Memory Built-In Self Test Integration

BAE SYSTEMS

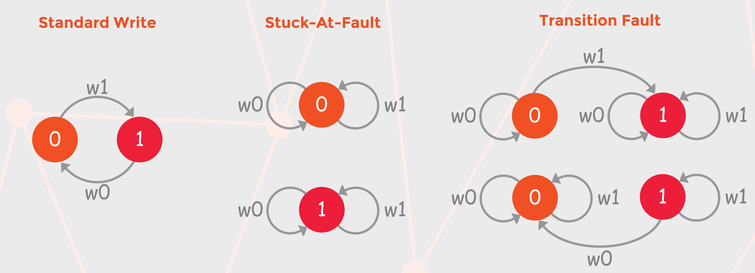
Osaze Y. Shears - ASIC Hardware Design, BAE Systems Inc., Manassas, Virginia
Computer Engineering, George Mason University

GEORGE MASON UNIVERSITY | Volgenau School of Engineering

I. Introduction

The decreasing size of transistors has allowed static random access memory (SRAM) hardware to store an increasing amount of data while retaining its compact size. Although we can store more data than ever before in such a small space, the amount of defects that the hardware is suspect to increases because of manufacturing errors and material inconsistencies. When chips are exposed to these defects, the overall yield for the design is affected.

Memory Defect Finite State Machines



II. Project Overview

To retain high yield percentages with increasing memory densities, memory built-in self-test (MBIST) hardware has been created for memory test and repair. MBIST hardware can become difficult to integrate into large designs because of the amount of connections required. The following issues have been addressed in this project in an effort to streamline the MBIST integration process.

Verification Issues: Due to the number of port connections between MBIST and each connected memory, ASIC designers may inadvertently wire unrelated ports together. This can cause a design to completely fail future verification.

Automation Issues: When MBIST is not instantiated at the same hierarchical level as memory, ASIC designers must implement several new ports for each hardware component enclosing the memories being tested.

III. Experimental Procedure: Verification

1. Research all MBIST-to-Memory combinations.
2. Create guidance on how to connect MBIST to memories.
3. Verify all connections were made correctly via test bench.

IV. Project Deliverables: Verification

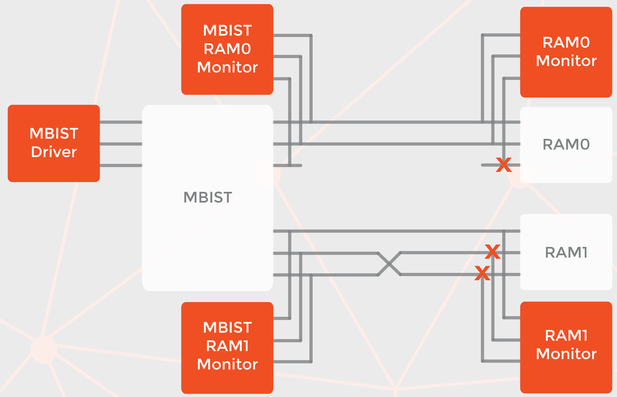
MBIST Connectivity Test Bench

- Uses drivers and monitors to evaluate MBIST connections
- Logic-1 (high) bit is driven through all MBIST-memory ports
- Comparisons made between bits at MBIST and at memory

MBIST Verification Plan

- Covers the process for extracting and simulating test bench and MBIST files from the MBIST generation software

Sample MBIST Connectivity Test Bench Failures



V. Experimental Procedure: Automation

1. Research methods for reducing ports in VHDL & Verilog.
2. Determine which methods are supported by logic synthesis.
3. Use the verification plan from Part 1 to validate connections.

VI. Project Deliverables: Automation

One Dimensional Vectors Method*

- High physical resource utilization and low code overhead
- MBIST data can be easily parsed from the vectors
- Construct supported in cross-language RTL synthesis

MBIST Template File Modifications

- Generation software was reconfigured to implement the one dimensional vectors method

Logic Signal Concatenation Methods

	Code Overhead	Parsing Difficulty	Physical Utilization	Cross-Language
Unmodified	High	Low	High	Yes
1D Vector w/ All Ports	Low	High	High	Yes
1D Vectors w/ Similar Ports*	Med	Med	Med	Yes
2D Vector w/ All Ports	Low	Low	Low	No
2D Vectors w/ Similar Ports	Med	Low	Med	No
Interfaces (VL)/ Records (VHDL)	Low	Low	High	No

VII. Future Work

Further improvements to the outcomes of this project may include:

- Improved physical resource utilization
- Software for inserting MBIST into a design
- Test Bench for simulating several MBIST engines simultaneously

VIII. Acknowledgments

Virginia Microelectronics Consortium (VMEC) for the opportunity to conduct this research and gain an invaluable experience as an ASIC designer.

Patty Schaefer and Danny Pirkl, who challenged and guided me as I conducted this work, and introduced me to several leaders within the company.

ASIC Hardware Design Team for providing further insights regarding MBIST.

Kyle Fowler and John Davis for hosting amazing intern events

BAE Systems employees that made my internship a great experience.