On Flip-Flop Selection for Multi-Cycle Scan Test with Partial Observation in Logic BIST

Shigeyuki Oshima, Takaaki Kato,
Yasuo Sato and Seiji Kajihara
Kyushu Institute of Technology, Iizuka, Japan
{oshima,kato,sato,kajihara}@aries30.cse.kyutech.ac.jp

Senling Wang
Ehime University, Matsuyama, Japan
wang@cs.ehime-u.ac.jp

***Abstract*— Multi-cycle test with partial observation for scan-based logic BIST is known as one of effective methods to improve fault coverage without increase of test time. In the method, the selection of flip-flops for partial observation is critical to achieve high fault coverage with small area overhead. This paper proposes a selection method under the limitation to a number of flip-flops. The method consists of structural analysis of CUT and logic simulation of test vectors, therefore, it provides an easy implementation and a good scalability. Experimental results on benchmark circuits show that the method obtains higher fault coverage with less area overhead than the original method. Also the relation between the number of selected flip-flops and fault coverage is investigated.**

Keywords— logic BIST, scan test, multi-cycle test, partial observation.

I. INTRODUCTION

Scan test and BIST (Built-In Self-Test) are commonly used as DFT (Design for Testability) of logic circuits [1-4]. Especially, the scan-based logic BIST is becoming indispensable for system test or field test [5]. Although its test patterns are generated by an on-chip random test pattern generator such as LFSR (Linear Feedback Shift Register), it is hard to achieve high fault coverage. Many techniques have been proposed to improve the fault coverage of logic BIST [6-9]. Test point insertion is a technique to insert control/observe points into a CUT (Circuit-Under-Test) where hard-to-detect faults exist [6]. However, the amount of area overhead is not negligible because the test points are composed of several gates such as multiplexers or flip-flops. Re-seeding technology uses multiple seeds of a LFSR to avoid the dependency of test patterns on the initial seed [7]. For the problem of low fault coverage by the random pattern test, a method [9] dealt with it with multi-cycle captures. Multi-cycle test is a method that performs capture operations more than twice between scan-in and scan-out operations. It was originally developed in partial scan test [8], and applied to logic BIST in [10]. It was shown that multi-cycle test can improve fault coverage with a little increase in test time [11-13]. Moreover, it was proved to be effective for capture power reduction in scan test [12-14]. During the multiple captures the circuit logical state gradually approaches a functional one, therefore its power level gets closer to the functional level which enables a low power at-speed testing for small delay faults.

An issue in the logic BIST with multi-cycle test is that the detectable faults at the early stage of multiple captures might be undetected in the subsequent captures because the faulty values are masked and disappear before reaching the pseudo-primary outputs or the primary outputs. To address this issue, a partial observation technique that observes a part of the FF (flip-flop) values during capture operation was proposed in [10].

Since partial observation requires additional DFT circuits which raises concern of area overhead. The selection of FFs for partial observation causes a trade-off between the fault coverage and the area overhead, however, only a few researches are addressing this problem [10][15]. In [10], a method that utilizes a testability measure (SCOAP [16]) for selection of FFs was proposed. Since SCOAP is the measure developed to make deterministic ATPG be efficient, it may not be an optimal measure for the test point insertion of logic BIST. Furthermore, 20% of FFs are selected for the partial observation in [10], and its area overhead (2%) may not be negligible for an area-critical design. In [15], the authors analyzed the mechanism of fault masking during multiple captures and proposed a method to select the Fault Detection-Strengthened FFs (FDS-FFs) which are able to gather more faulty values before the faulty effects vanish (are masked) in the subsequent capture cycles, by comprehensively evaluating various metrics related to the circuit structure. Although the FDS-FFs method achieved good fault coverage improvement, it did not consider the case that different selected FFs may detect the same fault sets, resulting in an increase of number of selected FFs.

