IEICE TRANS. INF. & SYST., VOL.E96–D, NO.9 SEPTEMBER 2013

PAPER Special Section on Dependable Computing

A Test Compaction Oriented Don't Care Identification Method Based on X-bit Distribution

Hiroshi YAMAZAKI[†a)], **Motohiro WAKAZONO**[†], ***Nonmembers***, **Toshinori HOSOKAWA**[††], ***and*** **Masayoshi YOSHIMURA**[†††], ***Members***

SUMMARY In recent years, the growing density and complexity of VLSIs have led to an increase in the numbers of test patterns and fault models. Test patterns used in VLSI testing are required to provide high quality and low cost. Don't care (X) identification techniques and X-filling techniques are methods to satisfy these requirements. However, conventional X-identification techniques are less effective for application-specific fields such as test compaction because the X-bits concentrate on particular primary inputs and pseudo primary inputs. In this paper, we propose a don't care identification method for test compaction. The experimental results for ITC'99 and ISCAS'89 benchmark circuits show that a given test set can be efficiently compacted by the proposed method.

key words: *X-bit, don't care identification, X-bit distribution, test compaction*

1. Introduction

In recent years, the growing density and complexity of very-large-scale integration (VLSI) circuits has caused an increase in the numbers of test patterns and fault models. Test patterns for not only stuck-at faults [1], [2] but also bridging faults [3]–[5] and transition faults [6], [7] are required for VLSI testing. Because the test cost is generally proportional to the number of test patterns, the test cost increases with the increase in the number of test patterns.

Test compaction [8] is one of the methods to solve the problem that the number of test patterns increases. Test compaction methods are generally classified into two types: a don't care based method and a fault simulation based method. A don't care based test compaction method reduces the number of test patterns by merging a test pattern with other compatible test patterns [9], [10]. A fault simulation based test compaction method reduces the number of test patterns by eliminating redundant test patterns by fault simulation. Reverse order fault simulation [11] and double detection [12] are examples of proposed fault simulation based methods for test compaction.

Some of the specified primary input (PI) and pseudo primary input (PPI) values in a test set may be changed to opposite logic values without losing fault coverage. Such PI and PPI values can be regarded as don't care (X)-bits. X-identification methods to identify many don't care inputs of test patterns in a given test set have been proposed [13]–[15].

In two proposed X-identification methods, XID [13] and DC-XID [14], the places of X-bits are changed by algorithms. XID identifies X-bits concentrated in particular test patterns. Therefore, XID may be less effective for application-specific fields. DC-XID was proposed for low power testing fields and controls the distribution of X-bits identified from an initial test set. DC-XID averages the number of detected faults for each test pattern. As a result, the number of X-bits in each test pattern becomes almost equal.

For test compaction as application-specific fields, XID and DC-XID may be less effective, because these two methods do not take into account the distribution of X-bits for PI (PPI). We presume that X-bits should be distributed at PI (PPI) for the effectiveness of test compaction.

In this paper, we hypothesize that a uniform number of X-bits in each PI (PPI) in a test set is effective for test compaction. The relationship between the X-bit variance for PI (PPI) and the number of test patterns after test compaction is analyzed. An X-identification problem for test compaction is formulated from the results of the analysis and a heuristic algorithm of X-identification is proposed.

This paper is organized as follows. Section 2 describes the relationship between X-bit variance for PI (PPI) and test compaction probability. Section 3 shows the correlation between X-bit variance for PI (PPI) and test compaction. Section 4 proposes an X-identification method for test compaction. Section 5 shows the experimental results for ISCAS'89 and ITC'99 benchmark circuits. Finally, Sect. 6 concludes the paper and discusses future work.

2. Background

2.1 Preliminaries

In this paper, full-scan design sequential circuits and combinational circuits are targeted. When test generation is applied to full-scan design sequential circuits, they can be treated as combinational circuits. Thus, we discuss X-identification methods for only combinational circuits henceforth. The number of PI's for a combinational circuit is denoted by N_{PI}.

Manuscript received January 18, 2013.
Manuscript revised April 28, 2013.

[†]The authors are with the Graduate School of Industrial Technology, Nihon University, Narashino-shi, 275–8575 Japan.
[††]The author is with the College of Industrial Technology, Nihon University, Narashino-shi, 275–8575 Japan.
[†††]The author is with the Department of Computer Science and Communication Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan.
a) E-mail: cihi12002@g.nihon-u.ac.jp

DOI: 10.1587/transinf.E96.D.1994

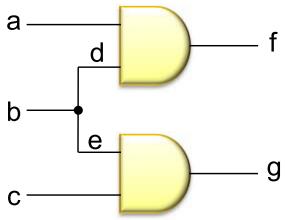

Fig. 1 Example circuit.

Table 1 Test set.

(a) T

	abc
t_1	110
t_2	101
t_3	010
t_4	011

(b) XT

	abc
xt_1	11X
xt_2	101
xt_3	010
xt_4	X11

Table 2 Test compaction operation $\cap_T$.

$V(xt_i, p_k)$ \ $V(xt_j, p_k)$	0	1	X
0	0	Φ	0
1	Φ	1	1
X	0	1	X

Some of the specified PI values in a test set may be changed to the opposite logic values without losing fault coverage. Such PI values can be regarded as don't care bits. Don't care bits can be set to the logic value "0" or "1". The don't care bit is denoted as "X" or "x" in a test pattern.

2.2 Formulation of X-identification

In this paper, an initial test set T is generated by the Automatic Test Pattern Generator (ATPG). Given a circuit C and the fully specified test set T, we compute the test set XT, which includes some X-bits and has the following properties.

(1) XT covers T.
(2) XT contains as many X-bits as possible.
(3) The fault coverage of XT is equal to that of T.

