Chapter 10

Boundary Scan and Core-Based Testing

Outline

- Introduction
- Digital Boundary Scan (1149.1)
- Boundary Scan for Advanced Networks (1149.6)
- Embedded Core Test Standard (1500)
- Comparison between 1149.1 and 1500

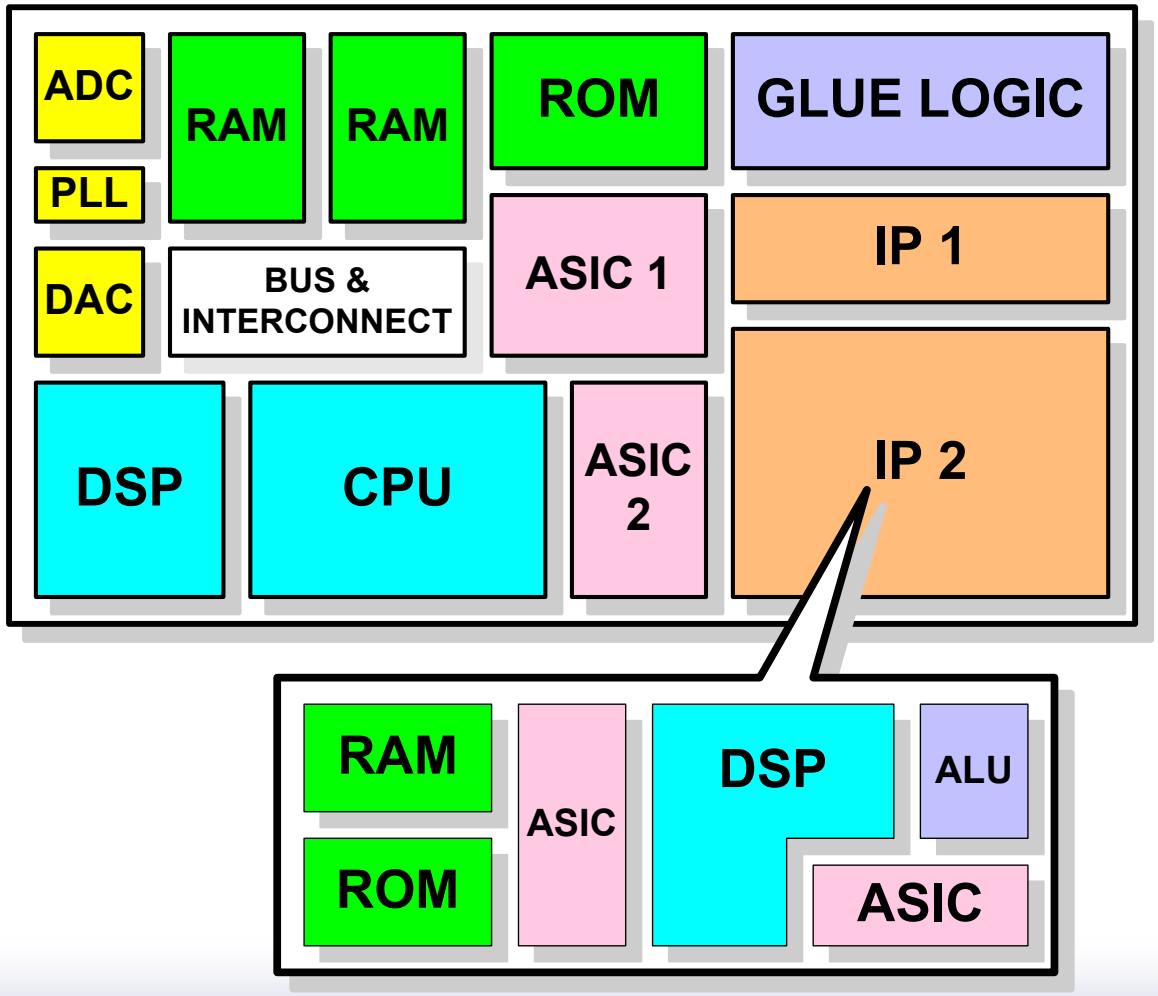
Boundary Scan

- 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

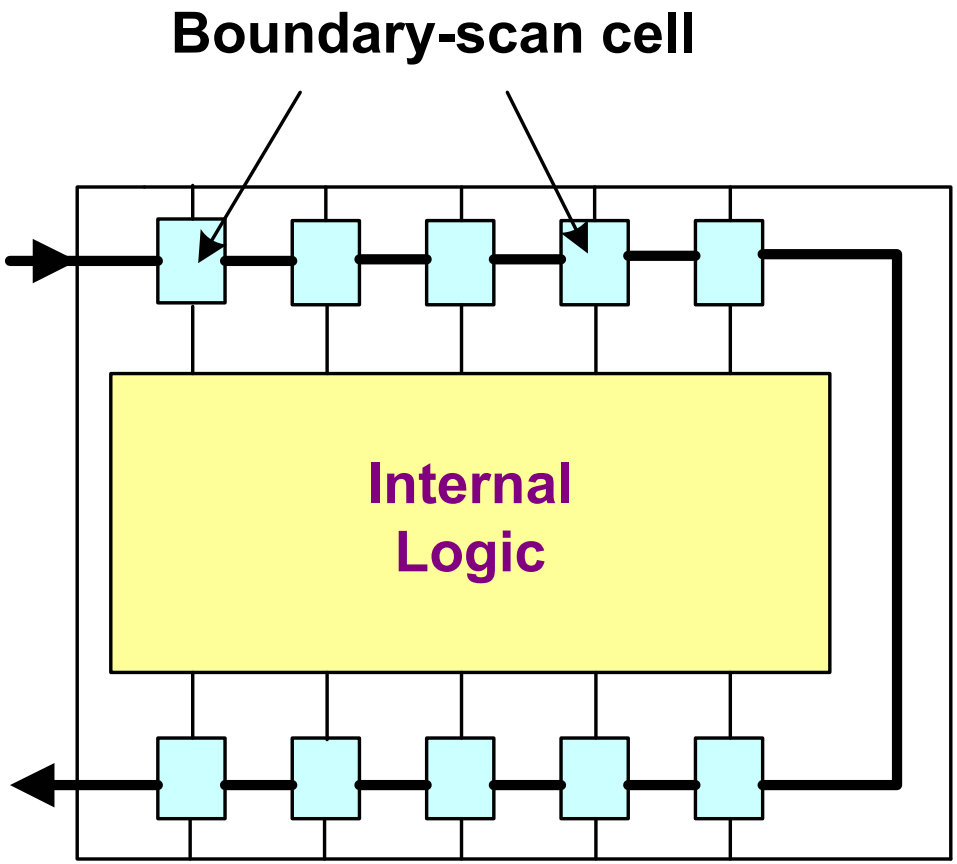
Core-Based SOC Design



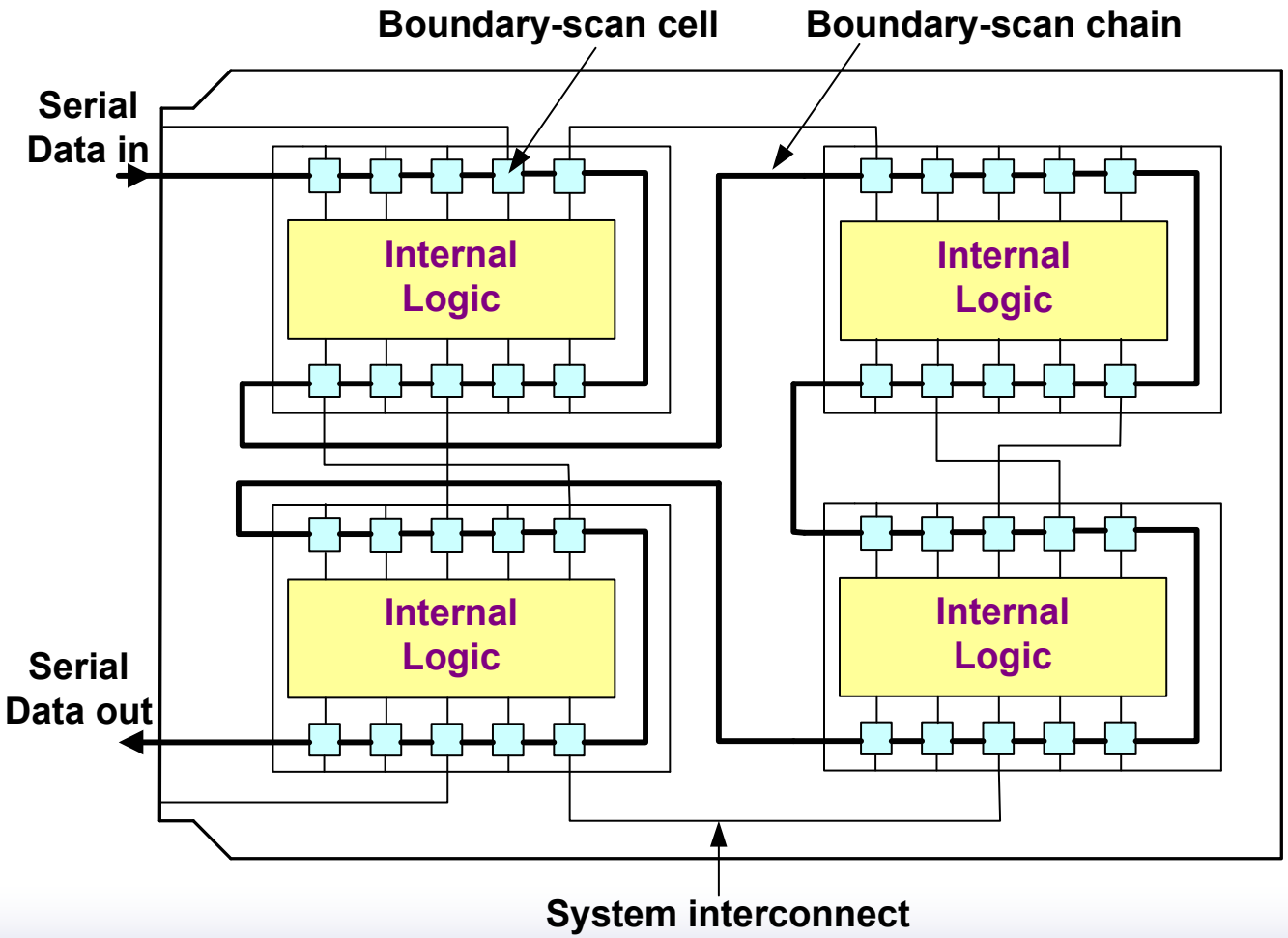
Digital Boundary Scan – 1149.1

- 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

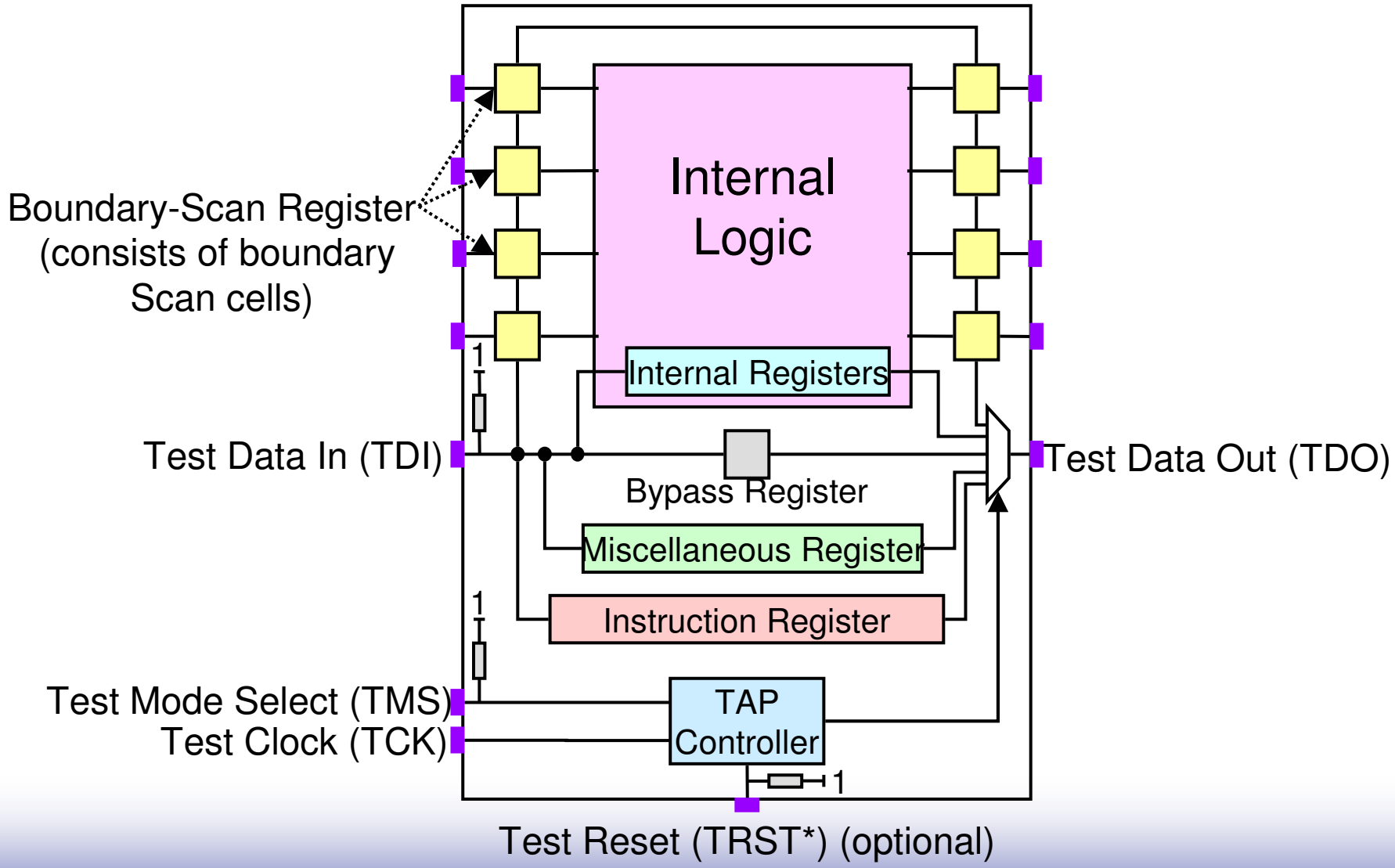
Basic Idea of Boundary Scan



A Board Containing 4 IC's with Boundary Scan



1149.1 Boundary-Scan Architecture



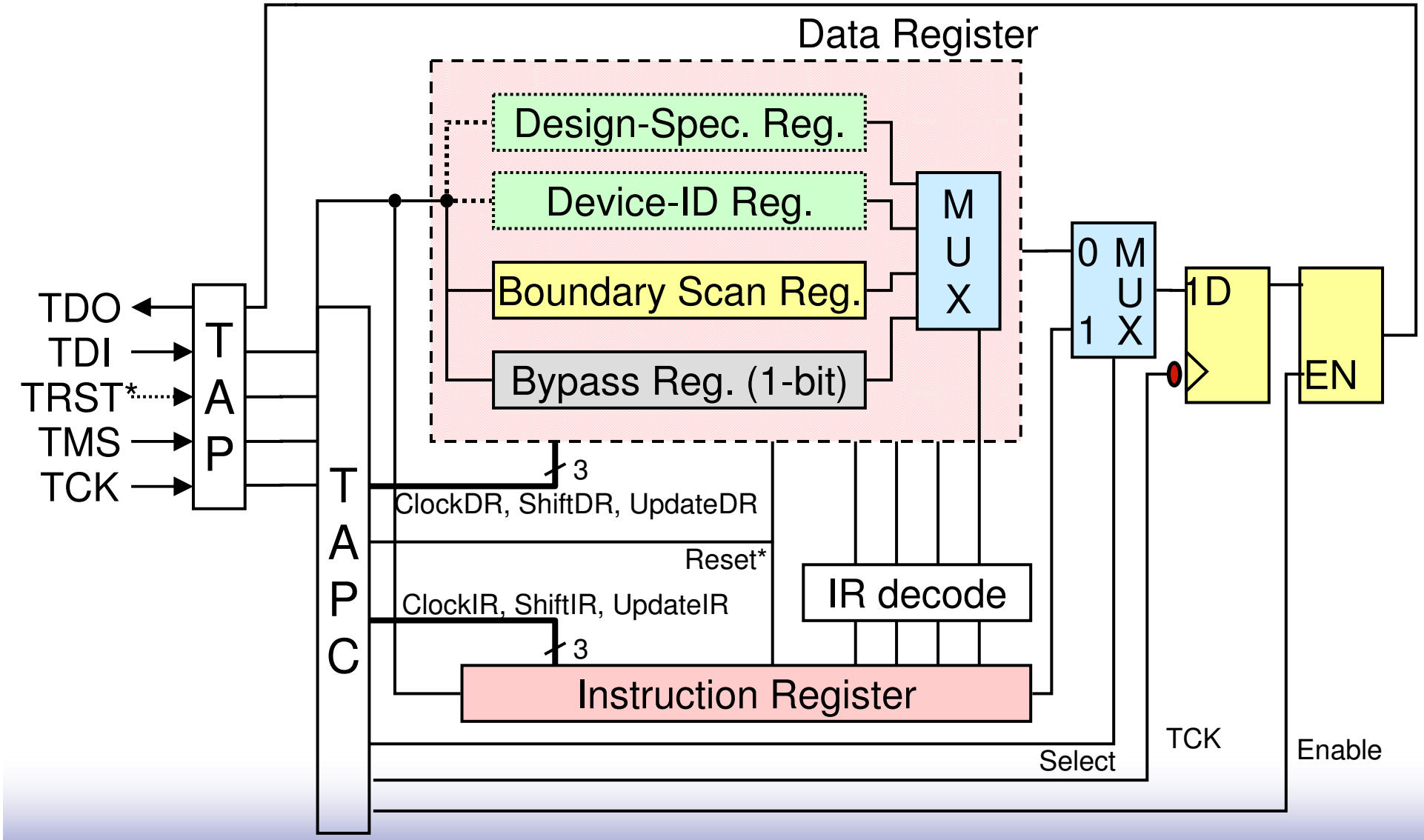
Hardware Components of 1149.1

- 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.

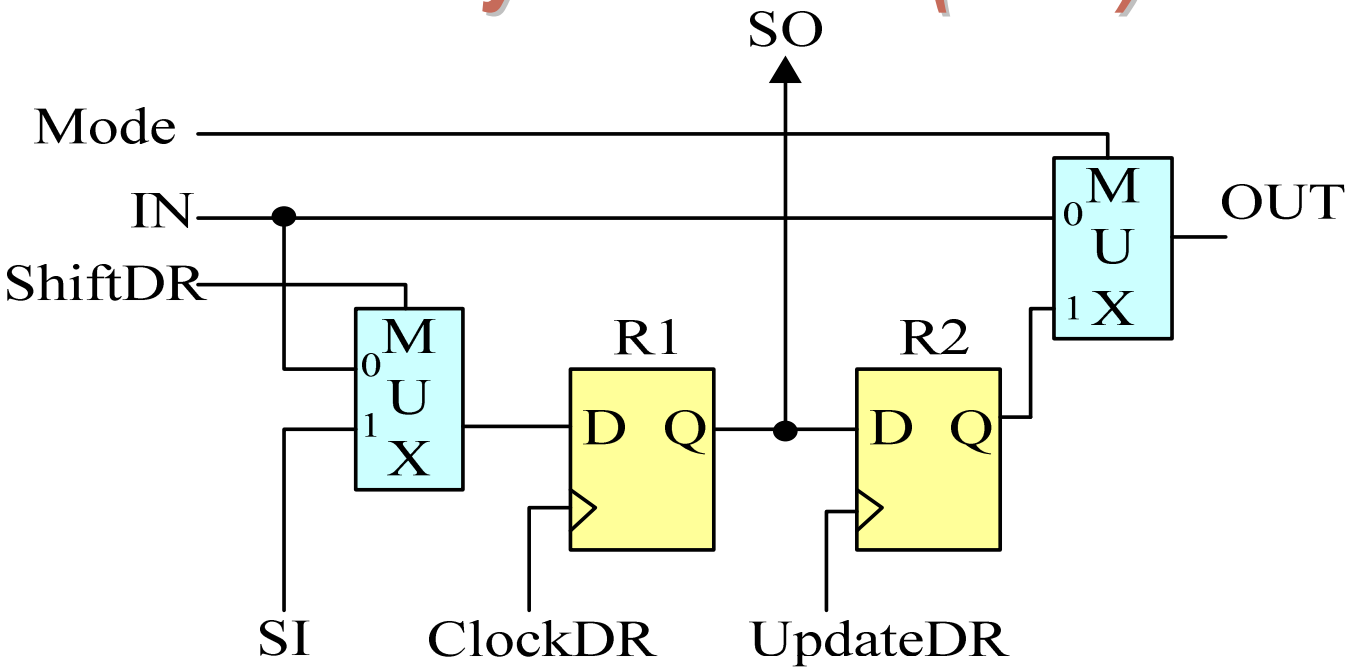
Boundary-Scan Circuitry in A Chip



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

A Typical Boundary-Scan Cell (BSC)

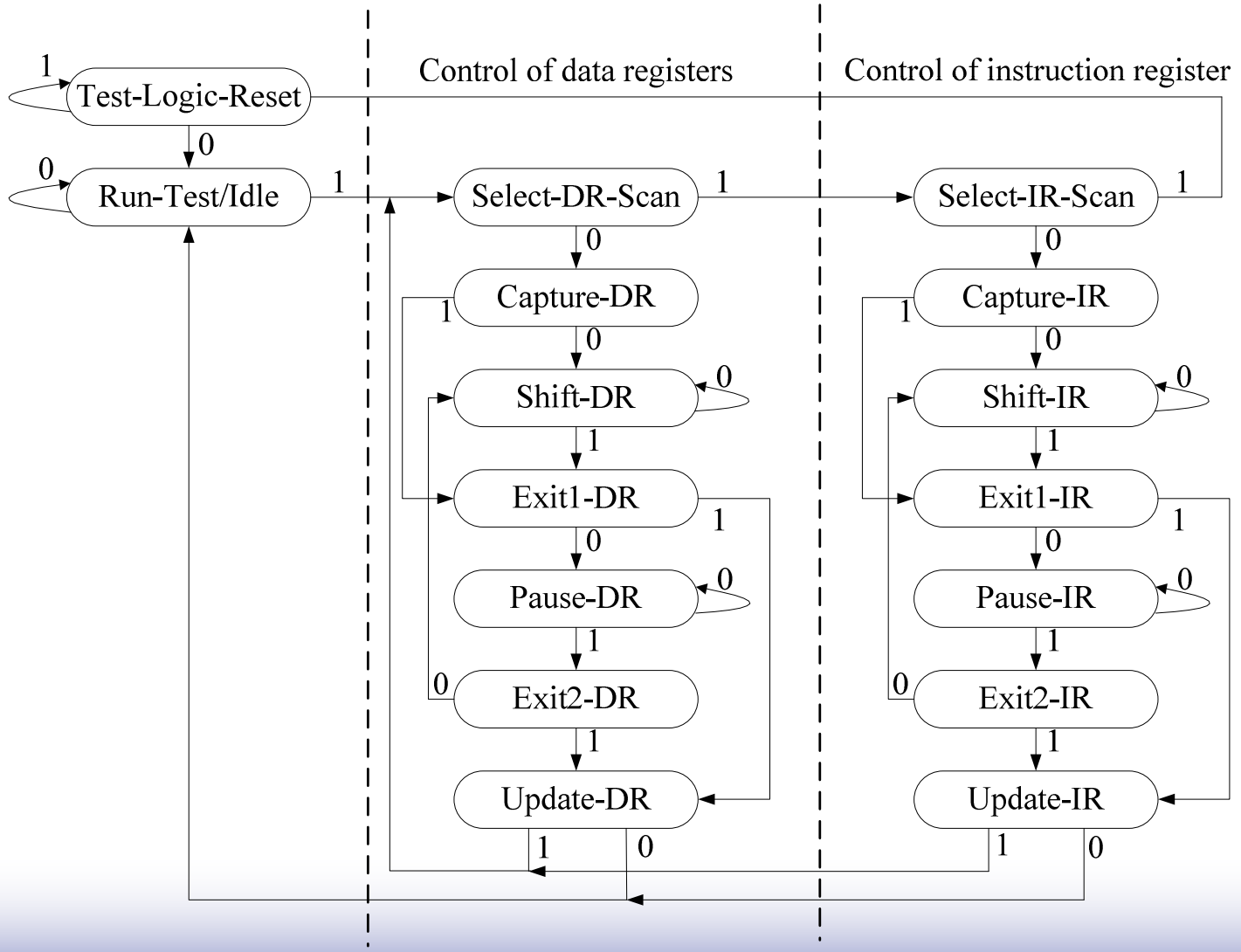


- Operation modes
  - Normal: IN → OUT (Mode = 0)
  - Shift: TDI → ... → IN → OUT → ... → TDO (ShiftDR = 1, ClockDR)
  - Capture: IN → R1, OUT driven by IN or R2 (ShiftDR = 0, ClcokDR)
  - Update: R1 → OUT (Mode_Control = 1, UpdateDR)

TAP Controller

- A finite state machine with 16 states
- Input: TCK, TMS
- Output: 9 or 10 signals included 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

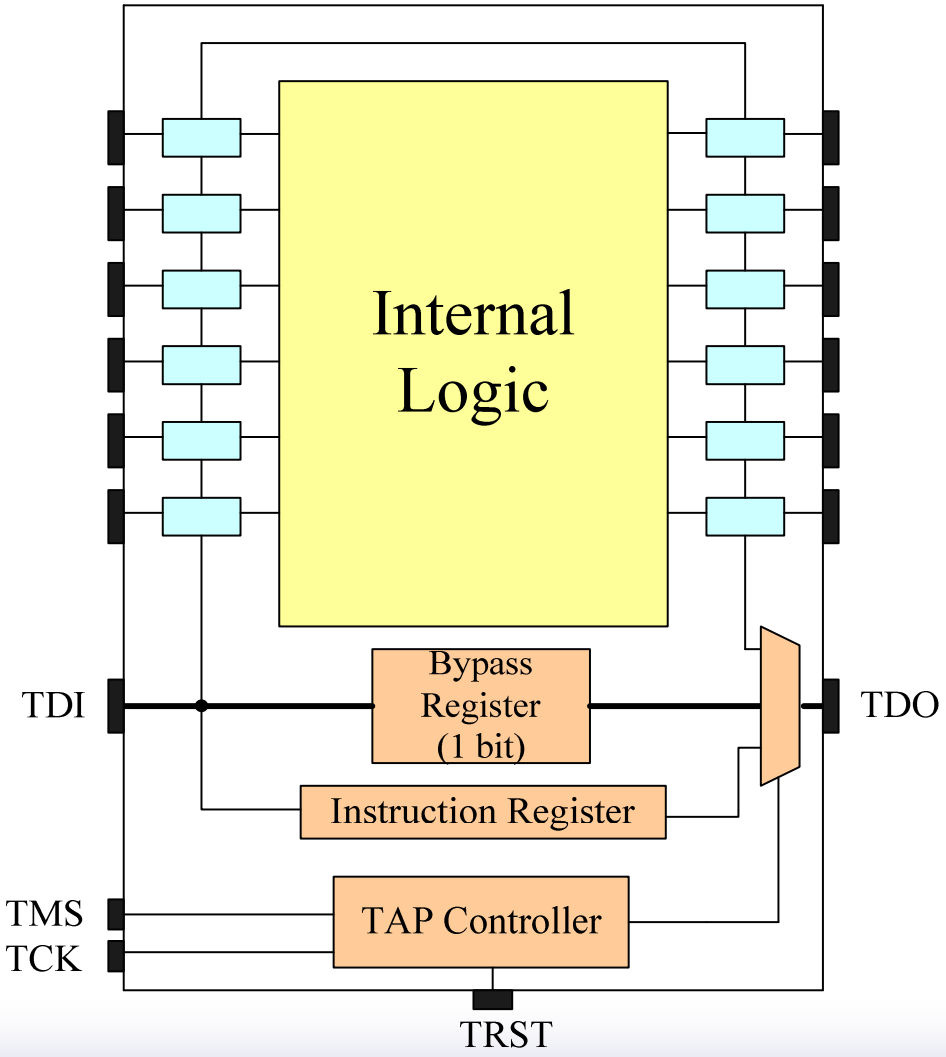
- Test-Logic-Reset: normal mode
- Run-Test/Idle: wait for internal test such as BIST
- Select-DR-Scan: initiate a data-scan sequence
- Capture-DR: load test data in parallel
- Shift-DR: load test data in series
- Exit1-DR: finish phase-1 shifting of data
- Pause-DR: temporarily hold the scan operation (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.

