$ holds. Let (N_{k_1}, N_{k_2}) and O_k be fan-ins and a fan-out at any gate $\#k$ in a sample NN circuit respectively. Now, the gate $\#k$ falls in fault, and for input-output pattern $(s_{i,j})$ (where, $\forall i, j \in \{0,1\}$) at the gate $\#k$ in a sample NN circuit, let $(t_{i,j})$ (where, $\forall i, j \in \{0,1\}$) be the input-output pattern of test circuit corresponding to the gate $\#k$. From the above assumption, there exists at least one fan-in (i, j) which $s_{i,j} \neq t_{i,j}$ (where, each $s_{i,j}$ and $t_{i,j}$ takes 0 or 1). Then, necessary conditions which the target gate to be fault are as follows.

[Condition 1] If $s_{i,j} = t_{i,j}$, then $(N_{k_1}, N_{k_2}) \neq (i, j)$. That is, with regard to any pattern that does not hold this condition, (i, j) never exists as a fan-in at the gate $\#k$.

From Condition 1, we consider only in the case satisfying $(N_{k_1}, N_{k_2}) = (i, j)$, and let O_k be an output of that case at the gate $\#k$ in a sample NN circuit. As for the error function, $\mathbf{E} = (1/2) \cdot \sum_m (S_m - T_m)^2$ holds.

[Condition 2] If $(s_{i,j}, t_{i,j}) = (1,0)$, then $\partial \mathbf{E} / \partial O_k > 0$, otherwise, if $(s_{i,j}, t_{i,j}) = (0,1)$, then $\partial \mathbf{E} / \partial O_k < 0$ holds.

Using this necessary conditions which the gate $\#k$ to be fault, we presents the diagnosis algorithm for combinatorial logic circuit. This procedure executes the concluding part of fault diagnosis, that is, locating and determining the fault type of the gate which might have different type from the sample NN circuit according to the analysis of sensitivity factors $\partial E / \partial O_j$'s given by the Backward. This analysis is composed of two-staged rating procedure and generates the candidate fault lists as follows.

[Primary rating]

To select candidate fault gates, this stage estimates the number of cases that satisfy the Condition 2, and that of cases do not satisfy the Condition 2, as follows (See Figure 4).

(1) generating a series of test input patterns randomly,
(2) feeding them to both TBox circuit and OBox circuit,
(3) extracting the unmatched cases between two circuits with respect to the corresponding primary output signals, then estimates sensitivity factors for all gates by back-tracking

only once from the unmatched outputs. These sensitivity data are summed up and analyzed for several unmatched patterns and then, the candidates of the defective gate and its defective gate type are extracted.

[Secondary rating]

According to the result of the primary rating, from the top to the bottom, the candidate fault gate $\#k$ of the OBox is temporary replaced by the NN gate of the type $(t_{i,j})$ and the error E is calculated between the original OBox and this modified OBox by repeating Forward to the latter one. Finally, from these errors, the fault gate can be detected. That is, for each candidate gate in primary rating, the steps take continuously as follows.

(4) changing the original gate type to the expected defect type while other gates are kept unchanged,

(5) feeding again unmatched test patterns to both TBox circuit and OBox circuit, the possible existence of inconsistency for primary signal outputs between two circuits are checked. If there is no inconsistency, the candidate (defective gate and its defective gate type) is recognized as a true candidate. This is because, if you have only one true candidate, then, the candidate is judged as the defective gate, or else, you can say either one of them might be the defective one.

4. Fault model detectable by NN fault diagnosis

The NN fault diagnosis proposed by the authors restricts to only the gate faults as a range, that is, it deals with the fault models that each of which comes from the defects observed as if a defective gate were replaced by another gate with different logic function [2]. This fault model detectable by our NN fault diagnosis, called the SSAG (Synthetic stuck-at-gate) fault model, consists of two classes. One is called the static SSAG fault model, which has never a logic fault associated with timing-related defects. The other is called the dynamic SSAG fault model, which has necessarily a logic fault associated with timing-related defects.

There are several kinds of fault models such as SSA (single stuck-at) fault model, bridging fault model, open fault model and SO (stuck-open) fault model [1]. On the contrary, the static SSAG model can cover these models uniformly. That is, the NN fault diagnosis method does not dare to specify the defect whether it is caused by short or by open, or furthermore, by compound of these. Conversely, it covers testing of a wide range of defects. Here, the delay fault associated with timing-related defects is divided into two different classes called the gate delay fault and the path delay fault. The gate delay fault is a special case of path delay fault and is detected at the system output as the delay fault (namely, it is included in dynamic SSAG fault model). In this section, based on the dynamic SSAG fault model, we construct an algorithm detecting the defect induced by gate delay fault.

