Implementation of Low Power BIST for Memory Architecture

Rajashekar D S[1], Vasundhara Patel K S[2]

[1]Dept. of ECE, BMS College of Engineering, Bengaluru, India
[2]Professor, Dept. of ECE, BMS College of Engineering, Bengaluru, India

---***---

Abstract - *As deep sub-micron systems evolve and incorporate more advanced technologies, such as shrinking transistor sizes and increasing memory densities, the memory arrangement and testing strategies for any post silicon faults can become increasingly complex. Conventional testing using ATE (Automatic Test Equipment) consumes more test time and test cost, so an automated test strategy for such designs is needed. Automated testing strategies like Memory BIST involves incorporating specialized circuitry within the memory itself to facilitate self-testing of the memory. This can significantly reduce the need for external test equipment and streamline the testing process while also minimizing the overall testing costs. Taking advantage of the same, Memory BIST architecture is implemented in this project work. The introduction of BIST circuitry on a chip can indeed bring advantages like reduced testing costs and self-testing capabilities, but it also leads to increased power consumption due to the additional logic and increased switching activity. To mitigate this issue and maintain low power dissipation, Implementation of a Clock gating technique is done in this project. Clock gating insertion has been done during the synthesis stage in Cadence Genus tool. Memory BIST insertion is carried out in Tessent MBIST Architect and the Power analysis is done in Cadence Genus Joules. Simulation results and Power analysis results proves that the MBIST insertion is done without any mis compares. Also the difference in power consumption before and after the insertion of ICG (Integrated Clock Gating) cells is observed.*

Key Words: MBIST; MARCH Algorithm; ICG; Tessent MBIST Architect tool; Cadence Genus Joules tool

1. INTRODUCTION

Today's cutting-edge semiconductor industry is characterized by a rapid pace of SoC development. The need for embedded memories keeps growing since more and more functions are added to the die. Sub-micron devices have many memories that call for smaller footprints and quicker access times. It becomes very important to test these memories for post-silicon defects.

BIST offers several advantages over traditional external testing methods that rely on specialized equipment or test vectors. By enabling circuits to test themselves without external intervention, BIST reduces the need for expensive and complex test equipment and interfaces. This simplifies the testing process while increasing test coverage, as circuits can generate and apply tailored test patterns. Furthermore, BIST facilitates faster and more reliable testing by performing self-checks and self-repairs during operation or at predefined intervals. The difficulties and constraints that come with using BIST techniques in CMOS circuit design must be taken into consideration. BIST raises the overhead and complexity of the circuit, which may have an impact on the cost, area, power, and speed.

1.1 MBIST Architecture

The Controller, Interface, and Wrapper that make up the Memory BIST (MBIST) Architecture are depicted in the image. The decoder, which is a part of the Controller, interprets the commands that are transmitted from outside via the ATE. Based on the commands, the control logic carries out several operations, which in turn regulate the BIST's Finite State Machine (FSM). FSM manages the process and explains the MBIST's whole flow.

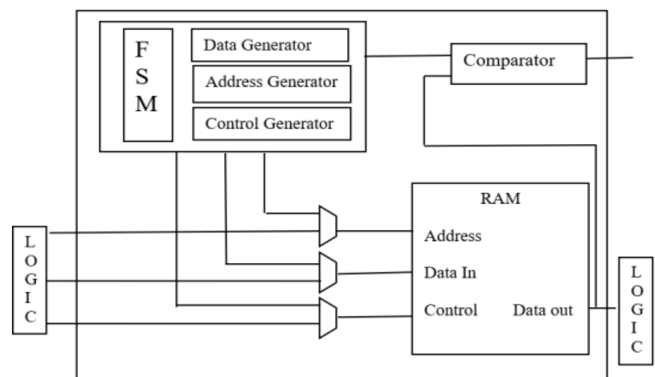


Fig 1. MBIST Architecture

The data generator, address generator, FSM, memory model, and comparator are all parts of the wrapper, as seen in the figure. The FSM starts the MBIST by choosing the appropriate row or column cells to test and executing a number of built-in algorithms. After reading the data again and comparing it to the golden response, a pass or fail decision is made. The ATE programs the FSM state externally and the background data generator uses that information to create a variety of patterns. Different memory cell addresses that can be read or written are generated by the address generator.