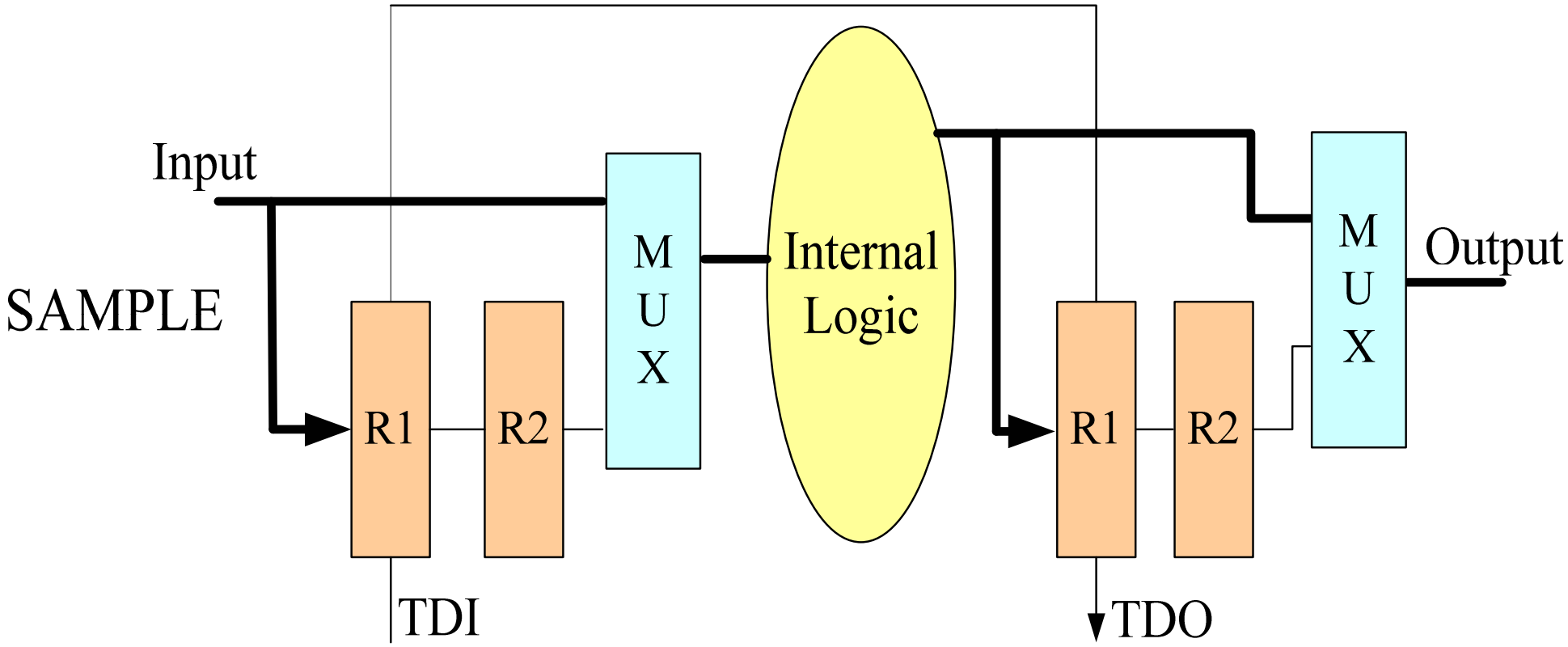
Instruction Set

- BYPASS
  - Bypass data through a chip
- SAMPLE
  - Sample (capture) test data into BSR
- PRELOAD
  - Shift-in test data and update BSR
- EXTEST
  - Test interconnection between chips of board
- Optional
  - INTEST, RUNBIST, CLAMP, IDCODE, USERCODE, HIGH-Z, etc.

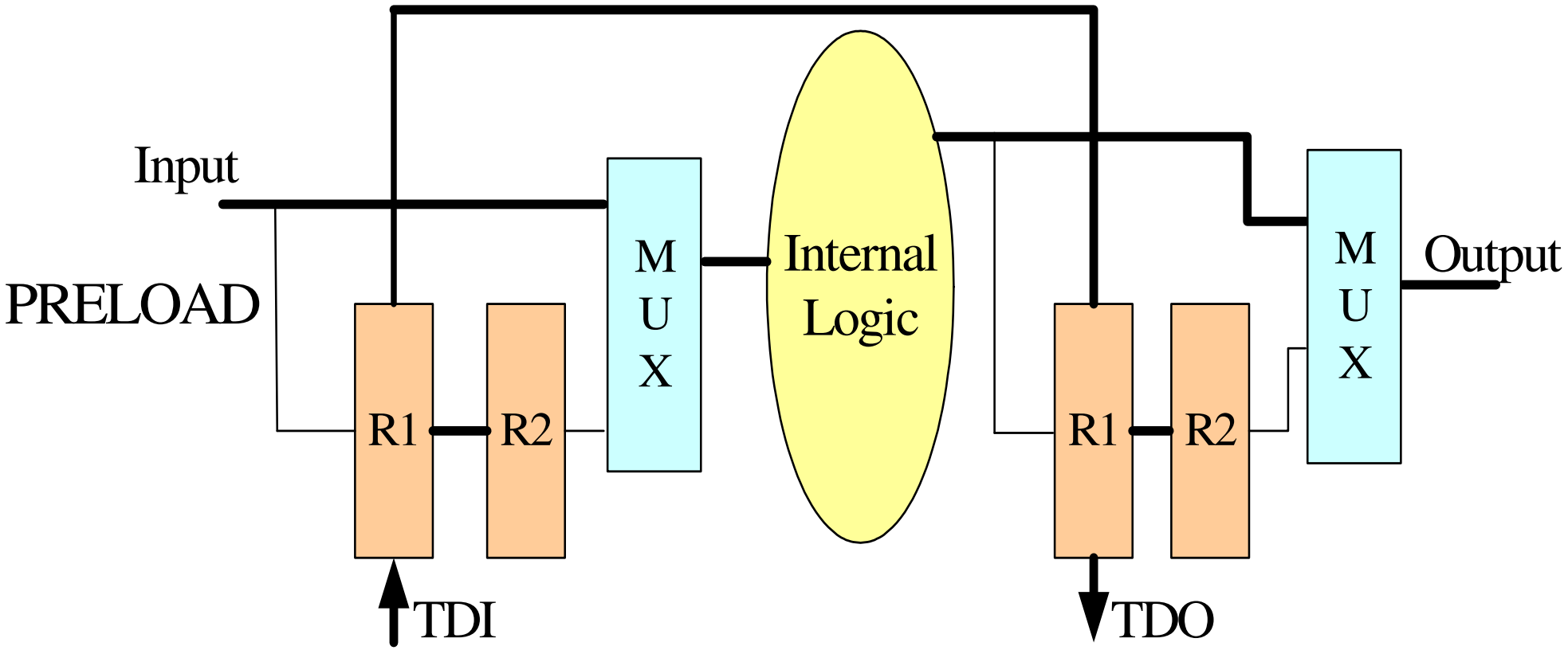
Execution of BYPASS Instruction



Execution of SAMPLE Instruction

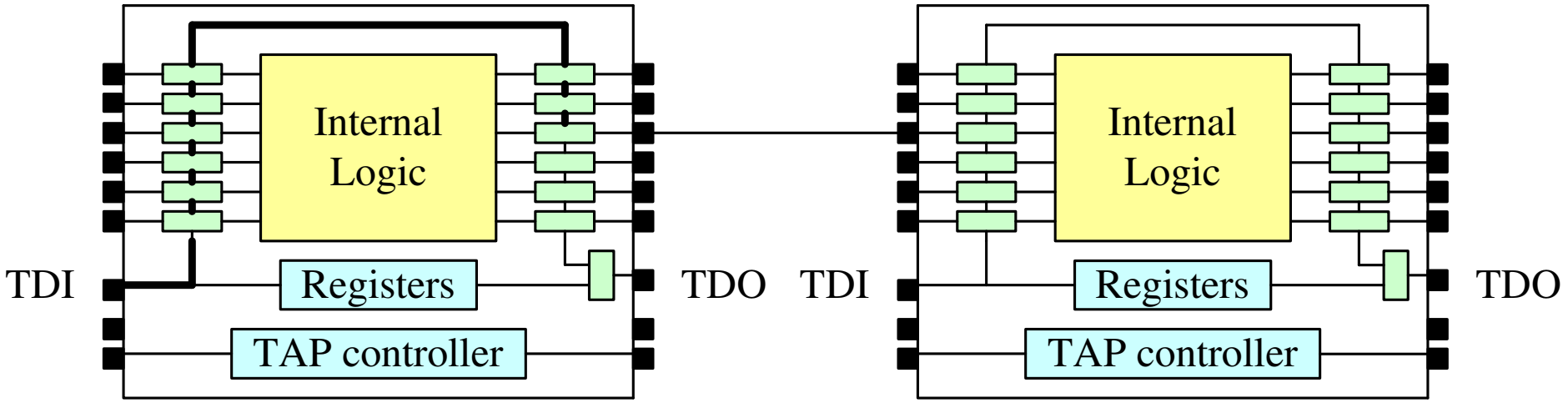


Execution of PRELOAD Instruction



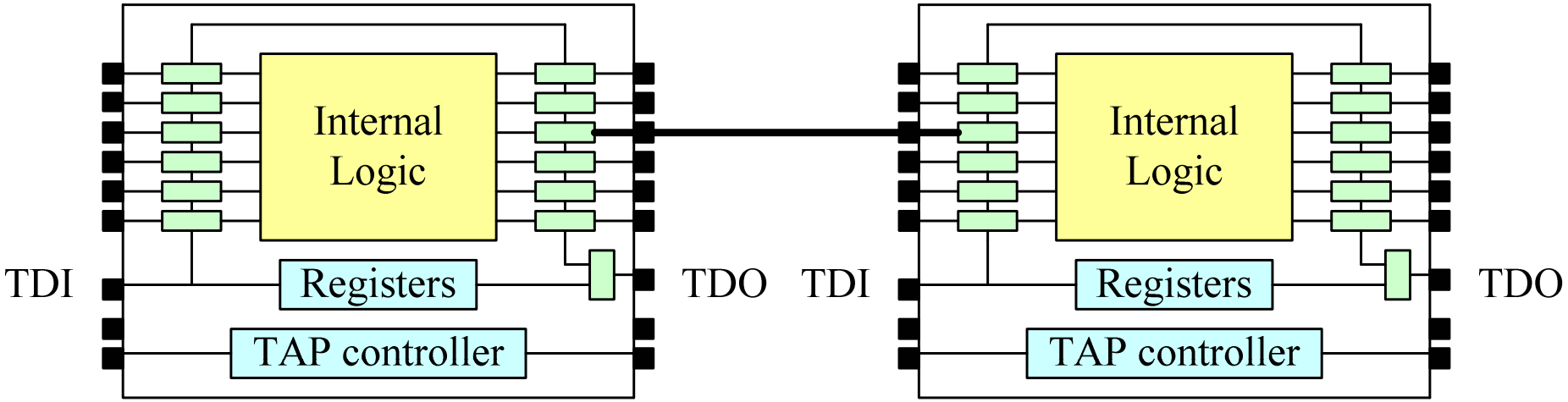
Execution of EXTEST Instruction (1/3)

- Shift-DR (Chip1)



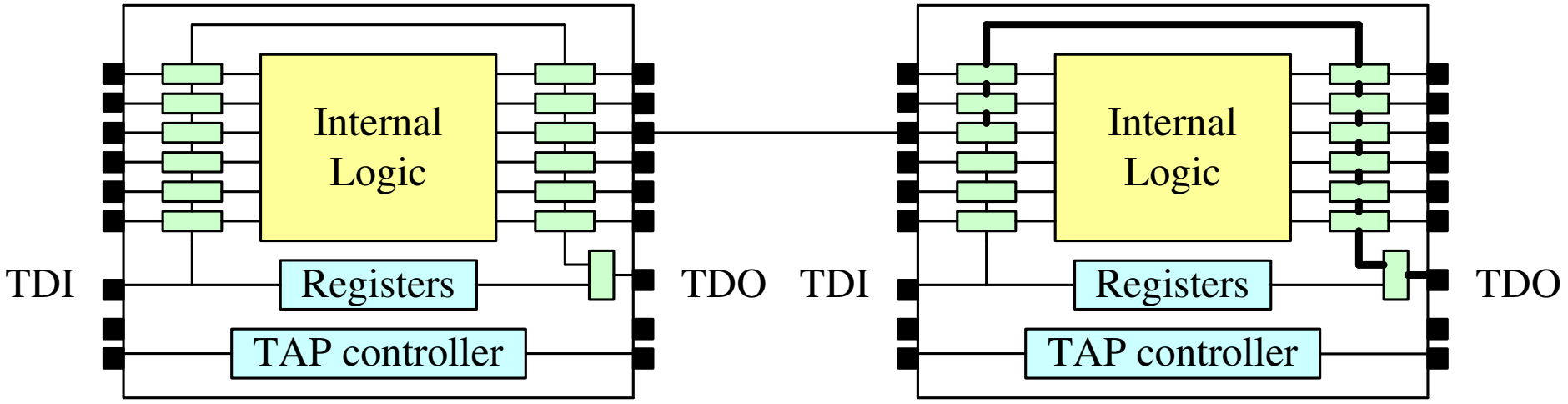
Execution of EXTEST Instruction (2/3)

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



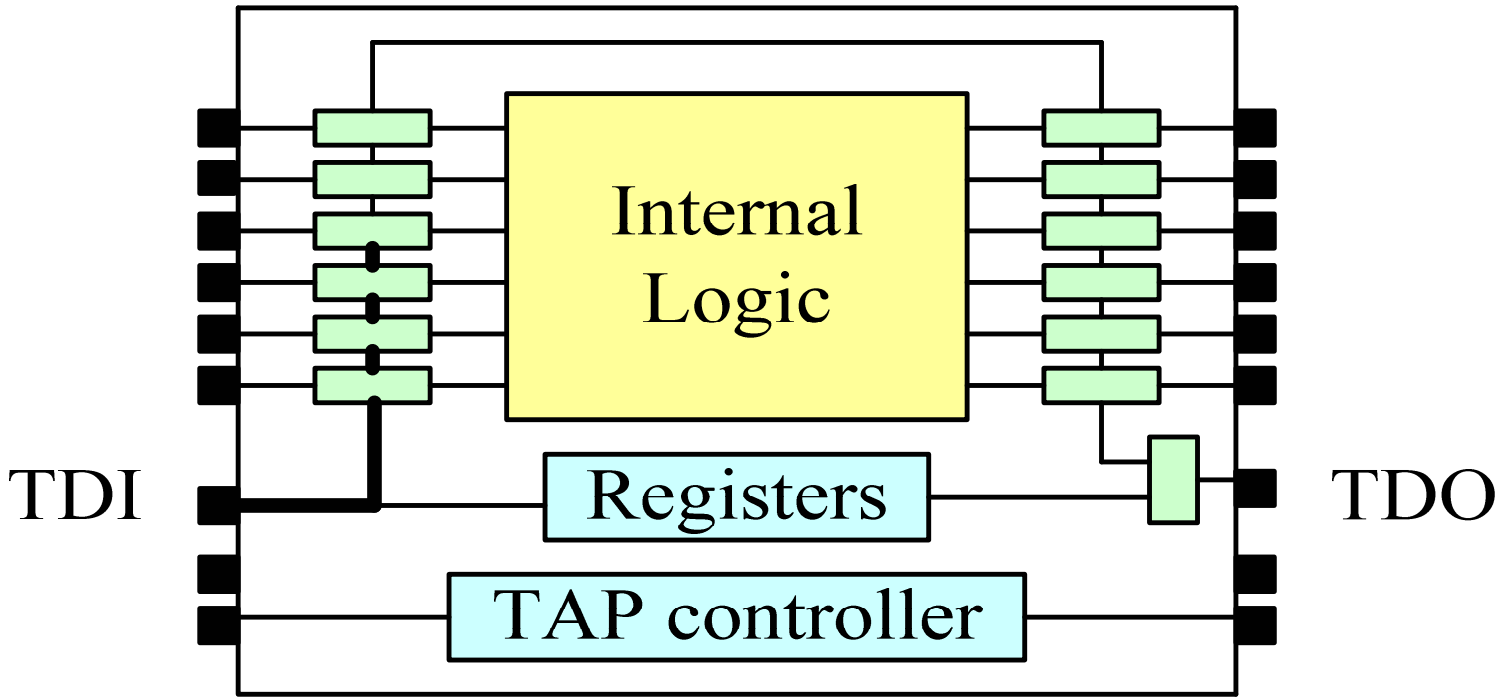
Execution of EXTEST Instruction (3/3)

- Shift-DR (Chip2)



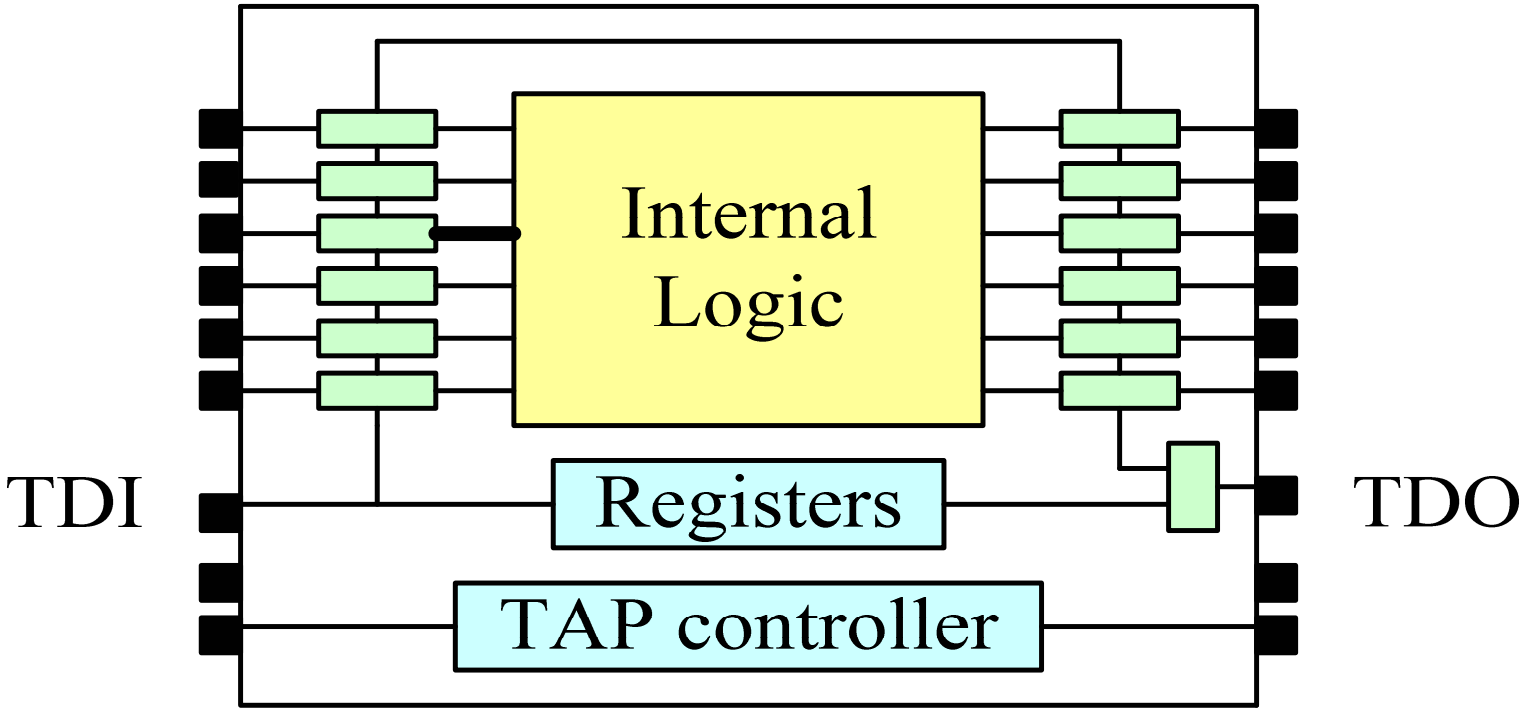
Execution of INTEST Instruction (1/4)

