Multi-Cycle Test with Partial Observation on Scan-Based BIST Structure

Yasuo Sato, Hisato Yamaguchi, Makoto Matsuzono, Seiji Kajihara
Department of Computer Science and Electronics
Kyushu Institute of Technology
Iizuka, 820-8502 Japan
JST CREST
Tokyo, 102-0075 Japan
{sato, yamaguchi, matsuzono, kajihara}@aries30.cse.kyutech.ac.jp

Abstract— **Field test for reliability is usually performed with small amount of memory resource, and it requires a new technique which might be somewhat different from the conventional manufacturing tests. This paper proposes a novel technique that improves fault coverage or reduces the number of test vectors that is needed for achieving the given fault coverage on scan-based BIST structure. We evaluate a multi-cycle test method that observes the values of partial flip-flops on a chip during capture-mode. The experimental result shows that the partial observation achieves fault coverage improvement with small hardware overhead than the full observation.**

Keywords- BIST; multi-cycle test; multiple observation; partial observation; scan-based BIST.

I. INTRODUCTION

Field test for reliability is becoming crucial for detecting unexpected failures due to aging or process variations. Especially, several aging mechanisms such as Hot Carrier Injection (HCI), Electro Migration (EM) and Negative Bias Temperature Instability (NBTI) are big concerns in deep sub-micron processes [1-2]. Some promising approaches that test a chip in testing mode are proposed [3-6]. However, these require a new technique which is somewhat different from the conventional manufacturing tests. Delay measurements, test environment monitoring, short time tests, and small memory-resource tests are mentioned in [6].

A small memory-resource test requires logic BIST techniques or test vector compression techniques that achieve high compression ratio. Many researches have done in this area, and the roadmap is shown in [7-8], which shows the potential solutions up to 67,000X compression using three technologies: test-cube compression, spatial compression, and time correlation compression. Although these suggest a great possible progress in future, the amount of on-chip or off-chip available memory size for a test purpose in an embedded system is often not so large to store compressed deterministic vectors for a large industrial chip because most part of memory is used for storing embedded software program codes. Logic BIST can be another candidate for field test. However, static or dynamic reseeding data is required for improving fault coverage [9]. In either approaches, more additional compression method is required.

The purpose of this paper is proposing a compression technique which consists with either test vector compression or logic BIST. It means that, if the original compression ratio is 500X and the additional compression ratio is 2X, then the total ratio will be improved to 1,000X. The proposed technique is based on multi-cycle test, which has multiple capture clocks during capture-mode between scan-in and scan-out for each test vector and a value of each flip-flop is observed at each capture cycle. In our earlier work [10], we confirmed up to 58 % test vector reduction rate (i.e. ~2X) for scan test with a full observation scheme. However, observing all of the flip-flops is not feasible from a viewpoint of hardware overhead. Therefore, this paper proposes a partial observation scheme for reducing hardware overhead, and the experimental result on scan-based BIST shows that it can achieve fault coverage improvement with small hardware overhead than the full observation.

This paper is organized as follows. Section II introduces the related works regarding multi-cycle test. Section III introduces the proposed multi-cycle test scheme. Section IV addresses the propose method to choose the partial flip-flops. Experimental results are discussed in Section V. Section VI concludes the paper.

II. RELATED WORKS

Fig. 1 shows the conventional capture clocks (Fig.1 (a): single capture clock for stuck-at-fault test, Fig.1 (b): double capture clocks for broad-side delay test [11]). Fig. 1 (c) shows the multi-cycle capture clock with more than 2 capture clocks. Let v be the first vector applied to a CUT from scan flip-flops and f be the function of the CUT, then fv is applied in the second capture clock cycle, $f(fv)= f^2v$ is applied in the third clock cycle, and in the same way $f^{n-1}v$ is applied in the nth clock cycle. Then, n test vectors ($v, fv, f^2v, , , f^{n-1}v$) are applied to the CUT in one test sequence. Assuming they are observable, it has promising potential to improve test coverage.

