On Simultaneous Shift- and Capture-Power Reduction in Linear Decompressor-Based Test Compression Environment

Xiao Liu and Qiang Xu
CUhk REliable computing laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: {xliu,qxu}@cse.cuhk.edu.hk

Abstract

Growing test data volume and excessive test power consumption in scan-based testing are both serious concerns for the semiconductor industry. Various test data compression (TDC) schemes and low-power X-filling techniques were proposed to address the above problems. These methods, however, exploit the very same "don't-care" bits in the test cubes to achieve different objectives and hence may contradict to each other. In this work, we propose a generic framework for test power reduction in linear decompressor-based test compression environment, which is able to effectively reduce shift- and capture-power simultaneously. Experimental results on benchmark circuits demonstrate that our proposed techniques significantly outperform existing solutions.

1 Introduction

With the ever-increasing test data volume for today's large integrated circuits (ICs), on-chip test compression has become the *de facto* design-for-testability (DfT) technique used in the industry. Given test cubes that feature a large percentage of don't-care bits (also known as *X-bits*), various test data compression (TDC) methodologies have been presented in the literature (as surveyed in [30]) and they can be broadly categorized into two categories: (i). nonlinear code-based schemes that use data compression codes to encode test cubes (e.g., [6, 11, 13]); (ii). linear decompressor-based schemes that decompress the input variables, using linear finite state machine (e.g., linear feedback shift registers (LFSR) or ring generator) and/or phase shifter implemented with XOR network (e.g., [3, 14, 20, 32, 37]). Generally speaking, linear decompressor-based schemes offer higher test compression ratio for test sets with very large percentage of *X-bits* than code-based schemes, and hence they are adopted in almost all commercial tools for test data compression.

At the same time, it is well known that the power dissipation of a circuit in scan-based testing can be significantly higher than that during normal operation, in both shift and capture mode [10]. Various techniques have been proposed in the literature to address this issue, which can be categorized into: (i). DfT-based techniques that change the circuit under test (CUT) for test power reduction (e.g., [2, 25, 36, 39]); (ii). low-power test scheduling algorithms that apply modular tests at different time according to given power constraints (e.g., [7, 12, 26]); (iii). power-aware automatic test pattern generation (ATPG) (e.g., [1, 22, 33]); (iv). post-ATPG *X-filling* solutions that manipulate the don't-care bits in test cubes to reduce circuit switching activities (e.g., [4, 17, 23, 34, 35]). As X-filling techniques are more compatible with existing design flow and do not need any circuit modification, they are very popular in the industry.

Since test compression methods and low-power X-filling techniques might take advantage of the very same *X-bits* in test cubes for different objectives, it is essential to develop a holistic solution that targets both issues together. In [16, 18], the authors considered to assign *X-bits* in the test cubes intelligently to reduce capture-power in code-based test compression environment. These techniques, however, cannot be applied to reduce test power in linear decompressor-based test compression environment, because the linear relationship among the *X-bits* in test cubes prevents them to be freely assigned with any value. Recently, Wu *et al.* [38] proposed a compression-compatible X-filling method for linear decompressor-based TDC techniques. In this work, after filling an *X-bit*, the authors identify those *X-bits* that are linearly related to it and conduct implication on them so that the test cube after X-filling is guaranteed to be compressible.

Considering the fact that filling one *X-bit* may imply the values for many other *X-bits* at the same time, both the X-filling order and the filled value should be determined by the set of *X-bits* instead of a single *X-bit*. However, in [38], the authors simply consider to fill the *X-bit* in a test cube that corresponds to the flip-flop with the largest fanout. Consequently, this technique is not quite effective for test power reduction. In contrast to [38], in this paper, we propose to concentrate on filling the free input variables (referred as *X-variable* hereafter) supplied to the linear decompressor, which are *genuinely free* to be assigned with any logic value. By evaluating the impact of *X-variables* on test power dissipation and ordering them intelligently during the X-filling process, we present a generic framework that is applicable to achieve significant power reduction in all scan test phases (i.e., shift-in,

978-1-4244-4867-8/09/$25.00 ©2009 IEEE

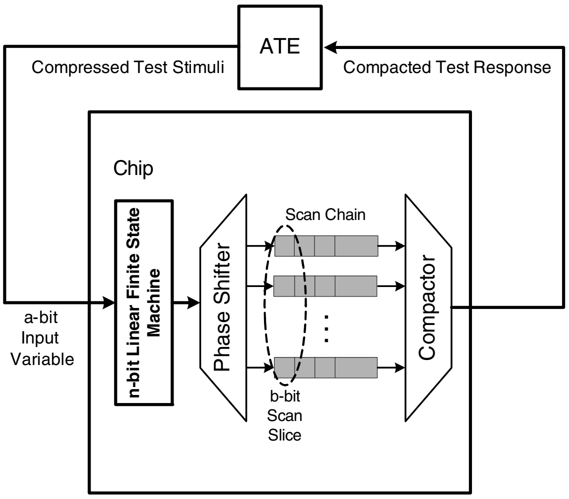

Figure 1. Linear decompressor-based test compression infrastructure.

shift-out and capture phases). In addition, we propose an effective post-processing step that can further reduce test power by flipping the initial filled values for X-variables. To verify the effectiveness of the proposed technique, we conduct experiments on large ISCAS'89 and ITC'99 benchmark circuits and the results show that our proposed technique significantly outperforms existing solutions.

The remainder of this paper is organized as follows. Section 2 reviews related work and motivates this work. In Section 3, we propose the generic power management framework. The applications of the framework for all phases of scan test are illustrated in detail in Section 4. Experimental results on benchmark circuits are then presented in Section 5. Finally, Section 6 concludes this work.

2 Preliminaries and Motivation

2.1 Linear Decompressor-Based Test Data Compression

Generally speaking, a linear decompressor is composed of a n-bit finite state machine (e.g., LFSR or ring generator) to generate test sequences and a *phase shifter* (typically implemented with an XOR network) used to expand these sequences to a large amount of scan chains with reduced linear dependencies, as shown in Fig. 1. To be specific, in each cycle, with a-bit *input variables* from tester, the sequential linear finite state machine generates a new n-bit state and they are expanded to form a b-bit *scan slice* through the phase shifter. Typically, b is much larger than a and hence we can achieve a high test compression ratio.

Consequently, the key idea of linear decompressor-based TDC technique is to generate the large-sized deterministic test cubes by expanding small *input variables*. For each deterministic test cube, its corresponding input variable can be computed by solving a set of linear equations (one equation for each specified bit). Since typically only 1 - 5% of the bits in a test vector are specified, most bits in a test cube do not need to be considered and the size of the input variables can thus be much smaller than that of a test vector.

