Partial Scan Design Based on Levelized Combinational Structure

Sung-Ju Park

Abstract

To overcome the large hardware overhead attendant in the full scan design, the concept of partial scan design has emerged with the virtue of less area and testability close to full scan. Combinational Structure has been developed to avoid the use of sequential test generator. But the patterns shifted on scan register have to be held for sequential depth period upon the aid of the dedicated HOLD circuit. In this paper, a new levelized structure is introduced aiming to exclude the need of extra HOLD circuit. The time to stimulate each scan latch is uniquely determined on this structure, hence each test pattern can be applied by scan shifting and then pulsing a system clock like the full scan but with much less scan flip-flops. Experimental results show that some sequential circuits are levelized by just scanning self-loop flip-flops.

I. Introduction

The difficulty of sequential test generation can be alleviated at the expense of full scan design. By considering all outputs and inputs of Flip-Flops(FF) as controllable and observable pseudo inputs and outputs, the test generation problem of sequential circuits is viewed as one in a combinational circuit. However this full scan design requires extra area overhead and performance degradation. Various partial scan design methodologies have been suggested [1, 4, 5, 6, 7, 8, 10]. In [5], a set of FFs is chosen guided by different testability measures in order to avoid the hard to observe and control FFs. Targeting for the faults escaped from functional level test generator, they enumerate all the FFs needed for each fault which is also escaped from the sequential test generator. This method is almost optimum, but another NP type covering problem must be solved and needs expensive sequential test generators. Quite a few graph theoretical approaches have been introduced [7, 8, 9, 10]. Initiated by the observation of the effects of feedback cycles in sequential test generator [5], different algorithms have been suggested to remove sets of cycles in different graph models. Minimal feedback vertex sets have been found using various heuristics [5, 9, 10].

In [7, 8] it is shown that if all pairs of vertices have equi-distance (equivalent distance) paths between them, test generation can be done by ignoring FFs and just applying a combinational test generator. We will call this structure as *combinational structure*. It has been observed that any fault detected using a sequential test generator can also be detected in this combinational structure. Hence test generation for the sequential circuit can be simplified to a combinational problem. Sequential Depth(*SD*) is defined as the number of non-scan FFs on the longest path between primary inputs (or scan FFs) and primary outputs (or scan FFs). The test application procedure of the patterns generated by the combinational Automatic Test Pattern Generator (ATPG) consists of scan shifting followed by pulsing SD number of system clocks. Suppose the Scan-path Length of a partial scanned circuit is *SL* and total *N* test patterns are required to test the circuit. Then the test application time for this circuit is $N \cdot (SL + SD)$. The levelized combinational structure proposed in this paper will reduce the test application time to $N \cdot (SL + 1)$. Furthermore the levelized structrue dose not require any disabling circuit on every scan FF which is needed for the eqi-distance combinational structure to hold the scanned data for *SD* system clocks[7].

This paper is organized as follows: In section 2, the levelized combinational structure is formally described. The graph model and its algorithm are addressed in section 3 and a test application procedure is described in section 4. Experimental results are shown in section 5 followed by conclusions in section 6.

II. Levelized Combinational Structure

In this section we shall consider the problem of converting a sequential circuit into a combinational structure with the property that the value of a scan FF on the scan register is only required

Manuscript received August 23, 1996; accepted May 23, 1997.

S. J. Park is with Department of Computer Science & Engineering Hanyang University in Ansan Kyunggi-Do, Korea.

at a certain time frame. Thus the combinational structure requires a scan shifting followed by only a single system clock in applying test patterns like the full scan design. We shall refer to such a structure as a *Levelized Combinational* (*LC*) structure. *Equi-Distance* (*ED*) structure, where the number of FFs between fanout stem and reconverged FFs is the same, was proposed in [7, 8] with the characteristics that the faults detected by the sequential test generator are also detected by the combinational test generator. However there are two disadvantages on the ED structure. First, the test application time is proportional to both SD and SL. Hence as sequential depth is increased the application time can be very lengthy. Second, HOLD function [7] has to be added to each scan FF so that the scanned patterns can be kept without being overloaded while the system clocks are applied. The Fig. 1(a) shows a level sensitive scan FF and Fig. 1(b) represents a scan FF with HOLD mode.

