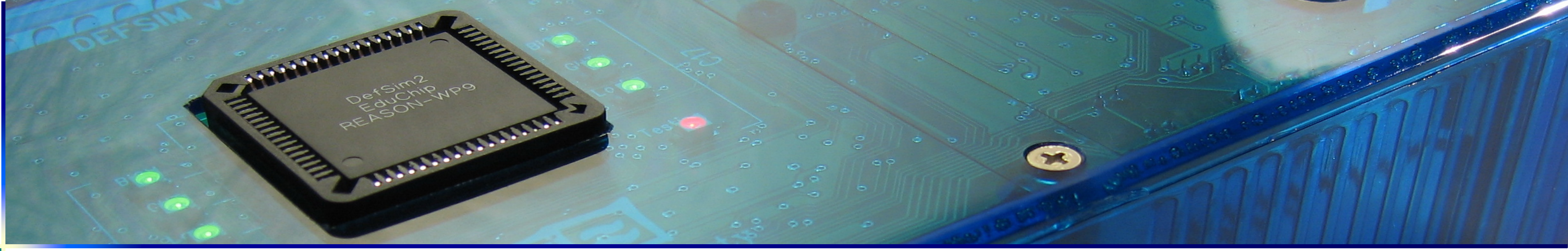
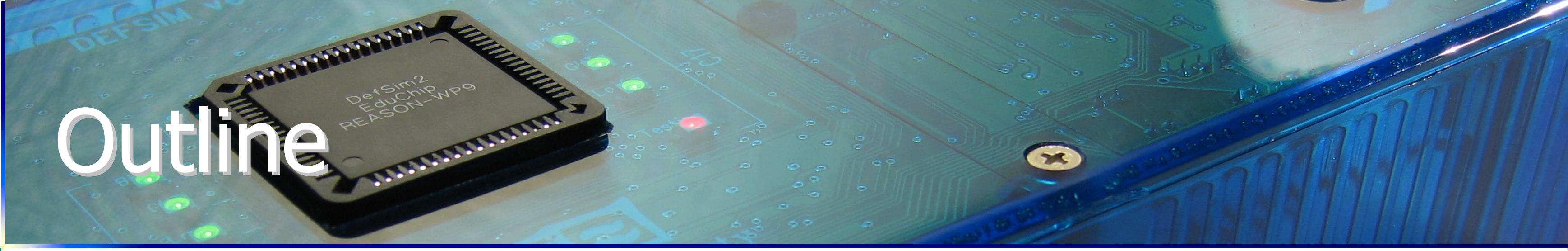


Board-level testing and IEEE1149.x Boundary Scan standard

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

February 2011

1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



Outline

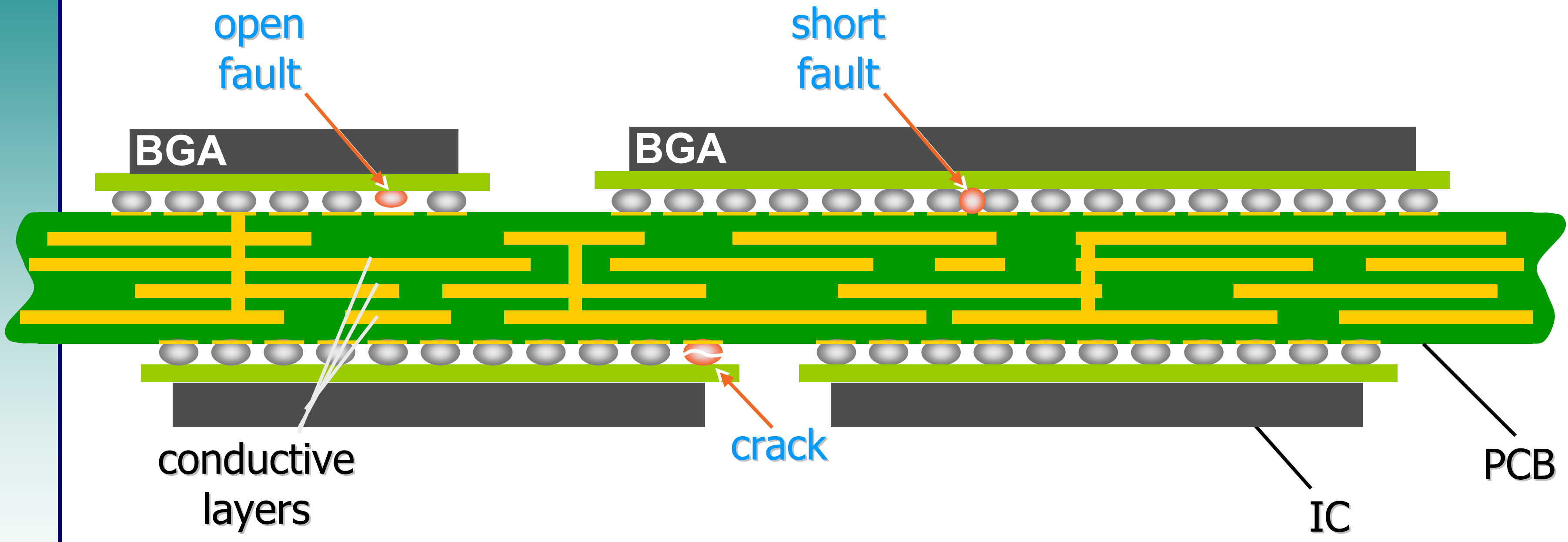
- Board level testing challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan

Industrial approach to board test

- Visual inspection
- Optical/x-ray inspection
- Smoke test ;-)
- Power distribution test
- Structural test
  - in-circuit test (ICT)
  - Boundary Scan (BS)
  - Test Processors/Cores (BIST)
- Functional test (FT)

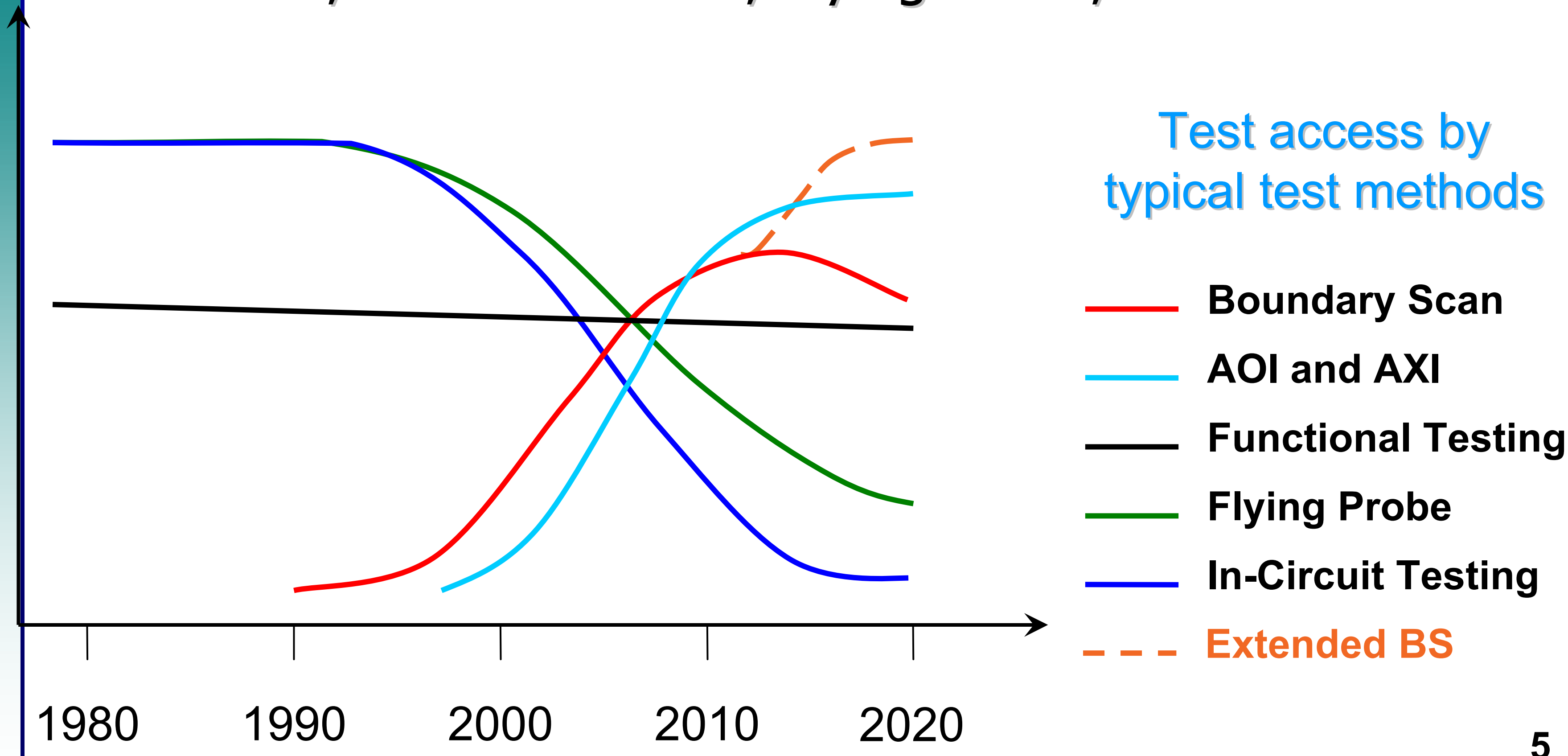
The test access problem…

Limited access (nail probing) for test, measurement, diagnostics

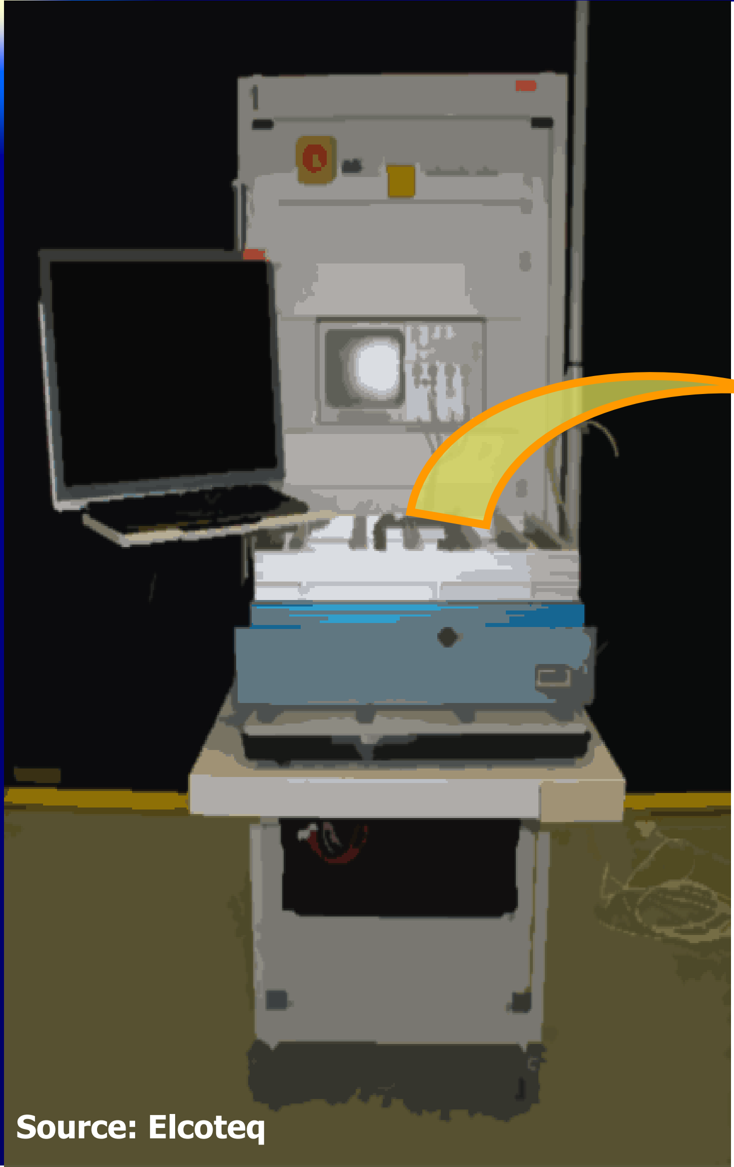


Test Access: Past and Present

- Past: test access was a problem
- Present: good test access by Boundary Scan combined with AOI, Functional Test, Flying Probe, etc.



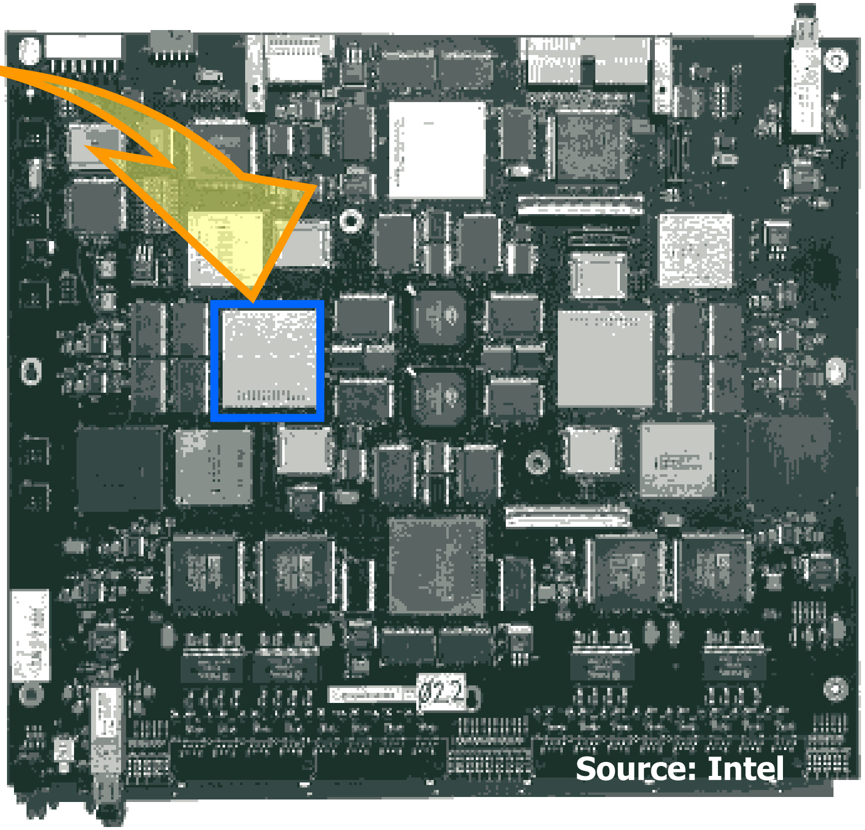
The Challenge of Board Testing



Source: Elcoteq

Tested chips placed on board

☝ Interconnects and soldering to be actually tested at board-level!



Source: Intel

Modeling of Interconnect Faults

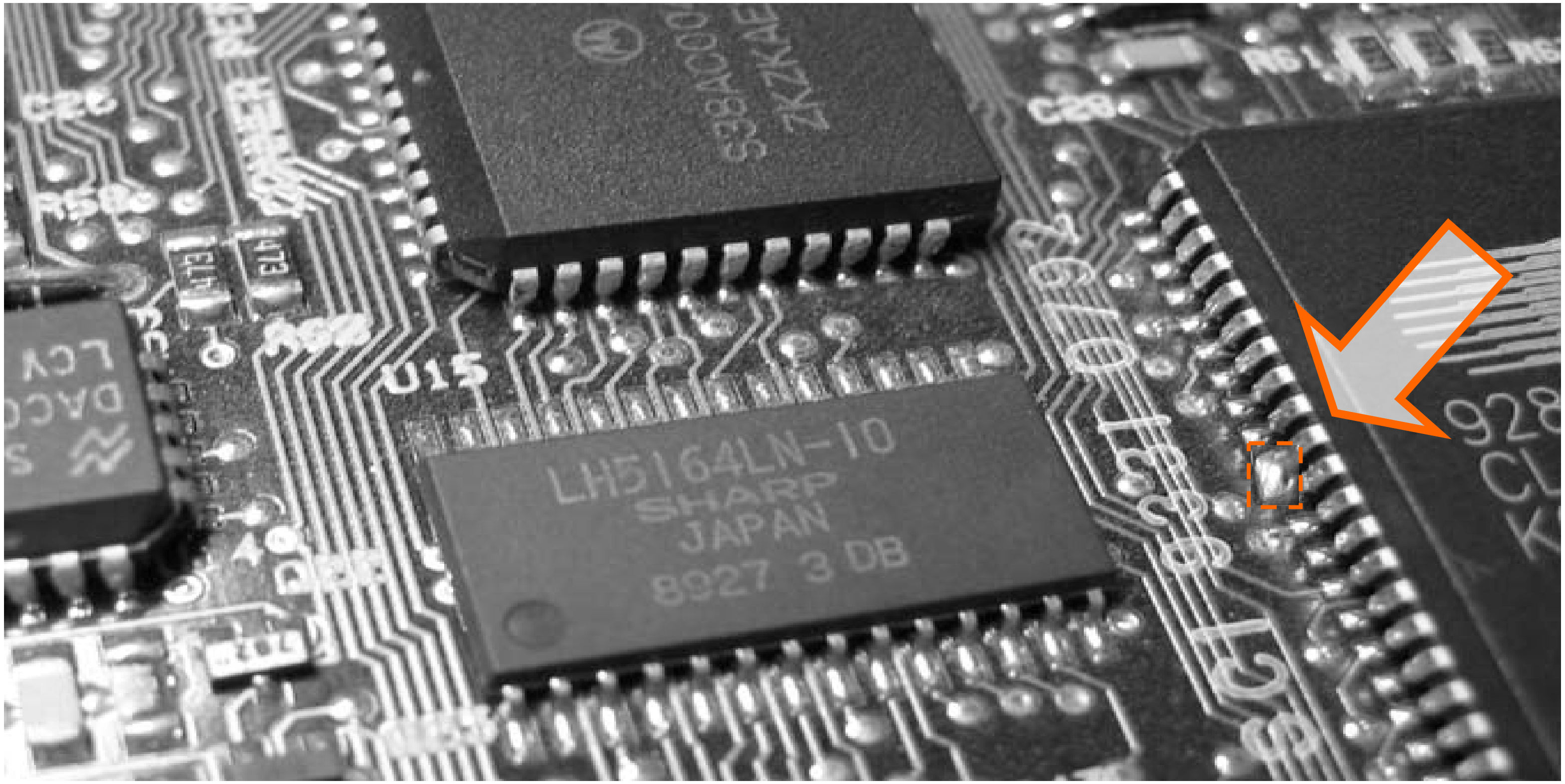
Net-level defect types and models

- Short faults
- Open faults
- Delay faults
- Noise/crosstalk
- Ground bounce
- …

Short faults, Open faults: static behavior

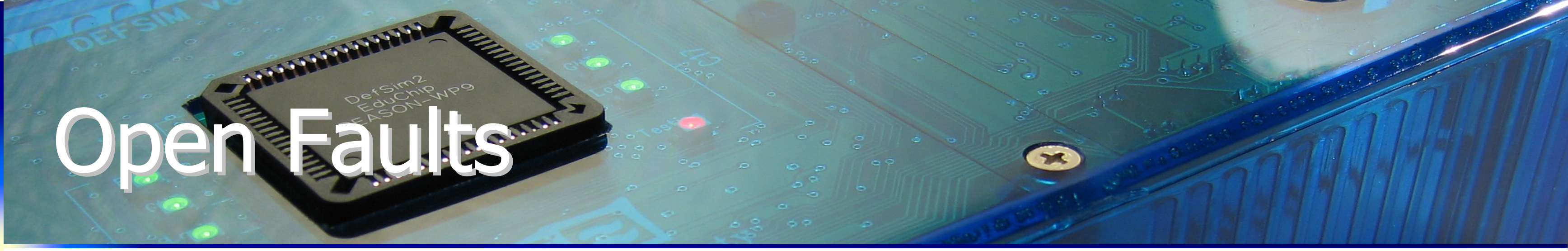
Open faults, Delay faults, Noise/crosstalk, Ground bounce: dynamic behavior

