Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 3
March 2025

Hierarchical DFT Approach for Testability

[1] Priyadarshini K, [2] Dr. Vasundhara Patel K S

VLSI Design and Embedded Systems, ECE

[1] MTech Student, BMSCE, Bangalore

[2] Professor, BMSCE, Bangalore

Abstract

The complexity and compactness of current chip designs have increased due to the rapid improvements in VLSI technology. To ensure that these designs function as intended, it is imperative that they can be tested after manufacture. After a chip is manufactured, defects can be found using Design for Testability (DFT) procedures, which eliminates the requirement for extensive functional testing on every one of the possibly many physical devices. Rather than testing each and every device in-depth, DFT optimizes the detection of problems by improving controllability and observability across all nodes within the system. To find problems on every single node, Automatic Test Pattern Generation (ATPG) was carried out on three distinct blocks in this project. WGLs that were created were used in simulations. All three blocks underwent Test Point Insertion (TPI), with the goal of enhancing the DC coverage. The outcomes of the three blocks were compared, an increase in pattern count along with a 0.4–2% improvement in DC coverage was observed.

Keywords: DFT, ATPG, Coverage Analysis, Test point Insertion.

1. Introduction

The four main phases of the IC design cycle are design, implementation, testing, and specifications. Design-for-Testability (DFT) becomes crucial during testing to confirm chips after fabrication. This is important since manufacturing processes aren't perfect, therefore each chip must pass stringent testing to make sure it satisfies standards. The "divide and conquer" tactic used by leading industry practices divides the chip into discrete functional pieces. Every block is independently targeted by ATPG, producing test patterns that are designed to verify each node. Strict temporal constraints are considered when validating these blocks. To ensure comprehensive testing of functionality and performance, CADENCE tools are specifically used to develop test patterns at the block level for a 6nm device.

To achieve sufficient detection of transition (AC) and stuck-at (DC) problems in accordance with specifications, test coverage analysis is necessary. After that, simulation techniques are used to validate the developed test patterns. More specifically, Cadence's Modus tool is used for conducting simulations. These simulations use the Waveform Generation Language (WGL) and are then converted into formats particular to testers for silicon testing. This guarantees that, prior to deployment, the chips are extensively tested to ensure they fulfill quality and performance standards.

Scan stitching, which connects Scan Input (SI) and Scan Output (SO) pins, integrates combinational logic and scan flip-flops in each block of the design. Top mapping is the technique by which these pins at the block level are aligned with equivalent top-level scan input and output pins. The available input pins allotted for DFT purposes are used to determine the longest scan chain length. This dictates how the flops are sewn together inside the pattern. The quantity of storage required for test patterns can also be decreased by using compression techniques.

In digital circuit design, the DFT scan approach is used to make integrated circuit (IC) testing and debugging easier. To collect and output internal states for testing, more logic is added to the circuit design using the DFT scan process. The circuit is divided into a number of smaller sections known as scan chains to perform the DFT scan technique. A

series of flip-flops connected in a serial format, along with extra control logic to enable the scan process, comprise each scan chain. The internal state of the circuit is captured by the scan chains and can subsequently be exported for testing and analysis.

The goal of Design for Test (DFT) approaches is to improve fault coverage by strengthening a chip's internal node control and observation capabilities. The term "controllability" describes the ease with which input pins can be changed by designating binary values (0 or 1). It measures how simple it is to set values on a circuit's primary inputs (PIs). Conversely, observability quantifies the ease with which the circuit can identify and transfer node values to the primary outputs (POs). Different DFT methods have been created to improve observability and controllability.

2. LITERATURE SURVEY

In the realm of semiconductor design and testing, achieving high testability is crucial to ensure the reliability and functionality of integrated circuits (ICs). Hierarchical Design for Testability (DFT) has emerged as a significant approach to address the complexities and challenges associated with testing modern ICs. This literature review explores the principles, methodologies, benefits, and challenges of hierarchical DFT in enhancing testability.

In paper [2], A Hierarchical DFT strategy is implemented to enhance SOC testability, partitioning the design into layout and DFT regions based on functionality. It builds test vectors with Modus tools for 16nm technology using the D-algorithm for ATPG with the goal of achieving high fault coverage (>99% stuck-at, >85% transition faults). Validation through IES simulations ensures accuracy before real silicon deployment, encompassing both block and device levels with and without timing constraints. Pin-level mapping and logical simulations further validate effective test pattern application and integration across design hierarchy.

