Lecture 24

EE 421 / CS 425
Digital System Design

Fall 2023

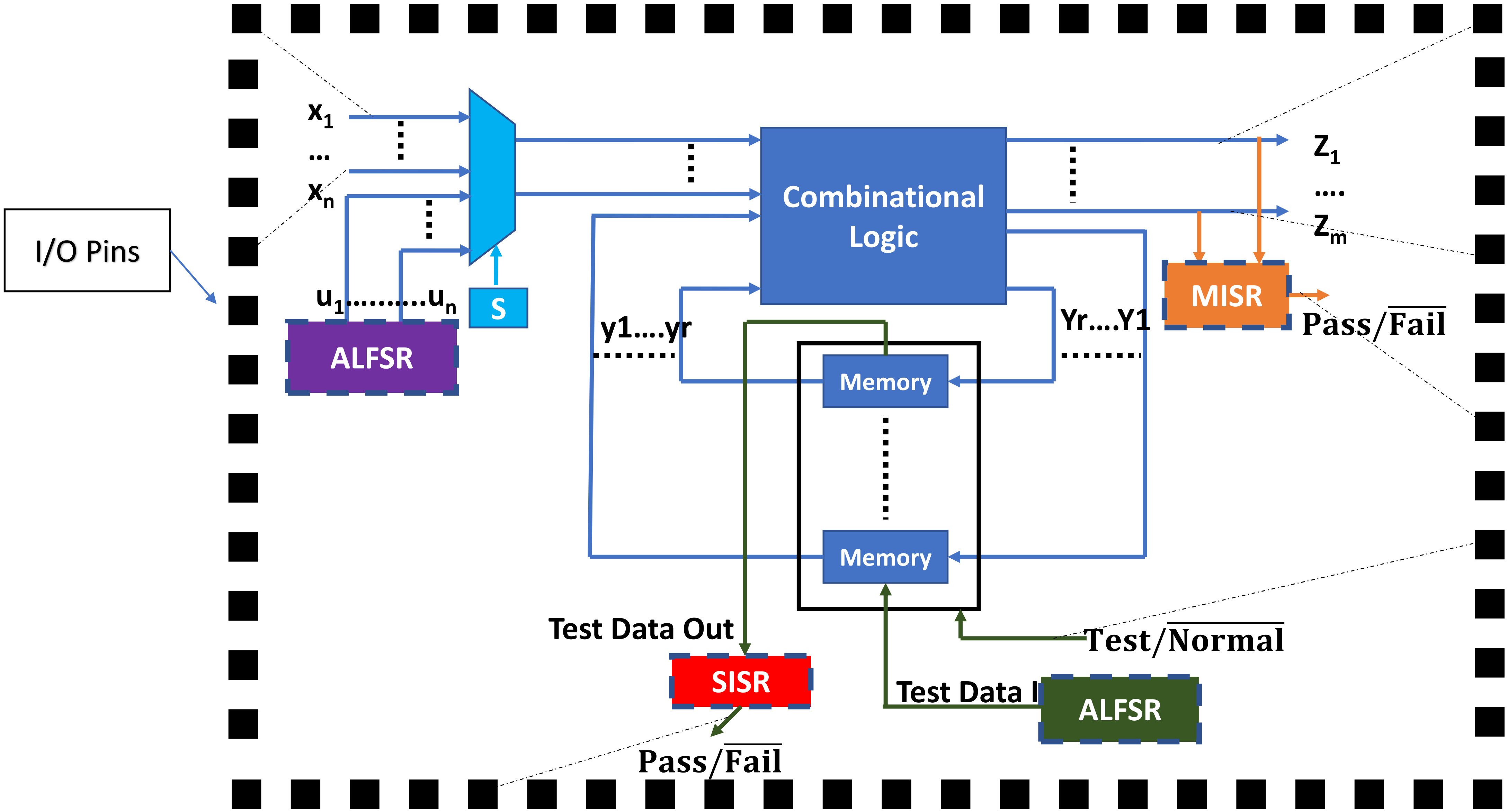
Shahid Masud

Topics

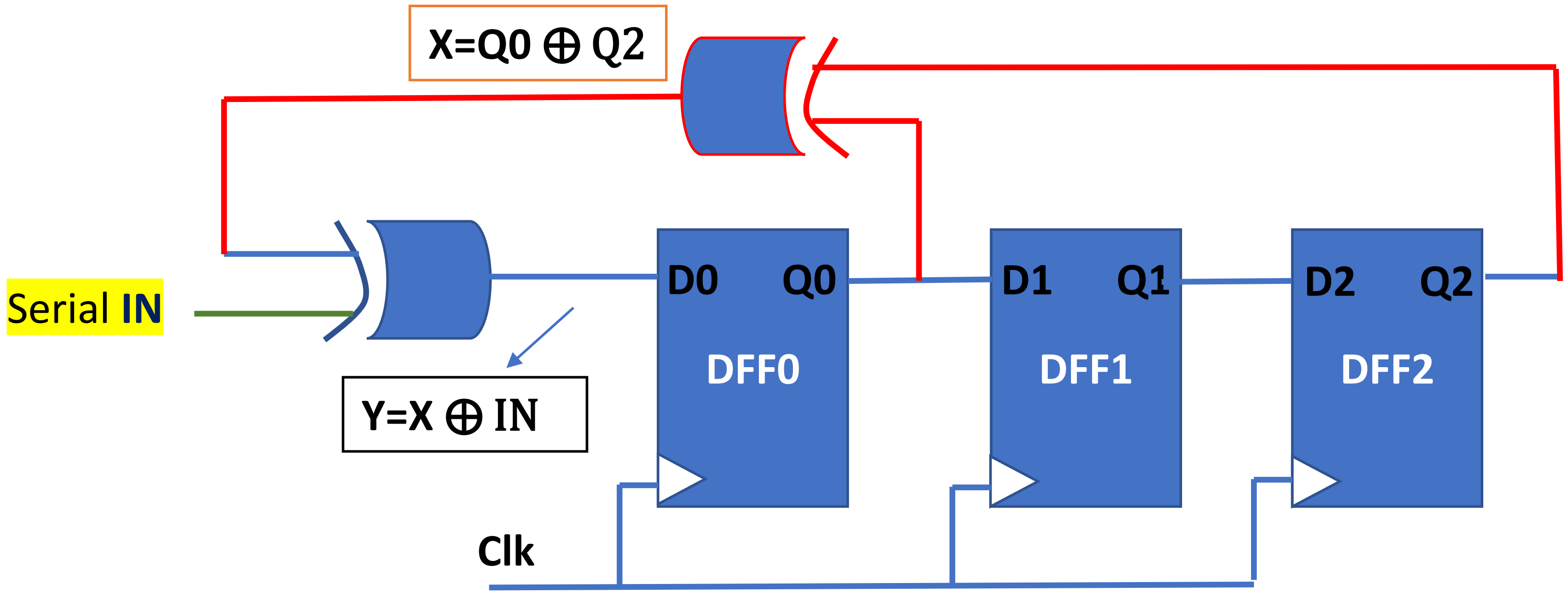
- **Recap: BIST and its Components**
- **SISR** – Serial Input Shift Signature Register
- **MISR** – Multi Input Signature Register
- **BILBO** – Built-In Logic Block Observer
- **Boundary Scan**
- **JTAG Standard**

QUIZ 5 today

Recap: Designing with BIST Circuits



SISR – Serial Input Shift Register



- The flip flops and feedback XOR gate form an ALFSR with some seed value
- The output from XOR is taken through another XOR with Serial IN bit stream
- The circuit continues to advance one bit at a time, with each clock edge
- Once all bits have been read through Serial IN, final status of Q0, Q1, and Q2 represents a signature pattern
- The pattern will only re-appear if the same bit pattern is presented at input
- No other pattern will produce the same final signature output at registers

SISR Example

Eg., Let seed value = ‘000’

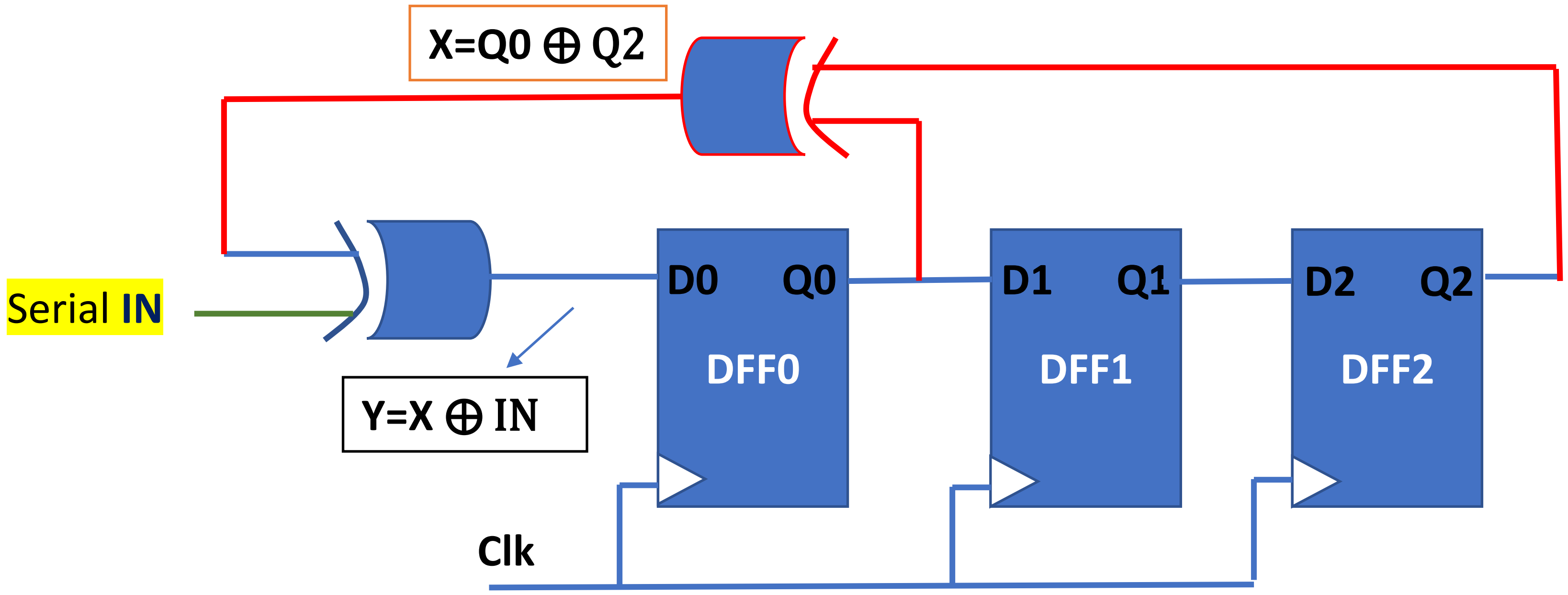
Generate Signature for input bit patterns ‘**011101**”

