EEDG/CE 6303: Testing and Testable Design

Mehrdad Nourani

Dept. of ECE
Univ. of Texas at Dallas

Session 09

Scan Design for Testability

Boundary Scan

Board-Level Testing

- Each chip used in a PCB is pretested by the chip's vendor and declared fault-free
- Testing at board level focuses on
  - Inter-chip interconnect faults
    - Opens
      - Caused when chip pins do not bond properly to board
      - That occur in PCB traces due to defects during PCB manufacturing
    - Shorts caused when extra solder flows between pins or PCB traces

Board-Level Testing (cont.)

- Testing at board level focuses on
  - Faults internal to chips
    - Faults induced due to improper handling, e.g., excessive heat or shock during PCB assembly
    - Faulty behaviors that become apparent only when a chip is integrated into a PCB, e.g., the ability to drive a large load
    - These defects
      - More likely to occur in pad drivers and pad receivers
      - However, may also occur in system logic, i.e., on-chip logic circuit
- Since repair possible, diagnosis also important
- Board-level DFT supports
  - Testing and diagnosis of faults in inter-chip interconnects (including pad drivers and receivers)
  - In-situ re-testing of system logic
  - Debugging of system design

Board-Level Testing (cont.)

- Possible approaches for board testing
  - —Consider the entire board as one circuit and generate tests: Not possible because
    - The circuit too large for ATPG tools
    - Net-list of most chips unavailable
  - —In-circuit testing
    - Probes were used to access input/output pins of chips
    - Care had to be taken to ensure probes driving values did not damage output pins
    - No longer used because
      - Pins are too small and too close to be reliably probed
      - Pins at bottoms of chips cannot be probed in multi-layered PCBs
  - —Boundary scan

Boundary Scan

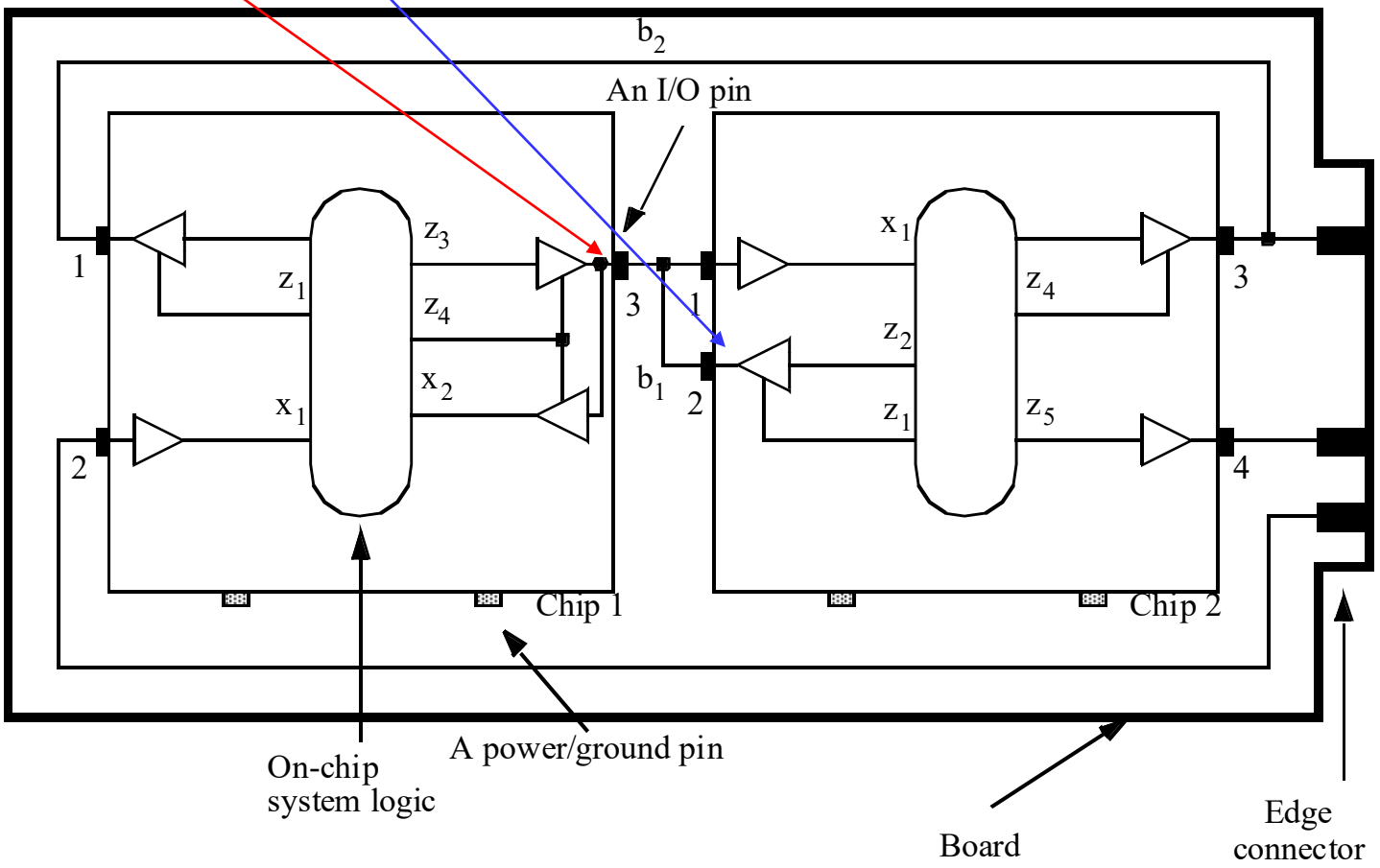
- Boundary scan used to test
  - Chips-on-board (COB) systems, where chips are mounted on a printed circuit board (PCB)
  - Multi-chip modules (MCM), where bare-die are integrated on a silicon or a PCB-like substrate
- A COB system is obtained by
  - Using a PCB that contains traces, i.e., metal connectors that constitute inter-chip interconnects
  - Bonding (e.g., soldering) chips on the PCB

Boundary Scan Structure

- Boundary scan incorporates DFT circuitry that allows direct access to chip input and output pins via scan chains
- A board contains chips from multiple manufacturers
- Boundary scan circuits must interoperate to achieve above objectives
- Hence, a standard, namely IEEE Std 1149.1, defined

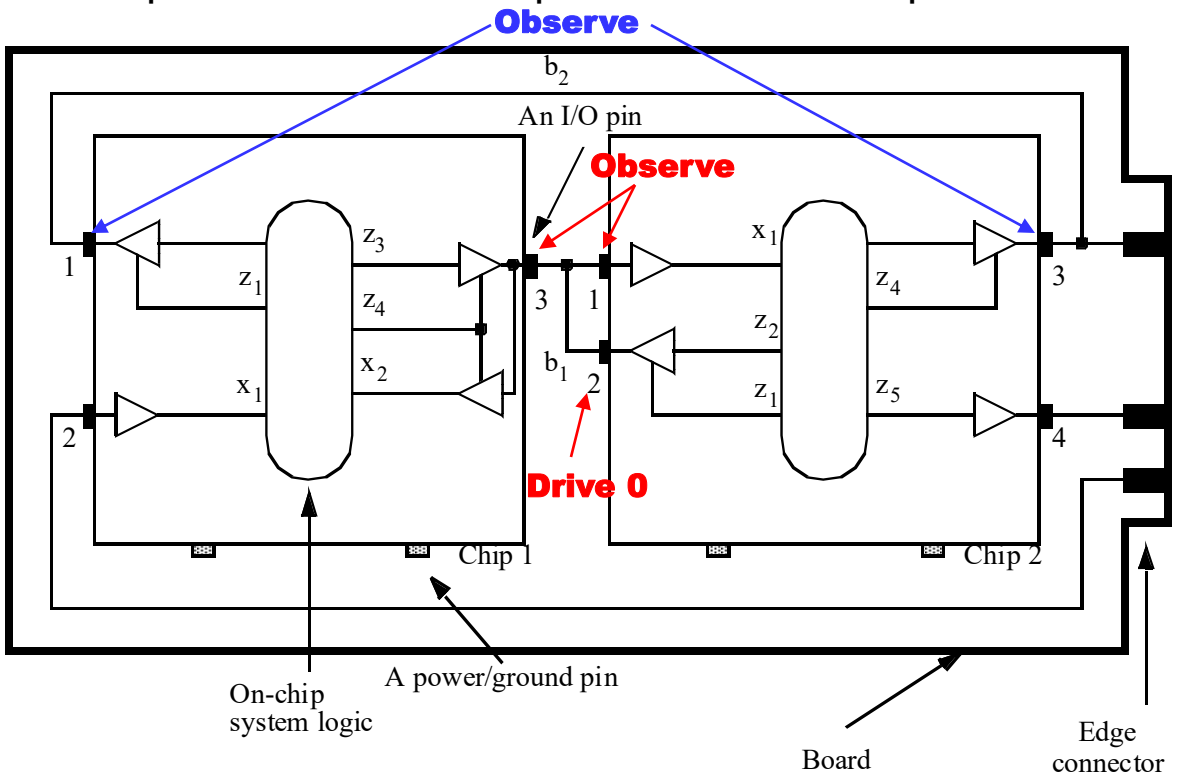
Example: A Board Without Boundary Scan

- Consider an example board. Especially note
  - Bi-directional driver/receiver – Pin 3 of Chip 1
  - Tri-state driver – Pin 2 of Chip 2



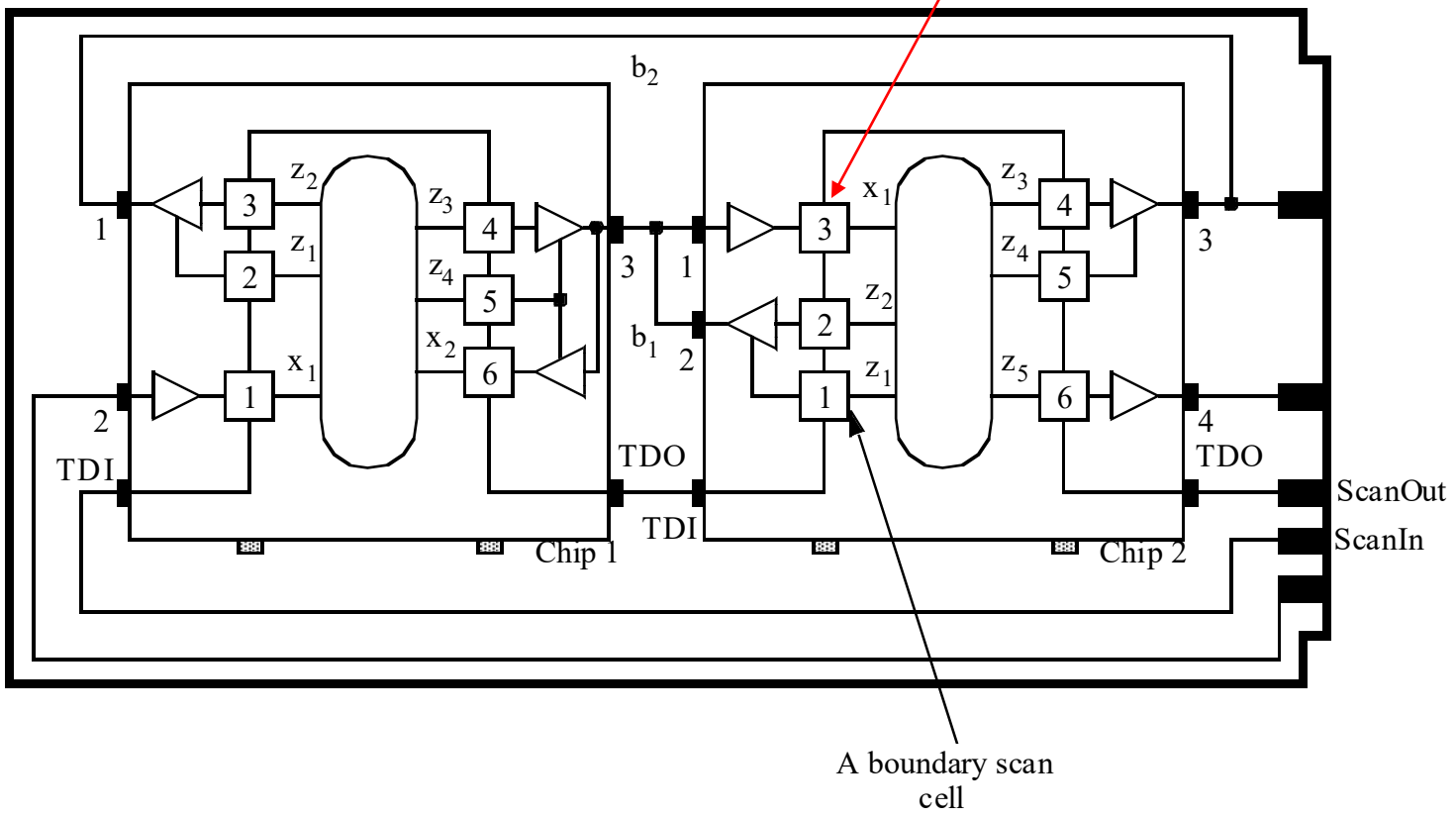
Example: Testing Without Boundary Scan (cont.)

