Partial BIST Insertion to Eliminate Data Correlation

Qiushuang Zhang and Ian Harris
Department of Electrical and Computer Engineering
University of Massachusetts at Amherst
E-mail: qzhang@ecs.umass.edu, harris@ecs.umass.edu

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 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 results 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. Based on this metric, we provide an exact method and a heuristic method to measure the data correlation. 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 exact algorithm rank on average among the top 11.6% of all possible solutions with the same number of flip-flops.

Keywords

BIST, Data Correlation, Reconvergent Fanout

I. Introduction

Pseudo-random Built-In Self-Test (BIST) is an important testing technique which enables at-speed and on-site testing and reduces the cost of automatic test generation and expensive test equipment. In general, pseudo-random test patterns are generated by a Linear Feedback Shift Register (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 enables 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 [10], reconvergent fanout is removed during high-level synthesis to reduce the level of pseudo-random data correlation and reduce number of conflicts encountered during the ATG process. In [9], a method to evaluate the correlation caused by

This research was funded in part by National Science Foundation grant number NSF-0081343

reconvergent fanout is proposed and the high correlation is broken by removing the reconvergent fanout in order to improve the testability.

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 follows. Sections II and III introduce a metric for correlation along a circuit path and describe how that metric is used to direct the BIST insertion process. Section IV describes exact and approximate methods to compute the correlation metric. Sections V and VI provide the algorithm and the results. Section VII presents conclusions and future work.

II. Correlation Metrics

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 [11]. A significant source of data correlation is *matched reconvergent fanout*, which exists when several reconvergent paths have the same sequential depth. Figure 1a 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 1b depicts matched reconvergent fanout paths that have 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 1c, so that the value of line X is not dependent on the value of line B.

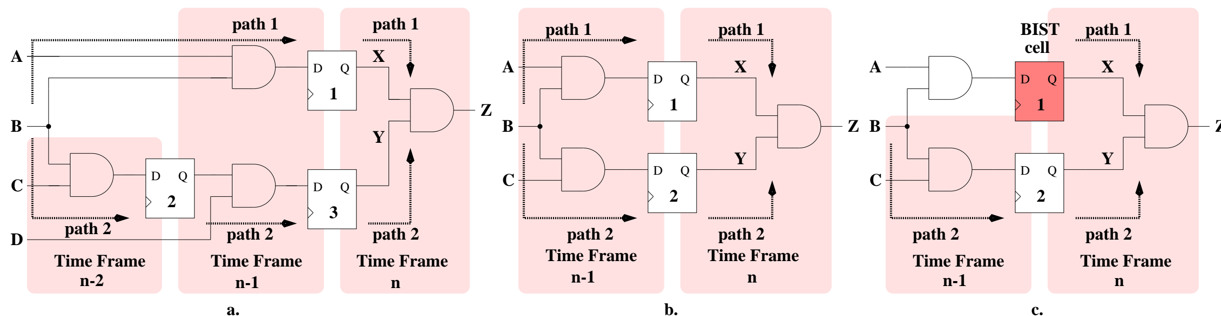


Fig. 1. (a) Mismatched reconvergent fanout (different sequential length). (b) Matched reconvergent fanout (same sequential length). (c) BIST insertion breaking reconvergent fanout.

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 of a combinational block depends on each input to that block. We define *serial correlation*

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	$(2^{n-1}-q)/2^n$

TABLE I
DISTRIBUTION TABLE

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 Table I. 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 $Z = 0$.

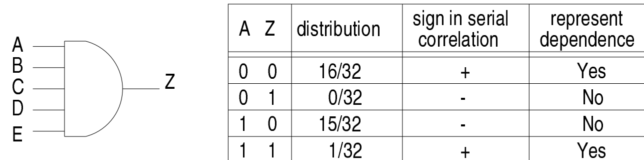
The covariance between input A and output Z is defined as

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

where μ_A and μ_Z are expected values of A and Z. $Prob.(A,Z)$ is probability of combination of (A, Z) in the 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 case of a 5-input AND gate as shown in Figure 2. The value of input A has a



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.

strong effect on 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. The effect of this 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* to alleviate the inaccuracy of serial correlation when an output is strongly dependent on multiple inputs. Normalized correlation between an input A and an output Z is defined in the following equation as serial correlation divided by maximum serial correlation between Z and all inputs to the combinational block driving Z.

