Basics of board-level testing and IEEE1149.x Boundary Scan standard

Artur Jutman
artur@ati.ttu.ee
TU Tallinn, ESTONIA

February 2016
http://www.pld.ttu.ee/~artur/labs/
IC Level Test (SoC as a system)
- Final test is often a go/no-go test (diagnosis for process control only)
- DFT is usually a must: scan test, compression, BIST, etc.

Board Level Test (Board as a system)
- A myriad of test, measurement, and inspection approaches available
- DFT is very beneficial: test points, boundary scan, BIST, etc.

Product Level Test (Board as a part of system)
- Qualification test before product shipping (functional + application)
- Board-Level DFT is used if any
- Diagnosis is essential to narrow down the failing replaceable module

In-Field Test (working conditions are part of the test setup)
- Product operates in real-life conditions
- Power-On Self-Test (POST), monitoring and self-check routines
Outline

• Board level test challenges

• Fault modeling at board level (digital)

• Test generation for interconnect faults

• IEEE 1149.1 Boundary Scan Standard

• Application of Boundary Scan
Board Testability: in the past

PCBA

Simple ICs described by functional/truth tables

Sufficient number of test points
Problem of Test Access

Limited access (nail probing) for test, measurement, diagnostics
Board Testability: today’s challenges

PCBA

Black Hole 1

Black Hole 2

Black Hole 3

Defects might be hiding inside (behind the horizon)

Defects might be spread among the chips (mismatch)

Timing defects on the board not covered with functional test

No failure found?
What are the reasons for NFF?

FP7 BASTION project industrial survey results

- Insufficient coverage
- No test for certain faults
- Missing fault model
- Missing test method
- Intermittent faults
- Ageing
- Lack of communication

Source: ASTER Technologies
Is the NFF problem Important?

NFF is a critical subject for the industry!
The Expensive side of NFF

- 70% of all product returns characterized as NFF (US, 2007)
- NFF amounted up to 50% of 13.8 billion USD

Source: Accenture

$13.8 Billion
2007
## Main Categories of Board Test Today

<table>
<thead>
<tr>
<th>Test Domain</th>
<th>Typical Test Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inspection</strong></td>
<td><em>Pre-reflow:</em> Solder Paste Inspection (SPI), Automated Optical Inspection (AOI)</td>
</tr>
<tr>
<td></td>
<td><em>Post-reflow:</em> Visual inspection, Automated X-ray Inspection (AXI), AOI</td>
</tr>
<tr>
<td><strong>Electrical test</strong></td>
<td>In-Circuit Test (ICT), Manufacturing Defect Analysis (MDA), Flying Probe Test (FPT)</td>
</tr>
<tr>
<td><strong>Scan test</strong></td>
<td>Boundary Scan (BS/JTAG) and other test techniques based on IEEE 1149.1 and related standards</td>
</tr>
<tr>
<td><strong>High-speed test &amp; measurement</strong></td>
<td><strong>Processor-centric</strong> automated test solutions, <strong>FPGA-centric</strong> automated test solutions, Bit-Error Rate Test (BERT)</td>
</tr>
<tr>
<td><strong>Embedded instrumentation</strong></td>
<td>BIST instrumentation (fixed hardware), Synthetic instrumentation on FPGA (flexible hardware)</td>
</tr>
<tr>
<td><strong>Functional test</strong></td>
<td>Test of interfaces and basic behavior, test of main functions (fit-for-function test)</td>
</tr>
</tbody>
</table>

Board Test is a compound of many different techniques
Typical Order of Test Phases

- Visual inspection
- Optical/x-ray inspection
- Smoke test ;-)
- Power distribution test
- Structural test
  - In-Circuit Test (ICT), Flying Probe Test (FPT)
  - JTAG/Boundary Scan (BS)
  - Test Processors/Cores (BIST)
- Functional test (FT)
Classical board test flow

Inspection & Measurement → ICT & Flying Probe → Boundary Scan / JTAG → Functional Test

