PAPER

Performance Analysis of Parallel Test Generation for Combinational Circuits

Tomoo INOUE[†], *Member*, Takaharu FUJII[†*], *Nonmember*, *and* Hideo FUJIWARA[†], *Member*

SUMMARY The problem of test generation for VLSI circuits computationally requires prohibitive costs. Parallel processing on a multiprocessor system is one of available methods in order to speedup the process for such time-consuming problems. In this paper, we analyze the performance of parallel test generation for combinational circuits. We present two types of parallel test generation systems in which the communication methods are different; *vector broadcasting (VB)* and *fault broadcasting (FB)* systems, and analyze the number of generated test vectors, the costs of test vector generation, fault simulation and communication, and the speedup of these parallel test generation systems, where the two types of communication factors; the *communication cut-off factor* and the *communication period*, are applied. We also present experimental results on the VB and FB systems implemented on a network of workstations using ISCAS '85 and ISCAS '89 benchmark circuits. The analytical and experimental results show that the total number of test vectors generated in the VB system is the same as that in the FB system, the speedup of the FB system is larger than that of the VB, and it is effective in reducing the communication cost to switch broadcasted data from vectors to faults.

key words: *test generation, parallel processing, performance analysis, interprocessor communication, speedup*

1. Introduction

The problem of test generation for VLSI circuits computationally requires prohibitive costs. Parallel processing on a multiprocessor system is one of available methods in order to speedup the process for such time-consuming problems, and many studies have reported various techniques for parallel test generation [1].

One important parallelization of test generation is dealing with different faults in parallel, referred to as *fault parallelism* [2]–[5]. In fault parallelism, a set of faults is partitioned into subsets, and allocated to processors on a network. Each processor performs test vector generation and fault simulation. Fujiwara and Inoue [2] considered a parallel test generation on a network of general purpose computers such as workstations using a *client-server model (CS model)*. They presented *granularity* which refers to the number of target faults allocated to a processor by each communication, and proposed the optimal granularity which minimizes the total processing time for test generation on the CS model. In [3], they further considered a *client-agent-server model (CAS model)* which can reduce the work load of the client processor by adding agent processors to the CS model, and showed that there exists an optimal scheme (an optimal pair of numbers of agent processors and client processors) which minimizes the total processing time for test generation on the CAS model. In their system, detected faults are transferred among processors. Patil and Banerjee [4] aimed at decreasing the number of generated test vectors, and presented a fault partitioning strategy which reduces duplication of test generation. Moreover, they introduced the *communication cut-off factor* to reduce the communication overhead. These approaches were implemented on a hypercube machine where each processor communicated with one another by sending generated test vectors. Klenke *et al.* [5] considered an implementation of the approaches presented in [4] on a network of workstations, and made an experiment on broadcasting faults instead of test vectors.

In this paper, we shall analyze the performance of parallel test generation for combinational circuits. We shall present two types of parallel test generation systems in which the communication methods are different; *vector broadcasting (VB)* and *fault broadcasting (FB)* systems. Based on the relationship between the number of test vectors and the number of unprocessed (identified as detectable or redundant) faults in a uniprocessor system, we shall analyze the number of generated test vectors, the costs of test vector generation, fault simulation and communication, and the speedup of these parallel test generation systems, where two types of communication factors; the *communication cut-off factor* [4] and the *communication period* (corresponding to *granularity* in [2], [3]), are applied. We shall also present experimental results on the VB and FB systems implemented on a network of workstations using ISCAS '85 [6] and ISCAS '89 [7] benchmark circuits.

2. Architecture of Parallel Processing Systems

In this paper, we consider a parallel processing system which consists of multiple processors connected to a single communication bus. In this network, the set of faults

Manuscript received November 1, 1995.
Manuscript revised May 23, 1996.

[†] The authors are with Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–01 Japan.
[*] Presently, with LSI Operation Unit, NEC Corporation, Kawasaki-shi, 211 Japan.

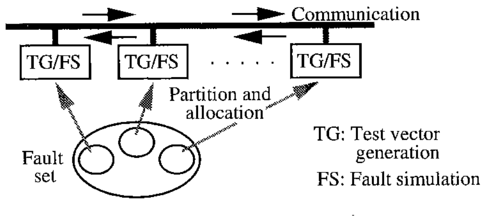


Fig. 1 Parallel test generation system.

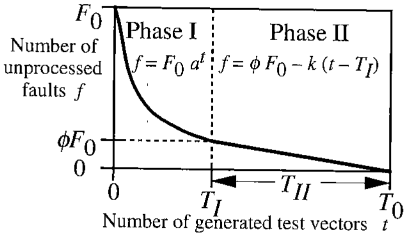


Fig. 2 Number of unprocessed faults.

is partitioned into subsets, and each subset of faults is allocated to eachprocessor (Fig. 1). The faults allocated to a processor are said to be the *target faults* for the processor. Each processor repeats test vector generation (TG) and fault simulation (FS) until all the target faults are *processed.* The result of a *process* for a fault is whether 1) the fault is detected, or 2) the fault is identified as undetectable, i.e., redundant.

