A Routing-Aware ILS Design Technique

S. Banerjee, IEEE Member, J. Mathew, IEEE Member, D. K. Pradhan, IEEE Fellow,
B. B. Bhattacharya, IEEE Fellow, S. P. Mohanty, IEEE Senior Member

***Abstract*— The Illinois Scan Architecture (ILS) consisting of several scan path segments is useful in reducing test application time of high density chips. In this paper, we propose a scheme of layout-aware as well as coverage-driven ILS design. The partitioning of the flip-flops into ILS segments is determined by their geometric locations, whereas the set of the flip-flops to be placed in parallel is determined by the minimum incompatibility relations among the corresponding bits of a test set, to enhance fault coverage in broadcast mode. This consequently, reduces the number of test patterns required in serial mode.**

I. Introduction

The serial full scan method reduces the cost of test generation and provides high fault coverage. However, its test application time and power dissipation is significantly high because of the inherent serial nature of the scan path. Among the existing methods that are used to tackle the problem of reducing the test application time, one approach is to configure the scan elements into multiple scan chains [1]. However, this increases the number of scan-in pins and scan-out pins during test and does not reduce the test data volume. In [2], authors present a test methodology that allows a single input data line to support multiple scan chains. The ILS is proposed to reduce the test application time and the test data volume [3], [4], [5], [6]. It is applicable to both stand alone cores or cores embedded in SoC, and requires a low-cost automatic test equipment. In the ILS structure, a long scan chain is divided into many short segments that are loaded in parallel with the same test vector. The outputs of the different scan segments are fed to an multiple input signature register $(MISR)$ to compress the responses [6].

In this paper, we propose layout-aware design of ILS architecture, which accounts the physical locations of the flip-flops, and provides tradeoffs among fault coverage, test application time/test data volume, and wiring cost. The proposed algorithm consists of three steps. First, scan cells are grouped based on their geometric proximity to form the different segments of the ILS; this reduces wiring cost. Second, the flip-flops in different segments are aligned based on some incompatibility information derived from a test set so that the fault coverage in broadcast mode is enhanced. As a result, the number of additional test patterns required in the serial mode reduces. Finally, some reordering of scan cells concurrently on all the segments, is performed to reduce scan-path length further. Experimental results on various benchmark circuits show that the proposed ILS structure provides significant reduction in test application time with high fault coverage while reducing wire length in the scan chains, compared to other structures designed by earlier methods.

S. Banerjee, J. Mathew, and D. K. Pradhan are with University of Bristol, UK., E-mail:pradhan@compsci.bristol.ac.uk. B. B. Bhattacharya is with Indian Statistical Institute, Kolkata, India, E-mail:bhargab@isical.ac.in, and S. P. Mohanty is with University of North Texas, E-mail:saraju.mohanty@unt.edu.

II. Layout-Aware ILS Generation Algorithm

The three basic steps of the algorithm are now described.

A. Partitioning of flip-flops into ILS segments

In this step, we partition the flip-flops into different groups based on their geometric neighborhood. The flip-flops nearest to each other are grouped, so that scan path length is minimized. Each group forms a segment of ILS, and the number of groups are equal to the number of required segments in the ILS structure, which may be a user-defined parameter. In order to partition the flip-flops into different segments, we first calculate the Euclidean distance of each flip-flop from the origin. We order the flip-flops according to these distances. Here, we assume that flip-flops with low Euclidean distance will have lower order. In this way, flip-flops are arranged in a list from low order to higher order. Now, we partition them into different groups sequentially from low order to higher order. For example, consider the *Fig. 1* with 10 flip-flops. Based on the Euclidean distances of these flip-flops from the origin, we order them as $FF1$, $FF2$, $FF4$, $FF6$, $FF5$, $FF3$, $FF8$, $FF7$, $FF9$, and $FF10$. Once ordered, we divide them into two partitions P_1 and P_2. Partition P_1 consists of flip-flops $FF1$, $FF2$, $FF4$, $FF6$, and $FF5$ while P_2 consists of $FF3$, $FF8$, $FF7$, $FF9$, and $FF10$. The runtime of this partition algorithm is linear and can be applied to a large circuits.

