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

Hiroshi YAMAZAKI††, 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 \( N_{PI} \).
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 $t_1$ 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 $xt_j$. $V(\text{xt}_i, \text{pt}_k)$ shown in Eq. (1) is an equation that expresses the value of primary input $\text{pt}_k$ in test pattern $\text{xt}_i$.

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

Table 2 shows a test compaction operation $\cap_T$ for $V(\text{xt}_i, \text{pt}_k)$ and $V(\text{xt}_j, \text{pt}_k)$. $\Phi$ means that $V(\text{xt}_i, \text{pt}_k)$ and $V(\text{xt}_j, \text{pt}_k)$ cannot be merged.

$$cxt_{pk} = V(\text{xt}_i, \text{pt}_k) \cap_T V(\text{xt}_j, \text{pt}_k) \quad (2)$$

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

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

As shown in Eq. (4), if the result of test compaction operation $\cap_T$ for $\text{xt}_i$ and $\text{xt}_j$ includes at least one $\Phi$, $\text{xt}_i$ and $\text{xt}_j$ are incompatible.

$$\exists p_k(cxt_{pk} = \Phi) \Rightarrow COM(\text{xt}_i, \text{xt}_j) = 0 \quad (4)$$

As shown in Eq. (5), if the result of test compaction operation $\cap_T$ for $\text{xt}_i$ and $\text{xt}_j$ does not includes $\Phi$, $\text{xt}_i$ and $\text{xt}_j$ are compatible.

$$\forall p_k(cxt_{pk} \neq \Phi) \Rightarrow COM(\text{xt}_i, \text{xt}_j) = 1 \quad (5)$$

$$BP(\text{xt}_i, \text{xt}_j, \text{pt}_k)$$ shown in Eq. (6) is an equation that expresses the test compaction probability for $\text{pt}_k$ value of $\text{xt}_i$ and $\text{pt}_k$ value of $\text{xt}_j$. If both $\text{pt}_k$ values of $\text{xt}_i$ and $\text{xt}_j$ are care-bits, Eq. (6) returns $P0(\text{xt}_i, \text{pt}_k) \times P0(\text{xt}_j, \text{pt}_k) + P1(\text{xt}_i, \text{pt}_k) \times P1(\text{xt}_j, \text{pt}_k)$; otherwise, it returns 1.

$$BP(\text{xt}_i, \text{xt}_j, \text{pt}_k) = \begin{cases} P0(\text{xt}_i, \text{pt}_k) \times P0(\text{xt}_j, \text{pt}_k) + P1(\text{xt}_i, \text{pt}_k) \times P1(\text{xt}_j, \text{pt}_k) & \text{if both } \text{pt}_k \text{ value of } \text{xt}_i \\ 1 & \text{otherwise (pt}_k \text{ value of } \text{xt}_i \text{ and/or } \text{xt}_j \text{ are X-bits)} \end{cases} \quad (6)$$

In Eq. (6), $P0(\text{xt}_i, \text{pt}_k)$ denotes probability that $\text{pt}_k$ value of $\text{xt}_i$ is 0, and $P1(\text{xt}_i, \text{pt}_k)$ denotes probability that $\text{pt}_k$ values of $\text{xt}_i$ is 1. If $\text{pt}_k$ values of $\text{xt}_i$ and/or $\text{xt}_j$ are X-bits, the $\text{pt}_k$ values are compatible. Therefore, test compaction probability is 1 when X-bits are included.

$$PCOM(\text{xt}_i, \text{xt}_j) = \prod_{k=1}^{N_{pt}} BP(\text{xt}_i, \text{xt}_j, \text{pt}_k) \quad (7)$$

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.

\[
X(x_i, p_k) = \begin{cases} 
1 & \text{if } p_k \text{ value of } x_i \text{ is an } X\text{-bit (care bit) } \\
0 & \text{otherwise (X-bit)} 
\end{cases}
\]  

(8)

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

\[
C(x_i, p_k) = \begin{cases} 
1 & \text{if } p_k \text{ value of } x_i \text{ is a care bit } \\
0 & \text{otherwise (X-bit)} 
\end{cases}
\]  

(9)

