PAPER

On the Effect of Scheduling in Test Generation

Tomoo INOUE†, *Member*, Hironori MAEDA†*, *Nonmember*, *and* Hideo FUJIWARA†, *Member*

SUMMARY The order of faults which are targeted for test-pattern generation affects both of the processing time for test generation and the number of generated test-patterns. This order is referred to as a *test generation schedule*. In this paper, we consider the effect of scheduling in test generation. We formulate the test generation scheduling problem which minimizes the cost of testing. We propose schedulings based on test-pattern generation time, *dominating probability* and *dominated probability*, and analyze the effect of these schedulings. In the analysis, we show that the total test-pattern generation time and the total number of test-patterns can be reduced by the scheduling according to the descending order of dominating probability prior to the ascending order of test-pattern generation. This is confirmed by the experiments using ISCAS'85 benchmark circuits. Further, in the experiments, we consider eight schedulings, and show that the scheduling according to the ascending order of dominated probability is the most effective of them.

key words: *test generation, test generation schedule, fault ordering, fault dominance, cost of testing*

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] 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 [4]-[7] 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 total test-pattern generation time and the total number of generated test-patterns. Then, we present schedulings based on test-pattern generation time, *dominating probability* and *dominated probability*, and analyze the effect of the schedulings. Finally, we present experimental results on the ISCAS '85 benchmark circuits [8].

2. Formulation of the Scheduling Problem

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

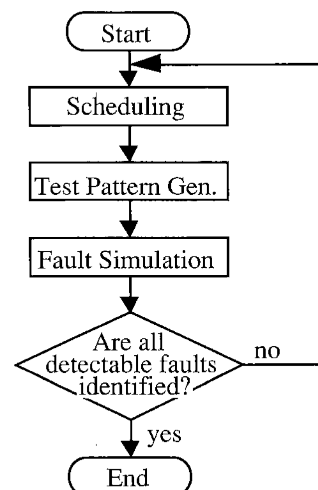


Fig. 1 Flow of test generation.

Manuscript received September 12, 1995.
Manuscript revised February 10, 1996.

† The authors are with Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-01 Japan.
* Presently, with 2nd ATE Division, ADVANTEST Co. Ltd.

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, we can consider that scheduling has much more effect on test-pattern generation time than that on fault simulation time. Hence, 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:

$$T_A(S_{Topt}) = \min_{s}\{T_A(S)\},$$

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:

$$L_A(S_{Lopt}) = \min_{s}\{L_A(S)\}.$$

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 $S = <f_1, f_2, \cdots, f_N>$. Let d_{ij} be the probability that fault f_i dominates fault f_j. Let g_i be the probability that test-pattern generation for the i-th fault f_i in schedule S is executed. Note that if a fault is undetectable (i.e., redundant), no test-pattern is generated. However, here we assume that all faults are detectable. 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, the probability that a test-pattern for the i-th fault is generated can be expressed as

$$\begin{cases} g_1 = 1 \\ g_i \prod_{k=1}^{i-1} (1 - g_k d_{ki}) \quad (2 \le i \le N) \end{cases} \tag{1}$$

Let t_i be the TPG time for the i-th fault f_i in schedule S. From Eq. (1), the total number of test-patterns by schedule S is given by

$$L(S) = \sum_{i=1}^{N} g_i. \tag{2}$$

The total TPG time by schedule S is given by

$$T(S) = \sum_{i=1}^{n} t_i g_i. \tag{3}$$

We shall express these equations as

$$L(S) = \sum_{s} g_i,$$

and

$$T(S) = \sum_{s} t_i g_i,$$

respectively.

In general, the probability d_{ij} that fault f_i dominates fault f_j depends on both of the dominating fault f_i and the dominated fault. However, here we consider two cases; (1) d_{ij} is independent of any dominated fault f_j, i.e., $d_{ij} = d_{i*}$ for all j, and (2) d_{ij} is independent of any dominating fault f_i, i.e., $d_{ij} = d_{*j}$ for all i. In the following, we define *dominating probability* for case (1) and *dominated probability* for case (2), respectively.