In this paper, we focus on achieving the best trade-off between the fault coverage improvement and the area overhead reduction, and propose a novel method of FF selection to maximize the fault coverage under the limitation of the given number of FFs used for partial observation. The proposed method statically analyzes the netlist of the circuit and extracts a set of combinational gates for each FF, where the logic value of gates may propagate to the FF. Then, the given number of FFs, which cover the gates as many as possible, are selected as the partial observation candidates. The experimental results show that the proposed method can derive better results than the original method [10]. The average fault coverage of benchmark circuits is improved by 1.35% at 2.5% of partial observation rate using the proposed method. Furthermore, the area overhead is reduced from the original 2.0% (reported in [10]) to only 0.25% for gaining the same level of fault coverage.

This paper is organized as follows. Section II describes the previous work relating the multi-cycle test. Section III

introduces the detail of the proposed selection method of FFs for partial observation. Section IV shows the experimental simulation results of the fault coverage and the area overhead on benchmark circuits. Section V concludes the paper.

II. PREVIOUS WORK

A. Multi-Cycle Test

Multi-cycle test is a scan test method to improve the fault coverage while suppressing the increase of test time [11-13]. Multi-cycle test performs capture operation more than twice between scan-in and scan-out operations. Figure 1 describes the behavior of test clock signal in a scan test. Let *M* be the number of clock cycles in the capture mode. For the detection of the stuck-at faults (typical static faults), *M* is equal to 1. For the detection of the transition delay faults (typical dynamic faults), *M* is set to 2. Multi-cycle test applies capture clocks for CUT *M*-times at the system clock rate (at-speed), and then scans-out the FF values captured at the last capture clock. Assuming that *M* is at most 20, it is far smaller than the typical scan chain length (usually several hundreds), which corresponds to the number of clock cycles in the scan-shift mode. Therefore, as for test application time, it does not increase so much when multi-cycle test is employed instead of the conventional tests. As for the fault coverage in multi-cycle test, the opportunity of fault excitation increases because of the extra capture clocks. It may increase fault coverage and may reduce the number of test patterns for achieving a specified fault coverage, which means it can shorten the total test application time. However, the opportunity of fault masking on fault propagation paths also increases and it becomes a risk of fault coverage loss. Any method to prevent this fault masking is required.

B. Multi-Cycle Test with Partial Observation

To avoid the fault masking in multi-cycle test, Partial Observation during the capture mode was introduced in [10]. Figure 2 shows the scheme of multi-cycle test with partial observation. The output of the scan-chain is connected to a response compactor (generally a MISR) in a conventional scan-based logic BIST, and the outputs of a part of FFs (denoted by shadowed red triangle) are directly connected to an additional response compactor which is typically composed of a space compactor and a MISR. Here, the random pattern generator can be any generator such as a simple LFSR or a LFSR with a phase shifter.

The response captured in all FFs at the last capture will be loaded into the response compactor through the scan chain. In addition, the values of the part of FFs will be loaded into the additional response compactor at each capture cycle. In the example of Figure 2, while values of FF1 and FF3 during the capture mode are observed by the response compactor, the value of FF2 is not observed. In this way, the faulty effect captured by FF1 or FF3 at any capture clock can be detected, and the fault coverage loss caused by fault masking is relaxed. However, since the area overhead due to the additional response compactor is almost proportional to the number of FFs used in partial observation [10], it is necessary to carefully select the FFs which really contribute the fault coverage improvement.