- Shift-DR



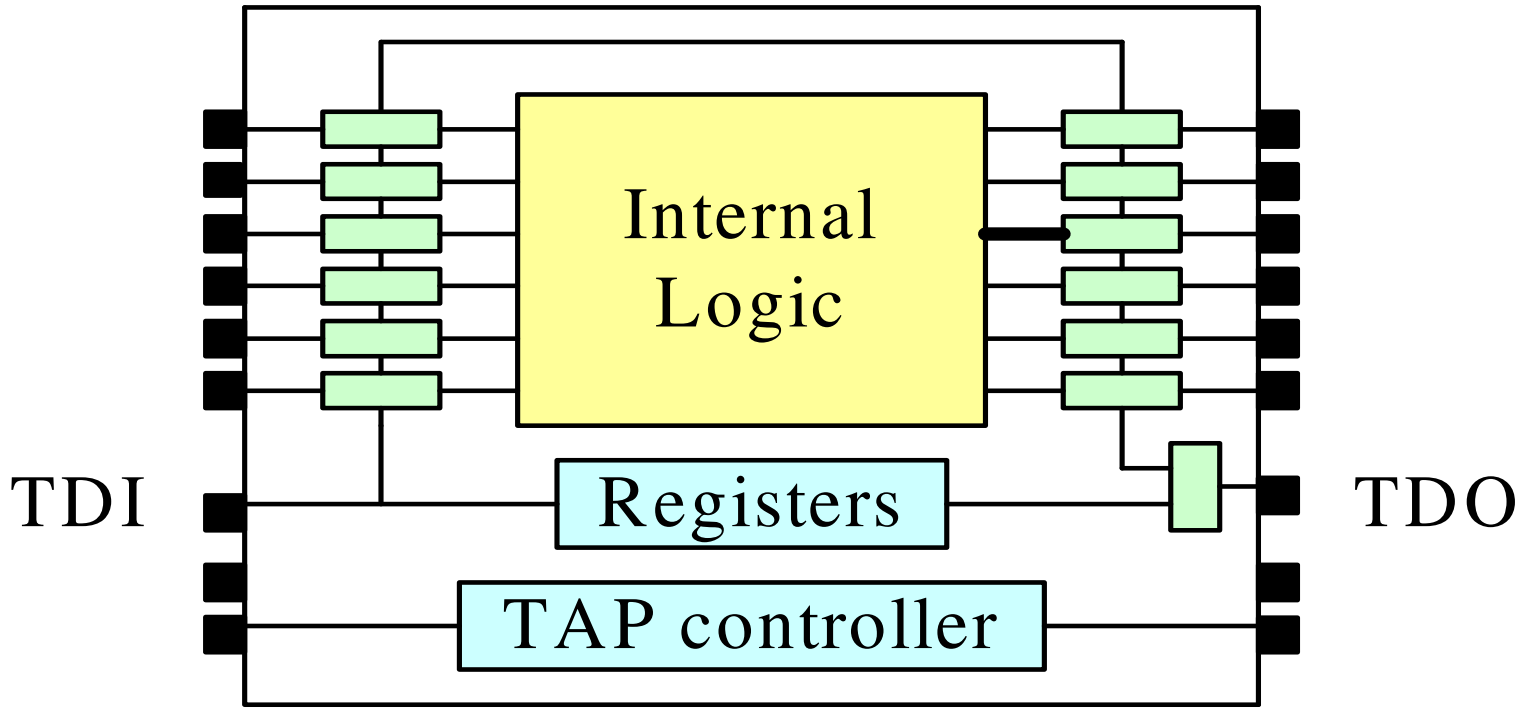
Execution of INTEST Instruction (2/4)

- Update-DR



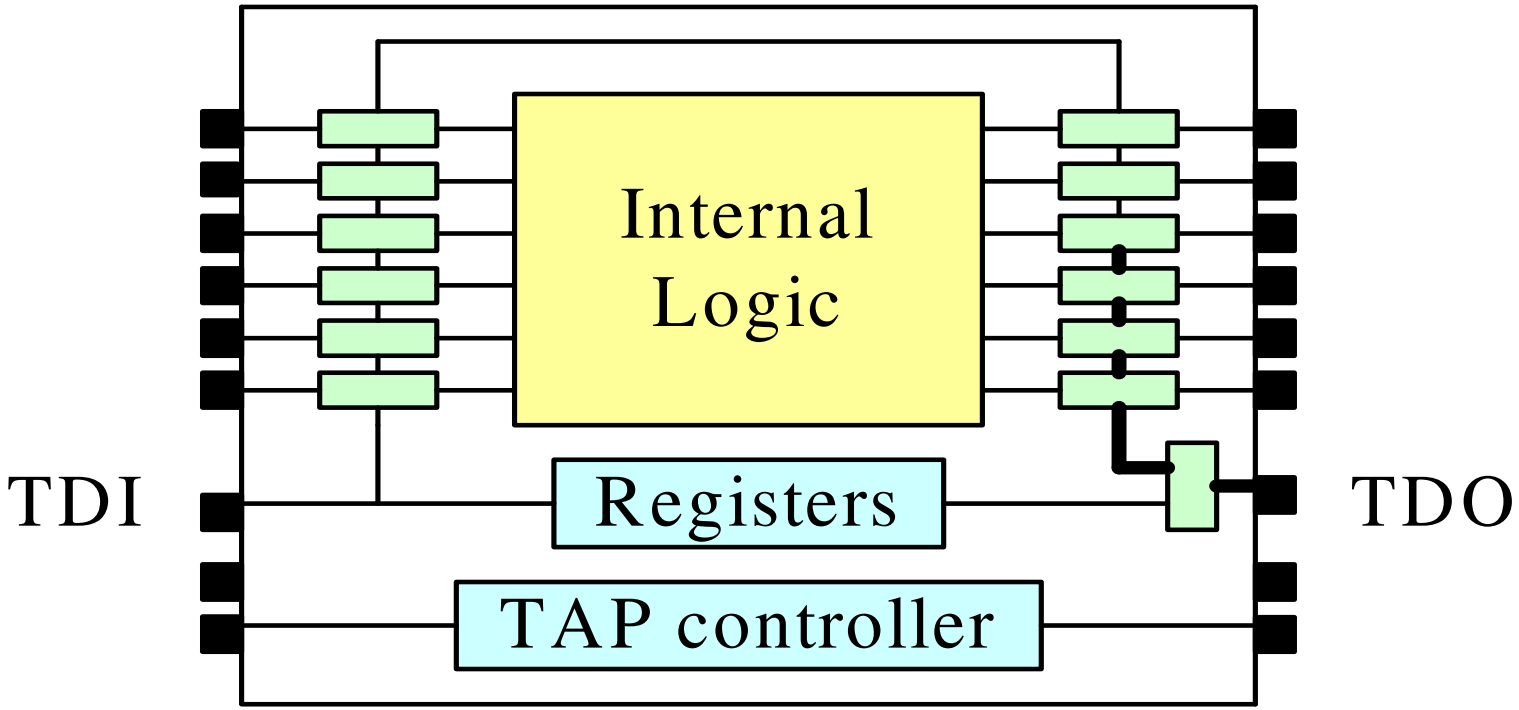
Execution of INTEST Instruction (3/4)

- Capture-DR



Execution of INTEST Instruction (4/4)

- Shift-DR



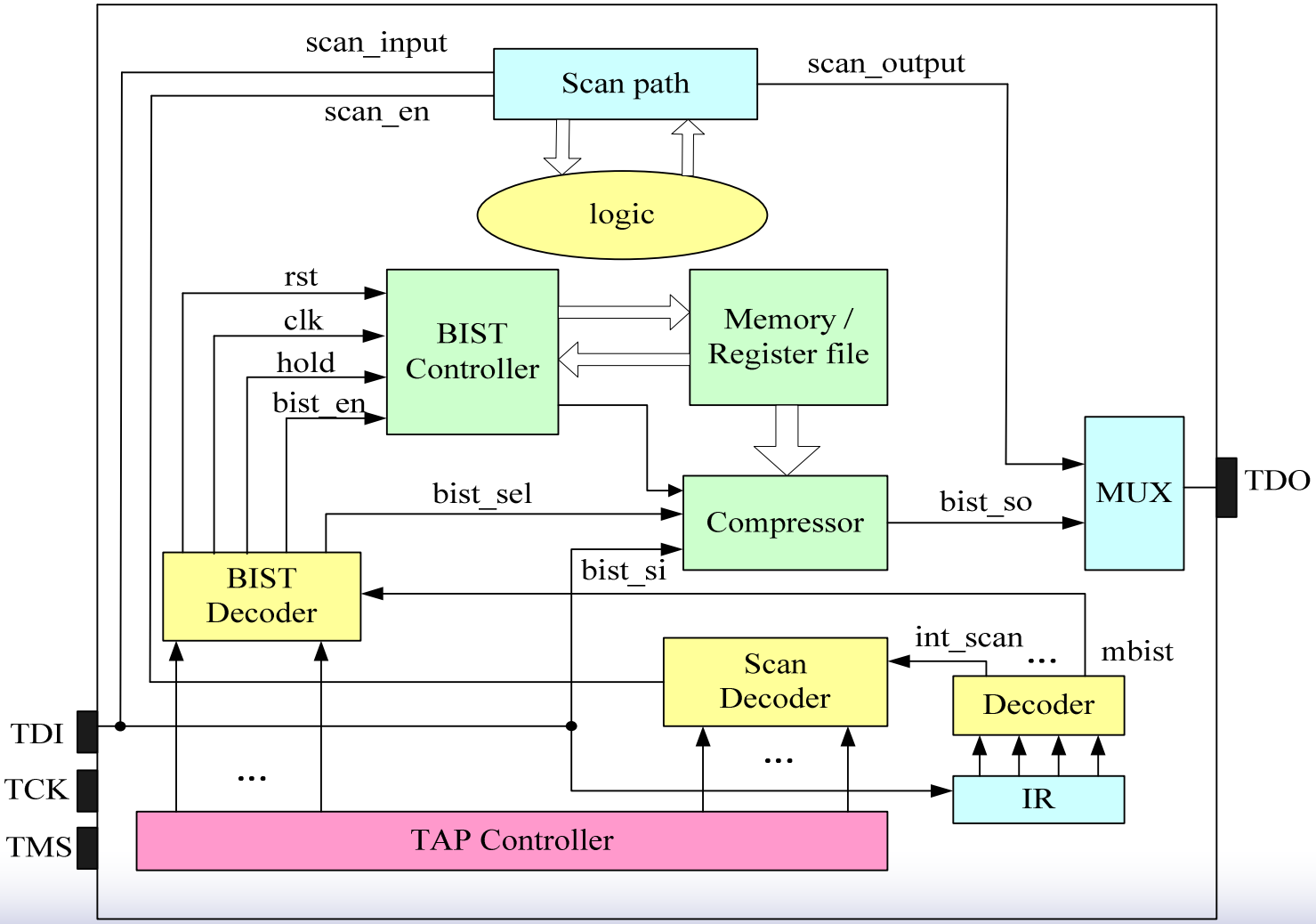
Boundary Scan Description Language (BSDL)

- Now a part of IEEE 1149.1-2001
- Purposes:
  - Provide standard description language for BS devices.
  - Simplify design work for BS – automated synthesis is possible.
  - Promote consistency throughout ASIC designers, device manufacturers, foundries, test developers and ATE manufacturers.
  - Make it easy to incorporation BS into software tools for test generation, analysis and failure diagnosis.
  - Reduce possibility of human error when employing boundary scan in a design.

Features of BSDL

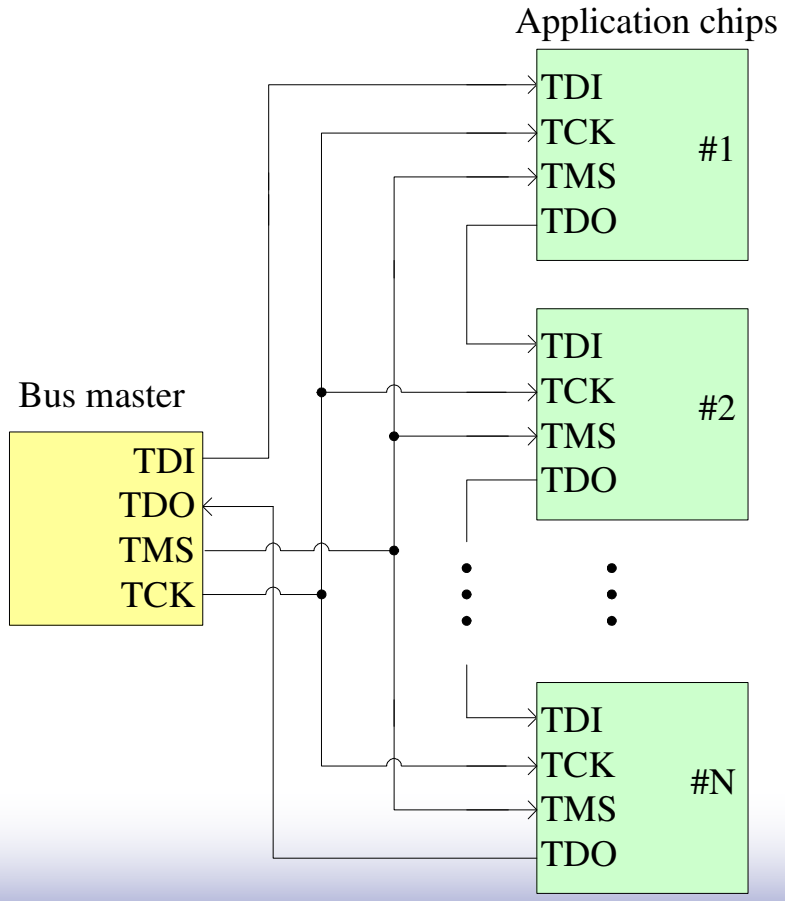
- Describes the testability features of BS devices that are compatible with 1149.1.
- S subset of VHDL.
- System-logic and the 1149.1 elements that are absolutely mandatory need not be specified.
  - Examples: BYPASS register, TAP controller, etc.
- Commercial tools to synthesize BSDL exist.

Scan and BIST Support with Boundary Scan



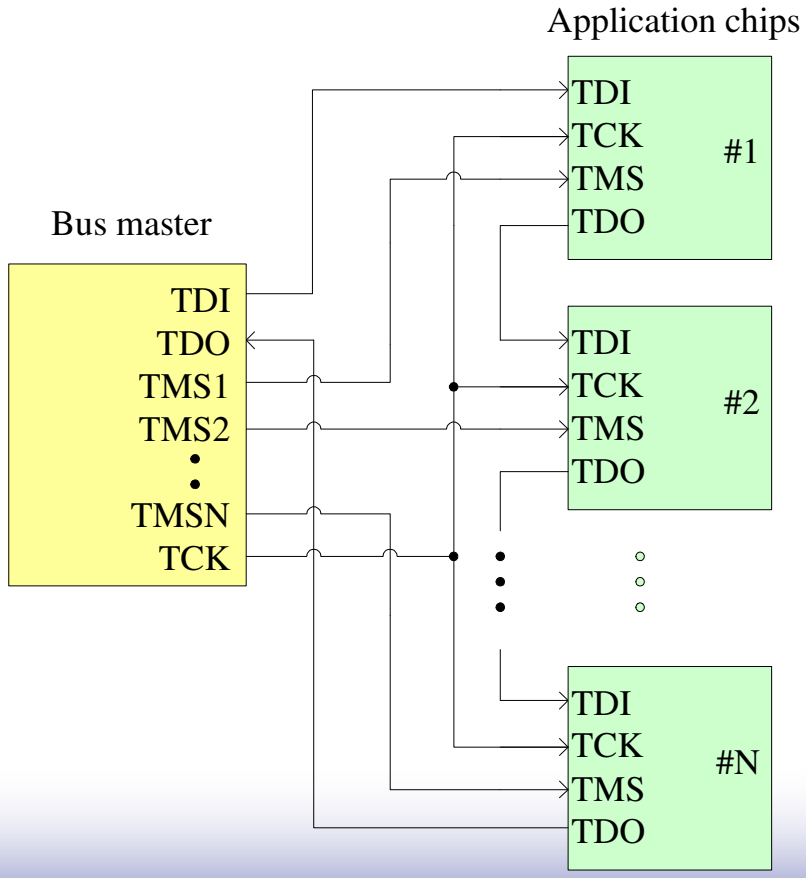
Bus Master for Chips with Boundary Scan (1/5)

- Ring architecture with shared TMS



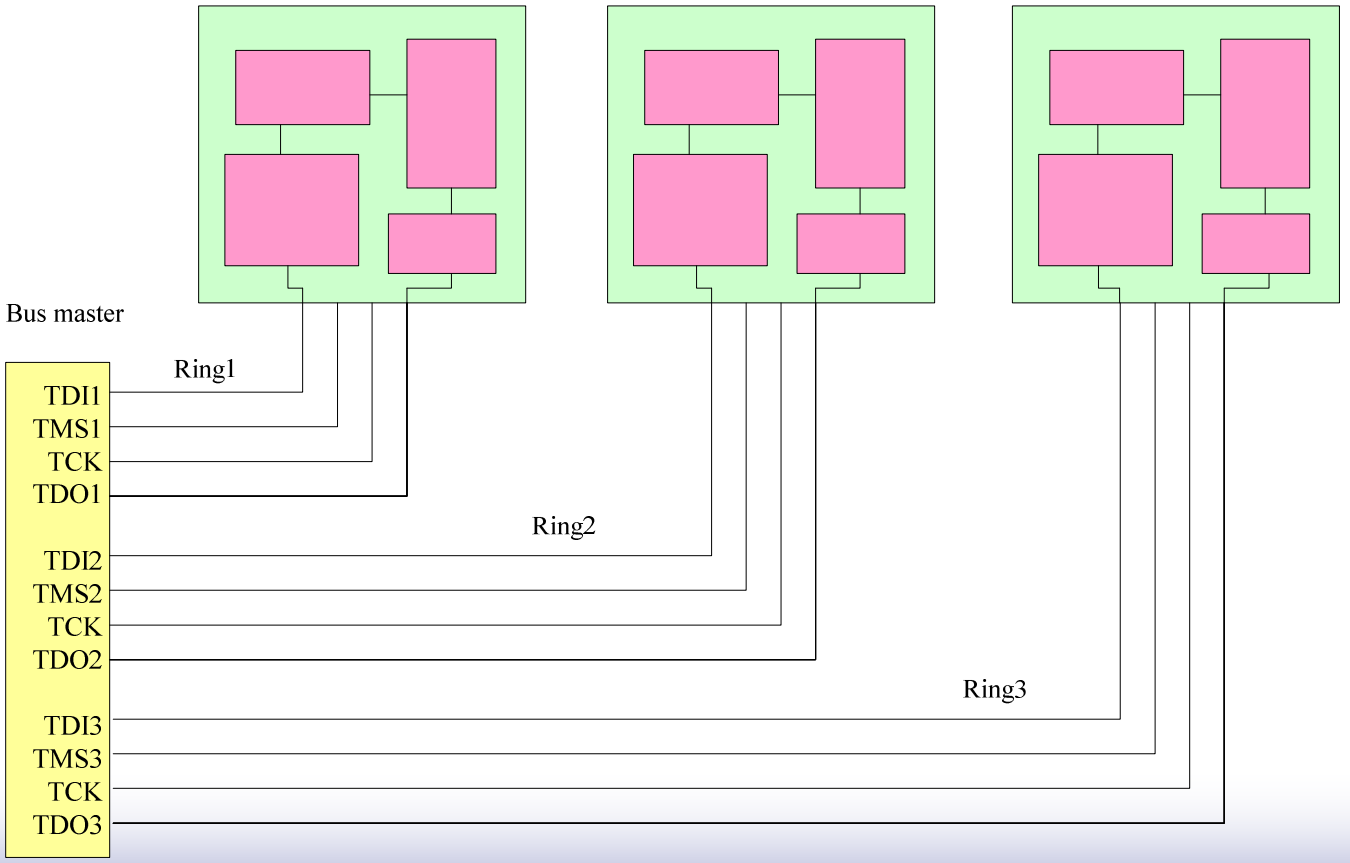
Bus Master for Chips with Boundary Scan (2/5)

- Ring architecture with separate TMS



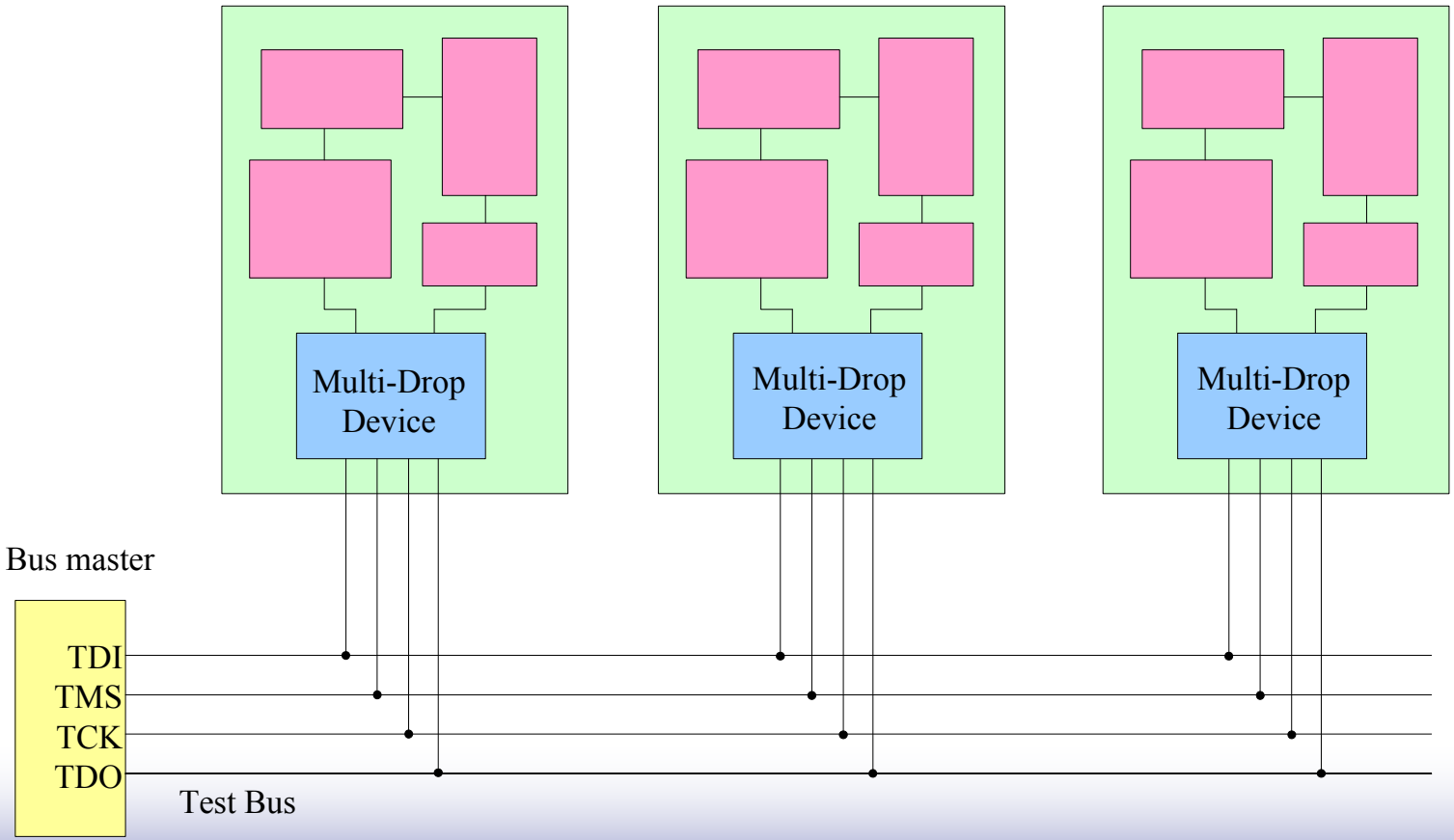
Bus Master for Chips with Boundary Scan (3/5)

- Star architecture



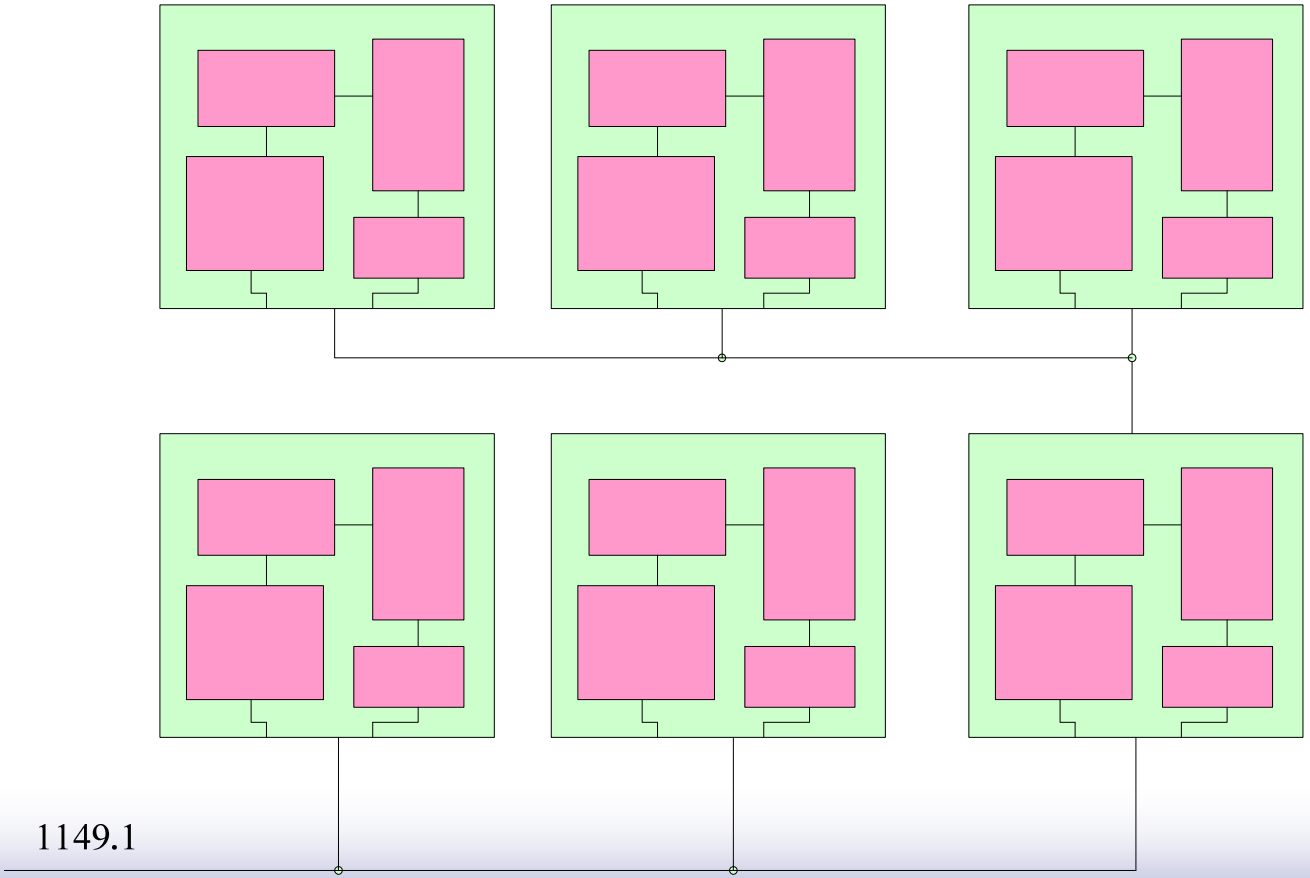
Bus Master for Chips with Boundary Scan (4/5)