The often-utilized architecture included in the DFT technique is compression logic. In paper [10], compression technique is used. In this technique, the scan chains are connected in a pipeline and the output of the last scan flop in a scan chain is passed through XOR logic. This technique reports on the reduction of the test size and the test run-time, but the area of the chip increases by the inclusion of this design in the DFT region. There comes a need to optimize the area and power. Research on trade-

offs between the power consumed and area occupied is being carried out.

In paper [3], insertion of scan chains is explained. Scan chain technique is used to target both the combinational and sequential logic for controllability and observability. For the post-scan stitch procedure, there are numerous Automatic Test Pattern Generation tools at one's disposal. The test patterns are validated by simulation techniques, and during simulation, the patterns are reported as pass or fail. A higher chip yield is obtained by scanning at a lower process technology.

In paper [17], the circuits are targeted for high-speed tests for the better performance of the circuits for at-speed. This paper comes with an effective technique which reports in reduction of test patterns during commit test technique, scan shift and capture operation, to control the switching activity of the scan input bits to reduce the power. The area overhead is also taken care of which reflects on the reduced cost of the chip. Three new techniques: launch-on-shift, launch-on-capture and mixed at-speed testing are proposed in this paper to support the low power and low-cost chip testing.

In paper [14], clock gating and non-clock gating structures are used for testing the device. Some modes used clock gating concept whereas the other modes ran with non-clock gating mechanism. The main aim was to reduce power consumption. The author came to the conclusion that the modes with clock gating structure reported better test coverage than the non-clock gating modes after executing and analyzing a variety of test cases. Run time was also found to be reduced when compared to the novel approach.

Built-in Self-Test (BIST) is the ATPG approach used for testing in paper [16]. BIST is an internal circuit that undergoes several loops to make accurate scan tests and reduces the run time when compared to conventional methods. This type of testing can be

used for the circuits that contain combinational blocks. This technique resulted in coverage improvement and reduced pattern size.

Paper [1] discusses an approach to enhance fault diagnosis and testability through the strategic insertion of test points, specifically as observation points. To identify optimal observation points, scores are calculated for signal lines associated with each fault pair that the existing test set fails to distinguish. Once the observation points are selected, the method organizes the primary outputs and the inserted observation points into groups, using XOR operations to compact the output responses within the same group. This partitioning strategy reduces the number of observed values while maintaining diagnostic effectiveness. The proposed approach is validated through experiments conducted on benchmark circuits.

3. METHODOLOGY

Fig. 1 depicts the project flow or methodology. Three blocks are used to construct the ATPG, and the WGLs are then used as simulation input. In the event that miscompares are found, ATPG debugging is carried out to remove any binary or X miscompares.

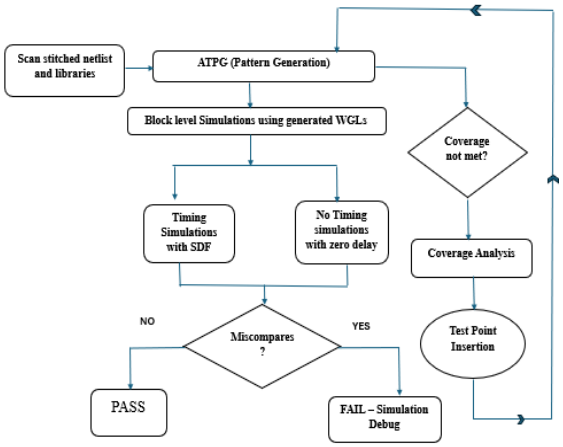

Fig 1: Project Flow

- ATPG: These patterns identify whether the semiconductor is operating correctly. By comparing the fault-free output, ATE ascertains if the circuit is free from manufacturing flaws when the Test pattern is applied. With ATPG, automated testing apparatus can differentiate between the proper circuit behavior and the defective circuit behavior resulting from flaws.
- Coverage Analysis: Test coverage produced by ATPG tools indicates an IC's quality. The goals for coverage are 85% for the transition test and 99% for the stuck-at test. When test coverage falls short of these objectives, the DFT engineer must be well-versed in design structure and have a close working familiarity with the ATPG tool in order to troubleshoot the poor test coverage.

The most difficult part of debugging low-coverage issues is figuring out why ATPG isn't able to generate a pattern to identify the errors and developing a plan to increase the coverage.

