Lecture 24

Design for Testability (DFT): Partial-Scan & Scan Variations

- Definition
- Partial-scan architecture
- Historical background
- Cyclic and acyclic structures
- Partial-scan by cycle-breaking
  - S-graph and MFVS problem
  - Test generation and test statistics
  - Partial vs. full scan
  - Partial-scan flip-flop
- Random-access scan (RAS)
- Scan-hold flip-flop (SHFF)
- Summary

Original slides copyright by Mike Bushnell and Vishwani Agrawal
Partial-Scan Definition

- A subset of flip-flops is scanned.

Objectives:
- Minimize area overhead and scan sequence length, yet achieve required fault coverage
- Exclude selected flip-flops from scan:
  - Improve performance
  - Allow limited scan design rule violations
- Allow automation:
  - In scan flip-flop selection
  - In test generation
- Shorter scan sequences
Partial-Scan Architecture

Combinational circuit

PI → CK1 → CK2 → TC → SCANIN

FF

FF

SFF

SFF

PO → SCANOUT
History of Partial-Scan

- Scan flip-flop selection from testability measures, Trischler et al., ITC-80; not too successful.
- Use of combinational ATPG:
  Agrawal et al., D&T, Apr. 88
    - Functional vectors for initial fault coverage
    - Scan flip-flops selected by ATPG
  Gupta et al., IEEETC, Apr. 90
    - Balanced structure
    - Sometimes requires high scan percentage
- Use of sequential ATPG:
  Cheng and Agrawal, IEEETC, Apr. 90; Kunzmann and Wunderlich, JETTA, May 90
  - Create cycle-free structure for efficient ATPG
Difficulties in Seq. ATPG

- Poor initializability.
- Poor controllability/observability of state variables.
- Gate count, number of flip-flops, and sequential depth do not explain the problem.
- Cycles are mainly responsible for complexity.

An ATPG experiment:

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of gates</th>
<th>Number of flip-flops</th>
<th>Sequential depth</th>
<th>ATPG CPU s</th>
<th>Fault coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLC</td>
<td>355</td>
<td>21</td>
<td>14*</td>
<td>1,247</td>
<td>89.01%</td>
</tr>
<tr>
<td>Chip A</td>
<td>1,112</td>
<td>39</td>
<td>14</td>
<td>269</td>
<td>98.80%</td>
</tr>
</tbody>
</table>

* Maximum number of flip-flops on a PI to PO path
## Benchmark Circuits

<table>
<thead>
<tr>
<th>Circuit</th>
<th>s1196</th>
<th>s1238</th>
<th>s1488</th>
<th>s1494</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>14</td>
<td>14</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PO</td>
<td>14</td>
<td>14</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>FF</td>
<td>18</td>
<td>18</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Gates</td>
<td>529</td>
<td>508</td>
<td>653</td>
<td>647</td>
</tr>
<tr>
<td>Structure</td>
<td>Cycle-free</td>
<td>Cycle-free</td>
<td>Cyclic</td>
<td>Cyclic</td>
</tr>
<tr>
<td>Sequential depth</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total faults</td>
<td>1242</td>
<td>1355</td>
<td>1486</td>
<td>1506</td>
</tr>
<tr>
<td>Detected faults</td>
<td>1239</td>
<td>1283</td>
<td>1384</td>
<td>1379</td>
</tr>
<tr>
<td>Potentially detected faults</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Untestable faults</td>
<td>3</td>
<td>72</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>Abandoned faults</td>
<td>0</td>
<td>0</td>
<td>76</td>
<td>97</td>
</tr>
<tr>
<td>Fault coverage (%)</td>
<td>99.8</td>
<td>94.7</td>
<td>93.1</td>
<td>91.6</td>
</tr>
<tr>
<td>Fault efficiency (%)</td>
<td>100.0</td>
<td>100.0</td>
<td>94.8</td>
<td>93.4</td>
</tr>
<tr>
<td>Max. sequence length</td>
<td>3</td>
<td>3</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>Total test vectors</td>
<td>313</td>
<td>308</td>
<td>525</td>
<td>559</td>
</tr>
<tr>
<td>Gentest CPU s (Sparc 2)</td>
<td>10</td>
<td>15</td>
<td>19941</td>
<td>19183</td>
</tr>
</tbody>
</table>
Cycle-Free Example

Circuit

Level = 1

F1

F2

2

F3

3

s - graph

Level = 1

F1

F2

2

F3

3
d_{seq} = 3

All faults are testable. See Example 8.6.
Theorem 8.1: A cycle-free circuit is always initializable. It is also initializable in the presence of any non-flip-flop fault.

Theorem 8.2: Any non-flip-flop fault in a cycle-free circuit can be detected by at most $d_{seq} + 1$ vectors.

