Squeezing Out More Test Coverage: Bridging the Gap Between Boundary Scan and Functional Test

Michael R. Johnson
ScanWorks BST Product Manager/Support Manager
August 31, 2021
Agenda

- Test Coverage Considerations
- Coverage Assessment Methodology
- iNEMI PCOLA/SOQ/FAM Framework
- Bridging the Gap with the Component Action
- Functional Test with the Component Action
- Component Action Facts
- Component Model Structure
- Component Model Base Commands
- Functional Test Examples
- Demonstration
- Summary
Every PCB manufacturer would like to have 100% test coverage of their PCBs, but is this achievable?

In manufacturing and test environments, time is money; PCB manufacturers must decide how much of each they are willing to invest to ensure a quality product.

PCB manufacturers must decide what percentage of PCB rejects and returns are acceptable due to reduced or untested areas of the PCB.

Each PCB manufacturer must decide what determines maximum test coverage for their product.
When asked, engineers will say the goal of a PCB test plan is 100% coverage.

Achieving maximum test coverage is an iterative process with coverage being improved with each “turn” or redesign of the PCB.

Following design for test (DFT) guidelines during PCB design and layout produces maximum coverage.
Coverage that is too low will result in warranty returns and high cost of product replacements in the field.

Coverage that is too high becomes expensive and can affect a product’s total cost, driving up price and making it uncompetitive in the market.

The goal becomes achieving optimum test coverage (i.e., best coverage achievable given those restraining factors).
## iNEMI PCOLA/SOQ/FAM Framework

<table>
<thead>
<tr>
<th>Structural Devices</th>
<th>P</th>
<th>Presence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
<td>Correctness</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>Orientation</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>Live</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>Alignment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structural Connections</th>
<th>S</th>
<th>Shorts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
<td>Opens</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>Quality</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Connections</th>
<th>F</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>At-Speed</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>Measurement</td>
</tr>
</tbody>
</table>

**PCOLA** tests are tests that verify the Presence of a chip in a socket or at a board location; that it is the Correct chip; and is Oriented correctly; receives power and is Live; and is Aligned correctly.

**Boundary Scan Tests and External Equipment (ICT, X-Ray) and Visual Inspection**

**SOQ** tests are tests that verify the connections/signals to the chip to be free of Shorts and Opens; and can assess the Quality of the connection’s solder joints.

**Boundary Scan Interconnect Test and X-Ray**

**FAM** tests are tests that can operate or verify a Feature of the chip; can be applied At-speed; and that allow a Measurement to be taken.

**Functional Tests**
PCB functional test coverage can be realized by using boundary scan.

The ScanWorks Component Action can “bridge the gap” between boundary scan and functional test to enhance PCB test coverage.
The ScanWorks Component action is a general-purpose action that extends functional test coverage to a wide array of non-boundary scan devices:

- ADCs/DACs
- Oscillators, clocks, PLLs
- LEDs
- Connectors
- I2Cs
- Ethernet PHYs
- Switches
- System monitors
The Component action uses the Tool Command Language (Tcl or Tickle) programming language along with a PCB's existing boundary scan resources to access non-boundary scan devices.

Any non-boundary scan device with boundary scan access, and an algorithm that can be modeled from its data sheet through Tcl, is a candidate for functional testing by the Component action.
The Tcl interpreter is included in ScanWorks.

The Tk package is also included in ScanWorks for creation of dialogs and GUIs.

Component models define the pins on the target device that require boundary scan access.

Component models are based on the operational algorithms of the non-boundary scan devices.

Component models are available for download from the ScanWorks Model Library for use as-is or as templates for model development.

```
set f [open $vectorFileName "w"]
puts $f "$VECTOR ADDRESS 10 9 8 7 6 5 4 3 2 25 24 23 21"
puts $f "$VECTOR DATA 19 18 17 16 15 13 12 11"
puts $f "$VECTOR WE 27"
puts $f "$VECTOR CE 20"
puts $f "$VECTOR OE 22"
close $f
```

# read the data bus
set actualBusValue [ca readBus DATA]
puts "DATA bus value is $actualBusValue"
if { $actualBusValue != $expectedBusValue($i) } {
    puts "Test failed: expected a data bus value of $expectedBusValue($i)"
    set returnValue 1
}

