Board-level testing and IEEE1149.x Boundary Scan standard

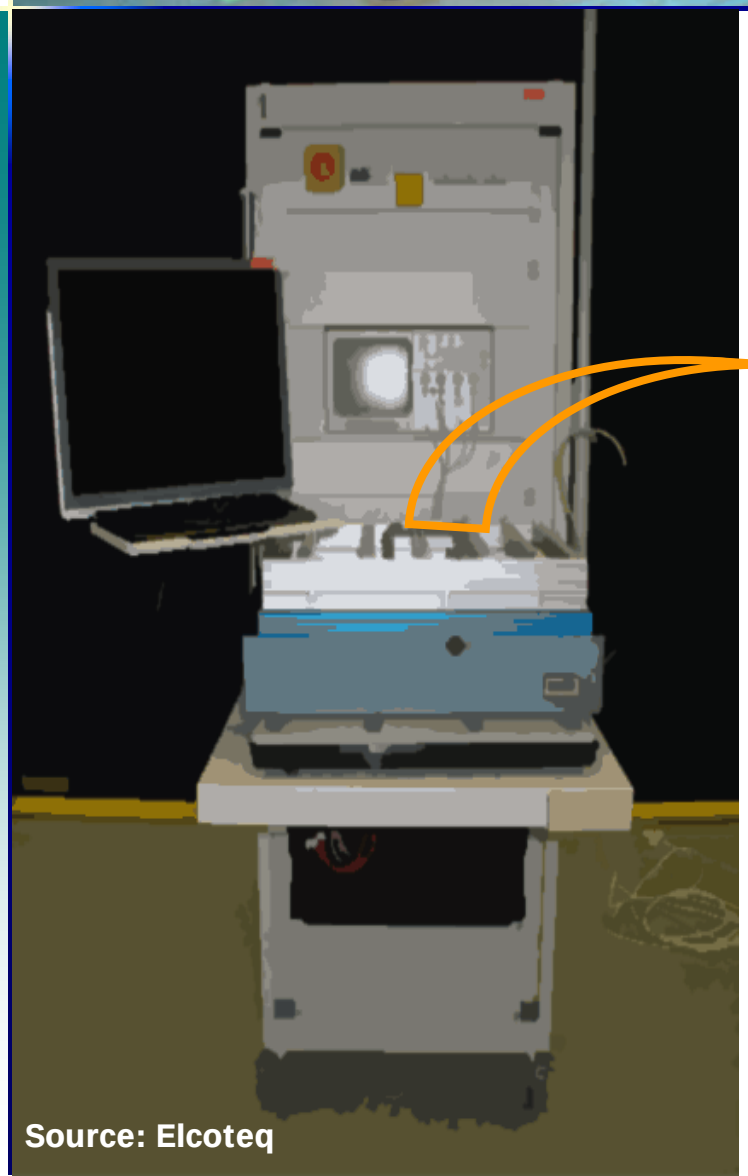
Artur Jutman
artur@ati.ttu.ee

October 2010

IEEE 1149.1 Boundary Scan Standard

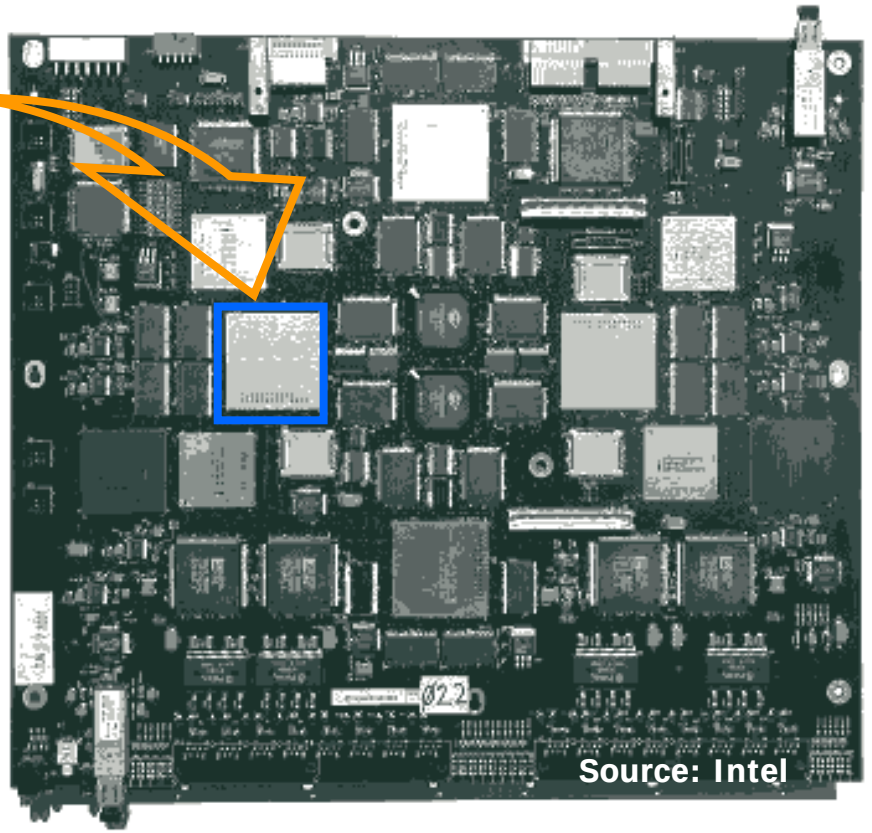
- Board level testing challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan

The Challenge of Board Testing



Tested chips placed on board

☝ Only interconnects to be tested!



Modeling of Interconnect Faults

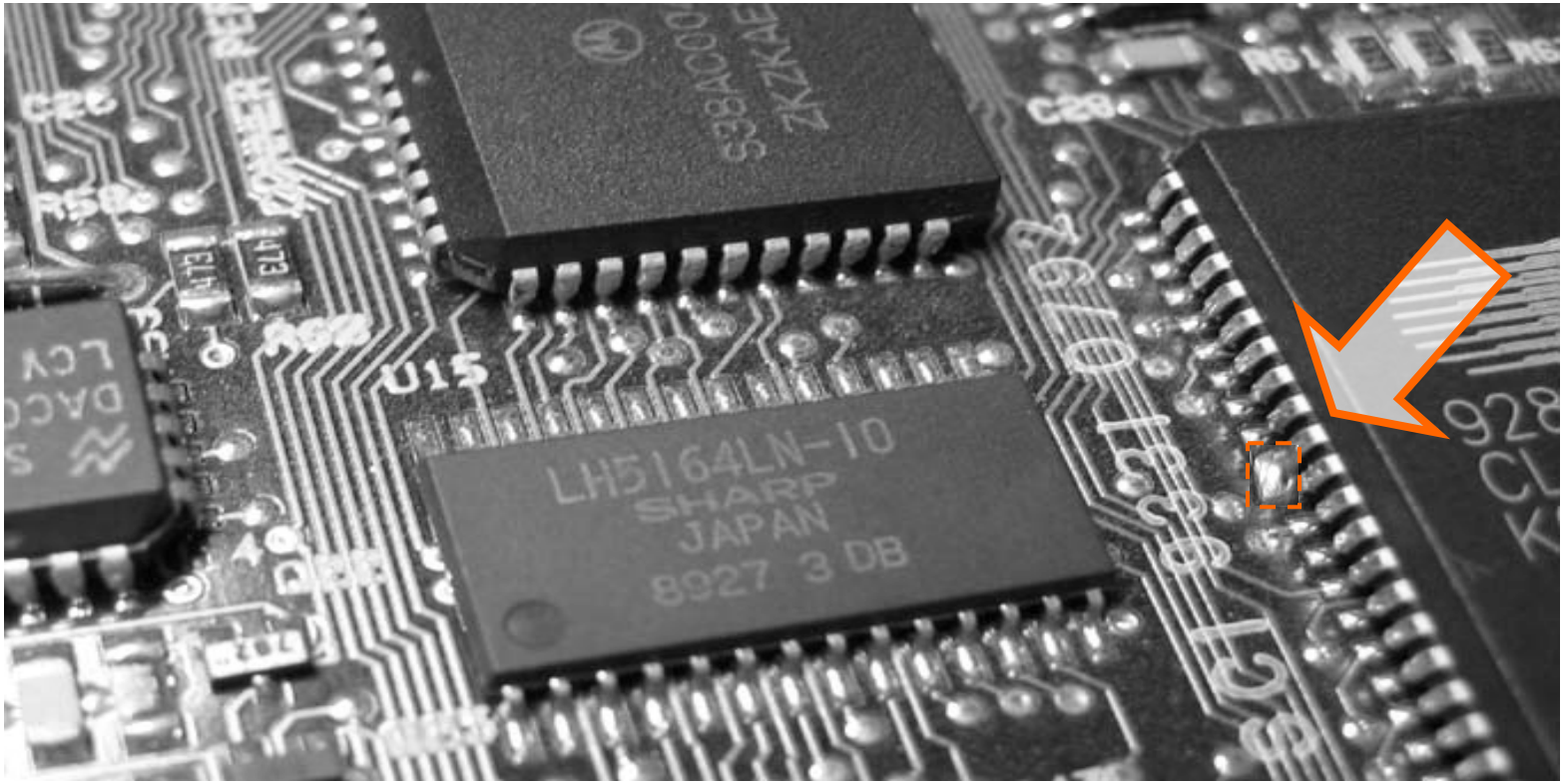
Net-level defect types and models

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

Short faults, Open faults: static behavior

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

Short Faults

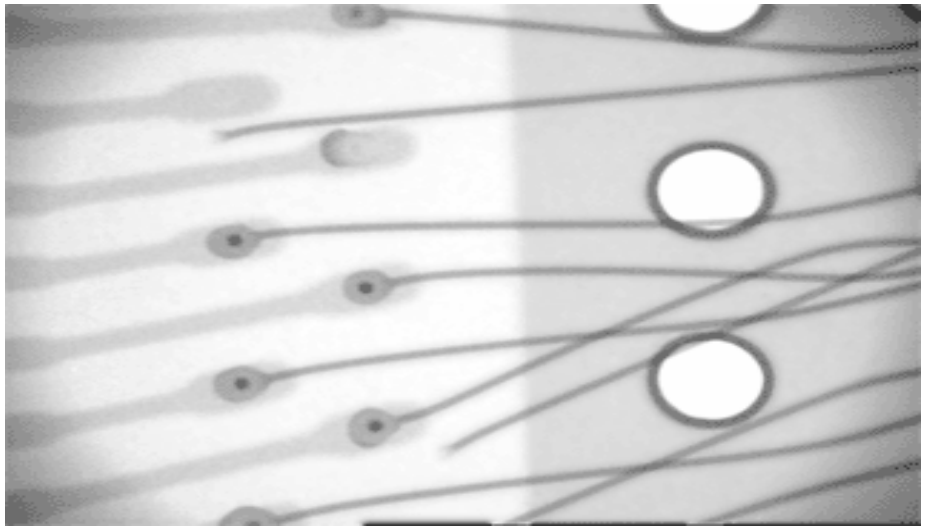


Possible shorts: bond wire, leg, solder, interconnect

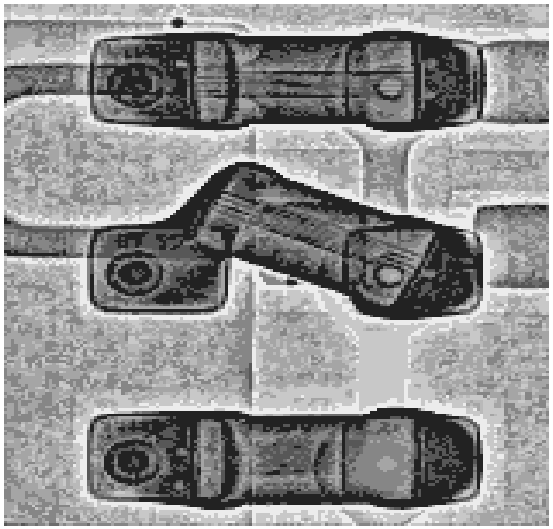
Shorts are usually modeled as wired-AND, wired-OR faults

Open Faults

Misplaced bond wire



Misplaced component



Possible opens: bond wire, leg, solder, interconnect

Opens usually behave like stuck-at or delay faults

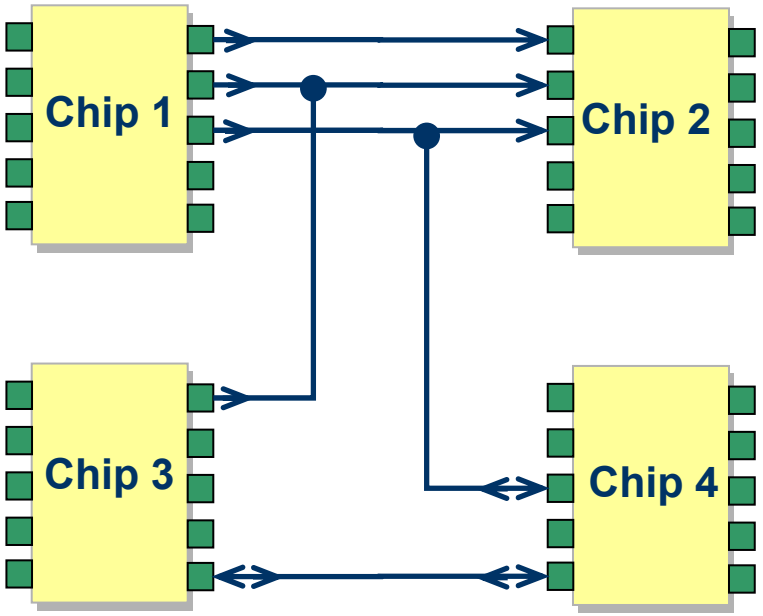
Tri-state connections

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.

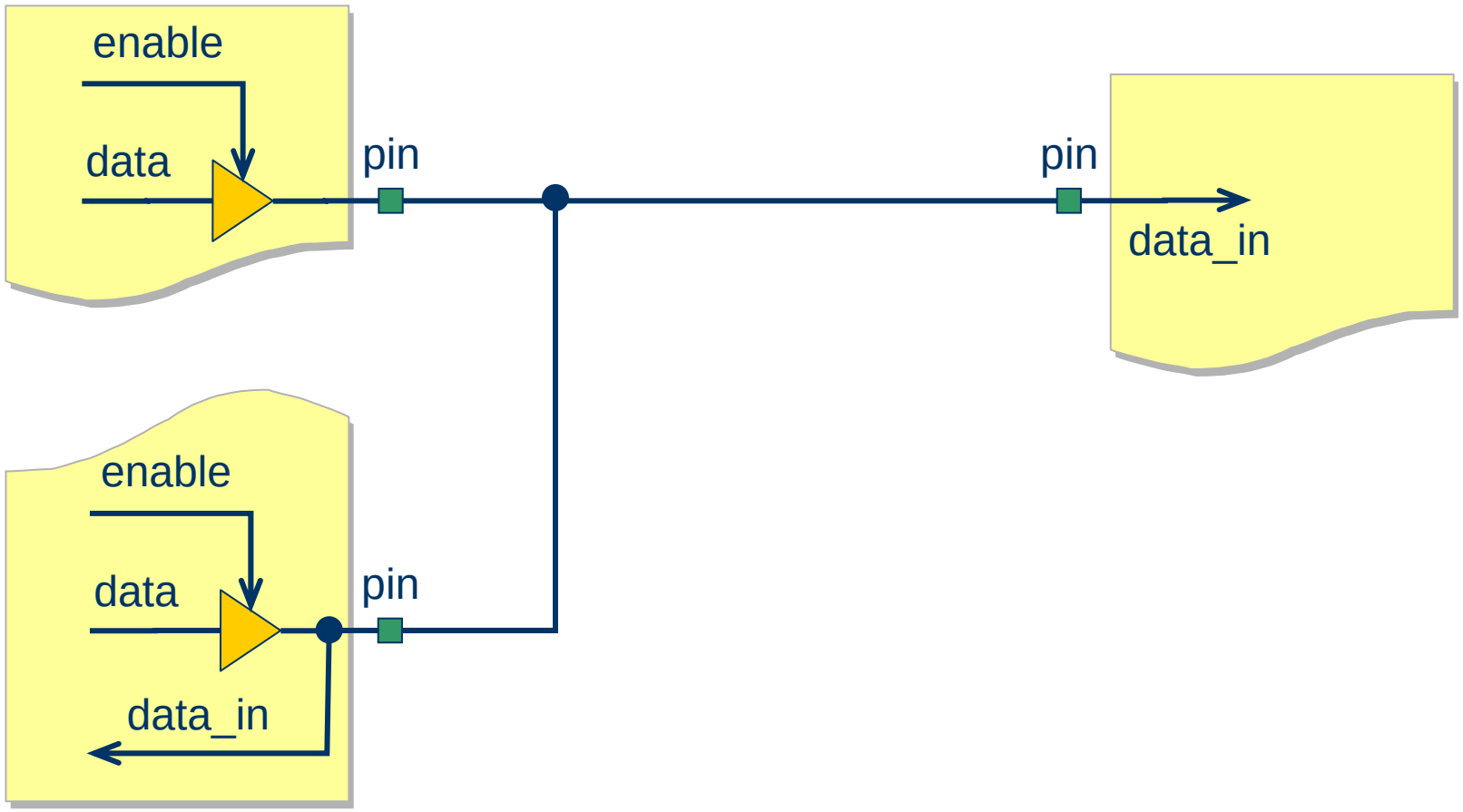
There are special "enable" signals that control this additional state of the I/O pins.