Clk	IN	X	Y	Q0	Q1	Q2
1	0	0	0	0	0	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	0	1
6	1	0	1	0	1	0
7	END			1	0	1

Signature produced

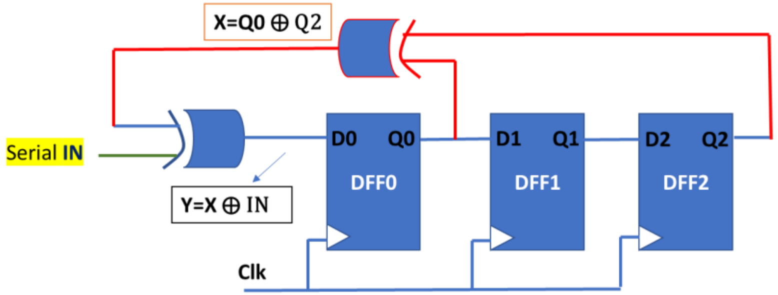
More register bits, and different seed value and different feedback XOR will generate new signatures

SISR – Serial Input Signature Register



- The flip flops and feedback XOR gate form an ALFSR with some seed value
- The output from XOR is taken through another XOR with Serial IN bit stream
- The circuit continues to advance one bit at a time, with each clock edge
- Once all bits have been read through Serial IN, final status of Q0, Q1, and Q2 represents a signature pattern
- The pattern will only re-appear if the same bit pattern is presented at input
- No other pattern will produce the same final signature output at registers

SISR Example



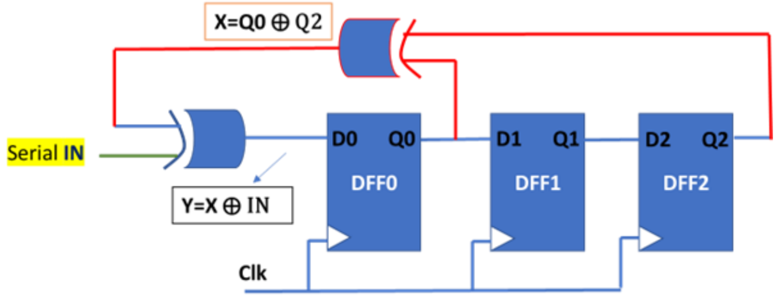
Eg., Let seed value = ‘000’
Generate Signature for input bit patterns ‘**011101**’

Clk	IN	X	Y	Q0	Q1	Q2
1	0	0	0	0	0	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	0	1
6	1	0	1	0	1	0
7	END			1	0	1

Signature produced

More register bits, and different seed value and different feedback XOR will generate new signatures

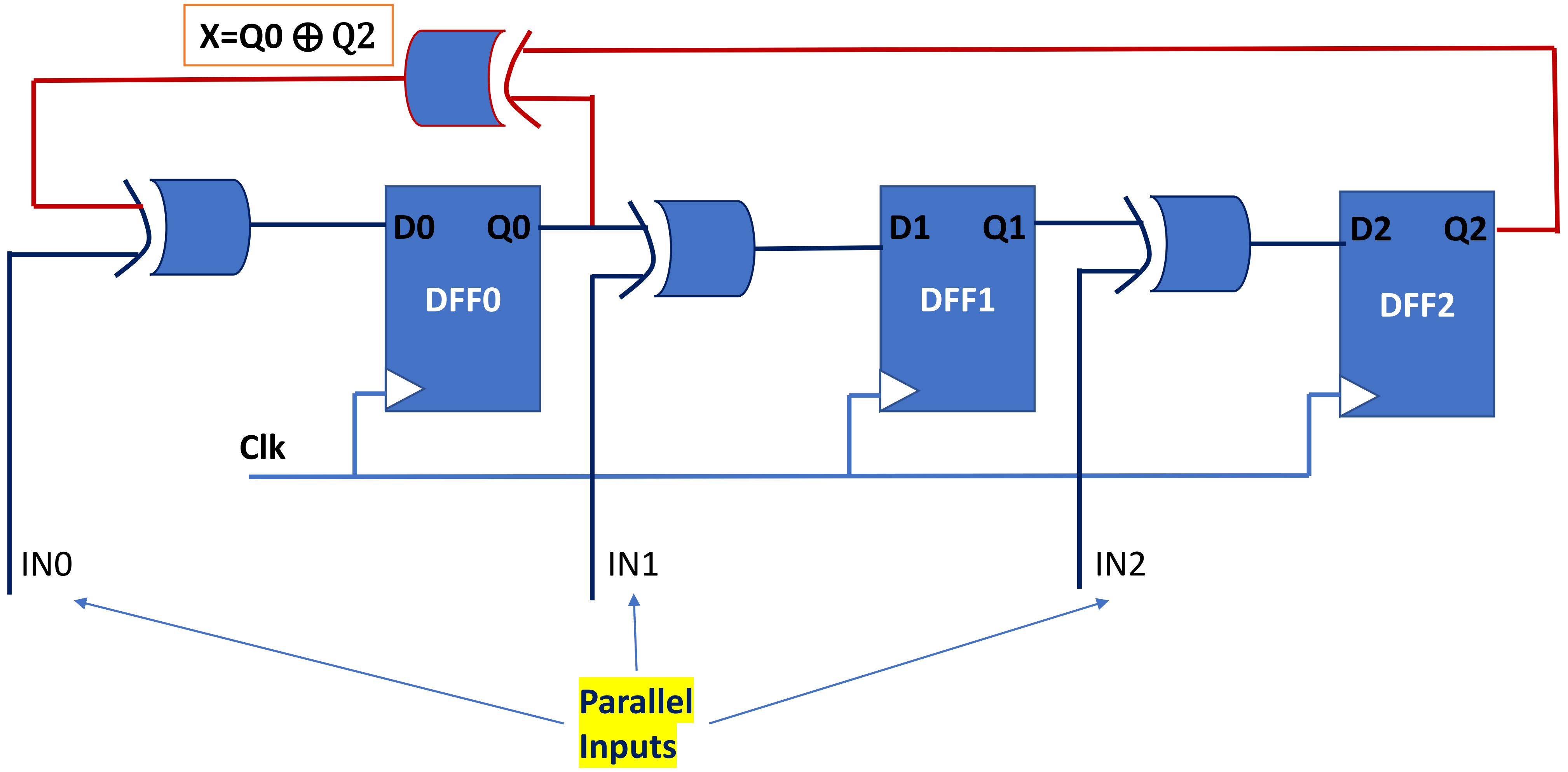
SISR Example 2 – Use sequence ‘011110’, Seed value ‘111’



Clk	IN	X	Y	Q0	Q1	Q2
1	0	0	0	1	1	1
2	1	1	0	0	1	1
3	1	1	0	0	0	1
4	1	0	1	0	0	0
5	1	1	0	1	0	0
6	0	0	0	0	1	0
7				0	0	1

Final Signature

MISR – Multi Input Signature Register



After all parallel inputs have been applied, one input set on each clock, the last status at Q0,Q1,Q2 is the signature

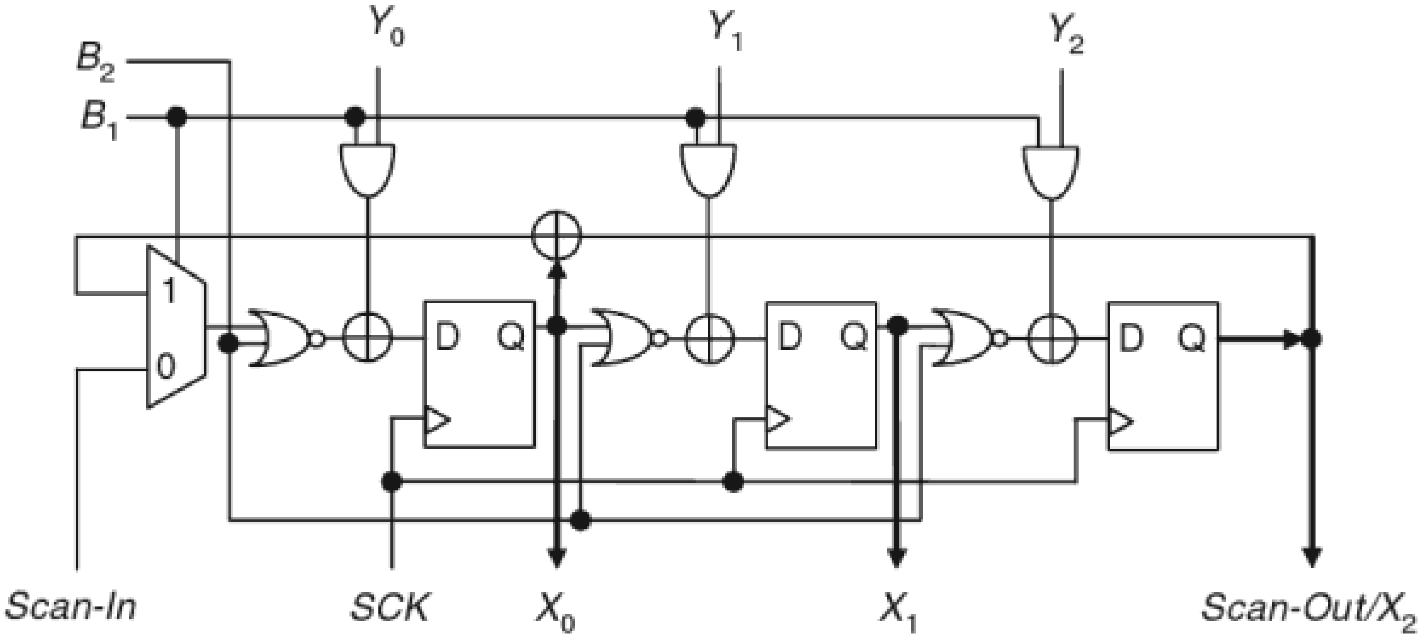
Example: Determine 'signature' when inputs {010, 101, 110, 111, 011} are applied at {IN0,IN1,IN2}