Pass → Pass → Pass

Bad Boards → IC NFF is here → bad ICs → BOARD-LEVEL NFFs are here

Repair & Retest

At-Speed test: enables detection of TIMING FAULTS

Static test
At-speed test

Yield

Unknown Bad Boards

Good Boards

Test Escapes

Products Shipped
Advanced state-of-the-art board test flow

Various static test phases

High-speed, at-speed structural test

Embedded Instrumentation

Pass

Bad Boards

At-speed structural test

Processor-centric test

Pass

Functional Test

Pass

Good Boards

Unknown Bad Boards

Yield

Test Escapes

Repair & Retest

Static test

At-speed test

Products Shipped

13
Test as early as possible!
Board Test Domain Comparison

**Structural Test**
- Provides measurable quality (defect coverage)
- Good for diagnosis and troubleshooting
- Provides feedback for process tuning
- Often intrusive
- Often do not cover dynamic faults

**Functional Test**
- Covers dynamic faults
- Fit-for-function proof
- Non-intrusive
- Black box testing
- Pass/fail result only
- Often unknown defect coverage (test quality)
- No feedback possible
Definition:

- **Black-box testing** (based on functional information only)
- **Test through functional inputs** (no DFT)

*Test Automation is crucial*
Definition:

- Test based on structural models:
  - device/system;
  - defects/faults;
  - properties/features.

- DFT-based test

Test Automation is normal
Closing the Timing Fault coverage gap

**Structural Test**
- ATPGs
- Automated diagnosis
- Known fault coverage

**Functional Test**
- Time consuming
- Manual work (expertise based)
- Studying field returns
- Coverage often unknown
- $$$

Test Escape Zone

Timing Faults

Structural Test
Towards a Dream Test Method

- Non-intrusive
- Should cover dynamic faults
- Should provide measurable quality (defect coverage)
- Good for diagnosis and troubleshooting
- Should provide feedback for process tuning
- Should provide fit-for-function proof
<table>
<thead>
<tr>
<th></th>
<th>ICT</th>
<th>Flying Probe</th>
<th>Boundary Scan</th>
<th>Processor-Centric Test</th>
<th>FPGA-Centric Test</th>
<th>Functional Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DUT access</strong></td>
<td>Fixed nails</td>
<td>“Flying” nails</td>
<td>Scan cells</td>
<td>μP core</td>
<td>FPGA</td>
<td>ATE / μP / FPGA</td>
</tr>
<tr>
<td><strong>Test access</strong></td>
<td>Low</td>
<td>Low</td>
<td>Digital</td>
<td>Digital</td>
<td>Digital</td>
<td>High</td>
</tr>
<tr>
<td><strong>Invasiveness</strong></td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Structural Fault Coverage</strong></td>
<td>Analog/Digital</td>
<td>Analog/Digital</td>
<td>Digital</td>
<td>Digital</td>
<td>Digital</td>
<td>Un-countable</td>
</tr>
<tr>
<td><strong>Timing Related Fault Coverage</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>High</td>
<td>High</td>
<td>Un-countable</td>
</tr>
<tr>
<td><strong>Test implementation cost</strong></td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>Test automation</strong></td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Test scalability</strong></td>
<td>Limited</td>
<td>Limited</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Qualification test</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Outline

• Board level test challenges

• Fault modeling at board level (digital)

• Test generation for interconnect faults

• IEEE 1149.1 Boundary Scan Standard

• Application of Boundary Scan
Combining Test Coverage from all different test methods

Common coverage metrics

PCOLA/
SOQ/FAM;
PPVS(F);
MPS(F)
## Test coverage strategies and corresponding metrics