Tri-state connections



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

Tri-state net structure



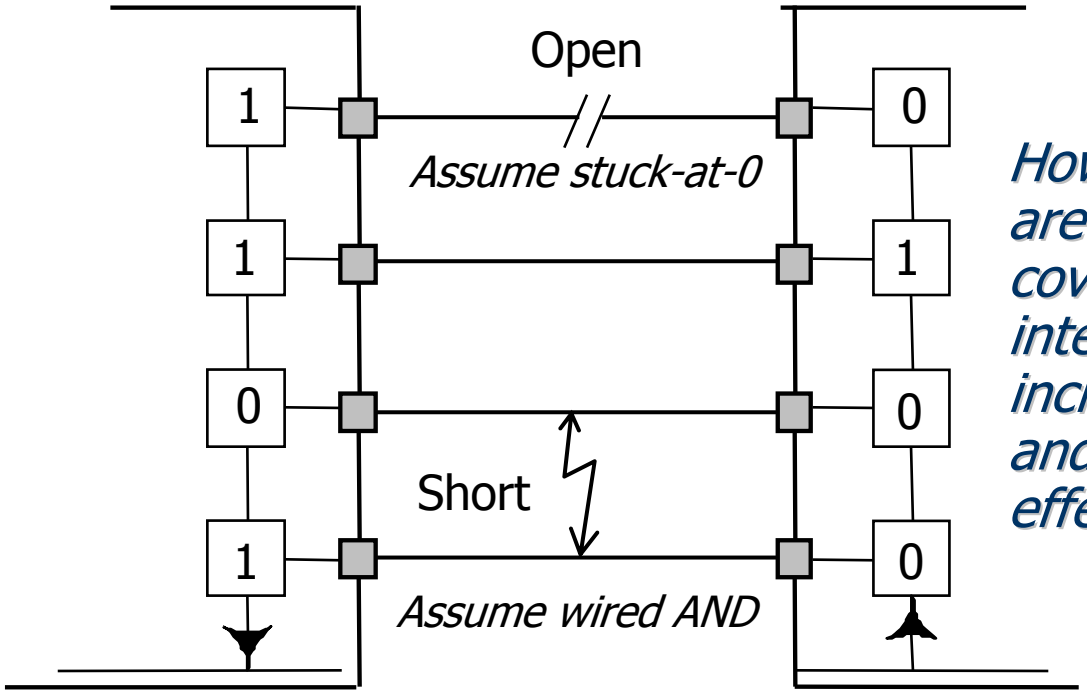
Specific faults in tri-state nets

- 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

IEEE 1149.1 Boundary Scan Standard

- 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

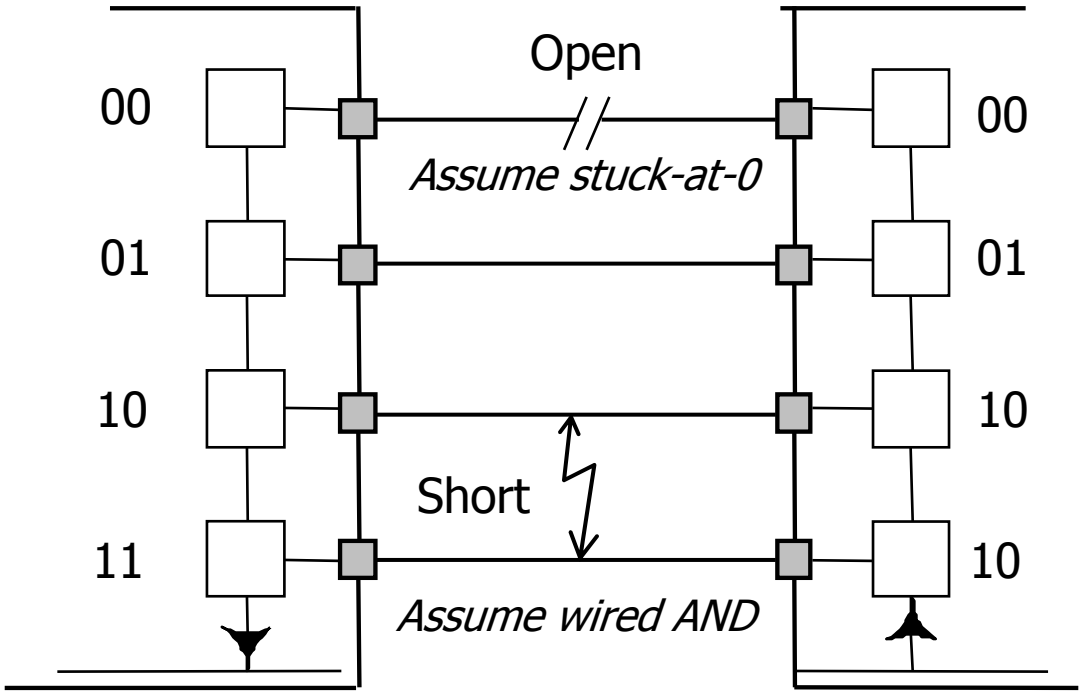
Test Generation Algorithms



How many vectors are enough to cover all possible interconnect faults including delays and other dynamic effects?

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

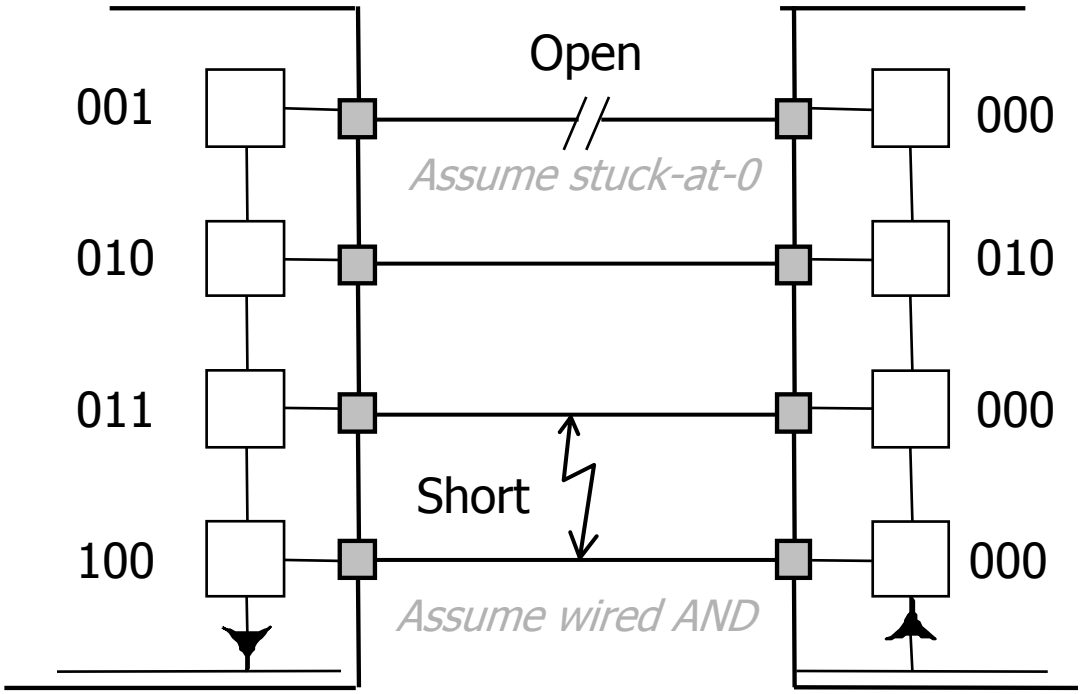
The Counting Sequence



What about opens?

Kautz [1] showed in 1974 that a sufficient condition to detect any pair of short circuited nets was that the serial codes must be unique for all nets. Therefore the test length is $\lceil \log_2(N) \rceil$

The Modified Counting Sequence

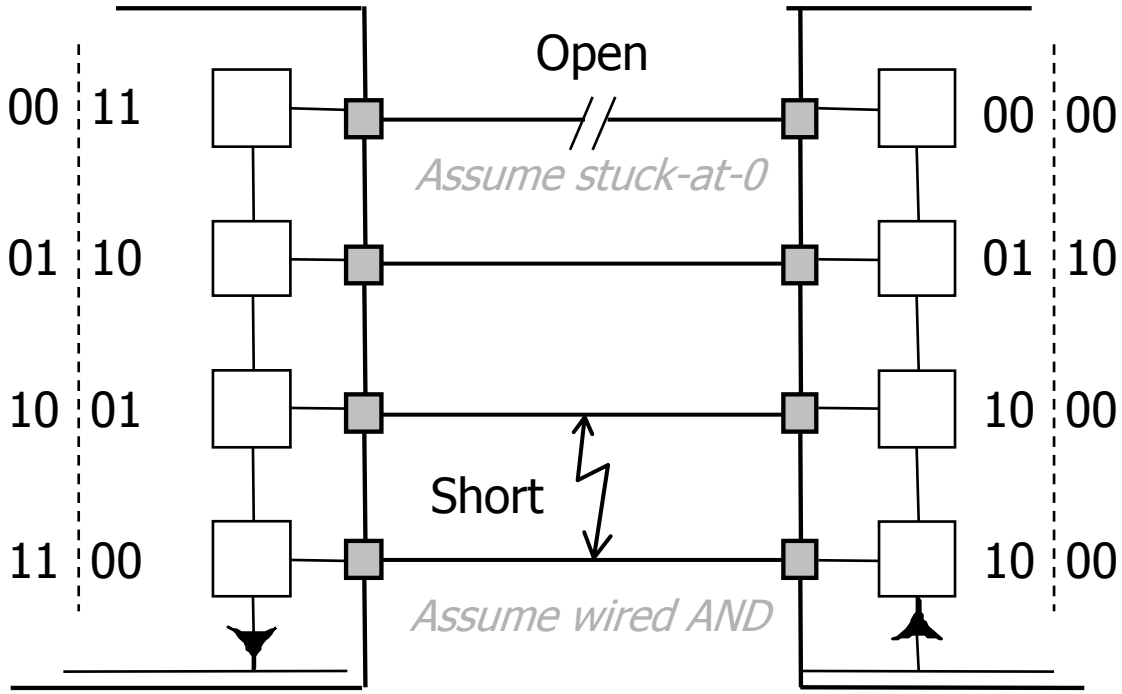


Some of the observed error responses are allowed codes.

How to improve the diagnosis?

All 0-s and all 1-s are forbidden 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 [2]

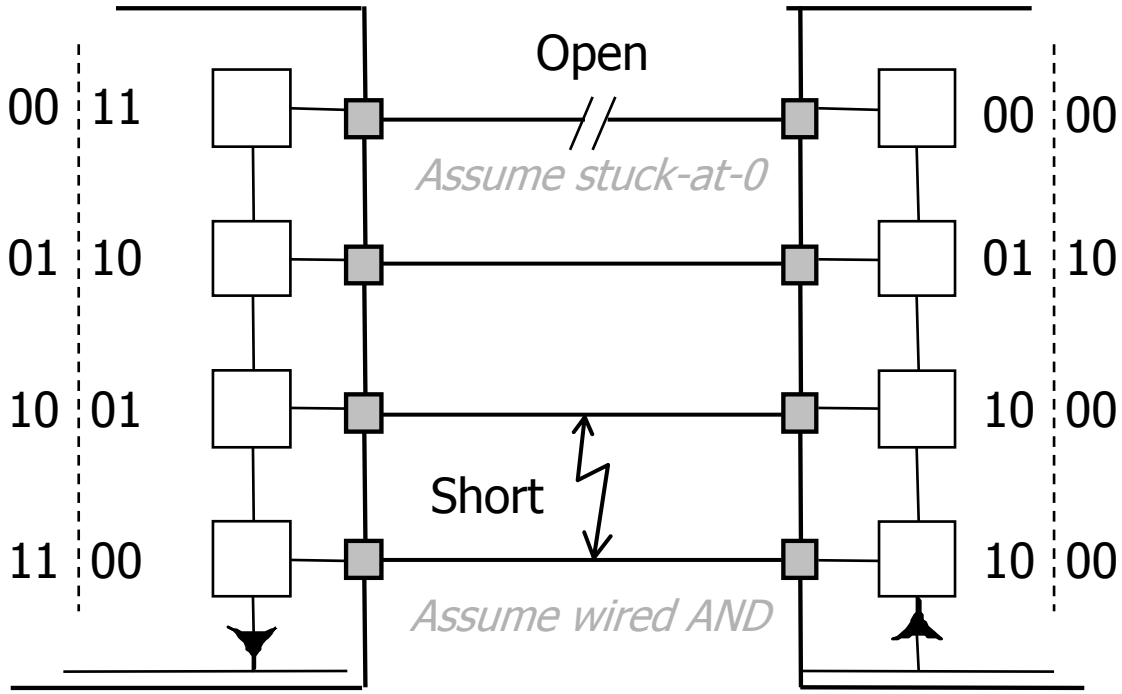
The True/Complement Code



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

To improve the diagnostic resolution Wagner proposed the True/Complement Code in 1987 [3].
The test length became equal $2\lceil \log_2(N) \rceil$

The True/Complement Code



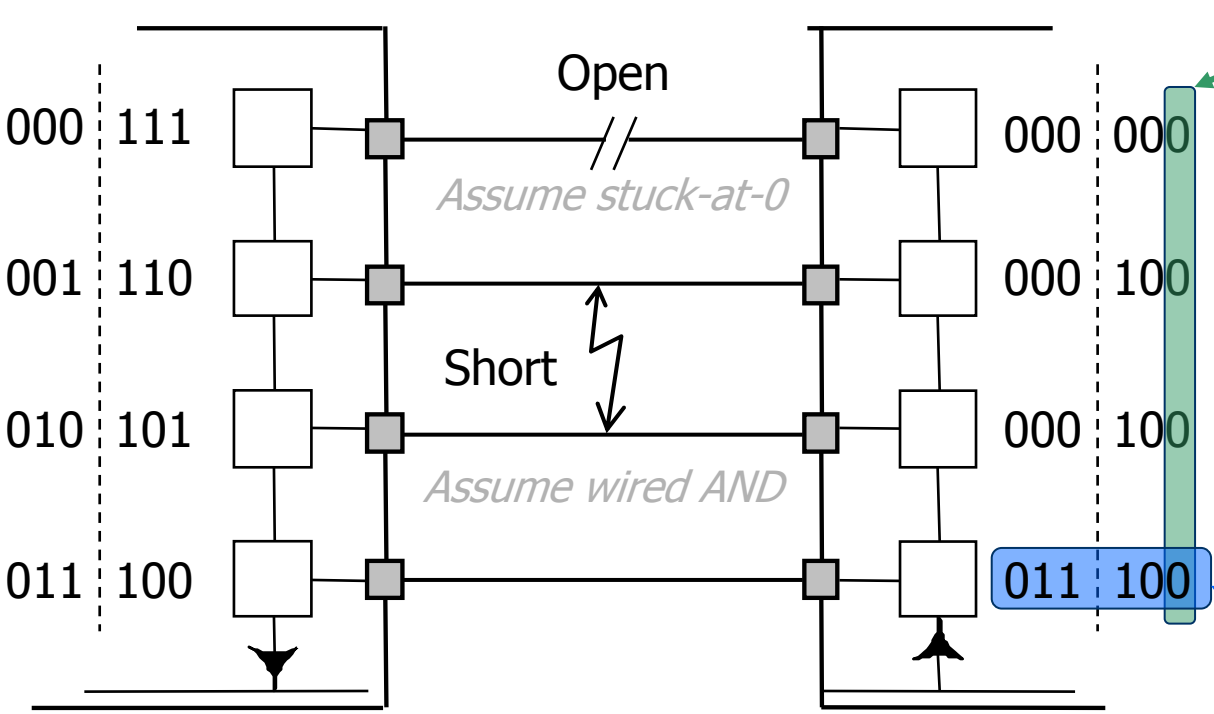
What about delay faults and other dynamic effects?

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

New!



