Testing Digital Systems II

Lecture 1: Introduction

Instructor: M. Tahoori

Today’s Lecture

- Logistics
- Course Outline
- Review from TDS I
Logistics

- Instructor: Mehdi Tahoori
  - Office: Room B2-313.1, Building 07.21
  - Email: mehdi.tahoori@kit.edu
  - Tel: 721-608-7778, Fax: 721-608-3962

- Lecture:
  - When: Wednesdays 14:00-15:30
  - Where: Room 301, Building 50.34

Logistics (cont)

- Requirements
  - Logic Design
  - Computer Architecture
  - Testing Digital Systems I
  - Background on
    - Algorithms and Programming
    - Hardware description languages (VHDL or Verilog)
Reference Books

Textbook


Recommended


Course Outline

- Basics
- Design for Testability (DFT)
- Built-in Self-test (BIST)
- More topics on test generation and DFT

I try to be flexible. The order and contents may be changed as we proceed.
Outline: Basics

- Review of TDS I
  - Digital design, test, and verification flow
  - Failures and errors
  - Fault models
  - Test pattern generation
    - Combinational ATPG
    - Sequential ATPG

Outline: Design For Testability (DFT)

- Ad-Hoc DFT techniques
- Internal scan design
- Boundary scan
Outline: BIST

- Test Pattern Generation
- Output Response Analysis
- BIST Architectures

Contents: More topics

- Fault list reduction and test compaction
- Delay testing
- Logic Diagnosis
- Memory testing
- Test Compression

- Other interesting topics if time permits
Overview of TDS I

VLSI Realization Process

Customer's need
Determine requirements
  specifications
Design synthesis and Verification
  Test development
  Fabrication
  Manufacturing test
Chips to customer
Definitions

- **Design synthesis:**
  - Given an Input-Output function, develop a procedure to manufacture a device using known materials and processes

- **Verification:**
  - Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given Input-Output function

---

Testing

- The process of determining whether a piece of device
  - Is functioning correctly, or
  - Is defective (broken or faulty)

- Equipment can be defective because it doesn't function
  - as designed, or
  - as specified
Testing Process

Components, Raw Materials

Manufacturing Process → Yield
Fraction of good parts
20%-90%

Testing

Shipped Parts
Quality: DPM
Defective Parts per Million
50-500 DPM

Testing as Filter Process

Good chips
Prob(good) = y

Fabricated chips

Defective chips
Prob(bad) = 1 - y

Prob(pass test) = high
Mostly good chips

Overkill
Escapes

Test
(yield loss)

Prob(fail test) = high
Mostly bad chips
Costs of Testing

- **Design for testability (DFT)**
  - Chip area overhead
  - Yield reduction
  - Performance overhead

- **Software processes of test**
  - Test generation
  - Fault simulation
  - Test programming and debugging

- **Manufacturing test**
  - *Automatic test equipment* (ATE) capital cost
  - Test center operational cost

Roles of Testing

- **Detection:**
  - Determination whether or not the *device under test* (DUT) has some fault.

- **Diagnosis:**
  - Identification of a specific fault that is present on DUT.

- **Device characterization:**
  - Determination and correction of errors in design and/or test procedure.

- **Failure mode analysis** (FMA):
  - Determination of manufacturing process errors that may have caused defects on the DUT.
Testing Principle

- Performed by
  - Automatic Test Equipment (ATE)
  - On-chip Built-in Self Test (BIST)

Testing Taxonomy

- Design Verification
- Explicit Testing
- Implicit Testing Or On-line Testing
- Characterization
- Production Test
- Reliability Test
- Qualification
- Diagnosis
- Wafer Sort
- Package Test
- IDDQ
- VLV
- Burn-in
- Min Vdd
- SHOVE
- System Test
- Gross Test
- DC parametric Test
- AC parametric Test
- Boolean Test
- Quasi-Boolean (Delay) Test
### Types of Test

