VLSI Testing

Lecture 25
18-322 Fall 2003
Announcement

- Homework 9 is due next Thursday (11/20)

- Exam II is on Tuesday (11/18) in class

- Review Session:
  - When: Next Monday (11/17) afternoon, 4pm – 6pm
  - Where: B131, HH
Outline

- Defects and Faults
  - Reasons for IC malfunctioning

- Fault Modeling
  - Types of faults (Stuck-At, bridge, Stuck Open)
  - Automatic Test Pattern Generation
  - Path Delay Fault

- Design for Testability
Why Testing?

- Manufacturing is imperfect

  - $Y = \frac{\text{No. of good chips on wafer}}{\text{Total no. of chips}}$

  - Yield ($Y$) depends on technology, chip area and layout
    - $Y$ decreases as the area of chip is increased
    - Defect density ($D$)
      - Modern technologies yield a value of 1-5 defects/cm$^2$
    - Yield starts out low (~10%) moves up (95%)

- High quality expectation

  - The earlier you detect a fault, the cheaper it is to fix
Reasons for IC Malfunction - Contamination, Defects and Faults

- Contamination / Instabilities - Process induced impurities and random fluctuations of process conditions

- Defects - Permanent deformation in IC layer which may but does not have to result in fault

- Faults - Functional misbehaviors i.e. IC malfunctions
Reasons for IC Malfunction - Defects and Faults
Reasons for IC Malfunction - Defects and Faults
Reasons for IC Malfunction - Defects and Faults
Reasons for IC Malfunction - Defects and Faults

\[ \begin{array}{c|c|c|c}
A & B & C & \text{Defect} \\
\hline
0 & 0 & 0 & 0 \quad 0 \\
0 & 0 & 1 & 0 \quad 0 \\
0 & 1 & 0 & 0 \quad 0 \\
0 & 1 & 1 & 1 \quad 1 \\
1 & 0 & 0 & 0 \quad 0 \\
1 & 0 & 1 & 1 \quad 0 \\
1 & 1 & 0 & 1 \quad 0 \\
1 & 1 & 1 & 1 \quad 1 \\
\end{array} \]

N1 - N13 short
Outline

✓ Defects and Faults
  ▲ Reasons for IC malfunctioning

■ Fault Modeling
  ▲ Types of faults (Stuck-At, bridge, Stuck Open)
  ▲ Automatic Test Pattern Generation
  ▲ Path Delay Fault

■ Design for Testability
Test Complexity

Exhaustive test $2^n + m$

Circuit with $n = 25$ and $m = 50$

1µsec/test

Exhaustive test time is over 1 billion years!
(Registers make life harder!)
Testing Strategies

- **Functional Test: (go/no go)**
  - Does the part work?
  - Do this fast & cheap

- **Diagnostic Test:**
  - What in the chip is broken?

- **Parametric Test:**
  - What is:
    - max clock frequency
    - min supply voltage
    - max operating temp
Test Implementation

- Runs Test Vectors/Programs on Device Under Test (DUT)
  - Goal: Find a SMALL set of test vectors that has a BIG fault coverage

- Testers
  - Clock rate in the range of GHz
  - Resolutions measured in psec
  - Large very fast memory
  - Cost 1 - 5 million dollars
Fault Models

• Modeling physical faults is complex
• Need models that simplify the behavior of faults
Stuck-At Fault

**Stuck-at-0**

**Stuck-at-1**
Bridge & Stuck Open

Bridging fault

Open fault
Automatic Test Pattern Generation (ATPG)

- Given a logic circuit:
  - Generate test program to cover all SA faults

- The D-Algorithm
  - The D-Calculus
    - Problem: Reconvergent Fanouts
D-Algorithm

- Step 1: Choose a fault to “insert”
  - Select from a fault dictionary

- Step 2: Activate (excite) the fault
  - Drive the faulty node to the opposite value of the fault
  - Example: for SA-1, drive the node to 0

- Step 3: Sensitize a path to an output
  - Propagate the fault so that it can be observed at the output pin
Path Sensitization

Goals: Determine input pattern that makes a fault controllable (triggers the fault, and makes its impact visible at the output nodes)

Fault enabling

Fault propagation

Techniques Used: D-algorithm, Podem
D-Algorithm

\[ D = 1/0 \]

value in good ckt

value in faulty ckt

\[ \overline{D} = 0/1 \]
### D-Algorithm

#### Five value logic simulation

<table>
<thead>
<tr>
<th>A</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D</td>
<td>Ø</td>
</tr>
<tr>
<td>Ø</td>
<td>D</td>
</tr>
</tbody>
</table>

\[ X = \overline{\text{NOT}(A)} \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 1</td>
</tr>
<tr>
<td>X</td>
<td>X X</td>
</tr>
<tr>
<td>D</td>
<td>D X</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø X</td>
</tr>
</tbody>
</table>

\[ X = AB \]
D-Algorithm

Five value logic simulation

\[ X = A + B \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>( \bar{D} )</td>
<td>( \bar{D} )</td>
</tr>
</tbody>
</table>
D-Algorithm

1/0 = D
D-Algorithm

Conflict!
Need backtracks

Reconvergent Fanout
Fault Simulation

Test Program

Fault Free Circuit

Circuit w/ One Fault

Compare

Random Number Generator, Genetic Algorithm, etc.
Path Delay Fault

- A defect can affect the speed of a path in the circuit

- Let’s see a Path Delay Fault example

![Diagram of a circuit with labeled inputs and outputs showing the effect of a path delay fault.]

T0 a slow↑ b 11 c slow↑ d 11 e slow↑ T1+ΔT
Path Delay Fault

Path \( \uparrow c-g2-g4-g5-x \)
Outline

✓ Defects and Faults
  ▲ Reasons for IC malfunctioning

✓ Fault Modeling
  ▲ Types of faults (Stuck-At, bridge, Stuck Open)
  ▲ Automatic Test Pattern Generation
  ▲ Path Delay Fault

■ Design for Testability
Scan-based Test

Modified to support two operation modes

Register → Combinational Logic A → Register

ScanIn

In

→ Out

ScanOut
Scan Based Methods

Level Sensitive Scan Design (LSSD) - IBM

Test Mode: OFF

Test Mode: ON
Boundary Scan (JTAG: IEEE 1149.1b)

Board testing becomes as problematic as chip testing
Built-In Self-Testing (BIST)

(Rapidly becoming more important with increasing chip-complexity and larger modules)
Linear-Feedback Shift Register (LFSR)

Pseudo-Random Pattern Generator