A Scheduling Problem in Test Generation

Tomoo Inoue, Hironori Maeda and Hideo Fujiwara

Graduate School of Information Science
Nara Institute of Science and Technology

Abstract The order of faults which are targeted for test-pattern generation affects both of the processing time for test generation and the number of test-patterns. This order is referred to as a *test generation schedule*. In this paper, we consider the test generation scheduling problem which minimizes the cost of testing. We analyze the effect of scheduling based on test-pattern generation time and *dominating probability*. Then, we present experimental results on the ISCAS'85 benchmark circuits.

Topic: Automatic Test Generation

Correspondence to:

Tomoo Inoue
Graduate school of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma
Nara 630-01, Japan
Phone: +81-7437-2-5222
Fax: +81-7437-2-5229
E-mail: inoue@is.aist-nara.ac.jp

1. Introduction

The cost of testing for logic circuits consists mainly of the cost of test generation and the cost of test application. The cost of test generation means the processing time to generate test-patterns for a given circuit. Several efficient test generation algorithms such as PODEM [1], FAN [2] and SOCRATES [3,4] are reported for combinational circuits. On the other hand, small test set is important for the reduction of the cost of test application. By applying test compaction algorithms as reported in [5-8] to test generation, small test sets can be obtained. However, since test compaction approaches require extra efforts such as deriving maximal independent fault sets, the total cost of testing is not always reduced.

Many test generation algorithms consist of two processes; test-pattern generation and fault simulation. In the test-pattern generation process, a fault is selected from a *fault list*, referred to as a *target fault*, and a test-pattern is generated for the fault. Then all the detectable faults by the test-pattern are identified in fault simulation process. These two processes are repeated until all detectable faults are identified. Here, the order of target faults selected from the fault list is referred to as the *test generation schedule*. The test generation schedule affects both the processing time for test generation and the test set size (the number of test-patterns), and accordingly there exists an optimal schedule which minimizes the test generation time and/or the test set size.

In this paper, we consider this scheduling problem in test generation for combinational logic circuits. First, we formulate the scheduling problem, and propose a relation called *fault dominance* to estimate the test-pattern generation time and the number of generated test-patterns. Then, we analyze the effect of scheduling based on test-pattern generation time and *dominating probability*. Finally, we present experimental results on the ISCAS'85 [10] benchmark circuits.

2. Formulation of the Scheduling Problem

The flow of our test generation process is illustrated in Figure 1. Let F be a set of faults of a given combinational circuit. Let A be a test-pattern generation algorithm in this process.

First, *Test Generation Scheduler* makes a test generation schedule S for fault set F, i.e., determines an order of target faults in fault set F to be test-generated. *Test-Pattern Generator* generates a completely-specified test-pattern for the target fault selected according to schedule S, by algorithm A. *Fault Simulator* identifies all the faults that are detected by the test-pattern. Test-pattern generation and fault simulation are both repeated until all detectable faults in F are identified. In this process the number of test-patterns and the processing time for test generation depend on schedule S as well as on test-pattern generation algorithm A. Hence, we can consider two optimal scheduling problems; one is to minimize the test generation time, and the other is to minimize the number of test-patterns for fault set F with algorithm A.

However, to simplify the analysis, we focus only on the processing time for test-pattern generation (denoted by *TPG time*) and the number of generated test-patterns hereafter. Let $T_A(S)$ be the total TPG time by schedule S with algorithm A. Let $L_A(S)$ be the total number of test-patterns by schedule S with algorithm A. One of the optimal scheduling problems is to find an optimal scheduling S_{Topt} which minimizes the total TPG time for fault set F with algorithm A:

$$T_A(S_{Topt}) = \min_S \{T_A(S)\}, \qquad (1)$$

and the other is to find an optimal scheduling S_{Lopt} which minimizes the total number of test-patterns for fault set F with algorithm A:

$$L_A(S_{Lopt}) = \min_S \{L_A(S)\}. \qquad (2)$$

Note that test-patterns generated by algorithm A are completely-specified, i.e., generated test-patterns by algorithm A include no don't-care value. Hence, we can define *fault dominance* as follows.

Fault Dominance: If the test-pattern for a fault f_i generated under an algorithm A detects another fault f_j, then fault f_i *dominates* fault f_j under algorithm A.

Unless otherwise noted, from now on we will omit the notation of algorithm A for simplicity.