To illustrate the LC concept, consider Fig. 2. In this figure, the leftmost two FFs constitute a scan-path and the dark squares represent non-scan FFs. The rightmost circles indicate primary outputs. Since both two circuits in Fig. 2 do not contain any reconverged FF, their structures are ED. Here we can generate 3 test patterns **(01, 10, 11)** using the combinational test generator just simplifying the FFs as lead connections since both of the circuits are ED structure.

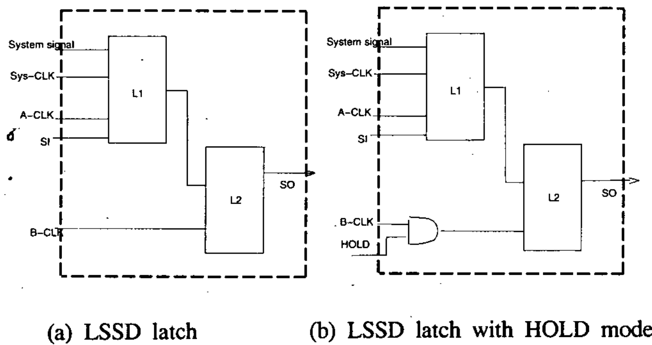
In Fig. 2(a), the test pattern for gate A stuck-at-1 and gate B stuck-at-1 faults is (0, 1) for the 1st scan FF(SFF1) and the 2nd scan FF(SFF2). However as shown in Fig. 3(a), the application time of the pattern for each fault is different. The time frame at which the SFF1 launches '0' is t2 for gate A s-a-1 while time frame of SFF1 for gate B s-a-1 is t0. Therefore scanned values (0, 1) on (SFF1, SFF2) have to be kept by holding the B-clocks on L2 latches during the SD period. On the other hand as shown in Fig. 3(b), the time frame at which the logic value on each scan FF has to be applieed to the internal logic circuit is the same regardless of the different tagret faults. For example the logic value of '1' on SFF2 are required only at t0 for both gate A and gate B stuck-at-1 in Fig. 3(b). Instead of scanning in all the values of a test pattern, in the LC structure only the values on the same level of SFFs need to be scanned in. The test application procedure for the LC structure can be summarized as follows:

- Scan in a test pattern on the scan FFs.
- Apply a system clock.
- Scan out the test response while scan in the next test pattern.

In the example in Fig. 2, the SL and SD of Fig. 2(a) are 2 and 4 respectively, thus $3 \cdot (2 + 4) = 18$ clocks are needed to apply 3 patterns. Although SD of Fig. 2(b) is 4, only $3 \cdot (2 + 1) = 9$ clocks are required for the LC structure independent of the SD.

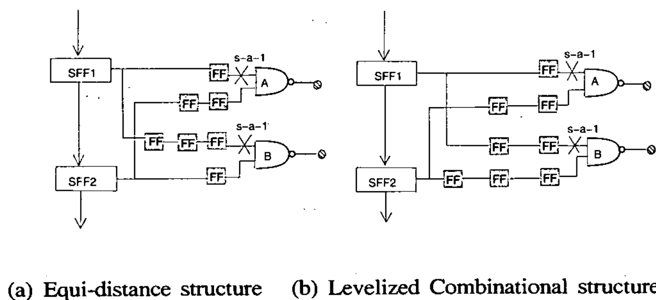
The main contributions of the LC structure compared with the ED structure are that HOLD type of extra circuit is not required and the test application time can be shortened.

In the next section, the graph model and algorithm to construct a LC structure are addressed.



(a) LSSD latch (b) LSSD latch with HOLD mode

Fig. 1. LSSD latch with and without HOLD mode.

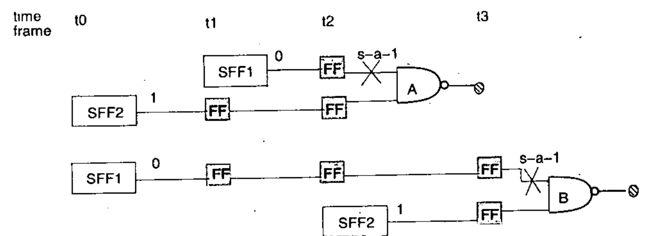


(a) Equi-distance structure (b) Levelized Combinational structure

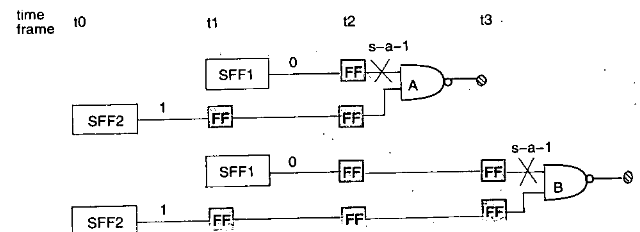
Fig. 2. Equi-distance and Levelized Combinational structure.