<table>
<thead>
<tr>
<th>Approach to fault modeling</th>
<th>Level of Abstraction</th>
<th>Examples of Defects</th>
<th>Test Coverage Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Targeting defects in material and defects caused by assembly process</td>
<td>Structural faults at physical level</td>
<td>Bad soldering, lifted/bent leads, bad component, misalignment, tombstone, etc.</td>
<td>PPVS, MPS, PCOLA/SOQ/FAM</td>
</tr>
<tr>
<td>Targeting pin-level and net-level defects</td>
<td>Structural and behavioral faults at logic level</td>
<td>Opens, shorts, bad driver (pin logic / buffer)</td>
<td>stuck-at for opens; zero, one and net dominance for shorts; stuck-driving and -not driving for pins</td>
</tr>
<tr>
<td>Functional problems caused by defects</td>
<td>System level malfunction (behavioral)</td>
<td>Booting failure, unstable operation, wrong behavior</td>
<td>Functional model based test coverage metrics</td>
</tr>
<tr>
<td>Performance-related faults mainly at interconnect lines, buses, interfaces, communication links</td>
<td>Mainly statistical (error rates); <em>structural approaches are missing but needed</em></td>
<td>High error rate (slow performance), crosstalk, jitter, delay fault</td>
<td>Bit error rates at communication links, but <em>no universal industry-wide structural fault coverage metric</em></td>
</tr>
</tbody>
</table>
Adequate coverage metrics is needed to:

- estimate test quality
- diagnose defects (and parametric mismatch),
- get feedback for test coverage improvement
- get feedback for manufacturing processes tuning

Available de-facto standards in board/system test:

- MPS - Philips Research
- PPVS - ASTER Technologies
- PCOLA/SOQ - Agilent Technologies
- PCOLA/SOQ/FAM - iNEMI

Production defects and material quality perspective
## Defect Categories: Correlation & Ambiguity

<table>
<thead>
<tr>
<th>MPS(F)</th>
<th>PPVS(F)</th>
<th>PCOLA/SOQ/FAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>Value</td>
<td>Correct</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Live</td>
</tr>
<tr>
<td>Placement</td>
<td>Polarity</td>
<td>Orientation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alignment</td>
</tr>
<tr>
<td>Presence</td>
<td>Presence</td>
<td>Presence</td>
</tr>
<tr>
<td>Solder</td>
<td>Solder</td>
<td>Short</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quality</td>
</tr>
<tr>
<td>Function</td>
<td>Function</td>
<td>Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At-Speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Measurement</td>
</tr>
</tbody>
</table>

Table from: ASTER Technologies

- **Fault models not needed as defects can be directly tested**
- **Detailed fault models are used in practice**
- **Ambiguous Category: Fault Models for TRFs need to be used**
Main difference between logic circuits and board-level systems is the way the components are connected.

Typical board-level interconnect uses tri-state logic: logic-0, logic-1, and “high impedance” (switched off) state. Common notation: 0,1,Z.

There are special “enable” signals that control this additional state of the I/O pins.
Tri-State Connections

- Nets with several drivers
- Nets with bi-directional pins
Tri-State Net Example
Typical board-level faults from the logic misbehavior perspective

Net opens
- stuck-at fault (0 or 1)
- delay fault

Net shorts
- zero dominance
  - wired AND (mutual 0-dominance)
- one dominance
  - wired OR (mutual 1-dominance)
- net dominance
  - strong driver fault

Driver faults
- stuck-driving fault
- stuck-not-driving fault
- stuck-at fault
The Big Picture

<table>
<thead>
<tr>
<th>Errors Observed</th>
<th>Fault Models</th>
<th>Actual Defects</th>
<th>Defect Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong Appearance</td>
<td>Faults are typically not abstracted from defects in this category (but some defects could result in opens, shorts, delays)</td>
<td>Bad Part, Dead Part, Wrong Part</td>
<td>Correct, Live, Orientation, Alignment, Presence, Shorts, Quality, Feature, Speed, Measurement</td>
</tr>
<tr>
<td>Wrong Response</td>
<td></td>
<td></td>
<td>Material, Placement, Solder</td>
</tr>
<tr>
<td>Missing</td>
<td></td>
<td></td>
<td>Function</td>
</tr>
</tbody>
</table>

- **C**: Correct
- **L**: Live
- **O**: Orientation
- **A**: Alignment
- **P**: Presence
- **S**: Shorts
- **O**: Opens
- **Q**: Quality
- **F**: Feature
- **A**: At Speed
- **M**: Measurement
### Timing Faults / Delay Faults

