Power Supply Noise Reduction for At-Speed Scan Testing in Linear-Decompression Environment

Meng-Fan Wu, *Student Member, IEEE*, Jiun-Lang Huang, *Member, IEEE*,
Xiaoqing Wen, *Senior Member, IEEE*, and Kohei Miyase, *Member, IEEE*

***Abstract*—Yield loss caused by excessive power supply noise has become a serious problem in at-speed scan testing. Although X-filling techniques are available to reduce the launch cycle switching activity, their performance may not be satisfactory in the linear-decompressor-based test compression environment. This paper solves this problem by proposing a novel integrated automatic test pattern generation scheme that efficiently and effectively performs compressible low-capture-power X-filling. Related theoretical principles are established, based on which the problem size is substantially reduced. The proposed scheme is validated by benchmark circuits, as well as an industry design in the embedded deterministic test environment.**

***Index Terms*—At-speed scan testing, embedded deterministic test (EDT), linear decompressor, power supply noise, test data compression, X-filling.**

I. Introduction

WHILE THE advanced IC fabrication technology empowers the designers to realize more versatile systems on a chip, it also poses new test challenges, for example, timing-related defects and growing test data volume.

A. Power Supply Noise in At-Speed Scan Testing

Modern circuits are more prone to timing-related errors because of the growing circuit complexity, escalating clock speed, and the lowered power supply voltage. As a result, delay testing becomes mandatory to ensure high test quality [2].

Delay testing, in general, adopts the two-pattern test approach. The first pattern sets the circuit state, and the second pattern activates the desired transition at the fault site. The fault is detected if the transition does not propagate to the target flip-flop(s) within the functional clock period. Fig. 1 shows the timing diagram of the ***launch-on-capture (LOC)*** at-speed scan testing scheme. The rising edges of the two capture cycles C_1 and C_2 correspond to the functional clock cycle, called the ***launch cycle*** hereinafter. If the transition launched at C_1 does not propagate to the target flip-flop(s) before C_2, the chip under test is classified as faulty.

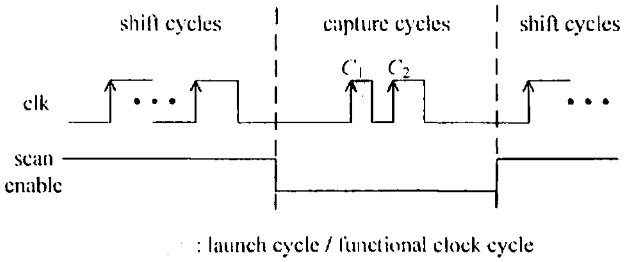


Fig. 1. LOC scheme.

The LOC scheme suffers yield loss caused by the power supply noise in the launch cycle. Conventional delay fault automatic test pattern generations (ATPGs) neglect the impact of launch-induced switching activity. The generated patterns may cause excessive switching activity in the launch cycle, which leads to abnormally high power network IR drop and results in extra gate propagation delays [3]. The extra delays may cause a timing-defect-free circuit under test to fail the delay fault test; this problem is referred to as the power-supply-noise-induced yield loss. In [4] and [5], it is reported that, in a 130-nm application-specified integrated circuit design running at 150-MHz clock frequency, some circuits pass the transition fault test only if the supply voltage is above 1.55 V; otherwise, they fail.

Previous works that aim at reducing the launch cycle power supply noise can be categorized into the architecture- and pattern-based classes. The partial capture scheme in [6] is architecture based; the noise-aware ATPG techniques [7]–[13] and the post-ATPG X-filling techniques [14]–[17] are pattern based. Pattern-based techniques are more compatible with any existing design flow and need no circuit modification.

Note that X-filling is very powerful whether it is used alone or integrated into an ATPG. The reason is that most test patterns, even after compaction, are sparsely specified [14]; thus, one can properly assign the X bits to effectively reduce the incurred switching activity.

B. Test Pattern Compression

Test data compression has become a necessity as a result of growing test data size in the new generation of technology. Fig. 2 shows a compressor–decompressor architecture. The decompressor decompresses the test stimulus from automatic

Manuscript received December 28, 2008; revised April 20, 2009 and July 27, 2009. Current version published October 21, 2009. An earlier version of this paper was published in the International Test Conference in 2008 [1]. This paper was recommended by Associate Editor A. Ivanov.

M.-F. Wu and J.-L. Huang are with the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: dreamvan.wu@gmail.com; jlhuang@cc.ee.ntu.edu.tw).

X. Wen and K. Miyase are with the Department of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka 820-8502, Japan (e-mail: wen@cse.kyutech.ac.jp; k_miyase@cse.kyutech.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2009.2030440

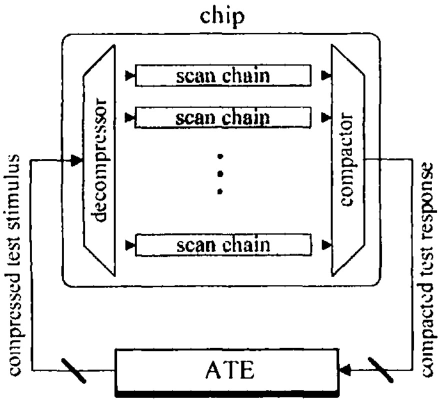


Fig. 2. Test data compression architecture.

test equipment (ATE), and the compactor compacts the test response.

Like X-filling, test pattern compression techniques properly assigns the X bits so that the patterns become compressible. Touba [18] provides a thorough survey on test data compression techniques. Among the various test pattern compression techniques, we will focus on the linear-decompressor-based scheme in this paper. Compared to the code-based and broadcast-scan-based schemes, it has the advantages of high compression rate and very little hardware overhead. Furthermore, it has been widely adopted in industry designs [18].

C. Motivation

Since the effectiveness of X-filling-based launch noise reduction techniques highly depends on the percentage of X bits left for assignment, the effect on power supply noise reduction is severely degraded if test pattern compression is performed first [5]. Similarly, applying launch noise reduction X-filling first will sacrifice the data compression performance.

To resolve this problem, there is a need to develop a launch noise reduction technique that is compatible with the utilized test pattern compression scheme. Although several previous techniques [19]–[23] combine test data compression and test power reduction, they only consider the shift-in induced switching activity and neglect the launch-induced switching activity. These techniques solve the average power issue rather than the power supply noise issue.

D. Contribution

The major contribution of this paper is the presentation of the first X-filling-based technique that generates test patterns with low launch cycle power supply noise in the linear-decompressor-based test pattern compression environment.

The proposed technique is based on the test pattern refinement flow in [10] and the justification-probability-based filling (JP-filling) technique in [17]; it has the following advantages.

1) A fault list shuffling mechanism is introduced to help escape from the local optima. This procedure is very effective for the large industry circuit.
2) An efficient compression-compatible JP-filling is developed to generate compressible patterns with low launch cycle supply noise. Theoretical principles are defined to speed up the X assignment process.