In most cases, a test vector detects more than one fault, and consequently a fault allocated to a processor may be able to be detected with the test vector that is generated another processor, i.e., a processor may process the faults that are processed on the other processors. In order to reduce this work duplication, the results of processes are transferred among processors. The transferred data can be considered as generated test vectors and processed faults, and accordingly two types of parallel test generation systems can be considered; the *Vector Broadcasting (VB) system* in which generated test vectors are broadcasted and the *Fault Broadcasting (FB) system* in which processed faults are broadcasted. Moreover, since the communication will be expensive, and restrain the increase of the overall speedup, the parameters of the *communication cut-off factor,* p_c, and the *communication period,* λ, are applied to decrease the communication cost.

Vector Broadcasting: Each processor makes a *fault list* of target faults. First, each processor selects a fault from the fault list, and generates a test vector for the fault. Then, it identifies all the detectable faults with the test vector by performing FS. When the number of generated test vectors reaches λ by repeating TG and FS, the processor sends the test vectors to all other processors, i.e., broadcasts the vectors. This broadcasting is repeated until the fault coverage of the target fault reaches p_c. On the other hand, when a processor receives test vectors from another processor, the processor performs FS with the received test vectors for the unprocessed faults in the fault list. These processes are repeated until all the target faults are processed on each processor.

Fault Broadcasting: Each processor makes a fault list of all faults in the circuit, not just its target faults. Each processor selects one of the target faults from the fault list, and generates a test vector for the fault. Then, it performs FS with the test vector for all the faults in the fault list. When the number of generated test vectors reaches λ by repeating TG and FS, the processor broadcasts not the test vectors but the faults processed by TG and FS during generating λ test vectors to all other processors. This broadcasting is repeated until the fault coverage of the target fault reaches p_c. On the other hand, when a processor receives processed faults from another processor, it updates its fault list with the received faults. After the completion of broadcasting, FS is performed for only target faults in the same way as the VB system. These processes are repeated until all the target faults are processed on each processor.

3. Performance Analysis

3.1 Test Generation in Uniprocessor System

The relationship between the number of unprocessed faults and the number of generated test vectors has been analyzed by Goel [8]. Based on his analysis, we assume that the relationship can be shown by the curve of Fig. 2. As shown in this figure, the curve is composed of two phases and is determined by F_0, a, k and ϕ, where F_0 is the total number of faults and a, k and ϕ are constants which depend on the circuit and the test generation algorithm, as follows.

(i) In the first phase, called **Phase I**, the number of faults processed with a test vector is proportional to the number of faults remaining before generating the test vector. Hence, the number of unprocessed faults f for t generated test vectors is expressed as

$$f = F_0 a^t \tag{1}$$

where $0 < a < 1$. When the number of unprocessed faults becomes ϕF_0 $(0 < \phi < 1)$, the phase turns from Phase I to the second phase, **Phase II**. Therfore, the number of unprocessed faults at the end of Phase I can be expressed as

$$\phi F_0 = F_0 a^{T_I} \tag{2}$$

where T_I is the number of generated test vectors in Phase I.

(ii) In Phase II, k faults are processed with each test vector. Hence, the number of unprocessed faults in Phase II can be expressed as

$$f = \phi F_0 - k(t - T_I) \tag{3}$$

where $k \geqq 1$. By letting T_0 be the total number of generated test vectors, from Eq. (3), we have

$$\begin{aligned} 0 &= \phi F_0 - k(T_0 - T_I) \\ \phi F_0 &= k T_{II} \end{aligned} \tag{4}$$

where T_{II} is the number of generated test vectors in Phase II.

From Eqs. (2) and (4), we have

$$T_I = \log_a \phi \tag{5}$$

$$T_{II} = \frac{\phi F_0}{k}. \tag{6}$$

Therefore, the total number of generated test vectors is given by

$$\begin{aligned} T(U) &= T_0 = T_I + T_{II} \\ &= \log_a \phi + \frac{\phi F_0}{k}. \end{aligned} \tag{7}$$

Throughout our analysis, we shall use the CPU time estimates as an estimate of the test generation cost. Let c_t be the cost of generating a test vector for a fault. From Eq. (7), the total TG cost is expressed as

$$\begin{aligned} C_{test}(U) &= c_t T_0 \\ &= c_t \left(\log_a \phi + \frac{\phi F_0}{k} \right). \end{aligned} \tag{8}$$

