BIST CONTROLLER DESIGN

Authors : Ganderla Chaithanya
Boddu Ajay

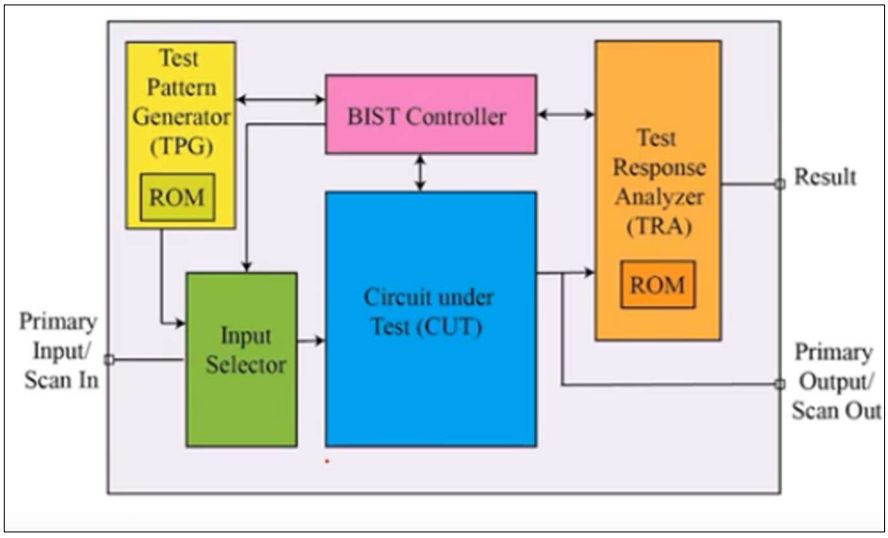
Description:
A **BIST (Built-In Self-Test) controller** is a dedicated on-chip hardware block designed to autonomously test the functionality of an integrated circuit or a subsystem without relying on expensive external test equipment or complex manual testing procedures. This approach significantly reduces testing time, minimizes production costs, and enables periodic self-checks for mission-critical systems where reliability is essential, such as automotive electronics, aerospace systems, and medical devices.

Purpose:

The main role of a BIST controller is to:

- Generate test patterns for the Circuit under test (CUT)
- Apply those patterns to the CUT
- Capture and compress the responses
- Compare the final result with a known good "Golden signature"
- Report whether the CUT is fault-free or faulty

Block diagram:



1. Test Pattern Generator (TPG)

- **Generates test inputs**: Produces a sequence of pseudo-random test vectors to stimulate the Circuit Under Test (CUT).
- **LFSR-based**: Uses a Linear Feedback Shift Register to create deterministic but pseudo-random bit patterns based on a chosen feedback polynomial. ($x^4 + x + 1$)

2. Input Selector

- **Switching between inputs**: Chooses whether the CUT receives normal functional inputs (Primary Input/Scan In) or BIST-generated patterns from the TPG.
- **Test mode control**: Controlled by the BIST Controller to route signals depending on whether the system is in normal mode or test mode.
- **Isolation**: Ensures functional input signals are isolated during self-test to prevent interference

3. Circuit Under Test (CUT)

- **Target of testing**: This is the hardware block whose functionality is being verified.
- **Receives test patterns**: Gets inputs from the Input Selector during test mode.
- **Produces responses**: Outputs from CUT are sent to the Test Response Analyzer for evaluation.

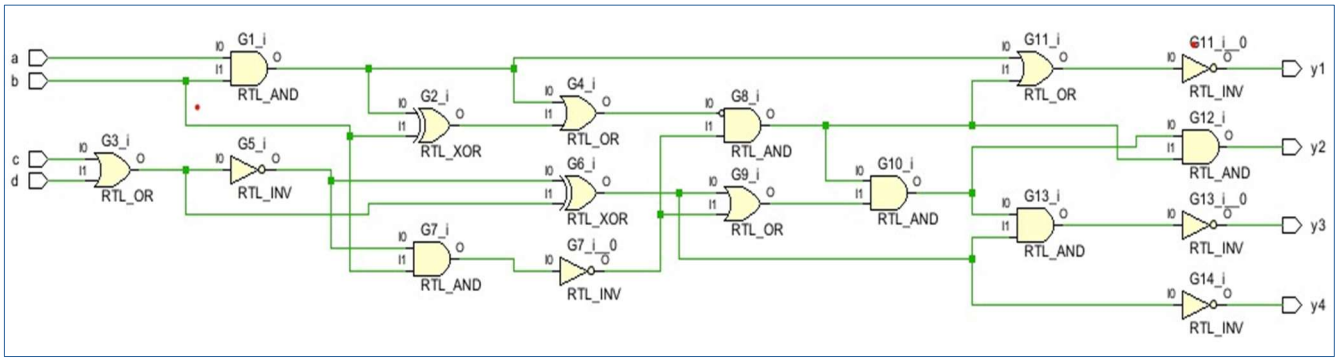
4. BIST Controller

- **Central control unit**: Orchestrates the entire self-test operation, including pattern generation, response collection, and signature comparison.
- **Sequencing**: Controls when to start/stop the LFSR and MISR, and when to switch the Input Selector.
- **Result decision**: Triggers the comparison between the generated signature and stored golden signature, then sets the final pass/fail flag.

