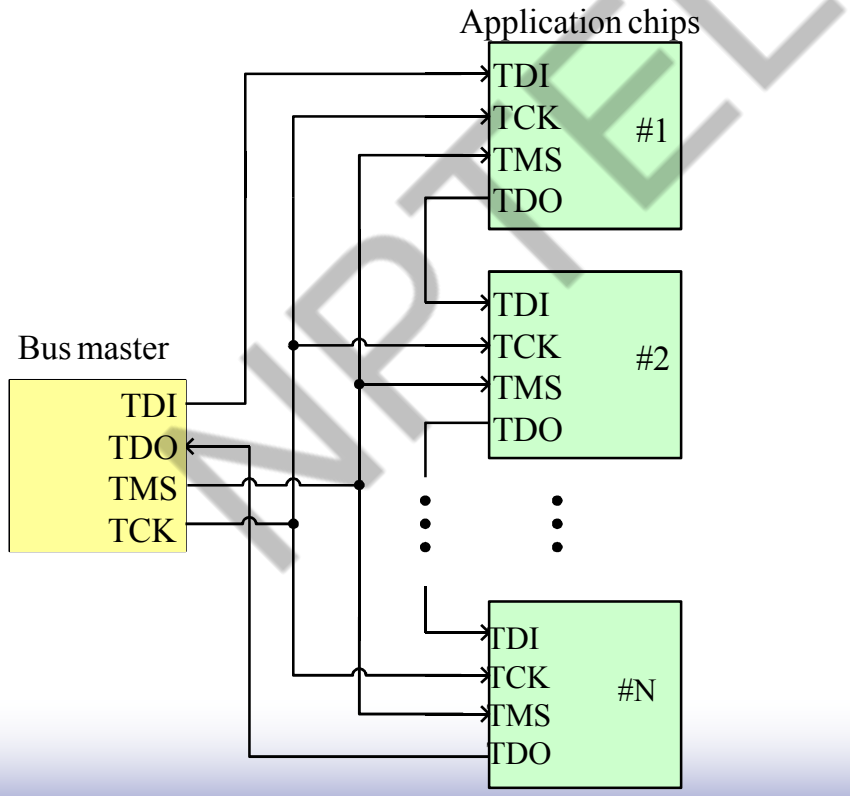
Lecture 42

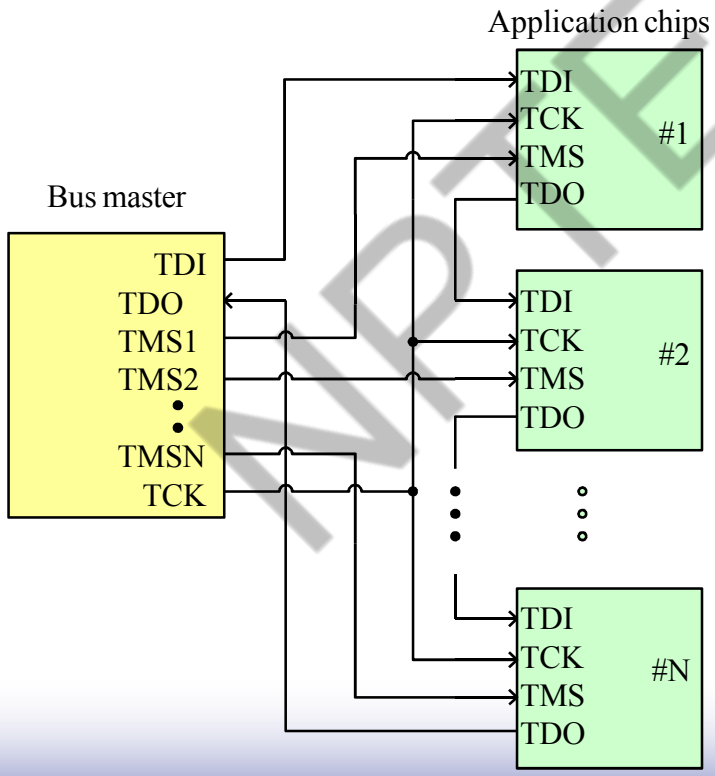
Bus Master for Chips with Boundary Scan (1/5)