The cost of FS with a vector is assumed to be composed of the factor c_{s0} which is independent of the number of simulated faults, and the factor c_{s1} which is proportional to the number of simulated faults, i.e., the number of unprocessed faults. Hence, from Eqs. (1) and (3), the total FS cost can be expressed as

$$\begin{aligned} C_{sim}(U) &= \sum_{t=1}^{T_I} \left(c_{s0} + c_{s1} F_0 a^{t-1} \right) \\ &\quad + \sum_{t=1}^{T_{II}} \left(c_{s0} + c_{s1} \left(\phi F_0 - k(t-1) \right) \right) \\ &= c_{s0} \left(\log_a \phi + \frac{\phi F_0}{k} \right) \\ &\quad + c_{s1} F_0 \left(\frac{1-\phi}{1-a} + \frac{\phi}{2} \left(1 + \frac{\phi F_0}{k} \right) \right). \end{aligned} \tag{9}$$

Therefore, the total cost of test generation in the uniprocessor system can be expressed as

$$C_{total}(U) = C_{test}(U) + C_{sim}(U). \tag{10}$$

We shall analyze the performance of the VB and FB systems under the above analytical results.

3.2 Test Generation in Multiprocessor Systems

In order to apply the analytical results of the uniprocessor system to the analysis of the multiprocessor systems, we have the following assumptions;

(i) All processors are uniform, i.e., the cost of TG for a fault and the cost of FS with a test vector are denoted by c_t, c_{s0} and c_{s1} for all processors.

(ii) Equal and homogeneous work load is allocated to each processor, i.e., an equal number of faults is allocated to each processor, and the ratio of easy-to-detect faults to hard-to-detect faults is the same for all processors.

(iii) Let S be the set of all faults for a given circuit. Let S' be any set of faults that are randomly extracted from S. The relationship between the number of test vectors and the number of unprocessed faults for S' is the same as that for S, i.e., the relationship between the number of test vectors and the number of unprocessed faults for S' is denoted by the parameters, q, ϕ and k, which denote that for S.

To simplify the discussion, we analyze the performance of the VB and FB systems provided that all processors communicate during Phase I, i.e., $0 \leqq p_c \leqq 1 - \phi$.

3.2.1 Number of Unprocessed Faults

First, let us consider the relationship between the number of unprocessed faults and the number of generated test vectors in the VB system.

Let N be the number of processors. Since F_0 faults are partitioned among N processors, each processor performs TG and FS for F_0/N faults. While the fault coverage is less than p_c, each processor broadcasts λ generated test vectors. If it is assumed that while a processor generates λ test vectors, another processor also generates λ test vectors for all processors, then when each processor broadcasts λ test vectors, it receives $\lambda(N-1)$ test vectors from $N-1$ processors. Since the λ test vectors are generated without respect to the received $\lambda(N-1)$ test vectors, some of faults detected with the generated test vectors may be also detected with the received test vectors. This work duplication will increase as the number of N increases, and as the number of the communication period λ increase. Therefore, we assume that the number of faults detected with λN test vectors generated in an N-processor system is equal to the number of faults detected with $(\lambda N)^r$ test vectors generated in the uniprocessor system, where $0 < r < 1$. Thus, the number of unprocessed faults can be expressed as

$$
\begin{aligned}
f_{vb} &= \frac{F_0}{N}a^{(\lambda N)^r b} && (F_0 \geq f \geq (1-p_c)F_0)\\
f_{vb} &= (1-p_c)\frac{F_0}{N}a^{t-T_{cvb}} && ((1-p_c)F_0 \geq f \geq \phi F_0)\\
f_{vb} &= \frac{\phi F_0}{N} - k(t - T_{Ivb}) && (\phi F_0 \geq f \geq 0)
\end{aligned} \tag{11}
$$

where b is the number of broadcasts, t is the number of generated test vectors, and T_{cvb} and T_{Ivb} are the numbers of test vectors generated before the completion of broadcasting and in Phase I, respectively.

Next, we consider the number of unprocessed faults in the FB system. In the FB system, while the fault coverage is less than p_c, FS is performed not only for the target faults but also for all the unprocessed faults in the fault list. After the fault coverage reaches p_c, each processor does not broadcast processed faults, and performs FS for only its target faults as well as that in the VB system. Thus, the number of unprocessed faults in the FB system can be expressed as

$$
\begin{aligned}
f_{fb} &= F_0 a^{(\lambda N)^r b} && (F_0 \geq f \geq (1-p_c)F_0)\\
f_{fb} &= (1-p_c)\frac{F_0}{N}a^{t-T_{cfb}} && ((1-p_c)F_0 \geq f \geq \phi F_0)\\
f_{fb} &= \frac{\phi F_0}{N} - k(t - T_{Ifb}) && (\phi F_0 \geq f \geq 0)
\end{aligned} \tag{12}
$$

where T_{cfb} and T_{Ifb} are the numbers of test vectors generated before the completion of broadcasting and in Phase I, respectively.

