Regular Paper

A Test State Reduction Method for FSMs with Non-Scan DFT Using Don't Care Inputs Identification Technique

Toshinori Hosokawa,† Hiroshi Date† and Michiaki Muraoka†

This paper proposes a method to reduce the number of states for testing an FSM (Finite State Machine) with non-scan DFT (Design for Testability) using a don't care inputs identification technique. The proposed method reduces the numbers of states required for the FSM testing using a don't care inputs identification technique and a state compaction technique although states required for FSM testing are classified into valid test states and invalid test states. The test length can be shortened by reducing the number of valid test states. Test area for the DFT decreases by reducing the number of invalid test states. Experimental results for MCNC'91 FSM benchmarks and practical FSMs show that, compared with a previous DFT method, the proposed method reduces the test area by 13 to 77 % and shorten the test lengths by 10 to 36 %.

1. Introduction

A design for testability (DFT) method is important for the design of reliable VLSI circuits. Scan design methods [1),2)] are one of the most popular DFT methods, so far. However, the scan design methods have the following disadvantages concerning test cost and test quality.

(1) The test circuits for DFT cause the degradation of performance and/or area due to DFT application at gate level.
(2) The test length is very long [3)].
(3) It is not suited for at-speed-testing [4)].

In order to drastically improve the above-mentioned disadvantages while keeping complete fault efficiency, non-scan DFT methods [5)~7)] for RTL (Register Transfer Level) circuits were proposed. The RTL circuits consist of a data path part and a controller part. The former is represented by hardware elements (e.g., registers, multiplexers, and operation modules) and signal lines, and the latter is represented by a finite state machine (FSM). A controller and a data path are connected with internal signal lines: control signal lines and status signal lines. A control signal line comes from the controller, and a status signal line comes from the data path. The focus of this paper is on FSMs. Non-scan DFT methods for FSMs have been proposed in Refs. 7) and 8). In Ref. 7), the non-scan DFT method at RTL was proposed to attain complete fault efficiency. The DFT method allows a given FSM to have invalid states during testing. In Ref. 8), test circuits are added at gate level to make invalid states valid using the information of FSMs.

Recently, a don't care (X) inputs identification technique was proposed [9)]. Given a test set for stuck-at-faults, the technique changes as many primary input values as possible to Xs without losing fault efficiency.

In this paper, a test state reduction method for FSMs with non-scan DFT using the X inputs identification technique and a state compaction technique is proposed to reduce test area and to shorten test length. The proposed method is built-in to the non-scan DFT method [7)] at RTL.

This paper is organized as follows. In Section 2, a non-scan DFT method for FSMs is summarized. In Section 3, a test state reduction method for FSMs with non-scan DFT using the X inputs identification technique is proposed. In Section 4, experimental results are shown by applying the proposed method to MCNC'91 FSM benchmarks [12)] and practical FSMs. Finally, Section 5 concludes this paper.

2. Non-Scan DFT Method for FSM

In this section, the non-scan DFT for FSMs [7)] is summarized. In FSMs, status signal lines are recognized as primary inputs, and control signal lines are recognized as primary outputs. An FSM has a reset state, and it is possible to go to the reset state regardless of the current state when the reset signal is activated. **Figure 1** shows an example of the FSM. In this figure, R is a reset signal, $s_i\,(i = 0, 1, 2, 3, 4, 5, \text{and } 6)$ is

† Design Technology Development Department, Semiconductor Technology Academic Research Center (STARC)

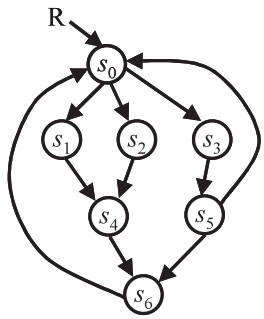

Fig. 1 Example of FSM.

