International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 12 | Dec 2023 www.irjet.net p-ISSN: 2395-0072

Advancing VLSI Design Reliability: A Comprehensive Examination of Embedded Deterministic Test (EDT) Logic insertion and its impact on Fault Classification and Transition Faults

Sandesh Y M[1], Sujatha K[2]

[1]*Dept. of Electronics and Communication Engineering, BMS College of Engineering Bangalore, India*
[2] *Dept. of Electronics and Communication Engineering, BMS College of Engineering Bangalore, India*

---***---

Abstract - Embedded Deterministic Test (EDT) logic insertion's impact on VLSI designs is comprehensively explored in this study. Our comparative analysis contrasts design with and without EDT logic, focusing on fault classifications and transition fault dynamics. Results indicate enhanced test and fault coverage with EDT insertion, albeit with increased test patterns. Significant shifts in fault sub-classes like atpg-untestable and tied-cells post-insertion highlight EDT's nuanced influence. This research presents empirical evidence to guide VLSI designers in optimizing design testability through strategic EDT logic integration.".

***Key Words*: EDT Integration, Fault Sub-classification, Scan-In and Scan-out, Test Data Volume, Design Rule Check, Lockup Latches, IP Generation, Multicycle Paths, Diagnostic Insights, Efficiency Trade-offs.**

1. INTRODUCTION

By including circuits on the chip, DFT in VLSI is a cutting-edge design method that makes testing a chip more affordable. To make all of the logic in the chip more testable, they boost the observability and controllability of internal nodes. Controllability is the capacity to program inputs to set a certain signal value at each internal node, while observability is the capacity to program inputs and monitor outputs to ascertain internal values. DFT also increases yield by offering high-quality debugging.

All of the VLSI circuits use DFT based on scan and ATPG because the system can deliver the highest level of efficiency and dependability in terms of coverage accomplishment of various fault models. The created test patterns identify circuit flaws and direct it to fix them. Because circuit complexity is rising in the modern VLSI age, failure rates are rising as well. This problem forces to develop additional patterns for identifying those flaws.

High test data volume and high memory requirements might be brought on by an increase in the number of faults in an ATE (Automatic test pattern equipment). EDT (Embedded Deterministic Testing) reasoning is used to get around this issue.

Different terms, such as flaws, errors, and faults, might be used to describe how an electrical system or equipment is faulty. Electronic system flaws are nothing more than unintentional deviations from the hardware's original design. The primary flaws detected in chips are those related to processes, aging, materials, and packages. When a signal from a malfunctioning system is produced, errors are created. An error is an outcome for which there are some flaws. A fault is nothing more than a defect's representation. The fault portion of the system is the primary focus of this work. Functional testing and structural testing are two categories for the entire testing process. Because structural testing is more efficient than functional testing, it is increasingly frequently used in modern VLSI design.

In the DFT context, the idea of fault modelling is quite important. Levels of fault modeling are taken into consideration before fault modelling. The three levels are behavior level, register transfer level, and transistor level. The notions of RTL level fault modelling are more pertinent to this study. The two RTL level fault models that this project is most interested in are the stalled at and at-speed fault models. The most basic fault modelling is stuck at fault modelling.

A signal line in the circuit is assigned a fixed (0 or 1) value to simulate the fault. A signal line is either the input or output of a logic gate or flip flop. The most common kind of stuck at faults are single stuck at faults, which are two faults per line stuck at 1 (sa-1 or sa1) and stuck at -0 (sa-0 or sa0).

At-speed fault modelling is another type of fault modelling. There is a lot of rivalry in the area of chip designs today about chip size. The chip's geometry will be smaller, and it may have errors or flaws that affect speed.

To compete in the market, the chip must operate at high speeds and frequencies; as a result, even a slight delay effect should have a significant impact on the design. The significance of at-speed testing is now brought up which examine the entire design to see if the system is functional at a particular clock rate. Path delay faults and transition delay faults are two categories for speed testing. At the gate terminals, transition flaws are visible. Path delay errors are measured along a particular path that the user has chosen.