1.2 MBIST Algorithms

Memory arrays in integrated circuits are tested for integrity and operation using MBIST (Memory Built-In Self-Test) methods. In order to identify defects including stuck-at faults, transition faults, coupling faults, retention faults, and others, these algorithms carry out a variety of read and write operations. The following are a few popular MBIST algorithms and are broadly classified into two types,

A. Classical Algorithms - They are subdivided into MSCAN or Zero-One Algorithm, Checkerboard Algorithm, Galloping Algorithm (GALPAT), Butterfly Algorithm

B. March Algorithms - MATS Algorithm , MATS+ Algorithm , MATS++ Algorithm March X Algorithm, March C Algorithm, March C- Algorithm

MARCH C- Algorithm Implementation

The March C- algorithm consists of a sequence of operations (read and write) that are applied to each memory cell in ascending and descending address order. The operations are designed to systematically test each cell for potential faults. Here is the sequence of operations in the March C- algorithm:
{⇳ (w0) ; ⇑ (r0,w1) ; ⇑ (r1,w0) ; ⇓ (r0,w1) ; ⇓ (r1,w0) ; ⇳ (r0)}

Steps of the March C- Algorithm

Step 1: Write 0 to all cells ⇳ (w0) i.e, Initialise all cells to 0 both in ascending and descending order

Step 2: Read 0, Write 1{⇑ (r0,w1)} For each cell, read the value and ensure it is 0, then write 1 to the cell. This step is performed in ascending address order.

Step 3: Read 1, Write 0{⇑ (r1,w0)} For each cell, read the value and ensure it is 1, then write 0 to the cell. This step is performed in ascending address order.

Step 4: Read 0, Write 1{ ⇓ (r0,w1)} For each cell, read the value and ensure it is 0, then write 1 to the cell. This step is performed in descending address order.

Step 5: Read 1, Write 0{ ⇓ (r1,w0)} For each cell, read the value and ensure it is 1, then write 0 to the cell. This step is performed in descending address order.

Step 6: Read 0 from all cells ⇳ (r0) i.e, read 0 from all the cells and ensure it is 0 both in ascending and descending order.

2. METHODOLOGY

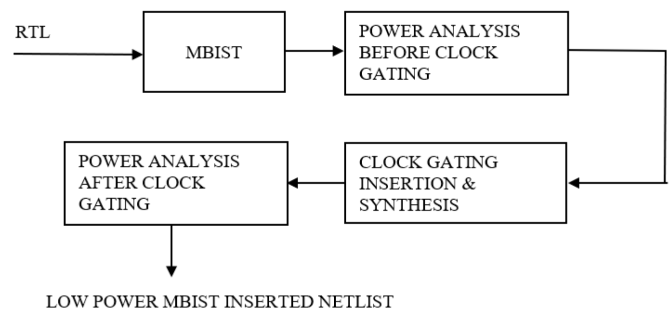


Fig 2. Block diagram Implementation

The implemented methodology shown above broadly consisting of four parts which are MBIST insertion, Power analysis before Clock gating, Clock Gating insertion and Power analysis after Clock Gating.

The brief flow of the project is explained as follows, RTL code has been fed to the tool(Tessent MBIST Architect) which inserts the MBIST logic to the design. Then we perform Synthesis on the MBIST inserted RTL that converts RTL code to Gate level Netlist. This netlist has been fed to the tool(Cadence Joules)which inserts ICG cells to the design that enables to lower the dynamic power dissipation. Hence able to test and detect the memory faults with reduced power dissipation.The detailed MBIST flow and Clock gating that has been implemented is discussed in the further sections.

Implementation of MBIST Flow

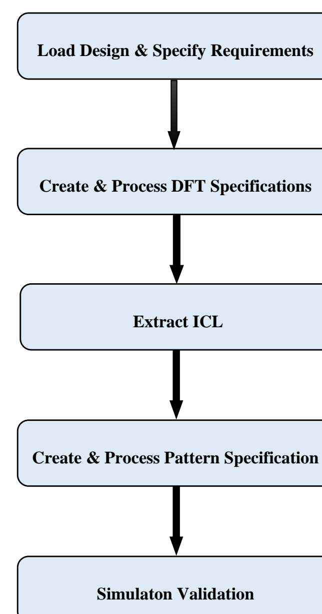


Fig 3. Implemented MBIST flow

Step 1 : Load Design and Specify Requirements

- RTL design is being loaded to the tool along with the necessary libraries.
- Specifying MBIST requirements involves the details about design information, memory information and DFT requirements.
- Design information such as number of clocks, frequency of clocks, UPF(Unity power factor) files. Memory information such as number of memories, type of memories, number of ports, test algorithm to be used. DFT requirements such as Max power per step, Max test time per step, Max memories per step.
- Once the design is loaded and specifying the requirements, we run the Design Rule Checking(DRC) to detect if any design issues are present

Step 2 : Create and Process DFT Specifications

- This step involves generation of DFT Specifications that are necessary for MBIST insertion. Also Modification of DFT Specification is also done as per the requirement during this step.
- Once the proper DFT Specifications are defined, then it'll generate and insert the MBIST hardware into the RTL design as per the specifications.