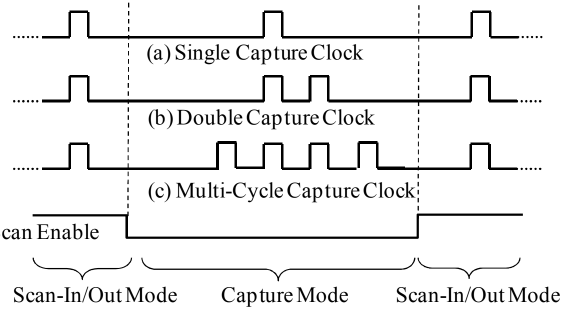


Figure 1. Multi-Capture Clocks.

Originally, multi-cycle test was used for partial scan test [12]. Unknown values of non-scan flip-flops are initialized with multiple capture clocks and the fault coverage is improved. However, a single capture clock is used for each scanned-in pattern. [13, 14] developed multiple test session schemes with different number of capture clocks using the BIST structure in [12]. They showed fault coverage improvement and/or test length reduction using the proposed schemes. A set of undetectable transition delay faults in the launch-off-capture scheme [11] are reported to be sensitizable with more than three capture cycles [15]. A test generation method treating the CUT as a sequential circuit is proposed in [16], and another test generation method avoiding sequential test generation is proposed in [17]. Further more; multi-cycle test is also known to generate more functional vectors than one or two cycle test. Paper [18] evaluated the Weighted Switching Activity (WSA) of a circuit applying 20 cycles and confirmed that WSA values are stabilized at rather low level.

However, as the values of flip-flops are observed only at the last capture clock in these methods, a faulty value might be masked before applying the last capture clock as shown in Fig. 2. For example, the faulty value of gate *A* is masked in the next time frame and it cannot be observed at the last capture. As the number of cycles becomes larger, such masking will be more possible. To tackle this problem, we proposed a technique that is based on a multi-cycle test, which has multiple capture clocks during capture-mode between scan-in and scan-out mode for each test vector and all of the values of flip-flops are observed at each capture cycle [10]. The observation of flip-flops during each capture cycle is done using XOR (Exclusive OR) based space compactor and Multiple Input Signature Register (MISR), which requires large hardware overhead. The number of test vectors was reduced 28 % reduction in average and 58 % reduction in maximum.

III. Proposed Multi-Cycle Scheme

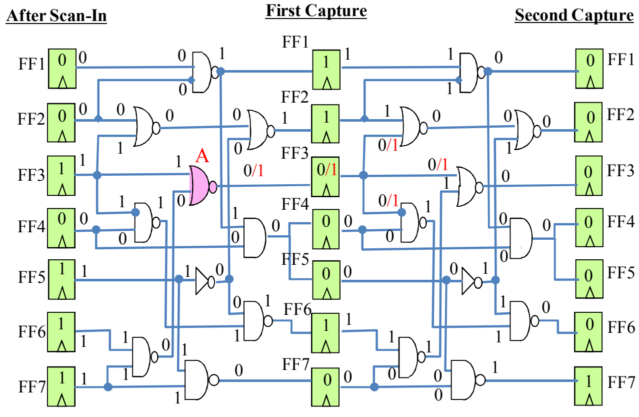
Fig. 3 shows the proposed multi-cycle test scheme with partial observation of flip-flops. Unlike the case of [10], we use scan-based logic BIST instead of full-scan test. This is because logic BIST with re-seeding will be more feasible for field test with small memory resource than compression-based deterministic full-scan test in the current technologies. In this scheme, we target to reduce the number of seeds that is needed for achieving the given fault coverage or to improve the fault coverage for the given number of seeds.

In the figure, input vectors to the combinational circuit under test (CUT) is provided to flip-flops (FFs) through scan chains from a test pattern generator (TPG), which can be a linear feedback signature register (LFSR) or a cellular Automata (CA) with re-seeding capability. The output values of CUT are captured into FFs at each clock cycle during test mode (capture mode). They are scanned out to Compactor *A*, which consists of an XOR based space compactor and a MISR. At the same time, a part of FFs are connected directly (i.e. without scan-out) to additional Compactor *B*, which also consists of an XOR based space compactor and a MISR. Note that we refer to the FFs connected to Compactor *B* as *partial FFs* in this paper. Primary inputs and outputs are isolated from the CUT using boundary scan cells in case of at-speed test. The figure is simplified to a single scan chain, but it can easily be enhanced to multiple scan chains.