Next we shall consider the probability that a test-pattern for a fault is generated. Let N be the total number of faults. Suppose that a test generation schedule $S = \langle f_1, f_2, \ldots, f_N \rangle$. Let d_{ij} be the probability that fault f_i dominates fault f_j. Let g_i be the probability that a test-pattern for the i-th fault f_i in schedule S is generated. Then, the probability that a test-pattern for the first fault f_1 is 1, i.e., $g_1 = 1$. The second fault f_2 is dominated by f_1 in probability d_{12}. The probability that a test-pattern for f_2 is generated is the probability that f_2 is not dominated by f_1. Hence, $g_2 = (1 - d_{12})$. Similarly, f_1 and f_2 dominate the third fault f_3 in probability d_{13} and in probability d_{23}, respectively. The probability that a test-pattern for fault f_3 is generated is the probability that f_3 is dominated neither by f_1 nor by f_2 for which a test-pattern is generated in probability g_2. Hence $g_3 = (1 - d_{13})\,(1 - g_2 d_{23})$. In this way, we can express the probability that a test-pattern for the i-th fault is generated can be expressed as

$$\begin{cases} g_1 = 1 \\ g_i = \displaystyle\prod_{k=1}^{i-1} \left(1 - g_k\, d_{ki}\right) \qquad (2 \le i \le N) \end{cases} \tag{3}$$

Note that the probability d_{ij} that fault f_i dominates fault f_j depends on the dominated fault f_j, i.e., $d_{ij} \neq d_{ik}$ if $j \neq k$ in general. However, here we consider the case that the probability is independent of any dominated fault, i.e., $d_{ij} = d_i$ for all j, and define the *dominating probability* as follows.

Dominating probability: Let the dominating probability d_i of a fault f_i be the average of the probabilities d_{ij} for all j, i.e.,

$$d_i = \frac{1}{N} \sum_{j=1}^{N} d_{ij} \quad . \tag{4}$$

By substituting d_k for d_{kj} in Equation (3), we have

$$\begin{cases} g_1 = 1 \\ g_i = \prod_{k=1}^{i-1} \left(1 - g_k d_k\right) \qquad (2 \le i \le N) \end{cases} \tag{5}$$

This equation can be expressed as

$$\begin{cases} g_1 = 1 \\ g_i = g_{i-1} \left(1 - g_{i-1} d_{i-1}\right) \qquad (2 \le i \le N) \end{cases} \tag{6}$$

Note that if $0 < d_{ij} \le 1$, then

$$g_i > g_{i+1} \quad \text{for } 1 \le \text{i} \le N - 1 \; . \tag{7}$$

The total number of test-patterns obtained by schedule S is

$$L(S) = \sum_{S} g_i \quad . \tag{8}$$

and the total TPG time by schedule S is

$$T(S) = \sum_{S} t_i g_i \quad . \tag{9}$$

Therefore, if we can predict fault characteristics such as dominating probability and TPG time for each fault *a priori*, we can estimate both of the total number of test-patterns and the total TPG time obtained by test generation scheduling based on these characteristics. In the next section, we shall analyze the effect of the test generation scheduling based on dominating probability and TPG time provided that these two characteristics are given *a priori*.

3. Analysis of the Effect of Scheduling

3.1 Scheduling Based on TPG Time

First, we consider a scheduling based on the TPG time for each fault. Here we assume that dominating probabilities for all faults f_i are equal, i.e., $d_i = d$. Then, from Equations (5) and (6), the probability g_i that a test-pattern for the i-th fault f_i is generated can be expressed as

$$\begin{cases} g_1 = 1 \\ g_i = \prod_{k=1}^{i-1} \left(1 - g_k d\right) \\ \quad = g_{i-1}\left(1 - g_{i-1} d\right) \qquad (2 \le i \le N) \end{cases} \tag{10}$$

Let S_h be a schedule or an order of faults. Let t_i be the TPG time for the i-th fault in order S_h. Suppose an arbitrary pair of adjacent faults (f_n, f_{n+1}) in order S_h such that $t_n > t_{n+1}$. Note that $1 \le n \le N - 1$. Let $t_h = t_n$ and $t_e = t_{n+1}$. Let g_i be the probability that a test-pattern for the i-th fault in order S_h is generated. From Equation (8) the total number of test-patterns obtained by schedule S_h is given by

$$L(S_h) = \sum_{S_h} g_i \tag{11}$$

From Equation (9) the total TPG time is given by

$$\begin{aligned} T(S_h) &= \sum_{S_h} t_i g_i \\ &= \sum_{i=1}^{n-1} t_i g_i + t_h g_n + t_e g_{n+1} + \sum_{i=n+2}^{N} t_i g_i \end{aligned} \tag{12}$$

Let S_e be the order obtained by exchanging the n-th and the $(n+1)$-th faults in order S_h. Let t'_i be the TPG time for the i-th fault in order S_e. That is,

$$\begin{cases} t'_n = t_e \\ t'_{n+1} = t_h \\ t'_i = t_i \qquad (i \ne n, n+1) \end{cases} \tag{13}$$

Let g'_i be the probability that a test-pattern for the i-th fault in order S_e is generated. From

Equation (8) the total number of test-patterns obtained by schedule S_e is given by

$$L(S_e) = \sum_{S_e} g_i' \quad . \tag{14}$$

From Equation (9) the total TPG time is given by

$$T(S_e) = \sum_{S_e} t_i' g_i'$$

$$= \sum_{i=1}^{n-1} t_i' g_i' + t_e g_n' + t_h g_{n+1}' + \sum_{i=n+2}^{N} t_i' g_i' \quad . \tag{15}$$

Since $d_i = d$ for all i , $g_i = g'_i$ for all i. Hence,

$$L(S_e) = L(S_h) \, . \tag{16}$$

From this equation we can easily see that any scheduling based on TPG time derives the same number of test-patterns provided that dominating probabilities for all faults are equal.