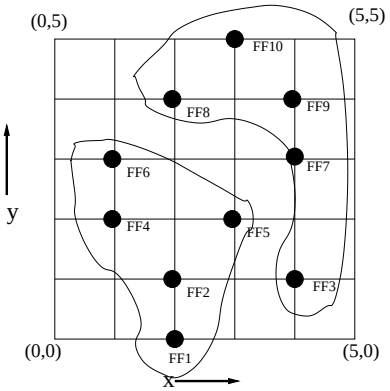

Fig. 1. partitioning of scan flip-flops

B. Determining the flip-flops to be placed in parallel

For a given test set of the original circuit, the algorithm analyzes the compatibility relationships among the flip-flops. The algorithm selects one flip-flop from each segment at a time, such that they have minimal incompatibility relationship

among themselves. All such flip-flops will be placed in parallel (at the same depth) in the ILS structure. This will give rise to high fault coverage in broadcast mode.

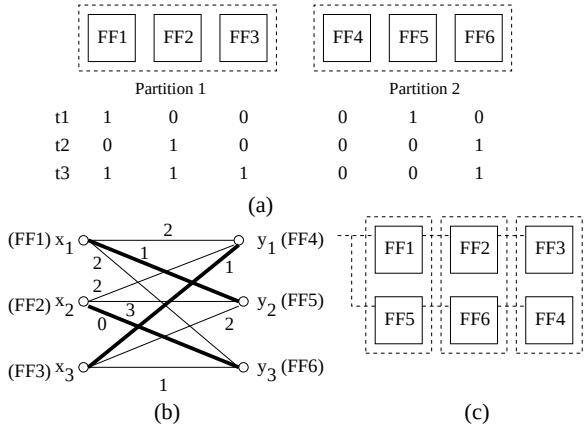

Fig. 2. Example of generation of ILS structure

For example, consider the test set shown in *Fig. 2.a*, which consists of three test vectors. The circuit has six flip-flops; hence each test vector is six-bit long. These six flip-flops are partitioned into two segments based on their coordinates. Each segment consists of three flip-flops: $FF1$, $FF2$, and $FF3$ are in segment 1, and $FF4$, $FF5$, and $FF6$ are in segment 2. The ILS structure will have two segments and each segment contains three flip-flops. The next task is to find out three groups of two flip-flops each, by selecting one flip-flop from each segment. The flip-flops belonging to a group will be placed at the same depth in the ILS structure. This selection problem based on minimal incompatibility can be solved by finding a perfect matching of a bipartite graph with minimum weight, which is described next.

The incompatibility distance (denoted as d) of two flip-flops belonging to two segments is defined as the number of conflicting bits in two corresponding column vectors of the test matrix. A graph $G(X, Y)$, called weighted incompatibility bipartite graph (WIBG) is then constructed, whose left set of vertices represents the flip-flops in one segment, and right set of vertices represents the flip-flops in the other segment. As shown in *Fig. 2b*, $X = \{x_1(FF1), x_2(FF2), x_3(FF3)\}$ denotes the set of flip-flops in the segment 1 and $Y = \{y_1(FF4), y_2(FF5), y_3(FF6)\}$ denotes the set of flip-flops in the segment 2. The weight on an edge represents the incompatibility distance between the two vertices connected by the edge. A zero weight denotes a compatible pair. The resultant WIBG of *Fig. 2a* is shown in *Fig. 2b*. In a WIBG $G(X, Y)$, a non-negative weight w_{ij} is assigned to each edge $x_i y_i$ of G. We seek a perfect matching M of graph G that minimizes the total weight $w(M)$. A matching in a graph G is a set of non-loop edge with no shared end points [7]. A perfect matching in a graph is a matching that saturates every vertex. In *Fig. 2b*, the minimum matching consists of three edges i.e. $M = \{x_1y_2, x_2y_3, x_3y_1\}$, and the weight of M is 2. Three groups of flip-flops are formed which are g_1 ($\{FF1, FF5\}$) connected by the edge x_1y_2, g_2 ($\{FF2, FF6\}$) connected by the edge x_2y_3, and g_3 ($\{FF3, FF4\}$) connected by the edge x_3y_1. In this way, an ILS structure is built where the flip-flops lying at the same depth in the scan structure will have a minimal incompatibility relation. The resultant ILS structure without interconnection among the flip-flops is shown in *Fig. 2c*. To find a perfect matching with minimum weight we replace each weight w_{ij} with $Z - w_{ij}$ for some large number Z (say 100). Then, we use an algorithm for computing maximum weight perfect matching, which is solvable in polynomial time.

When the number of segments is more than two, we form the groups by considering two segments at a time. After forming the groups of flip-flops between these two segments, another segment is considered. The process continues until all the flip-flops in each segment are grouped. For n segments (denoted as (n_p), e.g. $p_1, p_2, \ldots, p_{n_p}$), the process is repeated for $(n-1)$ times. The steps are outlined in *Alogirtim 1*.