III. Graph Model and Algorithm

The circuit is modeled as a graph such that a *vertex* represents



(a) Two-separted drawings of ED structure



(b) Two-separate drawings of LC structure

Fig. 3. Drawing separated upon the target faults in ED and LC structures.

a Primary Input, Primary Output or FF and an *arc* describes the connection between vertices. The sequential circuit of Fig. 4 can be modeled as in Fig. 5. Only vertices representing FFs are considered to be scanned (changed into pseudo inputs and pseudo outputs). From now on all the *cycles* in the directed graph are implicitly considered as those cycles in the *undirected* U^* graph. (We shall refer to this undirected graph as U^*.) Hence, although there exists no cycle in a directed graph, some cycles can be found in the undirected U^* graph.

Definition 1 : A *fundamental* cycle in the *undirected* U^* graph is defined as a cycle for which no subset of vertices constitutes another cycle.

Definition 2 : A cycle chosen from the *undirected* U^* graph is *proper* in the *directed* graph if the number of outgoing arcs and the number of incoming arcs are the same. Otherwise the cycle is *improper.*

Definition 3 : A directed graph is said to be *levelized* combinational if and only if all the cycles in the sense of the *undirected* U^* graph are *proper.*

Definition 4 : A directed graph will be said to be leveled if it has been labeled with the integers 0, 1, 2, ··· such that for any edge from A to B, the label of A is one less than the label for B.

Theorem 1 : All cycles are proper if and only if all the fundamental cycles are proper.

Proof : The fundamental cycles constitute a basis for all cycles, thus any cycle can be generated by the combination of fundamental cycles. We will show that any cycle which consists of n fundamental cycles is proper if all fundamental cycles are proper. The induction basis is clear since a cycle with a fundamental cycle is proper by the statement. Assume that a cycle with n - 1 fundamental cycles is proper. By adding one more fundamental proper cycle to this cycle, the arc set overlapped is removed and new arc set nonoverlapped between them is added. Since both cycles are proper, there exists proper labeling for the new arc set between two vertices which are intersections of two cycles. Since new arcs are proper between two vertices, we can label properly for all the vertices of the new cycle which consists of n fundamental cycles. The *only if* part is obvious since the fundamental cycles are a subset of all the cycles.

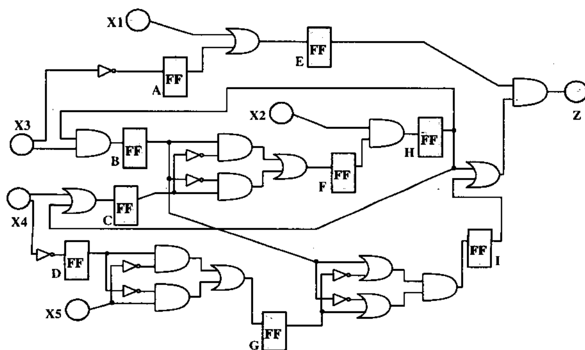


Fig. 4. A sequential circuit.

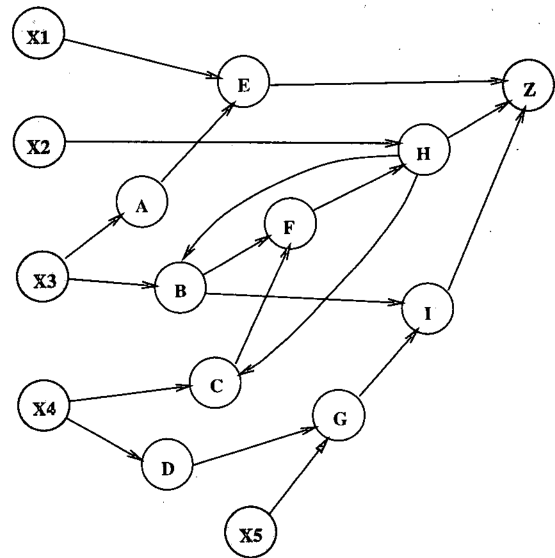


Fig. 5. Graph model.