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.,

$$d_{i*} = \frac{1}{N} \sum_{j=1}^{N} d_{ij}.$$ □

By substituting d_{k*} for d_{kj} in Eq. (1), we have

$$\begin{cases} g_1 = 1 \\ g_i = \sum_{k=1}^{i-1} (1 - g_k d_{k*}) \\ \quad = g_{i-1}(1 - g_{i-1} d_{i-1*}) \quad (2 \le i \le N) \end{cases} \tag{4}$$

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

$$g_i > g_{i+1} \tag{5}$$

for $1 \leq i \leq N-1$.

Dominated probability: Let the dominated probability d_{*j} of a fault f_j be the average of the probabilities d_{ij} for all i, i.e.,

$$d_{*j} = \frac{1}{N}\sum_{i=1}^{N} d_{ij}. \qquad \square$$

By substituting d_{*k} for d_{ik} in Eq. (1), we have

$$\begin{cases} g_1 = 1 \\ g_i = \prod_{k=1}^{i-1}(1 - g_k d_{*k}) \quad (2 \leq i \leq N) \end{cases}$$

Thus, if we can predict fault characteristics such as dominating probability, dominated 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. However, we have no general method to analyze the effect of the scheduling based on the dominated probability up to date. In the next section, we shall analyze the effect of the test generation scheduling based on dominating probability and TPG time provided that TPG time and dominating probability for each fault are given *a priori*. Then, for simplicity, we shall express dominating probability d_{i*} as d_i.

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 Eq. (4), the probability g_i that a test-pattern for the i-th fault f_i is generated can be expressed as:

$$\begin{cases} g_1 = 1 \\ g_i = g_{i-1}(1 - g_{i-1} d) \quad (2 \leq i \leq N) \end{cases}$$

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 $1 \leq n \leq N-1$. 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 Eq. (2) the total number of test-patterns obtained by schedule S_h is given by

$$L(S_h) = \sum_{S_h} g_i.$$

From Eq. (3) the total TPG time is given by

$$T(S_h) = \sum_{S_h} t_i g_i. \qquad (6)$$

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,

$$\begin{cases} t'_n = t_e,\ t'_{n+1} = t_h \\ t'_i = t_i \quad (i \neq n, n+1) \end{cases} \qquad (7)$$

Let g'_i be the probability that a test-pattern for the i-th fault in order S_e is generated. From Eq. (2) the total number of test-patterns obtained by schedule S_e is given by

$$L(S_e) = \sum_{S_e} g'_i.$$

From Eq. (3) the total TPG time is given by

$$T(S_e) = \sum_{S_e} t'_i g'_i. \qquad (8)$$

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

$$L(S_e) = L(S_h).$$

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 Eqs. (6), (7) and (8), the difference between these two total TPG time can be expressed as

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

Since $t_h > t_e$ and $g_n > g_{n+1}$ (from Inequality (5)),

$$T(S_h) - T(S_e) > 0.$$

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. $\square$

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 Eqs. (2) and (3) the total TPG time by a schedule S can be expressed as

$$T(S) = \sum_{S} t g_i = t L(S).$$

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 $1 \leq n \leq N-1$. 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 Eq. (4) we have

$$g_{n+1}=g_n(1-g_n d_f), \quad (9)$$

and

$$g_{n+2}=g_{n+1}(1-g_{n+1}d_m).$$

Thus the total number of test-patterns generated in order S_f can be expressed from Eq. (2) as

$$L(S_f)=\sum_{S_f} g_i. \quad (10)$$

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,

$$\begin{cases} d_n'=d_m,\ d_{n+1}'=d_f \\ d_i'=d_i \quad (i\neq n, n+1) \end{cases} \quad (11)$$