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

\[
A_X(XT) \text{ shown in Eq. (10) is the average value of the number of } X\text{-bits for PI in } XT.
\]

\[
A_X(XT) = \frac{1}{N_{PI}} \sum_{m=1}^{N_{PI}} \left( \sum_{k=1}^{N_{TX}^{(XT)}} X(x_i, p_m) \right)
\]  

(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 \( XT \).

\[
s^2(XT) = \frac{1}{N_{PI}} \sum_{i=1}^{N_{PI}} \left( A_X(XT) - \sum_{j=1}^{N_{TX}^{(XT)}} X(x_i, p_j) \right)^2
\]  

(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 \( XT_v = \{ x_{tm}, x_{tm} \} \) and \( XT_u = \{ x_{tm}, x_{tm} \} \) 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-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-ID \( u \) that makes the number of X-bits at each PI in a test set uniform. Therefore, X-bit variance for PI is \( s^2(XT_v) > s^2(XT_u) \).

\[
cc(x_i, x_j) \text{ shown in Eq. (12) is an equation that expresses the number of PI's whose } p_k \text{ values of } x_i \text{ and } x_j \text{ are care bits.}
\]

\[
cc(x_i, x_j) = \sum_{k=1}^{N_{PP}} (C(x_i, p_k) \times C(x_j, p_k))
\]  

(12)

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

\[
cx(x_i, x_j) = \sum_{k=1}^{N_{PP}} (X(x_i, p_k) \times C(x_j, p_k) + C(x_i, p_k) \times X(x_j, p_k))
\]  

(13)

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

\[
xx(x_i, x_j) = \sum_{k=1}^{N_{PP}} (X(x_i, p_k) \times X(x_j, p_k))
\]  

(14)

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

\[
cc(x_{tm}, x_{tm}) + 2xx(x_{tm}, x_{tm}) = cc(x_{tm}, x_{tm}) + cc(x_{tm}, x_{tm}) = N_{PP}.
\]

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

\[
cc(x_{tm}, x_{tm}) + cc(x_{tm}, x_{tm}) + xx(x_{tm}, x_{tm}) = cc(x_{tm}, x_{tm}) + cc(x_{tm}, x_{tm}) + xx(x_{tm}, x_{tm}) = N_{PP}.
\]

From \( s^2(XT_v) > s^2(XT_u) \Rightarrow xx(x_{tm}, x_{tm}) > xx(x_{tm}, x_{tm}) \),

\[
cc(x_{tm}, x_{tm}) > cc(x_{tm}, x_{tm}).
\]

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

\[
PCOM(x_{tm}, x_{tm}) = 0.5cc(x_{tm}, x_{tm}) \times 1.0s^2(x_{tm}, x_{tm}) \times 1.0xx(x_{tm}, x_{tm})
\]

\[
= 0.5cc(x_{tm}, x_{tm}).
\]

test compaction probability of \( XT_u \) is calculated as

\[
PCOM(x_{tm}, x_{tm}) = 0.5cc(x_{tm}, x_{tm}) \times 1.0s^2(x_{tm}, x_{tm}) \times 1.0xx(x_{tm}, x_{tm})
\]

\[
= 0.5cc(x_{tm}, x_{tm}).
\]

From \( cc(x_{tm}, x_{tm}) > cc(x_{tm}, x_{tm}) \), \( PCOM(x_{tm}, x_{tm}) < PCOM(x_{tm}, x_{tm}) \).

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, \( RXT \) were generated for 1000 kinds of variance values. As expected, \( RXT \) lost fault coverage
compared with $T$. In the second step, test compaction [9] was performed for each $RXT_c$.

### 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 $RXT_c$. 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.

**Inputs:** initial test set $T$

**Outputs:** test set with X-bits $XT$

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

**Minimization:** $s^2(XT), \text{subject to } D(T) = D(XT)$ (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
The inputs are a circuit (X-filling algorithm. Explanations of each step are given. set (DU\text{undetected faults in algorithm. This algorithm determines test pattern to detect undetected faults in U. Figure 4 shows the X-bit distribution X-filling algorithm. The series of processing for lines 9 to 19 is iterated for each test pattern tj is iterated for each test pattern xtj in XT (line 10). The PI values of tj required to detect only fi 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 fi, 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_{med} (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_{med} 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.