- **Extending the “A” in PCOLA/SOQ/FAM**

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Dynamic behavior</th>
<th>Static domain equivalent</th>
<th>IC-level fault equivalent</th>
<th>Test patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin-level</td>
<td>Pin delay fault</td>
<td>Slow to rise, Slow to fall</td>
<td>Stuck-at 1, Stuck-at 0</td>
<td>Transition delay faults</td>
</tr>
<tr>
<td>Connection-level</td>
<td>Pin-to-pin delay fault</td>
<td>Slow to rise, Slow to fall</td>
<td>N/A</td>
<td>Path-delay fault</td>
</tr>
<tr>
<td>Net-level</td>
<td>Crosstalk errors</td>
<td>Glitch, Delay</td>
<td>Short between nets</td>
<td>Crosstalk</td>
</tr>
</tbody>
</table>
Possible shorts: bond wire, leg, solder, interconnect

*Shorts are usually modeled as wired-AND, wired-OR faults*
Open Faults

Misplaced bond wire

Misplaced component

Possible opens: bond wire, leg, solder, interconnect

Opens usually behave like stuck-at or delay faults
Outline

- Board level test challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan
Consider a simple example...

... two circuits, 4 wires, plain topology

Test Vector or Parallel Test Vector = PTV

010011
010110
011001
011100

Code Word or Serial Test Vector = STV

011100

Driver

Receiver

010011
010110
011001
011100
Test generation for interconnect faults

... two circuits, 4 wires, plain topology

Opens usually behave like stuck-at or delay faults.
Shorts are usually modeled as wired-AND or wired-OR.

How many test vectors are enough to cover all possible shorts?
Kautz showed in 1974 that a sufficient condition to detect any pair of short circuited nets was that the serial codes (STV) must be unique for all nets. The corresponded test length is \( \lceil \log_2(N) \rceil \).
The Modified Counting Sequence

Assume stuck-at-0

Assume wired AND

All 0-s and all 1-s are forbidden STV codes because of open faults. Therefore the final test length is $\lceil \log_2(N+2) \rceil$

This method was proposed in 1982 by Goel & McMahon

How to improve the diagnosis?
To improve the diagnostic resolution Wagner proposed the True/Complement Code in 1987. The test length became equal $2^{{\left\lfloor \log_2(N) \right\rfloor}}$. All-0 and all-1 codes are not forbidden anymore!
The True/Complement Code

Important properties of the True/Complement Code are:
- There are equal numbers of 0-s and 1-s upon each line
- Hamming distance between any two code words is at least 2
- Some shorts and opens cannot be distinguished (e.g. n2/n3)

How to distinguish between opens and shorts?
Extended True/Complement Code

Idea: add two bits, that are the same at every STV code word
Shorts and stuck-at faults are now distinguishable
The test length is $2\lceil \log_2(N) \rceil + 2$
Comparison of main Test Sequences

<table>
<thead>
<tr>
<th>Counting</th>
<th>Modified</th>
<th>True/Compl.</th>
<th>Extended</th>
<th>Walking 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>001</td>
<td>111 000</td>
<td>01 111 000</td>
<td>10000000</td>
</tr>
<tr>
<td>001</td>
<td>010</td>
<td>110 001</td>
<td>01 110 001</td>
<td>01000000</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
<td>101 010</td>
<td>01 101 010</td>
<td>01000000</td>
</tr>
<tr>
<td>011</td>
<td>100</td>
<td>011 101</td>
<td>01 100 011</td>
<td>00010000</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>010 100</td>
<td>01 101 100</td>
<td>00010000</td>
</tr>
<tr>
<td>101</td>
<td>110</td>
<td>001 110</td>
<td>01 100 101</td>
<td>00001000</td>
</tr>
<tr>
<td>110</td>
<td>111</td>
<td>000 111</td>
<td>01 001 110</td>
<td>00000100</td>
</tr>
<tr>
<td>111</td>
<td></td>
<td></td>
<td>01 000 111</td>
<td>00000001</td>
</tr>
</tbody>
</table>

