Modeling Scan Chain Modifications For Scan-in Test Power Minimization*

Ozgur Sinanoglu and Alex Orailoglu

Computer Science and Engineering Department
University of California, San Diego
La Jolla, CA 92093

{ozgur, alex}@cs.ucsd.edu

ABSTRACT

Rapid and reliable test of SOCs necessitates upfront consideration of the test power issues. Special attention should be paid to scan-based cores as the test power problem is more severe due to excessive switching activity stemming from scan chain transitions during shift operations. We propose a scan chain modification methodology that transforms the stimuli to be inserted to the scan chain through logic gate insertion between scan cells, reducing scan chain transitions. We provide a mathematical analysis that helps model the impact of scan chain modifications on test stimuli transformations. Based on this analysis, we develop algorithms for transforming a set of test vectors into power-optimal test stimuli through cost-effective scan chain modifications. Even in the highly challenging case of fully specified test vectors, more than an order of magnitude reduction in scan-in power is attained by the proposed methodology, exceeding previous power reduction levels significantly.

1. INTRODUCTION

Testing of SOC cores in parallel helps reduce test application time which in turn improves time-to-market of the chip. Achieving such a parallelism though is a challenging task as exceeding certain power thresholds places the chip under test at risk of damage. Especially in scan-based cores, the test power problem is of significant concern, as during the shift cycles excessive switching activity occurs inside the scan cells. The rippling effect in a scan chain reflects into the circuit, resulting in a large number of unnecessary transitions at the circuit lines. Test power can be reduced by preventing the unnecessary rippling in the scan chain during the shift of the test data, consequently.

Numerous methodologies that aim at test power reduction in a scan-based environment have been proposed recently. The utilization of externally controlled gates [1, 2] has been shown to reduce test power drastically, albeit at the expense of performance degradation. Appropriate primary input assignment during the shift cycles [3, 4] helps reduce transition propagation from the scan chain to the circuit under test; however, the effectiveness of such techniques is limited, as typically circuits are controlled mostly by scan chains rather than primary inputs. Scan chain partitioning techniques [5, 6] have been proposed for test power reduction; the scan chain is decomposed into several partitions so as to have only one of the partitions active at a time, reducing the scan chain rippling. Test vector ordering and scan-latch ordering techniques [7], modification of test cube compaction [8] and test generation [9] procedures constitute a set of alternative techniques for reducing scan power dissipation. These techniques extract test power reductions yet at the expense of prolonged test application time [8, 9], performance degradation [1, 2], or layout constraint violations [7].

In traditional approaches, the stimulus inserted to the scan chain is one and the same as the test vector. Modification of the scan chain through logic gate insertion between the scan cells though shatters this equivalence. The modifications necessitate the insertion of a transformed stimulus which possibly differs from the actual test vector. As the scan chain transitions are strictly determined by the transitions embedded in the input stimulus (and not by the actual test vector to be applied to the circuit under test), scan chain modifications can be utilized to reduce the number of scan chain transitions. We have in the recent literature proposed scan chain modification techniques [10, 11, 12] which impose neither performance nor fault coverage degradation. In these techniques we have focused on scan chain modifications with block-contained impact on the test data transitions, resulting in techniques consisting of the decomposition of test data into blocks, with each block being handled individually. To ensure the block-contained impact of scan chain modifications, certain constraints have been imposed on the XOR gate insertions. Under these constraints, locally optimal scan chain modifications are performed; however, the overall test power reductions are far from the global optimum. Even though the techniques outlined ([10, 11, 12]) deliver significant power reductions in comparison to previous literature, the restriction to a subset of XOR gate insertion configurations significantly hampers the optimality in power reductions, failing to reap appreciable additional power reductions possible.

In the methodology we present in this paper, we explore the complete transformation space implementable by any possible XOR gate and inverter insertion, with no constraints imposed on the possible set of scan chain modifications whatsoever. Exploitation of any possible transformation leads to the identification of the globally optimal test power solution. Appropriately modeling the impact of scan chain modifications on stimulus transformation enables the implementation of the logic gate insertions that realize the globally optimal transformation. The resultant modifications of the scan chain yield typically more than an order of magnitude reduction in scan-in test power. Attaining reductions of this magnitude for *fully specified test vectors* sharply differentiates the proposed approach from previously proposed techniques wherein power reductions attained are strictly limited and correlated to the density of *don't cares*.

In the proposed methodology, gate delays are introduced on the scan path only, imposing no performance degradation. Furthermore, the fault (defect) coverage is also preserved as the utilization of bijective logic gates (inverters and XORs) guarantees the delivery of any test vector to the scan chain and the observation of any fault effect.

*The work of the first author has been supported through a graduate fellowship by IBM.

0-7803-8106-8/03 $17.00 Copyright 2003 IEEE

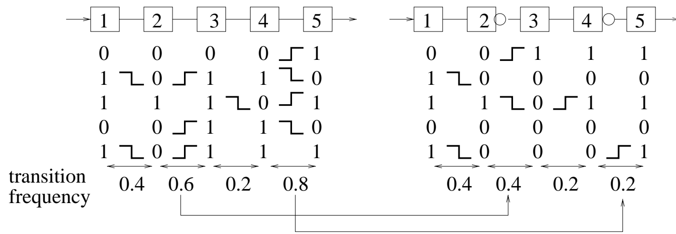

Figure 1: Impact of inverter insertion

In this paper, we model the impact of unfettered logic gate insertions on test stimuli transformations through a thorough mathematical modeling and analysis. We model the impact of scan chain modifications as band transformations on matrix structures and develop an algebra for the matrix evaluation and consequent placement of XOR/inverter gates on the scan chain. Through the exploitation of the modeled impact, we develop test power minimization algorithms that identify the *optimal* test stimuli transformation and implement them through minimal area overhead. Furthermore, we provide a power-area cooptimization algorithm which attains near-optimal test power reductions under certain design constraints.

This paper is structured as follows. In Section 2 we present the motivations underlying the proposed methodology which is followed up in Section 3 by the presentation of the matrix-based analysis we utilize to model the impact of XOR gate insertion. In section 4, we outline the algorithmic framework; the proposed algorithms for the identification of the optimal test stimulus transformation, the implementation of this transformation through minimal area overhead, and the incorporation of design constraints are outlined. Sections 5 and 6 present the experimental results and the conclusions, respectively.

2. MOTIVATION

The examination of the impact of logic gate insertion between scan cells on the consequent stimulus transformation raises two concrete questions. How can the transformation that maps the test vectors to the power optimal test stimuli be identified? Given the optimal test stimulus transformation, how should the scan chain be modified so as to implement this transformation cost-effectively?

The impact of inverter insertion on stimulus transformation is easier to model as inverters have local impact on the stimulus transitions. Insertion of an inverter between two scan cells necessitates the application of a stimulus which differs from the actual test vector in only the bits that are to pass through the inverter; the remaining stimulus bits equal the test vector bits in the corresponding positions. As the inversion of a string of bits happens to preserve the transitions between these bits, the only transition impacted by the inverter insertion is the one between the two bits only one of which is negated; inserting an inverter between two cells impacts solely the transition between the corresponding stimulus bits. Figure 1 illustrates the impact of inverter insertion on stimuli transitions; the original scan chain along with the test vectors is shown on the left part of the figure while the scan chain modified through inverter insertions along with the stimuli to be inserted is presented on the right hand side. The impact of inverter insertion on the transition frequency in the corresponding location is simply the negation of the transition frequency as illustrated in Figure 1.