Let g_i' be the probability that a test-pattern for the i-th fault in order S_m is generated. From Eq. (4) we have

$$g_{n+1}'=g_n'(1-g_n'd_m), \quad (12)$$

and

$$g_{n+2}'=g_{n+1}'(1-g_{n+1}'d_f).$$

Thus the total number of test-patterns generated in order S_m can be expressed from Eq. (2) as

$$L(S_m)=\sum_{S_m} g_i'. \quad (13)$$

From Eqs. (10) and (13) the difference between $L(S_f)$ and $L(S_m)$ can be expressed as

$$L(S_f)-L(S_m)=\sum_{i=1}^{n}(g_i-g_i')+(g_{n+1}-g_{n+1}') + (g_{n+2}-g_{n+2}')+\sum_{i=n+3}^{n}(g_i-g_i').$$

Since $d_i=d_i'$ for $i\leq n-1$ (Eq. (11)), from Eq. (4) we have

$$g_i=g_i' \quad (14)$$

for $i\leq n$. Hence,

$$\sum_{i=1}^{n}(g_i-g_i')=0.$$

From Eqs. (9), (12) and (14) and $d_f<d_m$, we have

$$g_{n+1}-g_{n+1}'=g_n^2(d_m-d_f)>0.$$

In the same way as the above inequality we have

$$g_{n+2}-g_{n+2}'=g_n^4 d_f d_m(d_m-d_f)>0. \quad (15)$$

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 Eq. (4) as

$$g_i-g_i'=(g_{i-1}-g_{i-1}')(1-(g_{i-1}+g_{i-1}')d_{i-1}). \quad (16)$$

If we assume that $d_i<1/2$ for all i, which is reasonable, then we have

$$(g_{i-1}+g_{i-1}')d_{i-1}<1 \quad (17)$$

in Eq. (16). Hence, from Eq. (16) and Inequalities (15) and (17), we have

$$g_i-g_i'>0$$

for $n+3\leq i\leq N$. Therefore,

$$\sum_{i=n+3}^{N}(g_i-g_i')>0.$$

Thus we have

$$L(S_f)-L(S_m)>0, \quad (18)$$

and accordingly

$$T(S_f)-T(S_m)>0. \quad (19)$$

Inequalities (18) and (19) 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 and that the dominating probability is less than 1/2 for all faults. □

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 $1\leq n\leq N-1$. 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,

$$\begin{cases} d_n'=d_{n+1},\ t_n'=t_{n+1},\ d_{n+1}'=d_n,\ t_{n+1}'=t_n \\ d_i'=d_i,\ t_i'=t_i \quad (i\neq n, n+1) \end{cases} \quad (20)$$

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 Eq. (4) we have

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

and

$$L(S_b)=\sum_{i=1}^{N} g'_i.$$

In the same way as the analysis in the previous section 3.2, we have $g_i=g'_i$ for $i \leq n$, $g_i<g'_i$ for $n<i\leq N$. Hence, the difference between the total number of test-patterns by order S_a and that by order S_b is expressed as

$$L(S_a)-L(S_b)<0.$$

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 Eq. (3) we have

$$T(S_a)=\sum_{i=1}^{N} t_i g_i,$$

and

$$T(S_b)=\sum_{i=1}^{N} t'_i g'_i.$$

From Eq. (4) we have

$$g_{n+1}=g_n(1-g_n d_n).$$

Since $g_i=g'_i$ for $i\leq n$, from Eq. (20) we have

$$g'_{n+1}=g'_n(1-g'_n d'_n)$$
$$=g_n(1-g_n d_{n+1}).$$

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

$$T(S_a)-T(S_b)=g_n^2(t_n d_{n+1}-t_{n+1}d_n)+\sum_{i=n+2}^{N} t_i(g_i-g'_i). \quad (21)$$