Short Faults



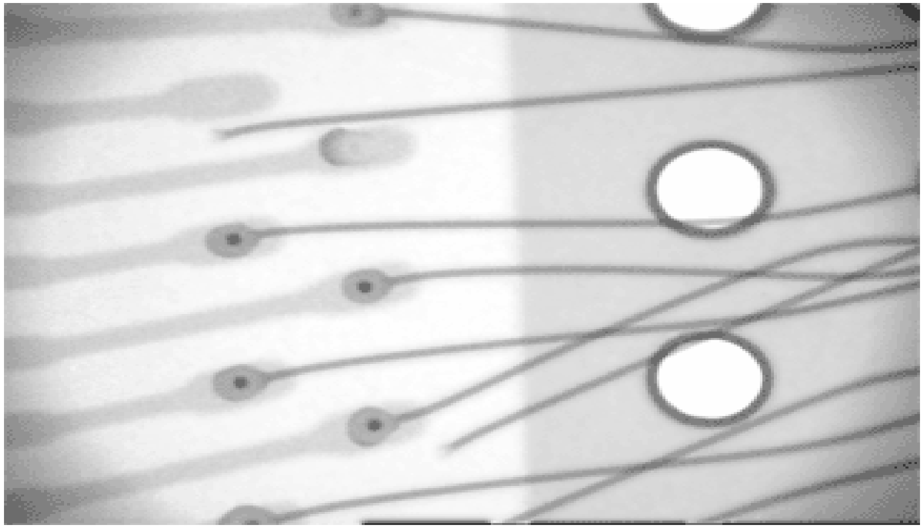
Possible shorts: bond wire, leg, solder, interconnect

Shorts are usually modeled as wired-AND, wired-OR faults

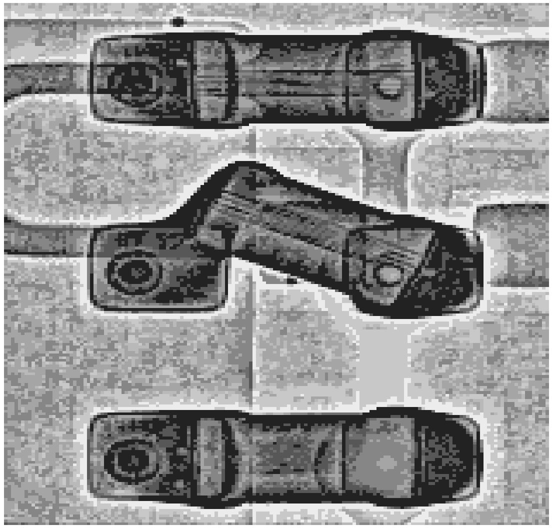


Open Faults

Misplaced bond wire



Misplaced component



Possible opens: bond wire, leg, solder, interconnect

Opens usually behave like stuck-at or delay faults

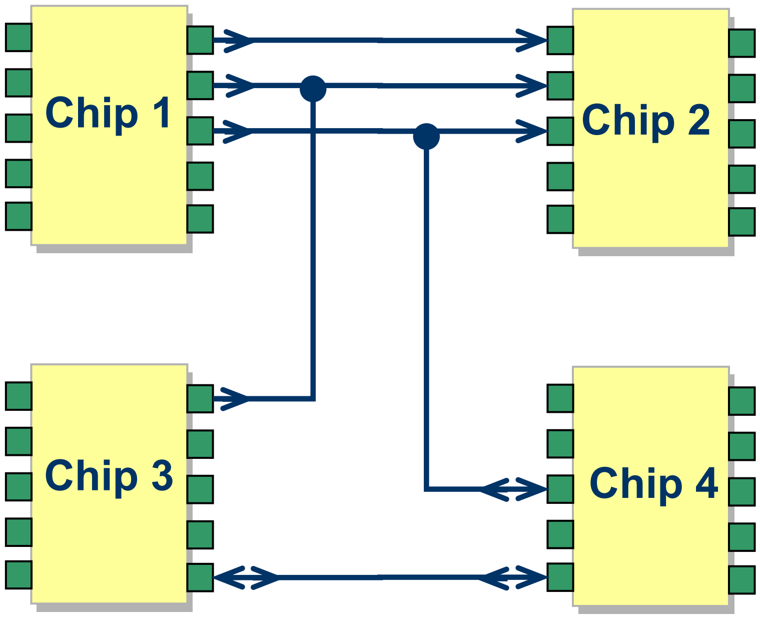
Tri-state connections

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.

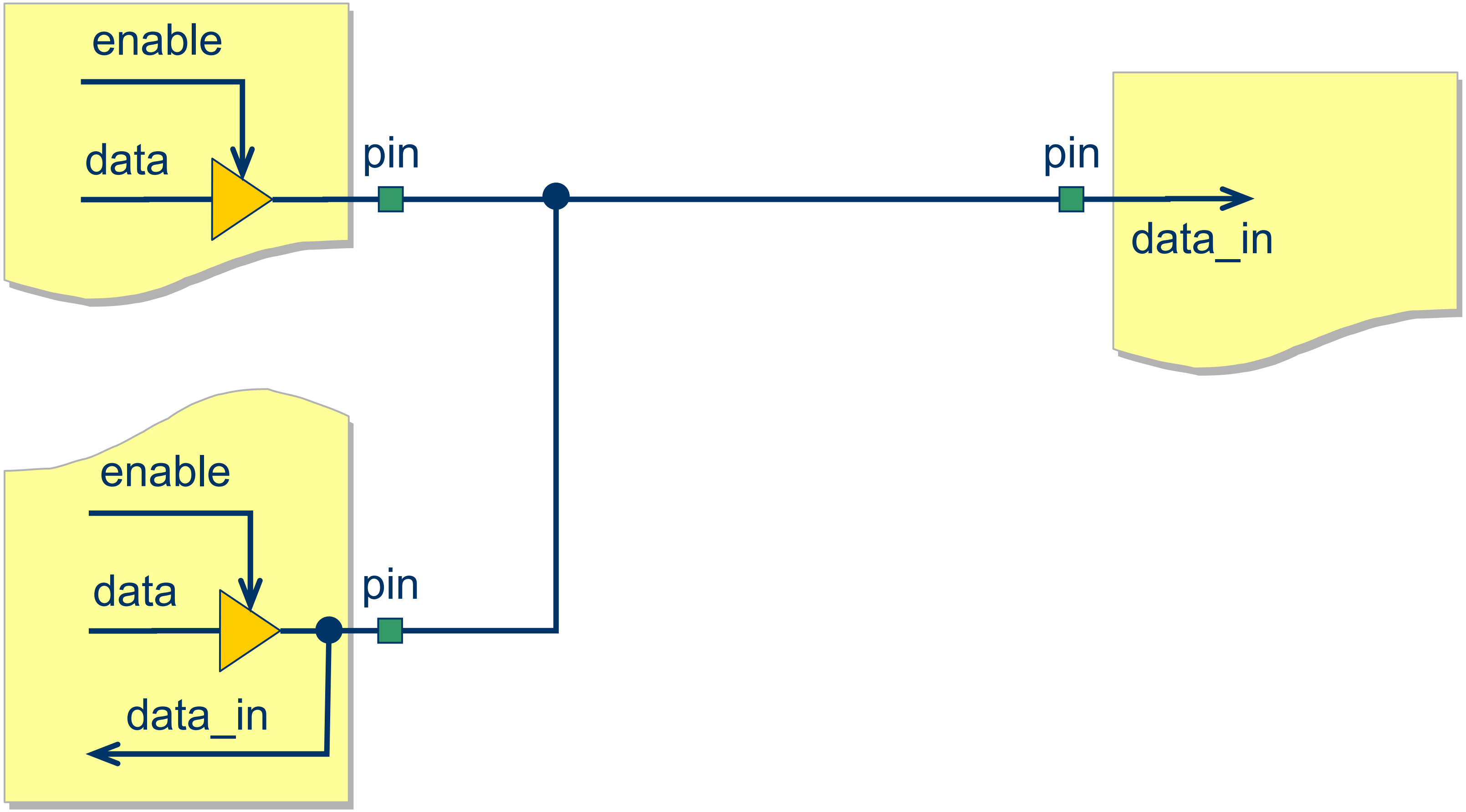
There are special “enable” signals that control this additional state of the I/O pins.

Tri-state connections



- Nets with several drivers
- Nets with bi-directional pins

Tri-state net structure



Specific faults in tri-state nets

Driver faults

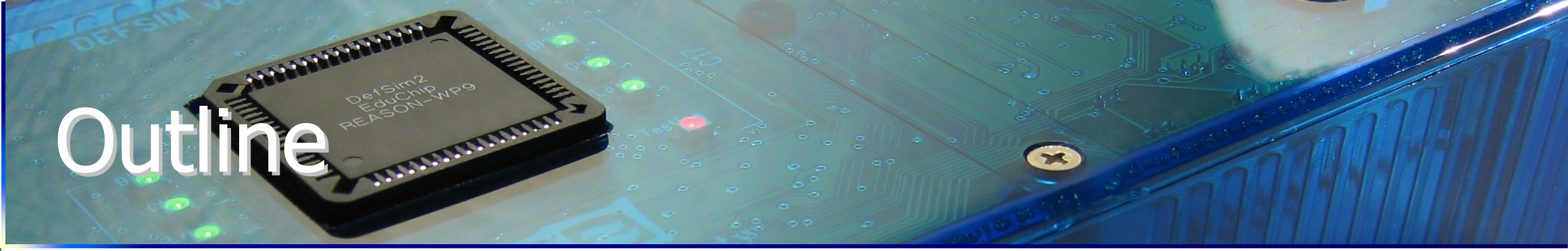
- stuck-driving fault
- stuck-not-driving fault
- stuck-at fault

Net opens

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

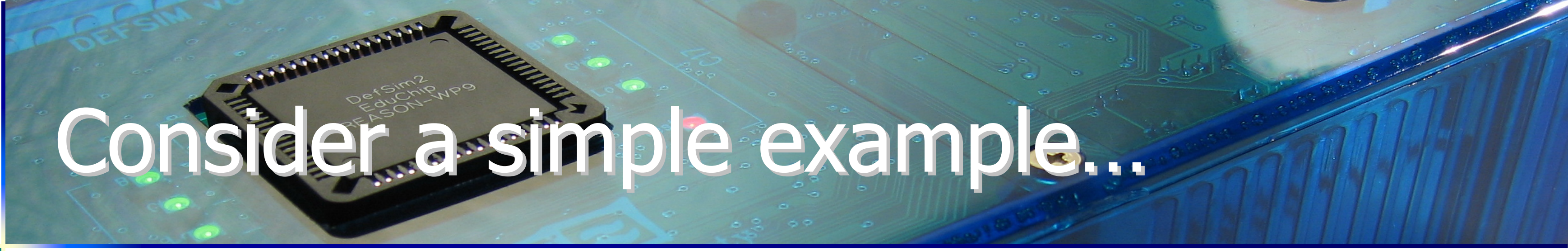
Net shorts

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



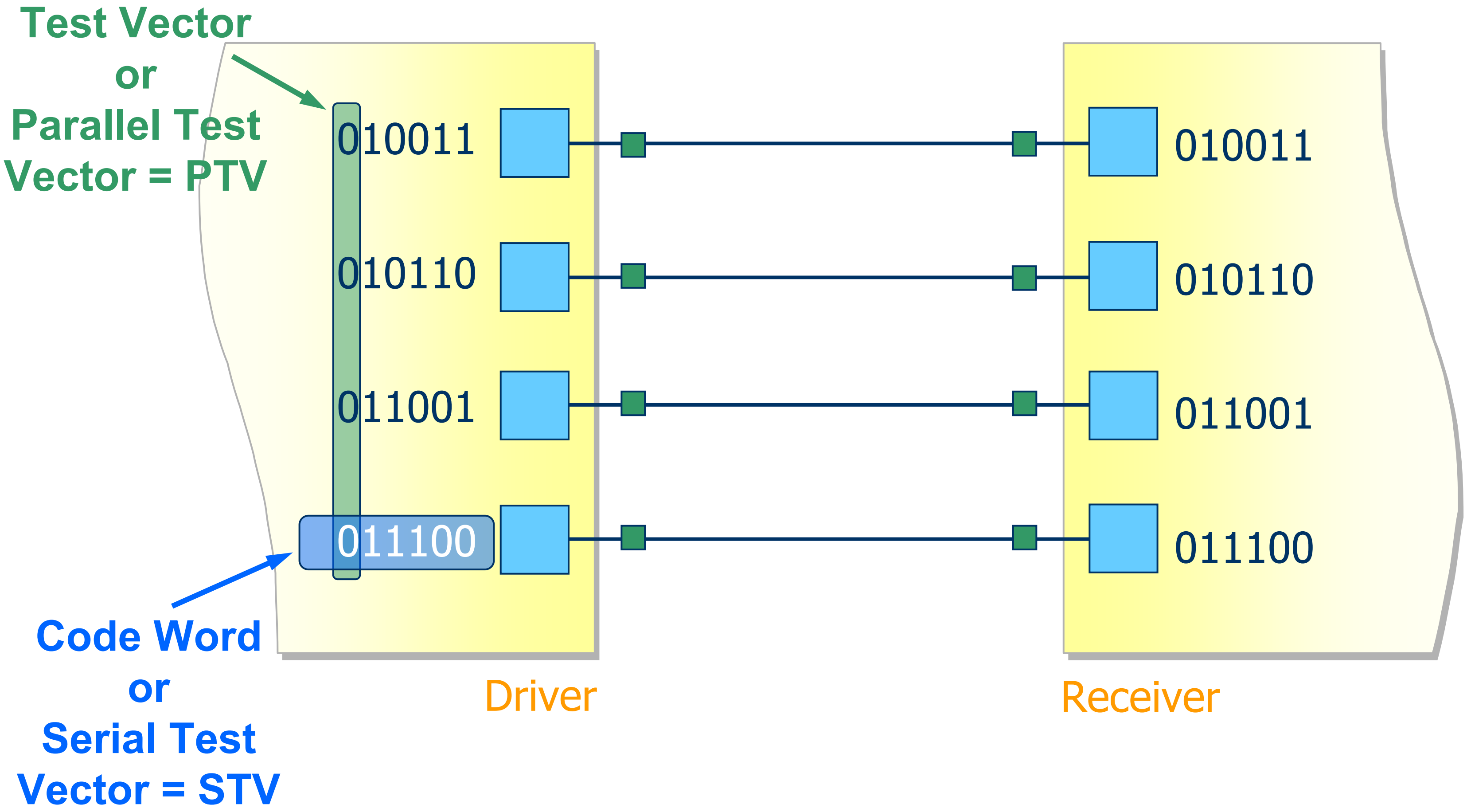
Outline

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



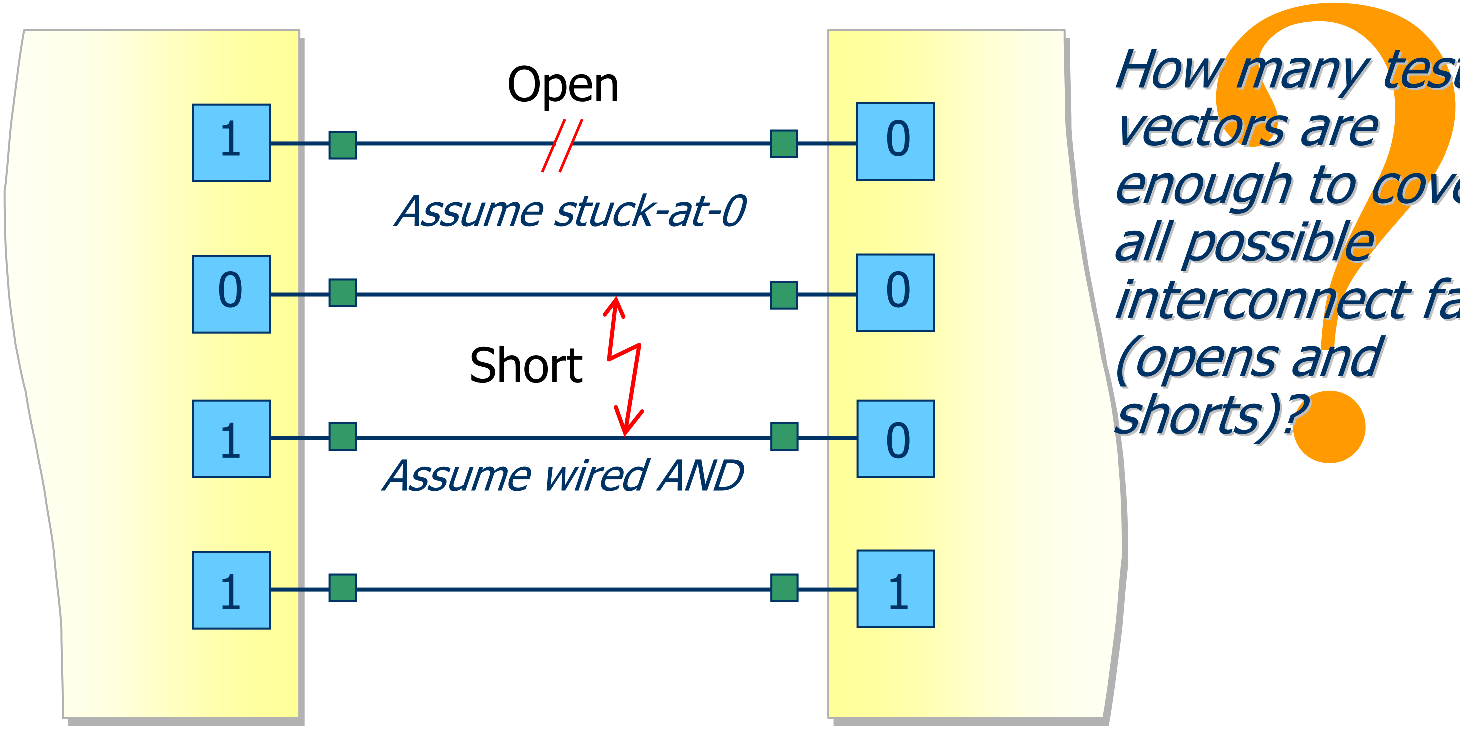
Consider a simple example…

… two circuits, 4 wires, plain topology



Test generation for interconnect faults

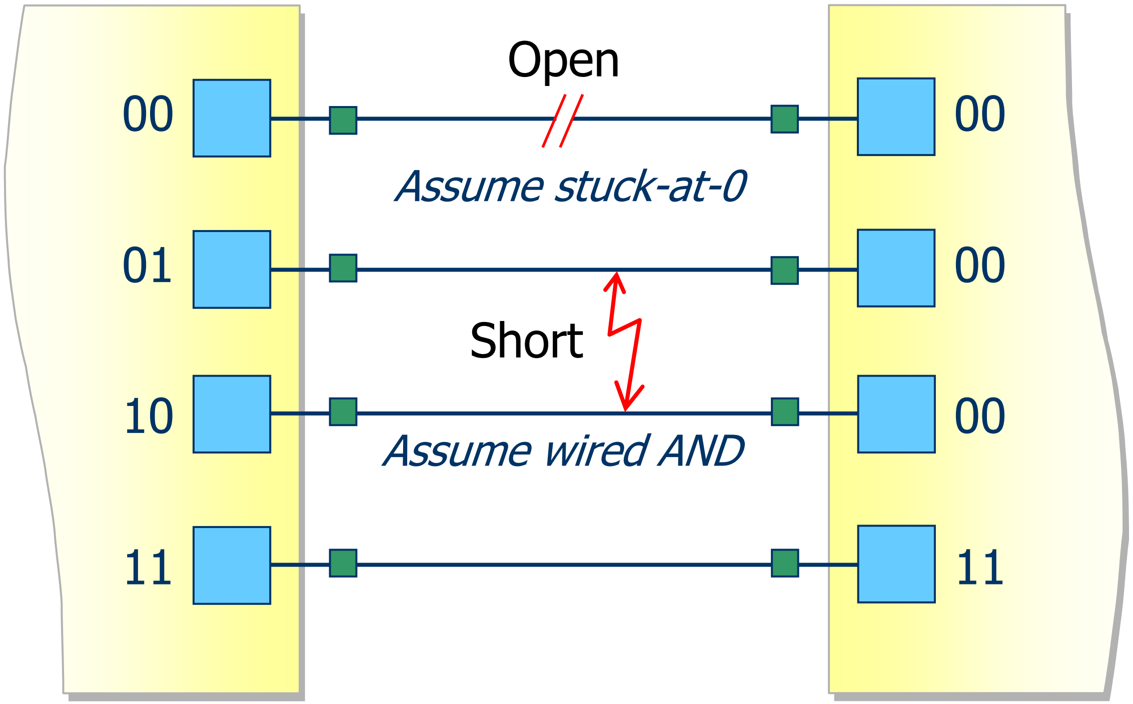
… two circuits, 4 wires, plain topology



How many test vectors are enough to cover all possible interconnect faults (opens and shorts)?