Note that when $p_c = 0$, no data is transferred among processors, and accordingly Eq. (11) is equal to Eq. (12), i.e.,

$$
\begin{aligned}
f_{nb} &= \frac{F_0}{N}a^{t} && (F_0 \geq f \geq \phi F_0)\\
f_{nb} &= \frac{\phi F_0}{N} - k(t - T_{Inb}) && (\phi F_0 \geq f \geq 0)
\end{aligned} \tag{13}
$$

where T_{Inb} are the number of test vectors generated during Phase I in this system. The system expressed as Eq. (13) is especially called the *Non-Broadcasting (NB)* system.

3.2.2 Number of Test Vectors and TG cost

Let B_{vb} be the total number of broadcasts in the VB system. When the fault coverage reaches p_c, broadcasting is finished. Hence, from Eq. (12) we have

$$
\begin{aligned}
(1-p_c)\frac{F_0}{N} &= \frac{F_0}{N}a^{(\lambda N)^r B_{vb}}\\
B_{vb} &= \frac{\log_a(1-p_c)}{(\lambda N)^r}.
\end{aligned} \tag{14}
$$

Since λ test vectors are broadcasted at each broadcasting, the number of test vectors generated during broadcasting in the VB system is expressed as

$$
\begin{aligned}
T_{cvb} &= \lambda B_{vb}\\
&= \frac{\lambda \log_a(1-p_c)}{(\lambda N)^r}.
\end{aligned} \tag{15}
$$

The numbers of unprocessed faults at the end points of Phase I and Phase II can be expressed as

$$\phi\frac{F_0}{N} = (1-p_c)\frac{F_0}{N}a^{T_{Ivb}-T_{cvb}} \tag{16}$$

$$0 = \phi\frac{F_0}{N} - kT_{IIvb} \tag{17}$$

where T_{IIvb} is the number of test vectors in Phase II. From these equations and Eqs. (5) and (6), we have

$$T_{Ivb} = T_I - \log_a(1-p_c) + T_{cvb} \tag{18}$$

$$T_{IIvb} = \frac{T_{II}}{N}. \tag{19}$$

Hence, the number of test vectors generated by each processor is

$$
\begin{aligned}
T_{0vb} &= T_{Ivb} + T_{IIvb}\\
&= T_I - \left(1 - \frac{\lambda}{(\lambda N)^r}\right)\log_a(1-p_c) + \frac{T_{II}}{N}.
\end{aligned} \tag{20}
$$

Therefore, the total number of test vectors generated in the VB system can be expressed as

$$
\begin{aligned}
T(VB) &= NT_{0vb}\\
&= T(NB) - N\left(1 - \frac{\lambda}{(\lambda N)^r}\right)\log_a(1-p_c)
\end{aligned} \tag{21}
$$

where

$$T(NB) = T(U) + (N-1)T_I. \tag{22}$$

From Eqs. (21) and (22), we can see that the total number of test vectors increases as the number of processors increases. However, we can also see that the total number of test vectors decreases as the communication cutoff factor increases, and as the communication period decreases (Fig. 3).

In the same way as the VB system, the total number of broadcasts in the FB system can be derived from Eq. (12), and the number is equal to that in the VB system, i.e.,

$$B_{fb} = B_{vb}. \tag{23}$$

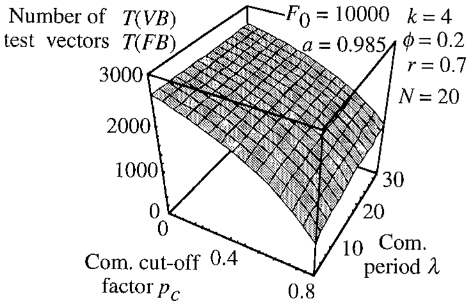


Fig. 3 Number of test vectors.

After the completion of broadcasting, each processor performs TG and FS in the same way as the VB system. Consequently, the total number of test vectors generated in the FB system is equal to that in the VB system, i.e.,

$$T(FB)=T(VB). \tag{24}$$

From Eq. (20), the total TG cost in the VB and FB systems is obtained in the same way as Eq. (8) as follows.

$$\begin{aligned} C_{test}(VB) &= C_{test}(FB) \\ &= C_{test}(NB) - c_t\left(1-\frac{\lambda}{(\lambda N)^r}\right)\log_a(1-p_c) \end{aligned} \tag{25}$$

where

$$C_{test}(NB) = c_t\left(\log_a\phi + \frac{\phi F_0}{kN}\right). \tag{26}$$

From Eqs. (25) and (26) we can see that in the VB and FB systems, the TG cost in Phase II can be reduced by multiprocessing, and the total TG cost can be further reduced by broadcasting compared with the NB system.