From Equation (13), Equation (15) can be also expressed as

$$T(S_e) = \sum_{i=1}^{n-1} t_i g_i + t_e g_n + t_h g_{n+1} + \sum_{i=n+2}^{N} t_i g_i \quad . \tag{17}$$

Hence, the difference between these two total TPG time can be expressed as

$$T(S_h) - T(S_e) = (t_h - t_e)(g_n - g_{n+1}) \, . \tag{18}$$

Since $t_h > t_e$ and $g_n > g_{n+1}$ (from Inequality (7)),

$$T(S_h) - T(S_e) > 0 \, . \tag{19}$$

This inequality means that when the TPG time for a fault is larger than that for the next fault in an order or a test generation schedule, the total TPG time can be reduced by exchanging these two faults. Hence, the ascending order of TPG time can be obtained by repeating this exchange until no exchange can be applied, and this order minimizes TPG time. Therefore, we have the following theorem.

Theorem 1: The scheduling according to the ascending order of TPG time minimizes the total TPG time provided that dominating probabilities for all faults are equal.

3.2 Scheduling Based on Dominating Probability

Next, we consider a scheduling based on dominating probability of each fault. Here we assume that TPG times for all faults f_i are equal, i.e., $t_i = t$. Then, from Equations (8) and (9) the total TPG time by a schedule S can be expressed as

$$\begin{aligned} T(S) &= \sum_{S} t g_i \\ &= t\, L(S) \end{aligned} \tag{20}$$

Let S_f be an order of faults. Let d_i be the dominating probability of the i-th fault in order S_f. Suppose an arbitrary pair of adjacent faults (f_n, f_{n+1}) in order S_f such that $d_n < d_{n+1}$. Note that $1 \le n \le N-1$. Let $d_f = d_n$ and $d_m = d_{n+1}$. Let g_i be the probability that a test-pattern for the i-th fault in order S_f is generated. From Equation (6) we have

$$g_{n+1} = g_n \left(1 - g_n d_f\right), \tag{21}$$

and

$$\begin{aligned} g_{n+2} &= g_{n+1} \left(1 - g_{n+1} d_m\right) \\ &= g_n \left(1 - g_n \left(d_f + d_m - 2\, g_n d_f d_m\right)\right) - g_n^4 d_f^2 d_m . \end{aligned} \tag{22}$$

Thus the total number of test-patterns generated in order S_f can be expressed from Equation (8) as

$$\begin{aligned} L(S_f) &= \sum_{S_f} g_i \\ &= \sum_{i=1}^{n} g_i + g_{n+1} + g_{n+2} + \sum_{i=n+3}^{N} g_i . \end{aligned} \tag{23}$$

Let S_m be the order obtained by exchanging the n-th and the $(n+1)$-th faults in order S_f. Let d'_i be the dominating probability of the i-th fault in order S_m. That is,

$$\begin{cases} d'_n = d_m \\ d'_{n+1} = d_f \\ d'_i = d_i & (i \neq n,\, n+1) \end{cases} \tag{24}$$

Let g'_i be the probability that a test-pattern for the i-th fault in order S_m is generated. From Equation (6) we have

$$g'_{n+1} = g'_n \left(1 - g'_n\, d_m\right), \tag{25}$$

and

$$\begin{aligned} g'_{n+2} &= g'_{n+1} \left(1 - g'_{n+1}\, d'_f\right) \\ &= g'_n \left(1 - g'_n \left(d_f + d_m - 2\, g'_n\, d_f d_m\right)\right) - g'^{\,4}_n\, d_f d_m^2 . \end{aligned} \tag{26}$$

Thus the total number of test-patterns generated in order S_m can be expressed from Equation (8) as

$$\begin{aligned} L(S_m) &= \sum_{S_m} g'_i \\ &= \sum_{i=1}^{n} g'_i + g'_{n+1} + g'_{n+2} + \sum_{i=n+3}^{N} g'_i . \end{aligned} \tag{27}$$

From Equations (23) and (27) the difference between $L(S_f)$ and $L(S_m)$ can be expressed as

$$\begin{aligned} &L(S_f) - L(S_m) \\ &= \sum_{i=1}^{n} \left(g_i - g'_i\right) + \left(g_{n+1} - g'_{n+1}\right) + \left(g_{n+2} - g'_{n+2}\right) + \sum_{i=n+3}^{N} \left(g_i - g'_i\right) . \end{aligned} \tag{28}$$