We show a simple example of X-identification for a single stuck-at fault. Suppose that test set T in Table 1 (a) is generated for the circuit shown in Fig. 1. Test set XT in Table 1 (b) is one of the solutions. Test pattern t_1 detects faults $a/0$, $b/0$, and $c/1$, where s/v denotes the stuck-at fault $v \in \{0, 1\}$ on the signal line s. Fault $a/0$ has to be detected by t_1, because no other test pattern can detect $a/0$. In contrast, fault $c/1$ does not have to be detected by t_1 because t_3 detects $c/1$, too. Hence, the value 0 at primary input c of t_1 becomes an X-bit. Similarly, the value 0 at primary input a of t_4 becomes an X-bit. Thus, test set XT in Table 1 (b) is obtained. XT is the set of test patterns xt_i that contain an X-bit.

2.3 Formulation of Test Compaction Probability

In this section, test compaction and test compaction probability are described for two test patterns xt_i and xt_j. $V(xt_i, p_k)$ shown in Eq. (1) is an equation that expresses the value of primary input p_k in test pattern xt_i.

$$V(xt_i, p_k) = \begin{cases} 0 & \text{if } p_k \text{ value of } xt_i \text{ is 0} \\ 1 & \text{if } p_k \text{ value of } xt_i \text{ is 1} \\ X & \text{otherwise (X-bit)} \end{cases} \quad (1)$$

Table 2 shows a test compaction operation $\cap_T$ for $V(xt_i, p_k)$ and $V(xt_j, p_k)$. ϕ means that $V(xt_i, p_k)$ and $V(xt_j, p_k)$ cannot be merged.

$$cxt_{pk} = V(xt_i, p_k) \cap_T V(xt_j, p_k) \quad (2)$$

From Table 2, the result of test compaction $cxt_{pk} \in \{0, 1, X, \phi\}$ is denoted by Eq. (2).

$COM(xt_i, xt_j)$ shown in Eq. (3) is an equation that expresses whether xt_i and xt_j are compatible or not. If xt_i and xt_j are compatible, Eq. (3) returns 1; otherwise, it returns 0.

$$COM(xt_i, xt_j) = \begin{cases} 0 & \text{if } xt_i \text{ and } xt_j \text{ are compatible} \\ 1 & \text{otherwise (incompatible)} \end{cases} \quad (3)$$

As shown in Eq. (4), if the result of test compaction operation $\cap_T$ for xt_i and xt_j includes at least one ϕ, xt_i and xt_j are incompatible.

$$\exists p_k (cxt_{pk} = \Phi) \Rightarrow COM(xt_i, xt_j) = 0 \quad (4)$$

As shown in Eq. (5), if the result of test compaction operation $\cap_T$ for xt_i and xt_j does not includes ϕ, xt_i and xt_j are compatible.

$$\forall p_k (cxt_{pk} \neq \Phi) \Rightarrow COM(xt_i, xt_j) = 1 \quad (5)$$

$BP(xt_i, xt_j, p_k)$ shown in Eq. (6) is an equation that expresses the test compaction probability for p_k value of xt_i and p_k value of xt_j. If both p_k values of xt_i and xt_j are care-bits, Eq. (6) returns $P0(xt_i, p_k) \times P0(xt_j, p_k) + P1(xt_i, p_k) \times P1(xt_j, p_k)$; otherwise, it returns 1.

$$BP(xt_i, xt_j, p_k) = \begin{cases} P0(xt_i, p_k) \times P0(xt_j, p_k) & \text{if both } p_k \text{ value of } xt_i \\ \quad + P1(xt_i, p_k) \times P1(xt_j, p_k) & \text{and } xt_j \text{ are care-bits} \\ 1 \quad \text{otherwise } (p_k \text{ values of } xt_i \text{ and/or } xt_j \text{ are X-bits}) \end{cases} \quad (6)$$

In Eq. (6), $P0(xt, p_k)$ denotes probability that p_k values of xt is 0, and $P1(xt, p_k)$ denotes probability that p_k values of xt is 1. If p_k values of xt_i and/or xt_j are X-bits, the p_k values are compatible. Therefore, test compaction probability is 1 when X-bits are included.

$PCOM(xt_i, xt_j)$ shown in Eq. (7) is an equation that expresses the test compaction probability for xt_i and xt_j.

$$PCOM(xt_i, xt_j) = \prod_{k=1}^{N_{PI}} BP(xt_i, xt_j, p_k) \quad (7)$$

2.4 X-bit Variance at Primary Input

In this section, we describe X-bit distribution of each PI. In this paper, variance is used to evaluate the X-bit distribution

of each PI.

$$X(xt_i, p_k) = \begin{cases} 1 & \text{if } p_k \text{ value of } xt_i \text{ is an X-bit} \\ 0 & \text{otherwise (care bit)} \end{cases} \quad (8)$$

In Eq. (8), if p_k value of xt_i is an X-bit, Eq. (8) returns 1; otherwise, it returns 0.

$$C(xt_i, p_k) = \begin{cases} 1 & \text{if } p_k \text{ value of } xt_i \text{ is a care bit} \\ 0 & \text{otherwise (X-bit)} \end{cases} \quad (9)$$

In Eq. (9), if p_k value of xt_i is a care bit, Eq. (9) returns 1; otherwise, it returns 0.

$A_X(XT)$ shown in Eq. (10) is the average value of the number of X-bits for PI in XT.

$$A_X(XT) = \frac{1}{N_{PI}} \sum_{m=1}^{N_{PI}} \left(\sum_{n=1}^{N_{TP}(XT)} X(xt_n, p_m) \right) \quad (10)$$

N_{PI} is the number of PI's. $N_{TP}(XT)$ is the number of test patterns in XT.

In Eq. (11), $s^2(XT)$ is the X-bit variance for PI in XT.

$$s^2(XT) = \frac{1}{N_{PI}} \sum_{i=1}^{N_{PI}} \left(A_X(XT) - \sum_{j=1}^{N_{TP}(XT)} X(xt_j, p_i) \right)^2 \quad (11)$$

2.5 Relationship between X-bit Variance at PI and Test Compaction Probability

XID and DC-XID do not consider the X-bit distribution for each PI. Therefore, XID and DC-XID may be less effective for test compaction. In this section, test compaction probabilities of test sets $XT_v = \{xt_{vn}, xt_{vm}\}$ and $XT_u = \{xt_{un}, xt_{um}\}$ are compared. XT_v and XT_u are generated from same initial test set $T = \{t_n, t_m\}$ by different X-identifications. We assume that the number of X-bits of XT_v and XT_u is equal. Care bit values of p_k for t_n and t_m are determined independently. Therefore, we assume that test compaction probability of each PI for t_n and t_m is

$$P0(x_i, p_k) \times P0(x_j, p_k) + P1(x_i, p_k) \times P1(x_j, p_k) = 0.5.$$

XT_v is generated by X-identification $X\text{-}ID_v$ that does not make the number of X-bits at each PI in a test set uniform. XT_u is generated by X-identification $X\text{-}ID_u$ that makes the number of X-bits at each PI in a test set uniform. Therefore, X-bit variance for PI is $s^2(XT_v) > s^2(XT_u)$.