- Multi-drop architecture



Bus Master for Chips with Boundary Scan (5/5)

- Hierarchical architecture



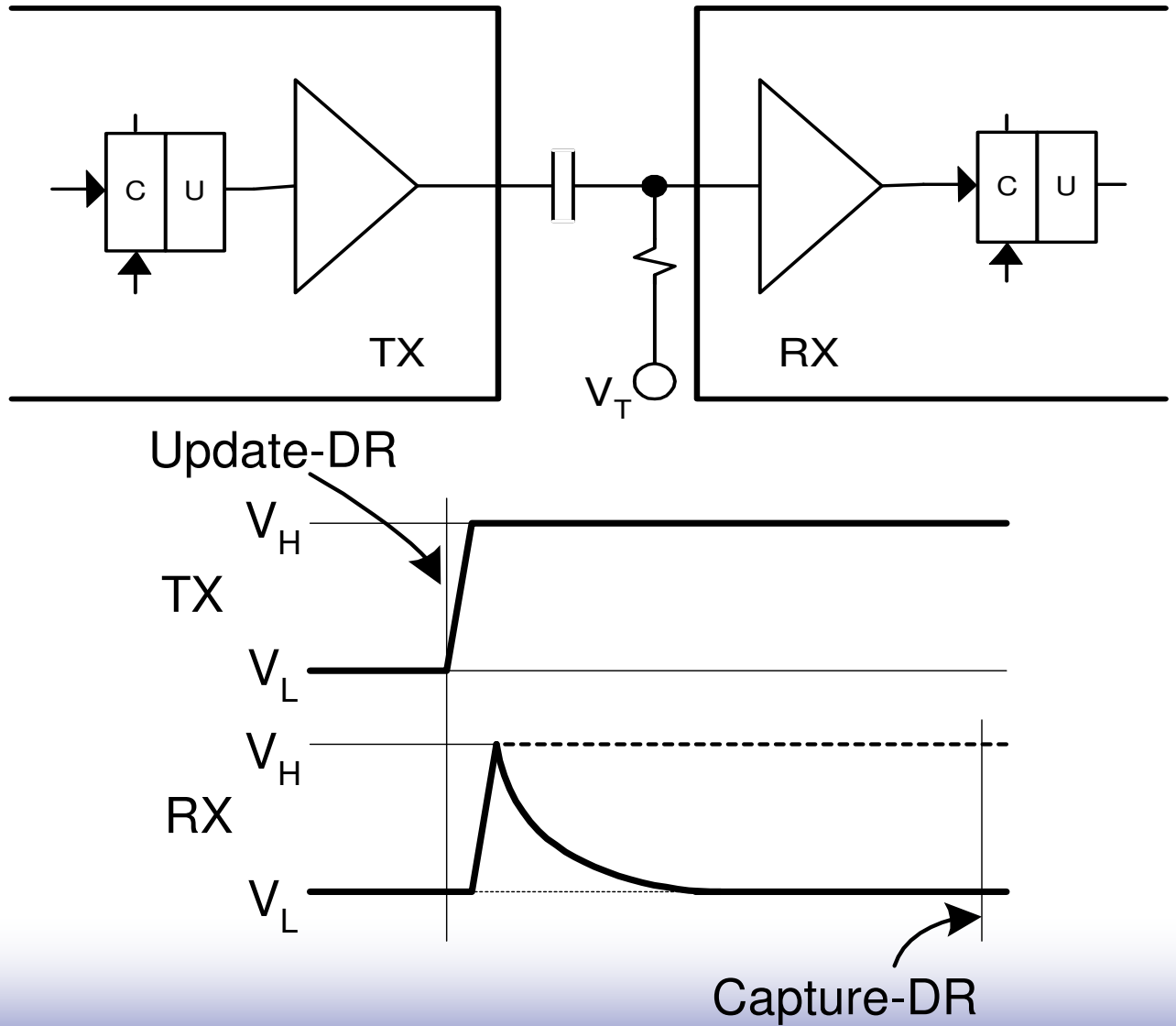
Boundary scan for advanced networks – 1149.6

- Rationale
- Analog test receiver
- Digital driver logic
- Digital receiver logic
- Test access port for 1149.6

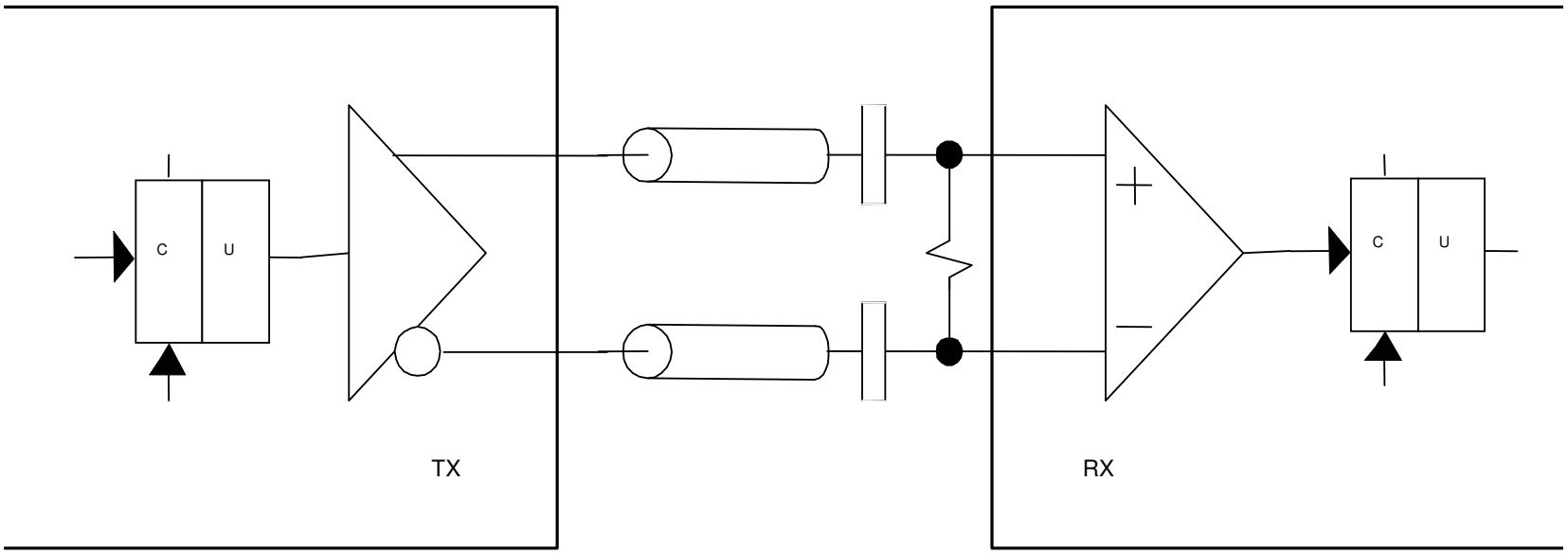
Rationale

- Advanced signaling techniques are required for multiple-mega-per-second I/O.
  - Differential or AC-coupling networks
- Coupling capacitor in AC-coupled networks blocks DC signals.
- DC-level applied during EXTEST may decay to undefined logic level.

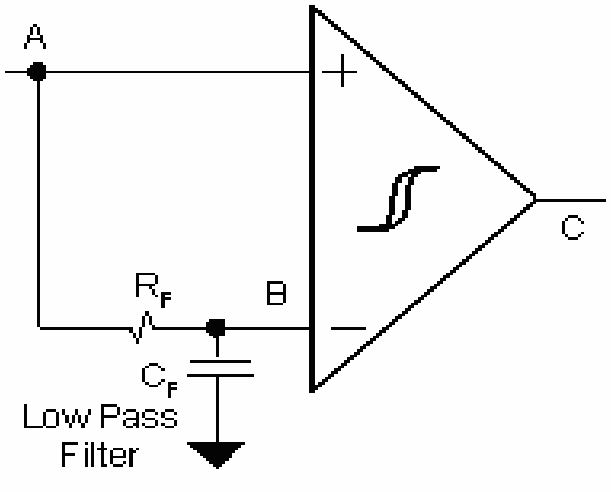
Capturing AC-Coupled Signal with 1149.1

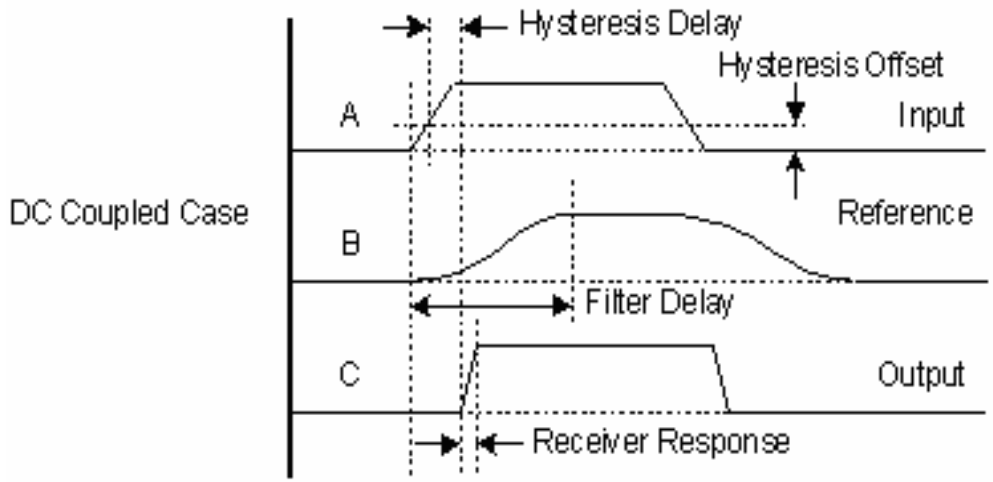


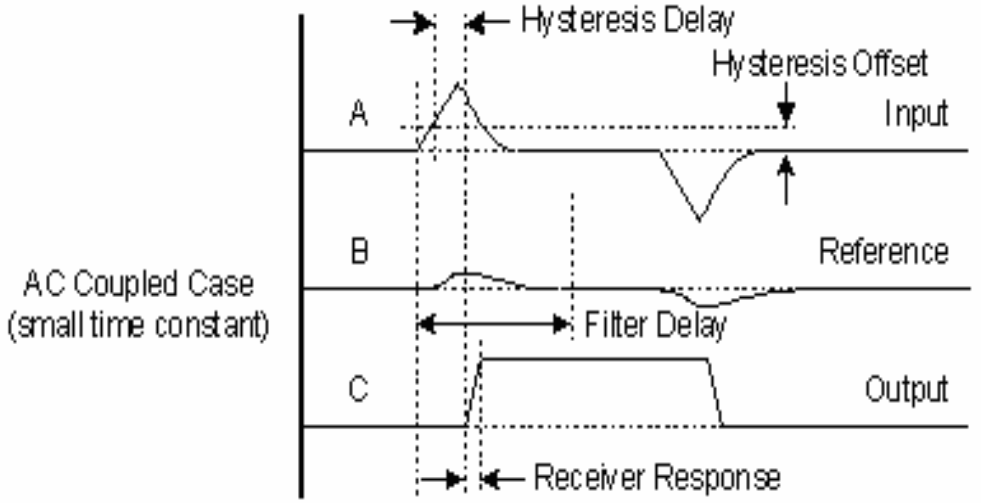
1149.1 Configuration for Differential Signaling



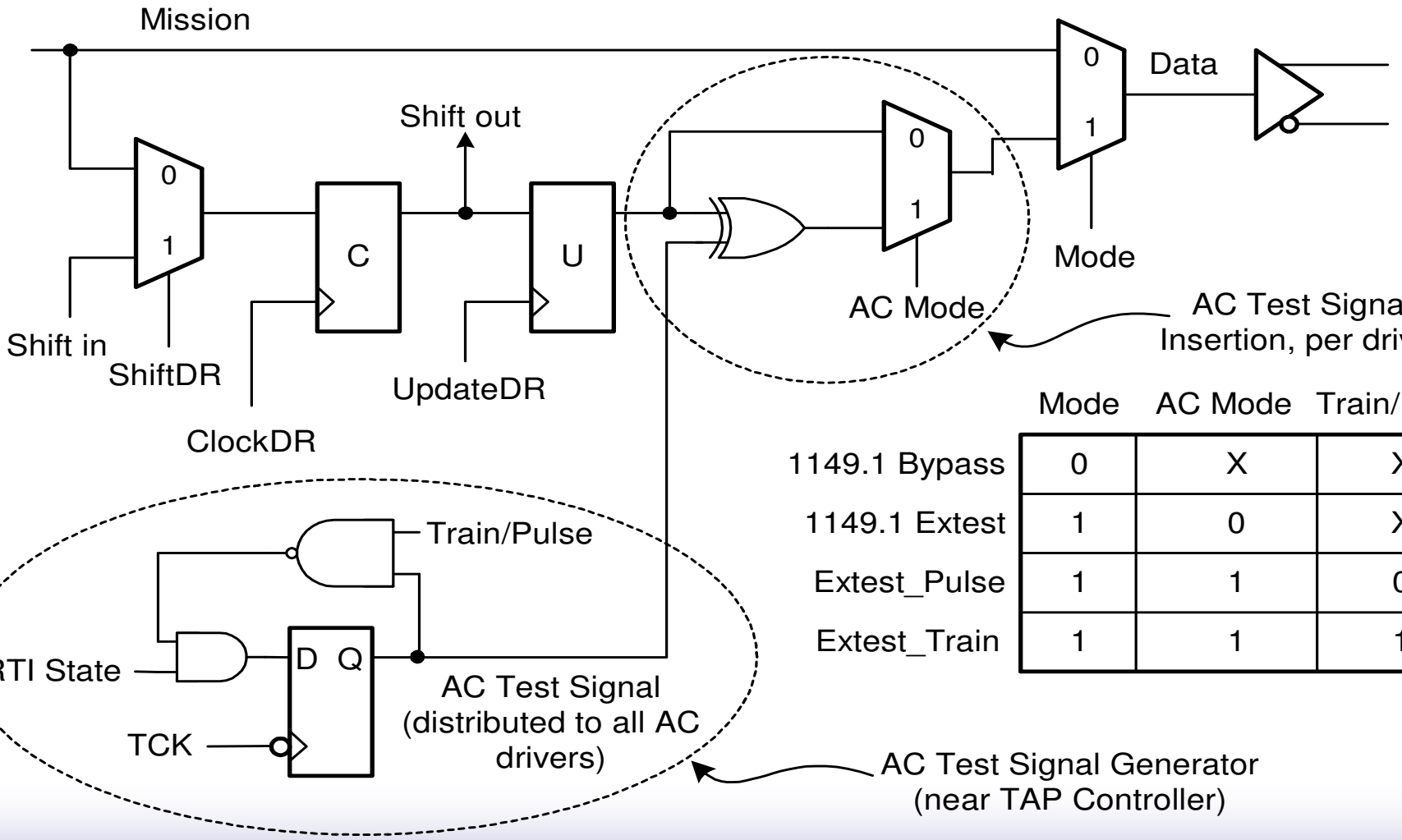
Analog Test Receiver Response to AC and DC-Coupled Signals





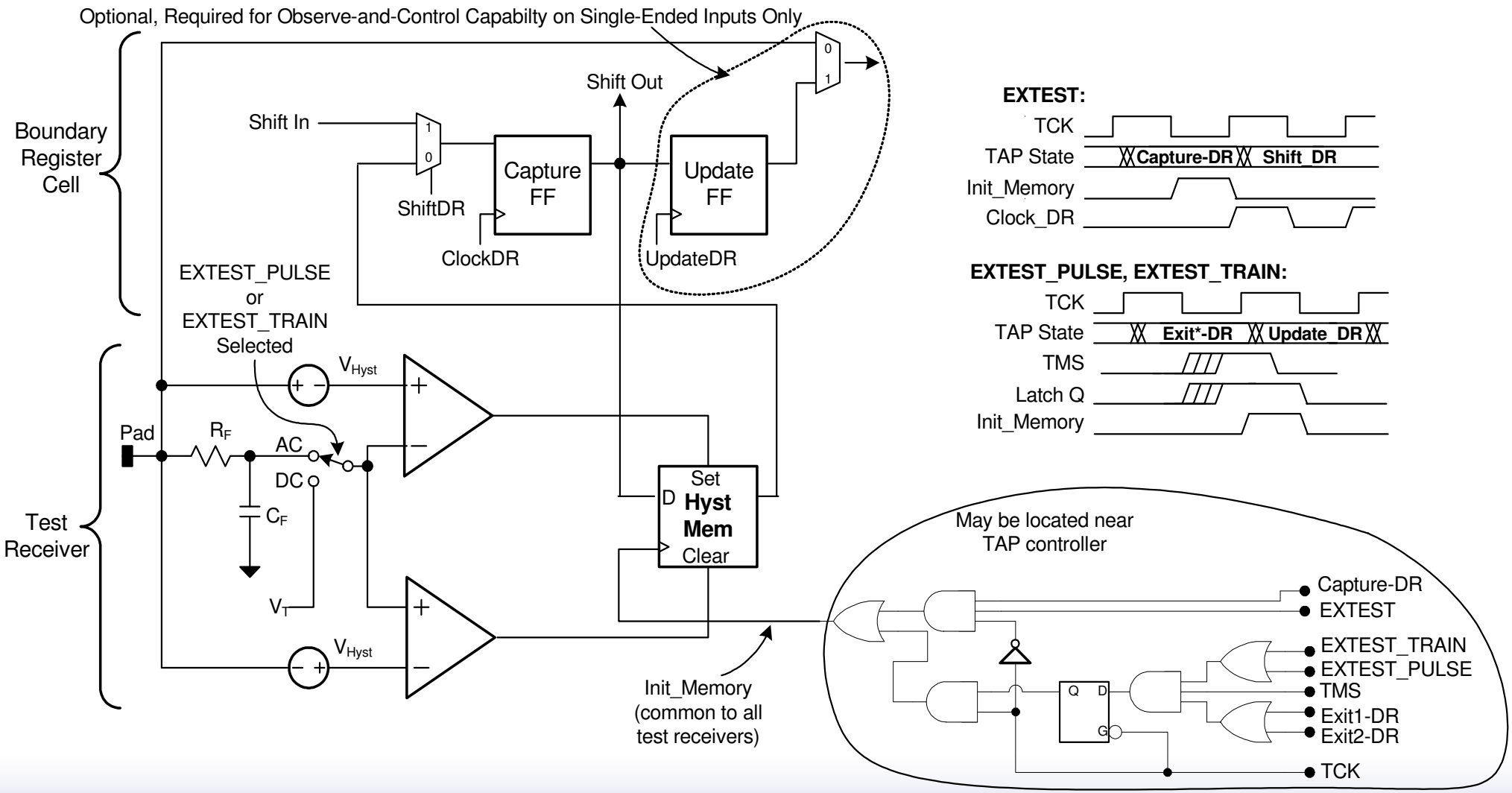


Digital Driver Logic



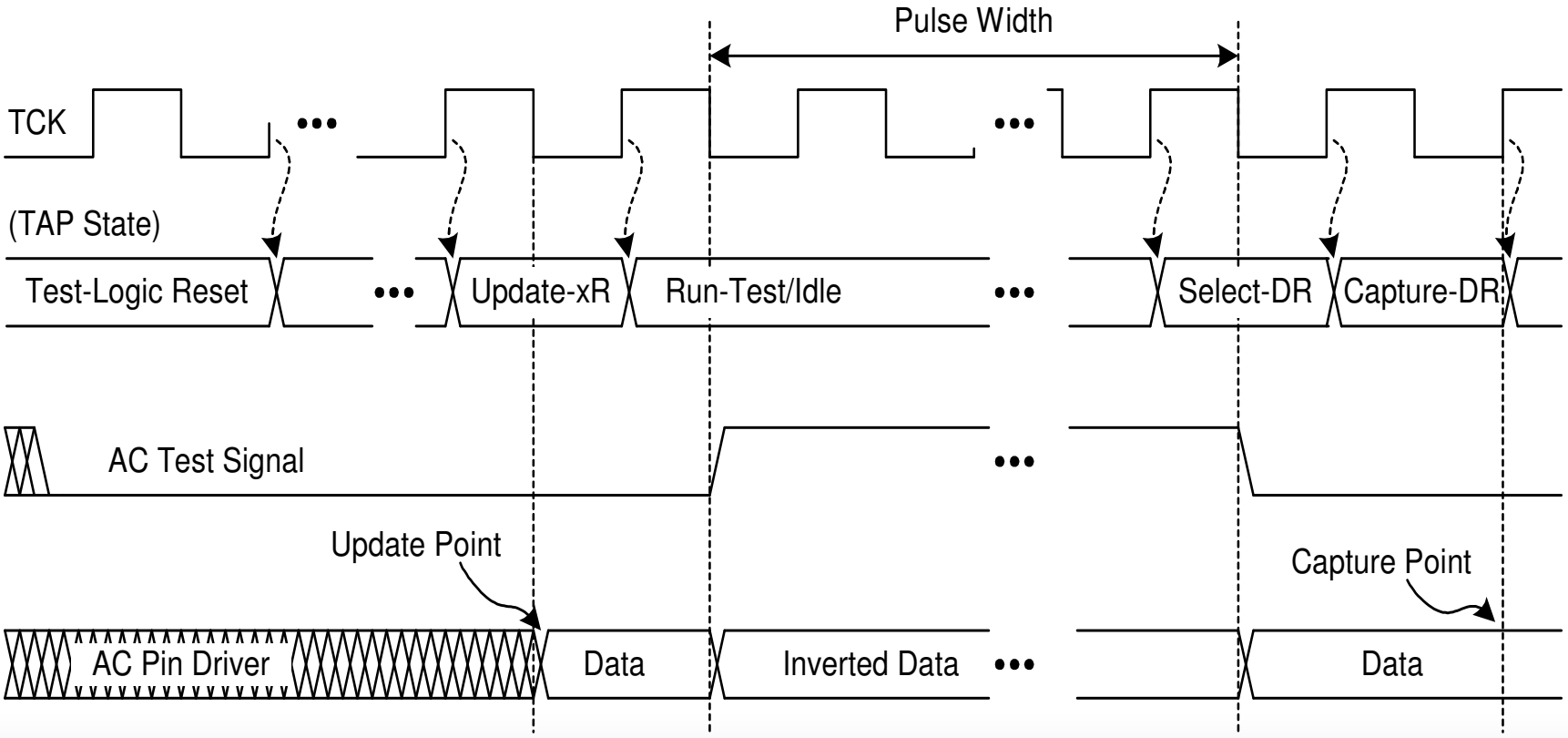
	Mode	AC Mode	Train/Pulse
1149.1 Bypass	0	X	X
1149.1 Extest	1	0	X
Extest_Pulse	1	1	0
Extest_Train	1	1	1

Digital Receiver Logic



NOTE: The generated clock (Init_Memory) shown is suitable for rising edge-sensitive behavior only.