A quick look at XOR gate insertion on a scan path reveals the challenges associated with the modeling of the impact of XOR gate insertion though. The example in Figure 2 illustrates the impact of the insertion of XOR gates on the scan path; the actual test vectors which happen to be the stimuli to be inserted to the unmodified scan chain are

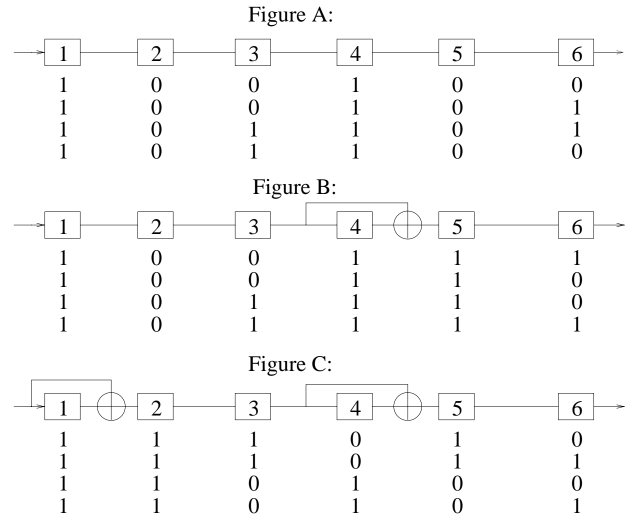

Figure 2: Scan chain modification

given in Figure 2.A, while Figures 2.B and 2.C illustrate the modified scan chains along with the stimuli to be inserted. It can be seen that the insertion of a single XOR gate (Figure 2.B) has an impact on various bit positions; the XOR gate eliminates all the transitions between the fourth[1] and the fifth bit positions whereas it introduces new transitions between the fifth and the sixth bit positions. The transitions between the first two stimuli bits can be eliminated by the insertion of another XOR gate between the first two scan cells (Figure 2.C); however, this modification cancels the benefit of the previous modification as transitions are introduced between the fourth and the fifth bit positions. To develop an algorithm that identifies the optimal scan chain modification, we need to understand and model the interference between various XOR gate insertions.

Successful modeling of the impact of XOR gates on the consequent stimulus transformation enables the decomposition of the problem into two parts: identification of the optimal stimulus transformation and the implementation of an appropriate scan chain modification that effects the transformation. As the test vector bits can be considered as individually transformed into new stimulus bits, the stimulus transformation operation can be represented by a binary matrix multiplication operation. The matrix whose rows denote the test vectors can be multiplied by the transformation matrix, forming the product matrix that denotes the transformed stimuli. The transformation matrix entries with a value of 1 in a column denote the actual test vector bits to be XORed in forming the transformed stimulus bit corresponding to the column. The transformation operation corresponding to the modified scan chains in Figures 2.B and 2.C are shown in the upper and the middle parts of Figure 3, respectively. Multiplication of the test vector matrix in Figure 2.A by the corresponding transformation matrix yields the matrix that denotes the stimuli to be inserted to the modified scan chains in Figure 2; the number of transitions is reduced from 14 down to 10 and 12, respectively. Actually, the optimal transformation matrix for the test vectors in Figure 2.A is given in the bottom part of Figure 3; the optimal transformation maps the actual test vectors to the stimuli set with the fewest possible number of transitions which is 4 in this example. The identification of the optimal transformation matrix poses another challenging issue due to the large size of this matrix; the number of stimulus transformation matrix entries equals

[1]The test vector bits with higher indices are shifted into the scan chain prior to the ones with lower indices. Similarly, lower numbered indices are assigned to the scan cells that are closer to the scan-in pin.

Transformation matrix in Fig. 2.B

$$\underset{\text{14 transitions}}{\begin{bmatrix}1&0&0&1&0&0\\1&0&0&1&0&1\\1&0&1&1&0&1\\1&0&1&1&0&0\end{bmatrix}}\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&1&1\\0&0&0&0&1&1\\0&0&0&0&0&1\end{bmatrix}=\underset{\text{10 transitions}}{\begin{bmatrix}1&0&0&1&1&1\\1&0&0&1&1&0\\1&0&1&1&1&0\\1&0&1&1&1&1\end{bmatrix}}$$

Transformation matrix in Fig. 2.C

$$\underset{\text{14 transitions}}{\begin{bmatrix}1&0&0&1&0&0\\1&0&0&1&0&1\\1&0&1&1&0&1\\1&0&1&1&0&0\end{bmatrix}}\begin{bmatrix}1&1&1&1&1&1\\0&1&1&1&1&1\\0&0&1&1&1&1\\0&0&0&1&0&1\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}=\underset{\text{12 transitions}}{\begin{bmatrix}1&1&1&0&1&0\\1&1&1&0&1&1\\1&1&0&1&0&0\\1&1&0&1&0&1\end{bmatrix}}$$

Optimal transformation matrix

$$\underset{\text{14 transitions}}{\begin{bmatrix}1&0&0&1&0&0\\1&0&0&1&0&1\\1&0&1&1&0&1\\1&0&1&1&0&0\end{bmatrix}}\begin{bmatrix}1&1&0&1&1&0\\0&1&1&0&1&1\\0&0&1&1&1&1\\0&0&0&1&1&1\\0&0&0&0&1&1\\0&0&0&0&0&1\end{bmatrix}=\underset{\text{4 transitions}}{\begin{bmatrix}1&1&0&0&0&1\\1&1&0&0&0&0\\1&1&1&1&1&1\\1&1&1&1&1&0\end{bmatrix}}$$

Figure 3: Stimuli transformation operation

the square of the number of scan cells. The typically large number of scan cells in practice necessitates a computationally effective strategy, consequently. Furthermore, the large size of the matrix imposes additional algorithmic challenges on the the placement of XOR/inverter gates on the scan chain to realize the optimal transformation.

The matrix equations given in Figure 3 can be reformulated so as to illustrate the mapping from the stimuli to be inserted to the actual test vectors; the consequent reformulation in Figure 4 somewhat hints at the relationship between the scan chain modifications and the stimulus transformations. In Figure 4, the mapping to the actual test vectors of the stimuli to be inserted to the modified scan chain is illustrated; the mapping happens by multiplying the stimuli by another transformation matrix which is actually the inverse of the transformation matrices given in Figure 3. The lefthand side of the equations in

Test vectors in Figure 2.A = Stimuli in Figure 2.B × (transformation matrix; discontinuity marked)

$$\begin{bmatrix}1&0&0&1&0&0\\1&0&0&1&0&1\\1&0&1&1&0&1\\1&0&1&1&0&0\end{bmatrix}=\begin{bmatrix}1&0&0&1&1&1\\1&0&0&1&1&0\\1&0&1&1&1&0\\1&0&1&1&1&1\end{bmatrix}\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\0&0&0&0&0&1\end{bmatrix}$$

Test vectors in Figure 2.A = Stimuli in Figure 2.C × (transformation matrix)

$$\begin{bmatrix}1&0&0&1&0&0\\1&0&0&1&0&1\\1&0&1&1&0&1\\1&0&1&1&0&0\end{bmatrix}=\begin{bmatrix}1&1&1&0&1&0\\1&1&1&0&1&1\\1&1&0&1&0&0\\1&1&0&1&0&1\end{bmatrix}\begin{bmatrix}1&1&0&0&0&0\\0&1&1&0&0&0\\0&0&1&1&1&0\\0&0&0&1&0&1\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}$$

Figure 4: Mapping of inserted stimuli to test vectors

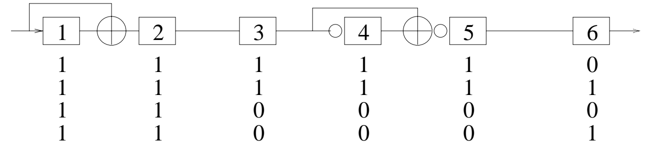

Figure 5: Utilization of inverters along with XORs

Figure 4 denotes the actual test vectors, while the righthand side, the mapping (matrix multiplication) operations. The transformation matrices in Figure 4 corresponding to the modified scan chains in Figures 2.B and 2.C are provided in the upper and the lower part of Figure 4, respectively. It is interesting to note that the transformation matrix corresponding to Figure 2.B embeds a discontinuity ($0 \rightarrow 1$) on the first off-diagonal right before the fifth column; the scan chain modification in Figure 2.B consists of an XOR gate insertion right before the fifth scan cell! The transformation matrix for the stimuli of Figure 2.C, on the other hand, includes three discontinuities. It is interesting to note that the ones on the first off-diagonal right before the second and the fifth columns account for the XOR gates inserted before the second and the fifth scan cells, respectively; however, the discontinuity on the second off-diagonal cannot be explained straightforwardly. An in-depth analysis should be carried out to establish the relationship between the scan chain modifications and the transformation matrix off-diagonals, consequently.

