Lecture 9

- The test problems
- Fault modeling
- Design for testability techniques
What is Testing?

- To examine a product to ensure that it functions correctly and exhibits the properties and capabilities it was designed to possess.

- Testing is an important activity in the life-cycle of an electronics product.

Life-Cycle of an Electronic System

- Specification
- Design
- Implementation
- Production
- Production Test
- Integration Test
- System Test
- Operation and Maintenance

Validation
Verification
Review
Inspection
Simulation

Test Preparation
and DFT

Testing
Causes of Incorrect Functions

- Design errors
  - usually consistent
- Fabrication (manufacturing) errors
  - often consistent, e.g., wrong components
  - usually mistakes by operators
- Fabrication (manufacturing) defects
  - inconsistent, e.g., impurity of materials
- Physical failures
  - wear-out
  - environmental factors

Defect Example
The Stuck-At Fault Model

- Every fault in a circuit changes its functionality as if some nodes were steadily tied to either logic 0 or 1.

A single stuck-at model is usually used. That is, we assume that only one node at a time is tied to 0 or 1.

Total no of faults = \(3^N - 1\)

Single Stuck-At (SSA) Fault Model

- Number of possible faults is small (2N).
- Capable of representing many different physical faults.
- Test patterns produced for SSA faults detect also many other faults.
- Facilitate the use of Boolean algebra mathematics in deriving test patterns.
- Technology independent.

→

- SSA fault coverage remains an established metric for test quality, and is widely used in the industry.
**Bridging Faults**

- Technology dependent (e.g., a short-circuit can lead to an AND-bridging, OR-bridging or dominance-bridging).
- Faults are related to the placement of the components.
- Fault activation and propagation are more difficult.

**Composition of Costs of Testing**

- **Cost of test equipment (hardware):**
  - A test controller (usually a computer).
  - Interface drivers and receivers and cable-connections.
  - System of probe-contacts.
  - A controlled environment (e.g., a temperature chamber).

- **Cost of software supports:**
  - Test pattern generation programs.
  - Test-design verification procedures (fault simulation and analysis).

- **Testing time:**
  - Test development time.
  - Test application time.
Test = Optimization

Testing and DFT are optimization problems:

- To minimize the total test cost, \( C(x) \), defined as:
  \[
  C_{\text{test}} = C_{\text{dev}} + C_{\text{appl}} + C_{\text{silicon}} + C_{\text{quality}}
  \]

- Subject to a set of constraints, \( G_i(x), i = 1, 2, \ldots k \)
  - test strategy and methods
  - fault coverage requirement
  - power consumption
  - ATE and tool limitation
  - test and DFT engineers
  - ...

Types of Testing

- Production test — testing of individual products to check whether faults are introduced during the manufacturing phase. It is assumed that the design is correct.

- System test — testing of the product in the environment where it is operating to ensure that it works correctly when interconnected with other components.

- Burn-in — testing at elevated temperature and voltage to accelerate and detect early life failures.

- Operation and maintenance test — testing of the product in the fielded for diagnosis or "preventive" purpose.

- Prototype test — testing to check for design faults during the system development phase. Diagnosis is required.
Lecture 9

- The test problems
- Fault modeling
- Design for testability techniques

Design for Testability (DFT)

- To take into account the testing aspects during the design process so that more testable designs will be generated.
- Advantages of DFT:
  - Reduce test efforts.
  - Reduce cost for test equipment.
  - Shorten time-to-market.
  - Increase product quality.
- Limitations:
  - Hardware overhead, 5-30%, and performance degradation.
  - Increased design complexity.
Controllability and Observability

- The key to design for testability is the ability to control and observe directly the internal states.
- Controllability of an internal node - ability to set it to a specific logic value.
- Observability of an internal node - ability to observe its specific logic value on primary output.

Ad Hoc DFT Techniques

Good design practices learnt through experience are used as guidelines:

- Test point insertion:
  - Very simple.
  - Can be used in principle anywhere.
  - Large overhead of I/O ports (→ multiplexing and addressing)
Ad Hoc DFT Techniques (Cont’d)

- Design circuits to be initializable.
- Provide easy reset.
- Eliminate asynchronous logic.
- Control/bypass internal clocks and oscillators.

