

Self-Freeze Linear Decompressors for Low Power Testing

V. Tenentes and X. Kavousianos

Dept. of Computer Science, University of Ioannina
Ioannina, Greece

tenentes@cs.uoi.gr, kabousia@cs.uoi.gr

Abstract—Even though linear decompressors constitute a very effective solution for compressing test data, they cause increased shift power dissipation during scan testing. Recently, a new linear decompression architecture was proposed which offers reduced shift power at the expense however of increased test data volume and test sequence length. In this paper we present a new linear encoding method which offers both high compression and low shift power dissipation at the same time. A new low-cost, test-set-independent scheme is also proposed which can be combined with any linear decompressor for reducing the shift power during testing. Extensive experiments show that the proposed method offers reduced test power dissipation, test sequence length and test data volume at the same time, with very small area requirements.

Keywords—scan testing; lfsr encoding; average power reduction

I. INTRODUCTION

Currently, the most widely adopted test strategy is Test Resource Partitioning. According to this strategy, the test data are stored in a compressed form in the ATE (Automatic Test Equipment) memory, and they are downloaded on chip where they are decompressed by embedded decompressors before they are applied on the core under test (CUT). Many efficient test data compression techniques have been presented so far in the literature. Some of them utilize compression codes [3, 4, 11] while others utilize various broadcasting schemes [24]. However, the most widely adopted test data compression strategy in industry is based on linear decompressors [13, 14, 18, 20]. Linear decompressors constitute an effective means for exploiting the large volumes of unspecified ('X') values existing in test data in order to maximize the compression.

Although linear decompressors are very effective in compressing the test data, they elevate the power dissipation during testing above the functional power budget of the circuit. They fill the 'X' values pseudorandomly and they increase thus both the shift and capture power during scan testing. In particular, shift power dissipation is caused by successive complementary logic values shifted into the scan chains which generate transitions at the scan cells (and inevitably at the combinational part of the circuit) as they travel through the scan chains. Increased switching activity during the scan in-out process is responsible for increased average power dissipation and consequently increased heat dissipation which elevates the temperature of the chip beyond the acceptable limits.

Many techniques have been presented for tackling the increased shift power dissipation during testing. X-filling

techniques exploit the unspecified ('X') bits of test data [6, 10, 16, 21], while structural methods modify either the scan chains [12] or the decompressor [7, 8, 19]. There are also structural methods which are based on power aware encoding algorithms [2, 5, 9, 15] as well as algorithmic methods, which are based on power-aware ATPG processes [17, 22].

One very efficient technique for reducing the shift power dissipation of linear based decompressors was proposed in [19]. According to this technique, whenever it is possible the linear decompressor (ring generator in the case at hand) feeds the scan chains with the same test data in successive scan clock cycles, and reduces that way the volume of successive complementary logic values shifted into the scan chains and consequently the shift power consumed. However, the linear decompressors proposed in [19] require an additional external control which in many cases increases considerably the volume of test data stored in the ATE memory.

In this paper a new power aware linear encoding method is proposed which offers low shift power dissipation and high compression efficiency at the same time. This technique exploits inherent properties of the test data to provide a fairly simple and low-cost weighted pseudorandom scheme which controls the decompression process and enables the power efficient encoding of test data, without the need of any additional control data. The major advantages of this scheme are: a) it constitutes a generic test-set-independent architecture, and b) it can be combined with any linear decompressor scheme for reducing shift power. Moreover, it offers a tradeoff between area overhead and shift power reduction. Extensive experiments show that the combined use of the proposed scheme with the test pattern generator proposed in [19] reduces the test data volume of [19] and achieves great power reductions with very small hardware cost.

II. BACKGROUND & MOTIVATION

Hereafter "test cube" refers to a test pattern consisting of specified ('0' or '1') and unspecified values ('X'), while "test vector" refers to a completely specified test pattern.

Figure 1 presents the classical scan based architecture. The CUT consists of c scan chains of length r (for simplicity we assume that all scan chains are of equal length). The compressed test data are downloaded from the ATE, they are decompressed using the embedded decompressor and they are shifted into the scan chains. For applying a test vector to the CUT the decompressor first generates r successive test slices of

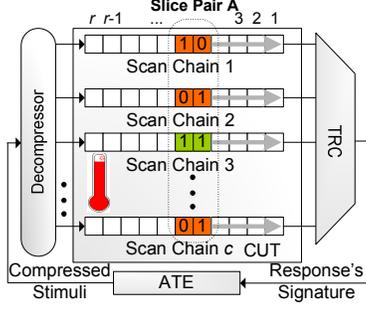


Figure 1. Switching activity caused by successive slices.

size c which are shifted into the scan chains to reach their respective scan slices (hereafter, the term test slice t_j refers to the test bits of test cube t which correspond to scan slice j with $j \in [1, r]$). After the last test slice of t (i.e. t_r) is shifted into the scan chains, t is applied to the CUT and the response is shifted out concurrently with the loading of the next test vector.