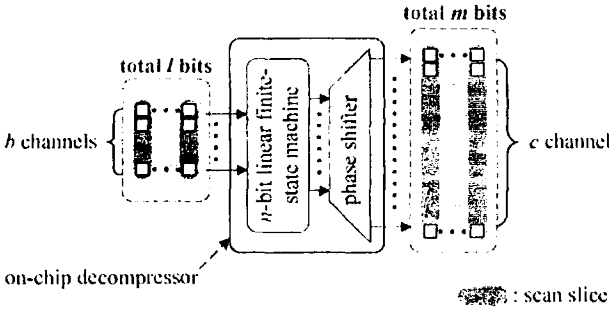


Fig. 3. Typical sequential linear decompressor.

The proposed technique has been validated with the 1989 IEEE International Symposium on Circuits and Systems (ISCAS'89), 1999 International Test Conference (ITC'99), and one industrial design. With minor test set size inflation, the proposed technique reduces the maximum launch cycle weighted switching activity (WSA) of the circuit by 25% for the three largest circuits and 17% for all the circuits.

E. Paper Organization

The organization of this paper is as follows. Section II gives the background of this paper, including linear-decompressor-based test compression and pattern-based power supply noise reduction. Section III illustrates the proposed low launch cycle noise pattern generation technique for linear-decompressor-based compression. Experimental results on benchmark and industrial circuits are shown in Section V. Finally, we conclude this paper in Section VI.

II. Background

A. Sequential Linear Decompression

Linear decompressors consist of only XOR gates, wires, and D flip-flops. A typical sequential linear decompressor is shown in Fig. 3 [24], [25]; it consists of an n-bit linear finite-state machine, such as linear feedback shift register, cellular automata, or ring generator [26], and a phase shifter. At each shift clock cycle, the linear finite-state machine receives b free variables from ATE and outputs n signals which represent the current state. The phase shifter is a combinational logic which consists of XOR gates and wires. It splits the n signals into c signals, which are inputs to the c scan chains. Assuming that the test pattern is m-bit wide and the compressed test stimulus is l-bit wide, we have

$$\frac{l}{b} = \frac{m}{c} = \text{number of scan slices.}$$

For the linear decompressor in Fig. 3, there is a Boolean matrix $\mathcal{M}_{m\times(l+n)}$ from which compressible test patterns are spanned. Let Y be the vector of free variables from ATE. If a

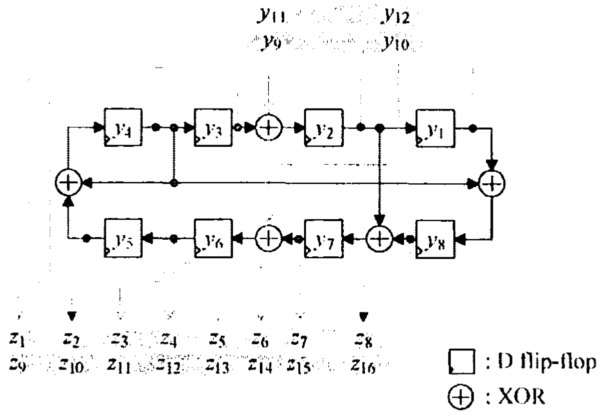

y_{11}
y_{12}
y_9
y_{10}
y_4
y_3
y_2
y_1
y_5
y_6
y_7
y_8
z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8
z_9 z_{10} z_{11} z_{12} z_{13} z_{14} z_{15} z_{16}
□ : D flip-flop
⊕ : XOR

Fig. 4. Example of sequential linear decompressor.

test vector V is compressible, the following linear system has at least one solution:

$$\mathcal{M}Y = V. \quad (1)$$

Fig. 4 shows an 8-bit ring generator with two external input channels as an example. The ring generator implements the primitive polynomial $x^8 + x^6 + x^5 + x + 1$; its initial state is represented by the free variables $y_1, y_2, \ldots, y_8$. Assuming that the test pattern is 16-b wide, denoted by $V = z_1, z_2, \ldots, z_{16}$, there are two scan slices: $(z_1, z_2, \ldots, z_8)$ and $(z_9, z_{10}, \ldots, z_{16})$. The free variables injected into the ring generator are (y_9, y_{10}) in the first scan slice and (y_{11}, y_{12}) in the second scan slice.

Given a partially specified test pattern V, let V_s denote the subvector of V that consists of the specified bits and $\mathcal{M}_s$ the submatrix of $\mathcal{M}$ that consists of the rows corresponding to the specified bits. V is compressible if there exists a solution for the following linear system:

$$\mathcal{M}_s Y = V_s. \quad (2)$$

Another way to determine whether V is compressible is to compute the rank of the corresponding augmented matrix; $[\mathcal{M}_s|V_s] - V$ is compressible if the rank of the augmented matrix is the same as that of the coefficient matrix, i.e., $\mathcal{M}_s$.

In general, multiple solutions exist for a sparsely specified test pattern; however, depending on which solution is used, the final fully specified pattern varies.

B. JP-Filling

The work in [17] presents an improved post-ATPG X-filling technique on [15] and [16]. Called JP-filling, this X-filling technique is both effective and scalable in minimizing launch cycle power supply noise. Given a partially specified test pattern, JP-filling aims at reducing the Hamming distance between the pattern and its output response. The direct result is reduced switching activity at flip-flops in the launch cycle, which indirectly brings down the launch cycle WSA.

The JP-filling flow is shown in Fig. 5. First, it performs three-valued $(0/1/X)$ logic simulation to derive the output response of the given partially specified pattern. Then, each pseudoprimary input (PPI)–pseudoprimary output (PPO) pair is classified as type A, B, C, or D according to Fig. 6. These pairs are processed in the following order: A, B, and C. (Type D needs no further processing.) For each type A pair, JP-filling assigns the PPO value to PPI. For each type B pair, it justifies PPO with the specified PPI value. As for the type C pair, the PPI and PPO are assigned either zero or one according to probability. Note that *all* type A pairs are processed at the same time, so do all type B pairs. Type C pairs of which the difference between zero and one probabilities is greater than a predefined threshold are also processed at the same time.