- Assume that
  - A receiver disconnected from its drivers due to opens interprets the input as a value 1
  - A control value 1 enables a tri-state/bi-directional driver
- To test for an open at net b1
  - Use a probe to apply 0 at Pin 2 of Chip 2
  - Use another probe to observe the value at Pin 3 of Chip 1 and Pin 1 of Chip 2
- To test for an short between b1 and b2
  - Probe above pins and Pin 1 of Chip 1 and Pin 3 of Chip 2



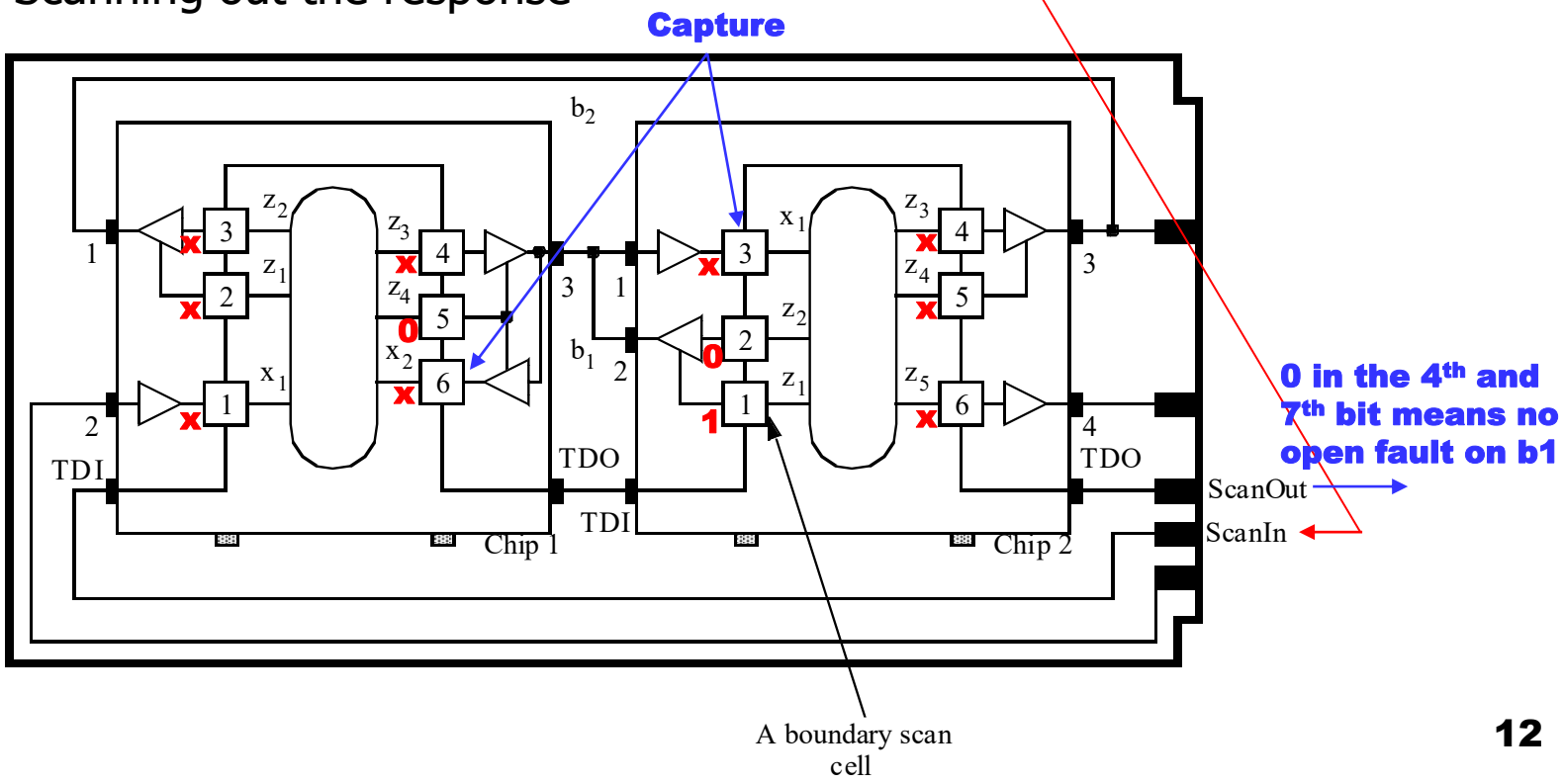
Example: A Board With Boundary Scan

- Boundary scan register (BSR)
  - Formed using boundary scan cells (BSCs)
    - At each input and output pin
    - As well as at control cells of tri-state and bi-directional pins



Example: Testing With Boundary Scan (cont.)

- Probing no longer necessary
  - E.g., an open at the simple receiver of net b1 tested by
    - Scanning in (x, x, x, x, 0, 1, x, 0, x, x, x, x) into BSCs
    - Applying the values scanned in BSCs to corresponding pins
    - Capturing the response from the input pins into corresponding BSCs
    - Scanning out the response



Key Issues & History

Key Issues on Boundary Scan

- Basic Concept & Structure
- Digital Boundary Scan (1149.1)
- Boundary Scan for Advanced Networks (1149.6)
- Embedded Core Test Standard (1500)
- Comparison between 1149.1 and 1500

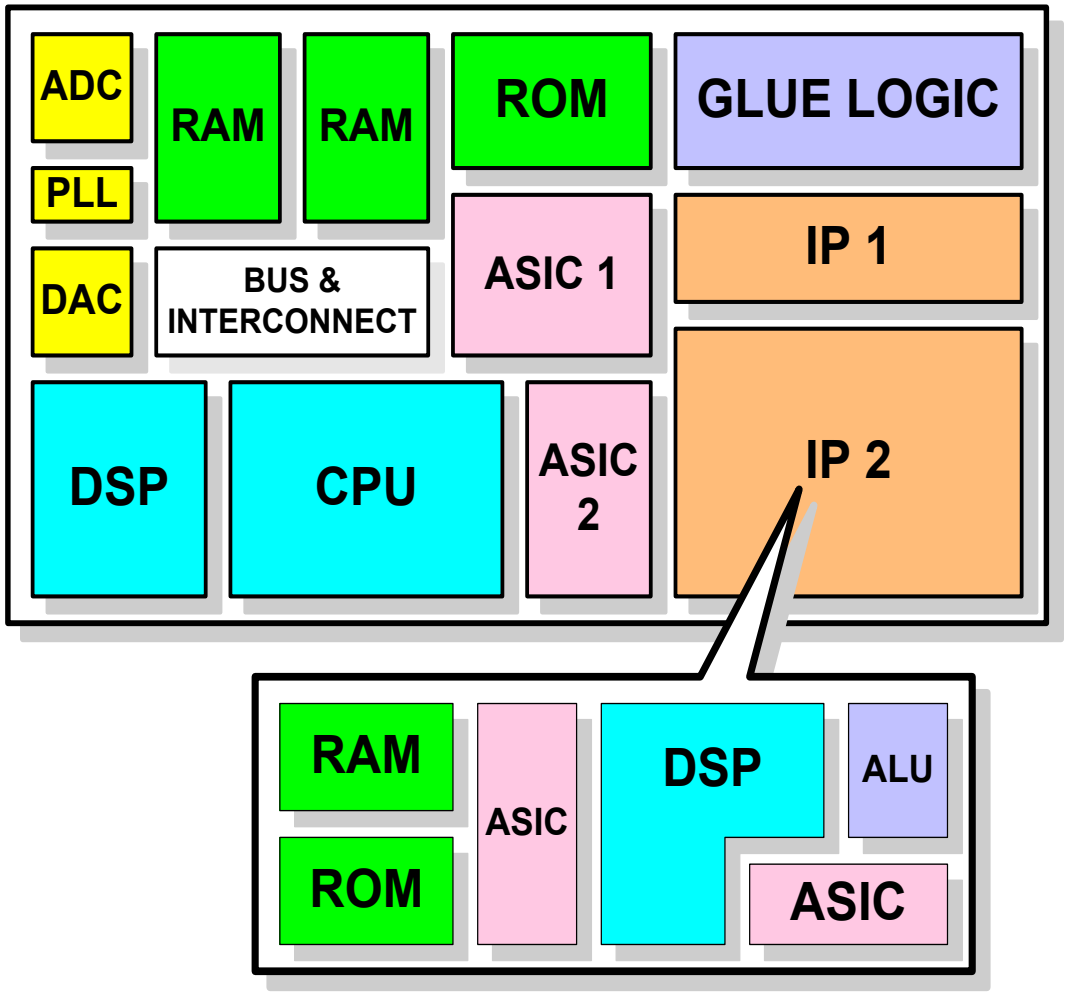
Boundary Scan

- Original objective: board-level digital testing
- Now also apply to:
  - MCM and FPGA
  - Analog circuits and high-speed networks
  - Verification, debugging, clock control, power management, chip reconfiguration, etc.
- History:
  - Mid-1980: JETAG
  - 1988: JTAG
  - 1990: First boundary scan standard – 1149.1

Boundary Scan Family

No.	Main target	Status
1149.1	Digital chips and interconnects among chips	Std. 1149.1-2001
1149.2	Extended digital serial interface	Discontinue
1149.3	Direct access testability interface	Discontinue
1149.4	Mixed-signal test bus	Std. 1149.4-1999
1149.5	Standard module test and maintenance (MTM) bus	Std. 1149.5-1995 (not endorsed by IEEE since 2003)
1149.6	High-speed network interface	Std. 1149.6-2003

Core-Based SOC Design

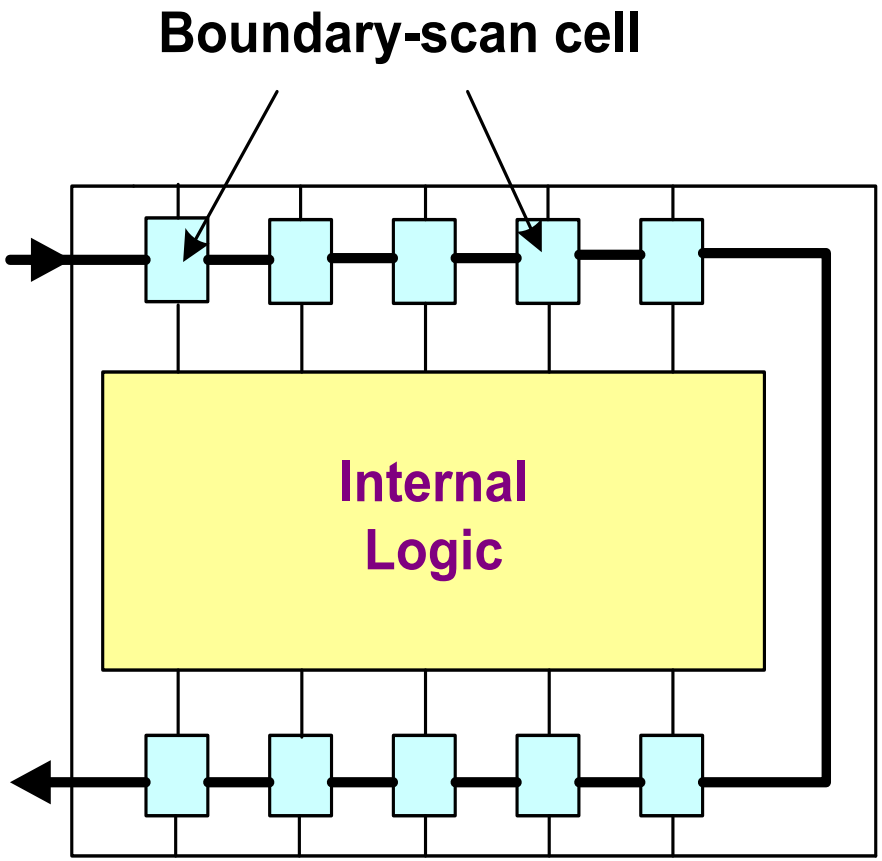


Digital Boundary Scan – IEEE 1149.1

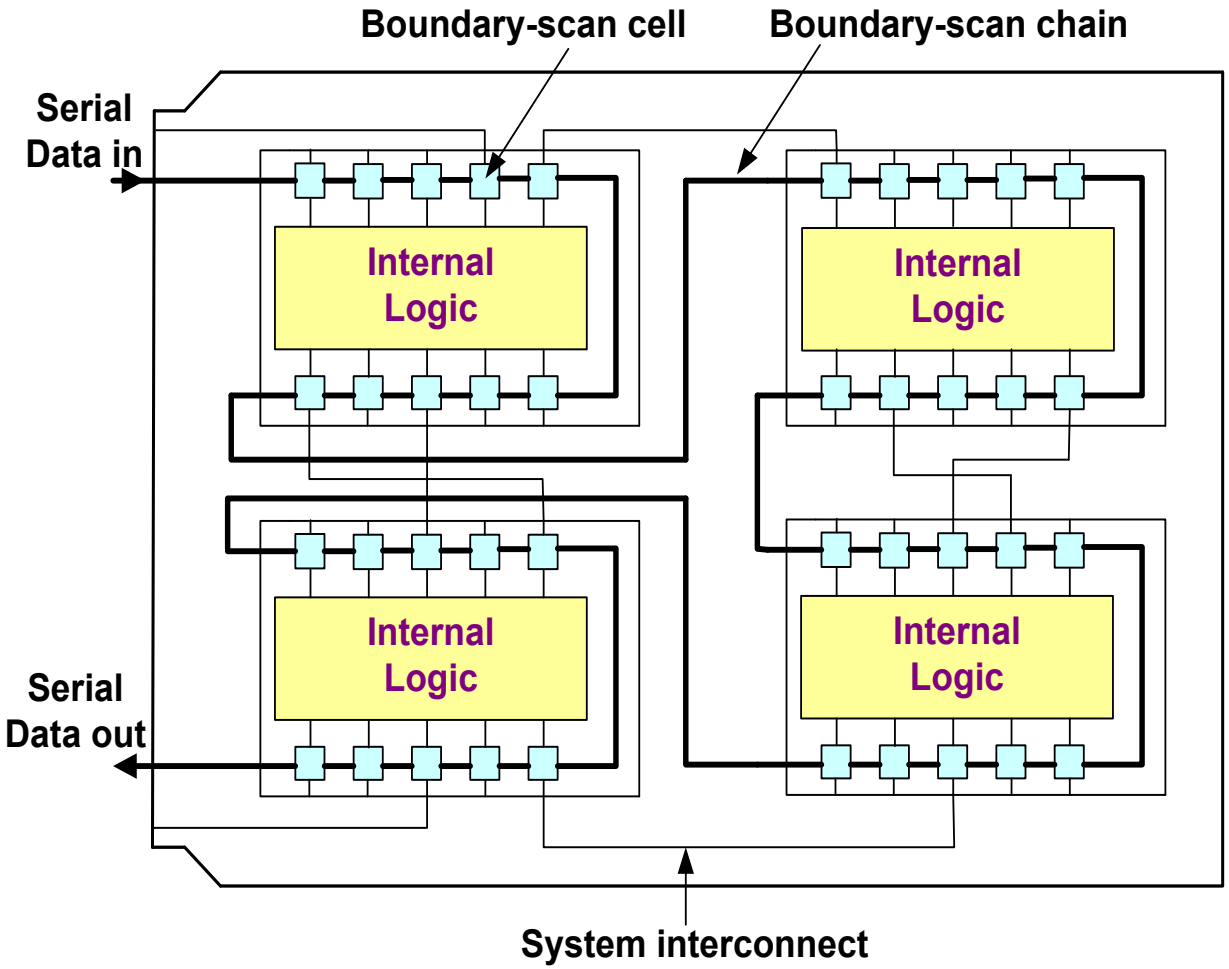
Digital Boundary Scan – 1149.1

- Basic concepts
- Overall test architecture & operations
- Hardware components
- Instruction register & instruction set
- Boundary scan description language
- On-chip test support
- Board/system-level control architectures