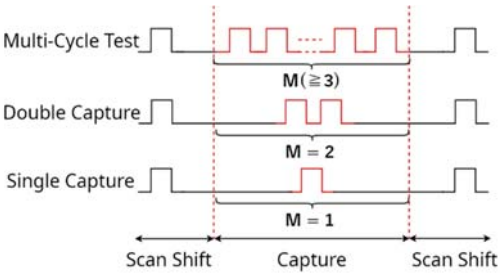

Fig.1 Overview of multi-cycle test

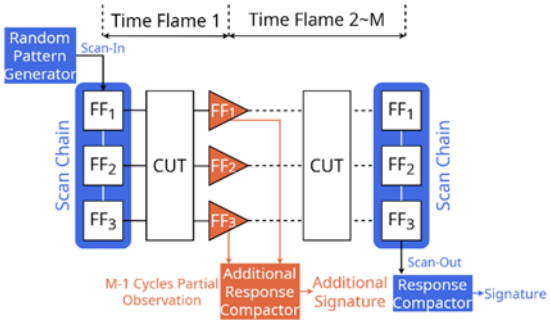

Fig 2. Example of partial observation

Depending on the FFs selected for partial observation, the detected faults (fault coverage) are different. In [10], the authors proposed a method to select the FFs for partial observation using the SCOAP testability measure, we call it S-method in this paper. According to their experiments, 20% of FFs in the CUT are selected for partial observation which achieved significant fault coverage improvement compared to the normal multi-cycle test, however, the effect was insignificant compared with the random selection method. Since SCOAP is the measure developed to make deterministic ATPG be efficient, it may not be an optimal solution for test point insertion of logic BIST necessarily. Moreover, 20% of FFs used for partial observation would be so large that it could not be allowed for industrial design. Because the ratio of FFs for partial observation determines the area overhead, it is needed to investigate the effect of partial observation with the smaller ratio.

In this paper, we focus on achieving the best trade-off between the fault coverage improvement and the area overhead increase for partial observation, and propose a novel method of FF selection to maximize the fault coverage under the limitation of a given number of FFs used for partial observation, which is described in the following section.

III. THE PROPOSE METHOD

A. FF selection based on input cone

The S-method selects FFs with low observability. For the increase of fault coverage by partial observation, many faults should be detected at selected FFs, and the faults to be detected at the selected flip-flops should be ones difficult to detect at unselected flip-flops. Therefore, we propose a novel FF selection method named "Non-Overlapping (NO) Method", which focuses on the number of faults included in the input cone of each FF. Note that the input cone of an FF is defined as a fan-out free sub-circuit consisting of logic gates and signal lines from which there is only one reachable

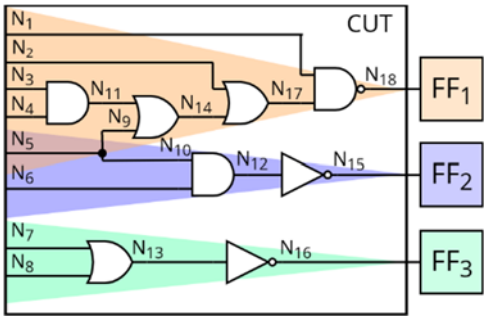

Fig 3. Example of input cones of FFs

path to the FF without passing through any other FFs. In other word, the input cone of an FF is a combinational circuit such that the output of the circuit is the input of the FF, and every input of the circuit is a primary input or a pseudo primary input. All faults in the input cone of an FF must be propagated to the FF for its detection. The larger the size of the input cone is, the more the number of detected faults by observing the FF will be. While the NO-method is constructed from this idea, we also propose an alternative method named "Capture-Toggle & Non-Overlapping Method" that considers not only the number of faults in the input cone but also the toggle count at each FF.

Figure 3 shows an example to explain the definition of input cones. Each FF is labeled as FF_i and each line is labeled as N_j where i and j is an index of FFs and lines, respectively. Let S_i be a set of lines included in the input cone of FF_i. For example, $S_1 = \{N_1, N_2, N_3, N_4, N_5, N_9, N_{11}, N_{14}, N_{17}, N_{18}\}$. In the same way, the input cone of FF_2 and FF_3 are $S_2 = \{N_5, N_6, N_{10}, N_{12}, N_{15}\}$ and S3 = $\{N_7, N_8, N_{13}, N_{16}\}$, respectively.

B. Non-Overlapping method

If FFs for partial observation are selected according to the size of input cone, i.e., the cardinality of S_i denoted by $|S_i|$, high fault coverage is expected. However, a part of an input cone may be overlapped with other input cones. It is sufficient that a fault is detected at one FF, and it is not necessary to detect it at two or more FFs. Therefore, it may be meaningless for improvement of fault coverage if two FFs of which the input cones have large overlapped area each other are selected for partial observation. NO-method selects FF for partial observation such that input cones of the selected FFs cover the circuit as large as possible. Suppose that the number of FFs in the circuit is n and the number of FFs to be selected is m. The procedure of the NO-method is given as follows:

Procedure: NO-method

Step1. Set a set of selected FFs $SFF = \varnothing$.