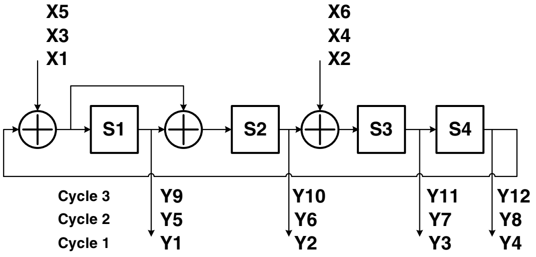

Figure 2. Example of 2-input 4-bit sequential linear decompressor.

$$\begin{pmatrix} 0\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1\,0\,1\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,1\,0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,1\,1\,1\,0\,1\,0\,0\,1 \\ 1\,0\,0\,1\,1\,0\,0\,1\,0\,0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{pmatrix} \tag{1}$$

$$s_4 \bigoplus x_1 = y_1 \tag{2}$$

We use a linear decompressor with 2-bit input variables and 4-bit finite state machine as an example to explain the above process (see Fig. 2). For the sake of simplicity, we omit the phase shifter in this example and we consider to supply test vectors to scan chains for three cycles only. The linear relationship between inputs and outputs of the decompressor can be presented by a system of linear equations $MX = Y$ (see Eq. 1), where X is a vector comprised of input variables and the initial seed for the linear FSM from ATE, Y is a vector for the actual test pattern applied to the circuit, and M indicates the linear characteristics of the decompressor. It is worth to note that all operations in this linear system are in the Galois field modulo 2. For example, the first equation in Eq. 1 can be simply represented as Eq. 2. Also, it should be highlighted that *one input variable in X can affect several related bits in Y due to the linear decompressor.*

Since test pattern are generated by expanding input variables with linear system, the possible test vectors that can be generated in the decompressor is a linear subspace spanned by such matrix. Consider a given test cube in which y_2, y_3, y_7, and y_{12} are specified bits while the remaining ones are all *X-bits*. The sub-set of equations in Eq. 1 are picked out (as shown in Eq. 3), and represented as $M_sX = Y_s$. According to linear algebra theorem, Eq. 3 is solvable (i.e., this test pattern is compressible) if and only if $Rank(M_s) = Rank(M_s \mid Y_s)$.

$$\begin{pmatrix} 1&0&0&1&1&0&0&0&0&0 \\ 0&1&0&0&0&1&0&0&0&0 \\ 1&0&0&1&1&0&0&1&0&0 \\ 1&0&0&1&1&0&0&1&0&0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ y_7 \\ y_{12} \end{pmatrix} \tag{3}$$

To obtain the rank of a matrix, we resort to the widely-used *Gaussian-Jordan elimination* method, and we obtain the elimination result $M_s'X = Y_s'$ as shown in Eq. 4. In the matrix M_s', the '1's in Column 1, 2 and 8 are the so-called *pivots* (i.e., the first non-zero entry when a row is used in elimination), for which the corresponding inputs s_1, s_2 and x_4 are to be fixed to make the equations solvable, while all the remaining inputs in vector X are *X-variables*.

$$\begin{pmatrix} \boldsymbol{\mathit{1}}&\mathbf{0}&0&1&1&0&0&\mathbf{0}&0&0 \\ \mathbf{0}&\boldsymbol{\mathit{1}}&0&0&0&1&0&\mathbf{0}&0&0 \\ \mathbf{0}&\mathbf{0}&0&0&0&0&0&\boldsymbol{\mathit{1}}&0&0 \\ \mathbf{0}&\mathbf{0}&0&0&0&0&0&\mathbf{0}&0&0 \end{pmatrix} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ s_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ \mathbf{x}_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ y_7 - y_2 \\ y_{12} - y_2 - y_7 \end{pmatrix} \tag{4}$$

$$y_2 \bigoplus y_7 = y_{12} \tag{5}$$

At the same time, from Eq. 4, it is obvious that only when the linear relationship shown in Eq. 5 (for the last equation in Eq. 4) is satisfied, $Rank(M_s') = Rank(M_s' \mid Y_s')$ is guaranteed. Therefore, suppose we have a test cube with $y_2 = 1$, $y_3 = 0$, $y_7 = 0$, and $y_{12} = 0$, it is not compressible even though the number of input variables are more than that of specified bits in the test vector. As another example, for a test cube with $y_2 = 1$, $y_3 = 0$, $y_7 = 0$, and $y_{12} = 1$, one possible solution is to assign s_4, x_1, and x_2 as '0's to fix $s_1 = 1$, $s_2 = 0$, $x_4 = 1$, and all other inputs with either '1' or '0'.

2.2 Low-Power X-Filling

It is likely that a CUT's power rating is violated in both shift mode and capture mode in at-speed scan tests. A vast amount of research work has been conducted to address excessive test power in the literature (as surveyed in [10, 21]). Among the various proposed techniques in the literature, low-power X-filling methods, being more compatible with existing design flow as it does not require to modify the CUT or re-run the time-consuming ATPG process, have received lots of attention from both academia and industry.

Butler *et al.* [4] proposed the so-called *adjacent fill* technique to reduce the shift-in power of CUTs, based on the *weighted transition metric (WTM)* for shift-power presented in [27]. Wen *et al.* presented the first low-capture-power X-filling method in [35] that tries to fill *X-bits* in the test stimuli to be as similar to test responses as possible. Although effective in terms of capture-power reduction, this method suffers from high computational complexity. To address this problem, Remersaro *et al.* [23] developed an efficient probability-based X-filling technique, namely *Preferred fill*, which tries to fill all *X-bits* in the test cube in one step, at the cost of less capture-power reduction. Later, Wen *et al.* [34] combined the benefits of the above two works and proposed a hybrid X-filling approach namely *JP-fill*. The above X-filling techniques target either shift-power reduction or capture-power reduction, but not both.

As shift and capture phases have different impact on the CUT's power consumption in scan-based testing, they should be dealt with differently. Elevated shift-power determines the CUT's accumulated power consumption and increases the thermal load that must be transported away from the circuit, which can cause structural damage to the silicon or the package [5, 10]. The main objective in shift-power reduction is thus to decrease it *as much as possible*, so that higher shift frequency can be applied to reduce the CUT's test time without damaging the circuit. Excessive capture-power, on the other hand, has little impact on the CUT's accumulated power consumption due to the short capture period, but it may cause serious IR-drop and prolong circuit delay in test mode only, thus leading to unnecessary test yield loss when at-speed testing is applied [28, 29, 31]. Consequently, the main objective in capture-power reduction is to keep it under a safe threshold to avoid excessive power supply noise that lead to test overkills. As long as this requirement is fulfilled, there is *no* need to further reduce capture-power.

Based on the above observation, Li *et al.* [17] proposed a so-called *iFill* technique that tries to reduce shift-power (including both shift-in power and shift-out power) as much as possible under given capture-power constraint, taking the impact of *X-bits* on both shift- and capture-power into consideration. We consider the same principle in this work for simultaneous shift- and capture-power reduction in test compression environment.

2.3 Linear Decompressor-Based Test Compression with Reduced Power

As discussed earlier, TDC schemes and low-power X-filling techniques manipulate the large amount of *X-bits* in test cubes for test volume reduction and decreased shift- and/or capture-power, respectively. These two objectives, however, may contradict with each other and it is hence essential to develop a *holistic* solution that targets both issues.

