Copyright © 2005 American Scientific Publishers
All rights reserved
Printed in the United States of America

*Journal of
Low Power Electronics*
Vol. 1, 1–12, 2005

# Efficient Test Set Modification for Capture Power Reduction

Xiaoqing Wen,[1,*] Tatsuya Suzuki,[1] Seiji Kajihara,[1] Kohei Miyase,[2] Yoshihiro Minamoto,[2] Laung-Terng Wang,[3] and Kewal K. Saluja[4]

[1] *Department of CSE, Kyushu Institute of Technology, Iizuka 820-8502, Japan*
[2] *Innovation Plaza Fukuoka, Japan Science and Technology Agency, Fukuoka 814-0001, Japan*
[3] *SynTest Technologies, Inc., 505 S. Pastoria Ave., Suite 101, Sunnyvale, CA 94086, USA*
[4] *Dept. of ECE, University of Wisconsin - Madison, Madison, WI 53706, USA*

(Received: 24 August 2005; Accepted: 29 November 2005)

The occurrence of high switching activity when the response to a test vector is captured by flip-flops in scan testing may cause excessive IR drop, resulting in significant test-induced yield loss. This paper addresses the problem with a novel method based on test set modification, featuring (1) a new constrained $X$-identification technique that turns a properly selected set of bits in a fully-specified test set into $X$-bits without fault coverage loss, and (2) a new LCP (low capture power) $X$-filling technique that optimally assigns 0's and 1's to the $X$-bits for the purpose of reducing the switching activity of the resulting test set in capture mode. This method can be readily applied in any test generation flow for capture power reduction without any impact on area, timing, test set size, and fault coverage.

**Keywords:** Low Capture Power, $X$-Identification, $X$-Filling.

## 1. INTRODUCTION

The combination of full-scan design and automatic test pattern generation (ATPG) forms the basis of scan testing. Due to its simplicity and efficiency, as well as wide support from designers, tool vendors, and tester makers, scan testing will continue to be a mainstream test scheme in foreseeable future.[1]

In a full-scan sequential circuit, all functional flip-flops (F/Fs) are replaced with scan F/Fs that operate in two modes: shift and capture. In shift mode, scan F/Fs form one or more scan chains directly accessible from a tester. This mode is used to load a test vector through shift-in or to observe a test response through shift-out, for the combinational portion of the sequential circuit. In capture mode, scan F/Fs operate as functional F/Fs and load the response of the combinational portion to a test vector preparatory to shift-out later in shift mode. Therefore, the task of testing a full-scan sequential circuit is reduced to that of testing its combinational portion, in that now it is sufficient to generate test vectors only for the combinational portion by using combinational ATPG.[1]

*Author to whom correspondence should be addressed.
Email: wen@cse.kyutech.ac.jp

Despite many advantages of scan testing, its applicability is being severely challenged recently by the following three problems: test data volume, test application time, and test power dissipation.

The problems of test data volume and test application time are caused by larger gate and F/F counts, longer scan chains, and the use of complex fault models, such as the transition delay fault model, all inevitable in the deep sub-micron (DSM) era. These problems can be addressed with test compaction, multi-capture clocking, and compression-decompression,[2] etc.

The power dissipation of a CMOS circuit consists of static dissipation due to leakage current and dynamic dissipation due to switching activity, with the latter being dominant. Dynamic power dissipation in scan testing occurs in both shift and capture modes. In shift mode, a test vector or a test response is shifted through all scan chains of a full-scan circuit serially, resulting in shift power dissipation. In capture mode, the response of the combinational portion of a full-scan circuit to a test vector is loaded into all F/Fs in parallel, resulting in pronounced capture power dissipation when the test vector and its corresponding response have opposite logic values for a large number of F/Fs. It has been shown that test power dissipation, including shift power dissipation and capture power

dissipation, is significantly higher than functional power dissipation.[3]

High test power dissipation is causing more and more serious problems in scan testing, especially for large-scale, high-speed, and low-power DSM integrated circuits.[4,5] Generally, there are two types of impact of test power dissipation: accumulative and instantaneous.

The accumulative impact often occurs in shift mode when a large number of clock pulses are applied. In shift mode, a test vector or a test response is shifted through all scan chains with several hundreds to several thousands of clock pulses, depending on the maximum scan chain length. As a result, the accumulative impact often manifests itself as excessive heat, which may permanently damage the circuit under test or reduce its reliability by causing accelerated electromigration.[6]

The instantaneous impact may occur in both shift and capture modes, even for one clock pulse. In this case, high switching activity for the clock pulse may cause IR drop, which is a drop in power supply voltage due to a current ($I$) flowing through a resistive transistor network ($R$). Such a power supply voltage drop in a circuit causes F/F malfunction and/or increases circuit delay. This can result in faulted test responses, leading to test-induced yield loss on top of process-related yield loss.[7,8]

Excessive test power dissipation in scan testing is becoming more and more pronounced and serious as a cost-factor and/or a yield-killer. Thus, there exists a strong need for test power reduction in order to maintain the applicability of scan testing. Generally, alleviating the accumulative impact requires reducing both average and peak power dissipation in shift mode, and alleviating the instantaneous impact requires reducing peak power dissipation in both shift and capture modes.

Many techniques have been proposed for shift power reduction, mostly based on four major approaches, namely scheduling, test vector manipulation, circuit modification, and scan chain modification. Test scheduling[9] takes the power budget into consideration when selecting modules to be tested simultaneously. Test vector manipulation includes power-aware ATPG,[10–12] static compaction,[13] test vector modification,[14] test vector reordering,[15] test vector compression,[16] and coding.[17] Circuit modification includes transition blocking,[18] clock gating,[19] and multi-duty scan.[8] Scan chain modification includes scan chain reordering,[16,20] scan chain partitioning,[21] and scan chain modification.[22]

Compared with shift power reduction, capture power reduction is more difficult due to the following reasons:

• Capture is often conducted for all clock domains simultaneously since capturing test responses one clock domain at a time may lead to excessive memory usage, long run time, and more test vectors in ATPG. Capturing for all clock domains simultaneously, however, dramatically increases the possibility of high switching activity.

• Capture clock timing is far more rigid than shift clock timing, making it difficult to achieve capture power reduction by manipulating capture clock timing in terms of frequencies and phases.

• Functional paths are activated in capture mode, making it risky to use the circuit modification approach to reduce state transitions at F/Fs for capture power reduction.

These characteristics of the capture mode in scan testing make it preferable to achieve capture power reduction through test vector manipulation, instead of clock or circuit modification. Test vector manipulation includes test vector generation[23–25,27] and test set modification.[26,27] Capture power reduction based on test set modification is especially beneficial since it can achieve capture power reduction without any impact on area, timing, test set size, and fault coverage. For this reason, this paper uses the approach of test set modification for capture power reduction.

Previous capture power reduction methods based on test set modification include those proposed in Ref. [26, 27]. They first conduct $X$-identification to find don't care bits ($X$-bits) in a fully-specified test set without incurring fault coverage loss, and then conduct $X$-filling to assign 0's and 1's to the $X$-bits for capture power reduction. However, these methods may suffer from the following problems:

• The run time of the method described in Ref. [26] may be long. This is because it identifies $X$-bits with a simulation-based procedure called bit-stripping, which requires conducting fault simulation for all test vectors and all bits in a test vector. In addition, $X$-bits are found by checking one bit at a time in one test vector, without taking vector-dependency in a test set and bit-dependency in a test vector into consideration. This may result in a smaller number of $X$-bits, thus limiting its effect in capture power reduction.

• The method described in Ref. [27] uses a more aggressive $X$-identification procedure,[28] which is based on justification and implication techniques that are typically used in ATPG, and targets a test set as a whole. The procedure needs less fault simulation time and can identify more $X$-bits, which increases the chance of achieving better effect in capture power reduction. However, the procedure provides no control on how identified $X$-bits are distributed in a test set, and some resulting $X$-bit distributions may not help much in capture power reduction. In addition, the method in Ref. [27] uses a simple $X$-filling technique that may not be globally effective in reducing capture power dissipation.