The utilization of inverters in exploiting high transition frequencies necessitates consideration of their impact as well in the computation of the optimal stimulus transformation; an XOR transformation that maps the actual test vectors to a set of stimuli with very high transition frequencies can still be useful, consequently. The example in Figure 5 illustrates the utilization of inverters to exploit the skewed transition frequencies created by the XOR gate insertion in Figure 2.C; insertion of inverters between the third and the fourth cells, and the fourth and the fifth cells eliminates all the transitions between the corresponding test stimuli bits, attaining the optimal number of transitions, 4.

3. IMPACT OF XOR GATE INSERTION

In this section, we present the modeling of the impact of XOR gate insertions on the consequent stimulus transformation implemented. Throughout this matrix-based analysis, we frequently refer to *matrix bands*, which denote the off-diagonals in the transformation matrices. We illustrate the strong connection between the matrix bands and the actual scan chain modifications. We thus establish the relationship between the XOR gate insertion and the transformation matrices, enabling the implementation of any possible stimulus transformation through the appropriate scan chain modification.

3.1 Test Stimulus Transformation

The impact of XOR gate insertion can be modeled as transformation functions. The *Stimulus Transformation* (ST) function maps the test vectors to the stimuli needed at the input of the scan chain; the *Stimulus Inverse Transformation* (SIT), on the other hand, maps the inserted stimuli to the data delivered into the scan cells, *i.e.*, the test vectors to be applied. Each of these transformations defines a set of single-bit mappings; the number of single-bit mappings equals the number of scan cells. Figure 6 illustrates these transformations for the scan chain modified through XOR gate insertion.

As the aforementioned transformation functions denote a set of single-bit mappings, they can be represented in matrix form. In a transformation matrix, a column corresponding to a bit position defines the

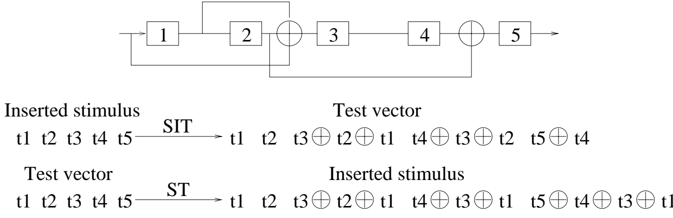

Figure 6: SIT and ST transformations

single-bit mapping for this bit; the leftmost column in the matrix corresponds to the bit associated with the leftmost scan cell. The entries in the matrix are binary; 1-entries in a column denote the bits to be XORed in forming the transformed bit. ST and SIT are square matrices with the number of rows (columns) equal to the number of scan cells. These matrices are upper triangular as no loops are formed in inserting XOR gates to the scan chain. Figure 7 illustrates ST and SIT in matrix forms for the example in Figure 6.

As the entries of these transformation matrices are binary, in matrix multiplication and summation operations, the XOR operation rather than an arithmetic summation applies in the case of the accumulation. Matrix multiplication and summation operations for two square matrices, $\mathbb{A}$ and $\mathbb{B}$, which are both $N X N$, can be defined as follows.

$$(\mathbb{A}\,\mathbb{B})_{i,j} \quad = \quad \bigoplus_k \mathbb{A}_{i,k}\;\mathbb{B}_{k,j} \tag{1}$$

$$(\mathbb{A} \oplus \mathbb{B})_{i,j} \quad = \quad \mathbb{A}_{i,j} \oplus \mathbb{B}_{i,j} \tag{2}$$

ST and SIT are inverse transformation functions; ST constitutes the mapping from a test vector to the corresponding stimulus to be inserted, while SIT, the inverse mapping. Consequently, the product of ST and SIT equals the identity matrix.

3.2 Relationship Between XOR Gate Insertion and Transformations

The ability to map any transformation function to the appropriate scan chain modification necessitates understanding the impact of XOR gate insertion in the scan chain. In this subsection, we analyze the relationship between XOR gate insertion and the resulting SIT implemented by the modified scan chain.

We first start with the impact of a single XOR gate insertion. The stimulus bits that are delivered into the scan cells without passing through the XOR gates remain intact. The stimulus bits that pass through the XOR gate however are transformed; as only a single XOR gate is inserted, the transformation functions for these bits are identical. Figure 8 illustrates a scan chain modified through the insertion of a three-input XOR gate between the third and the fourth scan cells. It can be seen from the corresponding SIT that the leftmost three bits remain intact; the transformation function for each of the rightmost three bits is the XOR of the bit with its two consecutive bits, resulting in a string of 1's between the fourth column and the rightmost column in

SIT:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

ST:

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 7: Transformations in matrix form

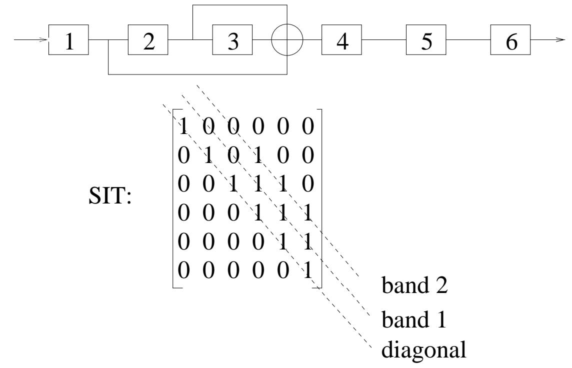

Figure 8: Impact of a single XOR gate insertion

the matrix on two off-diagonals. We therefore introduce *bands*, which are suitable for modeling the impact of XOR gate insertion; $B_{init}^{(d)}$ denotes a string of 1's between the $init^{th}$ and the rightmost columns, on the dth off-diagonal from the SIT matrix diagonal. The example SIT matrix consists of two bands, $B_4^{(1)}$ and $B_5^{(2)}$, in addition to the diagonal. In general a single n-input XOR gate insertion between the $s-1^{th}$ and s^{th} scan cells results in a transformation denoted by:

$$\mathbb{SIT} \quad = \quad \mathbb{I} \oplus \bigoplus_i B_s^{(d_i)} \tag{3}$$

wherein d_i denotes the number of scan cells between the XOR gate and the ith input tap and $\mathbb{I}$, the identity matrix.

To be able to map any upper triangular matrix to the appropriate scan chain modification that implements it, we introduce band lists, which denote the sum of a list of bands:

$$BL_{(l_1,l_2,..,l_n)}^{(d)} = \bigoplus_i B_{l_i}^{(d)} \tag{4}$$

The band list shown above denotes strings of 1's on the d^{th}off-diagonal, between the columns l_1 and l_2, l_3 and l_4 and so on; equivalently, if l_i exists in a band list, it creates a discontinuity ($0 \to 1$ or $1 \to 0$) on the band between the $l_i - 1^{th}$ and l_i^{th} columns. Any upper triangular matrix can hence be represented as the collection of various band lists; the example SIT in Figure 8, for instance, can be represented as $I \oplus BL_{(3)}^{(1)} \oplus BL_{(3,5)}^{(2)}$.