II. INTERNAL SCAN AND ATPG

A. Scan Insertion

Scan operations has three main stages: setup, load/unload, launch/capture. During setup, design is initialized and TDRs are programmed. The next step is load, where scan enable pin is pulled high before loading the value. In accordance with the length of the scan chain, shift cycles are offered. The scan enable will be set to zero to capture the value loaded when the specified number of shift cycles are completed. Therefore, loading and unloading will essentially take place at the same time, with loading occurring through scan-in and unloading occurring through scan-out.

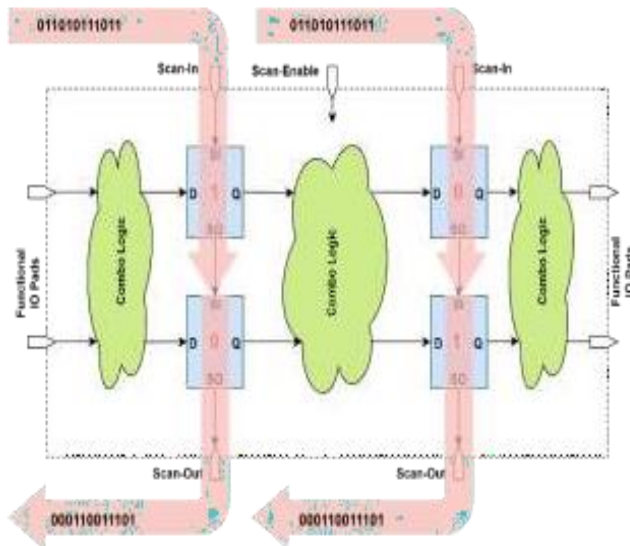

Figure 1 - Loading and Unloading the scan cells (Scan Enable = 1)

The idea of doing scan insertion involves replacing the flip-flops in the sequential circuit with scan flip-flops and applying the test logic to them.

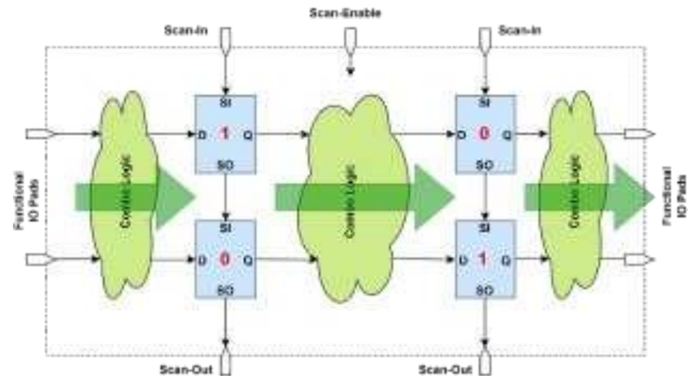

Figure 2 - Capturing the scan cells (Scan Enable = 0)

The scan-operation can be explained in the waveform as below.

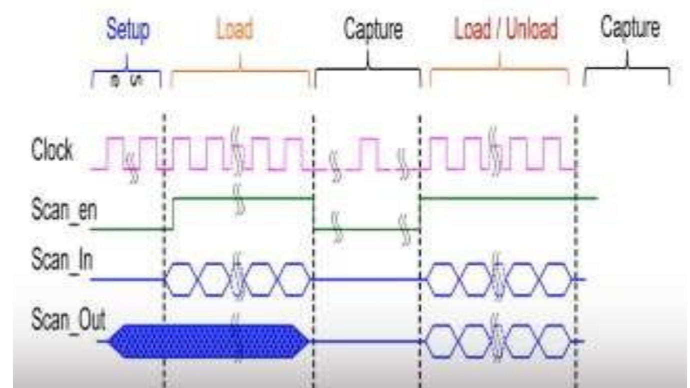

Figure 3 - Setup, load, capture, and unload operations

For scan insertion, some of the necessary input files are offered. These files mostly contain netlists in Verilog format, library files, and TCL scripts. Gate level netlist has to be created as the first step. Then the CTL models are read for the memory, which include the scan chain data. The time delay brought on by extensive wire connections in integrated circuits can be reduced via repeater insertion, which can be done at the RTL itself. Dedicated wrapper flops and shared wrapper flops are two different types of wrapper flops that may be introduced to improve the coverage. wrapper flops include core, input wrapper, and output wrapper logic.

Scan clocks, essential pin constraints, non-scan declaration, and all the other relevant information in the scan insertion constraints are specified.