Example 1 : 2 proper cycles are shown in Fig. 6. It can be seen that the new cycle generated by merging two cycles can be properly labeled.

Corollary 1 : A directed graph is levelized combinational if and only if all the fundamental cycles are proper.

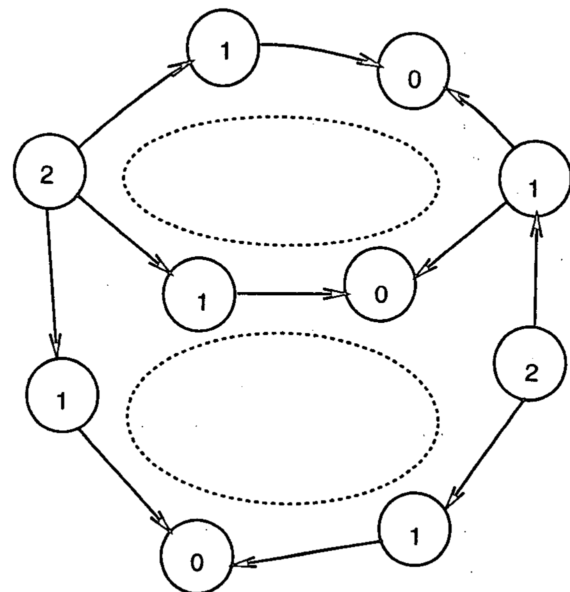


Fig. 6. Proper cycles labeled.

Proof : By the Definition 3 and Theorem 1.

The algorithm to generate all fundamental cycles can be described as following:

GENERATE_FUNDAMENTAL_CYCLES

Remove all the vertices with self-loop.

Repeat

Select an arc(F, T).
Remove this arc.

Find the shortest path between the two vertices F & T in the *undirected* U^* graph.

If this fundamental cycle comprising the arc(F, T) and shortest path is improper in the directed graph and unlisted, add it to the list.

Until (all arcs are visited)

Lemma 1 : The cycles generated by the above algorithm are not covered by other cycles in themselves. In other words, there exists no dominating relation between any pair of cycles.

Proof : Since the algorithm only enumerates improper fundamental cycles and equivalent cycles are avoided, no dominating relation exists between any pair of cycles.

Now all the improper fundamental cycles have been listed. We adopt a simple heuristic to choose minimal number of vertices whose removal results in the graph to be proper:

REMOVE_IMPROPER_CYCLES

Repeat

Remove the vertex which appears most frequently on the improper fundamental list.
Update the list by removing all the cycles containing the vertex removed.
Add new improper fundamental cycles to the list.

Until (No improper fundamental cycle can be found)

Theorem 2 : The graph can be leveled if and only if all cycles are proper.

Proof : Given a graph in which all cycles are proper, the labeling process is quite simple. We select any vertex, (say V), and label it with (say) l. Now all vertices with arcs into V are labeled with $l - 1$ and all with arcs from V with $l + 1$. We then process these newly labeled vertices in a similar manner. The only time that a problem might occur is when we label a vertex (say A) with $L(A)$ and note that there is an arc from A to a vertex B which is already labeled. How can we be sure that $L(B)$ will be at $L(A) + 1$? From the fact that B is already labeled, we know that there must be another (undirected) path, P, from A to B which together with the arc from A to B constitutes an undirected cycle. By definition this cycle is proper. Hence if we traverse this cycle from A first along P and then back to A from B, we will find that the number of incoming arcs which we traverse (say m) will exactly equal to the number of outgoing arcs. Thus $L(A)$ will be incremented (+ 1) m times and decremented (- 1) also m times. Hence when we return to A its label will again be $L(A)$. But this can only happen if $L(B) = L(A) + 1$.

Note that when we complete this labeling process the smallest label, l_0, will probably not be 0 and may in fact be a negative integer. However, if we now subtract l_0 from *all* labels then the graph becomes *leveled* as required.

Necessary condition easily follows the fact that no improper cycle can be leveled.