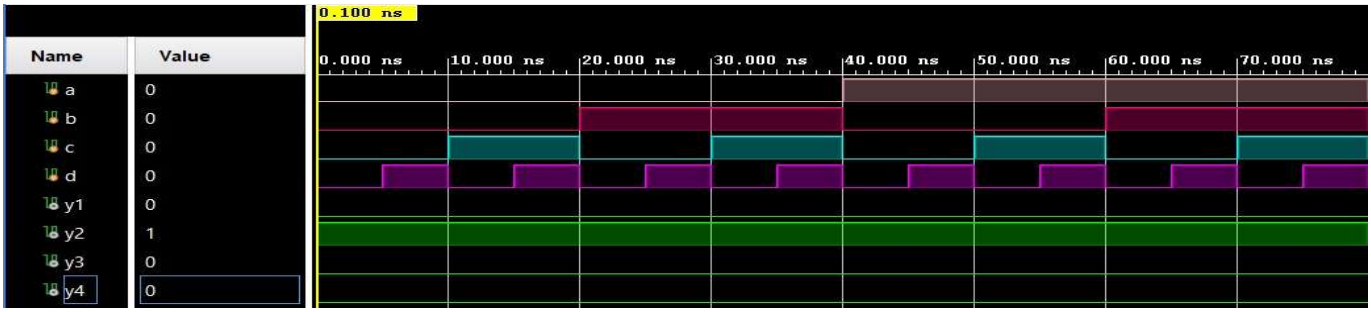
5. Output Response Analyzer (TRA)

- **Response compression**: Uses a Multiple Input Signature Register (MISR) to compress CUT's multiple output bits into a compact signature.
- **Fault detection**: Final MISR value is compared with the pre-stored golden signature to determine if the CUT is fault-free.

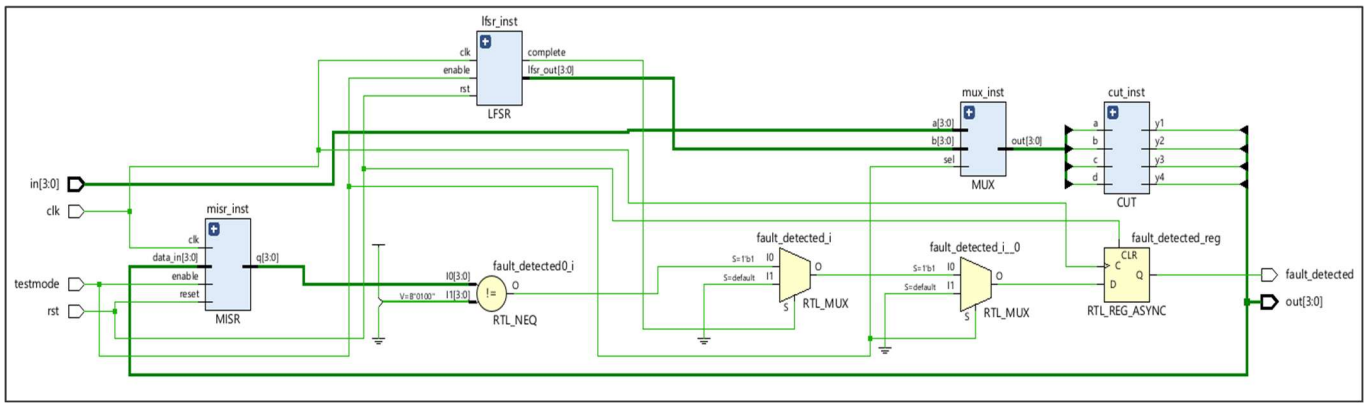
Circuit under Test:

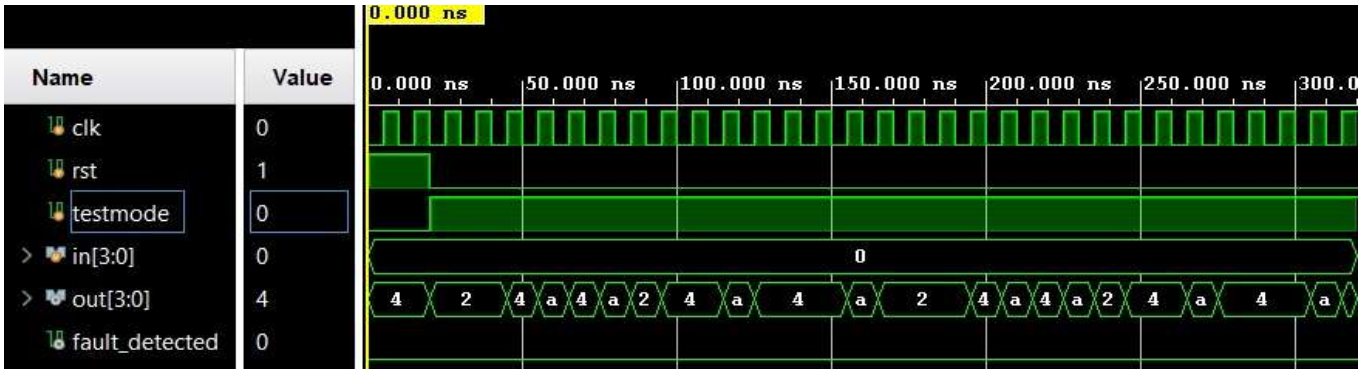


Testbench Results for CUT:

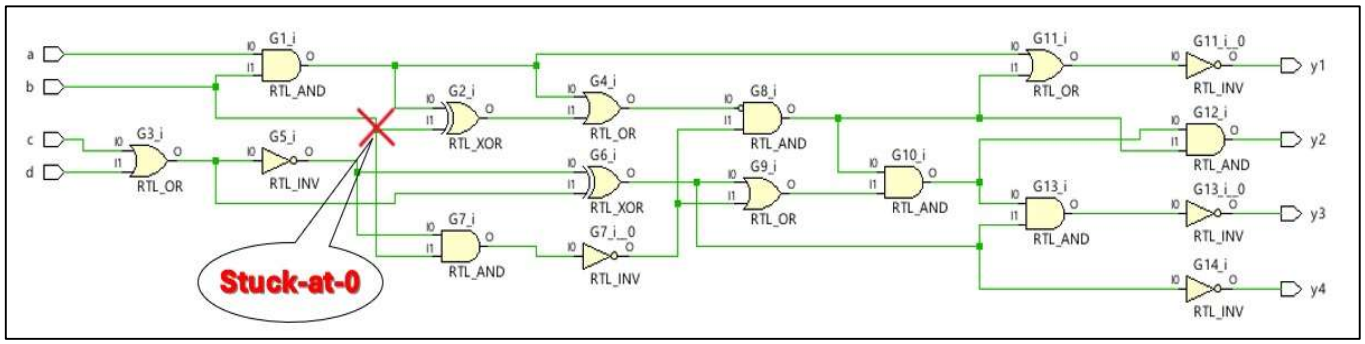


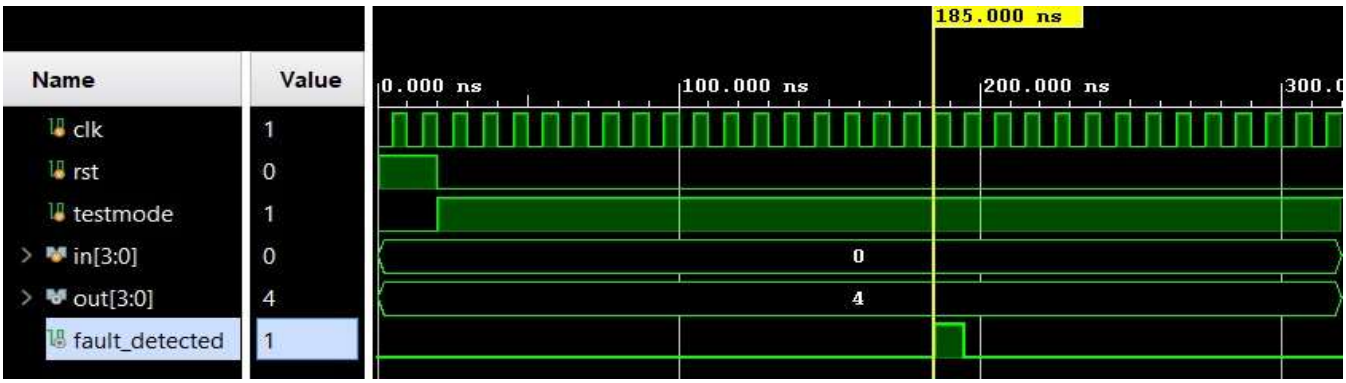
Testbench Results for BIST Controller with Fault-free CUT:



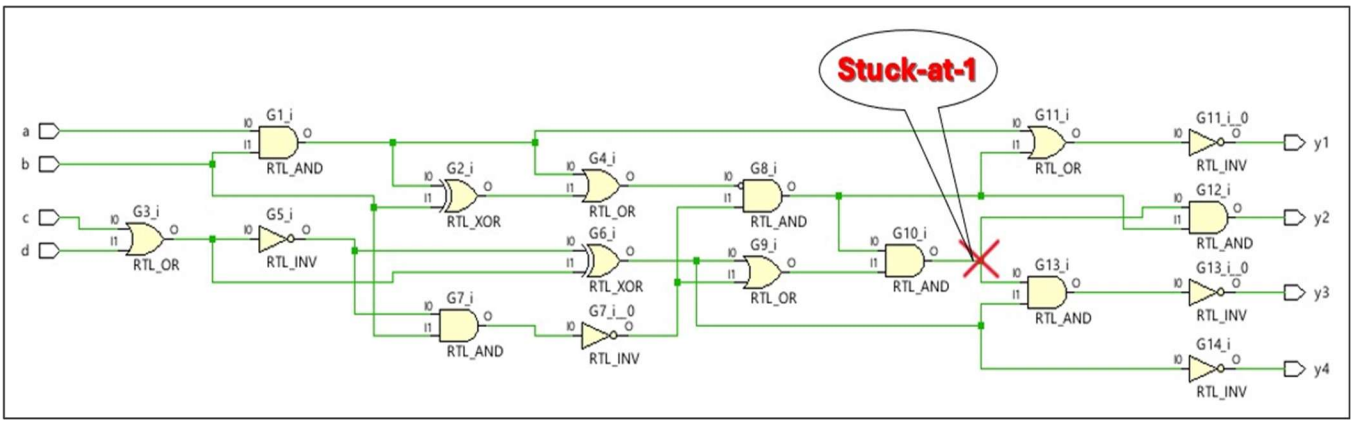


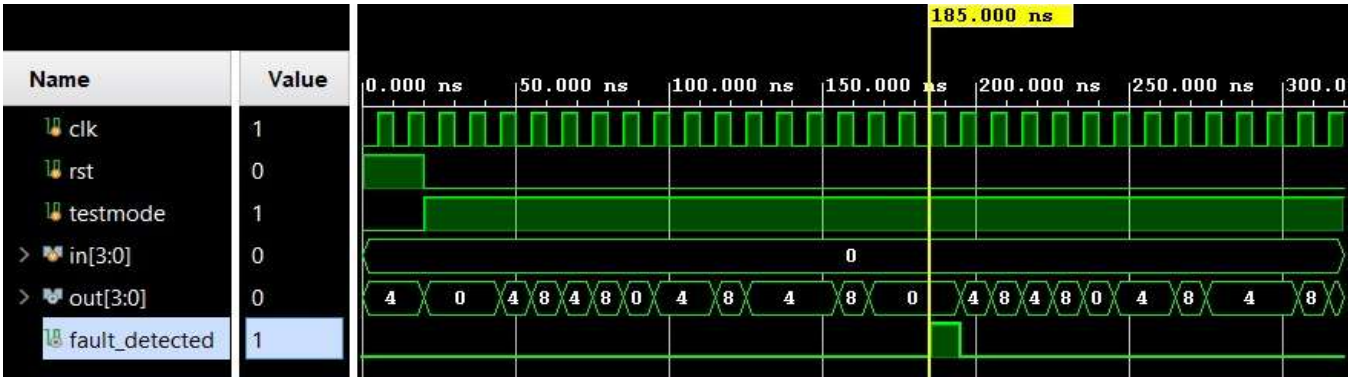
Testbench Results for BIST Controller with Faulty CUT(Stuck-at-0):





Testbench Results for BIST Controller with Faulty CUT(Stuck-at-1):





Specifications :

Metric	Value
No. of Inputs	6
No. of Outputs	5
No. of Clocks used	1
No. of Cells	72
Area	331.57 um^2
Delay	1090.33 ps
Power	119 uW

Conclusion :

The design and implementation of a BIST controller play a crucial role in modern DFT (Design for Testability) methodologies by enabling chips to test themselves efficiently and reliably. By integrating an LFSR-based test pattern generator, a MISR-based response analyzer, and a controller to orchestrate the testing process, BIST provides an effective mechanism for detecting manufacturing defects and operational faults without extensive external hardware support.