Since $d_i = d'_i$ for $i \leq n - 1$ (Equation (24)), from Equation (6) we have

$$g_i = g'_i \qquad \text{for } i \leq n . \tag{29}$$

Hence,

$$\sum_{i=1}^{n} \left(g_i - g'_i\right) = 0 . \tag{30}$$

From Equations (21), (25) and (29) and $d_f < d_m$, we have

$$g_{n+1} - g'_{n+1} = g_n^2 \left(d_m - d_f\right)$$

$$> 0 \,. \tag{31}$$

In the same way as the above inequality we have

$$g_{n+2} - g'_{n+2} = g_n^4 \, d_f \, d_m \left(d_m - d_f\right)$$

$$> 0 \,. \tag{32}$$

On the other hand, the difference between the probability that a test-pattern for the i-th fault is generated in order S_f and that in order S_m can be expressed from Equation (6) as

$$g_i - g'_i = \left(g_{i-1} - g'_{i-1}\right)\left(1 - \left(g_{i-1} + g'_{i-1}\right) d_{i-1}\right) . \tag{33}$$

If we assume that

$$d_i < \frac{1}{2} \qquad \text{for all } i, \tag{34}$$

which is reasonable, then we have

$$g_{i-1} + g'_{i-1} < 1 \tag{35}$$

in Equation (33). Hence, from Equation (33) and Inequalities (32) and (35), we have

$$g_i - g'_i > 0 \qquad \text{for } n + 3 \leq i \leq N. \tag{36}$$

Therefore,

$$\sum_{i = n+3}^{N} \left(g_i - g'_i\right) > 0 \,. \tag{37}$$

Thus we have

$$L(S_f) - L(S_m) > 0 , \tag{38}$$

and accordingly

$$T(S_f) - T(S_m) > 0 . \tag{39}$$

Inequalities (38) and (39) mean that when the dominating probability of a fault is smaller than that of the next fault in an order, the total TPG time as well as the total number of test-patterns can be reduced by exchanging these two faults. Hence, the descending order of dominating probability can be obtained by repeating this exchange until no exchange can be applied, and this order minimizes both of the total TPG time and the total number of test-patterns. Therefore, we have the following theorem.

Theorem 2: The scheduling according to the descending order of dominating probability minimizes both of the total TPG time and the total number of test-patterns provided that TPG times for all faults are equal.

3.3 Scheduling Based on TPG Time and Dominating Probability

Until now, we considered a scheduling based on either TPG time or dominating probability. Here we consider a scheduling based on both of TPG time and dominating probability.

Let S_a be an order of faults. Let d_i be the dominating probability of the i-th fault in order S_a. Let t_i be the TPG time for the i-th fault in order S_a. Suppose an arbitrary pair of adjacent faults (f_n, f_{n+1}) in order S_a such that $d_n > d_{n+1}$. Note that $1 \le n \le N - 1$. Let g_i be the probability that a test-pattern for the i-th fault in order S_a is generated.

Let S_b be the order obtained by exchanging the n-th and the (n+1)-th faults in order S_a. Let d'_i be the dominating probability of the i-th fault in order S_b. Let t'_i be the TPG time for the i-th fault in order S_b. That is,

$$\begin{cases} d'_n = d_{n+1}, \ t'_n = t_{n+1} \\ d'_{n+1} = d_n, \ t'_{n+1} = t_n \\ d'_i = d_i, \ t'_i = t_i \qquad (i \ne n, n+1) \end{cases} \tag{40}$$

Let g'_i be the probability that a test-pattern for the i-th fault in order S_b is generated.

First, let us consider the total number of test-patterns. From Equation (8) we have

$$L(S_a) = \sum_{i=1}^{N} g_i \tag{41}$$

and

$$L(S_b) = \sum_{i=1}^{N} g_i' \quad . \tag{42}$$

In the same way as the analysis in the previous subsection 3.2, we have

$$g_i = g'_i \qquad \text{for } i \le n \tag{43}$$

and

$$g_i < g'_i \qquad \text{for } n \le i \le N. \tag{44}$$

Hence, the difference between the total number of test-patterns by order S_a and that by order S_b is expressed as

$$L(S_a) - L(S_b) < 0 \quad . \tag{45}$$

In this way, the total number of test-patterns is derived only from the dominating probabilities of all faults independently of TPG time for any fault. Therefore, we have the following theorem.

Theorem 3: The scheduling according to the descending order of dominating probability minimizes the total number of test-patterns.

Next, let us consider the total TPG time. From Equation (9) we have

$$\begin{aligned} T(S_a) &= \sum_{i=1}^{N} t_i g_i \\ &= \sum_{i=1}^{n-1} t_i g_i + t_n g_n + t_{n+1} g_{n+1} + \sum_{i=n+2}^{N} t_i g_i \quad , \end{aligned} \tag{46}$$

and

$$T(S_b) = \sum_{i=1}^{N} t_i' g_i'$$

$$= \sum_{i=1}^{n-1} t_i' g_i' + t_n' g_n' + t_{n+1}' g_{n+1}' + \sum_{i=n+2}^{N} t_i' g_i' . \quad (47)$$