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

$corr_{max}(A,Z)$ 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:

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.

III. Partial BIST Insertion with Correlation

We perform partial BIST insertion by selecting a set of flip-flops to act as part of a BIST register. The flip-flops are selected with the goal of breaking all significant reconvergent fanout in the circuit. The normalized correlation measure is used to evaluate the significance of a set of reconvergent fanout paths in terms of their impact on correlation. To model the circuit we 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 the S-graph. The combinational logic of a sequential circuit can be partitioned into cones, where each cone is a 1-output combinational logic block such that its inputs are either primary inputs or outputs of flip-flops, and its output is either a primary output or an input of a 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 the S-graph. The normalized correlation of each input-output pair of a cone is labeled on the corresponding edge of the S-graph. If every edge on a pair of reconvergent fanout paths has high normalized correlation, then that reconvergent fanout will have significant effect on

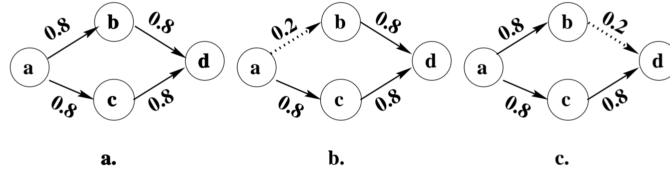


Fig. 3. (a) Reconvergent fanout which will impact test quality. (b) and (c) Two types of reconvergent fanout with negligible impact on test quality.

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. A dashed line indicates low normalized correlation.

Figure 3b shows low normalized correlation from a to b, indicating that the value of node b does 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. For this reason, the correlation between b and c only affects a small number of faults. The two types of reconvergent fanout shown in Figures 3b and 3c need not be considered because their impact on testability is small. Since many reconvergent fanouts do not need to be considered, our algorithm selects relatively few flip-flops, resulting in low area/performance overhead.

IV. Methods to Compute Normalized Serial Correlation

Normalized correlation measures the degree to which the value of an input to a combinational block can influence the value of an output to the combinational block. Computation of normalized serial correlation associated with each edge in the S-graph must be performed before partial BIST insertion can begin. We present two computation methods, the exact method which is based on the definition in Section II-B, and a heuristic method which measures the correlation based on combinational depth.

The exact method requires the generation of a truth table for the combinational blocks associated with each edge of the S-graph. Each truth table is used to create a distribution table as shown in Table I. Equations 2 and 3 are applied using the information in the distribution table to compute the normalized correlation associated with each edge in the S-graph. This method gives a more accurate correlation value at the cost of relatively high computational effort.

The computation time of the exact method can be impractical if a circuit is composed of complicated combinational blocks. To improve performance we introduce an alternative heuristic method to compute normalized correlation. The correlation between two nets is related to the combinational depth between the nets because each intermediate gate degrades the ability of the predecessor net to determine the successor net value. For example in Figure 4(a), net A has less direct influence on net Z than it does on net F, so the following inequality should hold, $\phi(A,Z) < \phi(A,F)$, where ϕ is the normalized correlation. The nature of the gates involved in the combinational path also impacts the correlation because a net may uniquely determine another net's value through a sequence of gates. This is seen in Figure 4(a) between nets D and Z. Although two gates separate these nets, the value of net D uniquely determines the value of net Z when $D = 0$. Since net D can uniquely determine Z but net A cannot, the following relationship should hold, $\phi(A,Z) < \phi(D,F)$. This small example shows that combinational depth alone is not sufficient.