$cc(xt_i, xt_j)$ shown in Eq. (12) is an equation that expresses the number of PI's whose p_k values of xt_i and xt_j are care bits.

$$cc(xt_i, xt_j) = \sum_{k=1}^{N_{PI}} (C(xt_i, p_k) \times C(xt_j, p_k)) \quad (12)$$

$cx(xt_i, xt_j)$ shown in Eq. (13) is an equation that expresses the number of PI's whose p_k values of xt_i and xt_j is different. Namely, p_k value of xt_i is a care bit and p_k value of xt_i is an X-bit, or p_k value of xt_i is an X-bit and p_k value of xt_i is a care bit.

$$cx(xt_i, xt_j) = \sum_{k=1}^{N_{PI}} \Big(X(xt_i, p_k) \times C(xt_j, p_k) + C(xt_i, p_k) \times X(xt_j, p_k) \Big) \quad (13)$$

$xx(xt_i, xt_j)$ shown in Eq. (14) is an equation that expresses the number of PI's whose p_k values of xt_i and xt_j are X-bits.

$$xx(xt_i, xt_j) = \sum_{k=1}^{N_{PI}} (X(xt_i, p_k) \times X(xt_j, p_k)) \quad (14)$$

From Eqs. (13) and (14), the number of X-bits of XT_v and XT_u is calculated as

$$cx(xt_{vm}, xt_{vn}) + 2xx(xt_{vm}, xt_{vn}) = cx(xt_{um}, xt_{un}) + 2xx(xt_{um}, xt_{un}).$$

Moreover, the number of PI's of XT_v and XT_u is calculated as

$$cc(xt_{vm}, xt_{vn}) + cx(xt_{vm}, xt_{vn}) + xx(xt_{vm}, xt_{vn}) = cc(xt_{um}, xt_{un}) + cx(xt_{um}, xt_{un}) + xx(xt_{um}, xt_{un}) = N_{PI}.$$

From $s^2(XT_v) > s^2(XT_u) \Rightarrow xx(xt_{vm}, xt_{vn}) > xx(xt_{um}, xt_{un})$,

$cc(xt_{vm}, xt_{vn}) > cc(xt_{um}, xt_{un})$.

From Eqs. (7), (12), (13) and (14), test compaction probability of XT_v is calculated as

$$PCOM(xt_{vm}, xt_{vn}) = 0.5^{cc(xt_{vm}, xt_{vn})} \times 1.0^{cx(xt_{vm}, xt_{vn})} \times 1.0^{xx(xt_{vm}, xt_{vn})} = 0.5^{cc(xt_{vm}, xt_{vn})}.$$

test compaction probability of XT_u is calculated as

$$PCOM(xt_{um}, xt_{un}) = 0.5^{cc(xt_{um}, xt_{un})} \times 1.0^{cx(xt_{um}, xt_{un})} \times 1.0^{xx(xt_{um}, xt_{un})} = 0.5^{cc(xt_{um}, xt_{un})}.$$

From $cc(xt_{vm}, xt_{vn}) > cc(xt_{um}, xt_{un})$, $PCOM(xt_{vm}, xt_{vn}) < PCOM(xt_{um}, xt_{un})$.
Therefore, the distribution of X-bits for each PI affects the efficiency of test compaction.

3. Impact of X-bit Variance of Test Compaction

3.1 Preliminary Experiments

We analyzed the relationship between the X-bit distribution at PI in a test set and the efficiency of test compaction. In this preliminary experiment, variance was used to evaluate the X-bit distribution of each PI. In the first step, X-bits were randomly substituted for care bits of a specified ratio in an initial test set T, which was generated by the ATPG tool. As the result, the random test set RXT was generated. From 1000 test sets, RXTs were generated for 1000 kinds of variance values. As expected, RXT lost fault coverage

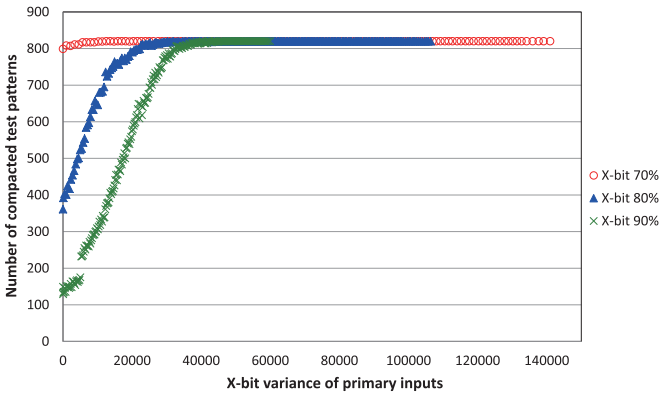

Fig. 2 Relation between X-bit variance and test compaction for b14.

compared with T. In the second step, test compaction [9] was performed for each RXT.

3.2 Preliminary Experimental Results

Figure 2 shows the preliminary experimental results for the ITC'99 b14 benchmark circuit. Initial test set T_c is a test set generated by the ATPG tool. T_c was compacted dynamically and statically. Test set RXT_c was generated by randomly substituting X-bits for care bits of a specified ratio in T_c. The specified X-bit ratios were 70%, 80%, and 90%. Each RXT_c lost fault coverage compared with T.