Step2. For every FF_i ($1 \le i \le n$), calculate S_i.

Step3. Sort all FFs in the decreasing order of $|S_i|$.

Step4. For i = 1 to m,

Select $FF_j \notin SFF$ such that the number of lines included in the input cones of FFs in $\{SFF \cup FF_j\}$ is the largest.

Add FF_j to SFF.

Step5. Flip-flops in SFF are used for partial observation.

In case of Figure 4, all FFs are sorted in the order of the size of input cones, FF_1 is added to SFF firstly because $|S_1|$

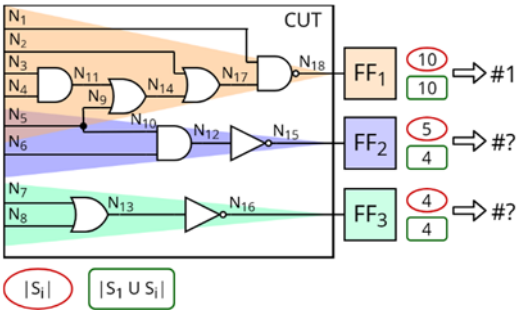

Fig 4. Concept of Non-Overlapping method

is the largest among the three FFs. Then, either FF_2 or FF_3 can be selected because both $|S_1 \cup S_2|$ and $|S_1 \cup S_3|$, which equal to 4, are same.

C. CaptureToggle & Non-Overlapping method

In the NO-method proposed in Section 3.B, when there are more than one FF with the largest number of lines at Step 4, the FF to be selected cannot be determined uniquely as described in Figure 4. In order to make the NO-method deterministically, we propose Capture-Toggle + Non-Overlapping method (C-NO method) to enable the ranking of all FFs based on the NO-method by considering multiple metrics as shown in Figure 5. Here, the Capture-Toggle count of a FF is defined as the number of toggles at the FF during the multiple captures in the capture mode. In the multi-cycle test, it is empirically known that the logic propagation converges and the number of toggle at a capture decreases as the number of captures increases. Then, when the count of a FF is large, the faults in its area are more likely to be detected at the FF than those in the other areas. The number of Capture-Toggle at each FF can be calculated by logic simulation.

In the C-NO method, the number of lines in the input cone $|S_i|$ is used as the first metric and the number of Capture-Toggle is used for the second metric to select an FF at Step 4 in the procedure of NO-method. An example is given in Table I. Suppose that we select two FFs from three. Because S_1 has the largest number of lines among S_1, S_2 and S_3, FF_1 is selected first. Next, since $|S_1 \cup S_2|$ is the same as

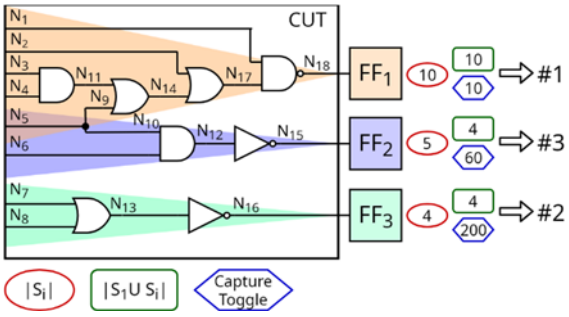

Fig 5. Concept of C-NO method

TABLE I. EVALUATION ELEMENTS OF EACH FF

Flip-flop	First metric: S_i	Second metric: Capture-Toggle	Priority
FF_1	$N_1, N_2, N_3, N_4, N_5, N_9, N_{11}, N_{14}, N_{17}, N_{18}$	40	1
FF_2	$N_5, N_6, N_{10}, N_{12}, N_{15}$	60	3
FF_3	N_7, N_8, N_{13}, N_{16}	200	2

$|S_1 \cup S_3|$, we check the Capture-Toggle number of FF_2 and FF_3. The Capture-Toggle of FF_3 is larger than that of FF_2, we select FF_3. Thus, we can select the FFs for partial observation in the order of FF_1, FF_3, FF_2.