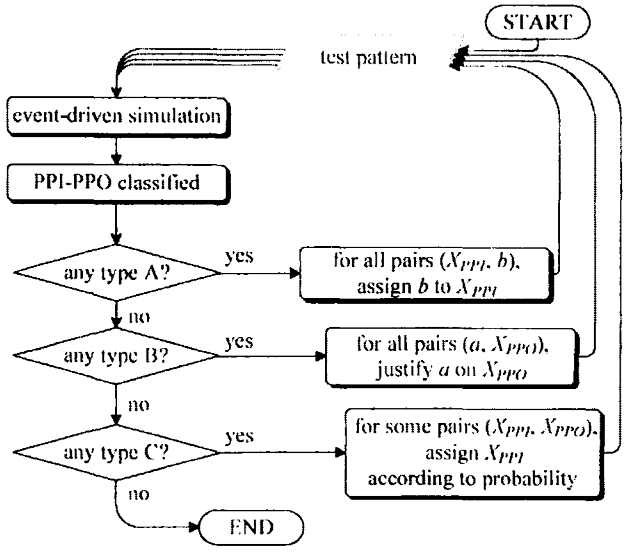

START
test pattern
event-driven simulation
PPI-PPO classified
any type A?
yes
for all pairs (X_{PPI}, b), assign b to X_{PPI}
no
any type B?
yes
for all pairs (a, X_{PPO}), justify a on X_{PPO}
no
any type C?
yes
for some pairs (X_{PPI}, X_{PPO}), assign X_{PPI} according to probability
no
END

Fig. 5. JP-filling flow in [17].

	PPI	PPO
Type A	X	0/1
Type B	0/1	X
Type C	X	X
Type D	0/1	0/1

Fig. 6. PPI–PPO classification in JP-filling.

C. Problem Definition

Assume that the LOC at-speed testing scheme and the linear-decompressor-based test pattern compression technique are utilized. The goal is to generate test patterns that induce low WSA in the launch cycle and are compressible by the linear decompressor. Meanwhile, compared to the standard compressible test pattern generation flow, the increase in test set size should be minimal.

The proposed technique is called "Compressible Supply Noise Reduced Test" and is abbreviated as CSNR-Test hereinafter for convenience.

III. CSNR-Test

Before describing further details of CSNR-Test, we will introduce the concept of implied and free X bits and show how to determine whether an X bit is a free or implied one.

A. Implied and Free X Bits

Definition 1: An X bit in a compressible pattern is ***0(1)*-*compressible*** if the resulting pattern is compressible after the X bit is assigned with 0(1).

Definition 2: An X bit in a compressible pattern is a ***free bit*** if it is both 0-compressible and 1-compressible.

Definition 3: An X bit in a compressible pattern is an ***implied bit*** if it is either 0-compressible or 1-compressible but not both.

Definition 4: The ***implied value*** of a 0-compressible implied bit is zero: the ***implied value*** of a 1-compressible implied bit is one.

Definition 5: A partially specified pattern is a ***free pattern*** if all its X bits are free bits.

The following example depicts the difference between free and implied bits. Consider the following linear system:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} X \\ X \\ 1 \\ 0 \end{bmatrix}. \tag{3}$$

In this example, z_3 and z_4 have been specified as 1 and 0, respectively. The corresponding linear equations are as follows:

$$y_1 \oplus y_2 \oplus y_3 = z_1 \tag{4}$$
$$y_1 \oplus y_3 = z_2 \tag{5}$$
$$y_1 \oplus y_4 = 1 \tag{6}$$
$$y_2 \oplus y_3 \oplus y_4 = 0. \tag{7}$$

From (4), (6), and (7), one arrives at

$$\begin{aligned} z_1 &= y_1 \oplus y_2 \oplus y_3 && (8) \\ &= (y_1 \oplus y_4) \oplus (y_2 \oplus y_3 \oplus y_4) && (9) \\ &= 1 \oplus 0 && (10) \\ &= 1 && (11) \end{aligned}$$

which shows that z_1's value has been implied by the specified z_3 and z_4 even though we have not explicitly specified it. From the $\mathcal{M}$ matrix's point of view, one can arrive at (11) because z_1's row vector, which is the first row vector in $\mathcal{M}$, can be spanned by those of z_3 and z_4. According to the definition, z_1 is a 1-compressible implied bit. On the contrary, we cannot find any linear combination of (6) and (7) to arrive at (5), i.e., the row vector of z_2 cannot be spanned by those of z_3 and z_4. z_2 can be assigned with either 0 or 1. Thus, it is 0-compressible and 1-compressible and, by definition, a free bit.

Theorem 1: Given a compressible test pattern V, if the row vector of an X bit can be spanned by the row vectors of the specified bits, the X bit is an implied bit; otherwise, it is a free bit.

Proof: If the row vector of an X bit can be spanned by some of the row vectors of the specified bits, the value of this X bit can be computed in the same way as we compute z_1 in the aforementioned example. Thus, this X bit is an implied one.

Now, let us prove the other half of the theorem. Since the pattern is compressible, we have

$$\text{Rank}\left([\mathcal{M}_s|V_s]\right) = \text{Rank}(\mathcal{M}_s) = r. \tag{12}$$

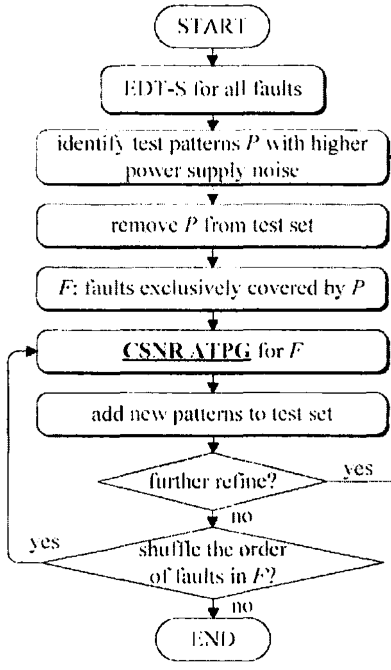

START
EDT-S for all faults
identify test patterns P with higher power supply noise
remove P from test set
F: faults exclusively covered by P
CSNR ATPG for F
add new patterns to test set
further refine?
yes
no
shuffle the order of faults in F?
yes
no
END

Fig. 7. Proposed CSNR-Test flow.

Consider an X bit whose row vector cannot be spanned by $\mathcal{M}_s$. Let us add its row vector to $\mathcal{M}_s$ and denote the resulting matrix by $\mathcal{M}'_s$. The rank of $\mathcal{M}'_s$ is $r+1$. As a result, the corresponding augmented matrix will also have a rank of $r+1$ no matter this X bit is assigned with 0 or 1. Thus, by definition, this X bit is a free bit. ■

Lemma 1: Given a compressible test pattern V, assigning an implied bit with its implied value does not affect the compressibility of the test pattern.

Proof: Assume that $\text{Rank}([\mathcal{M}_s|V_s]) = r$ before the assignment. Since the assigned X bit is an implied one, we have $\text{Rank}(\mathcal{M}'_s) = r$, where $\mathcal{M}'_s$ is as defined in Theorem I. The way the implied value is computed ensures that $\text{Rank}([\mathcal{M}'_s|V'_s]) = r$, where V'_s is the resulting pattern after the assignment. This proves the lemma. ■

One way to determine whether an X bit is free or implied is as follows.

1) Compute the basis of $\mathcal{M}_s$, and denote the set of row vectors in the basis by $\mathcal{B}$.
2) Check whether the X bit's row vector can be spanned by $\mathcal{B}$ or not. If positive, the X bit is implied; otherwise, it is free.