Example of Modification on 1149.1 TAP – Driver Behavior During EXTEST_PULSE



Embedded Core Test Standard - 1500

- SOC test problems
- Overall architecture
- Wrapper components and functions
- Instruction set
- Core test language
- Core test supporting and system test configurations
- Hierarchical test control and plug & play

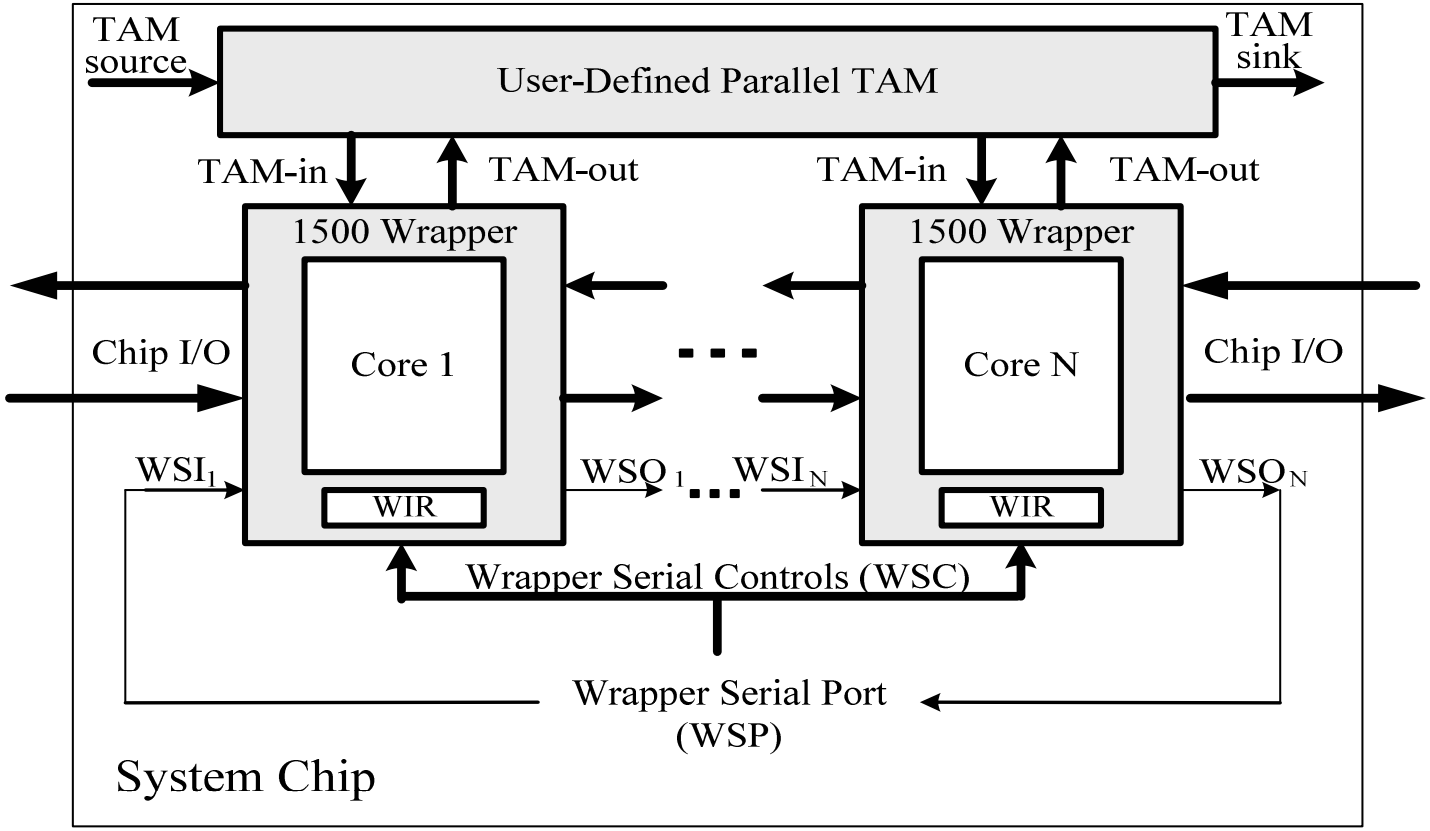
SOC Test Problems/Requirements (1/2)

- Mixing technologies: logic, processor, memory, analog
  - Need various DFT/BIST/other techniques
- Deeply embedded cores
  - Need Test Access Mechanism
- Hierarchical core reuse
  - Need hierarchical test management
- Different core providers and SOC test developers
  - Need standard for test integration
- IP protection/test reuse
  - Need core test standard/documentation

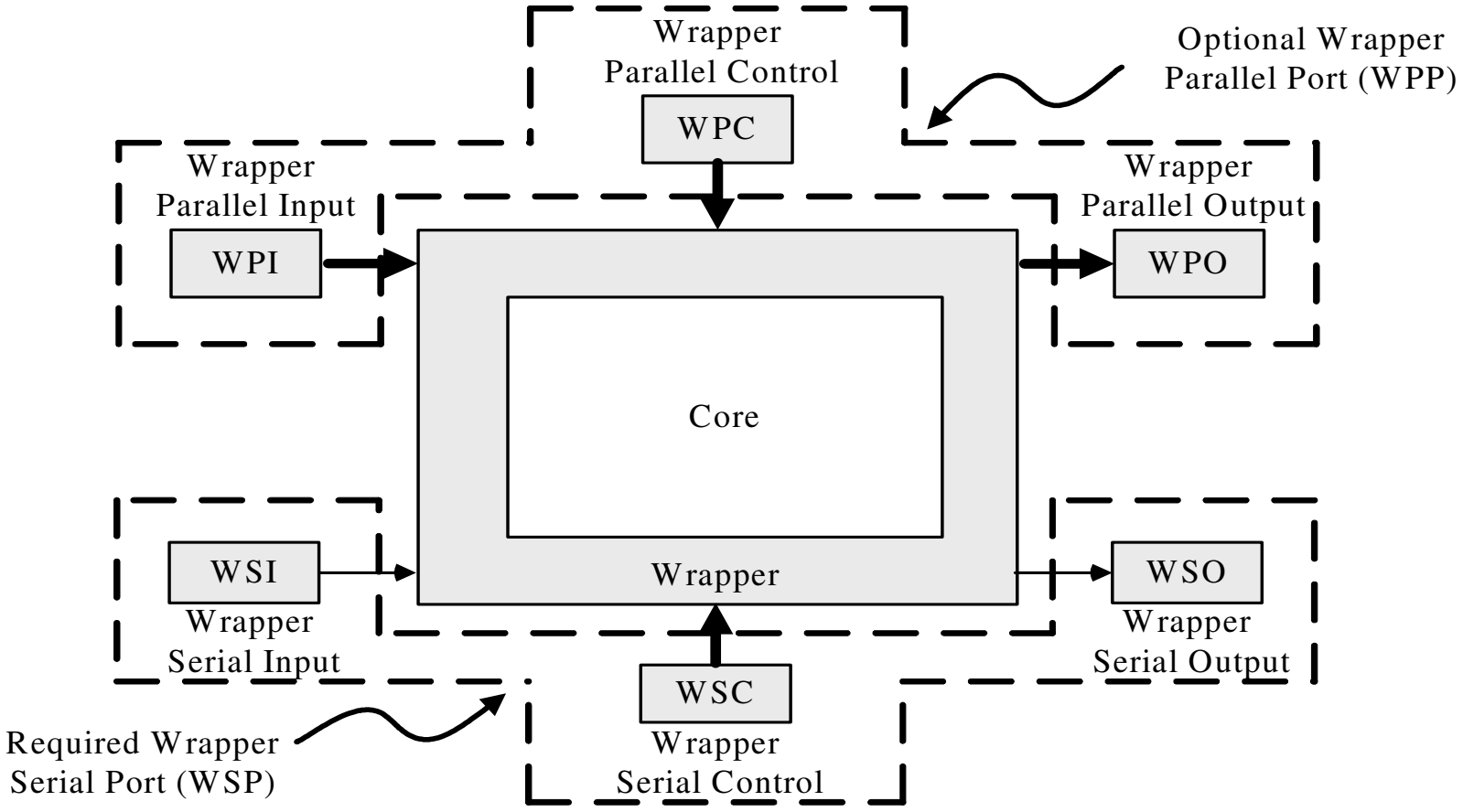
SOC Test Problems/Requirements (2/2)

- Higher-performance core pins than SOC pins
  - Need on-chip, at-speed testing
- External ATE inefficiency
  - Need “on-chip ATE”
- Long test application time
  - Need parallel testing or test scheduling
- Test power must be considered
  - Need lower power design or test scheduling
- Testable design automation
  - Need new testable design tools and flow

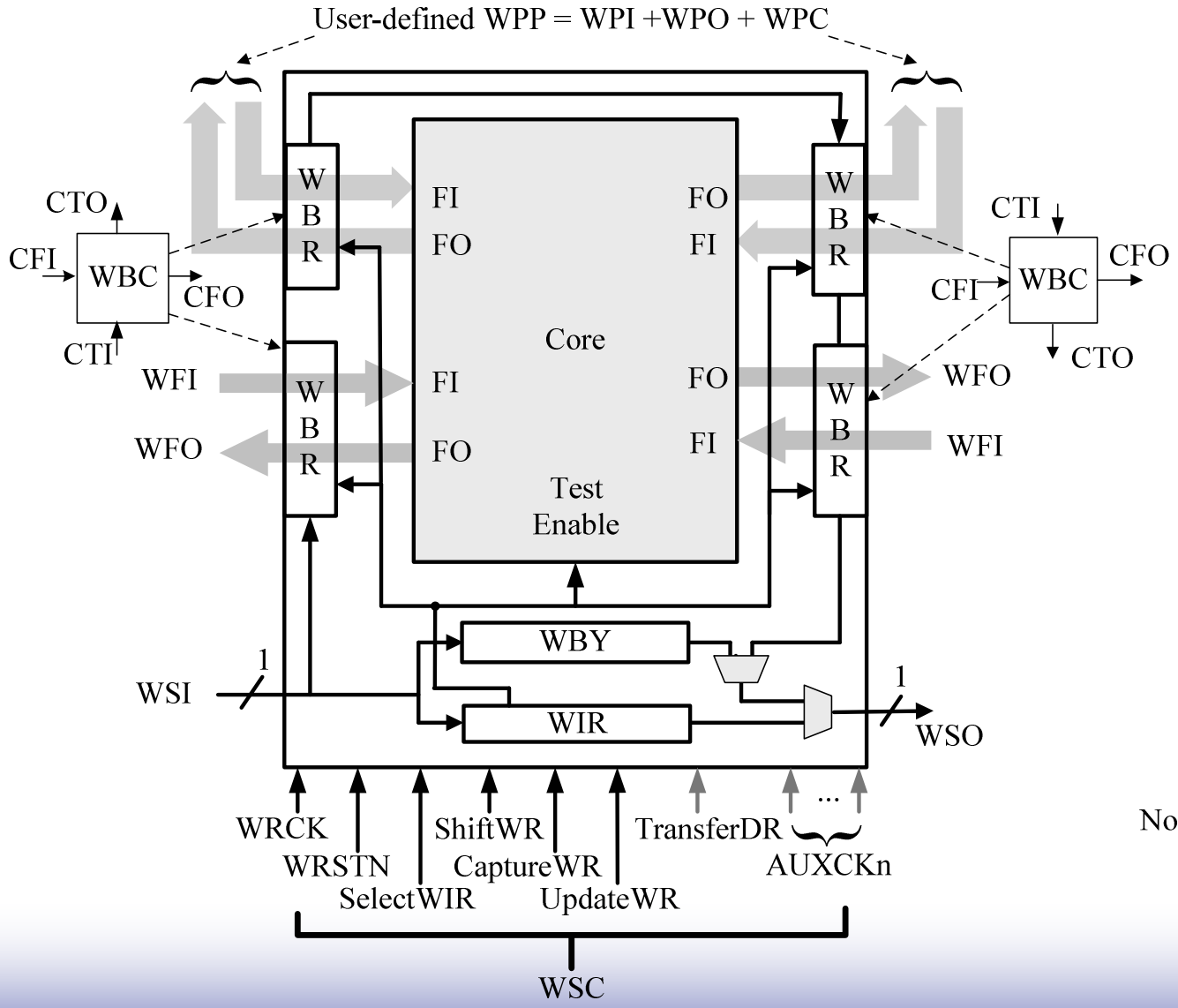
A System Overview of IEEE 1500 Standard



Test Interface of A Core Wrapper



Serial Test Circuitry of 1500 for a Core



CTI: Cell Test Input
CTO: Cell Test output
CFI: Cell Functional Input
CFO: Cell Functional Output
FI: Functional Input
FO: Functional Output
WFI: Wrapper Functional Input
WFO: Wrapper Functional Output
WBC: Wrapper Boundary Cell
WBR: Wrapper Boundary Register
WBY: Wrapper Bypass Register
WIR: Wrapper Instruction Register
WPP: Wrapper Parallel Port
WSC: Wrapper Serial Control
WSI: Wrapper Serial Input
WSO: Wrapper Serial Output
WSP: Wrapper Series Port

Note: WSP = WSI+WSO+WSC

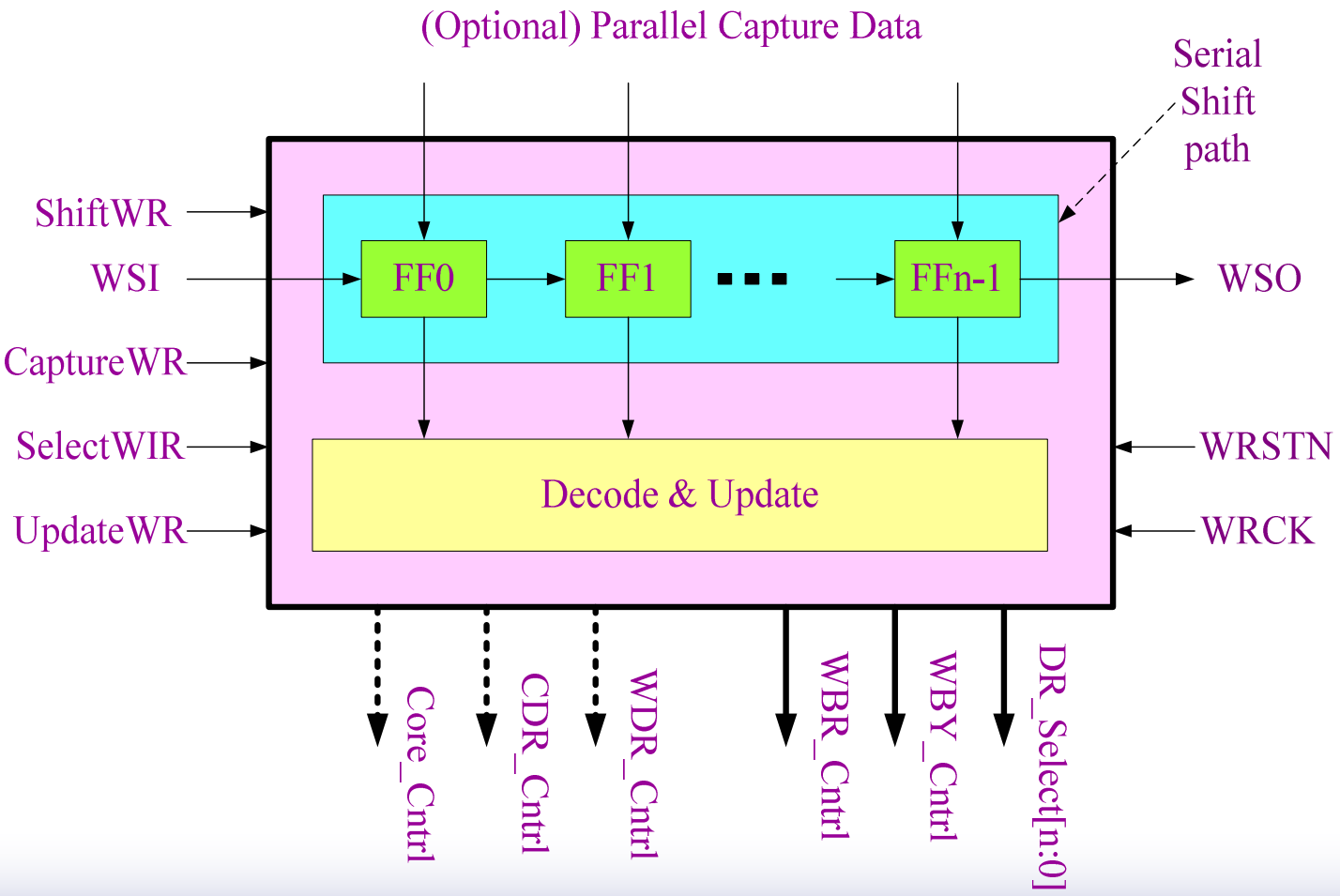
Wrapper Components

- Wrapper series port (WSP)
  - Wrapper series input (WSI), Wrapper series output (WSO), Wrapper series control (WSC)
- Wrapper parallel port (WPP) (optional)
  - Wrapper parallel input (WPI), Wrapper parallel output (WPO), wrapper parallel control (WPC)
- Wrapper instruction register (WIR)
- Wrapper bypass regiester (WBY)
- Wrapper data register (WBR)
  - Consists of wrapper boundary cells (WBC's)
- Core data register (CDR) (optional)

Wrapper Series Control (WSC) signals

- WRCK: wrapper clock terminal
- AUXCK*n*: Optional auxiliary clocks, where *n* is the number of the clocks.
- WRSTN: wrapper reset
- SelectWIR: determine whether WIR is selected
- CaptureWR: enable Capture operation
- ShiftWR: enable Shift operation
- UpdateWR: enable Update operation
- TransferDR: enable Transfer operation

Wrapper Instruction Register



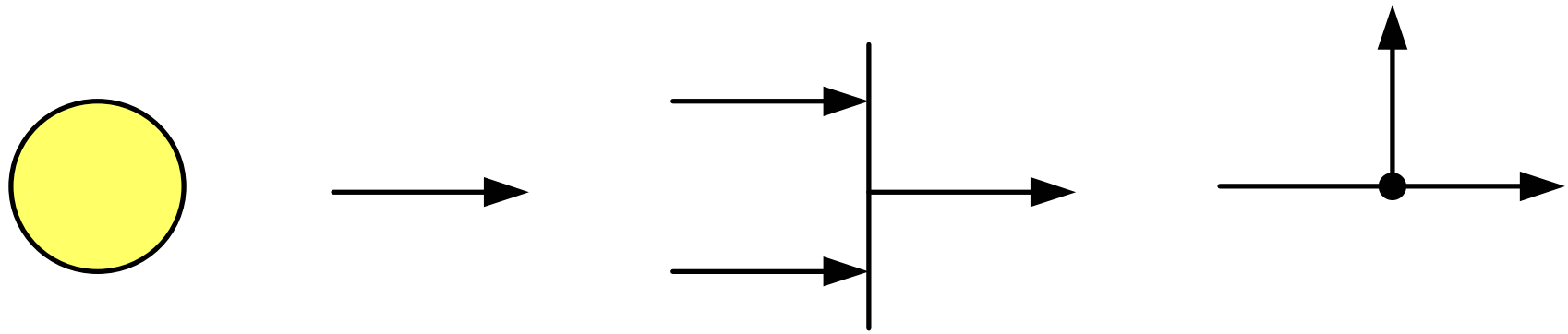
Wrapper Boundary Register (WBR)

- Consists of Wrapper boundary cells (WBC's)
- WBC
  - Terminals: Cell functional input (CFI), cell functional output (CFO), cell test input (CTI), cell test output (CTO)
  - Functional modes: Normal, inward facing, outward facing, nonhazardous (safe).
  - Operation events: Shift, capture, update, transfer, apply.

Events of WBR (WBC)

- Shift: data advance one-bit forward on WBR's shift path
- Capture: data on CFI or CFO are captured and stored in WBC
- Update: data stored in WBC's shift path storage are loaded into an off-shift-path storage of the WBC
- Transfer: move data to the storage closest to CTI or one bit closer to CTO
- Apply: a derivative event inferred from other events to apply data to functional inputs of cores or functional outputs of WBR

Four Symbols Used in Bubble Diagrams for WBC's



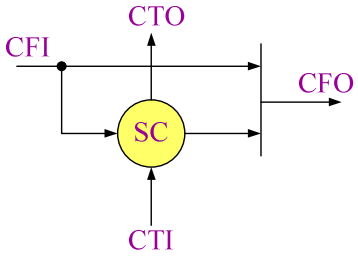
Storage element

Data path

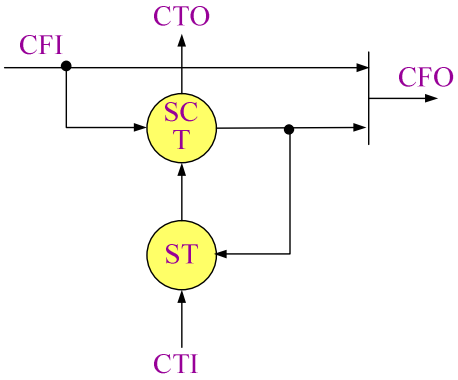
Decision point

Data paths from a source

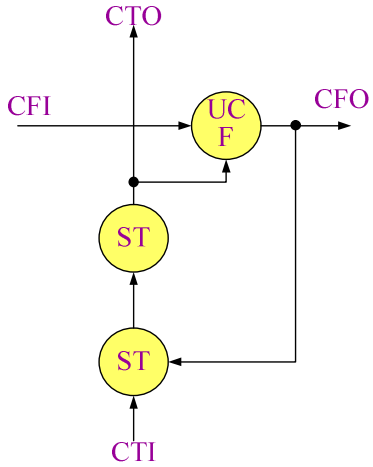
Some Typical WBC's Represented by Bubble Diagrams



