Journal of Information Systems Engineering and Management
2025, 10(23s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Analysis of Dual configurations of EDT for Compression Techniques in chip testing in VLSI

[1]Praveen K, [2]Dr Rajanna G S, [3]Dr Shivakumara swamy G M,

Dept of ECE, Srinivas University, Mukka, Mangalore, India

Dept of ECE, Srinivas University, Mukka, Mangalore, India

Dept of Electrical and Electronics, Bapuji Institute of Engineering and Technology, Davangere, India

Corresponding author: Praveen K, Dept of ECE, Srinivas University, Mukka, Mangalore, India

ARTICLE INFO

Received: 20 Dec 2024

Revised: 08 Feb 2025

Accepted: 21 Feb 2025

ABSTRACT

Design For Testability Plays an Important role in the silicon industry so it helps check the faults of the design in the post and pre-integrated chip manufacturing industry. Faults play a vital role in destroying the whole chip, if we are integrating the millions of transistors in a single chip causes many unwanted faults. To reduce the faults, we have so many techniques to reduce the test time and test data volume. Another type of technique we have followed here is dual configuration for EDT. It will be supported for both input configurations. Normally we concentrate on the Dual or Multi Configuration techniques for the design. According to the requirement of the industry level, mostly concentrate on the general configuration on the DFT level. The design will work for both the conditions EDT Configurations which have higher configurations and lower configurations. This design helps to check the possibility conditions of the different configurations; there is no need to change the entire design for the different design configurations for the same netlist. This design will help to reduce the burden of the DFT designer and the cost of the customer. The tool has the privilege of working with the dual set configuration for higher and lower, tool has certain limitations for the dual/Multiple EDT Configurations.

Keywords: SCAN, ATPG, Simulations, EDT, DUAL configuration, IC,

1. INTRODUCTION

Compression is one of the techniques that scan tests for data to be compressed, controlling, a huge numerous of scan chains from internal design with a small number of scan channels. Maintaining a high-quality test (i.e., support for all fault models). Add minimal test logic. By using compression, we can reduce tester time and test data volume.

The main contributions to the Compression are as follows: -

- Reduction of test data volume
- To reduce test time
- To handle with DFT allocated limited pins

- We making sub-chains by using compression
- By decreasing MCL based on compression ratio, we can reduce TT & TD
- Compression ratio = Internal scan chains/External channels (predefined by ATE)

Embedded Deterministic Test (EDT) is the processing technology used by the TestKompress tool, as the design increases then the testing data will be increased exponentially. if we have larger circuits, growth in the test data volume plays a significant role in the increase in test development price because of this test time will be increased as well as the test volume.

EDT is a technology that enhances traditional deterministic Automatic Test Pattern Generation (ATPG) with improved test data compression and reduced test time. Tessent TestKompress utilizes EDT to achieve significant techniques or technology related to scan test data compression. The concept involves controlling multiple internal scan chains through a limited number of scan channels. The key point is that these scan channels behave in a way that is indistinguishable from traditional scan chains from the perspective of the tester point of view. This allows for the application of compressed patterns using testers capable of handling traditional scan patterns. Essentially, it's a method to efficiently compress and manage scan test data during testing processes.

Tessent Shell streamlines the generation of compacted patterns tailored for chip designing processing through EDT controller logic. These condensed patterns, which are similar to traditional ATPG, adhere to ATPG constraints while mitigating bus contention for any testable fault. A set of these compacted patterns resides on the automatic tester equipment, every pattern is to apply the data to decompressor input signals and capture responses at compactor outputs.

During testing ATE supplies the compactor(compressor) pattern to the design circuit via the strategically positioned decompressor, located between the external scan channel pins and internal scan chains. Despite appearances to the tester, the design seemingly features a modest number of scan chains. Post-decompression, the compressed test patterns instigate the required values in the scan chains for effective fault detection.

Design functional input pins and output pins, adhering to conventional testing practices, inputs are directly controlled(forced) and observed(measured) by the ATE. Hardware compactors, positioned post-internal scan chains, minimize the number of internal chains, providing a more manageable quantity of external channels. The compactor condenses responses from scan cells, and the output of the compressed response is scrutinized on the ATE, safeguarding against fault-masking and corruption induced by X-states.

User-defined parameters, encompassing the design with enormous scan channels along with the equipped lockup latches, these integral to the RTL code. The tool autonomously configures the internal structure of the EDT hardware based on specified parameters, including the number and length of internal scan chains, and the clocking specifics of the first and last scan cell in each chain.

Supporting diverse formats, Tessent Shell accommodates serial and parallel test benches in the Verilog format. This versatility ensures compatibility with a broad range of testing environments and methodologies.

Dual configuration is the most common type technique that supports both configurations for the same design for two different test phases. We can reuse the same techniques for the number of inputs and number of outputs, but the generating patterns are considered according to the configuration techniques.

2. LITERATURE REVIEW

The prediction of the pattern counts and the scaling of the flipflops according to the author [1], worked on the multiple inputs and outputs behavior of the tool and how much the data compression of the tool will be taken care of according to the pattern count, test volume and test time compared with the actual design. They explained the RMS values while scaling down the chain length and count of the flops. Time-consuming for different numbers of inputs and outputs of the scan chains.

Estimation of test data volume for scan architectures with the different number of input channels, according to this author[2] they are mainly concentrated on the different number of input channels and output channels on the EDT test coverage, test cycles, and test data volume this is because of the changes in terms of the inputs and output of the scan chain and length of the flops they have written the multi patterns sizes from 1~100 number and how the tools will analysed according the condition. They analyzed the run time comparison of the design and worked on the patterns generated.

Author [3] work is based on the EDT compression based on the number of input and output channels of the compressor using the architecture checking the behavior of the tool about the compression ratio and other parameters. The compression may depend on the min 3% to 40% of the data compression. They are concentrated on the 4.16x compression of the test time and test data volume, on one side they concentrate on changing the input and output volume of channels.

Author [4] worked on the scan design constraints architecture of the scan design styles and the cases. They worked on the RTL design from the simulator and made a comparative analysis of the design.

In this journal, Authors [5] concentrated on the working of the design for testability in the machine learning collaboration so they have generated the ATPG patterns, Compression, MBIST and scan insertion in the machine learning method, by using the machine learning generated the test patterns and simulated all the patterns in the same method. The authors mainly worked on the chip testing flexibility. By applying these methods testing of the chip will be more efficient, reliable, fast, and sustainable ideas to the integrated circuits.

In this paper the authors [6] concentrated on the cell aware pattern generation, another method of the ATPG pattern generation which will easier to check the faults models easily. The author [6] This method works in the reduction of test time and test data volume up to 20 to 22 percent. This methodology worked more effectively and with standardization of the design. This cell aware patterns will check the internal functionalities as per traditionally as stuck at and transitional faults.

In this journal the authors [7] concentrated on the performance of the RSFQ circuits, they concentrated on the fabrication of the CMOS chip to get the desired performance of the chip. They proposed the new timing verification and delay timing information. The authors have worked on the

multiple pipelining system, to overcome this issue. Here they utilize the multicycle path to make it simple to generate the patterns. the characterization of the cell under process of modifications and finding the delay with excitation conditions, sensitization conditions, and propagation of the logic errors caused by timing violations due to process variations.

This author [8] has worked on the XOR-based low power controlling system to reduce the circuit based on the transitions in scan-based testing, this existing mode system should know the novel techniques for the testability. These proposed systems using the genetic algorithm enable the adaptive control system for scan chains according to the usages with X OR encoding capabilities, total time decreases by 20%on on average and up to 47% compared with the existing design without reducing the test coverage.

These authors [9] concentrated on the differentiation of the test power consumption for different types of fault models in the LOC (launch on Capture) and LOS (Launch on shift) models, here they concentrated mainly on the multiple clock domains and investigated the transition fault coverage the main work is on the Low power consumption of the devices through the clocks.

The challenges arising from shrinking transistor sizes and the exponential increase in transistor count in modern circuit designs, make it increasingly difficult to control and observe internal nodes, particularly in sequential circuits. Design for Testability (DFT) emerges as a crucial approach for fault detection in the circuit under test, offering the potential for reducing simulation duration with minimal area overhead. Numerous techniques exist under the umbrella of DFT for pattern simulation, among which scan compression and internal scan are compared in this paper. Through experiments conducted on various benchmark circuits, it is observed that both techniques lead to a significant reduction in simulation time while achieving increased coverage, albeit with a slight area overhead [10]. This comparative analysis provides valuable insights into the performance and trade-offs associated with different pattern simulation techniques under DFT, aiding in the selection of the most suitable approach for efficient fault detection in complex circuit designs.

Design for Testability (DFT), is a technique geared towards organizing designs in a self-regulating manner to enhance testability. It emphasizes the predominant use of the scan pattern method, which involves modifying the internal sequential circuitry of designs. Focusing on frequently employed industry-standard functional Register-Transfer Level (RTL) designs, the manuscript adopts a structured DFT approach for scan insertion and Automatic Test Pattern Generation (ATPG). Through this methodology, controllability and observability for clocks and reset signals in the chosen RTL designs are improved by addressing S rule and D rule violations with the addition of test logic. Notably, the proposed methodology [11] achieves high fault coverage and test coverage rates, exemplified by a Wallace tree multiplier design. This comprehensive approach not only identifies testable and untestable faults but also conducts fault simulation to detect faults using deterministic patterns generated by ATPG. Overall, the systematic methodology outlined in the paper contributes to advancing testability in RTL designs, providing valuable insights into fault detection and coverage, essential for ensuring the reliability and efficiency of complex integrated circuits.