3.2.3 FS Cost

First, let us consider the FS cost after the $(b-1)$-th broadcasting until the b-th broadcasting in the VB system. Each processor preforms FS with test vectors generated by itself once and with test vectors received from $(N-1)$ processors, and consequently FS is performed N times. Since the number of faults detected by performing FS with λN test vectors generated in an N-processor system is equal to that with $(\lambda N)^r$ test vectors generated in the uniprocessor system, we assume that each processor can detect several faults with the first $(\lambda N)^r$ test vectors and can detected no fault with the last $\lambda N-(\lambda N)^r$ test vectors by performing FS. Hence, the number of faults simulated between the $(b-1)$-th and b-th broadcastings is expressed as

$$\begin{aligned} n_{fvb}(b) &= \sum_{t=1}^{(\lambda N)^r}\frac{F_0}{N}a^{(\lambda N)^r(b-1)+(t-1)} \\ &\quad + (\lambda N-(\lambda N)^r)\frac{F_0}{N}a^{(\lambda N)^r b} \\ &= \frac{1-a^{(\lambda N)^r}}{1-a}\frac{F_0}{N}a^{(\lambda N)^r(b-1)} \\ &\quad + (\lambda N-(\lambda N)^r)\frac{F_0}{N}a^{(\lambda N)^r b}. \end{aligned} \tag{27}$$

Broadcasting is repeated B_{vb} times while the fault coverage is less than p_c, and after the completion of broadcasting, FS is performed only for the unprocessed faults in the fault list. Therefore, the total FS cost given by

$$\begin{aligned} & C_{sim}(VB) \\ &= \sum_{b=1}^{B_{vb}}(c_{s0}\lambda N + c_{s1}n_{fvb}(b)) \\ &\quad + \sum_{t=1}^{T_{Ivb}-T_{cvb}}\left(c_{s0}+c_{s1}(1-p_c)\frac{F_0}{N}a^{t-1}\right) \\ &\quad + \sum_{t=1}^{T_{IIvb}}\left(c_{s0}+c_{s1}\left(\phi\frac{F_0}{N}-k(t-1)\right)\right) \\ &= C_{sim}(NB) + c_{s0}\left(\frac{\lambda N}{(\lambda N)^r}-1\right)\log_a(1-p_c) \\ &\quad + \frac{c_{s1}F_0}{N}\frac{p_c}{1-a^{(\lambda N)^r}}\left(\lambda N-(\lambda N)^r\right)a^{(\lambda N)^r}, \end{aligned} \tag{28}$$

where

$$\begin{aligned} C_{sim}(NB) &= c_{s0}\left(\log_a\phi + \frac{\phi F_0}{kN}\right) \\ &\quad + \frac{c_{s1}F_0}{N}\left(\frac{1-\phi}{1-a}+\frac{\phi}{2}\left(1+\frac{\phi F_0}{kN}\right)\right). \end{aligned} \tag{29}$$

From Eq. (28) we can see that the total FS cost in the VB system is larger than that in the NB system.

In the FB system, while the fault coverage is less than p_c, FS is performed for all the unprocessed faults as shown Eq. (12). Therefore, the number of faults simulated between the $(b-1)$-th and b-th broadcastings is expressed as

$$\begin{aligned} n_{ffb}(b) &= \sum_{t=1}^{\lambda}F_0a^{(\lambda N)^r(b-1)+(t-1)} \\ &= \frac{1-a^{\lambda}}{1-a}F_0a^{(\lambda N)^r(b-1)}. \end{aligned} \tag{30}$$

After the completion of broadcasting, each processor in the FB system performs test generation as well as that in the VB system. Hence, in the same way as the VB system, the total FS cost of can be expressed as

$$\begin{aligned} C_{sim}(FB) &= C_{sim}(NB) \\ &\quad - c_{s0}\left(1-\frac{\lambda}{(\lambda N)^r}\right)\log_a(1-p_c) \\ &\quad + \frac{c_{s1}p_cF_0}{1-a}\left(\frac{1-a^{\lambda}}{1-a^{(\lambda N)^r}}-\frac{1}{N}\right). \end{aligned} \tag{31}$$

3.2.4 Communication Cost

In the VB and FB systems, each processor communicates with $(N-1)$ processors for each broadcasting. On the other hand, since all processors are connected via a single communication bus, the communication competition such that while a processor communicates with another processor, the others cannot communicate will occur. The competition will increases as the number of processors increases. Hence, the total costs of communication in the VB and FB systems are expressed as

$$C_{com}(VB) = \sum_{b=1}^{B_{vb}} c_{cvb}(b)N(N-1) \tag{32}$$

$$C_{com}(FB) = \sum_{b=1}^{B_{fb}} c_{cfb}(b)N(N-1) \tag{33}$$

where $c_{cvb}(b)$ and $c_{cfb}(b)$ are the costs of communication with a processor at the b-th broadcasting of each processor in the VB and FB systems, respectively. In the VB system, λ test vectors are broadcasted at each broadcasting, and hence we have

$$c_{cvb}(b) = c_{cvb0} + c_{cvb1}\lambda \tag{34}$$

where c_{cvb0} and c_{cvb1} are constants. In the FB system, the faults processed newly between the $(b-1)$-th and the b-th broadcasting are broadcasted at the b-th broadcasting. From Eq. (12), the number of faults processed newly until the b-th broadcasting after the $(b-1)$-th one can be expressed as