Step 3: ICL Extraction

- The main goal of ICL Extraction is automated generation of the interconnection information of various IJTAG building blocks (Instruments , SIBs, TDR's).
- ICL extraction process verifies the proper connectivity of the ICL modules that were inserted with the process_dft_specification. ICL extraction must pass with no violations in order to generate test patterns.

Step 4 : Pattern Specification

- Pattern specification defines a series of pattern wrappers for a given view of a design

Step 5 : Simulation Validation

- After creating and processing Pattern specification, the final step in the MBIST implementation flow is the Simulation validation. Here we run the simulation using Memory BIST verification patterns and then check the test bench simulations using Tessent shell environment.

Implementation of Clock Gating

Clock gating technique involves the insertion of ICG (Integrated Clock Gating) cells into the design. When designing digital circuits, an Integrated Clock Gating (ICG) cell is essential, especially for power control. By cutting off the clock signal to certain circuitry while it's not in use, a technique known as clock gating helps synchronous digital systems consume less dynamic power. The ICG cell is built to carry out this method efficiently. The typical ICG cell is as shown below.

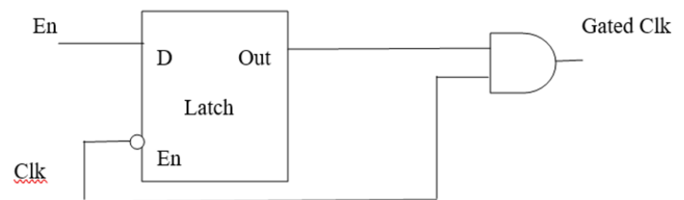


Fig 4. Clock Gating Cell

Operation of ICG cell

The ICG cell works by taking an enable signal and the original clock signal as inputs. When the enable signal is active, the clock signal is allowed to pass through to the gated clock output. When the enable signal is inactive, the clock signal is blocked, effectively disabling the clocking of the downstream circuitry.

When EN = 1, GCLK = CLK (the clock signal is passed through)
When EN = 0, GCLK = 0 (the clock signal is blocked)

3. RESULTS AND SIMULATIONS

MBIST Insertion & Validation

BIST_CLK - It controls the entire BIST operation as it drives the block that contains memories. If BIST_CLK is not pulsing properly, then the MBIST insertion and Validation fails.

MBIST_EN - This pin/signal describes whether to perform MBIST insertion or not. i.e, Tool will be able to perform MBIST only if MBIST_EN is set HIGH.

MBIST_GO - This signal will be flagged if any errors occurred or not during the execution of MBIST algorithm. MBIST_GO will be HIGH at start of the test. If any error, MBIST_GO transitions to LOW.

MBIST_DONE – This signal will be flagged whether the algorithm execution has been completed or not. MBIST_DONE will be LOW while test runs, goes HIGH when test is completed, returns LOW after controller results are extracted.

Checking GO and DONE signals,
DONE high + GO high means test ran to completion without errors
DONE low + GO high means test not yet complete and no errors yet
DONE high + GO low means test ran to completion with errors
DONE low + GO low means test not yet complete and error already encountered