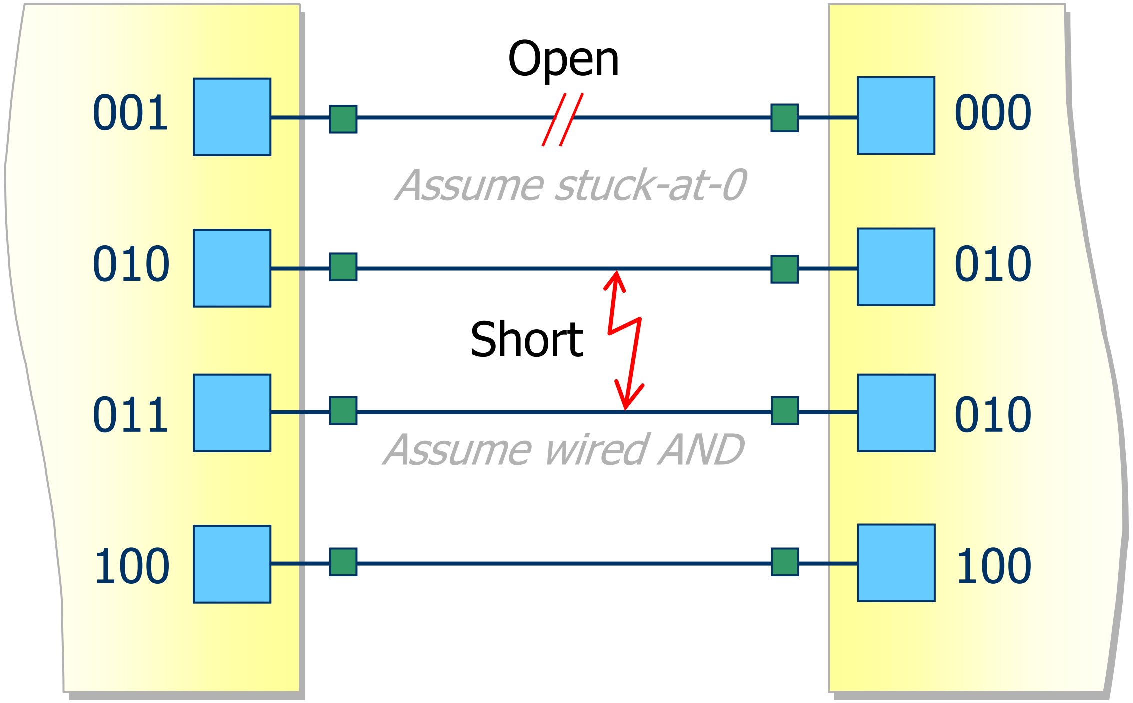
Opens usually behave like stuck-at or delay faults
Shorts are usually modeled as wired-AND or wired-OR

The Counting Sequence



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. Therefore the test length is $\lceil \log_2(N) \rceil$

The Modified Counting Sequence

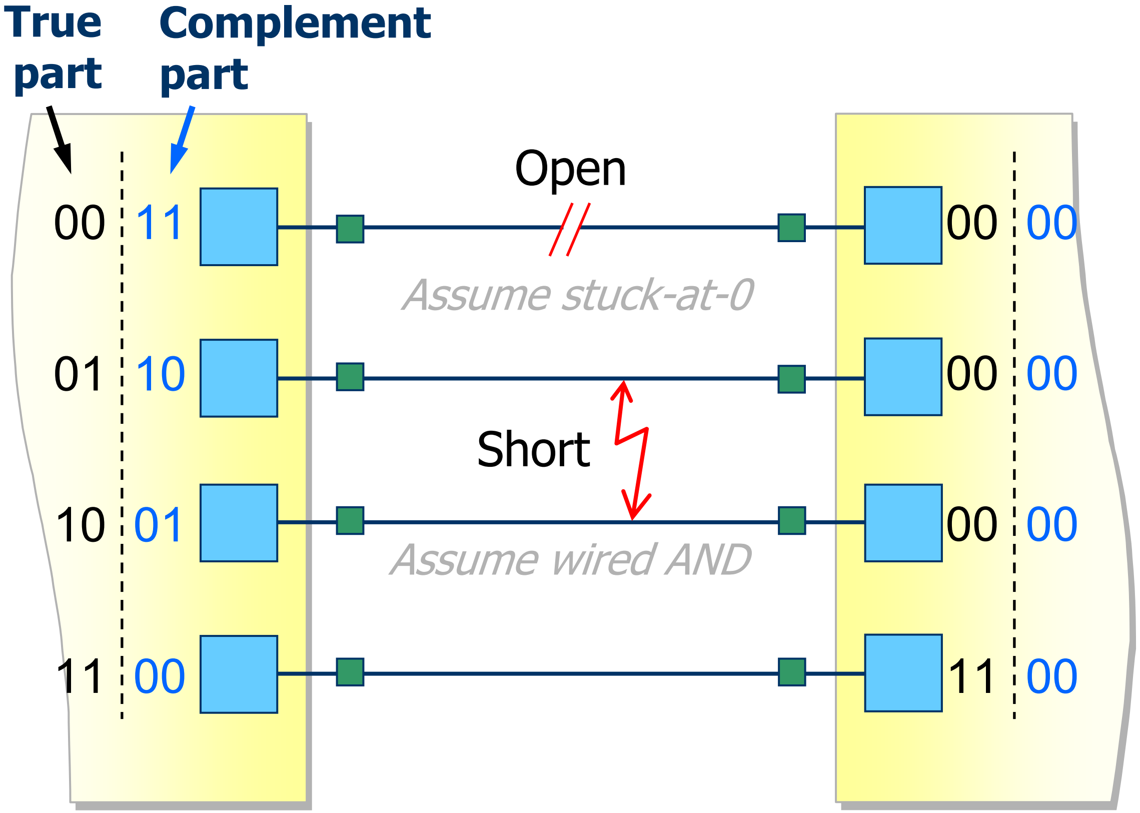


Some of the observed error responses are allowed or correct codes.

How to improve the diagnosis?

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

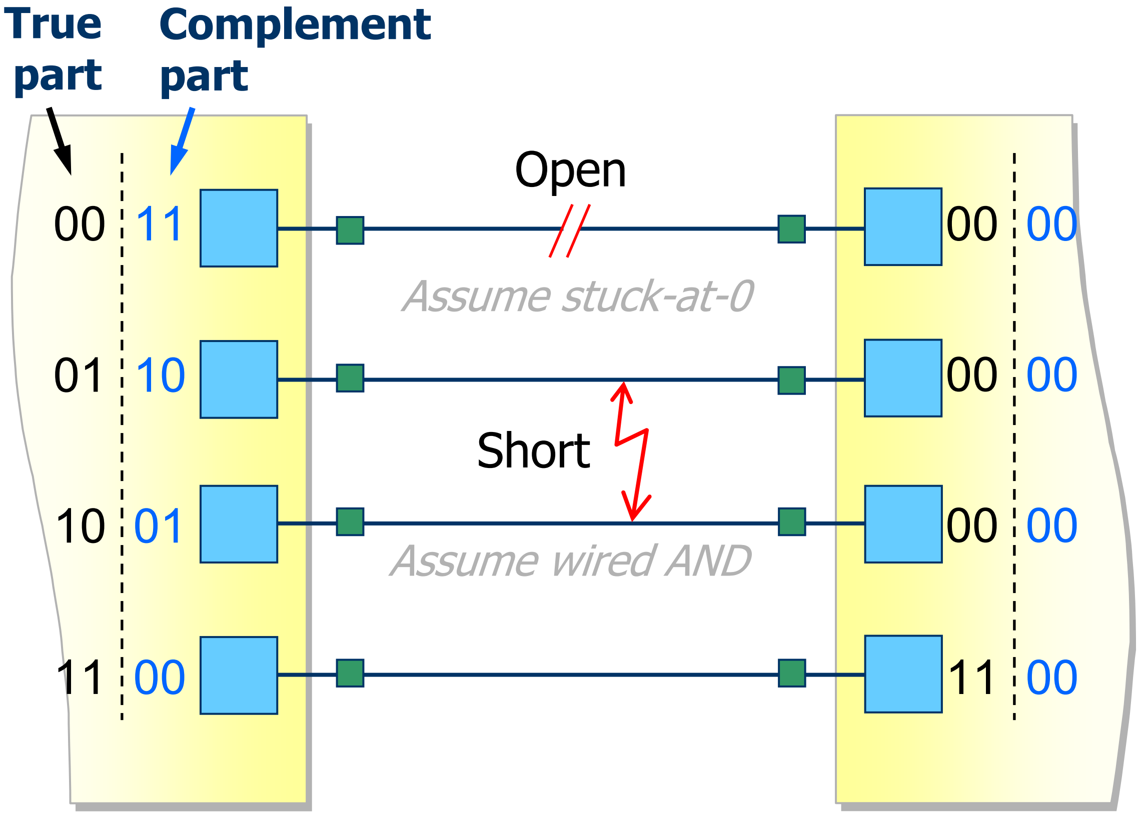
The True/Complement Code



All-0 and all-1 codes are not forbidden anymore!

To improve the diagnostic resolution Wagner proposed the True/Complement Code in 1987.
The test length became equal $2\lceil \log_2(N) \rceil$

The True/Complement Code

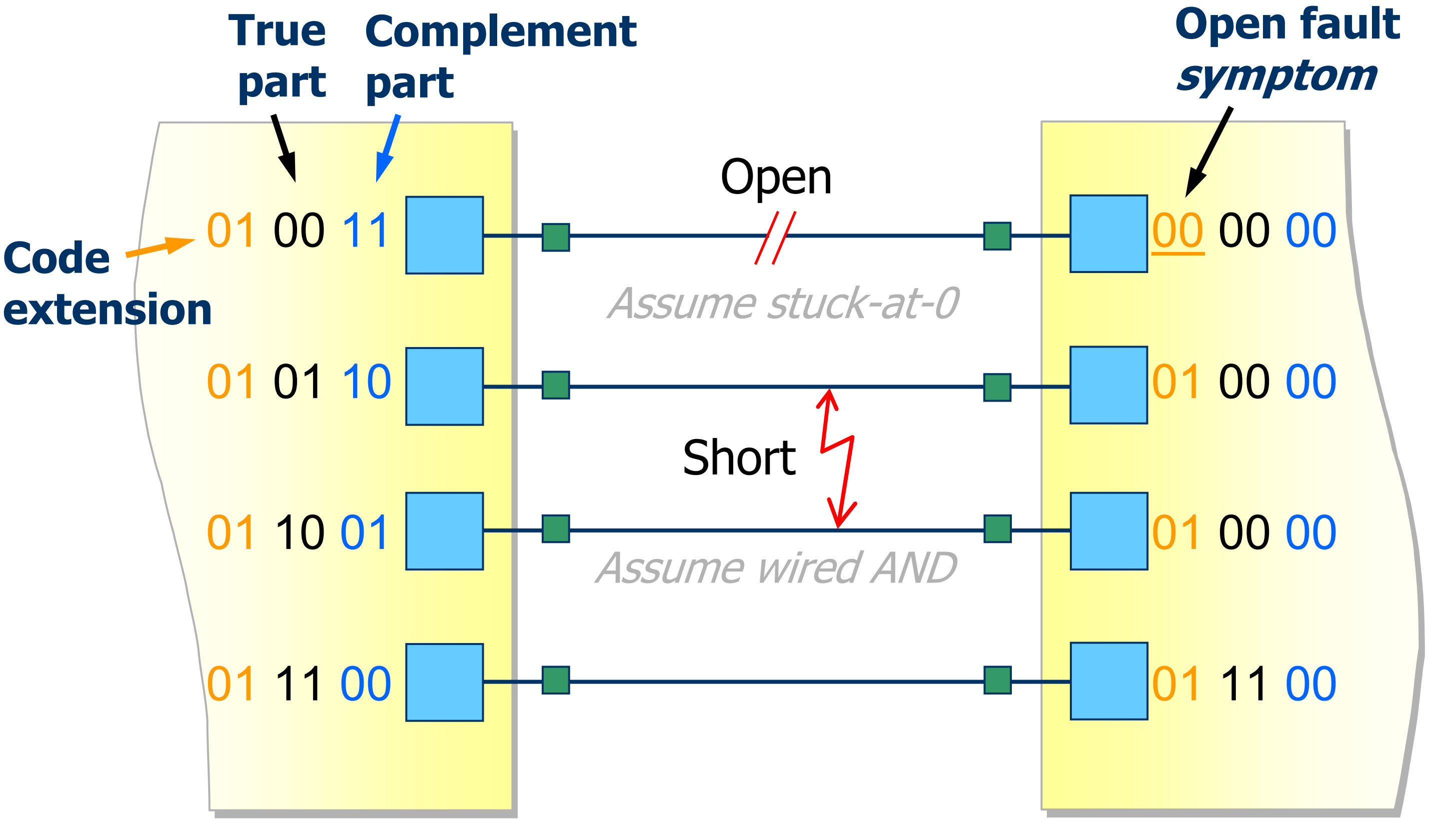


How to distinguish between opens and shorts?

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)

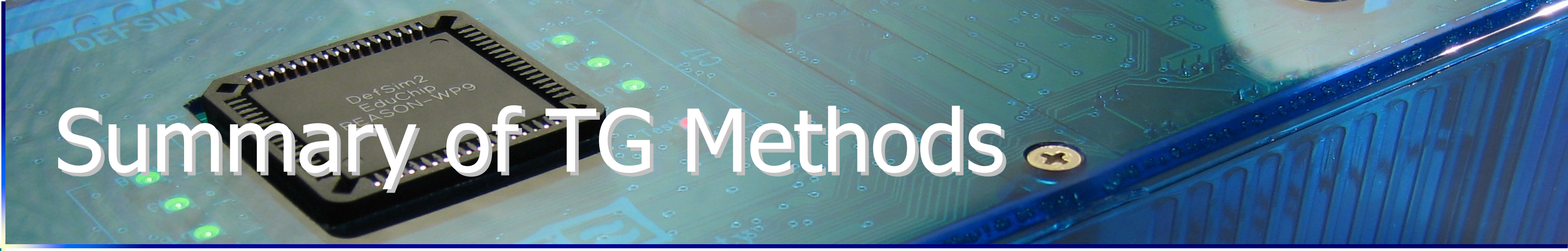
Extended True/Complement Code



Idea: add two bits, that are the same at every STV code word

Shorts and stuck-at faults are now *distinguishable*

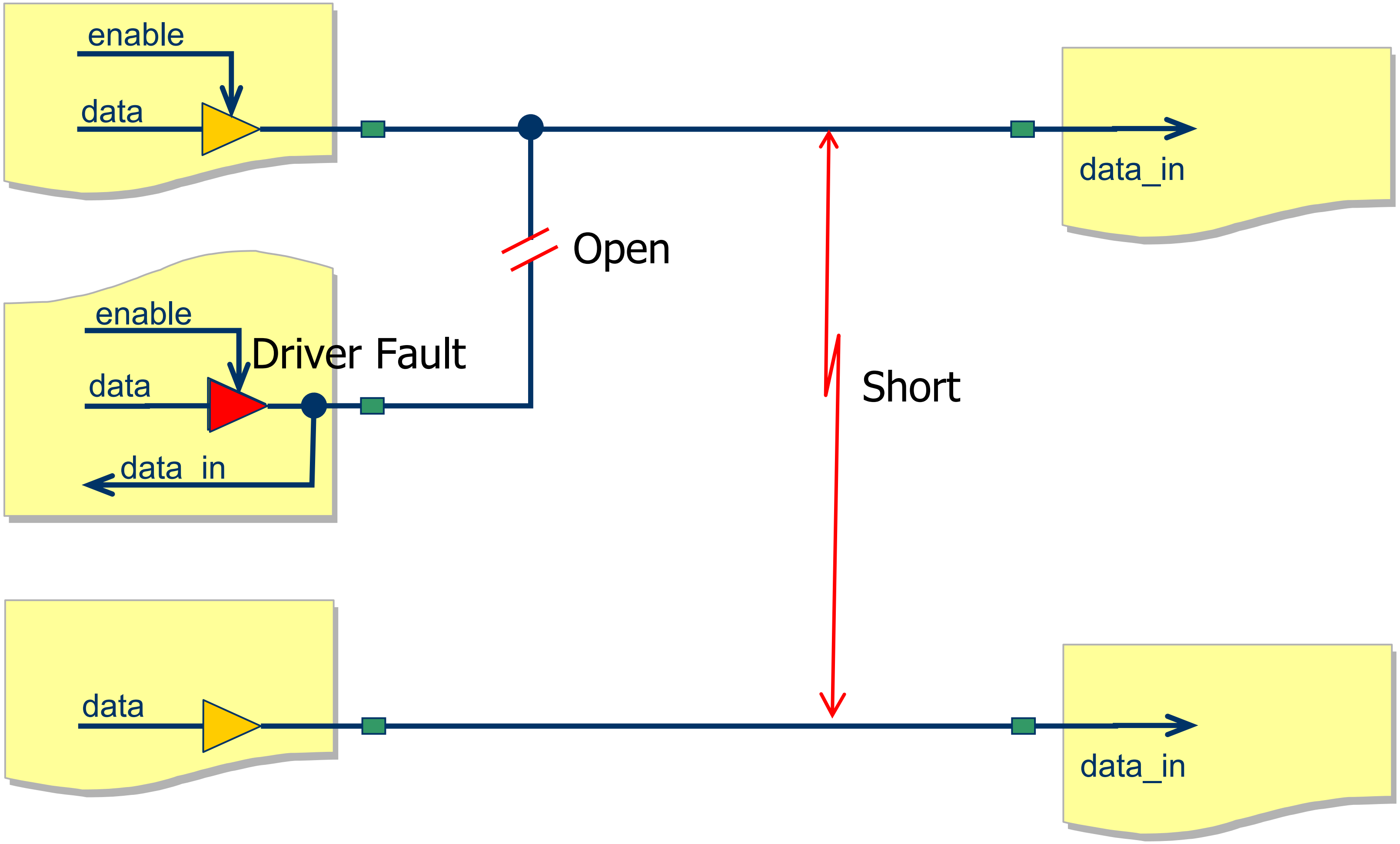
The test length is $2\lceil \log_2(N) \rceil + 2$