$$F_0 a^{(\lambda N)^r(b-1)} - F_0 a^{(\lambda N)^r(b-1)+\lambda}. \tag{35}$$

Hence, we have

$$c_{cfb}(b) = c_{cfb0} + c_{cfb1}(1-a^{\lambda})F_0 a^{(\lambda N)^r(b-1)} \tag{36}$$

where c_{cfb0} and c_{cfb1} are constants. From the above equations, the total costs of communication in the VB and FB systems can be given by

$$\begin{aligned} C_{com}(VB) &= (c_{cvb0} + c_{cvb1}\lambda) \\ &\quad \times \frac{\log_a(1-p_c)}{(\lambda N)^r} N(N-1) \end{aligned} \tag{37}$$

$$\begin{aligned} C_{com}(FB) &= \left(c_{cfb0}\frac{\log_a(1-p_c)}{(\lambda N)^r} \right. \\ &\quad \left. + c_{cfb1}\frac{p_c F_0}{1-a^{(\lambda N)^r}} \right) N(N-1). \end{aligned} \tag{38}$$

From these equations, the total cost of fault broadcasting is smaller than that of vector broadcasting when

$$\frac{\log_a(1-p_c)}{(\lambda N)^r} > \frac{c_{cfb1}\dfrac{p_c F_0}{1-a^{(\lambda N)^r}}(1-a^{\lambda})}{c_{cvb0} + c_{cvb1}\lambda - c_{cfb0}}. \tag{39}$$

Figure 4 shows the cost of the b-th broadcasting. As shown in this figure, $c_{cvb}(b)$ is constant and $c_{cfb}(b)$ decreases as the number of broadcasts b increases. Hence, if the two curves obtained from Eqs. (34) and (36) make a cross point as shown in Fig. 4, i.e., if

$$\begin{aligned} c_{fv0} + c_{fv1}F_0(1-a^{\lambda})a^{-(\lambda N)^r} &< c_{cvb0} + c_{cvb1}\lambda \\ &< c_{cfb0} + c_{cfb1}F_0(1-a^{\lambda}), \end{aligned} \tag{40}$$

then we can obtain a smaller cost of broadcasting by switching the communication data from test vectors to faults. Furthermore, if we switch the communication data when the number of broadcasts b reaches B_s:

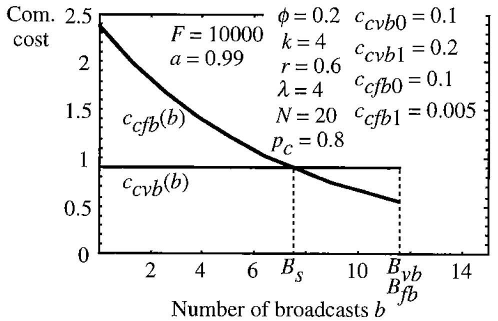
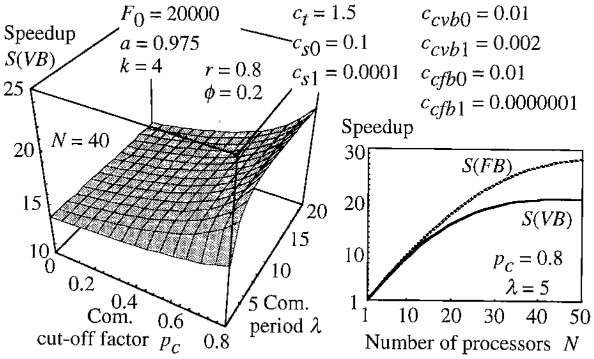


Fig. 4 Cost of each communication.



(a) vs. com. cut-off factor and com. period

(b) vs. number of processors

Fig. 5 Speedups of VB and FB systems.

$$B_s = \frac{1}{(\lambda N)^r}\log_a\left(\frac{c_{cvb0} + c_{cvb1}\lambda - c_{cfb0}}{c_{cfb1}F_0(1-a^{\lambda})}\right) + 1, \tag{41}$$

then we can obtain the minimum communication cost.

3.2.5 Speedup

The speedups of the VB and FB systems are expressed as

$$S(VB) = \frac{C_{total}(U)}{C_{test}(VB) + C_{sim}(VB) + C_{com}(VB)} \tag{42}$$

$$S(FB) = \frac{C_{total}(U)}{C_{test}(FB) + C_{sim}(FB) + C_{com}(FB)}. \tag{43}$$

Figure 5 (a) shows the speedup versus the communication cut-off factor and the communication period in the VB system. From this figure we can see that the speedup increases as the communication cut-off factor increases due to the decrease of the TG cost. Moreover, we can see that as the communication period increases, the speedup also increases, due to the decrease of the communication cost. Figure 5 (b) presents the speedup versus the number of processors in the VB and FB systems. From this figure we can see that the speedup increases as the number of processors increases. However, we can also see that the increase of the speedup is degraded by the increase of the communication cost when the number of processors is large. This figure shows a case that the total cost of test generation in the FB system is smaller than that in the VB system, and hence the speedup of the FB system is larger than that of the VB system.