After test compaction of RXT_c, $CRXT_c$ was generated. Test set $CRXT_c$ was compacted by Dsatur [9] which merges test patterns with X-bits. Each $CRXT_c$ kept fault coverage compared with each RXT_c. However each $CRXT_c$ and RXT_c lost fault coverage compared with T. In Fig. 2, the vertical axis is the number of test patterns in test set $CRXT_c$. The horizontal axis is the X-bit variance for PI in RXT_c. "X-bit 70%", "X-bit 80%", and "X-bit 90%" represent the experimental results of the X-bit ratios 70%, 80%, and 90% in RXT_c, respectively.

In the "X-bit 80%" and "X-bit 90%" graphs, the number of test patterns after test compaction decreased from the approximate X-bit variance of 30,000 or less. When the X-bit variance exceeded 30,000, the number of test patterns could not be reduced. In the "X-bit 70%" graph, the number of test patterns could not be reduced even if the X-bit variance was very small.

As a result, we confirmed that the X-bit distribution at the PI was effective for test compaction. In addition, we confirmed that the efficiency of test compaction depended on the X-bit ratio in a test set.

4. Test Compaction Oriented X-identification

4.1 Problem Formulation

From Sect. 3, we confirmed that to increase the X-bit ratio and to reduce the X-bit variance for PI was effective for test compaction. Therefore, an X-identification problem for test compaction is formulated as follows.

```
1.  Procedure test_compaction_oriented_X-Identification(C,T,)
2.  C : circuit , T : initial_test_set ;
3.  {
4.      for each test_pattern t_i in T {
5.          D = fault_simulation (C, t_i ) ;
6.      }
7.      EXT = essential_X-filling (C, D, T ) ;
8.      U = collect_undetected_fault (C, D, EXT ) ;
9.      XT = distribution_X-filling (C, D, U, T, EXT) ;
10.     return XT ;
11. }
```

Fig. 3 Test compaction oriented X-identification algorithm.

Inputs: initial test set T

Outputs: test set with X-bits XT

Constraint: X-bit ratio $\geq n$ (%)

Minimization: $s^2(XT)$, subject to $D(T) = D(XT)$ (15)

In this formulation, $s^2(XT)$ is the X-bit variance for PI in test set XT. $D(T)$ is the fault coverage by test set T. $D(XT)$ is the fault coverage by test set XT. n is the threshold value of the X-bit ratio ($0 \leq n \leq 100$). XT must have the X-bit ratio equal or more than n.

4.2 Test Compaction Oriented X-identification Algorithm

In this section, we propose an X-identification algorithm for test compaction. The algorithm aims to equalize the number of X-bits at each PI. Figure 3 shows the X-identification algorithm for test compaction. Explanations of each step are given. The inputs are a circuit (C) and an initial test set (T). In Fig. 3, a fault simulation is performed for a circuit by each initial test pattern t_i in T (line 4). From the result of the fault simulation, fault dictionary D is generated (line 5). Essential faults [12] are collected from fault dictionary D. The PI values of t_i required to detect the essential faults are calculated. These PI values are fixed to care bits, and the other values are X-bits. Then, test set EXT, which can detect all essential faults, is obtained (line 7). As EXT may detect faults other than the essential faults, a fault simulation is performed by EXT. From the result of the fault simulation, undetected fault set U is generated (line 8). The X-identification for test compaction is performed to equalize the number of X-bits at each PI for the undetected faults of EXT. As the result of the X-identification, test set XT is obtained (line 9). The total number of X-bits in XT is smaller than that of X-bits in EXT, since XT increase care bits to detect undetected faults in U. XT can detect all faults in D. The details are described in Sect. 4.3. An X-identified test set (XT) is outputted (line 10).

4.3 X-bit Distribution X-filling Algorithm

In this section, we propose an X-bit distribution X-filling

```
1.  Procedure distribution_X-filling (C, D, U, T, EXT) ;
2.  C : circuit , D : fault_dictionary ,
3.  U : undetected_fault_set , T : initial_test_set ,
4.  EXT : essential_fault_detectable_test_set ;
5.  {
6.      XT = EXT ;
7.      MCT = ∞ ;
8.      for each undetected_fault f_i in U {
9.          DT_fi = collect_f_detectable_test_pattern( f_i, D) ;
10.         for each detectable test pattern t_j for f_i in DT_fi {
11.             xt'_j = find_value(f_i , t_j ) ;
12.             CT = caluculation_cost( XT, xt'_j ) ;
13.             if ( MCT > CT) {
14.                 MCT = CT ;
15.                 t_mct = xt'_j ;
16.                 k = j ;
17.             }
18.         }
19.         XT = merge_test_pattern ( xt_k in XT , t_mct , XT) ;
20.         U = fault_simulation ( C, XT, U) ;
21.     }
22.     return XT ;
23. }
```

Fig. 4 X-bit Distribution X-filling algorithm.

algorithm. This algorithm determines test pattern to detect undetected faults in U. Figure 4 shows the X-bit distribution X-filling algorithm. Explanations of each step are given. The inputs are a circuit (C), a fault dictionary of initial test set (D), an undetected fault set (U), an initial test set (T) and an X-identified test set (EXT). EXT can detect all essential faults. In Fig. 4, test set XT is initialized to EXT (line 6). MCT which denotes the minimum value of X-bit distribution cost function is initialized to infinity (line 7). The series of processing for lines 9 to 19 is iterated for each undetected fault f_i in U (line 8). A test pattern set DT_{fi}, which detects undetected fault f_i, is obtained by fault dictionary D (line 9). The series of processing for lines 11 to 17 is iterated for each test pattern t_j in DT_{fi} (line 10). The PI values of t_j required to detect only f_i is calculated. These PI values are fixed to care bits, and the other values are X-bits. Then, test pattern xt'_j, which can detect fault f_i, is obtained (line 11). CT which denotes the value of X-bit distribution cost function is calculated by XT and xt'_j (line 12). The details are described in Sect. 4.4. If the value of CT is smaller than that of MCT (line 13), CT is substituted for MCT (line 14), xt'_j is substituted for a test pattern with a minimum cost value t_{mct} (line 15) and test pattern ID j is substituted for k which denotes ID of a test pattern with a minimum cost value (line 16). t_{mct} is merged with xt_k in XT, XT is updated (line 20). A fault simulation is performed for U by XT and detected faults are eliminated from U (line 21). Finally, test set XT is returned (line 23). XT can detect all faults in fault dictionary D.

