Outline

- What is Testing?
- Faults, ATPG & Fault Simulation
- Ad-Hoc
- Scan
- Logic BIST & Memory BIST
- P1149.1 & P1500
- 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

- **Defect**
  - Physical deviation from some specified properties

- **Test**
  - An experiment whose purpose is to detect the presence of defects and to diagnose the source of defects

- **Byproduct of testing**
  - Reliability measurement of the product and process
  - Quality assurance
  - Assistance to verification and validation
- Rule of thumb: spend 5-10% die area on DFT (10% of die is state and scan is 50% overhead per state!)
Verification vs Testing

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

- **Testing**
  - Correctness of manufactured hardware
  - Test generation and test application
  - Performed every manufactured devices
  - Responsible for quality of devices
## Testing Costs

- To detect problems early (Rule of Ten)

<table>
<thead>
<tr>
<th>Test</th>
<th>Failure</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip Test</td>
<td>Component Failure</td>
<td>$0.3</td>
</tr>
<tr>
<td>Board Test</td>
<td>Board Failure</td>
<td>$3</td>
</tr>
<tr>
<td>System Test</td>
<td>System Failure</td>
<td>$30</td>
</tr>
<tr>
<td>Field Test</td>
<td>Field Failure</td>
<td>$300</td>
</tr>
</tbody>
</table>
Test Cost vs Manufacturing Cost

Capital Investment for Manufacturing Test


Cost/Tr

0.1
0.01
0.001
0.0001
0.00001
0.000001
0.0000001

Si capital/transistor

Test capital/transistor

1997 Microprocessor Cost of Test trend Model [2000 ITRS]
A manufacturing defect is a finite chip area with electrically malfunctioning circuitry caused by errors in the fabrication process.

A chip with no manufacturing defect is called a good chip.

Fraction (or percentage) of good chips produced in a manufacturing process is called the yield.

Unclustered defects
Wafer yield = 12/22 = 0.55

Clustered defects (VLSI)
Wafer yield = 17/22 = 0.77
Bathtub Curve

Failure Rate

Infant Mortality

Working Life

Wearout

Early Failures

Random failure

Wearout Failures

Time
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 = \( \frac{BP}{GP + BP} \)
Quality of Test

- **Quality of Test**
  - Y : Yield
  - DL: Defect Level
  - d : defect coverage
  - DL = 1 - Y \(^{1-d}\)

- **Consider a 0.5 yield process**
  - To achieve 0.01 defect level, 99% coverage is required
  - To achieve 80% coverage, 0.2 defect level
Design for Testability

- Refers to those design techniques that make test generation and test application cost-effective

- Why need?
  - Difficulty in preparing test patterns

- Advantages
  - Test generation is easy
  - High quality testing

- Disadvantages
  - Area overhead
  - Timing overhead
How Much Testing Is Enough?

Test Development Time as a Percentage of Total Design Time

Testability of a Circuit (%)
Test Costs vs Manufacturing Costs

- Total costs
- Processing costs
- Packaging costs
- Testing costs

Costs vs No special test provision vs Additional silicon area Overhead for test purposes
Current Situation

- Increase of complexity
  - Moore’s Law
    - No. of transistors doubles every 18 months
  - Increase of speed
    - 30% increase per year
    - Timing and signal integrity
- Increase of test cost
  - 30-40% of overall cost
- Increase of ATE performance
  - 12% increase per year
- Time to Market
  - Exhaustive testing is no good
    - 8080 takes $10^{20}$ years at one million tests per second
Test Cost Factors

- Low Cost Test Equipment
- Low cost external ATE
- Efficient Test Engineering
- Spec. Design Methodology
- Sound Test Methodology
- Appropriate DFT
- Powerful Test CAD Tools
- Test Equipment
- Test Engineering
- Sound Test Methodology
- Appropriate DFT
- Powerful Test CAD Tools
In fabrication, defects get introduced from many sources:

- Contamination
- Metalization Defect
- Implant Defect
- Wafer Defect
- Oxide Defect
- Interconnect Defect
Logical Fault Models

- **Gate Level Faults**
  - **Stuck-at**
    - Short between signal and ground or power
  - **Bridging**
    - Short between two signals

- **Transistor Level Faults**
  - **Short**
    - Connecting points not intended to be connected
  - **Open (break)**
    - Breaking a connection
  - **Stuck-on (stuck-short)**
  - **Stuck-open (stuck-off)**

