Partial BIST Insertion to Eliminate Data Correlation

Abstract—

A new partial BIST insertion approach based on eliminating data correlation to improve pseudo-random testability is presented. Data correlation causes the circuit to be in a subset of the states more or less frequently, which leads to low fault coverage in pseudo-random test. One important cause of correlation is reconvergent fanout. Incorporating BIST test flip-flops into reconvergent paths will break correlation, however, breaking all reconvergent fanout is unnecessary since some reconvergent fanout result in negligible correlation. We introduce a metric to determine the degree of correlation caused by a set of reconvergent fanout paths. We use this metric to identify problematic reconvergent fanout which must be broken through partial BIST insertion. We provide an algorithm to break high correlation reconvergent paths. Our algorithm provides high fault coverage while selecting fewer BIST flip-flops than required using loop breaking techniques. Experimental results produced using our algorithm rank on average among the top 11.6% of all possible solutions with the same number of flip-flops.

I. Introduction

Pseudo-random Built-In Self-Test (BIST) is an important testing technique which enables at-speed and on-site testing while requiring neither automatic test generation nor expensive test equipment. In general, pseudo-random test patterns are generated by an LFSR and circuit responses are compressed by a signature analysis register [1]. Maximum fault coverage can be achieved at significant area and performance overhead cost by configuring all flip-flops as test registers. At the other extreme, very low overhead can be achieved with a fault coverage penalty by inserting test registers only at primary inputs and outputs. Partial BIST insertion enable exploration of the tradeoff between fault coverage and overhead by configuring only a subset of flip-flops as a test register.

The goal of partial BIST insertion is similar to that of partial scan insertion which has been solved by many methods. The goal is to minimize hardware overhead while improving fault coverage as much as possible. Many papers in partial scan break sequential loops since size of loops impacts test application time exponentially [2], [3], [4], [5], [6]. In [7], Stroele and Wunderlich present an algorithm to break all sequential loops for pseudo-random test with minimal hardware overhead using a branch-and-bound algorithm to select flip-flops.

Data correlation and its effects on pseudo-random testability has been investigated by several authors. Papachristou *et al* [8] show that register adjacency can cause bit-level correlation in a circular BIST architecture. Register adjacency is a specific case of reconvergent fanout in RTL circuits. In [9], reconvergent fanout is removed during high-level synthesis to reduce the level of pseudo-random data correlation and reduce to number of conflicts encountered during the ATG process.

In this paper, we present a method to select partial BIST flip-flops which eliminates correlation to improve test quality. We present a method to characterize reconvergent paths based on their impact on fault coverage. Our algorithm selects flip-flops to break only reconvergent paths which have significant effect on correlation and fault coverage. Our approach reduces area overhead compared to loop breaking approaches by limiting test insertion only to important reconvergent paths.

This paper is organized as following. Section 2 motivates the connection between reconvergent fanout and pseudo-random fault coverage. Sections 3 and 4 introduce a metric for correlation along a circuit path and describe how that metric is used to direct the BIST insertion process. Sections 5 and 6 provide the algorithm and the results. Section 7 presents conclusions and future work.

II. Motivation

In this work, test flip-flop insertion is performed to break reconvergent fanout paths and thereby reduce data correlation. Data correlation is targeted because it affects data entropy which is known to impact fault coverage [10]. A significant source of data correlation is *matched reconvergent fanout*, which exists when several reconvergent paths have the same sequential depth. Figure 2a shows *mismatched* reconvergent fanout which will *not* cause correlation because of two paths from B to Z with length of 2 and 3 respectively. There is no correlation between the inputs of the reconvergent gate because the values of lines X and Y in time frame n depend on the values of B in two different time frames, $n-1$ and $n-2$. Figure 2b depicts matched reconvergent fanout paths which are the same sequential depth. If we assume that the value of line B is random, then lines X and Y will have the value '00' approximately 50% of the time. This correlation will increase test application time for the detection of faults in gate Z which are not detected by a '00' input pattern. The correlation can be eliminated by configuring flip-flop 1 as a CBILBO register, as shown in Figure 2c, so that the value of line X is not dependent on the value of line B.

III. Correlation Metrics

Reconvergent fanout is so common in sequential circuits that it is too expensive to break all such fanout. Not all reconvergent fanout needs to be broken since some of reconvergent fanout has negligible effect on correlation. We have developed a metric which indicates the impact of reconvergent fanout on correlation.

