IPSJ Transactions on System LSI Design Methodology Vol. 4 140–149 (Aug. 2011)

Regular Paper

Scan FF Reordering for Test Volume Reduction in Chiba-scan Architecture

Kiyonori Matsumoto,[†1,*1] Kazuteru Namba[†1] and Hideo Ito[†1]

Scan architecture is one of designs for tests (DFTs). In scan architecture, some or all of flip-flops (FFs) in a circuit are serially connected and form a scan chain. The Chiba-scan is one of scan architectures facilitating delay testing. The Chiba-scan has many advantages such as small area overhead comparable to that of the standard scan architecture and complete fault coverage for robust testable path delay faults. However, its test volume is much larger than that of other scan architectures. This paper presents a test volume reduction method for robust path delay fault testing on the Chiba-scan. In this method, scan FFs are reordered. The experimental results give evidence that the proposed method reduces the number of test vectors by 18.4% for ISCAS89 benchmark circuits. Furthermore, the proposed method enables testing with 16.8% shorter test application time (TAT) and 18.3% lower required memory size for automatic test equipment (ATE) compared with those for the enhanced scan architecture.

1. Introduction

In modern VLSIs, delay fault testing is becoming of increasing importance with their increasing size and density [1),2)]. To detect delay faults, two-pattern testing is essential. In two-pattern testing, two test vectors, called the *initial vector* and *final vector*, are applied to circuits under test (CUTs) in two consecutive clock cycles. There are three well-known scan testing for two-pattern testing: Launch on Capture (LOC, also known as broad-side testing) [2),3)], Launch on Shift (LOS, also known as skewed-load testing) [2),4)] and enhanced scan testing [2),5)]. Although LOC and LOS use a standard scan architecture with a small area overhead, they do not give complete fault coverage for robust testable path delay faults. In contrast, enhanced scan testing uses an enhanced scan architecture and gives complete fault coverage. However, it requires a large area because an extra latch is added to every scan flip-flop (FF). Thus, it is rarely used.

To overcome these shortcomings, a scan architecture facilitating delay fault testing, called the *Chiba-scan architecture* [6)], was proposed. The Chiba-scan architecture allows arbitrary single-input-change two-pattern testing, which can provide robust path delay fault testing with complete fault coverage like enhanced scan architecture [⋆1]. Besides, it requires an area overhead almost as small as the standard scan architecture. However, it requires a large test volume to achieve high fault coverage. Thus, the Chiba-scan architecture requires a long test application time (TAT) and automatic test equipment (ATE) with a large memory, resulting in a high cost.

This paper presents a test volume reduction method for robust path delay fault testing on the Chiba-scan architecture. In the Chiba-scan architecture, scan FFs are connected to each other in a daisy chain fashion similarly to in standard and enhanced scan architecture. While the order of FFs does not affect robust path delay fault test volume in the enhanced scan architecture, it affects it in the Chiba-scan architecture. The proposed method reduces the test volume by reordering the FFs. The proposed method does not completely fix the order of the FFs, but it partitions the FFs into two groups and gives constraints about adjacency of the FFs on a scan chain.

This paper consists as follows: Section 2 explains the basic architecture and operation of the Chiba-scan. Section 3 discusses the reason why the test volume changes with the order of FFs in the Chiba-scan architecture. The proposed test volume reduction method is explained in Section 4, and its evaluation along with experimental results is given in Section 5. Finally, Section 6 concludes this paper.

2. Preliminaries

2.1 Chiba-scan

The Chiba-scan is one of the scan architectures facilitating two-pattern test-

†1 Chiba University
∗1 Presently with Canon Inc.

⋆1 The Chiba-scan architecture does not allow arbitrary multiple-input-change two-pattern testing. Thus, it can not always detect some delay faults we can detect only multiple-input-change two-pattern testing, such as some functional sensitizable path delay faults.

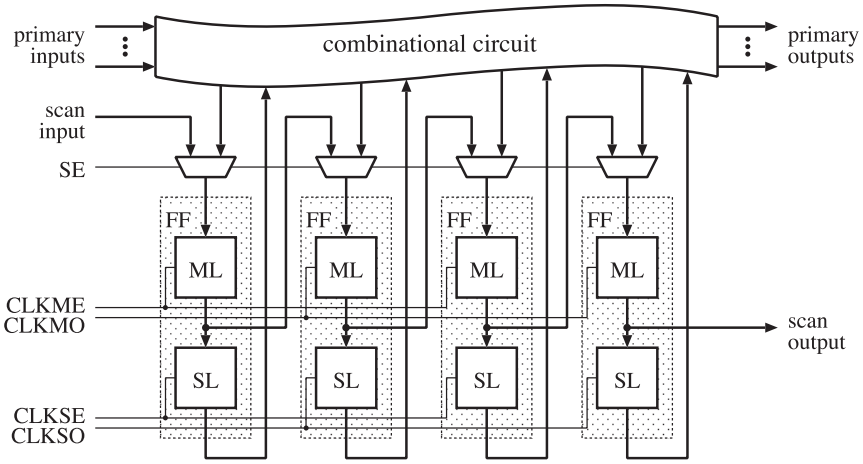

Fig. 1 Chiba-scan architecture.