To estimate correlation, combinational depth must be modified to consider cases where a net can uniquely determine the value of another net across multiple gates. We define a *correlation chain* as a set of gates connected in sequence where an input to the first gate in the chain can uniquely determine the output of the last gate in the chain. Correlation chains can be identified by examining the controlling and inverting values of the gates along the chain. The controlling value (c_p) of a gate p is the input value which independently determines the output value, and the inversion value (i_p) of a gate p indicates whether the gate function is positive or negative unate. Two gates p and q connected in sequence are part of the same correlation chain if the following condition holds, $c_p \oplus i_p = c_q$. If this condition holds then applying the value $c_p \oplus i_p$ at the input of gate p will uniquely determine the output of gate q. Because correlation is established along a correlation chain, the entire chain should be counted as a single gate when computing combinational depth to estimate correlation.

We define the **correlation level** between two nets as the combinational depth between the nets where each correlation chain is counted as a single gate. The correlation level is computed by following the shortest combinational path between the two nets. The gates contained in a correlation chain are merged into a single gate by applying DeMorgan's law as shown in Figures 4(a) and 4(b). The normalized correlation is inversely related to the correlation level; normalized correlation increases (decreases) as correlation level decreases (increases).

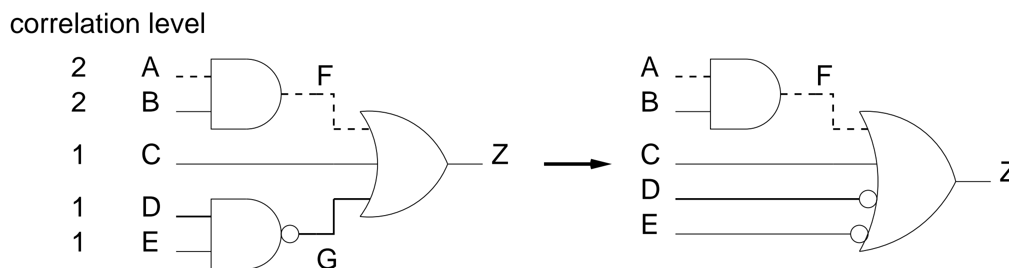


Fig. 4. A sample circuit of calculating correlation level.

The partial BIST insertion algorithm uses a *minimum correlation threshold* value to identify correlation which impacts testability significantly. The correlation information needed for each S-graph edge is binary, either correlation is above threshold or it is not. The computation of the correlation values of each S-graph edge is made more efficient by abandoning the calculation for an edge as soon as it can be ascertained to have correlation which is above threshold. The computation of correlation for an edge is performed in a breadth-first fashion which halts once correlation is found to exceed the threshold. The heuristic correlation computation algorithm is shown in Algorithm 1.

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 (or correlation level) associated with each edge in the S-graph are calculated. The normalized correlation data is used to prune the S-graph by removing edges whose correlation is below a threshold value. Figure 5a 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 (on a scale of 0 to 1), the S-graph is pruned to generate the S-graph shown in Figure 5b which contains 1/3 fewer edges. If we use a lower normalized correlation threshold, less edges are pruned and more flip-flops are configured as BIST cells. The threshold controls the tradeoff between the coverage

Algorithm 1 Compute Correlation ()

```
label output as level 1
put output into Queue Q, set the control values of the output to both 1 and 0
repeat
   dequeue the first gate g in Queue Q
   for each input l of gate g do
      level(l) = level(g) + 1
      find the gate p driven the input l of gate g
      if c_p ⊕ i_p = c_g then
         level(l) decrease by 1
      end if
      if level(l) ≤ threshold level then
         put l into Q
      end if
   end for
until Queue Q is empty
```

and hardware overhead. Our experiments show that threshold of best results for the benchmarks used in Table II is between 0.3 and 0.4, so we have selected 0.4 as our threshold of correlation.

When the heuristic method is used to estimate the correlation, a uniform correlation level threshold of 2 is applied in our experiments. Since the heuristic method will introduce errors in measuring the correlation, the heuristic method will select more flip-flops if it overestimates the correlation and will select less flip-flops if it underestimates the correlation. For this reason, the heuristic method is inferior to the exact method in measuring the correlation. However, this minor weakness enables great improvement in computation time.