Note that a free bit may become an implied one after some other free bits are specified. Thus, one should execute an X-bit classification process after each X assignment.

B. CSNR-Test Overview

The CSNR-Test flow is shown in Fig. 7, which is a modified version of that in [10]. First, an ATPG that generates compressible at-speed test patterns, e.g., the embedded deterministic test standard (EDT-Standard) [25], is utilized to obtain the initial compressible test set. CSNR-Test then enters the test set

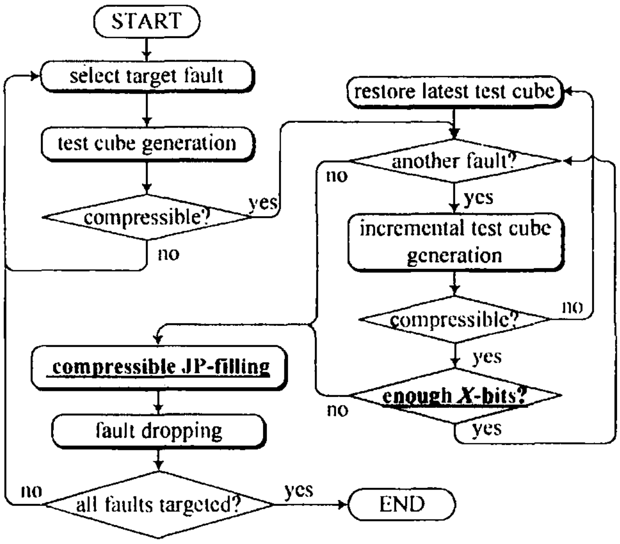


Fig. 8. Proposed CSNR ATPG flow.

refinement process to lower the launch cycle supply noise. In each refinement iteration, the set of patterns whose launch cycle WSA is greater than 99% of the maximum launch cycle WSA in the current test set is identified. These patterns form the set of high supply noise patterns, denoted by P, which is to be refined. The threshold is set to 99% so that the maximum launch cycle WSA is reduced by about 1% in each iteration.

Once identified, P is removed from the test set, and fault simulation is performed to identify the set of faults F that are exclusively detected by P. A launch cycle noise-aware ATPG targets the faults in F. If the newly generated patterns improve the maximum launch cycle WSA, they are accepted; otherwise, they are rejected. In the latter case, CSNR-Test randomly shuffles the order of faults in F and reenters the refinement process. These shuffles prevent CSNR-Test from being trapped in the local optima. If the CSNR-Test fails to improve the maximum launch cycle WSA for five consecutive iterations, i.e., five continuous shuffles, it terminates the refinement process. In our experiments, if more continuous shuffles are allowed, substantial improvement is observed for the three largest circuits.

C. CSNR ATPG

Fig. 8 shows the flow of the CSNR ATPG in Fig. 7. The flow is a modified version of the EDT-Standard. The blocks with underlined text indicate the augmented steps, including the new dynamic compaction constraint and the compressible JP-filling (CJP-filling)—the former ensures that the generated patterns still have enough X bits left, and the latter performs the compressible low launch noise X assignments.

Let s be the number of specified bits in test pattern V. CSNR ATPG stops incremental pattern generation for V and starts generating the next pattern if

$$s \geq 0.5 \cdot \frac{b \cdot m}{c}. \qquad (13)$$

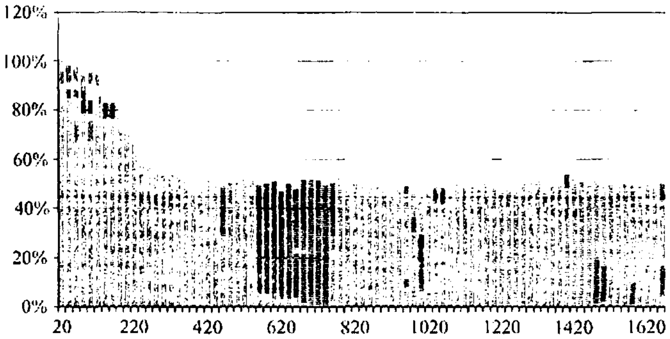


Fig. 9. Acceptance ratio versus assignment index plot.

The following experiment explains why we choose this threshold value. Assume that a 2-to-20 linear decompressor is used; this corresponds to ten times compression. The demonstration circuit is the ISCAS'89 benchmark circuit s38417 whose test pattern width is 1664. The experiment starts from a totally unspecified test pattern. Each time, we randomly select an unspecified bit and randomly assigns 0 or 1 to it. If the resulting test pattern is still compressible, this assignment is accepted or it is rejected, and we invert the assignment so that the pattern remains compressible. The reason to invert a rejected assignment is as follows. As the pattern is compressible before the rejected assignment, there exists a solution to (1) that sets the rejected bit to either 1 or 0. Thus, if we invert a rejected assignment, the resulting pattern must be compressible. At the end, we record which of the 1664 assignments are accepted and which are rejected.

This experiment is repeated 50 times; Fig. 9 shows the average acceptance rate versus assignment index plot. The curve is close to 100% at the beginning when the pattern is so sparsely specified that most available X bits are free ones. On the contrary, the curve eventually converges to 50% at the end when almost all the X bits left are implied ones. We are more interested in the inflection point that occurs at about assignment 164, around which the acceptance rate declines rapidly. Note that the inflection point index is close to the number of free variables from ATE, i.e., $l = (b \cdot m/c) = 166$.

If the compaction termination threshold is greater than 164, the following JP-filling will not perform well because the number of free bits available for low launch noise assignments decreases quickly or is too low. The chosen threshold is half the inflection point index, which is 83 in this example, to avoid the fast declining part before the inflection point and leave more room for launch noise reduction. Therefore, in CSNR ATPG, the threshold for general circuits is set to be $0.5 \cdot (b \cdot m/c)$.

One side effect of the added dynamic compaction constraint is the increased test set size. However, because CSNR ATPG only targets the faults that are exclusively detected by high noise patterns, there is no significant test set size increase in the experimental results.

IV. CJP-Filling

The CJP-filling is the core technology of CSNR ATPG. It tightly integrates launch cycle noise reduction and test pattern compression.