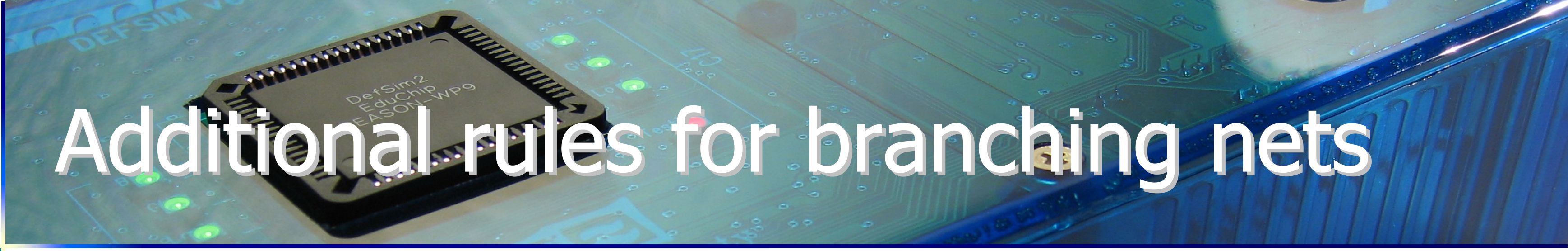


Summary of TG Methods

	Counting	Modified	True/Compl.	Extended	Walking	LaMa
	000 001 010 011 100 101 110 111	001 010 011 100 101 110	111 000 110 001 101 010 100 011 011 100 010 101 001 110 000 111	01 111 000 01 110 001 01 101 010 01 100 011 01 011 100 01 010 101 01 001 110 01 000 111	10000000 01000000 00100000 00010000 00001000 00000100 00000010 00000001	00001 00100 00111 01010 01101 10001 …
Length	$\lceil \log_2(N) \rceil$	$\lceil \log_2(N+2) \rceil$	$2\lceil \log_2(N) \rceil$	$2\lceil \log_2(N) \rceil+2$	N	$\lceil \log_2(3N+2) \rceil$
Example (N=10000)	14	14	28	30	10000	15
Hamming distance	1	1	2	2	2	2
Defects	Shorts	Shorts Opens	Shorts Opens /Delays/	Shorts Opens Delays	Shorts Opens /Delays/	Shorts Opens
Diagnostic Properties	Bad	Bad	Good	Very Good	Very Good	Very Good

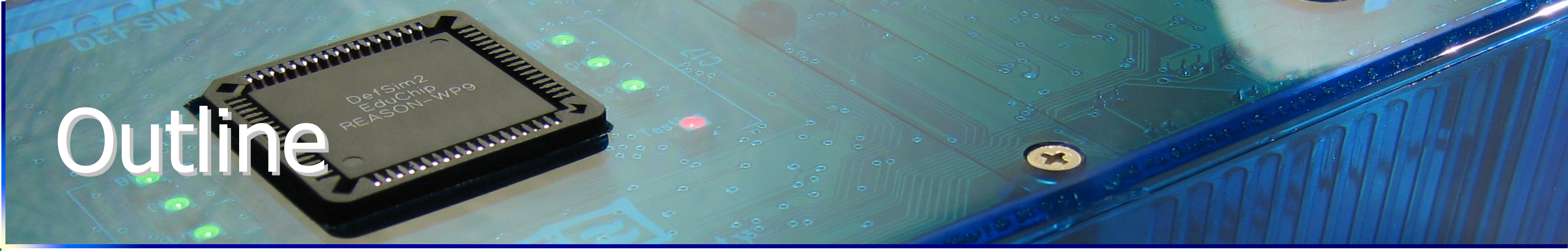
More complex case: branching nets





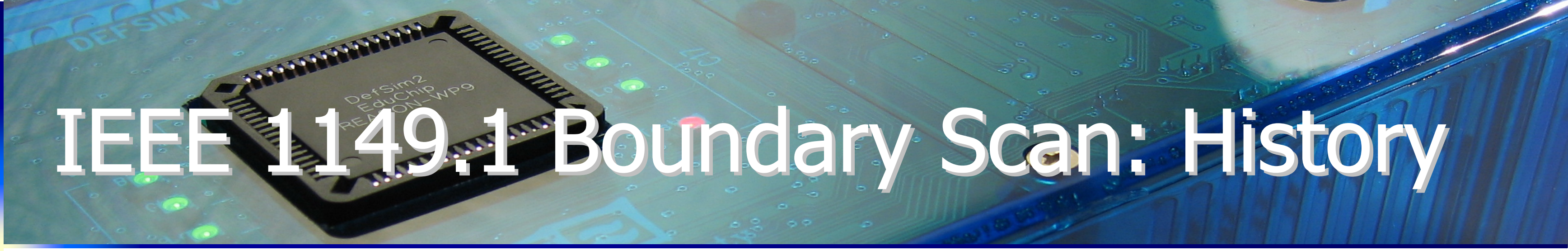
Additional rules for branching nets

- 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



Outline

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

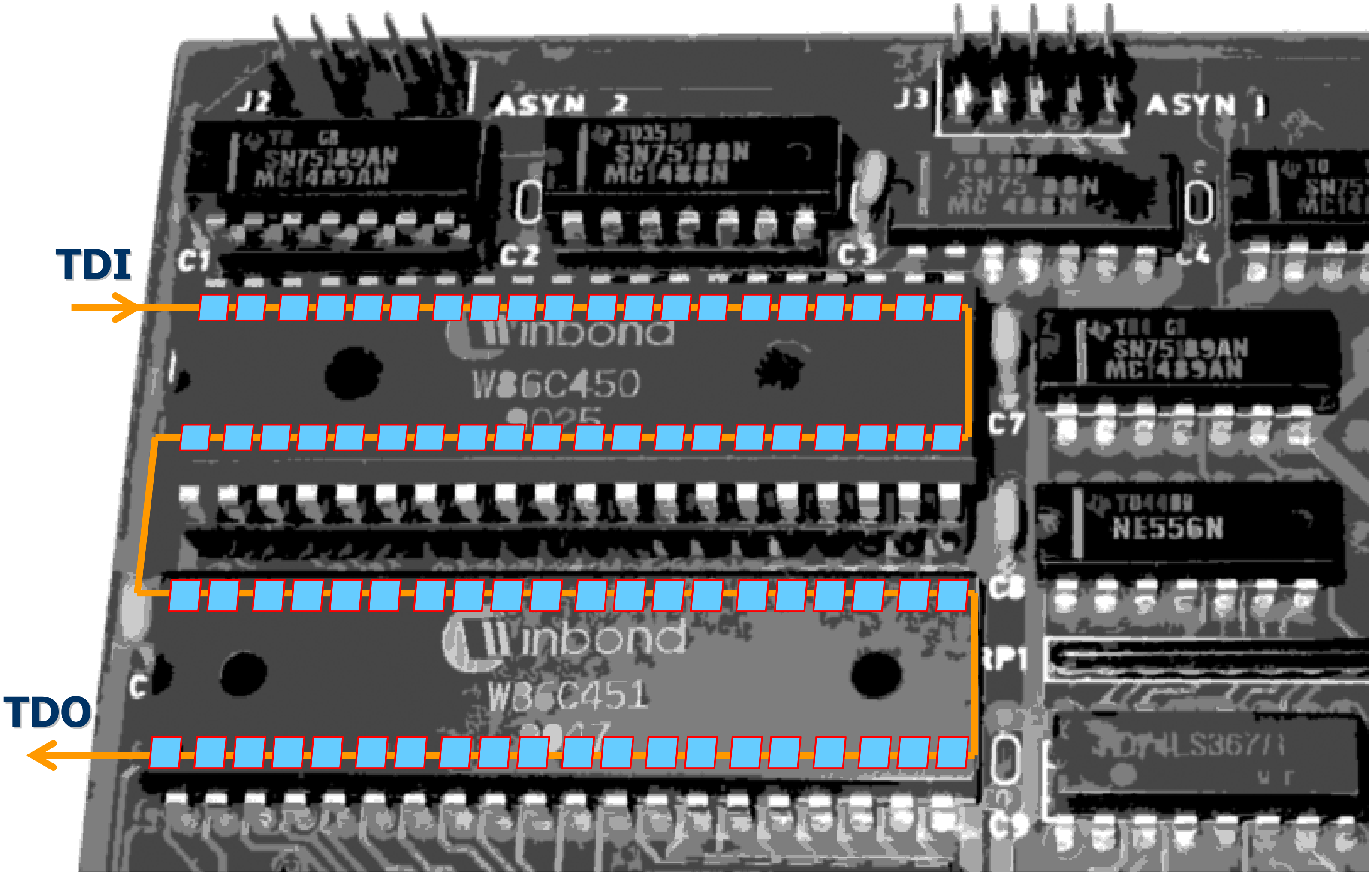


IEEE 1149.1 Boundary Scan: History

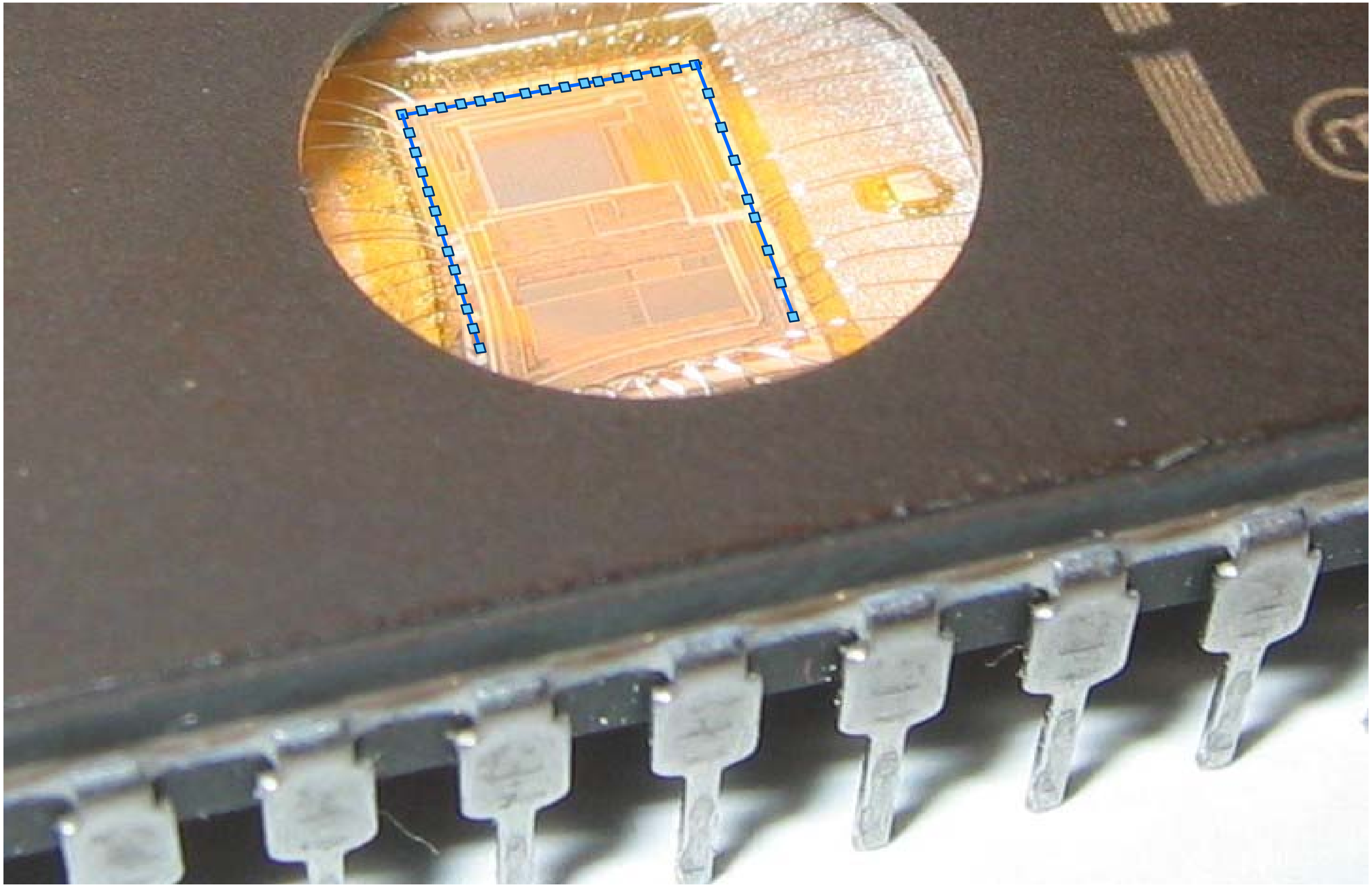
- Early 1980's – problem of test access to PCBs via "bed-of-nails" fixture
- Mid 1980's – Joint European Test Action Group (JETAG)
- 1986 – US companies involved: JETAG -> JTAG
- 1990 – JTAG Test Port became a standard [4]:

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

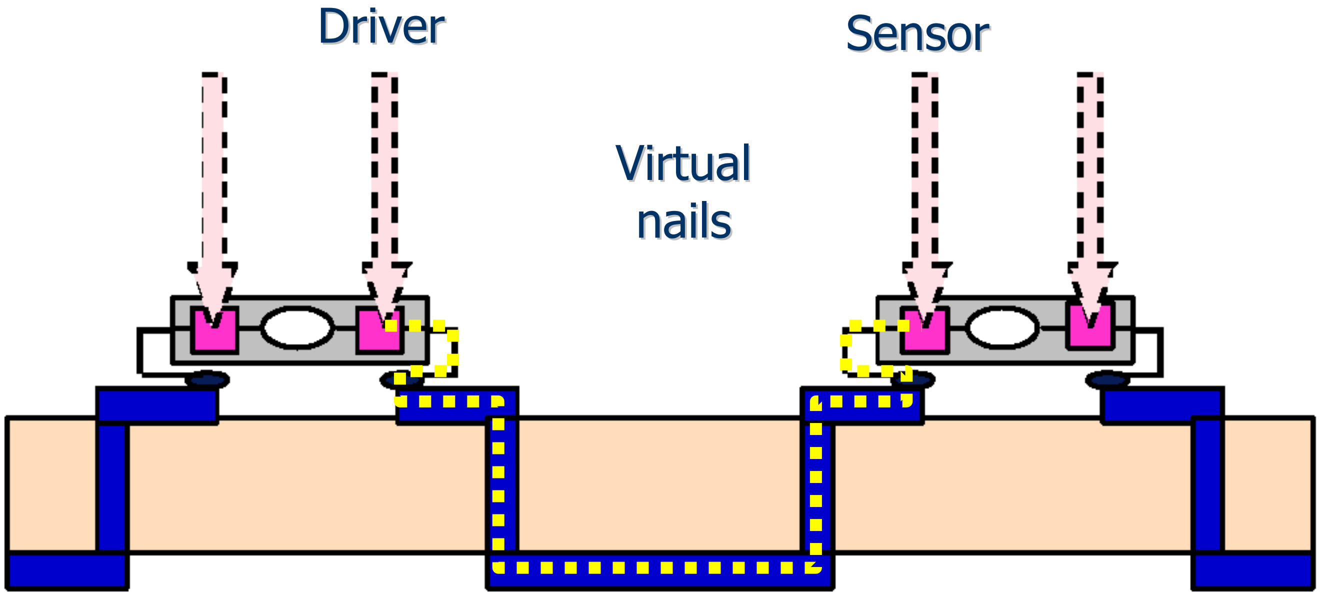
Test Access Via Boundary Scan



Test Access Via Boundary Scan



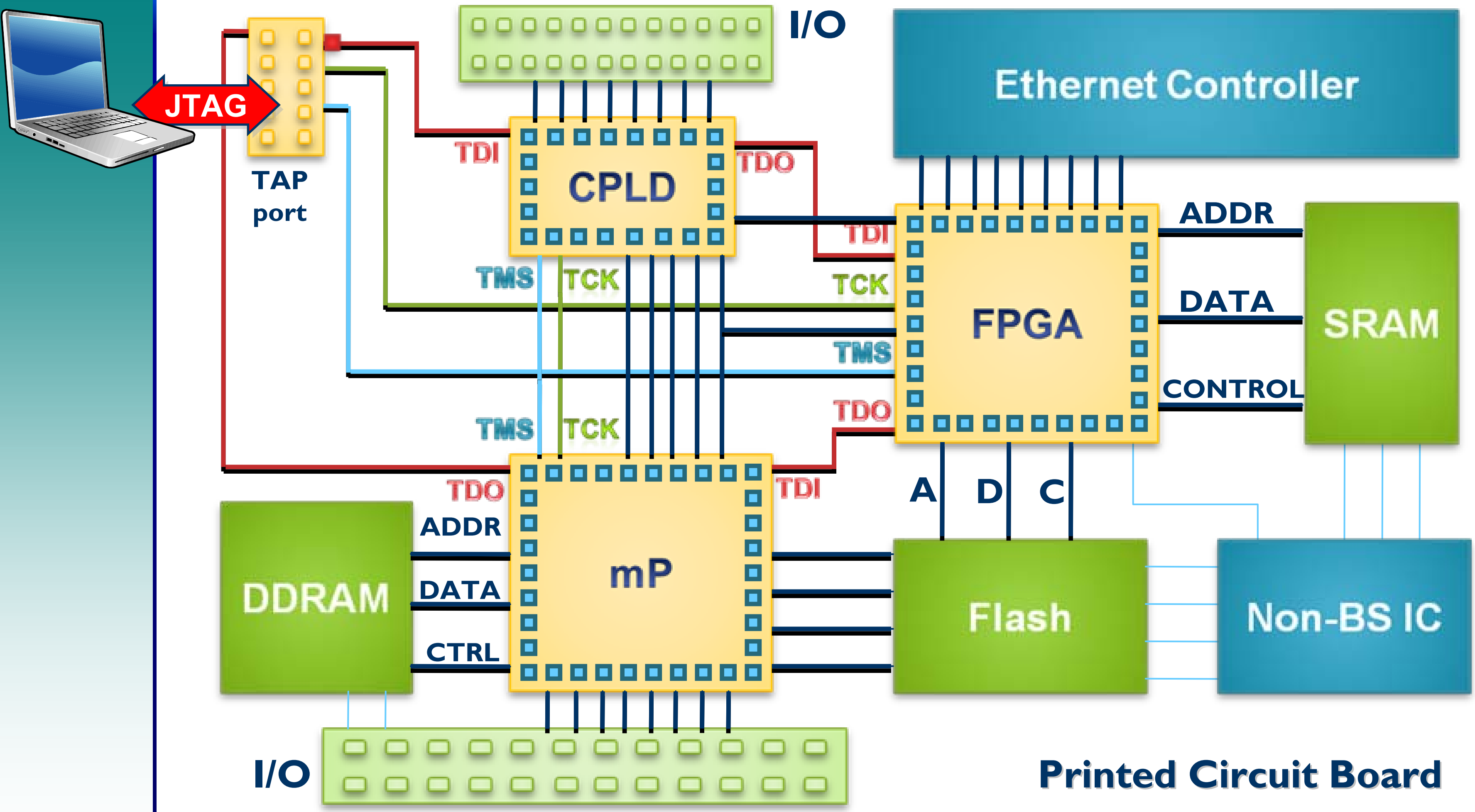
Test Access Via Boundary Scan



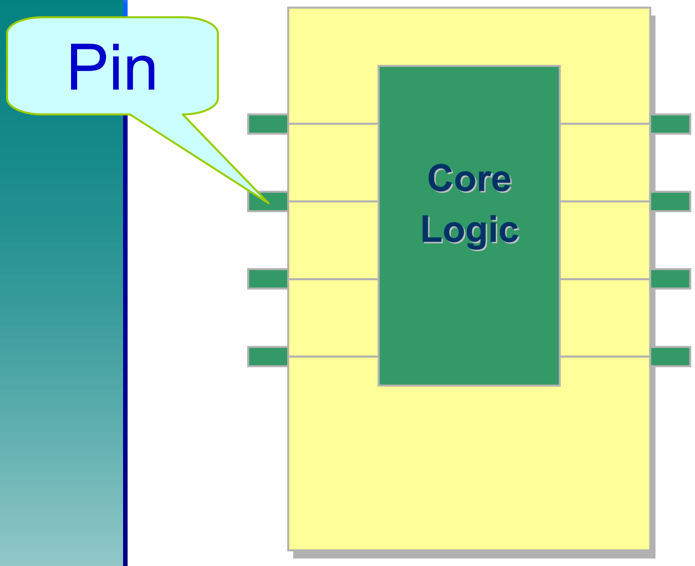
Defects covered:

driver scan cell, driver amp, bond wire, leg, solder, interconnect, solder, leg, bond wire, driver amp, sensor scan cell