The paper [12] presents a novel approach to path selection procedures in the context of identifying path delay faults for test generation. While traditional path selection criteria focus on positive indicators such as selecting the longest paths, these paths are often untestable, leading to challenges when dealing with a large number of paths. To address this issue, the paper introduces a negative path selection procedure, termed a path unselected procedure, which excludes paths from consideration for test generation. This approach aims to identify detectable path delay faults among a reduced set of paths, providing greater flexibility in test generation. The procedure involves linear-time traversals of the circuit to unselect fan-out branches, thereby excluding shorter paths from consideration. Additionally, it leverages information about detectable and undetectable path delay faults to refine the selection process further. The efficacy of this procedure is demonstrated through its application to benchmark circuits within an iterative test generation framework, to target manageable numbers of the longest path delay faults.

Rapid Single Quantum Flux [13] (RSFQ) logic, leveraging Josephson Junctions (JJs), is gaining renewed attention as a high-performance alternative beyond the physical scaling limits of CMOS technology. Its fabrication processes, characterized by larger feature sizes, result in significantly lower defect densities than CMOS [32]. Consequently, process variations and RSFQ-specific non-idealizes emerge as primary contributors to chip failures. Given its quantized pulse-based operation, RSFQ logic can interpret highly distorted pulses correctly but may suffer timing variations. Therefore, timing verification and delay testing assume critical importance in RSFQ technology. This paper delves into novel phenomena in RSFQ, particularly the emergence of single-pattern delay tests and the necessity to propagate delayed values through multiple pipeline stages. The study identifies conditions for delay excitation, sensitization, and propagation of logic errors induced by process variations by characterizing cells under process variations. Subsequently, a new Automatic Test Pattern Generation (ATPG) paradigm is proposed, leveraging these phenomena to select target delay subpaths and generate test patterns guaranteed to excite worst-case delays along each subpath. Monte Carlo simulation results for benchmark circuits with process variations validate the effectiveness of the proposed ATPG vectors.

The traditional approach to these analyses often involves long runtimes and significant disk storage due to the numerous cycles in ATPG patterns [14]. The paper proposes power and thermal predictions for test applications to mitigate these challenges. To enhance runtime efficiency, multiple machine learning (ML) models are utilized for power prediction, while decay surface models are employed for thermal prediction, reducing computational time. Furthermore, to optimize storage utilization, features are extracted from flip-flop values, eliminating the need for internal logic values from gate-level simulation. The results demonstrate promising accuracy, with mean absolute percentage error (MAPE) for power prediction less than 8% and mean absolute error (MAE) for thermal prediction less than 1.2°C. Moreover, the proposed approach enables transient thermal analysis of long ATPG patterns with a significant speedup in runtime (75X) and reduction in storage requirements (118X). Additionally, the scalability of predictions with test speed allows for the optimization of test time while ensuring thermal safety, enhancing the overall efficiency of the testing process.

The critical need for systematic evaluation of fault coverage and diagnosis capability in train traction systems [15], essential components of urban rail transit systems. Given the complexity and inherent risk of internal faults within these systems, assessing the effectiveness of testing equipment is vital for ensuring operational reliability and safety. To address this challenge, the study proposes the application of testability technology to analyze the test capability of train traction systems. In response to the uncertainties of actual tests, the study introduces a method for constructing fault diagnosis strategies tailored to unreliable testing conditions. Introducing the concept of test credibility, quantitatively evaluated using a cloud model, the study establishes a correlation matrix between fault occurrence and test credibility. Leveraging this framework, the study develops a single-fault diagnosis strategy for traction systems, employing information theory for comparative analysis

2.1 OBJECTIVE OF PROPOSED WORK:

1. Dual/Multi-channel configuration of the design.
2. Checking the Input and output channel configuration for the design.
3. Tool architecture of the Design Configuration
4. Report on the Design analysis.
5. Checking the limitation of the dual channel configuration of the tool.
6. Generate the DRC free violated scan and chain configurations
7. Generating the statical report for both the blocks.
8. Performing the debug for the Generated patterns.
9. Checking the coverage analysis for the dual blocks

3. METHODOLOGY

Block diagram of dual EDT configuration shown in figure 1, which contains both configurations high and low having different inputs. Inputs are directly given to the decompressor from high configuration to decompressor to the internal design to the compactor high configuration output. Otherwise, inputs to low configurations to decompressor to internal design to compactor.

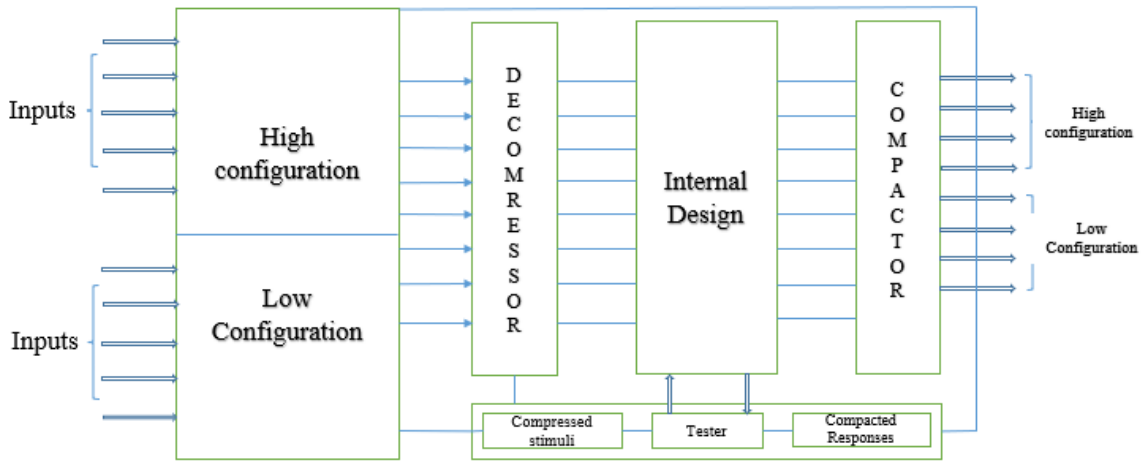


Figure 1: Block diagram of Dual configuration of EDT

Dual configurations of the DFT process are done from the inputs for the scan design process after they are converted into the low configuration and high configurations as the EDT process then test patterns are generated.

