Testability: Lecture 23
Design for Testability (DFT)

Shaahin Hessabi
Department of Computer Engineering
Sharif University of Technology

Adapted, with modifications, from lecture notes prepared by the book authors
Outline

- Iterative Logic Arrays
- Ad-hoc DFT methods
- *Scan design*
  - Design rules
  - Scan register
  - Scan flip-flops
  - Scan test sequences
  - Overheads
  - Scan design system
Iterative Logic Arrays (ILAs)

- Some circuits are easy to test, for example, ILAs
- Definition: An ILA is a $k$-dimensional array-like circuit composed of identical cells with uniform interconnections
- Array circuits can be tested for powerful fault models using relatively few tests
- Examples:
  - Arithmetic circuits
  - Ripple-carry adders
  - Array multipliers
  - Bit-sliced processors
  - Random-access memories: RAMs, ROMs
  - ILA models of sequential circuits
Example 1: Ripple-Carry Adder

- 1D array composed of full-adder cells
Example 1: Ripple-Carry Adder (cont’d)

- Assume the cell fault (CF) model, which implies that all single logic faults in all realizations will be detected.

- We must apply eight patterns to every cell and observe the responses.

- Six of the 8 patterns can be applied simultaneously to all cells; e.g., $A_iB_iC_i = 000$

- Faults in $F_{A_i}$ can be observed via $S_i$ or $C_{i+1}$

- Two of the 8 patterns cannot be applied simultaneously to all cells, namely $A_iB_iC_i = 001$ and 110, because $C_{in} \neq C_{out}$
Example 1: Ripple-Carry Adder (cont’d)

- The patterns $A_iB_iC_i = 001$ and $110$ can be applied simultaneously to alternating cells.
- All CF faults in an $n$-bit RC adder can be detected by 8 tests, independent of the array size $n$.
- The property of an $n$-cell ILA that all (cell) faults can be detected by a constant number test patterns for any $n$ is called $C$-testability.
Example 2: Gate Array

This is an ILA realization of a k-input AND function,

\[ k = 1, 2, 3, \ldots \]

- **Question 1**: Is an AND array C-testable?
- **Question 2**: What if the AND function is replaced by XOR?
Design for Testability (DFT)

- Test generation algorithms for logic circuits are complex (NP complete)
- Circuits containing, say, $10^6$ gates or $10^2$ flip-flops, may be too large for ATPG tools
- Heuristic methods are used for testing complex circuits such as microprocessors, RAMs, etc.
- Fault coverage of such methods can be low and hard to determine
- To ensure high levels of testability, design for testability (DFT) is often essential
Testability Goals

1. Maximize fault coverage
2. Minimize test application time
3. Minimize test data size
4. Minimize test generation effort

<table>
<thead>
<tr>
<th>Testability parameter</th>
<th>Testing method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PODEM</td>
</tr>
<tr>
<td>Fault coverage</td>
<td>good</td>
</tr>
<tr>
<td>Test data length</td>
<td>best</td>
</tr>
<tr>
<td>Test generation effort</td>
<td>worst</td>
</tr>
<tr>
<td>Suitability for BIST</td>
<td>bad</td>
</tr>
</tbody>
</table>
Definition

- **Design for testability** (DFT) refers to those design techniques that make test generation and test application cost-effective.

- DFT methods for digital circuits:
  - Ad-hoc methods
  - Structured methods:
    - Scan
    - Partial Scan
    - Built-in self-test (BIST)
    - Boundary scan
DFT

- In general, DFT deals with ways for improving controllability and observability.
- Costs associated with DFT:
  - Pins
  - Area/Yield
  - Performance
  - Design time
Objections to DFT

- Short-sighted view of management (schedule and costs)
- Life-cycle cost ignored by development management/contractors/buyers
- Area/functionality/performance myths
- Lack of knowledge by design engineers
- Testing is someone else’s problem
- Lack of tools to support DFT (this is improving….)
Ad-Hoc DFT Methods

- Good design practices learnt through experience are used as guidelines:
  - Avoid asynchronous (unclocked) feedback. \( \Rightarrow \) oscillation
  - Make flip-flops initializable. (clear or preset)
  - Avoid redundant gates. Avoid large fan-in gates.
    (controllability, observability)
  - Provide test control for difficult-to-control signals.
  - Avoid gated clocks.
  - Consider ATE requirements (tristates, etc.)

- Design reviews conducted by experts or design auditing tools.
Disadvantages of Ad-Hoc DFT Methods

- Circuits are too large for manual inspection.
- Experts and tools not always available.
- Test generation is often manual with no guarantee of high fault coverage.
- Design iterations may be necessary.

→ Use of ad-hoc DFT is usually discouraged for large circuits.
Ad Hoc Design Rules

- Partitioning
- Insert control/observation points (*test points*), e.g., a reset line for initialization
- Avoid redundancy
- Improve circuit structure, e.g., break global feedback during testing
- Provide clock access during testing
Partitioning (Divide and Conquer)

- Physically divide the system into multiple chips or boards.
- On board-level systems, use jumper wires to divide subunits.
- Has major performance penalties.
Partitioning using Degating

- Degating: another technique for separating modules on chip/board with lower performance penalties.