IV. EVALUATION EXPERIMENT

A. Exprerimental conditions

We conducted experiments on five ISCAS'89 benchmark circuits and five ITC'99 benchmark circuits. In the experiments, 30k patterns were generated by 16 bit LFSR (characteristic polynomial: $X^{16}+X^{15}+X^{13}+X^4+1$), the maximum scan chain length was 100 (the number of FFs $\leqslant$ 1600) or 200 (the number of FFs > 1600), the number of capture cycles was set to 20, and the fault simulation was performed using the single stuck-at fault model. The S-method, NO-method and C-NO method are denoted by S, NO, and C-NO in this section, were implemented to select 20%, 10%, 5%, 2.5% and 1% of FFs for partial observations, respectively. Where, C-NO used the number of Capture-Toggle of each FFs by logic simulation under the same condition.

We also measured the computing time for S, NO and C-NO to select the 20% observation FFs each benchmark circuits by in-house simulation tool. For S-method, the number of multi-cycle count was set to 20. The computing time of C-NO included the logic simulation time for Capture-Toggle. The Experiment conditions are shown Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and Memory 32GB.

B. Experimental Result

Table II shows the fault coverage of each 10 benchmark circuits using S, N and C-NO. Note that because redundant faults are counted as undetected faults in the fault list, the maximum fault coverage of these circuits cannot be 100%. The first row shows the benchmark circuit name, the number of nodes and the number of FFs each circuit. The results of Table II denote that the NO and C-NO method achieved more fault coverage improvement than S-method at any partial observation rate, and the fault coverages of C-NO are better than those of S with a few exceptions. Furthermore, the average coverage of C-NO at 2.5% is higher than or equal to that of S-method at 20%. This means that C-NO can get the same fault coverage with 2.5% observation FFs whereas S-method do with 20%, and this is over 87.5% reduction of the response observation circuit which consists of XOR gates and MISR. For s13207, the fault coverage of the existing method is higher at the partial observation rate 1%, but the coverages of C-NO are higher than S-method at other rates. Therefore, C-NO is the most efficient method for various circuits. Especially, the fault coverage of s38417 at 1% shows a great improvement with NO and C-NO. Although NO and C-NO were effective for ISCAS'89 circuits, they didn't show much improvement for ITC'99 circuits. This might be because of large gate size in ITC'99, and the analysis is remaining for the future work.

Table III shows the number of necessary test patterns to achieve 90% fault coverage. When the fault coverage did not achieve 90% by 30k patterns, the results of fault coverage by 50k patterns are given. It can be judged that even if the number of test patterns are increased, C-NO is the most effective in improving the fault coverage among the three methods.

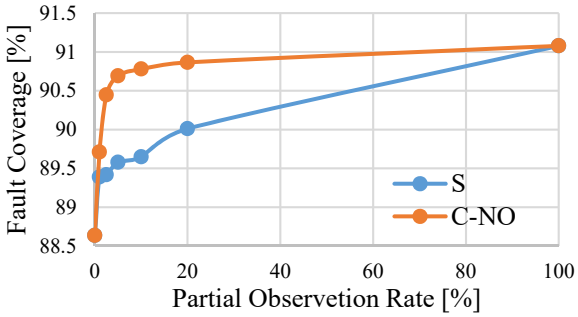

Fig 6. Average fault coverage

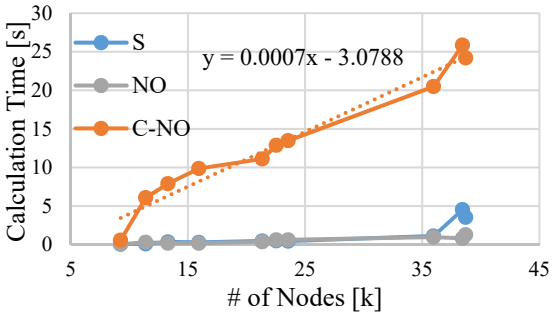

Fig 7. Computing time (20% observation FFs)

Figure 6 shows the average fault coverage of all circuits when increase the observation rate from 1% to 100%. In Figure 6, it can be seen that the fault coverage of C-NO is higher than or equal to the fault coverage of S method. From fault coverage results, S and C-NO per one observation FF improves the average fault coverage about 0.09% and 0.15% in partial observation rate 2.5%.