Algorithm 1 Generate the ILS structure

```
Set two integers i=1 and j=2.
for all segments in the design do
    Consider the segments p_i and p_j.
    Generate the WIBG G(X, Y) for p_i and p_j
    Replace the weight w_ij of each edge with Z − w_ij.
    Find maximum weight perfect matching M.
    Group the two FFs connected by each edge of the M.
    if |M| equal to n_e then
        There are n_e groups of FFs with two each.
    end if
    i = i + 1, and j = j + 1.
    if j is not equal to n_p then
        continue
    end if
end for
```

C. Ordering of the scan cells

Once FFs in each segment and the groups of FFs lying at the same depth has been identified, scan cell ordering with in a partition or branch can be done in two ways: (1) $Case-1$: Without considering the power dissipation during scan cell reordering which is done by solving the traveling salesman problem. (2) $Case-2$: A reordering technique has been employed for minimizing the test power consumption in ILS during both capture and shift cycles. The algorithm corresponds to $Case-1$ guarantees to produce minimum scan-path length. The $Case-2$ method reduces the dynamic power dissipation during scan shift cycle.

In $Case-1$, scan cell ordering within a branch or partition is done by using a weighted distance graph (WDG). The WDG is a directed graph where each group (obtained from phase-2) are vertices and two directed edges between each two groups i.e., e_{ij} from g_i to g_j, and e_{ji} from g_j to g_i. In addition to these vertices, WDG also contains one vertex corresponds to scan-in pin (si) and one corresponds to scan-out pin (so). As outputs of the ILS structure are fed either to $MISR$ or Compactor in the present case, we have considered only one scan-out pin. So, the WDG for the present example consists of 5 vertices which are g_1, g_2, g_3, si, and so. In WDG, in

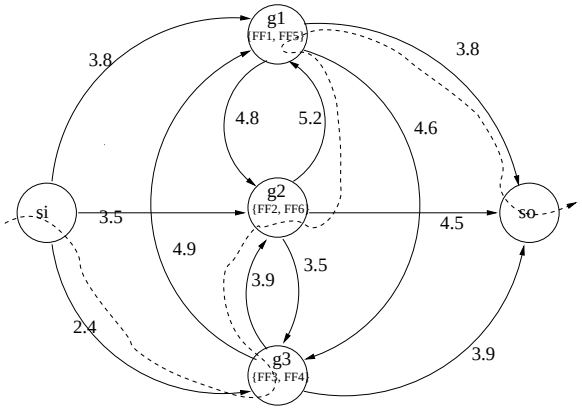

Fig. 3. The WDG for ordering scan cells

addition to the directed edges among the g_1, g_2 and g_3, there will be six more edges. Three edges are due to si ($si \rightarrow g_1$, $si \rightarrow g_2$, $si \rightarrow g_3$) and three edges due to so ($g_1 \rightarrow so, g_2 \rightarrow so, g_3 \rightarrow so$). The weight of the each edge depends on the average routing distance between the nodes. Let FF_i and FF_j be two scan cells. The distance $d(FF_i, FF_j)$ is the distance between the output port of FF_i and input port of FF_j. Similarly, the distance $d(si, FF_i)$ $(d(FF_i, so))$ is the distance between the si (output port of FF_i) and input port of FF_i (so). Now, a group of scan cells should be put at the first level of ILS structure which is close to scan-in pin (si). Similarly, at the last level group of scan cells should be close to the scan-out pin (so). Based on above, weight $w_{g_i \rightarrow g_j}$ is the average routing length between scan cells in g_i and g_j. Similarly, weight $w_{si \rightarrow g_i}$ $(w_{g_i \rightarrow so})$ is the average distance from si (scan cells in g_i) to scan cells in g_i (so). The final WDG for the *Fig. 2* is shown in *Fig. 3*. After creating WDG, the order of the scan cells can be determined by finding a shortest path from si to so. As here all the weights are positive, we have used Dijkstra's algorithm whose runtime is linear to find out the shortest path. In the present case best path should be $si \rightarrow g_3 \rightarrow g_2 \rightarrow g_1 \rightarrow so$ as shown by dotted line in *Fig. 3*. The final ILS structure of *Fig. 2* after scan cell reordering is shown in *Fig. 4*.