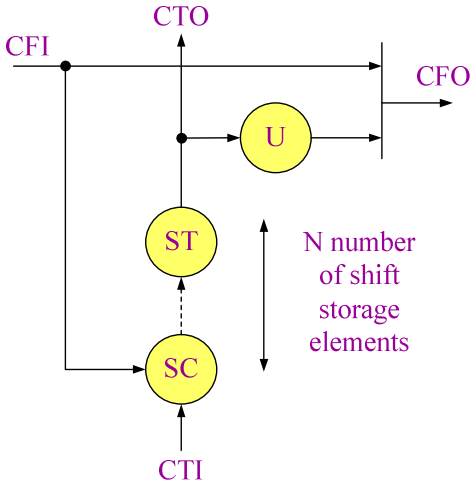
(a) WC_SD1_CII



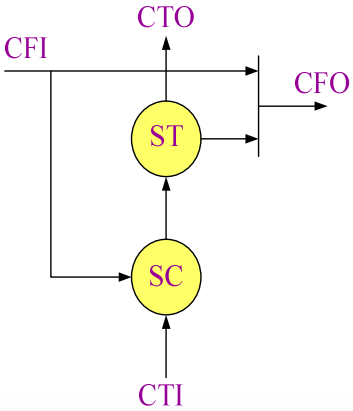
(b) WC_SD2_CIO



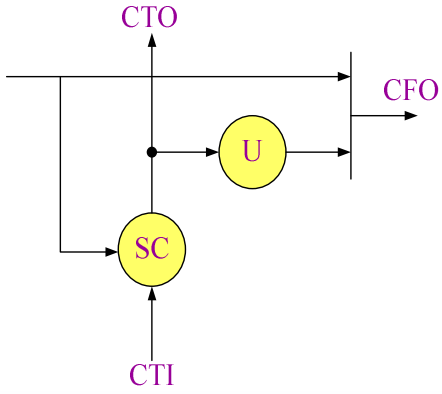
(e) WC_SD2_CIU_UF



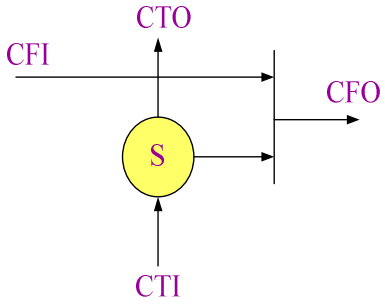
(f) WC_SDn_CII_UD



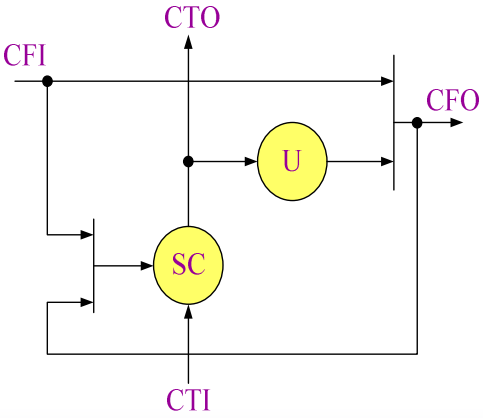
(c) WC_SD2_CII



(d) WC_SD1_CII_UD

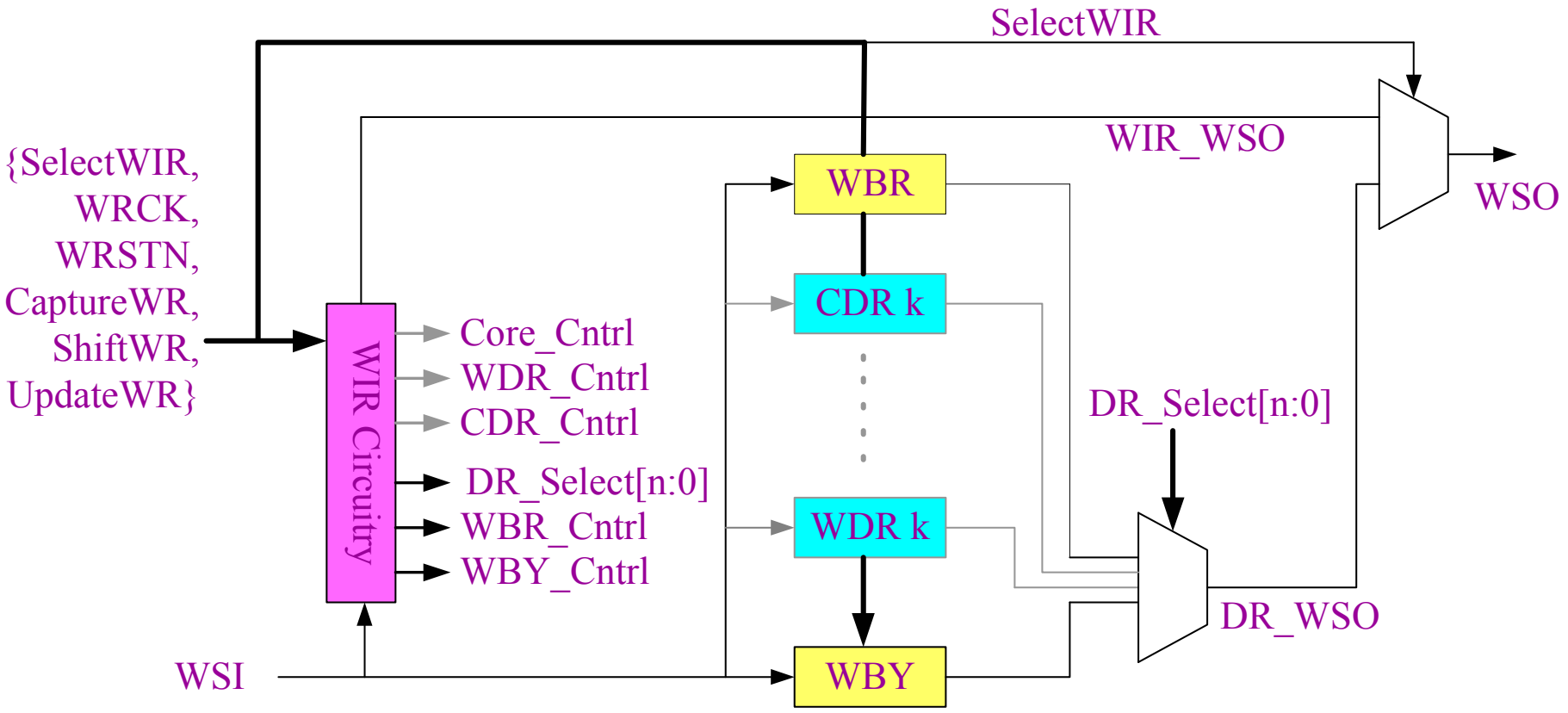


(g) WC_SD1_CN

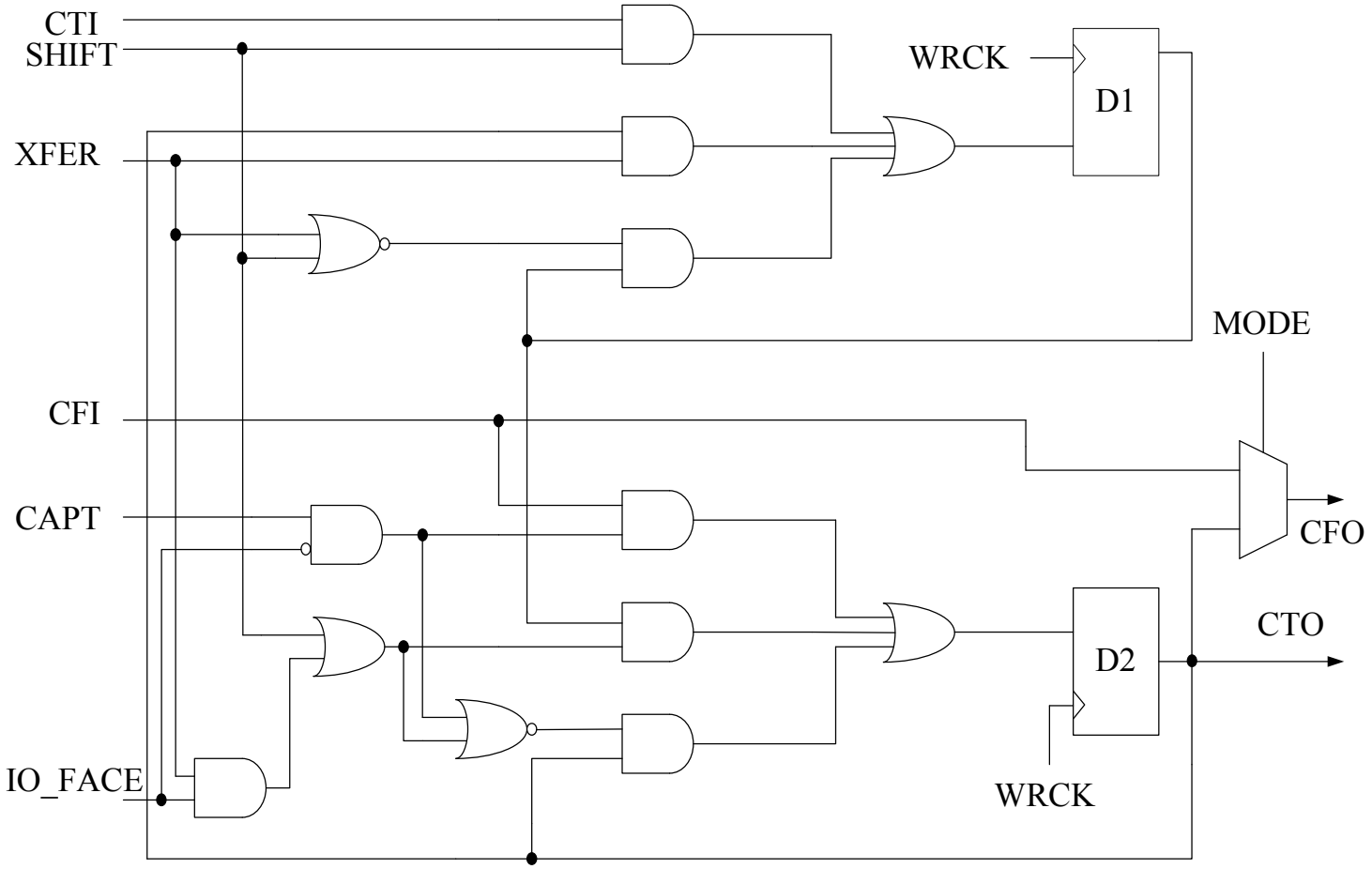


(h) WC_SD1_CBI_UD

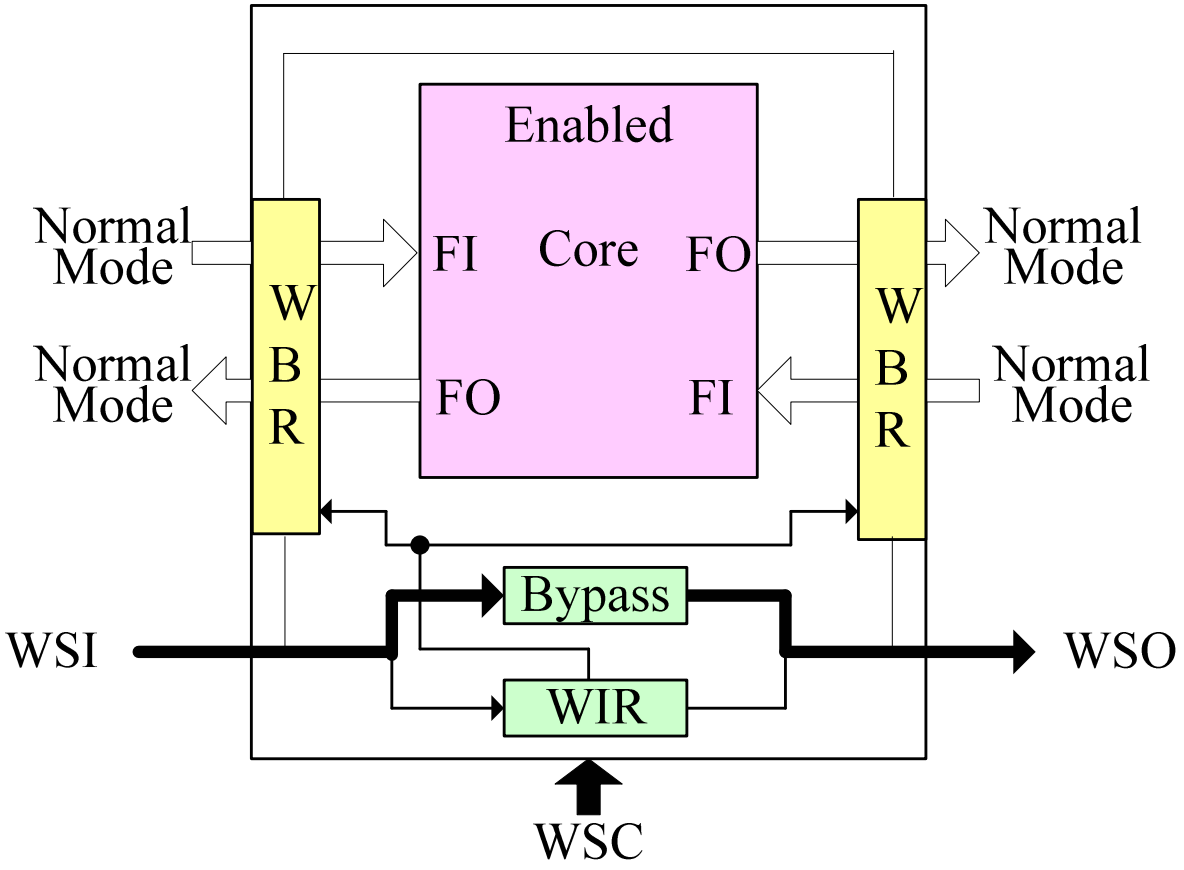
Example 10.1 - WIR Interface of WBY, WBR WDR(s) and CDR(s)



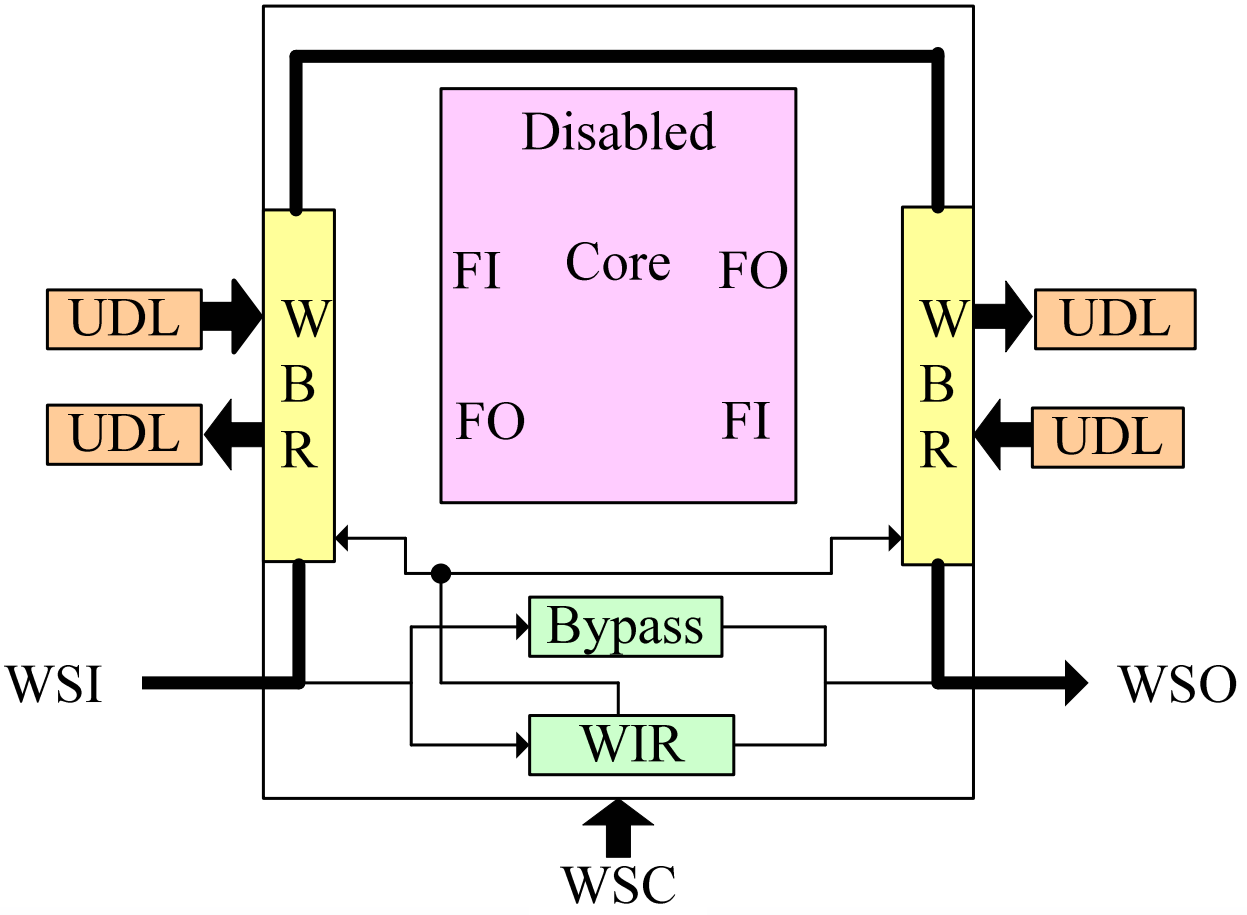
Example 10.2 - Schematic Diagram of WBC WC_SD2_CIO