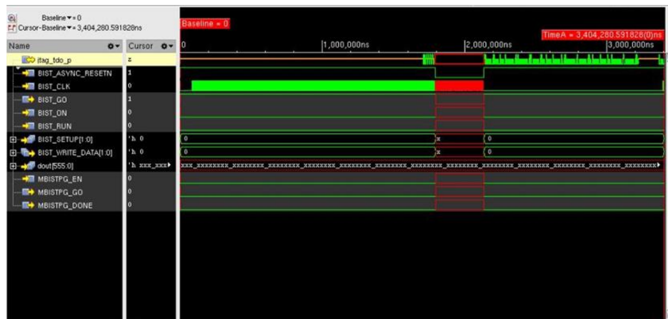


Fig 5. MBIST Insertion with Error 1

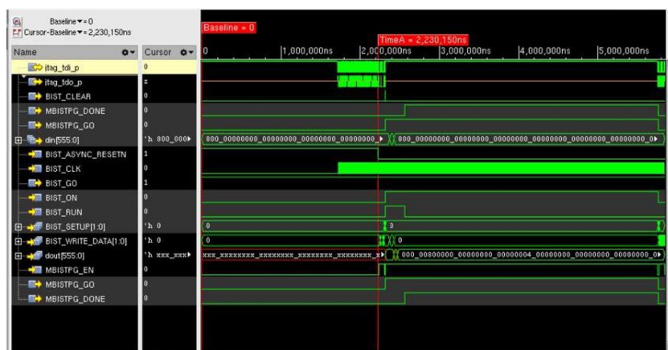


Fig 6. MBIST Insertion without Error 1

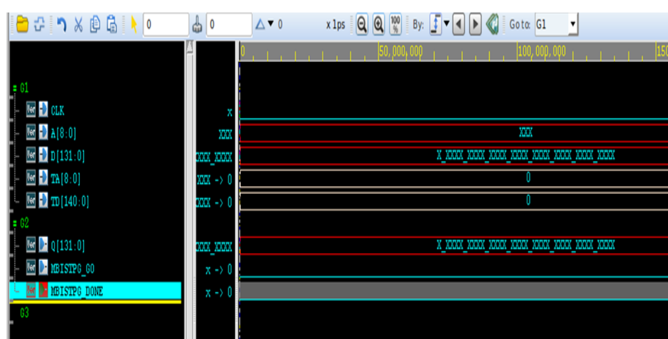


Fig 7. MBIST Insertion with Error 2

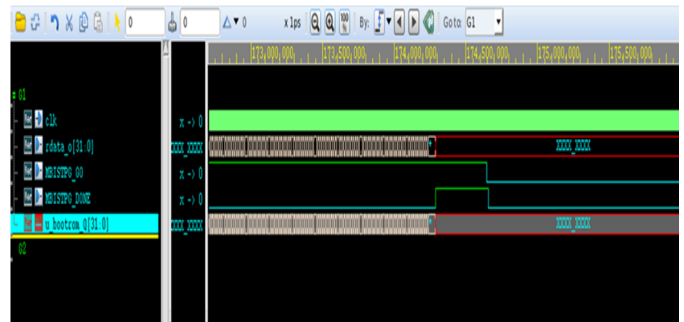


Fig 8. MBIST Insertion without Error 2

ICG Cell Insertion

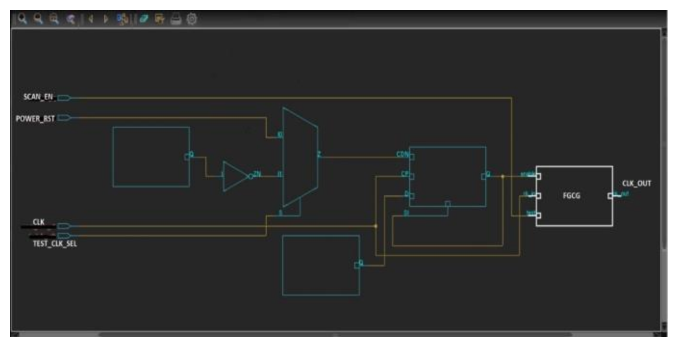


Fig 9. Clock gating cell

Clock Gating Instances report

Summary

Category	Number	%	Average Toggle Saving %
Total Clock Gating Instances	400	100.00	-
RC Clock Gating Instances	400	100.00	69.88
Non-RC Clock Gating Instances	0	0.00	0.00
RC Gated Flip-flops	4750	91.88	82.74
Non-RC Gated Flip-flops	0	0.00	0.00
Total Gated Flip-flops	4750	91.88	-
Total Ungated Flip-flops	420	8.12	-
Enable not found	210	50.00	-
Driven by a timing_model/Unresolved instance	10	2.38	-
Register bank width too small	200	47.62	-
Total Flip-flops	5170	100.00	-

Fig 10. Clock gating instances

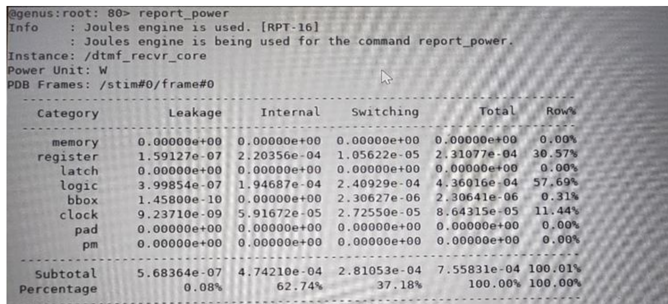
Power consumption before ICG cell insertion

