Design For Testability

2011.06.03

Sungho Kang

COMPUTER SYSTEMS & RELIABLE SOC LAB.
Agenda

• Introduction
• Test Methodology
• Design for Testability
• IEEE P1500
• Case Study
• Conclusion
VLSI Implementation

Customer’s need

Determine requirements

Write specifications

Design synthesis and Verification

Test development

Fabrication

Manufacturing test

Chips to customer
Design Constraints

- Area
- Speed
- Power
- Testability
Testing Process

- Rule of thumb: spend 5-10% die area on DFT (10% of die is state and scan is 50% overhead per state!)
SOC Design

- User Defined Core
- ROM
- DRAM
- RF/Analog Core
- IP
- IP
- UDL
- User Defined Core
- IO Pad
- IO Pad

COMPUTER SYSTEMS & RELIABLE SOC LAB.
Manufacturing Defects

In fabrication, defects get introduced from many sources:

- Contamination
- Metalization Defect
- Implant Defect
- Wafer Defect
- Oxide Defect
- Interconnect Defect

![Diagram of manufacturing defects](image)
Defect & Fault

- Permanent
- Intermittent
- Temporary
Clustered VLSI Defects

Unclustered defects
Wafer yield = 12/22 = 0.55

Clustered defects (VLSI)
Wafer yield = 17/22 = 0.77
Verification vs Test

◆ Verification
  ➢ Correctness of design
  ➢ Simulation, Emulation and Formal verification
  ➢ Performed once before manufacturing
  ➢ Responsible for quality of design

◆ Test
  ➢ Correctness of manufactured hardware
  ➢ Test generation and test application
  ➢ Performed every manufactured devices
  ➢ Responsible for quality of devices
Test Process

Parts for Manufacturing Line → Test Process

Bad parts which fail Test (BF)

Good parts which pass Test (GP)

Bad parts which pass Test (BP)

Defect Level = \( \frac{BP}{GP + BP} \)
Test Cost vs Manufacturing Cost

Capital Investment for Manufacturing Test

1997 Microprocessor Cost of Test trend Model [2000 ITRS]


Si capital/transistor

Test capital/transistor

COMPUTER SYSTEMS & RELIABLE SOC LAB.
**Single Stuck-at Fault (SSF)**

- Only one line in the circuit is faulty at a time
- The fault is permanent (as opposed to transient)
- The effect of the fault is as if the faulty node is tied to either Vcc (s-a-1), or Gnd (s-a-0)
- The function of the gates in the circuit is unaffected by the fault

---

**Fault: A s-a-1**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fault-Free Gate

![Fault-Free Gate Diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Faulty Gate

![Faulty Gate Diagram]
Automatic Test Pattern Generation (ATPG)

- Calculate the set of test patterns from a description of the logic network and a set of assumptions called fault models.
To detect a stuck-at fault in synchronous **sequential** logic, we can still use the familiar D algorithm, but it’ll take:

- One or more clock cycles to **activate** the fault.
- One or more clock cycles to **propagate** the fault effect.

In general, we’ll need a **sequence** of patterns to detect a fault!
Design-for-Testability (DFT)

◆ Definition
  - Any design effort to reduce test costs
  - The process of including special features to make a device easily testable

◆ Objective
  - To reduce in overall design cycle times and test costs without sacrificing the quality of the product
DFT Flow

1. Verilog/VHDL Netlist
2. DFT Rule Checking
3. Test Synthesis
4. DFT Rule Checking
5. ATPG
6. DFT Rule Checking
7. Full Timing Simulation

- Scan Design, Boundary Scan Design
- Memory BIST Design, Logic BIST Design
- Static Timing Analysis
- Test Pattern Generation
- Logic Simulation
- Test Vector Translation
Ad-Hoc DFT Methods

- Good design practices learnt through experience are used as guidelines:
  - Avoid asynchronous (unclocked) feedback.
  - Make flip-flops initializable.
  - Avoid redundant gates. Avoid large fanin gates.
  - Provide test control for difficult-to-control signals.
  - Avoid gated clocks.
  - Consider ATE requirements (tristates, etc.)

- Design reviews conducted by experts or design auditing tools.

- Disadvantages of ad-hoc DFT methods:
  - Experts and tools not always available.
  - Test generation is often manual with no guarantee of high fault coverage.
  - Design iterations may be necessary.
Test Points

- Employ test points to enhance controllability and observability
- Large demand on extra I/O pins
- Example
Initialization

- Flip-flop with explicit clear
  - Use explicit clear to all FFs
Partitioning

- Split large counters
Scan Test

- Convert each flip-flop to a scan register
  - Only costs one extra multiplexer
  - Normal mode: flip-flops behave as usual
  - Scan mode: flip-flops behave as shift register
- Contents of FFs can be scanned out and new values scanned in

A MUX is added

Diagram showing flip-flops and logic clouds with scan-in and scan-out connections.
Optimum Scan Layout

Active areas: XY and X’Y’

IO pad
Flip-flop cell
Routing channels
Interconnects

SFF cell

SCANIN

SCAN OUT

Computational Systems & Reliable SOC Lab.
Scan Chain Reordering

- To reduce the routing congestions
- To reduce the hold-buffer insertion during placement
  - May need skew-based optimization