BILBO – Built-In Logic Block Observer

- Combine the following into one circuit:
  - Pseudo Random Pattern Generator (ALFSR)
  - Serial Input Signature Register (SISR)
  - Multi-Input Signature Register (MISR)
  - Scan Registers

BILBO Implementation with 3 Registers

B_1	B_2	Operation mode
1	1	Normal
0	0	Scan
1	0	Mixed Test Generation and Signature Analysis
0	1	Reset



A BILBO Circuit Example

B1	B2	Operating Mode
0	0	Shift Register
0	1	PR PG (Pattern Gen)
1	0	Normal
1	1	MISR

Line Colour

Red = feedback ALFSR

Orange = Control Selection

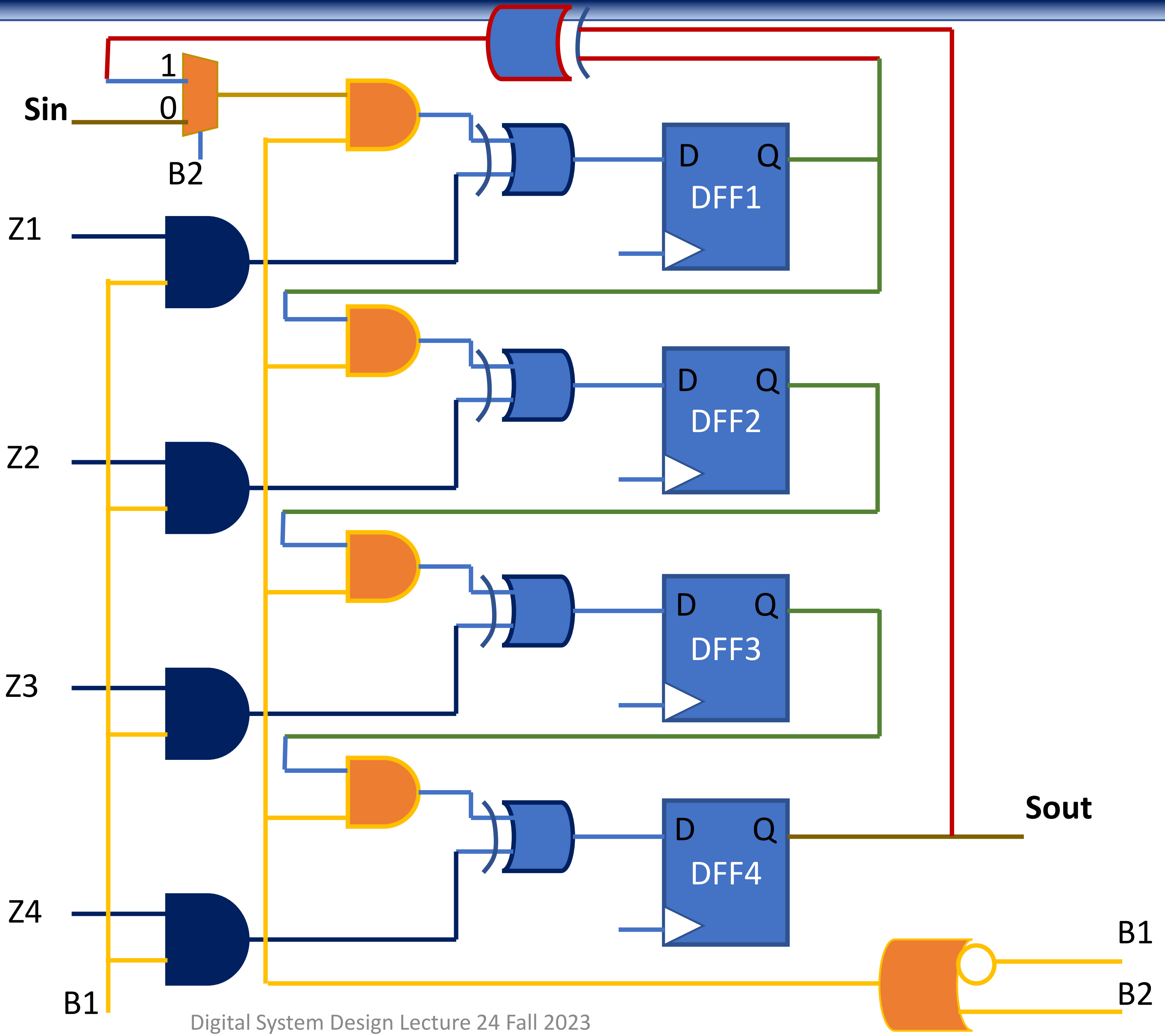
Navy Blue = MISR

Green = Scan Chain Sin to Sout

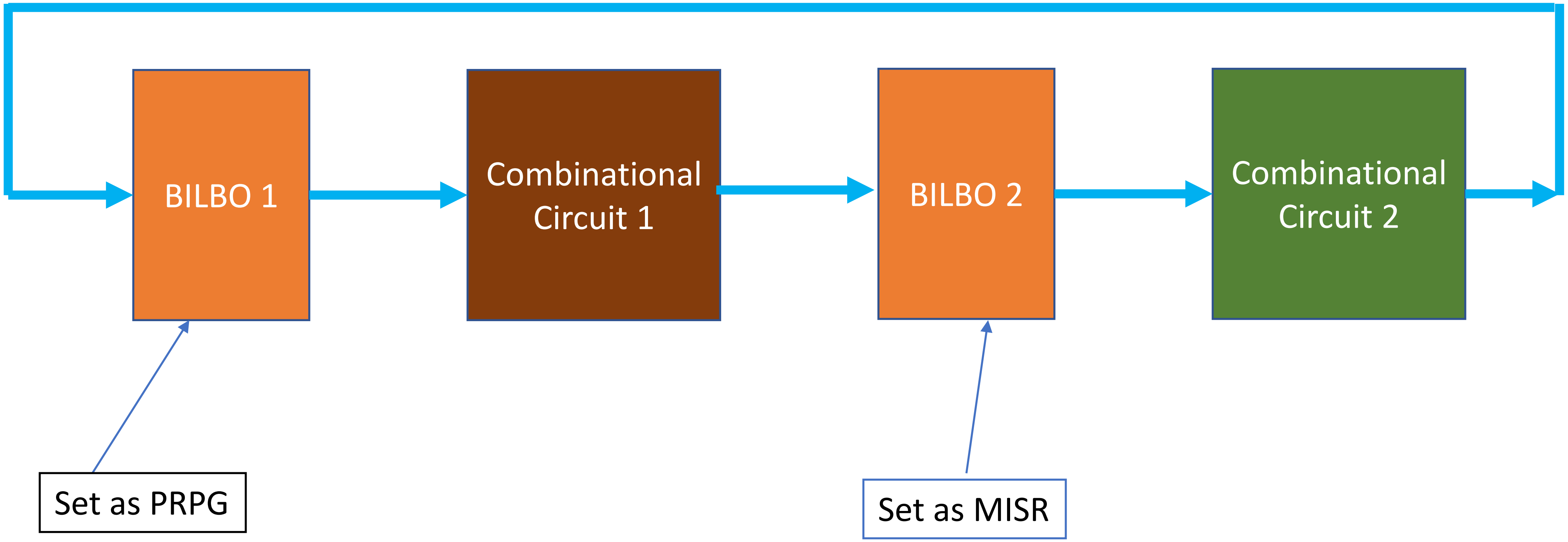
Brown = Select MUX input

Red and Green = SISR

PR PG = Pseudo Random Patr Gen



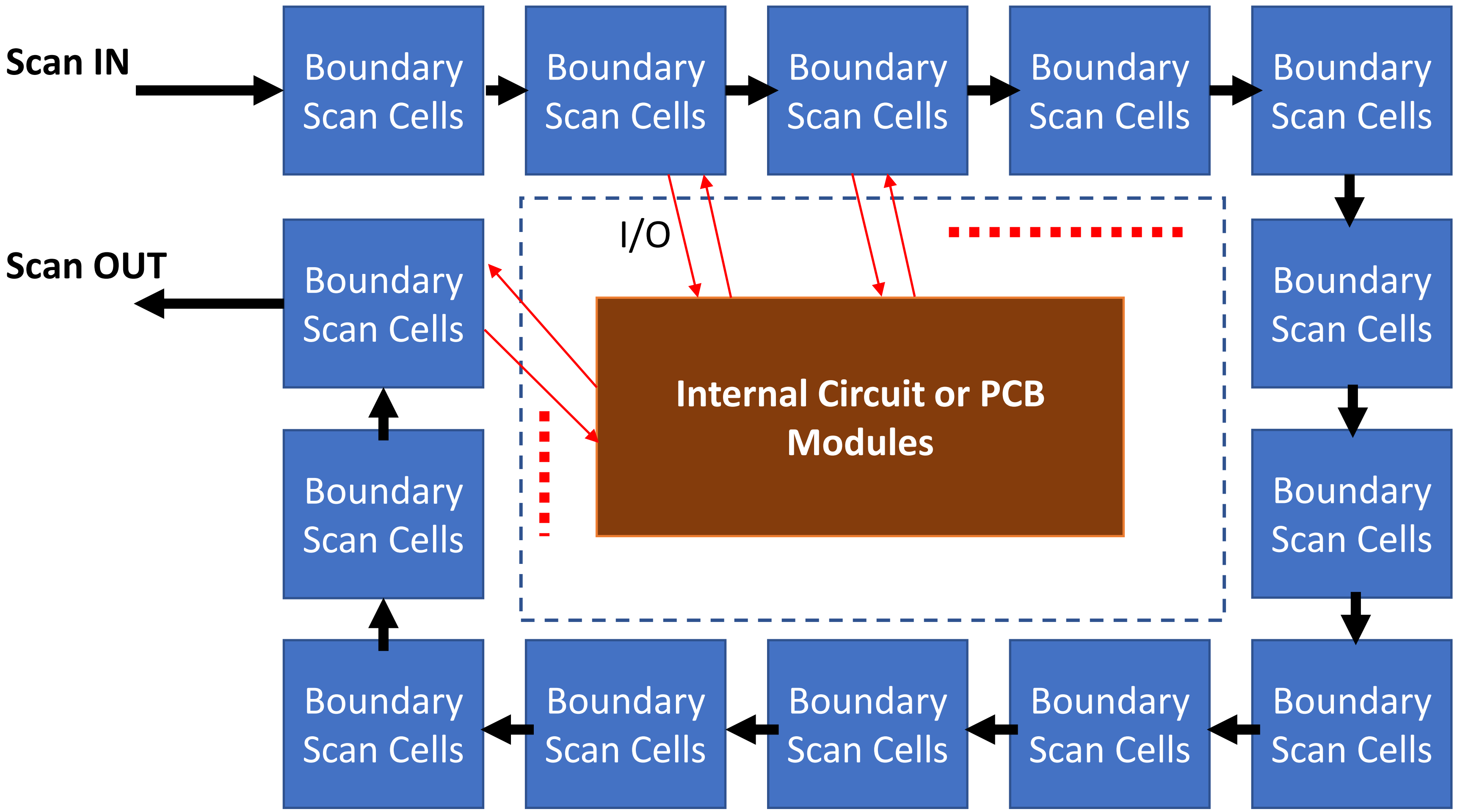