As it shown in Figure 1, every pair of successive test slices exhibits potential bitwise incompatibilities, i.e. pairs of successive complementary test bits loaded into the same scan chains. For example test slices denoted as "Slice Pair A" in Figure 1 are incompatible in the bit positions corresponding to scan chains 1, 2, c . As the test slices travel through the scan chains during the scan-in process, every pair of complementary successive test bits causes transitions in the scan chains which propagate through the combinational logic and cause switching activity to the CUT. However, test sets consist also of unspecified values which are exploited to reduce the number of incompatibilities between successive test slices.

Linear decompressors fill 'X' values pseudorandomly, and thus they fail to control the number of incompatibilities between successive test slices. Recently, the authors of [19] proposed a linear based encoding method which exploits the 'x' values, wherever they exist, to reduce incompatibilities between successive test slices, and thus to reduce shift power. According to this method, whenever a group of k ($k > 1$) successive test slices of a test cube are compatible (i.e., every slice in this group exhibits no bitwise incompatibilities with any other slice in this group) one test slice S_k is computed which is compatible with all k test slices. This slice is encoded using the ring generator and it is loaded into the scan chains for k successive clock cycles. This is achieved by the use of a shadow register shown in Figure 2 which can hold its contents if it is properly controlled. Specifically, instead of generating the first slice of this group, the ring generator generates slice S_k and it transfers this slice to the shadow register. This is called *Update* operation. During the next k successive clock cycles, the shadow register holds its contents and loads the scan chains with slice S_k . This is called *Hold* operation. The selection between these two operations of the shadow register requires additional control data which are either provided directly from the ATE (Figure 2a) or they are encoded as compressed stimuli (Figure 2b). In both cases the additional cost is considerable especially when the number of ATE channels is small, and the number of slices per vector is large.

In this paper we show that the additional control data can be completely eliminated by exploiting inherent properties of the

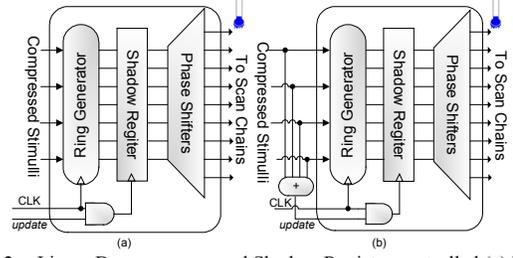


Figure 2. Linear Decompressor and Shadow Register controlled (a) by an additional "update" channel, (b) by compressed stimuli.

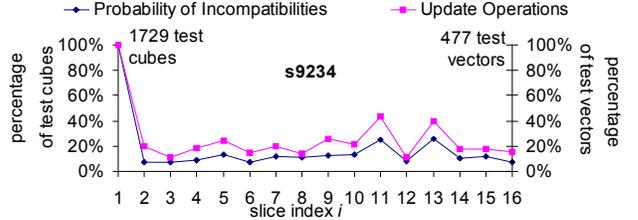


Figure 3. Incompatibilities and Update operations per slice

test data. Specifically, we show that during the generation of test slice t_j of any test cube t , the Update operation occurs with a unique probability. This probability depends solely on the test cubes and in particular on the probability test slice t_j to be incompatible with the test slices corresponding to its predecessor test slices (i.e. t_{j-1}, t_{j-2}, \dots) for any test cube t . By controlling the Update operation using predetermined weighted pseudorandom sequences generated based on these probabilities, the need for additional control data is eliminated. Pseudorandomly controlled Update and Hold operations provide very high power reduction and they can be easily implemented using embedded low-cost hardware modules. Let us see an example.

Example 1. An uncompact test set for s9234 was encoded using the method proposed in [19], for $r=16, c=16$. X-axis in Figure 3 shows the index of each scan slice. For each scan slice left y-axis presents the percentage of test cubes where the respective test slice was incompatible with its predecessor group of $k \geq 1$ successive compatible test slices (line labeled "Probability of incompatibilities"). The right y-axis presents the percentage of test vectors which triggered an Update operation at this scan slice (line labeled "Update Operation"). Note that the number of test vectors is smaller than the number of test cubes, as the ring generator encodes multiple test cubes into the same test vector (this elevates each slice's probability of incompatibility with its predecessors). Nevertheless, it is obvious that both cases yield similar results.