A. Serial correlation

In order to evaluate the effect of reconvergent fanout on correlation, we need some metrics to measure how strongly the output depends on the input. We define *serial correlation* as the correlation between an input and output of a single combinational block. The term "*serial*" is used to distinguish from correlation between parallel inputs of a combinational block.

Given the truth table of the combinational block with n inputs, we know exactly the distribution of all combinations of each input signal A and output signal Z, as shown in the following table. The value p indicates the total number of input values in which $A=0$ and $Z=0$, and the q value indicates the total number of input values in which $A=1$ and $Z=0$.

One Input A	Output Z	Distribution
0	0	$p/2^n$
0	1	$(2^{n-1}-p)/2^n$
1	0	$q/2^n$
1	1	$(2^{n-1}-q)/2^n$

Serial correlation between input A and output Z is defined as

$$corr(A,Z) = \sum_{(A,Z)} (A-\mu_A)(Z-\mu_Z)Prob.(A,Z) \qquad (1)$$

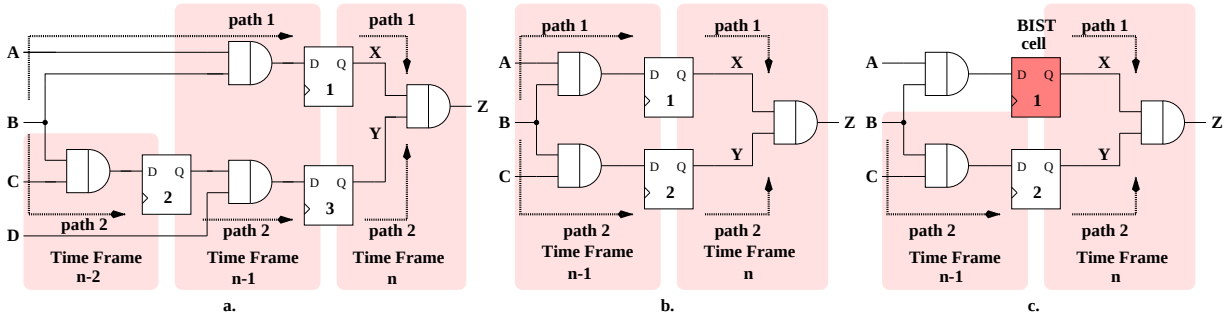
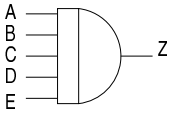


Fig. 1. (a) Mismatched reconvergent fanout (different sequential length). (b) Matched reconvergent fanout (same sequential length). (c) BIST insertion breaking reconvergent fanout.

where μ_A and μ_Z are expected values of A and Z. $Prob.(A,Z)$ is probability of combination of (A, Z) in truth table. In terms of p and q, the correlation between A and Z is

$$corr(A,Z) = \frac{p-q}{\sqrt{(2^n-p-q)(p+q)}} \tag{2}$$

High serial correlation between input A and output Z implies that output Z strongly depends on input A. However, serial correlation has some weakness in measuring the dependence of an output on an input. Serial correlation measures dependence between two data A and Z accurately only if Z is *not* strongly dependent on other inputs. The dependence of Z on other inputs can obscure the pairwise serial correlation metric. This effect can be seen in the can of a 5-input AND gate as shown in figure 2. The value of input A has a strong effect of



A Z	distribution	sign in serial correlation	represent dependence
0 0	16/32	+	Yes
0 1	0/32	-	No
1 0	15/32	-	No
1 1	1/32	+	Yes

Fig. 2. Distribution of one input A and output Z in truth table.

the value of the output because a '0' value on A implies a '0' value on Z. This is seen as a large distribution of the (0,0) combination which contributes positively to serial correlation. This positive correlation is negated by the large distribution of the (1,0) combination which is caused by the dependence of Z on the other inputs of the AND gate. Serial correlation of 5-input AND gate is 0.18 which does not accurately reflect the dependence of Z on the value of A.