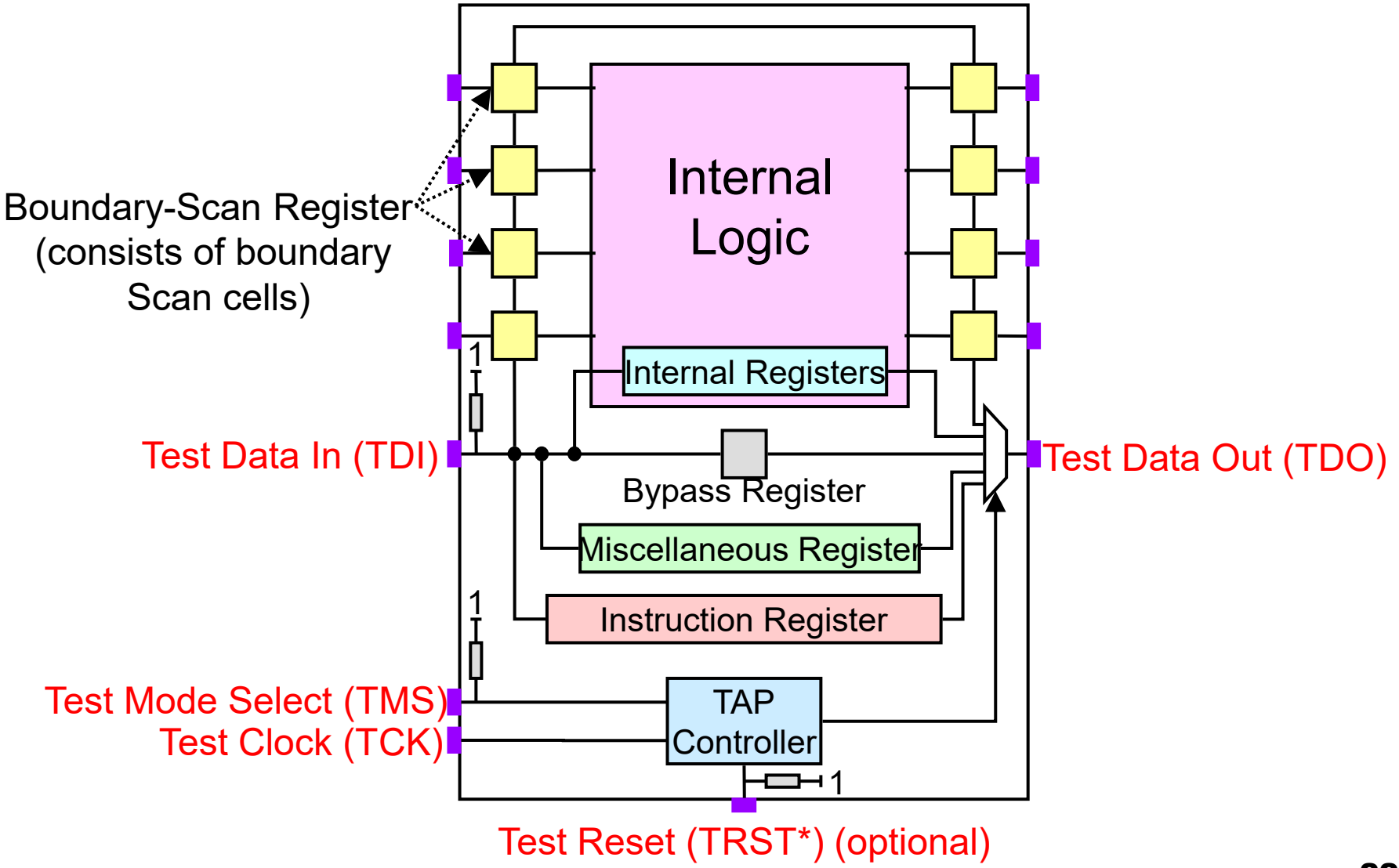
Basic Idea of Boundary Scan



A Board Containing 4 IC's with Boundary Scan



1149.1 Boundary-Scan Architecture



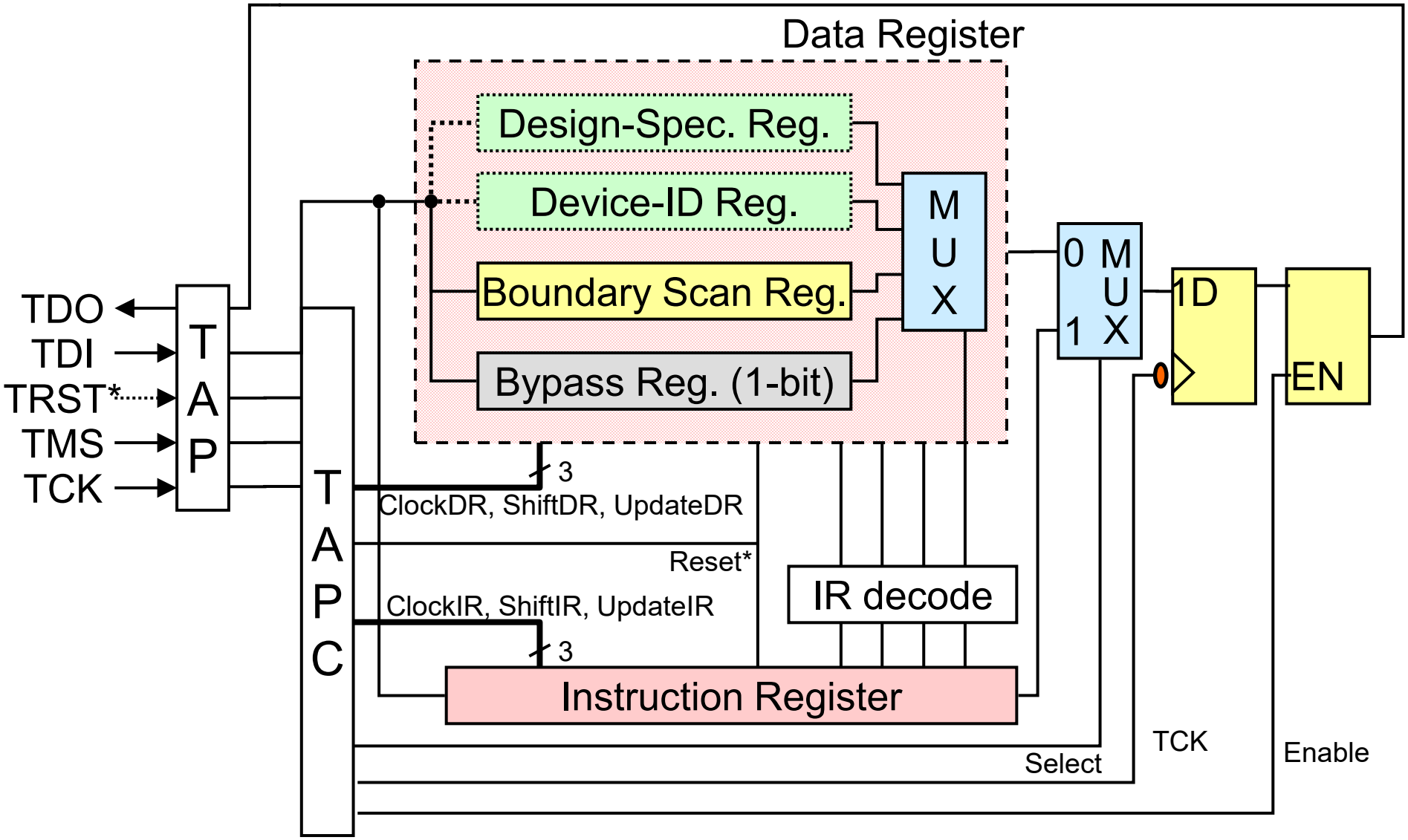
Hardware Components of 1149.1

- A test access port (TAP) consisting of :
  - —4 mandatory pins: Test data in (TDI), Test data out (TDO), Test mode select (TMS), Test clock (TCK), and
  - —1 optional pin: Test reset (TRST*)
- A test access port controller (TAPC)
- An instruction register (IR)
- Several test data registers
  - —A boundary scan register (BSR) consisting of boundary scan cells (BSCs)
  - —A bypass register (BR)
  - —Some optional registers (Device-ID register, design-specified registers such as scan registers, LFSRs for BIST, etc.)

Basic Operations

1. Instruction sent (serially) through TDI into instruction register.
2. Selected test circuitry configured to respond to the instruction.
3. Test pattern shifted into selected data register and applied to logic to be tested
4. Test response captured into some data register
5. Captured response shifted out; new test pattern shifted in simultaneously
6. Steps 3-5 repeated until all test patterns are applied.

Boundary-Scan Circuitry in a Chip



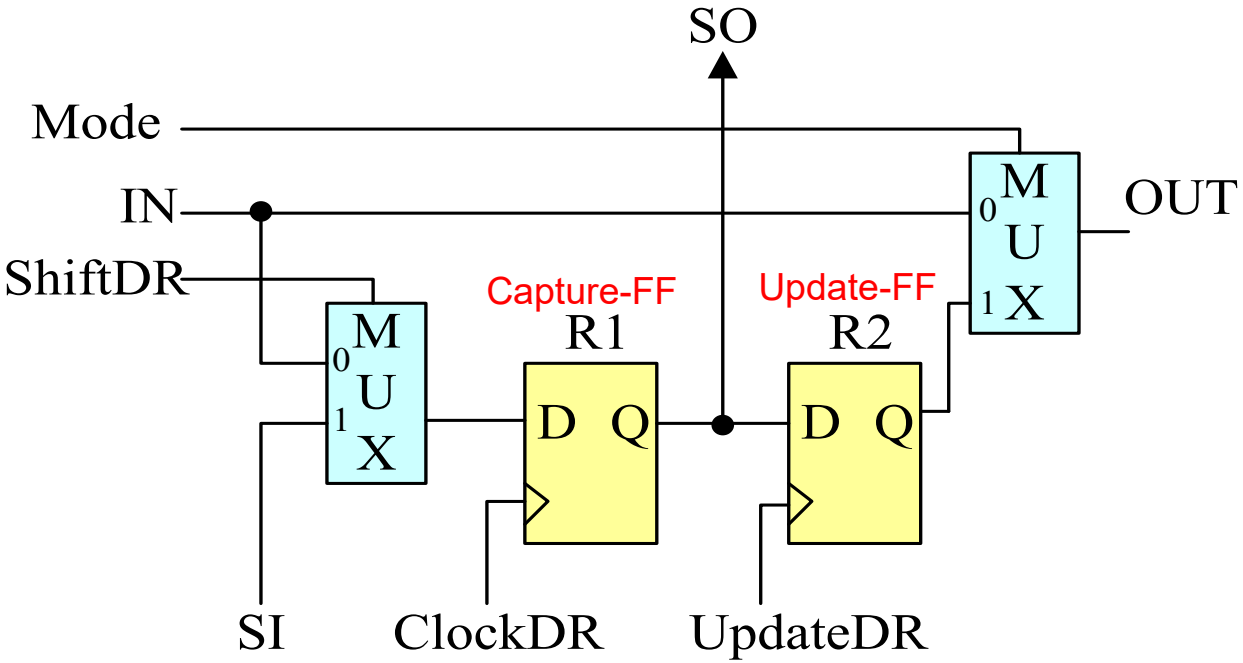
Data Registers

- **Boundary scan register**: consists of boundary scan cells
- **Bypass register**: a one-bit register used to pass test signal from a chip when it is not involved in current test operation
- **Device-ID register**: for the loading of product information (manufacturer, part number, version number, etc.)

MSB 31 28	27 12	11 1	LSB 0
Version (4 bits)	Part Number (16 bits)	Manufacturer Identity (11 bits)	'1' (1 bit)

- **Other user-specified data registers** (scan chains, LFSR for BIST, etc.)

A Typical Boundary-Scan Cell (BSC)

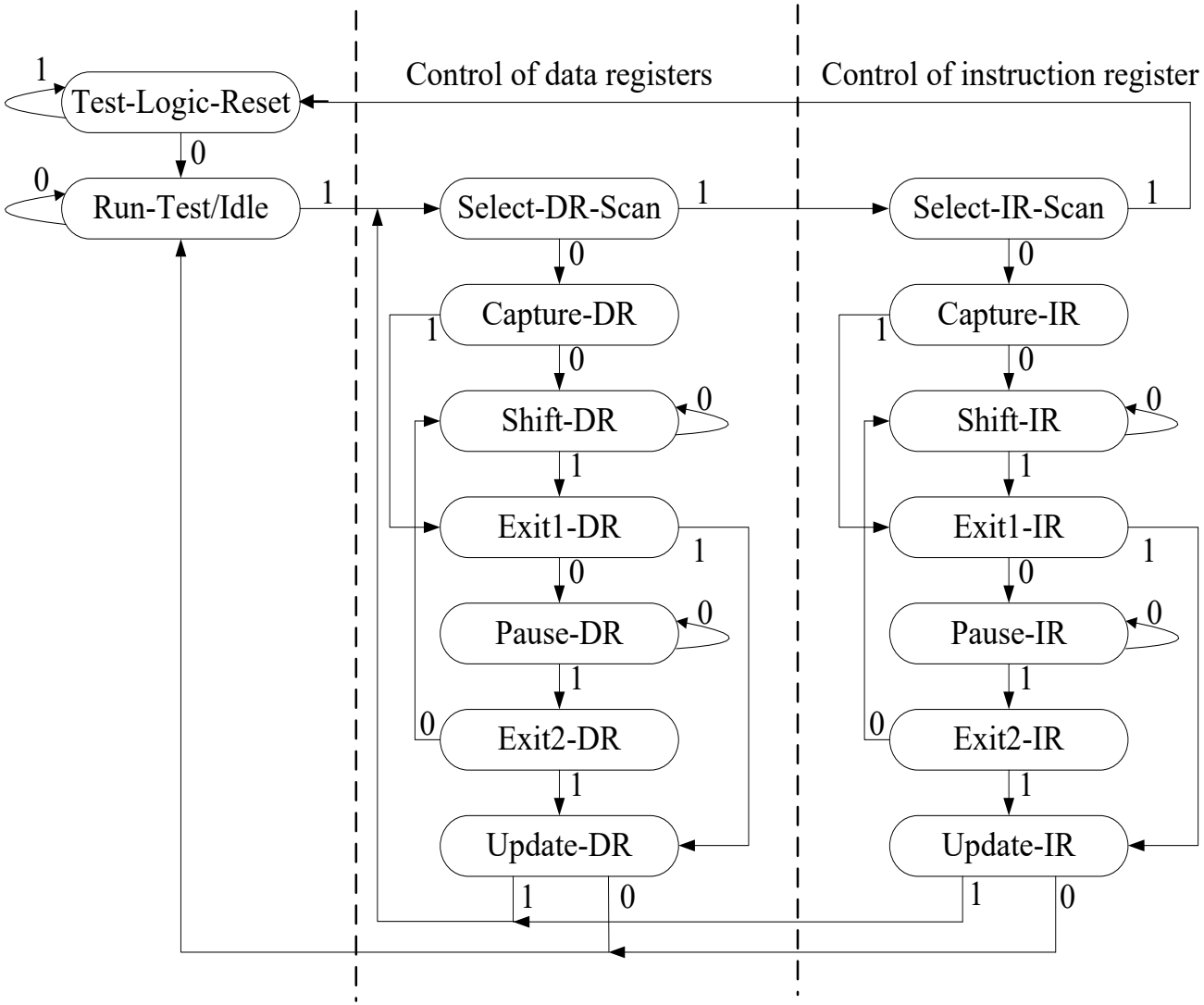


- Four main operation modes
  1. Normal: IN → OUT (Mode = 0)
  2. Shift: TDI → ... → SI → SO → ... → TDO (ShiftDR = 1, ClockDR)
  3. Capture: IN → R1, OUT driven by IN or R2 (ShiftDR = 0, ClockDR)
  4. Update: R1 → R2 → OUT (Mode = 1, UpdateDR)

TAP Controller

- Interprets the control inputs applied serially at TMS input (sampled at each rising edge of TCK)
- A finite state machine with 16 states
- Input: TCK, TMS
- Output: 9 or 10 signals , i.e. ClockDR, UpdateDR, ShiftDR, ClockIR, UpdateIR, ShiftIR, Select, Enable, TCK and TRST* (optional).