- **Delay Faults**

- **Temporary Faults**
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

<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: 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>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
Relative Performance of Fault Models

- HP(1996) : 25K std. cell design, 0.8micron
  - Full scan : 1497FFs, 33MHz
  - Functional pattern : 83.7% fault coverage
  - Stuck-at fault coverage : 99% with 284 patterns
  - Transition fault coverage : 99.4% with 554 patterns
## Fault Coverage and Efficiency

\[
\text{Fault coverage} = \frac{\text{# of detected faults}}{\text{Total # faults}}
\]

\[
\text{Fault efficiency} = \frac{\text{# of detected faults}}{\text{Total # faults} - \text{# undetectable faults}}
\]
Fault Simulation

Test Patterns

Fault Simulator

Fault Coverage

Circuit

Fault List
ATPG

Circuit \rightarrow ATPG \rightarrow Tests

\begin{align*}
\text{Fault Model} & \quad \text{Fault List}
\end{align*}
ATPG Example

- **G output s-a-1**

  ![Fault Excitation](image1)

  ![Fault Propagation](image2)
Sequential ATPG

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!
DFT Flow

1. Verilog/VHDL Netlist
2. DFT Rule Checking
3. Test Synthesis
   - DFT Rule Checking
   - Scan Design, Boundary Scan Design
   - Memory BIST Design, Logic BIST Design
4. DFT Rule Checking
5. ATPG
   - Static Timing Analysis
   - Test Pattern Generation
6. Full Timing Simulation
   - Logic Simulation
   - Test Vector Translation
Classification of DFT

- Ad-Hoc Design
  - Initialization
  - Adding extra test points
  - Circuit partitioning
- Structured Design
  - Scan design
  - Boundary Scan
  - Built-in Self Test
Partitioning

- Physically divide the system into multiple chips or boards
- On board-level systems, use jumper wires to divide subunits
- Can have performance penalties
Test Point Insertion

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

PI → Combinational logic → FFs → PO
Adding Scan Structure

A MUX is added
Automated Scan Design

- Rule violations
- Behavior, RTL, and logic
  Design and verification
- Gate-level netlist
- Scan design rule audits
- Scan hardware insertion
- Combinational ATPG
- Scan sequence and test program generation
- Chip layout: Scan-chain optimization, timing verification
- Design and test data for manufacturing
- Mask data
  - Test program
  - Scan chain order
- Combinational vectors
  - Scan netlist
Sources of DFT Rule Violation

- Reset/preset violations
  - Controllability through PIs or disabled during test
- Clock rule violations
  - Controllability/gating
- Tristate bus violations
  - Ensures there is no contention
- Bidirectional I/O violations
  - Controls direction to avoid contentions
- Latch violations
  - Ensures transparency during test mode
- Shift constraint violations
  - Ensures proper shifting of data through the scan chain
Scan Chain Reordering

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

Before

After
Built In Self Test

- Capability of a product to carry out an explicit test of itself
  - Test patterns are generated on-chip
  - Responses to the test patterns are also evaluated on chip
  - External operations are required only to initialized the built-in tests and to check the test results (go/no-go)
Linear Feedback Shift Register

- The state of shift register depends only on the prior state
LFSR Example

- Characteristic Polynomial: $1+x^2+x^3$
  - Initial condition (1,0,0): $x$
  - $Q1: x / (1+x^2+x^3)$
  - $Q2: x^2 / (1+x^2+x^3)$
  - $Q3: x^3 / (1+x^2+x^3)$
LFSR Example

- When initial state is 100
  - Q1 Q2 Q3
  - 1 0 0
  - 0 1 0
  - 1 0 1
  - 1 1 0
  - 1 1 1
  - 0 1 1
  - 0 0 1
  - 1 0 0
  - 0 1 0
  - 1 0 1

- When initial state is 000
  - Q1 Q2 Q3
  - 0 0 0
  - 0 0 0
MISR

- Multiple input signature analysis

![MISR Diagram]
Test-per-Clock

- Test patterns are applied to CUT every clock cycle
- Additional logic and delay are required between the input FF and CUT
- BILBO like type: Cannot perform compression and pattern generation concurrently
- Entire test is scheduled and divided into sessions
- Complex test control unit is required
STUMPS

- Self Testing Using MISR and Parallel SRSG
  - Centralized and separate BIST
  - Multiple scan paths
    - Reduction in test time
- Lower overhead than BILBO but takes longer to apply
Complete Fault Coverage

- Hybrid test pattern generation
  - Combination of pseudo-random test and deterministic test
- Deterministic test pattern generation
  - Store-and-generate scheme
  - Test set embedding scheme
    - Bit-flipping or Bit-fixing
    - LFSR reseeding or multiple-polynomial reseeding
- Hybrid BIST trade-off (random deterministic)

![Diagram illustrating hybrid BIST trade-off with short, long, and optimal test times and associated hardware overheads.]