From Equation (6) we have

$$g_{n+1} = g_n \left(1 - g_n\, d_n\right) \quad (48)$$

and

$$g_{n+1}' = g_n' \left(1 - g_n'\, d_n'\right) . \quad (49)$$

From Equations (40) and (43), Equation (49) can be expressed as

$$g_{n+1}' = g_n \left(1 - g_n\, d_{n+1}\right) . \quad (50)$$

Thus the difference between the total TPG time by order S_a and that by order S_b is

$$T(S_a) - T(S_b) = g_n^2 \left(t_n\, d_{n+1} - t_{n+1}\, d_n\right) + \sum_{i=n+2}^{N} t_i \left(g_i - g_i'\right) . \quad (51)$$

From Equation (44) we have

$$\sum_{i=n+2}^{N} t_i \left(g_i - g_i'\right) < 0 . \quad (52)$$

Hence, if

$$\frac{t_n}{t_{n+1}} \leq \frac{d_n}{d_{n+1}} , \quad (53)$$

then

$$T(S_a) - T(S_b) < 0 . \quad (54)$$

Therefore, we have the following theorem.

Theorem 4: The scheduling according to the descending order of dominating probability minimizes the total TPG time provided that

$$\frac{t_i}{t_j} \leq \frac{d_i}{d_j} \quad (55)$$

for any pair of faults (f_i, f_j) such that

$$d_i > d_j \text{ and } t_i > t_j . \tag{56}$$

As shown by Theorem 4, the total TPG time is not always minimized by the scheduling according to the descending order of dominating probability. However, since $d_n > d_{n+1}$ in Equation (51), if

$$t_n \le t_{n+1}, \tag{57}$$

then

$$T(S_a) - T(S_b) < 0 . \tag{58}$$

Further, as shown by Theorem 3, the total number of test-patterns is always minimized by the scheduling according to the descending order of dominating probability. Hence, we can consider that the scheduling according to the descending order of dominating probability prior to the ascending order of TPG time for each fault would be effective in reducing both of the total TPG time and the total number of test-patterns.

4. Experimental Results

We made experiments of test generation scheduling based on TPG time and dominating probability using the ISCAS'85 benchmark circuits [9] on a DECstation 5000/25. The FAN algorithm [2] was used as a test-pattern generator.

In order to obtain accurate fault characteristics for test generation such as TPG time and dominating probability, we computed the CPU time of test-pattern generation for each fault and the number of faults dominated by each fault on the benchmark circuits by using FAN. Note that the number of faults dominated by a fault over the total number of faults denotes the dominating probability of the fault in the above-mentioned analysis. Table 1 shows the computational results for the benchmark circuits.

Based on the above computed characteristics of faults, the following test generation schedules were implemented.

(*EF*) Ascending order of CPU time, called an *Easy Fault first* scheduling.

(*HF*) Descending order of CPU time, called a *Hard Fault first* scheduling.

(*DM*) Descending order of the number of dominated faults, called a *Dominating-Many fault first* scheduling.

(*DF*) Ascending order of the number of dominated faults, called a *Dominating-Few fault first* scheduling.

(*DE*) *DM* prior to *EF*: *DM* is applied first. For the faults that tie in *DM*, i.e., dominate the same number of faults, *EF* is applied.

(*ED*) *EF* prior to *DM*: EF is applied first. For the faults that tie in *EF*, i.e., require the same CPU time, *DM* is applied.

Figure 2 shows the total processing time (including fault simulation time) and the number of generated test-patterns for each scheduling method. From the results of the *EF* and *HF* schedulings, we can see that the total processing time by the *EF* scheduling is smaller than that by the *HF* scheduling for c3670, c3540, c6288 and c7552. These results coincide with our analytical results. As shown in Table 1, the standard deviation of TPG time for c7552 is large, i.e., there exist many hard-to-test faults and many easy-to-test faults in c7552. The *EF* scheduling seems to be effective particularly for such circuits. On the other hand, we can see that the total number of test-patterns by the *EF* scheduling is larger than that by the *HF* scheduling for all circuits except c6288. This is because dominating probability might not be independent of TPG time, e.g., an easy fault would dominate few faults.

From the results of the *DM* and *DF* schedulings, we can see that both the total processing time and the number of test-patterns by the *DM* scheduling are smaller than those by the *DF* scheduling for all circuits except c6288. These results coincide closely with the analytical results.

By comparing the results of the *DE* and *ED* schedulings, we can see that the total number of test-patterns by the *DE* scheduling is smaller than that by the *ED* scheduling for all circuits except c6288, and that the CPU time by the *DE* scheduling is also smaller than that by the *ED* scheduling for most circuits. These results coincide with the analytical result that the scheduling based on the descending order of dominating probability prior to the ascending order of TPG time is effective in reducing the total TPG time and the total number of test-patterns.

The experimental result for c6288 is exceptional. From Figure 2(g) we can see that by the *DM* scheduling, the total CPU time can be reduced, but the total number of test-patterns increases. This result implies that dominating-many faults are easy-to-test, and faults dominated by dominating-many faults are overlapping one another. On the other hand, from Figure 2(g) we can see that by the *DF* scheduling, the total number of test-patterns can be reduced, but the total CPU time increases. This result implies that dominating-few faults are hard-to-test, and faults dominated by dominating-few faults are different from one another. In this way, the scheduling based only on the dominating probability is not effective for such circuits as c6288 which has dominated faults overlap one another.