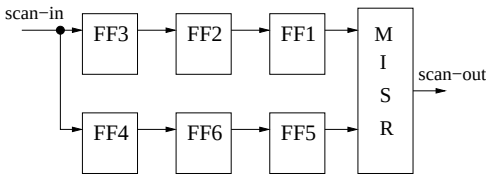

Fig. 4. The final ILS structure for *Fig. 2*

For $Case-2$, once the flip-flops in each segment and their linear order is determined, a reordering technique, similar to that in [8] is employed for minimizing test power consumption in the ILS for both capture and shift cycles. In our case, we need to order the scan cells in one segment only, and the scan cells lying in the other segments will be automatically ordered, as those lying at the same depth are fixed by the algorithm. The selection of a suitable segment for this purpose is guided by a heuristic that leads to maximum power saving [8].

III. Serial Mode

Once we obtain an ILS structure, we rerun the same $ATPG$ tool to generate a new set of test vectors with the logical constraints imposed on the secondary inputs of the circuit by segmentation of scan cells. However, in the presence of the constraints, some detectable faults in the original circuit may become untestable or hard-to-test in the broadcast mode. The undetectable faults can now be detected by reconfiguring dynamically the ILS structure into the serial mode, and by applying a few additional test patterns. *Fig. 5* illustrates the technique. The *Algorithm 2* described below, is used to determine the complete test patterns T for a circuit. The set T consists of two parts, T_B and T_S, i.e., the set of test patterns for the broadcast mode and for the serial scan mode respectively.

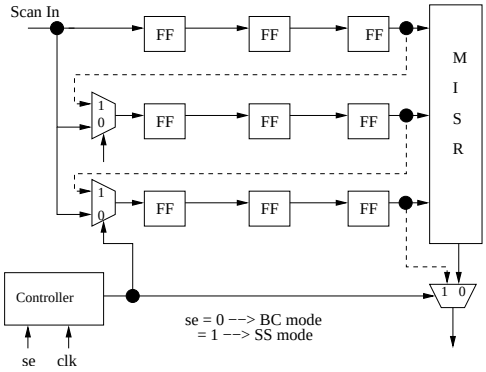

Fig. 5. The ILS with dynamic reconfiguration

Algorithm 2 Generate test patterns and test application time

1: Find the set of complete faults f_c in the circuit.
2: Insert the input constraints by using the ILS structure.
3: Run $ATPG$ to generate the set of test patterns T_B.
4: Use incremental fault simulation for each pattern of T_B to compute the detectable faults.
5: The detected faults are stored in f_d.
6: Compute the set of undetectable or hard-to-detect (HTD) faults $f_u = f_c - f_d$.
7: Set the circuit in serial scan chain mode.
8: Generate the set of serial test patterns T_S for the target faults f_u (using the $ATPG$).
9: Compute $T = T_B \bigcup T_S$.
10: Compute the test application time/test data volume for the complete test set T.

IV. Experimental Results

Experiments were performed on each circuit with $0.18\mu m$ standard cell library provided by the National Semiconductor. Results on scan wire length for these benchmark circuits are shown in *Table I*. The results show that scan-path length decreases when number of segments in the ILS structure increases. This is expected because as the number of partitions increases, neighborhoods FFs are grouped together. As a result, length of the scan-path decreases. On the average,

TABLE I
TOTAL WIRE LENGTH OF THE SCAN PATH W.R.T. # OF SEGMENTS

Number of segments in ILS	$s1423$ WL (μm)	Improvement over serial scan	CPU time (sec)	$s5378$ WL (μm)	Improvement over serial scan	CPU time (sec)	$s9234$ WL (μm)	Improvement over serial scan	CPU time (sec)
1 (serial scan)	$9.1E+3$	$--$	0.7	$5.1E+4$	$--$	0.7	$7.5E+4$	$--$	1.1
2	$8.7E+3$	$1.04X$	1.2	$4.4E+4$	$1.16X$	1.5	$7.2E+4$	$1.04X$	1.9
4	$8.5E+3$	$1.07X$	1.5	$4.2E+4$	$1.21X$	1.8	$7.1E+4$	$1.06X$	2.4
6	$8.3E+3$	$1.10X$	2.4	$4.1E+4$	$1.24X$	2.8	$7.1E+4$	$1.06X$	3.5

TABLE II
RESULTS ON TEST CYCLE REDUCTION USING PROPOSED ALGORITHMS

Number of Segments in ILS	$s9234$						$s13207$					
	Broadcast		Serial	total	total	TAT	Broadcast		Serial	total	total	TAT
	#Test patterns	Fault coverage	#Test patterns	test cycles or TAT	coverage (%)	improvement over full scan	#Test patterns	Fault coverage	#Test patterns	test cycles or TAT	coverage (%)	improvement over full scan
1	-	-	276	63432	98.25		-	-	412	276709	99.02	
2	264	80.05	70	46468	98.12	$1.36X$	384	79.70	72	178268	99.02	$1.55X$
4	253	71.28	94	36232	98.10	$1.75X$	367	70.20	134	119009	99.02	$2.32X$
6	248	65.23	97	31903	98.10	$1.98X$	323	61.24	110	120472	98.90	$2.28X$

the proposed method achieves $1.2X$ to $1.4X$ improvement in scan-path length compare to full scan methodology.