B. Normalized Correlation

We introduce *normalized correlation* defined as serial correlation divided by maximum serial correlation.

$$\phi(A,Z) = \frac{corr(A,Z)}{corr_{max}(A,Z)} \tag{3}$$

The maximum serial correlation adjusts the distortion in representation of dependence in serial correlation. Intuitively, $corr_{max}(A,Z)$ for an n input function is the maximum correlation between A and Z over all possible n input functions for which the total number of minterms is constant. If we define l to be the number of maxterms, then $l = p+q$ and $corr_{max}(A,Z)$ can be defined as follows:

If $l \leq 2^{n-1}$, then correlation is maximum when $p = l, q = 0$, from equation (2), get

$$corr_{max}(A,Z) = \sqrt{\frac{l}{2^n-l}}, \quad l \leq 2^{n-1} \tag{4}$$

In same way, get

$$corr_{max}(A,Z) = \sqrt{\frac{2^n-l}{l}}, \quad l > 2^{n-1} \tag{5}$$

Then the normalized correlation is, in terms of p and q is,

$$\phi(A,Z) = \frac{p-q}{p+q}, \quad p+q \leq 2^{n-1} \tag{6}$$

$$\phi(A,Z) = \frac{p-q}{2^n-(p+q)}, \quad p+q > 2^{n-1} \tag{7}$$

Normalized correlation differs from serial correlation because it indicates that an input A can directly control the value of an output Z, independent of the value of other inputs.

IV. Partial BIST Insertion with Normalized Correlation

Normalized correlation along a path can be used to predict the impact of reconvergent fanout on correlation. For the purpose of partial BIST insertion, use the S-graph [4] to model the interconnections between flip-flops. The vertices of the S-graph represent flip-flops and primary inputs and outputs, and an edge between nodes v_i and v_j represents the existence of a combinational path from v_i to v_j. Reconvergent fanout paths involving flip-flops are represented by reconvergent paths in S-graph. The combinational logic of a sequential circuit can be partitioned into cones, where each cone is a 1-output combinational logic such that its inputs are either primary inputs or outputs of flip-flops and its output is either primary output or input of flip-flop. Every input and output of a cone corresponds to a node in the S-graph and every input-output pair of a cone corresponds to an edge in S-graph. The normalized correlation of each input-output pair of a cone is labeled on the corresponding edge of S-graph. If every edge on all reconvergent fanout paths has high normalized correlation, then such reconvergent fanout will have significant effect on the testability of the circuit. Figure 3a shows reconvergent fanout which needs to be broken due to the high correlation along each edge involved. Figure 3b and 3c are two examples of reconvergent fanout without significant impact on test quality. Dashed line indicated low normalized correlation.

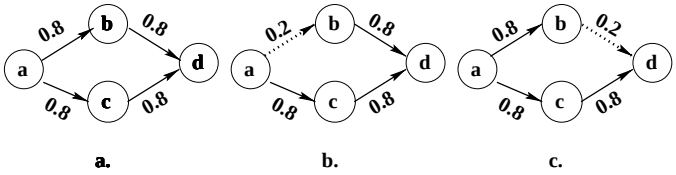


Fig. 3. (a) Reconvergent fanout which will impact test quality. (b) and (c) Two types of reconvergent fanout with negligible impact on test quality.

Figure 3b shows low normalized correlation from *a* to *b*. So, the value of node *b* will not strongly depend on the value of *a*. Therefore, *b* and *c* are relatively uncorrelated and this reconvergent fanout will not effect testability. Figure 3c shows low normalized correlation from *b* to *d*. Nodes *b* and *c* are correlated because both nodes depend on the value of node *a* in the previous time frame. This correlation will decrease the testability of the combinational block associated with *d*. However, low normalized correlation from *b* to *d*, implies that *b* has little impact on the detection of most faults. So correlation of *b* and *c* only affects a small number of faults. Our algorithm does not consider the two types of reconvergent fanout depicted in Figures 3a and 3b. Since many reconvergent fanouts do not need to be considered, our algorithm selects relatively few flip-flops, resulting in low area/performance overhead.

V. Partial BIST Insertion Algorithm

Our algorithm first constructs the S-graph of a sequential circuit and extracts the cones bounded by the nodes in S-graph. The normalized correlation values associated with each edge in the S-graph are calculated. This step is currently performed by exhaustive simulation of each combinational cone. The complexity of this step may be reduced by simulation of only a subset of input values, but we have chosen to perform simulation exhaustively to improve the accuracy of this experiment. The normalized correlation data is used to prune the S-graph by removing edges whose correlation is below a threshold value. Figure 4a is the original S-graph of benchmark s298, where the bold lines are used to represent a group of edges with the same source. By using a minimum correlation threshold of 0.4, the S-graph is pruned to generate the S-graph shown in Figure 4b which contains 1/3 less edges.