Length

<table>
<thead>
<tr>
<th>Length</th>
<th>$\lceil \log_2(N) \rceil$</th>
<th>$\lceil \log_2(N+2) \rceil$</th>
<th>$2 \lceil \log_2(N) \rceil$</th>
<th>$2 \lceil \log_2(N) \rceil + 2$</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example (N=10000)</td>
<td>14</td>
<td>14</td>
<td>28</td>
<td>30</td>
<td>10000</td>
</tr>
<tr>
<td>Hamming distance</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Defects

<table>
<thead>
<tr>
<th>Defects</th>
<th>Counting</th>
<th>Modified</th>
<th>True/Compl.</th>
<th>Extended</th>
<th>Walking 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shorts</td>
<td>Shorts</td>
<td>Shorts</td>
<td>Shorts</td>
<td>Shorts</td>
<td>Shorts</td>
</tr>
<tr>
<td>Opens</td>
<td>Opens</td>
<td>Opens</td>
<td>Opens/Delays</td>
<td>Opens/Delays</td>
<td>Opens/Delays</td>
</tr>
</tbody>
</table>

Diagnostic Properties

<table>
<thead>
<tr>
<th>Diagnostic Properties</th>
<th>Counting</th>
<th>Modified</th>
<th>True/Compl.</th>
<th>Extended</th>
<th>Walking 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad</td>
<td>Bad</td>
<td>Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
</tr>
</tbody>
</table>
More complex case: branching nets

- enable
- data
- data_in

- driver fault
- open

- short

- data
- data_in

- data_in
Additional rules for branching nets

- Every driver on the net should at least once drive low and at least once drive high.
- Every receiver should at least once sense 0 and at least once sense 1.
- Two or more drivers on the same net should never drive simultaneously.
- One can distinguish between a driver fault and open net by sensing back on a bi-directional pin.
## Memory Interconnect Test Principle

<table>
<thead>
<tr>
<th>Address 10bit</th>
<th>Data 32 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>walking 1</strong></td>
<td><strong>counting sequence (true/complement)</strong></td>
</tr>
<tr>
<td>00000000000</td>
<td>..........write here something unique........</td>
</tr>
<tr>
<td>10000000000</td>
<td>01010101010101010101010101010101</td>
</tr>
<tr>
<td>01000000000</td>
<td>00110011001100110011001100110011</td>
</tr>
<tr>
<td>00100000000</td>
<td>00001111100001111110000000111111</td>
</tr>
<tr>
<td>00010000000</td>
<td>00000000011111111000000000111111</td>
</tr>
<tr>
<td>00001000000</td>
<td>00000000000000000111111111111111</td>
</tr>
<tr>
<td>00000100000</td>
<td>10101010101010101010101010101010</td>
</tr>
<tr>
<td>00000010000</td>
<td>11001100110011001100110011001100</td>
</tr>
<tr>
<td>00000001000</td>
<td>11110000111100001111000001111000</td>
</tr>
<tr>
<td>00000000100</td>
<td>11111111000000000111111111111111</td>
</tr>
<tr>
<td>00000000010</td>
<td>11111111000001111111111110000000</td>
</tr>
<tr>
<td>00000000001</td>
<td>11111111111111110000000000000000</td>
</tr>
</tbody>
</table>
Memory Test: detection of shorts

- Any short between address lines will cause writing the corresponding PTV to address 0.
- Any short between data lines will cause writing wrong data to the correct address.
- Shorts between data and address lines need a double-lengths test to generate all 1/0 combinations between a particular address line and data lines. Such shorts will cause writing to address 0.
- **Fault detection:** read from all involved addresses then read from address 0.
- **Fault diagnosis:** after each write operation, read from address 0.

... assuming Wired-AND model
Memory Test: detection of opens

- Any s-a-0 at address lines will cause writing the corresponding PTV to address 0.
- Any s-a-0 at data lines will cause writing wrong data to the correct address.
- **Fault detection:** read from all involved addresses then read from address 0.
- **Fault diagnosis:** after each write operation, read from address 0.