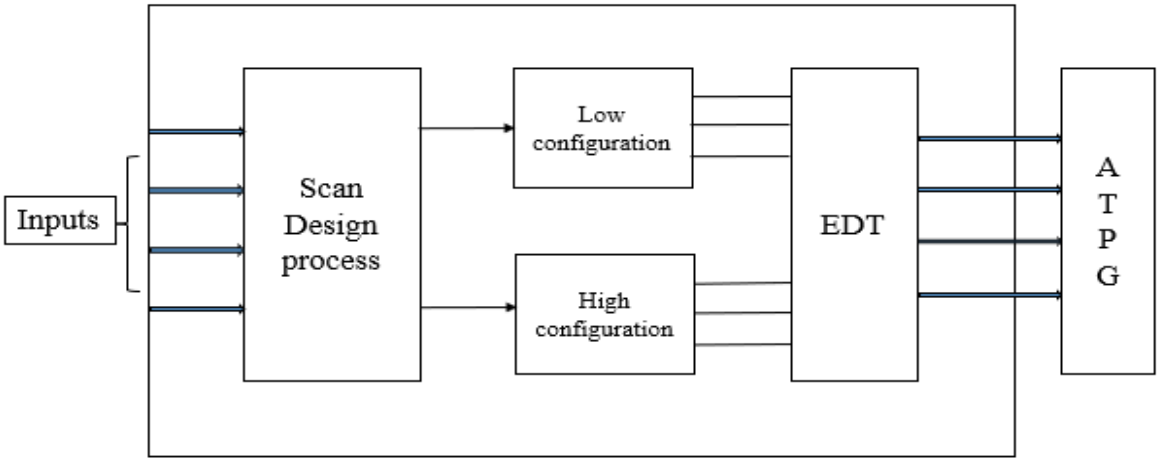


Figure 2: Dual Configuration DFT process

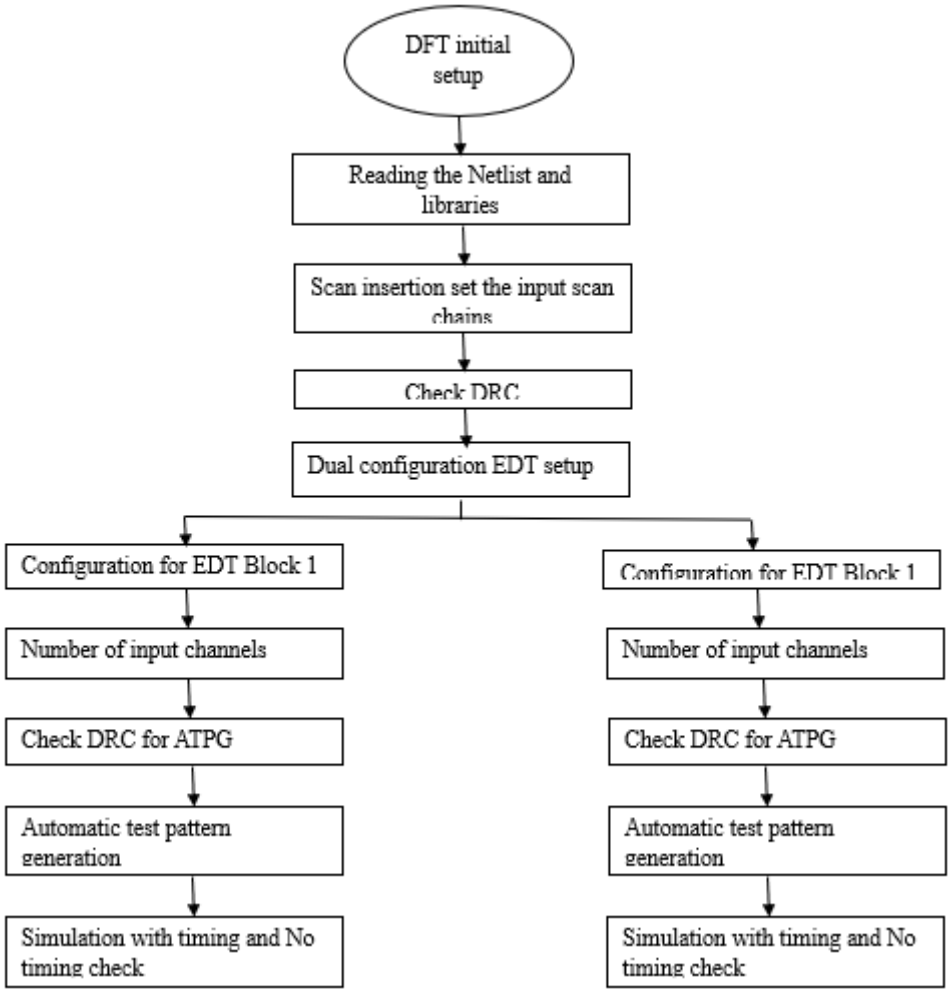


Figure 3: Flowchart of the Design for Dual configuration

The design and testing process for electronic circuits entails a series of meticulous steps to ensure functionality, reliability, and testability. As the above figure 3 explains flow chart represents the complete procedure for the dual configuration of EDT with different numbers of input channels. If any of the customers have the same design with a different number of inputs, effortlessly for single we can configure for both designs. It begins with the critical task of reading the Netlist and libraries, which provide comprehensive details of the circuit components and their characteristics. Following this, scan insertion becomes imperative to facilitate efficient testing, as it involves integrating input scan chains into the circuit design. Subsequently, we are checking the Design Rule Check (DRC) is conducted to validate compliance with testing guidelines and scan stitching properly. The setup of the Dual Configuration in Embedded deterministic Testing (EDT) will check the test time and test data volume of the design. We will check the fault coverage and reliability by configuring redundant components. Further, DRC checks are performed specifically tailored for Automatic Test Pattern Generation (ATPG), ensuring compatibility with design rules. The Automatic Test Pattern Generation process then automatically generates test patterns crucial for fault detection. Finally, simulations are conducted, both with and without timing considerations, to validate functional correctness and timing constraints. Each of these steps is indispensable, collectively culminating in a meticulously designed and thoroughly tested electronic circuit.

3.1 Dual Compression Configurations:

Enabling seamless setup and reuse of EDT logic for dual distinct test phases, like wafer (silicon slice) test and package(post-fabrication) test, involves initiating dual compression during the EDT logic configuration process as shown in figure 2. This entails defining separate configurations, thereby requiring the generation of an extra EDT pin to designate the active configuration. Each configuration prompts the creation of unique ATPG do-files and test procedure files.

In the case of the bypass mode, a single do-file and test procedure (proc) file are generated. The ATPG files resulting from these configurations are then utilized independently to generate test patterns for each specific configuration, mirroring the process for a single compression configuration. This approach ensures efficient handling of diverse test phases while maintaining a structured and organized testing framework. Within the modular flow, it is crucial to monitor compression properties usages among designed group guarantees. compression configurations are set up appropriately for each block as follows:

- A singular dual configuration or a pair of compression configurations is designed for the entire block, for all EDT blocks. This flexibility arises from the recognition that the parameters governing each EDT block within the design may diverge, necessitating distinct configurations.
- The channel parameters for each of the dual configurations exhibit variability across different blocks. To illustrate, in Block b1, the configuration named config_high features distinct parameters, specifically 2 input channels and 4 output channels. On the other hand, block b2, under the same configuration name (config_high), possesses different parameters with 1 input and 1 output channel, as visually depicted in the accompanying figure 3. This bespoke configuration approach allows for fine-tuned adjustments tailored to the unique characteristics of

each block. Figure 4 shows the dual configurations of the design for high and low for block1 and block 2

```
set_current_edt_block b1
set_current_edt_configuration config_high
set_edt_options –input 2 –output 4
set_current_edt_configuration config_low
set_edt_options –input 4 –output 5
set_current_edt_block b2
set_current_edt_configuration config_high
set_edt_options –input 1 –output 1
set_current_edt_configuration config_low
set_edt_options –input 3 –output 3
```

Figure 4: shows the Dual configuration of the EDT

- To establish a unified compression for a specific block, simply specify parameters exclusively for a single configuration. This streamlined approach eliminates the need for redundant configurations, ensuring clarity and efficiency in the configuration process for that particular block.

3.1.1 Limitations

Dual configurations pose a restriction wherein the configuration featuring a higher number of input channels must be either equal to or have the majority of a higher config of the output channels should be more than the other configuration. For instance, figure 5 below illustrates valid configurations where each case of the configuration having a higher input channel number and also boasts an equal or greater number of output channels.

Config1 = 4 input channels and 2 output channels
Config2 = 2 input channels and 1 output channels
Config1 = 2 input channels and 2 output channels
Config2 = 4 input channels and 2 output channels

Figure 5: Valid configurations for each configuration channel

Figure 4 shows the valid configurations for all the input and output channels. The configurations depicted in figure 6 below are deemed invalid, as, in each instance, the configuration with a higher input channel count exhibits a lower number of output channels. This inconsistency contravenes the specified criteria and renders these configurations non-compliant.

Config1 = 4 input channels and 1 output channels
Config2 = 2 input channels and 2 output channels
Config1 = 2 input channels and 2 output channels
Config2 = 4 input channels and 1 output channels.

Figure 6: Not Valid configurations for each configuration channel

Figure 6 shows valid configurations for the input and output channels. Explicitly defining channels for the high-compression configuration is not possible. By default, the high-compression

configuration utilizes the same input and output channels as those initially specified for the low-compression configuration.

The bypass mode is exclusively applicable to the lowest compression configuration. While it's possible to define the number of bypass chains for either configuration, it's imperative to ensure that the specified number does not surpass the count of input/output channels within the lowest compression configuration. This constraint safeguards the integrity and feasibility of the bypass mode operation.

Configuration 1 = 2 input channels and 2 output channels
Configuration 2 = 4 input channels and 4 output channels
The maximum number of bypass chains = 4

Figure 7. Bypass Mode channels for Low configuration

3.1.2 Dual Compression Configurations

Below are the steps to create dual EDT logic that has dual configuration for a single block(b1). The main requirement is Scan chains must be defined.

Procedure for how to create the dual configuration

1. Invoke Tessent Shell. Path of the tool has to be provided <home/tool/bin/tessent_home -shell

Figure 8 commands show the inputs of the design of the netlist. The tessent tool starts in setup mode. Which loads the necessary files for tool requirements like licenses and other supporting files

2. Provide the input for Tessent Shell commands

```
set_context dft -edt
read_verilog my_gate_scan.v
read_cell_library my_lib.aptg
set_current_design top
```

Figure 8. Tessent tool shell commands

3. Define the first compression configuration. First EDT compression name has to be provided, here we have taken config_high and set the input and output channels as 6 and 5 as shown in figure 9.

```
add_edt_configurations config_high
set_edt_options -input_channels 6 -output_channels 5
```

Figure 9: High configuration EDT having 6 input and 5 output channels

4. Define the second configuration. The second EDT compression name has to be provided, here we have taken config_low which has 2 input and 2 output channels. For example:

```
add_edt_configurations config_low
set_edt_options -input_channels 2 -output_channels 2
```

Figure 10: Low configuration EDT having 6 input and 5 output channels

For low configuration commands input for the design. Single configuration for a design block is the particular way to define parameters for single compression.

5. Set up the remaining parameters for the EDT logic.

6. Run_DRC and fix any violations. DRC will be performing the checking of the violation that occurred in the design, then it is automatically switching setup mode by running the command “set_system_mode analysis” or “check_design_rules”.