In order to estimate the scan power dissipation in this paper we will use the power dissipation metric proposed in [22]. This metric counts the number of invoked transitions in successive scan cells, while taking into account their relative positions. Let t_j^i, t_{j+1}^i be two successive test bits of test vector t loaded into scan chain i , scan slice j . The average shift power dissipated is given by the formula:

$$S_{av}(t) = 2[r(r+1)]^{-1} \sum_{i=1}^c \left[\sum_{j=1}^{r-1} (r-j)(t_j^i \oplus t_{j+1}^i) \right] \quad (1)$$

III. POWER AWARE ENCODING

In this section we will first present the statistical analysis of test data and then we will present the encoding method.

A. Test Data Analysis

Let TS be a test set consisting of N test cubes for testing a CUT with c scan chains of length r (i.e., each test cube consists of r test slices of size c bits). Hereafter, we will refer to every scan cell using its location in the scan chain structure (for example, scan cell (j, i) is the cell located at the scan slice j , scan chain i). Let $N_0(j, i)$, $N_1(j, i)$ be the number of test cubes of TS with logic value 0, 1 respectively at the scan cell (j, i) .

Definition 1: The *Zero (One) Fill Rate* of scan cell (j, i) is the probability scan cell (j, i) to be assigned to logic value '0' ('1') for any test cube of TS .

The *Zero, One Fill Rates* of scan cell (j, i) are denoted as $f_0(j, i)$, $f_1(j, i)$ and they are computed as follows: $f_0(j, i) = N_0(j, i)/N$, $f_1(j, i) = N_1(j, i)/N$, with $j \in [1, r]$, $i \in [1, c]$

Definition 2: The *Zero (One) Fill Rate* of scan slice j ($j \in [1, r]$) is the probability any scan cell of slice j to be assigned to logic '0' ('1') for any test cube of TS .

The *Zero, One Fill Rates* for slice j are denoted as $f_0(j)$, $f_1(j)$ respectively and they are computed using formulas:

$$f_0(j) = \frac{1}{c} \cdot \sum_{i=1}^c f_0(j, i), \quad f_1(j) = \frac{1}{c} \cdot \sum_{i=1}^c f_1(j, i) \quad \text{with } j \in [1, r]$$

Theorem 1: The probability two test slices x, y of any test cube in TS to be compatible is given by the formula:

$$P_{SC}(x, y) = (1 - f_0(x)f_1(y) - f_1(x)f_0(y))^c \quad (2)$$

Proof: Let x_i, y_i be two bits of test slices x, y corresponding to scan chain i . If x_i, y_i are both specified and complementary then test slices x, y are incompatible. The probability x_i, y_i to be incompatible is equal to $P_{inc}(x_i, y_i) = f_0(x)f_1(y) + f_1(x)f_0(y)$ and thus the probability x_i, y_i to be compatible is equal to $P_c(x_i, y_i) = 1 - P_{inc}(x_i, y_i)$. Slices x, y are compatible when all bit pairs $(x_1, y_1), (x_2, y_2), \dots, (x_c, y_c)$ are compatible. Thus $P_{SC}(x, y) = P_c(x_1, y_1) \cdot P_c(x_2, y_2) \dots P_c(x_c, y_c)$ which gives (2) ■

Lemma 1. The probability a group of k successive test slices $j, j+1, j+2, \dots, j+k-1$ of any test cube in TS to be compatible is

$$P_{gc}(j, j+1, \dots, j+k-1) = \prod_{a=j}^{j+k-2} \prod_{b=j+1}^{j+k-1} P_{sc}(a, b) \quad (3)$$

Proof: A group of successive test slices is compatible if every two slices in this group are compatible. Thus the probability $P_{gc}(j, j+1, \dots, j+k-1)$ is equal to the product of the probabilities $P_{sc}(a, b)$ of every possible slice pair a, b ($a, b \in [j, j+k-1]$), thus

$$P_{gc}(j, j+1, \dots, j+k-1) = \prod_{a=j}^{j+k-2} \prod_{b=j+1}^{j+k-1} P_{sc}(a, b). \quad \blacksquare$$

Let $u_j=1$ ($u_j=0$) denote the occurrence of an Update (Hold) operation during the generation of the test data loaded into scan slice j ($j \in [1, r]$). Then the *Update* vector $U=(u_1, u_2, \dots, u_r)$ represents the Update-Hold operations occurring at the shadow register during the generation of a vector. Since the first scan slice of each vector has no predecessors we set $u_1=1$, that is an update operation occurs always during the generation of the first slice of each vector. Let t be a test cube consisting of r test slices, i.e., $t = (t_1, t_2, \dots, t_r)$.

Lemma 2. Test cube t is encodable for Update vector $U=(u_1, u_2, \dots, u_r)$, if for every $j \in [1, r]$, $k < r$ with $u_j=1$ and $u_{j+1}=u_{j+2}=\dots=u_{j+k}=0$ ($j+k \leq r$) test slices $t_j, t_{j+1}, \dots, t_{j+k}$ are compatible.

