Design for Testability

Outline

• Ad Hoc Design for Testability Techniques
  – Method of test points
  – Multiplexing and demultiplexing of test points
  – Time sharing of I/O for normal working and testing modes
  – Partitioning of registers and large combinational circuits

• Scan-Path Design
  – Scan-path design concept
  – Controllability and observability by means of scan-path
  – Full and partial serial scan-paths
  – Non-serial scan design
  – Classical scan designs
Ad Hoc Design for Testability Techniques

Method of Test Points:

Block 1 \rightarrow Block 2

Block 1 is not observable, Block 2 is not controllable

Improving controllability and observability:

1- controllability:
- CP = 0 - normal working mode
- CP = 1 - controlling Block 2 with signal 1

0- controllability:
- CP = 1 - normal working mode
- CP = 0 - controlling Block 2 with signal 0
Ad Hoc Design for Testability Techniques

Method of Test Points:

Block 1 is not observable,
Block 2 is not controllable

Improving controllability:

Normal working mode:
CP1 = 0, CP2 = 1
Controlling Block 2 with 1:
CP1 = 1, CP2 = 1
Controlling Block 2 with 0:
CP2 = 0

Normal working mode:
CP2 = 0
Controlling Block 2 with 1:
CP1 = 1, CP2 = 1
Controlling Block 2 with 0:
CP1 = 0, CP2 = 1
Ad Hoc Design for Testability Techniques

Multiplexing monitor points:

To reduce the number of output pins for observing monitor points, multiplexer can be used:

$2^n$ observation points are replaced by a single output and $n$ inputs to address a selected observation point

Disadvantage:

Only one observation point can be observed at a time

Advantage:

$$(n + 1) \ll 2^n$$

Number of additional pins: $(n + 1)$

Number of observable points: $[2^n]$
Multiplexing monitor points:

To reduce the number of output pins for observing monitor points, a multiplexer can be used:

To reduce the number of inputs, a counter (or a shift register) can be used to drive the address lines of the multiplexer.

Disadvantage:

Only one observation point can be observed at a time.

Number of additional pins: 2
Number of observable points: $2^n$

Advantage: $2 < n <= 2^n$
Demultiplexer for implementing control points:

To reduce the number of input pins for controlling testpoints, demultiplexer and a latch register can be used.

Disadvantage:

$N$ clock times are required between test vectors to set up the proper control values.

Number of additional pins: $(n + 1)$
Number of control points: $2^{n-1} < N \leq 2^n$

Advantage: $(n + 1) << N$
Demultiplexer for implementing control points:

To reduce the number of input pins for controlling testpoints, demultiplexer and a latch register can be used.

To reduce the number of inputs for addressing, a counter (or a shift register) can be used to drive the address lines of the demultiplexer

**Disadvantage:**

\( N \) clock times are required between test vectors to set up the proper control values.

**Advantage:** \( 2^{\log_2 N} \)
Time-sharing of outputs for monitoring

To reduce the number of output pins for observing monitor points, time-sharing of working outputs can be introduced: no additional outputs are needed.

To reduce the number of inputs, again counter or shift register can be used if needed.

Number of additional pins: 1
Number of control points: N

Advantage: $1 << N$
Time-sharing of inputs for controlling

To reduce the number of input pins for controlling test points, time-sharing of working inputs can be introduced.

To reduce the number of inputs for driving the address lines of demultiplexer, counter or shift register can be used if needed.

Normal input lines

DMUX

Number of additional pins: 1
Number of control points: N

Advantage: $1 \ll N$
Example: DFT with MUX-s and DMUX-s

Given a circuit:
- CP1 and CP2 are not controllable
- CP3 and CP4 are not observable

DFT task: Improve the testability by using a single control input, no additional inputs/outputs allowed
Example: DFT with MUX-s and DMUX-s

Given a circuit:
CP3 and CP4 are not observable
→ Improving the observability

<table>
<thead>
<tr>
<th>T</th>
<th>Mode</th>
<th>MUX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Norm.</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Test</td>
<td>1</td>
</tr>
</tbody>
</table>

Result: A single pin T is needed
Given a circuit: CP1 and CP2 are not controllable → Improving the controllability