There are several design rules, including those for RAM, clocks, scan cell data, scannability, scan chain tracing, and power awareness, timelines, etc. All the logic specified for the flops will be stitched together in scan chains, where the maximum scan chain length can be chosen. The updated scan with scan inserted Verilog code for the specified design and two setup files are the real result of the tool.

The first setup-file lists every action taken by the program, while the other has all the data necessary to manage the scan chain.

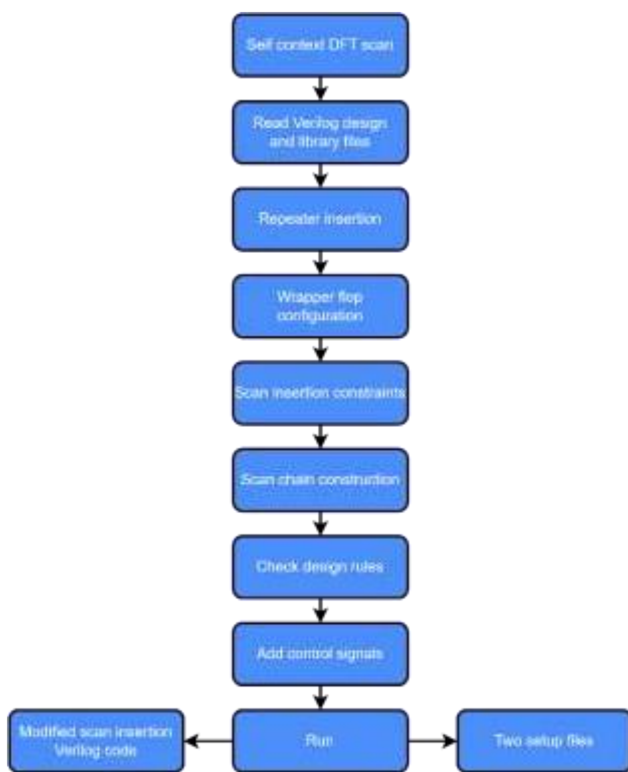

Figure 4 - Design flow for scan insertion Verilog code, and setup-files generation

B. ATPG Pattern Generation

Scan insertion tool generates a few output files, which are used as inputs for the pattern development. necessary patterns are received and coverage data is tested as a result of the pattern creation process. ATPG generate sequence of test patterns which detects the manufacturing defects. These patterns help to determine the cause of failures. ATPG consists of two steps. First step is generating the patterns and performing fault simulation to determine which faults the patterns detect.

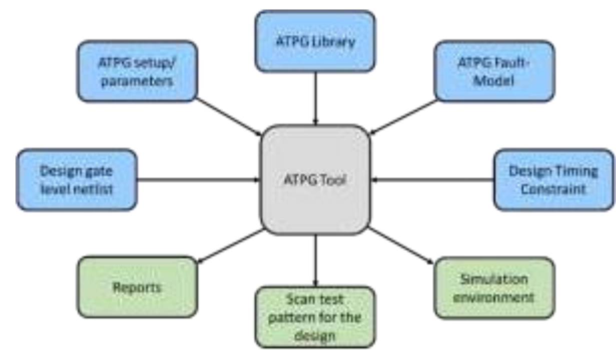

Figure 5 - Block diagram of a typical ATPG tool

After taking all the input files as mentioned in the figure tool will generate the reports such as DRC reports, scan cell reports, scan chain reports and so on. Scan test patterns will be generated by the tool and whatever the patterns generated by the tool can be used for running fault simulation to get the test coverage.

Befor delivering the patterns to test, simulating those patterns is mandatory. There are two types of simulations that can be performed. One is zero delay simulation where the pattern is simulate without adding delay on the gates and the second one is timing simulation where the patterns are simulated by reading the SDF(standard delay format) generated by physical design team.

Upon completion of simulation, signoff checks need to be perfomed where SOC to BLOCK level ATPG comparision is carried out. Here the TDR programming is reviewed from block to SOC and there should not be any mismatch. Pin constraints need to be validated at this stage assuming that the constraints are fully controllable from SOC. It also has to be made sure that there exists no cut-points but if it does, they should be moved to TDR programming.