Before

After
Logic BIST

Control unit sequences all activity

Pseudo-Random Pattern Generator

All flops placed into large number of relatively short scan chains

Multiple Input Signature Register compresses responses into a signature
Multiple Seed / Reseeding

- Seed ROM
  - Seed0
  - Seed1
  - Seed2

- BIST Controller

- LFSR

- CUT

- MISR

Test clock
Bit Fixing/Bit Flipping Method

Bit Flipping Circuit

Scan chain

Signature Analyzer

Sequential CUT

\[ t_0, t_1, \ldots, t_{L-1}, s_0, s_1, \ldots, s_{n-1} \]
Embedded Memory in SOC

![Bar Chart Showing Memory Area Over Time]

- **Year** 1999: **20%** Memory, **52%** Reused Logic, **28%** New Logic
- **Year** 2002: **52%** Memory, **28%** Reused Logic, **20%** New Logic
- **Year** 2005: **71%** Memory, **18%** Reused Logic, **11%** New Logic
- **Year** 2008: **83%** Memory, **17%** Reused Logic, **0%** New Logic
- **Year** 2011: **90%** Memory, **10%** Reused Logic, **0%** New Logic
- **Year** 2014: **94%** Memory, **6%** Reused Logic, **0%** New Logic

**Legend:**
- Red: % Area Memory
- Gray: % Area Reused Logic
- Blue: % Area New Logic
Memory Functional Model

Address Latch → Column Decoder

Row Decoder

Address

Column Decoder

Memory Cell Array

Write Driver

Sense Amplifiers

Data Register

Refresh Logic

Refresh

Data Flow

Control Flow

Address Decoder

Memory Cell Array

Read/Write Logic

Data Out

Data In

Read/Write and Chip Enable

Address

Data
Memory BIST Concept

Patterns: Writing/Reading 0s, 1s, Walking 0s, 1s, Galloping 0s, 1s

Controller

Pattern Generator

Comparator

Go/No-Go

BIST Module

Test Collar (MUX)

RAM

RAM Controller

Controller

Pattern Generator

Comparator

Go/No-Go

BIST Module

Test Collar (MUX)

RAM

RAM Controller
PMBIST

Architecture
Built-In Redundancy Analysis (BIRA)

- Analyze memory cell failures and utilize redundant cells to repair them efficiently.
- Inefficient memory cell repair in terms of repair costs and time.
- Identify an efficient method to repair memory cells.

Performance evaluation metrics:
- Repair rate
- Area overhead
- Rate of overused spares
- Analysis speed

Diagram illustrating the BIRA process.
Test Compression

- Compress the test input sequences
- Need a decompression units to make original test sequences
- Can be reduced for both limitations of ATE
  - The size of ATE memory
  - The width of ATE channel
- Can be reduced test application time
Test Compression Concepts

Smaller Memory (storing encoded test data)

Lower bandwidth
IEEE 1149.1 Device Architecture

- I/O Pad
- Boundary-scan cell
- Boundary-scan path
- BS Test bus circuitry
- IDCODE Register
- Instruction register
- Bypass register
- Glue Logic
- TDI
- TMS
- TRST
- TCK
- TDO
- Out
- S
The Principle of BS Architecture
SOC Design Evolution

- Emergence of very large transistor counts on a single chip
- Mixed technologies on the same chip
- Creation of Intellectual Property (IP)
- Reusable IP-based design
SoC with P1500 Wrapped Cores

- **TAM Source/Sink**
  - From chip I/O, test bus/rail/port, BIST, etc..

- **TAM In/Out**
  - 0 to n lines for parallel and/or serial test data, or test control

- **P1500 Wrapper Interface Port (WIP)**
  - From chip-level TAP controller, chip I/O, …
Test Wrapper

Wrapper Elements

Functional Core
Input

from/to
External Tester

scan chain

scan chain

Core

scan chain

scan chain

Functional Core
Output

Wrapper Test Controller

to/from TAB

from/to External Tester

from/to External Tester

Wrapper Test Controller
Test access mechanism (TAM)

- Defined as user-defined test data communication structure
- Carries test signals from source to module, and module to sink
- Tests module interconnects via test-wrappers
- May contain bus, boundary-scan and analog test bus components.
Test Scheduling

- Test scheduling problem
  - Determine the minimal number of test sessions required to test all IPs
  - Determine the minimal colors that can be assigned to the nodes of a graph such that no edge connects two nodes of the same color
- More complex when test time and power are considered

*Packed Bin = TAM Design + Test Schedule*
Conclusion

- Testable core design
- Logic BIST
- Test reuse
- Hierarchical testing
- Core access architecture
- Parallel access & bypass
- Core isolation
- IP-system test interface

Test Automation
- Automatic test pattern
- Fault simulation
- Testability measure
- Scan insertion & synthesis
- BIST circuit synthesis
- Boundary scan insertion & synthesis

Low cost external ATE

- Test spec.
- Test hardware control
- Test scheduling

- Analog Fault modeling
- Mixed signal Built-In Self Test
- Built-In Self Calibration

- Testable design
- Memory test algorithm
- Memory BIST, BISR

COMPUTER SYSTEMS & RELIABLE SOC LAB.