Test - References

- Rabaey, “Digital Integrated Circuits”
- Chapter 25, Testing of High-Performance Processors by D.K. Bhavsar
Testing is Expensive

- VLSI testers cost $1-5M
- Volume manufacturing requires large number of testers, maintenance
- Tester time costs are in ¢/sec
- Test cost contributes 20-30% to total chip cost

Typical Tester
Types of Testing

<table>
<thead>
<tr>
<th>Step</th>
<th>Error Source</th>
<th>Test Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Design flaws</td>
<td>Design ver.</td>
</tr>
<tr>
<td>Prototype</td>
<td>Design flaws</td>
<td>Functional test</td>
</tr>
<tr>
<td></td>
<td>Prototype flaws</td>
<td></td>
</tr>
<tr>
<td>Manufacture</td>
<td>Physical defects</td>
<td>Manuf. test</td>
</tr>
<tr>
<td>Shipping</td>
<td>Man. test, transport</td>
<td></td>
</tr>
<tr>
<td>System Integration</td>
<td>Same</td>
<td>Functional test</td>
</tr>
<tr>
<td>Service</td>
<td>Stress, Age</td>
<td>Diagnosis</td>
</tr>
</tbody>
</table>

Test Classification

- **Diagnostic test**
  - used in chip/board debugging
  - defect localization
- **“go/no go” or production test**
  - Used in chip production
- **Parametric test**
  - $x \in [v,i]$ versus $x \in [0,1]$
  - check parameters such as NM, $V_t$, $t_p$, T
Chip Debugging

- Design errors or fabrication defects?
- Micro-probing the die (1-0.1pF)
- E-beam
- Single-die repair (FIB)

Design for Testability

![Diagram of combinational and sequential logic modules]

- Combinational function: $2^N$ patterns
- Sequential engine: $2^{N+M}$ patterns

Exhaustive test is impossible or unpractical
Problem: Controllability/Observability

- Combinational Circuits:
  controllable and observable - relatively easy to determine test patterns
- Sequential Circuits: State!
  Turn into combinational circuits or use self-test
- Memory: requires complex patterns
  Use self-test

Generating and Validating Test-Vectors

- Automatic test-pattern generation (ATPG)
  - for given fault, determine excitation vector (called test vector) that will propagate error to primary (observable) output
  - majority of available tools: combinational networks only
  - sequential ATPG available from academic research
- Fault simulation
  - determines test coverage of proposed test-vector set
  - simulates correct network in parallel with faulty networks
- Both require adequate models of faults in CMOS integrated circuits
Fault Models

Most Popular - “Stuck - at” model

![Diagram of stuck-at model](image)

Covers almost all (other) occurring faults, such as opens and shorts (bridges).

A : x1 sa1
β : x1 sa0 or x2 sa0
γ : Z sa1

Problem with stuck-at model: CMOS open fault

![Diagram of CMOS open fault](image)

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z_{\text{not}}</td>
</tr>
</tbody>
</table>

Sequential effect

Needs two vectors to ensure detection!

Other options: use stuck-open or stuck-short models

This requires fault-simulation and analysis at the switch or transistor level - Very expensive!
Problem with stuck-at model: CMOS short fault

Causes short circuit between Vdd and GND for $A=C=0$, $B=1$

Possible approach:
Supply Current Measurement (IDDQ)
but: not applicable for gigascale integration

Generating and Validating Test-Vectors

- **Automatic test-pattern generation (ATPG)**
  - for given fault, determine excitation vector (called test vector) that will propagate error to primary (observable) output
  - majority of available tools: combinational networks only
  - sequential ATPG available from academic research

- **Fault simulation**
  - determines test coverage of proposed test-vector set
  - simulates correct network in parallel with faulty networks

- **Both require adequate models of faults in CMOS integrated circuits**
Test Approaches

- Ad-hoc testing
- Scan-based Test
- Self-Test

Problem is getting harder
- increasing complexity and heterogeneous combination of modules in system-on-a-chip.
- Advanced packaging and assembly techniques extend problem to the board level

Scan-based Test
Polarity-Hold SRL (Shift-Register Latch)

Introduced at IBM and set as company policy (LSSD)

Scan-Path Flip-Flop
Scan Flip-Flop in AMD K-6

Intel/HP Itanium 2

Naffziger, ISSCC'02
Scan-based Test —Operation

Scan Test

(a) With Clock-Mux Type Scan-flop

(b) With Data-Mux Type Scan-flop
Self-test

Rapidly becoming more important with increasing chip-complexity and larger modules

Linear-Feedback Shift Register (LFSR)

Pseudo-Random Pattern Generator
Signature Analysis

Counts transitions on single-bit stream
≡ Compression in time

BILBO

<table>
<thead>
<tr>
<th>$B_0$</th>
<th>$B_1$</th>
<th>Operation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Scan</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Pattern generation or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signature analysis</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reset</td>
</tr>
</tbody>
</table>
BILBO Application

Built-In Self-Test
Memory Self-Test

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

Boundary Scan (JTAG)

Board testing becomes as problematic as chip testing
Testing on-chip Logic

Testing Mixed Analog-Digital ICs
Embedded Software-Based Self-Testing for Programmable System Chips

- Low-cost tester
- High-quality at-speed test
- Low test overhead
- Non-intrusive

Loading test program at low speed
Self-test at operational speed
Unloading response signature at low speed

Test in normal operational mode
- No violation of power consumption
- More accurate speed-binning