III. EMBEDDED DETERMINISTIC TEST

As the size of the circuit increases, the volume of test data grows exponentially. Because bigger circuit tests take longer and require more memory to store the test data, the expanding test data volume has a major negative impact on test costs. As a result, test compression techniques are crucial for lowering test costs by lowering scan patterns while attempting to maintain test quality. EDT (EMBEDDED DETERMINISTIC TEST) is one of the most used hardware test compression methods.

Design teams must use new strategies for large designs at advanced process nodes to provide high-quality chips while keeping test time and cost under control. Any strategy that will shorten test duration, keep test expenses in check, and keep test quality high is likely to provide you an advantage over the competition. The outcomes have been observed on several industrial designs that incorporate EDT to demonstrate it's worth. By using EDT to minimize the number of patterns, you will also shorten the time it takes to build the ATPG patterns and validate them in simulation.

EDT consists of a decompressor and a series of scan chains and a compressor logic. EDT means an extra circuitry is embedded in an IC, which generates the deterministic patterns in-order to test the faults in the circuit. The Edt_Clock is given to the input to the decompressor in-order to generate the decompressor patterns which are going as an input to the scan chains. The Edt_Update signal is used to reset the decompressor and the compressor whenever the new patterns or new chains are loaded into the decompressor.

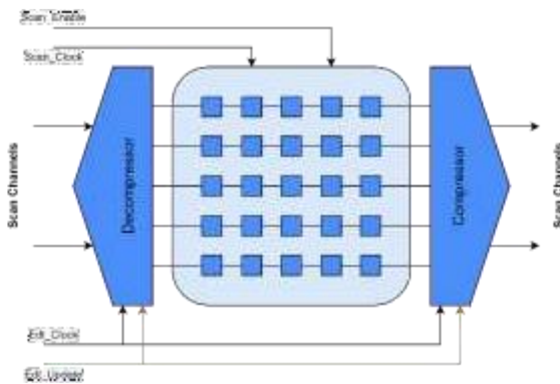

Figure 6 - Embedded Deterministic Test (EDT) mechanism

A decompressor is made up of a number of flip-flops coupled together in a feedback mechanism. The decompressor is made up of LFSR that are coupled by combinational logic. The scan chains are driven by the outputs which are plugged-out by the flipflops.

Compressor consist of an AND gate and XOR circuits. AND gate is used to mask the logics. Comparator XOR is used to compare between the two scan chains.

The Edt_Clock drives the flipflops within the decompressor, which is filled with LFSRs. However, the Scan_Clock is what drives the current scan chains. so, there is a shift in domain from the decompressor to the scan chains leads to the possibility of timing violations. To avoid timing violations, lockup latches are introduced in the circuitry.

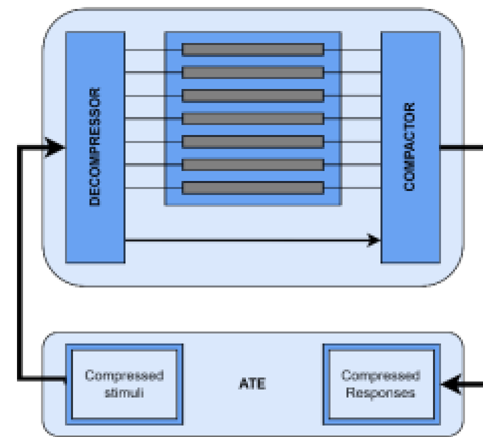

Figure 7 - EDT Block

In the initial stage of the Embedded Deterministic Test (EDT), the IP generation process takes precedence. This involves integrating control signals, external scan channels, decompressor, and compacter modules into the design. Subsequently, the second phase focuses on crafting patterns for the structured EDT designs. Setting the context to 'dft -edt' initiates the process, followed by parsing the input files—comprising library files and scan-inserted Verilog code. Upon thorough review of these files, the pivotal decision of selecting the top design is made. The received setup file, furnished by the tool, is then employed to impart comprehensive information regarding scan insertion.

Following the design rule check, the circuit undergoes thorough examination in the subsequent stage. Simultaneously, the user specifies the number of external scan channels required. In the IP creation phase, essential components like decompressor and compacter modules, control signals, and external scan channels are integrated into the EDT structured circuit. Each step in the procedure is accompanied by a check on the EDT logic's rules.