- Block level Simulations:

A testbench is created with the provides WGLs and is applied to DUT for simulating and checking the behavior of the design for the provided inputs.

a) ATPG tool will generate the patterns as per the design.
b) Simulation is needed to validate the generated patterns.

Cadence Xcelium is used for NO-Timing & Timing simulations.

Two ways of simulations:

i. Serial Simulation's using WGL
ii. Parallel simulation using modus generated TB

- Test Point Insertion:

Test point insertion is a circuit modification technique that adds extra gates into circuits during the design-for-test (DFT) phase and enhances circuit testability. Test points are added into the circuit to improve controllability and observability of a node in the circuit. The test points are categorized into two types: control points (CPs) and observation points (OPs).

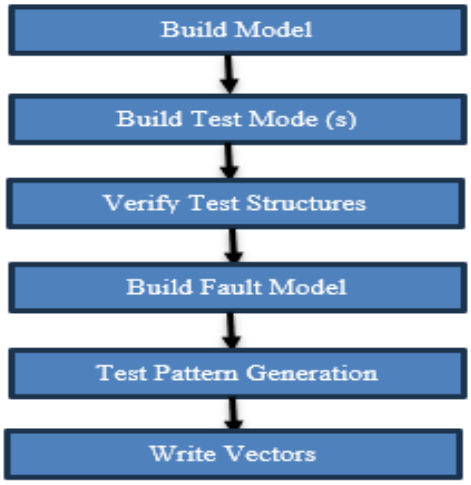

Fig 2: ATPG Flow

• Build Model: Builds the test model that the MODUS uses by reading in Verilog netlists, technology libraries, memory, and analog models. A logical hierarchy is included in the test model to make design analysis easier.

• Build Test Mode(s): Defines the test structures to be used during ATPG and identifying scan chains, compression structures, and other DFT-related logic and provides active and inactive logic.

• Verify Test Structures: Verify that scan chains and other test structures in the design adhere to guidelines aimed at ensuring testability. Severe warnings highlight structures that could lead to the generation of invalid vectors, while other warnings flag issues that might reduce achievable test coverage. Address all severe warnings prior to ATPG to thoroughly analyze the design database for potential issues such as scan chain integrity, clocking concerns, and contentions.

• Build Fault Model: This tool generates fault models essential for ATPG, aiding in both ATPG and diagnostics simulations. It constructs specific fault models for static and dynamic faults within the design, providing detailed statistics across different fault categories.

• Test pattern generation: The Modus tool from Cadence is primarily utilized for ATPG in 6nm technology. Test pattern generation involves determining the stimuli needed to validate a design's correctness and ensure it is free from manufacturing defects. To generate these test vectors, script files compatible with the Modus tool must be authored. The process includes creating models that encompass the netlist and 6nm technology libraries, configuring tests, defining structures, and verifying the design to identify potential issues. The output includes reports on fault models analyzed and their respective testing outcomes.

• Write vectors: This is used to write out the test vectors in the following format: STIL, WGL, Verilog, TDL. It generates vectors for simulation or to comply with IC manufacturers' manufacturing interface requirements. It also reports faults, detailing individual fault models and their current testing status.

4. RESULTS AND ANALYSIS

Three blocks are used in this project among which

a) The first block has DFT region which needs to be mapped to high level netlist (layout region), so macro-mapping is done here.
b) The second block is a RAM dominated block.
c) Third block has multiple clock domains. So, the concept of OCC is being used here.

There are 6 Testmodes that are run to achieve greater coverage. Among the 6 modes, 2 are AC modes and 4 are DC Modes. The AC modes are followed by the 4 DC modes, which include compression, non-compression and that mode which targets the memories.

Table 1: Blocks information

Block info	Macro map Block	RAM dominated block	OCC block
Gate Count	4 million (4457166)	2 million (2028945)	~1 million (994991)
Compression ratio	20	6	2
Scan bus width	168	168	168
Internal chain count	3360	1008	336
No. of clocks	9	1	4

a) Macro-map block: The larger SOC is partitioned into small regions to aid DFT and Layout execution. The partition can be done in any of the following way:
i. DFT region and Layout region are same
ii. DFT region only (Layout region is an upper hierarchy to this DFT region)
iii. Layout region only (may have multiple DFT regions within)

In case-1, both no-timing and timing simulations are executed at same level.

But in case-2, ATPG patterns are generated, and no-timing simulated at DFT region level.

But timing simulation must be exercised at its layout region, as the routed SDF is available at layout level. So, all the test ports at DFT region must be mapped to its layout region to perform timing simulation, hence the flow is referred as Macro-mapping flow.