In the previous section, we considered the dominating probability, which is the probability that a faults dominates other faults. On the other hand, we can also consider the *dominated probability* which refers to the probability that a fault is dominated by the other faults. Note that the dominated probability of a fault f_i is given by the number of faults that dominate fault f_i over the total number of faults. We computed the number of faults that dominate a fault for each fault by using FAN. Table 2 shows the computational results for the benchmark circuits. In Table 2, the number of faults that dominate a fault is denoted by the number of dominating faults. From this table we can see that the standard deviation of the number of dominating faults is large as compared with that of the number of dominated faults shown in Table 1 for all circuits. Thus, we can expect that the scheduling based on the dominated probability would be more effective than that based on the dominating probability.

Figure 3 shows the experimental results of the scheduling based on the number of dominating faults. In this figure, *DBF* means the scheduling according to the ascending order of the number of dominating faults, which is called the *Dominated-By-Few fault first* scheduling, and *DBM* means the scheduling according to the descending order of the number of dominating faults, which is called the *Dominated-By-Many fault first* scheduling. From Figure

3 we can see that both the total CPU time and the total number of test-patterns by the *DBF* scheduling are smaller than those by the *DBM* scheduling for all circuits without exception. Further, from Figures 3 and 4, we can see that the *DBF* scheduling is the most effective to reduce both of the total CPU time and the total number of test-patterns simultaneously for all circuits of all the schedulings.

5. Conclusions and Future Work

In this paper, we have considered a scheduling problem in test generation for combinational logic circuits. We have formulated the scheduling problem to minimize the cost of testing. Then, we have analyzed the effect of the scheduling based on *dominating probability* and *test-pattern generation time* for each fault. The analytical results show that the scheduling based on the descending order of dominating probability and the ascending order of test-pattern generation time reduces both of the total test-pattern generation time and the total number of test-patterns. Furthermore, we have presented experimental results on ISCAS'85 benchmark circuits. Experimental results coincide with the analysis for most of the benchmark circuits. We have also introduced *dominated probability*. Experimental results on dominated probability show that the scheduling according to the ascending order of dominated probability is the most effective of all the schedulings.

In this paper, we have considered schedulings based on the measures such as test-pattern generation time, dominating probability and dominated probability, where we have assumed that these measures are known *a priori*. The next problem is how to obtain these measures without consuming much time before test-pattern generation. The analysis of the effect of the scheduling based on dominated probability is also one of the remaining problems. We are also working on test generation scheduling for sequential circuits. We can expect that the scheduling for sequential circuits would be more effective than that for combinational circuits.

References

[1] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. Comput.*, vol. C-30, No.3, pp. 215-222, Mar. 1981.

[2] H. Fujiwara and T. Shimono, "On the acceleration of test pattern generation algorithms," *IEEE Trans. Comput.*, vol. C-32, No.12, pp. 1137-1144, Dec. 1983.

[3] M. H. Schulz, E. Trischler and T. M. Sarfert, "SOCRATES: A highly efficient automatic test pattern generation system," *IEEE Trans. Computer-Aided Design*, vol. 7, No. 1, pp. 126-137, Jan. 1988.

[4] M. H. Schulz and E. Auth, "Improved deterministic test pattern generation with applications to redundancy identification," *IEEE Trans. Computer-Aided Design*, vol. 8, No. 7, pp. 811-816, July 1989.

[5] S. B. Akers and C. Joseph, "On the role of independent fault sets in the generation of minimal test sets," *Proc. Int. Test Conf.*, pp. 1100-1107, 1987.

[6] I. Pomeranz, L. N. Reddy and S. M. Reddy, "COMPACTEST: A method to generate compact test sets for combinational circuits," *Proc. Int. Test Conf.*, pp. 194-203, 1991.

[7] G. J. Tromp, "Minimal test sets for combinational circuits," *Proc. Int. Test Conf.*, pp. 204-209, 1991.

[8] S. Kajihara, I. Pomeranz, K. Kinoshita, and S.M. Reddy, "Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits," *Proc. 30th ACM/IEEE Design Automation Conf.*, pp. 102-106, 1993.

[9] F. Brglez and H. Fujiwara, "A neutral netlist of ten combinational benchmark circuits and a target translator in FORTRAN," (Special Session on ATPG and Fault Simulation), *Proc. IEEE Int. Symp. Circuits and Systems*, June 1985.

[10] P. Agrawal, V.D. Agrawal and Joan Villoldo, "Test pattern generation for sequential circuits on a network of workstations," *Proc. 2nd Int. Symp. High Performance Distributed Computing*, pp. 114-120, July 1993.