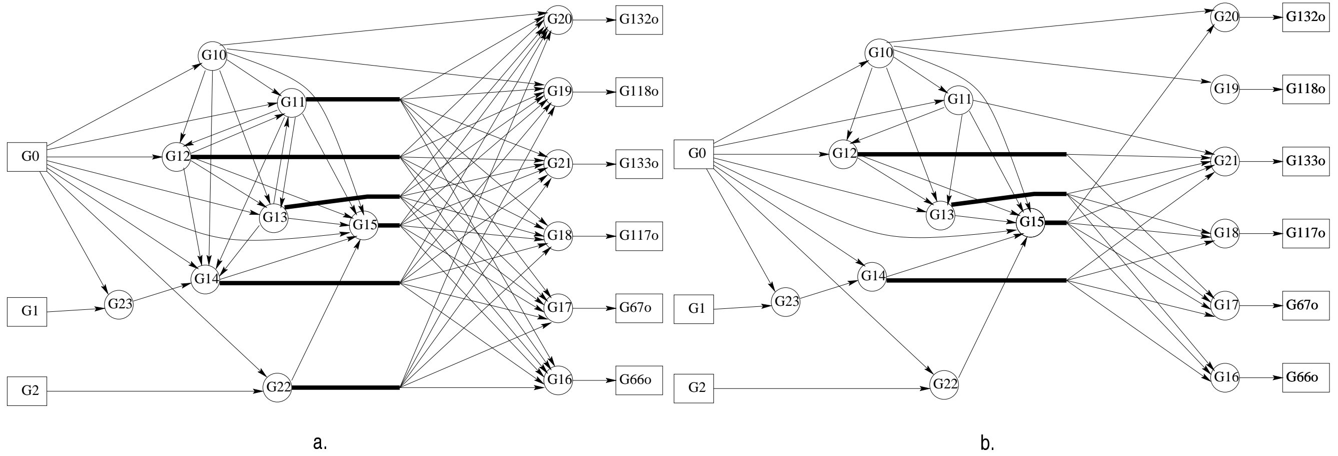


Fig. 5. (a) Original S-graph of s298, (b) Pruned S-graph of s298

The partial BIST selection problem can be reduced to a Minimum Feedback Vertex Set (MFVS) problem [12] which is NP-complete. Since an exact solution is not tractable, a heuristic approach must be used. Algorithm 2 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 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 shown in Algorithm 3 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.

Algorithm 2 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
```

Algorithm 3 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 → w_i → 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
```

VI. Results

This section shows two sets of results, obtained by the exact method and the heuristic method. We first analyze the results using the exact method in order to show the efficiency of breaking data correlation. Table II shows the results of applying the exact method to several ISCAS89 benchmark circuits. The experiments have been performed using a Pentium II 300MHz Linux computer. The correlation threshold value used in these results is 0.4. The fault coverage is obtained using PROOFS [13] to fault simulate the circuit with 10,000 LFSR-generated test vectors.

Column 5 of Table II shows the fault coverage achieved without BIST insertion by applying pseudo-random patterns only at the primary inputs and observing results at primary outputs. Column 6 shows fault coverage after performing BIST insertion using our exact approach to select flip-flops to act as part of a CBILBO register. We can see a significant 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

Benchmark	# of total FFs	# of selected FFs		fault coverage		rank	CPU time (sec.)
		loops	correlation	w/o BIST	with BIST		
s298	14	14	4	21%	84.0%	15.7%	0.06
s344	15	15	4	89%	99.4%	0.1%	0.88
s382	21	15	9	12%	93.2%	4.7%	2.86
s386	6	6	5	32%	98.9%	16.7%	0.95
s510	6	6	4	0%	99.8%	20.0%	94.54
s953	6	6	5	8%	95.7%	16.7%	138.28
s1196	18	0	2	91%	93.2%	7.5%	4486

TABLE II
PARTIAL BIST INSERTION AND FAULT COVERAGE.

in [7]. Our algorithm selects fewer flip-flops than Stroele's algorithm in all benchmark circuits except s1196 which contains no loops.

Since no fault coverage results are presented in [7], we cannot compare based on fault 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 compute the fault coverage associated with all these combinations and rank the fault coverage of our solution among all solutions as shown in column 7 of Table II. The average rank of our solutions is within the top 11.6%. Figure 6 shows the fault coverage distribution of all BIST insertion solutions for s382 involving 9 flip-flops. The arrow in the figure shows the position of our result with respect to all solutions.