BILBO Application



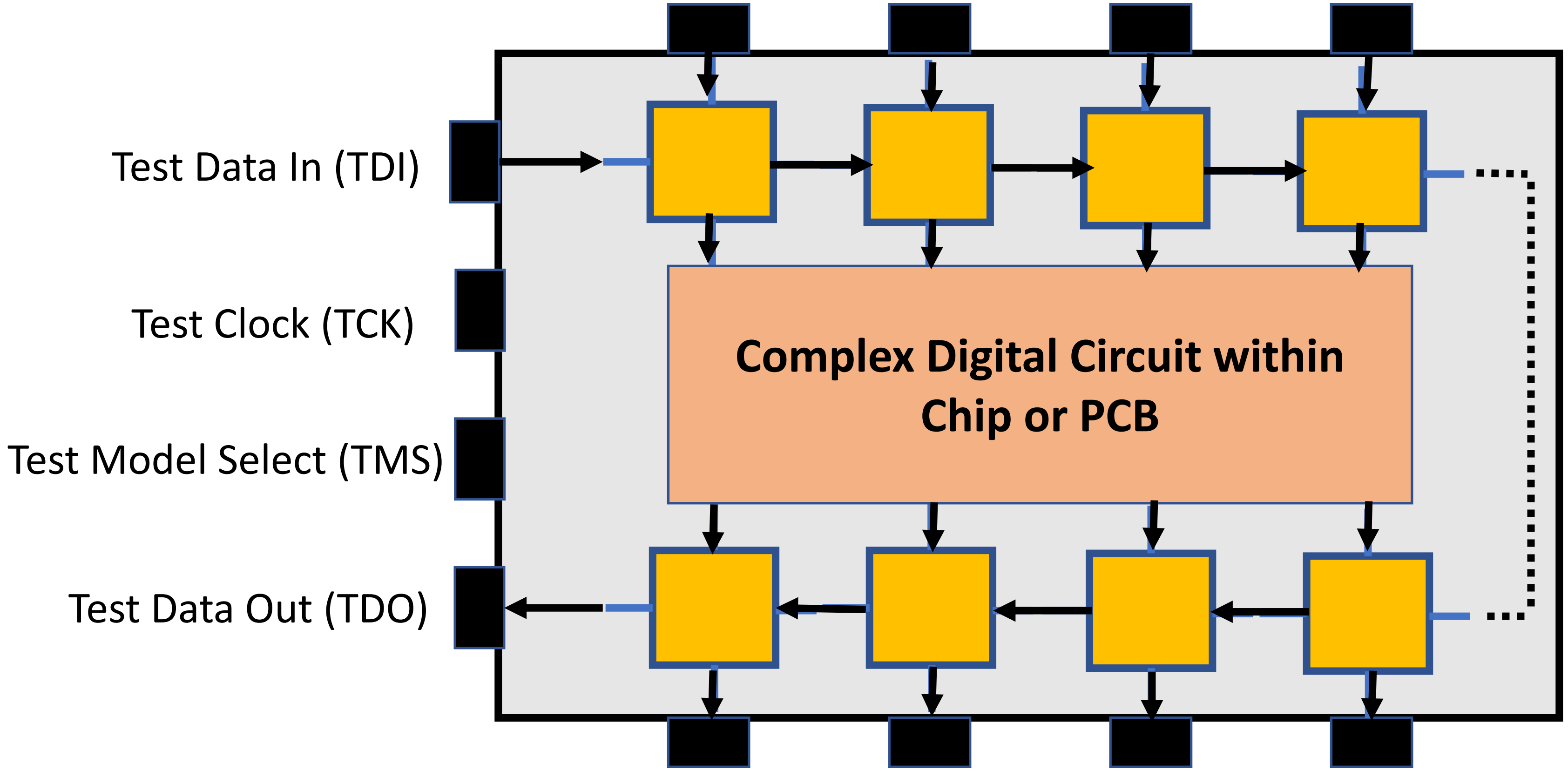
To Test Combinational Circuit 1, BILBO 1 is used as PRPG and BILBO 2 is used as MISR
In the Normal mode, both BILBO serve as registers for associated combinational logic block
To Scan data IN and OUT, both BILBO operate in Shift Register mode

Boundary Scan – Concept

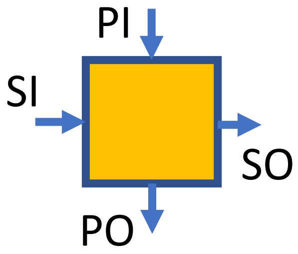
Place Scan cells along the boundary I/O Ports of an ASIC or PCB. They can be connected in the form of a Scan Chain to apply test vectors to internal circuit or PCB modules.



Boundary Scan Implementation



PI = Parallel In
PO = Parallel Out
SI = Serial IN
SO = Serial OUT



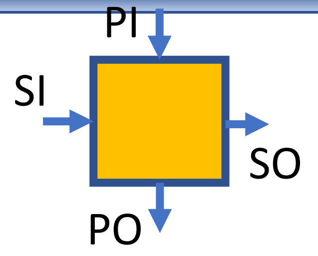
A Boundary Scan Cell

Boundary Scan Cell

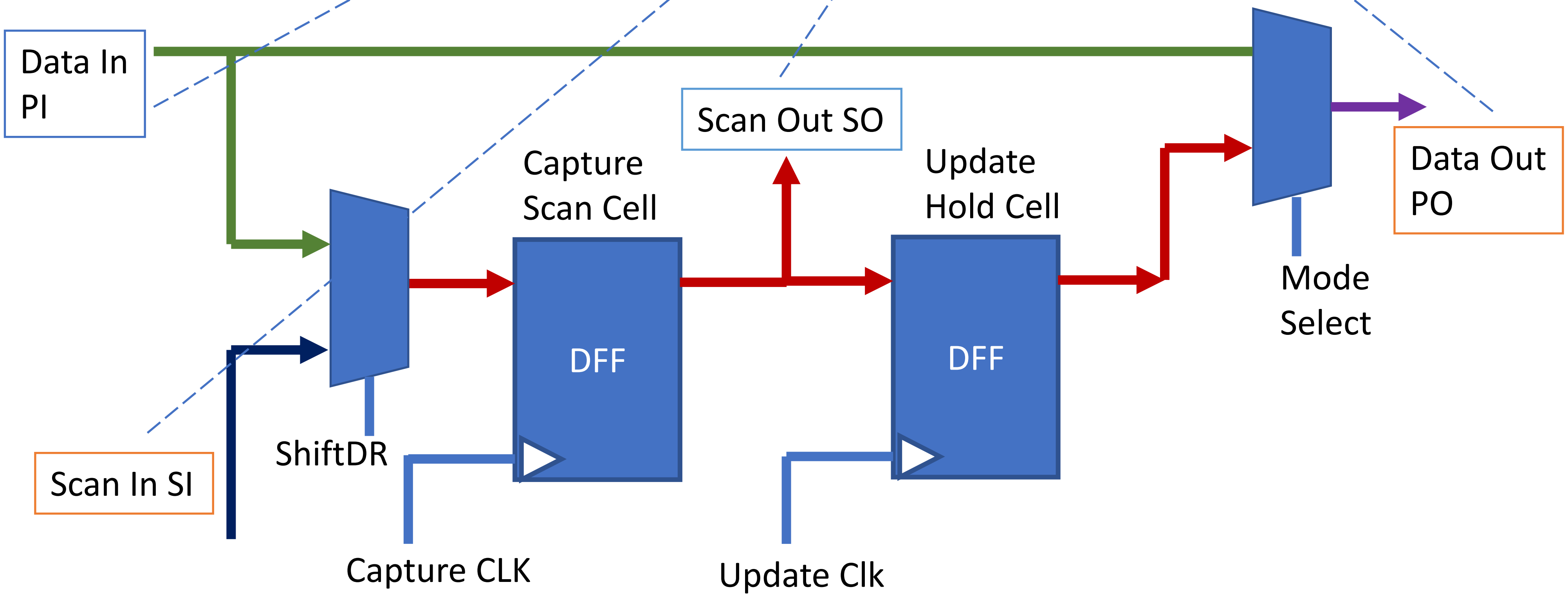
Each Boundary Scan Cell provides the functions:

- Capture data on parallel input PI
- Updates data on parallel output PO
- Serially Scan data from SO to its connected SI
- Transparently passes data from PI to PO

PI = Parallel In
PO = Parallel Out
SI = Serial IN
SO = Serial OUT



A Boundary Scan Cell



Four modes of operations: 1. NORMAL, 2. UPDATE, 3. CAPTURE, 4. SERIAL SHIFT

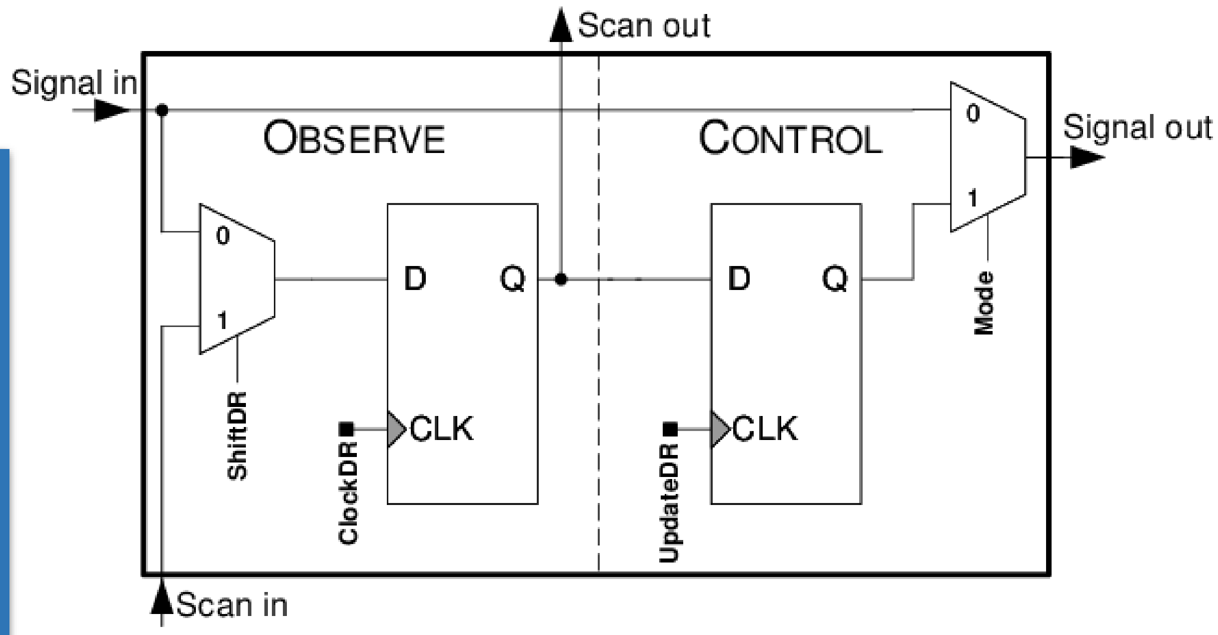
Boundary Scan Cell – Modes of Operation

Normal / Transparent: Data In (PI) is passed Out to Data Out (PO)

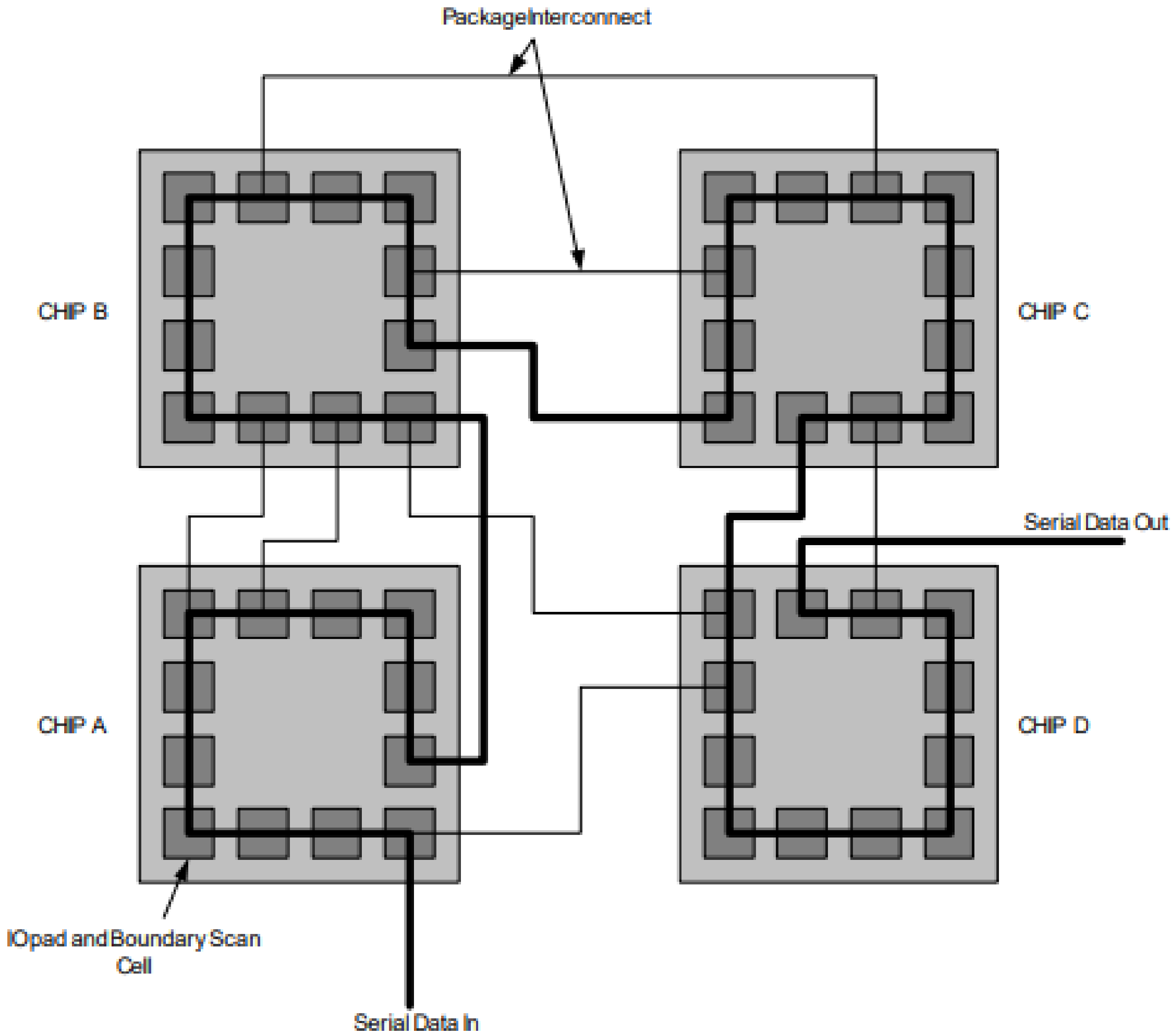
Capture: Test Data In will be captured into the 'Capture Scan Cell' in next Capture Clk

Update: Data stored in 'Capture Scan Cell' is propagated to 'Update Hold Cell' when Update Clk appears

Serial Shift: Test data is shifted from Serial In (SI) and test response can be scanned through Serial Out (SO)