Transitioning to the next stage involves pattern generation, wherein the modified design transforms into an EDT-structured configuration. These resulting compressed patterns are then dispatched to the external scan channels of the updated design. Subsequently, simulation is executed to compile comprehensive statistics, encompassing pattern quantity, total faults, discovered faults, test coverage, fault coverage, and more.

Comparative Results

A. Without EDT Insertion Result

Fault Classes		#faults (total)	
FU (full)		1752994	
UC	(uncontrolled)	25	(0.00%)
UO	(unobserved)	3744	(0.21%)
DS	(det simulation)	1270571	(72.48%)
DI	(det implication)	180452	(10.29%)
UU	(unused)	13634	(0.78%)
TI	(tied)	76185	(4.35%)
BL	(blocked)	17535	(1.00%)
AU	(atpg untestable)	190848	(10.89%)

Fault Sub - Classes		#faults (total)	
FU (full)			
AU	(atpg-untestable)		
BB	(black_boxes)	1165	(0.07%)
PC*	(pin_constraints)	61414	(3.50%)
TC*	(tied_cells)	71966	(4.11%)
CC*	(cell_constraints)	1	(0.00%)
MPO	(mask_po)	15572	(0.89%)
SEQ	(sequential_depth)	12126	(0.69%)
Unclassified		28604	(1.63%)
UC+UO			
AAB	(atpg_abort)	3769	(0.22%)

*Use "report_statistics -detailed_analysis" for details.

Coverage	
Test_coverage	88.17%
Fault_coverage	82.77%
Atpg_effectiveness	99.78%
#test patterns	1506
#basic_patterns	1494
#clock_sequential_patterns	12
#simulated patterns	1523

B. After EDT logic Insertion

Fault Classes		#faults (total)	
FU (full)		1752994	
UC	(uncontrolled)	6	(0.00%)
UO	(unobserved)	6552	(0.37%)
DS	(det simulation)	1264517	(72.13%)
DI	(det implication)	210968	(12.03%)
UU	(unused)	13634	(0.78%)
TI	(tied)	76185	(4.35%)
BL	(blocked)	17535	(1.00%)
AU	(atpg untestable)	163597	(9.23%)

Fault Sub - Classes		#faults (total)	
FU (full)			
AU	(atpg-untestable)		
BB	(black_boxes)	1165	(0.07%)
EDT	(edt_blocks)	1883	(011%)
PC*	(pin_constraints)	62036	(3.54%)
TC*	(tied_cells)	49950	(2.85%)
MPO	(mask_po)	15569	(0.89%)
SEQ	(sequential_depth)	3413	(0.19%)
Unclassified		29581	(1.69%)
UC+UO			
AAB	(atpg_abort)	6557	(0.37%)
UNS	(unsuccess)	1	(0.00%)

*Use "report_statistics -detailed_analysis" for details.

Coverage	
Test_coverage	89.66%
Fault_coverage	84.17%
Atpg_effectiveness	99.63%
#test patterns	5548
#basic_patterns	5376
#clock_sequential_patterns	172
#simulated_patterns	5554

C. Transition Faults Report for Without EDT Insertion

Fault Classes		#faults (total)	
FU (full)		1752994	
UC	(uncontrolled)	273	(0.02%)
UO	(unobserved)	13109	(0.75%)
DS	(det simulation)	1073610	(61.24%)
DI	(det implication)	191696	(10.94%)
UU	(unused)	13634	(0.78%)
TI	(tied)	120992	(6.90%)
BL	(blocked)	33132	(1.89%)
RE	(redundant)	31227	(1.78%)
AU	(atpg untestable)	275321	(15.71%)
Fault Sub - Classes		#faults (total)	
FU (full)			
AU	(atpg-untestable)		
BB	(black_boxes)	238	(0.01%)
PC*	(pin_constraints)	62690	(3.58%)
TC*	(tied_cells)	66235	(3.78%)
MPO	(mask_po)	14733	(0.84%)
SEQ	(sequential_depth)	15715	(0.90%)
Unclassified		114032	(6.50%)
UC+UO			
AAB	(atpg_abort)	13382	(0.76%)
*Use "report_statistics -detailed_analysis" for details.			
Coverage			
Test_coverage		81.42%	
Fault_coverage		72.18%	
Atpg_effectiveness		99.24%	
#test patterns		3375	
#clock_sequential_patterns		3375	
#simulated_patterns		3422	