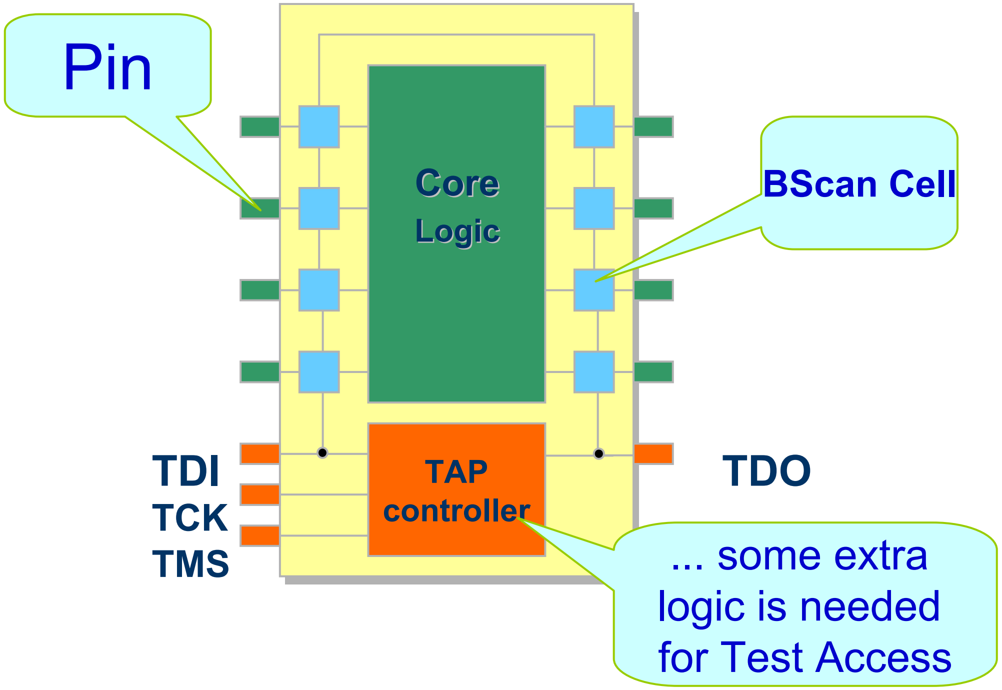
Test Access Via Boundary Scan



Boundary Scan basics



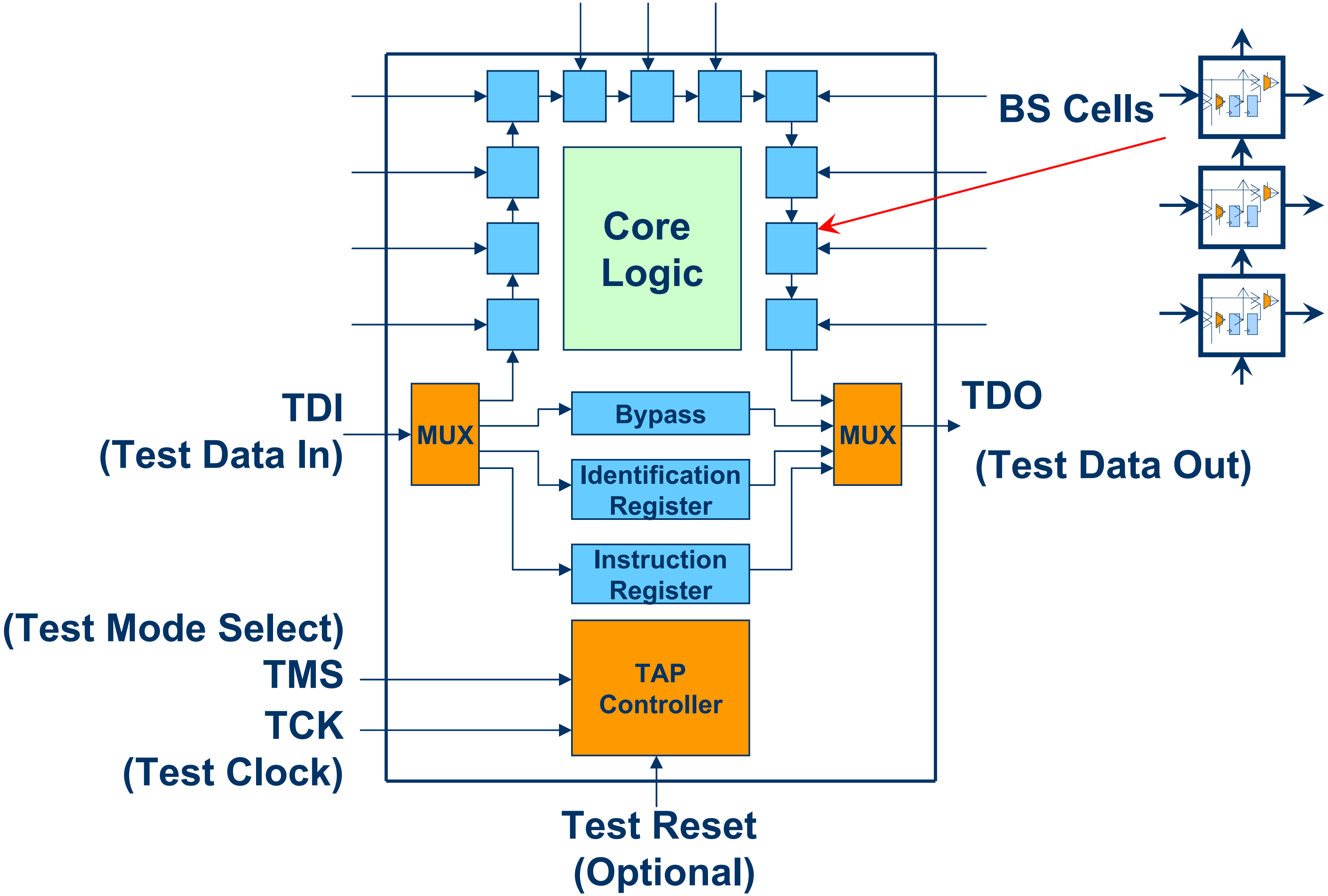
Non-BScan Device



BScan Device

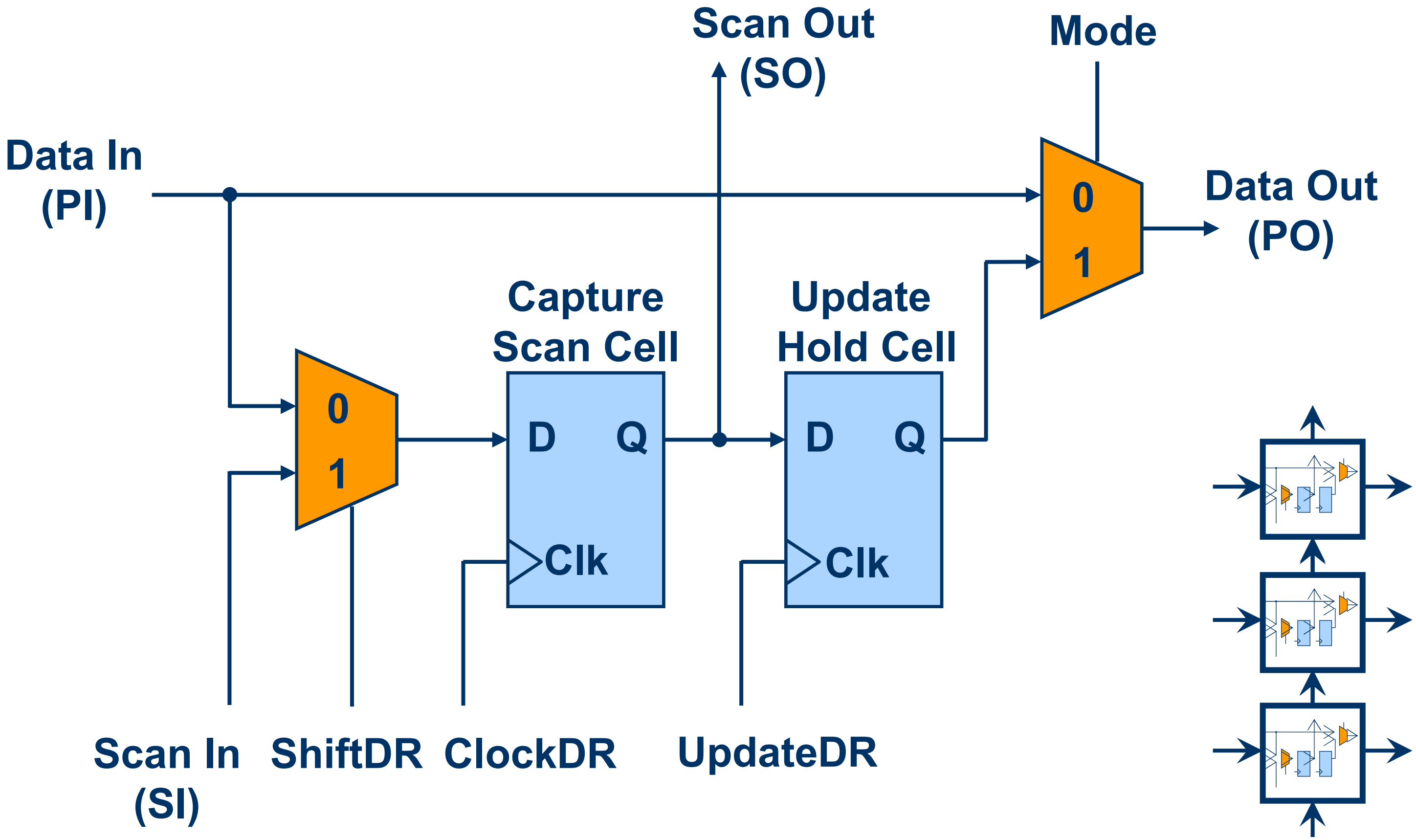
For describing Boundary Scan devices BSDL (Boundary Scan Description Language) models are used

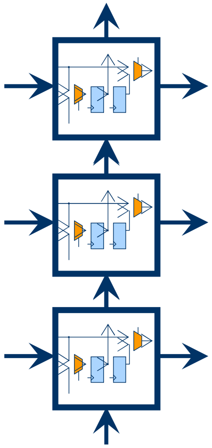
IEEE 1149.1 Device Architecture

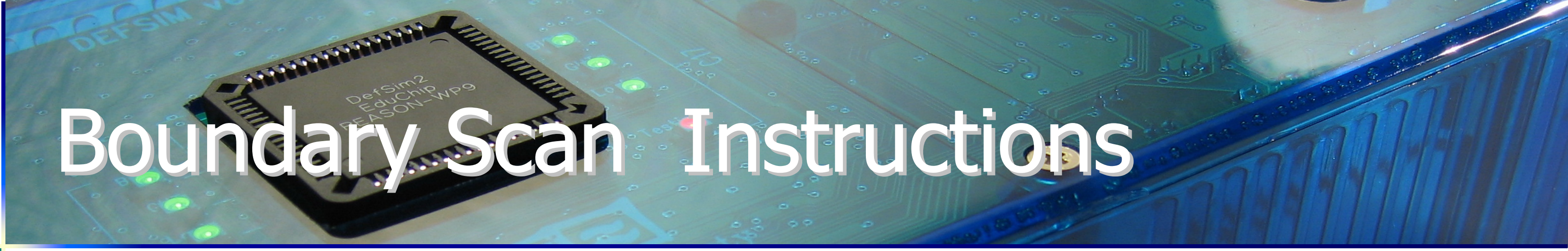


Typical Boundary Scan Cell (BC_1)

BC_1 is used both at input and output pins





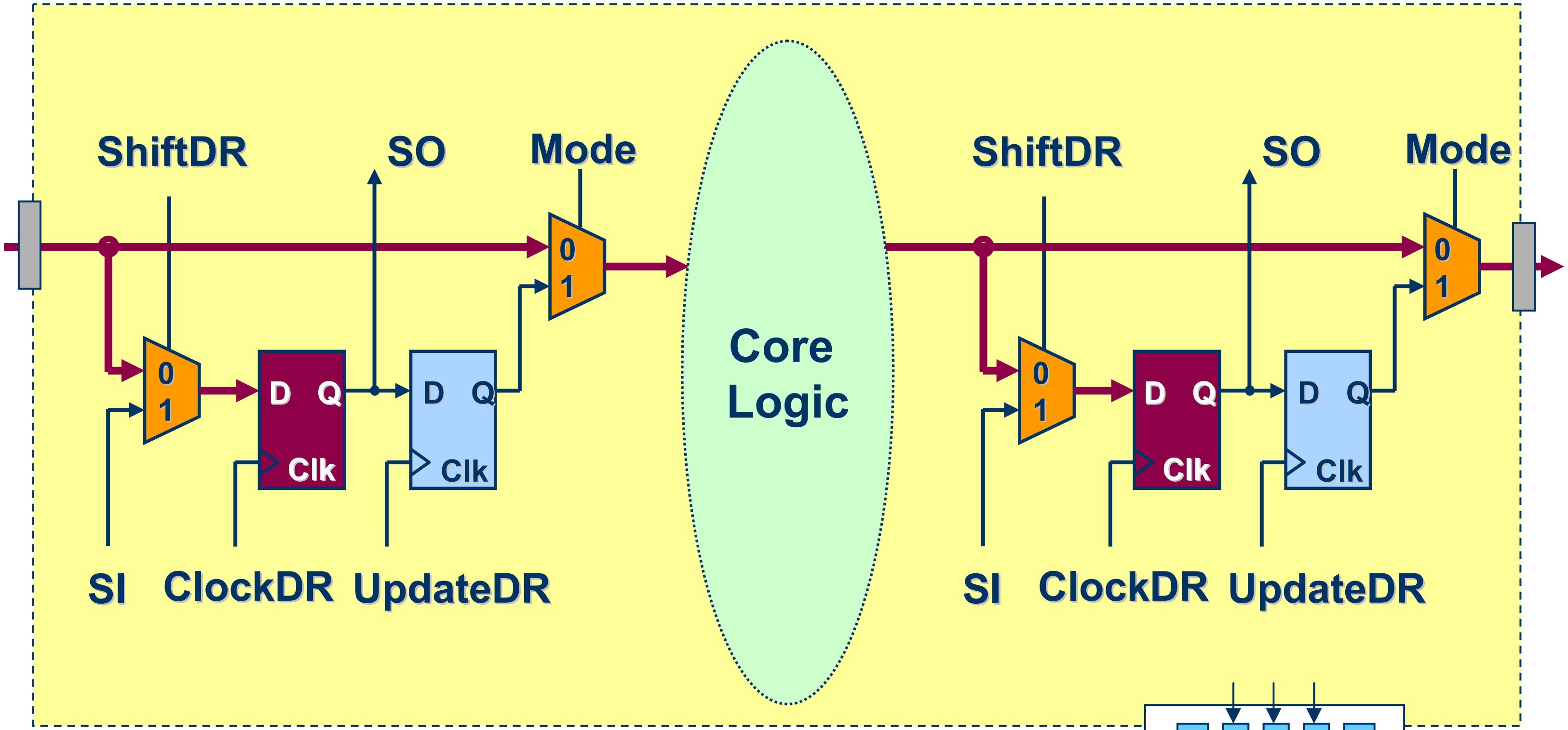


Boundary Scan Instructions

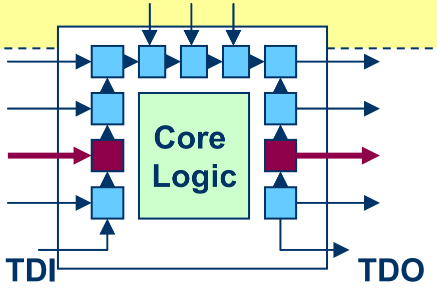
Instruction	Status
SAMPLE /PRELOAD	Mandatory
EXTEST	Mandatory
BYPASS	Mandatory
IDCODE	Optional
INTEST	Optional
CLAMP	Optional
HIGHZ	Optional
RUNBIST	Optional
USERCODE	Optional