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production Test</td>
<td>Tests to sort out defective manufactured parts</td>
</tr>
<tr>
<td>Wafer Sort or Probe</td>
<td>Test of each die while still on the wafer</td>
</tr>
<tr>
<td>Final or Package Test</td>
<td>Test of packaged chips and separation into classes or bins (military, commercial, industrial)</td>
</tr>
<tr>
<td>Acceptance Test</td>
<td>A test to demonstrate the degree of compliance of a device with purchaser's requirements</td>
</tr>
<tr>
<td>Sample Test</td>
<td>Test of some but not all parts</td>
</tr>
<tr>
<td>Go / No Go Test</td>
<td>Test to determine whether device meets specification</td>
</tr>
<tr>
<td>Characterization</td>
<td>Test to determine actual values of device AC and DC parameters and the interaction of parameters. Used to set final specifications and to identify areas to improve process to increase yield.</td>
</tr>
<tr>
<td>Stress Screening</td>
<td>Test with stress (high temperature, temperature cycling, voltage, vibration, etc.) applied to eliminate short life parts</td>
</tr>
<tr>
<td>Reliability Test (Accelerated Life Test)</td>
<td>Test after subjecting the part to extended high temperature or voltage to estimate time to failure in normal operation</td>
</tr>
<tr>
<td>Diagnostic Test</td>
<td>Test to locate failure site on failed part</td>
</tr>
<tr>
<td>Quality Test</td>
<td>Test by quality assurance department of a sample of each lot of manufactured parts. More stringent than final test.</td>
</tr>
<tr>
<td>On-line Test*</td>
<td>On-line testing to detect errors that occur during normal system operation.</td>
</tr>
<tr>
<td>System Test</td>
<td>Test by plugging a device into an actual system and running the system.</td>
</tr>
<tr>
<td>Design Verification</td>
<td>Verifying the correctness of a design</td>
</tr>
</tbody>
</table>

### Test Flow

- **Manufacturing**
  - In-line Wafer Tests
    - DC Parameters
    - Functional
    - Logic
    - Delay
  - Wafer Sort
    - GO/NO-GO

- **Packaged Device**
  - Package Test
  - Burn-In
  - Test escapes
  - Customer
  - Incoming inspection

- **System Integration**
  - System Test
  - Customer
  - Test escapes

- **Die**
  - **Fallout**

- **Masks**
Failures, Errors, Faults

- Chip is Defective if
  - it Doesn’t Function
    - as Specified, or
    - as Designed due to Presence of a Failure
- Error
  - Incorrect Signal Value
- Failure
  - Deviation from Designed Characteristics
- Fault
  - Models Effect of Failure on Logical Signals

Open

- Failure Mechanism: Particle on
- Failure Mode: Open Metal
- Electrical Fault: Open Signal
- Logical Fault Model: stuck-at-1
Short

Failure Mechanism: Particle on

Failure Mode: Shorted Metal

Electrical Fault

Logical Fault

Fault Model

- Fault model
  - Models effect of physical failure on logic network
  - Abstraction of physical situation
  - Used to describe the change in the logic function of a device caused by the defect.

- Various levels of abstraction are used
  - Functional (Board, Chip) level
  - Register transfer (Behavioral) level
  - Logic level
    - Gate library level
    - Elementary gate level
  - Switch level
  - Transistor (Spice) level
Fault Model Taxonomy

Fault Models for Logic Circuits

- "High" level or Functional level or RT-level
- Boolean Logic Network level
- Transistor level
  - Stuck-on, Stuck-open
  - Cross-check
  - Gate-to-source or Gate-to-drain shorts

Stuck Fault Models

- Structural logic-level fault model
  - Start with the circuit represented as a netlist of Boolean gates
  - Assumes faults only affect the interconnection between gates
- Single Stuck Fault
  - Logic network of elementary gates
    - AND, OR, NAND, NOR, NOT
  - One Line has Fixed 0 or 1 Value
  - Independent of other signal values
  - One fanout branch can be stuck
  - Most common model for Boolean test
  - Written \( L_i/h, h = 0 \) or \( 1 \)
- Multiple Stuck Fault
  - One or More Stuck Line Faults Present
- Pin Fault
  - Stuck Fault on I/O Connection of a Module
Stuck-at Fault Model