Table 1. Number of dominated faults and TPG time

circuit	# of faults	# of dominated faults				TPG time [ms]			
		minimum	maximum	mean	std.dev.*	minimum	maximum	mean	std.dev.*
c880	942	101	244	133.2	29.0	1.6	9.4	4.63	1.39
c1355	1574	0	464	324.4	108.4	5.5	86.7	19.37	7.58
c1908	1879	0	559	445.7	62.1	3.5	316.0	19.37	15.85
c2670	2747	0	621	436.1	108.3	4.7	150.4	16.44	14.49
c3540	3428	0	619	400.2	123.6	5.5	451.9	33.43	22.78
c5315	5350	0	974	718.4	179.4	9.0	92.6	22.39	9.27
c6288	7744	0	3861	1439.6	924.7	10.5	693.3	122.8	63.7
c7552	7550	0	1705	1148.3	236.1	12.9	400.4	45.81	26.85

*standard deviation

Table 2. Number of dominating faults

circuit	# of faults	# of dominating faults			
		minimum	maximum	mean	std.dev.*
c880	942	1	939	133.2	245.9
c1355	1574	0	1556	324.4	398.8
c1908	1879	0	1853	445.7	552.2
c2670	2747	0	2620	436.1	812.1
c3540	3428	0	3199	400.2	593.2
c5315	5350	0	5290	718.4	1368.7
c6288	7744	0	2912	1439.6	1323.8
c7552	7550	0	7294	1148.3	1844.4