In the case of multiple XOR gate insertion, the configuration in which the XOR gates are inserted has distinct impact on the transformation. Specifically, two distinct cases need to be considered. In the first case, the XOR gates inserted overlap as in the example given in Figure 9; the input of the XOR gate between the third and the fourth scan cells is taken from between the other XOR gate and the second scan cell. The resulting SIT is given in the same figure; it can be seen that the SIT can be obtained by taking the sum of the individual SITs corresponding to each XOR gate. In the same figure, the band list representation of the SIT is also provided.

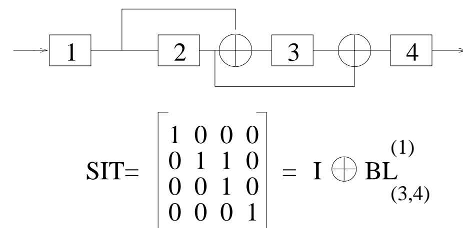

Figure 9: Overlapping modifications

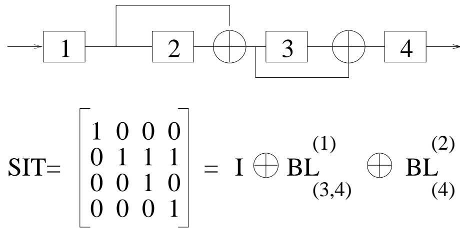

Figure 10: Nonoverlapping modifications

In the second XOR configuration, the XOR gates are non-overlapping as shown in Figure 10; the input of the XOR gate between the third and the fourth scan cells is taken from between the other XOR gate and the third scan cell this time. The sum of the two bands corresponding to the XOR gates fails to account for the SIT matrix provided in the same figure. Due to the interference of the two non-overlapping XOR gates, the SIT includes the product of the two bands in addition to the individual bands; in the case of two bands multiplied corresponding to non-overlapping XOR gates, the product of the two bands can be formulated, for $i_1 < i_2$, as:

$$B_{i_1}^{(d_1)}\, B_{i_2}^{(d_2)} = B_{i_2}^{(d_1+d_2)} \tag{5}$$

From the equation above, it can be seen that the product band might fall out of the boundaries of the matrix if $d_1 + d_2$ exceeds the matrix dimension; in this case, the product band can also be thought of as a band of all 0's.

In general, all possible non-overlapping XOR gate groups contribute to the stimulus inverse transformation, each with a band. Figure 11 illustrates a scan chain modified through four XOR gate insertions; in this figure, each XOR gate is denoted by the band it results in. In this example, $B_3^{(1)}$ and $B_4^{(1)}$ overlap; there is no other overlapping among the four XOR gates. The product terms, provided in the same figure, consist of two or three bands multiplied. The band list representation and the resulting SIT matrix is provided in the figure as well.

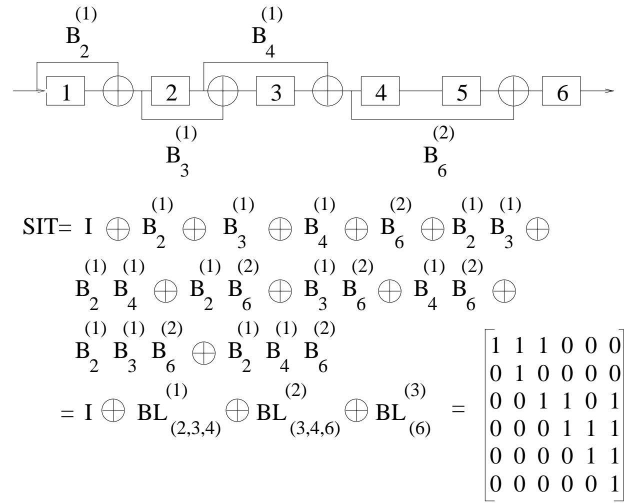

Figure 11: SIT computation

4. ALGORITHMIC FRAMEWORK

Having identified the impact of inverter and XOR gate insertion on test data transformations, in this section we present the proposed test power reduction algorithms we have developed based on the matrix-based modeling. We provide the analysis that identifies the impact of transformation matrix entries on test stimuli transitions. Based on this analysis, we first present the algorithm that efficiently identifies the optimal stimulus transformation in section 4.1. In section 4.2, we provide the algorithm that identifies the minimal area scan chain modification that implements the optimal transformation. Finally, the proposed power-area cooptimization algorithm is presented in section 4.3. In this algorithm, scan chain modifications and power-efficient test stimulus transformation identification are performed in an intertwined manner, enabling the identification of the best possible transformations attainable under any given set of design constraints.

4.1 Optimal Test Stimulus Transformation Identification

In this subsection, we present the proposed algorithm for identifying the optimal stimulus transformation, ST; the algorithm searches for the ST matrix that maps the actual test vectors into a set of transformed vectors (stimuli to be inserted to the scan chain) with maximized transition frequency skew.

The matrix representation of the test vector set, denoted by $\mathcal{T}V$, helps formulate the transformation operation; each test vector is denoted as a row in this matrix. The stimuli to be inserted to the scan chain, which can be represented as a matrix (SI) as well, can then be computed by multiplying $\mathcal{T}V$ and ST. Figure 12 illustrates the transformation operation; the test set consisting of three vectors is transformed by the ST matrix in Figure 7. It can be seen that all the transition frequencies are maximally skewed except for the leftmost one.

It should be noted that the stimulus transitions in distinct locations have distinct impact on the scan chain transitions. As the stimulus bits that are earlier shifted in pass through a larger number of scan cells, the transitions between these bits toggle a larger number of scan cells. Specifically, transitions between the ith and the $i+1^{th}$ stimulus bits toggle i scan cells. The weighted impact of the stimulus transitions should also be taken into account by a scan-in power reduction algorithm. In Figure 12, for instance, the only transition frequency which is not maximally skewed is the leftmost one; fortunately, the weight of this transition location is less than that of the other transition locations.

Developing a computationally feasible algorithm that identifies the optimal ST matrix necessitates an insight into the impact of matrix entries on the transitions between the transformed vector bits. Figure 13 illustrates all possible transformations for the three leftmost test vector bits; a decision tree is shown in this figure, wherein the nodes correspond to the transformed vector bits. An edge of the decision tree is denoted by the *condition function* (the condition that the edge is taken), and by the *transition function* (the condition that a transition occurs between the corresponding transformed bits), respectively. The condition function depends on the ST matrix entries whereas the transition function depends on the actual test vector bits. For instance, the leftmost test vector bit, t_1, can only be transformed into itself, whereas