Since $g_i<g'_i$ for $n<i\leq N$, we have

$$\sum_{i=n+2}^{N} t_i(g_i-g'_i)<0.$$

Hence, if

$$\frac{t_n}{t_{n+1}}\leq\frac{d_n}{d_{n+1}},$$

then

$$T(S_a)-T(S_b)<0. \quad (22)$$

Note that if $d_n>d_{n+1}$ and $t_n\leq t_{n+1}$, then Inequality (22) is satisfied. Therefore, if

$$\frac{t_i}{t_j}\leq\frac{d_i}{d_j}$$

for any pair of faults (f_i, f_j) such that $d_i>d_j$ and $t_i>t_j$, then Inequality (22) is satisfied for any n. Hence, we have the following theorem.

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

$$\frac{t_i}{t_j}\leq\frac{d_i}{d_j}$$

for any pair of faults (f_i, f_j) such that $d_i>d_j$ and $t_i>t_j$. □

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, as mentioned above, since $d_n>d_{n+1}$ in Eq. (21), if $t_n\leq t_{n+1}$, then

$$T(S_a)-T(S_b)<0.$$

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

In order to confirm the practical effect of scheduling in test generation, we made experiments of test generation scheduling based on TPG time, dominating probability and dominated probability using the ISCAS'85 benchmark circuits [8] on a DECstation 5000/25. The FAN algorithm [2] and the concurrent fault simulation algorithm [9] were used as a test-pattern generator and a fault simulator, respectively.

First, in order to obtain accurate fault characteristics for test generation such as TPG time, dominating probability and dominated probability, we computed the CPU time of test-pattern generation for each fault, the number of faults dominated by each fault and the number of faults that dominate each fault on the benchmark circuits by using FAN and a modified fault simulator. Note that the number of faults dominated by a fault over the total number of faults and the number of faults that dominate a fault over the total number of faults denote the above-mentioned dominating probability and dominated probability of the fault, respectively. The test-pattern generator FAN usually generates incompletely-specified patterns, i.e., a generated test-pattern for a fault may include a don't-care value. In this computation, we substituted a constant

Table 1 Fault characteristics in the ISCAS '85 benchmark circuits.

circuit	# of faults	TPG time [ms]				# of dominated faults				# of dominating faults			
		min.	max.	mean	s.d. *	min.	max.	mean.	s.d.*	min.	max.	mean.	s.d.*
c880	942	1.6	9.4	4.63	1.39	101	244	133.2	29.0	1	939	133.2	245.9
c1355	1574	5.5	86.7	19.37	7.58	0	464	324.4	108.4	0	1556	324.4	398.8
c1908	1879	3.5	316.0	19.37	15.85	0	559	445.7	62.1	0	1853	445.7	552.2
c2670	2747	4.7	150.4	16.44	14.49	0	621	436.1	108.3	0	2620	436.1	812.1
c3540	3428	5.5	451.9	33.43	22.78	0	619	400.2	123.6	0	3199	400.2	593.2
c5315	5350	9.0	92.6	22.39	9.27	0	974	718.4	179.4	0	5290	718.4	1368.7
c6288	7744	10.5	693.3	122.8	63.7	0	3861	1439.6	924.7	0	2912	1439.6	1323.8
c7552	7550	12.9	400.4	45.81	26.85	0	1705	1148.3	236.1	0	7294	1148.3	1844.4

* standard deviation

logic value for all don't-care values in test-patterns generated by FAN, in order to make completely-specified test-patterns independent of the order of target faults. Table 1 shows computational results for benchmark circuits. In Table 1, the number of faults dominated by a fault and the number of faults that dominate a fault are denoted by the number of dominating faults and the number of dominating faults, respectively. For a fault that was identified as a redundant one, both the number of faults dominated by the faults and the number of faults that dominate the fault were regarded as 0, and the CPU time for the fault was regarded as the time required to identify it. For a fault that were aborted to be test-generated, the number of faults dominated by the fault was regarded as 0, and the TPG time for the fault was regarded as the time required for the abortion.