ATPG complexity: To determine that a fault is untestable in a cyclic circuit, an ATPG program using nine-valued logic may have to analyze $9^{Nff}$ time-frames, where $Nff$ is the number of flip-flops in the circuit.
A Partial-Scan Method

- Select a minimal set of flip-flops for scan to eliminate all cycles.
- Alternatively, to keep the overhead low only long cycles may be eliminated.
- In some circuits with a large number of self-loops, all cycles other than self-loops may be eliminated.
The MFVS Problem

- For a directed graph find a set of vertices with smallest cardinality such that the deletion of this vertex-set makes the graph acyclic.

- The minimum feedback vertex set (MFVS) problem is NP-complete; practical solutions use heuristics.

- A secondary objective of minimizing the depth of acyclic graph is useful.

A 6-flip-flop circuit

s-graph
Test Generation

- Scan and non-scan flip-flops are controlled from separate clock PIs:
  - Normal mode – Both clocks active
  - Scan mode – Only scan clock active

- Seq. ATPG model:
  - Scan flip-flops replaced by PI and PO
  - Seq. ATPG program used for test generation
  - Scan register test sequence, 001100..., of length $n_{sff} + 4$ applied in the scan mode
  - Each ATPG vector is preceded by a scan-in sequence to set scan flip-flop states
  - A scan-out sequence is added at the end of each vector sequence

- Test length = $(n_{ATPG} + 2) n_{sff} + n_{ATPG} + 4$ clocks
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
Test Length Statistics

Circuit: TLC

- Without scan
- 9 scan flip-flops
- 10 scan flip-flops
## Partial vs. Full Scan: S5378

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Partial-scan</th>
<th>Full-scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of combinational gates</td>
<td>2,781</td>
<td>2,781</td>
<td>2,781</td>
</tr>
<tr>
<td>Number of non-scan flip-flops</td>
<td>179</td>
<td>149</td>
<td>0</td>
</tr>
<tr>
<td>(10 gates each)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of scan flip-flops</td>
<td>0</td>
<td>30</td>
<td>179</td>
</tr>
<tr>
<td>(14 gates each)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gate overhead</td>
<td>0.0%</td>
<td>2.63%</td>
<td>15.66%</td>
</tr>
<tr>
<td>Number of faults</td>
<td>4,603</td>
<td>4,603</td>
<td>4,603</td>
</tr>
<tr>
<td>PI/PO for ATPG</td>
<td>35/49</td>
<td>65/79</td>
<td>214/228</td>
</tr>
<tr>
<td>Fault coverage</td>
<td>70.0%</td>
<td>93.7%</td>
<td>99.1%</td>
</tr>
<tr>
<td>Fault efficiency</td>
<td>70.9%</td>
<td>99.5%</td>
<td>100.0%</td>
</tr>
<tr>
<td>CPU time on SUN Ultra II</td>
<td>5,533 s</td>
<td>727 s</td>
<td>5 s</td>
</tr>
<tr>
<td>200MHz processor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of ATPG vectors</td>
<td>414</td>
<td>1,117</td>
<td>585</td>
</tr>
<tr>
<td>Scan sequence length</td>
<td>414</td>
<td>34,691</td>
<td>105,662</td>
</tr>
</tbody>
</table>
Flip-flop for Partial Scan

- Normal scan flip-flop (SFF) with multiplexer of the LSSD flip-flop is used.
- Scan flip-flops require a separate clock control:
  - Either use a separate clock pin
  - Or use an alternative design for a single clock pin
Random-Access Scan (RAS)

Combinational logic

RAM $n_{ff}$ bits

Address decoder

Address scan register $\log_2 n_{ff}$ bits

PO

PI

CK

TC

SCANIN

ADDRESS

ACK

SCANOUT

Address scan register

$\log_2 n_{ff}$ bits
RAS Flip-Flop (RAM Cell)
RAS Applications

- Logic test: reduced test length.
- Delay test: Easy to generate single-input-change (SIC) delay tests.
- Advantage: RAS may be suitable for certain architecture, e.g., where memory is implemented as a RAM block.
- Disadvantages:
  - Not suitable for random logic architecture
  - High overhead – gates added to SFF, address decoder, address register, extra pins and routing
Scan-Hold Flip-Flop (SHFF)

- The control input HOLD keeps the output steady at previous state of flip-flop.

- Applications:
  - Reduce power dissipation during scan
  - Isolate asynchronous parts during scan test
  - Delay testing
Summary

- Partial-scan is a generalized scan method; scan can vary from 0 to 100%.
- Elimination of long cycles can improve testability via sequential ATPG.
- Elimination of all cycles and self-loops allows combinational ATPG.
- Partial-scan has lower overheads (area and delay) and reduced test length.
- Partial-scan allows limited violations of scan design rules, e.g., a flip-flop on a critical path may not be scanned.