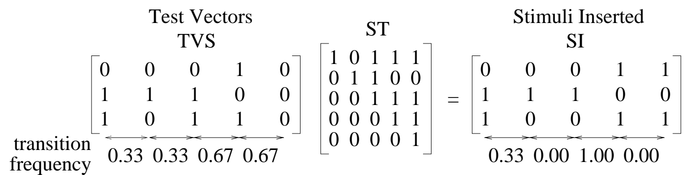

Figure 12: Transformed stimuli computation

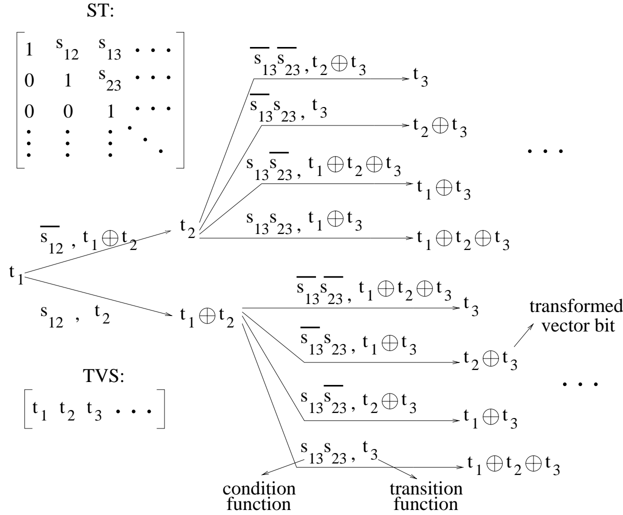

Figure 13: Impact of ST entries

two alternatives exist for t_2 depending on the value of s_{12}; if s_{12} is 0, t_2 remains intact, otherwise, it is transformed into $t_1 \oplus t_2$. In the first case, the transition function between the first two transformed bits equals $t_1 \oplus t_2$ whereas in the second case, it is simply t_2. It should be noted that between the second and the third transformed bits, each one of the transition functions on the upper four edges appear on one of the lower four edges as well. Whether t_2 remains intact or is transformed into $t_1 \oplus t_2$ has therefore no impact on selecting the optimal transition function between the second and the third transformed bits; regardless of what s_{12} is assigned to, s_{13} and s_{23} can be appropriately selected so as to pick the edge with the maximally skewed transition frequency. Each column of the optimal ST matrix can be computed in an independent manner, reducing the computational complexity of the problem significantly.

The test vector set, TV, can be multiplied by ST, with each entry of ST bound to a symbolic variable. The entries of the product matrix, SI, can be computed in terms of ST entries, enabling the denotation of the transitions between the transformed vector bits in terms of ST entries as well; the transitions between two consecutive SI columns can be computed by XORing the two columns bitwise, forming the corresponding *transition column*. The aforementioned independence of ST column computations enables an algorithm that searches the optimal assignment of an ST column so as to maximize the transition frequency skew corresponding to the transition column. Figure 14 illustrates the computation of SI and the transition columns for the example test vector set given in the same figure.

The algorithm consequently consists of a number of steps that equals the number of transition columns, or equivalently one less than the number of scan cells. In every step, an ST column is computed through solving two systems of equations which are formed based on the corresponding transition column; the first system of equations consists of setting the transition column entries to all 0's, resulting in the minimization of the transition frequency to 0.0. Similarly, the second system of equations aims at maximizing the transition frequency to 1.0; the transition column entries are set to all 1's. Among the two systems of equations, the preferable one is selected for assigning the ST column entries. In any step, the number of equations always equals the number of test vectors. However, the number of variables in the system depends on the transition column being handled; in the ith step (in handling the ith transition column), the variables in the system consist of the SIT symbolic entries in the $i+1^{th}$ column, resulting

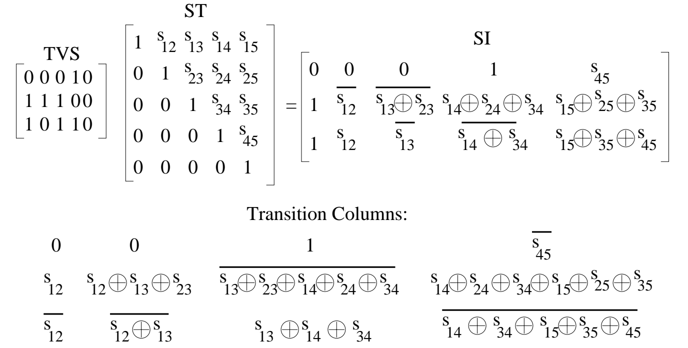

Figure 14: Computation of transition columns

in i variables. It is of course possible that neither one of the two systems of equations has a solution; this typically occurs when the transition columns with lower indices are being handled, resulting in more variables than the number of equations. In this case, ST entries are assigned so as to satisfy the maximal number of equations; the equations that cannot be satisfied contribute to the power cost. Among the two systems of equations, the one with the larger number of equations satisfied is selected for assigning the ST column entries.

The transition columns with indices larger than the number of test vectors typically have no contribution to test power as the corresponding system of equations has possibly multiple solutions, resulting in no unsatisfied equations. The example in Figure 15 illustrates the experimental data corresponding to the circuit $s1423$, with 91 scan cells and 63 test vectors; the y-axis denotes the number of equations that remain unsatisfied while the x-axis denotes the transition column index. From the graph, one can note that the transition columns with a higher index than the number of test vectors typically have no contribution to test power as the corresponding system of equations is solvable. The fact that the weight of the transition columns with lower indices (which are typically unsolvable as seen from the graph) are significantly smaller than those of the ones with higher indices increases the benefit of utilizing the proposed methodology.

The solution of a system of equations is computed by the Gaussian elimination technique [13]; the ST entries are thus assigned to appropriate values that maximize the transition frequency skew. It is possible that the system of equations is unsolvable in which case the Gaussian elimination technique terminates unsuccessfully; in such cases, we utilize a heuristic to maximize the number of equations that are satisfied. This heuristic, the details of which are omitted due to space constraints, is based on linear dependency identification among the equations in the system.

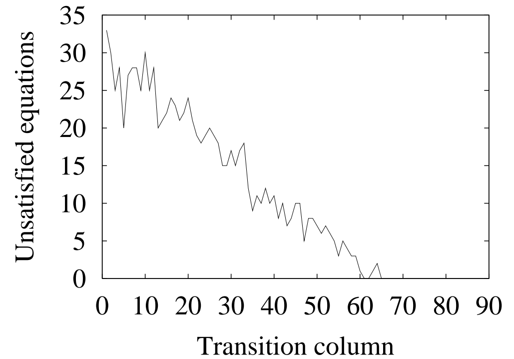

Figure 15: Solvability of equations for $s1423$

$$0 = 0 \quad s_{12} = 0 \quad \overline{s_{12}} = 0 \qquad \text{or} \qquad 0 = 1 \quad s_{12} = 1 \quad \overline{s_{12}} = 1$$

Figure 16: System of equations

Figure 16 illustrates the systems of equations to be solved for s_{12} for the leftmost transition column in Figure 14; setting s_{12} to 0 in the system of equations on the left hand side of the figure results in satisfying 2 out of 3 equations, failing to maximize the transition frequency skew and resulting in a transition frequency of 0.33. In the second transition column entries, s_{12} is substituted by 0 as it is already assigned to this value. Assigning both s_{13} and s_{12} to 1 results in minimizing the transition frequency to 0.0 in the second transition column. Similarly, the third and the fourth transition columns can be handled; s_{14}, s_{34}, s_{15}, s_{35} and s_{45} are assigned to 1 whereas s_{24} and s_{25} are set to 0, maximizing and minimizing the transition frequency in the third and the fourth transition columns, respectively.

4.2 Minimal Area Implementation of the Optimal Transformation

The SIT matrix based on which the scan chain is to be modified can be straightforwardly computed by inverting the ST matrix computed through the algorithm in the previous subsection. In this subsection, we not only illustrate how a scan chain is modified based on the SIT matrix but furthermore provide an algorithm for implementing it with minimal area overhead.

In Section 3.2, we have shown that overlapping and non-overlapping XOR configurations have distinct impact as the product term is additionally introduced in the latter case. We exploit this degree of freedom in implementing the SIT matrix by selecting the configuration (overlapping or nonoverlapping) that leads to the minimal area overhead.