Table II show the test application time (TAT) and fault coverage obtained for the ILS. On the average, the proposed algorithm achieves $1.3X$ to $1.9X$ reduction in the TAT compared to full scan. *Table III* shows that the proposed method reduces the test application time significantly compared to an earlier approach [4], with a high fault coverage and at the same time reduces the scan path wire length.

TABLE III
TEST CYCLES COMPARISON

Circuit Name	Previous Method [4]		Proposed Method	
	Config.	Cycles	Config.	Cycles
$s13207$	$ILS-360$	176732	$ILS-335$	178268
	$ILS-180$	99611	$ILS-168$	119009
	$ILS-90$	169751	$ILS-112$	120472
$s15850$	$ILS-400$	177219	$ILS-229$	134847
	$ILS-200$	93810	$ILS-150$	96751
	$ILS-100$	54673	$ILS-100$	84821

Table IV reports the power reduction (P_{wt}) obtained during scan testing by varying the number of segments in the ILS. For each circuit, we calculated two sets of P_{wt} for algorithm without and with power consideration, respectively. The power reduction P_{wt} is calculated as: $P_{wt} = (wt_k - wt_1)/wt_1 \times 100\%$ Where $k > 1$, wt_1 denotes the total weighted transitions when the number of segments in ILS is one, and wt_k is the total weighted transitions when the number of segments is k. The results indicates that when we select power reduction algorithm, P_{wt} is smaller. So, the present algorithm produces a trade-off between the wire length and power dissipation which user can select based on requirements.

V. CONCLUSIONS

The ILS architecture is capable of reducing test application time and test data volume significantly. In this paper, we proposed a layout-aware and coverage-driven design methodology for ILS architecture. The technique achieves significant improvements in fault coverage while at same time reduces the scan wire length to a great extent. The proposed methodology is also suitable for a highly compact test set, i.e., when the flip-flops have very weak or no compatibility.

TABLE IV
POWER DISSIPATION FOR CASE-1 (C-1) AND CASE-2 (C-2)

Segments in ILS	$s13207(P_{wt})$		$s15850(P_{wt})$		$s35932(P_{wt})$	
	(C-1)	(C-2)	(C-1)	(C-2)	(C-1)	(C-2)
2	6.9	6.7	5.2	4.9	7.2	7.0
4	7.4	7.2	5.7	5.4	7.8	7.4
6	8.2	7.8	6.5	6.3	8.4	8.3
8	8.8	8.5	7.3	7.3	9.2	9.0

VI. ACKNOWLEDGMENTS

The work was partially funded by the Engineering and Physical Science Research Council ($EPSRC$), United Kingdom, under Grant No: $EP/G032904/1$ and $EP/F030991/1$.

REFERENCES

[1] N. Nicolici and B.M. Al-Hashimi, "Multiple scan chains for power minimization during test application in sequential circuits," IEEE Transactions on Computers, June 2002, vol. 51, pp. 721 – 734.
[2] K-J. Lee, J-J. Chen, and C-H. Huang, "Broadcasting test patterns to multiple circuits," IEEE Transactions on TCAD, Dec 1999, vol. 18, pp. 1793 – 1802.
[3] A. Chandra, F. Ng, and R. Kapur, "Low Power Illinois Scan Architecture for Simultaneous Power and Test Data Volume Reduction," Proc. DATE, 10-14 March 2008, pp. 462 – 467.
[4] A.R. Pandey and J.H. Patel, "Reconfiguration technique for reducing test time and test data volume in Illinois Scan Architecture based designs," Proc. IEEE VTS, 28 April-2 May 2002, pp. 9 – 15.
[5] A.R. Pandey and J.H. Patel, "An incremental algorithm for test generation in Illinois scan architecture based designs," Proc. DATE, 2002, pp. 368 – 375.
[6] S. Banerjee, et al., "Layout-aware illinois scan design for high fault coverage coverage," Proc. ISQED, 2010, pp. 683–688.
[7] Douglas B. West, "Introduction to Graph Theory," Prentice-Hall of India, New Delhi, 2005.
[8] Y. Bonhomme, et al., "Power driven chaining of flip-flops in scan architectures," Proc. ITC, 7-10 Oct. 2002, pp. 796 – 803.