4.4 Cost Function of X-bit Distribution for PIs and X-bit Ratio

In this section, we present a cost function to control the X-bit distribution for PIs and X-bit ratio in a test set. This cost function is used in line 12 of Fig. 4.

$W(XT, p_n)$ shown in Eq. (16) is the number of care bits at primary input p_n in the test set XT with X-bits. $N_{TP}(XT)$ is the number of test patterns in XT.

$$W(XT, p_n) = \sum_{j=1}^{N_{TP}(XT)} C(xt_j, p_n) \tag{16}$$

$VX(xt_j, xt'_j)$ shown in Eq. (17) is the cost function to decide a test pattern to detect an undetected fault f_i on the proposed X-identification for test compaction. The number of test patterns in test set T to detect f_i is equal to two or more. One test pattern with the minimum cost of $VX(xt_j, xt'_j)$ is selected from XT to detect f_i.

$$VX(xt_j, xt'_j) = \sum_{n=1}^{N_{PI}} W(XT, p_n) \times X(xt_j, p_n) \times C(xt'_j, p_n) \tag{17}$$

In Eq. (17), xt'_j corresponding to t_j is a test pattern to detect only f_i, and xt_j corresponding to t_j can detect only essential faults. $X(xt_j, p_n)$ is the function for primary input p_n in xt_j. If the value of p_n in xt_j is an X-bit, $X(xt_j, p_n)$ returns 1; otherwise, it returns 0. $C(xt'_j, p_n)$ is the function for primary input p_n in test pattern xt'_j. If the value of p_n in xt'_j is a care bit, $C(xt'_j, p_n)$ returns 1; otherwise, it returns 0. Therefore, "$X(xt_j, p_n) \times C(xt'_j, p_n) = 1$" means that primary input p_n value in xt_j is an X-bit and primary input p_n value in xt'_j is a care bit. Thus, if xt'_j is selected to detect f_i, the number of care bits in xt_j increases by $\sum_{n=1}^{N_{PI}} X(xt_j, p_n) \times C(xt'_j, p_n)$. Therefore, the cost value is small when a small amount of the total number of X-bits at PI changes to care bits. Thus, when the value of $VX(xt_j, xt'_j)$ is small, the X-bit variance for PI is low and X-bit ratio is high.

An example of test pattern selection to detect undetected fault f_i is shown. Table 3 shows an example calculation of the number of care bits at PI in test set XT with X-bits. XT, obtained in line 6 of Fig. 4, can detect all essential faults. The test pattern IDs are denoted as xt_1 to xt_5. The primary input IDs are denoted as p_1 to p_7. The don't care bit is denoted as "X" and the care bit is denoted as "C".

The number of care bits for primary input p_1 is calculated. Primary input p_1 includes the care bits in test patterns xt_2 and xt_4, as shown in Table 3. Therefore, "$W(XT, p_1) = 2$" is calculated by Eq. (16).

Table 4 shows an example calculation of the cost of care bits to detect undetected fault f_i. An undetected fault f_i is detected by test patterns, t_1, t_3, and t_5 in T, whereas xt'_1, xt'_3, and xt'_5 detect only fault f_i.

As another example, we consider $VX(xt_3, xt'_3)$, which is

the value of the cost function required for test patterns xt_3

$$\begin{aligned} VX(xt_3, xt'_3) &= W(XT, p_1) \times X(xt_3, p_1) \times C(xt'_3, p_1) \\ &+ W(XT, p_2) \times X(xt_3, p_2) \times C(xt'_3, p_2) \\ &+ W(XT, p_3) \times X(xt_3, p_3) \times C(xt'_3, p_3) \\ &+ W(XT, p_4) \times X(xt_3, p_4) \times C(xt'_3, p_4) \\ &+ W(XT, p_5) \times X(xt_3, p_5) \times C(xt'_3, p_5) \\ &+ W(XT, p_6) \times X(xt_3, p_6) \times C(xt'_3, p_6) \\ &+ W(XT, p_7) \times X(xt_3, p_7) \times C(xt'_3, p_7) \\ &= (2 \times 1 \times 0) + (0 \times 1 \times 1) + (4 \times 0 \times 1) \\ &+ (5 \times 0 \times 0) + (2 \times 1 \times 0) + (2 \times 1 \times 0) \\ &+ (1 \times 1 \times 1) \\ &= 0 + 0 + 0 + 0 + 0 + 0 + 1 = 1 \end{aligned}$$

and xt'_3 to detect fault f_i. If test pattern xt'_3 detects fault f_i, the values of primary inputs p_2, p_3, and p_7 are care bits. $VX(xt_3, xt'_3)$ is calculated as follows.

The value of primary input p_3 of test pattern xt_3 is a care bit, as shown in Table 3. Thus, $X(xt_3, p_3) \times C(xt'_3, p_3)$ is 0, and so $VX(xt_3, xt'_3) = 1$ is calculated by Eq. (17). $VX(xt_1, xt'_1)$ and $VX(xt_5, xt'_5)$ are 3 and 2, respectively. From the results of the cost function, it is clear that the minimum value is $VX(xt_3, xt'_3)$. Therefore, test pattern t_3 to detect undetected fault f_i is selected. The values at p_2 and p_7 of xt_3 change from X-bits to care bits. The number of care bits of xt_3 increases from two to four. Thus, XT is updated.