The algorithm processes a single SIT band in each step; the bands closer to the diagonal are handled earlier as the product of a number of bands consists of bands with higher indices than the bands that are multiplied as can be observed in Equation 5 in Section 3.2. Subsequent to each step, the SIT matrix implemented so far by the XOR insertions is computed so as to account for the product bands as well. The band that the algorithm is to implement in the ith step therefore consists of the difference of the ith SIT band and the ith band of the matrix that has been implemented so far by the XOR gate insertions; the difference operation is simply an XOR operation of the band lists. In every step, the XOR gates that help implement the band being handled in the step are inserted into the scan chain. Whenever a decision is to be made regarding whether an XOR tap is to be configured as overlapping or nonoverlapping, a decision variable is bound to this XOR configuration; a 0 value for this variable denotes an overlapping configuration whereas a value of 1 denotes a nonoverlapping configuration. The bands implemented by the XOR insertions therefore depend on the decision variables; as the difference bands to be implemented depend on the decision variables as well, conditional XOR taps might occur, with the condition being a function of the decision variables. Subsequent to handling all the bands, the decision variables are assigned so as to minimize the number of XOR taps; the maximal number of conditional XOR taps are eliminated by appropriately setting the decision variables, hence deciding judiciously whether the corresponding configurations will be overlapping or nonoverlapping.

The example in Figure 17 illustrates this algorithm for the SIT matrix in Figure 7. In the first step, the first band of the SIT matrix is implemented by inserting a single XOR gate. In the second step, the second SIT band is implemented by inserting two XOR taps; the de-

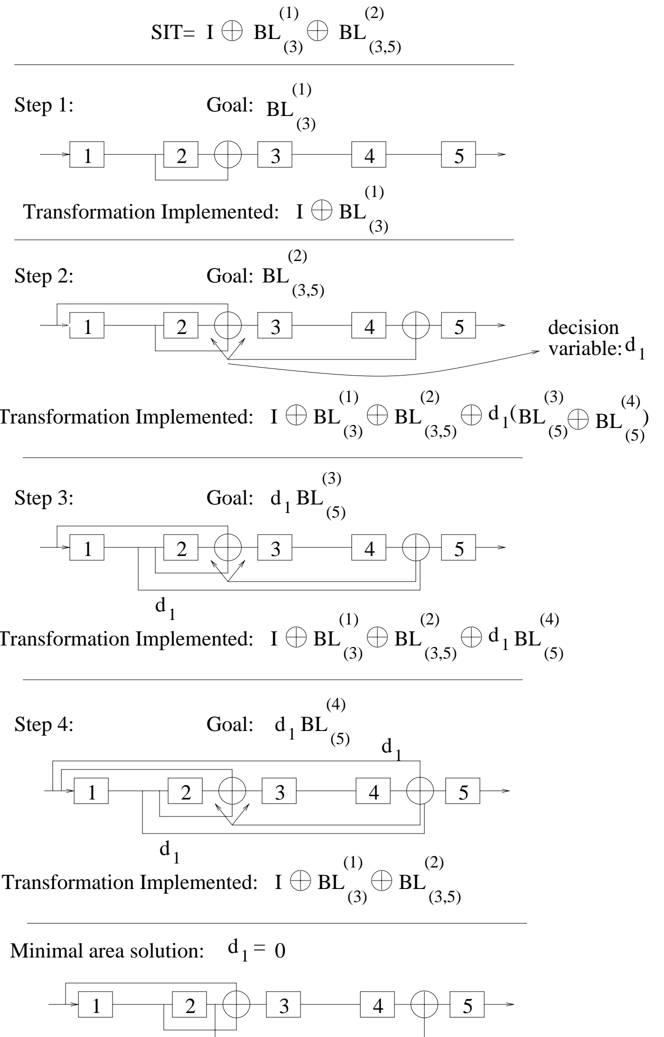

Figure 17: Minimal area implementation

cision regarding whether the rightmost XOR tap is to be overlapping with the other XOR taps is denoted by d_1. The transformation implemented by the modified scan chain at the end of step 2 consists of the third and fourth bands as a function of d_1. As the third band of SIT consists of all 0's, the difference band to be implemented in the third step is a function of d_1 as well; the XOR tap inserted in this step is a conditional one, with the condition being d_1 itself. At the end of the fourth step, it can be seen that two conditional XOR taps exist, with both conditions being the same. The decision variable, d_1, is assigned to 0 so as to eliminate both XOR taps; the modification in the second step is thus made overlapping, minimizing the area overhead. The minimal area implementation is provided at the bottom of the figure.

4.3 Power-Area Cooptimization

The algorithms explained in the previous two subsections identify the optimal transformation and implement it with minimal possible area; however, near-optimal power solutions might exist which can be implemented at significantly reduced area cost. In this section, we present a power-area cooptimization algorithm that computes power-wise sub-optimal transformations and performs scan chain modifications in an intertwined manner.

ST:

$$\begin{bmatrix} 1 & s_{12} & s_{13} & s_{14} & s_{15} \\ 0 & 1 & s_{23} & s_{24} & s_{25} \\ 0 & 0 & 1 & s_{34} & s_{35} \\ 0 & 0 & 0 & 1 & s_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

SIT:

$$\begin{bmatrix} 1 & s_{12} & s_{13}\oplus s_{12}s_{23} & s_{14}\oplus s_{13}s_{34}\oplus s_{12}(s_{24}\oplus s_{23}s_{34}) & (s_{15}\oplus s_{14}s_{45}\oplus s_{13}(s_{35}\oplus s_{45}s_{34}) \oplus s_{12}(s_{25}\oplus s_{24}s_{45}\oplus s_{23}(s_{35}\oplus s_{34}s_{45}))) \\ 0 & 1 & s_{23} & s_{24}\oplus s_{23}s_{34} & s_{25}\oplus s_{24}s_{45}\oplus s_{23}(s_{35}\oplus s_{34}s_{45}) \\ 0 & 0 & 1 & s_{34} & s_{35}\oplus s_{34}s_{45} \\ 0 & 0 & 0 & 1 & s_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

TVS

$$\begin{bmatrix} 1\,1\,0\,0\,1 \\ 0\,0\,1\,0\,0 \\ 1\,0\,1\,0\,1 \\ 1\,0\,1\,1\,0 \end{bmatrix} \longrightarrow \text{SI}: \begin{bmatrix} 1 & \overline{s_{12}} & s_{13}\oplus s_{23} & s_{14}\oplus s_{24} & \overline{s_{15}\oplus s_{25}} \\ 0 & 0 & 1 & s_{34} & s_{35} \\ 1 & s_{12} & \overline{s_{13}} & s_{14}\oplus s_{34} & \overline{s_{15}\oplus s_{35}} \\ 1 & s_{12} & \overline{s_{13}} & \overline{s_{14}\oplus s_{34}} & s_{15}\oplus s_{35}\oplus s_{45} \end{bmatrix}$$

Transition Columns:

$$\begin{matrix} s_{12} & \overline{s_{12}\oplus s_{13}\oplus s_{23}} & s_{13}\oplus s_{23}\oplus s_{14}\oplus s_{24} & \overline{s_{14}\oplus s_{24}\oplus s_{15}\oplus s_{25}} \\ 0 & 1 & \overline{s_{34}} & s_{34}\oplus s_{35} \\ \overline{s_{12}} & \overline{s_{12}\oplus s_{13}} & \overline{s_{13}\oplus s_{14}\oplus s_{34}} & \overline{s_{14}\oplus s_{34}\oplus s_{15}\oplus s_{35}} \\ \overline{s_{12}} & \overline{s_{12}\oplus s_{13}} & s_{13}\oplus s_{14}\oplus s_{34} & \overline{s_{14}\oplus s_{34}\oplus s_{15}\oplus s_{35}\oplus s_{45}} \end{matrix}$$

Figure 18: Cooptimization algorithm initialization

As the optimal power transformation and the minimal area implementation operations are performed based on the entries of ST and SIT matrices, respectively, an intertwined algorithm necessitates the representation of the symbolic entries of one matrix in terms of the symbolic entries of the other one. The cooptimization algorithm therefore initially computes straightforwardly the symbolic SIT entries in terms of symbolic ST entries as the two matrices are the inverses of each other. Figure 18 illustrates the ST and SIT matrices for an example scan chain of five scan cells; the computation of the transition columns based on an example test vector set, $\mathcal{TV}$, is illustrated as well.