State Diagram of TAP Controller

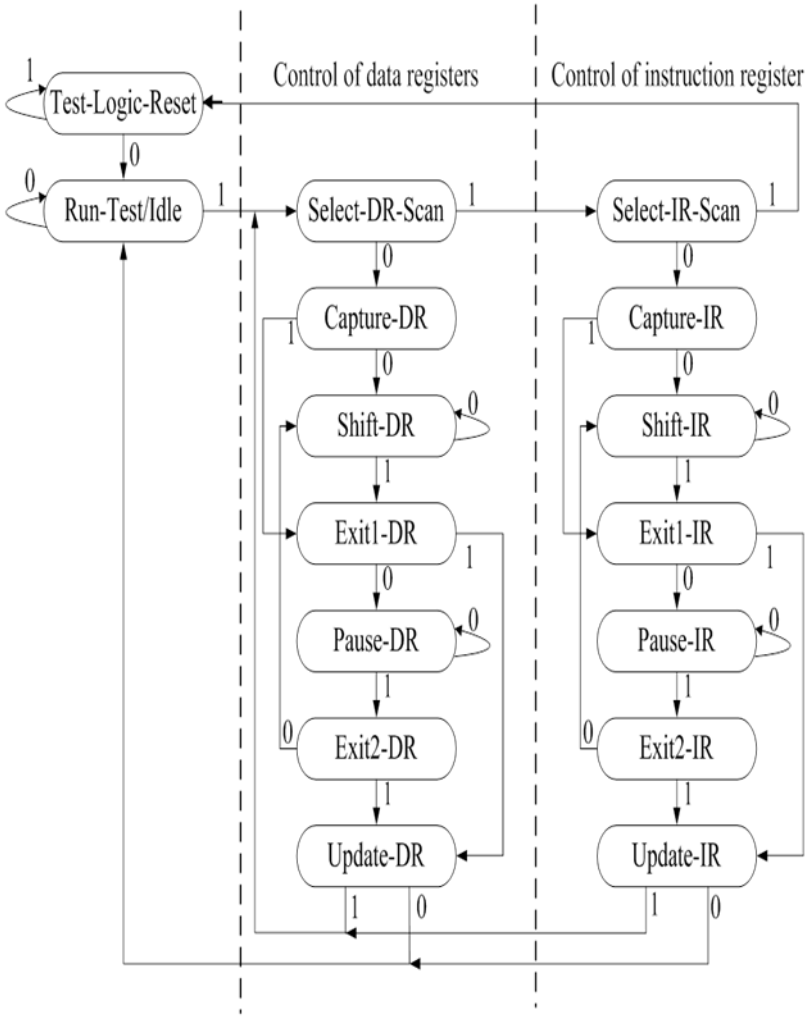


Main Functions of TAP Controller

- Providing control signals to
  - —Reset BS circuitry
  - —Load instructions into instruction register
  - —Perform test capture operation
  - —Perform test update operation
  - —Shift test data in and out

States of TAP Controller

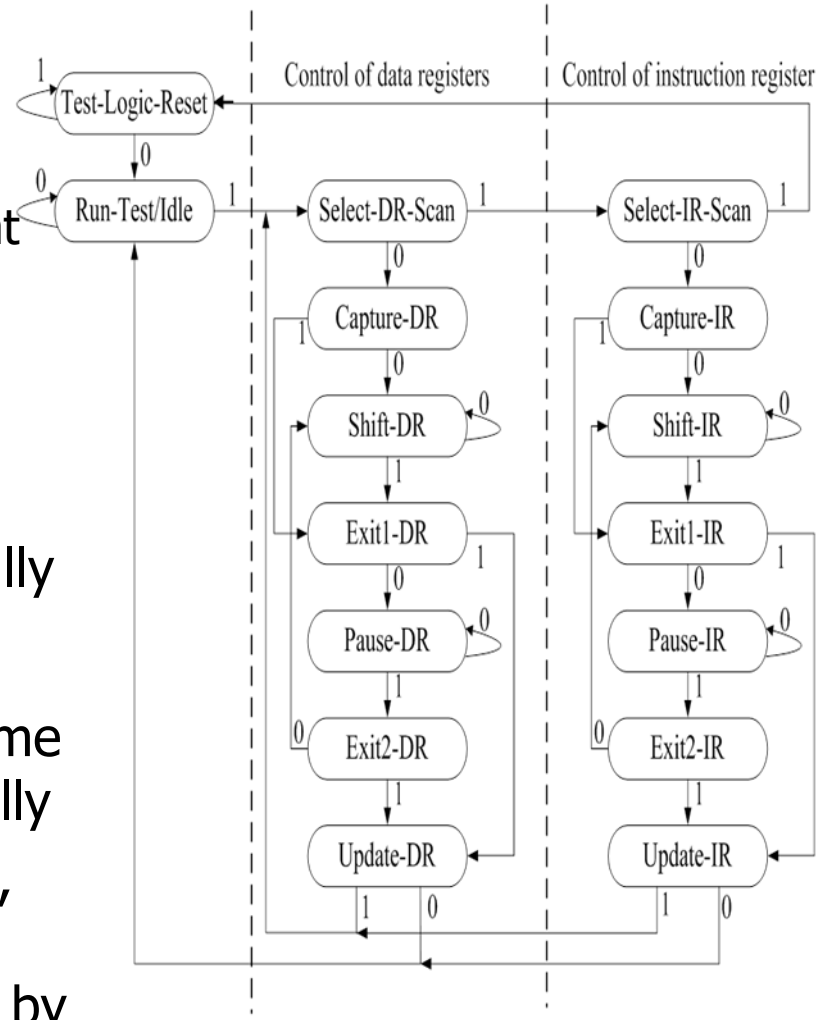
- Test-Logic-Reset: normal mode
- Run-Test/Idle: wait for internal test such as BIST
- Select-DR-Scan: initiate a data-scan sequence
- Capture-DR: load test data in parallel
- Shift-DR: load test data in series
- Exit1-DR: finish phase-1 shifting of data
- Pause-DR: temporarily hold the scan operation (e.g., allow the bus master to reload data)
- Exit2-DR: finish phase-2 shifting of data
- Update-DR: parallel load from associated shift registers



Note: Controls for IR are similar to those for DR.

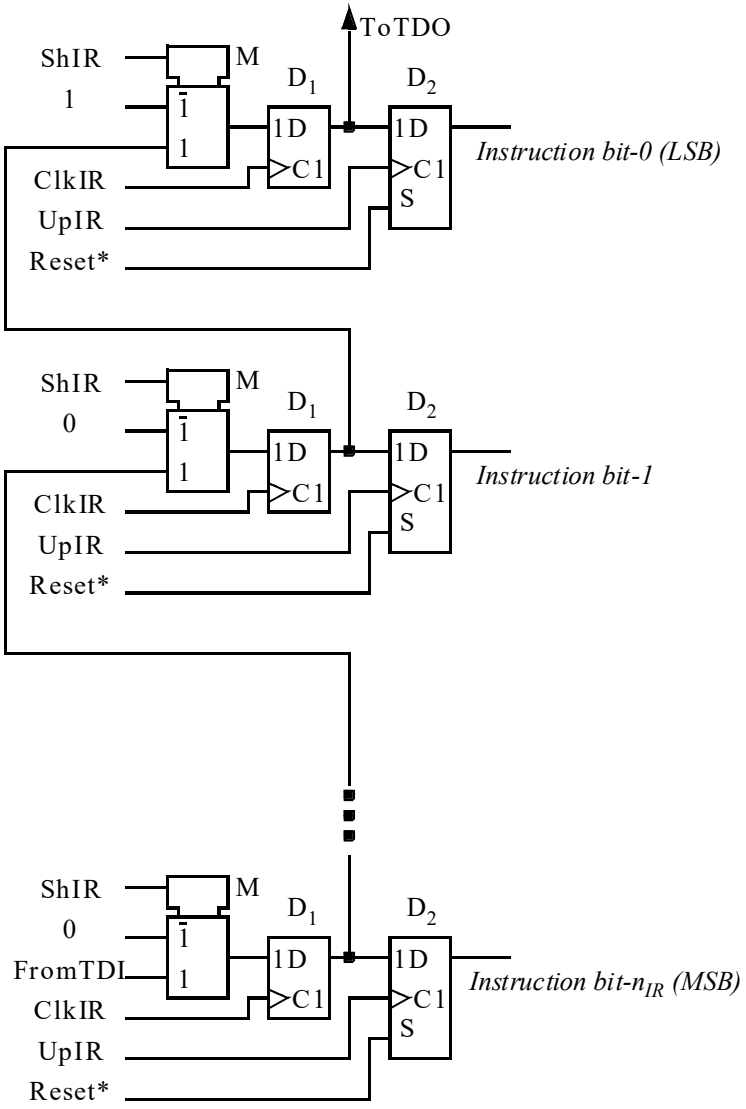
More on TAP Controller States

- Test-Logic-Reset state
  - Entered when
    - A chip is powered up, or
    - Five or more consecutive 1s applied at TMS, or
    - 0 applied at TRST* input
  - Disables test logic by initializing IR to the BYPASS instruction
  - Allows system logic to operate normally
- Run-Test/Idle state
  - For most instructions, test logic become idle and system logic operates normally
  - For some instructions, e.g., RUNBIST, self-test is performed on the system logic for a specified number of clocks by applying an appropriate number of 0s at TMS



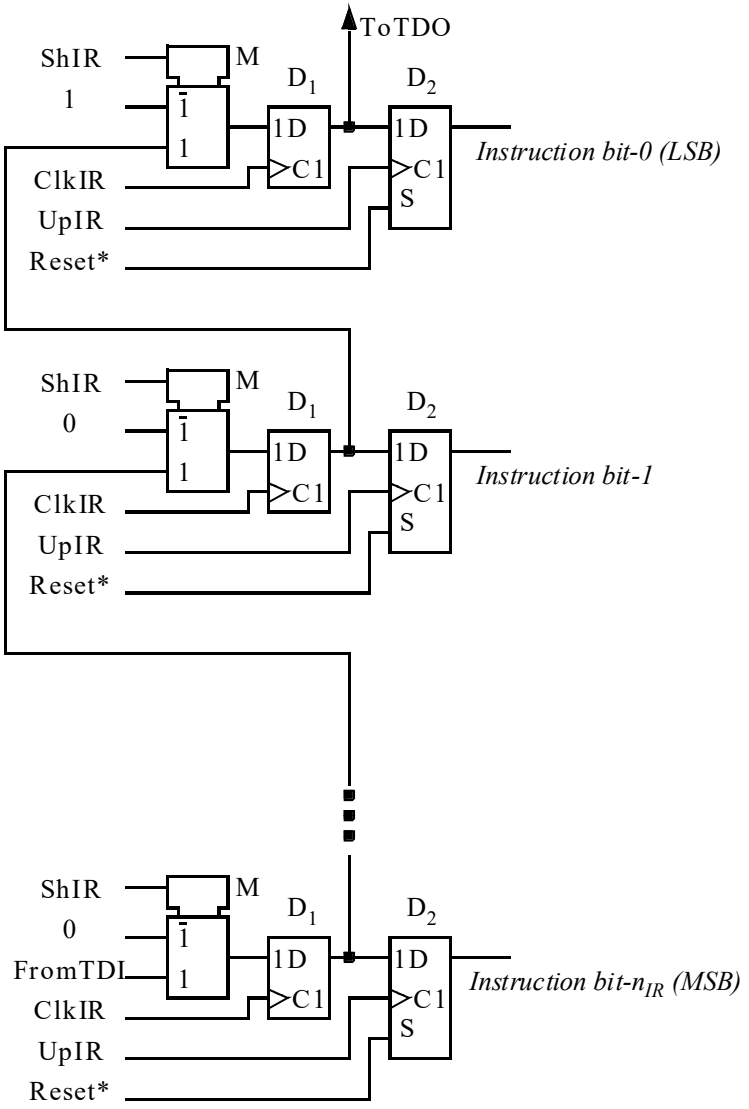
Operation of Instruction Register

- Instruction register
  - If instructions n_{IR}-bits long, then register has n_{IR} bits
  - The bit close to TDO is considered LSB
  - The multiplexers and D_1 FFs form the shift-register within IR
  - The D_2 FFs constitute the output stage
  - The contents of the shift-register can be changed without changing the output



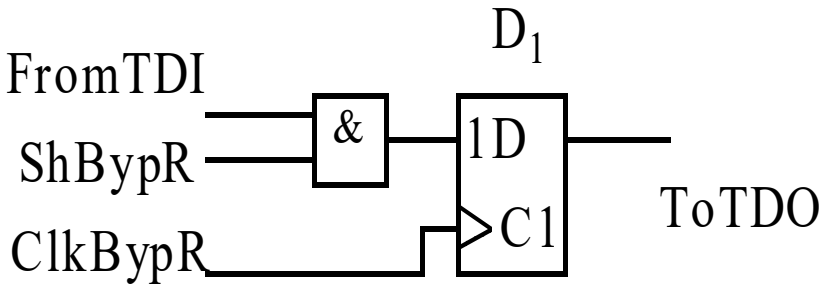
Operation of Instruction Register (cont.)