Proof: Since $u_j=1$, during the generation of the test slice t_j the shadow register will be updated from the linear generator with a test slice s_j , and since $u_{j+1}=\dots=u_{j+k}=0$ then the same slice s_j will be loaded into scan slices $j, j+1, \dots, j+k$. If test slices $t_j, t_{j+1}, \dots, t_{j+k}$ are compatible then for every $i, i \in [1, c]$ the test bits of all test slices corresponding to scan chain i are either unspecified or exhibit the same logic value ('0' or '1'). Then, the test slice s_j can be computed as follows: for every $i \in [1, c]$ if any of the test slices $t_j, t_{j+1}, t_{j+2}, \dots, t_{j+k}$ exhibit a logic value $v='0'$ or $v='1'$ the respective bit of s_j is set equal to v , else it is left unspecified. In that way s_j is compatible with all test cubes $t_j, t_{j+1}, t_{j+2}, \dots, t_{j+k}$ and thus test cube t is encodable. ■

The most power-efficient Update vector is $U=[1, 0, \dots, 0]$ which can be used for encoding only those test cubes which have all their slices compatible. On the other hand, the most power consuming but in the same time highly efficient in respect to its encoding ability Update vector is $U=[1, 1, \dots, 1]$. This vector can encode any test cube which is encodable by the decompressor. In order to maximize the power efficiency of linear decompressors without compromising their encoding efficiency, we need to maximize the volume of zeros in the Update vector of the decompressor and minimize at the same time the probability any test cube to become un-encodable. However, it is rather unlikely that a single Update vector will suffice to encode all test cubes. In the following, we will show that multiple Update vectors which achieve these goals can be generated in a weighted-pseudorandom fashion.

Let R_j be the probability of an Update operation during the generation of scan slice j ($1-R_j$ is the probability of a Hold operation during the generation of scan slice j). We denote hereafter as *Pseudorandom-Configuration Vector* or simply as *Configuration*, the probability vector $R=[R_1, R_2, \dots, R_r]$. Since $u_1=1$, we also set $R_1=1$.

Theorem 2: The probability any test slice in TS corresponding to scan slice j to be encodable using configuration vector $R=[R_1, R_2, \dots, R_r]$ is given by the formula

$$P_E(j) = \sum_{m=1}^j R_m \cdot P_{gc}(m, \dots, j) \cdot \prod_{k=m+1}^j (1-R_k) \quad (4)$$

Proof: Any arbitrary test slice t_j corresponding to scan slice j is encodable if either the update operation occurs during the generation of this slice or if the update operation occurs during the generation of a predecessor slice t_k (of the same test cube) and all test slices $t_k, t_{k+1}, t_{k+2}, \dots, t_j$ are bitwise compatible. Therefore, for slice j we have the following cases:

1. $P_1=R_j$ is the probability of an update operation at slice j .
 2. $P_2=(1-R_j)R_{j-1}P_{gc}(j-1, j)$ is the probability the update operation to occur at slice $j-1$ (and not at slice j) and at the same time test slices $j-1, j$ to be compatible.
 - ...
 - j. $P_j=(1-R_j)(1-R_{j-1})\dots(1-R_2)R_1P_{gc}(1, 2, \dots, j)$ is the probability the update operation to occur at slice 1 (and not at slices 2..j) and test slices 1, 2, ..., j to be compatible.
- Thus $P_E(j)=P_1+P_2+\dots+P_j$ which gives (4).

Finally, since every test cube is encodable when all its test slices are encodable, we have that the overall probability P_{ET} for any test cube in TS to be encodable using configuration vector $R=[R_1, R_2, \dots, R_r]$ is given by the formula:

$$P_{ET}(R) = P_E(1) \cdot P_E(2) \dots P_E(r) \quad (5)$$

Besides the encoding ability of the decompressor, the Configuration vector R affects also the switching activity during the scan-in process, which is calculated as follows.