Figure 7 shows the computing time of each method to select the 20% observation FFs for all the ten benchmark circuits. The computing time of C-NO was the largest among the proposed methods because it needs the logic simulation time for calculating the Capture-Toggle for each FF. The computing time shows a proportional relationship to the number of nodes which is (# of Node) ×0.0007 from the results. In case of the 100 million nodes level design, we presume that C-NO of calculation time is under one day from the result. Thus, the proposed method can calculate within practical time for industrial designs.

C. Area Evaluation

In order to compare the area overhead [9], we use the same large data model with that used in [9]. The total number of gates is 17M, the number of transistors in FF is 26, the number of transistors in XOR is 12, the total number of FFs is 523 k, and the number of bits in MISR is 523 bits assuming 1% of the total number of FFs. Given the number of transistors required for XOR-tree is $G(\alpha)$, it is given by the following equation [9].

$$G(\alpha)=g\times(m\alpha - n)$$

Here, α is the partial observation rate, g is the number of XOR transistors, m is the number of FFs, and n is the

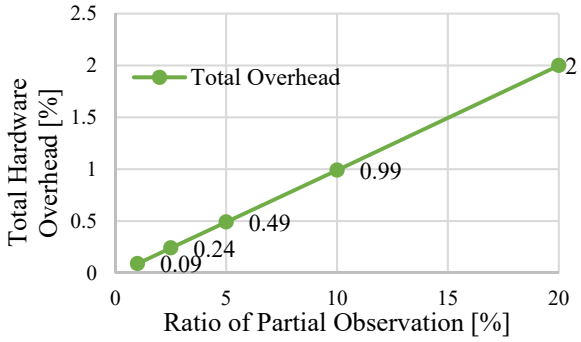

Fig 8. Estimated area overhead

number of bits of MISR. Fig. 8 shows the area overhead by the estimated partial observation.

Note the area overhead, the fault coverage is higher when using the proposed method at the partial observation rate of 2.5% than S-method when the partial observation rate is 20%. Therefore, it is possible to reduce FF required for partial observation by 87.5% or more, and the area overhead can be reduced from 2% to 0.24%.

V. CONCLUSIONS

In this paper, we proposed more efficient FF selection methods for partial observation with multi-cycle test in scan-based logic BIST. The proposed method (Capture-Toggle & Non-Overlapping) increased the average fault coverage of the benchmark circuits by 1.03% at partial observation rate 2.5%=compared with the existing method that used SCOAP. The hardware overhead is also reduced from 2.0% to 0.25% for gaining the same level of fault coverage.

By using the C-NO method, area overhead can be reduced for various circuits. For the ITC'99 circuit, the proposed Non-Overlapping method and the C-NO method showed almost no change in fault coverage compared with the existing method. The future work is to develop a method that can improve fault coverage independently of circuit characteristics and its scale.

REFERENCES

[1] M. Bushnell, V. Agrawal, *Essential of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*, Kluwer Academic Publisher, 2000.

[2] P. Girard, N. Nicolici, X. Wen, *Power-Aware Testing and Test Strategies for Low Power Devices*, Springer, 2009.

[3] S.Dikic, L.-J Fritz, D.Dell'Aquia, "BIST and Fault Insertion Re-use in Telecom Systems," *Proc. Int'l Test Conf.*, pp.1011-1016, 2001.

[4] F. Corno, M. Rebaudengo, M. S. Reorda, M. Violante, "A new BIST architecture for low power circuits," *Proc. European Test Workshop*, pp. 160-164, 1999.