Algorithm 1 describes the greedy constructive approach used to select BIST flip-flops in the reduced S-graph. The algorithm computes the number of unbroken reconvergent fanout paths in which each flip-flop *v* is contained, ($rec_cnt(v)$). The flip-flop contained in the largest number of reconvergent fanout paths is selected as a BIST flip-flop. After each iteration, the **UpdateReconvergenceCount** function is called to update the rec_cnt values of each node in the S-graph to reflect the selection of a new BIST flip-flop. The process is iterated until all reconvergent fanout paths are broken. In our algorithm, only reconvergence of length 2 is considered because shorter paths have the strongest impact on correlation.

VI. Results

Table 1 shows the results of applying our algorithm to several ISCAS89 benchmark circuits. The experiments were performed on a PII-300 linux computer. The fault coverage is obtained using PROOFS [11] to fault simulate the circuit with 10,000 LFSR generates test vectors.

Column 5 contains the fault coverage without BIST insertion, applying pseudo-random patterns only at the primary inputs and observing results at primary outputs. Column 6 contains fault coverage after

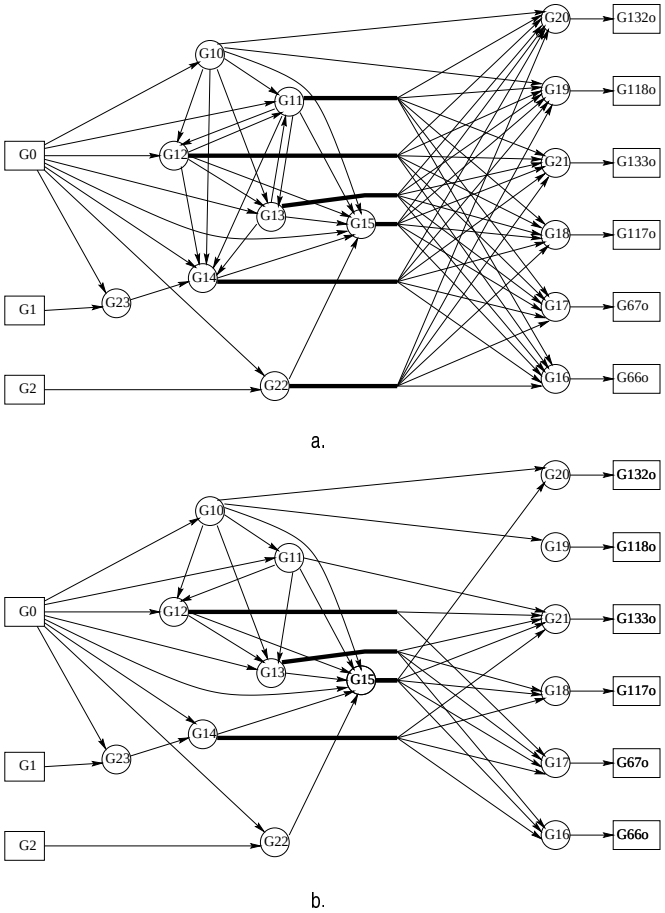


Fig. 4. (a) Original S-graph of s298, (b) Pruned S-graph of s298

Algorithm 1 BIST Flip-Flop Insertion (S-Graph G)

```
create S-graph
compute normalized correlation for each edge in S-graph
prune all S-graph edges with subthreshold correlation
UpdateReconvergenceCount(G)
repeat
    select the node w with maximum rec_cnt(v) as BIST cell
    delete all edges incident to node w
    UpdateReconvergenceCount(G)
until rec_cnt(v) = 0 for each node in G
```

performing BIST insertion using our approach to select flip-flops to act as a CBILBO register. We can see a great improvement of testability of most circuits after BIST insertion. Column 3 indicates the number of flip-flops configured as a CBILBO register using the loop breaking method presented in [7]. Our algorithm selects less flip-flops than Stroele's algorithm in all benchmark circuits except s1196 which contains no loops.

Since no fault coverage result are presented in [7], we cannot compare the coverage. In order to evaluate the fault coverage quality of our results, we simulate fault coverage of *all* BIST insertion solutions with the same number of selected flip-flops. For example, in circuit s344 there are 1365 BIST insertion solutions using 4 out of 15 flip-flops. We simulate fault coverage of all these combinations and rank the fault coverage of our solution among all solutions. Ranking results are listed in column 7. The average rank of our solutions is within the

Benchmark	# of total FFs	# of selected FFs		fault coverage		rank	CPU time (sec.)
		loops	correlation	w/o BIST	with BIST		
s298	14	14	4	0.21	0.840	15.7%	0.06
s344	15	15	4	0.89	0.994	0.1%	0.88
s382	21	15	9	0.12	0.932	4.7%	2.86
s386	6	6	5	0.32	0.989	16.7%	0.95
s510	6	6	4	0.00	0.998	20.0%	94.54
s953	6	6	5	0.08	0.957	16.7%	138.28
s1196	18	0	2	0.91	0.932	7.5%	4486

TABLE I
PARTIAL BIST INSERTION AND FAULT COVERAGE.