\[ W(XT, p_n) = \sum_{j=1}^{N_{TP}(XT)} C(x_{tj}, p_n) \]  

(16)

\[ VX(x_{tj}, x_{t'j}) = \sum_{n=1}^{N_{TP}} W(XT, p_n) \times X(x_{tj}, p_n) \times C(x_{t'j}, p_n) \]  

(17)

In Eq. (17), x_{tj} corresponding to tj is a test pattern to detect only fi, and xtj corresponding to tj can detect only essential faults. X(x_{tj}, p_n) is the function for primary input p_n in x_{tj}. If the value of p_n in x_{tj} is an X-bit, X(x_{tj}, p_n) returns 1; otherwise, it returns 0. C(x_{tj}, p_n) is the function for primary input p_n in test pattern xtj. If the value of p_n in xtj is a care bit, C(x_{tj}, p_n) returns 1; otherwise, it returns 0. Therefore, “X(x_{tj}, p_n) \times C(x_{tj}, p_n) = 1” means that primary input p_n value in x_{tj} is an X-bit and primary input p_n value in xtj is a care bit. Thus, if xtj is selected to detect fi, the number of care bits in x_{tj} increases by \sum_{n=1}^{N_{TP}} X(x_{tj}, p_n) \times C(x_{t'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(x_{tj}, x_{t'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 fi 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_5. 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_3 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 fi. An undetected fault fi is detected by test patterns, t_1, t_2, and t_3 in T, whereas xt'_{t_1}, xt'_{t_2}, and xt'_{t_3} detect only fault fi.

As another example, we consider VX(x_{t_5}, x_{t'_{t_5}}), which is...
the value of the cost function required for test patterns $x_{t3}$

$$
VX(x_{t3}, x'_{t3}) = W(XT, p_1) \times X(x_{t3}, p_1) \times C(x'_{t3}, p_1) \\
+ W(XT, p_2) \times X(x_{t3}, p_2) \times C(x'_{t3}, p_2) \\
+ W(XT, p_3) \times X(x_{t3}, p_3) \times C(x'_{t3}, p_3) \\
+ W(XT, p_4) \times X(x_{t3}, p_4) \times C(x'_{t3}, p_4) \\
+ W(XT, p_5) \times X(x_{t3}, p_5) \times C(x'_{t3}, p_5) \\
+ W(XT, p_6) \times X(x_{t3}, p_6) \times C(x'_{t3}, p_6) \\
+ W(XT, p_7) \times X(x_{t3}, p_7) \times C(x'_{t3}, p_7) \\
= (2 \times 1 \times 0) + (0 \times 1 \times 1) + (4 \times 0 \times 1) \\
+ (5 \times 0 \times 0) + (2 \times 1 \times 0) + (2 \times 1 \times 0) \\
+ (1 \times 1 \times 1) \\
= 0 + 0 + 0 + 0 + 0 + 1 = 1
$$

and $x'_{t3}$ to detect fault $f_i$. If test pattern $x'_{t3}$ detects fault $f_i$.

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_VX. VX considers both the X-bit ratio and X-bit variance for each PI. XID considers only X-bit ratio. PI_VX considers only X-bit variance for each PI. DC-XID considers only X-bit distribution for each test pattern. PI_VX identifies X-bits to minimize the X-bit variance for PI. The algorithm of PI_VX is almost the same as that of VX. The difference is cost function of line 12 of Fig. 4. The cost function of PI_VX 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 3 shows 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_VX. VX considers both the X-bit ratio and X-bit variance for each PI. XID considers only X-bit ratio. PI_VX considers only X-bit variance for each PI. DC-XID considers only X-bit distribution for each test pattern. PI_VX identifies X-bits to minimize the X-bit variance for PI. The algorithm of PI_VX is almost the same as that of VX. The difference is cost function of line 12 of Fig. 4. The cost function of PI_VX 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 $T_{uc}$.

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, $N_{PT}$ denotes the number of the PI, $N_{TP}(T_{uc})$ denotes the number of test patterns in initial uncompacted test set $T_{uc}$, $%X\text{-bit}$ denotes the X-bit ratio in X-identified test set $X_{Tuc}$, $s^2(X_{Tuc})$ denotes the X-bit
Table 6 Results of X-identification of compacted initial test sets $T_c$.