Result: A single pin $T$ is needed
Example: DFT with MUX-s and DMUX-s

Result: A single pin $T$ is needed

<table>
<thead>
<tr>
<th>Q</th>
<th>Mode</th>
<th>DMUX</th>
<th>MUX 1</th>
<th>MUX 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Norm</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>Contr</td>
<td>0</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>010</td>
<td>Test</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>Obs</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>Obs</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>101</td>
<td>Obs</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
Ad Hoc Design for Testability Techniques

Examples of good candidates for control points:
- control, address, and data bus lines on bus-structured designs
- enable/hold inputs of microprocessors
- enable and read/write inputs to memory devices
- clock and preset/clear inputs to memory devices (flip-flops, counters, ...)
- data select inputs to multiplexers and demultiplexers
- control lines on tristate devices

Examples of good candidates for observation points:
- stem lines associated with signals having high fanout
- global feedback paths
- redundant signal lines
- outputs of logic devices having many inputs (multiplexers, parity generators)
- outputs from state devices (flip-flops, counters, shift registers)
- address, control and data busses
Fault redundancy and testability

\[ y = x_1 \lor (x_1 \lor x_2) x_4 \lor x_3 x_4 \]

\[ \frac{\partial y}{\partial x_2} \equiv 0 \quad \text{Faults at } x_2 \text{ not testable} \]

Redundant gates are removed:

Fault at \( x_{12} \) not testable

\[ y = x_1 \lor x_4 \lor x_3 x_4 = x_1 \lor x_4 \lor x_3 \]

Remaining gate
Ad Hoc Design for Testability Techniques

Logical redundancy:

Redundancy should be avoided:

- If a redundant fault occurs, it may invalidate some test for nonredundant faults
- Redundant faults cause difficulty in calculating fault coverage
- Much test generation time can be spent in trying to generate a test for a redundant fault

Redundancy intentionally added:

- To eliminate hazards in combinational circuits
- To achieve high reliability (using error detecting circuits)

Hazard control circuitry:

Redundant AND-gate
Fault \( \equiv 0 \) not testable

Additional control input added:

- \( T = 1 \) - normal working mode
- \( T = 0 \) - testing mode
Ad Hoc Design for Testability Techniques

Fault redundancy:

Error control circuitry:

Testable error control circuitry:

E = 1 if decoder is fault-free
Fault 1 not testable

Additional control input added:
T = 0 - normal working mode
T = 1 - testing mode

E = 1 if decoder is fault-free
Ad Hoc Design for Testability Techniques

Partitioning of registers (counters):

16 bit counter divided into two 8-bit counters:

Instead of $2^{16} = 65536$ clocks, $2 \times 2^8 = 512$ clocks needed

If tested in parallel, only 256 clocks needed
Ad Hoc Design for Testability Techniques

Partitioning of large combinational circuits:

The time complexity of test generation and fault simulation grows faster than a linear function of circuit size. Partioning of large circuits reduces these costs. I/O sharing of normal and testing modes is used. Three modes can be chosen: - normal mode - testing C1 - testing C2 (bolded lines)

How many additional inputs are needed?
Scan-Path Design

The complexity of testing is a function of the number of feedback loops and their length.

The longer a feedback loop, the more clock cycles are needed to initialize and sensitize patterns.

Scan-register is a register with both shift and parallel-load capability.

T = 0 - normal working mode
T = 1 - scan mode

Normal mode: flip-flops are connected to the combinational circuit.

Test mode: flip-flops are disconnected from the combinational circuit and connected to each other to form a shift register.
Scan-Path Design and Testability

Two possibilities for improving controllability/observability

- MUX
- DMUX
- SCAN IN
- SCAN OUT

Technical University Tallinn, ESTONIA
In parallel scan path flip-flops can be organized in more than one scan chain

Advantage: time ↓
Disadvantage: # pins ↑
In partial scan instead of full-scan, it may be advantageous to scan only some of the flip-flops.