Several techniques have been proposed to reduce shift-power in linear decompressor-based test compression environment [9, 15, 19, 24]. These methods mainly try to reduce switching activities during the shift-in phase only. Czysz *et* *al.* [8] considered to reduce both shift- and capture-power in the embedded deterministic test (EDT) environment [20], by introducing DfT structures to the circuit to keep a set of scan cells in a constant state during testing. This technique can significantly reduce test power, however, it is not compatible with conventional design flow since it is a pattern-dependent and circuit-dependent approach.

Recently, Wu *et al.* [38] proposed a compression-compatible X-filling method for capture-power reduction in linear decompressor-based test compression environment. In this method, in each run, they try to fill the *X-bit* in a test cube that corresponds to the flip-flop with the largest fanout, based on *JP-fill* [34]. After filling an *X-bit*, [38] identifies the corresponding *pivot* by Guassian-Jordan elimination (e.g., the *'1'* in the bolded columns shown in Eq. 4). Then, the related *X-variable* is fixed to guarantee the filling of this *X-bit*. Next, an implication procedure is used to fill those *X-bits* that are already *implied* by existing fixed input variables. As the example in Eq. 4 and Eq. 5 shows, when y_2 and y_7 are specified bits, y_{12} is automatically implied by the XOR of these two bits to make the test pattern compressible. The filling process ends when all *X-variables* have been fixed.

2.4 Motivation and Our Contributions

There are several limitations in [38] that makes it less effective for test power reduction. Firstly, and most importantly, this approach targets on filling a single *X-bit* in a test cube in each run and fix one *X-variable* for it. According to [20], the test data compression ratio for industrial circuits can be as high as a few hundred, which means, on average, one input variable can determine the value of several hundred bits in a test vector. Consequently, fixing the value of an *X-variable* that guarantees to avoid one transition in an *X-bit* may even increase capture-power instead of reducing it, since its impact on the other related *X-bits* is unknown. Secondly, due to the linear relationship among *X-bits* in test cubes, the X-filling order has a significant impact on the effectiveness of such techniques, but in [38], the authors simply order the X-filling process according to the fan-out sizes of the *X-bits*. Finally, as we can only estimate the power impact of *X-bits* during the incremental X-filling process, the filled value based on inaccurate estimation may actually result in test power increase.

Motivated by the above, we propose a novel test power reduction framework in linear decompressor-based test compression environment that have the following features:

- we target at filling an *X-variable* instead of filling a particular *X-bit* in each run;
- we propose novel X-filling ordering mechanisms that are effective for shift- and/or capture-power reduction;
- we present a post-processing procedure that tries to flip the initial filled value for X-variables to reduce the impact of inaccurate estimation for their power impact during incremental X-filling;

3 Proposed Framework for Power Reduction in Test Compression Environment

The flowchart for our generic test power reduction framework is shown in Fig. 3. With a given compressible test cube, similar to [38], the original linear system is transformed with Gaussian-Jordan elimination to find the pivots with respect to the specified bits in this pattern. The corresponding input variables are fixed accordingly. Next, those *X-bits* that are already implied with existing linear space are filled. For the remaining *X-variables* (if any), we perform *X-variable* filling and *X-variable* flipping, respectively.

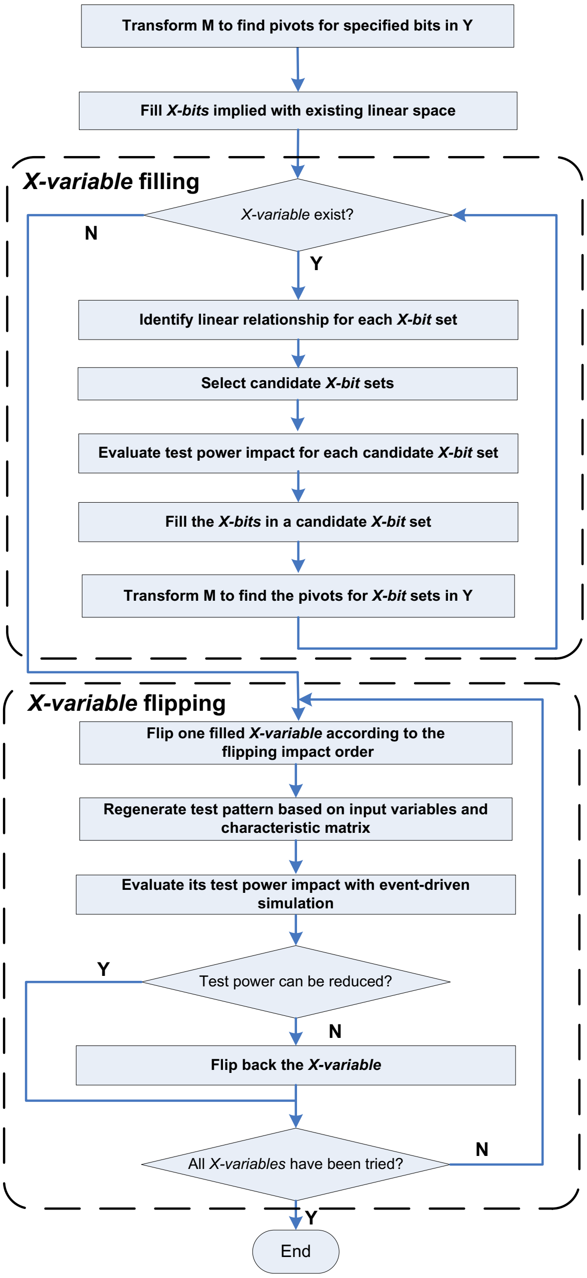
Transform M to find pivots for specified bits in Y
Fill X-bits implied with existing linear space
X-variable filling
X-variable exist?
N
Y
Identify linear relationship for each X-bit set
Select candidate X-bit sets
Evaluate test power impact for each candidate X-bit set
Fill the X-bits in a candidate X-bit set
Transform M to find the pivots for X-bit sets in Y
X-variable flipping
Flip one filled X-variable according to the flipping impact order
Regenerate test pattern based on input variables and characteristic matrix
Evaluate its test power impact with event-driven simulation
Y
Test power can be reduced?
N
Flip back the X-variable
N
All X-variables have been tried?
Y
End

Figure 3. Flowchart for the proposed generic test power reduction framework.

The power impact for fixing an *X-variable*, however, can only be estimated after we know which *X-bits* are affected by this *X-variable*. Therefore, before fixing an *X-variable*, we first identify the linear relationship among the set of *X-bits* in the test cube corresponding to each *X-variable*. With this information, we can order the *X-variable* filling process according to the characteristics of different types of test power. Finally, to overcome the limitation for inaccurate power estimation during the incremental X-filling process, a post-processing *X-variable* flipping process is conducted.