Boundary Scan Working Modes

SAMPLE/PRELOAD instruction – sample mode

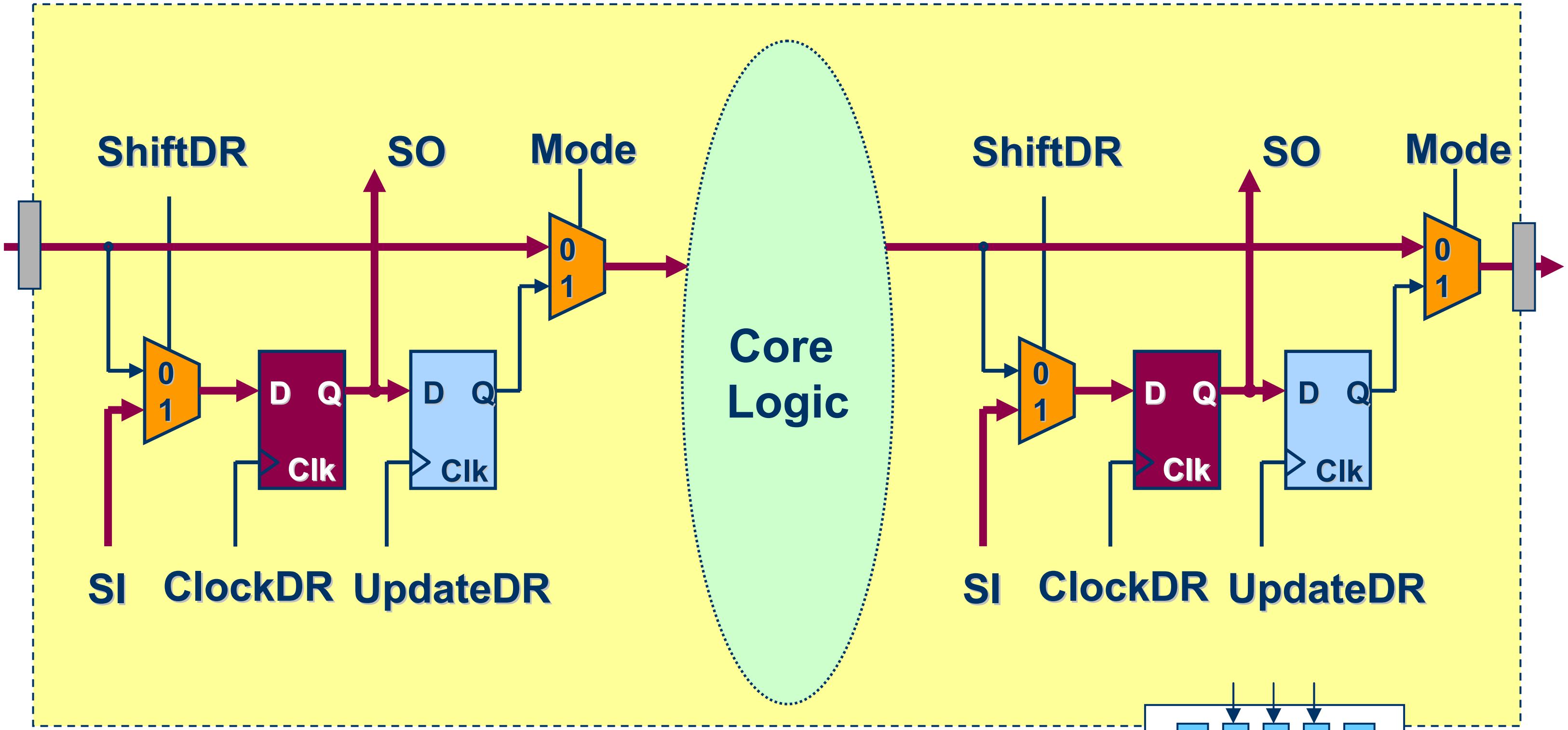


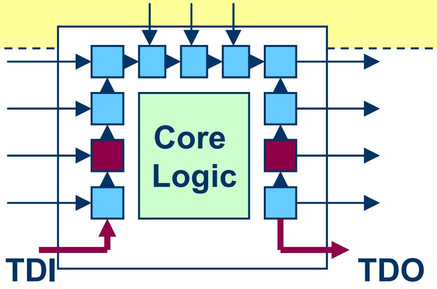
Get snapshot of normal chip output signals



Boundary Scan Working Modes

SAMPLE/PRELOAD instruction – preload mode

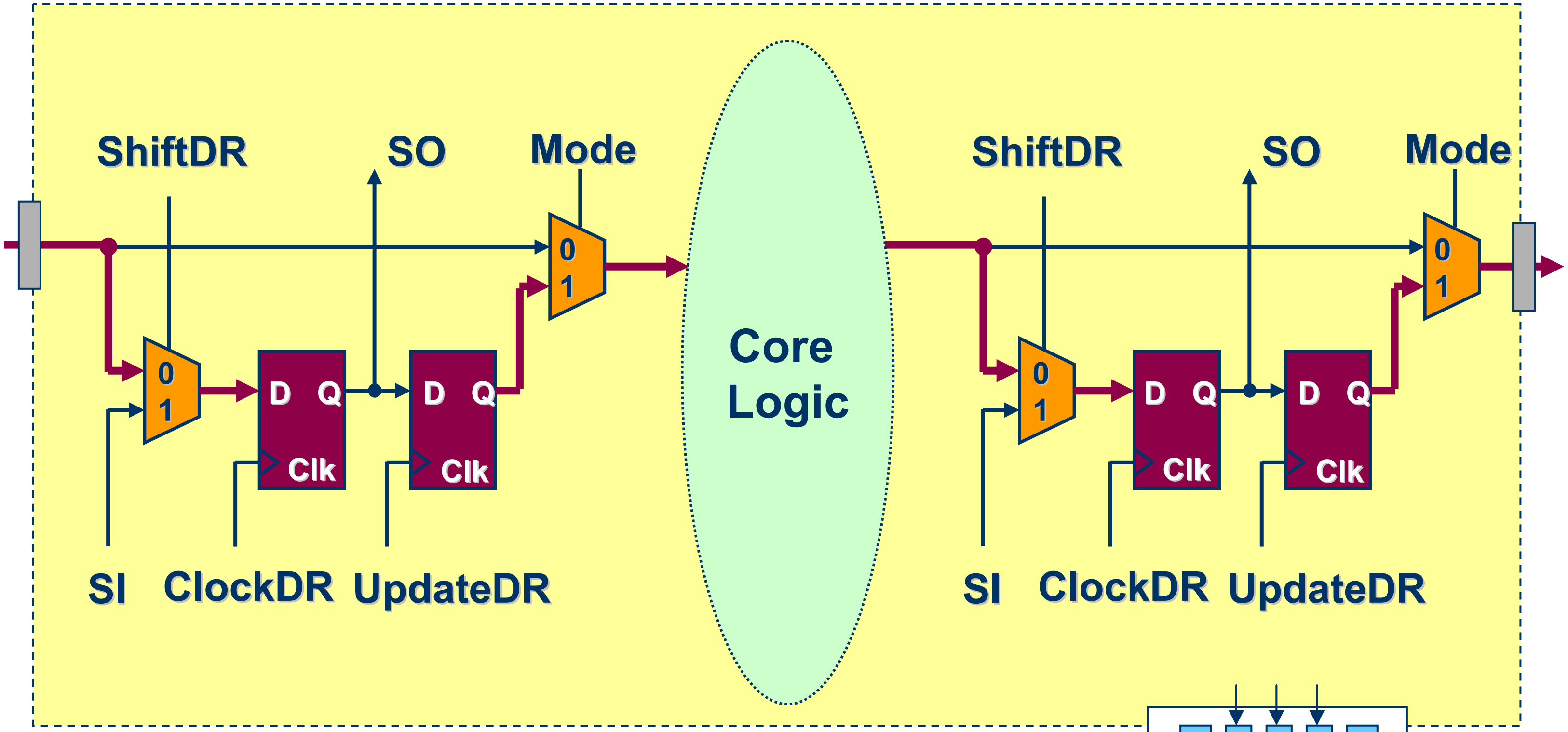




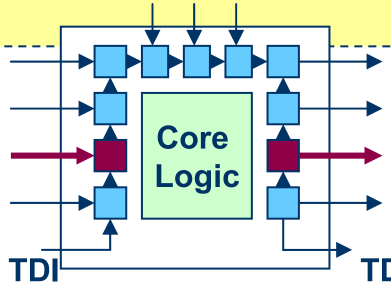
Shift out snapshot data and shift in new test data to be used later

Boundary Scan Working Modes

EXTEST instruction – driving and sensing:

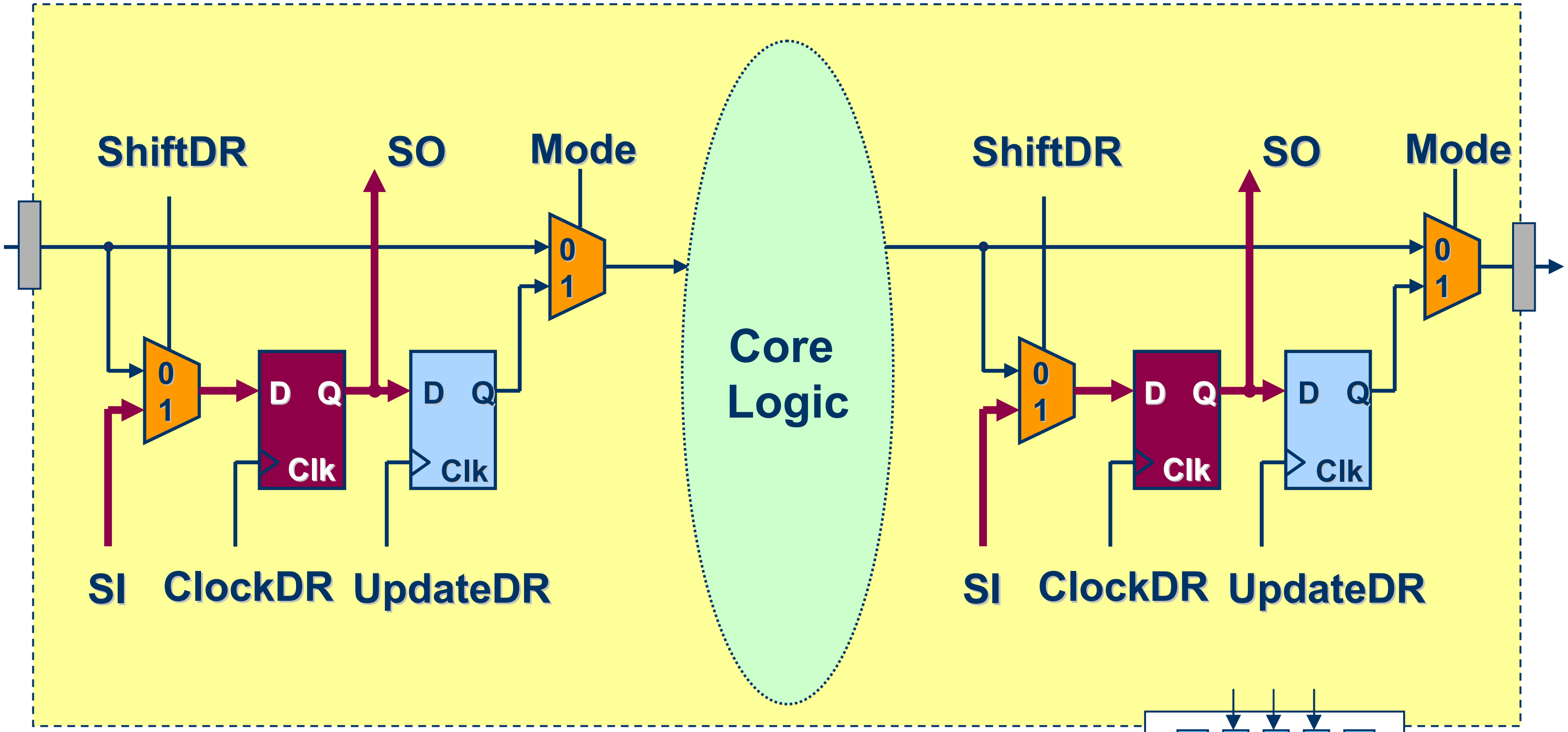


Test off-chip circuits and board-level interconnections

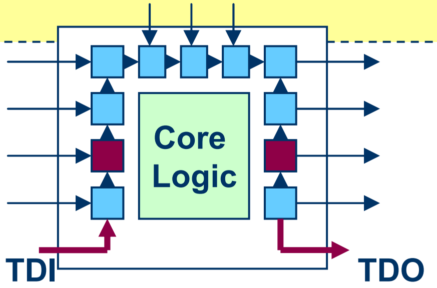


Boundary Scan Working Modes

EXTEST instruction – shifting



Shift out snapshot data and shift in new test data to be used later



Typical BS Interconnect Test Flow

BS mode	Test bus actions	Test data manipulations	Test
PRELOAD	IRshift + DRshift	Loading the first test vector to BS register (vector includes control/disable values for other devices on the bus)	Vector 1 loaded
EXTEST	IRshift + DRshift	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	Vector 1 applied and analyzed
EXTEST	DRshift	1. Applying vector 2 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	Vector 2 applied and analyzed
EXTEST	DRshift	1. Applying vector *N* to the DUT 2. Capturing test responses from DUT in BS reg. 3. Reading back test responses	Vector *N* applied and analyzed

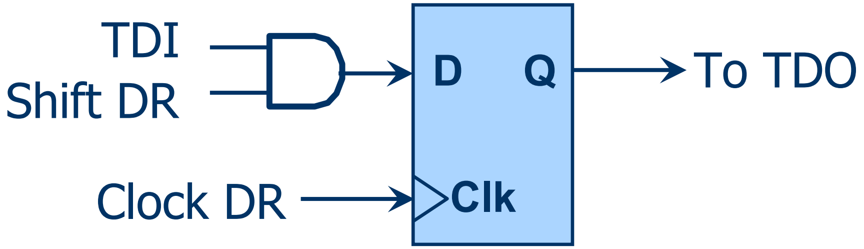
Time

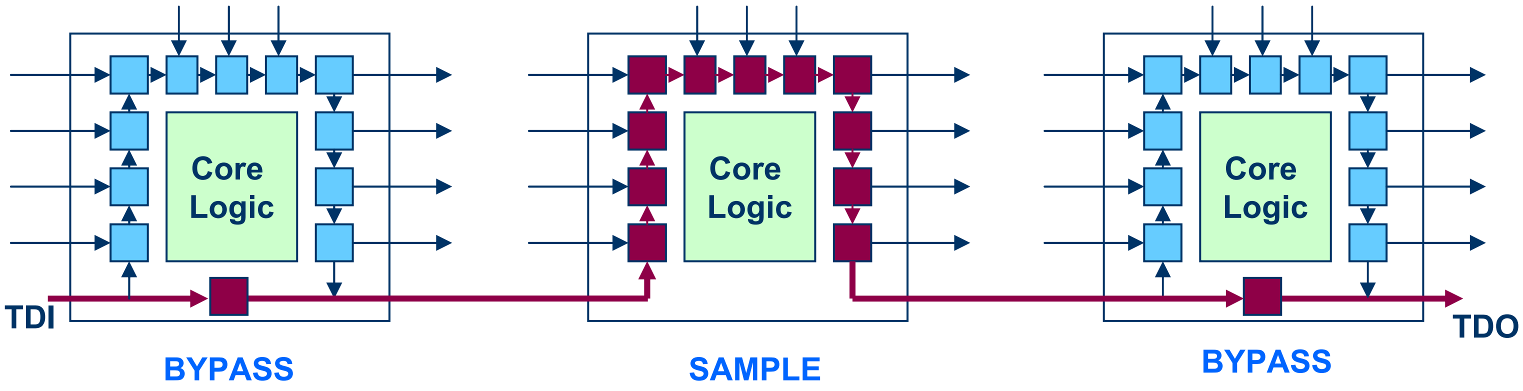
N test vectors: (N+1) DRshifts + 2 IRshifts ≈ (N+1) DRshifts

Boundary Scan Working Modes

BYPASS instruction:

Bypasses the corresponding chip using 1-bit register





Similar instructions: CLAMP, HIGHZ

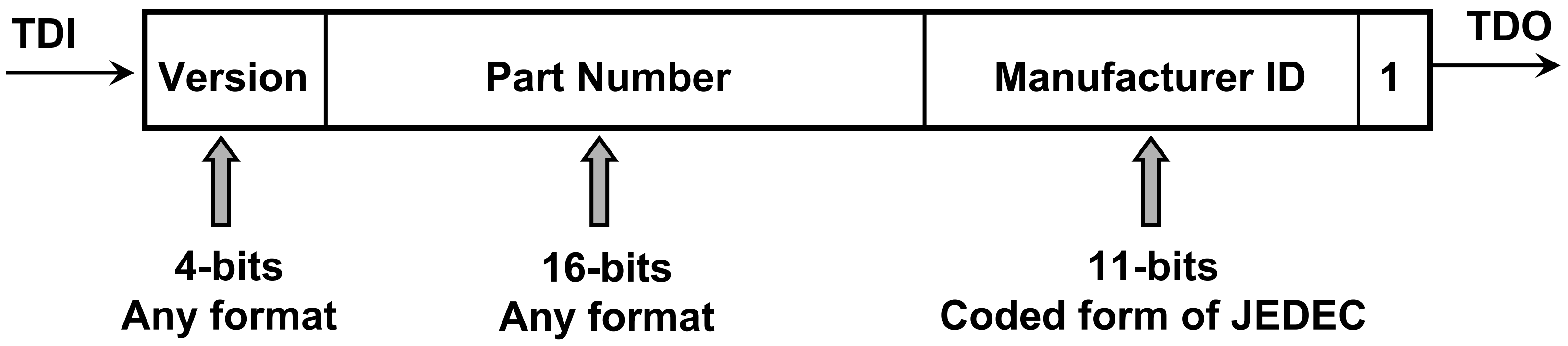
Boundary Scan Working Modes

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



Blind Interrogation

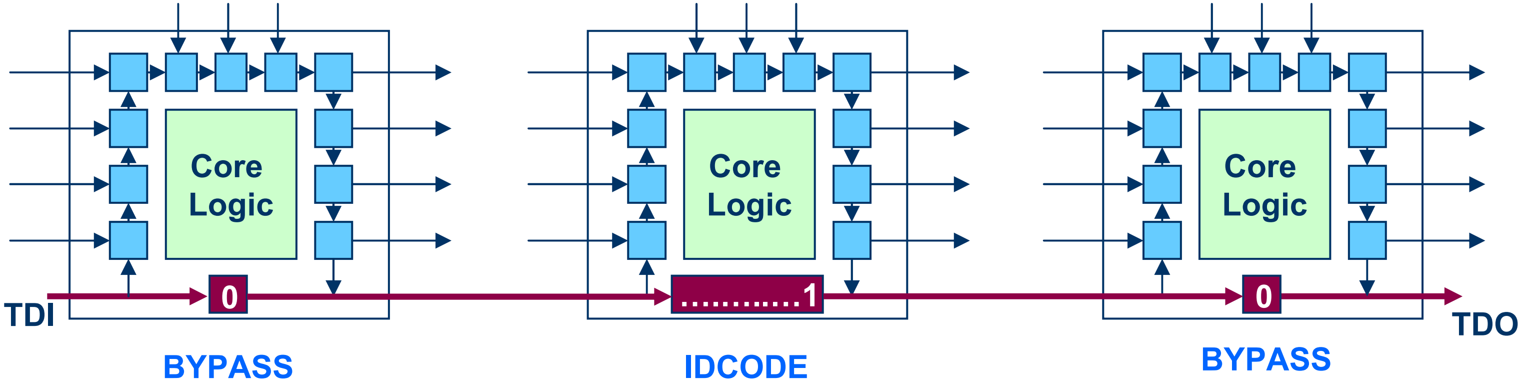
Default instruction:

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

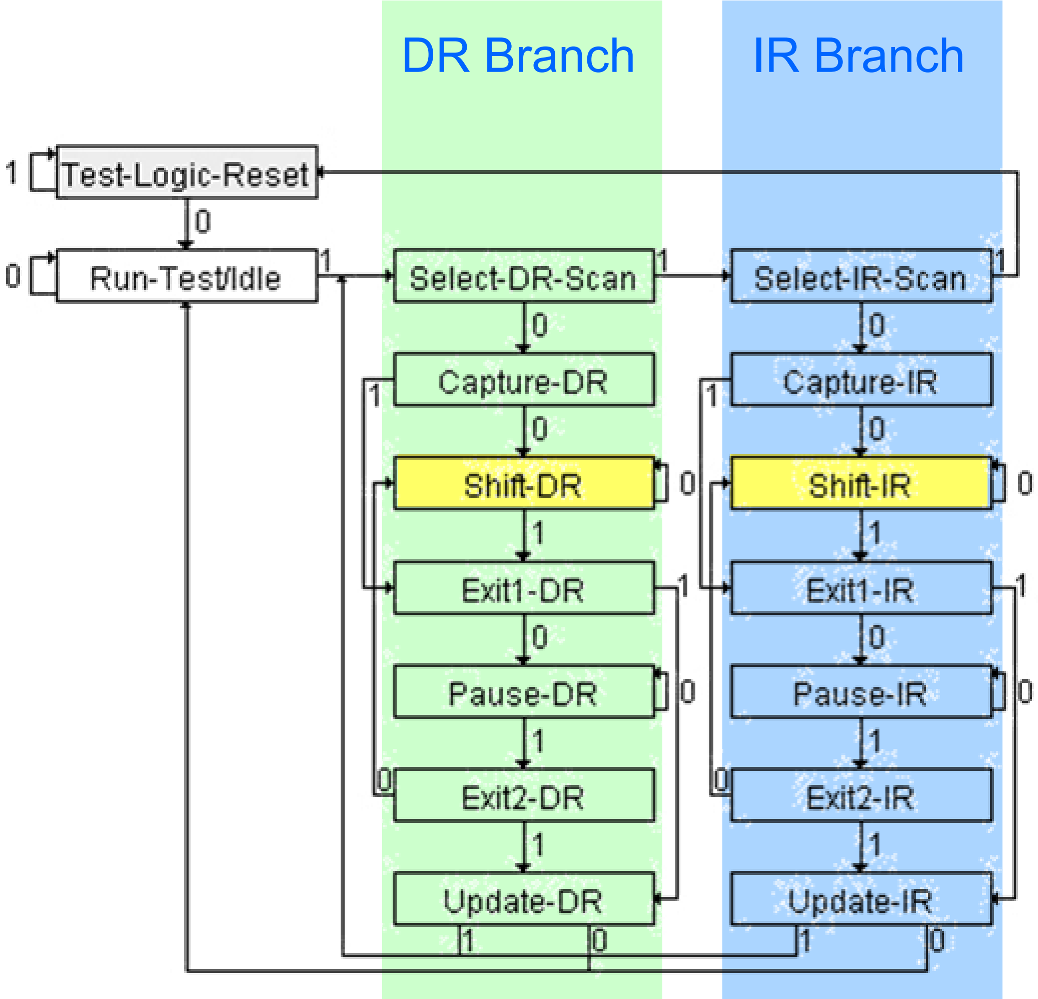
Default capture bits:

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

Example: 0…………………………….10



TAP Controller State Diagram

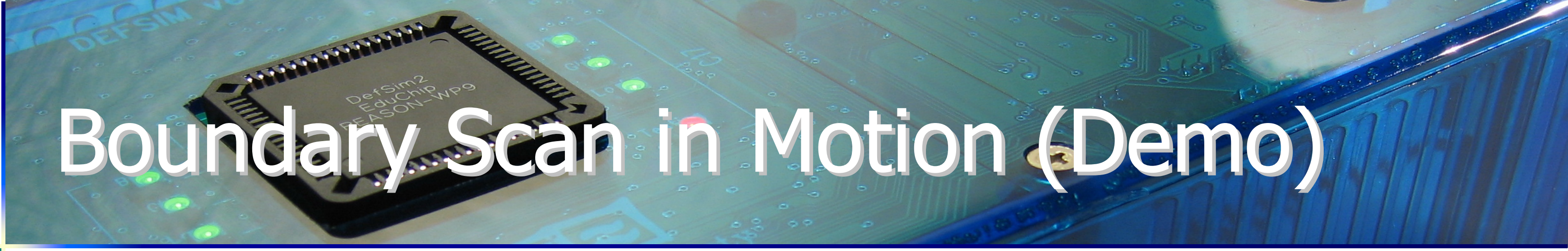


The TAP state diagram has two main branches and two idle states.

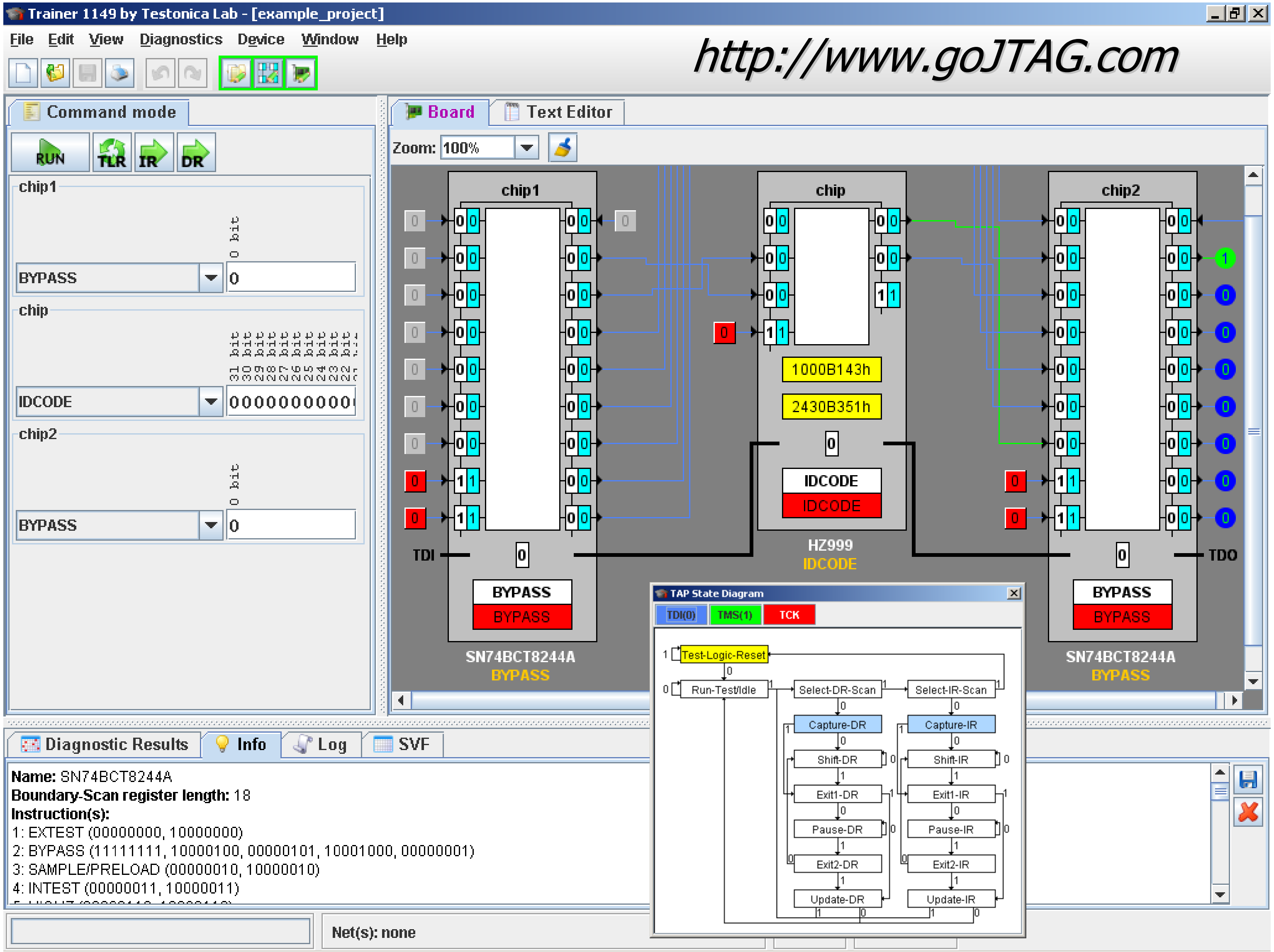
Shift IR and Shift DR states are used to insert instructions and test data into the BS device. These are the most important states.

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

TMS signal is used to move through the states



Boundary Scan in Motion (Demo)



http://www.goJTAG.com

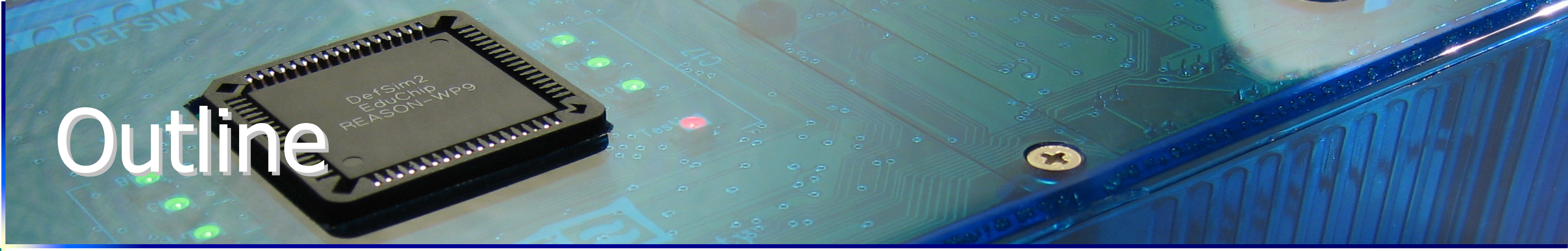
chip1: BYPASS 0

chip: IDCODE 00000000000

chip2: BYPASS 0

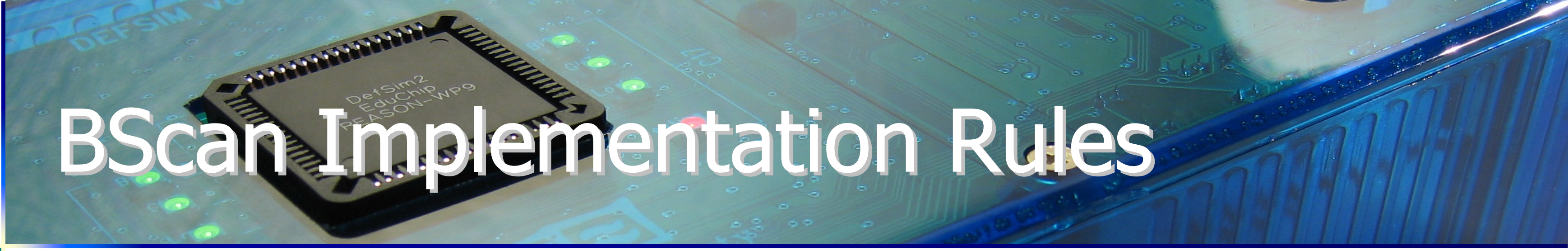
Name: SN74BCT8244A
Boundary-Scan register length: 18
Instruction(s):
1: EXTEST (00000000, 10000000)
2: BYPASS (11111111, 10000100, 00000101, 10001000, 00000001)
3: SAMPLE/PRELOAD (00000010, 10000010)
4: INTEST (00000011, 10000011)

Net(s): none



Outline

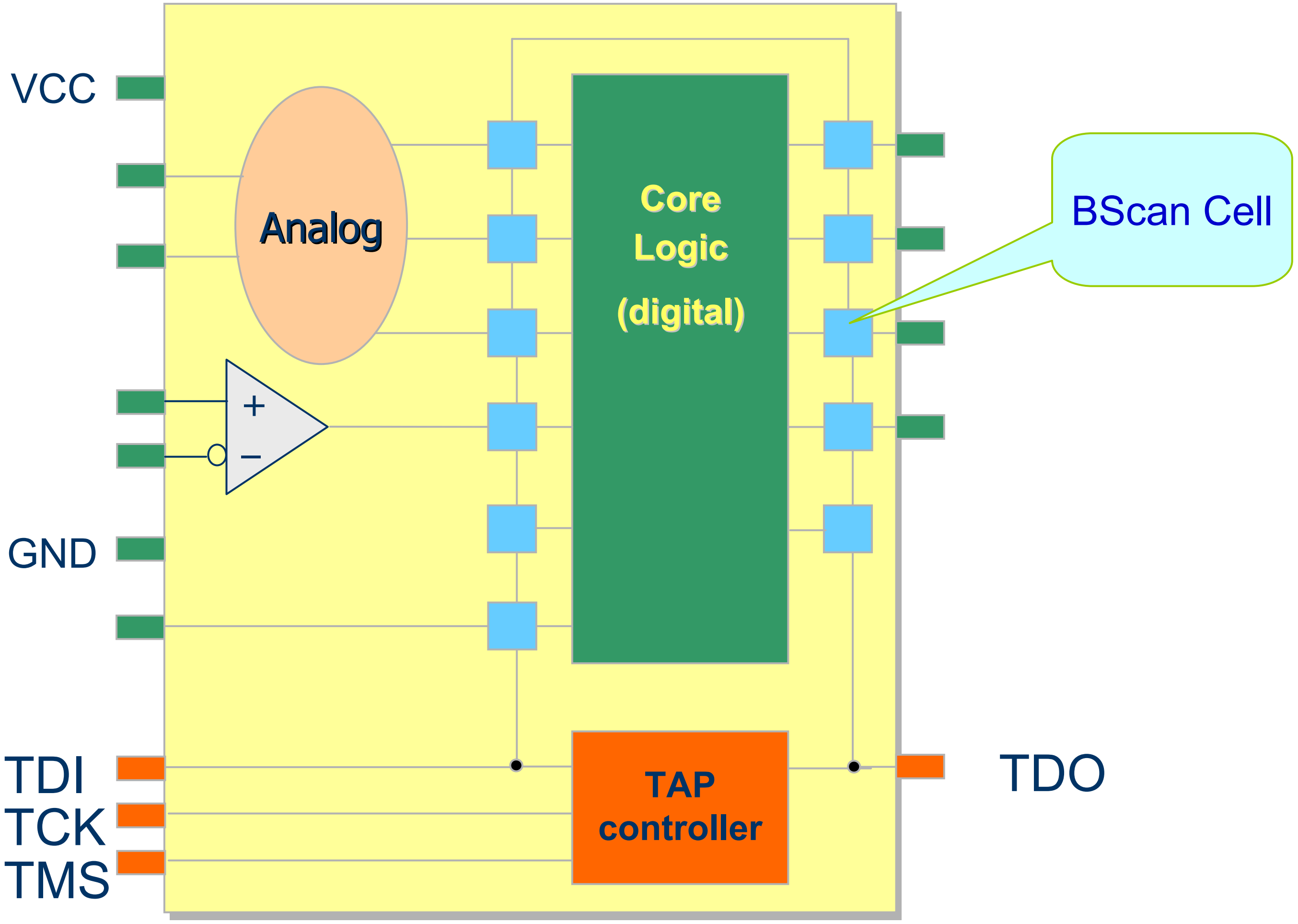
- Board level testing challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan



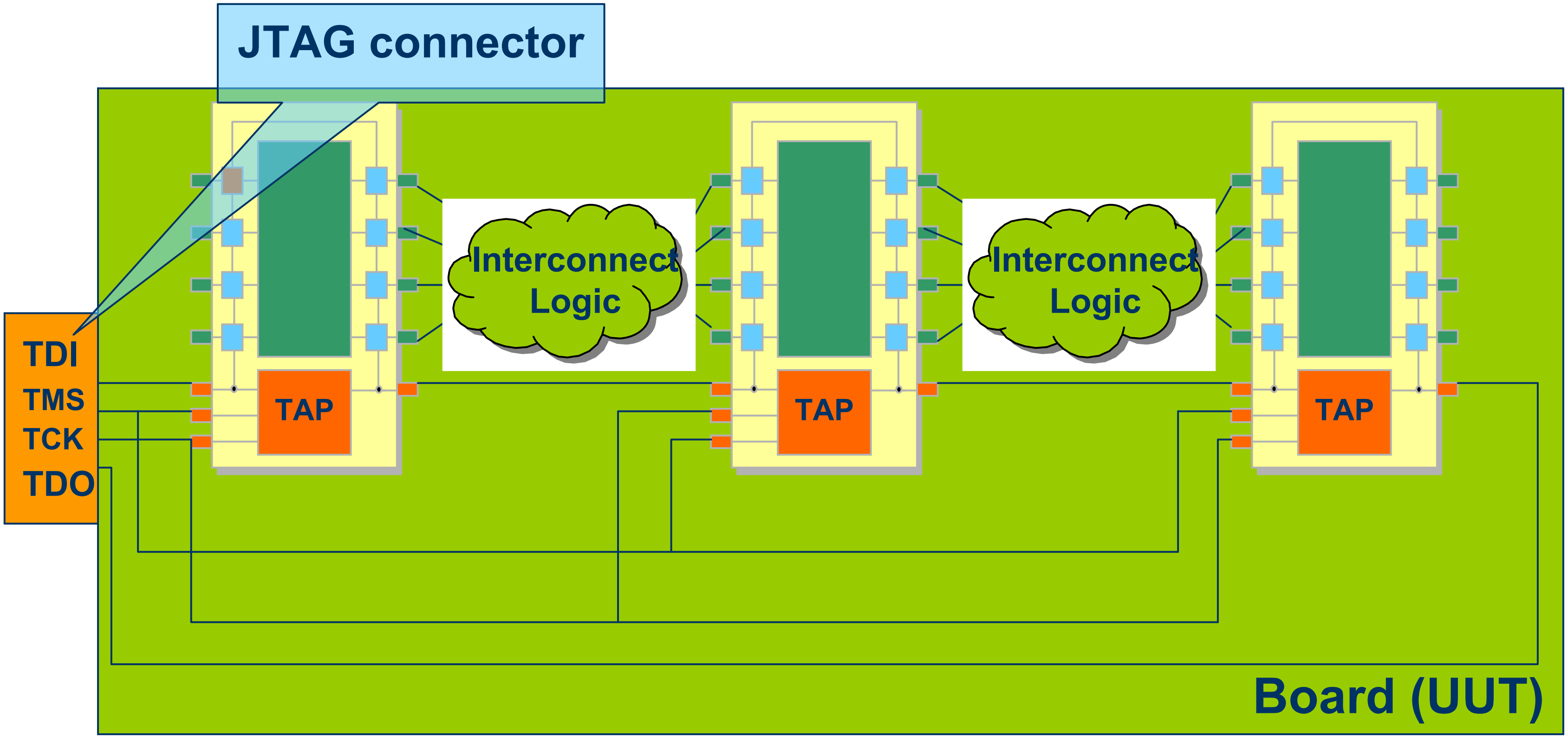
BScan Implementation Rules

- 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



Board-level Test using Boundary Scan



Infrastructure test (generated by using BSDL models)

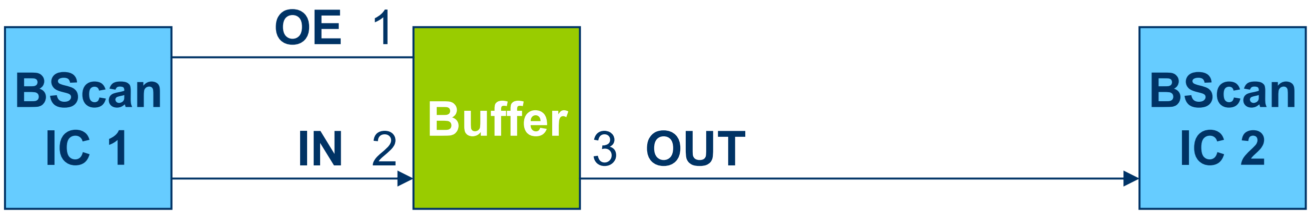
Interconnect test (BSDL models + interconnection netlist)