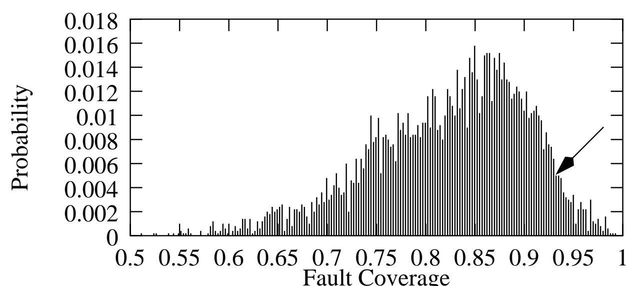


Fig. 6. Fault coverage distribution of benchmark s382.

Table III shows the results of applying the heuristic method with threshold correlation level of 2. Column 2 of Table III shows the total number of flip-flops in each benchmark. Columns 3, 4 and 5 are the number of selected flip-flops, fault coverage and CPU time using our heuristic method. Column 6 is the number of selected flip-flops using the loop breaking method [7].

Comparing our heuristic method to the loop breaking method we can see that our heuristic method selects fewer flip-flops for most benchmarks except s1196 and s1238. We also present the fault coverage results of our heuristic method in column 4. The fault coverage of s208, s420 and s838 are low because the circuit cannot be initialized. The coverage of the other benchmarks are above 0.70 except s444. Since we use a uniform threshold in all the benchmarks, the error introduced by heuristic method may result in fewer flip-flops selected and lower fault coverage. Figure 7 shows the relation between correlation level, the number of

Benchmark	# of total FFs	breaking correlation			breaking loop
		sel. FFs	coverage	CPU time(s)	sel. FFs
s208	8	5	30%	<1	8
s298	14	7	97%	<1	14
s344	15	3	98%	<1	15
s349	15	3	98%	<1	15
s382	21	7	74%	<1	15
s386	6	3	70%	<1	6
s400	21	7	74%	<1	15
s420	16	11	27%	<1	16
s444	21	5	55%	<1	15
s510	6	5	100%	<1	6
s526	21	14	98%	<1	21
s526n	21	14	98%	<1	21
s641	19	2	93%	2	7
s713	19	0	81%	2	7
s820	5	4	83%	<1	5
s832	5	4	82%	<1	5
s838	32	23	26%	1	32
s953	29	5	96%	<1	6
s1196	18	2	92%	1	-
s1238	18	2	87%	1	-
s1423	74	28	84%	13	71
s1488	6	5	93%	<1	6
s1494	6	5	92%	<1	6
s5378	179	21	84%	9	30

TABLE III
PARTIAL BIST INSERTION AND FAULT COVERAGE BY HEURISTIC METHOD.

flip-flops selected and fault coverage. When the correlation level increases, more flip-flops are selected and higher fault coverage is achieved.

An advantage of the heuristic method over the exact method is the greatly reduced CPU time. The results in Table II take up to 4486 seconds using the exact method, while they take less than or equal to 1 second using the heuristic method. The time complexity of the heuristic method is polynomial in the number of flip-flops or the number of gates of circuits.

Table IV compares the results of the heuristic method to the results of the exact method. To make the results comparable, we have adjusted the threshold of the heuristic method to select the same number of flip-flops as the exact method. In Table IV, column 2 is the number of flip-flops selected in each benchmark. Columns 3 and 4 are the fault coverage results of both methods based on the same number of BIST cells listed in column 2. We see that the exact method has higher fault coverage than the heuristic method when the number of flip-flops selected by the two approaches is equal. The heuristic method achieves great improvement in computation time by sacrificing only a small amount of fault coverage.