4. Experimental Results

The VB and FB systems were implemented on a network (Ethernet) of workstations (DEC5000/25's), where the FAN [9] algorithm was used as a test vector generator for each processor. We made experiments using five benchmark circuits of ISCAS'85 [6] combinational circuits and the combinational parts of ISCAS'89 [7] sequential circuits. Here we present part of the results due to the limitation of space. Experimental results show similar tendencies for all circuits.

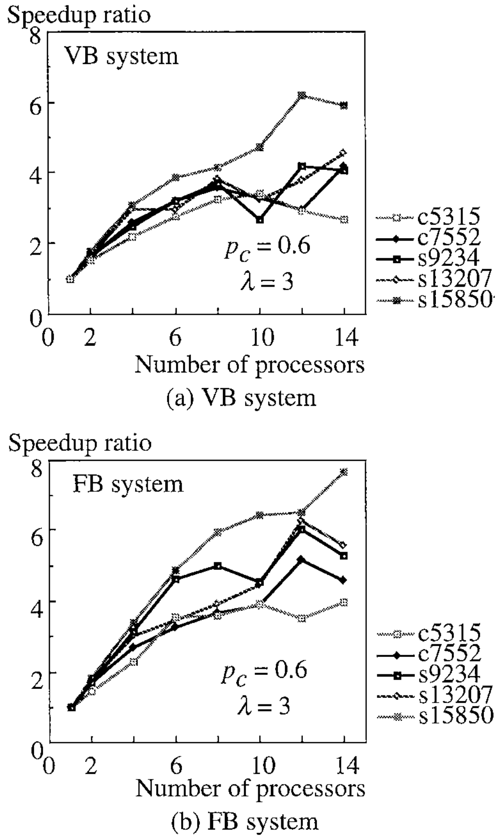
Number of Test Vectors

Figure 6 (a) shows the total number of test vectors generated for s15850 in the VB system. From this figure we can see that the total number of generated test vectors increases as the number of processors and the communication period increase, and as the communication period decreases. Figure 6 (b) shows the total numbers of test vectors generated for s15850 in the the VB and FB systems. From this figure we can see that the total number of test vectors generated in the VB system is almost the same as that in the FB system, and that the total number of generated test vectors can be reduced by broadcasting. These results coincide closely with the analysis in Sect. 3.

Speedup

Figure 7 presents the speedups of the VB and FB systems. From this figure we can see that the speedups increase as the number of processors increases for all circuits. By comparing the curves in Fig. 7 (a) with those in 7 (b), we can see that the speedup of the FB systems is larger than that of the VB systems for all circuits. This is because the total FS cost in the VB system is larger than that in the FB system.

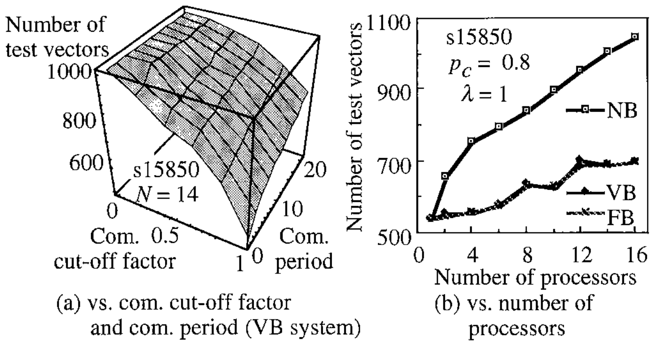
Figure 8 shows the speedup versus the communication cut-off factor and the communication period in the FB system. From this figure we can see that the speedup increases as the communication cut-off factor increases due to the reduction of the TG cost, and as the communication period increases due to the reduction of the communication cost. However, we can also see that when the communication cut-off factor is large, the speedup decreases, and hence there exists the optimal communication period which maximizes the speedup (in this case, the maximum speedup is 8.35 when the communication cut-off factor is 0.8 and the communication period is 11). This result does not coincide with the analytical result in Sect. 3 that the speedup increases monotonically as the communication cut-off



(a) VB system

(b) FB system

Fig. 7 Experimental results: Speedups of VB and FB systems.



(a) vs. com. cut-off factor and com. period (VB system)

(b) vs. number of processors

Fig. 6 Experimental results for s15850: Number of test vectors.