This paper addresses the above problems with a novel capture power reduction method based on test set modification. Two unique techniques are proposed: First, a constraint generation technique is used in $X$-identification that turns a properly selected set of bits in a fully-specified test set into $X$-bits in order to achieve a more effective $X$-bit

distribution. Second, a partial-symbolic $X$-simulation technique is used in $X$-filling to determine more effective logic values for $X$-bits. These two techniques make it possible to achieve more effective capture power reduction.

Note that the new capture power reduction method of this paper can be applied effectively with any shift power reduction method that are not based on test set modification. For example, one can use the multi-duty scan method[8] for shift power reduction as it just uses different shift clock phases in shift mode to prevent a large number of scan F/Fs from operating simultaneously. Since this method is totally independent from any test set, it can work effectively with any capture power reduction based on test set modification. This way, reduction in both shift power and capture power can be achieved in one scheme.

The rest of the paper is organized as follows. Section 2 describes the research background. Section 3 presents the new method for capture power reduction. Section 4 shows experimental results, and Section 5 concludes the paper.

## 2. BACKGROUND

### 2.1. Capture Power Reduction

A general full-scan circuit is shown in Figure 1, which consists of a combinational portion with primary inputs (PIs) and primary outputs (POs), as well as scan F/Fs corresponding to pseudo primary inputs (PPIs) and pseudo primary outputs (PPOs). In Figure 1, $v$ is a test vector. The PI and PPI bits in $v$ are denoted by $\langle v\text{: PI}\rangle$ and $\langle v\text{: PPI}\rangle$, respectively. The combinational portion implements logic function $f$, and its functional response to $v$ is $f(v)$. The PO and PPO bits in $f(v)$ are denoted by $\langle f(v)\text{: PO}\rangle$ and $\langle f(v)\text{: PPO}\rangle$, respectively.

If a bit $x$ in $\langle v\text{: PPI}\rangle$ and its corresponding bit $y$ in $\langle f(v)\text{: PPO}\rangle$ have opposite logic values at a scan F/F as shown in Figure 2, a capture transition occurs at its output in capture mode when SE (Scan Enable) is 0. In this case, the bit $x$ in the test vector $v$ is called a capture-transition-causing bit. It has been demonstrated that the number of capture transitions for a test vector is closely correlated with the level of switching activity caused by the test vector.[13] Therefore, capture power reduction is often realized through reducing capture transitions.[23–27]

Given a fully-specified test set $T$, the maximum number of capture transitions per test vector in $T$ is denoted by

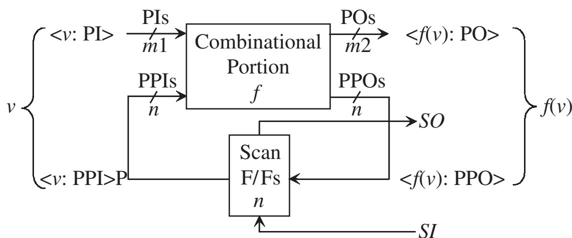



**Fig. 1.** A general full-scan circuit.

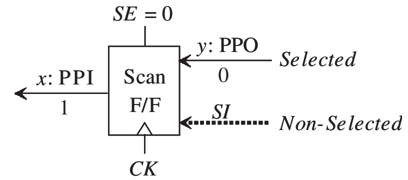



**Fig. 2.** Capture transition at a scan F/F.

MCT($T$). In order to reduce the instantaneous impact of capture power dissipation in scan testing conducted with $T$, it is necessary to reduce MCT($T$) as much as possible. In this paper, we try to achieve this goal by modifying $T$ without affecting its size or fault coverage.

Modifying a given test set $T$ for capture transition reduction typically involves two steps: $X$-Identification and $X$-filling. $X$-Identification is to convert some bits in $T$ into $X$-bits without reducing its fault coverage. $X$-filling is to assign proper 0's and 1's to the $X$-bits so that MCT($T$) is reduced. More discussions on these two steps are given in 2.2 and 2.3.

### 2.2. X-Identification

$X$-identification described in Ref. [26] is based on a procedure called bit-stripping. It is performed on one test vector $v$ at a time for a test set $T$. Each test vector $v$ is first fault-simulated to determine the set of essential faults, denoted by $F(v)$, that are only detected by $v$ and by no other test vector in $T$. Then the first bit in $v$ is tentatively changed to an $X$-bit and 3-valued fault simulation is conducted to check if all the faults in $F(v)$ are still detected. If so, the bit is kept as an $X$-bit, otherwise the bit is restored to its previous value. This process is repeated for all bits in $v$.

The run time of bit-stripping is roughly proportional to the product of the number of faults, the number of test vectors, and the number of bits in a test vector, which can be long for a large circuit. In addition, the number of $X$-bits identified by bit-stripping depends on the order of test vectors processed in a test set and the order of bits processed in a test vector. No solution is available yet for finding an optimal order to maximize the number of identified $X$-bits.

$X$-identification described in Ref. [27] is based on a more aggressive procedure called XID. It is performed on all test vectors in a whole test set simultaneously, instead of on one individual test vector at a time. First, essential faults for all test vectors are determined, and then logic values at some bits in the whole test set for detecting all essential faults of each test vector are obtained with justification and implication techniques that are typically used in ATPG. Other bits whose logic values are not determined are left as $X$-bits after additional adjustment is done to detect all faults.

The run time of XID is roughly proportional to the product of the number of faults and the number of test vectors,

| <v1: PPI> | <f(v1): PPO> | <v2: PPI> | <f(v2): PPO> |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 3 | | 2 | |

(a) Original capture transitions

| <v1: PPI> | <f(v1): PPO> | <v2: PPI> | <f(v2): PPO> |
|---|---|---|---|
| 0 | 0 | $X \leftarrow$ **0** | 0 |
| $X \leftarrow$ **0** | 0 | 0 | 0 |
| 0 | 1 | $X \leftarrow$ **1** | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 2 | | 0 | |

(b) $X$-bit distribution A

| <v1: PPI> | <f(v1): PPO> | <v2: PPI> | <f(v2): PPO> |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| $X \leftarrow$ **0** | 0 | 0 | 0 |
| $X \leftarrow$ **1** | 1 | $X \leftarrow$ **1** | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | | 1 | |

(c) $X$-bit distribution B

**Fig. 3.** Impact of $X$-distribution.

which is generally shorter than that of bit-stripping.[26] In addition, more $X$-bits are often identified by the XID procedure since all test vectors are processed simultaneously with sophisticated and algorithmic techniques of justification and implication.

However, the XID procedure is unconstrained in that it may convert any bit in a test set $T$ into an $X$-bit, resulting in an unpredictable $X$-bit distribution. This is undesirable since some $X$-bit distributions may not be effective in reducing MCT($T$). This is illustrated in Figure 3. As shown in Figure 3(a), the original test set $T$ has two test vectors $v1$ and $v2$, with 3 and 2 capture transitions, respectively. That is, MCT($T$) is 3. Figure 3(b) and Figure 3(c) show two $X$-identification results for $T$, both with 3 $X$-bits. For $X$-bit distribution A, $X$-filling can reduce the number of capture transitions of $v1$ to 2 and that of $v2$ to 0. In this case, MCT($T$) becomes 2. For $X$-bit distribution B, $X$-filling can reduce the number of capture transitions of $v1$ to 1 and that of $v2$ to 1. In this case, MCT($T$) becomes 1. Obviously, $X$-bit distribution B is better for the purpose of reducing MCT($T$).

Another issue with the XID procedure is that it may identify $X$-bits at non-capture-transition-causing bits. In Figure 3(a), for example, the first bit in $v1$ is a non-capture-transition-causing bit. Obviously, turning this bit into an $X$-bit will adversely affect capture transition reduction.

## 2.3. X-Filling

After $X$-bits are identified by $X$-identification for a test set $T$, $X$-filling is conducted to assign proper 0's and 1's to the $X$-bits so that $MCT(T)$ is reduced. There are two approaches to $X$-filling, depending on whether only $X$-bits in a test vector are targeted or $X$-bits in the response to the test vector are also targeted. Generally, taking $X$-bits in a test vector and its response into consideration in $X$-filling can lead to better results in capture power reduction. The LCP (low capture power) $X$-filling method described in Ref. [27] is a typical method based this approach.