The parameters of this algorithm are the *power tolerance* and *area constraint* per column; the power tolerance denotes the deviation from the minimal power for the corresponding transition column while the area constraint corresponds to the maximal number of XOR gates that can be expended in a step. In every step, the algorithm computes the power cost of any possible XOR gate insertions, subject to the constraint that the number of XOR gates cannot exceed the area constraint. The power costs are computed by taking into account the weight of the transitions as well; the power cost associated with the ith transition column equals the number of equations that remains unsatisfied multiplied by i. The area cost consists simply of the number of XOR gates expended. If a transformation with an area cost less than the constraint leads to a power cost within the power tolerance window, it is implemented; otherwise, the minimal power transformation that can be implemented within the area constraint is selected.

In a manner similar to the optimal transformation identification algorithm, the cooptimization algorithm proceeds by handling a single transition column in each step. In addition, the corresponding column of the SIT matrix is computed as well, enabling the power-area tradeoff exploration. The ST matrix entries are assigned so as to attain a power cost within the tolerance window and also to have an SIT column that necessitates fewer XOR gates than the maximal number allowed. Throughout the course of the algorithm, XOR gates are inserted in an overlapping manner[2] consistently; although this is a pessimistic approach area-wise, it significantly enhances computational efficiency. Upon the termination of the algorithm, the algorithm in the previous subsection is executed on the fully computed SIT matrix so as to implement the sub-optimal power transformation with minimal area overhead.

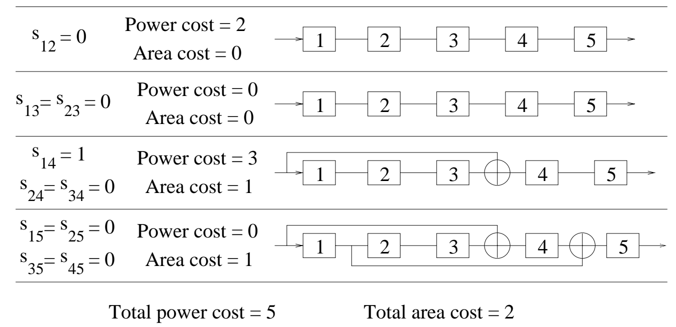

Figure 19: Execution of the cooptimization algorithm

The example in Figure 19 illustrates the step-by-step execution of the cooptimization algorithm for the example test set in Figure 18 with power tolerance and area constraint set to 3 and 1, respectively. In the first step, s_{12} is assigned to 0, resulting in a power cost of 2 transitions and no area cost; in the first transition column, two equations remain unsatisfied, resulting in a power cost of 2, while no discontinuity is introduced on the first band of SIT, leading to no area cost. In the second step, both s_{13} and s_{23} are assigned to 0; as all the equations can be satisfied in the second transition column with no discontinuity on the first and the second bands of SIT, both power and area costs equal 0. In the third step, attaining a power cost within the power tolerance with no area cost is infeasible; s_{14} is assigned to 1 whereas both s_{34} and s_{24} are assigned to 0. The power cost equals 3 in this step, as only one equation on the transition column cannot be satisfied, while the area cost is 1, as a discontinuity is introduced on the third SIT band which necessitates the insertion of an XOR gate. In the final step, similar to the third step, attaining a power cost within the power tolerance with no area cost is infeasible; appropriate assignments are made so as to reduce the power cost to 0. However, as another discontinuity is introduced on the third SIT band, the area cost in this step consists of a single XOR gate.

Execution of the proposed cooptimization algorithm with the appropriate values for the power tolerance and area constraint parameters helps identify the best possible test data transformation power-wise at reasonable area cost. In the case of multiple transformations attaining approximately equal power reductions but at distinct area costs, the algorithm selects the most cost-effective transformation area-wise. A single power-wise effective transformation that beats the others by a significant margin, on the other hand, is selected by the algorithm as long as it is affordable area-wise. The proposed cooptimization is able to adjust itself in exploiting the tradeoff between power and area, consequently.

Similar to the area constraints, layout constraints can also be incorporated to the cooptimization problem. A slight modification to the proposed cooptimization algorithm therefore needs to impose additional constraints on the length of the XOR taps, restricting certain transformations. Near-optimal power reductions can be thus attained with no layout constraint violations.

[2]Alternatively, always nonoverlapping XOR insertion can be employed.

Circuit	Scan cells	Test vectors
s713	54	42
s953	45	100
s1423	91	63
s5378	214	145
s9234	247	217
s13207	700	248
s15850	611	155
s35932	1763	22
s38417	1664	883
s38584	1464	681

Table 1: Benchmark circuits

5. EXPERIMENTAL RESULTS

The proposed test power reduction scheme has been applied to several fully-scanned circuits in ISCAS89 [14]. The test vectors that are used to compute the test power reductions achieved by the proposed methodology are generated by ATALANTA [15]. Table 1 presents general information about the benchmarks used to evaluate our scheme; the number of scan cells and test vectors are provided.

Table 2 demonstrates the optimal test power reductions attained by the proposed methodology; the reductions in the total number of transitions in the scan chain during the shift of the test stimuli are provided.[3] The second column of the table shows the ratio of the number of scan cells to the number of test vectors while the third column, the optimal scan-in power reduction attained by the proposed methodology. It can be seen that the power reductions attained are strongly correlated with the scan cell to test vector ratio; in Section 4.1, we have shown that the number of equations to be solved equals the number of test vectors. As the scan cell to test vector ratio increases, we can expect enhanced test power results, consequently. Two extreme cases consist of the circuits $s953$ and $s35932$; in $s953$, the number of test vectors exceed the number of scan cells, resulting in a comparatively limited test power reduction, whereas in $s35932$, the number of scan cells is significantly larger than the number of test vectors, resulting in several orders of magnitude reduction in test power. The fact that the test power reduction attained by the proposed methodology is correlated with the scan cell to test vector ratio is a promising result, as for typical circuits in industry, utilization of effective test set compaction tools pushes these ratios in favor of the proposed technique. The experimental results confirm this observation as more than an order of magnitude reduction in test power is attained for larger benchmark circuits. It is important to note that the proposed methodology is applied to fully specified test vectors; the methodology we propose attains significant power reductions with no exploitation of the *don't care* bits in test data whatsoever.

Table 3 demonstrates the results of the proposed power-area cooptimization algorithm; power reduction and area cost results are presented for various values of the power tolerance and area constraint parameters. In our experiments, we adjust the power tolerance parameter as a function of the minimum possible test power, denoted by P_{min}, and the number of scan cells, denoted by SC. It can be seen from the table that significant reductions in test power can still be achieved yet at reasonable area cost. The proposed cooptimization algorithm offers various solutions to the designer, enabling the exploitation of the tradeoff.

[3]Test power reductions are computed by comparing the number of scan chain transitions of the proposed scheme against those incurred during a traditional scan test; [8] shows that the number of scan chain transitions and the test power dissipation are strongly correlated.

Circuit	SC/TV	Optimal power reduction (%)
s713	1.29	80.7
s953	0.45	33.6
s1423	1.44	85.5
s5378	1.48	85.7
s9234	1.14	73.9
s13207	1.52	85.9
s15850	3.94	97.9
s35932	80.14	99.9
s38417	1.88	90.9
s38584	2.15	93.1

Table 2: Optimal scan-in power reductions attained by the proposed methodology

Table 4 presents the scan-in power reduction comparisons against the previously proposed ([11, 10]) scan chain modification schemes. In [11], the scan chain is modified through inverter insertions only, whereas in [10] XOR gates are utilized as well; however, in [10], the scan chain modifications are subject to the constraint that whenever an XOR gate is inserted, another one is to be inserted in the consecutive location, simplifying the test data transition frequency analysis. The experimental results confirm the benefit of eliminating this constraint on the XOR gate insertions; exploration of any possible scan chain modification boosts the test power reductions tremendously.