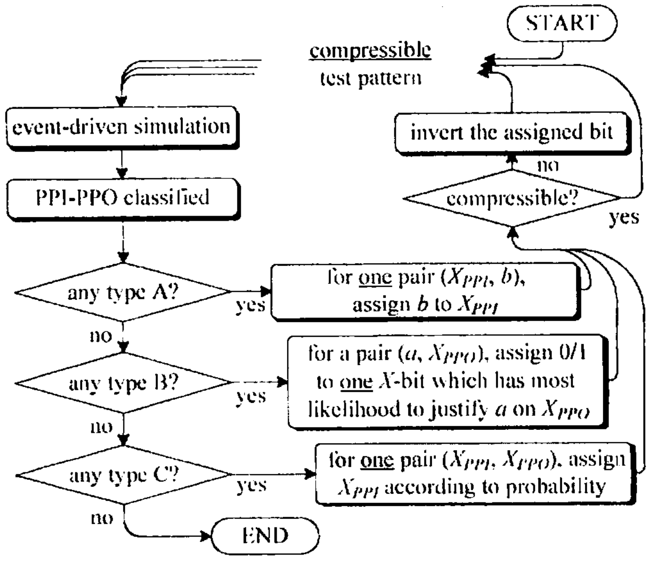

START
compressible
test pattern
event-driven simulation
PPI-PPO classified
any type A?
yes
for one pair (X_{PPI}, b), assign b to X_{PPI}
no
any type B?
yes
for a pair (a, X_{PPO}), assign 0/1 to one X-bit which has most likelihood to justify a on X_{PPO}
no
any type C?
yes
for one pair (X_{PPI}, X_{PPO}), assign X_{PPI} according to probability
no
END
compressible?
yes
no
invert the assigned bit

Fig. 10. Naive CJP-filling flow.

A. *Naive CJP-Filling Flow*

A naive flow for CJP-filling is shown in Fig. 10. The modifications made to the original flow are as follows.

1) The multibit assignments in JP-filling (for type A, B, and C PPI–PPO pairs) are replaced by single-bit assignments. The reason is to improve the acceptance rate and to facilitate the following compressibility check procedure. For type B pairs, we choose the one whose flip-flop has the largest weight and apply *backtrace()* to find the unspecified PPI that has the most likelihood to justify the chosen PPO to its corresponding PPI value.
2) A compressibility checker examines whether the test pattern is compressible after the single-bit assignment. If positive, the assignment is accepted; otherwise, the assignment is rejected, and the assigned bit is inverted.

Note that the initial test pattern is compressible because it is generated by a compression-aware ATPG. Moreover, the process of inverting rejected assignments ensures that the test pattern is always compressible.

The aforementioned flow is inefficient because it assigns one X bit at a time. According to the dynamic compaction constraint in (13), CJP-filling will call the time-consuming compressibility checker for about $m - l/2$ times, which substantially slows it down.

B. *Proposed CJP-Filling Flow*

Compared to the naive approach, the proposed CJP-filling substantially improves the CPU time by enabling multibit assignments (for type A and type C pairs) and by avoiding making unnecessary assignments on the implied bits.

Fig. 11 shows the proposed CJP-filling flow. The flow can be divided into two phases. The upper part constitutes Phase I, which keeps the pattern free. The lower part constitutes Phase II, which makes compressible launch noise reduction assignments. The loop is repeated until the pattern is fully specified.

1) Phase I: In Phase I, we first derive or update the basis associated with the current pattern ("update basis"), from ATPG or Phase II. Based on the updated basis, the X bits are classified as implied or free in "X classification." All the implied bits are assigned with their respective implied values; these assignments must be compressible according to Lemma I. The process of identifying and assigning the implied bits in Phase I prevents CJP-filling from making unnecessary or unacceptable assignments in Phase II. This substantially reduces the number of times the loop is executed and thus improves the CJP-filling efficiency.

2) Phase II: In Phase II, event-driven simulation is first performed to derive the output response of the current pattern. Then, PPI–PPO pairs are classified (Fig. 6), and they are processed in the following ways.

1) Type A ($X, 0/1$). We first perform compatible-free-bit-set identification (CFBS Identification) which identifies a set of X bits that can be *arbitrarily* assigned at the same time without affecting the compressibility. Then, the original JP-filling method is utilized to assign these X bits; the resulting pattern is guaranteed compressible.
2) Type B ($0/1, X$). The process is the same as that in the naive flow. As a single assignment is made and the test pattern is free, this assignment is compressible.
3) Type C (X, X). Like Type A, a set of X bits that can be concurrently assigned is first identified (CFBS Identification). Then, the original JP-filling method is utilized to assign these X bits.

Note that Phase I and CFBS Identification guarantee that the row vector of each newly assigned X bit in Phase II cannot be spanned by those of the previously and newly assigned bits (with itself excluded). Thus, the "update basis" process in Phase I simply adds row vectors of the newly assigned bits in Phase II to the basis.

The following theorem establishes the foundation of the CFBS Identification algorithm.

Theorem 2: A set of free bits can be randomly assigned at the same time, and the resulting pattern is still compressible if, for any of the free bits in this set, its row vector cannot be spanned by the union of the basis before assignment and the row vectors of the other free bits in this set.

Proof: Denote the set of free X bits by $\mathcal{X}$. Assume that the size of $\mathcal{X}$ is q and $\text{Rank}(\mathcal{M}_s) = r$. The property of $\mathcal{X}$ ensures that, after the concurrent random assignments, we have $\text{Rank}(\mathcal{M}'_s) = r + q$ and, as a result, $\text{Rank}([\mathcal{M}'_s|V'_s]) = r + q$. This proves the theorem. ■

The CFBS Identification heuristic for type A and C pairs is as follows. These pairs are sorted in ascending order according to the flip-flop weight, i.e., the fan-out size. This way, flip-flops with larger weights are considered first, as they have more impact on launch noise reduction. The selection process is as follows.

1) Pick the first unprocessed pair in the list as the target pair.
2) If the row vector of the target pair's X bit cannot be spanned by the current basis, the target pair's X bit is selected, and its row vector is added to the basis.
3) If not all pairs are processed, go to 1).