- Short test time: Heavy H/W overhead
- Long test time: Light H/W overhead
- Optimal test time: Optimal H/W overhead
Multiple Seed / Reseeding

- Seed0
- Seed1
- Seed2

Seed ROM

Test clock

BIST Controller

LFSR

CUT

MISR
Bit Fixing/Bit Flipping Method

Sequential CUT: $t_0$, $t_1$, ..., $t_{L-1}$

Scan chain: $s_0$, $s_1$, ..., $s_{n-1}$

Signature Analyzer

Bit Flipping Circuit
Why Need Test Compression?

- Today’s SoC environments
  - Large and complex VLSI circuits
  - Need an enormous amount of test data
- Limitations of ATE based test methods
  - Channel width and memory size
  - Modified or more expensive ATE must be required
- Reducing test data by eliminating useful test patterns
  - Can be reduced for the size of the ATE memory
  - Decreases the accuracy of testing
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
### Embedded Memory Testing

Source: ITRS 2001 – Percentage of Logic Forecast in SoC Design

<table>
<thead>
<tr>
<th>Year</th>
<th>Node (nm)</th>
<th>% Area New Logic</th>
<th>% Area Reused Logic</th>
<th>% Area Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>180</td>
<td>64</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>2002</td>
<td>130</td>
<td>32</td>
<td>16</td>
<td>52</td>
</tr>
<tr>
<td>2005</td>
<td>100</td>
<td>16</td>
<td>13</td>
<td>71</td>
</tr>
<tr>
<td>2008</td>
<td>70</td>
<td>8</td>
<td>9</td>
<td>90</td>
</tr>
<tr>
<td>2014</td>
<td>35</td>
<td>2</td>
<td>4</td>
<td>94</td>
</tr>
</tbody>
</table>
Memory Functional Model
MBIST Basic Architecture

- **Fault Model**
  - Stuck-at Fault
  - Address Decoding Fault
  - Coupling Fault
  - Pattern-Sensitive Fault
- **Memory Test Algorithms**
  - March Test
  - Checkerboard
  - Zero-One

![Diagram of MBIST Basic Architecture](image)
Concept of Boundary Scan

- Improve testability by reducing the requirements placed on the physical test equipment

- Also called
  - JTAG (Joint Test Action Group) Boundary Scan Standards
  - IEEE P1149.1

- Why use it?
  - Testing interconnections among chips
  - Testing each chip
  - Snapshot observation of normal system data

- Why testing boards?
  - To test board is easier than to test systems
IEEE 1149.1 Device Architecture

- **I/O Pad**
- **BS Test bus circuitry**
- **Boundary-scan cell**
- **Boundary-scan path**
- **IDCODE Register**
- **Instruction register**
- **Bypass register**
- **Glue Logic**
- **S_{out}**

- **TAP**
- **TDI**
- **TMS**
- **TRST**
- **TCK**
- **TDO**
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
**System Chip 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 Access

- No Direct Physical Access Method
  - Test access mechanism is required
- Today’s chip is tomorrow’s core
Core Level Testing Summary

- Testable core design
  - Logic BIST
  - Test reuse
  - Hierarchical testing
- Core access architecture
  - Parallel access & bypass
  - Core isolation
- User Defined Core
- IP
- DRAM
- UDL
- RF/Analog Core
- Test Access
- SoC Test Controller
- Mem BIST
- 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
- • Testable design
- • Analog Fault modeling
  - Mixed signal
  - Built-In Self Test
  - Built-In Self Calibration
- • Memory test algorithm
  - Memory BIST, BISR
- • Testable core design
  - Logic BIST
  - Test reuse
  - Hierarchical testing
Test Challenges

- Test quality
- Test cost reduction
- At-speed test
- Reduction of design efforts
- Test design reuse