Lecture 5: Built-in Self Test (I)

Instructor: M. Tahoori

Outline

- Introduction (Lecture 5)
- Test Pattern Generation (Lecture 5)
  - Pseudo-Random
  - Pseudo-Exhaustive
- Output Response Analysis (Lecture 6)
  - Duplication
  - Response Compaction
    - Signature Analysis
- BIST Architectures (Lecture 7)
### Built-In Self Test

**Definition:**
- Capability of a product
  - chip, multichip assembly, or system
- To carry out an explicit test of itself

**Requires**
- Test Pattern Generation
- Output Response Analysis
- One or both integral to the product
- Minimal external test equipment required

---

### A Typical Logic BIST System

**Structural off-line BIST**
Built-In Self Test

- Why BIST (Built-In Self Test)?
  - Improved product quality
  - Faster debug
  - Better diagnosis
  - Thorough test — very many high-speed patterns
  - Economical production test
  - Improved field test and maintainability
- What are its drawbacks?
  - Initial design investment
  - Possible performance or area overhead

BIST Techniques

- Enhanced functional self-test software routines
- Exhaustive and pseudo-exhaustive
- Pseudo-random (PR-BIST)
BIST Techniques

- Enhanced functional self-test software routines
  - For field test and diagnosis
  - Advantage:
    - No hardware modifications
  - Disadvantages:
    - Low hardware fault coverage
    - Low diagnostic resolution
    - Slow to operate
    - Labor intensive, low fault coverage

- Exhaustive and pseudo-exhaustive
  - + Thorough test of stuck faults
  - + Minimal simulation required
  - - Difficult to implement for arbitrary designs

BIST Techniques

- Pseudo-Random (PR-BIST)
  - Separate (Serial Scan-Loaded Test Patterns)
    - External Pattern Generation, Response Analysis
  - Embedded PR-BIST (System Bistables Reconfigured)
    - BILBO - Multiple Test Configurations
    - Circular
BIST Attributes: Fault Characteristics

Fault classes tested
- Single-stuck faults in functional circuitry
- Combinational faults in functional circuitry
- Delay faults
- Interchip wiring and chip I/O connections

Fault coverage
- Percentage of faults guaranteed to be detected

BIST Attributes: Cost Characteristics

Area overhead
- Additional active area, interconnect area
  - Test controller
  - Hardware pattern generator
  - Hardware response compacter
  - Testing of BIST hardware

Pin overhead: Additional pins required for testing
Performance penalty: Added path delays
Yield loss: Due to increased area
Reliability reduction: Due to increased area
BIST Attributes: Other Characteristics

- Generality
  - Degree of function dependence
- Time required to execute test
- Diagnostic resolution
- Engineering changes
  - Effect on BIST structure
- Functional circuitry
  - Scan path?
  - Design changes?

---

BIST Attributes: Other Characteristics

- Test Pattern Generation
  - Exhaustive
  - Pseudo-Exhaustive
  - Pseudo-Random
- Response Analysis
  - LFSR
  - Duplication
Exhaustive and Pseudo-Exhaustive Test

- Exhaustive Test of n-Input Combinational Circuit
  - Apply all \( N = 2^n \) Patterns
- Pseudo-Exhaustive Test of Combinational Circuit
  - Subdivide the Circuit into Segments
  - Apply all Possible Inputs to each Segment
- Input patterns — \( 2^m \) m-bit patterns
  - binary counter
  - Gray counter
  - m-stage Modified ALFSR

Test Patterns

- Stored Off Line
  - Patterns are generated and stored
  - Simulation used to identify patterns for removal
- "Just-in-Time"
  - Patterns are generated during test application
    - External tester generates patterns
    - Patterns generated on same chip or board as device under test
  - +Easy to Generate
  - +Detect Non- single-stuck faults
  - -Long
    - Coverage Expensive to Determine
Random vs Pseudorandom

- Random Source
  - Patterns can occur more than once
  - Non-reproducible
- Pseudorandom Source
  - All (possibly except all-0 pattern) Patterns
    - Occur Before Any Pattern Repeats
  - Reproducible

Test Pattern Generator
Pseudo-Random Test Pattern Generator

- **Four-stage ALFSR — Standard or External Form**
  - Autonomous Linear Feedback Shift Register

  ![Diagram of Four-stage ALFSR](image)

  - Output Sequence: \( a(t + 4) = a(t + 3) \oplus a(t) \)
  - Generating Function: \( f(x) = x^4 + x^3 + 1 \)
  - Feedback Vector: \( H = <h_4, h_3, ..., h_0> = <1, 1, 0, 0, 1> \)

Pseudo-Random Test Pattern Generator

- **Operator Notation**
  - \( X^i a(t) = a(t+i) \)
  - \( (X^3 + X + 1) a(t) = a(t+3) + a(t+1) + a(t) \)
### Standard Form ALFSR

- **Output Sequence:** \( a(t + N) = \sum_{i=0}^{N-1} h_i a(t + i) \mod 2 \)
- **Generating Function:** \( f(x) = \sum_{i=0}^{N} h_i x^i \mod 2 \)

### Four-Stage Modular ALFSR (Divider)

- **Generating Function:** \( f(x) = x^4 + x^3 + 1 \)
- **Feedback Vector:** \( H = \langle h_4, h_3, \ldots, h_0 \rangle = \langle 1, 1, 0, 0, 1 \rangle \)
### Modular ALFSR (Divider)

- **Generating Function:** \( f(x) = \sum_{i=0}^{N} h_i x^i \mod 2 \)

### 4-stage standard and modular LFSRs

- **4-stage Standard LFSR**
  \[ f(x) = 1 + x^2 + x^4 \]

- **4-stage Modular LFSR**
  \[ f(x) = 1 + x + x^4 \]
### Primitive VS Non-Primitive

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Non-Primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x^3 + x^1 + 1 )</td>
<td>( x^3 + x^2 + x + 1 )</td>
</tr>
</tbody>
</table>

Lc = 7
Lc = 4

### Four-Stage Modified ALFSR

- de Bruijn Counter
- Period = 16
N-Stage Modified ALFSR

- (de Bruijn Counter): Period = $2^N$

Minimized Four-Stage Modified ALFSR
Test Architecture

- m-stage ALFSR generates L m-bit patterns
  - L is test length
  - $M = 2^m - 1$ is number of patterns generated
- n is number of inputs for network under test (NUT)
  - $N = 2^n$ is exhaustive test length for NUT
    - Patterns generated on same chip or board
      - as device under test