*standard deviation

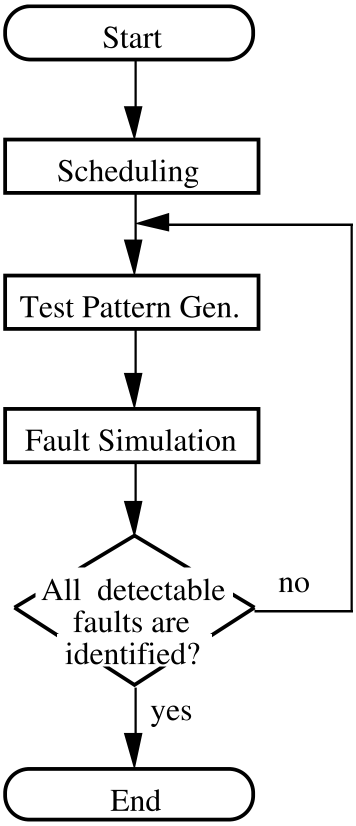

Figure 1. Flow of test generation

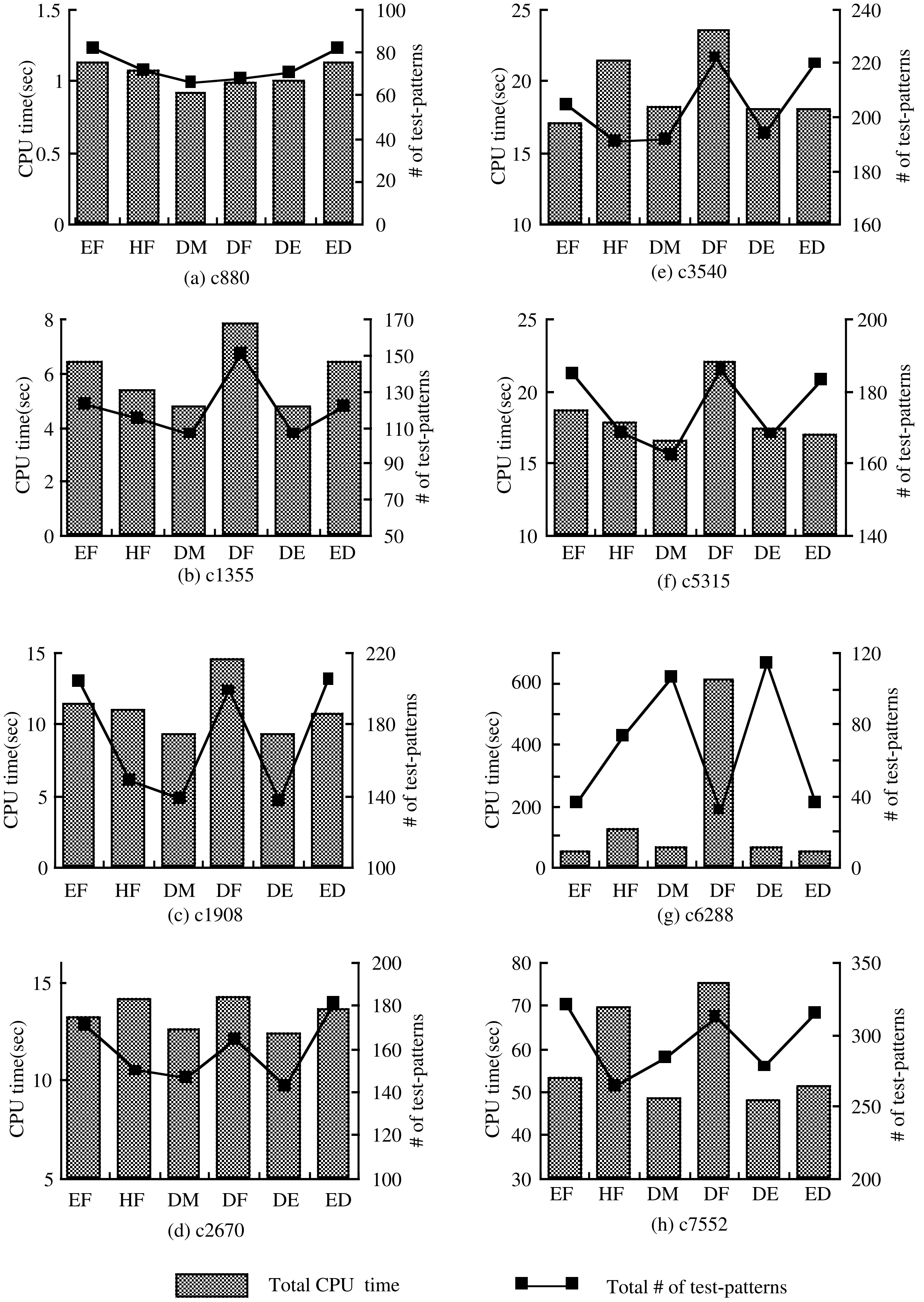

Figure 2. Experimental results:
Scheduling based on TPG time and # of dominated faults

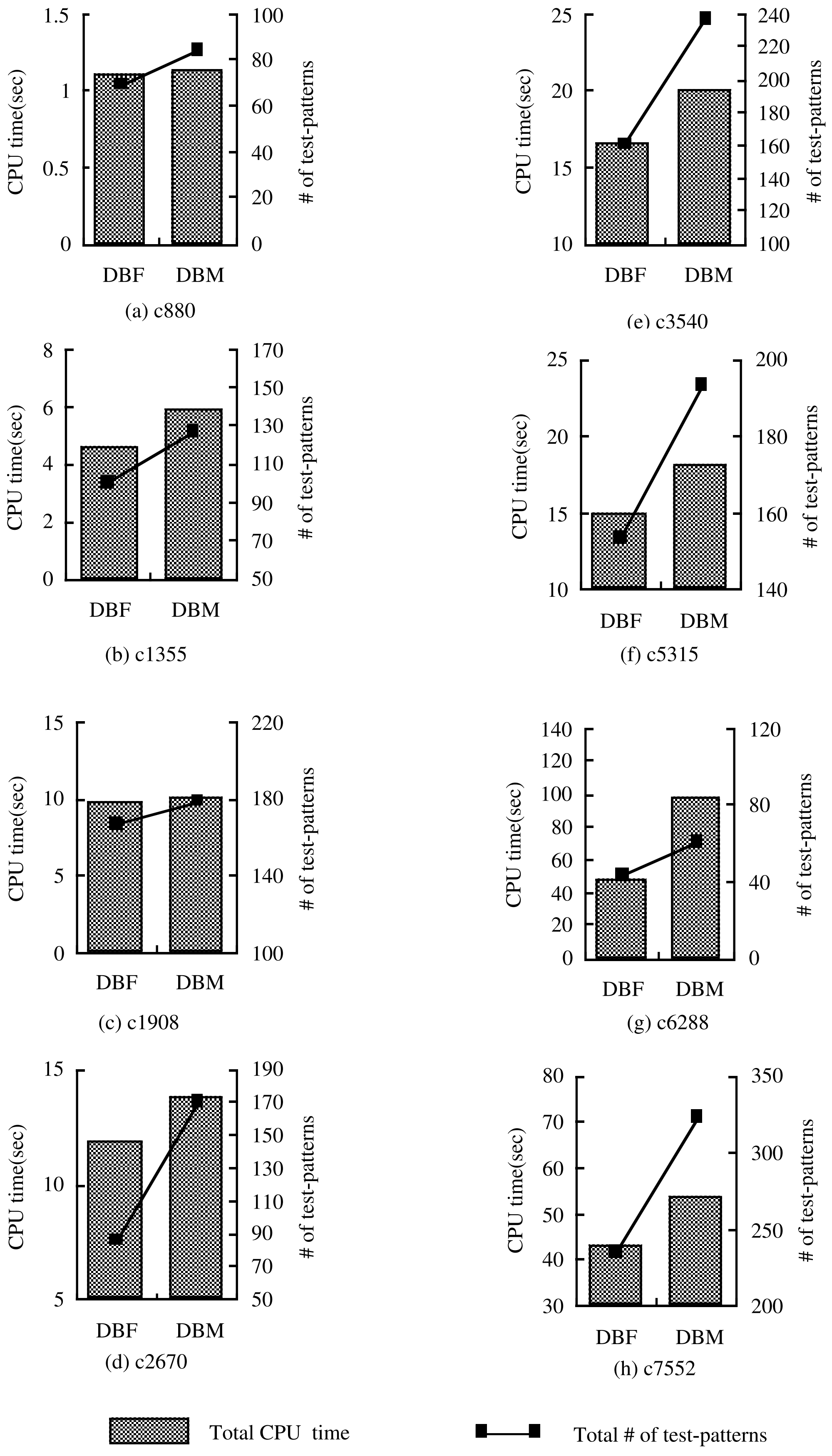

Figure 3. Experimental results:
Scheduling based on # of dominating faults