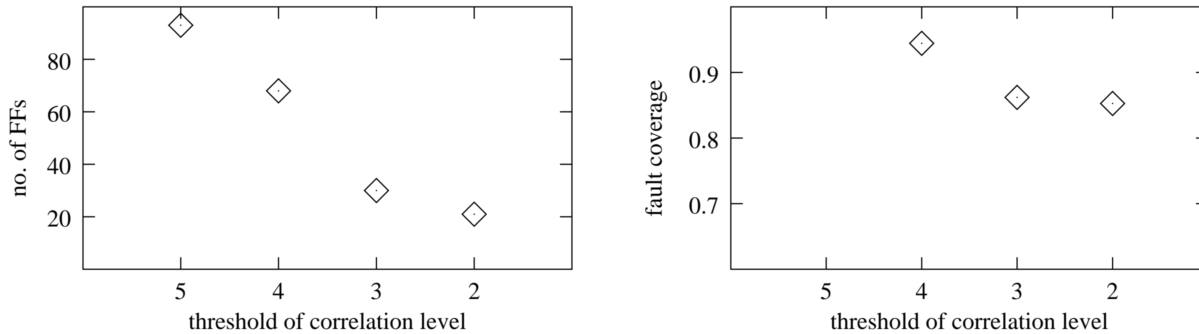


Fig. 7. Relation of correlation level, the number of flip-flops selected and fault coverage of benchmark s5378.

Benchmark	**BIST FFs**	**coverage (heu.)**	**coverage (exct.)**
s298	4	79.9%	84.0%
s344	4	98.2%	99.4%
s382	9	87.7%	93.2%
s386	5	98.9%	98.9%
s510	4	96.3%	99.8%
s953	5	95.6%	95.7%
s1196	2	92.4%	93.2%

TABLE IV

COMPARISON OF HEURISTIC AND EXACT METHODS OF PARTIAL BIST INSERTION.

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. 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, 1994, 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, 1994, 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, June 1989, 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, 1995, pp. 486-491.

[9] Q. Zhang, and I. Harris, *"Partial BIST Insertion to Eliminate Data Correlaiton,"* 1999 IEEE International Conference on Computer-Aided Design, pp.395-398 .

[10] 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.

[11] S. Chiu, and C. A. Papachristou, *"A Design for Testability Scheme with Applications to Data Path Synthesis,"* 28th Design Automation Conference, 1991, pp. 271-277.

[12] M. R. Garey and D. S. Johnson, *"Computers and Intractability: A Guide to the Theory of NP-Completeness,"* W. H. Freeman and Company, 1979.

[13] 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.

[14] M. Abramovici, M.A. Breuer, and A.D. Friedman, *"Digital Systems Testing and Testable Design,"* IEEE press, 1994.

Response to the Comments

Reviewer 1

Comment 1.1 Unfortunately, the problems with the substandard English have not been resolved. ... The authors need to seriously proofread the paper, as opposed to simply cleaning up the specific errors that reviewers point out. Specifically, the authors to look at these issues (a writing handbook covers these):

- *when to use "that", and when to use "which" (throughout the paper, "which" is used incorrectly.)*
- *when to use a comma, and when to use a semi-colon (throughout the paper, commas are used incorrectly to splice together two complete sentences).*
- *when to use "affect", and when to use "effect".*
- *when to use "less", and when to use "fewer".*

Response 1.1 We have proofread the document, looking for the problems identified by the author as well as other english problems. We believe that the vast majority of these problems have been fixed.

Comment 1.2 Also:

- *running a spell checker would pick up mistakes: on page 6, the words "Normalixed", "wuation", and the phrase "as as"; "correlaiton" on page 18.*
- *page 3: "matched reconvergent fanout paths which are the same sequential depth" should be "matched reconvergent fanout paths that have the same sequential depth"*
- *the wording of "This positive correlation is negated...." on page 5 is vague.*
- *Many, many sentences start with "So...". This should be avoided.*
- *on page 7, "Dashed line indicated low normalized correlation" should be "A dashed line indicates low normalized correlation".*
- *First sentence in second paragraph of page 8 is incomplete. ("The exact method requires....") The sentence after that also has problems: "The each truth table..."*
- *On page 11, it is hard to see what the "Then" in "Then the partial BIST solution can be reduced...." refers to. It should be replaced by something like: "Once the graph has been reduced by pruning edges of low correlation..."*
- *On page 11, "improvement of the computation time" should be "improvement in the computation time".*
- *On page 13, the sentence "First analyze the results by the exact method..." has no subject.*
- *On page 13, "Since no fault coverage result are presented" should be "Since no fault coverage results are presented"*
- *English in page 14 is a mess, and hard to follow as a result.*
- *In table IV, the second column should have a much more descriptive heading than "number limit". Perhaps, "number of BIST flip-flops inserted"?*

Response 1.2 All of the english problems identified by the reviewer have been fixed.