In order to conduct LCP $X$-filling for a test vector $v$ with at least one $X$-bit, 3-valued logic simulation on {0, 1, $X$} is first conducted for $v$ to obtain its response $f(v)$ as shown in Figure 1. Then, the LCP $X$-filling method processes each bit-pair between $\langle v$: PPI$\rangle$ and $\langle f(v)$: PPO$\rangle$ if there is an $X$-bit in $\langle v$: PPI$\rangle$, $\langle f(v)$: PPO$\rangle$, or both. Such a bit-pair is called an $X$-bit-pair, which has three types as shown in Table I.

Depending on the type of an $X$-bit-pair, the LCP $X$-filling method first determines necessary logic values for the $X$-bits in the $X$-bit-pair, and then use techniques of assignment, justification, or their combination to bring the determined logic values to the $X$-bits. Examples are shown in Figure 4.

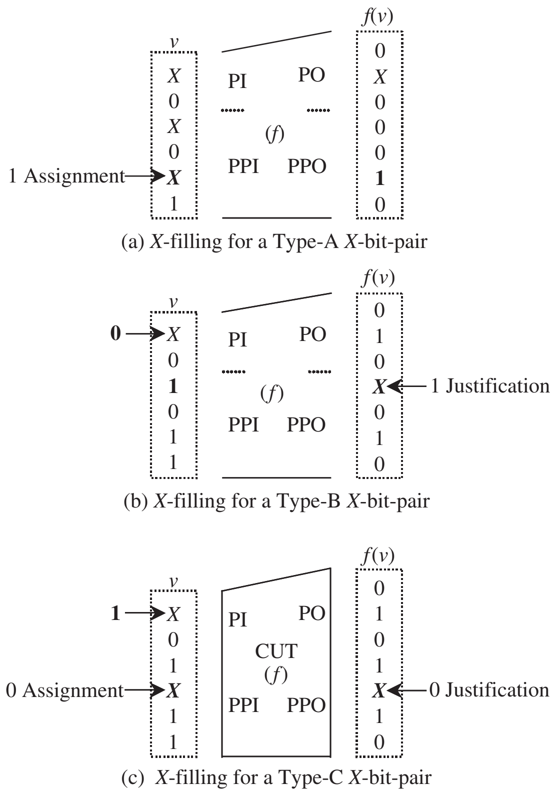
Obviously, $X$-filling for a Type-A $X$-bit-pair is the easiest and will always be successful since it only needs to assign a logic value to a pseudo primary input. $X$-filling for a Type-B $X$-bit-pair is relatively difficult since it requires justifying a logic value on a pseudo primary output. $X$-filling for a Type-C $X$-bit-pair is also relatively difficult and may take a longer time since it may need two assignment-justification trials. The effort of $X$-filling for each type of $X$-bit-pair is also summarized in Table I.

However, the LCP $X$-filling method is conducted for one $X$-bit-pair at a time. Especially, the logic values for the $X$-bits in an $X$-bit-pair is determined by the rules shown in Table I, without taking the dependency among different $X$-bit-pairs into consideration. This may adversely affect capture transition reduction. An example is shown in Figure 5. It is assumed that $\langle v\text{: PPI}\rangle = \langle a, b, c\rangle = \langle 1, 1, 1\rangle$ and $\langle f(v)\text{: PPO}\rangle = \langle x, y, z\rangle = \langle 0, 1, 1\rangle$ before $X$-identification as shown in Figure 5(a). That is, initially, the test vector $v$ has only one capture transition. Now assume that one $X$-bit is identified at $a$ by $X$-identification as shown in

**Table I.** Types of $X$-bit-pair.

| | | | LCP $X$-filling | |
|---|---|---|---|---|
| | $\langle v$: PPI$\rangle$ | $\langle f(v)$: PPO$\rangle$ | Logic value for $X$ | Effort |
| Type-A | $X$ | $b2$ | $b2$ | Low |
| Type-B | $b1$ | $X$ | $b1$ | Medium |
| Type-C | $X$ | $X$ | 0–0<br>1–1 | High |

(b1, b2: 0 or 1).



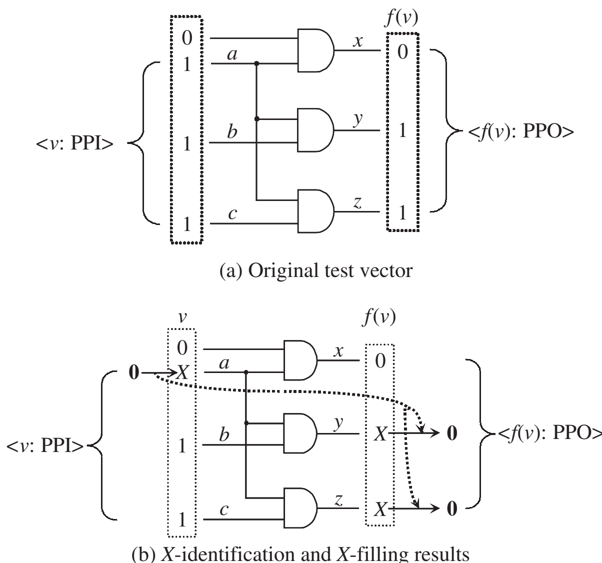

(a) $X$-filling for a Type-A $X$-bit-pair

(b) $X$-filling for a Type-B $X$-bit-pair

(c) $X$-filling for a Type-C $X$-bit-pair

**Fig. 4.** LCP $X$-filling.

Figure 5(b). In this case, 3-valued logic simulation will find three $X$-bit-pairs: $\langle a, x\rangle = \langle X, 0\rangle$, $\langle b, y\rangle = \langle 1, X\rangle$, and $\langle c, z\rangle = \langle 1, X\rangle$. Assume that $\langle a, x\rangle = \langle X, 0\rangle$ is processed first in $X$-filling. Based on the rules shown in Table I, the logic value for the $X$-bit on $a$ is determined to be 0. However, assigning 0 to $a$ automatically brings 0 to both $y$ and $z$, creating two new capture transitions. This is because there usually is dependency among different $X$-bit-pairs.




(a) Original test vector

(b) $X$-identification and $X$-filling results

**Fig. 5.** Dependency among $X$-bit-pairs.

In this case, 1 is a better choice for the $X$-bit on $a$ although it results in a capture transition.

The example in Figure 5 shows that, determining logic values for $X$-bits in an $X$-bit-pair locally by using the rules shown in Table I only for that particular $X$-bit-pair may not lead to a good result. Generally, it is preferable to determine logic values for $X$-bits in an $X$-bit-pair globally by taking dependency among different $X$-bit-pairs into consideration.

### 2.4. Motivation

As described in 2.2 and 2.3, previous capture power reduction methods based on test set modification have the following problems in $X$-identification and $X$-filling:

- Unconstrained $X$-identification may not lead to an effective $X$-bit distribution in which non-capture-transition-causing bits keep their values and capture-transition-causing bits are selectively turned into $X$-bits so that $\mathrm{MCT}(T)$ can be effectively reduced for a test set $T$.
- Dependency among different $X$-bit-pairs is ignored in $X$-filling. Determining logic values for the $X$-bits in an $X$-bit-pair without taking other $X$-bit-pairs into consideration adversely affects the effect of $X$-filling.

Therefore, there is a strong need for improving $X$-identification and $X$-filling, the key procedures in any capture power reduction method based on test set modification. In the following, we address the above problems with two unique techniques: constrained $X$-identification for finding an effective $X$-bit distribution and $X$-bit-pair-dependency-aware $X$-filling for effective capture transition reduction.