7. Initiate the generation of the EDT logic. For comprehensive details, individual dofiles and procedure files are crafted for each configuration. The configuration name seamlessly integrates with the designated prefix through the write_edt_files command. Upon specifying the EDT, employ the command write_edt_files to generate the requisite files constituting the EDT.

For example**: write_edt_files Newly**

In the naming convention, "newly" functions as the string prefix appended to the files, and you can provide the switch "-replace", enables the tool if any pre-existing files having the same name will be replaced. The specific EDT logic files affected by this naming convention depend on the placement of the EDT logic, encompassing the following:

- ✓ **Newly_edt_top.v** (external EDT) — This is a Top_level wrapper cell which will instantiate the EDT core logic circuitry and output channel which are allocated with the multiplexers.
- ✓ **Newly_edt_top_rtl.v** (internal EDT) — This file contains the Core design netlist with the integration of EDT logic seamlessly linked between internal scan chains and I/O pads, all while lacking a gate_level description of the EDT logic.
- ✓ **Newly_edt.v** — this file contains the RTL logic design description.
- ✓ **Newly_core_blackbox.v** (external EDT) — This file contains the Black boxes description present in the core of the synthesized netlist.
- ✓ **Newly_dc_script.scr** — In this which contains the DC synthesis script.
- ✓ **Newly_rtlc_script.scr** — in this file contains EDT logic RTL Compiler synthesis script.
- ✓ **Newly_edt.dofile** — This file contains the Dofile for the ATPG supporting for pattern generation.
- ✓ **Newly_edt.testproc**— This procedural file is also required for the supporting files for ATPG pattern generation.
- ✓ **Newly_bypass.dofile** — This Bypass Mode Dofile is required for uncompressed ATPG patterns generation
- ✓ **Newly_bypass.testproc**— Procedure file is for bypass mode for an uncompressed ATPG. In this file contains the set up the EDT clock how much time required to pulse before the scan chain. For shifting the clocks we have “*-pulse_edt_before_shift_clock”* switch along with set_edt_options command. Default, the EDT and scan chain will be undergoing shift clocks

simultaneously pulsing. Configuring the EDT logic in this manner imparts independence from the scan chain clocking, offering the following advantages:

• Simplifies the process of developing EDT logic for a design during RTL logic performing since there's no need for scan chain clocking information. Further details on creating EDT logic at the design stage can be found for reference.

• Eliminates the necessity for lockup latches present in between the scan chains EDT logic due to the guaranteed correct timing facilitated by the clock.

• Streamlines clock routing by having the lockup laches for bypass mode scan chains. These chains are powered by the EDT clock rather than system clocks. This eradicates the requirement to route system clocks to the EDT logic. To leverage this feature, the shift clock speed must be capable of accommodating two distinct clock pulses within a single shift cycle, potentially leading to an increase in test time.

4. RESULT

The results are based on the number of configurations of the EDT architecture can able to support the analysis are made depending on the architecture of the design without modifying the scan configuration. This helps reduce time and effort of the designer the following results are taken from the tessent mentor graphics tool.

```
set_context dft -edt
read_verilog scan_outputs/scan_stitched.v
read_cell_library tsmc13.mdt
set_current_design
dofile scan_outputs/ATPG_setup_files.dofile
tessent_scan_setup
add_edt_configuration high_config
set_edt_options -channels 5
add_edt_configurations low_config
set_edt_options -channels 2
set_edt_pins bypass my_bypass_pin
set_edt_pins configuration my_configuration_pin
set_system_mode analysis
report_edt_configurations -all
report_edt_pins
write_edt_files dual_outputs/edt -verilog -replace
~
~
~
~
```

Figure 11: Dual configuration of EDT.

Figure 11 shows the setting up of the EDT configurations with high and low configurations having the input 5 channels and output 5 channels for high configurations. Low configurations have 2 inputs and 2 outputs. These are the above commands which are provided for EDT design. Write_edt_files commands provide the supporting files for the ATPG like testproc file and dofiles separately for both the configurations.

```
//
set time scale 1.000000 ns ;
set strobe_window time 10 ;
timeplate gen_tp1 =
    measure po 10 ;
    pulse clock 20 10 ;
    period 40 ;
end;
procedure test_setup =
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
        force SCAN_EN 0 ;
        force TESTMODE 1 ;
        force edt_clock 0 ;
    end;
end;
procedure shift =
    scan_group grp1 ;
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
        force_sci ;
        force_edt_update 0 ;
        measure_sco ;
        pulse edt_clock ;
        pulse tx_clk ;
    end;
end;
procedure load_unload =
    scan_group grp1 ;
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
        force SCAN_EN 1 ;
        force TESTMODE 1 ;
        force edt_clock 0 ;
        force edt_update 1 ;
        force my_bypass_pin 0 ;
        force my_configuration_pin 0 ;
        force reset_tx 1 ;
        force tx_clk 0 ;
        pulse edt_clock ;
    end ;
    apply shift 26;
end;
```

Figure 12: EDT Test proc file.

Figure 12 shows the example of the test proc file produced by the tessent tool it contains the time plates for the shift, capture and unload procedure of the high configurations.

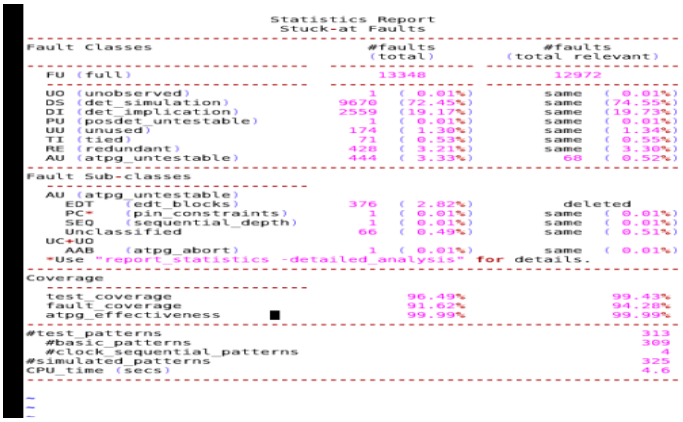


Figure 13: Statiscal report

Figure 13 shows the statistical report for the design constraints for fault classes with the coverage report. In this report, we can able to see the number of patterns present and the CPU usage time is also provided.

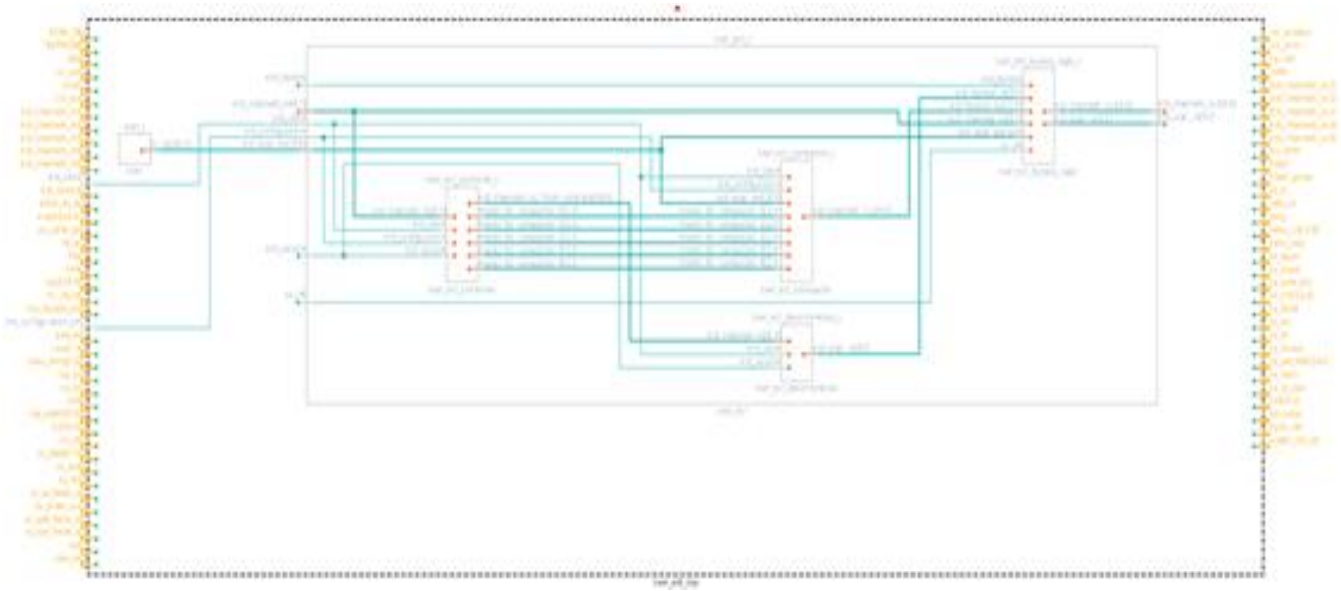


Figure 14: High Config EDT architecture

The tool provided architecture for the high configurations which contain the input and output channels configured for the decompressor and compactor, this architecture is taken from the open_visualizer by using tree view of the block. It provided the complete flow of the design having the 5 number of input channels and 5 number of output channels. Decompressor size and internal design channels are provided. The compressor is taken from the internal design channels into the output channels as shown in Figure 14