Based on the fault characteristics (CPU time, number of dominated faults and number of dominating faults) obtained from the above computation, we considered the following eight schedules.

(*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.

(*DBF*) Ascending order of the number of dominating faults, called a *Dominated-By-Few fault first* scheduling.

(*DBM*) Descending order of the number of dominating faults, which is called a *Dominated-By-Many fault first* scheduling.

Note that the first six schedules correspond to the schedulings analyzed in the previous section. The last two schedules correspond to the schedulings according to the ascending order of dominated probability and the descending order of dominated probability. We implemented these eight schedules in test generation, and obtained the total processing time (including fault simulation time) and the total number of test-patterns on ISCAS '85 benchmark circuits. Here, don't-care values in generated test-patterns were specified randomly.

Table 2 and Fig. 2 show the total processing time (including fault simulation time) and the number of generated test-patterns for each scheduling method. From these results we can see that the *EF* and *DM* schedulings can reduce the total processing time for the most circuits compared with the *HF* and *DF* schedulings, and 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 the circuits except c6288. These results coincide with our analytical results. 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 the circuits except c6288. This is because dominating probability might not be independent of TPG time. 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. In c6288 dominating-many faults might be easy-to-test, and faults dominated by dominating-many faults would overlap one another. Further, dominating-few

Table 2 Experimental results.

	Total processing time [sec]								Number of test-petterns							
	EF	HF	DM	DF	DE	ED	DBF	DBM	EF	HF	DM	DF	DE	ED	DBF	DBM
c880	1.14	1.07	0.92	0.99	1.00	1.12	1.10	1.13	82	71	66	67	70	82	69	84
c1355	6.46	5.30	4.81	7.85	4.80	6.39	4.56	5.88	123	115	106	151	106	122	100	126
c1908	11.4	10.9	9.23	14.5	9.26	10.7	9.73	10.1	204	149	139	199	138	205	167	179
c2670	13.2	14.1	12.5	14.2	12.3	13.6	11.8	13.7	171	150	146	165	142	181	86	170
c3540	17.0	21.5	18.2	23.5	18.0	18.0	16.4	19.9	204	191	192	222	194	220	161	237
c5315	18.7	17.9	16.4	22.1	17.4	17.1	14.9	18.1	185	169	162	186	168	183	153	193
c6288	47.9	120.2	61.2	616.9	61.3	49.6	47.4	98.0	36	73	106	32	114	36	43	61
c7552	53.1	69.5	48.3	75.4	49.9	51.0	42.8	53.6	320	263	284	312	278	315	234	324

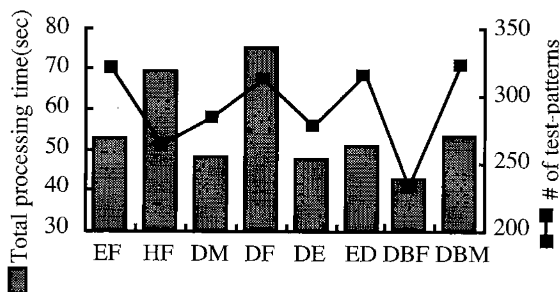


Fig. 2 Experimental results for c7552.

faults might be hard-to-test, and faults dominated by dominating-few faults would be different from one another. In this way, the scheduling based only on the dominating probability is not effective for such circuits as c6288.

From the results of the DBF and DBM schedulings in Table 2 and Fig. 2, 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. Furthermore, 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. This is because the standard deviation of the number of dominating faults is large as compared with that of the number of dominated faults for all circuits.

5. Conclusions and Future Work

In this paper, we considered a scheduling problem in test generation for combinational logic circuits. We proposed schedulings based on test-pattern generation time, dominating probability and dominated probability, and analyzed the effect of these schedulings. In the analysis, we showed that the total test-pattern generation time and the total number of test-patterns can be reduced by the scheduling according to the descending order of dominating probability prior to the ascending order of test-pattern generation. This was confirmed by the experiments using ISCAS '85 benchmark circuits. Further, in the experiments, we considered eight schedulings, and showed that the scheduling according to the ascending order of dominated probability is the most effective of them.