## 3. NEW METHOD FOR CAPTURE POWER REDUCTION

### 3.1. General Flow

The general flow of the new method for modifying a test set to reduce its capture power dissipation is shown in Figure 6. The procedure starts from an initial fully-specified test set $T_{\mathrm{int}}$. Note that during the generation of $T_{\mathrm{int}}$, $X$-filling may be needed for $X$-bits existing in intermediate test cubes. $X$-filling at this stage should be targeted for detecting more secondary faults in order to reduce the size of $T_{\mathrm{int}}$. After $T_{\mathrm{int}}$ is generated, a constrained $X$-identification procedure is conducted to find $X$-bits for $T_{\mathrm{int}}$ in an $X$-bit distribution that is effective for capture transition reduction. The result is an intermediate test set $T_x$. Then, $X$-filling for the $X$-bits in $T_x$ is conducted with an $X$-bit-pair-dependency-aware LCP $X$-filling procedure. $X$-filling at this stage is targeted for reducing the number of capture transitions. The resulting fully-specified test set $T_{\mathrm{fin}}$ is the final test set with lower capture power than $T_{\mathrm{int}}$.

The details of constrained $X$-identification and $X$-bit-pair-dependency-aware LCP $X$-filling are described in 3.2, and 3.3, respectively.

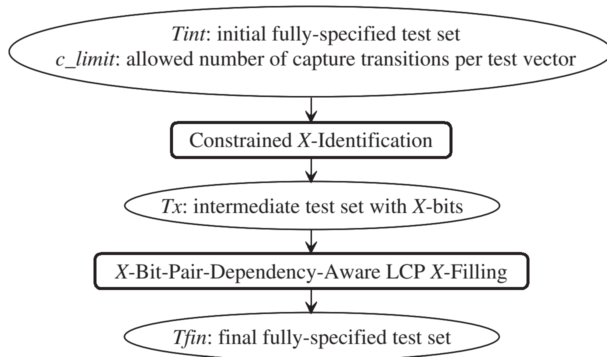


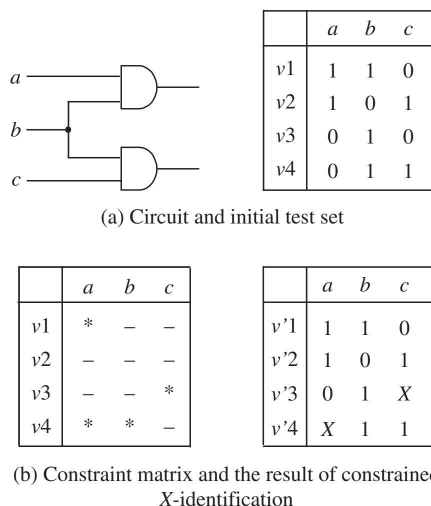

**Fig. 6.** General procedure for capture power reduction.

## 3.2. Constrained X-Identification

### *3.2.1. Basic Procedure and Requirements*

The basic procedure of constrained $X$-identification is the same as that in Ref. [29]. Similar to unconstrained $X$-identification,[28] it operates on all test vectors in a test set simultaneously using justification and implication techniques to maximize the number of identified $X$-bits. The unique feature of our procedure is that it limits the $X$-bit search at a set of pre-specified bits, called changeable bits, while keeping the logic values for the rest of bits, called fixed bits, unchanged. The changeable/fixed bit specification is provided by a constraint matrix.

Figure 7 shows an example of constrained $X$-identification. Figure 7(a) shows a circuit and its initial fully-specified test set. Figure 7(b) shows a constraint matrix, where "*" and "–" indicate changeable bits and fixed bits, respectively. The result of constrained $X$-identification based on the constraint matrix is also shown in Figure 7(b). This example also shows that, while no $X$-bit will be identified at any fixed bit, usually only part of changeable bits can be turned into $X$-bits.




| | a | b | c |
|---|---|---|---|
| v1 | 1 | 1 | 0 |
| v2 | 1 | 0 | 1 |
| v3 | 0 | 1 | 0 |
| v4 | 0 | 1 | 1 |

(a) Circuit and initial test set

| | a | b | c |
|---|---|---|---|
| v1 | * | – | – |
| v2 | – | – | – |
| v3 | – | – | * |
| v4 | * | * | – |

| | a | b | c |
|---|---|---|---|
| v'1 | 1 | 1 | 0 |
| v'2 | 1 | 0 | 1 |
| v'3 | 0 | 1 | X |
| v'4 | X | 1 | 1 |

(b) Constraint matrix and the result of constrained $X$-identification

**Fig. 7.** Example of constrained $X$-identification.

### *3.2.2. Constraint Matrix Generation*

The key task in constrained $X$-identification is to generate an effective constraint matrix. Obviously, non-capture-transition-causing bits should be specified as fixed bits. On the other hand, capture-transition-causing bits should be selectively specified as changeable bits. The criterion of selecting changeable bits for a test set $T$ is to make sure that the resulting $X$-distribution can effectively reduce MCT($T$) as discussed in 2.2. The procedure of generating a constraint matrix for a test set is proposed as follows:

Step 1: Identify the capture-transition-causing bits and non-capture-transition-causing bits for each test vector by conducting 2-valued logic simulation.

Step 2: Specify the non-capture-transition-causing bits for all test vectors as fixed bits.

Step 3: Select a set of capture-transition-causing bits for a test vector and specify them as changeable bits. Repeat this step for all test vectors.

Step 4: Specify the remaining capture-transition-causing bits for all test vectors as fixed bits.

The method for selecting changeable bits for a test vector at Step 3 is described in the following. We first discuss the basic idea for changeable bit selection and then provide an algorithm for selecting changeable bits.

The concept for selecting changeable bits from capture-transition-causing bits in a test vector is illustrated in Figure 8. The basic idea is that, only when $n$, the number of capture-transition-causing bits for a test vector, is greater than *c_limit* (the number of allowed capture transitions per test vector as shown in Figure 6, then we select $d$ changeable bits from the $n$ capture-transition-causing bits, where $d = n - c_limit$. In the example shown in Figure 8, the number of capture-transition-causing bits of $vi$ is greater than *c_limit* by a difference of $d$. In this case, we select $d$ bits from its capture-transition-causing bits as changeable bits. As for $vj$, no changeable bits are selected from its capture-transition-causing bits since the number of capture-transition-causing bits of $vj$ is smaller than *c_limit*.

Introducing the concept of *c_limit* allows us to avoid over-conducting $X$-identification or conducting

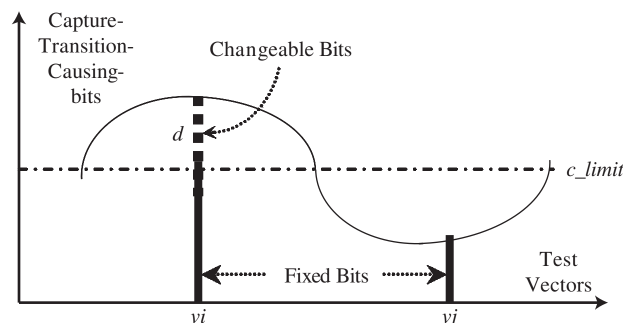



**Fig. 8.** Concept of constraint matrix generation.

unnecessary $X$-identification for a test vector. This way, more $X$-identification capability is reserved for other test vectors. As a result, a globally better effect can be achieved in capture transition reduction for the whole test set. Obviously, the final effect of capture transition reduction is dependent on the value of *c_limit*. If it is too large, fewer changeable bits will be specified in a constraint matrix, resulting in fewer $X$-bits, and thus less capture power reduction. If it is too small, a large number of changeable bits will be specified in a constraint matrix, making the constrained $X$-identification procedure[29] unable to successfully find $X$-bits on all changeable bits, again leading to unsatisfactory capture power reduction effect. Heuristically, *c_limit* can be set as $\alpha \times \text{MCT}(T)$ for a given test set $T$, where $0 < \alpha < 1$.