Table 3 Care bits in test set XT and $W(XT, p_n)$.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
xt_1	x	x	c	c	x	c	x
xt_2	c	x	x	c	c	x	c
xt_3	x	x	c	c	x	x	x
xt_4	c	x	c	c	c	x	x
xt_5	x	x	c	c	x	c	x
$W(XT, p_n)$	2	0	4	5	2	2	1

Table 4 Care bits to detect fault f_i.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	$VX(xt_j, xt'_j)$
xt'_1	c	c	x	x	x	x	c	3
xt'_3	x	c	c	x	x	x	c	1
xt'_5	c	c	x	x	x	c	x	2
$W(XT, p_n)$	2	0	4	5	2	2	1	

5. Experimental Results

In this section, we describe the experimental results of the proposed method. The evaluation items are the X-bit ratio, the X-bit variance for PI, the execution time for X-identification and the number of test patterns after test compaction. The proposed method (VX) was compared with XID [13], DC-XID [14] and PI_Vari. VX considers both the X-bit ratio and X-bit variance for each PI. XID considers only X-bit ratio. PI_vari considers only X-bit variance for each PI. DC-XID considers only X-bit distribution for each test pattern. PI_Vari identifies X-bits to minimize the X-bit variance for PI. The algorithm of PI_Vari is almost same as that of VX. The difference is cost function of line 12 of Fig. 4. The cost function of PI_Vari used Eq. (11). The applied circuits were ITC'99 benchmark circuits and ISCAS'89 benchmark circuits. Initial test set T was generated by the ATPG tool "TetraMAX™" (Synopsys). The target fault model was a single stuck-at fault model. Two initial test sets were prepared for each circuit. One was an initial uncompacted test set T_{uc}. The other was an initial compacted test set T_c.

Table 5 shows the X-bit ratio, the X-bit variance for PI and the execution time of X-identification for uncompacted initial test sets T_{uc}. In Table 5, N_{PI} denotes the number of the PI, $N_{TP}(T_{uc})$ denotes the number of test patterns in initial uncompacted test set T_{uc}, *%X-bit* denotes the X-bit ratio in X-identified test set XT_{uc}, $s^2(XT_{uc})$ denotes the X-bit

Table 5 Results of X-identification of uncompacted initial test sets T_{uc}.

Circuits	N_{PI}	$N_{TP}(T_{uc})$	XID			DC-XID			PI_Vari			VX (Proposed)			FC(%)
			%X-bit	$s^2(XT_{uc})$	Time(sec)	%X-bit	$s^2(XT_{uc})$	Time(sec)	%X-bit	$s^2(XT_{uc})$	Time(sec)	%X-bit	$s^2(XT_{uc})$	Time(sec)	
s13207	650	589	93.73	2252	**1.13**	95.51	1884	4.46	95.57	**1530**	5.27	**95.77**	1558	5.04	99.04
s15850	600	562	93.74	1040	**1.38**	94.24	1126	11.10	**94.62**	**689**	12.46	**94.62**	749	15.98	97.95
s35932	1763	80	86.03	15	**5.22**	87.70	12	73.40	**88.73**	**7**	85.87	88.51	8	116.57	90.04
s38417	1524	1185	96.94	1524	**9.04**	97.34	1131	66.50	97.50	**745**	81.63	**97.58**	783	97.37	99.73
s38584	1462	862	96.23	1818	**11.30**	96.49	1538	800.67	96.50	**931**	319.82	**96.67**	966	283.36	95.27
b14	277	1317	82.02	78512	**27.64**	80.74	78215	174.08	80.82	**54584**	199.80	**82.43**	56903	361.15	99.47
b15	485	909	90.75	14044	**25.57**	89.77	16639	307.13	**92.38**	**8715**	466.87	91.78	9454	497.10	97.26
b17	1452	3019	96.16	30017	**65.97**	95.96	29116	1699.44	**96.84**	**20184**	3866.55	96.63	20918	3550.99	97.83
b20	522	2138	86.94	60510	**65.90**	86.09	60007	629.02	87.17	**38127**	789.90	**87.64**	41266	514.68	99.48
b21	522	2316	87.94	68423	**54.67**	86.92	67539	679.45	87.78	**44401**	803.35	**88.29**	47875	544.77	99.44
b22	767	2770	91.24	57956	**96.29**	90.46	60311	1286.69	91.30	**36979**	843.53	**91.62**	40734	665.25	99.56

Table 6 Results of X-identification of compacted initial test sets T_c.

Circuits	N_{PI}	$N_{TP}(T_c)$	XID			DC-XID			PI_Vari			VX (Proposed)			FC(%)
			%X-bit	$s^2(XT_c)$	Time(sec)	%X-bit	$s^2(XT_c)$	Time(sec)	%X-bit	$s^2(XT_c)$	Time(sec)	%X-bit	$s^2(XT_c)$	Time(sec)	
s13207	650	267	87.86	1676	**2.66**	91.13	1384	10.66	91.23	**1302**	17.31	**91.66**	1315	17.50	99.04
s15850	600	130	79.02	430	**2.91**	79.74	451	27.14	80.96	**372**	32.17	**81.32**	383	30.90	97.95
s35932	1763	21	56.81	5	**8.29**	58.61	3	91.38	58.54	**2**	117.64	**59.09**	**2**	147.87	90.04
s38417	1524	104	75.54	401	**12.10**	76.78	387	297.48	76.49	**332**	218.72	**77.38**	351	185.31	99.73
s38584	1462	143	**83.65**	451	**12.33**	82.46	482	410.04	82.72	**394**	190.14	83.61	407	183.81	95.27
b14	277	749	**77.14**	21755	**20.46**	74.97	23272	188.99	75.46	**20450**	201.05	75.73	20961	220.77	99.47
b15	485	459	85.41	5086	**18.82**	84.29	5916	308.24	**87.31**	**4160**	276.53	86.22	4641	307.84	97.26
b17	1452	1056	91.33	13530	**145.81**	90.73	14229	3509.52	**92.18**	**11960**	3627.86	91.88	12302	2461.18	97.83
b20	522	754	**72.90**	18401	**41.48**	70.68	18921	737.28	71.45	**15886**	631.87	71.49	17296	624.06	99.48
b21	522	762	**72.92**	20468	**41.21**	70.90	20537	745.84	71.76	**17666**	628.82	71.80	19194	629.65	99.44
b22	767	640	**72.14**	18375	**61.92**	70.65	18694	1426.80	70.70	**16023**	1088.42	71.25	17415	1005.29	99.56

Table 7 Results of test compaction of X-identified test sets XT_{uc}.