- In the *Shift-IR* state of TAP controller
  - *ShIR* held high and gated clock *ClkIR* generated to shift in bits from TDI into the shift register
- In the *Capture-IR* state
  - *ShIR* held low and *ClkIR* generated to clock 10…0 into the shift register
- In above cases, the instruction at outputs remains unchanged
  - Holds current instruction in place
  - Prevents application of intermediate values in the shift-register as an instruction
- In *Update-IR* state of the TAP controller
  - Clock *UpIR* generated to load the contents of shift-register into the output stage to update the instruction



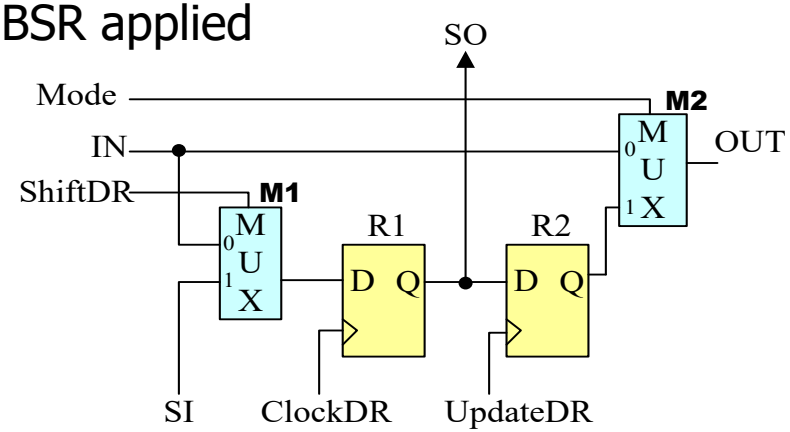
Operation of Bypass Register

- Bypass register
  - —Mandated by the standard
  - —Used to connect TDI to TDO via a 1-bit connection
  - —Used when
    - This chip is not being tested
    - But other chips or their interconnects might be
  - —When BYPASS instruction at the outputs of IR, in the Shift-DR state of the TAP controller
    - ShBypR is high and ClkBypR generated to shift value from TDI to TDO



Operation of Boundary Scan Register

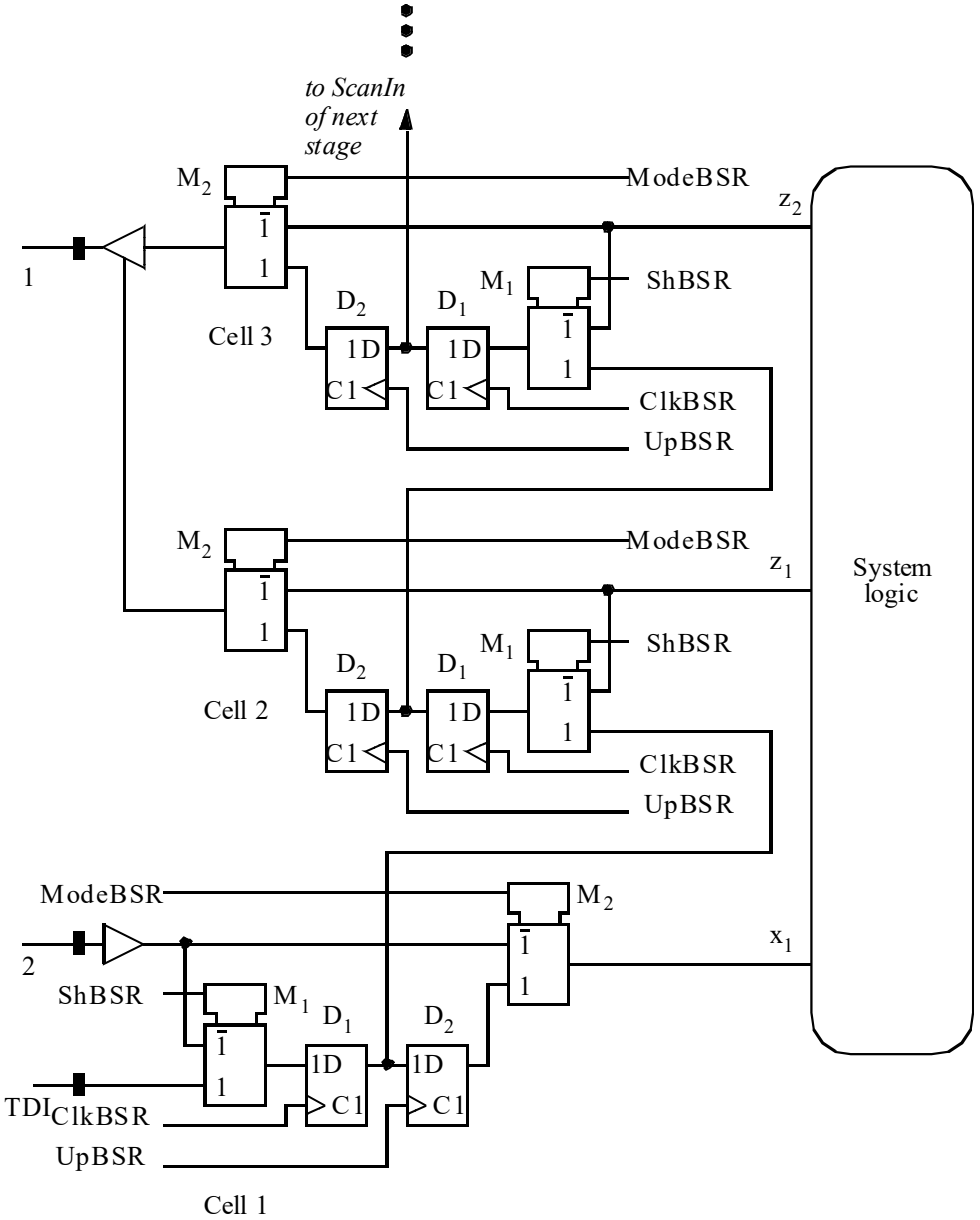
- Boundary scan register (BSR)
  - Inserted between the on-chip system logic and pad logic at chip pins
  - Used to support the mandatory EXTEST and SAMPLE/PRELOAD instructions and many other optional instructions
  - In this design, the main components are
    - R_1 FFs connected to form a shift register
    - M_1 multiplexers control whether normal or scan data loaded into R_1 FFs
    - R_2 FFs hold the test data applied by the register (similar to the output stage of IR)
    - Hence the shift register within can capture response and/or scan-out/scan-in data without changing the values at the output stage
    - M_2 multiplexers control whether the normal values or values at the output stage of BSR applied



Examples of Mode of Operations

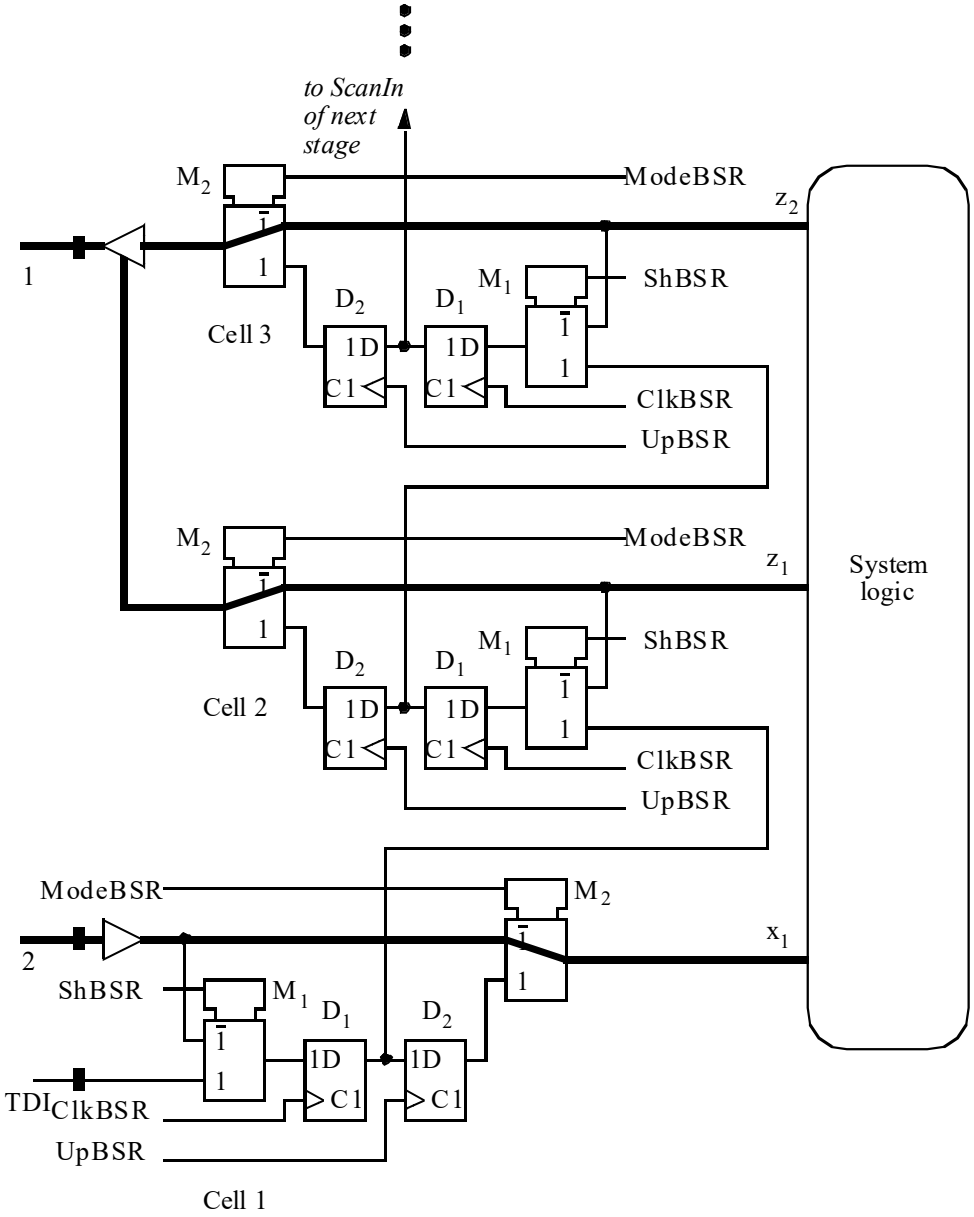
Example Configuration

- Input: x1
- Output: z1 and z2
- Number of boundary scan cells: 3



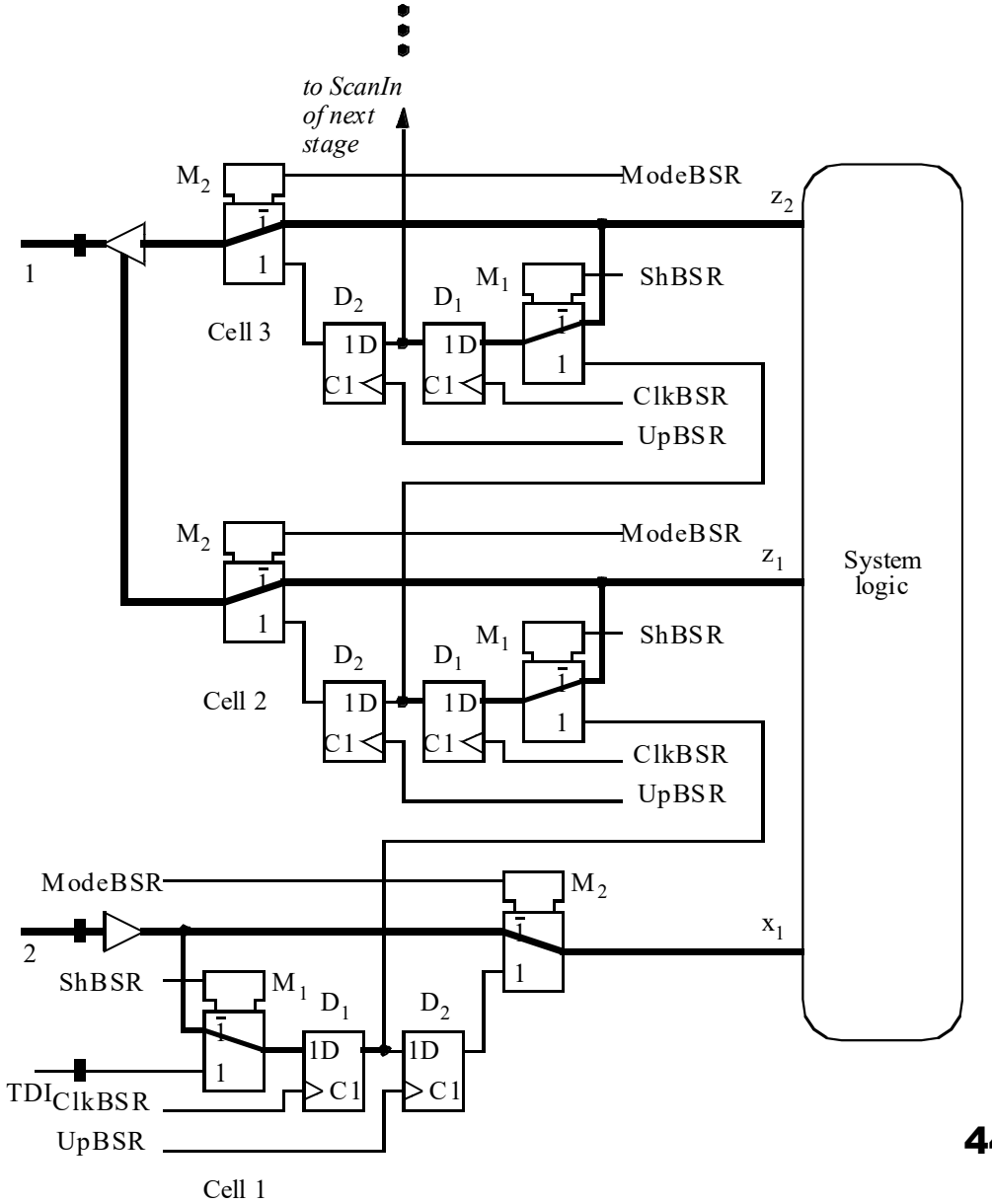
Normal Mode

- 0 applied at *ModeBSR*
  - The value at *NormalIn* appears at *NormalOut*
  - That is, the system logic operates in normal mode
- Gated clocks *ClkBSR* and *UpBSR* disabled
- The value at *ShBSR* does not affect operation