Fig. 4 shows the test flow for multi-cycle logic BIST with partial observation. Let *N* be the number of capture clocks. A vector is scanned into FFs, the capture clock is applied *N* times with concurrent compression of the partial FF values with Compactor *B*, and the FF values are scanned out to Compactor *A* after the last capture. This process is repeated for number of vectors and number of seeds. At the end of successive tests, signatures in Compactor *A* and *B* are read and analyzed.

Using this scheme, lower hardware overhead will be achieved but fault coverage will be lower than [10]. Therefore, we set the following problems to investigate in.

- How do we choose the partial FFs to observe their values at Compactor *B*?
- What ratio of FFs is feasible to minimize the coverage loss?



Correct Value/Faulty Value

Figure 2. Fault Masking.

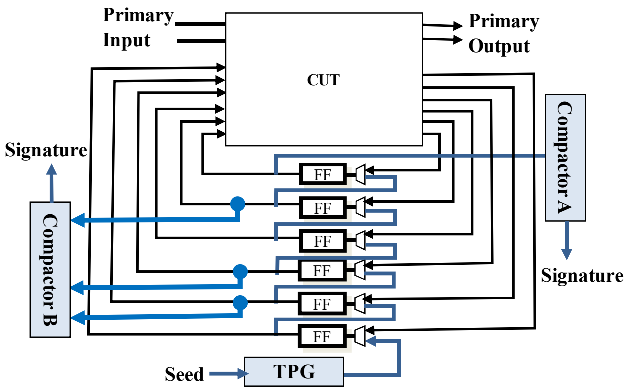


Figure 3. Multi-Cycle Test with Partial Observation

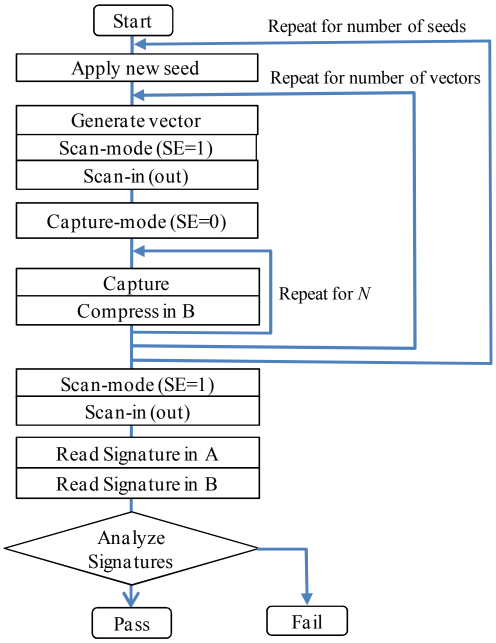
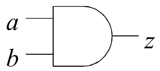


Figure 4. Multi-Cycle Logic BIST Flow



$$\mathrm{CC0}(z) = \mathit{min}\,(\mathrm{CC0}(a), \mathrm{CC0}(b))$$
$$\mathrm{CC1}(z) = \mathrm{CC1}(a) + \mathrm{CC1}(b) + 1$$
$$\mathrm{CO}(a) = \mathrm{CO}(z) + \mathrm{CC1}(b) + 1$$
$$\mathrm{CO}(b) = \mathrm{CO}(z) + \mathrm{CC1}(a) + 1$$

Figure 5. Example of SCOAP Calculation of AND Gate

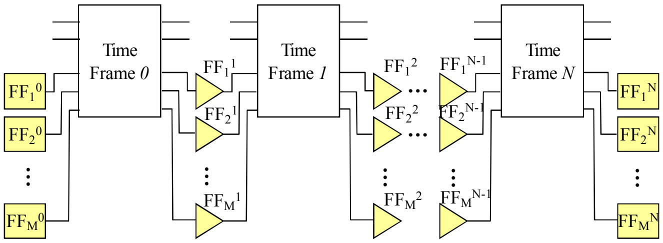


Figure 6. Time Frame Expansion of CUT

IV. METHOD TO CHOOSE PARTIAL FLIP-FLOPS

The following three methods, which choose partial FFs to observe their values, were compared to each other.