Theorem 3: The average scan-in switching activity SC_{av} for any test cube t in TS under Configuration $R=[R_1, R_2, \dots, R_r]$ is:

$$SC_{av}(R) = \frac{c}{r(r+1)} \sum_{j=1}^{r-1} (r-j)R_{j+1} \quad (6)$$

Proof: Let t_j, t_{j+1} be two successive test slices, and let t_j^i, t_{j+1}^i be the test bits of these slices which correspond to scan chain i . Relation (1) gives the average switching activity for any test cube t in TS . The term $t_j^i \oplus t_{j+1}^i$ in (1) is equal to '1' if t_j^i, t_{j+1}^i are different else it is equal to '0'. Given a Configuration vector R , these bits can be different only if an update operation occurs during the generation of slice t_{j+1} . Since R_{j+1} is the probability of an update operation at slice t_{j+1} and $1/2$ is the probability t_{j+1}^i to be generated complementary to t_j^i (assuming linear independent generation) the probability these test bits to be different is $P_{diff}(t_j^i \oplus t_{j+1}^i) = R_{j+1}/2$. Then, relation (1) becomes

$$S_{av}(t) = 2[r(r+1)]^{-1} \sum_{i=1}^c \left[\sum_{j=1}^{r-1} (r-j) P_{diff}(t_j^i, t_{j+1}^i) \right]$$

and provided that t is generated using configuration R we have

$$SC_{av}(R) = 2[r(r+1)]^{-1} \sum_{i=1}^c \left[\sum_{j=1}^{r-1} (r-j) \frac{R_{j+1}}{2} \right] \text{ which gives (6). } \blacksquare$$

In the next Section we will give an algorithm to compute the configuration vector $R=[R_1, R_2, \dots, R_r]$ for any given set of test cubes, which maximizes the switching activity reduction and does not violate a minimum encoding probability $P_{ET}(R)$.

B. Encoding Algorithm

The flowchart of the proposed encoding method is shown in Figure 4 (Figure 4a presents step E3 in details). The main target of the encoding method is to calculate the configuration R which offers the minimum average switching activity without compromising the encoding efficiency of the decompressor. This is shown in Figure 4a. Specifically, $R=[R_1, R_2, \dots, R_r]$ is initially set equal to $[1, 1, \dots, 1]$ which is the configuration offering the maximum encoding probability $P_{ET}(R)=1$. Then, the values of R_j ($j \in [2, r]$) are iteratively decreased until $P_{ET}(R)$ drops below a pre-determined threshold P_{min} or when all R_2, R_3, \dots, R_r reach their minimum values and they cannot be further reduced. We remind that as the values of R_j decrease, both the average switching activity during scan-in and the encoding probability $P_{ET}(R)$ decrease too.

During every iteration, $r-1$ candidate configurations A_2, A_3, \dots, A_{r-1} are generated based on R . Specifically, the candidate

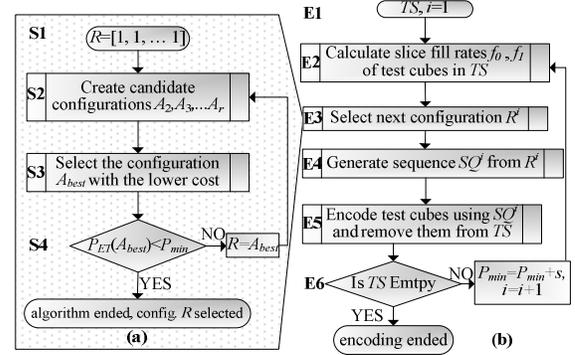


Figure 4. a) Configuration selection algorithm, b) Test set encoding.

configuration A_j ($j \in [2, r]$) is derived from R by decreasing the probability R_j by a predetermined value p (all the other probabilities remain intact). Thus $A_j=[R_1, R_2, \dots, R_j-p, \dots, R_r]$, with $j \in [2, r]$ (note that R_1 is set always equal to 1). Next, candidate configurations are evaluated using the following formula

$$Cost(A_j) = \frac{\Delta P_{ET}(A_j)}{\Delta SC_{av}(A_j)} \quad \text{with} \quad \begin{aligned} \Delta P_{ET}(A_j) &= P_{ET}(A_j) - P_{ET}(R) \\ \Delta SC_{av}(A_j) &= SC_{av}(A_j) - SC_{av}(R) \end{aligned} \quad (7)$$

$\Delta P_{ET}(A_j)$ is the reduction of the encoding probability and $\Delta SC_{av}(A_j)$ is the average switching activity reduction of A_j compared to R . The candidate A_{best} with the lower value of $Cost(A_{best})$ is selected and R is set equal to A_{best} .

Usually, one configuration does not suffice to encode all test cubes. Thus, multiple configurations must be generated using the algorithm shown in Figure 4b. The algorithm begins with set TS of test cubes and it selects the first configuration, let say R^1 using the algorithm shown in Figure 4a. Based on R^1 , it generates a weighted pseudorandom bit sequence SQ^1 , which controls the Update operation during the decompression process (the generation of this sequence is based on pseudorandom properties of simple hardware modules as we will show in the next section). Using SQ^1 the encoding process attempts to encode as many test cubes as possible and it drops the encoded test cubes from TS . This process is repeated and configurations R^2, R^3, \dots (and thus sequences SQ^2, SQ^3, \dots) are selected, until TS becomes empty. At each iteration, the value of P_{min} increases by a step s in order to favor the encoding ability of the next configurations and decrease thus their volume, at the expense however of an increase in the switching activity. Note that relations (2)-(6) are computed in each iteration using the reduced set of test cubes.