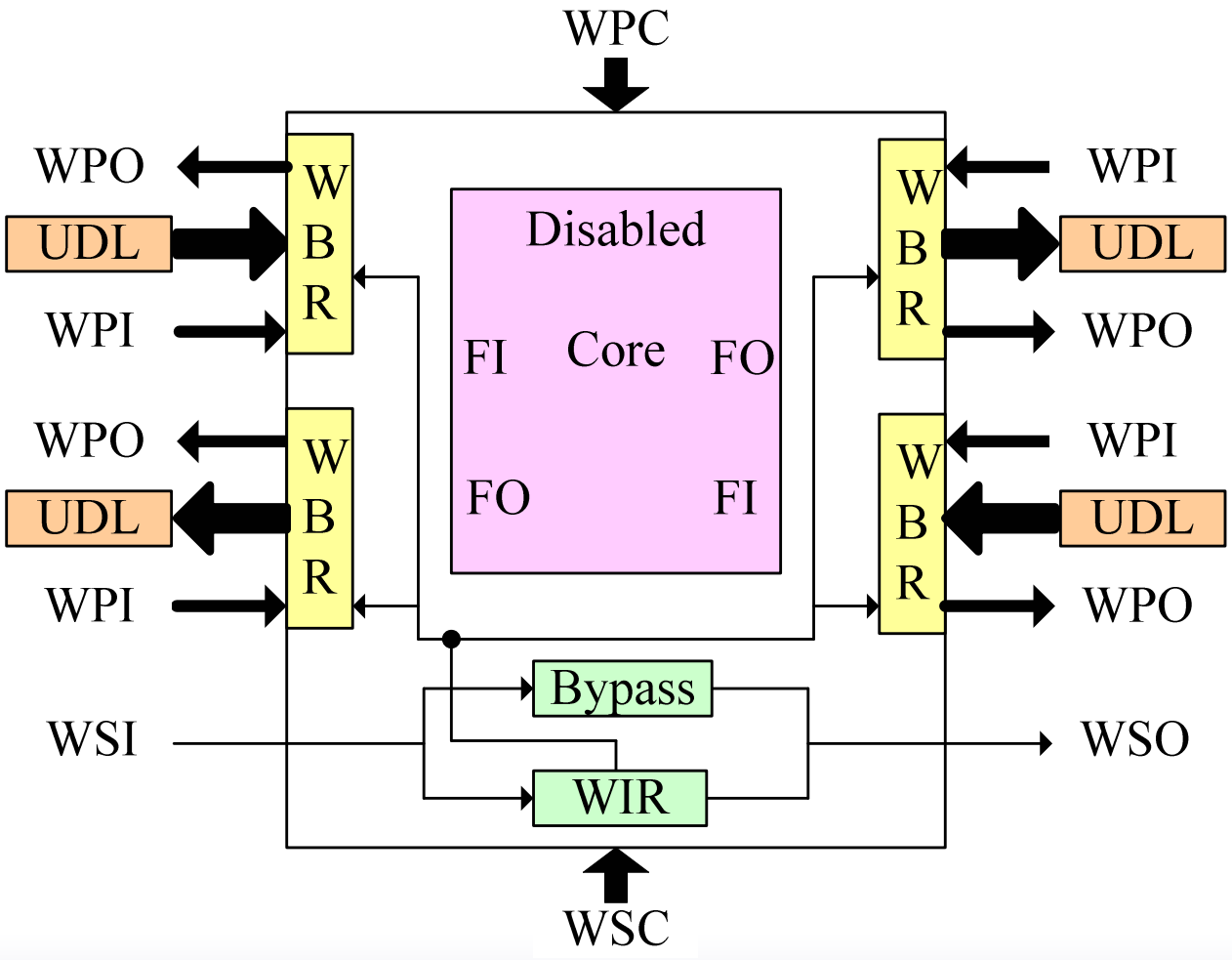
WS_BYPASS Instruction



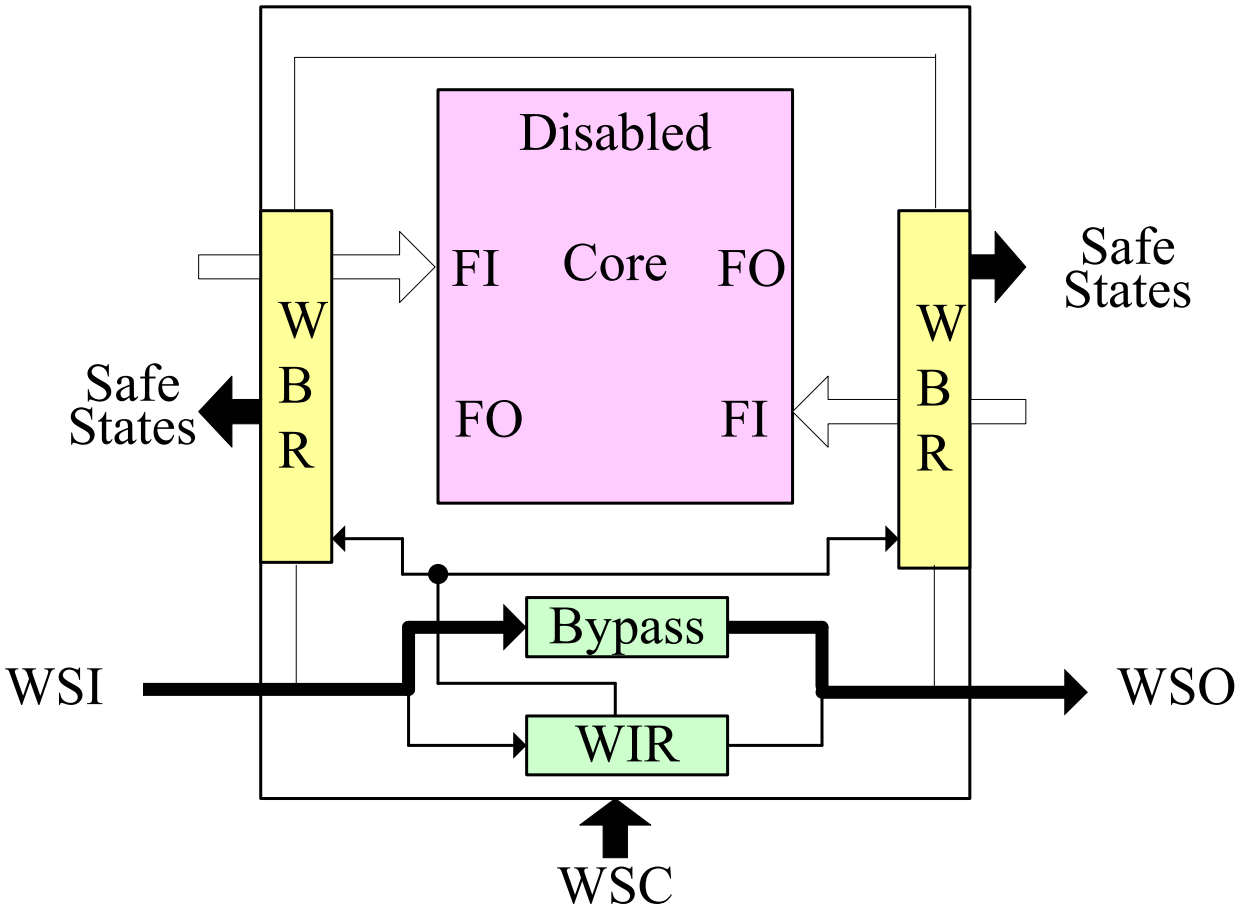
WS_EXTEST Instruction



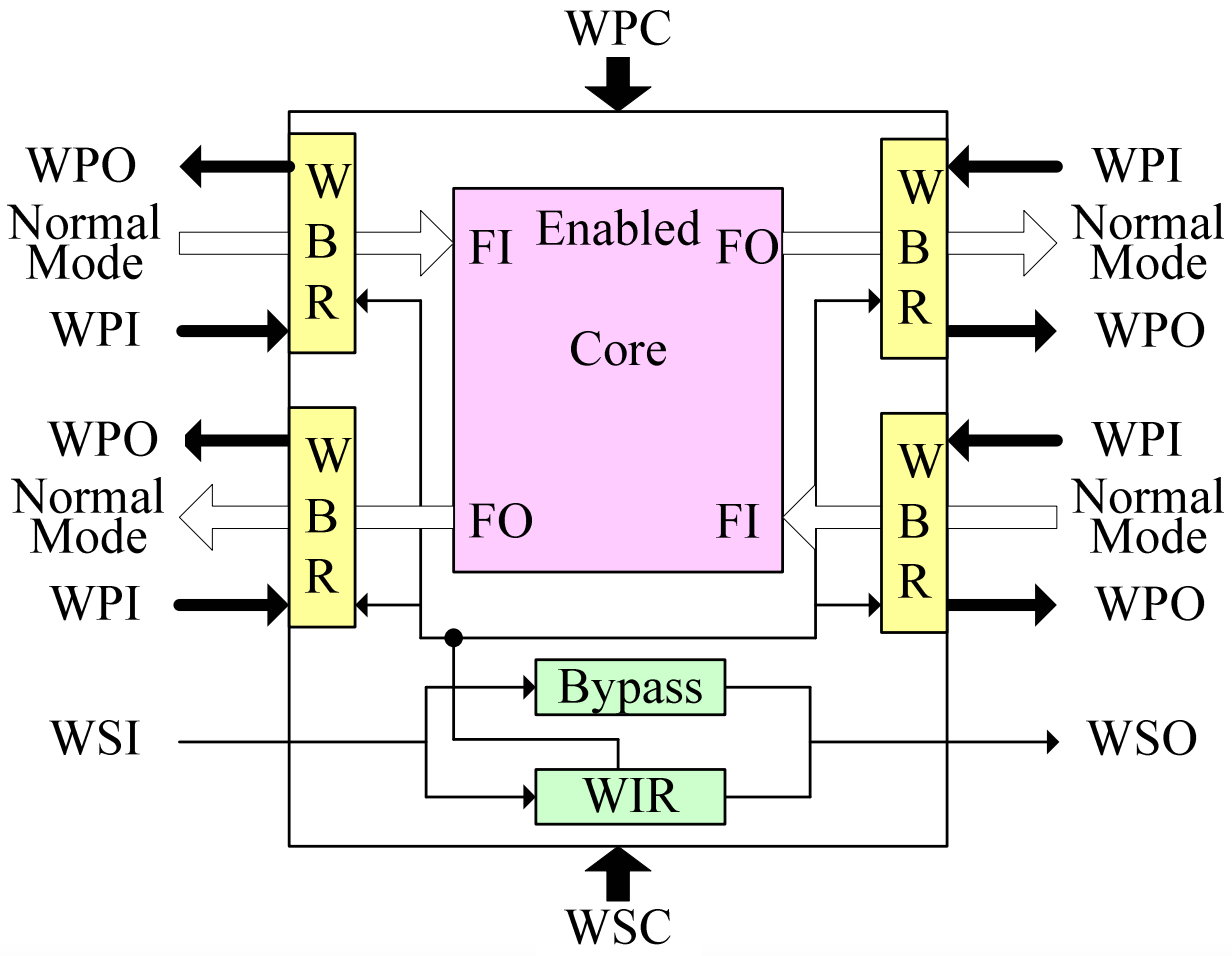
WP_EXTEXT Instruction



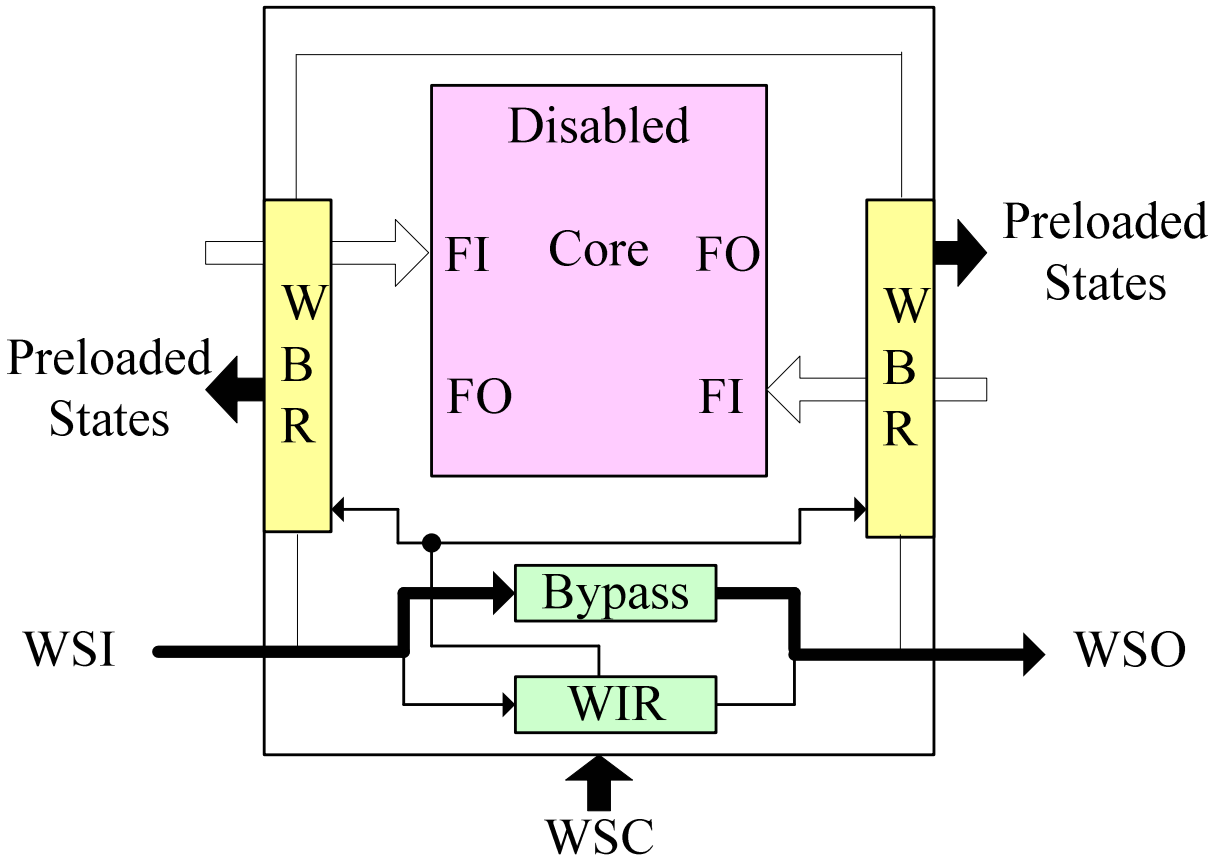
WS_SAFE Instruction



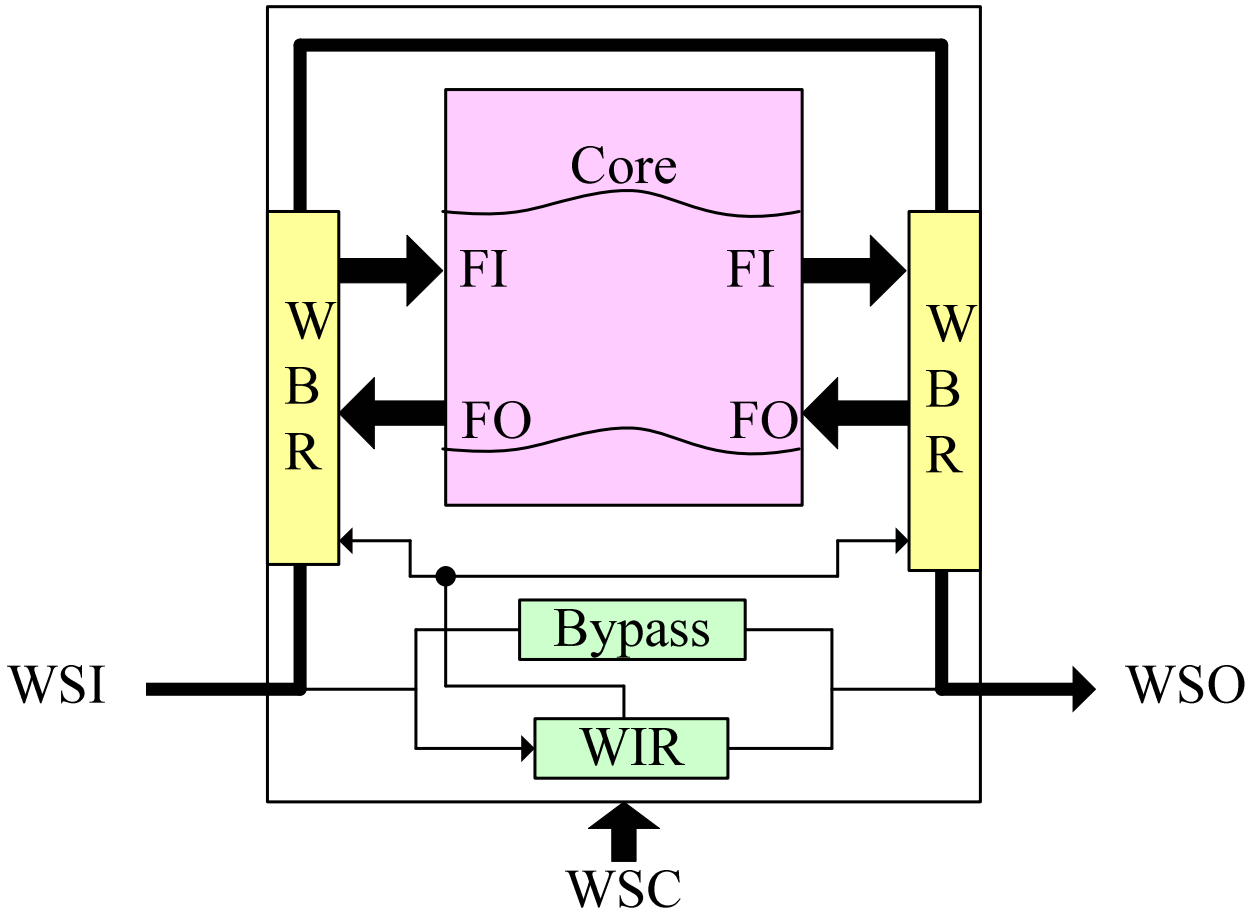
WS_PRELOAD Instruction



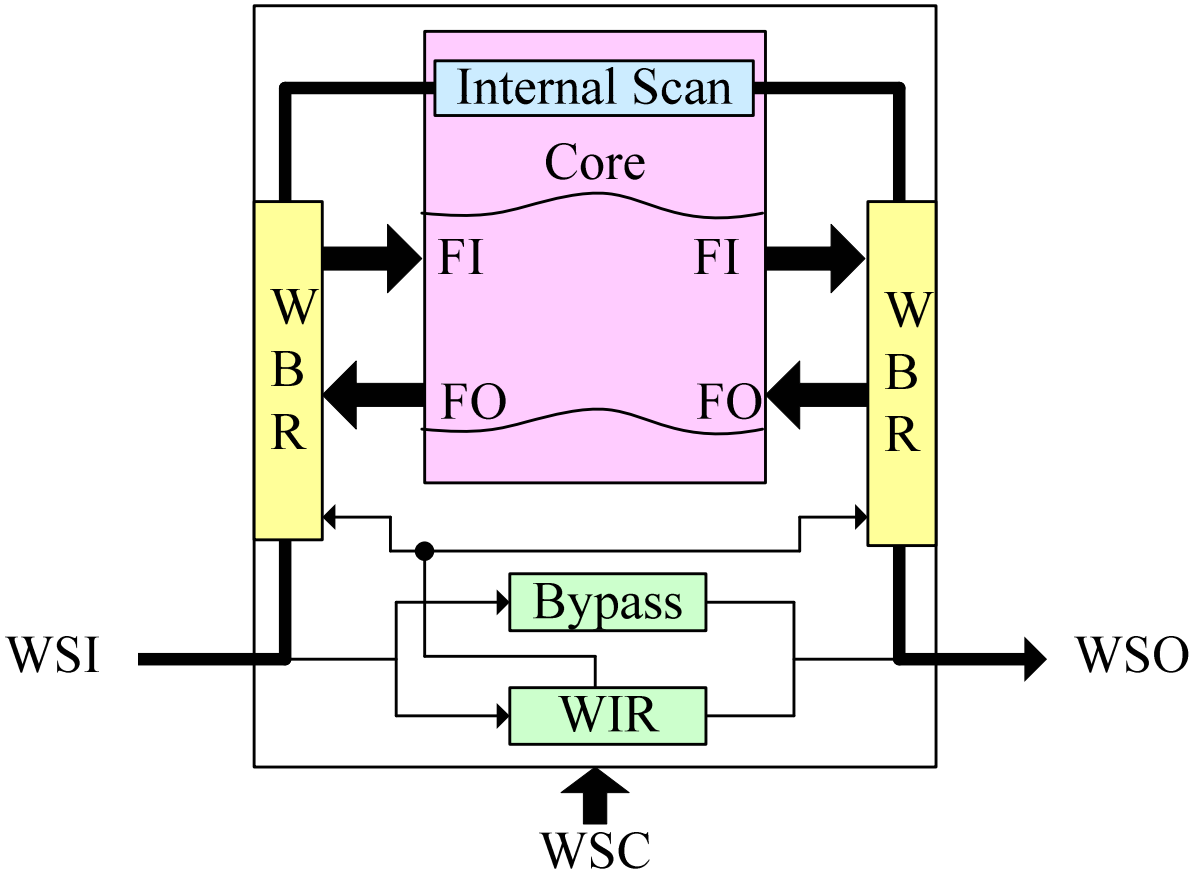
WP_PRELOAD Instruction



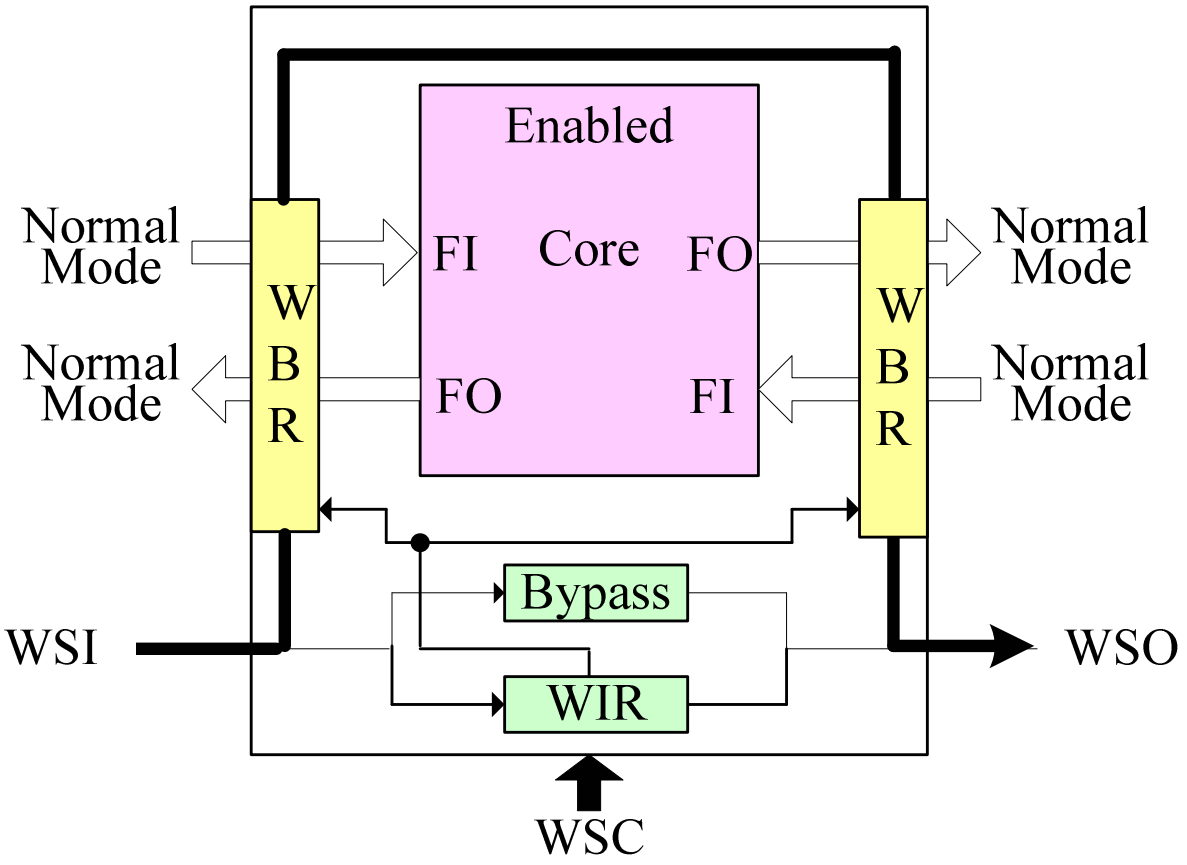
WS_CLAMP Instruction



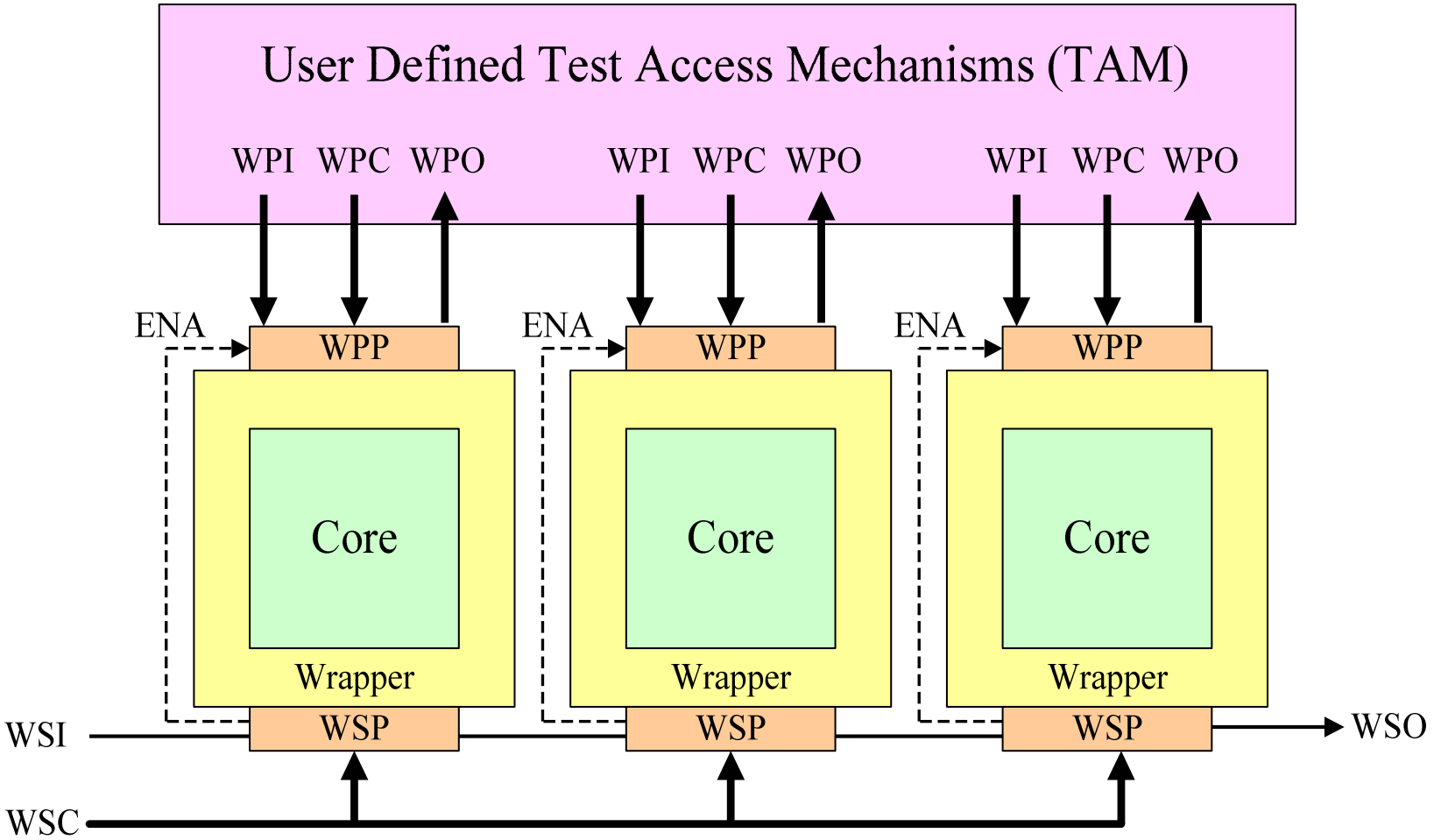
WS_INTEST Instruction



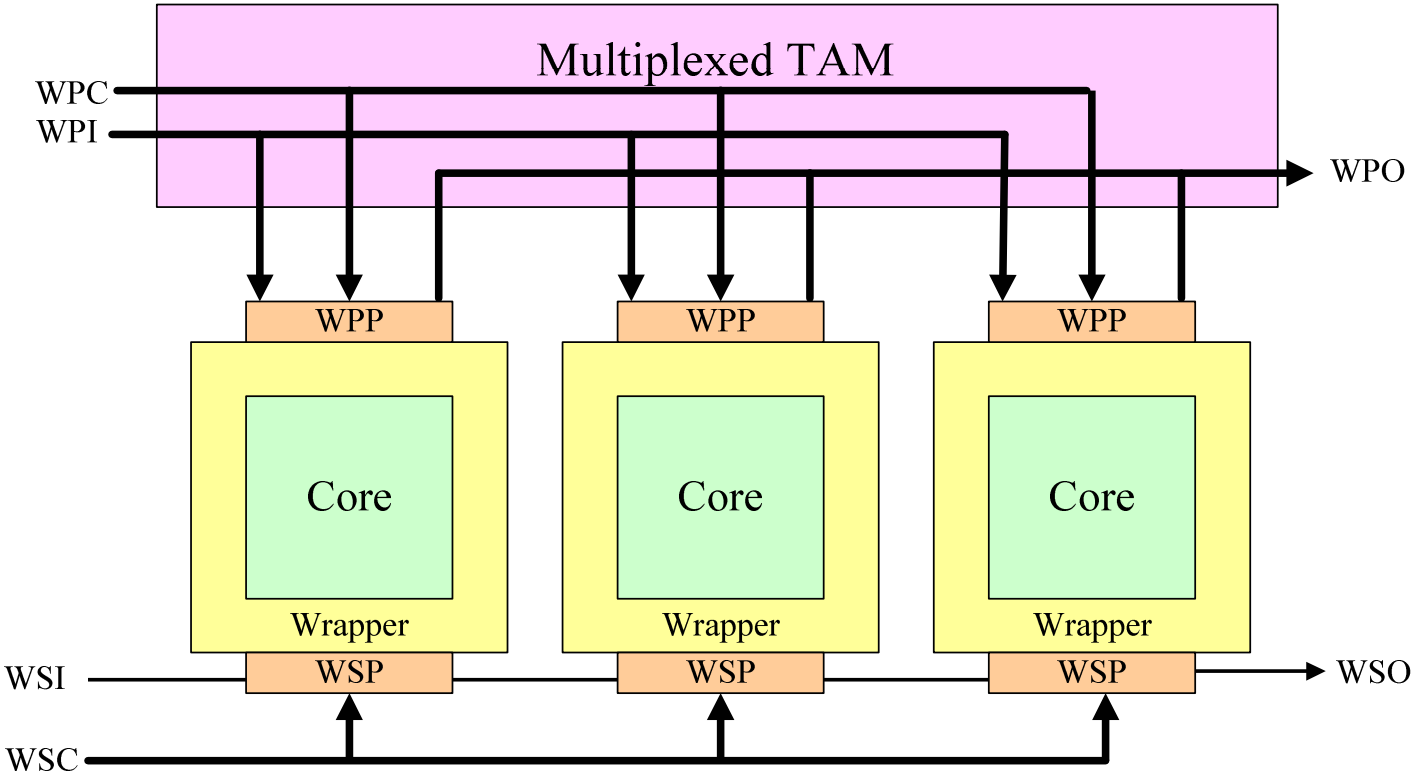
WS_INTEST_SCAN Instruction



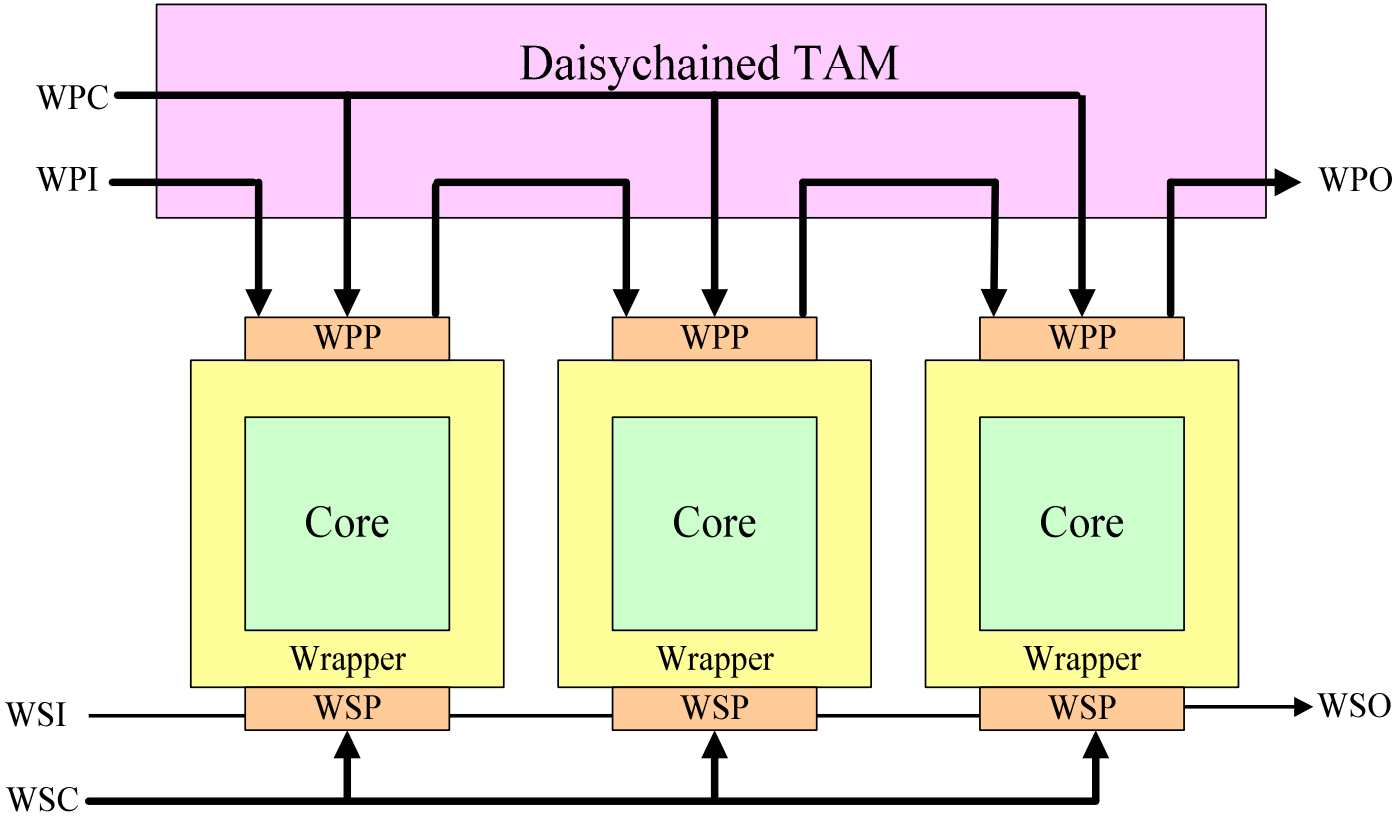
General Parallel TAM Structure



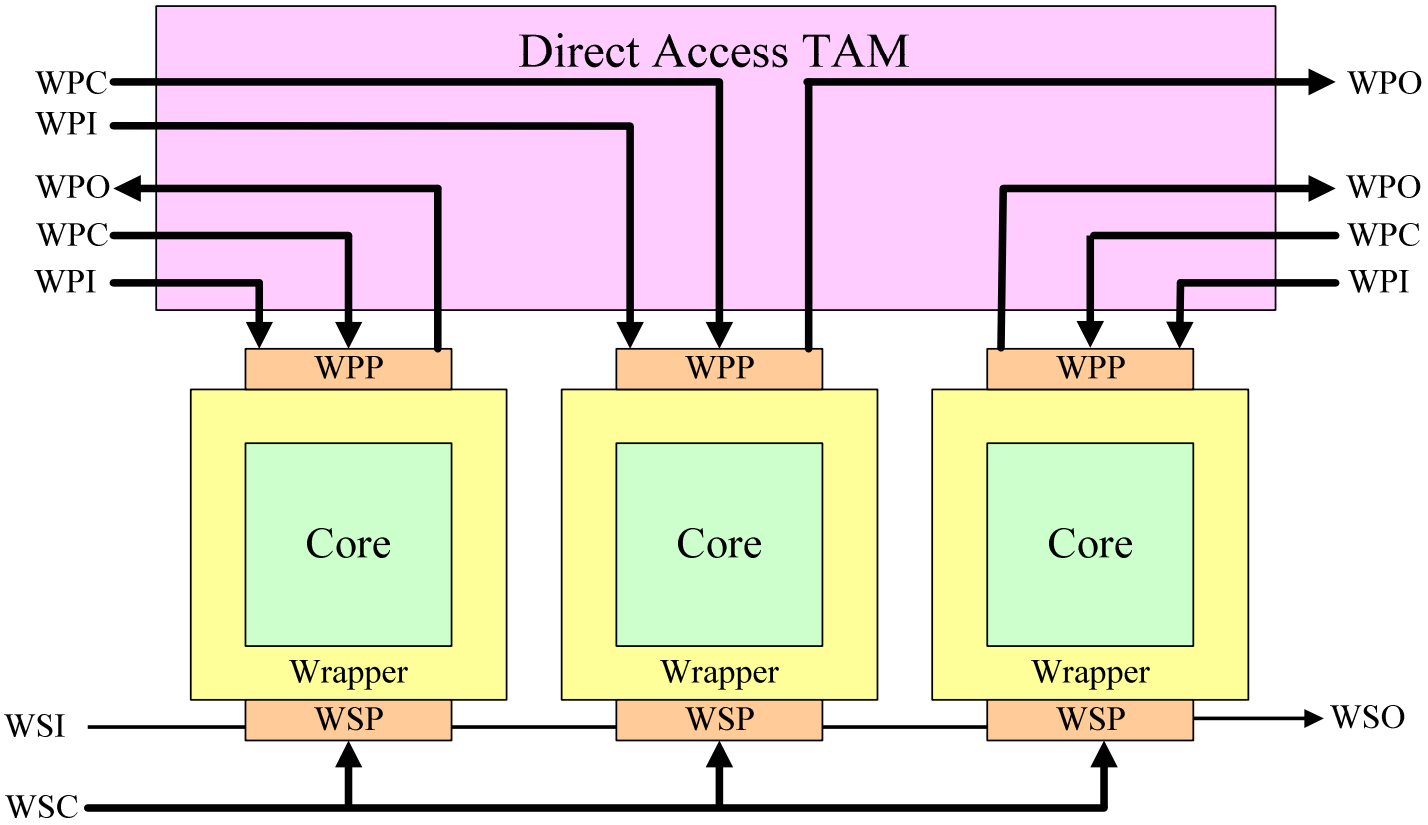
Multiplexed TAM Architectures



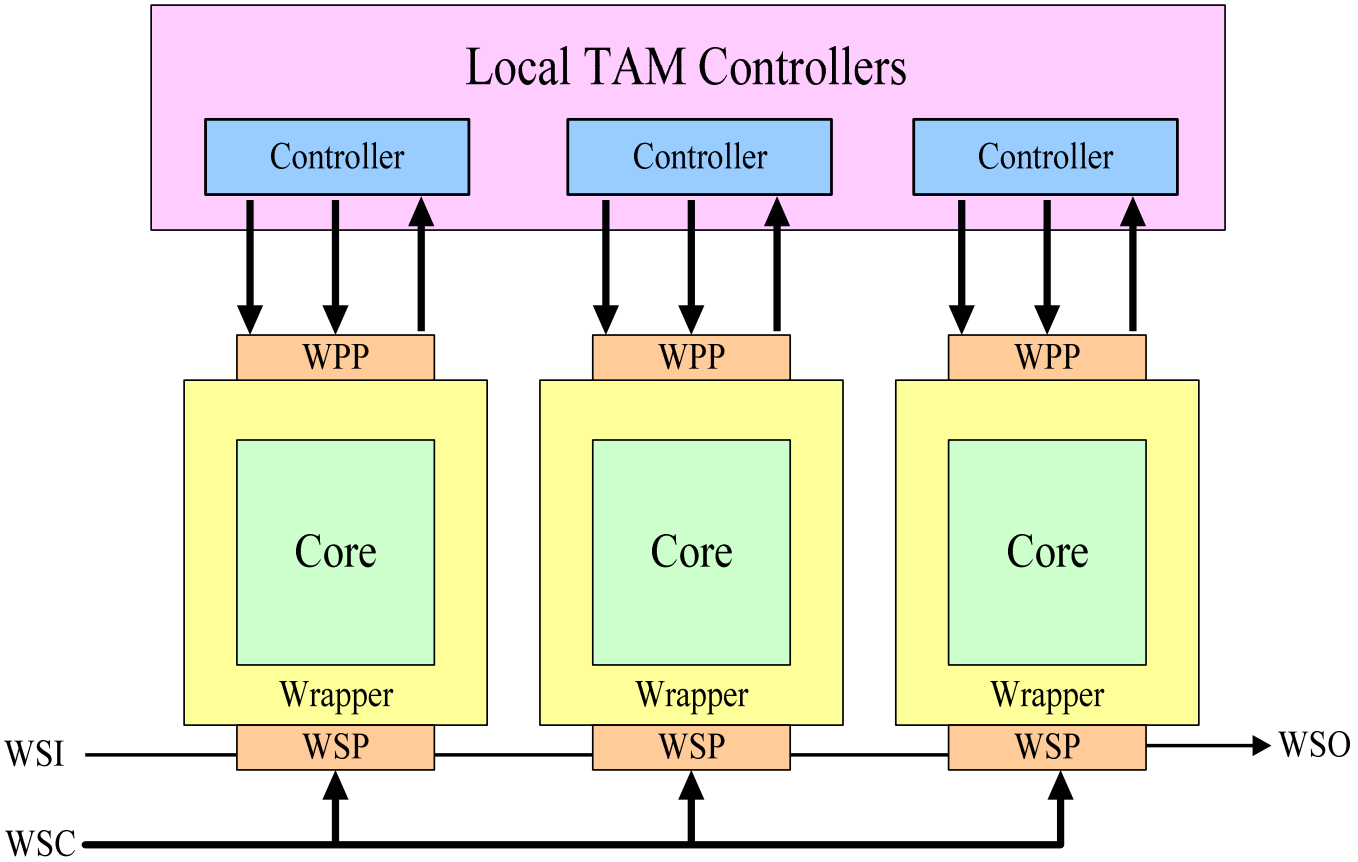
Daisy chained TAM Architecture



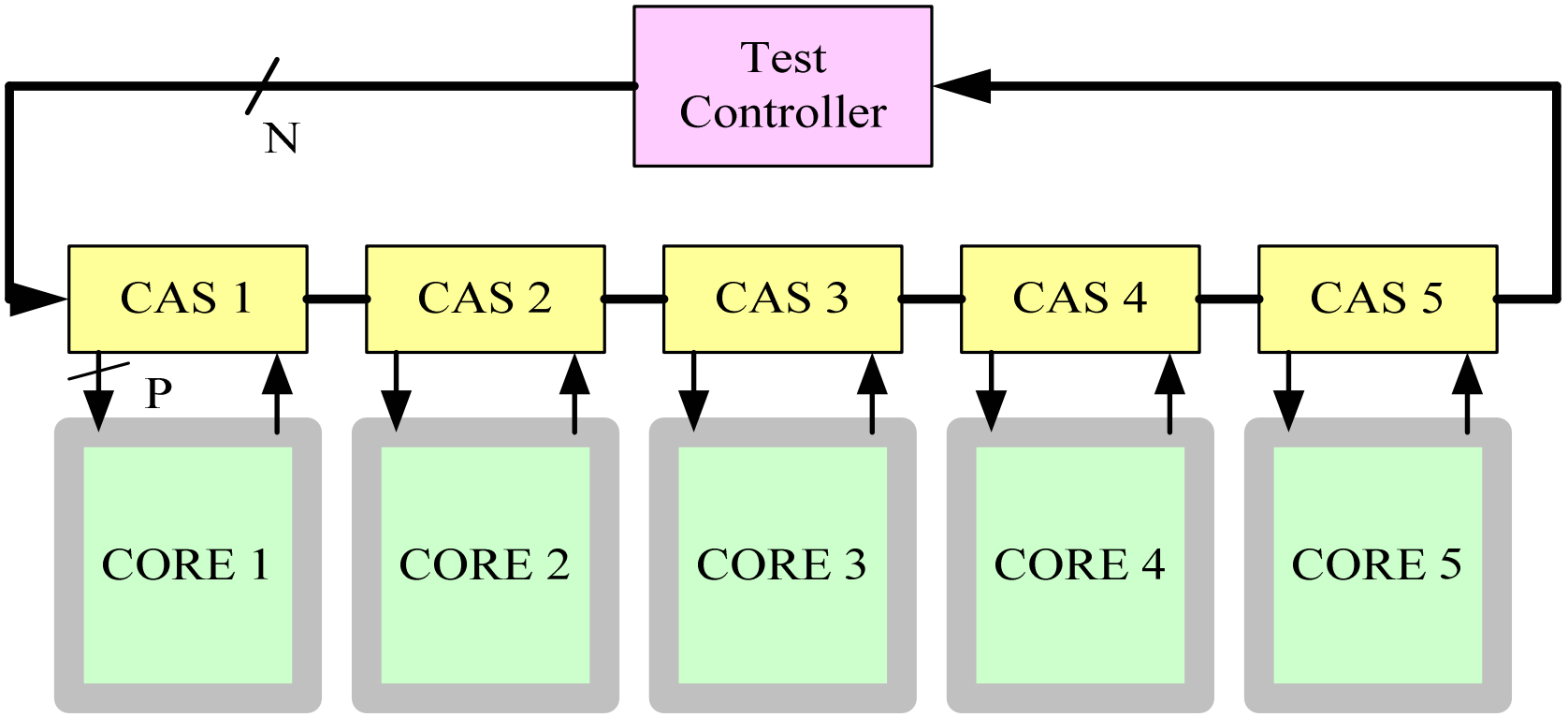
Direct Access TAM Architectures



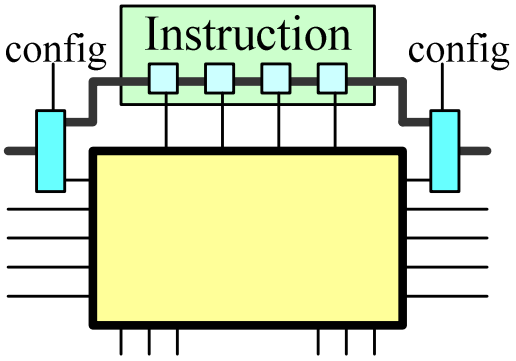
Local Controller TAM Architectures



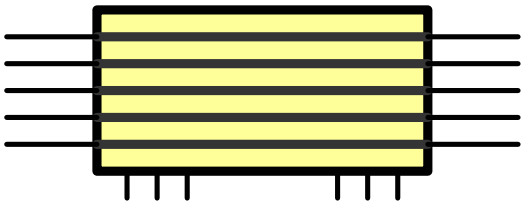