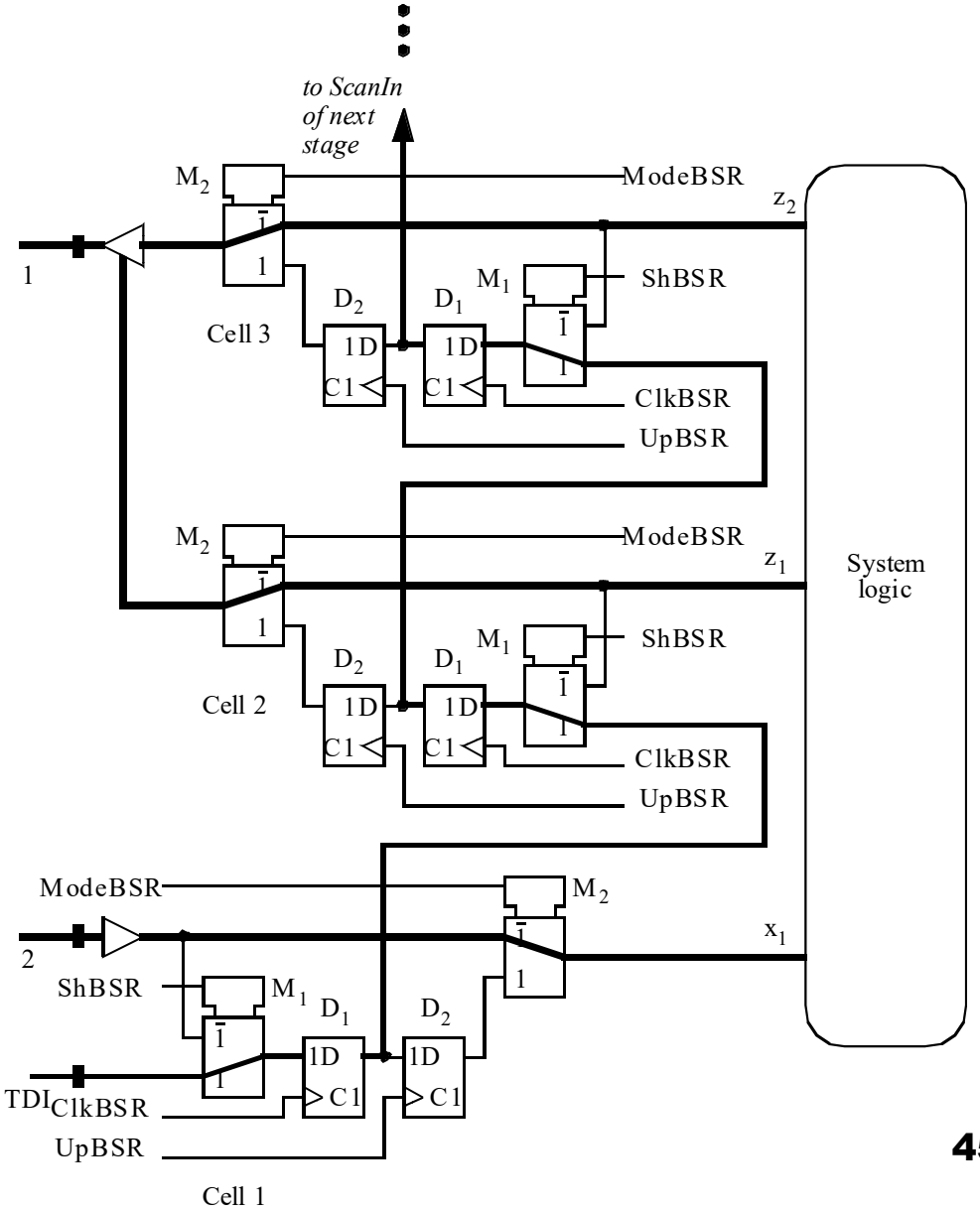
Normal-Sample Mode

- 0 applied at *ModeBSR*
  - The value at *NormalIn* appears at *NormalOut*
  - That is, the system logic operates in normal mode
- Gated clock *UpBSR* disabled
- 0 applied at *ShBSR*
- One clock pulse applied to *ClkBSR* to capture the values at *NormalIn* into the corresponding D_1 FFs
  - This clock synchronized with system clock that controls data at *NormalIn* inputs
  - This occurs only when TAP controller in *Capture-DR* state



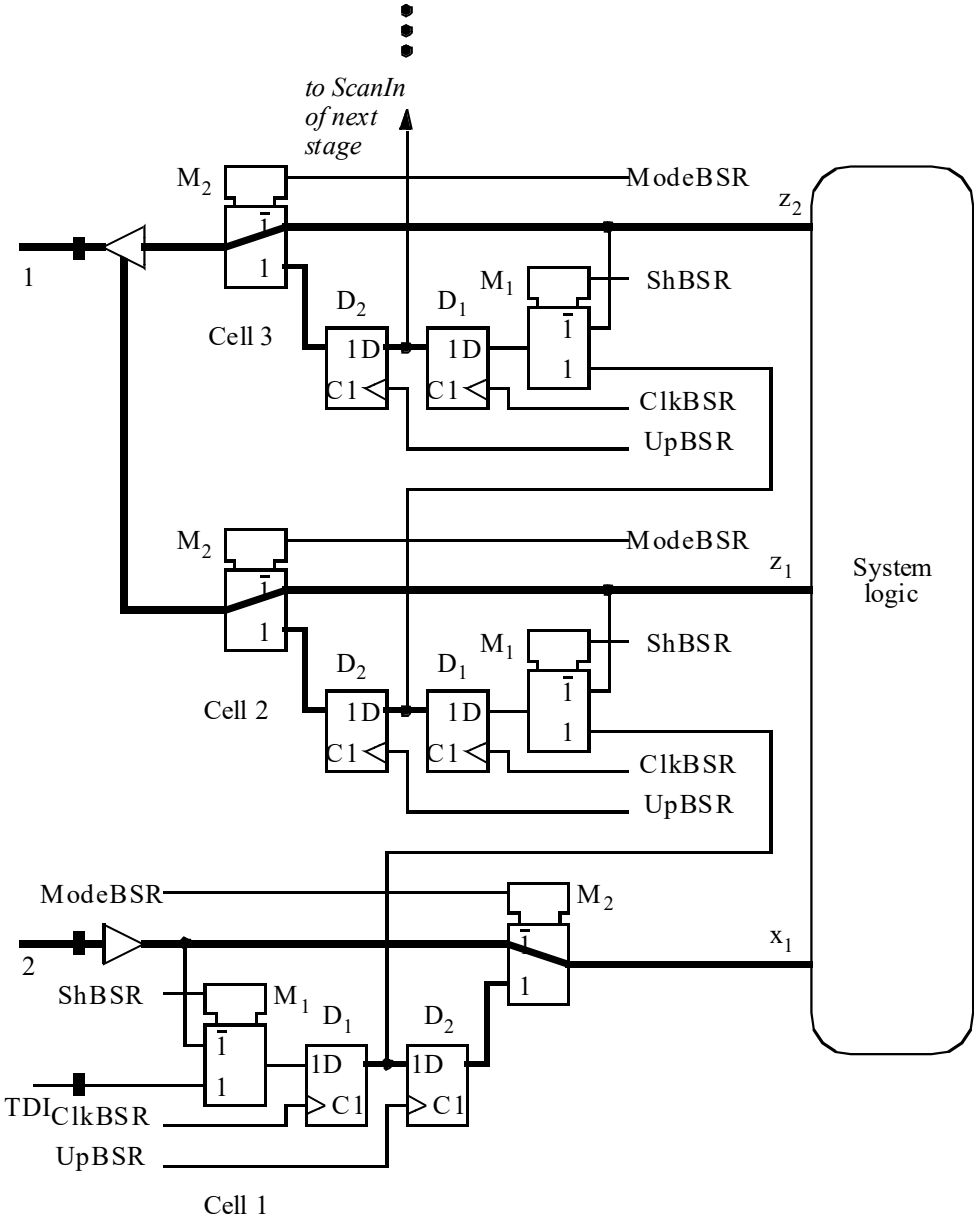
Normal-Scan Mode

- 0 applied at *ModeBSR*
  - The value at *NormalIn* appears at *NormalOut*
  - That is, the system logic operates in normal mode
- Gated clock *UpBSR* disabled
- 1 applied at *ShBSR*
- A sequence of clock pulses applied to *ClkBSR* to
  - Scan out any response previously captured in D_1 FFs
  - Scan in appropriate values into D_1 FFs from TDI
  - This occurs only when TAP controller in *Shift-DR* state



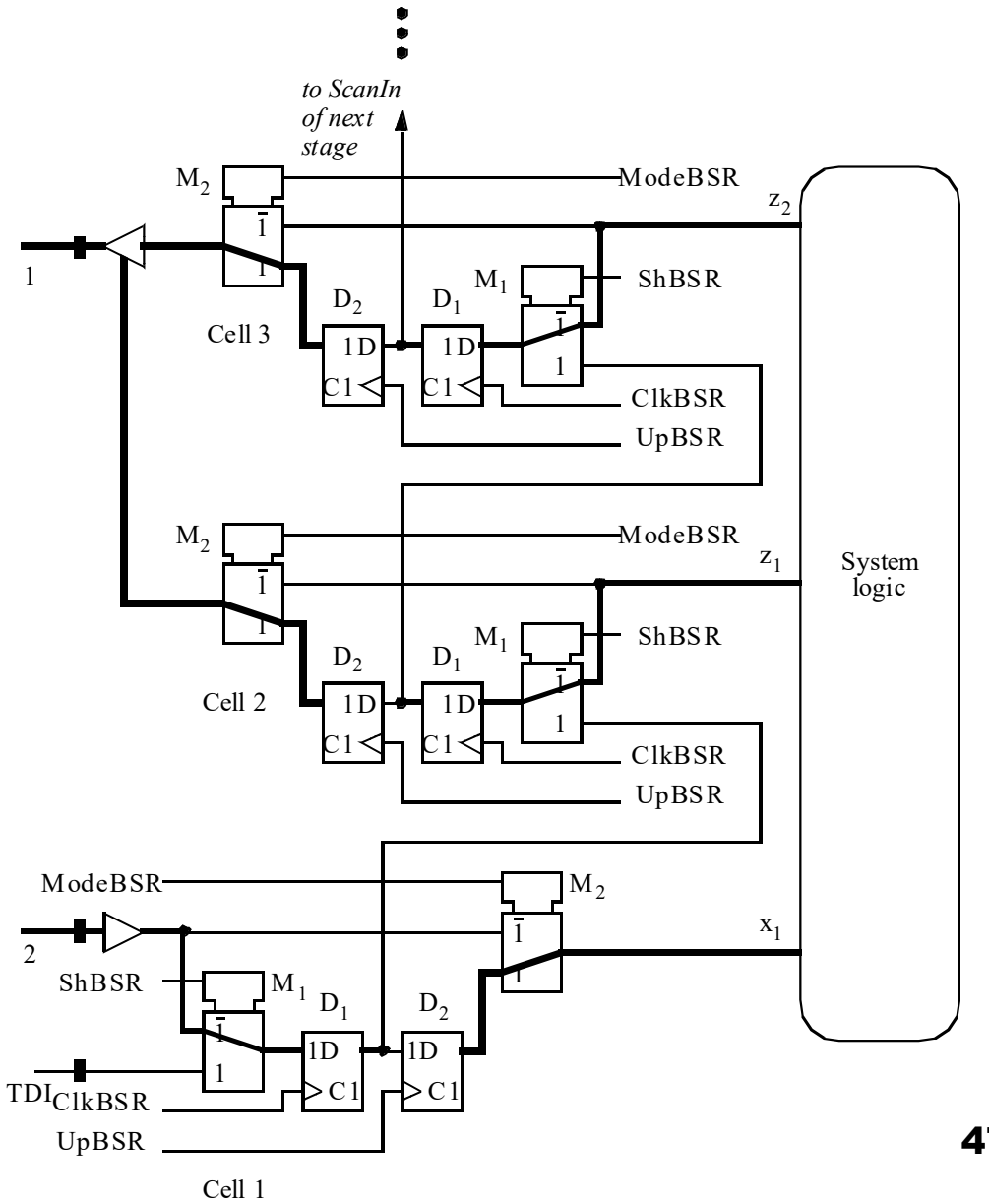
Normal-Preload Mode

- 0 applied at *ModeBSR*
  - The value at *NormalIn* appears at *NormalOut*
  - That is, the system logic operates in normal mode
- *ClkBSR* held inactive, *ShBSR* unimportant
- One clock pulse applied at *UpBSR*
  - To load the data currently held in D_1 FFs into corresponding D_2 FFs
  - This occurs only when TAP controller in *Update-DR* state



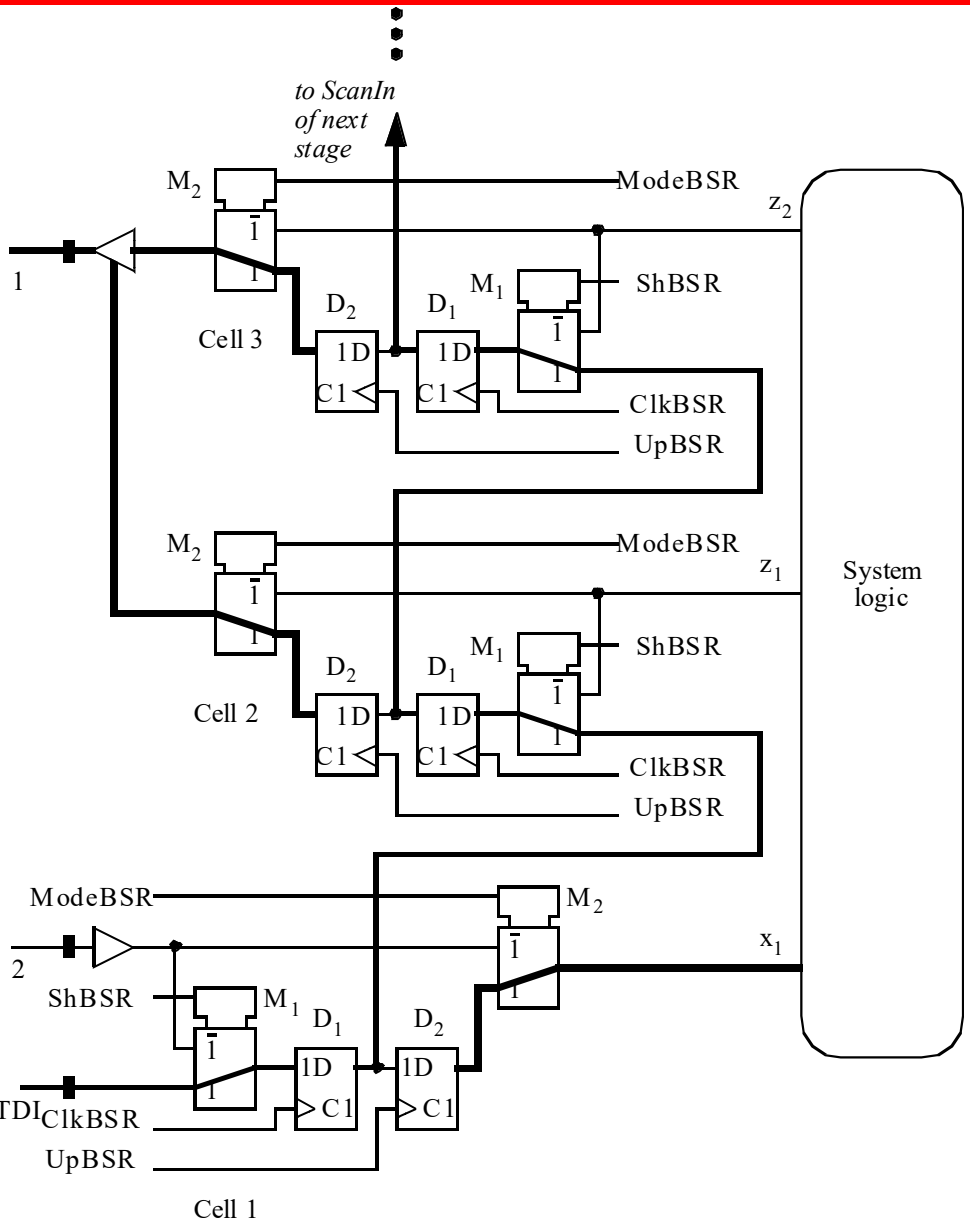
Test-Capture Mode

- 1 applied at *ModeBSR*
  - Values at D_2 FFs appear at *NormalOut*
  - That is, the system logic operates in test mode
- Gated clock *UpBSR* disabled
- 0 applied at *ShBSR*
- One clock pulse applied to *ClkBSR* to capture the values at *NormalIn* into the corresponding D_1 FFs
  - This occurs only when TAP controller in *Capture-DR* state