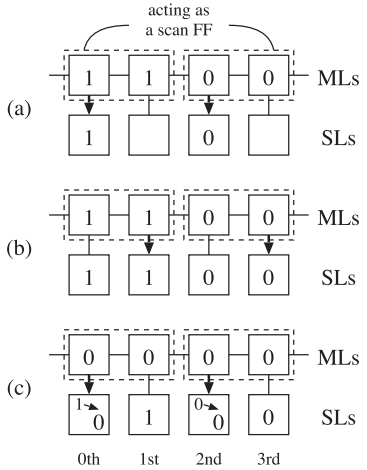

Fig. 2 Example of test vector application in Chiba-scan.

ing[6]. Its advantages are as high fault coverage as enhanced scan testing for robust path delay fault testing and as small area as standard scan architecture. **Figure 1** illustrates an example of a Chiba-scan architecture that consists of four scan FFs. These FFs are based on master-slave FFs. Every FF contains master and slave latches, which are represented by ML and SL, respectively. This scan architecture is based on a muxed-D type scan architecture, with a scan enable signal SE. This architecture has four clocks: CLKME, CLKMO, CLKSE and CLKSO.

The main characteristic of the Chiba-scan is the scan output of each scan FF. The scan outputs of Chiba-scan FFs are the outputs of master latches while those of standard scan FFs are the outputs of slave latches. In the Chiba-scan, every pair of adjacent master latches acts as a master-slave FF and becomes a part of a scan chain. In the Chiba-scan, test vector pairs (pairs of initial and final vectors) are applied into a CUT and their test responses are observed as follows:

(1) Select the scan inputs as the inputs of master latches, and shift-in the even-numbered bits of an initial vector by enabling SE and controlling CLKME and CLKMO, i.e., by supplying a clock signal and its inverted clock signal to CLKME and CLKMO, respectively.
(2) Set CLKSE to 1, and then reset CLKSE. (The even-numbered bits are copied from master latches, which acted as a scan chain in Step (1), to slave latches of the even-numbered FFs.)
(3) Scan-in the odd-numbered bits of the initial vector into the slave latches of the odd-numbered FFs in the same way as in Steps (1) and (2).
(4) Shift-in the even-(odd-) numbered bits of the final vector.
(5) Disable SE and rise CLKSE (CLKSO). (The even-(odd-) numbered bits of the final vector are launched.)
(6) Rise CLKME (CLKMO). (The Chiba-scan starts capturing the even-(odd-) numbered bits of the test response.)
(7) Fall CLKME (CLKMO). (The even-(odd-) numbered bits of the test response are captured into the corresponding master latches.)
(8) Shift-out and observe the captured test response by enabling SE and controlling CLKME and CLKMO.
(9) Go back to Step (1) until all test vector pairs have been applied and all test responses have been observed.

Figure 2 gives an example of applying a test vector pair. The initial and final vectors of the pair are (1,1,0,0) and (0,1,0,0), respectively. First, the even-

numbered bits of the initial vector (1,0) are shifted into the master latches. They are then copied into the slave latches as shown in Fig. 2 (a). Subsequently, in a similar way, the odd-numbered bits of the initial vector (1,0) are transferred to the slave latches of the corresponding FFs as shown in Fig. 2 (b). Finally, to apply a transition, the final vector (0,1,0,0) is launched as follows: In this example, the only different bit between the initial and final vectors is the zeroth bit, i.e., an even-numbered bit. In other words, the odd-numbered bits of the final vector are equal to those of the initial vector. Thus, by changing the values of the even-numbered bits, we can apply the final vector. In the same way as that for the even-numbered bits of the initial vector, those of the final vector (0,0) are launched as shown in Fig. 2 (c).

It is important to note that in Chiba-scan testing, only either even- or odd-numbered bits of final vectors can be applied, and the remaining bits must be equal to the corresponding bits in the initial vectors. Even if we forcibly apply transitions at both even- and odd-numbered bits, undesired transitions can occur, as described in Section 3 in detail. In addition, in Chiba-scan testing, only either odd- or even-numbered bits of test responses are observed. Thus, we cannot use test vector pairs for enhanced scan testing on the Chiba-scan architecture. For Chiba-scan testing, we have to generate and compact test vector pairs considering these restrictions and using a special automatic test pattern generator (ATPG).