A. *Random method*

Partial FFs of the specified ratio are randomly chosen. This method is evaluated for reference.

B. *Simulation-based method*

The method consists of the following steps;

Step 1. Assume all of the FFs are observable.

Step 2. Perform fault simulation and calculate a coverage contribution of each FF.

Step 3. Select the FF that has the largest coverage increase.

Step 4. Remove the faults detected by the selected FF (or FFs) from the fault list.

Step 5. Repeat Step 2 to Step 4 for the remaining FFs and the remaining faults until the ratio of selected FFs reaches the specified ratio.

This method maximizes the fault coverage for a specified ratio of FFs. However, the fault simulation should be performed after the scan chain has fixed. That means, after placement & routing have completed, selected FFs should be connected to Compactor *B* in Fig.3 and this requires layout iterations, which will increase the design term. It also requires iterations of fault simulation, which results in huge computation time. Therefore, this method is also used for reference to get the maximum fault coverage improvement.

C. *Controllability & Observability Analysis Method*

The SCOAP testability analysis method [19] is used for statically estimating the observability of FFs. Fig. 5 shows an example of SCOAP calculation for an AND gate, where CC0(*s*) is a combinational 0-controlability of signal line *s*, CC1(*s*) is a combinational 1-controlability of *s* and CO(*s*) is a combinational observability of *s*. Partial FFs to be observed is selected as follows:

Step 1. Transform a CUT to an N^{th} time frame expansion combinational circuit (see Fig. 6), where 1st FFs (FF_1^0, FF_2^0, , , FF_M^0) are replaced with primary inputs, intermediate FFs (FF_1^i, FF_2^i, , , FF_M^i: i=1 to *N*-1) are replaced with through buffers and the last FFs (FF_1^N, FF_2^N, , , FF_M^N) are replaced with a primary output.

Step 2. Calculate CC0/1 and CO value of each gate in the expanded circuit.

Step 3. For each FF_j (*j*= 1 to *M*), list up CO value of the i^{th} buffer FF_j^i (*i*=1 to *N-1*).

Step 4. Select the FF_k^i ($1 \leq i \leq N\text{-}1$, $1 \leq k \leq M$) that has the largest CO value.

Step 5. Add primary outputs to *N*-1 flip-flops FF_k^i (*i*= 1 to *N*-1).

Step 6. Repeat Step 2 to Step 5 while the ratio of selected FFs is less than the specified ratio.

V. EXPERIMENTAL RESULTS

The three methods introduced in the previous section were evaluated using ISCAS'89 benchmark circuits. FFs in the circuits were stitched in a single scan chain. 100 test vectors generated by 16 bit LFSR were supplied into the scan chain in serial. Only a seed of "0001000100010001" was used for the evaluation. Pseudo-random vectors were also provided to primary inputs using LFSR in parallel, which has the same bit length as the number of primary inputs and has the initial seed value of all 1s. Aliasing in Compressor *A* and *B* were not evaluated for simplicity.

Table I shows the results on fault coverage evaluation of the three methods with 5 cycles of multiple captures. The 2nd row shows the coverages when no observation is done during intermediate captures. The 3rd row shows the coverages when all of the FFs are observed during captures as shown in [10]. They are the maximum fault coverages by the observations and are nearly 6% higher in average than no observation case. It implies the effectiveness of the observations during captures. The rows of "Sim-based" show the results of the simulation-based method with 20% and 50% observations respectively. From the results, it can be seen that the coverages show almost the same as those in the 3rd row, which are the maximum values. The rows of “SCOAP” show the results of the controllability & observability analysis method using SCOAP. The rows of “Random” are the results of the random method. The results of “SCOAP” are 1.88% (= 72.44 - 70.56) lower with 20% observation and 1.21% (= 72.49 - 71.28) lower in average with 50% observation than those of “Sim-based”, respectively. However, they show 3.9% (= 70.56 – 66.66) higher with 20% observation and 4.62% (= 71.28 – 66.66) higher with 50% observation than that of no-observation’s. These values show the effectiveness of “SCOAP” and even 20% observation shows nearly 4% improvement. The comparison between “SCOAP” and “Random” shows nearly 1 to 2% improvements due to the proposed “SCOAP” algorithm.