Example: counter – even bits joined in the scan-register.
Partial Scan Path

Hierarchical test generation with Scan-Path:

Control Part:

Data Part:

Scan-In

Scan-Out

Bus
Testing with Minimal DFT

Hierarchical test generation with Scan-Path:

Control Part

Scan-In

Data Part

Scan-Out

Bus
In random access scan each flip-flop in a logic network is selected individually by an address for control and observation of its state

Example:
Delay fault testing
Selection of Test Points

Test point selection approaches

• Improving testability for any set of pseudo-random patterns (Pseudorandom BIST)
  – Testability measures are used to characterize the controllability and observability of the circuit (independently of the test applied)

• Improving testability for a given sequence of vectors (Functional BIST)
  – Fault simulation is used for measuring the fault coverage

Methods that are used:
  – logic simulation,
  – fault simulation,
  – evaluation (measuring) of controllability and observability
Random BIST vs Functional BIST

Random BIST

UUT

Test generator

Signature

Reference

Go/NoGo

HW overhead

Random test set

Traditional functional testing

UUT

Result

Reference

Go/NoGo

Normal operation

Deterministic functional test set

Functional BIST

UUT

Result

Signature

Reference

Go/NoGo

HW overhead

Random BIST vs Functional BIST

Random BIST

UUT

Test generator

Signature

Reference

Go/NoGo

HW overhead

Random test set

Traditional functional testing

UUT

Result

Reference

Go/NoGo

Normal operation

Deterministic functional test set

Functional BIST

UUT

Result

Signature

Reference

Go/NoGo

HW overhead

Improving Testability by Inserting CPs

Test sequence → Fault Simulation → Selection of CPs

Circuit modification

Fault coverage 100%

Circuit

Not detected faults
Functional BIST
Selection of Test Points

Method: Simulation of given test patterns

• Identification of the faults that are detected
• The remaining faults are classified as
  – A: Faults that were not excited
  – B: Faults at gate inputs that were excited but not propagated to the gate output
  – C: Faults that were excited but not propagated to circuit output
• The faults A and B require control points for their detection
• The faults C may be detected by either by observation points or by control points
• Control points selection should be carried out before observation points selection
Classification of Not-Detected Faults

Classes A and B need controllability

Class A:
Fault $x_3 \equiv 1$ is not activated

Class B:
Faults at $x_5$ are not propagated through the gate

Class C:
Faults at $x_1$ are not propagated to the output

Class C needs either controllability or observability

Always 1
## Selection of Test Points

**Classification of faults**

### Given test:

<table>
<thead>
<tr>
<th>No</th>
<th>Test patterns</th>
<th>Fault table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inputs</td>
<td>Intern. points</td>
</tr>
<tr>
<td>1</td>
<td>1 2 3 4 5</td>
<td>a b c</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

### Not detected faults:

<table>
<thead>
<tr>
<th>Class</th>
<th>Faults</th>
<th>Missing signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$x_1/0$: $x_1 = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$b/0$: $b = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$x_3/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$a/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$x_2/0$: $x_1x_2 = 01$ OK</td>
<td></td>
</tr>
</tbody>
</table>
## Selection of Test Points

### Classification of faults

**Given test:**

<table>
<thead>
<tr>
<th>No</th>
<th>Test patterns</th>
<th>Fault table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inputs</td>
<td>Intern. points</td>
</tr>
<tr>
<td>1</td>
<td>1 2 3 4 5</td>
<td>a b c</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

**Not detected faults:**

<table>
<thead>
<tr>
<th>Class</th>
<th>Faults</th>
<th>Missing signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$x_1/0$: $x_1 = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$b/0$: $b = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$x_3/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$a/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$x_2/0$: $x_1x_2 = 01$ OK</td>
<td></td>
</tr>
</tbody>
</table>
### Selection of Test Points

#### Classification of faults

**Given test:**

<table>
<thead>
<tr>
<th>No</th>
<th>Test patterns</th>
<th>Fault table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inputs</td>
<td>Intern. points</td>
</tr>
<tr>
<td>1</td>
<td>1 0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 1 0</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

Not detected faults:

<table>
<thead>
<tr>
<th>Class</th>
<th>Faults</th>
<th>Missing signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$x_1/0$: $x_1 = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$b/0$: $b = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$x_3/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$a/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$x_2/0$: $x_1 x_2 = 01$ OK</td>
<td></td>
</tr>
</tbody>
</table>
Selection of Test Points

Classification of faults

Given test:

<table>
<thead>
<tr>
<th>No</th>
<th>Test patterns</th>
<th>Fault table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inputs</td>
<td>Intern. points</td>
</tr>
<tr>
<td>1</td>
<td>1 2 3 4 5</td>
<td>a b c</td>
</tr>
<tr>
<td>2</td>
<td>0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