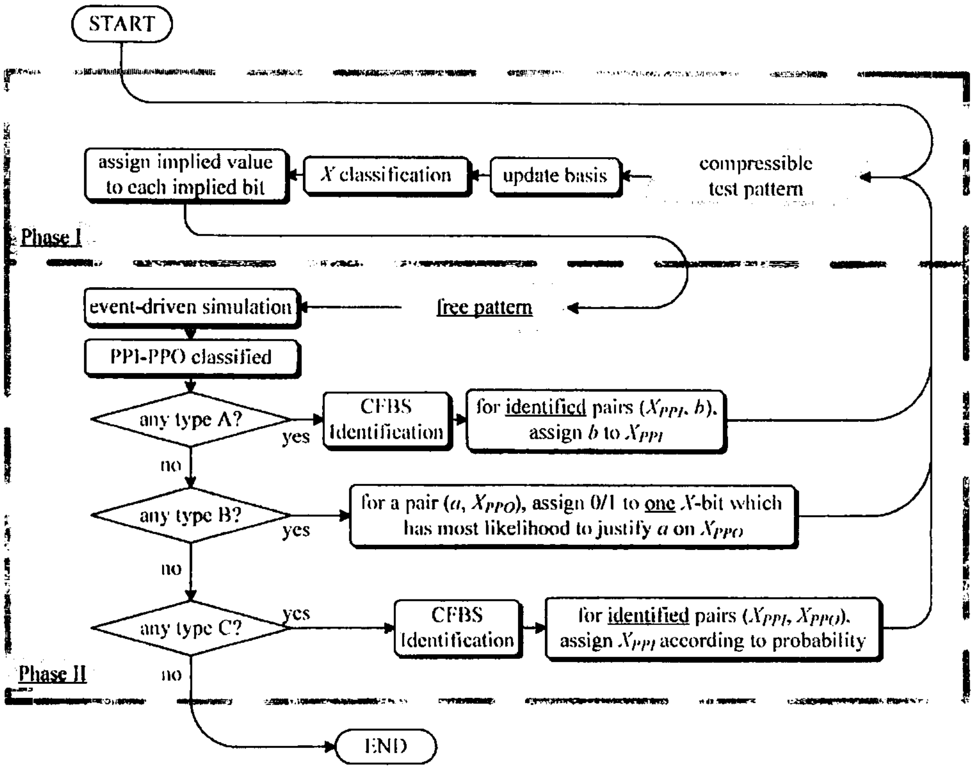

START
assign implied value to each implied bit
X classification
update basis
compressible test pattern
Phase I
event-driven simulation
free pattern
PPI-PPO classified
any type A?
yes
CFBS Identification
for identified pairs (X_{PPI}, b), assign b to X_{PPI}
no
any type B?
yes
for a pair (a, X_{PPO}), assign 0/1 to one X-bit which has most likelihood to justify a on X_{PPO}
no
any type C?
yes
CFBS Identification
for identified pairs (X_{PPI}, X_{PPO}), assign X_{PPI} according to probability
Phase II
no
END

Fig. 11. Proposed CJP-filling flow.

According to Theorem II, the X bits selected this way can be randomly assigned at the same time. Therefore, the JP-filling methods for type A and C pairs can be employed to concurrently target these selected X bits.

CFBS Identification further improves the CJP-filling efficiency. The reasons are as follows.

1) CFBS Identification reduces the number of times the loop is executed because it enables multiassignments.
2) The incurred CPU overhead is small because CFBS Identification implicitly classifies the X bits of the processed pairs and performs "update basis" for the selected bits—the X bits that are not selected are implied bits while row vectors of the selected bits have been added to the basis. In other words, some of the operations in "update basis" and "X classification" are performed in CFBS with a better use model.

C. CJP-Filling Performance Analysis

The following theorem provides the foundation of the performance analysis.

Theorem 3: Given a compressible test pattern V, the maximum number of times the CJP-filling loop is performed is the number of free variables from ATE minus the size of the basis before CJP-filling.

Proof: At the beginning, the rank of $\mathcal{M}_s$ is equal to the size of the basis before CJP-filling. At the end, the rank of $\mathcal{M}_s$, i.e., $\mathcal{M}$, will be smaller than or equal to the number of free variables. Thus, the theorem is proved if we show that the rank of $\mathcal{M}_s$ increases in each loop. This is true if type B pairs are processed because a single free bit assignment is made. This is also true for type A and C pairs, because if q bits are concurrently assigned, the CFBS Identification process ensures that the rank of $\mathcal{M}_s$ increases by q. This proves the theorem. ■

In the following, we compare the performance of the naive and proposed CJP-filling flows. First, their loop run times are about the same. The reason is that the X-bit classification, the CFBS Identification, and the compressibility check are all based on Gaussian elimination. The time complexity of Gaussian elimination is $O(m \cdot l^2)$. Second, the number of times the proposed CJP-filling loop is executed is less than the number of free variables (Theorem III), while the naive CJP-filling executes its loop for about $m - l/2$ times. Finally, the speed-up factor is about $(m - l/2/l) = (m/l) - 0.5$. Note that the first term is the decompressor's compression rate.

Following the example in Section III-C, the number of X bits left for CJP-filling is at least

$$\lfloor 1664 \times (1 - (1/2)(1/10)) \rfloor = 1\,580$$

and the number of free variables is

$$\lceil 1664 \times (1/20) \rceil \cdot 2 + 8 = 176.$$

Thus, the speedup is about 1 580/176 ≈ 9. Note that the analysis is for the CJP-filling process only.

D. Discussions

The CFBS Identification process still has much room for improvement. Currently, it focuses more on multiple assignment

Circuit	Scan architecture	CSNR-Test X-ratio	Test patterns		Maximum launch WSA		Detected faults		CPU* (s)
			EDT-S	CSNR-Test Increase (%)	EDT-S	CSNR-Test Reduction (%)	EDT-S	CSNR-Test Loss (%)	
s35932	20×89	0.95	58	13.79	19085	8.40	50955	0.00	154
	40×45	0.975	60	20.00	19085	6.31	50955	0.00	60
	80×23	0.9875	70	20.00	19085	4.11	50958	0.00	40
s38417	20×84	0.95	495	72.73	19928	20.79	74038	0.11	1253
	40×42	0.975	817	35.37	19465	16.81	73661	-0.02	864
	80×21	0.9875	1046	1.05	18868	7.88	71868	-0.03	346
s38584	20×74	0.95	552	-1.45	17737	10.20	54137	0.01	294
	40×37	0.975	817	1.96	17711	7.89	54245	0.00	325
	80×19	0.9875	1010	1.09	17654	3.29	52628	-0.05	262
b17s	20×73	0.95	2031	-0.49	14461	21.33	92847	0.73	2039
	40×37	0.975	1797	3.23	13828	14.88	87240	0.12	1125
	80×19	0.9875	1548	0.58	13943	8.76	81262	-0.09	572
b22s	20×39	0.95	1121	2.85	15425	8.83	59943	-0.04	315
	40×20	0.975	674	0.00	15243	7.35	53772	0.09	146
	80×10	0.9875	476	0.21	15319	1.25	50053	-0.11	64
D1	20×110	0.95	1729	5.61	14473	32.58	131404	0.01	3594
	40×55	0.975	1671	3.11	14288	27.90	130217	0.07	1556
	80×28	0.9875	1445	1.38	13658	11.76	127927	-0.07	740

*: CSNR-ATPG run time with the EDT-Standard run time excluded.

Fig. 12. Experimental results for transition faults under different scan architectures.

and less on launch noise reduction. As a result, the following problems arise and require further investigation.

1) CFBS Identification favors the X bits whose corresponding flip-flops have larger weights. However, this does not guarantee that the sum of the selected bits' weights is the maximum.
2) CFBS Identification does not consider the impact of the implied bits on launch noise.

Note that minimizing launch cycle WSA could adversely increase the defect level if the resulting WSA becomes less than that in the functional mode. This can be solved by modifying the refinement algorithm—only the patterns that exceed the given launch cycle WSA upper bound are refined.

V. Experimental Results

The experimental setup to validate CSNR-Test is as follows.

1) The on-chip decoder consists of a two-to-eight ring generator in Fig. 4 and a phase shifter which splits the eight signals to internal scan chains. Three scan architectures (20/40/80 internal scan chains) are employed for each circuit. Therefore, the compression rates are 10, 20, and 40 times.
2) The three biggest benchmark circuits from ISCAS'89, the two biggest benchmark circuits from ITC'99, and an industry circuit with $50k$ gates and 1077 scan cells are used.
3) The transition fault model is targeted.