Table II shows the effect of cycle increase. The meanings of columns “Sim-based”, “SCOAP” and “Random” are the same as those of Table I. The 3rd column shows the number of capture cycles. As seen in the table, the effect of cycle increases from 5 to 10 is less than 1% and seems not so much. On the contrary, there are a little decrease of the fault coverages in three cases of s5378 at “Sim-based”, s5378 at “Random” and s15850 at “Random”.

Fig. 7 and Fig. 8 are the analysis of this phenomenon with 20% observations. Unlike the prosperous increase of coverages in Fig. 8, Fig. 7 shows drastic decrease of the coverages of “Random” as the number of cycles increases. “Sim-based” and “SCOAP” show slightly the same trend. This suggests that the circuit contains some kind of logic that the faulty values are masked as the clock advances. This decrease is alleviated with algorithmic selection of observing FFs such as “Sim-based” or “SCOAP”. Table III is the comparison of CPU time that shows the efficiency of “SCOAP” in comparison with “Sim-based”.

In Fig. 9 and Fig. 10, the number of test vectors is increased from 1,000 to 10,000 for analyzing the effect of the proposed multi-cycle test with observations. From these figures the following observations can be obtained:

- The fault coverage is stabilized around 3,000 to 4,000 vectors.
- The stabilized fault coverage of the proposed test is 2-4% higher than a single capture test (The same result shown in Table I).
- When we focus on the number of vectors to achieve the stabilized fault coverage, the figures show that less than half vectors are enough (i.e. ~2X compression).

TABLE I. FAULT COVERAGE EVALUATION WITH 5 CYCLES

Circuit Name	Observation methods & Ratio							
	---		Sim-based		SCOAP		Random	
	0%	100%	20%	50%	20%	50%	20%	50%
s5378	48.30	65.05	64.74	65.05	63.00	64.81	53.47	60.00
s9234	59.09	64.88	64.88	64.88	60.40	60.66	59.39	60.76
s13207	61.40	65.86	65.85	65.85	63.98	64.66	62.20	63.51
s15850	68.61	72.76	72.76	72.76	71.06	71.92	70.41	71.58
s38417	81.67	84.23	84.23	84.23	83.33	83.74	83.28	83.76
s38584	80.90	82.20	82.19	82.20	81.58	81.88	81.12	81.47
Average	66.66	72.50	72.44	72.49	70.56	71.28	68.31	70.18

TABLE II. FAULT COVERAGE EVALUATION WITH DIFFERENT CYCLES

Circuit Name	Observation methods & # of cycle						
	---	Sim-based		SCOAP		Random	
	1	5	10	5	10	5	10
s5378	63.61	64.74	64.68	63.00	63.18	53.47	53.23
s9234	54.48	64.88	66.35	60.40	61.74	59.39	60.86
s13207	62.78	65.85	66.56	63.98	64.07	62.20	62.56
s15850	69.37	72.76	73.42	71.06	71.39	70.41	70.29
s38417	77.95	84.23	85.13	83.33	84.05	83.28	83.90
s38584	76.44	82.19	83.42	81.58	82.15	81.12	81.81
Average	67.44	72.44	73.26	70.56	71.10	68.31	68.77

TABLE III. COMPARISON OF CPU TIME

Circuit	Sim-based (sec)	SCOAP (sec)
s5378	1.67	0.03
s9234	3.86	0.06
s13207	8.60	0.18
s15850	11.86	0.20
s38417	73.34	1.08
s38584	89.23	1.57