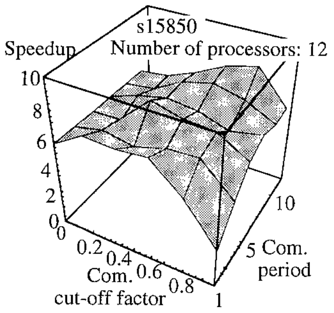
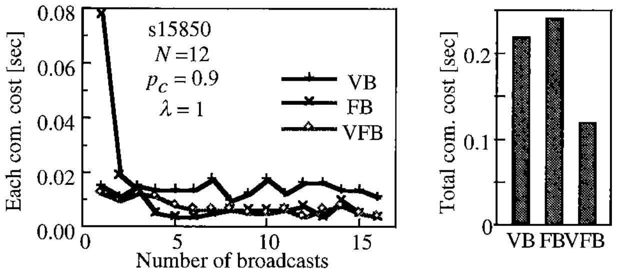


Fig. 8 Experimental results for s15850: Speedup of FB system.



(a) Each com. cost (b) Total com. cost

Fig. 9 Experimental results for s15850: Communication cost.

factor increases. This may be because we analyzed the performance provided that broadcasting is performed only during in Phase I. Hence, we can consider that broadcasting in Phase II is not effective in reducing the TG cost. Moreover, we can also consider that the relationship between the number of test vectors and the number of unprocessed faults for actual circuits is not clearly partitioned into Phase I and Phase II.

Communication Cost

Figure 9 (a) illustrates the cost of each communication between two processors versus the number of broadcasts for s15850. From this figure we can see that the shapes of the curves are similar to those in Fig. 4 obtained from the analysis under the theoretical models. Furthermore, we can see that there exists a cross point of these curves when the number of broadcasts is 3, and hence we also implemented the VFB system in which the broadcasted data are switched from vectors to faults. Figure 9 (b) shows the total costs of communication in the VB, FB and VFB systems. As shown in this table, switching broadcasted data from vectors to faults is effective in the reduction of the communication cost.

5. Conclusions

In this paper, we analyzed the performance of parallel test generation for combinational circuits in the *vector broadcasting (VB)* and *fault broadcasting (FB)* systems where the *communication cut-off factor* and the *communication period* were applied. We also presented experimental results on the VB and FB systems implemented on a network of workstations using ISCAS '85 and ISCAS '89 benchmark circuits. The analytical and experimental results show that the total number of test vectors generated in the VB system is the same as that in the FB system, the speedup of the FB system is larger than that of the VB, and it is effective in reducing the communication cost to switch broadcasted data from vectors to faults. As future work, we will apply the results of this work to the parallel test generation for sequential circuits.

Acknowledgement

The authors would like to express their cordial thanks to Dr. Toshimitsu Masuzawa of Nara Institute of Science and Technology for his fruitful discussions and comments. 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] R.H. Klenke, R.D. Williams, and J.H. Aylor, "Parallel-processing techniques for automatic test pattern generation," IEEE Comput., pp.71–84, Jan. 1992.

[2] H. Fujiwara and T. Inoue, "Optimal granularity of test generation in a distributed system," IEEE Trans. Comput.-Aided Des. Intergrated Circuits & Syst., vol.9, no.8, pp.885–892, Aug. 1990.

[3] H. Fujiwara and T. Inoue, "Optimal granularity and scheme of parallel test generation in a distributed system," IEEE Trans. Parallel & Distributed Syst., vol.6, no.7, pp.677–686, July 1995.

[4] S. Patil and P. Banerjee, "Performance trade-offs in a parallel test generation/fault simulation environment," IEEE Trans. Comput.-Aided Des. Intergrated Circuits & Syst., vol.10, no.12, pp.1542–1558, Dec. 1991.

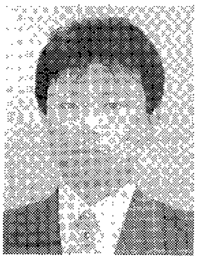
[5] R.H. Klenke, L. Kaufman, J.H. Aylor, R. Waxman, and P. Narayan, "Workstation based parallel test generation," Proc. 1993 Int'l Test Conf., pp.419–428, 1993.

[6] F. Brglez and H. Fujiwara, "A neutral netlist of ten combinational benchmark circuits and target translator in FORTRAN," Proc. Int'l Symp. on Circuits & Syst., June 1985.

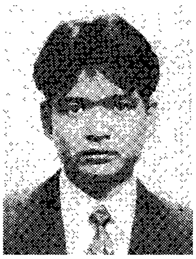
[7] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of seqential benchmark circuits," Proc. Int'l Symp. on Circuits & Syst., pp.1929–1934, 1989.

[8] P. Goel, "Test generation cost analysis and projections," Proc. 17th Ann. Design Automation Conf., pp.77–84, 1980.

[9] H. Fujiwara and T. Shimono, "On the acceleration of test pattern generation algorithms," IEEE Trans. Comput., vol.C-32, pp.1137–1144, Dec. 1983.



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 and the Information Processing Society of Japan.



Takaharu Fujii 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 NEC Corporation, Kanagawa, Japan, and engages in development of microprocessors for multimedia applications. His resarch interests include processor architecture, test generation and parallel processing.

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.