Table 2: Macromap block information

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	83.54	77.68	32170	0	4632764
AC_mode 2	88.26	82.62	3099	0	446540
DC_mode 1	94.60	82.62	1412	14642	229098
DC_mode 2	94.72	82.62	447	15380	67468
RAM sequential	94.88	82.62	383	25493	105713
Full Scan	95.01	0.00	449	25497	1232135

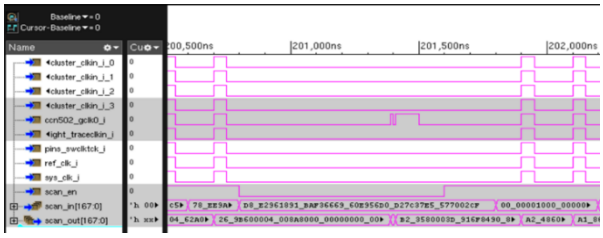

Fig 3: Macro map block simulations

b) RAM dominated block:

This block has more RAM instances. We can observe the AC coverage is extremely low as the block has more RAM instances which is going untested in AC mode. So an additional top up AC mode for memory is used to improve the AC coverage.

Table 3: RAM dominated block information

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	72.98	66.52	6672	0	1054488
AC_mode 2	74.52	68.80	2441	0	385990
DC_ mode 1	80.62	68.80	1456	4007	232752
DC_mode 2	80.71	68.80	142	4004	22759
RAM sequential	95.18	68.80	681	8674	384336
Full Scan	95.25	0.00	1	8674	1790

Table 4: Top up AC mode for memory

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	72.98	66.52	6356	0	1004568
AC_mode 2	74.52	68.80	2634	0	416484
RAM sequential AC mode	91.34	81.09	206	45365	121008
DC_ mode 1	94.12	81.10	1454	44195	231988
DC_mode 2	94.19	81.10	160	44175	25584
RAM sequential	95.21	81.10	670	8309	407600
Full Scan	95.27	0.00	1	8309	1790

In Table 4, we can see an improvement in AC coverage from 68.8% to 81.10% after top up mode was used.

Top up modes for memory is necessary to test all the untestable faults when the block has more RAM instances or modules instantiated in them.

Fig 4: RAM dominated block simulations

c) OCC block:

OCC stands for on chip clock control. It is the logic inserted on the SOC for controlling clocks during silicon testing on ATE (Automatic test Equipment) to ensure following requirements are met. ATPG has independent control of each clock domain to improve coverage, reduce pattern count, and achieve safe clocking with minimal user intervention.

OCC provides the appropriate number of clock pulses for each pattern during capture.

In order to apply slow and rapid patterns, it manipulates slow or fast clocks during capture.

It regulates the precise location of the capture window's capture pulse onset.

Table 5: OCC block information

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	86.67	81.59	7923	0	1323471
AC_mode 2	88.94	84.40	2664	0	445218
DC_ mode 1	95.16	84.83	442	2469	79500
DC_mode 2	95.21	84.83	60	2536	10549
RAM sequential	95.26	84.83	47	2554	28393
Full Scan	95.33	0	1	2554	670

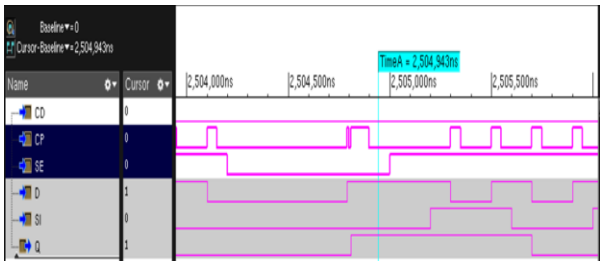

Fig 5: OCC block simulations

TEST POINT INSERTION:

Test point insertion is carried out to improve DC coverage. DC mode untested faults is targeted, and all these faults are reported into a file which is taken as input to analyze how many testpoints and where those Control or observe points needs to be inserted. The DFT engineer can mention the maximum testpoints which should be inserted into the design keeping the area constraint in picture. Then the tool gives the final count of how many testpoints can be inserted into the design.

An experiment was conducted in which a total of 2000 test points were inserted to each block using the MODUS tool.

For Macro map block, the tool was able to insert all the 2000 test points among which 1972 were observe points and 28 were control points.