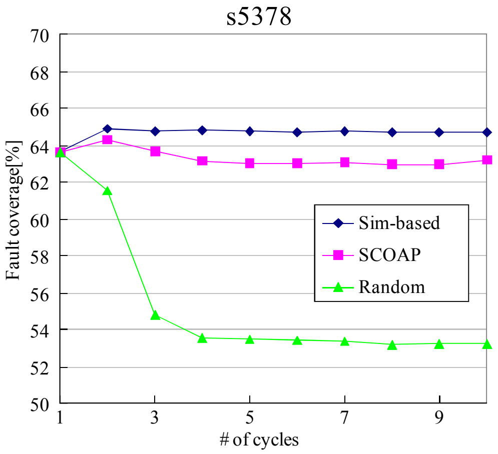


Figure 7. Fault Coverage Evaluation with Different Cycles (s5378)

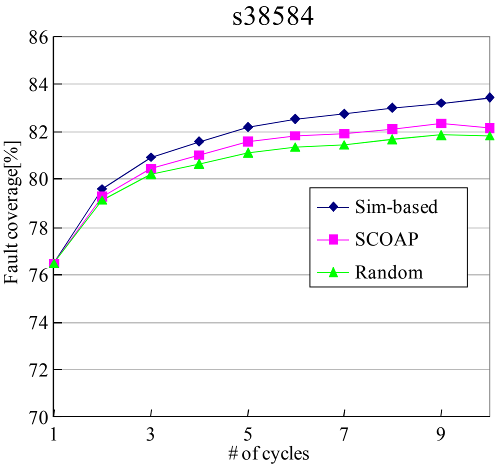


Figure 8. Fault Coverage Evaluation with Different Cycles (s38584)

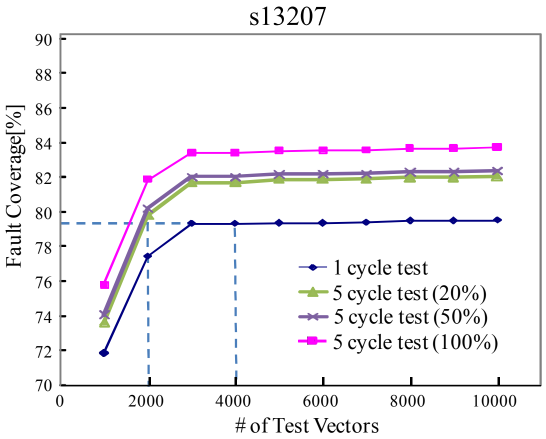


Figure 9. Effect of Multi-Cycle Test for Vecor Reduction (s13207)

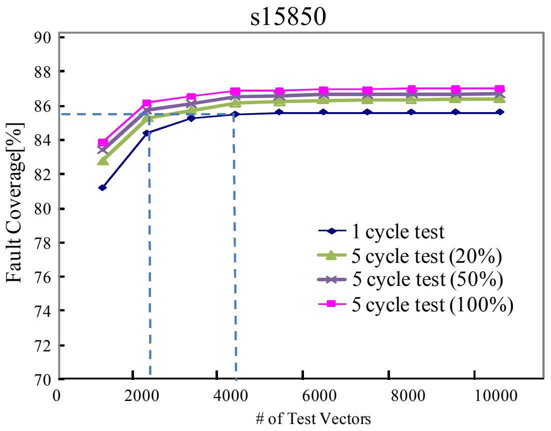


Figure 10. Effect of Multi-Cycle Test for Vecor Reduction (s15850)

As a space compactor, many methods such as Parity-tree, Zero-aliasing compactor and X-compact are proposed [20]. As discussing these are not our themes, we adopted a simple scheme of a compactor, which consists of a parity tree (XOR-tree) and MISR as shown in Fig. 11. It is not appropriate to discuss hardware overhead of the proposed scheme on small circuits. Therefore, we established a large data model that is based on the SoC model in the Test and Test Equipment chapter of ITRS2009 edition [1]. The model is as follows:

- Total gates: 17 M gates.
- Transistors in a FF: 26 transistors.
- Transistors in a XOR (g): 12 transistors.
- Number of FFs (m): 523 k (This is derived under the assumption that FFs occupy 20% of a chip).
- MISR bit (n): 523 (1% of FF number is assumed).
- Level of XOR-tree (p): a parameter.
- Ratio of FFs to be observed (α): a parameter.