![Diagram of clock degating and module partitioning](image-url)
Test Point Insertion

- Make hard-to-control internal signals controllable via extra primary inputs and logic (CP = control points)
- Make hard-to-observe internal signals observable via extra primary outputs and logic (OP = observation points)
Test Point Insertion (cont’d)

Control-Point Sites

- Clock lines
- Global reset lines
- Inputs of state-control devices
- Lines of high fan-out (fan-out stems)
- Control (especially tristate control) lines of buses
- All bus lines in bus-structured designs
- Control inputs to (embedded) RAMS and ROMs
- Some enable/hold/select control lines
- Lines identified by testability measuring programs as having low controllability
Test Point Insertion (cont’d)

Observation-Point Sites

- “Buried” (not directly accessible) control/status lines
- Outputs of state-control devices
- Lines of high fan-out (fan-out stems)
- Outputs of high fan-in circuits, e.g. parity generators
- Logically redundant nodes
- Global feedback paths
- Output lines in bus-structured designs
- Lines identified by testability measuring programs as having low observability

Main Limitation

- Availability of (spare) input/output terminals
  - Add MUX’s to reduce number of I/O pins
  - Serially shifts control point values
    - Long testing time
Test Point Insertion (cont’d)

Memory Control/Observation
DFT: Circuit Restructuring

- Example: Counter Design
DFT: Timing Control

- Avoid asynchronous timing
- Make clocks observable and controllable during testing
Design Rule Summary

- Partition large hard-to-test circuits into small testable components
- Design controllable (e.g. initializable) and observable units with careful selection of control/test points
- Allow global feedback paths to be opened/closed
- Avoid redundancy, or allow it to be overridden during testing
- Avoid asynchronous circuits, provide access to clock signals

Conclusions

- Often ad hoc design modification is too late to significantly improve a circuit's testability
- Develop a systematic test plan at the start of the design process
Scan Design

- Circuit is designed using pre-specified design rules.
- Test structure (hardware) is added to the verified design:
  - Add a test control (TC) primary input.
  - Replace flip-flops by scan flip-flops (SFF) and connect to form one or more shift registers in the test mode.
  - Make input/output of each scan shift register controllable/observable from PI/PO.
- Use combinational ATPG to obtain tests for all testable faults in the combinational logic.
- Add shift register tests and convert ATPG tests into scan sequences for use in manufacturing test.
Scan Design Rules

1. Use only clocked D-type of flip-flops for all state variables.
2. At least one PI pin (test control) must be available for test; more pins, if available, can be used.
3. All clocks must be controlled from PIs.
4. Clocks must not feed data inputs of flip-flops.
Correcting a Rule Violation

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

![Diagram of Scan Flip-Flop (SFF)]

- **Logic overhead**
- **MUX**
- **Master latch**
- **Slave latch**
- **D flip-flop**

**CK** (Clock)

- **Master open**
- **Slave open**

**TC** (Test Control)

- Normal mode, D selected
- Scan mode, SD selected

Sharif University of Technology

Testability: Lecture 23

Slide 29 of 43
Level-Sensitive Scan-Design Flip-Flop (LSSD-SFF)

D flip-flop

Master latch

Slave latch

MCK

SCK

SD

TCK

Logic overhead

Normal mode

Scan mode

Sharif University of Technology

Testability: Lecture 23
Adding Scan Structure

PI → Combinational logic → SFF → SCANOUT

PO

TC or TCK

Not shown: CK or MCK/SCK feed all SFFs.
Tests for Full-Scan Circuits

- Test generation for combinational logic only
- Separate the test vectors and response data, based on PI, PO and state (F) variables: \( t_i = t_i^I, t_i^F \quad i = 1, 2, n \)
  \[ r_i = r_i^O, r_i^F \]
- Test application:
  1. Scan-in \( t_i^F \) by setting the circuit in test mode
  2. Apply \( t_i^I \)
  3. Observe \( r_i^O \)
  4. Set the circuit in functional mode and capture the response \( r_i^F \) into scan register
  5. Scan-out \( r_i^F \) while scanning-in \( t_{i+1}^F \) by setting the circuit in test mode
  6. \( i \leftarrow i + 1. \) Go to 2
Combinational Test Vectors
Combinational Test Vectors (cont’d)

Sequence length = \((n_{\text{comb}} + 1) \times n_{\text{sff}} + n_{\text{comb}}\) clock periods

- \(n_{\text{comb}} =\) number of combinational vectors
- \(n_{\text{sff}} =\) number of scan flip-flops
Testing Steps