IV. ARCHITECTURE

The low power decompression architecture is shown in Figure 5. It consists of the Test Data Decompression Unit (TDU) and the proposed Freeze Control Unit (FCU). TDU is a classical decompression architecture and it consists of the linear decompressor, the shadow register, and the phase shifter. Even though any linear decompressor can be used, we use in this paper ring generators [18], as in the case of [19]. FCU generates the *update* signal which controls the shadow register based on the configuration R (when *update*=1 the Update

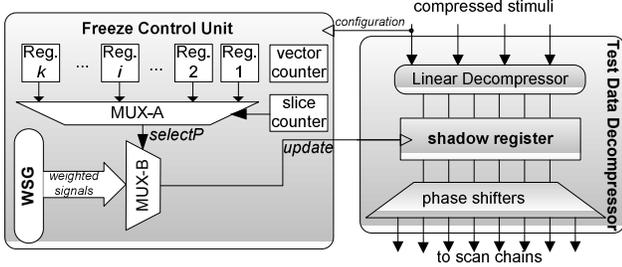


Figure 5. Self-freeze architecture

operation is applied). It consists of a set of r registers which store the configuration vector R_1, R_2, \dots, R_r , the slice counter which selects the register for the next generated test slice, and the Weighted Signal Generation Unit (WSG) which generates a set of weighted pseudorandom signals with pre-determined weights. WSG unit generates a set of n pseudorandom signals $WS_0, WS_1, \dots, WS_{n-1}$ with probabilities $W_0 < W_1 < \dots < W_{n-1}$ respectively. Specifically, signal WS_i is assigned to logic value '1' with probability W_i and to logic value '0' with probability $1 - W_i$. Depending on the configuration R , register j is loaded from the ATE before the decompression begins with a value d in the range $[0, n-1]$. d selects the input of MUX-B which corresponds to signal WS_d with probability W_d equal to R_j . Slice counter counts from 1 to j and whenever it is equal to j , register j selects signal WS_d which is driven to the update input of the shadow register. Thus the Update operation is applied with probability R_j during the generation of the test data loaded into scan slice j .

Many techniques have been presented in the past for designing WSG units, like [1], [23]. In this paper we use a small LFSR which is loaded initially with a randomly selected seed, and a few AND gates of 2, 3 and 4 inputs driven by the LFSR cells (note that this small LFSR operates only as a pseudorandom generator and it does not participate in the decompression process). Since each LFSR cell is set to the logic value '1' with probability $P_1=1/2$, every q -input AND gate produces a weighted pseudorandom signal at its output with probability P_1 equal to $(1/2)^q$. By using three AND gates of 2, 3 and 4 inputs driven by different LFSR cells, and by using both the normal and the inverted outputs of the AND gates, we generate signals with the following P_1 probabilities: 0.0625, 0.125, 0.25, 0.5, 0.75, 0.875, 0.9375. Note that during the encoding process (Figure 5a) the values R_j are selected only from this set (i.e. p is adjusted each time in such a way that values from this set are selected). Additionally, the encoding of the test cubes is done using the pseudorandom sequences generated at the outputs of the WSG unit. In other words, after the calculation of a configuration R^i , the WSG unit is simulated and it generates a pseudorandom sequence SQ^i using signals $WS_0 \dots WS_{n-1}$. The encoding process uses this predetermined sequence SQ^i to encode test cubes.

We have to note that the area overhead of the architecture shown in Figure 5 increases as the number of slices (and thus the number of registers) increases. To overcome this problem an area-efficient alternative architecture is also proposed which reduces the number of registers at the expense of a slight performance degradation. Specifically, k ($k < r$) registers are

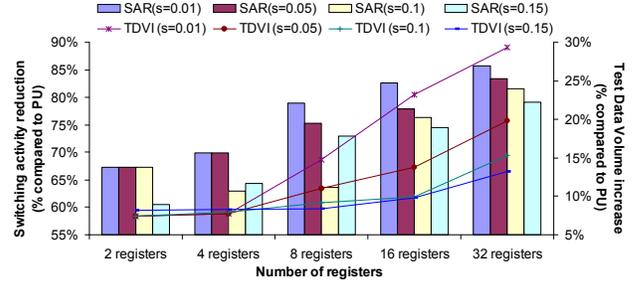


Figure 6. Switching Activity Reduction and Test Data Volume trade-off.

used and every register corresponds to more than one slices. The registers are assigned to scan slices in a modulo- k fashion. For example, register j is used for controlling the update signal during the generation of scan slices $j, j+k, j+2k, \dots$ (note that scan cell 0 is excluded from this process because an update operation occurs always during the generation of this slice). In this case, the encoding method shown in Figure 4a is modified accordingly in order to consider the reduced set R_1, R_2, \dots, R_k . Thus, the process begins with set R where $R_j = R_{j \bmod k}$ and at each iteration k candidate configurations are generated.