2.2 Nine-valued Logic

Nine-valued logic is usually used in delay fault test generation [7], and consists of nine symbols: S0, S1, T0, T1, H0, H1, U0, U1 and XX. Each symbol represents a value pair (a pair of values in a vector pair). The definitions of these nine symbols are shown in **Table 1**. For every hazard possible value, the initial value is equal to the final value. However, unlike stable values, it is possible that a hazard occurs and the value is inverted momentarily. If such an inversion occurs at an input of a FF at the same time as a clock transition, the inverted value is incorrectly stored in the FF.

Table 1 Nine-valued logic.

S0/S1	stable 0/1 value
T0/T1	transition 1→0 / 0→1
H0/H1	hazard possible 0/1 value
U0/U1	S0, T0 or H0 / S1, T1 or H1
XX	don't care

3. Variation of Test Volume with Scan FF Order

In the Chiba-scan architecture, unlike in enhanced scan architectures, the test volume for robust testable path delay faults is affected by the order of scan FFs forming the daisy chain. The reason for this is explained in the rest of this section.

Usually, to reduce vector volume, i.e., the number of vector pairs, we test multiple faults with a single vector pair, which we can obtain by compacting multiple vector pairs for these faults by filling don't care bits. Below, an issue related to test compaction in the Chiba-scan is discussed using the example test vector pair shown in **Fig. 3**. In this example, transitions occur at the first, third and fifth bits, indicated with rectangles. An ATPG for Chiba-scan generates this vector pair by targeting at least three path delay faults, the inputs of which are connected to the first, third and fifth FFs. **Figure 4** illustrates the state in

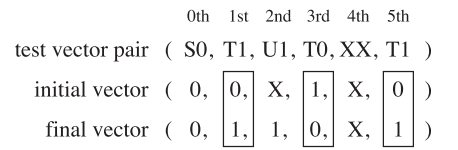

Fig. 3 Example of vector pair acceptable in Chiba-scan.

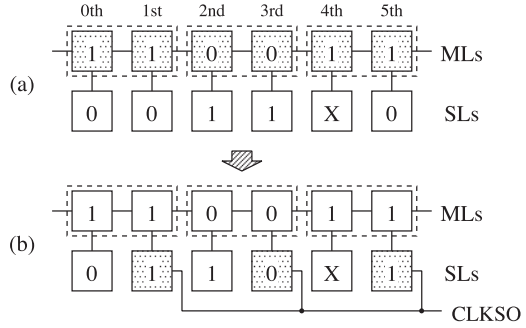

Fig. 4 Final vector application for vector pair in Fig. 3.

	0th	1st	2nd	3rd	4th	5th
test vector pair	(T1,	S0,	U1,	T0,	T1,	XX)
initial vector	(0,	0,	X,	1,	0,	X)
final vector	(1,	0,	1,	0,	1,	X)

Fig. 5 Example of vector pair not acceptable in Chiba-scan.

the Chiba-scan before and after the final vector application. In this figure, the vector pair shown in Fig. 3 is used. In Fig. 4 (a), the initial vector is stored in the slave latches and the odd-numbered bits of the final vector, (1,0,1), are stored in the master latches. Since adjacent master latches, such as those of the zeroth and first FFs, act as a FF, the values of the master latches of the odd-numbered FFs must be equal to those of the adjacent even-numbered FFs. As shown in Fig. 4 (b), by setting CLKSO to 1 in Step (5) of Chiba-scan testing described in Section 2, we can copy the odd-numbered bits of the final vector into their corresponding slave latches to launch transitions.

Secondly, we consider the vector pair shown in **Fig. 5**, which we can obtain by changing the order of the bits in the vector pair shown in Fig. 3, specifically, by swapping the zeroth and first bits, and the fourth and fifth bits. In this vector pair, transition values, T0 and T1, are at the zeroth, third and fourth bits, which indicates that this pair targets path delay faults on paths whose inputs are connected to the zeroth, third and fourth FFs. An example of final vector application of this test vector is illustrated in **Fig. 6**. In this figure, shaded latches have been changed from the initial vector to the final vector. Initially, the values of CLKSE and CLKSO are 0's. By rising CLKSE, we can launch transitions at the zeroth and fourth bits as shown in Fig. 6 (b). However, in this case, we cannot launch a transition at the third bit, because the corresponding clock CLKSO is kept at 0. On the other hand, as illustrated in Fig. 6 (c), by setting CLKSO to 1, although we can launch a transition at the third bit, we cannot at the zeroth and fourth bits. By rising both CLKME and CLKMO, we can launch transitions at all three bits. However, in this case, as shown in Fig. 6 (d), undesired final values are launched at the first and second bits. This is because the values of the master latches of adjacent FFs must be the same as each other. The value of the master latch of the first FF is equal to that of the zeroth FF, and their values