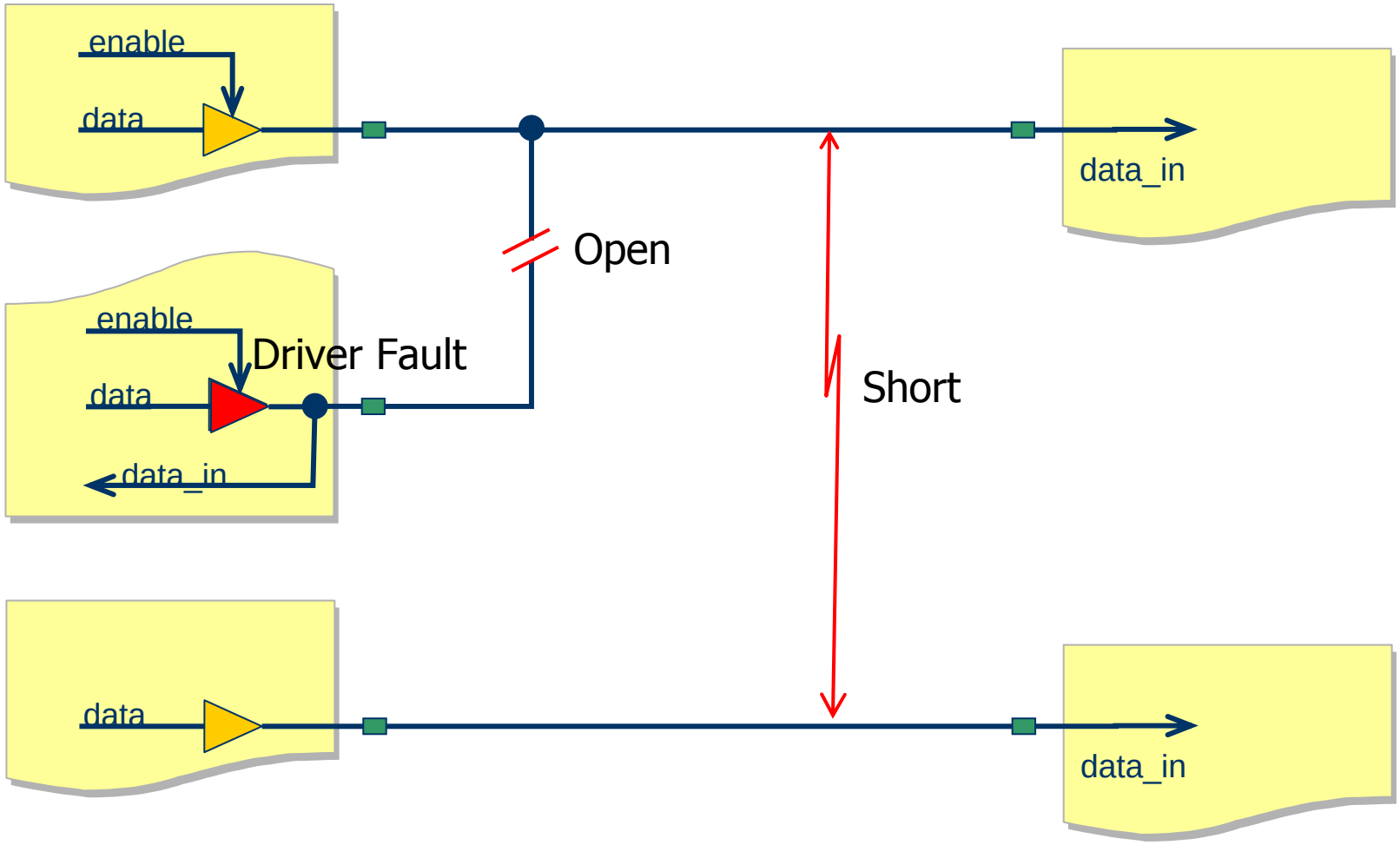
Add two bits, that are the same at every code word
Shorts and stuck-at faults are now *distinguishable*
The test length is $2\lceil \log_2(N) \rceil + 2$

Summary of TG Methods

Updated!

	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

New!

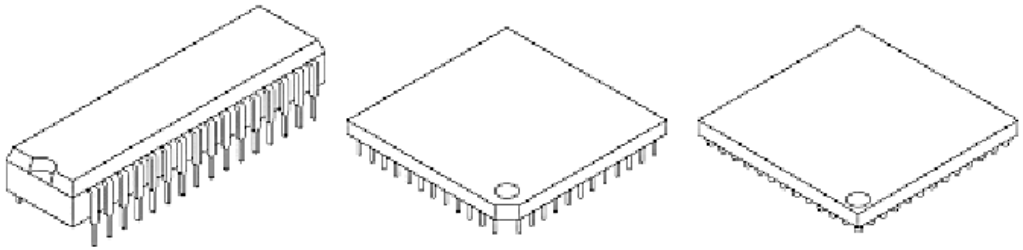
- Every driver on the net should at least once drive low and at least once drive high
- Every input should at least once sense 0 and at least once sense 1
- Two drivers should never drive simultaneously
- One can distinguish between a driver fault and open net by sensing back on a bi-directional pin

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)

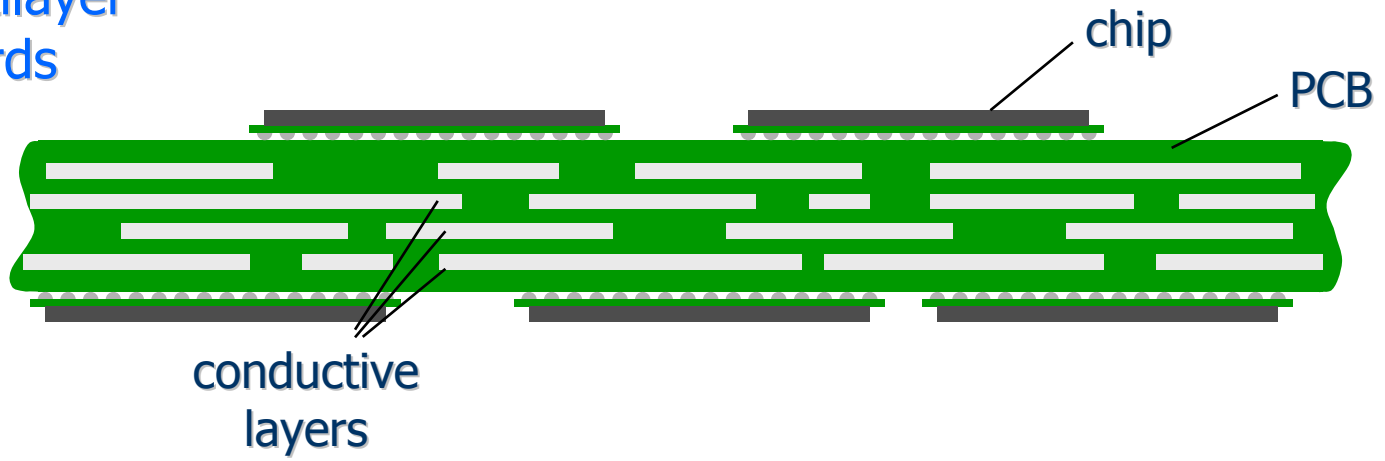
Limitations of the Nail Probing

Packaging styles

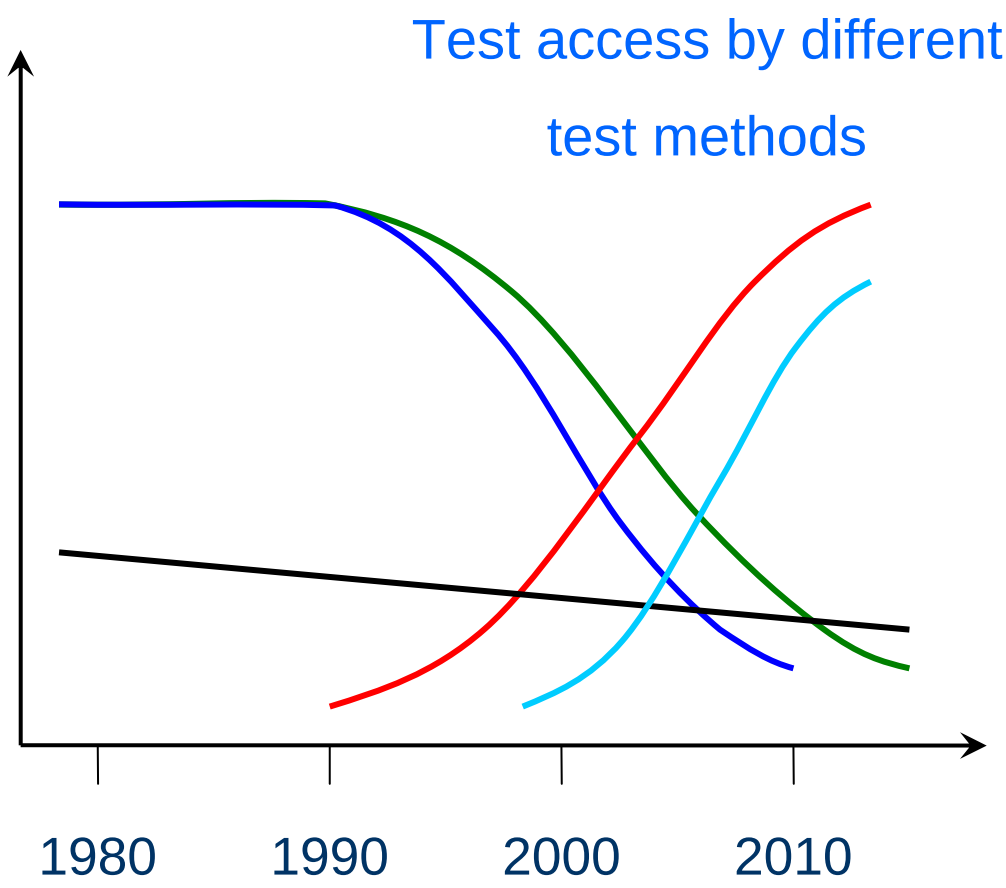


DIP PGA BGA

Multilayer boards



Test Access Methods: Usage Trends



IEEE 1149.1 Boundary Scan Standard

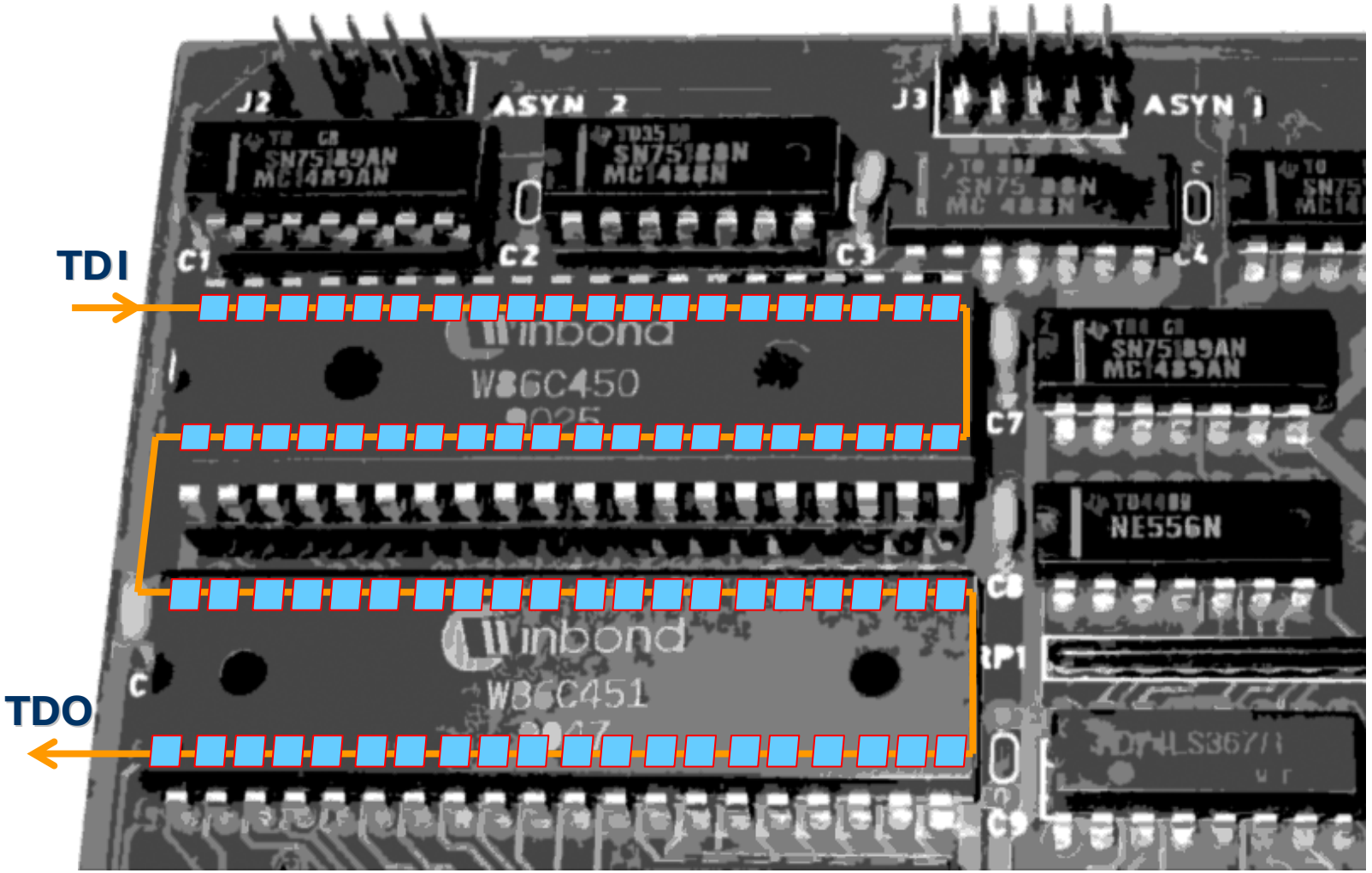
- Board level testing challenges
- Fault modeling at board level (digital)
- Test generation for interconnect faults
- IEEE 1149.1 Boundary Scan Standard
- Application of Boundary Scan

IEEE 1149.1 Boundary Scan: History

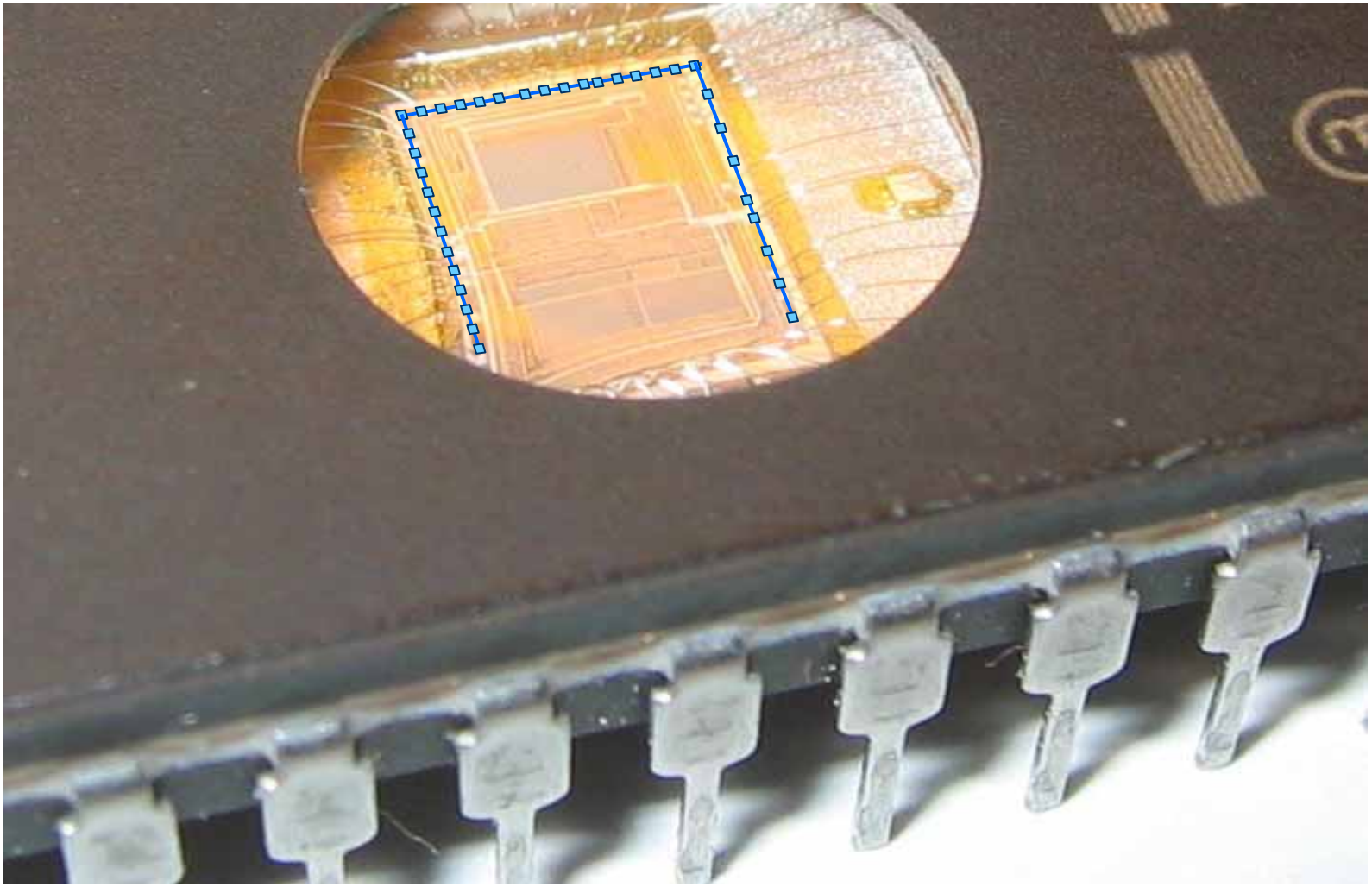
- Early 1980's – problem of test access to PCBs via "bed-of-nails" fixture
- Mid 1980's – Joint European Test Action Group (JETAG)
- 1986 – US companies involved: JETAG -> JTAG
- 1990 – JTAG Test Port became a standard [4]:

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

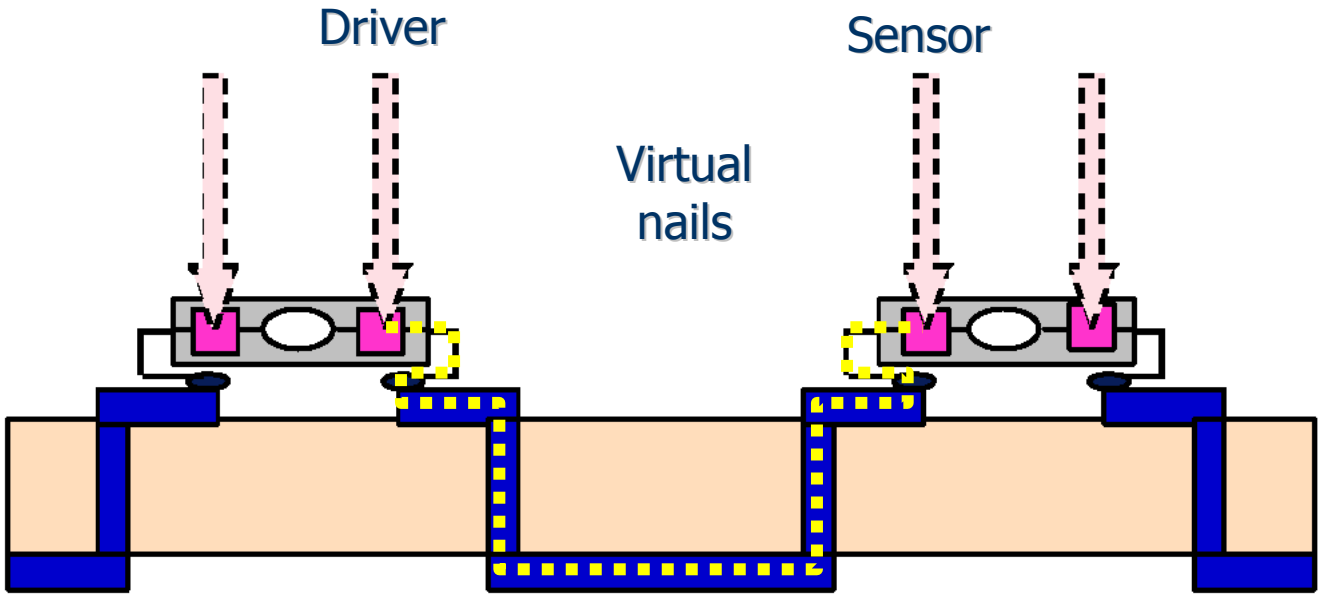
Test Access Via Boundary Scan



Test Access Via Boundary Scan



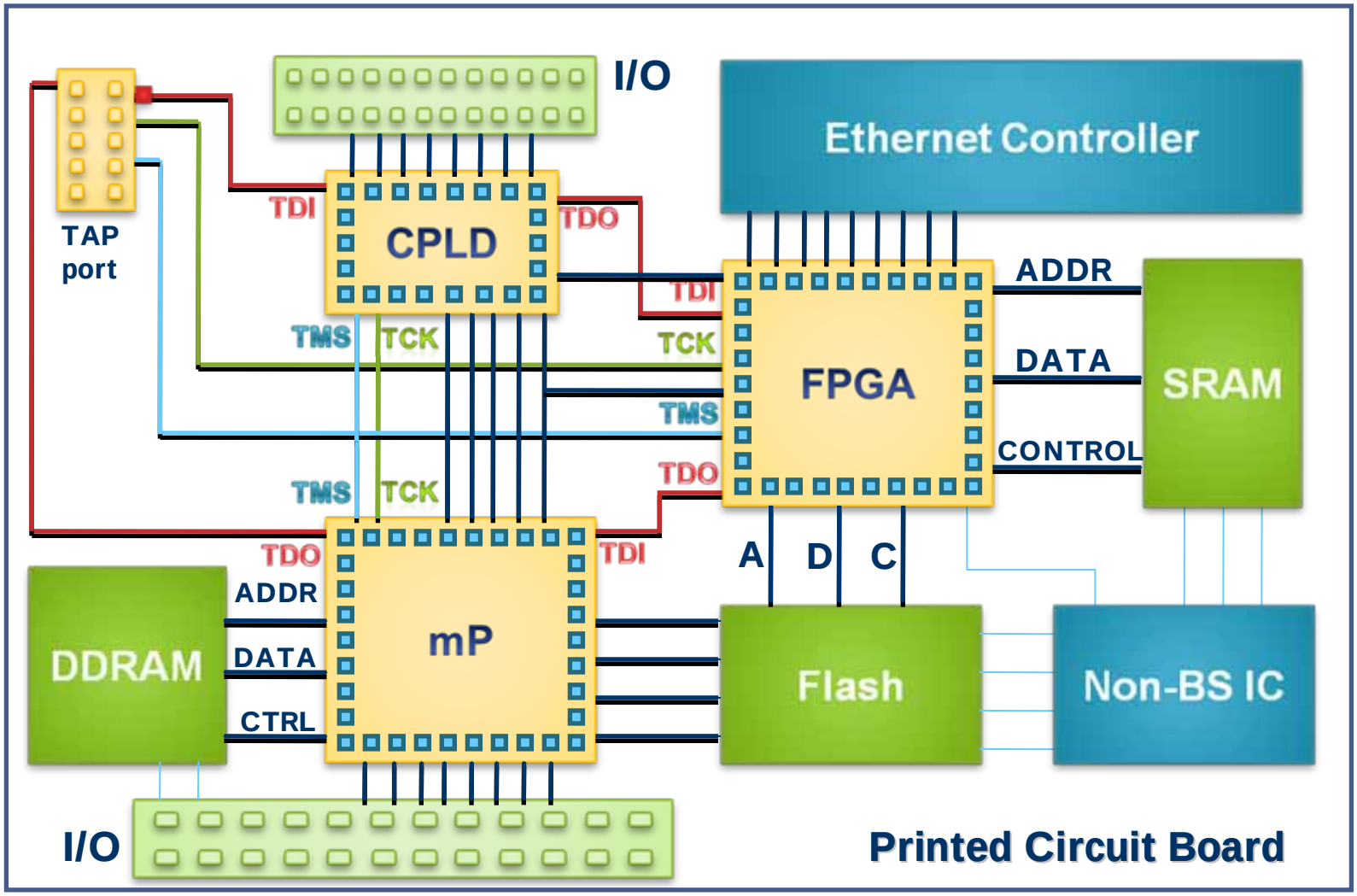
Test Access Via Boundary Scan



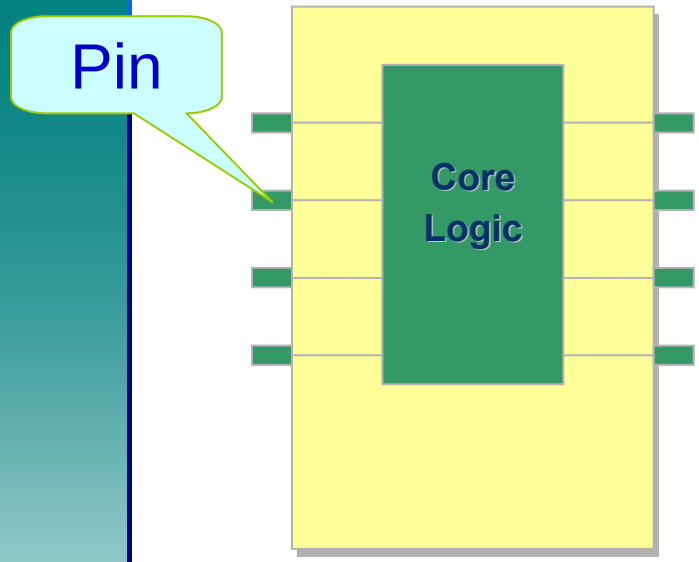
Defects covered:

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

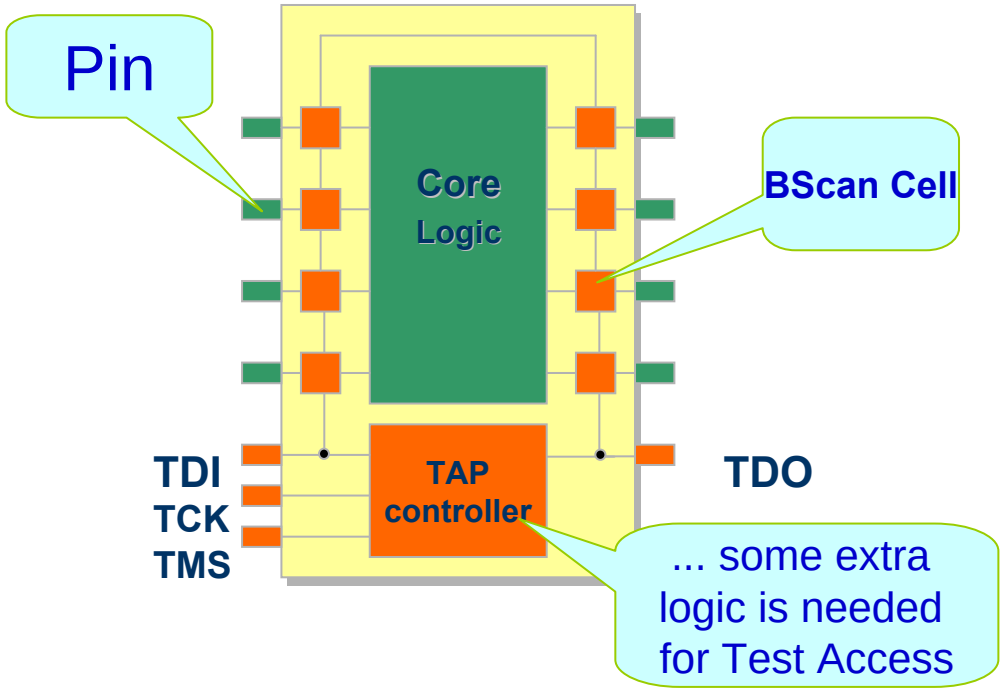
Test Access Via Boundary Scan



Boundary Scan basics



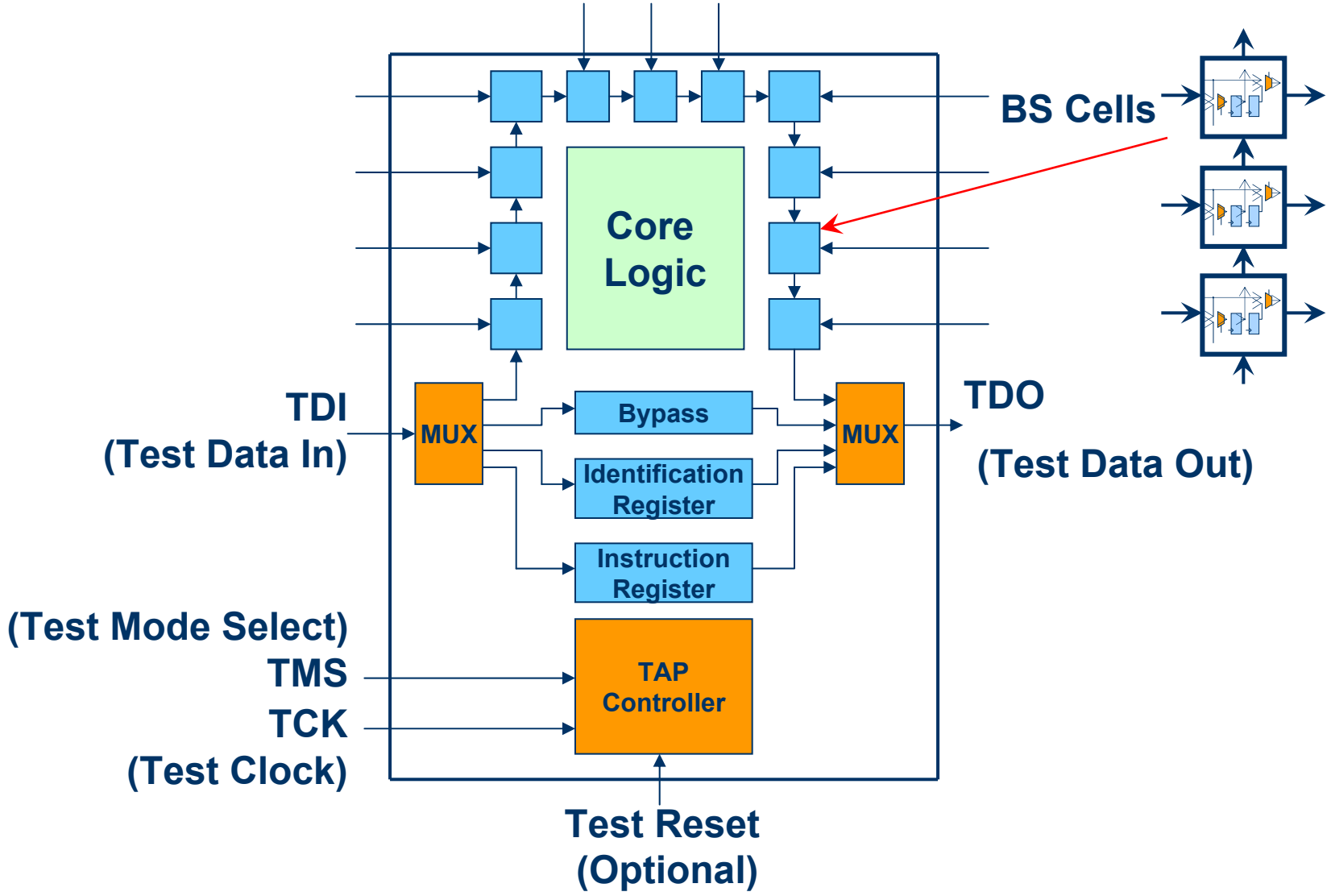
Non-BScan Device



BScan Device

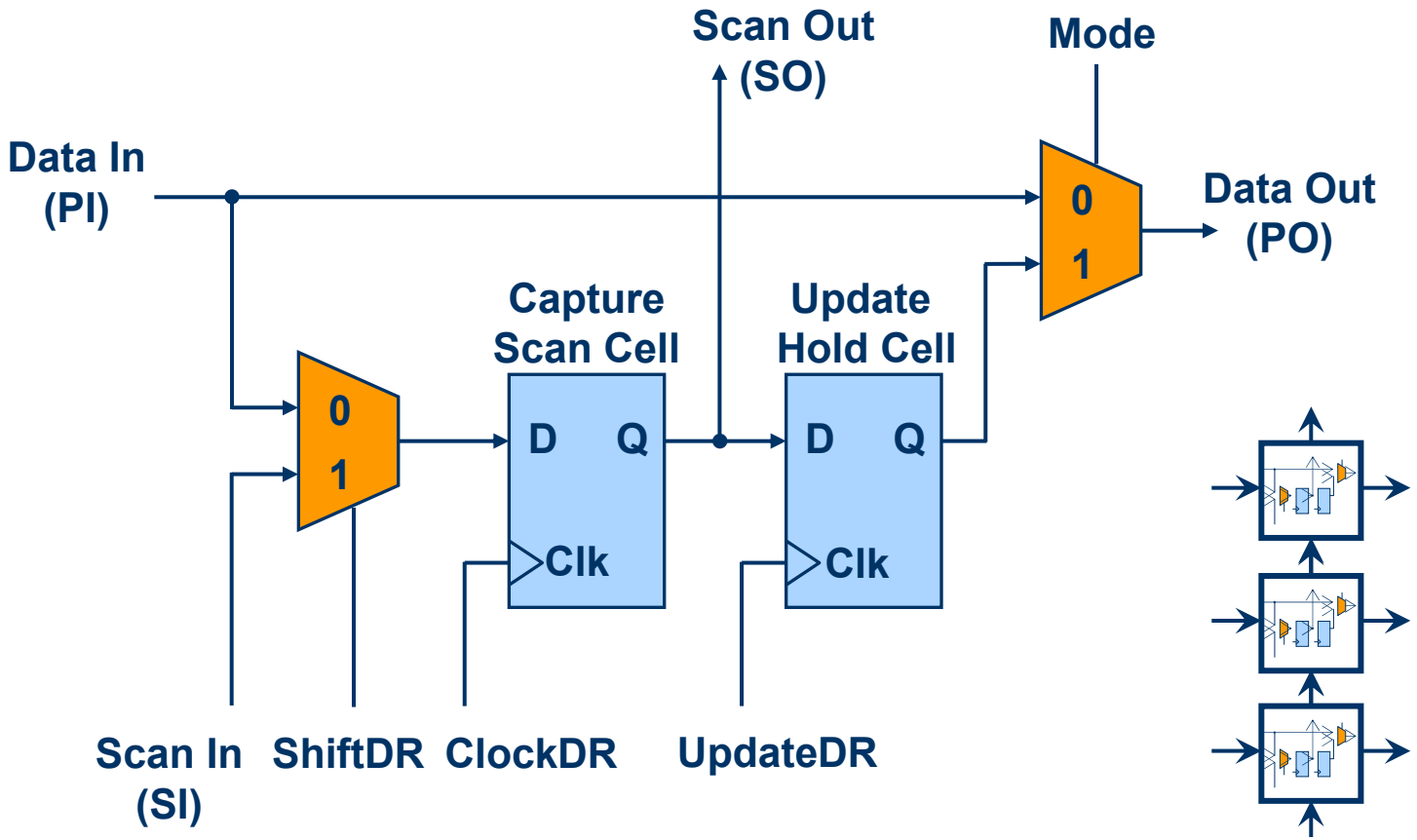
For describing Boundary Scan devices BSDL (Boundary Scan Description Language) models are used

IEEE 1149.1 Device Architecture



Typical Boundary Scan Cell (BC_1)