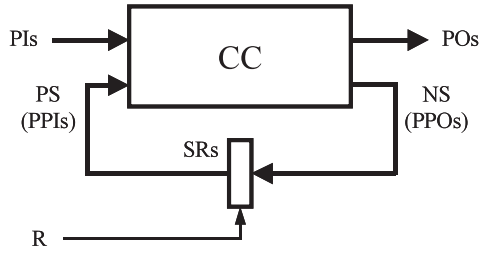

Fig. 2 Model of a synthesized sequential circuit.

a state, where s_0 is the reset state. If a state is reachable from the reset state, the state is called a valid state. Otherwise, it is called an invalid state.

2.1 DFT

A procedure of the non-scan DFT method for FSMs is given bellow.

Step 1: Logic synthesis

Given an FSM, a sequential circuit is synthesized by a tool of logic synthesis. **Figure 2** shows a model of a synthesized sequential circuit from the FSM. In this figure, CC is a combinational logic block, SRs are state registers, PIs are primary inputs, POs are primary outputs, R is a reset signal line, PS is a present state, NS is a next state, PPIs are pseudo primary inputs, and PPOs are pseudo primary outputs.

Step 2: Combinational test generation

From the synthesized sequential circuit, the CC extracted by replacing the SRs with PPIs and PPOs is defined as a combinational test generation model. Test patterns are generated for the combinational test generation model.

If the values of PPIs in a test pattern are a valid state, the test pattern is said to be a valid test pattern [7] and the values of PPIs are called a valid test state [7]. Otherwise, the test pattern is called an invalid test pattern [7] and the values of PPIs are said an invalid test state. Thus, the generated test patterns are classified into valid test patterns and invalid test patterns. Valid test states and invalid test states are referred to as test states.

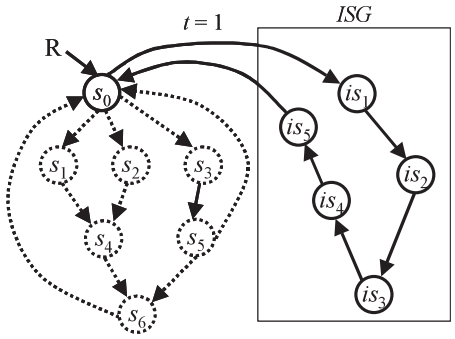

Fig. 3 Example of FSM with DFT.

Step 3: DFT

An invalid test state generator (*ISG*) is designed. All the invalid test states can be sequentially traversed from the reset state by the *ISG*. The function of the *ISG* is added to the original FSM. When the value of an additional primary input t, which gives a mode switching signal, is 1, the *ISG* is activated. **Figure 3** shows an example of the FSM with the *ISG*. In this figure, $is_i\,(i = 1, 2, 3, 4, \text{and } 5)$ means an invalid test state, and t means a mode switching signal. Additional primary outputs *t_out*, which are state output signals, are also added to the FSM to observe a next state.

Thus, faults which are propagated to PPOs are detected at *t_out*. The bit width of *t_out* is the same as the number of registers in SRs [11]. A hold function is added to the SRs. When the value of an additional primary input H, which gives a hold/load signal, is 0, the SRs operate as load mode. Otherwise, the SRs operate as hold mode. This hold mode is utilized to reduce test application time. The details are given in Section 2.2. The hold mode of the SRs can be implemented by setting 0 to the clock signal line. In this case, H is unnecessary. **Figure 4** shows the model of a synthesized sequential circuit from an FSM with the DFT.

2.2 Test Length

(1) Applying valid test patterns

A transition sequence to traverse all the valid test states from the reset state can set all PPI values to the SR. Such a transition sequence is called a *valid test state traversing sequence* [7]. The shortest *valid test state traversing sequence* is obtained by solving the traveling salesman problem [10]. For a valid test state, there may exist two or more valid test patterns which con-

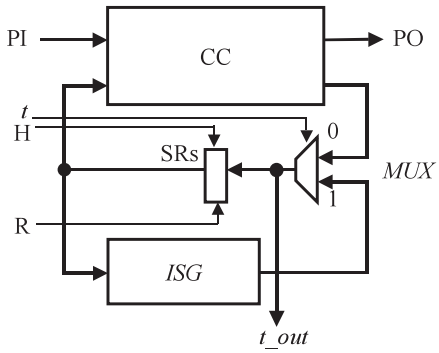

Fig. 4 Model of a synthesized sequential circuit of FSM with DFT.

tain the valid test state. Therefore, the test length can be reduced if the PI values of the test patterns are applied one after another with holding the PPI values at the SR.