[5] Y. Sato, S. Kajihara, Y. Miura, T. Yoneda, S. Ohtake, M. Inoue, H. Fujihara, "A Circuit Failure Prediction Mechanism (DART) for High Field Reliability", *Proc. Int'l Conf. on ASIC*, pp. 581-584, 2009.

[6] B. H. Seiss, P. M. Trouborst, M. H. Schulz, "Test Point Insertion for Scan-Based BIST", *Proc. of Euro. Test Conf.*, pp. 253-262, 1991..

[7] D. Xiang, X. Wen, L-T. Wang, "Low-Power Scan-Based Built-In Self-Test Based on Weighted Pseudorandom Test Pattern Generation and Reseeding," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VOL.25, pp. 942-953, 2016.

[8] C.-J. Lin, Y. Zorian, S. Bhawmik, "PSBIST: A Partial Scan Based Built-In Self Test Scheme", *Proc. Int'l Test Conf.*, 1993.

[9] S. Upadhyaya, L. Nachman, K. K. Saluja and R. Reuse, "A Novel Approach to Random Pattern Testing of Sequential Circuits," *IEEE Transactions on Computers*, vol. 47, no. , pp. 129-134, 1998.

[10] Y. Sato, H. Yamaguchi, M. Matsuzono, S. Kajihara, "Multi-Cycle Test with Partial Observation on Scan-Based BIST Structure," *Proc. Asian Test Symp.*, pp. 54-59, 2011.

[11] I. Pomeranz, S.M. Reddy, "Forming multi-cycle tests for delay faults by concatenating broadside tests," *Proc. VLSI Test Symp.*, pp. 51-56, 2010.

[12] J. Abraham, U. Goel, A. Kumar, "Multi-cycle sensitizable transition delay faults," *Proc. VLSI Test Symp.*, pp. 306-313, 2006.

[13] G. Bhargava, M. Gaurav, D. Meehl, J. Sage, "Achieving serendipitous N-detect mark-offs in multi-capture-clock scan patterns," *Proc. Int'l Test Conf.*, pp. 1-7, 2007.

[14] Y. Sato, S. Wang, T. Kato, K. Miyase, S. Kajihara, "Low Power BIST for Scan-Shift and Capture Power," *Proc. Asian Test Symp.*, pp. 173-178, 2012.

[15] S. Wang, H. T. Al-Awadhi, S. Hamada, Y. Higami, H. Takahashi, . Iwata and J. Matsushima, "Structure-Based Methods for Selecting Fault-Detection-Strengthened FF under Multi-Cycle Test with Sequential Observation," *Proc. Asian Test Symp.*, pp. 209-214, 2016.

[16] L. H. Goldstein, E. L. Thigpen, "SCOAP: Sandia Controllability/Observability Analysis Program," *Proc. Design Automation Conf.*, pp. 190-196, 1980.