Stuck-at Fault Model

- Short to VDD
- Open signal lead

Notation: \( A/1 \) or \( A_1 \)

Fault Detection

- An input combination detects a fault in a logic network if
  - the response of the faulty logic network to that input combination is different from that of the fault-free network
  - The input combination is called a test pattern for the fault

- Fault detection requires:
  - A test activates or provokes the fault \( f \).
  - \( t \) propagates the error to an observation point
    - e.g. primary output
  - A line whose value changes with \( f \) present is said to be sensitized to the fault site.
Single Stuck-at

- 14 faults
  - 2 faults (SA0, SA1) per each line
- $ABCD = 1100$ detects $F/1$
  - Faulty and fault-free outputs different
- $ABCD = 1101$ does NOT detect $F/1$
  - Faulty and fault-free outputs are the same

![Logic Diagram]

Untestable Fault

- A fault that does not affect the logical behavior of a circuit (redundant fault)
  - Untestable by Particular Test Procedure

Causes
- Redundant Circuitry
  - Design Error
- Hazard Control Circuitry
- Error Detection Circuitry
  - Parity Check
- Excess Components
  - Needed for Performance, not Functionality

![Logic Diagram]
Bridging Fault

- Bridging faults appear when two or more normally distinct signal lines in a Boolean logic network are unintentionally shorted together and create wired logic.
- A feedback bridging fault is a special type of bridging fault which is created when one of the two shorted signal lines depends on the other signal line in the fault-free circuit.
  - May cause oscillation or latch
- If a fanout branch of a signal line is involved in a bridge
  - Logic value on the fanout stem and the other fanout branches of that signal line will be the same as the logic value on the fanout branch which is involved in the bridge
Timing Failures

- Logic Network has a Timing Failure if and only if
  - it fails to operate correctly at its specified speed
- BUT
  - may produce correct outputs when operated at either
    - a slower or faster speed
- Compared to Stuck-at or bridging fault models
  - Static faults
    - Incorrect values at any speed

Path Delay Fault

- Path delay fault present
  - propagation delay of at least one path
    - from primary input to primary output exceeds clock interval
  - Models multiple or distributed defects
  - Issue: can path be sensitized, occur in operation?
- Each path delay fault
  - associated with a particular path
  - between primary input and output
  - either
    - all paths
    - all sensitizable paths
    - longest paths
      - static timing analyzer
Transition Fault

Definition
- A localized timing failure is large enough such that the delay of all paths through some gate to observable outputs exceed the clock interval.

Each transition fault
- associated with a particular gate input or gate output
- either a 0 to 1 transition or a 1 to 0 transition
  - (two transition faults)
    - slow-to-rise, slow-to-fall
  - propagated to some primary output

Transition Fault

two input combinations are needed

initialization pattern
- places an initial value at the fault site.
  - The initial value is 0 for a slow-to-rise transition fault,
  - and 1 for a slow-to-fall transition fault.

transition propagation pattern
- places the final transition value
  - 1 for a slow-to-rise transition fault, and 0 for a slow-to-fall transition fault
  - propagates the transition to an observable output
Fault Simulation

- Fault Simulator
  - A program that models a design with fault present
- Inputs:
  - A circuit
  - A sequence of test vectors
  - A fault model
    - Usually single-stuck faults
    - Sometimes multiple-stuck, bridging faults, ...
- Determines
  - Fault coverage (fault grading)
  - Set of undetected faults (Areas of Low Fault Coverage)
  - Generates fault dictionary (Fault diagnosis)
  - Test compaction