Now that $d$, the number of changeable bits, is determined by using *c_limit*, next we show how to select $d$ changeable bits from $n$ capture-transition-causing bits, where $n >$ *c_limit*. The basic idea is to select a capture-transition-causing bit on which the possibly identified $X$-bit is easy to handle later in $X$-filling. An example is shown in Figure 9, where $v$ is the initial test vector, and its response $f(v)$ is obtained by 2-valued logic simulation. Obviously, $v$ has two capture-transition-causing bits: $a$ and $b$. Suppose that one changeable bit needs to be selected from them. In order to make the selection, $v$ is first converted to $v'$ by turning all capture-transition-causing bits into $X$s. Then, 3-valued simulation with {0, 1, $X$} is conducted for $v'$ to obtain $f(v')$. Comparing $v'$ and $f(v')$, it is clear that there are two $X$-bit-pairs corresponding to $a$ and $b$, respectively. The $X$-bit-pair with regard to $a$ is of Type-C, while the $X$-bit-pair with regard to $b$ is of Type-A, as defined in Table I. As shown in Figure 4, $X$-filling for a Type-A $X$-bit-pair will always be successful with a simple assignment operation. That is, it is easier to handle a Type-A $X$-bit-pair in $X$-filling than a Type-C $X$-bit-pair. Therefore, it is preferable to select $b$ as a changeable bit

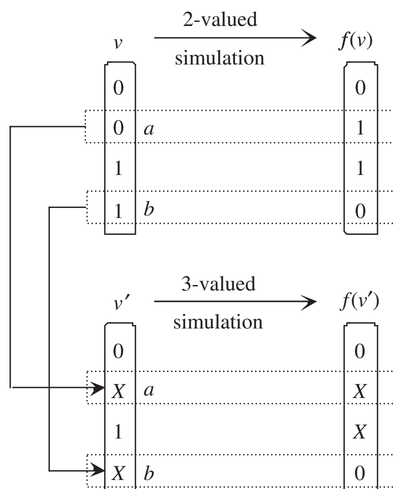



**Fig. 9.** Changeable bit selection.

in order to make constrained $X$-identification to try to find an $X$-bit on $b$.

Based on the basic idea illustrated in Figure 9, the algorithm for selecting $d$ changeable bits for a test vector $v$ is proposed as follows:

Step 1: Identify all capture-transition-causing bits for $v$ by 2-valued logic simulation.

Step 2: Convert $v$ to $v'$ by replacing all capture-transition-causing bits with $X$s, and identify all $X$-bit-pairs by conducting 3-valued logic simulation with {0, 1, $X$}.

Step 3: First, select all capture-transition-causing bits corresponding to Type-A $X$-bit-pairs as changeable bits. If more changeable bits need to be selected, then select capture-transition-causing bits corresponding to Type-C $X$-bit-pairs as changeable bits.

In Step 3, if there are multiple Type-C $X$-bit-pairs from which changeable bits need to be selected, we select those with larger reach values. Here, the reach value of a Type-C $X$-bit-pair $\langle a, b\rangle$, where $a$ and $b$ are bits in $\langle v'$: PPI$\rangle$ and $\langle f(v')$: PPO$\rangle$, respectively, is defined as the number of $X$s in $v'$ that is structurally reachable from $b$. Obviously, the larger the reach value, the more likely a successful justification of a required logic value on $b$.

### 3.3. X-Bit-Pair-Dependency-Aware X-Filling

As shown in Figure 6, after constrained $X$-identification is conducted and an intermediate test set with $X$-bits is obtained, $X$-filling is then conducted, one test vector at a time. The purpose is to fill $X$-bits in a test vector with proper logic values so that the number of capture transitions is reduced as much as possible.

In the following, we propose an $X$-bit-pair-dependency-aware LCP (low capture power) $X$-filling method, whose general procedure is shown in Figure 10. It is similar to the LCP $X$-filling method described in Ref. [27] in that it also uses techniques of assignment, justification, and their combinations for bringing necessary logic values to the $X$-bits in an $X$-bit-pair for capture transition reduction, as illustrated in Figure 4. The difference is in how necessary logic values are determined for the $X$-bits in an $X$-bit-pair.

The $X$-bit-pair-dependency-aware LCP $X$-filling method collects dependency information among different $X$-bit-pairs in a test vector through partial-symbolic $X$-simulation in the procedure A of Figure 10. The information is used in determining proper logic values for the $X$-bits in an $X$-bit-pair in the procedure B of Figure 10. Taking $X$-bit-pair dependency into consideration in logic value determination for $X$-bits makes it possible for the new $X$-filling method to achieve a globally better result in capture transition reduction.

The details of dependency information collection and logic value determination are described in 3.3.1 and 3.3.2,

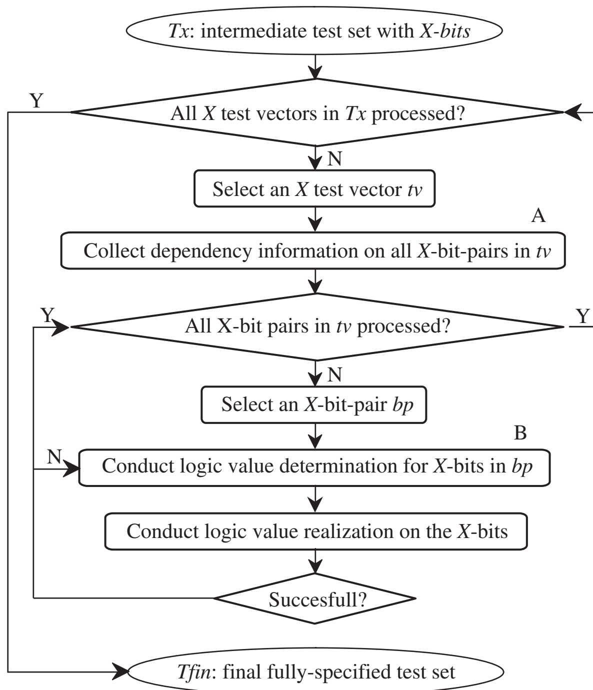



**Fig. 10.** New LCP $X$-filling procedure.

respectively. Detailed information on how to realize the determined logic values by assignment and justification in $X$-filling can be found in Ref. [27].

### *3.3.1. Dependency Information Collection*

$X$-bit-pair dependency information is collected through partial-symbolic $X$-simulation, whose basic concept was originally proposed for fault diagnosis.[30] An extension of this analysis technique is made in this paper since it helps in exploring the dependency among all $X$-bit-pairs in a test vector.

Suppose that the test vector under $X$-filling is $v'$, which has $m$ $X$-bits identified from a fully-specified initial test vector $v$. $v'$ is called an $X$ test vector. Partial-symbolic $X$-simulation is conducted for $v'$ as follows: First, the $m$ identical $X$-bits in $v'$ are replaced with $m$ different $X$-symbols, $X1, X2, \ldots, Xm$. The resulting test vector, $v''$, is called a symbolic $X$ test vector. Then partial-symbolic propagation is conducted for $v''$ by repeatedly applying the following two rules:

> Rule 1: If the result of a logic function is the inversion of an $X$ symbol $Xi$ ($\overline{X}i$), the result is represented with $\overline{X}i$ ($Xi$).
>
> Rule 2: If the result of a logic function is neither a logic value (0 or 1) nor the inversion of an $X$ symbol $Xi$ ($\overline{X}i$), the result is represented with a new $X$ symbol.

These rules are devised to preserve only the logic inversion function but ignore all other logic functions in partial-symbolic propagation. The purpose is to reduce memory usage while retaining useful structural and logic information as much as possible. Note that partial-symbolic $X$-simulation is conducted on the combinational portion of a full-scan sequential circuit. As a result, only primary combinational logic functions need to be considered in Rule 1 and Rule 2.

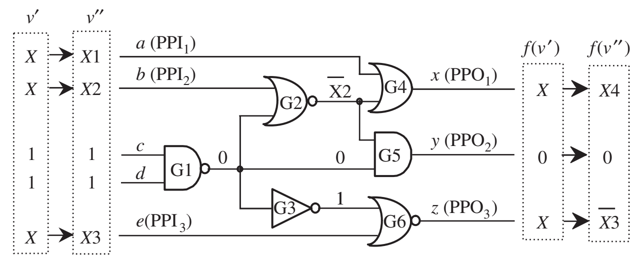



**Fig. 11.** Partial-symbolic $X$-simulation.

Figure 11 shows an example of partial-symbolic $X$-simulation. The $X$ test vector $v'$ has three $X$-bits, and its corresponding symbolic $X$ test vector $v''$ is obtained by replacing the three $X$-bits with three different $X$-symbols, $X1$, $X2$, and $X3$. Partial-symbolic propagation is conducted by repeating Rule 1 and Rule 2. For example, since the output of G2 is the inversion of $X2$, it is represented with $\overline{X}2$. Since the output of G4 is OR($X1$, $\overline{X}2$), it is represented with a new $X$-symbol, $X4$.