Under the above assumptions and notations, number of transistors required for XOR-tree (G(α)) will be as follows.

$$G(\alpha) = g \times m\alpha \times \left\{ \frac{1}{2} + \left(\frac{1}{2}\right)^2 + \cdots + \left(\frac{1}{2}\right)^p \right\}$$

$$= g \times m\alpha \times \left(\frac{1}{2}\right) \times \frac{1 - \left(\frac{1}{2}\right)^p}{1 - \frac{1}{2}}$$

$$= g(m\alpha - n)$$ Here, $m\alpha \times \left(\frac{1}{2}\right)^p = n$ is used.

Using this equation, the hardware overhead was estimated as shown in Fig. 12. Ratio of MISR is a constant of 0.2% for α. Adding XOR gates, the total hardware overhead is 9.3% for full-observation, whereas, it can be reduced to 2.0% at 20% partial observation.

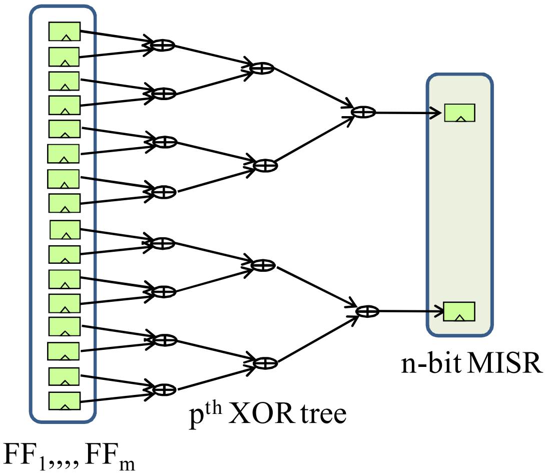


Figure 11. Space Compactor (XOR-Tree) + time Compactor (LFSR)

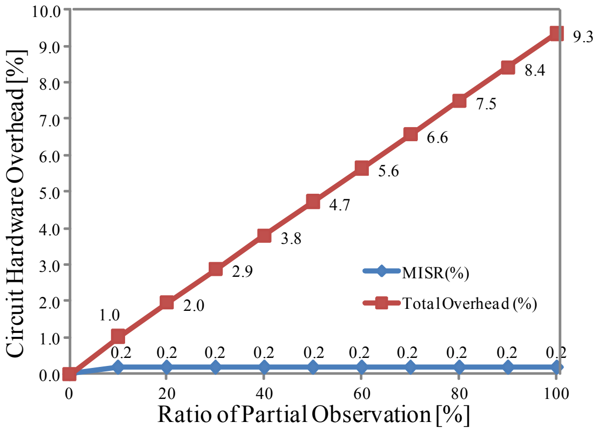


Figure 12. Hardware Overhead Estimation

VI. CONCLUSIONS

This paper addressed the multi-cycle test with partial observation, which is effective to improve fault coverage or to reduce test vectors. The proposed method can be used in conjunction with on either conventional logic BIST or deterministic compression test. The selection algorithm of observing flip-flops based on the controllability and observability analysis method of SCOAP was also proposed and its effectiveness regarding fault coverage improvement and CPU time reduction were evaluated.

Experimental evaluation shows that 20% observation of flip-flops improves nearly 4% fault coverage from the conventional 1 cycle test that is 1.88% lower than the full-observation improvement. The hardware overhead of 20% observation is 2.0% and it is far lower than 9.3% of the full observation. This means that the partial observation method achieves fault coverage improvement with small loss and with lower hardware overhead than the full observation method. Seeing from the view of vector number, this coverage improvement corresponds to nearly half reduction (i.e. ~2X compression), which is effective to reduce test data volume or test time reduction.

Our final goal is achieving high fault coverage with small seed data on scan-based BIST structure. The effect of seed data reduction will be evaluated in future.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Yoshinobu Higami, Ehime University, Japan, for his invaluable discussion in fault simulation of sequential circuits.

REFERENCES

[1] W. Wang, V. Reddy, A. T. Krishnan, R. Vattikonda,S. Krishnan, and Y. Cao, “Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” *IEEE Trans. on Device and Material Reliability*, VOL. 7, NO. 4, Dec. 2007.