Circuits	$N_{TP}(T_{uc})$	XID		DC-XID		PI_Vari		VX (Proposed)		FC(%)
		#Dsatur	#DD	#Dsatur	#DD	#Dsatur	#DD	#Dsatur	#DD	
s13207	589	284	276	285	275	**271**	270	**271**	269	99.04
s15850	562	178	168	170	159	**150**	149	**150**	**147**	97.95
s35932	80	**37**	**36**	79	64	40	39	40	39	90.04
s38417	1185	187	185	157	155	**138**	**138**	140	140	99.73
s38584	862	210	209	193	188	174	172	**171**	**170**	95.27
b14	1317	1013	916	1009	896	918	870	**911**	**866**	99.47
b15	909	627	521	654	535	568	514	**552**	**504**	97.26
b17	3019	1315	1260	1294	1246	1228	1205	**1192**	**1174**	97.83
b20	2138	1283	1164	1300	1176	1133	**1080**	**1122**	1090	99.48
b21	2316	1390	1259	1398	1283	**1232**	**1194**	1250	1208	99.44
b22	2770	1152	1082	1147	1092	983	970	**964**	**957**	99.56

variance for PI in X-identified test set XT_{uc}, *Time*(*sec*) denotes the execution time of X-identification for initial test set T_{uc} and $FC(\%)$ denotes fault coverage. Fault coverage of X-identified test set was the same as that of initial test set. The *%X-bit* of VX is about 1% higher than that of XID and DC-XID for all circuits. The *%X-bit* of VX is about 0.4% higher than that of PI_Vari for s13207, s15850, s38417, s38584, b14, b20, b21, and b22. The *%X-bit* of VX is about 0.3% lower than that of PI_Vari for s35932, b15, and b17. The $s^2(XT_{uc})$ of VX is smaller than that of XID, and DC-XID for all circuits. The $s^2(XT_{uc})$ of VX is almost same as those of PI_Vari. The $s^2(XT_{uc})$ of VX was reduced from 17 to 48% (average 33%) as compared with XID and DC-XID. The *Time*(*sec*) of XID is smaller than that of DC-XID, PI_Vari, and VX for all circuits.

Table 6 shows the X-bit ratio, the X-bit variance for PI, and the execution time of X-identification for compacted initial test sets T_c. In Table 6, N_{PI} denotes the number of the PI, $N_{TP}(T_c)$ denotes the number of test patterns in initial compacted test set T_c, *%X-bit* denotes the X-bit ratio in X-identified test set XT_c, $s^2(XT_c)$ denotes the X-bit variance for PI in X-identified test set XT_c, *Time*(*sec*) denotes the execution time of X-identification for initial test set T_c and $FC(\%)$ denotes fault coverage. Fault coverage of X-identified test set was the same as that of initial test set. The *%X-bit* of VX is about 1% higher than that of *DC-XID* for all circuits. The *%X-bit* of VX is about 0.5% higher than that of PI_Vari for all circuits except for b15 and b17. The *%X-bit* of VX is about 2% higher than that of XID for s13207, s15850, s35932, s38417, b15, and b17. The *%X-bit* of VX is about 1% lower than that of XID for s38584, b14, b20, b21, and b22. The $s^2(XT_c)$ of VX is smaller than that of XID, and DC-XID for all circuits. The $s^2(XT_c)$ of VX is almost same as those of PI_Vari. The $s^2(XT_c)$ of VX was reduced from 3 to 60% (average 13%) as compared with XID and DC-XID. The *Time*(*sec*) of XID is smaller than that of DC-XID, PI_Vari, and VX for all circuits.

Table 7 shows the number of test patterns in the test set after test compaction of initial uncompacted test set T_{uc}. Two test compaction methods were applied to each test set after X-identification. One was Dsatur [9] which merges test patterns with X-bits. The other was double detection [12] which is fault simulation based test compaction. In Table 7, *#Merge* denotes the number of test patterns after applying

Table 8 Results of test compaction of X-identified test sets XT_c.

Circuits	$N_{TP}(T_c)$	XID		DC-XID		PI_Vari		VX (Proposed)		FC(%)
		#Dsatur	#DD	#Dsatur	#DD	#Dsatur	#DD	#Dsatur	#DD	
s13207	267	262	262	261	260	260	260	**259**	**259**	99.04
s15850	130	127	126	127	127	125	123	**124**	**121**	97.95
s35932	21	**21**	**21**	**21**	**21**	**21**	**21**	**21**	**21**	90.04
s38417	104	104	104	104	104	**103**	**103**	**103**	**103**	99.73
s38584	143	140	139	143	143	**135**	**134**	**135**	135	95.27
b14	749	728	710	737	708	716	705	**715**	**703**	99.47
b15	459	393	357	398	367	378	356	**372**	**352**	97.26
b17	1056	1033	1022	1028	1022	1016	1014	**1015**	**1012**	97.83
b20	754	733	729	728	727	**721**	719	**721**	**717**	99.48
b21	762	739	737	737	737	**725**	**722**	726	725	99.44
b22	640	**621**	**621**	638	638	624	624	625	624	99.56

test compation based on Dsatur to an initial test set. *#DD* denotes the number of test patterns after applying test compaction based double detection to a test set compacted by Dsatur. *FC*(%) denotes fault coverage. Fault coverage of test set after test compaction was the same as that of initial test set. The *#Merge* of VX is smaller than that of XID, DC-XID, and PI_Vari for all circuits except for s35932, s38417, and b21. The *#Merge* of VX was reduced maximum 188 patterns (average 71 patterns) as compared with XID, DC-XID, and PI_Vari. As for s35932, XID has the smallest number of test patterns. As for s38417, and b21 PI_Vari has the smallest number of test patterns. The *#DD* of VX is smaller than that of XID, DC-XID, and PI_Vari for s13207, s15850, s38584, b14, b15, b17 and b22. The *#DD* of VX was reduced maximum 135 patterns (average 39 patterns) as compared with XID, DC-XID, and PI_Vari. As for s35932, XID has the smallest number of test patterns. As for s38417, b20, and b21 PI_Vari has the smallest number of test patterns.