Note the following: 1) The ATPG engine used here is fan-out-oriented based [27], while that in [1] is path-oriented decision-making based, and 2) we also implement an EDT-Standard test pattern generator for comparison.

A. EDT-Standard Versus CSNR-Test

Fig. 12 shows the experimental results. The first column lists the circuit names (D1 is the industry circuit). The second column is the scan architecture in the form of (the number of internal scan chains) × (the number of scan slices), and the third column is the corresponding ATPG X-ratio constraint according to (13). The results are summarized as follows.

1) *Test set inflation.* The numbers of EDT-Standard test patterns are listed in column 4; the CSNR-Test-induced test set inflation percentages are shown in column 5. For s35932 and s38417, up to 72.73% test set inflation is observed; for the other circuits, the inflation ranges from −1.45% to 5.61%.
2) *Maximum launch cycle WSA.* Columns 6 and 7 compare the maximum launch cycle WSAs between EDT-Standard and CSNR-Test. For the 20 scan-chain configurations, the maximum WSA reduction averages 17%, ranging from 8.40% (s35932) to 32.58% (D1). As the number of scan chains and the compression rate grow, the achieved WSA reduction decreases; this may be due to the trend that the launch cycle WSA of EDT-Standard goes down with the growing scan-chain count.
3) *Fault coverage.* Columns 8 and 9 compare the achieved fault coverages of EDT-Standard and CSNR-Test, respectively. For EDT-Standard, one can observe that, as the compression rate increases, the number of undetected faults also increases, particularly for b17s, b22s, and D1. The reason should be that the same ring generator is utilized for all three scan configurations. However, compared to EDT-Standard, CSNR-Test causes negligible fault coverage loss, ranging from −0.11% to 0.73%.

B. Impact of X-Ratio Threshold

To assess the impact of X-ratio threshold, we repeat the experiments for the 2-to-20 configuration (20 scan chains) with different X-ratio thresholds. The setup and results are shown in Fig. 13. Column 3 lists the three X-ratio threshold values. In

Circuit	Scan architecture	CSNR-Test X-ratio	Test patterns		Maximum launch WSA		Detected faults		CPU* (s)
			EDT-S	CSNR-Test	EDT-S	CSNR-Test	EDT-S	CSNR-Test	
				Increase (%)		Reduction (%)		Loss (%)	
s35932	20×89	0.975 0.95 0.925	58	29.31 13.79 10.34	19085	9.55 8.40 7.04	50955	0.00 0.00 0.00	208 154 130
s38417	20×84	0.975 0.95 0.925	495	180.00 72.73 45.45	19928	20.02 20.79 20.22	74038	0.03 0.11 0.03	2521 1253 939
s38584	20×74	0.975 0.95 0.925	552	3.44 -1.45 -0.72	17737	11.32 10.20 10.61	54137	0.00 0.01 0.01	376 294 320
b17s	20×73	0.975 0.95 0.925	2031	5.07 -0.49 -1.33	14461	22.11 21.33 21.28	92847	0.65 0.73 0.45	2720 2039 1974
b22s	20×39	0.975 0.95 0.925	1121	4.28 2.85 5.00	15425	8.83 8.83 12.47	59943	-0.06 -0.04 0.05	323 315 449
D1	20×110	0.975 0.95 0.925	1729	10.64 5.61 5.55	14473	33.72 32.58 32.99	131404	0.05 0.01 0.02	4394 3594 3514

*: CSNR-ATPG run time with the EDT-Standard run time excluded.

Fig. 13. Experimental results for transition faults under different X-ratio thresholds.

addition to 0.95 which is according to (13), two more thresholds 0.95 and 0.9875 are used. The impact of X-ratio threshold on test set inflation, WSA reduction, and fault coverage is summarized as follows.

1) *Test set inflation.* Column 6 shows that higher X-ratio threshold tends to cause more test set inflation; the reason is that CSNR ATPG has less bits to specify for fault detection. Except for some cases in s35932 and s38417, the inflation is within 11% of the EDT-Standard.
2) *Maximum launch cycle WSA.* Launch cycle WSA reduction is shown in column 7; no apparent trend with respect to X-ratio threshold is observed.
3) *Fault coverage.* The fault coverage losses in column 9 show no apparent trend with respect to the X-ratio threshold.

C. Further Analysis

To validate the effectiveness of the proposed CJP-filling and CFBS techniques, we compare the performances of the following three approaches.

1) *Proposed.* This is the proposed CSNR-Test technique.
2) *Without CFBS.* In this version, the CFBS Identification feature is removed, which disables multibit assignments.
3) *Naive.* In this version, the proposed CJP-filling is replaced by the naive CJP-filling (Fig. 10).

The results for maximum launch WSA reduction are shown in Fig. 14. It is interesting that the CFBS Identification improves not only CPU times (as it is intended to) but also the maximum WSA reduction. The reasons should be CFBS favors flip-flops with larger weights. Fig. 15 shows the WSA reduction cost. It is defined as CPU time(s) divided by reduction ratio; thus, lower is better. Clearly, the proposed CJP-filling technique substantially speeds up the X-filling process. CFBS Identification achieves further speedup.

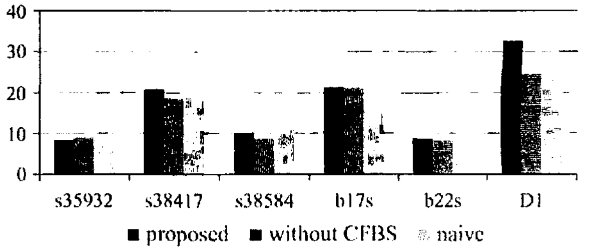


Fig. 14. Comparison of WSA reduction ratios (in percent).

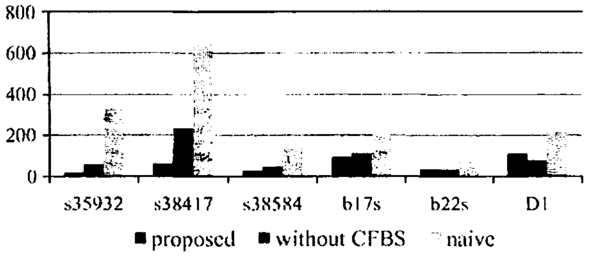


Fig. 15. Comparison of WSA reduction costs.

VI. Conclusion

This paper has presented the first work to overcome the problem of launch-induced yield loss in the linear-decompressor-based test data compression environment. To speed up the X-filling process, X classification and CFBS Identification techniques are developed and integrated into the proposed CSNR-Test flow. The proposed ATPG flow has been validated by benchmark circuits and an industry circuit; the results show that CSNR-Test is a promising flow.

In the future, we will 1) investigate techniques to improve CFBS Identification by making it noise reduction aware and 2) include the circuit-layout-related information in the cost function.

