



Boundary-Scan Test / IEEE1149.x

Mario Blunk

Blunk electronic / Inh. Dipl. Ing. Mario Blunk / Buchfinkenweg 3 / 99097
Erfurt / Deutschland / www.blunk-electronic.de

2019-09-26

Abstract

Digitale Schaltungsstrukturen sind im Zuge der permanenten Miniaturisierung kaum noch mechanisch kontaktierbar. Angeschlossene Peripherie bestehend aus Leuchtdioden, Displays, Motoren, Ventilen, Schaltern und Sensoren ist im Funktionstest umständlich testbar, oder es sind Testroutinen in Firmware nötig. Boundary-Scan entschärft diese Problematik und ist in den IEEE1149.x Standards seit nunmehr zwanzig Jahren genormt. Dieser Vortrag vermittelt die Grundlagen des Standards, die Implementierung der nötigen Strukturen während der Produktentwicklung und die Möglichkeiten dieses Testverfahrens. Boundary-Scan unterstützt Design, Inbetriebnahme von Prototypen und Systemen, Qualitätssicherung in Fertigung und Reparatur. Bestückungs- und Verdrahtungsfehler oder Ausfälle im Feld werden bis herab auf Bauteil- und Netzebene detektiert. Testroutinen in Firmware können somit entfallen oder stark reduziert werden. Die Möglichkeiten der sinnvollen Kombination mit anderen Testverfahren wie ICT, FPT, AOI oder Funktionstest werden vorgestellt. Anhand eines realen Testsystems wird die praktische Anwendung in einfachen Beispielen demonstriert.

1. Wie funktioniert Boundary-Scan / IEEE1149.x ?
2. Anwendung
3. Kombination mit anderen Testverfahren
4. Ausrüstung
5. Demo

Wie funktioniert Boundary-Scan ? 1

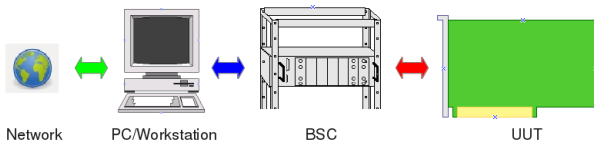


Figure 1: System Überblick

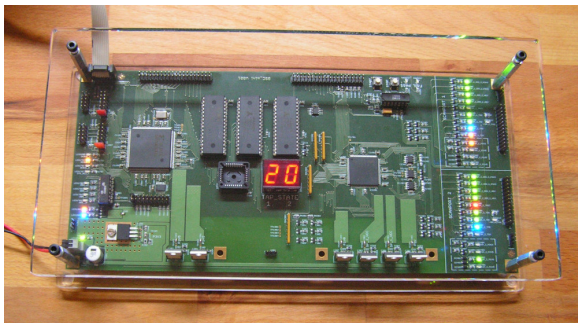


Figure 2: Boundary-Scan Controller (BSC)