- The same procedures can be repeated for stuck-at 1 opens and Wired-OR shorts using complementary test data (walking-0 code).
Outline

- Board level test challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- **IEEE 1149.1 Boundary Scan Standard**
- Application of Boundary Scan
• Early 1980’s – problem of test access to PCBs via “bed-of-nails” fixture

• Mid 1980’s – Joint European Test Action Group (JETAG)

• 1986 – US companies involved: JETAG -> JTAG

• 1990 – JTAG Test Port became a standard [4]:

IEEE Std. 1149.1: Test Access Port and Boundary Scan Architecture comprising serial data channel with a 4/5-pin interface and protocol
Test Access Via Boundary Scan
Test Access Via Boundary Scan
Test Access Via Boundary Scan

Defects covered:

driver scan cell, driver amp, bond wire, leg, solder, interconnect, solder, leg, bond wire, driver amp, sensor scan cell
Test Access Via Boundary Scan
Boundary Scan basics

For describing Boundary Scan devices, BSDL (Boundary Scan Description Language) models are used.
IEEE 1149.1 Device Architecture

- **Core Logic**
- **BS Cells**
- **TAP Controller**
- **Identification Register**
- **Instruction Register**
- **Bypass**
- **Test Reset (Optional)**
- **TDI (Test Data In)**
- **TDO (Test Data Out)**
- **TMS (Test Mode Select)**
- **TCK (Test Clock)**
Typical Boundary Scan Cell (BC_1)

BC_1 is used both at input and output pins
## Boundary Scan Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE / PRELOAD</td>
<td>Mandatory</td>
</tr>
<tr>
<td>EXTEST</td>
<td>Mandatory</td>
</tr>
<tr>
<td>BYPASS</td>
<td>Mandatory</td>
</tr>
<tr>
<td>IDCODE</td>
<td>Optional</td>
</tr>
<tr>
<td>INTEST</td>
<td>Optional</td>
</tr>
<tr>
<td>CLAMP</td>
<td>Optional</td>
</tr>
<tr>
<td>HIGHZ</td>
<td>Optional</td>
</tr>
<tr>
<td>RUNBIST</td>
<td>Optional</td>
</tr>
<tr>
<td>USERCODE</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Boundary Scan Working Modes

**SAMPLE/PRELOAD instruction – sample mode**

Get snapshot of normal chip output signals
Boundary Scan Working Modes

SAMPLE/PRELOAD instruction – preload mode

Shift out snapshot data and shift in new test data to be used later
EXTEST instruction – driving and sensing:

Test off-chip circuits and board-level interconnections
Boundary Scan Working Modes

**EXTEST instruction – shifting**

Shift out snapshot data and shift in new test data to be used later.
### Typical BS Interconnect Test Flow

<table>
<thead>
<tr>
<th>BS mode</th>
<th>Test bus actions</th>
<th>Test data manipulations</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRELOAD</td>
<td>IRshift + DRshift</td>
<td>Loading the first test vector to BS register (vector includes control/disable values for other devices on the bus)</td>
<td>Vector 1 loaded</td>
</tr>
<tr>
<td>EXTEST</td>
<td>IRshift + DRshift</td>
<td>1. Applying vector 1 to the DUT 2. Capturing test responses from DUT in BS reg. 3. Reading back test responses and loading new test vector to BS register</td>
<td>Vector 1 applied and analyzed</td>
</tr>
<tr>
<td>EXTEST</td>
<td>DRshift</td>
<td>1. Applying vector 2 to the DUT 2. Capturing test responses from DUT in BS reg. 3. Reading back test responses &amp; and loading new test vector to BS register</td>
<td>Vector 2 applied and analyzed</td>
</tr>
<tr>
<td>EXTEST</td>
<td>DRshift</td>
<td>1. Applying vector ( N ) to the DUT 2. Capturing test responses from DUT in BS reg. 3. Reading back test responses</td>
<td>Vector ( N ) applied and analyzed</td>
</tr>
</tbody>
</table>

**Time**

\[
\text{N test vectors: } (N+1) \text{ DRshifts} + 2 \text{ IRshifts} \approx (N+1) \text{ DRshifts}
\]
**Boundary Scan Working Modes**

**BYPASS instruction:**

Bypasses the corresponding chip using 1-bit register

Similar instructions: **CLAMP, HIGHZ**
**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.