V. EXPERIMENTS

The proposed method was developed using the C programming language. We conducted experiments on test sets for complete stuck-at coverage generated using a commercial ATPG tool for the largest ISCAS'89. All the shift power estimations were done using formula (1).

Figure 6 presents the test data volume (TDV) increase (right y-axis) and the switching activity reduction (SAR at the left y-axis) of the proposed technique against the power unaware dynamic encoding (PU) method. In both cases the s13207 benchmark circuit was used assuming $c=16, r=44$ and the proposed method was applied for 2, 4, 8, 16 and 32 registers and various values of parameter s ($s=0.01, s=0.05, s=0.1$ and $s=0.15$). It is obvious that as the number of registers increase, the pseudorandom sequences reflect more accurately the specific requirements of the scan slices and thus the switching activity reduction improves. It is worth noting however, that even a relatively small number of registers suffices to achieve very high reduction of the switching activity. On the other hand, the TDV increases as the number of registers increase, because more data are required for loading the registers for every configuration. In respect with parameter s , it is obvious that small values of s improve the power reduction compared to PU but also increase the test data volume. The reason is that small values of s favor the switching activity reduction at the expense however of generation of more configurations.

Table I presents the results of a) the proposed technique using 8 registers, b) the power unaware dynamic encoding (PU) and c) the deterministic freeze method (DF) presented in [19] and re-implemented here. We note that in the implemented DF method we assume that the control data are sent from the ATE to the CUT using an extra channel (Figure 2a). In all cases 8 or 16 scan chains and 1 or 2 ATE channels were used (excluding the control channel for DF). Note that both DF and

TABLE I. RESULTS & COMPARISONS

circuit	SA reduction %		TSL			TDV without repeat			TDV with repeat		
	DF	Prop.	PU	DF	Prop.	PU	DF	Prop.	PU	DF	Prop.
s5378	90%	78%	250	463	392	7	25	10	3	6.3	5.2
s9234	80%	67%	309	611	419	19	38	26	7	14	10
s13207	96%	78%	276	432	380	24	57	33	18	28	22
s15850	94%	80%	293	511	347	22	60	27	17	30	20
s38417	98%	85%	626	2374	924	65	493	96	33	124	48
s38584	98%	82%	267	1088	373	49	300	68	37	150	51

TDV reported in Kbits

PU methods were implemented by omitting the fault simulation step in order to provide fair comparisons with the proposed method (the fault simulation step can be trivially included in all cases). The first column presents the circuit's name, while the next two columns present the average switching activity reduction of both DF and the proposed method against the power unaware method (PU). The next three columns in Table I present the test sequence length of the PU, the DF and the proposed technique. The results indicate that the proposed technique achieves a vast reduction of the average switching activity (67%-85%). Note that, the proposed method is inferior compared to DF with respect to the switching activity. However this is attributed to the high TSL of the DF method which is a consequence of the trend of DF to minimize the volume of Update operations and to limit thus the ability of the decompressor to encode multiple test cubes on the same generated vector. As a result, the number of generated vectors (i.e. the TSL) increases considerably especially for large test sets. Nevertheless, the switching activity of the proposed technique remains significantly lower than PU and thus the probability to comply with the functional power budget of the CUT (which is the most important target of any low power testing technique) is still very high.

The next six columns present the test data volume (TDV) comparisons between the DF and the proposed method. The first three of these columns report the TDV results assuming that the repeat command is not supported by the ATE, while the next three report the TDV results assuming that the repeat command is supported by the ATE. As it has already been mentioned in [19] the use of the repeat command considerably reduces the TDV. The proposed method achieves very high TDV reduction against DF in both cases (in the range of [30%-81%] whenever the repeat command is not supported and in the range of [17%-66%] whenever the repeat command is supported).

Finally, we synthesized the proposed scheme for 8 registers. The hardware overhead of the proposed FCU unit is less than 100 gate equivalents (one gate equivalent corresponds to a 2-input nand gate). This overhead is less than the 25% of the overhead of the TDU unit. Additionally, we note that the same decompressor can be used for testing any number of cores, which makes its application very attractive to modern SoCs.

VI. CONCLUSIONS

In this paper we presented a new linear encoding method which exploits inherent properties of test data to reduce the scan-in switching activity during testing. A simple and low-cost scheme was also presented which can be combined with

any linear-decompressor architecture and achieves very high reduction of the switching activity at the expense of only a small increase on the test data volume. Compared to the state-of-the-art power aware linear encoding method, the proposed method provides comparable shift power reduction with considerably lower test data volume.