D. Transition Faults Report After EDT Insertion.

Fault Classes		#faults (total)	
FU (full)		1752994	
UC	(uncontrolled)	273	(0.02%)
UO	(unobserved)	14656	(0.84%)
DS	(det simulation)	1066454	(60.89%)
DI	(det implication)	222605	(12.71%)
UU	(unused)	13634	(0.78%)
TI	(tied)	120992	(6.91%)
BL	(blocked)	33132	(1.89%)
RE	(redundant)	31227	(1.78%)
AU	(atpg untestable)	250021	(14.26%)
Fault Sub - Classes		#faults (total)	
FU (full)			
AU	(atpg-untestable)		
BB	(black_boxes)	238	(0.01%)
EDT	(edt_blocks)	1652	(0.09%)
PC*	(pin_constraints)	63668	(3.63%)
TC*	(tied_cells)	43348	(2.47%)
MPC	(multicycle_paths)	1678	(0.10%)
MPO	(mask_po)	14730	(0.84%)
SEQ	(sequential_depth)	7106	(0.41%)
Unclassified		117601	(6.71%)
UC+UO			
AAB	(atpg_abort)	14929	(0.85%)
*Use "report_statistics -detailed_analysis" for details.			
Coverage			
Test_coverage		83.04%	
Fault_coverage		73.60%	
Atpg_effectiveness		99.15%	
#test patterns		11782	
#clock_sequential_patterns		11782	
#simulated_patterns		11823	

Comparative analysis

1. Test Coverage Assessment:
For Stuck-At Faults:
Without EDT Insertion:
   - Test Coverage: 88.17%
   - Fault Coverage: 82.77%
After EDT Insertion:
   - Test Coverage: 89.66%
   - Fault Coverage: 84.17%
For Transition Faults:
Without EDT Insertion:
   - Test Coverage: 81.42%
   - Fault Coverage: 72.18%

After EDT Insertion:
   - Test Coverage: 83.04%
   - Fault Coverage: 73.60%

Inference: EDT logic insertion showcases an improvement in both test and fault coverage for both stuck-at and transition faults.

2. Test Efficiency Analysis:
For Stuck-At Faults:
   - Without EDT Insertion: 1506 test patterns.
   - After EDT Insertion: 5548 test patterns.
For Transition Faults:
   - Without EDT Insertion: 3375 test patterns.
   - After EDT Insertion: 11782 test patterns.

Inference: Despite the enhanced coverage, there's a noticeable increase in the number of test patterns when EDT logic is inserted, indicating potential efficiency trade-offs.

3. Overhead Quantification:
   - Notably, with EDT insertion, there's an emergence of additional fault sub-classes like "EDT blocks" and "multicycle_paths" which were not present without EDT.

Inference: This suggests that the incorporation of EDT logic introduces additional elements into the VLSI design that could be contributing to overheads.

4. Trade-off Evaluation:
The incorporation of EDT has shown mixed results:
   - The number of "atpg untestable" faults reduced after EDT insertion for both stuck-at and transition faults, suggesting better testability.
   - However, there was a significant increase in the number of test patterns post-EDT insertion, pointing towards potential efficiency trade-offs.
   - Notably, for stuck-at faults, there's an increase in unobserved faults after EDT insertion, which may raise concerns.

5. Recommendation Formulation:
Given the data:

- EDT logic insertion can enhance test and fault coverage. Designers might opt for EDT when higher fault coverage is a priority.
- However, the surge in the number of test patterns after EDT logic insertion suggests a potential escalation in test application time. If test efficiency is a concern, careful consideration is required before integrating EDT.
- Designers need to weigh the increased fault and test coverage against the additional test patterns and potential overheads introduced by the EDT logic.

IV. CONCLUSION

The study's findings reveal a notable shift in fault dynamics upon implementing EDT (Embedded Deterministic Test) logic insertion. Specifically, without EDT insertion, the dominant fault classes include Deterministic Simulation (DS) with 72.48% and ATPG Untestable (AU) faults contributing to 10.89%. However, post-EDT insertion, there is a mild decrease in DS faults to 72.13%, and a noticeable reduction in AU faults to 9.23%.