<table>
<thead>
<tr>
<th>TDI</th>
<th>Version</th>
<th>Part Number</th>
<th>Manufacturer ID</th>
<th>TDO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4-bits</td>
<td>16-bits</td>
<td>11-bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Any format</td>
<td>Any format</td>
<td>Coded form of JEDEC</td>
<td></td>
</tr>
</tbody>
</table>
Blind Interrogation

Default instruction:

- IDCODE (but it is not mandatory)
- BYPASS (if IDCODE is not implemented)

Default capture bits:

- 0 in BYPASS register
- 1 – first bit in IDCODE register

Example: 0...............................................10
The TAP state diagram has two main branches and two idle states.

**Shift IR** and **Shift DR** states are used to insert instructions and test data into the BS device. These are the most important states.

The number of states is exactly 16 (to avoid some undefined states).

**TMS** signal is used to move through the states.
Outline

- Board level test challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan
BScan Implementation Rules

- One or more BSC at each system input or output of on-chip system logic (core logic)
- BSC may be connected to chip-internal signals
- No BSC on:
  - TAP pins (TCK, TMS, TDI, TDO, TRST)
  - Compliance Enable Pins
  - Non-digital pins (e.g. analog pins, power pins)
- No logic between BSC and I/O pin it is connected to (a buffer is allowed)
BScan Implementation Examples

BScan Cell

Analog

Core Logic (digital)

TAP controller

VCC

GND

TDI

TCK

TMS

TDO
Automatically generated:

- Infrastructure test (generated by using BSDL models)
- Interconnect test (BSDL models + interconnection netlist)
Extended testability via Boundary Scan

BS devices:
- Microprocessor
- FPGA/PLD

Non-BS devices
- RAM
- Flash
- Clock generator
- Cluster logic
- Buffers/MUXes
- Pull-up/Pull-down resistors
- Jumpers
- D/A converter
- External connectors (analog and digital)
Typical test types and their order of application:

- **Infrastructure test**
- **Interconnect test**
  - One needs to specify **behavioral models for non-BS components** to get acceptable test coverage (no standard description format exists)
- **Memory test**
- **External connectors test**
- **Clock oscillator test**
- **Cluster logic test (semi-automated)**
- **LED or display test (can be assisted by camera/sensor)**
- **FLASH test/program/read ID** – **in-system programming**
Unidirectional buffer

Bidirectional buffer
Cluster Logic Test (Manual)

Cluster’s truth table is needed

Truth table

000 0
001 1
010 1
...
111 1
To generate test:

- Specify constraints that will select only one device
- RAM/Flash model in special format that provides description of read/write protocol
- Combination of walking and counting sequences is used
Testing External Connectors

Only interconnect test will be performed!

The real protocol of external connector is not tested
Boundary Scan Test Development
Typical Workflow

1. **CAD Netlist**
   - Constraints
   - Scan-path configuration

2. **Parser**
   - Netlist with drive/sense constraints
   - Interconnect ATPG

3. **Compiled Test Program**
4. **Test Coverage Report**

5. **BSDL Models**
   - Models for non-BS components (buffers, ram, flash, etc)

6. **Compiled Test Data**

- **Functional models for RAM/Flash**
- **RAM Test, Flash Test/Program Generator**
- **Test/Program Generator**
- **Test Coverage Report**
Boundary Scan Standard has become absolutely essential:

- No longer possible to test printed circuit boards with bed-of-nails tester
- Not possible to test multi-chip modules at all without it
- Supports BIST, external testing with Automatic Test Equipment, and boundary scan chain reconfiguration as BIST pattern generator and response compacter
- Has got a widespread usage
Importance of Boundary-Scan to Production Goals

- **Not important**: we do not use 2%
- **Highly important**: cannot meet goals if it fails to work 49%
- **Moderately important**: we can work around most problems 30%
- **We are just dabbling in boundary scan -- not sure about it**: 8%
- **It helps fill in coverage -- other tests do the most for us**: 11%

Source: iNEMI Boundary Scan Adoption Report published at ITC 2009 conf.
Source: iNEMI Boundary Scan Adoption Report published at ITC 2009 conf.
Boundary Scan – Evolution