Algorithm 2 UpdateReconvergenceCount (S-Graph G)

for each node v in G **do**
  $rec\_cnt(v) = 0$
end for
for each pair of nodes u and v in G **do**
  find all paths from u to v with sequential length of 2, $u \rightarrow w_i \rightarrow v$
  if the number of paths ≤ 2 **then**
    for each intermediate node w_i **do**
      $rec\_cnt(w_i)$ increases by 1
    end for
  end if
end for

top 11.6%. Benchmarks s298 and s382 have 6 flip-flops connected to outputs directly. Configuring these flip-flops as BIST cells will not increase the testability of circuits, so the total combinations do not account these flip-flops. Figure 5 shows the fault coverage distribution of all BIST insertion solutions for s344 involving 4 flip-flops. The arrow in the figure shows the position of our result with respect to all solutions.

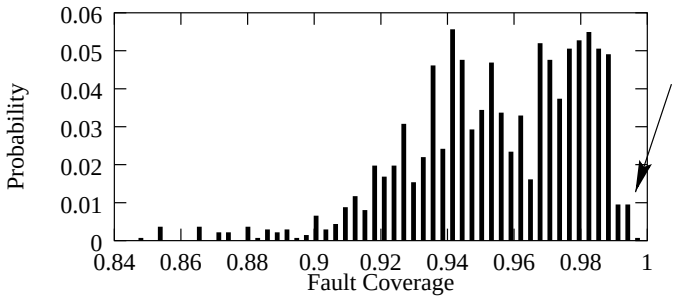


Fig. 5. Fault coverage distribution of benchmark s344.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we present a new method to insert BIST flip-flops which breaks reconvergent fanout paths to reduce data correlation. We have motivated the need to eliminate data correlation and our experimental results have demonstrated the benefits of this approach in terms of reduced overhead and high fault coverage. Determining normalized reconvergent fanout is currently a time consuming process, so in the future, we will investigate the use of sampling and structural analysis to derive the same information.

REFERENCES

[1] J. Savir, G. S. Ditlow, and P. H. Bardell, *"Random Pattern Testability,"* IEEE Transactions on Computers, Vol. C-33, No. 1, January 1984, pp. 79-90.

[2] S. T. Chakradhar, and S. Dey, *"Resynthesis and Retiming for Optimum Partial Scan ,"* 31st Design Automation Conference, pp. 87-93.

[3] S. T. Chakradhar, A. Balakrishnan, and V. D. Agrawal, *"An Exact Algorithm for Selecting Partial Scan Flip-Flops,"* 31st Design Automation Conference, pp. 81-86.

[4] K. T. Cheng, and V. D. Agrawal, *"A Partial Scan Method for Sequential Circuits with Feedback,"* IEEE Transactions on Computers, Vol. 39, No. 4, April 1990, pp. 544-548.

[5] R. Gupta, R. Gupta, and M.A. Breuer, *"BALLAST: A Methodology for Partial Scan Design,"* 19th International Symposium on Fault-Tolerant Computing, pp. 118-125.

[6] D. H. Lee, and S. M. Reddy, *"On Determining Scan Flip-Flops in Partial-Scan Designs,"* 1990 IEEE International Conference on Computer-Aided Design, pp. 322-325.

[7] A. P. Stroele, and H. J. Wunderlich, *"Hardware-Optimal Test Register Insertion,"* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 6 June 1998, pp. 531-539.

[8] J. Carletta, and C. Papachristou, *"Structural Constraints for Circular Self-Test Paths,"* 13rd IEEE VLSI Test Symposium, pp. 486-491.

[9] I. G. Harris, and A. Orailoglu, *"Testability Inprovement in High-Level Synthesis Through Reconvergence Reduction,"* Proceedings of the Asilomar Conference on Signals Systems and Computers, October 1995.

[10] S. Chiu, and C. A. Papachristou, *"A Design for Testability Scheme with Applications to Data Path Synthesis,"* 38th Design Automation Conference, pp. 271-277.

[11] T. M. Niermann, W.-T. Cheng, and J. H. Patel, *"PROOFS: A Fast, Memory-Efficient Sequential Circuit Fault Simulator,"* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 11, No. 2, Feb 1992, pp. 198-207.