The fault sub-classes also changed, particularly within tied cells (TC) and sequential depth (SEQ) post-EDT insertion. There is a significant reduction in tied cell faults from 4.11% to 2.85% and in sequential depth faults from 0.69% to 0.19%.

Regarding transition faults, before EDT insertion, the ATPG untestable faults stood at a substantial 15.71%. This figure saw a reduction to 14.26% after implementing the EDT logic, signifying its efficacy in mitigating such faults.

Furthermore, the test coverage, fault coverage, and ATPG effectiveness showed consistent improvement post-EDT insertion. The number of test patterns required increased, which could be attributed to the added EDT logic complexity.

Overall, the EDT logic insertion demonstrates its potential in optimizing the fault landscape, enhancing coverage metrics, and reinforcing the robustness of the test environment.

FUTURE SCOPE

Enhanced EDT Logic Design: While the current study highlighted the benefits of EDT logic insertion in fault optimization, there's potential for further refinement in the EDT design to ensure even greater reductions in ATPG untestable faults and other prominent fault classes.

Advanced Diagnostic Techniques: Future work can explore the integration of advanced diagnostic algorithms to better understand and classify the newly identified and unclassified faults post-EDT insertion.

Deep Learning and Fault Analysis: With the rise of machine learning, employing deep learning techniques could pave the way for predictive fault analysis, enabling preemptive measures even before fault manifestation.

Holistic Design-for-Test (DFT) Strategies: Beyond EDT, it would be valuable to investigate other DFT strategies in tandem with EDT. Combining multiple strategies might yield a synergetic effect on test and fault coverage.

Real-world Case Studies: While the current study provides a solid foundation, applying the EDT insertion methodology to a diverse set of real-world chip designs can offer insights into its scalability and versatility.

Power and Performance Metrics: Future research could also evaluate the impact of EDT insertion on power consumption and chip performance, ensuring that enhanced testability doesn't compromise on operational efficiency.

Adaptive Testing Environments: Building adaptive test environments that can modify their testing strategies based on the observed faults can lead to faster and more efficient fault detection and correction cycles.

Quantum Computing and Fault Analysis: As the world moves closer to quantum computing, understanding its unique fault dynamics and how EDT or similar methodologies can be adapted for it presents a vast uncharted territory for exploration.

V. REFERENCES

[1] "Test pattern generation and clock disabling for simultaneous test time and power reduction"Jih-Jeen Chen; Chia-Kai Yang; Kuen-Jong Lee 28 February 2003, IEEE, jornel paper.

[2] "Sequential circuit ATPG using combinational algorithms" Xiaoming Yu; M. Abramovici 2013, IEEE.

[3] "Combinational Automatic Test Pattern Generation for Acyclic Sequential Circuits" YongChangKim, Vishwani D. Agrawal, and Kewal K. Saluja 2005 IEEE journal paper.

[4] "High-Level Test Synthesis With Hierarchical Test Generation for Delay-Fault Testability" Sying-Jyan Wang, Member, IEEE, and Tung-Hua Yeh, Student Member, IEEE 2009.

[5] IEEE Standard Boundary Scan 1149.1 An Introduction1991, by Electro International, 1991.

[6] Efficient methodology for boundary scan insertion and pattern generation for MCM based designs. Conference paper 2008.

[7] DFT Methodologies for Reducing Shift Power of Compression Architecture for 28NM ASIC. Conference paper IEEE 2018.

[8] Christos Papameletis, Vivek Chickermane "Optimized Physical DFT Synthesis of Unified Compression and LBIST for Automotive Applications". IEEE- 2019.

[9] S. Emara, D. Romanov, G. W. Roberts, S. Aouini, M. Parvizi, and N. Ben-Hamida, "Optimized periodic bitstreams for DC signal generation used in dynamic calibration applications," IEEE Open J. Circuits Syst., vol. 1, no. 1, pp. 3–12, Mar. 2020.

[10] H. Zhu, W. Yang, G. Engel, and Y.-B. Kim, "A two- parameter calibration technique tracking temperature variations for current source mismatch," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 64, no. 4,
pp. 387–391, Apr. 2017.