Core Access Switch (CAS) Architecture



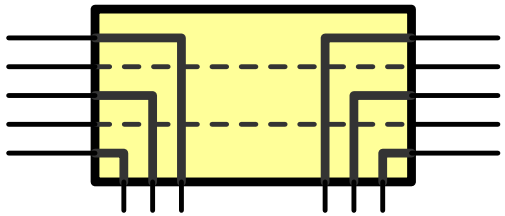
Different Functional Modes of CAS



Configuration mode

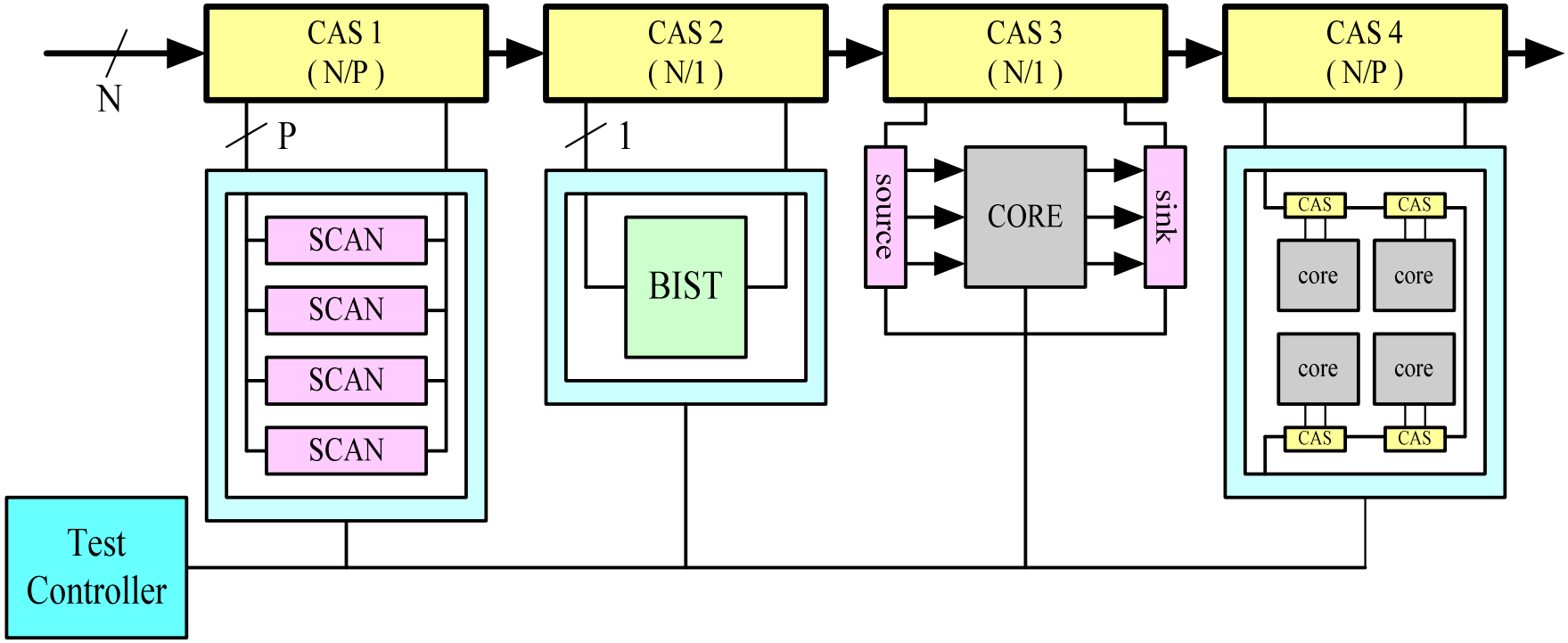


Bypass mode

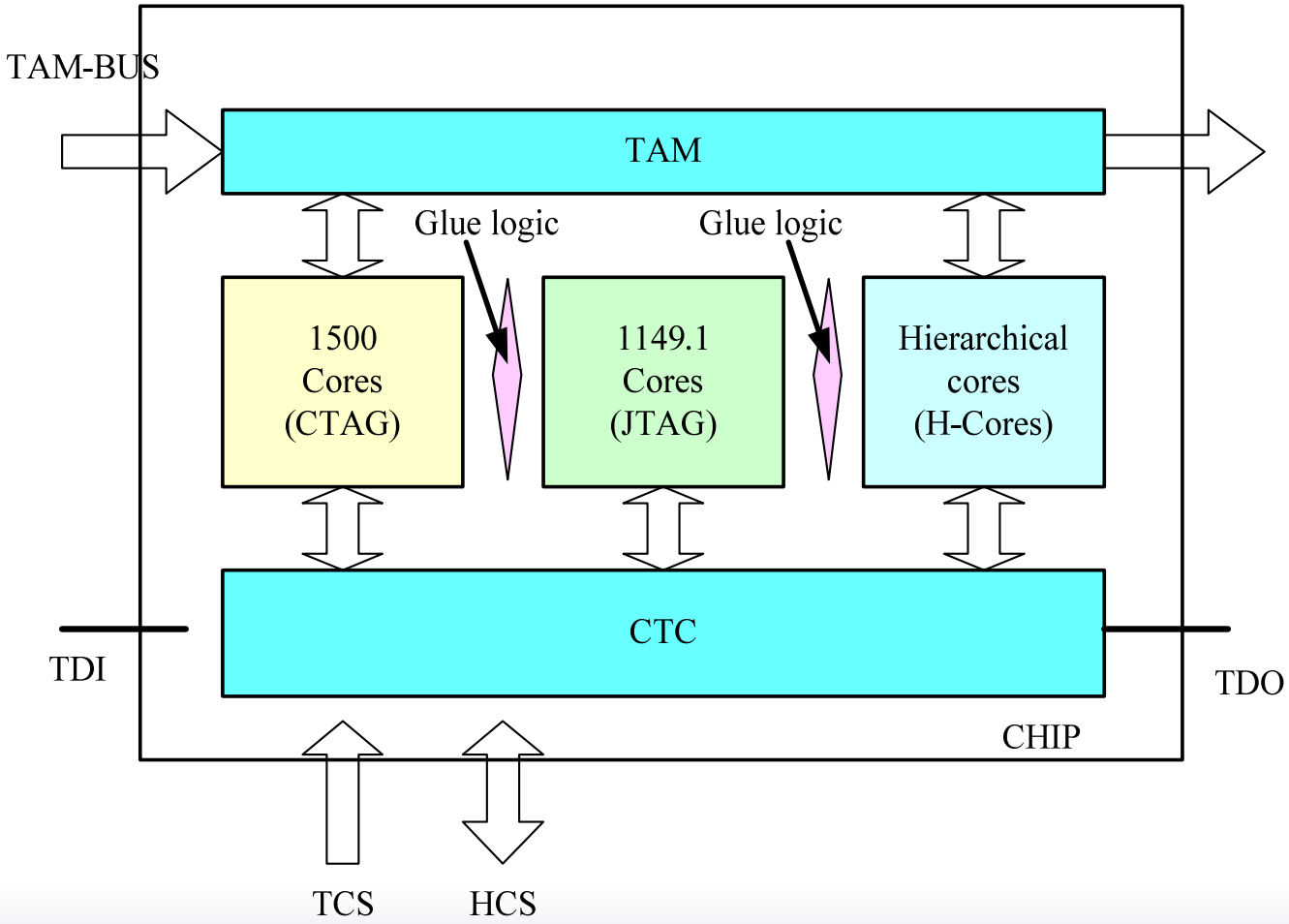


Test mode

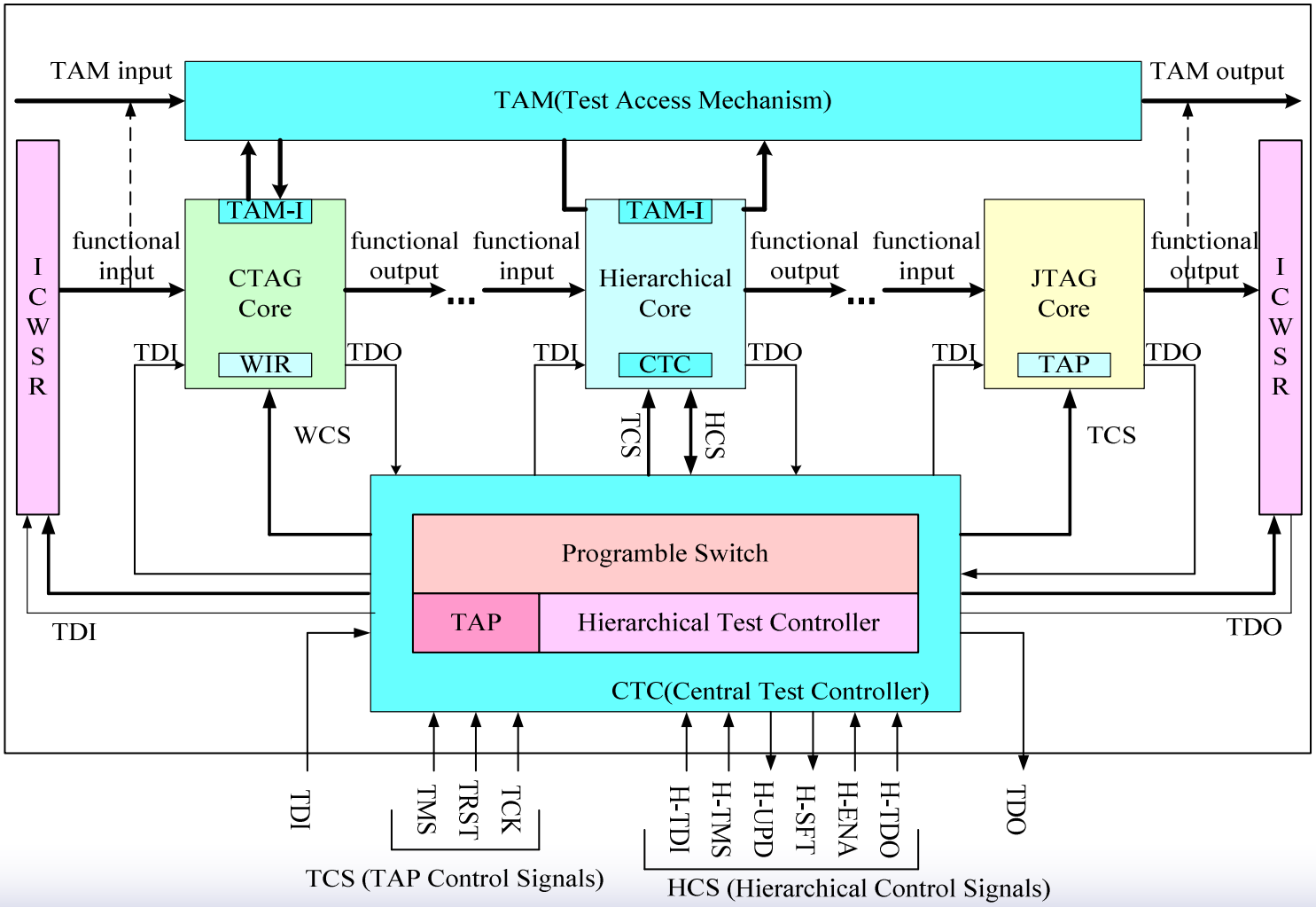
Various Types of Test Supporting Using CAS Structure



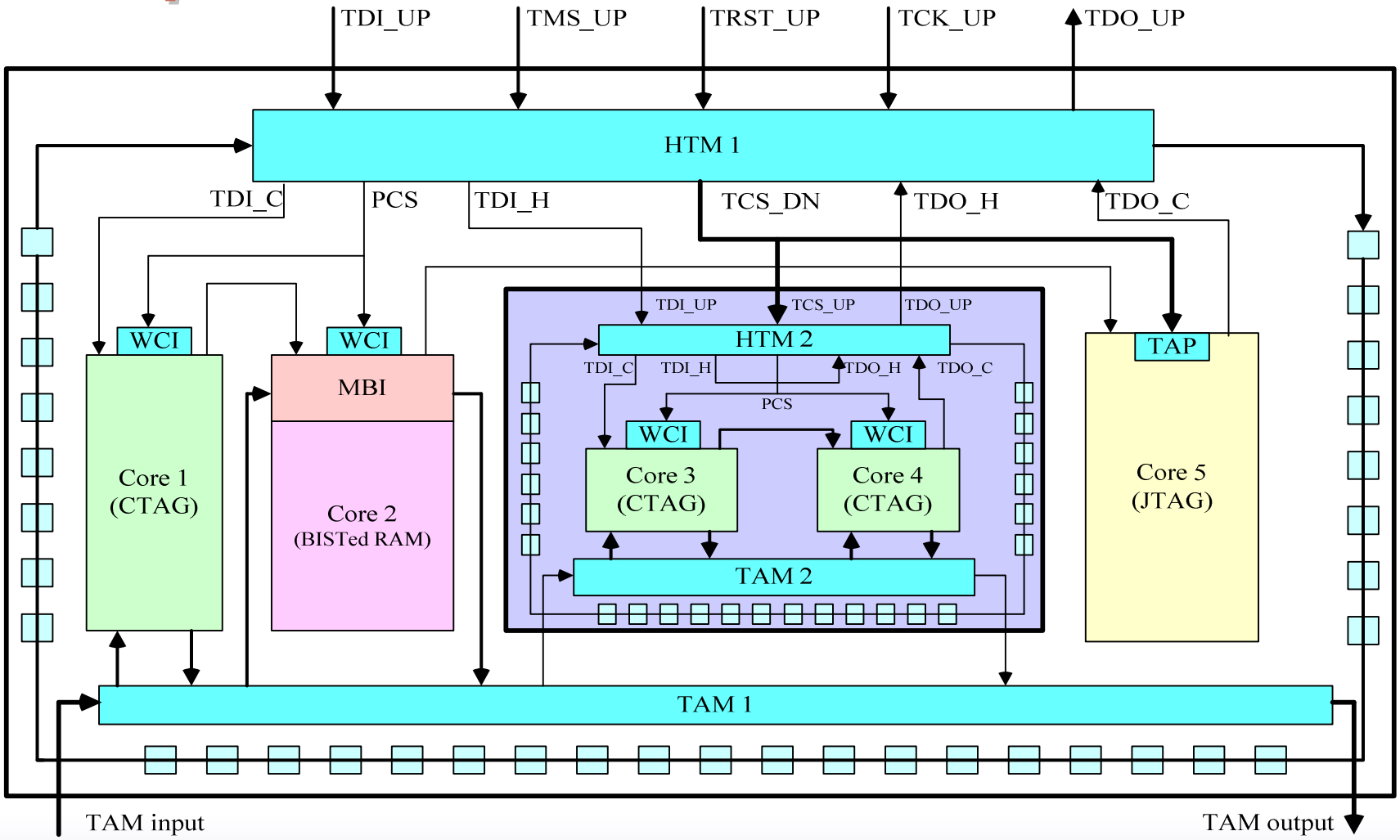
A Hierarchical Test Architecture Supporting Plug & Play Feature



Detailed I/O and CTC of The Hierarchical Test Architecture



A Hierarchical Test Architecture with I/Os Compatible to 1149.1



Comparison between 1149.1 and 1500

	1149.1	1500
Purpose	Board-level	Core-based
Parallel Mode	No	Yes
Extra Data/Control I/Os	Mandatory: TDI, TDO, TMS, TCK Optional: TRST	Mandatory: WSI, WSO, 6 WSC Optional: TransferDR, WPP, AUXCKn(s)
FSM	Yes	No
Transfer Mode	No	Yes
Latency between operations	Yes	No
Mandatory Instructions	EXTEST, BYPASS, SAMPLE, PRELOAD	WS_EXTEST, WS_BYPASS, one Wx_INTEST, WS_PRELOAD (cond. required)