![Diagram](Image)

Ad Hoc DFT Techniques (Cont’d)

- Redundancy must be avoided or controlled.

\[ z = ab + bc + \overline{ac} = ab + \overline{ac} \]

A Hazard-Free Circuit

- High sequential depth should be avoided.
- Combinational feedback loops should be avoided since they result in additional hidden storage elements.
Disadvantages of Ad-Hoc Methods

- Experts and tools not always available.
- Test generation must often be manually performed with no guarantee of high fault coverage.
- Design iterations may be needed, which is very time consuming.

Structural DFT Techniques

Can be applied to any design in a systematic way:

- Full scan.
- Partial scan.
- Boundary scan.
- Built-in self-test (BIST).
Basic Scan Technique

- Sequential circuits have poor controllability and poor observability.
- To address this problem, the flip-flops are connected into a long shift register.

Scan Design

- Circuit is designed using pre-specified design rules.
- Test structure is added to the completed design:
  - Add a test control (TC) primary input.
  - Replace flip-flops by scan flip-flops (SFF), and connect them to form one or several shift registers in the test mode.
  - Make input/output of each scan shift register controllable/observable from PI/PO.
- Full scan = all flip-flops are included in the scan path:
  - The problem of ATPG for sequential circuits is reduced to ATPG for combinational ones.
Scan Design Rules

- Use only clocked D-type of flip-flops for all state variables.
- At least one PI pin must be available for test; more pins, if available, can be used.
- All clocks must be controlled from PIs.
- Clocks must not feed data inputs of flip-flops.

Correcting a Rule Violation

- All clocks must be controlled from PIs.
Scan Flip-Flop (SFF)

Scan Overheads

- IO pins: One pin necessary.
- Area overhead:
  - Gate overhead = \(\frac{4 \times Nff}{(Ng + 10 \times Nff)} \times 100\%\),
    where \(Ng\) = No. of comb. gates; \(Nff\) = No. of flip-flops.
    - Example: \(Ng = 100k\) gates, \(Nff = 2k\) flip-flops, overhead = 6.7%.
    - More accurate estimate must consider scan wiring and layout area.
- Performance overhead:
  - Multiplexer delay added in combinational path; approx. two gate-delays.
  - Flip-flop output loading due to one additional fanout; approx. 5-6%.
Partial Scan

- A subset of FFs is selected for the scan path.
- Advantages:
  - reduced area overhead.
  - avoid performance degradation.
  - reduced test application time.
- Disadvantages:
  - design time may increase.
  - need a sequential ATPG tool.
  - need a method for selecting the partial scan subset.

Partial-Scan Architecture
A Partial-Scan Method

- Select a minimal set of flip-flops for scan to eliminate all cycles.

- Alternatively, in order to keep the overhead low, only long cycles will be eliminated.

- In some circuits with a large number of self-loops, all cycles other than self-loops will be eliminated.

Partial Scan Example

Circuit: TLC, 355 gates, 21 flip-flops

<table>
<thead>
<tr>
<th>Scan flip-flops</th>
<th>Max. cycle length</th>
<th>Depth*</th>
<th>ATPG CPU s</th>
<th>Fault sim CPU s</th>
<th>Fault cov.</th>
<th>ATPG vectors</th>
<th>Test seq. length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>14</td>
<td>1,247</td>
<td>61</td>
<td>89.01%</td>
<td>805</td>
<td>805</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>10</td>
<td>157</td>
<td>11</td>
<td>95.90%</td>
<td>247</td>
<td>1,249</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>5</td>
<td>32</td>
<td>4</td>
<td>99.20%</td>
<td>136</td>
<td>1,382</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>3</td>
<td>13</td>
<td>4</td>
<td>100.00%</td>
<td>112</td>
<td>1,256</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>52</td>
<td>1,190</td>
</tr>
</tbody>
</table>

* Cyclic paths ignored
Summary

- Testing is mainly used to find physical defects introduced during the manufacturing and operation phases.

- It is an expensive and complex task, and is becoming more and more difficult with the development of complex systems.

- Testability must be taken into account at all stages of the design and synthesis process.
  - In particular, early testability consideration prevents costly design iterations.

- The key to successful testing lies in the design process!