4.1 Interpretation of dynamic SSAG fault model

It is assumed that all the static faults are already removed by the NN diagnosis method [2] and do not exist in the target circuit. That is, at the stationary state, the target circuit has not a logical dependency on its history of primary input patterns and its output pattern is unique and correct for the primary input pattern. Furthermore, it is assumed that the circuit has neither SA0 fault, nor SA1 fault, nor SO fault. The delay fault is one of the dynamic faults related to the transient state induced by an input change. That is, if a delay fault appears in a transient state, the circuit has a logical fault associated with timing-related defects. The delay testing involves a series of time-dependent input test patterns instead of the single input pattern. That is, at least two patterns are to be considered. The one is the input pattern relating directly to the fault and the other is its preceding one. The path delay testing needs the analysis of a propagation path for the primary input signal. On the contrary, as to the gate delay fault, we have only to take into consideration the propagation of the delay from the gate.

If the notation $(s_0; t\ s_1)$ (where, $s_0, t, s_1 \in \{0,1\}$) means that the output takes s_0 in stationary state for the 1-st input, t in a transient state after feeding the 2-nd input, and s_1 in stationary state after a while, the observation at the system output terminal be represented as follows.

Type 1: (0;0 0), (0;1 1), (1;0 0), (1;1 1)

Type 2: (0;0 1), (1;1 0)

Type 3: (0;1 0), (1;0 1)

Here, except for the Type 1, the remaining types are corresponding to delay faults. That is, the Type 2 represents a pure delay, and the Type 3 represents a static 0-hazard and a static 1-hazard, respectively. Thus, if the last 2 bits of the observation represents either 0 1 or 1 0, the target gate is regard as delay fault.

4.2 Dynamic NN diagnosis for gate delay test

According to the facts shown in the item 4.1, the dynamic NN diagnosis method is constructed to find a gate to which a gate delay fault is attributed by modifying the NN diagnosis method (say, static NN diagnosis method) which works efficiently for static gate faults. In the end of the item 4.1, it is shown that the difference between the transient- and stationary response for the 2-nd input pattern means the existence of a gate delay fault. This is reasonably translated to the static fault where the circuit has an illegal primary signal output corresponding to the transient response different from the correct stationary output for the 2-nd input pattern. Therefore, applying a static NN method to this test according to the SSAG model, we have one or more defective candidate gates of defective gate type (Primary rating). The defective gate type data for the candidates are used only as the reference information and ignored in the next Secondary rating. That is, for each candidate, the following evaluation is given in this stage. First of all, in the sample circuit, the NN gate logic of the candidate gate is temporarily changed such that the fan-out for the second fan-in signals takes that for the first fan-in signals when a pair of primary input signal patterns is fed consecutively to the circuit. Then, for each unmatched input patterns recorded in the Primary rating stage, the transient output patterns are compared between the test circuit and the corresponding sample NN circuit. If there is no inconsistency, the candidate (defective gate and its defective gate type) is recognized as a true candidate. This is because, if you have only one true candidate, then, the

candidate is judged as the defective gate, or else, you can say either one of them might be the defective one (Secondary rating). This is the principle of the dynamic NN diagnosis method that is constructed by slightly modifying the static NN method.

5. Numerical experiments

The proposed testing method is verified by numerical simulations according to the NN fault diagnosis algorithm described in the previous section. In order to do this, in place of using a real circuit for the test circuit, we take an equivalent NN circuit to it. A test circuit is defined as defective if either one of its gates works as a different type as listed in Table 1, from the sample circuit. This definition is adopted because it can include any kind of gate faults such as gate placement error and gate delay error. Then, the NN diagnosis method can operate as a simulation program on computer and perform the numerical experiments. The simulation algorithm of NN diagnosis method is formulated in the C++ language, and it was implemented on a platform of the Macintosh G3. We take the combinatorial logic circuit shown Figure 5 as a test case of numerical experiment, and the simulation results are shown for the 5 cases in Table 2.

The Case 1 ~ Case 3 in Table 2 represent the simulation results for the static NN diagnosis. For example, the first case (Case 1) means the type G2 is placed at the Gate #1 instead of the type G1(NOR) as a fault. The Secondary rating concluded the Gate #1 with the type G0 is the most probable, because it has full mark (in other words, it has the error 0). In this case, for any fan-out at the Gate #1, notice that the type G0 is equivalent to the type G2. So, each of these (namely, G0 and G2) has the error 0. That is, though the static NN diagnostic method did not catch the pre-set fault exactly, it did take out correctly possible candidates equivalent to the given test circuit. This was also confirmed by the simulation results in Case 2 and Case 3.