<table>
<thead>
<tr>
<th>Models</th>
<th>Date Posted</th>
</tr>
</thead>
<tbody>
<tr>
<td>atmel_at93c46b_64x16_ver5.tcl</td>
<td>September, 26 2012 08:58:00</td>
</tr>
<tr>
<td>atmel_at93c66a_256x16_ver4.tcl</td>
<td>September, 26 2012 09:07:00</td>
</tr>
<tr>
<td>Broadcom_BCM5482_PHY_BGA.tcl</td>
<td>May, 03 2013 02:48:00</td>
</tr>
<tr>
<td>Broadcom_BCM5785_XOR.tcl</td>
<td>May, 03 2013 02:49:00</td>
</tr>
<tr>
<td>CY6264.tcl</td>
<td>April, 05 2012 09:19:00</td>
</tr>
</tbody>
</table>
The Proc loadComponentActionSupport procedure loads all of the necessary ASSET Component Action classes when the action is run.

Pins on the target non-boundary scan device to be functionally tested are listed in the proc Build procedure:
- This information is used along with UUT netlist to find access to the target non-boundary scan device.

The proc Run procedure defines the algorithm necessary to functionally test the target non-boundary scan device:
- Uses the buses defined in the proc Build and the user-supplied commands to communicate with the target non-boundary scan device.
Component Model Base Commands

- Command: setBus
  - Sets the TDI buffer for the boundary scan cells used to access the pins of the non-Boundary Scan device

- Command: disableBus
  - If possible, disables the boundary scan driver cells used to drive the pins

- Command: DRScan
  - Performs a DR scan to scan in the contents of the TDI buffer and captures the data in the TDO buffer that is scanned out

- Command: readBus
  - Performs a DR scan and reads the data from the TDO buffer for the bus pin

See the documentation “Using Component Actions” at c:\scanworks\doc for more information
Functional Test Example

LED Verification

Boundary Scan ICs

ScanWorks Component Action

Tcl Model File
Algorithm
Blink LEDs

Operator input GUIs created with Tk

Visual Inspection
Optical Sensor

Boundary Scan ICs

TDI
TMS
TCK

Operational Circuit

Operator Input
Component Action Example

Did the D5 LED blink?

Yes
No

© 2021, ASSET InterTech, Inc.
Functional Test Example

Connector Verification

Boundary Scan ICs

Line Driver

Opens coverage

Shorts coverage

ScanWorks Component Action

Tcl Model File Algorithm Loopback test

TDO

© 2021, ASSET InterTech, Inc.
Functional Test Example

Clock Verification

Boundary Scan ICs

20 MHZ Oscillator

Test for oscillator clock transitions

ScanWorks Component Action

Tcl Model File Algorithm Clock presence

TDI

TMS

TCK

TDO
Functional Test Example

ADC with Temperature Measure

Test for device temperature (U2) vs. PCB ambient temperature along with power supply measurements

ADC Converter

Boundary Scan ICs

ScanWorks Component Action

Tcl Model File
Algorithm Temp measure

TDI
TDO
TMS
TCK

SCLK
SDIN
SDOUT
CSn
CH0
CH1
CH2
CH3
CH4
CH5

~1.96 VDC
~2.00 VDC
~1.20 VDC
~2.18 VDC

~1.96 VDC
~2.00 VDC
~1.20 VDC
~2.18 VDC

Ambient Temperature Sensor

Temperature Sensor
Demonstration

ADC with Temperature Measure

Boundary Scan ICs

Test for device temperature (U2) vs. PCB ambient temperature along with power supply measurements

ADC Converter

ScanWorks Component Action

SourcePoint

Platform for Software Debug and Trace

© 2021, ASSET InterTech, Inc.
Bridging the boundary scan and functional test gap with the Component action has many positive results:

- Adds functional testing to areas of the PCB that might have gone untested
- One station for structural, programming and functional testing reduces PCB handling and manufacturing cost
- Reduces need for large equipment saving factory floor space

Failing devices can be identified during manufacturing where the cost to repair is lower as compared to latter production phases or at product release.

Functional test can be implemented through 3rd party applications such as LabVIEW, TestStand, along with boundary scan test.

ASSET provides a library of Component models for use as-is or as templates for model development so it’s easy to get started.


Questions and Contact Information

Contact Information:
Michael R. Johnson
7161 Bishop Rd. Ste. 250
Plano, TX. 75024

mjohnson@asset-intertech.com
www.asset-intertech.com
Real Insight from Code to Silicon