BC_1 is used both at input and output pins

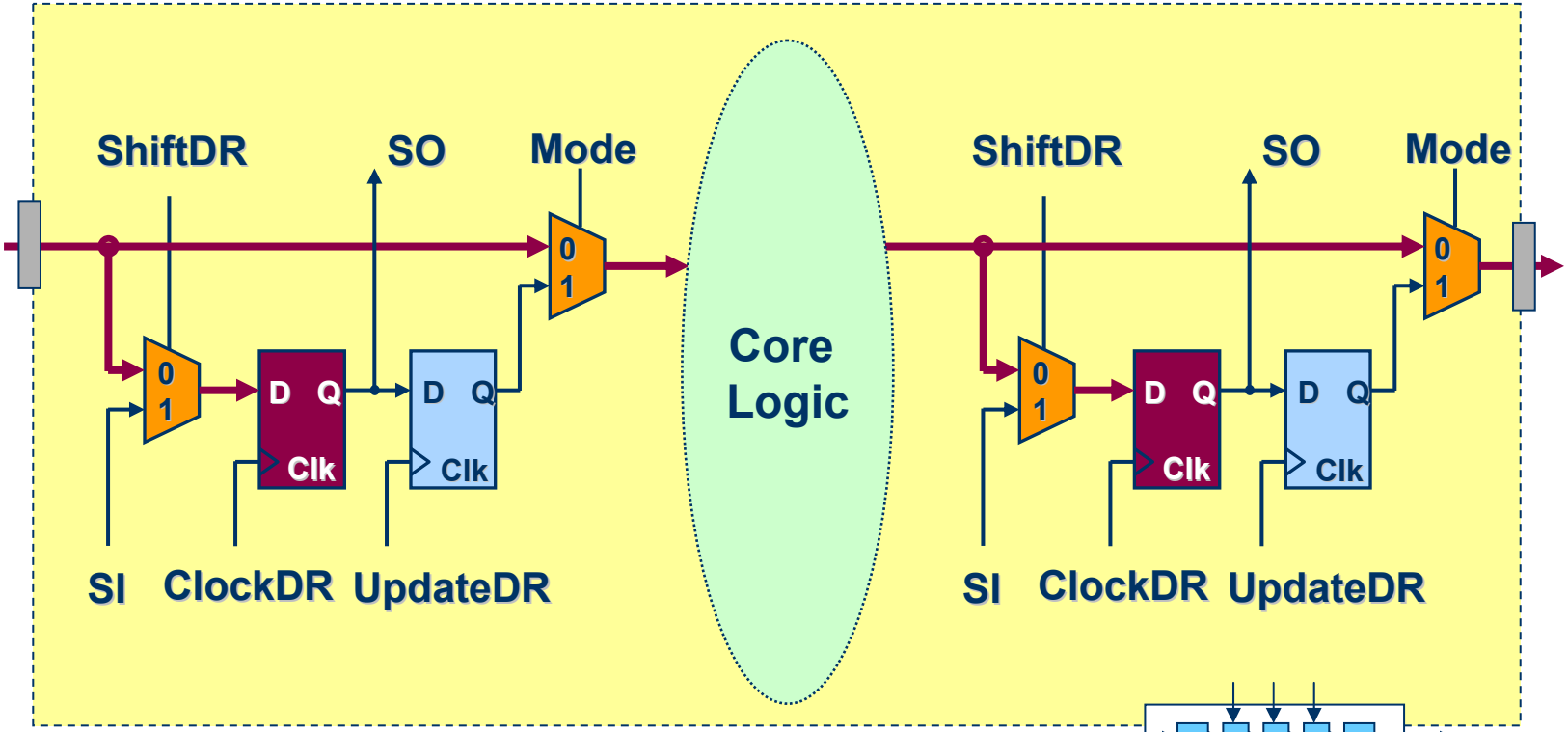


Boundary Scan Instructions

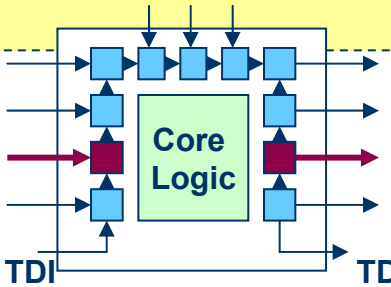
Instruction	Status
SAMPLE /PRELOAD	Mandatory
EXTEST	Mandatory
BYPASS	Mandatory
IDCODE	Optional
INTEST	Optional
CLAMP	Optional
HIGHZ	Optional
RUNBIST	Optional
USERCODE	Optional

Boundary Scan Working Modes

SAMPLE/PRELOAD instruction – sample mode

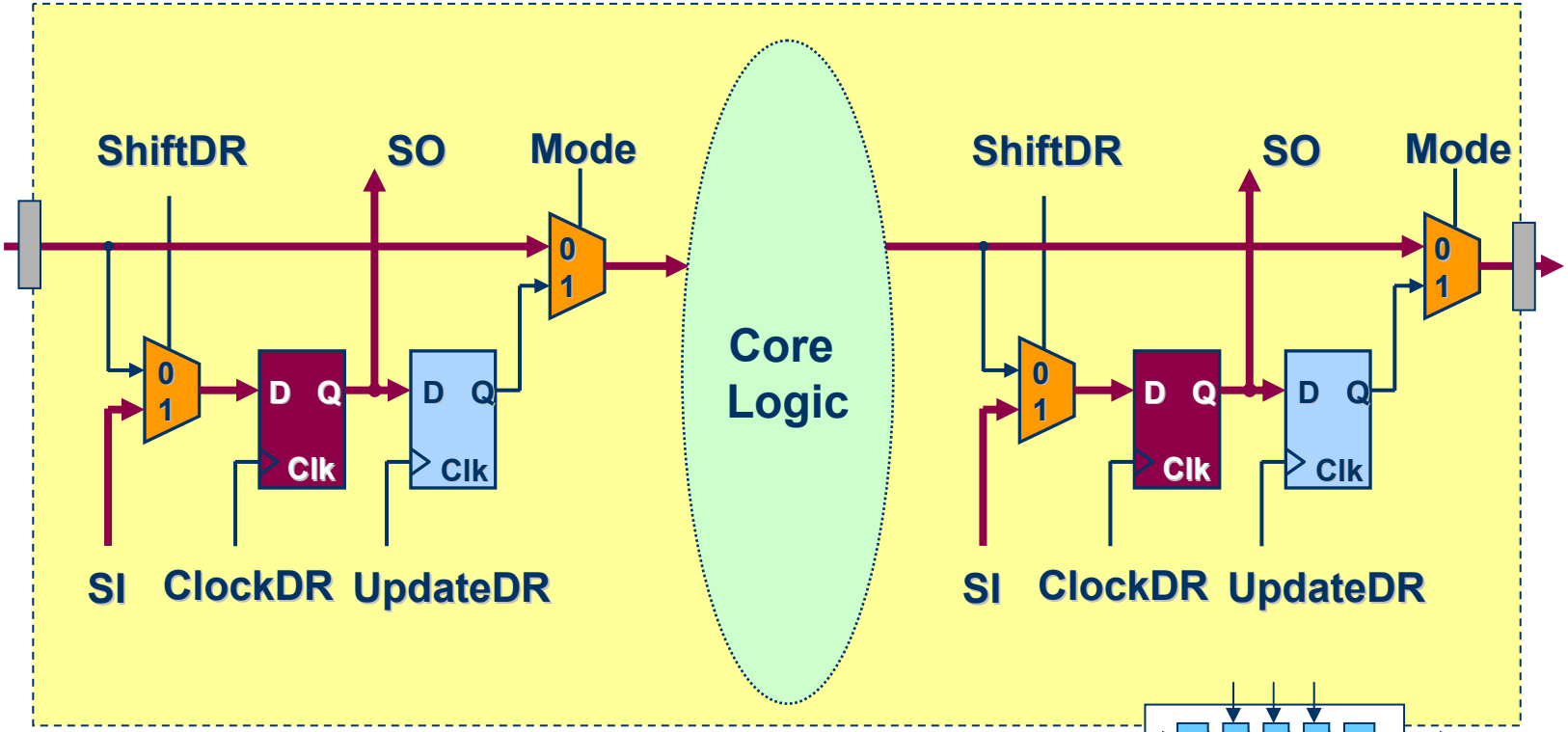


Get snapshot of normal chip output signals

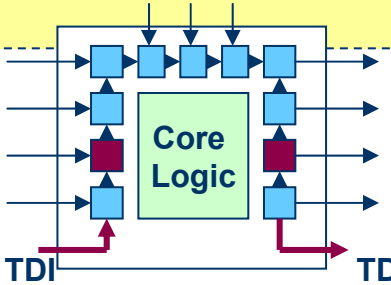


Boundary Scan Working Modes

SAMPLE/PRELOAD instruction – preload mode

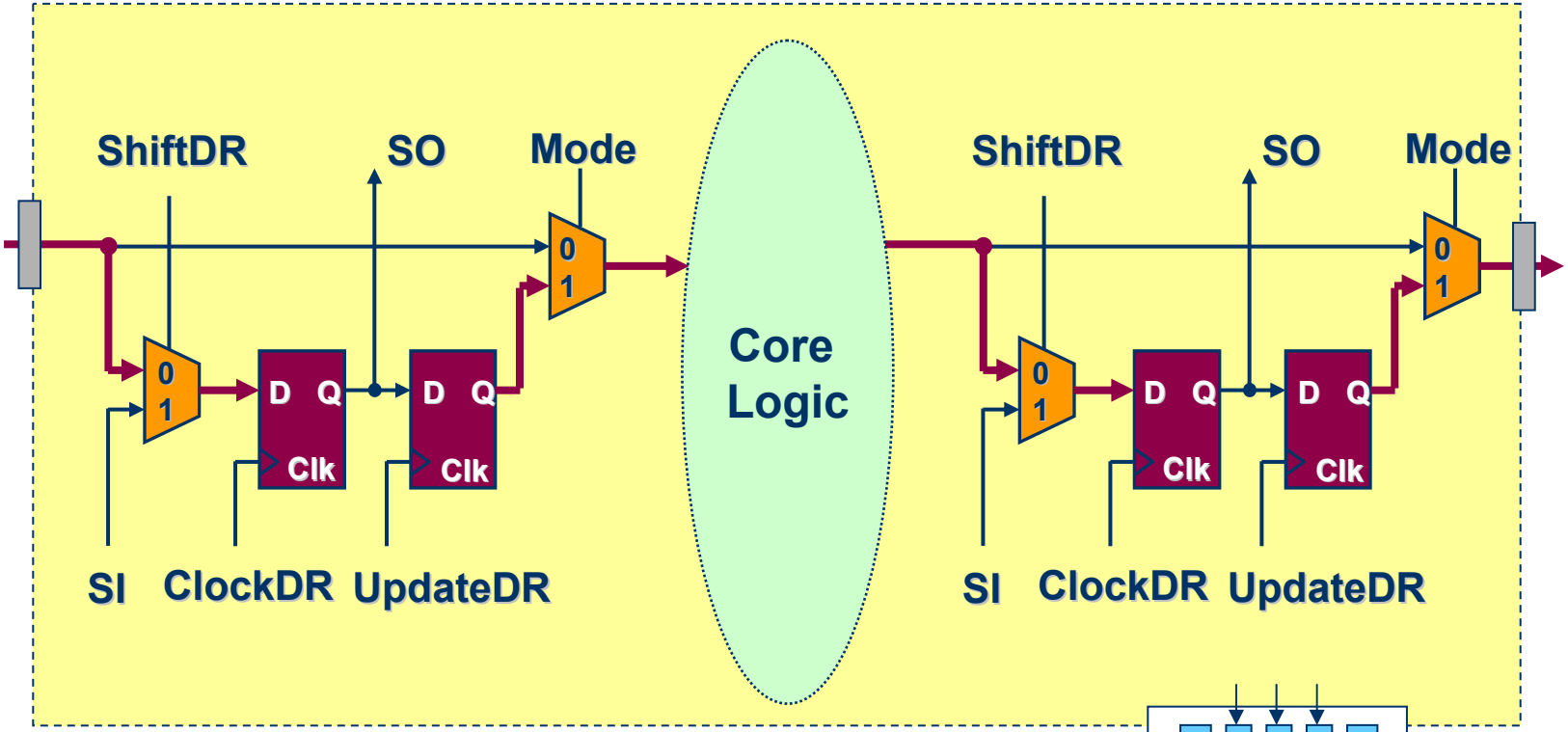


Shift out snapshot data and shift in new test data to be used later

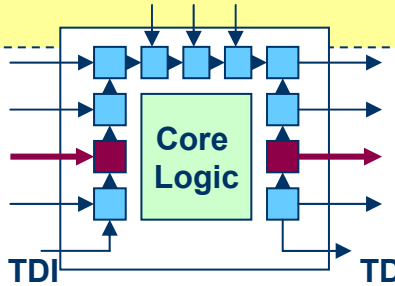


Boundary Scan Working Modes

EXTEST instruction – driving and sensing:

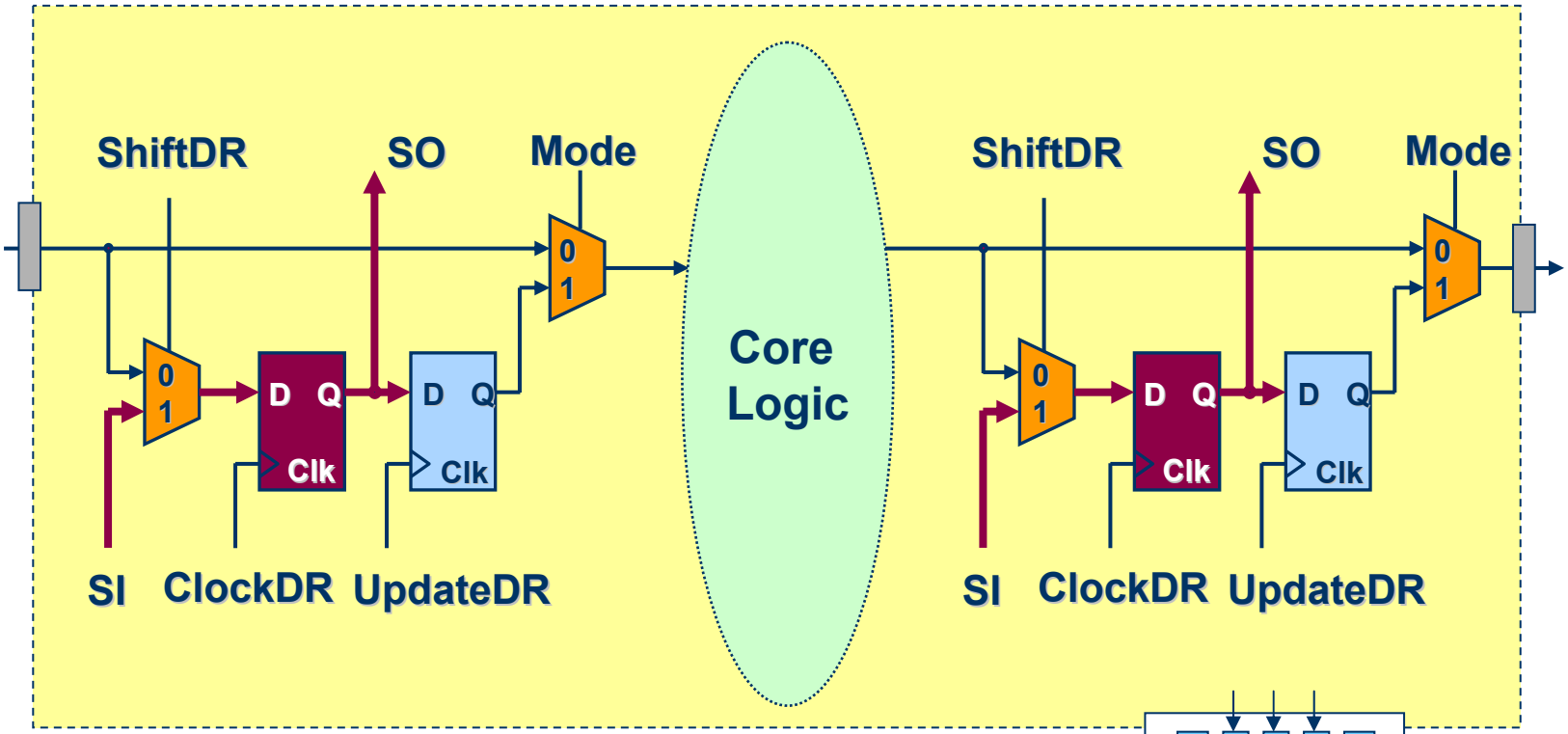


Test off-chip circuits and board-level interconnections

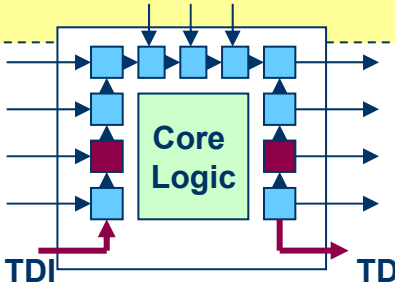


Boundary Scan Working Modes

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

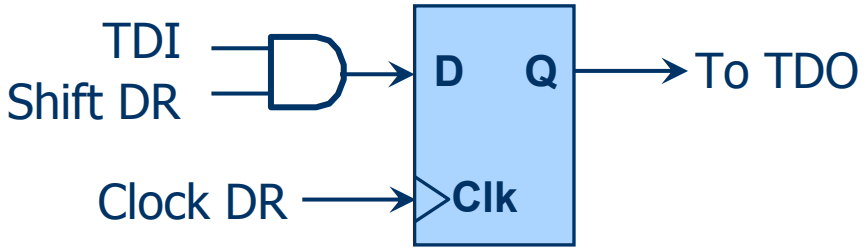
Time

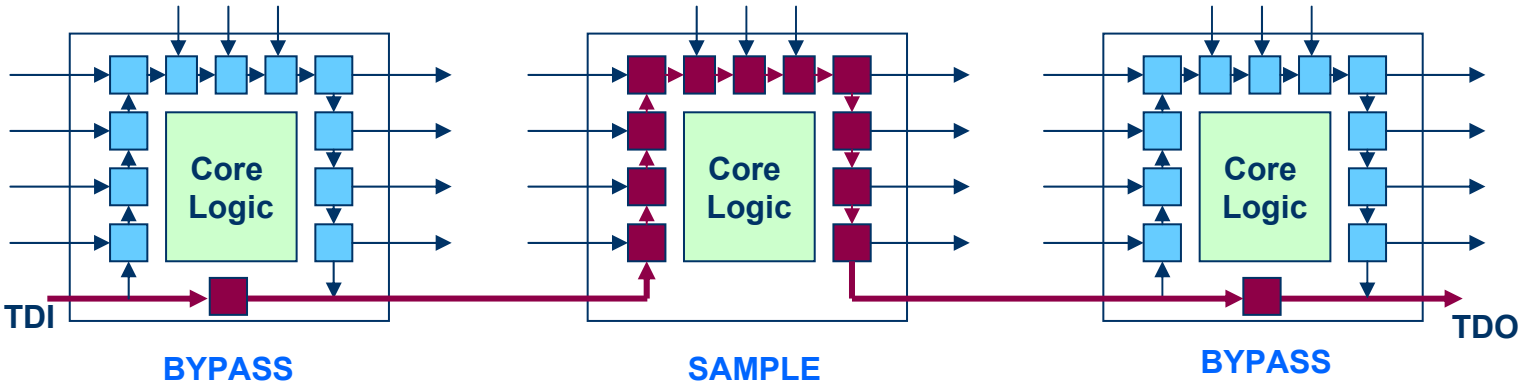
N test vectors: (N+1) DRshifts + 2 IRshifts ≈ (N+1) DRshifts