Let us precisely describe the procedures to change the Fig. 4 circuit into LC structure. Three simple branches from Fig. 5 are excluded in Fig. 7 and five improper fundamental cycles are generated as Table 1. For example, by applying the GENERATE_FUNDAMENTAL_CYCLE, the improper fundamental cycles for the edge *FH* are generated as *BFH* and *CFH*. Upon the REMOVE_IMPROPER_CYCLE algorithm, the vertex *H* is chosen as the best *FF*. After scanning the *FF*, *H*, as shown in Fig. 7, total three improper cycles are generated as Table 2. Then the *FF B* is scanned and Fig. 7 is changed to Fig. 8. By repeating the process, finally five vertices, *H*, *B*, *C*, *E*, *I*, are chosen to generate the LC structure of Fig. 9. The scanned FFs appear as shaded circles splitted into pseudo outputs(left side) and pseudo inputs(right side) and all the vertices are properly labeled, that is, all the starting vertices of the edges are labeled as one higher number than the

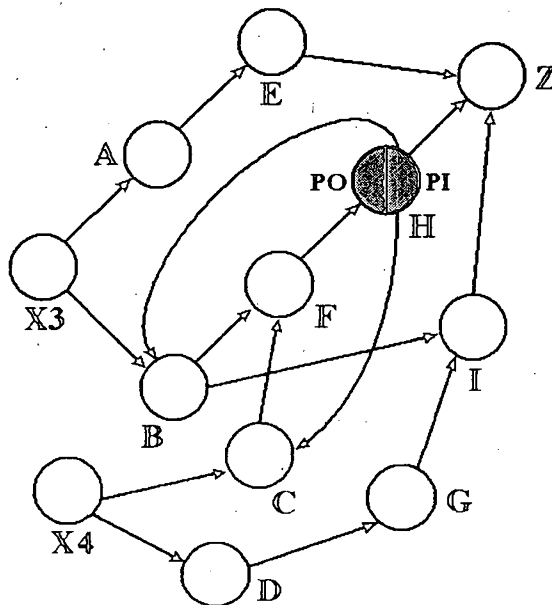


Fig. 7. Change into levelized combinational structure: step 1.

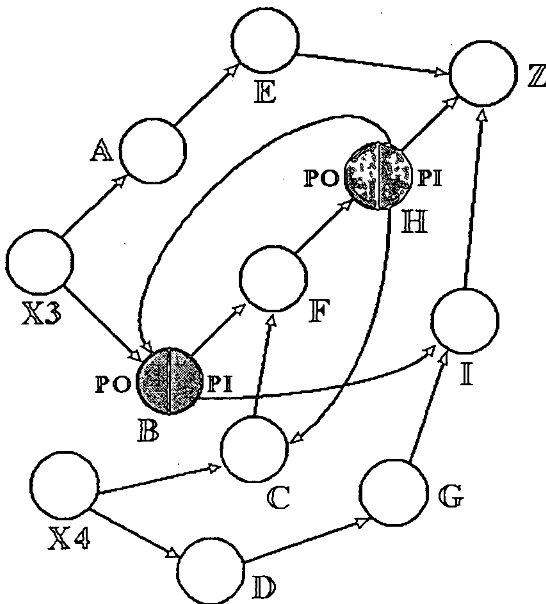


Fig. 8. Change into levelized combinational structure: step 2.

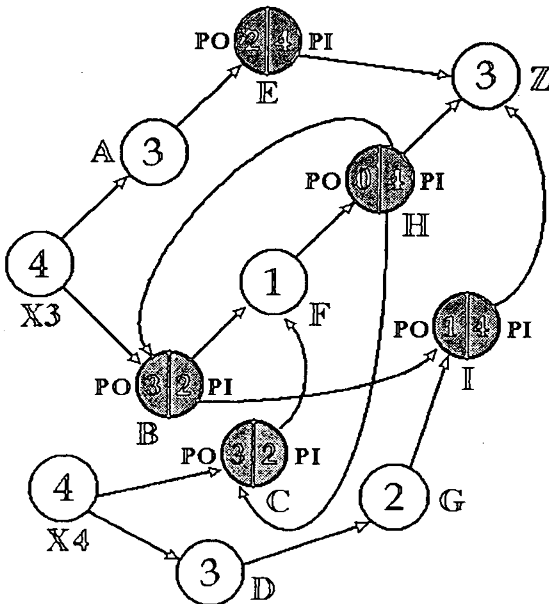


Fig. 9. Change into levelized combinational structure.

ending vertices. The labeling process can be summarized as following:

- Take any output and label it as number 0.
- Label all the adjacent vertices driving this output as 1.
- Decrease the label by one to the vertices driven by the current vertex.
- Increase the label by one to the vertices driving the current vertex.
- Keep labeling until all are labeled.

