Basics of Test Technology

Adit D. Singh
Electrical & Computer Engineering
Auburn University
Introduction

- Many integrated circuits contain fabrication defects upon manufacture
- Only 20-60% for high end circuits manufactured may be defect free
- ICs must be carefully tested to screen out faulty parts before integration in systems
- Latent faults that cause early life failure must also be screened out through “burn-in” stress tests
IC Testing is a Difficult Problem

- Need $2^3 = 8$ input patterns to exhaustively test a 3-input NAND
- $2^N$ tests needed for N-input circuit
- Many ICs have > 100 inputs

$$2^{100} = 1.27 \times 10^{30}$$

Applying $10^{30}$ tests at $10^9$ per second (1 GHZ) will require $10^{21}$ secs = 400 billion centuries!

- Only a very few input combinations can be applied in practice
IC Testing in Practice

For high end circuits

- A few seconds of test time on very expensive production testers
- Many thousand test patterns applied
- Test patterns carefully chosen to detect likely faults
- High economic impact
  - Test costs are approaching manufacturing costs

Despite the costs, testing is always imperfect!
How well must we test?

Approximate order of magnitude estimates

- Number of parts per typical system: 100
- Acceptable system defect rate: 1% (1 per 100)
- Therefore, required part reliability
  
  1 defect in 10,000
  
  100 Defects Per Million (100 DPM)

Requirement ~100 DPM for commercial ICs
~1000 DPM for ASICs
“Zero Defect” target for automotive
How well must we test?

Assume 2 million ICs manufactured with 50% yield

- 1 million GOOD >> shipped
- 1 million BAD >> test escapes cause defective parts to be shipped

For 100 BAD parts in 1M shipped (DPM=100)
Test must detect 999,900
out of the 1,000,000 BAD

For 100 DPM: Needed Test Coverage = 99.99%
DPM and System Failure Probability

Defective Parts per Million parts shipped:

- ~ 100 DPM (0.01%) for commercial ICs
  - System with 10 ICs => 0.1% Failure Probability
  - System with 100 ICs => 1.0% Failure Probability
  - System with 500 ICs => 5.0% Failure Probability

- < 10 DPM Automotive Industry

Target: “Zero” defects!
Classical Yield Models

Two classes of Manufacturing Defects

- Gross or area defects
- Random Spot Defects

In mature well controlled processes, die yield is mostly limited by random spot defects
  - impossible to completely eliminate
The simplest defect distribution model for semiconductor wafers assumes that random spot defects are uniformly distributed.

\[
\text{Die Yield} = e^{-\lambda}
\]

\[
\lambda = \text{Average number of defects per Die} = \text{Defect Density (\sim 0.2 - 1.0 per cm}^2) \times \text{Die Area}
\]
Die Yield = $e^{-\lambda}$

$\lambda$ = Average number of defects per Die 
= Defect Density 
$\times$ Die Area

- Yield = $1/e$ (37%) for $\lambda = 1$
- Defect Density $\sim 0.5$ defects/sq cm
- Largest Die are $\sim 2$ sq cms
The Poisson model has been found to consistently underestimate yield. This suggests, defects on semiconductor wafers are not uniformly distributed but are clustered. For a given total number of defects on the wafer, defect clustering results in more die with multiple defects, and therefore more defect free die (higher yield).
DPM depends on Yield

For Test Coverage: 99.99%

(Escapes 100 per million defective)

- 1 Million Parts @ 10% Yield
  
  0.1 million GOOD >> shipped
  0.9 million BAD >> 90 test escapes
  
  \[ DPM = \frac{90}{0.1} = 900 \]

- 1 Million Parts @ 90% Yield

  0.9 million GOOD >> shipped
  0.1 million BAD >> 10 test escapes
  
  \[ DPM = \frac{10}{0.9} = 11 \]
Testing Large Complex Die

Testing large, complex low yielding die is the biggest challenge

- Higher DPM even for equally effective (similar “coverage”) tests because of lower yields

- Difficult to achieve high coverage testing for large complex die

- DPM increases non linearly with die complexity
Real Defect Types

Actual manufacturing defects, flaws, variability etc. can have very complex interactions leading to unpredictable anomalous electrical behavior

- Permanent or hard faults
- Difficult to achieve high coverage testing for large complex die
- DPM increases quite non-linearly with die complexity
Test Basics
Input \((a_1, a_2, a_3 \ldots an)\) is a \textbf{test} for fault \(\alpha\) iff
\[
f(a_1, a_2, a_3 \ldots an) = /= f_\alpha(a_1, a_2, a_3 \ldots an)
\]

\textbf{Note:} We are only interested in knowing if the DUT is faulty, not in diagnosing or locating the fault.
For an n input circuit, there are $2^n$ input combinations. Ideally we must test for all possible faulty functions. This will require an exhaustive test with $2^n$ inputs.

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Since we cannot apply the exhaustive test, set our best bet is to target likely faults!
Defects Faults and Errors

A **Defect** is a physical flaw in the device, i.e. a shorted transistor or an open interconnect.

A **Fault** is the logic level manifestation of the Defect, i.e. a line permanently stuck at a low logic level.

An **Error** occurs when a fault causes an incorrect logic value at a functional output.
Test Basics

Likely defects
- Depend on the circuit, layout, process control
- Difficult to obtain

Simplify the problem by targeting only Logical Faults

Fault Model

Physical Defects Faults  Logical
The Stuck-at Fault Model

Assumes defects cause a signal line to be permanently stuck high or stuck low

- s-a-0  Stuck-at 0
- s-a-1  Stuck-at 1

- How good is this model?
- What does it buy us?
Stuck-at Test for NAND4

Fault List:
Possible Faults \{A/0, A/1, B/0, B/1, C/0, C/1, D/0, D/1, Y/0, Y/1\}

<table>
<thead>
<tr>
<th>Test</th>
<th>Faults Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1</td>
<td>A/0, B/0, C/0, D/0, Y/1</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>A/1, Y/0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>B/1, Y/0</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>C/1, Y/0</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>D/1, Y/0</td>
</tr>
</tbody>
</table>

Test Set size = \(n+1\) not \(2^n\)
Was reasonable for Bipolar technologies and NMOS

Less good for CMOS