Boundary Scan Working Modes

BYPASS instruction:

Bypasses the corresponding chip using 1-bit register





Similar instructions: CLAMP, HIGHZ

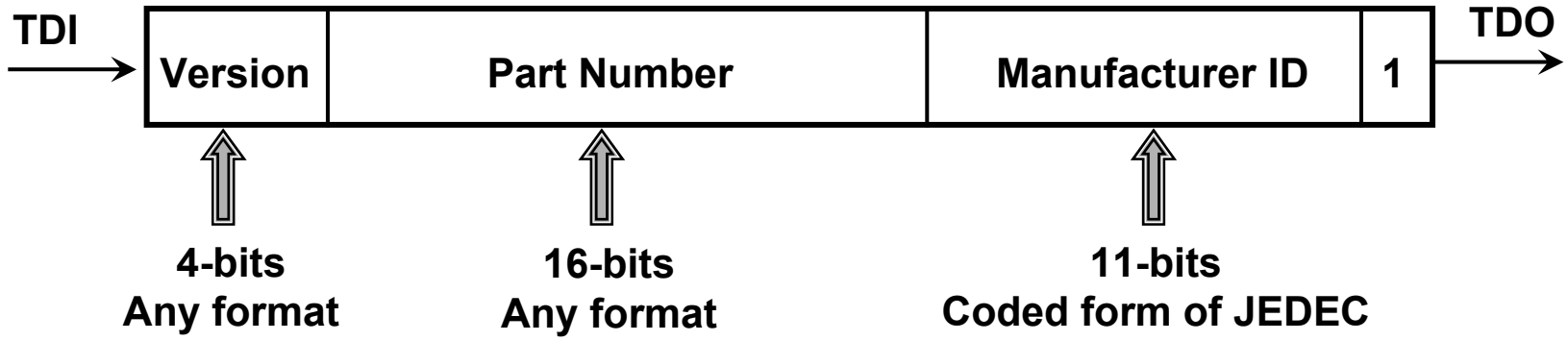
Boundary Scan Working Modes

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

New!

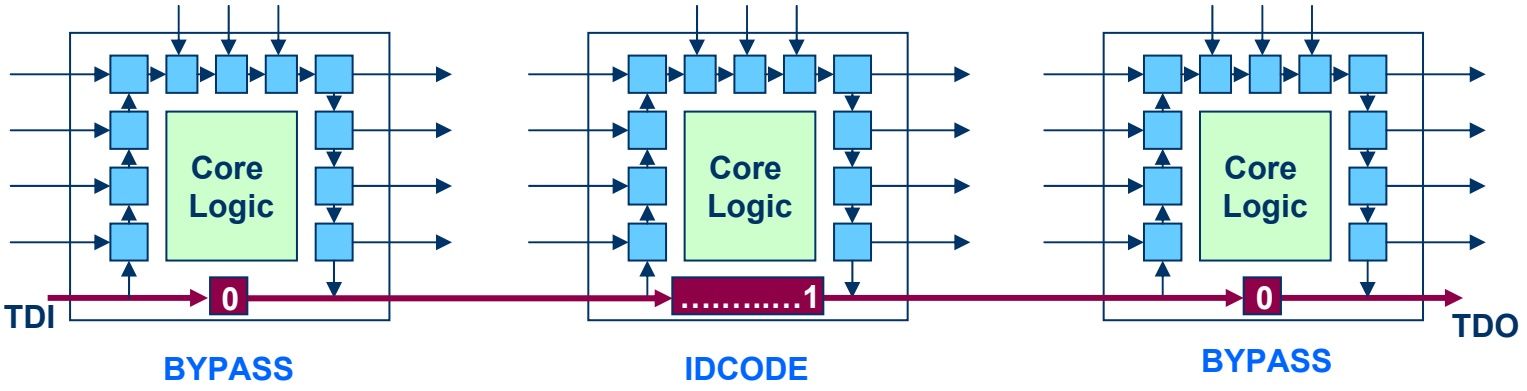
Default instruction:

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

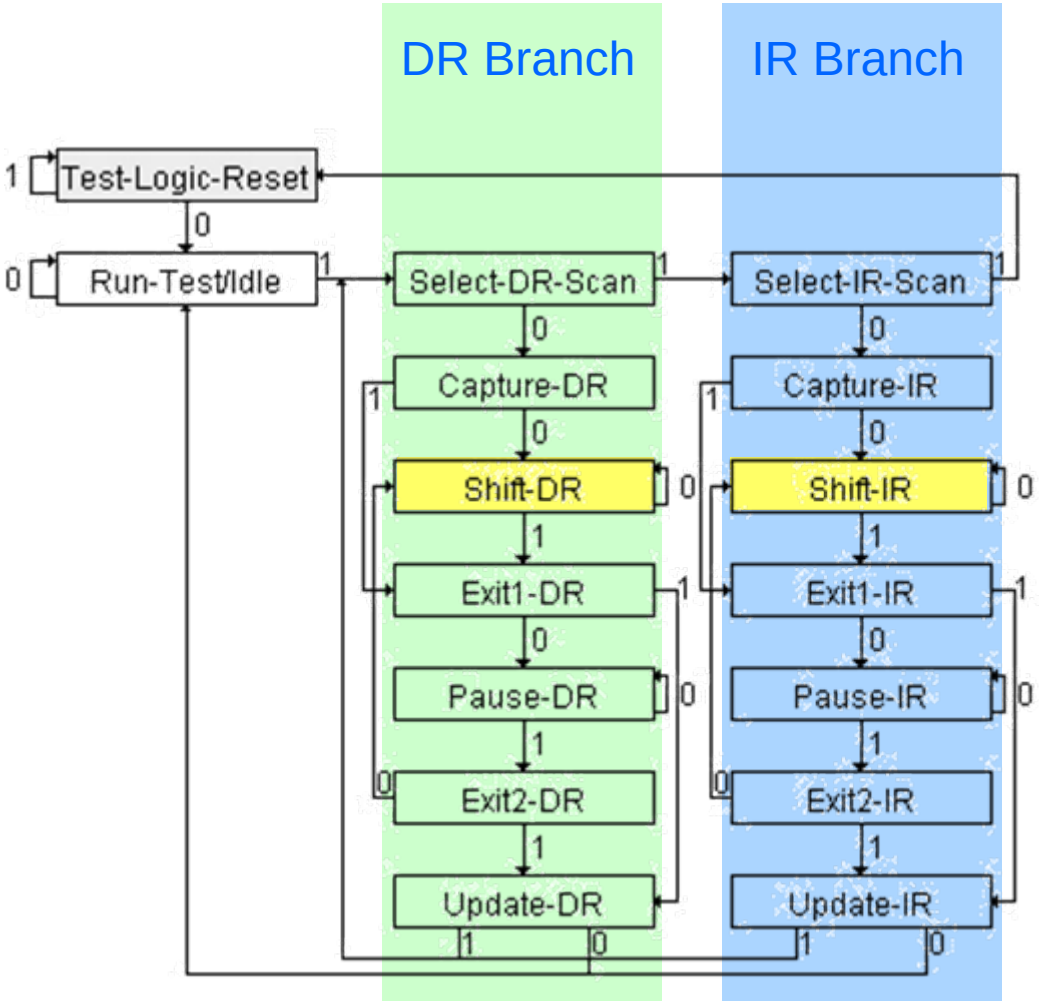
Default capture bits:

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

Example: 0.....10



TAP Controller State Diagram



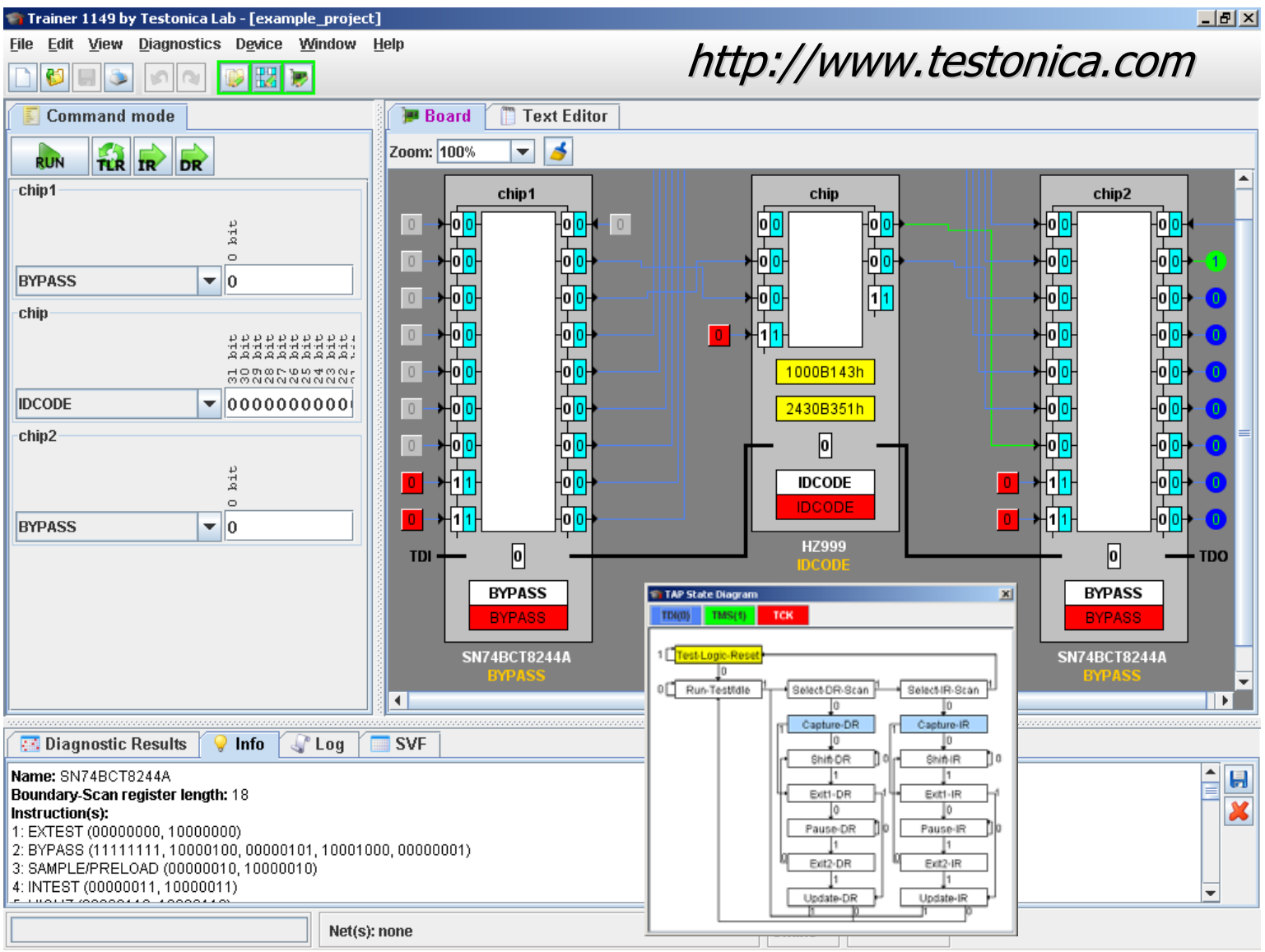
The TAP state diagram has two main branches and two idle states.

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)



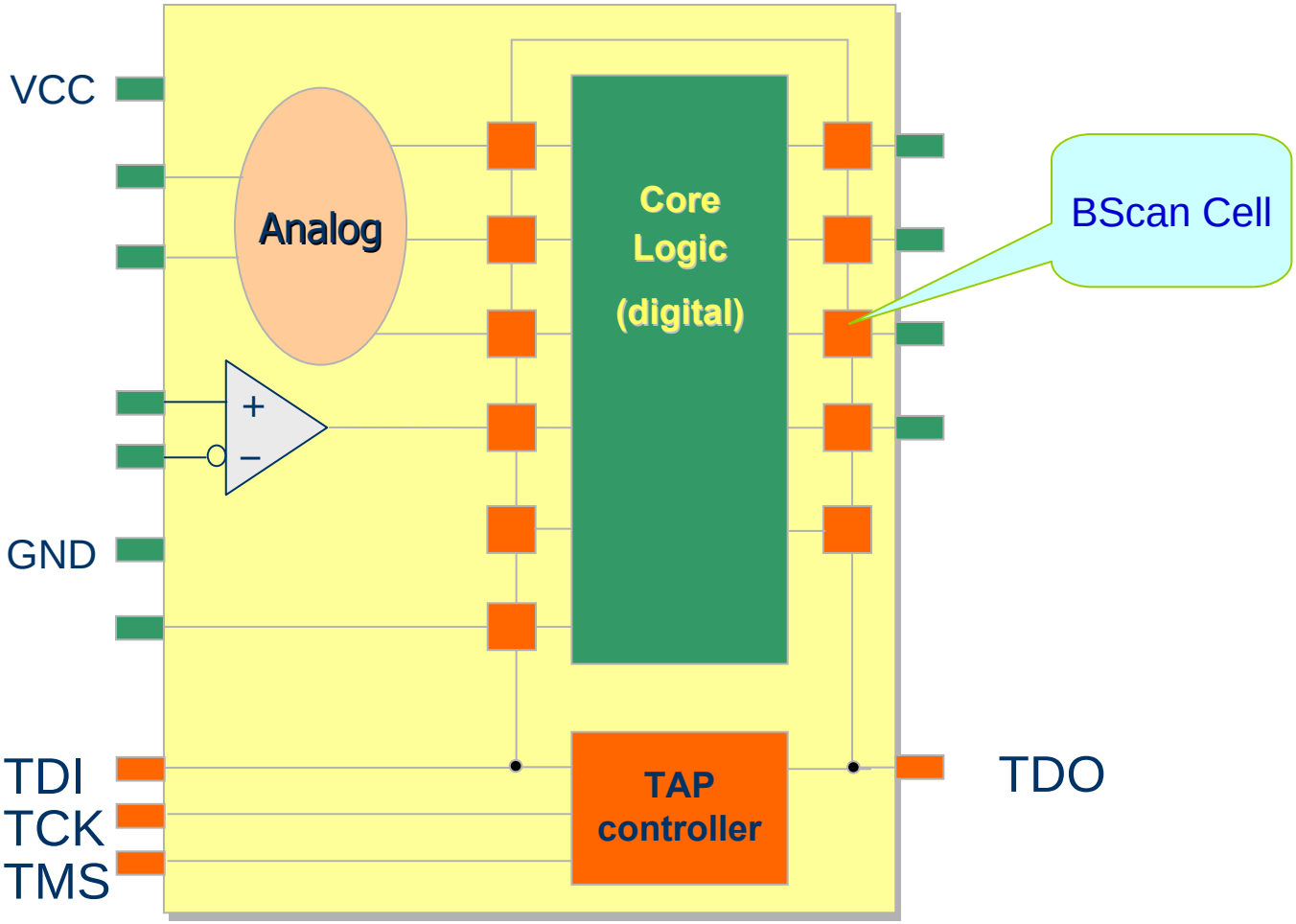
IEEE 1149.1 Boundary Scan Standard

- 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

BSC implementation

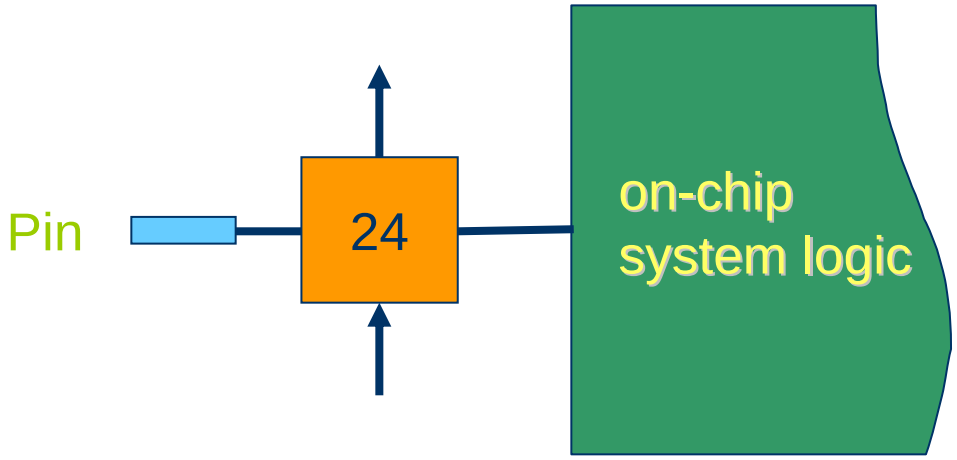
- One or more BSC at each system input or output of on-chip system logic (core logic)
- BSC may be connected to chip-internal signals
- No BSC on:
  - TAP pins (TCK, TMS, TDI, TDO, TRST)
  - Compliance Enable Pins
  - Non-digital pins (e.g. analog pins, power pins)
- No logic between BSC and I/O pin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