(2) Applying invalid test patterns

Each of the invalid test patterns is applied in the same way as a valid test pattern using the *ISG*.

(3) Test length for FSM testing

Let N_{pat}, L_{vt}, and N_{is} be the number of test patterns, the length of a *valid test state traversing sequence*, and the number of invalid test states, respectively. The test length for FSM testing is expressed by the following equation [7].

$$L_{vt} + N_{is} + N_{pat} + 2 \qquad (1)$$

3. Test State Reduction Method Using X Inputs Identification Technique

3.1 X Inputs Identification Technique

In this sub-section, the X inputs identification technique [9] is summarized. Given a combinational circuit and its test pattern set T in which every primary input value of test patterns has been specified to either 0 or 1, a test pattern set T' including some Xs is generated. A test pattern set T' has the following properties:

(1) T' covers T.
(2) T' contains as many Xs as possible.
(3) Fault efficiency of T' is equal to that of T.

Figure 5 shows an example of a combinational circuit. Suppose that the test pattern set T in **Table 1** (a) was generated for the circuit shown in Fig. 5. The test pattern set T' in Table 1 (b) is one of the solutions. The test pattern t_1 detects the faults $pi_1/0$, $pi_2/0$, and

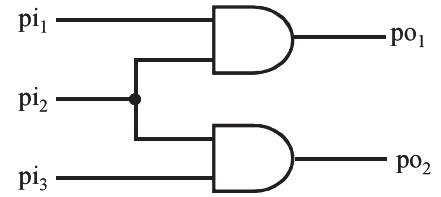

Fig. 5 Example of a combinational circuit.

Table 1 Example of X inputs identification.

(a) T

	pi_1	pi_2	pi_3
t_1	1	1	0
t_2	1	0	1
t_3	0	1	0
t_4	0	1	1

(b) T'

	pi_1	pi_2	pi_3
t_1'	1	1	X
t_2'	1	0	1
t_3'	0	1	0
t_4'	X	1	1

$pi_3/1$, where s/v denotes stuck-at v fault on signal line s. While $pi_1/0$ has to be detected by t_1, the fault $pi_3/1$ does not have to be detected by t_1 because t_3 also detects it. Hence the value 0 at pi_3 in t_1 becomes an X. Similarly, the value 0 at pi_1 in t_4 becomes an X. Thus, the test pattern set T' shown in Table 1(b) is obtained.

3.2 Problem Formulation

Let T be a test pattern set which is generated for a combinational test generation model of an FSM. Suppose that test pattern set T' including some Xs is generated from test pattern set T without any Xs using the X inputs identification technique. Let S and S' be state sets for FSM testing corresponding to T and T', respectively. A state for FSM testing is the values of PPIs in a test pattern.

Given state sets S, S' and an FSM, irreducible state set S'' is generated to reduce the number of states for FSM testing.

3.3 Test State Reduction Method

In this sub-section, a test state reduction method for FSMs with non-scan DFT using the X input identification technique is described. State $s_i \in S$ changes to $s_i' \in S'$ by applying the X inputs identification technique and s_i' may include Xs. If s_i' and valid test state $s_j'\,(i! = j, s_j' \in S')$ can be merged, the number of states for the FSM testing is reduced.