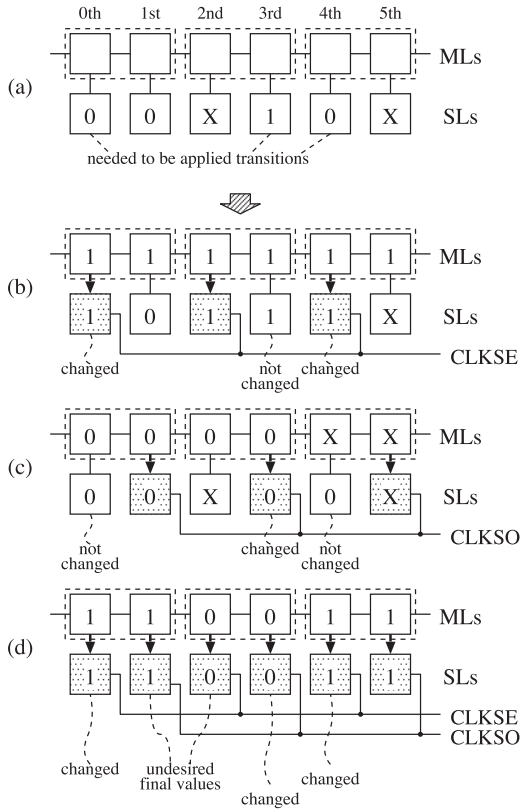

Fig. 6 Final vector application for vector pair in Fig. 5.

are 1's in order to launch the transition 0→1 at the zeroth bit. Upon the rising of CLKSO, at the first FF, the logic 1 is launched, and thus the value of the slave latch changes from 0 to 1. In sum, the vector pair shown in Fig. 5 cannot be launched in the Chiba-scan, and thus ATPGs for Chiba-scan are not allowed to generate this pair. We cannot test the faults on paths whose inputs are the zeroth, third and fourth FFs with a single test vector pair.

In addition, the faults on the paths whose outputs are connected to the even-numbered FFs and to the odd-numbered FFs cannot be tested simultaneously.

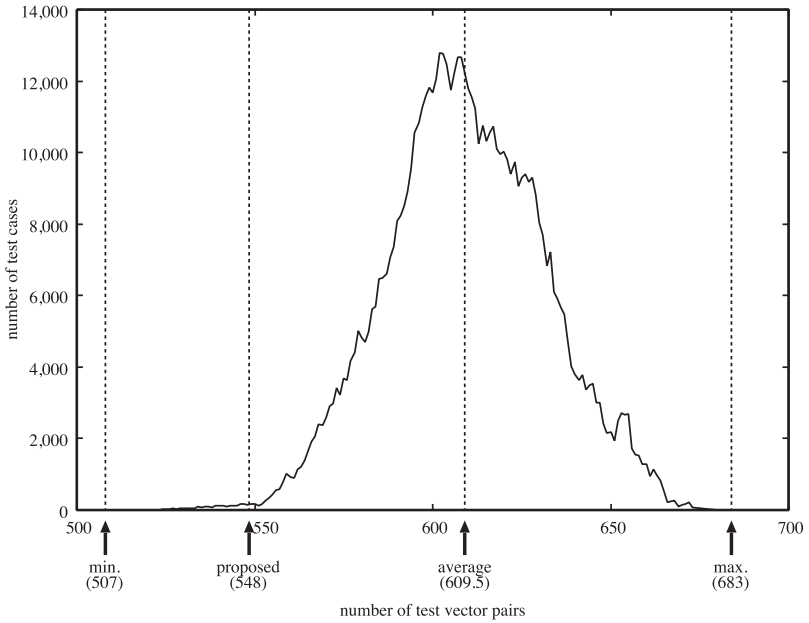

Fig. 7 Number of vector pairs for Chiba-scan randomly reordered for s953.

This is because the even-numbered and odd-numbered bits of test responses cannot be shifted-out and observed at the same time.

As explained above, in the Chiba-scan architecture, the order of scan FFs affects the test volume. **Figure 7** shows an evaluation result illustrating this fact. This evaluation uses the s953 ISCAS89 benchmark circuit and generates 664,344 test cases. For every test case, scan FFs are randomly reordered, and a set of test vector pairs for robust path delay fault testing [⋆1] are generated. The y-axis shows the number of test cases, and the corresponding number of test vector pairs is indicated on the x-axis. For example, the ATPG generates 600 test vector pairs in 11,680 out of 664,344 cases. The dashed line labeled "proposed" will be explained later. The maximal number of test vector pairs is 683 while the minimal number is 507, a difference of 176. This fact indicates that we can reduce the test volume by changing the order of FFs.