Generally, partial-symbolic $X$-simulation can collect more dependency information on $X$-bit-pairs than 3-valued logic simulation, as demonstrated by the example of Figure 11. Here, the result of conducting 3-valued logic simulation on $v'$ only reveals a simple $X$-bit-pair $\langle X, X\rangle$ for $\langle e, z\rangle$. The result of conducting partial-symbolic $X$-simulation on $v''$, on the other hand, reveals a symbolic $X$-bit-pair $\langle X3, \overline{X}3\rangle$ for $\langle e, z\rangle$, which shows that it is impossible to bring the same logic value to both $e$ and $z$. Obviously, such detailed dependency information can not be obtained from 3-valued logic simulation.

### *3.3.2. Logic Value Determination*

Suppose that $X$-filling needs to be conducted for an $X$ test vector $v'$, and it is necessary to determine logic values for the $X$-bits in $v'$ and $f(v')$, where $f(v')$ is the response to $v'$. The algorithm for logic value determination is proposed as follows:

> Step 1: Conduct partial-symbolic $X$-simulation on $v''$, which is the symbolic $X$ test vector corresponding to $v'$. All symbolic $X$-bit pairs are identified by comparing $\langle v''$: PPI$\rangle$ $\langle f(v'')$: PPO$\rangle$, where $f(v'')$ is the response to $v''$.
>
> Step 2: For each symbolic $X$-bit pair in the form of $\langle Xi, b2\rangle$, $\langle b1, Xi\rangle$, or $\langle b1, \overline{X}i\rangle$, calculate the logic preference of $Xi$ adaptively with the rules shown in Table II. Here, $b1$ and $b2$ are arbitrary logic values. Note that the initial 0-preference and 1-preference for $Xi$ are set to 0.

**Table II.** Rules for logic preference calculation.

| $\langle v''$: PPI$\rangle$ | $\langle f(v'')$: PPO$\rangle$ | $Xi$ 0-preference | $Xi$ 1-preference |
|---|---|---|---|
| $Xi$ | $b2$ | $+1$ if $b2=0$ | $+1$ if $b2=1$ |
| $b1$ | $Xi$ | $+1$ if $b1=0$ | $+1$ if $b1=1$ |
| $b1$ | $\overline{X}i$ | $+1$ if $b1=1$ | $+1$ if $b1=0$ |

(b1, b2: 0 or 1).

Step 3a: When the logic value for the $X$-bit $Xi$ in a symbolic $X$-bit pair in the form of $\langle Xi, b2\rangle$, $\langle b1, Xi\rangle$, or $\langle b1, \overline{X}i\rangle$ needs to be determined, select 0 (1) if its 0-preference (1-preference) is greater than its 1-preference (0-preference) for $Xi$.

Step 3b: When the logic values for the $X$-bits $a$ and $b$ in a symbolic $X$-bit pair in the form of $\langle a, b\rangle = \langle Xi, Xi\rangle$ need to be determined, select 0 (1) for both $a$ and $b$ first. If this selection cannot be realized in $X$-filling, then select 1 (0) for both $a$ and $b$.

Step 3c: When the logic values for the $X$-bits $a$ and $b$ in a symbolic $X$-bit pair in the form of $\langle a, b\rangle = \langle Xi, \overline{X}i\rangle$ need to be determined, select 0 (1) for $a$ and 1 (0) for $b$ first. If this selection cannot be realized in $X$-filling, then select 1 (0) for $a$ and 0 (1) for $b$.

Step 3d: When the logic values for the $X$-bits $a$ and $b$ in a symbolic $X$-bit pair in the form of $\langle a, b\rangle = \langle Xi, Xj\rangle$ or $\langle Xi, \overline{X}j\rangle$ need to be determined, select the same logic value for both $a$ and $b$ first. If this selection cannot be realized in $X$-filling, then select opposite logic values for $a$ and $b$.

Obviously, Step 1 is for collecting more dependency information on $X$-bit-pairs through partial-symbolic $X$-simulation. Step 2 is for taking $X$-bit-pair-dependency into consideration in order to determine the logic value for an $X$-bit more effectively. Step 3 is for determining logic values in all possible cases. As a result, more effective logic values can be determined for the $X$-bits of an $X$-bit-pair in $X$-filling.

An example of logic value determination is shown in Figure 12, which shows the result of partial-symbolic $X$-simulation for the same circuit and the same initial test vector as shown in Figure 5. There are three symbolic $X$-bit-pairs: $\langle a, x\rangle = \langle X1, 0\rangle$, $\langle b, y\rangle = \langle 1, X1\rangle$, and $\langle c, z\rangle = \langle 1, X1\rangle$. According to the rules shown in Table II, the 0-preference for $X1$ is 1 and the 1-preference for $X1$ is 2. Therefore, when it is necessary to determine a logic value for the $X$-bit $X1$ in the symbolic $X$-bit-pair $\langle a, x\rangle = \langle X1, 0\rangle$, logic value 1 will be selected. This will result in a capture transition for $\langle a, x\rangle$ but will avoid two capture transitions for $\langle b, y\rangle$ and $\langle c, z\rangle$. Obviously, this is a better selection than the one shown in Figure 5. Therefore, compared with the previous logic value determination method summarized in Table I, the new logic value determination method generally leads to more effective $X$-filling for capture transition reduction.

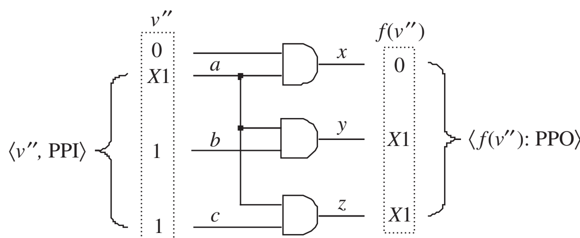



**Fig. 12.** Example of logic value determination.

## 4. EXPERIMENTAL RESULTS

The proposed method for capture power reduction was implemented in $C$, and experiments were conducted on ISCAS' 89 circuits to verify its effectiveness. The machine used was a PC with a 1.0 GHz Pentium III CPU and 512 MB memory.

Table III shows the basic information on the circuits used in experiments. The numbers of PIs, POs, and F/Fs for each circuit are shown under "# of PIs", "# of POs", and "# of F/Fs", respectively. An internally-developed ATPG tool was used to generate stuck-at test sets, and the number of test vectors and the fault coverage for each circuit are shown under "# of Vectors" and "Fault Coverage", respectively. In addition, the original maximum number of capture transitions for each circuit is shown under "Original Max. Trans".