Wie funktioniert Boundary-Scan ? 2

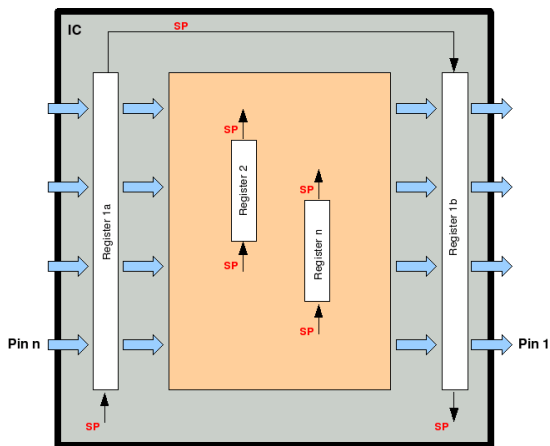


Figure 3: im IC

Wie funktioniert Boundary Scan ? 3

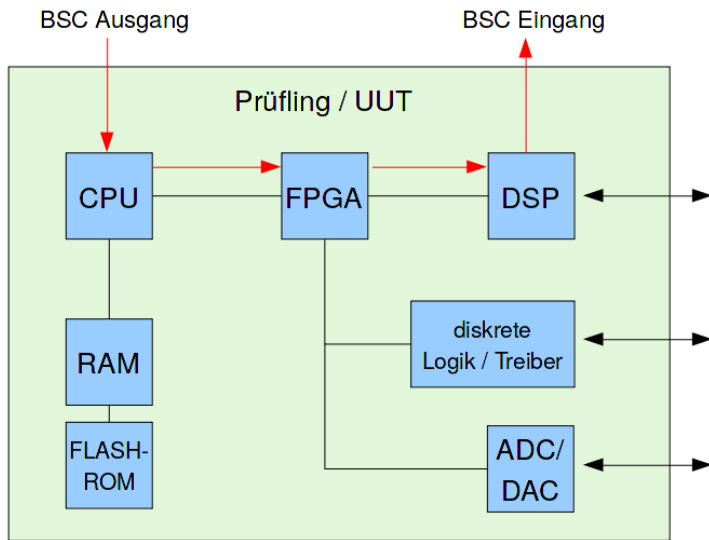


Figure 4: On-Board

Wie funktioniert Boundary-Scan ? 4

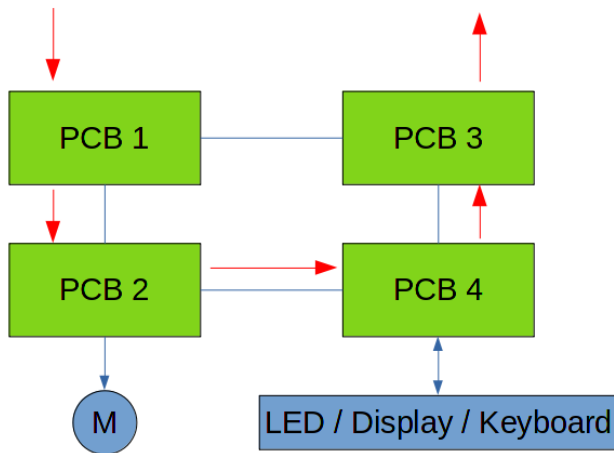
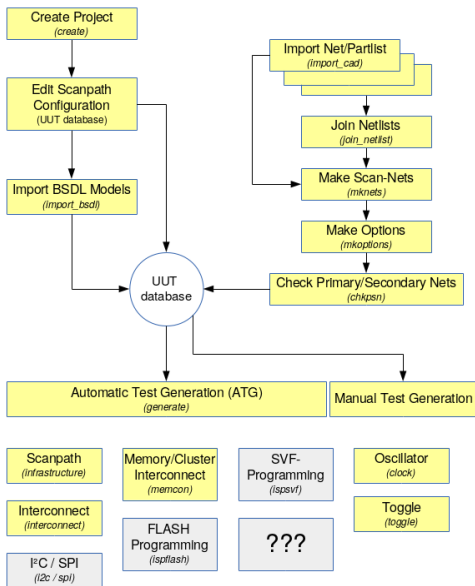


Figure 5: Systemtest

Testgenerierung 1



Testgenerierung 2

```
# infrastructure and interconnect tests
bsmcl generate $database_default infrastructure infra
bsmcl compile $database_default infra

bsmcl generate $database_default interconnect intercon
bsmcl compile $database_default intercon

bsmcl generate $database_ram_out_1 memconnect ram_out_1 IC602 models/AS6C4008.txt NDP32
bsmcl compile $database_ram_out_1 ram_out_1

bsmcl generate $database_ram_out_2 memconnect ram_out_2 IC603 models/AS6C4008.txt NDP32
bsmcl compile $database_ram_out_2 ram_out_2

bsmcl generate $database_ram_in memconnect ram_in IC601 models/AS6C4008.txt NDP32
bsmcl compile $database_ram_in ram_in

#bsmcl generate $database_clock_test clock clk_slow IC801 154 10 1
#manually modified for intrusive mode
bsmcl compile $database_clock_test clk_slow

#bsmcl generate $database_clock_test clock clk_master IC801 181 10 0.1
#manually modified for intrusive mode
bsmcl compile $database_clock_test clk_master

#cluster tests
bsmcl compile $database_default cluster_digital
bsmcl compile $database_default cluster_analog

#vccio
bsmcl compile $database_default vccio

#LED - requires vccio set to 3V3 and a very low tck frequency
#bsmcl generate $database_vccio_3v3 interconnect intercon_slow
bsmcl compile $database_vccio_3v3 intercon_slow

bsmcl load infra
bsmcl load intercon
bsmcl load ram_out_1
bsmcl load ram_out_2
bsmcl load ram_in
bsmcl load cluster_digital
bsmcl load cluster_analog
bsmcl load clk_slow
bsmcl load clk_master
bsmcl load vccio
bsmcl load intercon_slow

bsmcl run infra
bsmcl run intercon
```