⋆1 Target faults are all robust testable path delay faults except ones that are not detected because of time-out on the ATPG.

4. Scan FF Reordering Method

4.1 Overview of Proposed Method

In the proposed method, scan FFs are reordered to reduce the number of vector pairs. The proposed method uses iterative optimization algorithms. However, the number of vector pairs cannot be used in iterative algorithms practically because its calculation requires test generation with very long time. Instead of this number, the proposed method uses a parameter S defined in Section 4.2, the large number of which roughly means a small number of vector pairs. This definition uses *compactability*. If two vector pairs can be compacted, the pairs are compactable. Compactability is explained in detail in Section 4.3.

The outline of the proposed method is as follows: First, for every path delay fault, a vector pair is generated with an ordinary ATPG for the enhanced scan testing. Every generated vector pair targets only a single fault and includes some unspecified values. It is noted that the generated vector pairs are only used in the proposed reordering method and not used in manufacturing testing. Second, compactabilities are calculated for any combination of generated vector pairs. Third, we reorder FFs maximizing the parameter S. Finally, vector pairs used in manufacturing testing are generated with an ATPG for the Chiba scan testing.

4.2 FF Reordering Algorithm

As described above, in the proposed method scan FFs are reordered according to compactabilities. The FF reordering algorithm for CUTs with n scan FFs, FF_i $(0 \leq i < n)$, is described below:

(1) Let M be an n-dimensional square matrix, and initialize all elements in M with 0.

(2) Generate vector pairs P_j for every targeted robust testable path delay fault f_j $(0 \leq j < \mathrm{NOF})$, where NOF is the number of path delay faults.

(3) Examine compactabilities for all combinations of P_{j_0} and P_{j_1} such that $j_0 < j_1$. If the vector pairs P_{j_0} and P_{j_1} is compactable, perform Steps (3-a) and (3-b):

(3-a) Let FF_{i_0} and FF_{i_1} be FFs connected to the inputs of the paths where

f_{j_0} and f_{j_1} occur, respectively. Increment the value of M_{i_0,i_1} in the matrix M by 1.

(3-b) Let $\mathrm{FF}_{i'_0}$ and $\mathrm{FF}_{i'_1}$ be FFs connected to the outputs of the paths where f_{j_0} and f_{j_1} occur, respectively. Increment the value of $M_{i'_0,i'_1}$ by 1.

(4) Let FFe and FFo be sets of FFs initialized with even- and odd-numbered FFs, respectively. Let S be a parameter such that

$$S = \left(\sum_{\mathrm{FF}_{i_0},\mathrm{FF}_{i_1} \in \mathrm{FFe}} M_{i_0,i_1} \right) + \left(\sum_{\mathrm{FF}_{i_0},\mathrm{FF}_{i_1} \in \mathrm{FFo}} M_{i_0,i_1} \right).$$

(5) Swap a FF in FFe and one in FFo to increase S. Repeat Step (4) to maximize S (or until time out).

Finally, the even- and odd-numbered FFs in the reordered scan chain appear in FFe and FFo, respectively.

4.3 Compactability of Vector Pairs

Here, we give the definition of compactabilities used in the proposed method. This definition uses *compatibility*, the definition of which is as follows: If and only if two vector pairs are compatible, we can make them the same by filling unspecified values in them. Let P_0 and P_1 be vector pairs that the ATPG generates targeting path delay faults occurring on paths p_0 and p_1. Two vector pairs P_0, P_1 are compactable if and only if they satisfy the following two conditions:

(1) The pairs P_0 and P_1 are compatible.

(2) The FFs connected to the outputs of the paths p_0 and p_1 differ from each other, or the values at the outputs of p_0 and p_1 are the same.

Figure 8 shows an example of two vector pairs. Figure 8 (a) illustrates a compactable vector pair, and (b), (c) show not compactable pairs. This figure also shows the FF number connected to the output of the path where each targeted fault occurs, and its observed transition value.

In Fig. 8 (a), the vector pairs are compatible because we can make both pairs into the same pair (T1,S0,T0,S1) by filling unspecified values. In addition, the FFs connected to the outputs, i.e., the second and fourth FFs, differ from each other. Thus, the above two conditions are satisfied. In Fig. 8 (b), the vector pairs are not compatible because the shaded values differ from each other and we cannot make the pairs the same. In Fig. 8 (c), the vector pairs are compatible.

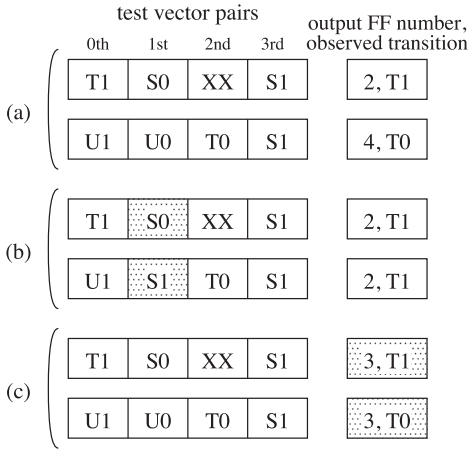

Fig. 8 Examples of vector pairs: (a) compactable, (b) (c) not compactable (shaded parts are in conflict).