Power Unit: W
PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	2.06660e-07	1.09016e-04	5.36861e-08	1.09277e-04	13.47%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	1.99045e-07	9.68840e-05	5.28760e-04	6.25843e-04	77.13%
bbox	1.45800e-10	0.00000e+00	1.20750e-06	1.20765e-06	0.15%
clock	9.07039e-09	6.23255e-05	1.27060e-05	7.50406e-05	9.25%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	4.14922e-07	2.68226e-04	5.42727e-04	8.11368e-04	100.00%
Percentage	0.05%	33.06%	66.89%	100.00%	100.00%

Fig 11. Power report before ICG cell

Power consumption before ICG cell insertion



```
@genus:root: 80> report_power
Info    : Joules engine is used. [RPT-16]
        : Joules engine is being used for the command report_power.
Instance: /dtmf_recvr_core
Power Unit: W
PDB Frames: /stim#0/frame#0
```

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	1.59127e-07	2.20356e-04	1.05622e-05	2.31077e-04	30.57%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	3.99854e-07	1.94687e-04	2.40929e-04	4.36016e-04	57.69%
bbox	1.45800e-10	0.00000e+00	2.30627e-06	2.30641e-06	0.31%
clock	9.23710e-09	5.91672e-05	2.72550e-05	8.64315e-05	11.44%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	5.68364e-07	4.74210e-04	2.81053e-04	7.55831e-04	100.01%
Percentage	0.08%	62.74%	37.18%	100.00%	100.00%

Fig 12. Power report after ICG cell

Unlike the power report of the design before ICG insertion, which has consumed 8113680000 nW. It is clearly evident that the power dissipation after the insertion of ICG cells has been reduced and consumed only 7558310000 nW, which means the power dissipation has been reduced by 7%. The power report is shown above and also the power values obtained has been tabulated as shown below.

Table 1. Power measurements of MBIST inserted netlist before and after Clock Gating

	Total power consumption before clock gating (W)	Total power consumption after clock gating (W)
MBIST inserted Netlist	8.11368E-04	7.55831E-06

4. CONCLUSIONS AND FUTURESCOPE

The work that has been carried out as a part of this project is majorly on the MBIST Insertion for memory architecture to enable the integrated circuits for self-testing on its own and Clock Gating insertion for reduced power consumption.

The implementation has been carried out at RTL level and the design is being synthesized to get the gate level netlist for power analysis. Firstly, MBIST insertion is done and power report is generated for that design and the readings are noted without insertion of ICG cells. Then, the same MBIST flow is repeated again and power report is generated after the insertion of ICG cells. It was clear from the obtained results that the power consumption has been reduced significantly.

While testing and detecting memory module errors was achievable in this work, but the errors were not repaired. Additionally, there is potential to use cutting edge low power implementation approaches to further minimize the power usage.

In the future,

- It is possible to add more BISR (Built in Self Repair) hardware to the functional logic to enable both the detection and correction of defects.
- There is a scope to use advanced low power approaches like FSM partitioning, Multilevel voltage scaling, and Supply voltage scaling.

REFERENCES

[1] A. Manzone, P. Bernardi, M. Grosso, M. Rebaudengo, E. Sanchez and M. S. Reorda, "Integrating BIST techniques for on-line SoC testing," 11th IEEE International On-Line Testing Symposium, French Riviera, France, 2005, pp. 235-240, doi: 10.1109/IOLTS.2005.38.

[2] S. B. Ghale and N. P, "Design and Implementation of Memory BIST for Hybrid Cache Architecture," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 26-31, doi: 10.1109/ICCES51350.2021.9489225.

[3] T. Q. Bui, L. D. Pham, H. M. Nguyen, V. T. Nguyen, T. C. Le and T. Hoang, "An Effective Architecture of Memory Built-In Self-Test for Wide Range of SRAM," 2016 International Conference on Advanced Computing and Applications (ACOMP), Can Tho, Vietnam, 2016, pp. 121-124, doi: 10.1109/ACOMP.2016.026.

[4] S. S. Rao Dannina and U. V. Ratna Kumari, "Error Tolerant MBIST Design with Efficient High Throughput and Low Power Reversible Techniques," 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2021, pp. 230-235, doi: 10.1109/CSNT51715.2021.9509611.