REFERENCES

- [1] N. Ahmed, M. Tehranipoor and M. Nourani, "Low Power Pattern Generation for BIST Architecture," in Proc. Int. Symp. on Circuits and Systems (ISCAS'04), vol. 2, pp. 689-692, 2004.
- [2] K. Chandra, K. Chakrabarty, "A unified approach to reduce SOC test data volume, scan power and testing time", IEEE Trans. on CAD Vol. 22, No 3, March 2003 pp 352 – 363.
- [3] A. Chandra and K. Chakrabarty, "System-on-a-chip test data compression and decompression architectures based on Golomb codes," IEEE Trans. Computer-Aided Design, vol. 20, pp. 355-368, Mar. 2001.
- [4] A. Chandra, K. Chakrabarty "Test Data Compression and Test Resource Partitioning for System-on-Chip using Frequency-Directed Run-Length (FDR) Codes," IEEE Trans. on Comp., vol. 52, no. 8, pp. 1076-1088.
- [5] K. Chandra, K. Chakrabarty, "Low-power scan testing and test data compression for system-on-a-chip", IEEE Trans. on CAD, Vol. 21, No 5, May 2002 pp:597-604
- [6] A. Chandra, R. Kapur, "Bounded Adjacent Fill for Low Capture Power Scan Testing", 26th IEEE VTS, 2008, pp.: 131-138.
- [7] D. Czys, et al., "Low-Power Test Data Application in EDT Environment Through Decompressor Freeze", IEEE Trans. on CAD of Integrated Circuits and Systems 27(7): 1278-1290 (2008).
- [8] D. Czys, et al., "Low Power Embedded Deterministic Test", IEEE VTS 2007, pp. 75-83.
- [9] P. Girard, et al., "A test vector inhibiting technique for low energy BIST design", in Proc. VTS, 1999, pp. 407-412.
- [10] S. Kajihara, K. Ishida, K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing", Proc. VTS, pp. 160-165, 2002.
- [11] X. Kavousianos, E. Kalligeros and D. Nikolos, "Optimal Selective Huffman Coding for Test-Data Compression," IEEE Trans. on Computers, vol. 56, no. 8, pp. 1146-1152, June 2007.
- [12] Ko H. F., Nicolici N., "Automated Scan Chain Division for Reducing Shift and Capture Power During Broadside At-Speed Test", IEEE Trans. on CAD, vol. 27, No 6, Page(s): 2092-2097.
- [13] B. Koenemann, "LFSR-Coded Test Patterns for Scan Designs", Proc. European Test Conf. (ETC 91), VDE Verlag, 1991, pp. 237-242.
- [14] C.V. Krishna, A. Jas, and N.A. Toubia, "Test Vector Encoding Using Partial LFSR Reseeding", Proc. Int'l Test Conf., 2001, pp. 885-893.
- [15] J. Lee, and N. Toubia, "Low Power Test Data Compression Based on LFSR Reseeding", in Proc. ICCD, 2004, pp. 180-185.
- [16] 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", DATE March 10-14, 2008, p. 1184.
- [17] Lin H.-T., Li J.C.-M., "Simultaneous capture and shift power reduction test pattern generator for scan testing", IET Proceedings, Computers & Digital Techniques, vol. 2, No 2, pp. 132-141, 2008.
- [18] G. Mrugalski, J. Rajska, and J. Tyszer, "Ring Generators — New Devices for Embedded Test Applications", IEEE Trans. CAD, vol. 23, no. 9, Sept. 2004, pp. 1306-1320.
- [19] G. Mrugalski, et al., "New Test Data Decompressor for Low Power Applications", Design Automation Conference 2007, pp. 539-544.
- [20] Janusz Rajska, et al., "Embedded deterministic test" IEEE Trans. on CAD, vol. 23, No. 5, pp. 776-792 (2004)
- [21] S. Remersaro, et al., "Scan Based Tests with Low Switching Activity", IEEE Design & Test of Computers, May-June 2007, pp. 268-275.
- [22] R. Sankaralingam, R. Oruganti, N. Toubia, "Static compaction technique to control scan vector power dissipation," Proc. VTS, pp. 35-40, 2000.
- [23] S. Wang and S. Gupta, "LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation," in Proc. Int. Test Conf., pp. 85-94, 1999.
- [24] L.-T. Wang et al., "UltraScan: Using Time-Division Demultiplexing-Multiplexing (TDDM/TDM) with VirtualScan for Test Cost Reduction", Proc. Int'l Test Conf., 2005, pp. 946-953.