Fault simulator in VLSI Design

Verified design netlist
Verification input stimuli
Fault simulator
Modeled fault list
Test stimuli
Fault coverage?
Low
Adequate
Stop
Remove tested faults
Test compactor
Delete vectors
Add vectors
Test generator
Specific-Fault Oriented Test Generation

Two fundamental test generation steps

- **ACTIVATE**, Excite, Provoke or Setup the Fault
  - Make Fault OBSERVABLE, Fault Sensitization
  - Find Primary Input Values that Cause Error Signal in Faulty Circuit
  - For Single-Stuck-at-\( v \) Fault
    - Place \( v' \) at Fault Site

- **PROPAGATE** the Resulting Error to a Primary Output
  - Path Sensitization
  - Find Primary Input Values that Sensitize Error Signal to Primary Output

Example: Test for \( c/0 \) is \( w,x,y = 0,1,1 \)

- **ACTIVATE** Fault \( c/0 \)
  - Set \( x = y = 1 \) to make \( c=1 \)
    - in Fault-free Circuit
  - **PROPAGATE** Value on \( c \) to \( f \)
    - Set \( w =0 \) to sensitize \( c \) to \( f \)
Test Generation Using Path Tracing

- **Notation**
  - D Signal Value
    - 1 in Fault-free Circuit, 0 in Faulty Circuit
  - D' or D Signal Value
    - 0 in Fault-free Circuit, 1 in Faulty Circuit
  - X
    - Signal Value is Unspecified

- **Truth table for AND**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>X</th>
<th>D</th>
<th>D'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>D</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>D</td>
<td>X</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>D</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

Path Sensitization Method

- **Fault Sensitization**
  - Force tested node to opposite of fault value
- **Fault Propagation (path sensitization)**
  - Propagate the effect to one or more POs
- **Line Justification**
  - Justify internal signal assignments made to activate and sensitize fault

- **These three steps may result in conflict**
  - Different values are assigned to the same signal
  - Require **backtracking**
Path Sensitization Method

- Try path f - h - k - L
  - Requires A = 1, j = 0, E = 1
- Blocked at j
  - Since there is no way to justify 1 on i

```
Try simultaneous paths f - h - k - L and g - i - j - k - L
```

- Blocked at k because
  - D-frontier (chain of D or \( \overline{D} \)) disappears
### Path Sensitization Method

- Final try: path $g \rightarrow i \rightarrow j \rightarrow k \rightarrow L$
  - test found!

```
A  0
B  1
C  1
D  1
E  1
```

```
A  0
B  1
C  1
D  1
E  1
```

### Specific-Fault Oriented Test Generation

- Three Approaches
  - **D Algorithm**: Internal Line Values Assigned (Roth-1966)
    - D-cubes
    - Bridging faults
    - Logic gate function change faults
  - **PODEM**: Input Values Assigned (Goel – 1981)
    - X-Path-Check
    - Backtracing
  - **FAN**: Input and Internal Values Assigned (1983)
Sequential ATPG

- **Approach**
  - Convert Finite State Machine to Corresponding Iterative Network
    - Multiple Time Frames (Iterative Cells) Needed for Justification and Propagation
  - One Fault in Sequential Circuit
    - Many faults in Corresponding Iterative Network
    - Use 9-valued signals

- **Issues**
  - Order of Justification and Propagation
  - Simulation Values
  - Test Point Insertion (Partial Scan)

---

**General Case**

- Huffman model of sequential circuit with edge-triggered D-flip-flops

- Any sequential circuit with edge-triggered D-FF can be directly converted into an iterative network
Iterative Logic Array Expansion

- To detect a fault, a sequence of vectors may be needed

Example

- Test for P SA0
  - Provoke Fault on P1: \( a_1 = 0, b_1 = 1 \)
  - Propagate Fault to S2:
    - \( C_0 = 1 \)
      - Need to consider last time frame: \( a_0 = 1, b_0 = 1, \text{Cin} = X \)
    - \( a_2 = 0, b_2 = 0 \)