Reviewer 2

Comment 2.1 The result tables doesn't seem to present a fair comparison. What I would have liked to see in one of the Tables is a comparison between the BIST (full scan) vs BIST (partial scan - your method). Under these conditions, the fault coverage, area overhead, test points, etc. should be compared. Then you could show us the real benefits of the proposed methodology.

Response 2.1

Comment 2.2 In one of the comments you mention that "pseudo-random BIST is not applicable to all circuits, so we cannot expect that fault coverage (FC) will always be acceptable". That's a very good point. To improve the FC for circuits that have BIST, usually test points are necessary, and in addition, top-up ATPG patterns are required. We all know that scan cells are the best kind of test points as it increases both the controllability and observability of circuits. So - it seems to me that you are suggesting the following flow. You can start by getting rid of scan cells that need not be scanned based on your heuristics. Then, do fault simulation or testability analysis to determine the FC, and if the FC is low, insert lots of test points, which again may mean converting the remaining sequential elements to scan cells.

Response 2.2 We assume that a designer wishes to use BIST in his/her design. Given that assumption, the flow that we are suggesting is to perform partial BIST insertion using our heuristics, and then evaluate the fault coverage by fault simulation. If fault coverage is insufficient then adjust the correlation level threshold until the fault coverage goals are met. If fault coverage goals cannot be met using our approach then some other method for enhancing testability would be required, but we leave the choice of an additional test enhancement method to the designer.

Comment 2.3 Secondly, in any practical BIST application, top-up patterns are necessary to improve the overall coverage for manufacturing test. Do you realize that making the circuit partial scan - the complexity of generating deterministic vectors may increase substantially.

Response 2.3 We do understand that a using partial scan will increase ATPG complexity as compared to using full scan. If the area and performance overheads of full scan are acceptable by the designer then it is preferable to partial scan from a tool standpoint. We, and other researchers in partial scan/BIST, are targeting designs in which the area and performance overhead incurred by full scan is too great to be acceptable. For such designs, partial scan/BIST are reasonable alternatives, even at the risk of increased ATPG complexity.

Comment 2.4 Another practical limitation is as follows. One of the other reasons why full scan is used predominantly in the industry is the ease of doing system level debug or fault diagnosis. As a result, partial scan never seemed to be an attractive solution to the industry.

Response 2.4 The reviewer brings up a limitation of partial scan/BIST approaches. Although this problem exists, it should not prohibit exploration into partial scan/BIST which has other advantages. Instead, this problem should be seen as an area for research. In fact scan and BIST-based diagnosis techniques are currently a very active research area, in part for the reason mentioned by the reviewer.

Reviewer 3

Comment 3.1 On page 9, par. 2 (and in your Responce 3.6) you state that normalized correlation is inversely proportional to the correlation level. For me, it can be understood as either (1) There is a constant C so that normalized correlation = C / correlation level holds, or (2) If normalized correlation increases, correlation

level decreases, but not necessarily strictly proportional. It should be made clear what definition is meant. (For me, Responce 3.6 indicates that definition (2) is meant, but, in this case, the term 'inversely proportional' is somewhat misleading.

Response 3.1 This point is now clarified in Section IV, page 6, second paragraph.

Comment 3.2 On page 9, c_p*,* c_q *and* i_p *should be introduced before using them (actually, c without index is introduced and for i it can be figured out that it is the inverting value, but I think it would be better to read).*

Response 3.2 These terms are now properly introduced in Section IV, at the last line of page 5 and at the top of page 6.

Comment 3.3 Section IV line 8: The each =¿ Each

Response 3.3 We have corrected this error.