In order to consider the effect of scheduling in test generation, we assumed that fault characteristics such as test-pattern generation time, dominating probability and dominated probability of all faults were given *a priori*. Hence, a remaining problem is to predict accurate fault characteristics for scheduling before test generation with small computation time. Controllability/observability measures [10] might be used to estimate test-pattern generation time, dominating probability and dominated probability.

Acknowledgements

The authors would like to thank Dr. Toshimitsu Masuzawa of Nara Institute of Science and Technology, Japan for his helpful discussion. This work was supported in part by Grant-in-Aid for Scientific Research No. 06680322 from the Ministry of Education, Science and Culture, Japan.

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, March 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 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.

[4] 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.

[5] 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.

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

[7] S. Kajihara, I. Pomeranz, K. Kinoshita, and S.M. Reddy,

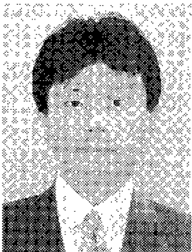
"Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits," Proc. 30th ACM/ IEEE Design Automation Conf., pp. 102–106, 1993.

[8] F. Brglez and H. Fujiwara, "A neutral netlist of ten combinational benchmark circuits and a target translator in FORTRAN," Proc. IEEE Int. Symp. Circuits and Systems, June 1985.

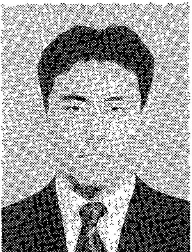
[9] E. G. Ulrich and T. Baker, "The concurrent simulation of nearly identical digital networks," Proc. 10th Design Automation Workshop, pp. 145–150, 1973.

[10] L. H. Goldstein, "Controllability/observability analysis of digital circuits," IEEE Trans. Circuits. & Syst., vol. CAS-26, pp. 685–693, Sept. 1979.

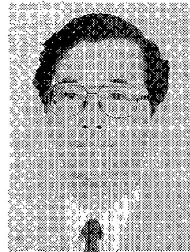
[11] P. Agrawal, V. D. Agrawal, and J. Villoldo, "Test pattern generation for sequential circuits on a network of workstations," Proc. 2nd Int. Symp. High Performance Distributed Computing, pp. 114–120, July 1993.



Tomoo Inoue received the B.E. degree in electronics and communication engineering and the M.E. degree in electrical engineering from Meiji University, Kawasaki, Japan in 1988 and 1990, respectively. From 1990 to 1992, he was engaged in research and development of microprocessors at Matsushita Electric Industrial Co., Ltd., Osaka, Japan. Since 1993 he has been a Research Associate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His research interests include test generation, synthesis for testability and parallel processing. Mr. Inoue is a member of the IEEE, the Institute of Electronics and the Information Processing Society of Japan.



Hironori Maeda received the B.E. degree in information system engineering from Osaka University, Osaka, Japan, in 1993, and the M.E. degree in information processing from Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 1995. He is currently working at ADVANTEST Co. Ltd., Japan, and engaged in research and development of automatic test equipments. His research interests include test generation, test compaction and high level synthesis.



Hideo Fujiwara received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. In 1981 he was a Visiting Research Assistant Professor at the University of Waterloo, and in 1984 he was a Visiting Associate Professor at McGill University, Canada. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Prof. Fujiwara's research interests are in the design and test of computers, including design for testability, built-in self-test, test pattern generation, fault simulation, computational complexity, parallel processing, neural networks and expert systems for design and test. He is the author of *Logic Testing and Design for Testability* (MIT Press, 1985). Dr. Fujiwara is a *fellow* of the IEEE as well as a member of the Information Processing Society of Japan. He received the IECE Young Engineer Award in 1977 and IEEE Computer Society Certificate of Appreciation Award in 1991.