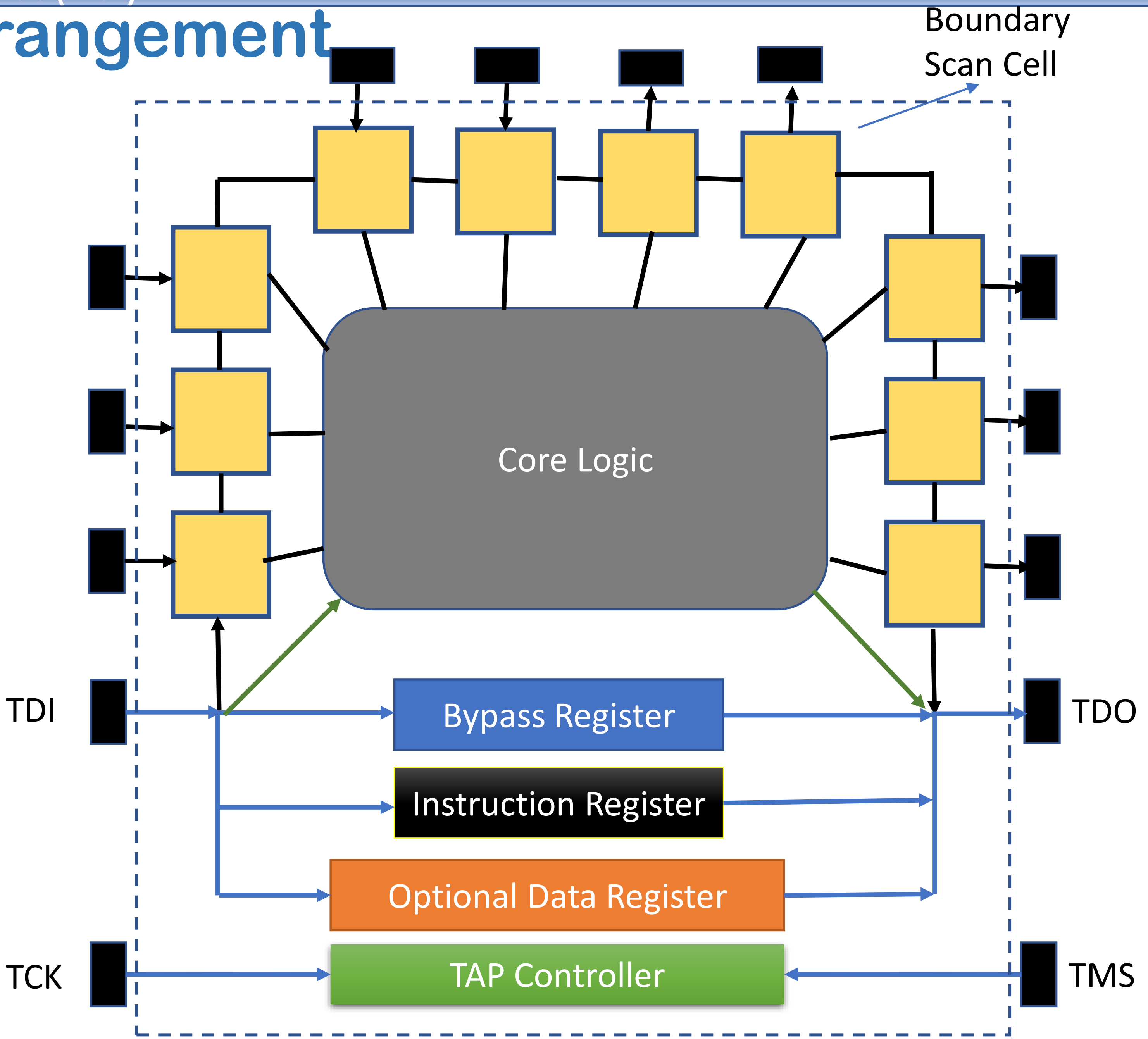
Boundary Scan Example – Testing Multiple Chips



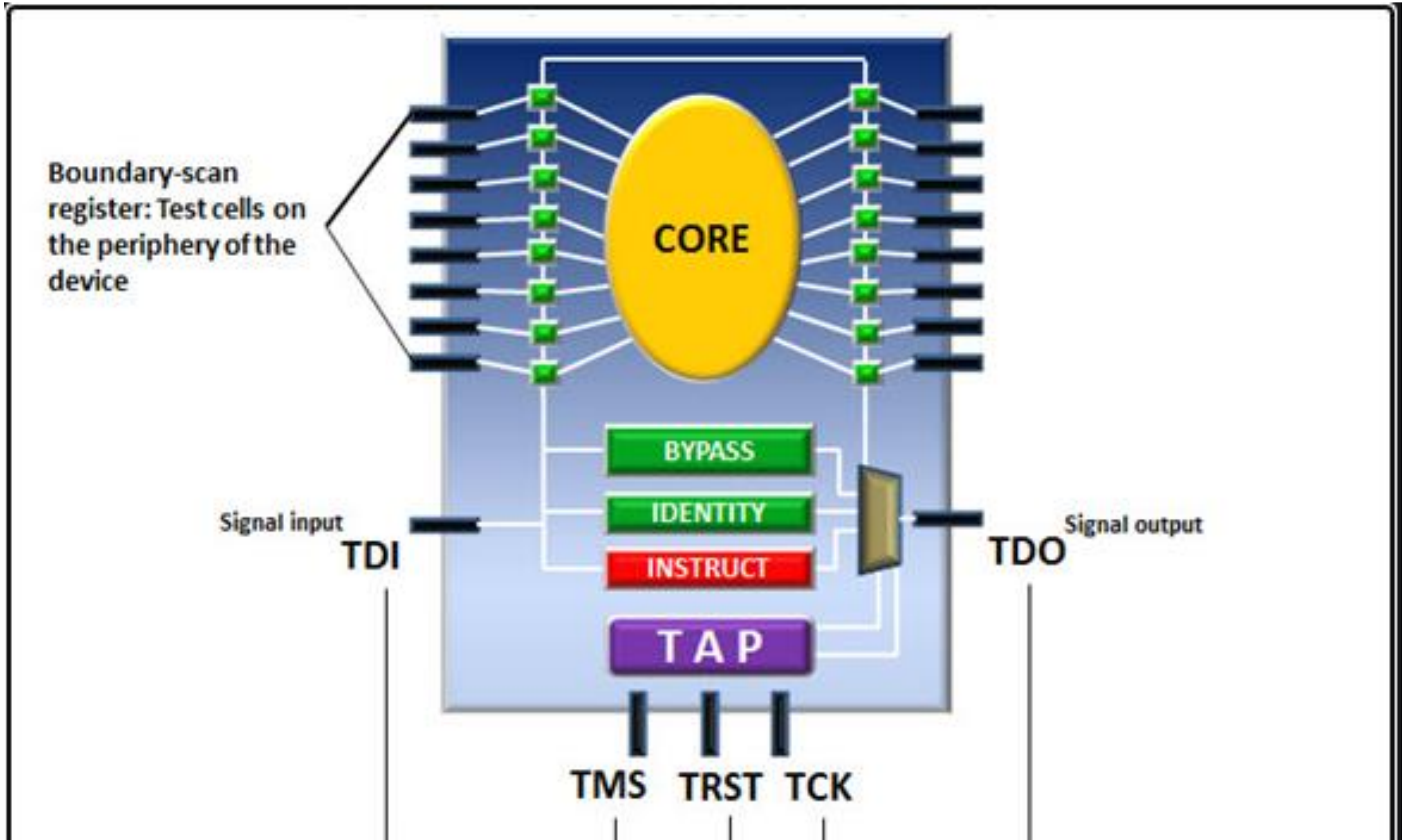
JTAG Standard

- IEEE Joint Test Action Group (JTAG) Standard 1149.1 was finalized in 1990. Latest release was in 2001
- The standard is based on Boundary Scan and Test Access Port architecture
- Allows complete controllability and observability of the boundary pins of a JTAG compatible device under software control
- During test modes, all input signals are captured for analysis and all output signals are preset to test internal devices.
- The operation of these scan cells is software controlled through the Test Access Port (TAP) Controller and the instruction register
- See block diagram next page

JTAG Arrangement



JTAG



Ref: from web

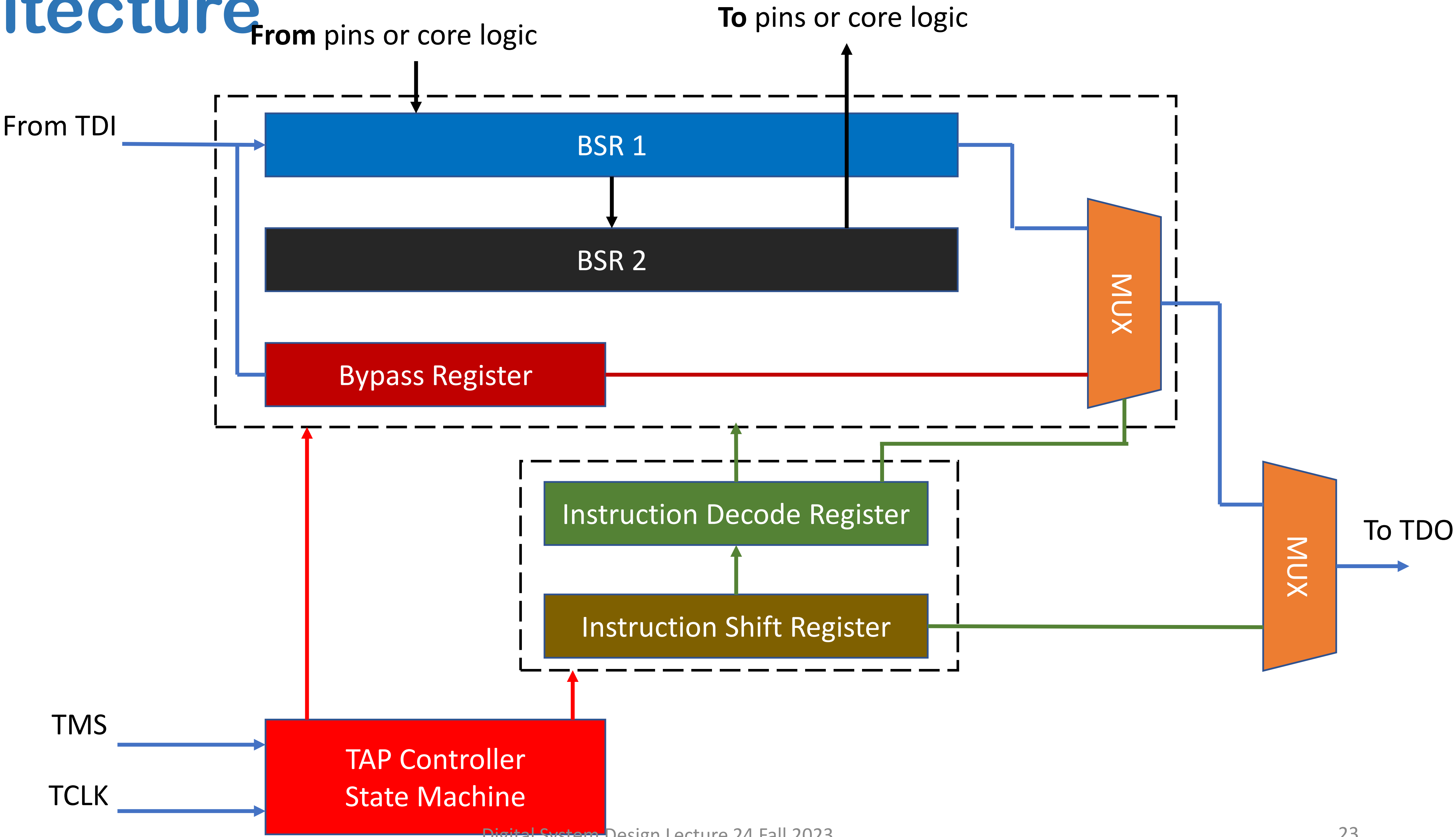
Ref: from web

Test Signals in JTAG TAP – Test Access Port

4 or 5 Pin connections are provided for TAP – Test Access Port

TDI: Test Data Input, this data is sifted serially into Boundary Scan Registers
TCK: Test clock signal
TMS: Test Mode Select
TDO: Test Data Out, this is serial output from Boundary Scan Registers
TRST: Test Reset, resets the TAP controller and test logic, optional pin

Boundary Scan TAP Architecture



Basic JTAG Instructions

BYPASS: This instruction allows TDI serial data to through a 1-bit Bypass register on the IC instead of boundary can registers. In this way, one or more IC on the PCB may be bypassed while other IC are being tested.