Table 6: Coverage after Test point insertion on Macromap block

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	83.53	77.67	29484	0	4245980
AC_mode 2	88.24	82.61	3253	0	468716
DC_mode 1	95.60	82.61	1247	19907	188489
DC_mode 2	95.70	82.61	2628	20066	388656
RAM sequential	97.12	82.61	704	10917	362758
Full Scan	97.25	0.00	987	10914	2701935

For RAM dominated block, a maximum of 2000 testpoints was given to the tool, out of which only 699 testpoints could be inserted by the tool in the design. Out of which 687 observe points and 12 control points.

Table 7: Coverage after Test point insertion on RAM dominated block

Testmode	DC Coverage	AC Coverage	Patterns	Possibly Tested	Vector Count
AC_mode 1	72.87	66.41	6242	0	986548
AC_mode 2	74.52	68.81	4745	0	750022
RAM sequential AC	91.39	80.83	199	42076	115879
DC_ mode 1	94.51	80.83	810	40833	130342
DC_mode 2	94.51	80.83	26	40797	4397
RAM sequential	95.55	80.83	655	4248	398891
Full Scan	95.61	0.00	1	4248	1790

For OCC block, a maximum of 2000 testpoints was given to the tool, but the tool was unable to insert any testpoint into the design.

After comparing the coverage of all the blocks before and after Test point insertion we can observe that the pattern count has increased along with improvement in the coverage. Since the intention was to improve the coverage, the increase in pattern count factor cannot be compromised. But if the goal is to reduce pattern count and test time, then a different approach should be used to insert testpoints into the design.

Table 8: Comparing coverage of blocks before and after TPI

Block name	Before fault count (DC=AC)	After fault count (DC=AC)	Before coverage (DC)	After coverage (DC)
Macro map block	11725544	11729712	95.01%	97.25%
RAM dominated block	5549736	5551206	95.25%	95.61%
OCC block	2564304	-	95.33%	-

One of the objectives of the project was to achieve 99% DC coverage and 85% AC coverage. But we can see that the expected coverage is not met. Sometimes the faults will remain untested on Primary input and output pins which will get tested in Top Interconnect test or using some other advanced testing methods.

5. CONCLUSION AND FUTURE SCOPE

Because of the quick advancement of VLSI technology, chip designs have become more intricate and compact, necessitating post-fabrication testability to guarantee that the devices fulfill criteria. Defects can be found with the help of Design-for-Testability (DFT) approaches, which minimizes the requirement for extensive testing on a variety of devices. DFT minimizes testing

inefficiencies while optimizing error detection through increased controllability and observability.

The efficacy of the hierarchical DFT approach is noteworthy since it makes use of Automatic Test Pattern Generation (ATPG) to generate compact test vectors that effectively guarantee thorough fault coverage. Block-level integrity and compatibility are further guaranteed by validation procedures. Targeted coverage analysis and test point insertion help close gaps when certain blocks don't match coverage requirements, enabling the attainment of intended testing targets. In this project, despite an increase in pattern count, TPI was added to the blocks to track the progress in coverage. As TPI receives DC untested fault reports as input, there was a rise in DC coverage of 0.36% to 2.24%.

FUTURE SCOPE

Ultimately, in the highly competitive semiconductor industry, these tactics are essential for preserving the performance and dependability of contemporary chip designs. Future developments in technology, rising complexity, and the demand for effective testing methods will probably influence DFT and TPI in VLSI. The upcoming trends will concentrate more on automation, machine learning, and integration with more comprehensive design processes. Using AI in the testing environment might make it easier for the tool to find difficult-to-detect errors and using new testing strategies can help to increase test coverage and decrease runtime.

References

[1] Y. Higami, T. Inamoto, S. Wang, H. Takahashi and K. K. Saluja, "Improving of Fault Diagnosis Ability by Test Point Insertion and Output Compaction," 2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), Jeju, Korea, Republic of, 2023, pp. 1-5, doi: 10.1109/ITC-CSCC58803.2023.10212844.

[2] E Renold Sam Vethamuthu, S Sivanantham, R Sakthivel. "Implementation of Hierarchical DFT Approach for Better Testability", 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 2018.

[3] Ms. Janki Chauhan, Mr. Chintan Panchal, “Scan Methodology and ATPG DFT Techniques at Lower Technology Node”, Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC),pp.508-514, 2017.

[4] Jan Burchard, Felix Neubauer, Pascal Raiola, Dominik Erb and Bernd Becker, “Evaluating the Effectiveness of D-chains in SAT-based ATPG”, 18th IEEE Latin American Test Symposium (LATS), 2017.