Figure 6 shows the non-scan DFT algorithm for FSMs based on the proposed test state reduction method. Function *non_scan_dft* is described as follows. First, given an FSM, sequential circuit G is synthesized from the FSM by logic synthesis (line 2). Next, combinational test generation model CC is extracted from G (line 3) and test pattern set T is generated for CC (line 4). Next, test pattern set

```
1.  non_scan_dft (FSM) {
2.    Synthesize sequential circuit G from FSM by logic synthesis;
3.    Extract combinational test generation model CC from G;
4.    Generate test pattern set T for CC;
5.    Generate test pattern set T' including some Xs using X
        inputs identification technique;
6.    S'' = compact_state_set (T, T', FSM);
7.    Add DFT function to FSM;
8.  }
9.  compact_state_set (T, T', FSM) {
10.   S = extract states corresponding to the values of PPIs from T;
11.   S' = extract states corresponding to the values of PPIs from T';
12.   S' = delete state s_i corresponding to the values of PPIs in test pattern
        t_i from S' if the values of t_i in T' are all Xs;
13.   Make out valid test state compaction buffer VTS_BUF;
14.   for (each s_i' ∈ S') {
15.     if (s_i' includes X) {
16.       Check whether s_i' and a state in VTS_BUF are compatible or not;
17.       if (Compatible) {
18.         Update VTS_BUF;
19.         Delete s_i' from S';
20.       }
21.     }
22.     else if (s_i == a valid state) {
23.       Delete s_i' from S';
24.     }
25.   }
26.   Make out invalid test state compaction buffer ITS_BUF;
27.   for (each s_i' ∈ S') {
28.     Check whether s_i' and a state in ITS_BUF are compatible or not;
29.     if (Compatible) {
30.       Update ITS_BUF;
31.     }
32.     else {
33.       Add s_i' to ITS_BUF;
34.     }
35.   }
36.   S'' = states with on-flag in VTS_BUF and states in ITS_BUF;
37.   Return S'';
38. }
```

Fig. 6 Algorithms of non-scan DFT and state reduction for FSM testing.

T' including some Xs is generated from T using the X inputs identification technique (line 5). Next, from T, T', and the FSM, reduced state set S'' for FSM testing is generated by function *compact_state_set* (line 6). Finally, the invalid test states in S'', mode switching signal (t), hold/load signal (H), and primary outputs (t_out) are added to the given FSM (line 7).

Function *compact_state_set* is described as follows. T, T', and the FSM are given. First, states corresponding to the values of PPIs in test patterns are extracted from T, and let the state set be S (line 10). Next, the states corresponding to the values of PPIs in test patterns are extracted from T' and let the state set be S' (line 11). State s_i corresponding to the values of PPIs in test pattern t_i is deleted from S' if the values of t_i in T' are all Xs (line 12). Function *compact_state_set* has two major processes.

(1) Valid test state compaction

In this process, it is checked whether state $s'_i \in S'$ which includes Xs, is merged into a valid state or not. A valid test state compaction buffer *VTS_BUF* is made out (line 13). Each valid state is set into each state part in *VTS_BUF*. If the valid state exists in S', "ON" is set into the flag part and the valid state is referred to as the state with on-flag. Otherwise, "OFF" is set into the flag part and the valid state is referred to as the state with off-flag. If s'_i is merged into a state with off-flag, the flag part is changed from "OFF" to "ON".

Next, for $s'_i \in S'$, the following steps are iterated (line 14).

(case 1: s'_i includes X) First, it is checked whether s'_i and a state part of each row in *VTS_BUF* can be merged or not (line 16). The test pattern compaction technique proposed in Ref. 11) is used for this state compaction. The state compaction is preferentially tried from a state with on-flag. If si' and any states with on-flag in *VTS_BUF* are not compatible, then it is checked whether s'_i and a state with off-flag in *VTS_BUF* are compatible or not. If they are compatible [11] (line 17), the flag part of each row in *VTS_BUF* is updated (line 18) and s'_i is deleted from S' (line 19).

(case 2: not case 1 and s_i is a valid state) s'_i is deleted from S' (line 23).

(2) Invalid test state compaction

In this process, it is checked whether state $s'_i \in S'$ which includes Xs, is merged with an invalid test state or not. Invalid test state compaction buffer *ITS_BUF* is made out (line 26). Each row in *ITS_BUF* is a state.