However, the FFs connected to the outputs are identical, and their values differ. Therefore, the vector pairs are not compactable.

If many combinations of vector pairs are compactable, many vector pairs can be compacted and the number of vector pairs becomes small. As already discussed, in the Chiba-scan, faults on paths whose inputs / outputs are connected to even- and odd-numbered FFs cannot be tested simultaneously. Thus, for faults tested by compactable vector pairs, if possible, we should connect the inputs / outputs of the paths to FFs with numbers of the same parity, i.e., both even-numbered FFs or both odd-numbered FFs.

5. Evaluation

This section gives evaluation results using ISCAS89 benchmark circuits with a single Chiba-scan chain. The test generation targeted all robust testable path delay faults but some faults whose test vector pairs were not generated due to a time-out on the ATPG.

Table 2 shows the number of vector pairs for the enhanced scan architecture, the Chiba-scan architecture without the proposed method and that with the proposed method. Every test vector pair has been compacted to detect several

Table 2 Number of vector pairs for enhanced scan, original Chiba-scan and Chiba-scan with proposed method.

circuit					number of vector pairs			PM/ES	PM/CS
name	#faults	#FFs	#PI	#PO	ES	CS	PM	(%)	(%)
s838.1	3,428	32	34	1	1,767	1,831	1,798	101.8	98.2
s953	2,302	29	16	23	426	680	548	128.6	80.6
s1196	3,581	18	14	14	679	689	685	100.9	99.4
s1423	28,696	74	17	5	5,164	6,502	6,138	118.9	94.4
s5378	18,656	179	35	49	970	1,171	1,144	117.9	97.7
s9234.1	21,389	211	36	39	1,553	2,656	1,672	107.7	63.0
s13207	27,603	669	31	121	2,613	3,648	2,648	101.3	72.6
s35932	21,783	1,728	35	320	74	140	75	101.4	53.6
s38584.1	92,239	1,426	38	304	2,192	3,376	2,541	115.9	75.3
average								110.5	81.6

#faults : number of robust testable path delay faults.
#FF : number of FFs in the CUT.
#PI : number of primary inputs.
#PO : number of primary outputs.
ES : results for enhanced scan.
CS : results for Chiba-scan without the proposed reordering method.
PM : results for Chiba-scan with the proposed reordering method.
PM/ES, PM/CS : ratio of results.

faults using a single test vector.

In the CUTs without the proposed reordering, every scan-FF is connected in the order of occurrence in the original benchmark lists. The numbers of vector pairs using the proposed method are on average 18.4% and up to 41.4% lower than those without the proposed method. However, they are about 10.5% greater than those for the enhanced scan architecture. This is because the Chiba-scan architecture only allows vector pairs such that transition can appear at only bits with numbers of the same parity, while the enhanced scan architecture allows arbitrary vector pairs.

From Fig. 7 in Section 3, we find that the proposed method effectively reduces the test volume. From Table 2, the number of test vector pairs for the proposed method is 548, which is indicated by the line labeled "proposed" in Fig. 7. The cumulative relative frequency of the test data using the proposed method is 0.0033. This means the following: Assume that we want to obtain orders of FFs providing test volume as low as or lower than that with the proposed method. To obtain such an order by reordering FFs randomly, we have to generate about 300 orders and the corresponding test sets.

Table 3 Test application time (clock).

circuit	ES	CS	PM	PM/ES(%)	PM/CS(%)
s838.1	116,686	91,566	89,916	77.1	98.2
s953	25,618	30,955	24,949	97.4	80.6
s1196	25,838	19,990	19,874	76.9	99.4
s1423	774,748	734,763	693,631	89.5	94.4
s5378	349,558	316,845	309,542	88.6	97.7
s9234.1	658,894	846,042	532,638	80.8	63.0
s13207	3,502,758	3,668,399	2,662,899	76.0	72.6
s35932	259,348	364,024	195,414	75.3	53.7
s38584.1	6,258,820	7,228,729	5,440,994	86.9	75.3
average				83.2	81.6

Table 4 Required memory size in ATE (bit).