Table IV shows the experimental results for the previous capture power reduction method based on an unconstrained $X$-identification procedure[28] and the old LCP $X$-filling procedure.[27] The percentage of $X$-bits found by the unconstrained $X$-identification procedure and CPU time for each circuit are shown under "$X$" and "CPU" of "Unconstrained $X$-Identification", respectively. It can be seen that on average, 65.3% of the bits in a fully-specified test sets can be turned into $X$-bits without any fault coverage loss. $X$-filling for the identified $X$-bits was conduced by the old LCP procedure[27] that ignores $X$-bit-pair-dependency in logic value determination for $X$-bits. The maximum number of capture transitions, the reduction ratio with regard to the original maximum number of capture transitions, and CPU time for each circuit are shown

**Table III.** Basic information.

| Circuit | # of PIs | # of POs | # of F/Fs | # of Vectors | Fault Coverage | Original Max. Trans. |
|---|---|---|---|---|---|---|
| s1238 | 14 | 14 | 18 | 125 | 94.9 | 18 |
| s1423 | 17 | 5 | 74 | 24 | 99.1 | 49 |
| s5378 | 35 | 49 | 179 | 100 | 99.1 | 102 |
| s13207 | 31 | 121 | 669 | 235 | 98.5 | 380 |
| s15850 | 14 | 87 | 597 | 97 | 96.7 | 282 |
| s35932 | 35 | 320 | 1728 | 12 | 89.8 | 1548 |
| s38417 | 28 | 106 | 1636 | 87 | 99.5 | 590 |
| s38584 | 12 | 278 | 1452 | 114 | 95.9 | 925 |

**Table IV.** Experimental result for the previous method.

| | Unconstrained $X$-Identification | | Random $X$-Filling | Old LCP $X$-Filling | | |
|---|---|---|---|---|---|---|
| Circuit | $X$ (%) | CPU (Sec.) | Max. Trans. | Max. Trans. | Red. Ratio (%) | CPU (Sec.) |
| s1238 | 55.0 | 0.1 | 14 | 9 | 50.0 | 0.0 |
| s1423 | 41.1 | 0.1 | 40 | 34 | 30.6 | 0.0 |
| s5378 | 71.0 | 1.3 | 109 | 91 | 10.8 | 0.2 |
| s13207 | 91.6 | 8.6 | 305 | 244 | 35.8 | 5.9 |
| s15850 | 76.1 | 5.3 | 262 | 173 | 38.7 | 1.8 |
| s35932 | 34.4 | 4.0 | 1532 | 1517 | 2.0 | 0.3 |
| s38417 | 73.4 | 13.8 | 600 | 323 | 45.3 | 24.7 |
| s38584 | 79.7 | 17.5 | 802 | 437 | 52.8 | 22.3 |
| Ave. | 65.3 | — | — | — | 33.2 | — |

under "Max. Trans.", "Red. Ratio", and "CPU" of "Old LCP $X$-Filling", respectively. For the comparison purpose, the result of randomly filling the $X$-bits for each circuit is also shown in Table IV under "Max. Trans." of "Random $X$-Filling". Obviously, even the old LCP $X$-filling procedure can achieve a better capture power reduction effect than random $X$-filling.

Table V shows the experimental results for the new capture power reduction method proposed in this paper. $X$-identification was conducted by a constrained $X$-identification procedure[29] based on the constraint matrix generation technique proposed in 3.2. $X$-filling was conducted by the new $X$-bit-pair-dependency-aware LCP $X$-filling procedure proposed in 3.3. The meaning of each column is the same as that of the corresponding column in Table IV.

From Table IV, it can be seen that the previous capture power reduction method achieved 33.2% reduction for the maximum number of capture transitions on average. From Table V, it can be seen that the proposed capture power reduction method achieved 43.3% reduction for the maximum number of capture transitions on average. This is 30.4% improvement in the reduction ratio for the maximum number of capture transitions. Since the number of capture transitions has strong correlation with the number of node transitions in a circuit,[13] and the number of node transitions directly reflects capture power dissipation, the experimental results show that the new techniques proposed in this paper are indeed effective in reducing capture power dissipation.

**Table V.** Experimental result for the proposed method.

| | Constrained $X$-Identification | | Random $X$-Filling | New LCP $X$-Filling | | |
|---|---|---|---|---|---|---|
| Circuit | $X$ (%) | CPU (Sec.) | Max. Trans. | Max. Trans. | Red. Ratio (%) | CPU (Sec.) |
| s1238 | 32.6 | 0.1 | 14 | 5 | 72.2 | 0.0 |
| s1423 | 10.8 | 0.1 | 43 | 34 | 30.6 | 0.0 |
| s5378 | 25.2 | 1.2 | 108 | 70 | 31.4 | 0.5 |
| s13207 | 33.1 | 10.3 | 333 | 198 | 47.9 | 5.5 |
| s15850 | 20.2 | 5.2 | 252 | 132 | 53.2 | 1.8 |
| s35932 | 12.8 | 3.7 | 1533 | 1482 | 4.3 | 0.6 |
| s38417 | 20.3 | 15.3 | 592 | 295 | 50.0 | 5.6 |
| s38584 | 20.6 | 14.9 | 785 | 396 | 57.2 | 7.0 |
| Ave. | 21.9 | — | — | — | 43.3 | — |

Note that, in order to generate a constraint matrix for constrained $X$-identification, it is necessary to set *c_limit* (the number of allowed capture transitions per test vector) as described in 3.2.2. In the experiments, we set *c_limit* as $10\% \times \mathrm{MCT}(T)$, where $T$ is an initial fully-specified test set.

Also note that, unconstrained $X$-identification resulted in 65.3% of $X$-bits on average for a fully-specified test as shown in Table IV, while constrained $X$-identification resulted in 21.9% of $X$-bits on average for a fully-specified test as shown in Table V. Obviously, even though constrained $X$-identification resulted in a smaller number of $X$-bits, nonetheless it achieved better results in capture power reduction. This demonstrates that $X$-bit distribution, not only the number of $X$-bits, indeed plays an important role in improving the effect of capture power reduction.

## 5. CONCLUSION

This paper proposed a new capture power reduction method based on test set modification, featuring two unique techniques. The first technique is for algorithmically generating a constraint matrix for constrained $X$-identification. This makes it possible to obtain a more effective $X$-bit distribution for capture power reduction. The second technique is for determining a more effective logic value for an $X$-bit in $X$-filling by taking dependency among different $X$-bit-pairs into consideration. This makes it possible to achieve a globally better result in capture power reduction. Since the proposed method is for post-processing a test set, it can be readily applied in any test generation flow for capture power reduction without incurring impact on area, timing, test set size, and fault coverage. The result is a lowered risk of test-related yield loss due to IR drop caused by excessive capture power dissipation.

Areas for future work include (1) evaluating capture power reduction effects by using a power analysis tool, (2) determining *c_limit* algorithmically, (3) extending the basic idea of this paper to dynamic compaction in ATPG, and (4) combining the test set modification approach with the DFT (design for testability) approach in order to further reduce test power in general.

**Acknowledgments:** This work was partly supported by Grant-in-Aid for Scientific Research of Japan Society for the Promotion of Science under the Grant No. 17500039. The authors are also grateful to all reviewers for their helpful comments.