[5] Aijiao Cui, Yanhui Luo, Chip-Hong Chang. "Static and Dynamic Obfuscations of Scan Data Against Scan-Based Side-Channel Attacks", IEEE Transactions on Information Forensics and Security, 2017.

[6] Dan Trock, Rick Fisette, “Recursive Hierarchical DFT Methodology with Multi-Level Clock Control and Scan Pattern Retargeting”, IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1128-1131, 2016.

[7] Zhou Jiang, Dong Xiang and Kele Shen, “A Novel Scan Segmentation Design for Power Controllability and Reduction in At-Speed Test”, IEEE 24th Asian Test Symposium, pp.7-12,2015.

[8] Renji Ono, Satoshi Ohtake. "A Method of Diagnostic Test Generation for Transition Faults", 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC), 2015.

[9] Rahul Malhotra, Sujay Deb, Fabio Carlucci,” A novel approach to Reusable Time economized STIL based pattern development”, IEEE Conference on VLSI circuits, 2015.

[10] A. Chandra, S. Chebiyam, and R. Kapur, “A Case Study on Implementing Compressed DFT Architecture”, IEEE 23rd Asian Test Symposium, pp.336-341, 2014.

[11] C.Vasanthanayaki, A.Azhagu Jaisudhan Pazhani, Jincy Johnson, “VLSI implementation of low power multiple single input change (MSIC) test pattern generation for BIST scheme”, IEEE Fifth International Symposium

on Electronic System Design, pp.187- 191, 2014.

[12] Kai Hu, Tsung-Yi Ho and Krishnendu Chakrabarty, “Test Generation and Design- for Testability for Flow-Based mVLSI Microfluidic Biochips”, IEEE 32nd VLSI Test Symposium (VTS), 2014.

[13] Samah Mohamed Saeed and Ozgur Sinanoglu, “Design for Testability Support for Launch and Capture Power Reduction in Launch-Off-Shift and Launch-Off-Capture Testing”, IEEE transactions on Very Large-Scale Integration (VLSI) systems, pp.1-6, 2013.

[14] Kun-Han Tsai, Shuo Sheng,” Design Rule Check on the Clock Gating Logic for Testability and Beyond”, IEEE Int. Test Conf., Paper 15.2, pp.1-8,2013.

[15] Deepa Jose, P. Nirmal Kumar, L. Saravakanthan, R. Dheeraj,” Computer Aided Partitioning for Design of Parallel Testable VLSI Systems”, IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp.1363-1366, 2013.

[16] Takanori Moriyasu, Satoshi Ohtake,” A Method of LFSR Seed Generation for Scan Based BIST Using Constrained ATPG”, Seventh International Conference on Complex, Intelligent, and Software Intensive Systems, pp.755-759, IEEE, 2013.

[17] Samah Mohamed Saeed and Ozgur Sinanoglu, “Design for Testability Support for Launch and Capture Power Reduction in Launch-Off-Shift and Launch-Off-Capture Testing”, IEEE transactions on Very Large-Scale Integration (VLSI) systems, pp.1-6, 2013.

[18] Mohammad Tehranipur, Ke Peng, Krishnendu Chakrabarty. "Test and Diagnosis for Small-Delay Defects", Springer Science and Business Media LLC, 2012.

[19] K. Miyase, Y. Uchinodan, K. Enokimoto, Y. Yamato, X. Wen, S. Kajihara, “Effective Launch-to-Capture Power Reduction for LOS Scheme with Adjacent-Probability-Based X- Filling”, IEEE Asian Test Symposium, pp.90-95, 2011.

[20] Anshuman Chandra, Rohit Kapur. "Bounded Adjacent Fill for Low Capture Power Scan Testing”, 26th IEEE VLSI Test Symposium (vts 2008), 2008.

[21] V.R. Devanathan, C.P. Ravikumar, V. Kamakoti, “Reducing SoC Test Time and Test Power in Hierarchical Scan Test: Scan Architecture and Algorithms”, 20th International Conference on VLSI Design (VLSID'07) IEEE, 2007.

[22] Jeff Remmers, Moe Villalba, Richard Fisette, “Hierarchical DFT approach- A case study”, IEEE Int. test conference, paper 30.2, pp 847-856, 2004.

[23] Xcelium and Modus User guide from Cadence website, https://www.cadence.com/