Table 5 provides the comparison of the proposed scheme against the previously published ([7, 6]) scan chain re-routing techniques.[4] The proposed methodology attains significantly better results compared to the other approaches for all the circuits except for $s953$. It is important to note that the scan chain re-routing techniques in [7, 6], whose applicability may sometimes be limited due to the stringent layout constraints, constitute orthogonal approaches that can be applied in conjunction with the methodology we propose with no diminution in the benefits offered by either methodology.

6. CONCLUSION

To shorten the SOC test application process, parallelism among core tests should be achieved. Testing of SOCs that is both rapid and reliable necessitates however that test power reductions be attained, as exceeding certain power thresholds may damage the chip under test. In this paper, we address the test power issues in scan-based environments where this problem is more acute due to the scan chain transitions during the shift of test data. The methodology we propose aims at reducing scan chain transitions during shift cycles stemming from the insertion of test stimuli, significantly decreasing the circuit switching that these transitions reflect into otherwise.

The technique we present is based on scan chain modifications to enable the insertion of a test stimulus with reduced transitions embedded. While both the actual test vector delivery into the scan cells and the observation of any fault effects are guaranteed, the reduced rippling that occurs in the modified scan chain during shift cycles helps mitigate the test power dissipation problem in scan-based environments. The types of modifications utilized consist of inverter and XOR gate insertion; no performance penalty is imposed as the modifications are performed on the scan path, keeping intact the functional operation of the circuit timing-wise.

Based on a sophisticated mathematical analysis that has helped us model the relationship between the scan chain modification and the

[4]For comparison purposes, we have re-implemented the techniques in [7, 6].

Circuit	Power tolerance, Area constraint							
	$0.5P_{min}/SC$, 2 XORs		P_{min}/SC, 2 XORs		$0.5P_{min}/SC$, 3 XORs		P_{min}/SC, 3 XORs	
	Power red.(%)	Area cost(%)	Power red.(%)	Area cost(%)	Power red.(%)	Area cost(%)	Power red.(%)	Area cost(%)
s713	68.3	15.1	60.7	11.8	74.4	18.3	70.2	16.2
s953	1.2	11.0	-34.7	8.7	18.4	13.7	5.7	11.4
s1423	74.4	12.9	69.2	10.1	80.2	16.1	75.8	13.4
s5378	76.2	10.1	72.3	7.9	82.1	13.2	80.5	11.2
s9234	61.3	6.8	46.4	5.9	68.7	10.1	64.7	8.4
s13207	74.9	11.4	71.1	8.8	80.8	14.7	77.6	12.2
s15850	95.7	8.4	94.9	7.1	96.3	13.8	96.1	11.9
s35932	99.9	10.8	99.9	8.3	99.9	14.2	99.9	11.5
s38417	84.8	9.6	80.4	7.2	88.6	13.5	85.3	12.1
s38584	88.5	8.6	81.6	6.9	91.7	12.4	89.7	10.3

Table 3: Results of the proposed cooptimization algorithm

consequent test data transformations, we have implemented algorithms not only for identifying the optimal test data transformation but futhermore for the minimal area implementation of the optimal transformation. Moreover, we have developed a power-area cooptimization algorithm that identifies the near-optimal test data transformation in terms of power cost at significantly reduced area cost, ensuring the applicability of the proposed scheme in practice.

To demonstrate the efficacy of the proposed approach, we have applied it on several ISCAS89 benchmark circuits. The experimental results indicate the benefits of being able to explore arbitrary combinations of inverter and XOR gate insertion configuration in scan chain modification. Even in the case of fully specified test vectors, typically more than an order of magnitude reduction in test power is attained, favoring the proposed methodology over the previously proposed techniques.

7. REFERENCES

[1] H. J. Wunderlich and S. Gerstendorfer, “Minimized power consumption for scan based BIST”, in *ITC*, pp. 85–94, 1999.

[2] R. Sankaralingam and N.A. Touba, “Inserting Test Points to Control Peak Power During Scan Testing”, in *DFT*, pp. 138–146, 2002.

[3] T. Huang and K. Lee, “An input control technique for power reduction in scan circuits during test application”, in *ATS*, pp. 315–320, 1999.

[4] N. Nicolici, B. M. Al-Hashimi and A. C. Williams, “Minimisation of power dissipation during test application in full-scan sequential circuits using primary input freezing”, *IEEE TCOMP*, vol. 47, n. 2, pp. 256–262, 1998.

[5] L. Whetsel, “Adapting scan architectures for low power operation”, in *ITC*, pp. 863–872, 2000.

[6] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch and H.J. Wunderlich, “A modified clock scheme for a low power BIST test pattern generator”, in *VTS*, pp. 306–311, 2001.

[7] V. Dabholkar, S. Chakravarty, I. Pomeranz and S. M. Reddy, “Techniques for minimizing power dissipation in scan and combinational circuits during test application”, *IEEE TCAD*, vol. 17, n. 12, pp. 1325–1333, 1998.

[8] R. Sankaralingam, R. R. Oruganti and N. A. Touba, “Adapting scan architectures for low power operation”, in *VTS*, pp. 35–40, 2000.

[9] R. Sankaralingam, B. Pouya and N.A. Touba, “Reducing Power Dissipation During Test Using Scan Chain Disable”, in *VTS*, pp. 319–324, 2001.

[10] O. Sinanoglu, I. Bayraktaroglu and A. Orailoglu, “Test Power Reduction Through Minimization of Scan Chain Transitions”, in *VTS*, pp. 166–171, 2002.

[11] O. Sinanoglu, I. Bayraktaroglu and A. Orailoglu, “Scan Power Reduction Through Test Data Transition Frequency Analysis”, in *ITC*, pp. 844–850, 2002.

[12] O. Sinanoglu, I. Bayraktaroglu and A. Orailoglu, “Reducing Average and Peak Test Power Through Scan Chain Modification”, *JETTA*, 2003.

[13] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, 1973.

[14] F. Brglez, D. Bryan and K. Kozminski, “Combinational Profiles of Sequential Benchmark Circuits”, *IEEE ISCAS*, vol. 14, n. 2, pp. 1929–1934, May 1989.

[15] H. K. Lee and D. S. Ha, *On the Generation of Test Patterns for Combinational Circuits*, Technical Report 12-93, Department of Electrical Eng., Virginia Polytechnic Institute and State University.

Circuit	Inv. only [11]	Consecutive X(N)ORs [10]	Proposed
s713	12.2	16.4	80.7
s953	9.2	11.8	33.6
s1423	7.5	11.3	85.5
s5378	8.6	11.4	85.7
s9234	5.1	8.0	73.9
s13207	7.3	11.0	85.9
s15850	5.8	8.3	97.9
s35932	8.0	12.2	99.9
s38417	8.7	11.3	90.9
s38584	10.1	13.9	93.1

Table 4: Scan-in power reduction (%) comparisons against previous scan chain modification schemes

Circuit	Scan cell reordering [7]	Scan chain partitioning [6]	Proposed
s713	22.3	42.7	80.7
s953	15.4	44.8	33.6
s1423	17.9	49.1	85.5
s5378	19.7	42.5	85.7
s9234	13.6	38.9	73.9
s13207	14.1	40.9	85.9
s15850	17.5	49.3	97.9
s35932	15.8	44.7	99.9
s38417	18.2	51.2	90.9
s38584	16.0	48.4	93.1

Table 5: Scan-in power reduction (%) comparisons against scan chain re-routing schemes