[2] V. Reddy, J. Carulli, A. Krishinan, W. Bosch, and B. Burgess, “IMPACT OF NEGATIVE BIAS TEMPERATURE INSTABILITY ON PRODUCT PARAMETRIC DRIFT,” *Int. Test Conf.*, pp. 148-155, Oct. 2004.

[3] O. Khan and S. Kundu, “A Self-Adaptive System Architecture to Address Transistor Aging,” *Design Automation and Test in Europe.*, pp.81-86, April 2009.

[4] Y. Li, S. Makar, and S. Mitra, “CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns,” *Design Automation and Test in Europe.*, pp. 885-890, March 2008.

[5] Y. Sato, S. Kajihara, Y. Miura, T. Yoneda, S. Ohtake, M. Inoue, and H. Fujiwara, “A Circuit Failure Prediction Mechanism (DART) for High Field Reliability,” *Int. Conf. on ASIC*, pp. 581-584, Oct. 2009.

[6] Y. Sato, S. Kajihara, M. Inoue, T. Yoneda, S. Ohtake, H. Fujiwara, and Y. Miura, “Circuit Failure Prediction by Field Test (DART) with Delay-Shift Measurement Mechanism,” *Integrated Circuits and Devices in Vietnam*, pp. 5-10, Aug. 2010.

[7] J. Rajski, “We have got compression, what next?,” *IEEE European Test Sympo.*, Keynote speech, May. 2009.

[8] *International Technology Roadmap for Semiconductors*, 2009 Edition. http://www.itrs.net/

[9] N. A. Touba, “Survey of Test Vector CompactionTechniques,” *IEEE Design & Test of Computers*, July-August 2006, pp. 294-303.

[10] S. Kajihara, M. Matsuzono, H. Yamaguchi, Y. Sato, K. Miyase, and X. Wen, “On Test Pattern Compaction with Multi-Cycle and Multi-Observation Scan Test,” *Int. Sympo. on Communications and Information Technologies*, pp. 723-726, Oct. 2010.

[11] J.Savir, "On broad-side delay testing," *VLSI Test Sympo.*, pp.284-290, April 1994.

[12] C. Lin, Y. Zorian and S. Bhawmik, ”PSBIST: A partial-scan based built-in self-test scheme,” *Int. Test Conf.*, pp. 507-516, Oct. 1993.

[13] H.-C. Tai, K.-T. Cheng and S. Bhawmik, “Improving The Test Quality for Scan-based BIST Using A General Test Application Scheme,” *Design Automation Conf.*, pp. 748-753, June 1999.

[14] Y. Huang, I. Pomeranz, S. M. Reddy and J. Rajski, “Improving the proportion of At-Speed Tests in Scan BIST,” *Int. Conf. on Computer-Aided Design*, pp. 459-463, Nov. 2000.

[15] J. Abraham, U. Goel, and A. kumar, ”Multi-Cycle Sensitizable Transition Delay Faults,” *IEEE VLSI Testing Sympo.*, pp. 306-311, April 2006.

[16] K-T. Cheng, “Transition Fault testing for Sequentil Circuits,” *IEEE Trans. on Computer-Aided Design of ICs and Systems*, VOL. 12, NO. 12, Dec. 1993.

[17] I. Pomeranz and S. M. Reddy, ”Forming Multi-Cycle Tests for Delay Faults by Concatenating Broadside Tests,” *IEEE VLSI Testing Sympo.*, pp. 51-56, April 2010.

[18] E. K. Moghaddam, J. Rajski, S. M. reddy and M. Kassab, “At-Speed Scan Test with Low Switching Activity,” *IEEE VLSI Test Sympo.*, pp.177-182, April 2010.

[19] L. H. Goldstein, and E. L. Thigpen, “SCOAP: Sadia Controllability / Observability Analysis Program,” *Design Automation Conf.*, pp.190-196, June. 1980.

[20] L.-T. Wang, C.-W. Wu and X. Wen,”VLSI TEST PRINCIPLES AND ARCHITECTURES,” *Elsevvier*, ISBN 13: 978-0-12-370597-6, 2006