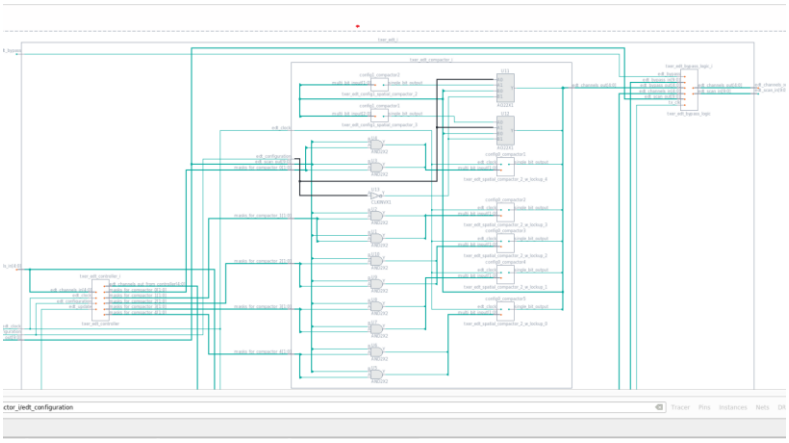


Figure 15: Inside the Compactor Design

Figure 15 shows the internal diagrams of the compactor architecture which will have the internal design and the compactor connections done internally by the tool. After EDT is inserted the tool undergoes ATPG(automatic test pattern generation). Patterns are generated for the serial scan, serial chain, Parallel scan, and parallel chain.

While generating the ATPG patterns we need to take care of the violations present in the design. Resolve the issue for the DRC generate the patterns and start the simulation. Above figure 16 represent the waveforms of the parallel chain. "parallel chain" describes a specific implementation or optimization technique when computing DFTs in parallel on multiple processing units or cores in a parallel computing environment. This could involve breaking down the DFT computation into smaller parts and processing them concurrently on different computing resources to speed up the overall calculation.

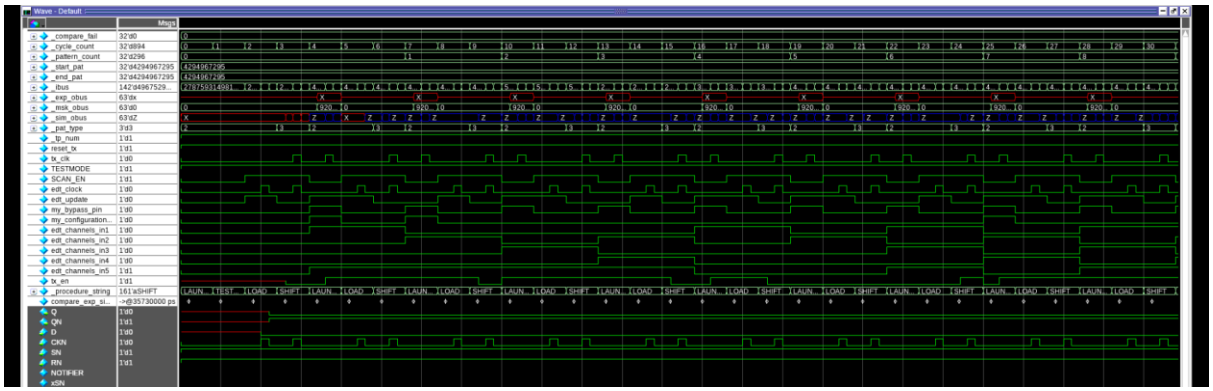


Figure 16: Waveforms for Parallel chain

While generating the ATPG patterns we need to take care of the violations present in the design. Resolve the issue for the DRC generate the patterns and start the simulation. Above figure 16 represents the waveforms of parallel chains. "parallel chain" describes a specific implementation or optimization technique when computing DFTs in parallel on multiple processing units or cores in a parallel computing environment. This could involve breaking down the DFT computation into smaller parts and processing them concurrently on different computing resources to speed up the overall calculation.

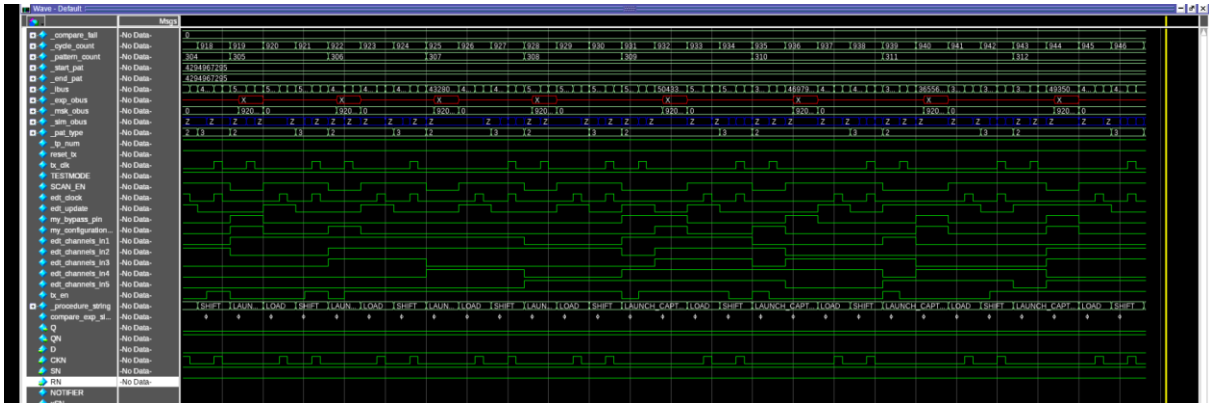


Figure 17: waveforms for the serial chain.

"serial chain" is a term used to describe a specific method of testing electronic circuits to ensure their functionality and reliability. DFT is an important aspect of designing integrated circuits (ICs) and other electronic systems to make them more easily testable during manufacturing and throughout their operational life. Figure 16 shows the Serial chain waveforms.

Serial scan in DFT refers to a method of applying test patterns and capturing test responses through a serial chain of storage elements within a digital circuit. This technique is essential for ensuring the reliability and testability of integrated circuits. These storage elements are taken into consideration for checking the filpflop by changing the input parameters and observing the outputs. Figure 18 shows the waveforms for the serial scan elements.

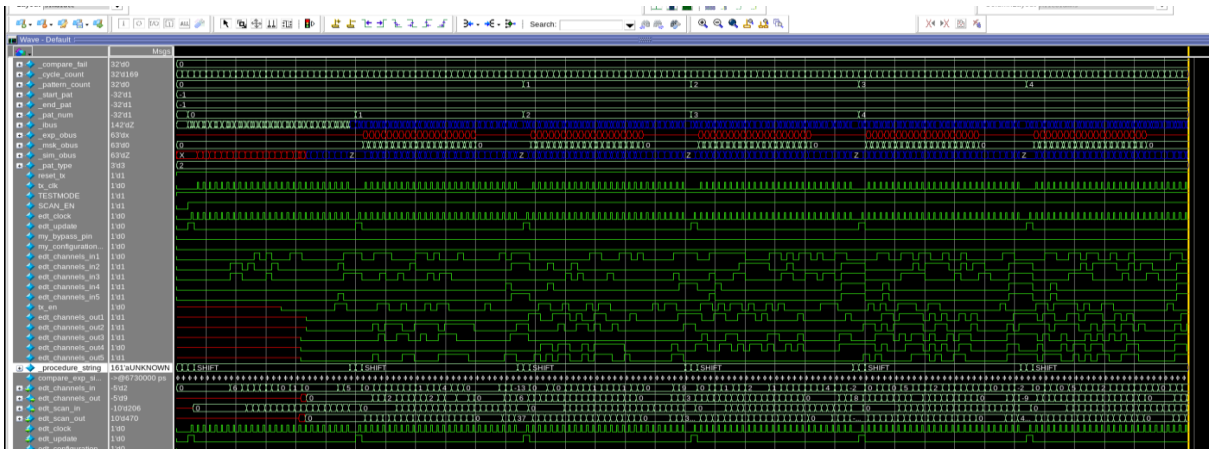


Figure 18: Waveforms for serial scan

Parallel scan is a technique used to facilitate the testing of integrated circuits. It's an alternative to the more common serial scan approach, allowing for the efficient application of test patterns and the capture of test responses within a digital circuit. Parallel scans can be advantageous in certain situations, particularly when testing large and complex digital circuits. Figure 18 shows the waveforms for the parallel scan