BSC implementation



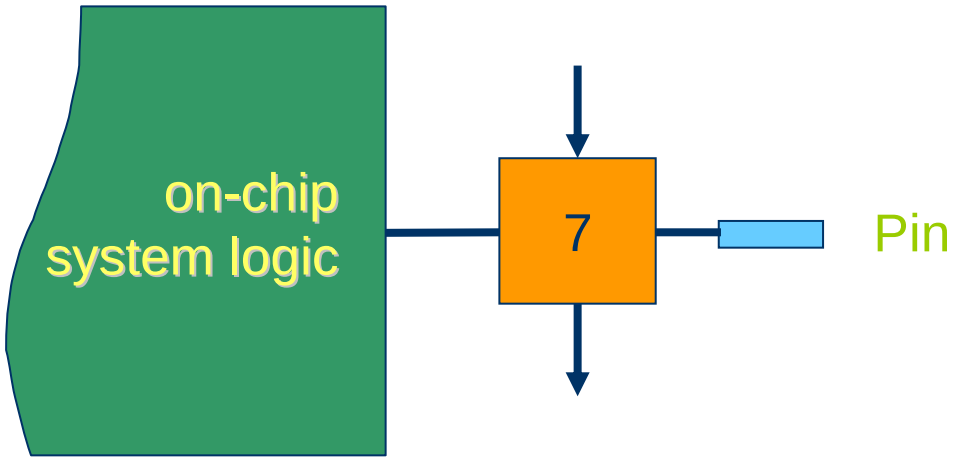
BSC implementation

Input:



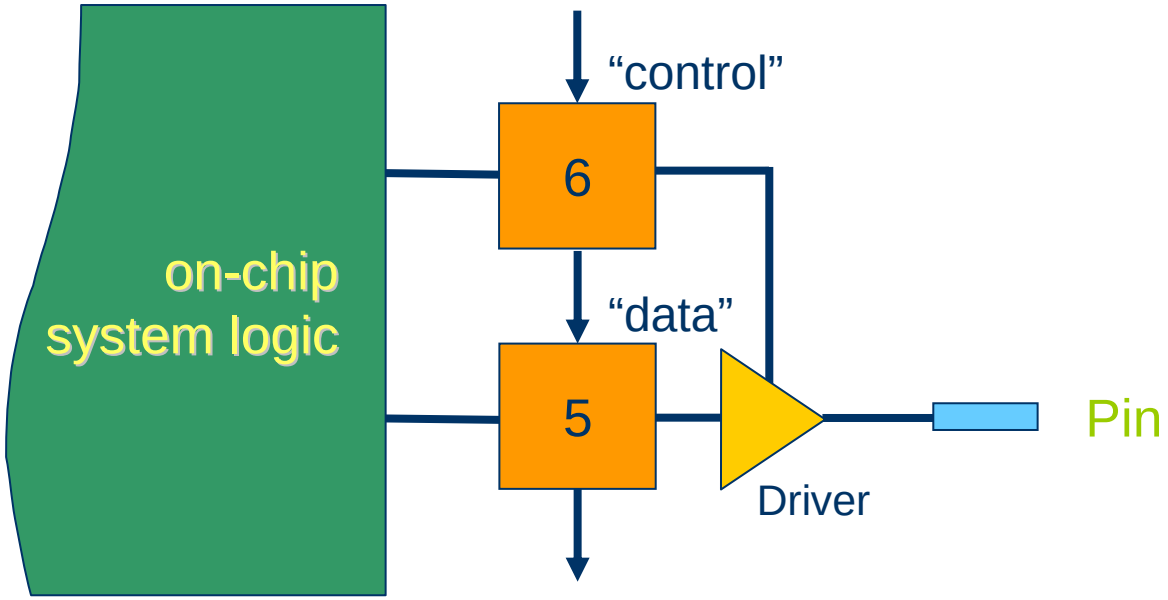
BSC implementation

2-state output:



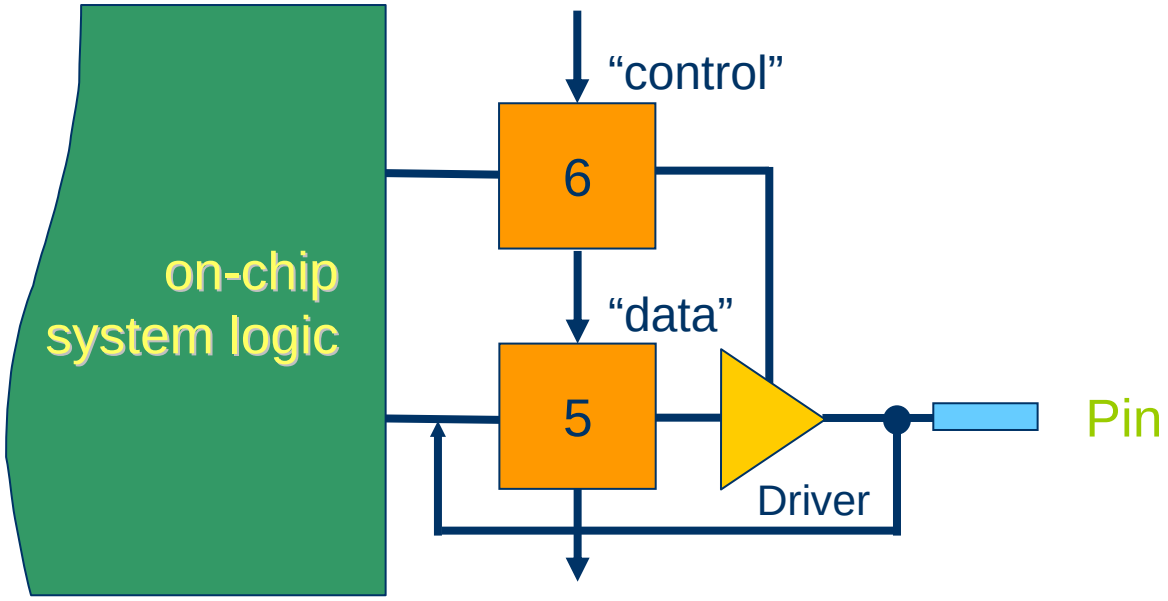
BSC implementation

3-state output:



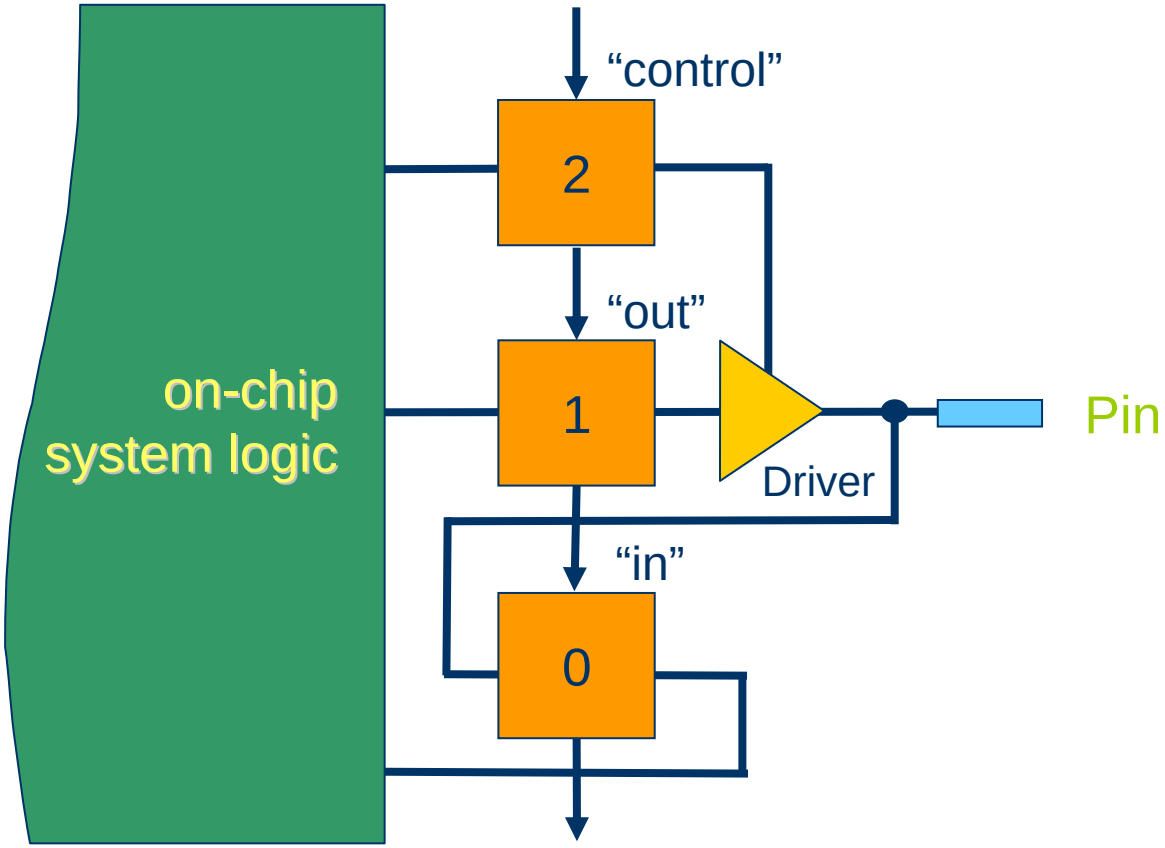
BSC implementation

Bi-directional, 2-cell:



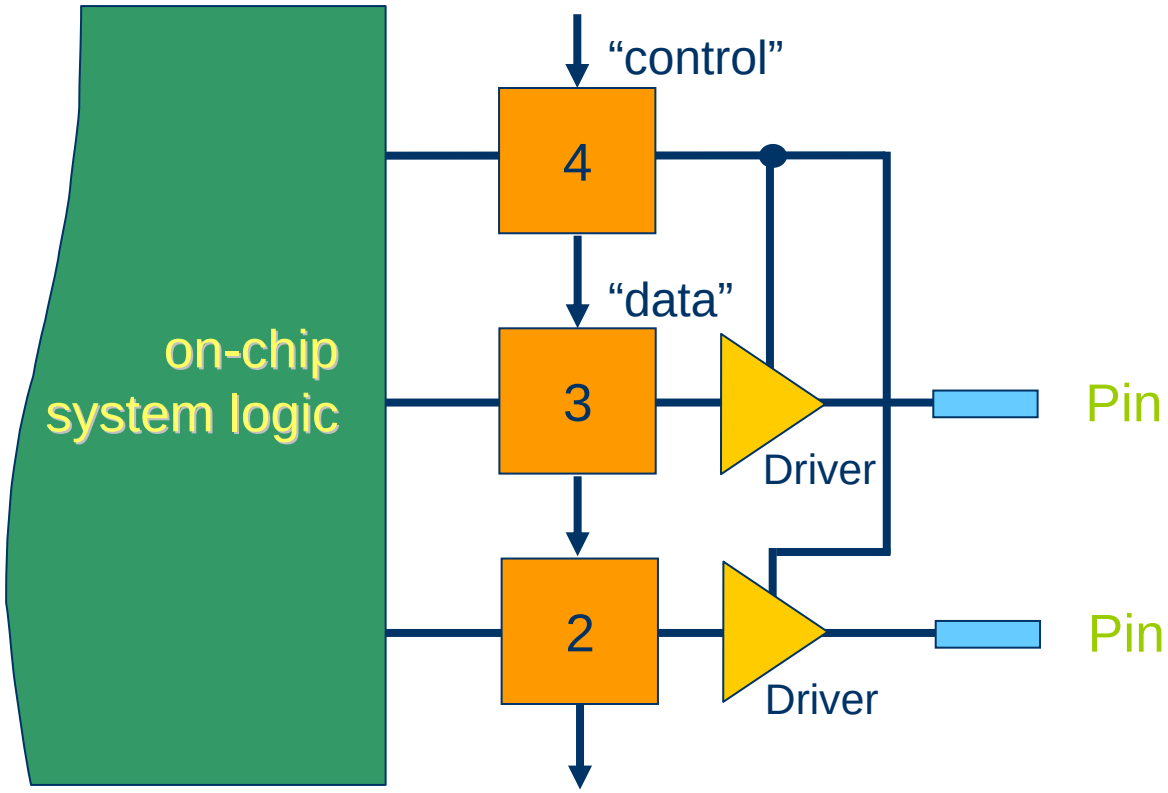
BSC implementation

Bi-directional, 3-cell:

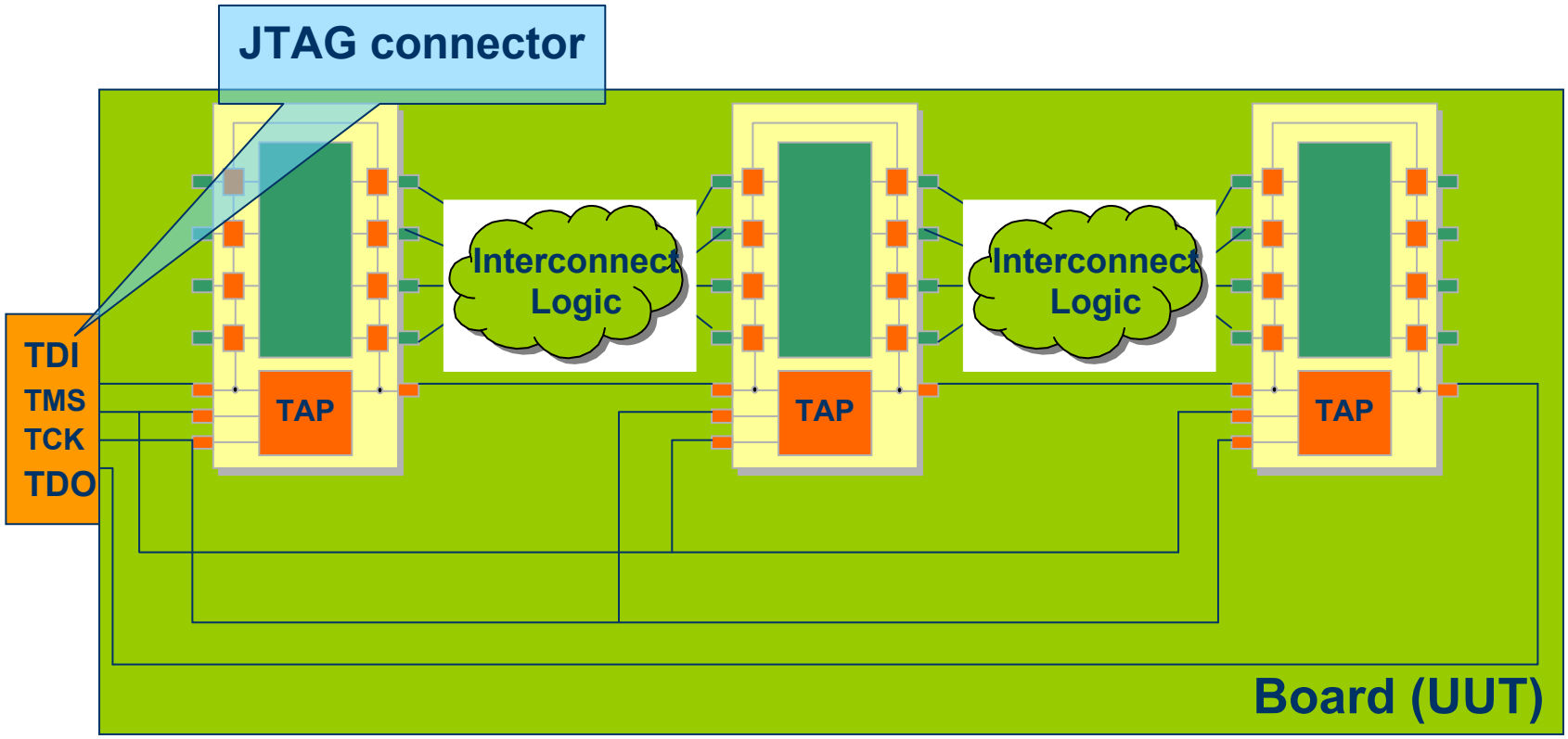


BSC implementation

3-state outputs, shared control cell:



Board-level Test using Boundary Scan (in theory)



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

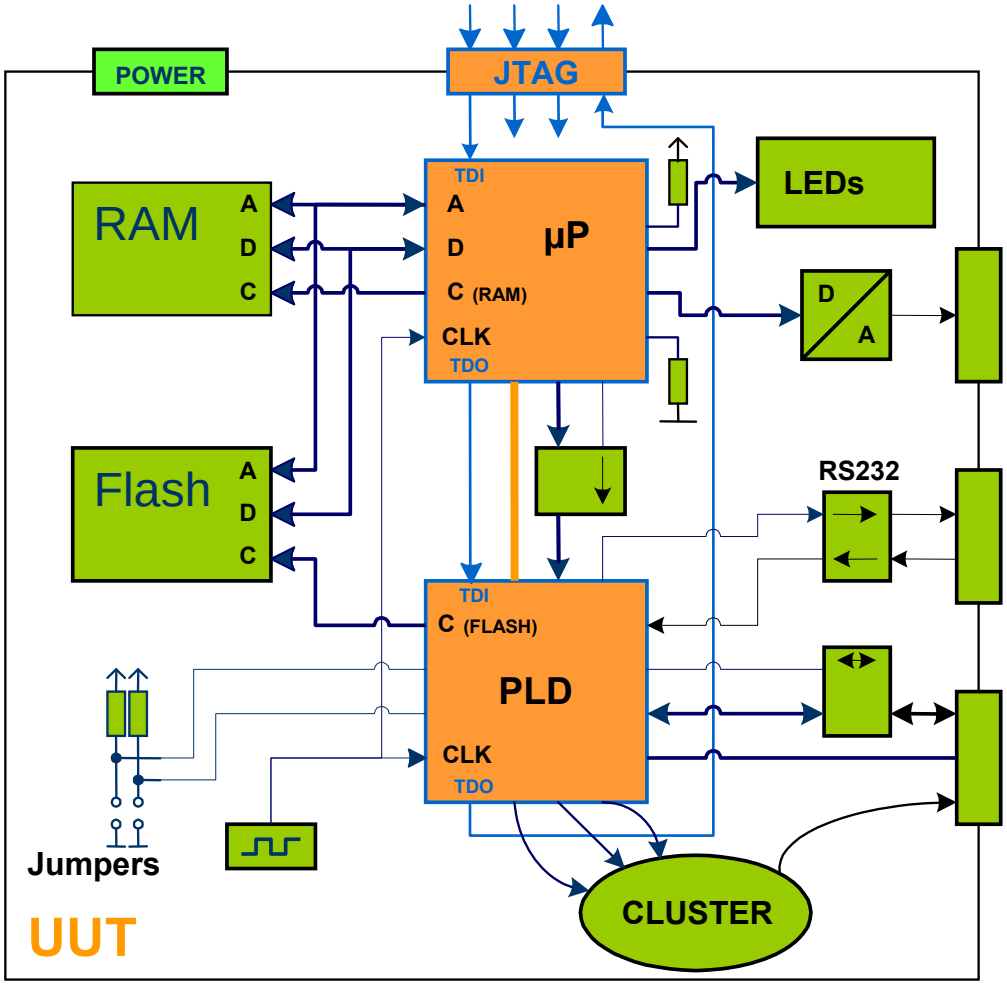
RAM Test

External connectors test

LED or display test (can be assisted by a camera/sensor)

FLASH test/program/read ID – in-system programming

Board-level Test using Boundary Scan

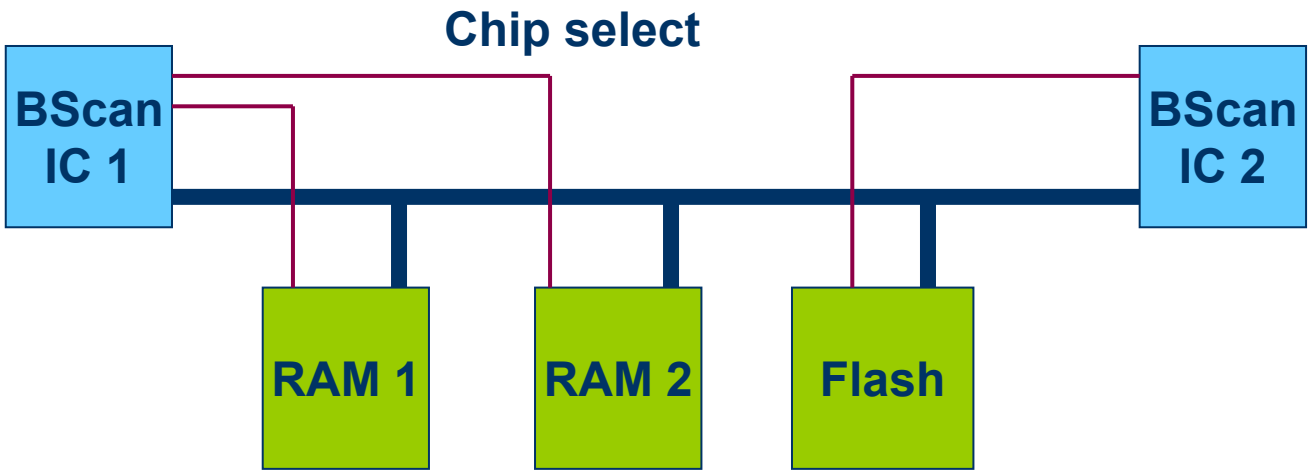


- Microprocessor
- PLD

- RAM
- Flash
- Clock generator
- Cluster logic
- Buffers/MUXes
- Pull-up/Pull-down resistors
- Jumpers
- D/A converter
- External connectors (analog and digital)

Interconnect Test through Clusters

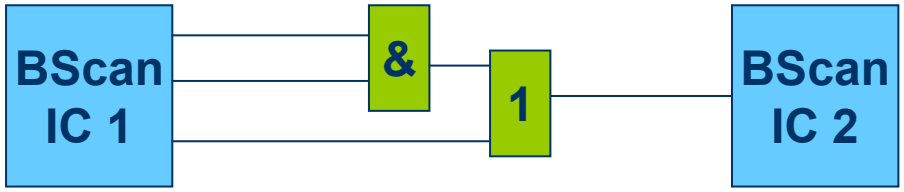
RAM / Flash Test



To generate test:

- Specify constraints that will select only one device
- RAM/Flash model in special format that provides description of read/write protocol

Cluster Logic Test (Manual)

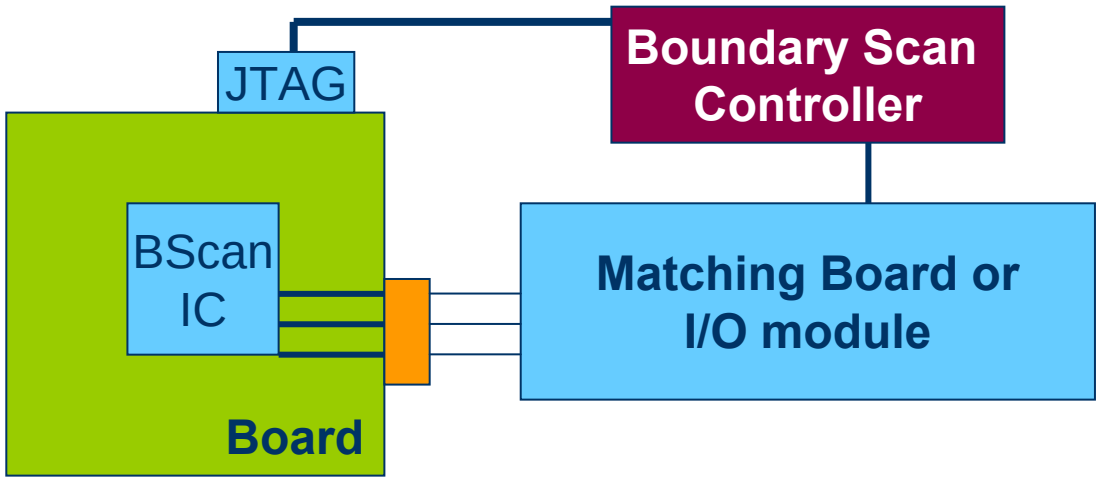


Provide cluster's truth table

Truth table

000 0
001 1
010 1
…
111 1

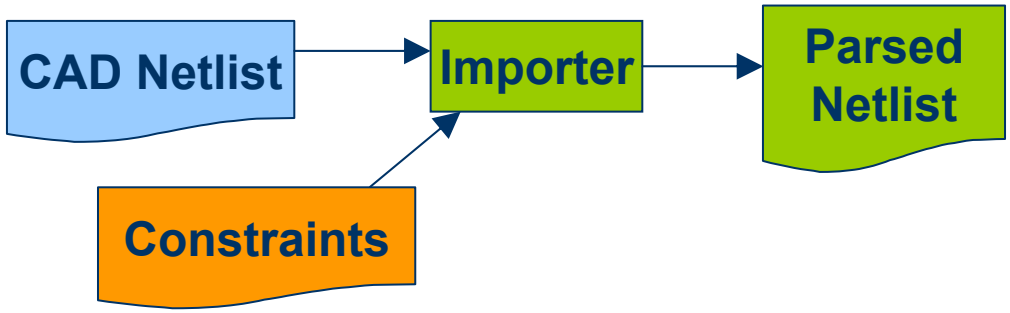
External connectors



Only interconnect test will be performed!

The real protocol of external connector is not tested

Boundary Scan Test Development
Typical workflow

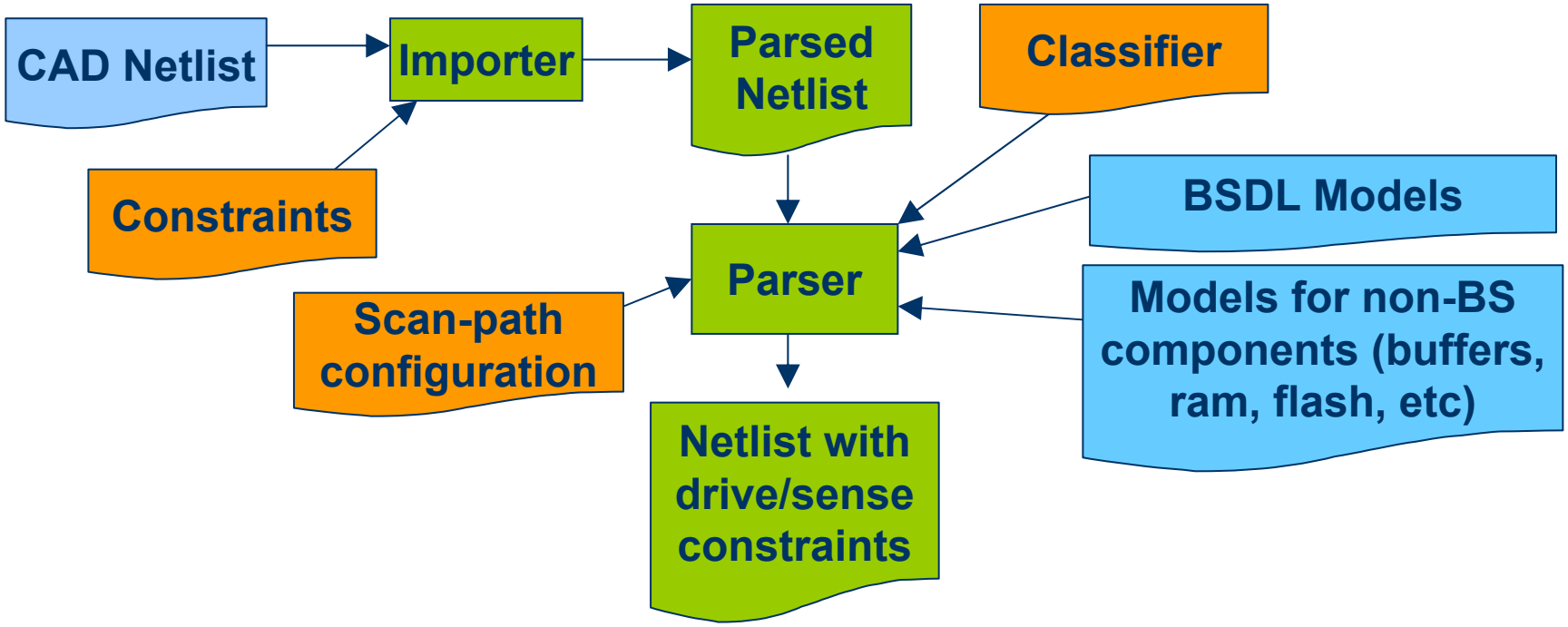


- Description of boards come in different formats (depends on CAD system used by designer) ➔
- CAD Import is the first step

Common problems:

- Netlist doesn't fully correspond to board
- CAD Importer does not work correctly

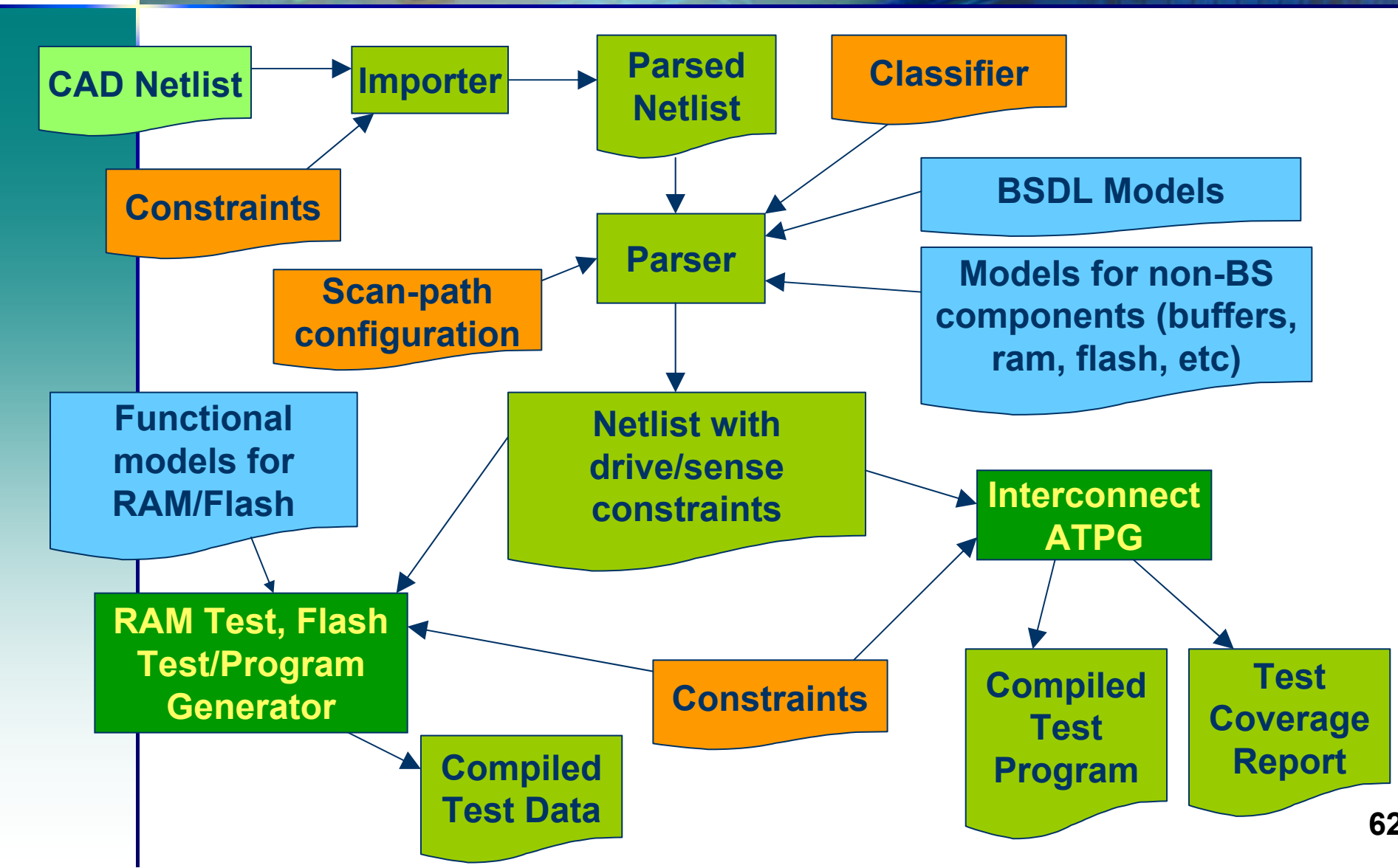
Boundary Scan Test Development
Typical workflow



Common problems:

- BSDL file is not available
- Model of non-BS component is absent in the library

Boundary Scan Test Development Typical workflow



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)

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:
  - Trainer 1149 by Testonica Lab (http://www.testonica.com/1149/download.htm)
  - Scan Coach by Goepel Electronic (http://www.goepel.com/content/html_en/index.php?site=bs_BScanCoach)
  - 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
- Active Universities:
  - Porto, Portugal
  - Tallinn, Estonia
  - Liberec and Prague, Czech Republic
  - Lubljana, Slovenia