ACKNOWLEDGMENT

The authors would like to thank the Semiconductor Technology Academic Research Center, Japan, for providing the industry circuit.

REFERENCES

[1] M.-F. Wu, J.-L. Huang, X. Wen, and K. Miyase, "Reducing power supply noise in linear-decompressor-based test data compression environment for at-speed scan testing," in *Proc. Int. Test Conf.*, 2008, pp. 13.1.1–13.1.10.

[2] X. Lin, K.-H. Tsai, C. Wang, M. Kassab, J. Rajski, T. Kobayashi, R. Klingenberg, Y. Sato, S. Hamada, and T. Aikyo, "Timing-aware ATPG for high quality at-speed testing of small delay defects," in *Proc. Asian Test Symp.*, 2006, pp. 139–146.

[3] T. Yoshida and M. Watari, "MD-scan method for low power scan testing," in *Proc. Asian Test Symp.*, 2002, pp. 80–85.

[4] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, "A case study of IR-drop in structured at-speed testing," in *Proc. Int. Test Conf.*, 2003, pp. 1098–1104.

[5] S. Ravi, "Power-aware test: Challenges and solutions," in *Proc. Int. Test Conf.*, 2007, pp. 1–10.

[6] S. Wang and W. Wei, "A technique to reduce peak power current and average power dissipation in scan designs by limited capture," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2007, pp. 810–816.

[7] F. Corno, P. Prinetto, M. Redaudengo, and M. Reorda, "Test pattern generation methodology for low power consumption," in *Proc. VLSI Test Symp.*, 1998, pp. 453–457.

[8] S. Wang and S. K. Gupta, "An automatic test pattern generator for minimizing switching activity during scan testing activity," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 8, pp. 954–968, Aug. 2002.

[9] X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K. K. Saluja, L.-T. Wang, K. S. Abdel-Hafez, and K. Kinoshita, "A new ATPG method for efficient capture power reduction during scan testing," in *Proc. VLSI Test Symp.*, 2006, pp. 58–65.

[10] M.-F. Wu, K.-S. Hu, and J.-L. Huang, "An efficient peak power reduction technique for scan testing," in *Proc. Asian Test Symp.*, 2007, pp. 111–114.

[11] V. Devanathan, C. Ravikumar, and V. Kamakoti, "Glitch-aware pattern generation and optimization framework for power-safe scan test," in *Proc. VLSI Test Symp.*, 2007, pp. 167–172.

[12] S. Ravi, V. R. Devanathan, and R. Parekhji, "Methodology for low power test pattern generation using activity threshold control logic," in *Proc. Int. Conf. Comput.-Aided Des.*, 2007, pp. 526–529.

[13] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Supply voltage noise aware ATPG for transition delay faults," in *Proc. VLSI Test Symp.*, 2007, pp. 179–186.

[14] K. M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Lewis, "Minimizing power consumption in scan testing: Pattern generation and DFT techniques," in *Proc. Int. Test Conf.*, 2004, pp. 355–364.

[15] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing," in *Proc. Int. Test Conf.*, 2005, pp. 1019–1028.

[16] S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz, and J. Rajski, "Preferred fill: A scalable method to reduce capture power for scan based designs," in *Proc. Int. Test Conf.*, 2006, pp. 32.2.1–32.2.10.

[17] X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L.-T. Wang, "A novel scheme to reduce power supply noise for high-quality at-speed scan testing," in *Proc. Int. Test Conf.*, 2007, pp. 25.1.1–25.1.10.

[18] N. A. Touba, "Survey of test vector compression techniques," *IEEE Des. Test Comput.*, vol. 23, no. 6, pp. 294–303, Apr. 2006.

[19] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Low power mixed-mode BIST based on mask pattern generation using dual LFSR reseeding," in *Proc. IEEE Conf. Comput. Des.*, 2002, pp. 474–479.

[20] J. Lee and N. A. Touba, "Low power test data compression based on LFSR reseeding," in *Proc. IEEE Conf. Comput. Des.*, 2004, pp. 180–185.

[21] J. Lee and N. A. Touba, "LFSR-reseeding scheme achieving low-power dissipation during test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 396–401, Feb. 2007.

[22] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer, "Low power embedded deterministic test," in *Proc. VLSI Test Symp.*, 2007, pp. 75–83.

[23] G. Mrugalski, J. Rajski, D. Czysz, and J. Tyszer, "New test data decompressor for low power applications," in *Proc. DAC*, 2007, pp. 539–544.

[24] C. Krishna, A. Jas, and N. Touba, "Test vector encoding using partial LFSR reseeding," in *Proc. Int. Test Conf.*, 2001, pp. 885–893.

[25] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 776–792, May 2004.

[26] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generators—New devices for embedded test applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 9, pp. 1306–1320, Sep. 2004.

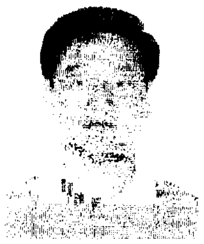
[27] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. C-32, no. 12, pp. 1137–1144, Dec. 1983.

Meng-Fan Wu (S'07) received the B.S. degree in computer science and engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2005. He is currently working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

His current research interests focus on signal integrity issue in very large scale integration testing.

Jiun-Lang Huang (S'96–M'99) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1992 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara (UCSB), in 1995 and 1999, respectively.

From 2000 to 2001, he was an Assistant Research Engineer with the Department of Electrical and Computer Engineering, UCSB. In 2001, he joined the National Taiwan University, where he is currently an Associate Professor in the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering. His main research interests include design for test and built-in self-test for mixed-signal systems, and very large scale integration system verification.



Xiaoqing Wen (S'89–M'93–SM'08) received the B.E. degree from Tsinghua University, Beijing, China, in 1986, the M.E. degree from Hiroshima University, Hiroshima, Japan, in 1990, and the Ph.D. degree from Osaka University, Osaka, Japan, in 1993.

He was an Assistant Professor with Akita University, Akita, Japan, from 1993 to 1997. He was a Visiting Researcher with the University of Wisconsin, Madison, from October 1995 to March 1996. He joined SynTest Technologies, Inc., Sunnyvale, CA, in 1998 and served as its Chief Technology Officer until 2003. He joined the Kyushu Institute of Technology, Iizuka, Japan, in 2004, where he is currently a Professor. His research interests include very large scale integration test, diagnosis, and testable design.

Dr. Wen is a member of the Institute of Electronics, Information and Communication Engineers and the Reliability Engineering Association of Japan.

Kohei Miyase (S'01–M'05) received the Ph.D. degree in computer science and systems engineering from the Kyushu Institute of Technology, Iizuka, Japan, 2005.

In 2007, he joined the Department of Computer Science and Electronics, Kyushu Institute of Technology, where he is currently an Assistant Professor. His research interests include test compression, design for testability, low power test, and fault diagnosis.