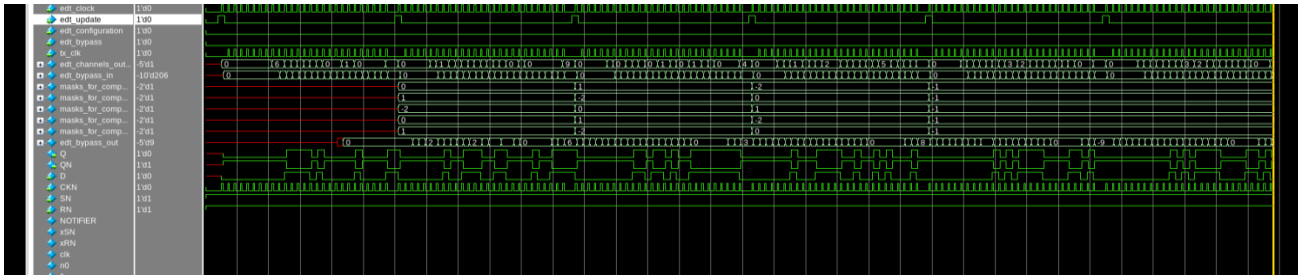


Figure 19: waveforms for parallel scan

3.1.3 Low Config Pattern Generation

Design for Testability (DFT) is an approach used to make ICs easier to test during manufacturing and throughout their operational life. The goal is to detect and diagnose faults or defects within the IC efficiently. Low configuration pattern generation in DFT is used to strike a balance between achieving sufficient test coverage and minimizing the resources and time required for testing. The main intention of this configuration is to reduce the human effort for multiple inserting of the scan design. The design will be helpful for both input configuration there is no need to do the scan insertion the tessent tool will take care of all the input configurations.

```
#SET CONTEXT TO PERFORM ATPG
set_context patterns -scan
# READ THE SCAN INSERTED  NETLIST AND LIBRARY
read_verilog /home/varaprasad/praveen/case1/feb23/level1/case11/dual_outputs/edt_edt_top_gate.v
read_cell_library tsmc13.mdt
# SET THE TOP MODULE
set_current_design
add_black_boxes -auto
#READ EDT INSERTED DOFILE

#dofile /home/varaprasad/praveen/case1/feb23/level1/case11/dual_outputs/edt_high_config_edt.dofile
dofile /home/varaprasad/praveen/case1/feb23/level1/case11/dual_outputs/low_config/edt_low_config_edt.dofile

set_system_mode analysis

set_fault_type -help
set_fault_type Stuck
add_faults -all

report_edt_configurations
create_patterns
#check_design_rules
report_faults -all >> dual_reports1/all_faults.rpt
#create_patterns
report_statistics >> dual_reports1/report_statistics.rpt
#GENERATE ATPG PATTERNS
write_patterns dual_patterns_outputs_low/dual_edt_serial_scan_patterns_outputs_low/txer_stuck_serial.v -verilog -PATTERN_SEts scan -serial -rep
write_patterns dual_patterns_outputs_low/dual_edt_parallel_scan_patterns_outputs_low/txer_stuck_parallel.v -verilog -PATTERN_SEts scan -parallel -rep
write_patterns dual_patterns_outputs_low/dual_edt_parallel_chain_patterns_outputs_low/txer_stuck_parallel_chain.v -verilog -PATTERN_SEts chain -parallel -rep
write_patterns dual_patterns_outputs_low/dual_edt_serial_chain_patterns_outputs_low/txer_stuck_Serial_chain.v -verilog -PATTERN_SEts chain -serial -rep
```

Figure 20: Low Config pattern generation

Figure 20 shows the generating the patterns for different configurations, while performing the ATPG pattern generation we need to take that we have to read the different do files for low and high configuration with the EDT inserted netlist. Create pattern command used to generate all the set of patterns like serial and parallel (scan and chain). We have to write the patterns separately in different directories separate testbenches are created for every pattern.

Figure 20 shows the test proc file for the low configuration. Test procedure" or "testproc" typically refers to a document or set of instructions that outlines the steps for testing an integrated circuit (IC) or other electronic design. A test procedure may include details on how to apply test patterns, set up test equipment, and perform various tests to verify the functionality and performance of the design. This test proc file contains the shift, capture load, and unload procedure will be represented here there

is one difference between the high and low configurations is my_configuration_pin, it is forced to make it 0(Zero) for high configuration, and if it is 1(one) then considered as low configuration pattern generation. Depending upon the number of chain lengths we can make the number of shift cycles.

```
//
set time scale 1.000000 ns ;
set strobe_window time 10 ;

 timeplate gen_tp1 =
    force_pi 0 ;
    measure_po 10 ;
    pulse_clock 20 10 ;
    period 40 ;
 end;

 procedure test_setup =
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
         force SCAN_EN 0 ;
         force TESTMODE 1 ;
         force edt_clock 0 ;
    end;
 end;

 procedure shift =
    scan_group grp1 ;
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
         force_sci ;
         force edt_update 0 ;
         measure_sco ;
         pulse edt_clock ;
         pulse tx_clk ;
    end;
 end;

 procedure load_unload =
    scan_group grp1 ;
    timeplate gen_tp1 ;
    // cycle 0 starts at time 0
    cycle =
         force SCAN_EN 1 ;
         force TESTMODE 1 ;
         force edt_clock 0 ;
         force edt_update 1 ;
         force my_bypass_pin 0 ;
         force my_configuration_pin 1 ;
         force reset_tx 1 ;
         force tx_clk 0 ;
         pulse edt_clock ;
    end ;
    apply shift 26;
 end
```

Fig 21: Low config test proc file

Figure 21 shows the low configuration test procedural file. While doing the ATPG pattern generation for stuck-at faults and transition delay faults, the statistical reports are to be considered. Here FU(full) class can be the ATPG tool that can be testable that maybe the unobserved, detectable simulation, detectable implication, possible detectable untestable, unused pins, tied pins, redundent, and ATPG untestable shown in figure 22.

```
                    Statistics Report
                     Stuck-at Faults
-------------------------------------------------------------------------
Fault Classes                    #faults          #faults
                                 (total)       (total relevant)
-------------------------------  ----------  -----------------------
  FU (full)                         13348            13098
  -----------------------------  ----------  -----------------------
  UO (unobserved)                       1   ( 0.01%)        same   ( 0.01%)
  DS (det_simulation)                9670   (72.45%)        same   (73.83%)
  DI (det_implication)               2685   (20.12%)        same   (20.50%)
  PU (posdet_untestable)                1   ( 0.01%)        same   ( 0.01%)
  UU (unused)                         174   ( 1.30%)        same   ( 1.33%)
  TI (tied)                            71   ( 0.53%)        same   ( 0.54%)
  RE (redundant)                      428   ( 3.21%)        same   ( 3.27%)
  AU (atpg_untestable)                318   ( 2.38%)          68   ( 0.52%)
-------------------------------------------------------------------------
Fault Sub-classes
  -----------------------------
  AU (atpg_untestable)
    EDT    (edt_blocks)               250   ( 1.87%)          deleted
    PC*    (pin_constraints)            1   ( 0.01%)        same   ( 0.01%)
    SEQ    (sequential_depth)           1   ( 0.01%)        same   ( 0.01%)
    Unclassified                       66   ( 0.49%)        same   ( 0.50%)
  UC+UO
    AAB    (atpg_abort)                 1   ( 0.01%)        same   ( 0.01%)
  *Use "report_statistics -detailed_analysis" for details.
-------------------------------------------------------------------------
Coverage
  -----------------------------
  test_coverage                      97.48%                99.44%
  fault_coverage                     92.56%                94.33%
  atpg_effectiveness                 99.99%                99.99%
-------------------------------------------------------------------------
#test_patterns                                              321
  #basic_patterns                                           317
  #clock_sequential_patterns                                  4
#simulated_patterns                                         332
CPU_time (secs)                                             4.6
-------------------------------------------------------------------------
```

Figure 22: Low Config Statistical Report.

In ATPG untestable(AU) means the tool isn't able to test the particular design level due to some reasons like edt blocks, pin constraints, and sequential depth, and the overall report is provided in the Low Configuration Statiscal Report.

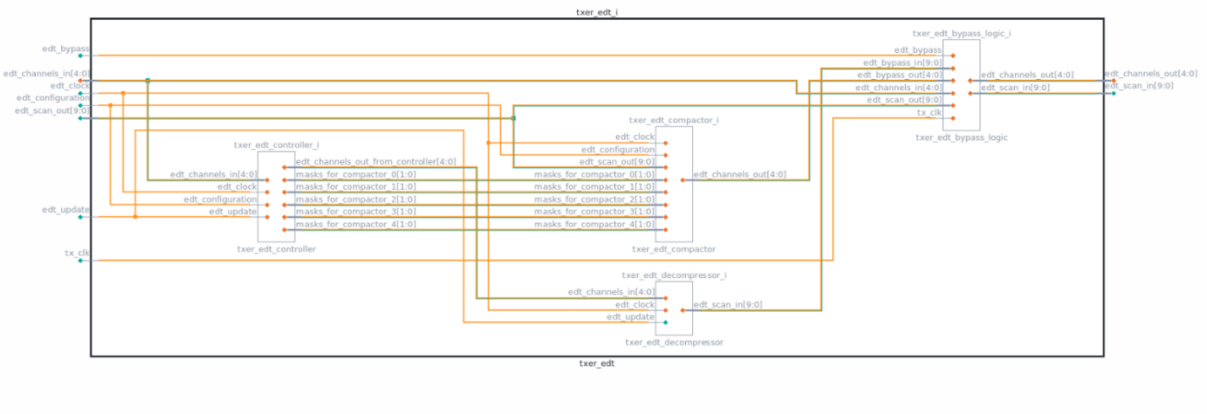


Figure 23: Low config Architecture

Similarly, the figure shows the architecture of the low configuration which we have seen in this diagram as shown in figure 23 and figure 24 shows an internal diagram for the low configuration internal diagram on the compactor.