3.1 Linear Relationship Identification

Consider the example decompressor shown in **Fig. 2**, suppose only y_4 is a specified bit '1' in a given test cube while all others are don't-cares. With Gaussian-Jordan elimination, the pivot is found to be in Column 3 of Eq. 6. Thus, the corresponding input variable s_3 is fixed and filled as '1' to make the equation solvable.

$$\begin{pmatrix} 0\,0\,\mathbf{0}\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,\mathbf{0}\,1\,1\,0\,0\,0\,0\,0 \\ 0\,1\,\mathbf{0}\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,\boldsymbol{1}\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,\mathbf{0}\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,\mathbf{0}\,1\,1\,0\,1\,0\,0\,0 \\ 1\,0\,\mathbf{0}\,1\,1\,0\,0\,1\,0\,0 \\ 0\,1\,\mathbf{0}\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,\mathbf{0}\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,\mathbf{0}\,0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,\mathbf{0}\,1\,1\,0\,1\,0\,0\,1 \\ 1\,0\,\mathbf{0}\,1\,1\,0\,0\,1\,0\,0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \mathbf{s}_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \mathbf{1} \\ y_5-1 \\ y_6-1 \\ y_7 \\ y_8 \\ y_9 \\ y_{10}-1 \\ y_{11}-1 \\ y_{12} \end{pmatrix} \tag{6}$$

We then pick out the equations with respect to unspecified y_i from Eq. 6, and re-order them in the characteristic matrix by treating the value of each row vector as a binary number, which yields $M'_r X = Y'_r$ shown in the following.

$$\begin{pmatrix} \mathbf{1\,0\,0\,1\,1\,0\,0\,1\,0\,0} \\ \mathbf{1\,0\,0\,1\,1\,0\,0\,1\,0\,0} \\ 1\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,1\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,1\,0 \\ \mathbf{0\,1\,0\,0\,0\,1\,0\,0\,0\,0} \\ \mathbf{0\,1\,0\,0\,0\,1\,0\,0\,0\,0} \\ 0\,0\,0\,1\,1\,0\,1\,0\,0\,1 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_7 \\ \mathbf{y}_{12} \\ y_2 \\ y_{10}-1 \\ y_9 \\ \mathbf{y}_3 \\ \mathbf{y}_8 \\ y_{11}-1 \\ y_6-1 \\ y_1 \\ y_5-1 \end{pmatrix} \tag{7}$$

Here we have an important observation that several rows in M'_r are identical. Thus, we divide Y_r into the following *X-bit sets*: $\{y_7, y_{12}\}$, $\{y_2\}$, $\{y_{10}-1\}$, $\{y_9\}$,$\{y_3, y_8\}$, $\{y_{11}-1\}$, $\{y_6-1\}$, $\{y_1\}$, and $\{y_5-1\}$, where each set corresponds to one group of rows with the same elements in M'_r. The linear relationship among the *X-bits* in an *X-bit set* conforms to the following theorem.

Theorem 1 For a linear system $M_s X = Y_s$ where all non-zero rows in M_s are identical, all the items in Y_s should be **identical**.

Proof: In a matrix M_s with identical non-zero rows, every row can be spanned by any other one. Hence, we have $Rank(M_s) = 1$. To guarantee $M_s X = Y_s$ to be solvable, we need to have $Rank(M_s|Y_s) = Rank(M_s) = 1$. This equation holds if and only if all the bits in Y_s are identical. □

It is important to note that, setting the values for any of these *X-bit sets* essentially requires an *X-variable* to be fixed accordingly[1]. For our example, the linear relationship $y_7 = y_{12}$ and $y_3 = y_8$ can be obtained, indicated by Eq. 7. The other *X-bits* in the test cube remain to be independent at this moment. After setting the value for an *X-bit set* (and hence the corresponding *X-variable* is fixed), however, these independent bits may become correlated. For example, suppose we set the value for the *X-bit set* $\{y_9\}$ to be '1', it then requires us to fix s_2 since the corresponding *pivot* is in Column 2[2], and we have a further transformed system $M''_r X = Y''_r$, as shown in Eq. 8. Then, the *X-bit sets* are updated as $\{y_7, y_{12}\}$, $\{y_2\}$, $\{y_3, y_8\}$, $\{y_{11}-1\}$, $\{y_6-1\}$, and $\{y_5-1, y_{10}\}$, and we have a new linear relationship among X-bits: $y_{10} \bigoplus y_5 = 1$, obtained from the fact that the Gaussian-Jordan elimination process implicitly merges the original *X-bit sets* $\{y_{10}-1\}$ and $\{y_5-1\}$. From this example, we can observe that, *during the X-variable filling process, the number of X-bit sets keeps decreasing, and for any X-bit set, its size either remains the same or is enlarged through merging*.

$$\begin{pmatrix} 1\,0\,0\,1\,1\,0\,0\,1\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0\,1\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ \mathbf{0\,0\,0\,0\,0\,0\,1\,0\,0\,0} \\ 0\,1\,0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0\,1 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ \mathbf{0\,0\,0\,0\,0\,0\,1\,0\,0\,0} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} y_7 \\ y_{12} \\ y_2 \\ \mathbf{y}_{10} \\ 1 \\ y_3 \\ y_8 \\ y_{11}-1 \\ y_6-1 \\ y_1 \\ \mathbf{y}_5-1 \end{pmatrix} \tag{8}$$

It is also important to note that, *the linear relationship among X-bits depend on the filling order of the X-variables and their filled values*. For the above example, $\{y_{10}-1\}$ and $\{y_5-1\}$ in Eq. 7 cannot be merged together to form a new *X-bit set* if another *X-variable* (e.g., s_4 instead of s_2) is fixed first. Also, their linear relationship is changed to $y_{10} = y_5$ when we fix s_2 that leads to $y_9 = 0$.

3.2 X-Variable Filling

With the capability to identify linear relationship among *X-bits* during the *X-variable* filling process, in each run (to fix one *X-variable*), it is possible to estimate the test power impact

[1] Multiple *X-bit sets* may require to fix the same *X-variable*.

[2] Note, s_2 is fixed but its to-be-filled value is not determined yet since it is related to some other unknown *X-variables*.

for filling every *X-bit sets* with logic '1' or '0' and select the filling option that leads to the minimum test power consumption. This intuitive greedy heuristic, however, is associated with extremely long computational time, since we have many *X-bit sets* (much more than the number of *X-variables*) and we need to conduct extensive simulations to estimate shift-out power and capture-power. As a result, in this work, we first try to select a few candidate *X-bit sets* that might have high test power impact and we fix the value for one of them that results in the least test power consumption (see Fig. 3).

For the selection of candidate *X-bit sets*, we tried many different kinds of metrics to rank these sets and eventually we simply use the size of the *X-bit sets* to rank them, which gives satisfactory results with short runtime. That is, we select those sets that contain large number of *X-bits* as the candidate *X-bit sets*. The effectiveness of this simple evaluation metrics is due to the following reasons:

- Larger *X-bit sets* typically affect more circuitries in the CUT and hence their power impact tend to be higher.
- As shown in Theorem 1, all the items in an *X-bit set* have to be identical. For instance, for the set $\{y_{10}, y_5 - 1\}$ with linear relationship $y_{10} \bigoplus y_5 = 1$, y_{10} and y_5 are either $\{1, 0\}$ or $\{0, 1\}$, but cannot be $\{1, 1\}$ or $\{0, 0\}$. Consequently, the flexibility to set *X-bits* in an *X-bit set* to be their desired values that are able to reduce test power is significantly reduced with the growth of *X-bit set* size. As a result, it is beneficial to balance the size of the *X-bit sets* during the *X-variable* filling process. At the same time, as shown earlier in Section 3.1, *X-bit sets* tend to be merged and their sizes can only grow during the *X-variable* filling process. Therefore, it is better to fill large *X-bit sets* earlier to avoid generating extremely large *X-bit sets* at the end of the *X-variable* filling process.