Board-level Test using Boundary Scan

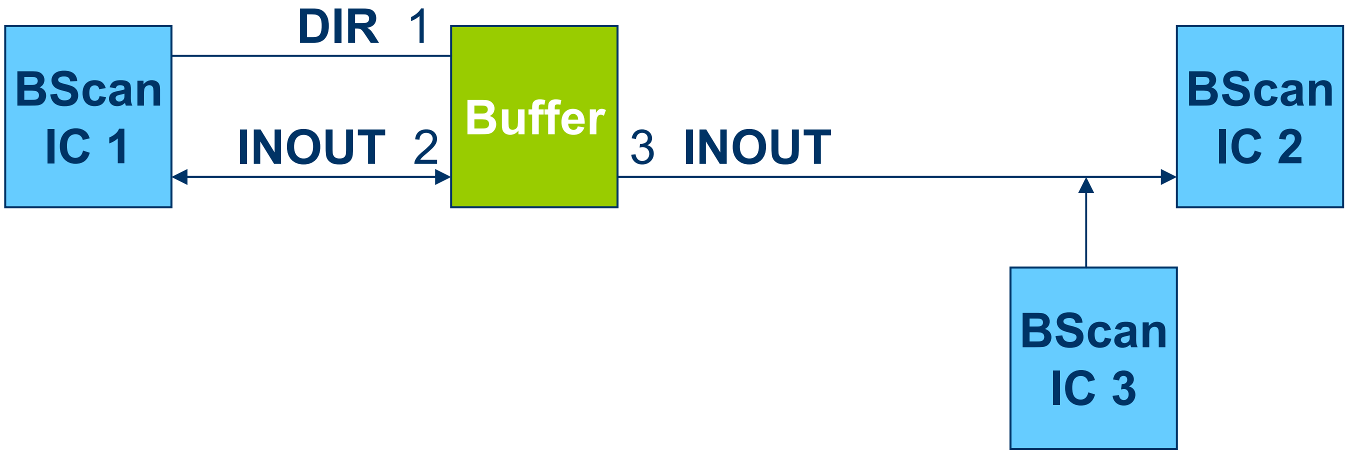
- Infrastructure test
- Interconnect test
  - One needs to specify behavioral models for non-BS components to get acceptable test coverage
  - No standard description format exists
- Additional tasks:
- Cluster logic test – semi-automated
- RAM Test
- External connectors test
- LED or display test (can be assisted by a camera/sensor)
- FLASH test/program/read ID – in-system programming

Interconnect Test through Clusters

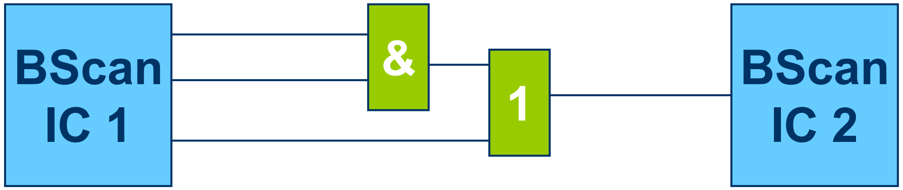
Unidirectional buffer



Bidirectional buffer



Cluster Logic Test (Manual)

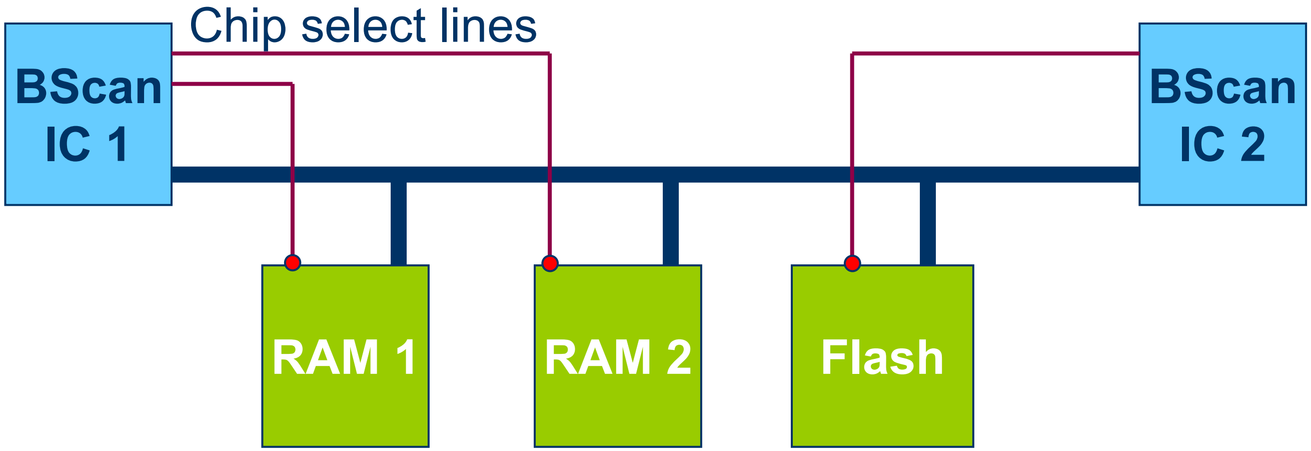


Cluster’s truth table is needed

Truth table

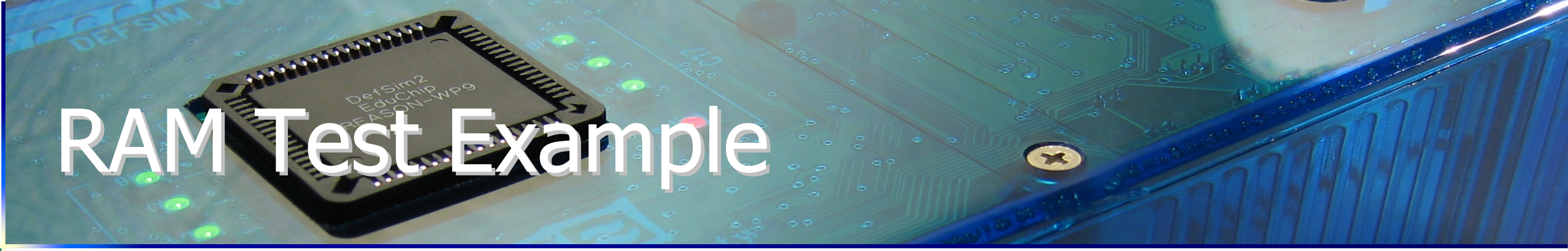
000 0
001 1
010 1
…
111 1

RAM / Flash Test



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



RAM Test Example

Address 10bit	**Data** 32 bit
walking 1	*counting sequence (true/complement)*
0000000000	…….write here something unique…….
1000000000	01010101010101010101010101010101
0100000000	00110011001100110011001100110011
0010000000	00001111000011110000111100001111
0001000000	00000000111111110000000011111111
0000100000	00000000000000001111111111111111
0000010000	10101010101010101010101010101010
0000001000	11001100110011001100110011001100
0000000100	11110000111100001111000011110000
0000000010	11111111000000001111111100000000
0000000001	11111111111111110000000000000000

RAM Test Example: detection of shorts

… assuming Wired-AND model

- 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 at address 0.

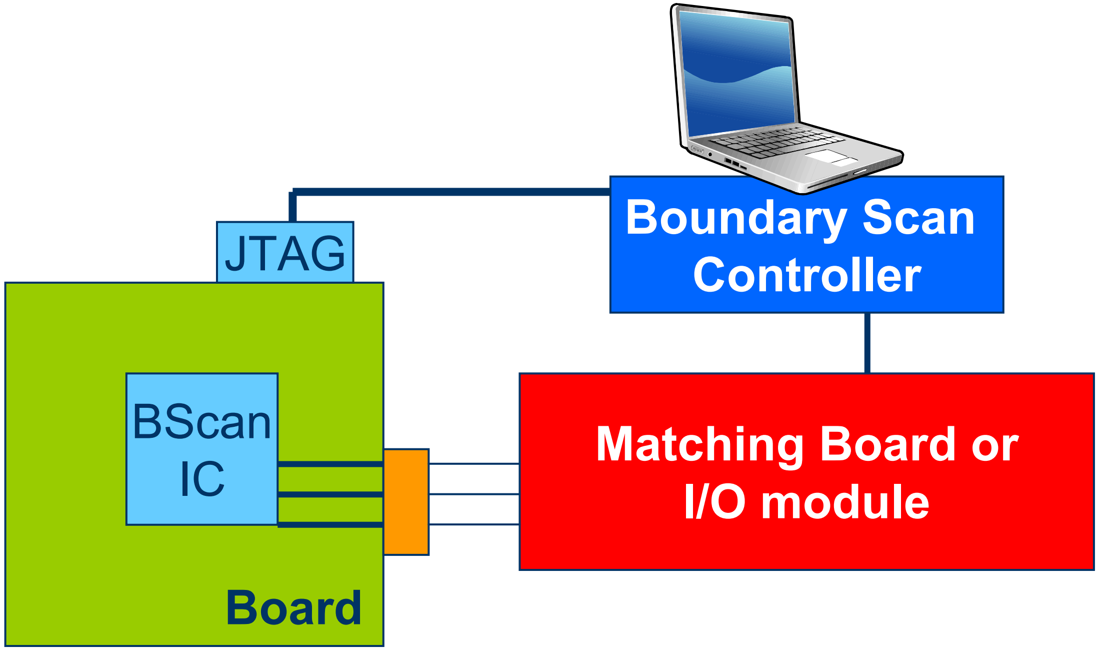
RAM Test Example: detection of opens

… assuming stuck-at 0 model

- 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 at address 0.

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

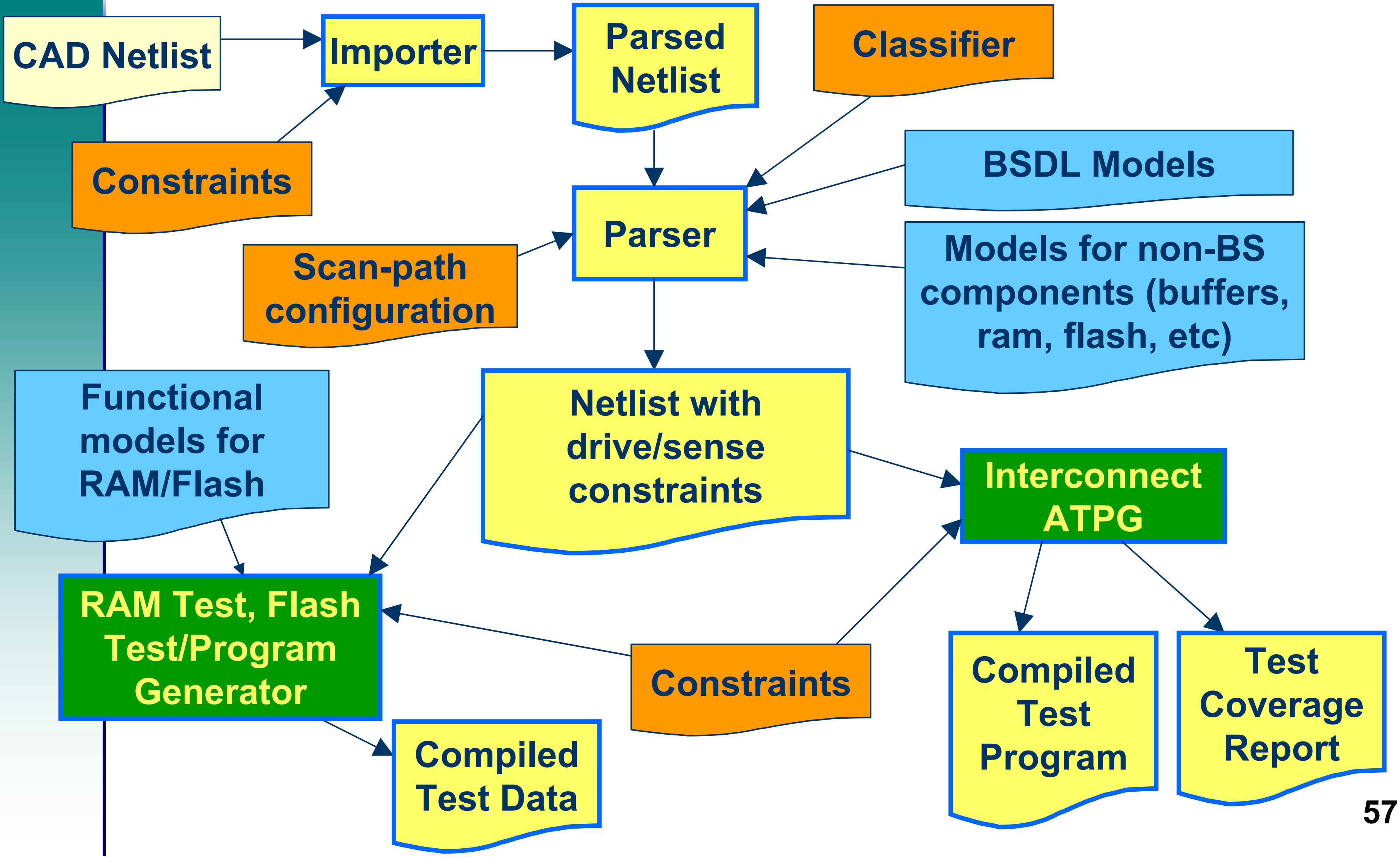
Testing External Connectors



Only interconnect test will be performed!

The real protocol of external connector is not tested

Boundary Scan Test Development Typical Workflow



CAD Netlist

Importer

Parsed Netlist

Classifier

Constraints

BSDL Models

Parser

Scan-path configuration

Models for non-BS components (buffers, ram, flash, etc)

Functional models for RAM/Flash

Netlist with drive/sense constraints

Interconnect ATPG

RAM Test, Flash Test/Program Generator

Constraints

Compiled Test Program

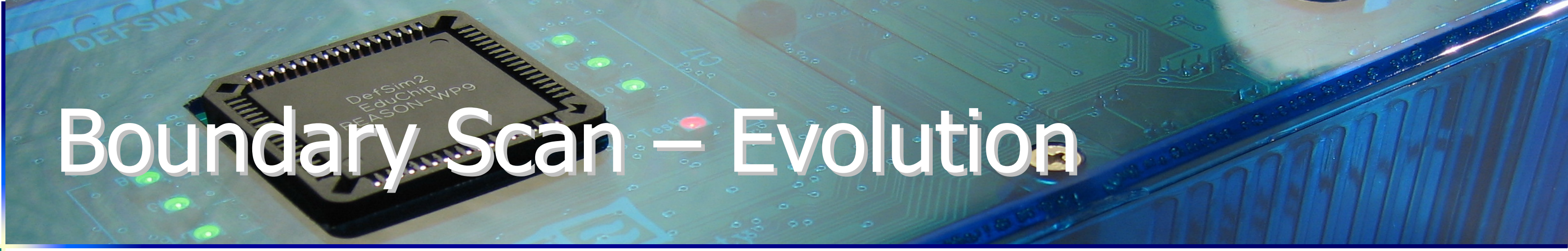
Test Coverage Report

Compiled Test Data

IEEE 1149.1 Summary

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
- Now getting widespread usage



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

P1687 – IJTAG – Internal JTAG (component testing, BIST)

1500 – Embedded Core Test (SoC testing)

1532 – In-System Configuration of Programmable Devices

P1581 – Static Component Interconnection Test Protocol and Architecture (memory-to-BS_chip links testing)

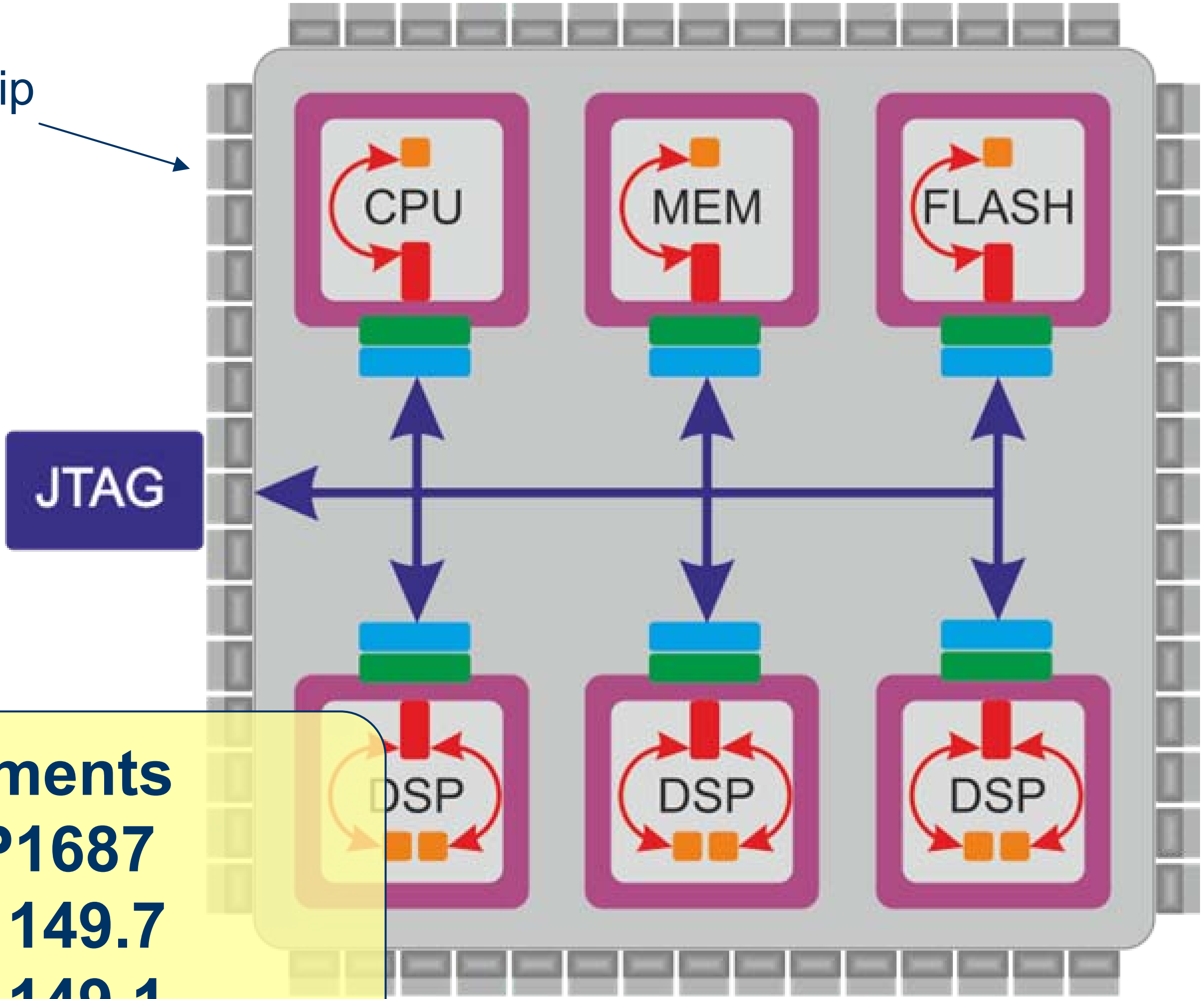
5001 – NEXUS – Global Embedded Processor Debug Interface (SW development, debug, and emulation)

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 P1149.8.1 – improved observability for measurement (from the JTAG standpoint) by implementing a capacitive sensing technology
- IEEE P1149.1-2011 – solves signaling issues on the buses by driver initialization procedures
- IEEE P1687 – 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

New and Emerging Standards Combined

Board or chip



- 🟧 **Instruments**
- 🟥 **IEEE P1687**
- 🟦 **IEEE 1149.7**
- 🟩 **IEEE 1149.1**
- 🟪 **IEEE P1149.1-2011**

What to look further

Leading BScan companies:

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

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
- Lecture notes by Ben Bennetts (try googling)