<table>
<thead>
<tr>
<th>Circuits</th>
<th>$N_T(T_c)$</th>
<th>%X-bit</th>
<th>$s^2(X_{T_c})$</th>
<th>Time (sec)</th>
<th>%X-bit</th>
<th>$s^2(X_{T_c})$</th>
<th>Time (sec)</th>
<th>%X-bit</th>
<th>$s^2(X_{T_c})$</th>
<th>Time (sec)</th>
<th>FC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>650</td>
<td>87.86</td>
<td>1676</td>
<td>2.66</td>
<td>91.13</td>
<td>1384</td>
<td>10.66</td>
<td>91.23</td>
<td>1302</td>
<td>17.31</td>
<td>99.04</td>
</tr>
<tr>
<td>s15850</td>
<td>600</td>
<td>79.02</td>
<td>430</td>
<td>2.91</td>
<td>79.74</td>
<td>451</td>
<td>27.14</td>
<td>80.96</td>
<td>372</td>
<td>32.17</td>
<td>97.95</td>
</tr>
<tr>
<td>s35932</td>
<td>1763</td>
<td>56.61</td>
<td>5</td>
<td>8.20</td>
<td>58.61</td>
<td>3</td>
<td>91.38</td>
<td>58.54</td>
<td>2</td>
<td>117.64</td>
<td>90.04</td>
</tr>
<tr>
<td>s38417</td>
<td>1524</td>
<td>75.54</td>
<td>401</td>
<td>12.10</td>
<td>76.78</td>
<td>387</td>
<td>297.48</td>
<td>76.49</td>
<td>332</td>
<td>216.72</td>
<td>97.28</td>
</tr>
<tr>
<td>s38584</td>
<td>1462</td>
<td>83.65</td>
<td>451</td>
<td>12.33</td>
<td>82.46</td>
<td>482</td>
<td>410.04</td>
<td>82.72</td>
<td>394</td>
<td>190.14</td>
<td>95.27</td>
</tr>
<tr>
<td>b14</td>
<td>277</td>
<td>77.14</td>
<td>2175</td>
<td>20.46</td>
<td>74.97</td>
<td>23272</td>
<td>188.99</td>
<td>75.46</td>
<td>20450</td>
<td>201.05</td>
<td>99.47</td>
</tr>
<tr>
<td>b15</td>
<td>485</td>
<td>85.41</td>
<td>5086</td>
<td>18.82</td>
<td>84.29</td>
<td>5916</td>
<td>308.24</td>
<td>87.31</td>
<td>4160</td>
<td>276.53</td>
<td>97.26</td>
</tr>
<tr>
<td>b17</td>
<td>1452</td>
<td>91.33</td>
<td>13530</td>
<td>145.81</td>
<td>90.73</td>
<td>14229</td>
<td>3509.52</td>
<td>92.18</td>
<td>11960</td>
<td>3627.68</td>
<td>91.88</td>
</tr>
<tr>
<td>b20</td>
<td>522</td>
<td>72.90</td>
<td>18401</td>
<td>41.48</td>
<td>70.68</td>
<td>18921</td>
<td>737.28</td>
<td>71.45</td>
<td>15886</td>
<td>631.87</td>
<td>94.48</td>
</tr>
<tr>
<td>b21</td>
<td>522</td>
<td>72.92</td>
<td>20468</td>
<td>41.21</td>
<td>70.90</td>
<td>20537</td>
<td>745.84</td>
<td>71.76</td>
<td>17666</td>
<td>626.82</td>
<td>94.48</td>
</tr>
<tr>
<td>b22</td>
<td>767</td>
<td>72.14</td>
<td>18375</td>
<td>61.92</td>
<td>70.65</td>
<td>18694</td>
<td>1426.80</td>
<td>70.70</td>
<td>16023</td>
<td>1088.42</td>
<td>99.56</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Circuits</th>
<th>$N_T(T_{uc})$</th>
<th>%X-bit</th>
<th>$s^2(X_{T_{uc}})$</th>
<th>FC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>589</td>
<td>284</td>
<td>276</td>
<td>99.04</td>
</tr>
<tr>
<td>s15850</td>
<td>562</td>
<td>178</td>
<td>168</td>
<td>97.95</td>
</tr>
<tr>
<td>s35932</td>
<td>80</td>
<td>37</td>
<td>36</td>
<td>90.04</td>
</tr>
<tr>
<td>s38417</td>
<td>1185</td>
<td>187</td>
<td>185</td>
<td>99.73</td>
</tr>
<tr>
<td>s38584</td>
<td>862</td>
<td>210</td>
<td>209</td>
<td>95.27</td>
</tr>
<tr>
<td>b14</td>
<td>1317</td>
<td>1031</td>
<td>916</td>
<td>99.47</td>
</tr>
<tr>
<td>b15</td>
<td>909</td>
<td>627</td>
<td>521</td>
<td>97.26</td>
</tr>
<tr>
<td>b17</td>
<td>3019</td>
<td>1315</td>
<td>1260</td>
<td>97.83</td>
</tr>
<tr>
<td>b20</td>
<td>2138</td>
<td>1283</td>
<td>1164</td>
<td>94.48</td>
</tr>
<tr>
<td>b21</td>
<td>2316</td>
<td>1390</td>
<td>1259</td>
<td>99.56</td>
</tr>
<tr>
<td>b22</td>
<td>2770</td>
<td>1152</td>
<td>1082</td>
<td></td>
</tr>
</tbody>
</table>