- Scan register must be tested prior to application of scan test sequences.
  - A shift sequence 00110011 . . . of length \( n_{sff} + 4 \) in scan mode (TC=0) produces 00, 01, 11 and 10 transitions in all flip-flops and observes the result at SCANOUT output.
    - Covers most, if not all, single stuck-at faults in FFs, and
    - verifies the correctness of the shift operation of the scan register.
- Add \( n_{sff} + 4 \) to sequence length obtained in previous slide;
  - Total scan test length: \((n_{comb} + 2) n_{sff} + n_{comb} + 4 \) clock periods.
- Example: 2,000 scan flip-flops, 500 comb. vectors, total scan test length \( \sim 10^6 \) clocks.
- Multiple scan registers reduce test length.
Multiple Scan Registers

- Multiple scan registers reduce test length.
- Scan flip-flops can be distributed among any number of shift registers, each having a separate *scanin* and *scanout* pin.
- Sequence length is determined by the longest scan shift register.
- One *test control (TC)* pin is essential (scanin and scanout for each chain can share PI and PO pins, respectively).

![Diagram of Multiple Scan Registers]
Scan Design Advantages/Disadvantages

- **Advantages:**
  1. By adding controllability/observability to the state variables, scan design also eases functional testing.
  2. The testing problem is transformed from one of sequential circuit testing to one of combinational circuit testing.

- **Disadvantages:**
  1. IO pins: One pin necessary.
  2. Additional area for latches/FFs (area overhead)

    - *Gate overhead* = \( \left[ \frac{4 \times n_{f_{FF}}}{n_g + 10 \times n_{FF}} \right] \times 100\% \), where \( n_g = \text{comb. gates}, \ n_{FF} = \text{flip-flops} \)

    Example: \( n_g = 100k \text{ gates}, \ n_{FF} = 2k \text{ flip-flops} \), overhead = 6.7%.

    - More accurate estimate must consider scan wiring and layout area.
Scan Design Advantages/Disadvantages

3. Additional time required to latch the next state into the registerers (speed overhead)
   - Multiplexer delay added in combinational path; approx. two gate-delays.
   - Flip-flop output loading due to one additional fan-out; approx. 5-6%.

4. Additional time required to scan in/out test vectors and responses (testing overhead)

5. Clock generation and distribution is more difficult.
ATPG Example: S5378

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Full-scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of combinational gates</td>
<td>2,781</td>
<td>2,781</td>
</tr>
<tr>
<td>Number of non-scan flip-flops (10 gates each)</td>
<td>179</td>
<td>0</td>
</tr>
<tr>
<td>Number of scan flip-flops (14 gates each)</td>
<td>0</td>
<td>179</td>
</tr>
<tr>
<td>Gate overhead</td>
<td>0.0%</td>
<td>15.66%</td>
</tr>
<tr>
<td>Number of faults</td>
<td>4,603</td>
<td>4,603</td>
</tr>
<tr>
<td>PI/PO for ATPG</td>
<td>35/49</td>
<td>214/228</td>
</tr>
<tr>
<td>Fault coverage</td>
<td>70.0%</td>
<td>99.1%</td>
</tr>
<tr>
<td>Fault efficiency</td>
<td>70.9%</td>
<td>100.0%</td>
</tr>
<tr>
<td>CPU time on SUN Ultra II, 200MHz processor</td>
<td>5,533 s</td>
<td>5 s</td>
</tr>
<tr>
<td>Number of ATPG vectors</td>
<td>414</td>
<td>585</td>
</tr>
<tr>
<td>Scan sequence length</td>
<td>414</td>
<td>105,662</td>
</tr>
</tbody>
</table>
Automated Scan Design

- Rule violations
- Scan design rule audits
- Combinational ATPG
- Combinational vectors
- Scan sequence and test program generation
- Scan chain order
- Design and test data for manufacturing
- Test program
- Gate-level netlist
- Behavior, RTL, and logic Design and verification
- Scan hardware insertion
- Chip layout: Scan-chain optimization, timing verification
- Scan netlist
- Mask data
- Design and test data for manufacturing
Timing and Power

- Small delays in scan path and clock skew can cause race condition.
- Large delays in scan path require slower scan clock.
- Dynamic multiplexers: Skew between TC and $\overline{TC}$ signals can cause momentary shorting of D and SD inputs.
- Random signal activity in combinational circuit during scan can cause excessive power dissipation.
Summary

- Scan is the most popular DFT technique:
  - Rule-based design
  - Automated DFT hardware insertion
  - Combinational ATPG

- Advantages:
  - Design automation
  - High fault coverage; helpful in diagnosis
  - Hierarchical: scan-testable modules are easily combined into large scan-testable systems
  - Moderate area (~10%) and speed (~5%) overheads
Summary

- Disadvantages:
  - Area overhead
    - Due to larger flip-flops
    - Due to extra routing
  - Possible performance degradation
    - Extra gate delay due to the multiplexer
    - Extra capacitive loading delay due to scan wiring at the flip-flop output
  - Large test data volume and long test time
  - Basically a slow speed (DC) test
  - Not applicable to all designs (e.g. asynchronous designs, designs violating scan design rules)
  - High power dissipation during testing