circuit	ES	CS	PM	PM/ES(%)	PM/CS(%)
s838.1	291,555	243,523	239,134	82.0	98.2
s953	60,492	76,840	61,924	102.4	80.6
s1196	65,184	53,742	53,430	82.0	99.4
s1423	1,347,804	1,215,874	1,147,806	85.2	94.4
s5378	636,320	558,567	545,688	85.8	97.7
s9234.1	1,155,432	1,415,648	891,176	77.1	63.0
s13207	5,722,470	5,548,608	4,027,608	70.4	72.6
s35932	412,476	538,440	288,450	69.9	53.6
s38584.1	10,210,336	10,911,232	8,212,512	80.4	75.3
average				81.7	81.6

Tables 3 and **4** show the TAT and the required memory size of ATE, respectively. The TATs (clock cycles) for the enhanced scan and Chiba-scan, $\mathrm{TAT_{ES}}$ and $\mathrm{TAT_{CS}}$, can be calculated by the following equations [6]:

$$\mathrm{TAT_{ES}} = 2mn + 2m + 2n,$$
$$\mathrm{TAT_{CS}} = \frac{3}{2}mn + 2m + \frac{n}{2},$$

where m is the number of vector pairs and n is the number of FFs. The corresponding required ATE memory sizes, $\mathrm{MEM_{ES}}$ and $\mathrm{MEM_{CS}}$, are given by the following [6]:

$$\mathrm{MEM_{ES}} = m\left(3n + 2n_{\mathrm{i}} + n_{\mathrm{o}}\right),$$
$$\mathrm{MEM_{CS}} = m\left(2n + 2n_{\mathrm{i}} + n_{\mathrm{o}}\right),$$

where n_{i} and n_{o} are the numbers of primary inputs and primary outputs, respec-

Table 5 Calculation time for reordering and test generation.

circuit	caluculation time (second) CS	PM	time increase (%)
s838.1	3	8	166.7
s953	1	4	300.0
s1196	3	8	166.7
s1423	79	151	91.1
s5378	52	176	238.5
s9234.1	295	462	56.6
s13207	2,599	3,246	24.9
s35932	296	1,078	264.2
s38584.1	5,750	22,326	288.3
average			177.4

Table 6 Calculation time with and without parallel computations.

circuit	calculation time (second) PM (1 core) re.	gen.	sum	PM (4 core) re.	gen.	sum	multicore effect (%)	time increase (%)
s838.1	4	4	8	1	4	5	62.5	66.7
s953	4	1	4	1	1	2	50.0	100.0
s1196	5	3	8	1	3	4	50.0	33.3
s1423	61	90	151	18	90	108	71.5	36.7
s5378	100	76	176	27	76	104	59.1	100.0
s9234.1	93	369	462	47	369	415	89.8	40.7
s13207	440	2,806	3,246	126	2,806	2,932	90.3	12.8
s35932	791	287	1,078	223	287	510	47.3	72.3
s38584.1	15,798	6,528	22,326	4,175	6,528	10,703	47.9	86.1
average							63.2	61.0

re.: reordering time, gen.: test generation time, sum: sum of re. and gen.

tively.

In the Chiba-scan, for every CUT, the ATE memory sizes is proportional to the number of vector pairs m and the TATs are almost proportional to m. Therefore, the relative decreases in the TATs and ATE memory sizes by the proposed reordering are nearly equal to that in the number of vector pairs, 18.4% on average. The TAT and the ATE memory size for the Chiba-scan with the proposed method are, on average, 16.8% and 18.3% smaller than those for the enhanced scan despite the large number of vector pairs. This is because the half of each final vector is not scanned-in and thus not stored in the ATE in the Chiba-scan unlike in the enhanced scan.

Table 5 shows the calculation time for the reordering of FFs (only for the proposed method) and for test generation. This experiment was performed on a machine with a Windows XP x86 operating system, an Intel Core 2 Quad Q8300 processor and 4 GB DDR2 RAM. The "time increase" column shows the relative time increase for the proposed reordering, given by

$$\text{time increase} = \frac{\text{PM}}{\text{CS}} - 1.$$

The proposed method requires a long calculation time, particularly for large CUTs, due to the reordering time. In fact, there are many path delay faults in large CUTs and, in the proposed reordering, we must calculate compactability for all combinations of path delay faults.

In the proposed reordering, almost all calculations can be made in parallel, and thus we can reduce calculation time by parallel computations. **Table 6** shows the calculation time with a single core and four cores, respectively. The "PM (1 core)" and "PM (4 core)" columns show the calculation times for Chiba-scan using the proposed reordering with a single core and four cores, respectively. The "multicore effect" column shows the ratio of the calculation time with four cores to that with a single core, given by

$$\text{multicore effect} = \frac{\text{PM (4 cores)}}{\text{PM (1 core)}}.$$

The "time increase" column shows the relative time increase for the proposed reordering with four cores, given by

$$\text{time increase} = \frac{\text{PM (4 cores)}}{\text{CS}} - 1.$$

By using parallel computations, we can reduce 36.8% of the calculation times, on average. When in use of parallel computations with four cores, the calculation time with the proposed method is 51.6% larger than that without reordering.