## References


1. M. Abramovici, M. Breuer, and A. D. Friedman, Digital systems testing and testable design. *IEEE Press* (**1994**).
2. L.-T. Wang, X. Wen, H. Furukawa, F. Hsu, S. Lin, S. Tsai, K. S. Abdel-Hafez, and S. Wu, VirtualScan: a new compressed scan technology for test cost reduction. *Proceedings of the IEEE International Test Conference* (**2004**), pp. 916–925.
3. Y. Zorian, A distributed BIST control scheme for complex VLSI devices. *Proceedings of the IEEE VLSI Test Symposium* (**1993**), pp. 4–9.
4. P. Girad, Survey of low-power testing of VLSI circuits. *IEEE Design and Test of Computers* (**2002**), Vol. 19, pp. 82–92.
5. N. Nicolici and B. M. Al-Hashimi, *Power-constrained testing of VLSI circuits*. Kluwer Academic Publishers (**2003**).
6. J. Altet and A. Rubio, *Thermal testing of ICs*. Kluwer Academic Publishers (**2002**).
7. J. Saxena, K. M. Butler, V. B. Jayaram, and S. Kundu, A case study of IR-drop in structured at-speed testing. *Proceedings of the IEEE International Test Conference* (**2003**), pp. 1098–1104.
8. T. Yoshida and M. Watari, A new approach for low power scan testing. *Proceedings of the IEEE International Test Conference* (**2003**), pp. 480–487.
9. R. Chou, K. K. Saluja, and V. Agrawal, Scheduling tests for VLSI systems under power constraints. *IEEE transactions on VLSI systems* (**1997**), Vol. 5, pp. 175–185.
10. S. Wang and S. Gupta, ATPG for heat dissipation minimization during test application. *IEEE Transactions on computers* (**1998**), Vol. 47, pp. 256–262.
11. F. Corno, P. Prinetto, M. Redaudengo, and M. Reorda, A Test Pattern Generation Methodology for Low Power Consumption. *Proceedings of the IEEE VLSI Test Symposium* (**1998**), pp. 453–459.
12. K. M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Levis, Minimizing power consumption in scan testing: pattern generation and DFT techniques. *Proceedings of the IEEE International Test Conference* (**2004**), pp. 355–364.
13. R. Sankaralingam, R. Oruganti, and N. Touba, Static Compaction Techniques to Control Scan Vector Power Dissipation. *Proceedings of the IEEE VLSI Test Symposium* (**2000**), pp. 35–40.
14. S. Kajihara, K. Ishida, and K. Miyase, Test vector modification for power reduction during scan testing. *Proceedings of the IEEE VLSI Test Symposium* (**2002**), pp. 160–165.
15. V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Transactions on Computer-Aided Design* (**1998**), Vol. 17, pp. 1325–1333.
16. A. Chandra and K. Chakrabarty, Combining low power scan testing and test data compression for system-on-a-chip. *Proceedings of the ACM/IEEE Design Automation Conference* (**2001**), pp. 166–169.
17. A. Chandra and K. Chakrabarty, Reduction of SoC test data volume, scan power, and testing time using alternating run-length codes. *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design* (**2002**), pp. 673–678.
18. A. Hertwig and H. Wunderlich, Low power serial built-in self-test. *Proceedings of the IEEE European Test Workshop* (**1998**), pp. 49–53.
19. R. Sankaralingam, R. Oruganti, and N. Touba, Reducing power dissipation during test using scan chain disable. *Proceedings of the IEEE VLSI Test Symposium* (**2001**), pp. 319–324.
20. Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, Power driven chaining of flip-flops in scan architectures. *Proceedings of the IEEE international test conference* (**2002**), pp. 796–803.
21. J. Saxena, K. Butler, and L. Whetsel, A Scheme to reduce power consumption during scan testing. *Proceedings of the IEEE International Test Conference* (**2001**), pp. 670–677.
22. O. Sinanoglu and A. Orailoglu, Scan power minimization through stimulus and response transformations. *Proceedings of the IEEE International Design, Automation and Test in Europe Conference* (**2004**), pp. 404–409.
23. W. Li, S. M. Reddy, and I. Pomeranz, On Reducing Peak Current and Power during Test. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI* (**2005**), pp. 156–161.
24. W. Li, S. M. Reddy, and I. Pomeranz, On reducing power dissipation during capture cycle of broadside delay testing. *Proceedings of the IEEE Latin America Test Workshop* (**2005**), pp. 71–76.
25. X. Wen, Y. Yamashita, S. Morishima, S. Kajiihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, Low-capture-power test generation for scan-based at-speed testing. *Proceedings of the IEEE International Test Conference* (**2005**), pp. 39–2.
26. R. Sankaralingam and N. A. Touba, Controlling peak power during scan testing. *Proceedings of the IEEE VLSI Test Symposium* (**2002**), pp. 153–159.
27. X. Wen, H. Yamashita, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, On low-capture-power test generation for scan testing. *Proceedings of the IEEE VLSI Test Symposium* (**2005**), pp. 265–270.
28. K. Miyase and S. Kajihara, XID: don't care identification of test patterns for combinational circuits. *IEEE Transactions on Computer-Aided Design* (**2004**), Vol. 23, pp. 321–326.
29. K. Miyase, S. Kajihara, I. Pomeranz, and S. M. Reddy, don't care identification on specific bits of test patterns. *Proceedings of the ACM/IEEE International Conference on Computer Design* (**2002**), pp. 194–199.
30. X. Wen, T. Miyoshi, S. Kajiihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, On per-test fault diagnosis using the $X$-fault model. *Proceedings of the ACM/IEEE International Conference on Computer Design* (**2004**), pp. 633–640.


### Xiaoqing Wen

Xiaoqing Wen *received his B.E. degree in Computer Engineering from Tsinghua University, China, in 1986, his M.E. degree in Information Engineering from Hiroshima University, Japan, in 1990, and his Ph.D degree in Applied Physics from Osaka University, Japan, in 1993. From 1993 to 1997, he was a Lecturer at Akita University, Japan. He was a Visiting Researcher at University of Wisconsin-Madison from October 1995 to March 1996. He joined SynTest Technologies, Inc. in 1998 and served as its CTO until 2003. From 2004, he has been an Associate Professor at the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan. His research interests include VLSI test, diagnosis, and testable design. He is a member of the IEEE, the IEICE, and the REAJ.*

### Tatsuya Suzuki

Tatsuya Suzuki *received his B.E. degree in Computer Science and Electronics from Kyushu Institute of Technology, Japan, in 2005. He is currently working towards his M.E. degree in the Graduate School of Computer and Systems Engineering, Kyushu Institute of Technology, Japan. His research interest is low power testing of VLSI circuits.*

**Seiji Kajihara**

Seiji Kajihara *received his B.S. and M.S. degrees in Information Engineering from Hiroshima University, Japan, and his Ph.D degree in Applied Physics from Osaka University, Japan, in 1987, 1989, and 1992, respectively. From 1992 to 1995, he worked with the Department of Applied Physics, Osaka University, Japan, as an Assistant Professor. In 1996, he joined the Department of Computer Science and Electronics of Kyushu Institute of Technology, Japan, where he is currently a Professor. His research interest includes test generation, delay testing, and design for testability. He received the Young Engineer Award from the IEICE in 1997 and the Yamashita SIG Research Award from the IPSJ in 2002. He is a member of the IEEE, the IEICE, and the IPSJ. He also serves on the editorial board of the Journal of Electronic Testing: Theory and Applications.*

**Kohei Miyase**

Kohei Miyase *received his B.E., M.E., and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology, Japan, in 2000, 2002, and 2005, respectively. From 2005, he has been a researcher at Innovation Plaza Fukuoka, Japan Science and Technology Agency, Japan. His research interests include test compaction, test compression, design for testability, low power test, and fault diagnosis. He received the Excellent Student Award of the IEEE Fukuoka Section in 2005. He is a member of the IEEE, the IEICE, and the IPSJ.*

**Yoshihiro Minamoto**

Yoshihiro Minamoto *received his B.E. degree in Computer Science and Communications Engineering from Kyushu University, Japan, in 1983. From 1983 to 2005, he worked at Oki Electric Industry Co., Japan. From 2005, he has been a researcher at Innovation Plaza Fukuoka, Japan Science and Technology Agency, Japan. His research interests include test generation and design for testability. He is a member of the IEEE, the IEICE, and the IPSJ.*

**Laung-Terng Wang**

Laung-Terng Wang *received his B.S. and M.E. degrees in Electrical Engineering from National Taiwan University, Taiwan, in 1975 and 1977, respectively, and his M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1982 and 1987, respectively. He was a Lecturer at the Department of Electrical Engineering and Computer Science of Stanford University from 1989 to 1991. In January 1990, he founded SynTest Technologies, Inc. headquartered in Sunnyvale, California. Since then, he has led the company to grow to more than 60 full-time employees and 200 customers worldwide. Prior to founding SynTest, he worked at several technology companies, including Intel and Daisy Systems. He has published more than 30 technical papers and filed more than 20 US patents in the areas of test compression, logic built-in self-test, design for testability, and design for debug/diagnosis. He is a Senior Member of the IEEE.*

**Kewal K. Saluja**

Kewal K. Saluja *obtained his B.E. degree in Electrical Engineering from the University of Roorkee, India, his M.S. and Ph.D degrees in Electrical and Computer Engineering from the University of Iowa in 1967, 1972, and 1973, respectively. He is currently with the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison as a Professor, where he teaches courses in logic design, computer architecture, microprocessor based systems, VLSI design and testing, and fault-tolerant computing. Prior to this, he was at the University of Newcastle, Australia. He has held visiting and consulting positions at various national and international institutions including University of Southern California, Hiroshima University, Nara Institute of Science and Technology, and the University of Roorkee. He has also served as a consultant to the United Nations Development Program. He served as an Editor of the IEEE Transactions on Computers (1997–2001), and he is an Associate Editor for the letters section of the Journal of Electronic Testing: Theory and Applications (JETTA) published by Kluwer. He is a Fellow of the IEEE and a Fellow of the JSPS.*