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,
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 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.

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 y2, y3, y7, and y12 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 M5X = Y5. According to linear algebra theorem, Eq. 3 is solvable (i.e., this test pattern is compressible) if and only if Rank(M5) = Rank(M5 | Y5).
Gaussian-Jordan elimination

elimination result (Eq. 4) is satisfied, the linear relationship shown in Eq. 5 (for the last equation in y

Therefore, suppose we have a test cube with

from both academia and industry.

flow as it does not require to modify the CUT or re-run the

filling methods, being more compatible with existing design

various proposed techniques in the literature, low-power X-

power in the literature (as surveyed in [10, 21]). Among the

of research work has been conducted to address excessive test

mode and capture mode in at-speed scan tests. A vast amount

2.2 Low-Power X-Filling

It is likely that a CUT’s power rating is violated in both shift

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-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 y2 and y7 are specified bits, y12 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.

![Figure 3. Flowchart for the proposed generic test power reduction framework.](image-url)
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.

\[
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
s_1 \\
s_2 \\
s_3 \\
s_4 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{pmatrix}
= 
\begin{pmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8 \\
y_9 \\
y_{10}
\end{pmatrix}
\]  
(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 \( M'_X = Y'_s \) shown in the following.

\[
\begin{pmatrix}
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
s_1 \\
s_2 \\
s_3 \\
s_4 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{pmatrix}
= 
\begin{pmatrix}
y_7 \\
y_{12} \\
y_2 \\
y_9 \\
y_{10} \\
y_3 \\
y_8 \\
y_{11} \\
y_{12} \\
y_{12}
\end{pmatrix}
\]  
(7)

Here we have an important observation that several rows in \( M'_X \) are identical. Thus, we divide \( Y_s \) into the following X-bit sets: \{\( y_7, y_{12} \), \{\( y_2 \), \{\( y_9 \), \{\( y_{10} \), \{\( y_3, y_8 \), \{\( y_11 \), \{\( y_6 \), \{\( y_1 \), and \{\( y_{12} \), \( y_{12} \)\} in Eq. 7 cannot be merged together to form a new X-bit set if another X-variable (e.g., \( s_3 \) 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, \( s_3 \) 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 \( y_{10} \oplus y_5 = 1 \), \( y_{10} \) 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 \( FI_{\text{out}} = FI_{\text{in}} \times WP \). 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:
$$CTP = \sum_{i=1}^{n} (P_{i1}(s) \times P_{i0}(r) + P_{i0}(s) \times P_{i1}(r))$$  \hspace{1cm} (12)$$

, where $P_{i1/0}(s/r)$ 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:

$$WTM = \sum_{i=1}^{n} \sum_{j=1}^{m-1} (S_{i,j} \oplus S_{i,j+1}) \times j$$  \hspace{1cm} (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 define a probability-based weighted transition metric (PWTM) to evaluate the shift power, where $P_{i,j}(1/0)$ is the probability to have ‘1/0’ as the logic value in scan cell $(i,j)$.

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

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.
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 ΔAve. and Δpeak 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 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-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.

### Table 2. Comparison of capture-power reduction.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>X-Filling in [38]</th>
<th>Random Fill + X-Variable Flipping</th>
<th>X-Variable Filling</th>
<th>X-Variable Filling + Flipping</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave.</td>
<td>Peak</td>
<td>Ave.</td>
<td>ΔAve.</td>
</tr>
<tr>
<td>s13207</td>
<td>216.2</td>
<td>254</td>
<td>199.0</td>
<td>8.0%</td>
</tr>
<tr>
<td>s35932</td>
<td>765.6</td>
<td>842</td>
<td>782.8</td>
<td>-2.2%</td>
</tr>
<tr>
<td>s38417</td>
<td>381.5</td>
<td>557</td>
<td>366.6</td>
<td>-4.7%</td>
</tr>
<tr>
<td>s38584</td>
<td>355.5</td>
<td>701</td>
<td>319.3</td>
<td>10.2%</td>
</tr>
<tr>
<td>b22</td>
<td>67.7</td>
<td>148</td>
<td>50.2</td>
<td>25.8%</td>
</tr>
<tr>
<td>b17</td>
<td>61.9</td>
<td>179</td>
<td>24.5</td>
<td>60.4%</td>
</tr>
</tbody>
</table>

Δ: Difference ratio between X-Filling in [38] and proposed methods
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 524, 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.