If s'_i does not include any Xs, s'_i is set into each row in *ITS_BUF* and s'_i is deleted from S'.

Next, for each $s'_i \in S'$, the following steps are iterated (line 27).

First, it is checked whether s'_i and each row in *ITS_BUF* can be merged or not. The state compaction is preferentially tried from a state with few numbers of Xs in *ITS_BUF*. If they are compatible [11] (line 29), the state in *ITS_BUF* is updated (line

Table 2 Original state set S and state set S' after X inputs identification.

(a) original states S

s_1	000
s_2	110
s_3	011
s_4	010
s_5	101
s_6	001
s_7	011
s_8	100
s_9	111

(b) states with Xs S'

s_1'	000
s_2'	XX0
s_3'	01X
s_4'	010
s_5'	1XX
s_6'	001
s_7'	X11
s_8'	100
s_9'	X1X

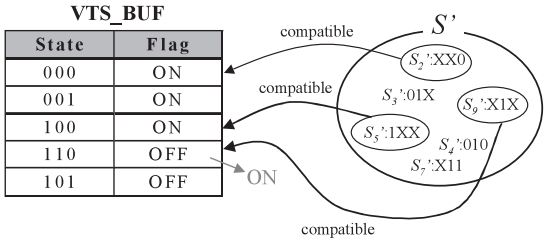

Fig. 7 Example of valid test state compaction buffer.

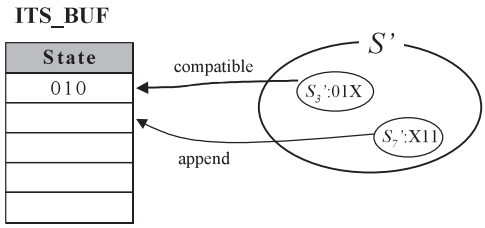

Fig. 8 Example of invalid test state compaction buffer.

30). Otherwise (line 32), s_i' is added to a row in *ITS_BUF* (line 33).

Finally, states with on-flag in *VTS_BUF* and states in *ITS_BUF* are inserted into a state set S'' (line 36) and S'' is returned (line 37).

Example: Consider the states in **Table 2**, **Fig. 7**, and **Fig. 8**. Note that these states (s_1 to s_9, and s_1' to s_9') are different from those of Fig. 1, Fig. 2, and **Fig. 9**. Table 2 shows an original test state set S and state set S' after the application of the X inputs identification. In Table 2, (a) shows S and (b) shows S'. s_1, s_2, s_5, s_6, and s_8 are valid test states, and s_3, s_4, s_7, and s_9 are invalid test states. Figure 7 shows an example of compaction of states in S' and states in *VTS_BUF*. All the valid states $\{000, 001, 100, 110, 101\}$ are set into the state parts in *VTS_BUF*.

Because s_1' is 000, s_6' is 001, and s_8' is 100, "ON" are set into the flag parts of 000, 001, and 100 in *VTS_BUF*. State XX0 (s_2') and state 1XX (s_5') are merged into state 000 with on-flag and state 100 with on-flag in *VTS_BUF*, respectively. State X1X (s_9') is not compatible with any states with on-flag in *VTS_BUF*, but is compatible with 110 with off-flag in *VTS_BUF*. The flag part of state 110 changes from "OFF" to "ON". State 01X (s_3') and state X11 ($s7'$) are not compatible with any states in *VTS_BUF*. Figure 8 shows an example of compaction of states in S' and states in *ITS_BUF*. Because state 010 (s_4') does not include X, state 010 is set into *ITS_BUF*. State 01X ($s3'$) is compatible with 010 in *ITS_BUF*. Because state X11 ($s7'$) is not compatible with any states in *ITS_BUF*, state X11 is appended to *ITS_BUF*. Thus, the test state set S'', obtained at this test state reduction, is $\{000, 001, 100, 110, 010, X11\}$. The number of valid test states is reduced from 5 to 4 and the number of invalid test states is reduced from 3 to 2.