After selecting several candidate *X-bit sets*, we first evaluate their test power impact when setting to be '1' or '0' (detailed in Section 4), and we select the setting that leads to minimum test power consumption. The corresponding *X-variable* is fixed accordingly. Afterwards, the matrix is transformed again to find the pivots for the rest of the to-be-filled *X-bit sets*. The above procedure iterates itself until all *X-variables* are fixed and filled.

3.3 X-Variable Flipping

One of the limitations of the above greedy *X-variable* filling method is that, it fills one *X-variable* at a time according to its current estimated test power impact. Since there are lots of X-bits during the incremental X-filling process (especially in the beginning), the estimated test power impact cannot be very accurate and hence the filled value may not be beneficial for test power reduction. To tackle this problem, we propose to conduct *X-variable* flipping as a post-processing procedure to further reduce test power consumption (see Fig. 3).

Apparently, it is not computationally-feasible to try all flipping combinations, and we only attempt to flip each *X-variable* once. Whenever we flip an *X-variable*, we obtain a new test pattern and its test power is evaluated according to our power metrics (detailed in Section 4). If test power is reduced, we accept the flipped value for this *X-variable*; otherwise, we try to flip another *X-variable*.

Due to the linear relationship among *X-variables*, their flipping order has a high impact on whether we accept it or not, and different flipping orders may result in distinct test patterns (and hence different test power consumption). In this paper, we define the *flipping impact (FI)* of *X-variables* and use it to order the *X-variable* flipping process.

Generally speaking, flipping one *X-variable* will lead to the flipping of many bits in the original test pattern. To obtain the so-called flipping impact of an *X-variable*, we implement a circuit analysis procedure to calculate the sum of flipping probabilities of all scan cells that are driven by this *X-variable*. We start by initializing the flipping possibility for the *X-variable* to be 1, and then propagate it through its logic cone. Every time a gate is passed, the flipping possibility for the driven scan cells through this gate is likely to be reduced by the other inputs. To estimate such weakening effect, we define a series of *weakening parameters (WP)* calculated by the following equations for various types of gates, in which n is the number of inputs to the gate and $P_i(0/1)$ is the probability of the logic values on input i of the gate.

$$WP_{\text{and/nand}} = \prod_{i=1}^{n-1} P_i(1) \tag{9}$$

$$WP_{\text{or/nor}} = \prod_{i=1}^{n-1} P_i(0) \tag{10}$$

$$WP_{\text{not/xor/xnor}} = 1 \tag{11}$$

The idea behind the above definitions is that all the inputs except for the flipping one should be set as non-controlling value (e.g., '1' for 'and' gate) to propagate this flipping effect. Hence, after the FI goes through a gate, it is weaken as $FI_{out} = FI_{in} \times WP$. These FI values related to an *X-variable* are then summed up as the *flipping impact* for this variable. It is worth noting that, $P_i(0/1)$ is calculated by probability-based simulation with the assumption that the primary inputs to the circuit are "unknown" and hence their probabilities to be logic '1'/'0' are 0.5.

4 Simultaneous Shift- and Capture-Power Reduction with our Proposed Framework

4.1 Capture-Power Reduction

In order to avoid unnecessary test yield loss, scan capture-power should be kept under a safe limit. We use the transition activities in scan cells before and after launch cycle to evaluate capture-power [35]. With the initial stimuli and response probabilities calculated as in [23], we can obtain the capture transition probability (CTP) for the CUT as follows:

$$CTP = \sum_{i=1}^{n} (P_{i1}(s) \times P_{i0}(r) + P_{i0}(s) \times P_{i1}(r)) \qquad (12)$$

, where $P_{i1/i0}(s/r)$ is the probability to have logic '1/0' as the value of test stimulus/response in the i^{th} scan cell and n is the number of scan cells. Here we utilize an one time-frame even-driven simulation to obtain the response probabilities. We use this power metric for the capture-power evaluation in the generic flow that is described in Section 3 for capture-power reduction.

4.2 Shift-Power Reduction

Power consumption can be significantly higher than normal function during shift phase of scan testing, due to the excessive transition between adjacent scan cells in a scan chain. The so-called weighted transition metric (WTM) was proposed to evaluate shift-in and shift-out power caused by these logic value differences. That is, the shift-in/shift-out power of the CUT is estimated by:

$$WTM = \sum_{i=1}^{n} \sum_{j=1}^{m-1} (S_{(i,j)} \bigoplus S_{(i,j+1)}) \times j \qquad (13)$$

, where n is the number of scan chains, m is the number of scan cells in a scan chain and $S_{(i,j)}$ is the logic value of j^{th} scan cell in the i^{th} scan chain. Then the shift power is the total WTM of shift-in and shift-out power.

Since the test pattern is not full specified during the filling process, we defined a probability-based weighted transition metric (PWTM) to evaluate the shift power, where $P_{(i,j)1/(i,j)0}$ is the probability to have '1/0' as the logic value in scan cell (i,j).

$$PWTM = \sum_{i=1}^{n} \sum_{j=1}^{m-1} (P_{(i,j)0} \times P_{((i,j+1)1} + P_{(i,j)1} \times P_{(i,j+1)0}) \times j \qquad (14)$$

The shift-in power from test stimuli can be calculated directly based on scan chain distribution. On the other hand, a time-consuming two time-frame simulation is required to get the response probabilities for evaluating the shift-out power. Note that, the candidate *X-bit set* selection metrics will add the position weight on each *X-bit* in a to-be-selected set by considering its shift-in power impact. We use this power metric for the shift-power evaluation in the generic flow that is described in Section 3 for shift-power reduction.