Not detected faults:

Class | Faults | Missing signals
A | $x_1/0$: $x_1 = 1$ | is missing
B | $x_3/0$: $x_3 = 11$ | is missing
C | $x_2/0$: $x_1x_2 = 01$ | OK
Selection of Test Points

Classification of faults

Given test:

<table>
<thead>
<tr>
<th>No</th>
<th>Test patterns</th>
<th>Fault table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inputs</td>
<td>Intern. points</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5</td>
<td>a b c</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

Not detected faults:

Class | Faults | Missing signals |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$x_1/0$: $x_1 = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$b/0$: $b = 1$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$x_3/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$a/0$: $x_3 a = 11$ is missing</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$x_2/0$: $x_1 x_2 = 01$ OK, but path activation is missing</td>
<td></td>
</tr>
</tbody>
</table>
**Selection of Test Points: Procedure**

1. **Selection of control points:**
   - Once *control point candidates* are identified for the faults A and B, a *minimum* number of control points (CP) can be identified.
   - This can be formulated as a *minimum coverage problem* where a minimum CPs are selected such that at least one CP candidate is included for each fault in A and B.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CP3</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CP5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

*Faults, not detected*

*Selected control points*
Selection of Test Points: Procedure

1. Selection of control points:

Control point candidates

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Faults, not detected

Selected control points
Selection of Test Points: Procedure

2. Selection of observation points

- Once the CPs are selected, the given test patterns are augmented to accommodate the additional inputs associated with the CPs and fault simulation is performed
- The fault class C is updated
- For each fault, in C the circuit lines to which the effect of the fault propagates, are identified as a potential observation point candidates
- A minimum covering problem is formulated and solved to find the observation points to be added

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CP3</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CP5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Minimization of control points
Selection of Test Points: Procedure

1. Selection of observation points:

Control point candidates

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Faults, not detected

Selected control points

Technical University Tallinn, ESTONIA
Selection of Test Points

Minimization of test points:
Not detected faults:
Class A: \(x_1/0, b/0\)
Class B: \(x_3/0, a/0,\)

Test point coverage:

<table>
<thead>
<tr>
<th>Potential control points</th>
<th>Not detected faults</th>
<th>To be selected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(x_1=1)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>(x_3=1)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>(a=1)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>(b=1)</td>
<td>+</td>
</tr>
</tbody>
</table>

Not detected faults:

<table>
<thead>
<tr>
<th></th>
<th>(x_1/0)</th>
<th>(x_3/0)</th>
<th>(a/0)</th>
<th>(b/0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

Test patterns:

<table>
<thead>
<tr>
<th>No</th>
<th>Inputs</th>
<th>Intern. points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

Technical University Tallinn, ESTONIA
Insertion of Test Points

Test point for $x_1/0$

All faults detected:
Class A: $x_1/0$, $b/0$
Class B: $x_3/0$, $a/0$,

Corrected circuit:
Insertion of Test Points

Two test points:

Selected test points:

Class A: \( x_1/0 \rightarrow x_1=1 \) (control point)

Class C: \( x_2/0 \) (observable point)

Test patterns:

<table>
<thead>
<tr>
<th>No</th>
<th>Inputs</th>
<th>Intern. points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 1 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 1 0 1</td>
<td>1 0 0 0</td>
</tr>
</tbody>
</table>

Corrected circuit:

To be observed

Technical University Tallinn, ESTONIA
Minimization of monitoring points:

To reduce the number of output pins for observing monitor points, exor gates can be used:
Selection of Test Points

Minimization of monitoring points:

To reduce the number of output pins for observing monitor points, signature analyzers can be used:

- **With SA - Accurate**
- **With EXOR – Not accurate**

**Space and time compaction**

**Space**

**With MUX**

**Time**

**Additional outputs**

**Out**

**Scan out**

**Scan in**

**MUX**

**Counter**

**SA**

**C**
Boundary Scan Architecture

TDI – Test Data IN
TMS – Test Mode Select
TCK – Test Clock
TDO – Test Data OUT
TAP – Test Access Port
Boundary Scan Architecture