Figure 6: automatisierbar via UNIX/Linux Shell-Skripte

Test Ausführung 1

```
Test FAILED! Diagnosis:
failed scanpath   : 1
step id (dec)    : 7
srx length (dec) : 242 (one-based)
srx fail pos (dec): 9 (one-based)
scan type       : SDR
device position  : 3
device name     : IC303
register        : BOUNDARY
failed bit pos. : 8 (zero-based)
expected       : HIGH
net class      : PU

secondary net    : CT_D0

JP402 pin 2
IC303 pin 15
IC302 pin 3

primary net     : D0

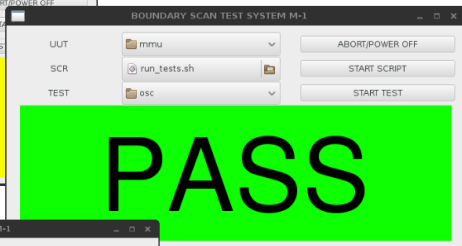
RN401 pin 2
JP406 pin 2
IC203 pin 11
IC202 pin 11
IC201 pin 13
IC200 pin 13

stuck at LOW or Pull-Up resistor missing !
```

Figure 7: Fehlerbericht

Test Ausführung 2

Einsatz in Fertigung: GO/NOGO - Test



Anwendung

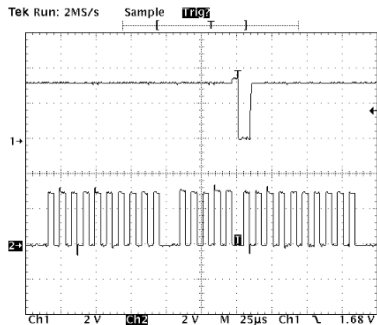
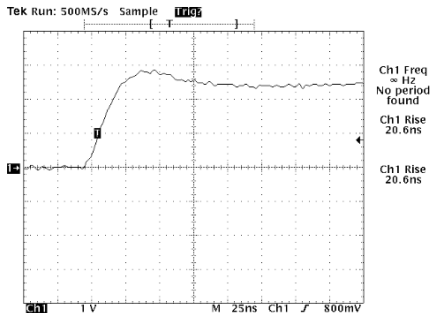
1. Board-Test (PCBA) in Serienfertigung
2. digitale Strukturen (evtl. mixed Signal)
3. Test von PCBA-Prototypen
4. Geräte- und Systemtest
5. Agile Hardware-Entwicklung
6. In-System-Programmierung (ISP)

Kombination mit anderen Testverfahren

1. Sichtkontrolle / AOI / AXI als Vorprüfung
2. In-Circuit-Test / FPT (analoge Strukturen, mixed Signal)
3. Funktionstest (Peripherie)
4. Schnittstellen zwischen Testsystemen nötig
5. offene Datenbasis des UUT
6. OpenSource der Testgeneratoren

Ausrüstung 1

1. Spannungsversorgung (Strombegrenzung !)
2. Multimeter
3. Oszilloskop
4. optional Logiktester / Bus-Scanner



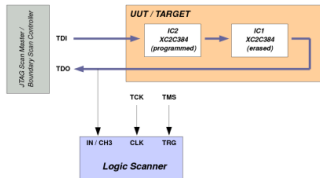
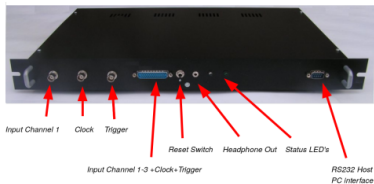
Ausrüstung 2

1. Spannungsversorgung (Strombegrenzung !)
2. Multimeter
3. Oszilloskop
4. optional Logiktester / Bus-Scanner

```
Session Edit View Bookmarks Settings Help
LOGIC SCANNER V2.9.1

IN  H          FLT OFF      TRM OFF  CH3
DRV  L   OFF    WEAK1      STAT
CLK  L          FLT OFF    + EDGE   TRM OFF
TRG  L          FLT OFF    - EDGE   TRM OFF

-----
TRACE ONE-SHOT      MAN TRG
MEM  00000101 00000001 10000000 00000000 00000000 00000000
EXPECT PATTERN: 05 MATCH AKN?      TAP: Shift-IR
-----
AUDIO IN      VOL 00
BEEP OFF
```



Literaturquellen I

- [1] Mario Blunk / Blunk electronic *Zuverlässigkeit in der Elektronik*.
<http://www.blunk-electronic.de/pdf/zuverlaessigkeit.pdf>
- [2] Mario Blunk / Blunk electronic *Testverfahren der Elektronik*.
http://www.blunk-electronic.de/pdf/testverfahren_der_elektronik.pdf
- [3] Mario Blunk / Blunk electronic *Boundary-Scan Training Teil 1*.
http://www.blunk-electronic.de/pdf/bst_teil_1.pdf
- [4] Mario Blunk / Blunk electronic *Boundary-Scan Training Teil 2*.
http://www.blunk-electronic.de/pdf/bst_teil_2.pdf
- [5] Mario Blunk / Blunk electronic *Agile HW Entwicklung*.
http://www.blunk-electronic.de/pdf/agile_HW/agile_HW_development.pdf