	DfT profiles			capture-power		shift-power	
circuit	#dff	#pattern	X%	Ave.	Peak	Ave.	Peak
s13207	638	336	95.53%	238.2	280	11785.8	13069
s35932	1728	73	94.52%	861.5	932	73758.3	81870
s38417	1636	1128	96.10%	454.5	554	62865.3	69765
s38584	1426	865	95.84%	408.2	759	53364.3	57616
b22	735	1788	95.74%	89.6	168	13724.2	16994
b17	1415	2199	96.41%	92.1	200	50588.9	56868

Table 1. DfT profiles and Random fill results for benchmark circuits

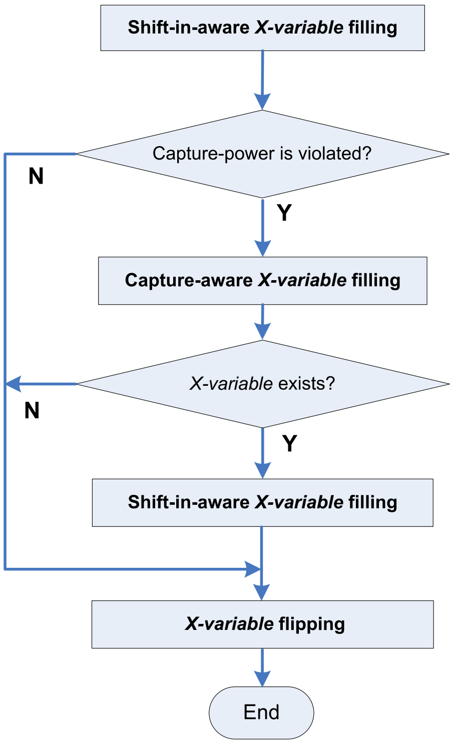
Shift-in-aware *X-variable* filling
Capture-power is violated?
N
Y
Capture-aware *X-variable* filling
X-variable exists?
N
Y
Shift-in-aware *X-variable* filling
X-variable flipping
End

Figure 4. Flowchart for the overall test power reduction framework.

4.3 Test Power Reduction Framework

We propose an overall flow as depicted in Fig. 4 to reduce power consumption during all phases of scan test, i.e., shift-in, capture and shift-out phases. Recall that, our objective for the overall test power reduction flow is to reduce shift-power as much as possible under given capture-power constraint.

Given a compressible test cube, in our proposed flow, the *X-variables* are filled first by targeting shift-in power reduction only to accelerate the process. Then, the fully-specified pattern is evaluated to check whether capture-power constraint is violated. If not, *X-variable* flipping is conducted to reduce both shift-in and shift-out power. Otherwise, the original test cube is reloaded and we target capture-power reduction during the re-filling process. The filling process terminates once the pattern's capture-power can be guaranteed to be under the given safe limit. The remaining *X-variables* are then filled by considering shift-in power reduction again.

It is important to note that, during the *X-variable* flipping process, a variable is flipped only when the shift-power can be reduced and at the same time the capture-power does not exceed the given threshold.

5 Experimental Results

To evaluate the effectiveness of the proposed approaches, we conduct three sets of experiments on ISCAS'89 and ITC'99 benchmark circuits on a 2.13GHz PC with 2GB RAM, including capture-power reduction, shift-power reduction and simultaneous shift- and capture-power reduction.

circuit	X-Filling in [38]		Random Fill +*X-Variable* Flipping					*X-Variable* Filling					*X-Variable* Filling + Flipping				
	Ave.	Peak	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time
s13207	216.2	254	199.0	8.0%	229	9.8%	10.4	169.7	21.5%	199	21.7%	23.6	168.5	22.1%	199	21.7%	38.4
s35932	765.6	842	782.8	-2.2%	828	1.7%	25.3	635.6	17.0%	744	11.6%	37.0	634.1	17.2%	744	11.6%	70.4
s38417	381.5	557	363.6	4.7%	414	25.7%	617.9	285.3	25.2%	417	25.1%	1083.9	283.3	25.7%	389	30.2%	2087.7
s38584	355.5	701	319.3	10.2%	358	48.9%	376.1	280.5	21.1%	605	13.7%	473.4	258.2	27.4%	361	48.5%	1112.9
b22	67.7	148	50.2	25.8%	93	37.2%	188.4	50.1	26.0%	116	21.6%	653.8	44.1	34.9%	84	43.2%	959.0
b17	61.9	179	24.5	60.4%	57	68.2%	1429.6	29.7	52.0%	115	35.8%	5510.1	17.7	71.4%	55	69.3%	8272.8
Ave.				17.8%		31.9%			27.1%		21.6%			33.1%		**37.4%**	

Δ: Difference ratio between X-Filling in [38] and proposed methods

Table 2. Comparison of capture-power reduction.

We use the same linear decompressor as that in [38], which is composed of a 2-input 8-bit ring generator and a phase shift to expand 8 signals to a 20-bit scan slice (i.e. the number of scan chains is 20). The compressible test cubes used in these experiments are provided by Wu *et al.* [38], targeting on transition faults. The DfT profiles for circuits are summarized in Table 1, where the number of scan cells, the number of test cubes, and the percentages of *X-bits* in the test cubes are listed in Columns 2-4, respectively. This table also presents the test power metrics obtained by a baseline approach *Random fill* (see Columns 5-8), wherein we randomly fill X-variables as '0' or '1', and then expand them by the characteristic matrix to obtain the corresponding test patterns.

5.1 Results for Capture-Power Reduction

Table 2 presents the comparison of capture-power between the proposed methods and an existing one [38], assuming all X-variables are used to control capture transitions. Here, *Ave.* and *peak* indicate the average and peak number of capture transitions in all scan cells, respectively. In addition, we denote by $\Delta_{\text{Ave.}}$ and Δ_{Peak} the transition reduction induced by using the proposed methods, compared with X-filling [38]. Note, we are more interested in the peak value when considering capture-power consumption.

The X-filling results for [38] is provided by the authors. Based on the proposed flow, three possible methods are compared against it. The first one is to fill X-variables randomly and then conduct *X-variable* flipping with the *FI* order as described in Section 3.2. As can be observed from Table 2, this simple and fast method can achieve better results than the one proposed in [38], except for one case of average power for s35932. The second method incudes the proposed *X-variable* filling process only, which, on average, leads to 27.1% average power reduction and 21.6% peak power reduction, respectively, compared with [38]. Here, the number of candidate *X-bit sets* is set as 3. The runtime for *X-variable* filling technique is higher than that of *X-variable* flipping, as we need to conduct more simulations to evaluate the capture-power impacts of candidate X-bit sets before fixing each *X-variable*.

We also observe that, for some benchmark circuits (e.g., s35932), *X-variable* filling leads to less capture-power than *X-variable* flipping; while for others (e.g., s38584) , it does not. When combining them together as the third method, we can achieve further capture-power reduction. In particular, we obtain, on average, 37.4% less peak capture transition, at the cost of longer computational time. Note that, there is a special case in terms of peak power for s38584, where the *X-variable* flipping achieves the minimum capture-power. We attribute this phenomenon to the fact that the *X-variable filling*, in spite of its effectiveness, is essentially a greedy heuristic that results in a deterministic solution. Conducting *X-variable* flipping on top of *Random fill*, by contrast, may explore a certain solution space that might include better solutions when compared with the deterministic one.

5.2 Results for Shift-Power Reduction

The shift-power comparison between the proposed methods is illustrated in Table 3, in which all X-variables are filled to reduce shift-power. Note, we are more interested in the average values when considering shift-power.

In Table 3, "Random Fill + *X-Variable* Flipping" is to fill *X-variable* randomly and then conduct flipping based on their impact order; "Shift-in-Aware *X-Variable* Filling" is to greedily fill *X-variable* by evaluating the candidate X-bit sets' shift-in power; Similarly, "Shift-Aware *X-Variable* Filling" also utilize the proposed X-filling method, but it evaluates both shift-in and shift-out power instead of shift-in power only; "Shift-in-Aware *X-Variable* Filling + Flipping" combined "Shift-in-Aware *X-Variable* Filling" with a post-processing *X-variable* flipping procedure, during which we consider both shift-in and shift-out power and the flipping order is based on the impact order described in earlier section.