Table 7 summarizes factors behind area increases compared with a CUT without scan architecture. In this table, n denotes the number of FFs in the CUT. Every test architecture has checked factors. All scan architectures have n MUXs selecting between normal and scan inputs, and scan chains. Only the enhanced scan contains n extra latches. Chiba-scan architecture requires three extra clock signals. In other words, it has four clock signals while the other architectures

Table 7 Factors leading to area increases compared to a CUT with non-scan architecture.

test architecture	n MUXs	scan chain	n extra latches	3 extra clocks	complex scan routing
standard scan	√	√			
enhanced scan	√	√	√		
Chiba-scan w/o reordering	√	√		√	
Chiba-scan with reordering	√	√		√	√

have only one. The FF reordering can prolong the physical routing for a scan chain. The "complex scan routing" column indicates this fact, which is the only difference between the Chiba-scan architecture with and without the proposed reordering. However, the proposed method only determines whether every FF is even-numbered or odd-numbered, and does not determine absolute orders. Therefore, we can deduce that the routing is not complex and that its area overhead is not much larger than that without reordering.

6. Conclusion

In this paper, a test volume reduction method has been proposed for robust path delay fault testing on the Chiba-scan architecture. In the proposed method, FFs are reordered. The Chiba-scan with the proposed method requires a shorter TAT and a smaller memory size of ATE than the enhanced scan. In fact, the experimental results using ISCAS89 benchmark circuits show that, on average, its TAT is 16.8% shorter and its memory size is 18.3% smaller.

This paper did not quantitatively discuss the physical routing of the scan chain and its area overhead. However, we assume that this is a trivial problem because the proposed method does not determine the order of FFs, but the parities of their numbers, i.e., whether they are even or odd.

This work targets only robust path delay fault testing. To discuss un-targeted fault models, e.g., non-robust testable path delay faults and transition faults, is an important future work. The proposed method requires a long calculation time for reordering because it examines compactability for all combinations of two path delay faults. Future works also include the development of a method for reducing the calculation time in the proposed method.

Acknowledgments This research was partially supported by grants from Kurata Memorial Hitachi Science and Technology Foundation and the Grant-in-Aid for Young Scientists (B) No. 21700053.

References

1) Wang, L.T., Wu, C.W. and Wen, X.: *VLSI test principles and architectures: design for testability*, Morgan Kaufmann (2006).
2) Krstic, A. and Cheng, K.T.: *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers (1998).
3) Savir, J. and Patil, S.: On broad-side delay test, *Proc. IEEE VLSI Test Symp.*, pp.284–290 (1994).
4) Savir, J. and Patil, S.: Skewed-load transition test: Part II, coverage, *Proc. IEEE Int'l Test Conf.*, pp.714–722 (1992).
5) Dervisoglu, B.I. and Stong, G.E.: Design for testability: using scan-path techniques for path-delay test and measurement, *Proc. IEEE Int'l Test Conf.*, pp.365–374 (1991).
6) Namba, K. and Ito, H.: Scan design for two-pattern test without extra latches, *IEICE Trans. Inf. & Syst.*, Vol.E88-D, No.12, pp.2777–2785 (2005).
7) Pramanick, A.K. and Reddy, S.M.: On the detection of delay faults, *Proc. IEEE Int'l Test Conf.*, pp.845–856 (1988).

(Received November 11, 2010)
(Revised February 24, 2011)
(Accepted April 24, 2011)
(Released August 10, 2011)

(Recommended by Associate Editor: *Kazumi Hatayama*)

Kiyonori Matsumoto received his B.E. and M.E. from Chiba University in 2009 and 2011, respectively. He joined Canon Inc. in 2011.

Kazuteru Namba received his B.E., M.E. and Ph.D. from Tokyo Institute of Technology in 1997, 1999 and 2002, respectively. He joined Chiba University in 2002. He is currently an Assistant Professor of Graduate School of Advanced Integration Science, Chiba University. His current research interests include dependable computing. He is a member of IPSJ, IEICE and IEEE.

Hideo Ito was born in Chiba, Japan, on June 1, 1946. He received his B.E. degree from Chiba University in 1969 and his D.E. degree from Tokyo Institute of Technology in 1984. He joined Nippon Electric Co. Ltd. in 1969 and Kisarazu Technical College in 1971. Since 1973, he has been a member of Chiba University. He is currently a Professor of Graduate School of Advanced Integration Science. His research interests include easily testable VLSI design, defect-tolerant VLSI design, VLSI architecture, fault-tolerant computing, and dependable computing. He is a fellow of the IEICE, and a member of the IPSJ and the IEEE.