Table 1. Improper fundamental cycles: step 1.

Cycle No.	Vertices of Improper Cycle
Cycle 1	X3 A E Z H B
Cycle 2	B H F
Cycle 3	C F H
Cycle 4	B I Z H
Cycle 5	X4 C H Z I G D

Table 2. Improper fundamental cycles: step 2.

Cycle No.	Vertices of Improper Cycle
Cycle 1	X3 B H Z E A
Cycle 2	H Z I B
Cycle 3	H Z I G D X4 C

- If the smallest label is negative, add that absolute amount to all labels so that the smallest becomes 0.

More precise analysis for the application of test patterns is addressed in the next section.

IV. Application of Test Vectors

Scan techniques can be classified into an Edge Sensitive Scan Design(ESSD) and a Level Sensitive Scan Design(LSSD). The data is fetched on the rising or falling edge of the clock in the ESSD. On the other hand in the LSSD, the data is fetched while the clock is on. In this paper we only consider the LSSD but the same method can be applied to the ESSD. The application of test patterns in the Level Sensitive Scan Design is performed in the following way:

1. Apply scan (A and B) clocks to scan in a test pattern.
2. Stimulate PI values (off the system clock).
3. Measure the PO values.
4. Capture the internal signal value on the L1 latch by a system clock.
5. Apply the B clock, and then A and B clocks to shift out the scan data.

Example 2 : The sequential circuit shown in Fig. 4 is changed into a LC structure as in Fig. 9. Five out of nine FFs are scanned and a total of five patterns are generated to detect the stuck-at faults. It will take 30 clocks (5* [5 + 1] = 30 clocks: where 5 = number of test patterns, 5 = scan path length, 2 = sequential depth) are needed to apply the test patterns.

V. Experimental Results

Experimental results are shown in Table 3 and Table 4. In Table 3, the 4th column describes the number of FFs of each ISCAS89 sequential benchmark circuit [2] and the 5th column represents the total number of self-loops which also must be broken for the equi-distance structure, hence we will call this as the Lower Bound for the Equi-Distance Vertex Set. The cardinality of Levelized Combinational Vertex Set(LCVS) is shown in the 6th column and the percentage of the scanned FFs upon the total FFs is described in the last column. Some of the benchmark circuits, such as s3271 (fpunit), s3384(mult), s4863(design2) and s6669(bound), are taken from ISCAS89 ADDENDUM benchmarks [3] announced in 1993. The s953, s3384(mult), s4863(design2) and s5378 requires only 20.6%, 39.3%, 32.7% and 51.9% of scan FFs respectively compared with the full scan design.

Although some sequential circuits require almost all the FFs, in general we can see the large difference between LCVS and total number of FFs needed for full scan design. It may be noted that the sequential circuits become almost LCVS just by scanning the self- loop FFs. Table 4 shows the fault coverages of non scan, LC scan, and full scan designs. HITEC sequential test generator developed from the University of Illinois has been used to generate test patterns for each circuits. The CPU times spent for the test pattern generation on the SUN Sparc 5 were also described in the table. The fault coverage of the LC scan was very close to the full scan but s9234 and s5378.

VI. Conclusions

To simplify the test generation and test application problem frequently arising from the partial scan design, a new *levelized combinational* structure has been developed. It completely changes the sequential problem into combinational problem like the full scan design. Unlike the equi-distance structure, the system clock does not have to be pulsed for the sequential depth period for each scanned test pattern and thus the HOLD circuit for each scan FF is not required. It has been experimentally observed that some sequential circuits could be levelized by just scanning self-loop FFs. This partial scan design would be especially well suited for the chip with regular structure.

Table 3. Cardinality of levelized combinational vertex set.