Compared with the baseline method (i.e., Random fill), all the above methods are able to reduce shift-power significantly. Among them, "Shift-in-Aware *X-Variable* Filling" consumes the shortest time since it does not require to conduct expensive simulation to evaluate shift-out power. Compared to it, "Random Fill + *X-Variable* Flipping" requires longer runtime, and at the same time, its shift-power is higher. Hence, it is not very effective. However, by combining the above two methods together, we are able to achieve the lowest shift-power (on average, 32.3% compared with Random fill), while the running time is kept in a reasonable range. By evaluating the shift-in and shift-out power during *X-Variable* filling, "Shift-Aware *X-Variable* Filling" is able to reduce shift-power slightly more than "Shift-in-Aware *X-Variable* Filling" (29.7% vs. 27.8%). This method, however, is associated with extremely long runtime since it requires to conduct expensive two time-frame simulation to evaluate the shift-out power of each candidate set during the filling process.

circuit	Random Fill + *X-Variable* Flipping					Shift-in-Aware *X-Variable* Filling				
	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time
s13207	10703.0	9.2%	11894	9.8%	17.1	9673.5	17.9%	11383	12.9%	6.3
s35932	42510.2	42.4%	73714	10.0%	34.5	49851.6	32.4%	68302	16.6%	15.8
s38417	55117.9	12.3%	59656	14.5%	1093.6	44481.3	29.2%	57433	17.7%	183.7
s38584	48754.2	8.6%	52459	9.0%	691.2	43782.5	18.0%	50105	13.0%	100.6
b22	12401.3	9.6%	14872	12.5%	355.9	9500.5	30.8%	14905	12.3%	36.9
b17	45144.2	10.8%	49618	12.7%	2669.9	31075.5	38.6%	44919	21.0%	260.7
Ave.		15.5%		11.4%			27.8%		15.6%	

circuit	Shift-Aware *X-Variable* Filling					Shift-in-Aware *X-Variable* Filling+ Flipping				
	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time
s13207	9186.7	22.1%	11017	15.7%	43.5	9361.6	20.6%	11114	15.0%	22.0
s35932	48591.0	34.1%	65382	20.1%	58.4	36178.9	50.9%	66031	18.3%	46.9
s38417	43121.8	31.4%	55434	20.5%	1995.1	43484.3	30.8%	55749	20.1%	1230.9
s38584	41884.7	21.5%	47677	17.3%	1038.7	42810.7	19.8%	49037	14.9%	753.9
b22	9343.7	31.9%	14835	12.7%	1455.8	9263.2	32.5%	14368	15.5%	368.1
b17	31866.8	37.0%	44851	21.1%	11236.0	30751.0	39.2%	44919	21.0%	2830.5
Ave.		29.7%		17.9%			**32.3%**		17.5%	

Δ: Difference ratio between Random fill and proposed methods

Table 3. Comparison of shift-power reduction.

5.3 Results for Simultaneous Shift- and Capture-Power Reduction

Table 4 presents the results for the proposed overall framework (shown in Section 4.3) for simultaneous shift- and capture-power reduction. The capture-power threshold is set to be the peak transitions obtained in [38].

By using this method, the average shift-power can be reduced significantly (similar to the results obtained from targeting shift-power reduction only), while the capture transition remains under a safe threshold. For instance, with this method, the peak capture transition for s13207 is 253, which is less than the threshold value 254, while the average shift-power can be reduced to 9508.9, which is even less than the power value 9673.5 obtained with "Shift-in-Aware *X-Variable* Filling" method as shown in Table 3.

We also observe that, the average capture transition in this technique is higher than that in [38]. This is because, as discussed earlier, the objective of capture-power reduction is to keep it under the safe limit. As long as this objective is achieved, there is no need to further reduce it and the remaining X-variables are utilized to reduce shift-power as much as possible. Since most of the X-variables are utilized for shift-power reduction with the capture-power threshold set according to [38], we can achieve significant shift-power reduction with the proposed overall framework for test power reduction (on average, 28.7% reduction compared with Random fill). The runtime is also affordable since "Shift-in-Aware *X-Variable* Filling" is conducted in most cases.

circuit	capture-power			shift-power				Exe.
	Ave.	Peak	Th.	Ave.	$\Delta_{\text{Ave.}}$	Peak	Δ_{Peak}	Time
s13207	226.0	253	254	9508.9	19.3%	11497	12.0%	25.1
s35932	739.0	836	842	50468.7	31.6%	73954	9.7%	62.0
s38417	415.8	545	557	43489.0	30.8%	57495	17.6%	1232.1
s38584	408.6	700	701	43078.8	19.3%	51190	11.2%	778.8
b22	85.0	147	148	9266.3	32.5%	14368	15.5%	374.5
b17	100.9	179	179	30932.7	38.9%	47768	16.0%	2927.9
Ave.					**28.7%**		13.7%	

Δ: Difference ratio between Random fill and proposed method

Table 4. Experimental results for the overall test power reduction framework.

6 Conclusion and Future Work

In this work, we propose a generic framework for reducing test power in linear-decompressor-based test compression environment during all scan test phases. We propose to target at filling an *X-variables* instead of a particular *X-bit* in each filling process, by identifying the linear relationship among *X-bits* in test cubes. In addition, we propose novel X-filling ordering mechanisms for effective shift- and/or capture-power reduction. Moreover, we present a novel post-processing procedure that tries to flip the initial filled value for X-variables to reduce the impact of inaccurate estimation for their power impact. With the above techniques, our proposed solution significantly outperforms existing methods, as demonstrated in our experimental results on benchmark circuits.

For our future work, we plan to introduce algorithms that are able to concurrently fill multiple *X-variables* to reduce the runtime of the proposed approach and conduct experiments on large industrial circuits.

7 Acknowledgements

The authors wish to thank Mr. Meng-Fan Wu and Prof. Jiun-Lang Huang from National Taiwan University for providing the test cubes and their X-filling results.

This work was supported in part by the General Research Fund CUHK417406, CUHK417807, and CUHK418708 from Hong Kong SAR Research Grants Council, and in part by a grant N_CUHK417/08 from the NSFC/RGC Joint Research Scheme.

References

[1] N. Ahmed, M. Tehranipoor, and V. Jayaram. Supply Voltage Noise Aware ATPG for Transition Delay Faults. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 179–186, 2007.

[2] S. Almukhaizim and O. Sinanoglu. Peak Power Reduction Through Dynamic Partitioning of Scan Chains. In *Proceedings IEEE International Test Conference (ITC)*, paper 9.2, 2008.

[3] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, and T. Onodera. Extending OPMISR Beyond 10x Scan Test Efficiency. *IEEE Design & Test of Computers*, 19(5):65–73, 2002.