Table 8 shows the number of test patterns in the test set after test compaction of initial compacted test set T_c. *FC*(%) denotes fault coverage. Fault coverage of test set after test compaction was the same as that of initial test set. The *#Merge* of VX is smaller than that of XID, DC-XID, and PI_Vari for all circuits except for b21, and b22. The *#Merge* of VX was reduced maximum 26 patterns (average 8 patterns) as compared with XID, DC-XID, and PI_Vari. As for b21, PI_Vari has the smallest number of test patterns. As for b22, XID has the smallest number of test patterns. The *#DD* of VX is smaller than that of XID, DC-XID, and PI_Vari for all circuits except for s38584, b21, and b22. The *#DD* of VX was reduced maximum 15 patterns (average 6 patterns) as compared with XID, DC-XID, and PI_Vari. As for s35932, XID has the smallest number of test patterns. As for s38584, and b21 PI_Vari has the smallest number of test patterns. As for b22, XID has the smallest number of test patterns.

6. Conclusion

In this paper, we analyzed the relationship between X-bit variance for PI and the number of test patterns after test compaction. As the result of preliminary experiments, we formulated an X-identification problem for test compaction. From the formulation, we proposed a heuristic algorithm of X-identification for test compaction. The experimental results of our proposed method showed that the number of test patterns after test compaction is reduced for most circuits and the experimental results showed that the X-bit variance for PI is reduced for all circuits. The experimental results also showed that the X-bit variance for PI and the X-bit ratio affects the number of final test patterns after test compaction. Future work includes improving the don't care identification algorithm for test compaction and studying don't care identification for other fault models.

References

[1] H. Fujiwara, Logic Testing and Design for Testability, MIT Press, 1985.
[2] M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design, IEEE Press, 1995.
[3] J. Ferguson and J. Shen, Extraction and Simulation of Realistic CMOS Faults Using Inductive Fault Analysis, International Test Conference, pp.475–484, 1988.
[4] M. Renovell, P. Huc, and Y. Bertrand, "The concept of resistance interval: A new parametric model for realistic resistive bridging fault," 13th IEEE VLSI Test Symposium, pp.184–189, 1995.
[5] T.M. Storey and W. Maly, CMOS Bridging Fault Detection, International Test Conference, pp.842–851, 1990.
[6] S.J. Wang, Y.T. Chen, and K. Shu-Min Li, "Low capture power test generation for launch-off-capture transition test based on don't-care filling," IEEE International Symposium on Circuits and Systems, pp.3683–3686, 2007.
[7] A. Krstic and K.T. Cheng, Delay Fault Testing for VLSI Circuits, Kluwer Academic Pub Press, 1998.
[8] N. Jha and S. Gupta, Testing of Digital Systems, Cambridge University Press, 2002.
[9] D. Brelaz, "New methods to color the vertices of a graph," Communications of the ACM, vol.22, no.4, pp.251–256, 1979.

[10] P. Goel and B.C. Rosales, "Test generation and dynamic compaction of tests," Digest of papers of Test Conf., pp.189–192, 1979.
[11] L.N. Reddy, I. Pomeranz, and S.M. Reddy, "ROTCO: A reverse order test compaction technique," IEEE EURO-ASIC Conf., pp.189–194, 1992.
[12] S. Kajihara, I. Pomeranz, K. Kinoshita, and S.M. Reddy, "Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits," 30th ACM/IEEE Design Automation Conference, pp.102–106, 1993.
[13] K. Miyase and S. Kajihara, "XID: Don't care identification of test patterns for combinational circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.23, no.2, pp.321–326, 2004.
[14] K. Miyase, K. Noda, H. Ito, K. Hatayama, T. Aikyo, Y. Yamato, H. Furukawa, X. Wen, and S. Kajihara, "Effective IR-drop reduction in at-speed scan testing using distribution-controlling x-identification," IEEE/ACM International Conference on Computer-Aided Design, pp.52–58, 2008.
[15] M.A. Kochte, C.G. Zoellin, M.E. Imhof, and H.J. Wunderlich, "Test set stripping limiting the maximum number of specified bits," 4th IEEE International Symposium on Electronic Design, pp.581–586, 2008.

Hiroshi Yamazaki received the B.E. and M.E. degrees in Mathematical information engineering, College of industrial technology, Nihon University, in 2010 and 2012, respectively. He is graduate student of D.E. in Nihon University. His research interests are don't care identification, design for testability, and SAT-based test generation.

Motohiro Wakazono received the B.E. and M.E. degrees in Mathematical information engineering, College of industrial technology, Nihon University, in 2008 and 2010, respectively. He is with Hitachi ULSI Systems Co., Ltd. Presently his research interests are don't care identification, and don't care filling.

Toshinori Hosokawa received the B.E. degree in Electronics and Communication Engineering from Meiji University, Kawasaki, Japan, in 1987. He also received the Ph.D. degree from Meiji University in 2001. He was with Matsushita Electric Industrial Co., Ltd from 1987 to 2003. He was temporarily with Semiconductor Technology Academic Research Center (STARC) from 2000 to 2003. He was also a lecturer at Meiji University in 2001 and 2002. Presently he is a Professor at Department of Mathematical Information Engineering, College of Industrial Technology, Nihon University, Chiba, Japan. His research interests are test generation, fault simulation, design for testability, synthesis for testability, high level testing, logic simulation engine, hardware Trojan detection and hardware/software co-verification. He is a member of IEEE (Institute of Electrical & Electronics Engineers) and IPSJ (Information Processing Society of Japan).

Masayoshi Yoshimura received B.E. and M.E. degrees from Osaka University, Osaka, Japan in 1996 and 1998, respectively. In 1998, he joined Matsushita Electric Industry Co., Ltd. After working on EDA for digital systems there, he received Ph.D degree from Osaka University in 2003. In Octorver 2004, he joined Fukuoka Laboratory for Emerging & Enabling Technology of SoC (FLEETS), and engaged in EDA projects until March 2007. Currently, he is assistant professor in Kyushu University. His research interest includes automatic test pattern generation, design for testability and. He is a member of IPSJ and IEEE.