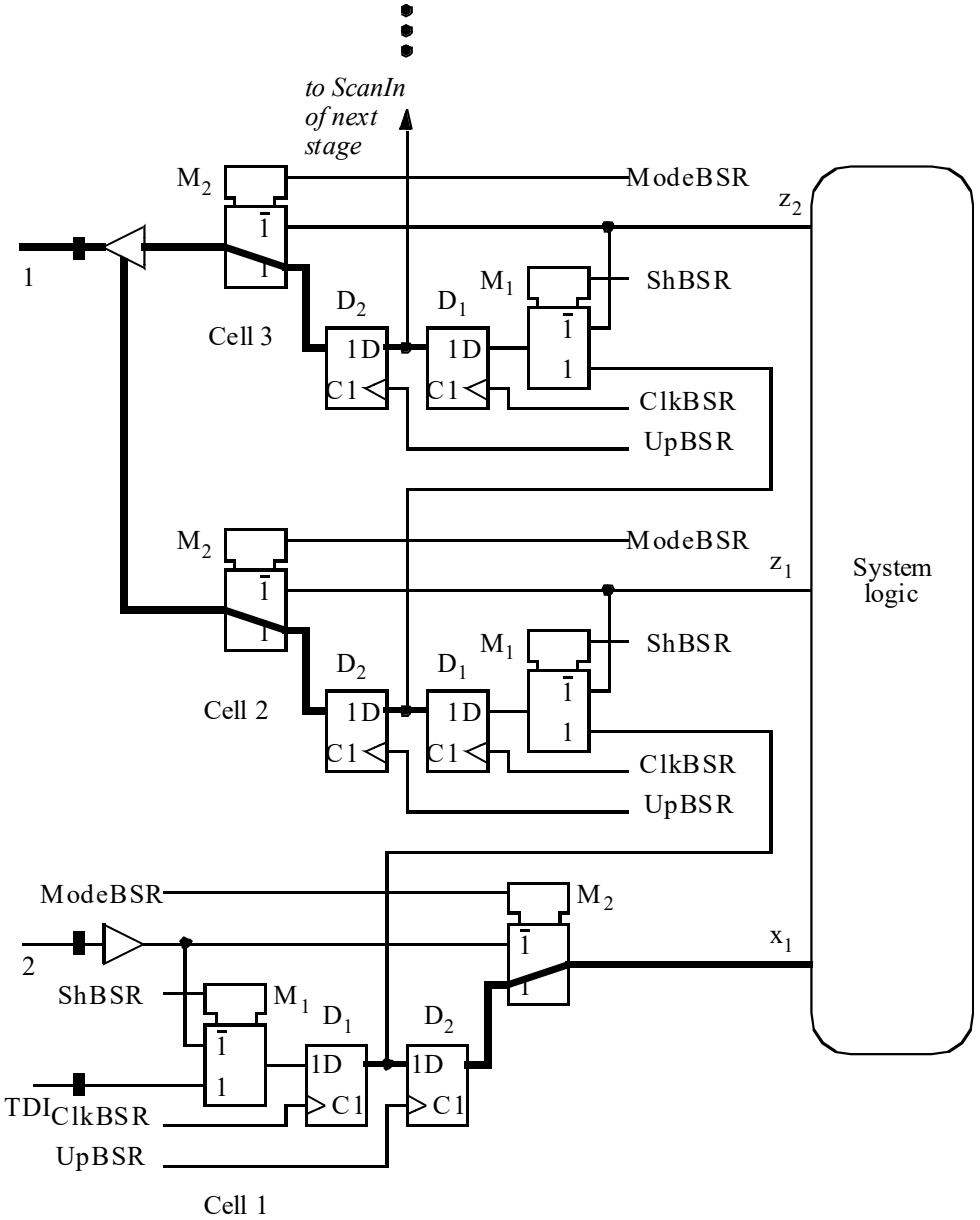
Test-Scan Mode

- 1 applied at *ModeBSR*
  - Values at D_2 FFs appear at *NormalOut*
  - That is, the system logic operates in test mode
- *UpBSR* held inactive
- 1 applied at *ShBSR*
- A sequence of clock pulses applied to *ClkBSR* to
  - Scan out any response previously captured in D_1 FFs
  - Scan in appropriate values into D_1 FFs from TDI
  - This occurs only when TAP controller in *Shift-DR* state



Test-Update Mode

- 1 applied at *ModeBSR*
  - Values at D_2 FFs appear at *NormalOut*
  - That is, the system logic operates in test mode
- *ClkBSR* held inactive, *ShBSR* unimportant
- One clock pulse applied at *UpBSR*
  - To load the data currently held in D_1 FFs into corresponding D_2 FFs
  - This occurs only when TAP controller in *Update-DR* state



Instruction Set – IEEE 1149.1

Instruction Set

- BYPASS
  - Bypass data through a chip
- SAMPLE
  - Sample (capture) test data into BSR
- PRELOAD
  - Shift-in test data and update BSR

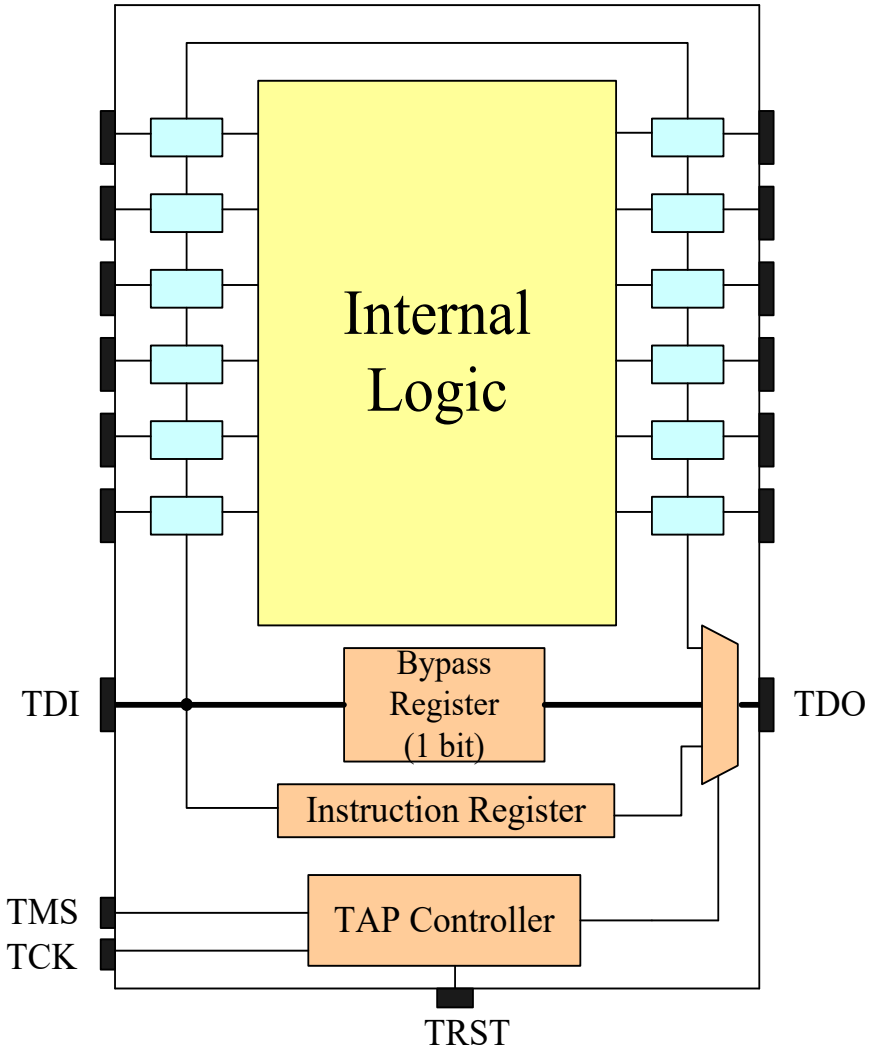
SAMPLE/PRELOAD
(internal logic remains in its functional mode

- EXTEST
  - Test interconnection between chips of board
- Optional
  - INTEST, RUNBIST, CLAMP, IDCODE, USERCODE, HIGH-Z, etc.

About Optional Instructions

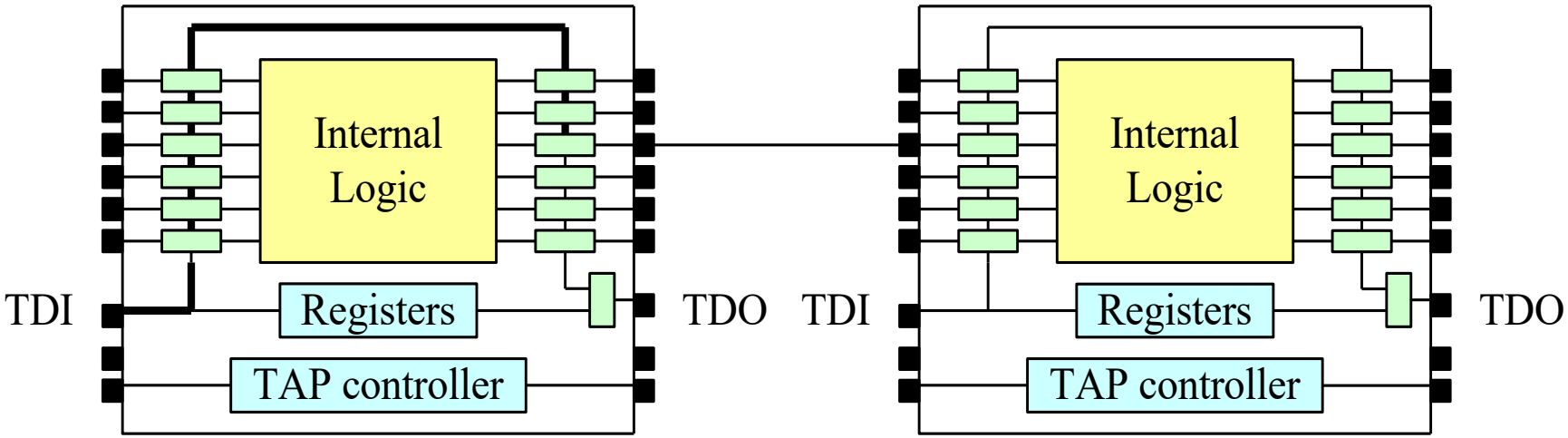
- **INTEST**: Selects the boundary scan registers preparatory to applying tests to the internal logic of device.
- **RUNBIST**: Targets a self-test result register between TDI and TDO which holds the pass/fail result in BIST test.
- **IDCODE**: Selects ID registers between TDI and TDO, preparatory to loading the internally-held 32-bit identification code and reading it out through TDO.
- **USERCODE**: Selects the same 32-bit ID register but allows an alternative 32-bits of identity data to be loaded and serially shifted out. This is used for dual-personality devices (e.g. code indicates the way a device like PLD/FPGAs programmed).
- **CLAMP**: Works like a preload that is followed by bypass. It is used to set up safe guarding values on the outputs of certain devices in order to avoid bus contention problems.
- **HIGH-Z**: Similar to Clamp, but it leaves the device output pins in a high-Z state rather than drive fixed logic 1 or 0 values. Similar to Clamp, it also selects the bypass register.