- TDI
- TDO
- Boundary Scan Registers
- Internal logic
- Device ID. Register
- Bypass Register
- Instruction Register (IR)
- Data Registers
Boundary Scan Cell

Used at the input or output pins
Boundary Scan Working Modes

**SAMPLE mode:**

Get snapshot of normal chip output signals (*monitoring mode*)

---

Sample diagram showing the flow of data with labels such as `ShiftDR`, `SO`, `Mode`, `D Q`, `CLK`, `SI`, `UpdateDR`, and `Component boundaries`. The diagram highlights data transfer paths activated in SAMPLE mode.
Boundary Scan Working Modes

**PRELOAD mode:**

*Put data on boundary scan chain before next instruction*

---

**INPUT CELL**

- **ShiftDR**
- **SO**
- **Mode**
- **D Q**
- **CLK**
- **SI**
- **ClockDR**
- **UpdateDR**

**OUTPUT CELL**

- **ShiftDR**
- **SO**
- **Mode**
- **D Q**
- **CLK**

---

**On-chip System Logic**

---

**Component boundaries**

**Data transfer paths activated in PRELOAD mode.**
Boundary Scan Working Modes

Extest instruction:

Test off-chip circuits and board-level interconnections

Data transfer paths activated in EXTEST mode.
Boundary Scan Working Modes

**INTEST instruction**

*Feeds external test patterns in and shifts responses out*

![Diagram of Boundary Scan Working Modes](image-url)
Boundary Scan Working Modes

Bypass instruction:

Bypasses the corresponding chip using 1-bit register
Boundary Scan Working Modes

**IDCODE instruction:**

Connects the component device identification register serially between TDI and TDO in the Shift-DR TAP controller state.

Allows board-level test controller or external tester to read out component ID.

**Required** whenever a JEDEC identification register is included in the design.

- **Version:** 4-bits, Any format
- **Part Number:** 16-bits, Any format
- **Manufacturer ID:** 11-bits, Coded form of JEDEC
Fault Detection with Boundary Scan

Assume stuck-at-0
Assume wired AND
Open
Short
Kautz showed in 1974 that a sufficient condition to detect any pair of short circuited nets was that the “horizontal” codes must be unique for all nets. Therefore the test length is $\lceil \log_2(N) \rceil$.
Any Fault Detection with Boundary Scan

Assume stuck-
at-0

Assume wired AND

Open

Short

Suspected wired AND short

Suspected open fault SAF/0

Ambiguity

All 0-s and all 1-s are forbidden codes because of stuck-at faults. Therefore the final test length is \( \lceil \log_2 (N+2) \rceil \)
Fault Diagnosis with Boundary Scan

To improve the diagnostic resolution we have to add one bit more.
Synthesis of Testable Circuits

\[ y = x_1 x_3 \lor x_1 x_2 \]

Test generation:

\[ y = x_1 x_3 \lor x_1 x_2 \]

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4 test patterns are needed
Synthesis of Testable Circuits

Two implementations for the same circuit:

\[ y = x_1 x_3 \lor x_1 x_2 \]

Here:
4 test patterns are needed

Here:
Only 3 test patterns are needed
Synthesis of Testable Circuits

Given: \[ y = x_1 x_3 \lor x_1 x_2 \]

\[ y = c_0 \oplus c_1 x_3 \oplus c_2 x_2 \oplus c_3 x_2 x_3 \oplus c_4 x_1 \oplus c_5 x_1 x_3 \oplus c_6 x_1 x_2 \oplus c_7 x_1 x_2 x_3 \]

Calculation of constants:

<table>
<thead>
<tr>
<th>( f_i )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( y )</th>
<th>( \Sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_0 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

New:

\[ y = 1 \oplus x_3 \oplus x_1 \oplus x_1 x_3 \oplus x_1 x_2 \]
Synthesis of Testable Circuits

Test generation method:

\[ y = 1 \oplus x_3 \oplus x_1 \oplus x_1 x_3 \oplus x_1 x_2 \]

Roles of test patterns:

\[
\begin{array}{ccc}
X_1 & X_2 & X_3 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]
Amusing testability:

**Theorem:** You can test an arbitrary digital system by only 3 test patterns if you design it appropriately.

**Proof:**

```
011 & 001 → 011 & 001 & ?
101
```

**Solution:** System → FSM → Scan-Path → CC → NAND