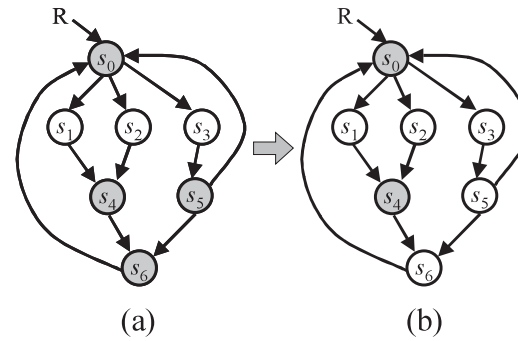

Fig. 9 Example of the reduction of the number of valid test states.

3.4 Effect of Test State Reduction for FSM Testing

The proposed test state reduction method can reduce both the number of invalid test states and the number of valid test states. Thus, the total number of test states for the FSM is reduced. In this sub-section, each effect is described.

(1) The number of invalid test states

The test area for the *ISG* is decreased by reducing the number of invalid test states. For example, in Fig. 3, the number of invalid test states is reduced from 5 (is_1 to is_5) to 3 (is_1, is_2, and is_4) by applying the proposed test state reduction method. Only the invalid test states is_1, is_2, and is_4 are added to the original FSM (Fig. 1). Thus, the area of the *ISG* shown in Fig. 4 is reduced.

(2) The number of valid test states

If the number of valid test states is reduced, the length of a *valid test state traversing sequence* is shortened. Thus, test length is shortened. For example, in Fig. 9, the number of valid test states is reduced from 4 (s_0, s_4, s_5, and s_6) to 2 (s_0 and s_4) by applying the proposed test state reduction method. In Fig. 9, states with shadow express valid test states. In Fig. 9 (a), the *valid test state traversing sequence* is a sequence which causes the following state transitions ($s_0 \rightarrow s_1 \rightarrow s_4 \rightarrow s_6 \rightarrow s_0 \rightarrow s_3 \rightarrow s_5$). In Fig. 9 (b), the *valid test state traversing sequence* is a sequence which causes the following state transitions ($s_0 \rightarrow s_1 \rightarrow s_4$). Thus, the length of the *valid test state traversing sequence* is shortened from 7 to 3.

Because the reduction of invalid test states has priority over the reduction of valid test states, the number of valid test states may increase by compacting invalid test states and valid states. However, it is considered that such a case seldom occurs.

4. Experimental Results

In this section, experimental results of the proposed non-scan DFT method for MCNC'91 benchmarks [12] and practical FSMs are shown. **Table 3** shows the characteristics of the FSMs. In this table, #States is the number of valid states, #PIs is the number of primary inputs, #POs is the number of primary outputs, #FFs is the number of flip-flops, and Area is the area size of synthesized sequential circuits by logic synthesis. Here, the area size is estimated using library cell area. "IDCT" and "DCT" are practical FSMs. As for MCNC'91 benchmarks [12], we applied our non-scan DFT method to only FSMs with many numbers of states (more than 40). In our experiment, we used the logic synthesis tool Design Compiler (Synopsys) with a sample library of Synopsys, the test generation tool TetraMax (Synopsys), and the X inputs identification program used in Ref. 9). **Table 4** shows the experimental results. In this table, Circuit denotes the name of FSMs, "Ref. 7)" denotes a non-scan DFT method proposed in Ref. 7), and "Ours" denotes the proposed non-scan DFT method with the test state reduction.

Table 3 FSM characteristics.