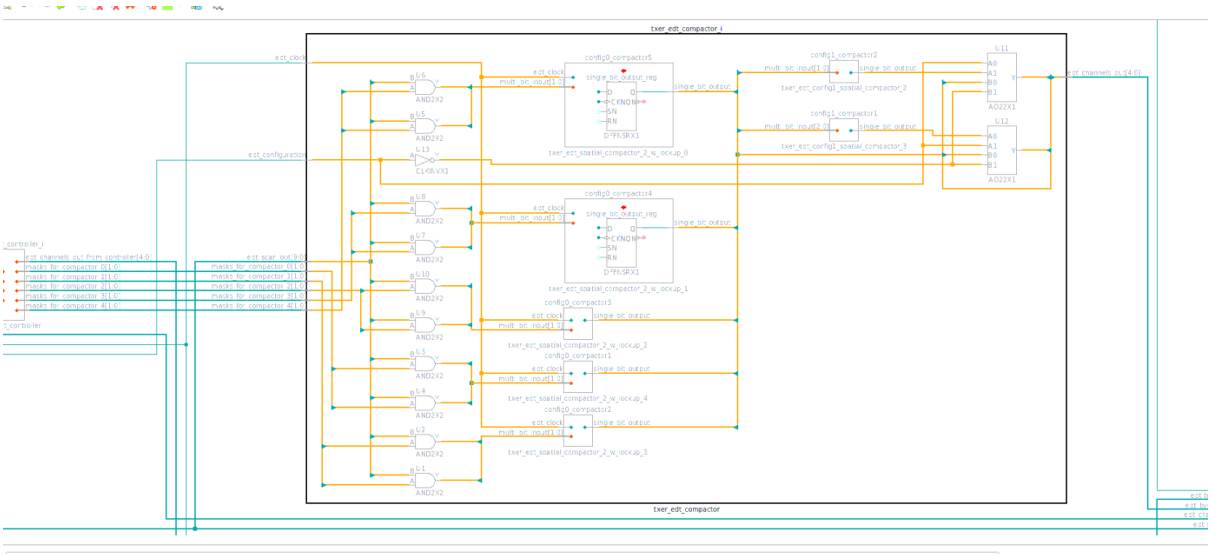


Figure 24: Low Configuration compactor internal diagram

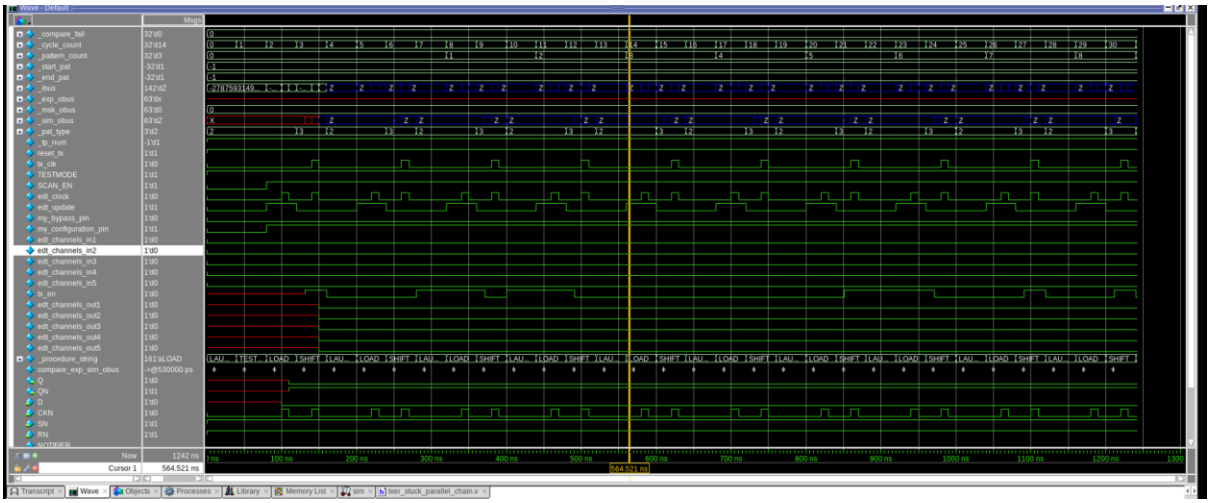


Figure 25: Parallel chain Simulated waveforms.

Figure 24 shows the internal diagram of the compactor, with the internal. By using this we have generated the ATPG pattern for serial scan serial chain, and parallel capture and chain. As part of the design for testability flow, we are going for the simulations of those patterns.

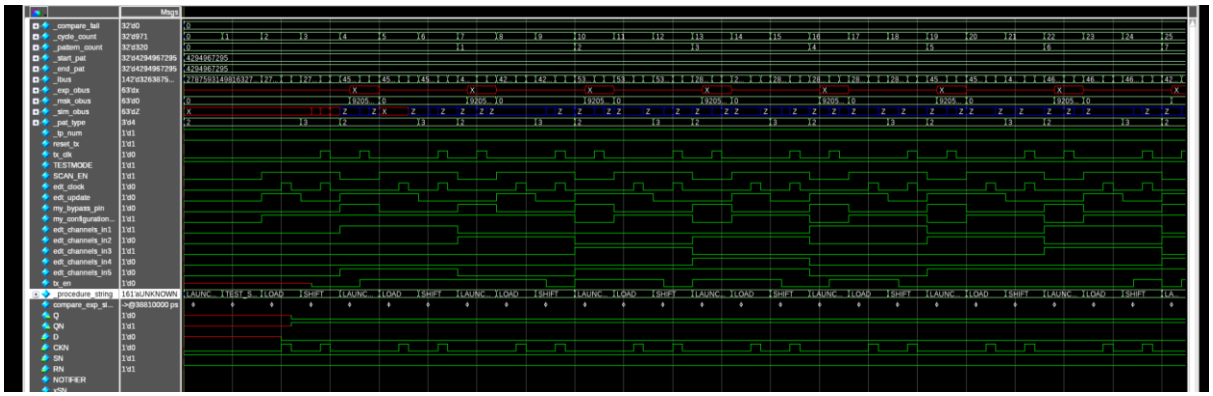


Figure 26: Parallel scan simulated waveforms.

Figure 26 shows the parallel capture simulated waveforms and Figure 27 shows the serial simulated waveforms.

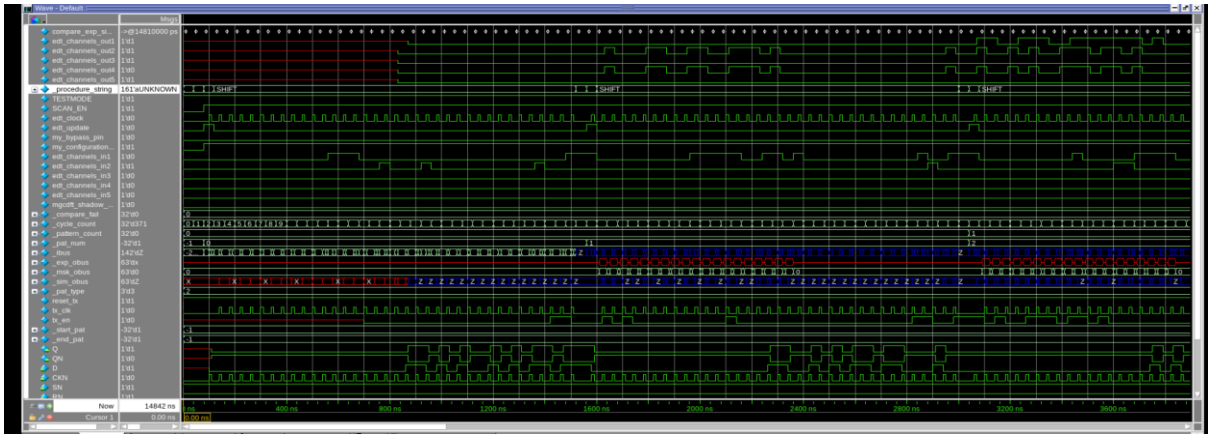


Figure 27: Serial chain Simulated waveforms

5. ADVANTAGES AND DISADVANTAGES

5.1 Advantages:

- ✓ Improved fault coverage: DUAL EDT provides better fault coverage compared to a single scan chain because it enables the testing of more paths and states within the circuit. This helps identify and detect more potential defects and manufacturing errors.
- ✓ Reduced test time: DUAL EDT can lead to shorter test times because it allows for concurrent testing of the two embedded scan chains. This parallelism can significantly reduce the overall testing duration, which is crucial for high-volume semiconductor manufacturing.
- ✓ Enhanced diagnosis and debugging: DUAL EDT makes it easier to pinpoint the location of faults or defects within the integrated circuit. This improved diagnosis capability is valuable for identifying and addressing issues during the manufacturing and debugging phases.

- ✓ Efficient testing of large designs: DUAL EDT is particularly useful for testing complex and large-scale integrated circuits where traditional testing methods may be inadequate. The parallel scan chains help manage the complexity and enable comprehensive testing.