SAMPLE/PRELOAD: This instruction is used to scan (read) the boundary scan registers without interfering with the normal operation of core logic. Data is transferred to or from the core logic from or to the IC pins without interference. Test data can be shifted into the BSR.

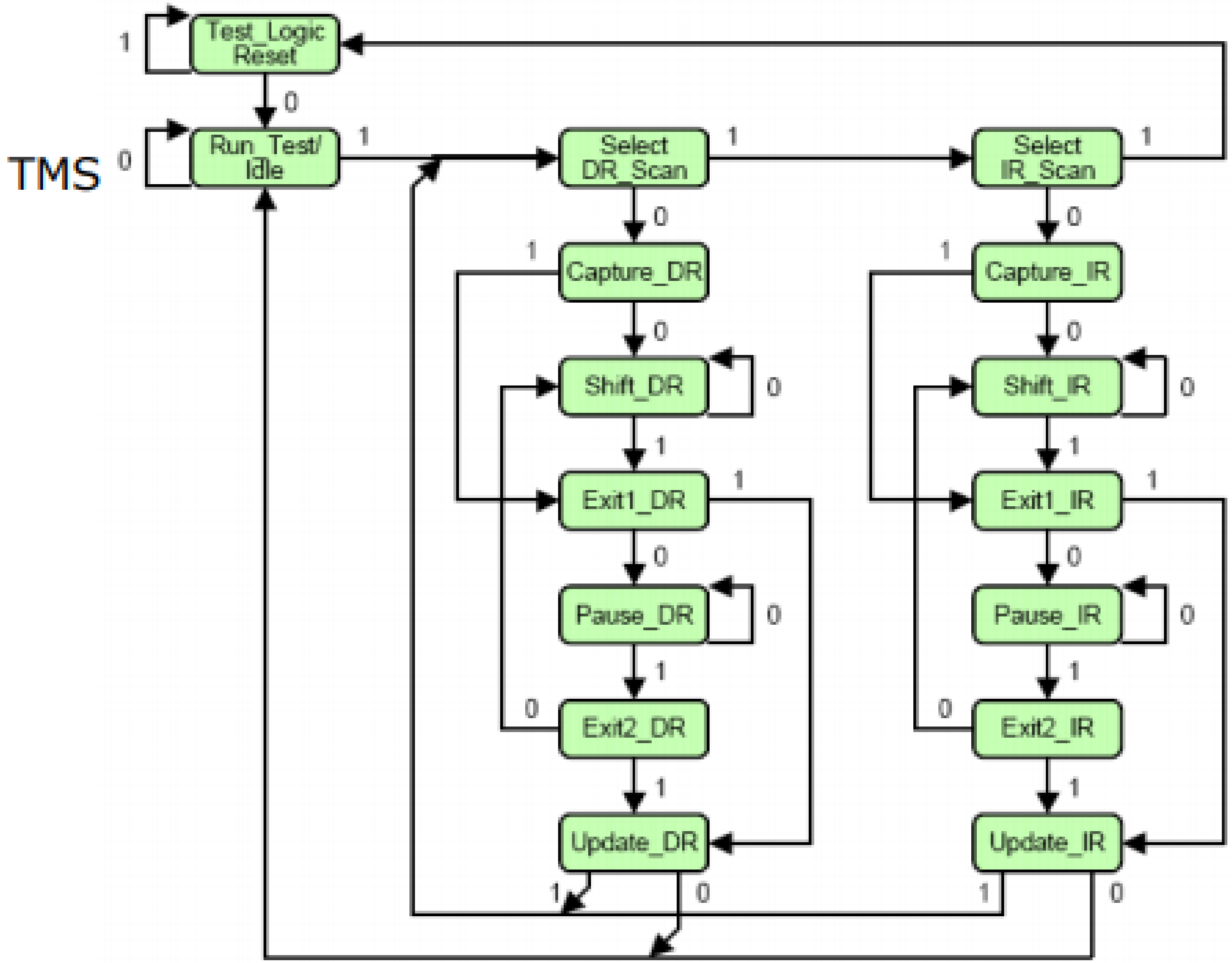
EXTEST: This instruction allows board-level interconnect testing. Test data is shifted into the BSR and then it goes to the output pins. Data from the input pins is captured by the BSR.

INTEST (optional): To test core logic, data shifted into BSR takes place of data from input pins, and output data from core logic is loaded into BSR.

RUNBIST (optional): This instruction causes special BIST logic within the IC to execute. PRPG, SISR and MISR can be invoked.

TAP Controller State Machine – In Standard

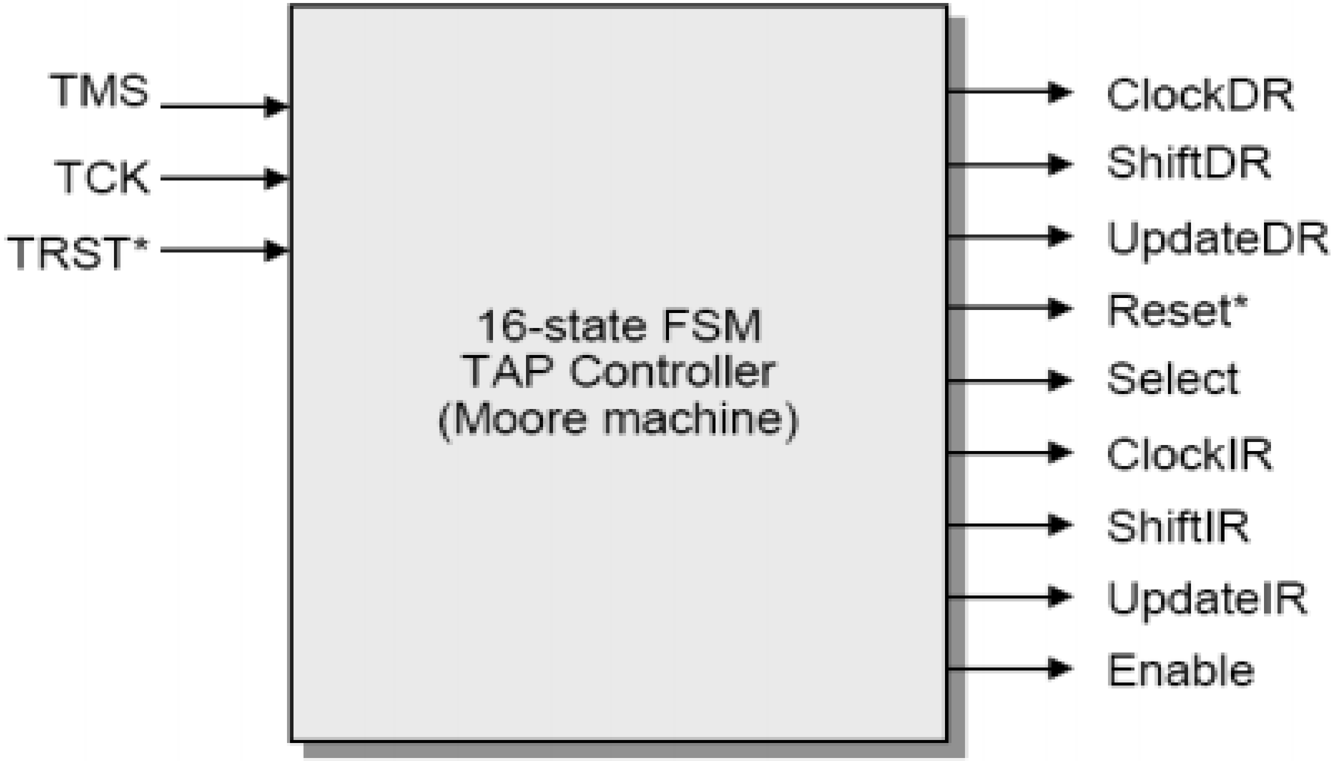
- A state transition occurs on the positive edge of TCK, and the controller output changes on the failing edge of the TCK