Execution of BYPASS Instruction



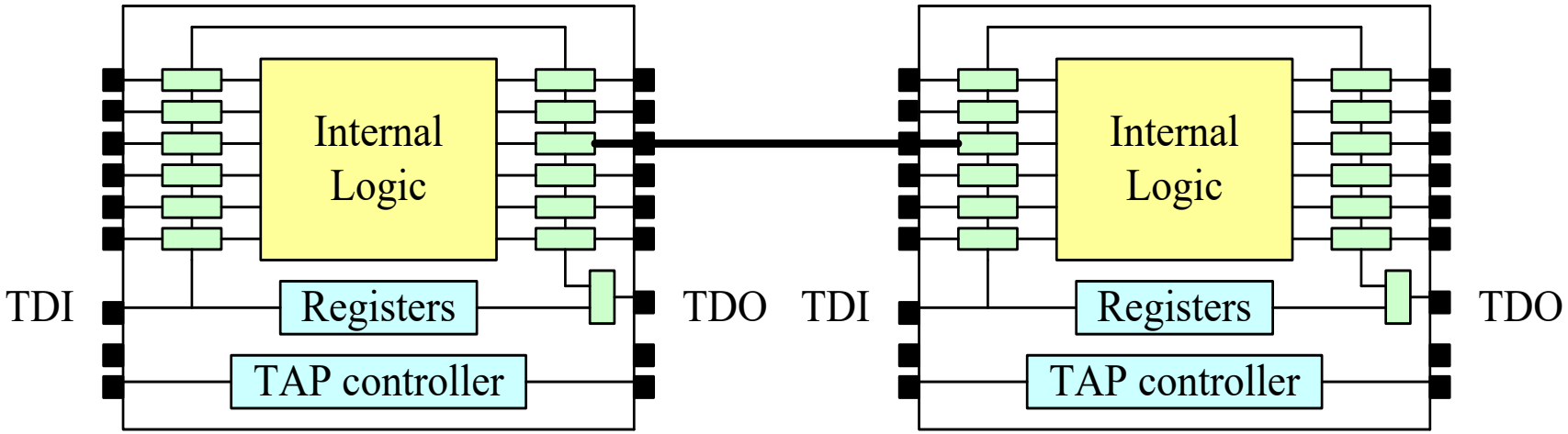
Execution of EXTEST Instruction (1/3)

- Shift-DR (Chip1)



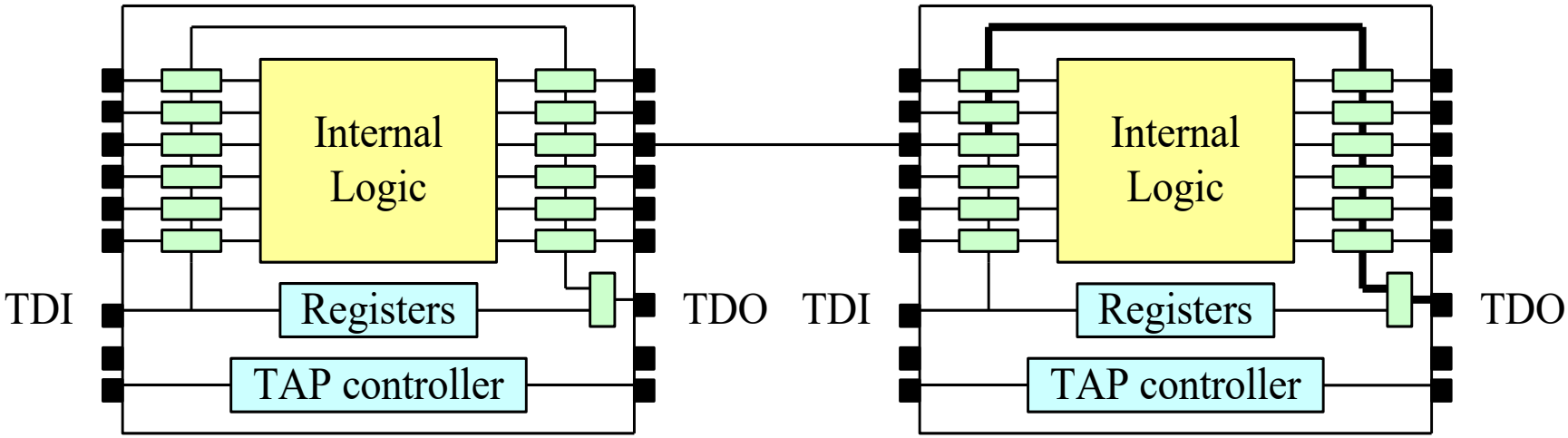
Execution of EXTEST Instruction (2/3)

- Update-DR (Chip1)
- Capture-DR (Chip2)



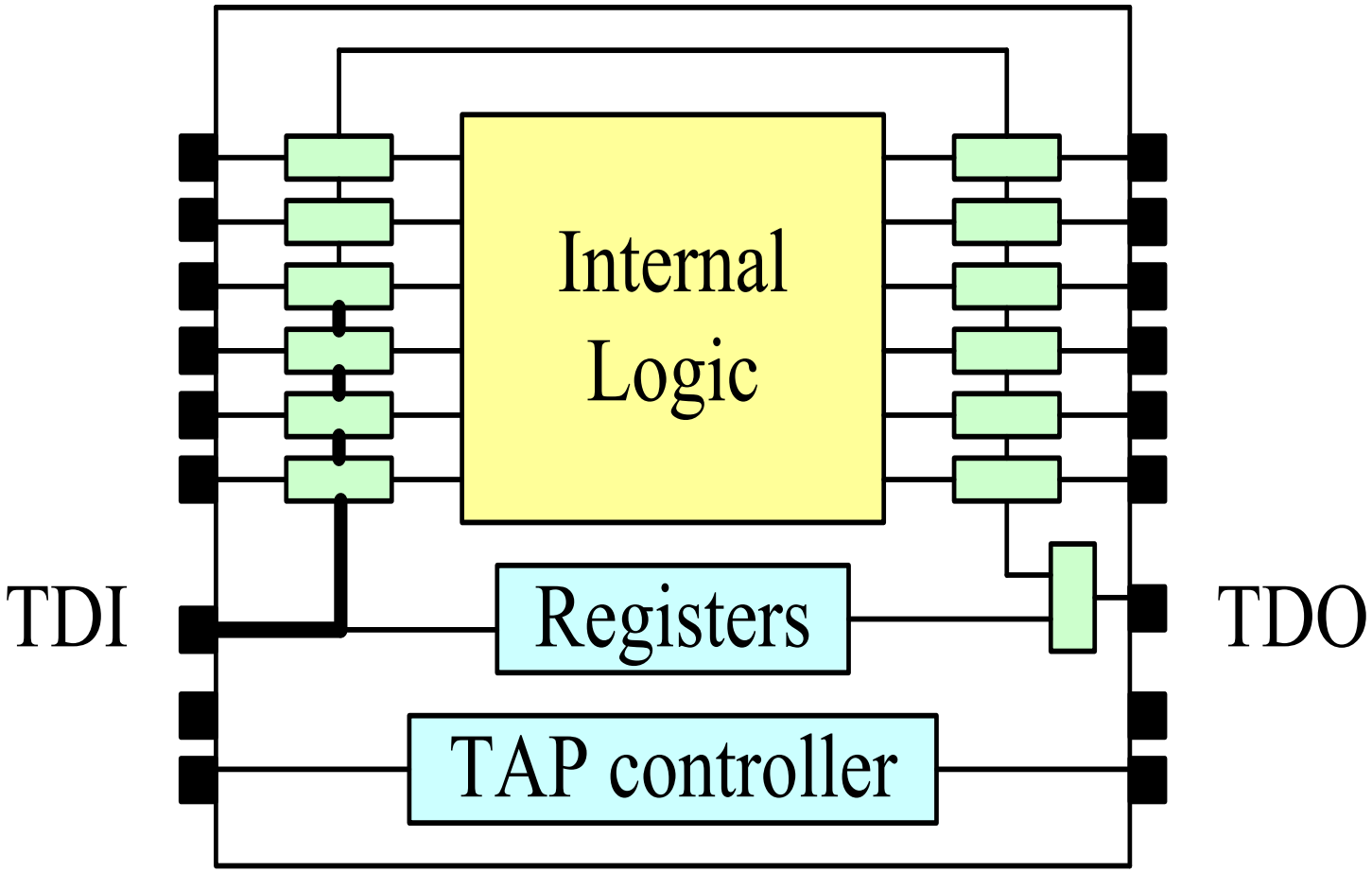
Execution of EXTEST Instruction (3/3)

- Shift-DR (Chip2)



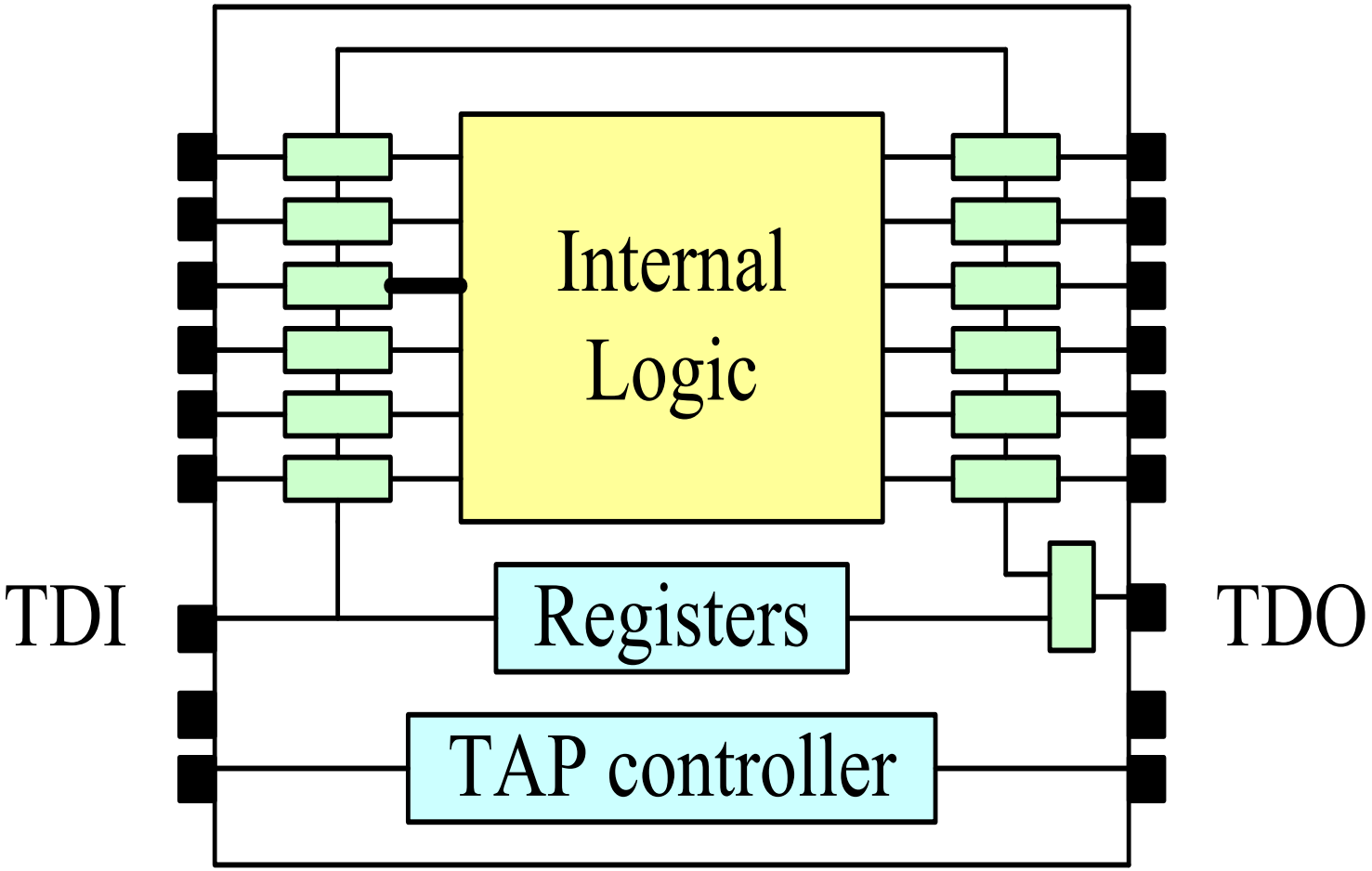
Execution of INTEST Instruction (1/4)

- Shift-DR



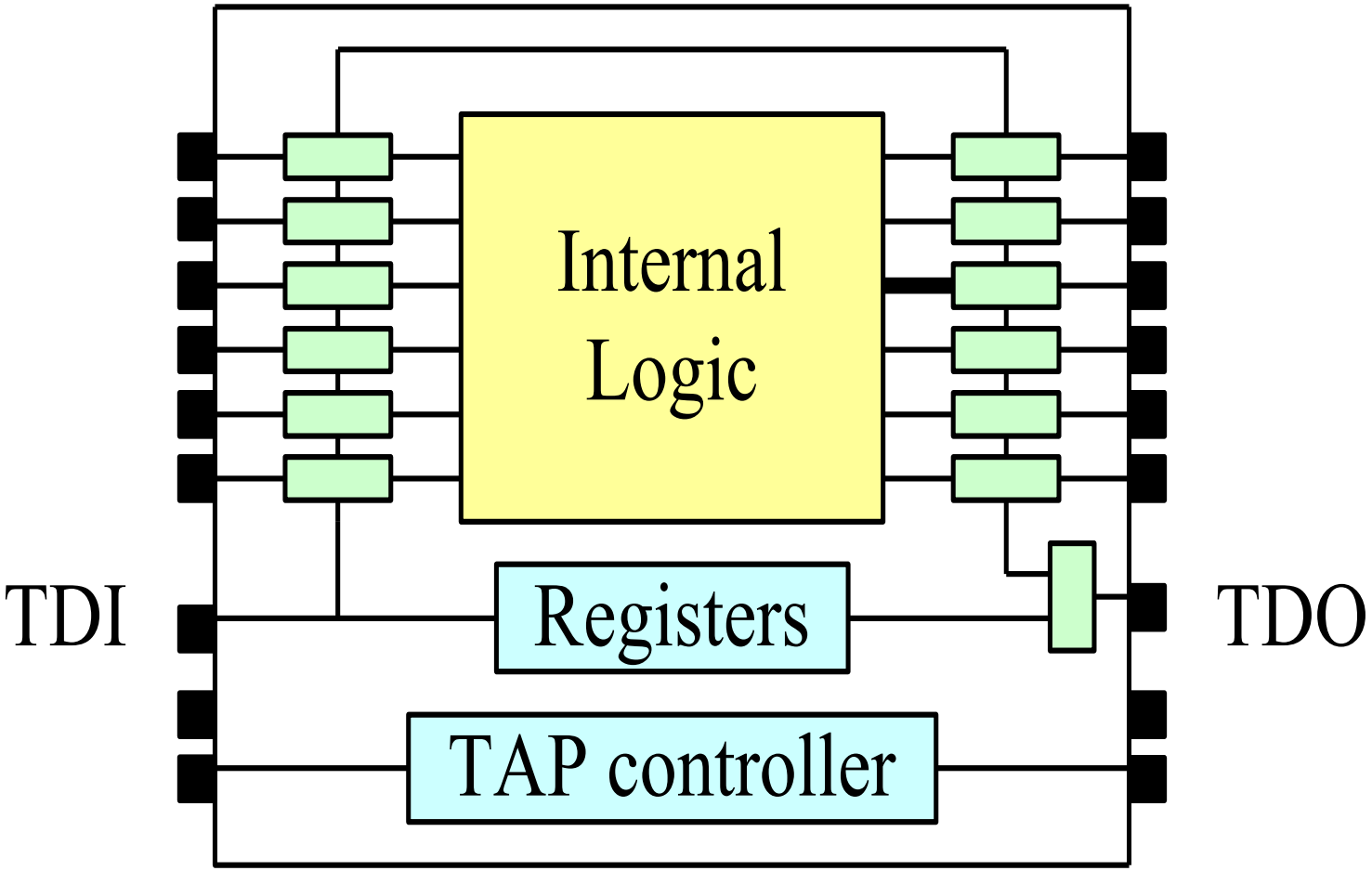
Execution of INTEST Instruction (2/4)

- Update-DR



Execution of INTEST Instruction (3/4)

- Capture-DR



Execution of INTEST Instruction (4/4)

- Shift-DR