Circuit	PIs	POs	FFs	LB	LCVS	% (LCVS/FFs)
s444	3	6	21	15	15	71.4
s713	35	23	19	15	15	78.9
s953	16	23	29	6	6	20.6
s1238	14	14	18	0	16	88.8
s1423	17	5	74	71	72	97.2
s3271	26	14	116	111	111	95.6
s3384	43	26	183	72	72	39.3
s4863	49	16	104	34	34	32.7
s5378	35	49	179	0	93	51.9
s6669	83	55	239	164	164	68.6
s9234	19	22	228	150	217	95.2

Table 4. Fault coverages on no scan, LC scan, and full scan.

Circuit	No Scan(%)	sec	LC Scan(%)	sec	Full Scan(%)	sec
s444	53.59	0.27	97.05	0.18	97.05	0.28
s713	81.93	0.35	93.46	0.18	93.46	0.27
s953	82.48	0.18	100.0	0.47	100.0	0.37
s1238	94.69	0.68	94.90	0.85	94.91	0.57
s1423	38.15	0.35	99.08	0.57	99.08	0.57
s3271	97.83	1.68	100.0	1.03	100.0	0.88
s3384	88.85	0.82	99.85	1.00	100.0	0.97
s4863	95.19	1.20	96.07	1.08	100.0	1.17
s5378	68.35	1.23	70.15	1.93	99.13	1.93
s6669	99.00	1.30	99.38	1.43	100.0	1.47
s9234	0.45	0.43	19.12	0.78	93.47	2.93

Acknowledgement

This research was supported by the Korea Science and Engineering Foundation under contract number KOSEF 95-0100-29-3.

References

[1] M. Abramovici, J. J. Kulikowski and R. K. Roy, "The Best Flip-Flops to Scan", Proceedings, IEEE Int'l Test Conf., pp. 116-173, Oct. 1991.

[2] F. Brglez, D. Bryan and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits", Procedding, IEEE International Symposium on Circuits and Systems, pp. 1929-1934, May 1989.

[3] C. Gloster Jr., "Dynamic Scan Testing: Investigation A new Paradigm", Technical Report TR93-06, MCNC.

[4] S. T. Chakradhar, A. Balakrishnam and V. D. Agrawal, "An Exact Algorithm for Selecting Partial Scan Flip Flops", Proceeding, DAC, pp. 81-86, 1994.

[5] K. T. Cheng and V. D. Agrawal, "An Economical Scan Design for Sequential Logic Test Generation", Proceedings, Symposium on Fault-Tolerant Computing, pp. 28-35, June 1989.

[6] V. Chickermane and J. H. Patel, "An Optimization Based Approach to the Partial Scan Design problem", Proceedings, IEEE Int'l Test Conf., pp. 377-386, Oct. 1990.

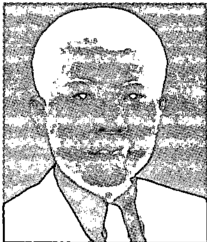
[7] R. Gupta and M. A. Breuer, "BALLAST: A Methodology for partial Scan Design", Digest of Papers of the 9th Int'l Symposium on Fault-Tolerant Computing, pp. 118-125, June 1989.

[8] A. Kunzmann and H. J. Wunderlich, "An Analytical Approach to the Partial Scan Problem", Journal of Electronic Testing: Theory and Applications, pp. 163-174, Jan. 1990.

[9] D. H. Lee and S. M. Reddy, "On Determining Scan Flip-Flops in Partial Scan Designs", Proceedings, ICCAD, pp. 322-325, Nov. 1990.

[10] S. J. Park and S. B. Akers, "A Graph Theoretic Partial Scan Design By K-Cycle Elimination", Proceedings, IEEE Int'l Test Conf., pp. 303-311, Sept. 1992.

[11] D. Xiang, S. Venkataraman, and W. K. Fuchs, "Partial Scan Design Based on Circuit State Information", Proceedings, DAC, pp. 807-812, June 1966.



Sung-Ju Park received the B.S. in Electronics from Hanyang Univ. in 1983 and the M.S. and Ph.D. in ECE dept. from Univ. of Massachusetts at Amherst in 1988 and 1992 respectively. He was with Gold Star from 1983 to 1986 and worked for the IBM Microelectronics, Endicott, NY as a development staff from 1992 to 1995. Since March 1995, he has been with the computer science dept. in Hanyang Univ. as an assistant professor. His research interests include Scan Design, BIST, ATPG, Test Synthesis, design verification, ASIC design, and graph theory.