[4] K. M. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis, and G. Hetherington. Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques. In *Proceedings IEEE International Test Conference (ITC)*, pages 355–364, Oct. 2004.

[5] C. Shi and R. Kapur. How Power Aware Test Improves Reliability and Yield. EE Times, Sept. 15, 2004.

[6] A. Chandra and K. Chakrabarty. Reduction of SOC Test Data Volume, Scan Power and Testing Time Using Alternating Run-Length Codes. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 673–678, 2002.

[7] R. M. Chou, K. K. Saluja, and V. D. Agrawal. Scheduling Tests for VLSI Systems Under Power Constraints. *IEEE Transactions on VLSI Systems*, 5(2):175–184, June 1997.

[8] D. Czysz, M. Kassab, X. Lin, G. Mrugalski, J. Rajski, and J. Tyszer. Low Power Scan Shift and Capture in the EDT Environment. In *Proceedings IEEE International Test Conference (ITC)*, paper 13.2, 2008.

[9] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer. Low-Power Test Data Application in EDT Environment Through Decompressor Freeze. *IEEE Transactions on Computer-Aided Design*, 27(7):1278–1290, July 2008.

[10] P. Girard. Survey of Low-Power Testing of VLSI Circuits. *IEEE Design & Test of Computers*, 19(3):80 – 90, May-June 2002.

[11] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Variable-Length Input Huffman Coding for System-on-a-Chip Test. *IEEE Transactions on Computer-Aided Design*, 22(6):783–796, June 2003.

[12] V. Iyengar and K. Chakrabarty. Precedence-Based, Preemptive, and Power-Constrained Test Scheduling for System-on-a-Chip. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 368–374, Marina del Rey, CA, May 2001.

[13] A. Jas, J. Ghosh-Dastidar, M.-E. Ng, and N. A. Touba. An Efficient Test Vector Compression Scheme using Selective Huffman Coding. *IEEE Transactions on Computer-Aided Design*, 22(6):797–806, June 2003.

[14] C. Krishna, A. Jas, and N. Touba. Test Vector Encoding Using Partial LFSR Reseeding. In *Proceedings IEEE International Test Conference (ITC)*, pages 885–893, 2001.

[15] J. Lee and N. A. Touba. LFSR-Reseeding Scheme Achieving Low-Power Dissipation During Test. *IEEE Transactions on Computer-Aided Design*, 26(2):396–401, Feb. 2007.

[16] J. Li, X. Liu, Y. Zhang, Y. Hu, X. Li, and Q. Xu. On Capture Power-Aware Test Data Compression for Scan-Based Testing. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 67–72, 2008.

[17] J. Li, Q. Xu, Y. Hu, and X. Li. iFill: An Impact-Oriented X-Filling Method for Shift- and Capture-Power Reduction in At-Speed Scan-Based Testing. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1184–1189, 2008.

[18] X. Liu and Q. Xu. A Generic Framework for Scan Capture Power Reduction in Fixed-Length Symbol-based Test Compression Environment. In *Proceedings Design, Automation, and Test in Europe (DATE)*, accepted, 2009.

[19] G. Mrugalski, J. Rajski, D. Czysz, and J. Tyszer. New Test Data Decompressor for Low Power Applications. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 539–544, 2007.

[20] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee. Embedded Deterministic Test. *IEEE Transactions on Computer-Aided Design*, 23(5):776–792, May 2004.

[21] S. Ravi. Power-Aware Test: Challenges and Solutions. In *Proceedings IEEE International Test Conference (ITC)*, lecture 2.2, 2007.

[22] S. Ravi, V. R. Devanathan, and R. Parekhji. Methodology for Low Power Test Pattern Generation Using Activity Threshold Control Logic. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 526–529, 2007.

[23] S. Remersaro, X. Lin, Z. Zhang, S. Reddy, I. Pomeranz, and J. Rajski. Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs. In *Proceedings IEEE International Test Conference (ITC)*, paper 32.2, 2006.

[24] 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 *Proceedings International Conference on Computer Design (ICCD)*, Sept. 2002.

[25] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici. Scan Architecture with Mutually Exclusive Scan Segment Activation for Shift- and Capture-Power Reduction. *IEEE Transactions on Computer-Aided Design*, 23(7):1142–1153, October 2004.

[26] S. Samii, E. Larsson, K. Chakrabarty, and Z. Peng. Cycle-Accurate Test Power Modeling and its Application to SoC Test Scheduling. In *Proceedings IEEE International Test Conference (ITC)*, paper 32.1, 2006.

[27] R. Sankaralingam, R. R. Oruganti, and N. A. Touba. Static Compaction Techniques to Control Scan Vector Power Dissipation. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 35–40, 2000.

[28] J. Saxena, K. M. Butler, V. B. Jayaram, and S. Kundu. A Case Study of IR-Drop in Structured At-Speed Testing. In *Proceedings IEEE International Test Conference (ITC)*, pages 1098–1104, Oct. 2003.

[29] S. Sde-Paz and E. Salomon. Frequency and Power Correlation between At-Speed Scan and Functional Tests. In *Proceedings IEEE International Test Conference (ITC)*, paper 13.3, 2008.

[30] N. A. Touba. Survey of Test Vector Compression Techniques. *IEEE Design & Test of Computers*, 23(4):294–303, April 2006.

[31] J. Wang, D. M. H. Walker, A. Majhi, B. Kruseman, G. Gronthoud, L. E. Villagra, P. v. d. Wiel, and S. Eichenberger. Power Supply Noise in Delay Testing. In *Proceedings IEEE International Test Conference (ITC)*, page 17.3, 2006.

[32] L.-T. Wang, X. Wen, S. Wu, Z. Wang, Z. Jiang, B. Sheu, and X. Gu. VirtualScan: Test Compression Technology Using Combinational Logic and One-Pass ATPG. *IEEE Design & Test of Computers*, 25(2):122–130, March-April 2008.

[33] S. Wang and S. K. Gupta. ATPG for Heat Dissipation Minimization During Test Application. *IEEE Transactions on Computers*, 47(2):256–262, February 1998.

[34] 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 *Proceedings IEEE International Test Conference (ITC)*, paper 25.1, 2007.

[35] 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 *Proceedings IEEE International Test Conference (ITC)*, pages 1019–1028, November 2005.

[36] L. Whetsel. Adapting Scan Architectures for Low Power Operation. In *Proceedings IEEE International Test Conference (ITC)*, pages 863–872, Oct. 2000.

[37] P. Wohl, J. A. Waicukauski, S. Patel, F. DaSilva, T. W. Williams, and R. Kapur. Efficient Compression of Deterministic Patterns into Multiple PRPG Seeds. In *Proceedings IEEE International Test Conference (ITC)*, paper 36.1, 2005.

[38] 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 *Proceedings IEEE International Test Conference (ITC)*, paper 13.1, 2008.

[39] Q. Xu, D. Hu, and D. Xiang. Pattern-Directed Circuit Virtual Partitioning for Test Power Reduction. In *Proceedings IEEE International Test Conference (ITC)*, paper 25.2, 2007.