The Case 4 and Case 5 in Table 2 represent the simulation results for the dynamic NN diagnosis, where the gate delay defect is set to the Gate #1 and the Gate #2 respectively as an assumed delay fault. The Case 4 shows that the Secondary rating concludes the Gate #1 is the most probable in delay fault. In this case, the diagnosis method did catch the pre-set delay fault exactly. The Case 5 shows that the Secondary rating concludes 6 gates (that is, the Gate #2, #9, #19, #23, #29 and #30) are the most probable in delay fault, where the assumed delay fault Gate #2 is included in these results. In this case, the diagnosis method did catch the delay path caused by the pre-set delay fault Gate #2 such as #2 → #9 (→ #14) → #19 → #23 → #29 → #30.

As shown by the simulations, our NN diagnosis method works well according to its designed functions just as expected.

Table 1. The all types of gate

Input		Gate type							
I1	I2	G0	G1	G2	G3		G13	G14	G15
0	0	0	1	0	1		1	0	1
1	0	0	0	1	1		0	1	1
0	1	0	0	0	0	...	1	1	1
1	1	0	0	0	0		1	1	1
Gate name		s-a-0	NOR					OR	s-a-1

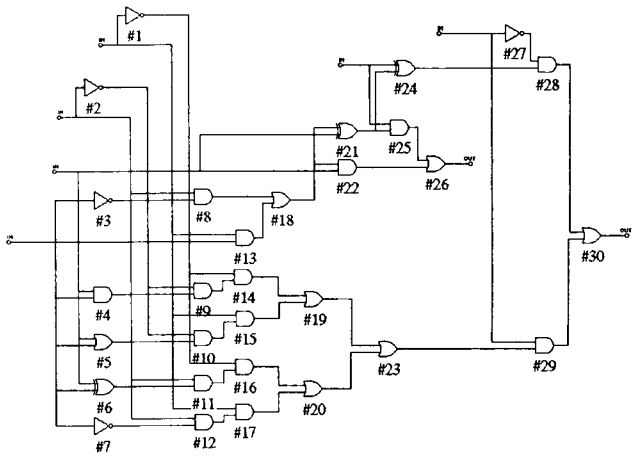


Figure 5. Test circuit

Table 2. Simulation results

Rating	[Case 1] Gate #1 : G1(NOR) --> G2									
Primary	#1	#2	#4	#5	#6	#7	#9	#10	#11	#12
Type	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0
Secondary	100	-29	18	-41	42	-29	18	-41	42	-29
Primary	#14	#16	#19	#20	#23	#24	#29	#30		
Type	G0	G0	G12	G12	G8	G0	G0	G10		
Secondary	18	42	18	42	-29	-91	-29	-29		

Rating	[Case 2] Gate #18 : G14(OR) --> G4									
Primary	#3	#8	#18	#21	#22	#24	#25	#26	#28	#30
Type	G0	G0	G12	G4	G0	G9	G0	G8	G2	G13
Secondary	100	100	100	0	-22	-41	-22	-41	-41	-73

Rating	[Case 3] Gate #29 : G8(AND) --> G6(EXOR)									
Primary	#29	#30								
Type	G6	G11								
Secondary	100	55								

Rating	[Case 4] Setting delay : Gate #1									
Primary	#1	#23	#29	#30						
Type	G8	G9	G2	G11						
Secondary	100	29	29	29						

Rating	[Case 5] Setting delay : Gate #2									
Primary	#2	#9	#10	#14	#19	#23	#29	#30		
Type	G8	G4	G4	G8	G13	G13	G2	G11		
Secondary	100	100	0	29	100	100	100	100		

6. Conclusions

In this paper, new fault model called SSAG is proposed, and is applied to any test for gate defect. According to the SSAG model, by extending the existing NN method (static NN diagnosis method), another new NN diagnosis method (dynamic NN diagnosis model) is developed for testing defective circuits with delay faults. Its advantageous features are proved by numerical experiments.

Integration of the two methods, that is, constructing the integrated SSAG-NN method based on the static NN method and the dynamic one is one of the future works.

References

[1] Aitkin, R. C., Nanometer technology effects on fault models for IC testing, IEEE Computer Magazine, pp.46-51, Nov. 1999.

[2] Tatsumi, H., Murai, Y., Tokumasu, S., Logic circuit diagnosis by using neural networks, 31st IEEE Int. Symp. on Multiple-Valued Logic, pp.345-350, May 2001.