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. Jian-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.

Table 3. Comparison of shift-power reduction.

<table>
<thead>
<tr>
<th>circuit</th>
<th>Random Fill + X-Variable Flipping</th>
<th>Shift-in-Aware X-Variable Filling</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>10703.0  9.2%  11894 9.8%  17.1</td>
<td>9673.5  7.9%  11383 12.9%  6.3</td>
</tr>
<tr>
<td>s35932</td>
<td>42510.2  42.4%  73714 10.0%  34.5</td>
<td>49851.6  32.4%  68302 16.6%  15.8</td>
</tr>
<tr>
<td>s38417</td>
<td>55117.9  12.3%  59656 14.5%  1093.6</td>
<td>44481.3  29.2%  57433 17.7%  183.7</td>
</tr>
<tr>
<td>s38584</td>
<td>48754.2  8.0%  52459 9.0%  691.2</td>
<td>43782.5  18.0%  50105 13.0%  100.6</td>
</tr>
<tr>
<td>b22</td>
<td>12401.3  9.6%  14872 12.5%  355.9</td>
<td>9500.5  30.8%  14905 12.3%  36.9</td>
</tr>
<tr>
<td>b17</td>
<td>45144.2  10.8%  49618 12.7%  2689.9</td>
<td>31075.5  38.6%  44919 21.0%  260.7</td>
</tr>
<tr>
<td>Ave.</td>
<td>15.5%   11.4%</td>
<td>27.8%   15.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>circuit</th>
<th>Shift-in-Aware X-Variable Filling+ Flipping</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>9186.7  22.1%  11017 15.7%  43.5</td>
</tr>
<tr>
<td>s35932</td>
<td>48591.0  34.1%  65382 20.1%  58.4</td>
</tr>
<tr>
<td>s38417</td>
<td>43121.8  31.4%  55434 20.5%  1995.1</td>
</tr>
<tr>
<td>s38584</td>
<td>41884.7  21.5%  47677 17.3%  1038.7</td>
</tr>
<tr>
<td>b22</td>
<td>9343.7  31.9%  14835 12.7%  1455.8</td>
</tr>
<tr>
<td>b17</td>
<td>31866.8  37.0%  44851 21.1%  11236.0</td>
</tr>
<tr>
<td>Ave.</td>
<td>29.7%   17.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>circuit</th>
<th>capture-power</th>
<th>shift-power</th>
<th>Exe.</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>226.0  253  254</td>
<td>9508.9  19.3%  11497 12.0%  25.1</td>
<td></td>
</tr>
<tr>
<td>s35932</td>
<td>739.0  836  842</td>
<td>30498.7  31.6%  73954 9.7%  62.0</td>
<td></td>
</tr>
<tr>
<td>s38417</td>
<td>415.8  545  557</td>
<td>43489.0  30.8%  57495 17.6%  1232.1</td>
<td></td>
</tr>
<tr>
<td>s38584</td>
<td>408.6  700  701</td>
<td>43078.8  19.3%  51190 11.2%  778.8</td>
<td></td>
</tr>
<tr>
<td>b22</td>
<td>85.0  147  148</td>
<td>9286.3  32.5%  14368 15.5%  374.5</td>
<td></td>
</tr>
<tr>
<td>b17</td>
<td>100.9  179  179</td>
<td>30932.7  38.9%  47768 16.0%  2927.9</td>
<td></td>
</tr>
<tr>
<td>Ave.</td>
<td>28.7%  13.7%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Δ: Difference ratio between Random fill and proposed methods
References