5.2 Disadvantages:-

- ✓ Higher design and verification effort: Designing and verifying DUAL EDT configurations can be more challenging and time-consuming compared to traditional DFT techniques. The added complexity may require specialized expertise and tools.
- ✓ Compatibility issues: DUAL EDT may not be compatible with all designs or semiconductor processes. Some circuits may not be suitable for this configuration due to layout constraints, technology limitations, or other factors.
- ✓ Cost implications: The extra hardware and design effort associated with DUAL EDT can increase the overall cost of semiconductor production. This may impact the economic viability of using this approach for certain applications.

5.3 Discussion of Dual Configuration:

The dual configuration architecture in Design for Testability (DFT) refers to a methodology that incorporates two distinct configurations within a semiconductor design to enhance testability and fault coverage. This approach aims to address the limitations of traditional single-configured designs by providing greater flexibility and efficiency in testing.

One aspect of the dual configuration architecture involves the inclusion of redundant circuitry or specialized test structures alongside the primary functional circuitry. These redundant elements are strategically placed to facilitate testability features such as Built-In Self-Test (BIST), scan chains, or boundary scan cells. By integrating redundant structures into the design, manufacturers can improve fault coverage and diagnostic capabilities during testing, leading to higher product quality and reliability.

Additionally, the dual configuration architecture may encompass different modes or states of operation within the device. For example, a semiconductor chip may have separate configurations for normal operation and test mode. In test mode, certain features or functions may be activated or deactivated to facilitate testability, debugging, or diagnosis processes without interfering with normal device operation. This dual-mode approach allows for more comprehensive testing while minimizing disruption to end-user functionality.

The discussion surrounding the dual configuration architecture in DFT often focuses on its advantages and trade-offs. On the one hand, this approach offers enhanced testability, fault coverage, and diagnosis capabilities, which can lead to improved product quality and reliability. On the other hand, implementing dual configurations may incur additional design complexity, area overhead, and power consumption.

6. CONCLUSION

Design for Testability (DFT) is a critical aspect of the semiconductor industry, as it plays a pivotal role in identifying and mitigating faults in integrated chip manufacturing. Given the complexity of modern semiconductor designs, the integration of millions of transistors into a single chip creates the potential for numerous faults. To address this challenge, various techniques have been developed to reduce test time and test data volume.

One such technique is the dual configuration for Embedded Deterministic Test (EDT), which accommodates both higher and lower configuration needs. This approach offers flexibility and adaptability to different industry requirements. It ensures that a single design can cater to a wide range of EDT configurations without the need to overhaul the entire design for each specific netlist. This versatility not only simplifies the work of DFT designers but also helps reduce costs for customers.

While dual EDT configurations provide significant benefits, it's important to note that there are limitations and considerations associated with these configurations. These include the need for specialized tools and compatibility with specific design and manufacturing requirements.

In summary, DFT techniques, such as dual EDT configurations, are instrumental in maintaining the integrity and quality of semiconductor chips in the face of growing complexity and potential faults, ultimately contributing to the reliability and success of the silicon industry.

7. REFERENCES

[1] F. -J. Tsai et al., "Prediction of Test Pattern Count and Test Data Volume for Scan Architectures under Different Input Channel Configurations," 2020 IEEE International Test Conference (ITC), Washington, DC, USA, 2020, pp. 1-10, doi: 10.1109/ITC44778.2020.9325219.

[2] F. -J. Tsai et al., "Estimation of Test Data Volume for Scan Architectures with Different Numbers of Input Channels," 2020 IEEE International Test Conference in Asia (ITC-Asia), Taipei, Taiwan, 2020, pp. 130-135, doi: 10.1109/ITC-Asia51099.2020.00034.

[3] C. -S. Ye et al., "Efficient Test Compression Configuration Selection," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 7, pp. 2323-2336, July 2022, doi: 10.1109/TCAD.2021.3099100.

[4] S. D. Talatule, P. Zode and P. Zode, "A secure architecture for the design for testability structures," 2015 19th International Symposium on VLSI Design and Test, Ahmedabad, India, 2015, pp. 1-6, doi: 10.1109/ISVDAT.2015.7208090.

[5] Z. Zhao and Z. Cai, "A Review of Intelligent Design for Test Based on Machine Learning," 2023 International Symposium of Electronic Design Automation (ISEDA), Nanjing, China, 2023, pp. 116-120, doi: 10.1109/ISEDA59274.2023.10218713.

[6] K. S. Das and A. Zala, "Optimizing cell-aware ATPG pattern volume to keep test cost competitive," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2020, pp. 1-6, doi: 10.1109/CONECCT50063.2020.9198358.

[7] F. Wang and S. K. Gupta, "An Effective and Efficient Automatic Test Pattern Generation (ATPG) Paradigm for Certifying Performance of RSFQ Circuits," in IEEE Transactions on Applied Superconductivity, vol. 30, no. 5, pp. 1-11, Aug. 2020, Art no. 1300711, doi: 10.1109/TASC.2020.2965933.

[8] M. Li, Z. Shi, Z. Wang, W. Zhang, Y. Huang and Q. Xu, "Testability-Aware Low Power Controller Design with Evolutionary Learning," 2021 IEEE International Test Conference (ITC), Anaheim, CA, USA, 2021, pp. 324-328, doi: 10.1109/ITC50571.2021.00046.

[9] S. Pandey, N. S. Murty and R. Ranjan, "Test Power and Transition Fault Coverage Comparison Between LOC and LOS Test Scheme for Multiple Clock Domain Circuits," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 2017, pp. 1-4, doi: 10.1109/ICCIC.2017.8524188.

[10] Patel, Parul et al. “Comparative Analysis of Simulation Techniques: Scan Compression and Internal Scan.” *International Journal on Cybernetics & Informatics* (2023): n. pag.

[11] Gowda, Madhu. “Indonesian Journal of Electrical Engineering and Computer Science.” (2022).

[12] Pomeranz, Irith. “Path Unselection for Path Delay Fault Test Generation.” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31 (2023): 267-275

[13] Li, Mingye et al. “Methods for testing path delay and static faults in RSFQ circuits.” *2022 IEEE 40th VLSI Test Symposium (VTS)* (2022): 1-7

[14] Liang, Zhe-Jia et al. “High-Speed, Low-Storage Power and Thermal Predictions for ATPG Test Patterns.” *2023 IEEE International Test Conference (ITC)* (2023): 206-215.

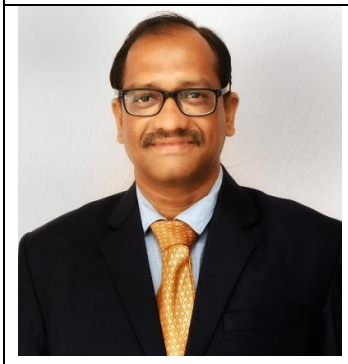
[15] Li, Mengwei et al. “Sequential-Fault Diagnosis Strategy for High-Speed Train Traction Systems Based on Unreliable Tests.” *Applied Sciences* (2023): n. pag.

* **Sri. Praveen K** working as DFT Engineer, Siliciom Technologies Pvt Ltd., **BE, Electronics, and Communication Engg M.Tech – Digital Electronics.** Now doing his **Ph.D** degree from Srinivas University, Mangaluru Professor having vast experience in teaching and industry, six years working as Assistant professor in Electronics and Communication Department at BIET, Davangere, three years working Assistant professor in Electronics and Communication Department at Bheema Institute of Technology and Science, Adoni, Andhra Pradesh, now Professor working in Industry as a Design for Testability Engineer in Siliciom Technologies Pvt Ltd, Bengaluru. Worked for several clients for Siemens, INTEL, Google, and Tech-Mahindra Cerium. His research interests include area VLSI Design, Digital System Design, DFT, RTL Design, and Front-end, and Back-end Design.

e-mail: praveen.kuntanahal@gmail.com

Dr. Rajanna presently working as a Research Professor at the Department of Electronics and Communication Engineering, Institute of Engineering and Technology, Affiliated to Srinivas University, Mangaluru, Karnataka, India. He has been working in various positions in teaching since 1984. Started at SJMIT Chitradurga, Karnataka. Professor received his Bachelor's Degree from M S Ramaiah Institute of Technology, Bangalore University, Karnataka. Walchand College of Engineering, Sangli. Shivaji University, Maharashtra. Professor completed his PhD degree from JNTU Hyderabad. His teaching experience is more than Thirty-five years with his research interests include an area of Power Electronics, Control Systems, Artificial Intelligence, Digital System Design, and Soft Computing.

e-mail: kgsrajanna@gmail.com



Dr. Shivakumaraswamy G M is working as an Assistant Professor at the Department of Electrical and Electronics Engineering, Bapuji Institute of Engineering and Technology, Affiliated with Visvesvaraya Technological University, Davangere, Karnataka, India. He has been working at Bapuji Institute of Engineering and Technology since 2004. Professor Shivakumar received his Bachelor's Degree from Kuvempu University, Shivamogga, in 2000 and Master's Degree from Visvesvaraya Technological University, Belagavi, in 2005. Recently Professor completed his PhD degree from Srinivas University, Mangaluru in 2023. His teaching experience is more than Nineteen years with his research interests include an area Privacy Preserving Data, Artificial Intelligence, Power Systems, Renewable Energy, Digital System Design.

e-mail: gms20mar@gmail.com