TABLE II. FAULT COVERAGE FOR 30,000 PSEUDO RANDOM PATTERNS

Circuit				b14	b15	b20	b21	b22	s9234	s13207	s15850	s38417	s38584	**Ave.**
# of nodes				11400	21355	22555	23563	35972	9256	13300	15934	38445	38710	
# of FFs				245	449	490	490	735	228	669	597	1636	1452	
Method	*S*	*Partial observation rate*	*20%*	89.43	92.97	88.47	89.84	89.52	83.48	89.88	89.36	96.77	90.4	**90.01**
			10%	89.38	92.97	88.47	89.75	89.52	83.4	89.63	89.22	94.1	90.04	**89.65**
			5%	89.38	92.97	88.47	89.75	89.52	83.4	89.52	88.82	94.06	89.89	**89.58**
			2.5%	89.38	92.97	88.44	89.75	89.51	83.34	88.64	88.2	94.06	89.89	**89.42**
			1%	89.38	92.97	88.39	89.75	89.50	83.34	88.62	88.09	94.06	89.79	**89.34**
	NO		*20%*	89.43	92.99	88.57	89.98	89.62	89.23	91.13	89.13	97.39	91.11	**90.86**
			10%	89.43	92.98	88.54	89.93	89.61	89.23	90.98	88.8	97.37	90.85	**90.77**
			5%	89.43	92.97	88.50	89.91	89.60	89.23	90.63	88.63	97.25	90.53	**90.67**
			2.5%	89.43	92.97	88.47	89.89	89.59	89.14	85.28	88.63	97.05	90.07	**90.05**
			1%	89.43	92.97	88.47	89.87	89.59	87.37	84.84	88.00	96.99	89.76	**89.73**
	C-NO		*20%*	89.43	93.01	88.60	89.95	89.62	89.23	91.13	89.12	97.41	91.16	**90.87**
			10%	89.43	93.01	88.53	89.89	89.60	89.23	90.98	88.8	97.41	90.92	**90.78**
			5%	89.43	93.00	88.51	89.89	89.59	89.23	90.67	88.75	97.31	90.54	**90.69**
			2.5%	89.43	92.99	88.51	89.89	89.59	89.14	89.14	88.63	97.1	90.04	**90.45**
			1%	89.43	92.97	88.47	89.87	89.59	87.37	84.62	88.00	97.03	89.76	**89.71**
0%				89.38	92.97	88.37	89.75	89.49	83.34	82.07	87.30	94.06	89.64	**88.64**

TABLE III. NUMBER OF TEST PATTERNS FOR 90% FAULT COVERAGE OR FAULT COVERAGE FOR 50,000 PATTERNS

Circuit				b14	b15	b20	b21	b22	s9234	s13207	s15850	s38417	s38584
# of nodes				11400	21355	22555	23563	35972	9256	13300	15934	38445	38710
# of FFs				245	449	490	490	735	228	669	597	1636	1452
Method	*S*	*Partial observation rate*	*20%*	(89.69)	3,400	(88.93)	32,800	(89.92)	(84.32)	38700	(89.41)	900	16,700
			10%	(89.69)	3,400	(88.93)	35,200	(89.92)	(84.26)	(89.80)	(89.26)	5,400	27,400
			5%	(89.69)	3,400	(88.93)	35,200	(89.92)	(84.26)	(89.69)	(88.87)	5,400	39,100
			2.5%	(89.69)	3,400	(88.91)	35,200	(89.89)	(84.22)	(88.83)	(88.38)	5,400	39,100
			1%	(89.69)	3,400	(88.88)	35,200	(89.87)	(84.22)	(88.83)	(88.32)	5,400	(89.97)
	NO		*20%*	(89.69)	3,300	(89.15)	31,300	46300	(89.84)	11900	(89.25)	600	5,600
			10%	(89.69)	3,400	(89.14)	31,300	46300	(89.84)	12000	(89.01)	600	8,300
			5%	(89.69)	3,400	(89.10)	31,300	48700	(89.84)	14100	(88.84)	700	14,100
			2.5%	(89.69)	3,400	(89.05)	31,300	(89.96)	(89.78)	(85.66)	(88.84)	700	24,500
			1%	(89.69)	3,400	(89.05)	32,800	(89.96)	(88.29)	(85.23)	(88.19)	900	(89.94)
	C-NO		*20%*	(89.69)	3,000	(89.17)	31,300	46300	(89.84)	11900	(89.21)	600	5,300
			10%	(89.69)	3,000	(89.10)	31,300	48700	(89.84)	12000	(89.01)	600	8,000
			5%	(89.69)	3,300	(89.07)	31,300	(89.98)	(89.84)	13700	(88.96)	700	14,100
			2.5%	(89.69)	3,300	(89.07)	31,300	(89.96)	(89.78)	(89.47)	(88.84)	700	26,500
			1%	(89.69)	3,400	(89.05)	32,800	(89.96)	(88.29)	(85.01)	(88.19)	1,000	(89.94)
0%				(89.69)	3,400	(88.87)	(89.69)	(89.86)	(84.22)	(82.49)	(87.51)	5,400	(89.82)