Circuit	#States	#PIs	#POs	#FFs	Area
planet	48	7	19	6	471
planet1	48	7	19	6	471
s1488	48	8	19	6	484
s1494	48	8	19	6	470
s298	218	3	6	8	1206
s510	47	19	7	6	317
IDCT	40	11	40	6	209
DCT	45	13	45	6	232

#VTSs denotes the number of valid test states, #ITSs denotes the number of invalid test states, TA denotes the additional test circuit area for DFT, and TL denotes the test length. RTA (%) and RTL (%) are expressed the following equations.

$$R_{TA} = (TA\ of\ [7] - TA\ of\ Ours)/TA\ of\ [7]$$
$$R_{TL} = (TL\ of\ [7] - TL\ of\ Ours)/TL\ of\ [7]$$

The numbers of valid test states were reduced by 41 to 52 % and the test lengths were shortened by 10 to 36 %. Test lengths are calculated using the equation (1). Because RTL depends on the structure of FSMs, this information must be considered when compacting valid test states in order to improve R_{TL}. There are two factors that reduce the test circuit area. One is the reduction of the numbers of invalid test states. The other is the increase of the number of Xs in the invalid test states. The number of the invalid test states was reduced by 0 to 46 % and test circuit area were reduced by 13 to 77 %. The ratio of the numbers of reduced invalid test states for $s1488$ and $s1494$ as shown in Table 4 is greater than that for other circuits. The ratio of the numbers of Xs in invalid test states for $s1488$ and $s1494$ (16.67%) was greater than that for other circuits. Therefore, test area for these circuits was much reduced.

Table 4 Experimental results.

Circuit	[7]				Ours					
	#VTSs	#ITSs	TA	TL	#VTSs	#ITSs	TA	R_{TA}(%)	TL	R_{TL}(%)
planet	47	13	72	204	23	8	42	41.67	129	36.76
planet1	47	13	72	204	23	8	42	41.67	129	36.76
s1488	48	15	71	223	24	8	16	77.46	155	30.49
s1494	48	15	77	219	24	8	25	67.53	168	23.29
s298	216	2	61	730	127	2	53	13.11	560	23.29
s510	47	3	53	127	26	2	22	58.49	111	12.60
IDCT	40	3	51	102	19	2	39	23.53	90	11.76
DCT	45	2	20	109	22	2	9	55.00	98	10.09

5. Conclusion

This paper proposed a test state reduction method for FSMs with non-scan DFT using the X inputs identification technique. The proposed method could reduce the numbers of invalid test states and valid test states using the X inputs identification technique and a state compaction technique. Experimental results for MCNC'91 FSM benchmarks and practical FSMs show that the proposed method reduces test area by 13 to 77% and shorten the test lengths by 10 to 36 %.

Future work includes:

(1) proposing a compaction technique for valid test states which takes a valid test state traverse into account to shorten test lengths furthermore, and

(2) proposing an X inputs identification technique which focuses on the values of PPIs in test patterns to reduce the numbers of invalid test states and valid test states furthermore.

Acknowledgments This work was sponsored by NEDO (New Energy and Industrial Technology Development Organization) as VCDS Project (SoC advanced design technology development project). Authors would like to thank Professor Hideo Fujiwara, Professor Michiko Inoue, and Professor Satoshi Ohtake of Nara Institute of Science and Technology for their valuable discussion and comments. Authors would like to thank Professor Seiji Kajihara of Kyushu Institute of Technology for the offer of the don't care inputs identification program. Authors would like to thank Dr. Rafael K. Morizawa of Semiconductor Technology Academic Research Center for his valuable comments.