Ref:
VLSI Test Technology and Reliability, 2009-2010 CE Lab, TUDelft

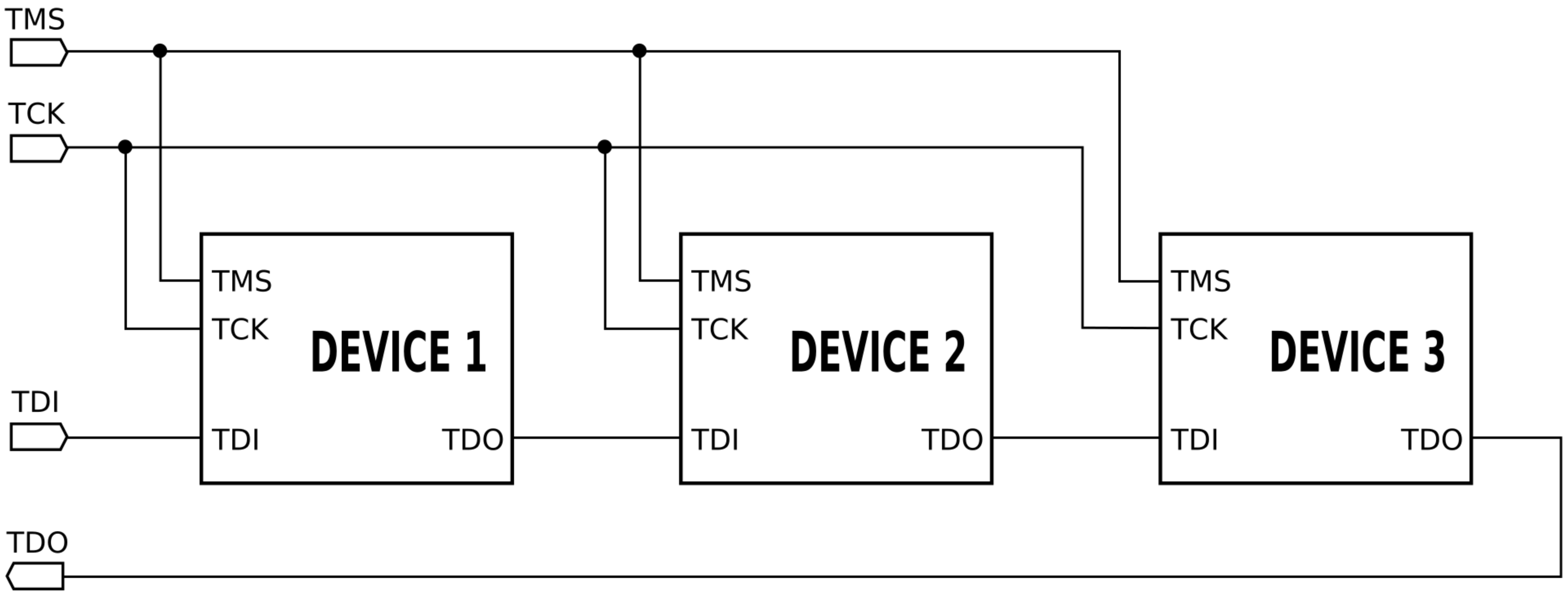
TAP Controller

- 16 finite state machine which produce the various control signals
  - Dedicated signals to IR (ClockIR, ShiftIR, UpdateIR)
  - Dedicated signals to DR (ClockDR, ShiftRD, UpdateDR)
  - Reset*: distributed to IR and targeted DR
  - Select: distributed to output mux
  - Enable: distributed to output drive amplifier

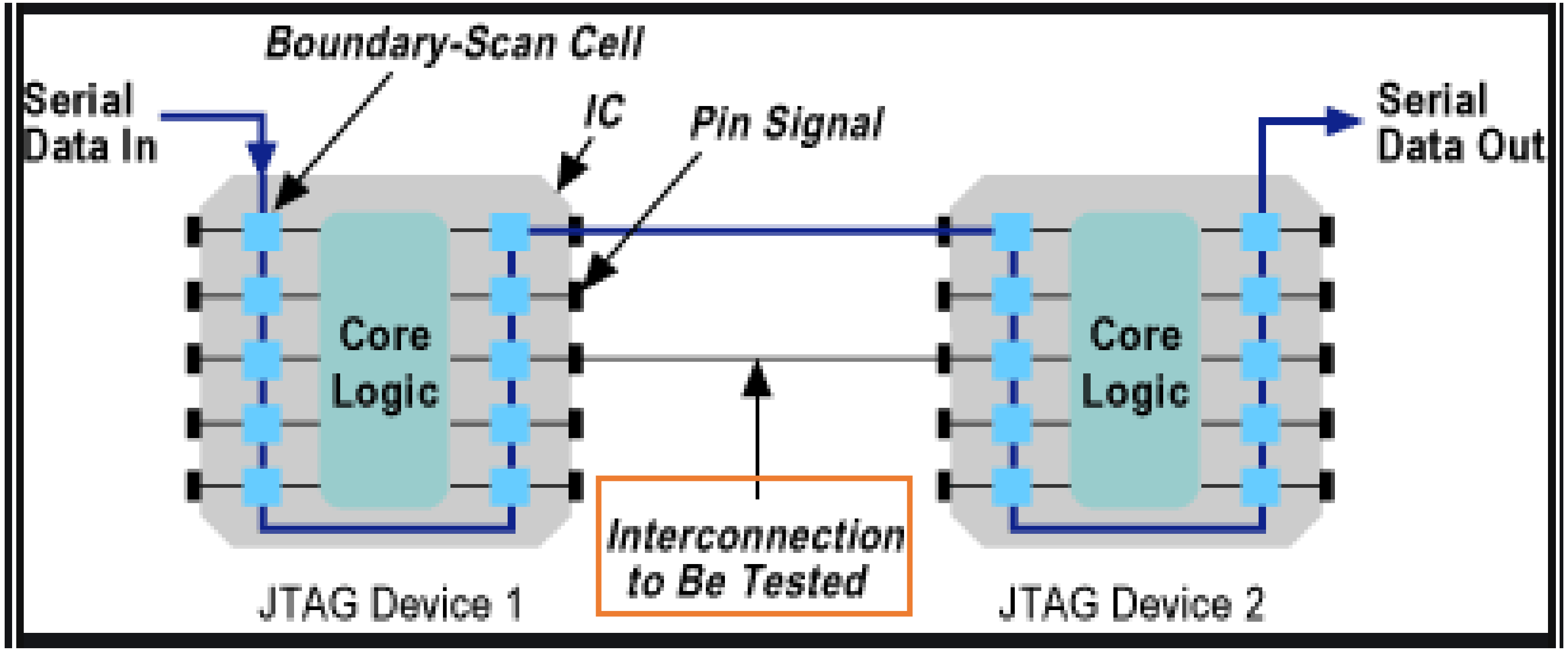


Ref:

Daisy Chain JTAG – for multiple devices

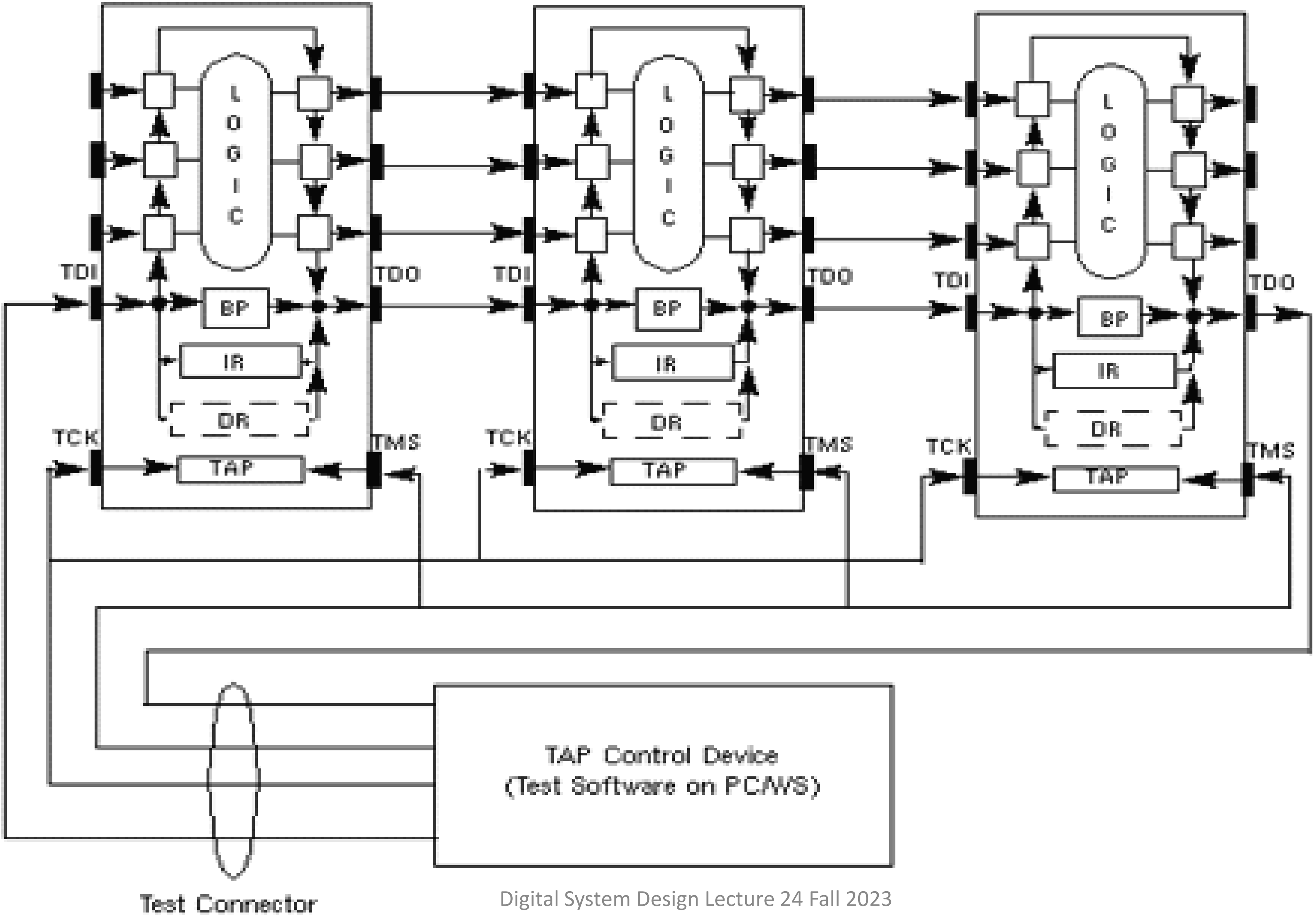


Testing Chip Interconnections



Pre-load the data on BSR of IC1
Transfer through PCB connections to IC2
Read the data from BSR of IC2

TAP Controller for multiple devices



Ref: from web