[5] A. S. Abu-Issa, "Energy-Efficient Scheme for Multiple Scan-Chains BIST Using Weight-Based Segmentation," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 3, pp. 361-365, March 2018, doi: 10.1109/TCSII.2016.2617160.

[6] M. Mosalgi and G. Hegde, "Power Optimized TPG for BIST Architecture," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 2017, pp. 1-4, doi: 10.1109/ICCIC.2017.8524577.

[7] P. Liu, J. Wu, Z. You, M. Elimu, W. Wang and S. Cai, "Defect Analysis and Parallel March Test Algorithm for 3D Hybrid CMOS-Memristor Memory," 2018 IEEE 27th Asian Test Symposium (ATS), Hefei, China, 2018, pp. 25-29, doi: 10.1109/ATS.2018.00016.

[8] A. Kincel and M. Balaz, "MBIST for LEON3 processor core cache," 2013 IEEE 16th International Symposium on

Design and Diagnostics of Electronic Circuits & Systems (DDECS), Karlovy Vary, Czech Republic, 2013, pp. 287-288, doi: 10.1109/DDECS.2013.6549836.

[9] A. S. Abu-Issa and S. F. Quigley, "Bit-Swapping LFSR and Scan-Chain Ordering: A Novel Technique for Peak- and Average-Power Reduction in Scan-Based BIST," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 5, pp. 755-759, May 2009, doi: 10.1109/TCAD.2009.2015736.

[10] W. Kang, C. Lee, H. Lim and S. Kang, "Optimized Built-In Self-Repair for Multiple Memories," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2174-2183, June 2016, doi: 10.1109/TVLSI.2015.2499387.

[11] Y. J. M. Shirur, H. R. Lakshmi and V. S. Chakravarthi, "Implementation of Area Efficient Hybrid MBIST for Memory Clusters in Asynchronous SoC," 2014 Fifth International Symposium on Electronic System Design, Surathkal, India, 2014, pp. 226-227, doi: 10.1109/ISED.2014.57.

[12] T. Tewary and A. Sen, "A novel approach to realize built-in-self-test(BIST) enabled UART using VHDL," 2014 First International Conference on Automation, Control, Energy and Systems (ACES), Adisaptagram, India, 2014, pp. 1-6, doi: 10.1109/ACES.2014.6808031.

[13] Dong Xiang, Ming-Jing Chen, Jia-Guang Sun and Fujiwara, "Improving test quality of scan-based BIST by scan chain partitioning," 2003 Test Symposium, Xi'an, China, 2003, pp. 12-17, doi: 10.1109/ATS.2003.1250773.

[14] K. Lee, S. Lee, J. Park, I. Lee and S. Kang, "A Low-Power BIST Scheme Using Weight-Aware Scan Grouping and Scheduling for Automotive ICs," in IEEE Access, vol. 9, pp. 116115-116132, 2021, doi: 10.1109/ACCESS.2021.3105429.

[15] S. R. Chand, V. Srinivas, T. V. Sai, M. Sailaja and T. Madhu, "Fault diagnosis for using TPG low power dissipation and high fault coverage," 2010 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 2010, pp. 1-5, doi: 10.1109/ICCIC.2010.5705884.

[16] J. Wibbenmeyer and C. -I. H. Chen, "Built-In Self-Test for Low-Voltage High-Speed Analog-to-Digital Converters," in IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 6, pp. 2748-2756, Dec. 2007, doi: 10.1109/TIM.2007.908343.

[17] D. Xiang, X. Wen and L. -T. Wang, "Low-Power Scan-Based Built-In Self-Test Based on Weighted Pseudorandom Test Pattern Generation and Reseeding," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 942-953, March 2017, doi: 10.1109/TVLSI.2016.2606248.

[18] D. Ghosh, S. Bhunia and K. Roy, "A technique to reduce power and test application time in BIST," Proceedings. 10th IEEE International On-Line Testing Symposium, Funchal, Portugal, 2004, pp. 182-183, doi: 10.1109/OLT.2004.1319684.

[19] V. Thirunavukkarasu, R. Saravanan and V. Saminadan, "Performance of low power BIST architecture for UART," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 2016, pp. 2290-2293, doi: 10.1109/ICCSP.2016.7754103.

[20] C. Zoellin, H. -j. Wunderlich, N. Maeding and J. Leenstra, "BIST Power Reduction Using Scan-Chain Disable in the Cell Processor," 2006 IEEE International Test Conference, Santa Clara, CA, USA, 2006, pp. 1-8, doi: 10.1109/TEST.2006.297695.