References

1) Fujiwara, H.: *Logic Testing and Design for Testability*, The MIT Press (1985).
2) Abramovici, M., Breuer, M.A. and Friedman, A.D.: *Digital systems testing and testable design*, IEEE Press (1995).
3) Hosokawa, T., Yoshimura, M. and Ohta, M.: Design for testability strategies using full/partial scan designs and test point insertions to reduce test application times, *Proc. Asia and South Pacific Design Automation Conference*, pp.485–491 (2001).
4) Maxwell, P.C., Aitken, R.C., Johansen, V. and Chiang: The effect of different test sets on quality level prediction: When is 80% better than 90%?, *Proc. International Test Conference*, pp.358–364 (1991).
5) Ohtake, S., Wada, H., Masuzawa, T. and Fujiwara, H.: A Non-Scan DFT Method at Register-Transfer Level to Achieve Complete Fault Efficiency, *Proc. Asia and South Pacific Design Automation Conference*, pp.599–604 (2000).
6) Ohtake, S., Nagai, S., Wada, H. and Fujiwara, H.: A DFT Method for RTL Circuits to Achieve Complete Fault Efficiency Based on Fixed-control Testability, *Proc. Asia and South Pacific Design Automation Conference*, pp.331–334 (2001).
7) Ohtake, S., Masuzawa, T. and Fujiwara, H.: A Non-Scan DFT Method for Controllers to Achieve Complete Fault Efficiency, *Proc. Asian Test Symposium*, pp.204–211 (1998).
8) Das, D.K., Ohtake, S. and Fujiwara, H.: New DFT Techniques of Non-Scan Sequential Circuits with Complete Fault Efficiency, *Proc. Asian Test Symposium*, pp.263–268 (1999).
9) Kajihara, S. and Miyase, K.: On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits, *Proc. Int'l Conf. of CAD*, pp.364–369 (2001).
10) Garey, M.R. and Johnson, D.S.: Computer and Intractability, W.H. Freeman and Company (1979).
11) Niermann, T.M., Roy, R.K., Patel, J.H. and Abraham, J.A.: Test Compaction for Sequential Circuits, *IEEE Trans. on Computer-Aided Design*, Vol.11, No.2, pp.260–267 (Feb. 1992).
12) Yang, S.: Logic synthesis and optimization benchmarks user guide, Technical Report 1991-IWLS-UG-Saeyang, Microelectronics Center of North Carolina (1999).

(Received October 16, 2002)
(Accepted March 4, 2003)

Toshinori Hosokawa received the B.E. degree in Electronics and Communication Engineering from Meiji University, Kawasaki, Japan, in 1987. He also received the D.E. degree from Meiji University in 2001. In 1987, he joined Matsushita Electric Industrial Co., Ltd. He has worked on logic simulation engine, automatic test pattern generation, fault simulation, design for testability and high level testing. In 2000, he was temporarily transferred to Semiconductor Technology Academic Research Center (STARC), where he is currently working on testing for system on a chip and hardware/software co-verification. He was also a lecturer at Meiji University from 2001 to 2002. He is a member of IEEE and IPSJ.

Hiroshi Date received the B.S. and M.S. degrees in Science of Mathematics from Kyushu University, Fukuoka, Japan, in 1985 and 1987, respectively. He also received the D.E. degree from Kyushu University in 1993. From 1987 to 1996, he was with Hitachi Research Laboratory, Hitachi Ltd. and was engaged in VLSI-CAD and parallel processing. From 1990 to 1993, he joined Institute for New Generation Computer Technology (ICOT) and worked on VLSI-CAD using parallel processing. From 1996 to 2001, he was with Institute of Systems & Information Technologies/ KYUSHU. From 1999 to 2001, he was an associate professor of Kyushu University. From 2001 to 2002, he was with Abel systems Inc. Since 2000, he has joined Semiconductor Technology Academic Research Center. Since 2002, he has been a chairman & CEO of System J D Co., Ltd. His research interests include design & test methodology and CAD technologies for system LSI, and information security. He is a member of IEICE, IPSJ, TTTC, ACM, and IEEE.

Michiaki Muraoka received the B.E. degree in Electrical Engineering from Keio University, Tokyo, Japan in 1974. He worked on the research and development of system and semiconductor EDA system, especially in the logic and RTL design and silicon compilation at OKI Electric from 1974 to 1988. He joined Matsushita Electric Industrial Company where he worked on the research and development of EDA technology and IP based design methodology as the technical leader from 1988 to 2000. He was temporarily assigned to STARC in 2000 as a senior manager of VCDS Development Group. His current research interests are in the areas of the next generation design methodology and EDA technology for the giga scale SoC (System on Chip).