1149.4 – Mixed-Signal Test Bus (testing analog signals)
1149.6 – Boundary-Scan Testing of Advanced Digital Networks (testing high speed links)
1149.7 – CJTAG – Compact JTAG (debug)
1149.8.1 – Sensing using capacitive plate
1687 – IJTAG – Internal JTAG (component testing, BIST)
1500 – Embedded Core Test (SoC testing)
1532 – In-System Configuration of Programmable Devices
1581 – Static Component Interconnection Test Protocol and Architecture (memory-to-BS_chip links testing)
1838 – TSV interconnect test in 3D chips (also at-speed)
5001 – NEXUS – Global Embedded Processor Debug Interface (SW development, debug, and emulation)
New Standards and Their Purposes

- IEEE 1149.7 – improved **flexibility** of the JTAG bus by relaxing topology requirements and by addressing resources; potential data throughput improvements
- IEEE 1149.8.1 – improved **observability** for measurement (from the JTAG standpoint) by implementing a capacitive sensing technology
- IEEE 1149.1-2013 – solves **signaling** issues on the buses by driver initialization procedures
- IEEE 1687 – introduces the concept of **embedded instrumentation** for test application, measurement and diagnosis tasks
- Processor emulation standards (e.g. NEXUS) and solutions – converts a **MPU** into a test instrument
## IEEE Scan-Based Board-Level Test Access

<table>
<thead>
<tr>
<th>Main Target Application</th>
<th>Main Purpose</th>
<th>Essential Technology</th>
<th>Target Fault Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IEEE 1149.1 – Boundary Scan</strong></td>
<td>Manufacturing test of PCBA</td>
<td>Test access (TA) improvement</td>
<td>On-chip scan registers</td>
</tr>
<tr>
<td><strong>IEEE 1149.4 – Mixed-Signal Test Bus</strong></td>
<td>Measurement: analog values</td>
<td>TA improvement</td>
<td>On-chip switches</td>
</tr>
<tr>
<td><strong>IEEE 1149.6 – BST of Advanced Digital Networks</strong></td>
<td>Testing LVDS high-speed nets</td>
<td>Test trough AC-coupled nets</td>
<td>On-chip pulse generators</td>
</tr>
<tr>
<td><strong>IEEE 1149.7 – Reduced-pin and Enhanced TAP</strong></td>
<td>Board test; SW debug</td>
<td>Flexible 2-pin high-speed TA</td>
<td>SERDES, addressing</td>
</tr>
<tr>
<td><strong>IEEE 1149.8.1 – Pin Toggle and Contactless Sensing</strong></td>
<td>Interconnect test of PCBA</td>
<td>Links to passive components</td>
<td>Capacitive sense plate</td>
</tr>
<tr>
<td><strong>IEEE P1149.10 – High Speed Test Access Port (TAP)</strong></td>
<td>All of the above</td>
<td>High-speed test data exchange</td>
<td>Reuse of high speed I/O pins</td>
</tr>
<tr>
<td><strong>IEEE 1687 – Embedded Instrumentation Access</strong></td>
<td>IC test, debug, diagnosis</td>
<td>Instrument access standard</td>
<td>Reconfigurable scan chains</td>
</tr>
</tbody>
</table>
New and Emerging Standards Combined

Instruments

IEEE P1687
IEEE 1149.7
IEEE 1149.1
IEEE P1149.1-2013
What to try further

Training software:
- goJTAG – open source project (http://www.goJTAG.com/)
- Trainer 1149 by Testonica Lab (http://www.testonica.com/1149/download)
- Scan Coach by Goepel Electronic (http://www.goepel.com/index.php?id=1418&L=4)
- Scan Educator by Texas Instruments (http://focus.ti.com/docs/toolsw/folders/print/scan_educator.html)

Literature:
- Kenneth P. Parker, The Boundary-Scan Handbook

Leading BScan companies:
- Goepel Electronic (http://www.goepel.com/)
- ASSET Intertech (http://www.asset-intertech.com/)
- JTAG Technologies (http://www.jtag.com/)