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 0.5% 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, b17, and b22. The %X-bit of VX is about 1% lower than that of XID for s38584, b14, b20, b21, and b22. The $s^2(X_{T_{uc}})$ of VX is smaller than that of XID, and DC-XID for all circuits. The $s^2(X_{T_{uc}})$ of VX is almost the same as those of PI Vari. The $s^2(X_{T_{uc}})$ of VX was reduced from 17 to 48% (average 33%) as compared with XID and DC-XID. The Time (sec) of VX is smaller than that of 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, $N_{T_{uc}}$ denotes the number of the PI, $N_{T_{uc}}(T_c)$ 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 $XT_c$, $s^2(X_{T_c})$ 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 test sets. 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 $s^2(X_{T_c})$ of VX is smaller than that of XID, and DC-XID for all circuits. The $s^2(X_{T_c})$ of VX is almost the same as those of PI Vari. The $s^2(X_{T_c})$ of VX was reduced from 3 to 60% (average 13%) as compared with XID and DC-XID. The Time (sec) of VX 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 $XT_c$.

<table>
<thead>
<tr>
<th>Circuits</th>
<th>$N_{\text{v}(T_c)}$</th>
<th>XID</th>
<th>DC-XID</th>
<th>PI Vari</th>
<th>VX (Proposed)</th>
<th>FC(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>267</td>
<td>262</td>
<td>262</td>
<td>261</td>
<td>260</td>
<td>259</td>
</tr>
<tr>
<td>s15850</td>
<td>130</td>
<td>127</td>
<td>126</td>
<td>127</td>
<td>125</td>
<td>124</td>
</tr>
<tr>
<td>s35932</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>s38417</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>103</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>s38584</td>
<td>143</td>
<td>140</td>
<td>139</td>
<td>143</td>
<td>135</td>
<td>135</td>
</tr>
<tr>
<td>b14</td>
<td>749</td>
<td>728</td>
<td>710</td>
<td>737</td>
<td>716</td>
<td>715</td>
</tr>
<tr>
<td>b15</td>
<td>459</td>
<td>393</td>
<td>357</td>
<td>398</td>
<td>378</td>
<td>372</td>
</tr>
<tr>
<td>b17</td>
<td>1056</td>
<td>1033</td>
<td>1022</td>
<td>1028</td>
<td>1016</td>
<td>1015</td>
</tr>
<tr>
<td>b20</td>
<td>754</td>
<td>733</td>
<td>729</td>
<td>728</td>
<td>721</td>
<td>721</td>
</tr>
<tr>
<td>b21</td>
<td>762</td>
<td>739</td>
<td>737</td>
<td>737</td>
<td>725</td>
<td>726</td>
</tr>
<tr>
<td>b22</td>
<td>640</td>
<td>621</td>
<td>621</td>
<td>638</td>
<td>624</td>
<td>625</td>
</tr>
</tbody>
</table>

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


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 2010 and 2012, 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 October 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.