- Ring architecture with shared TMS



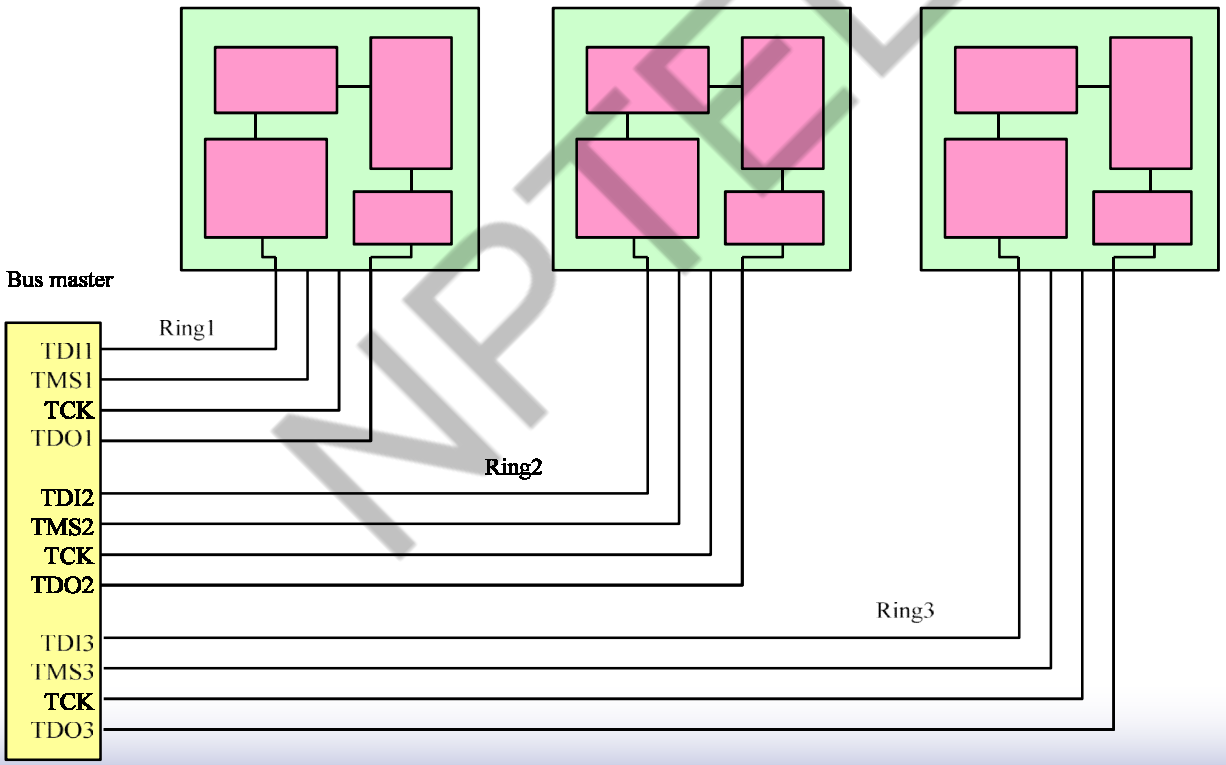
Bus Master for Chips with Boundary Scan (2/5)

- Ring architecture with separate TMS



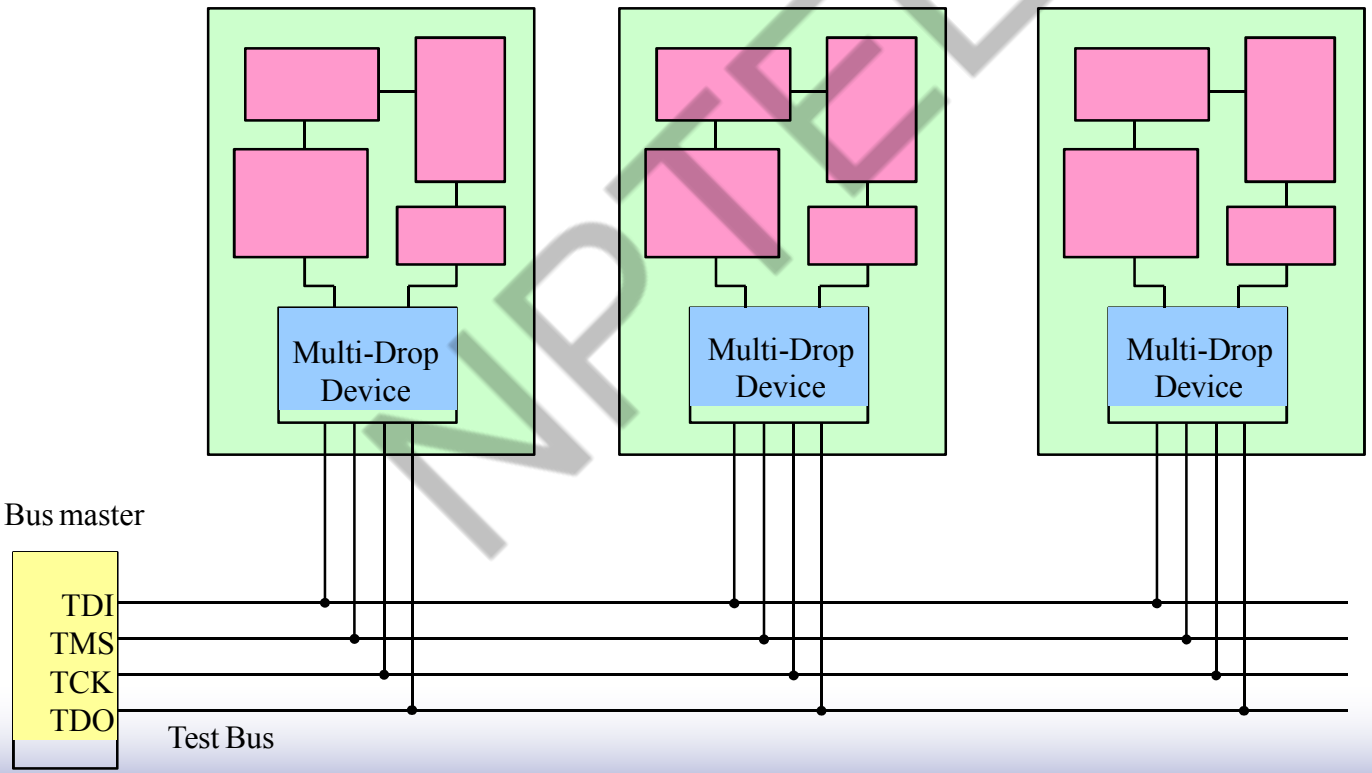
Bus Master for Chips with Boundary Scan (3/5)

- Star architecture



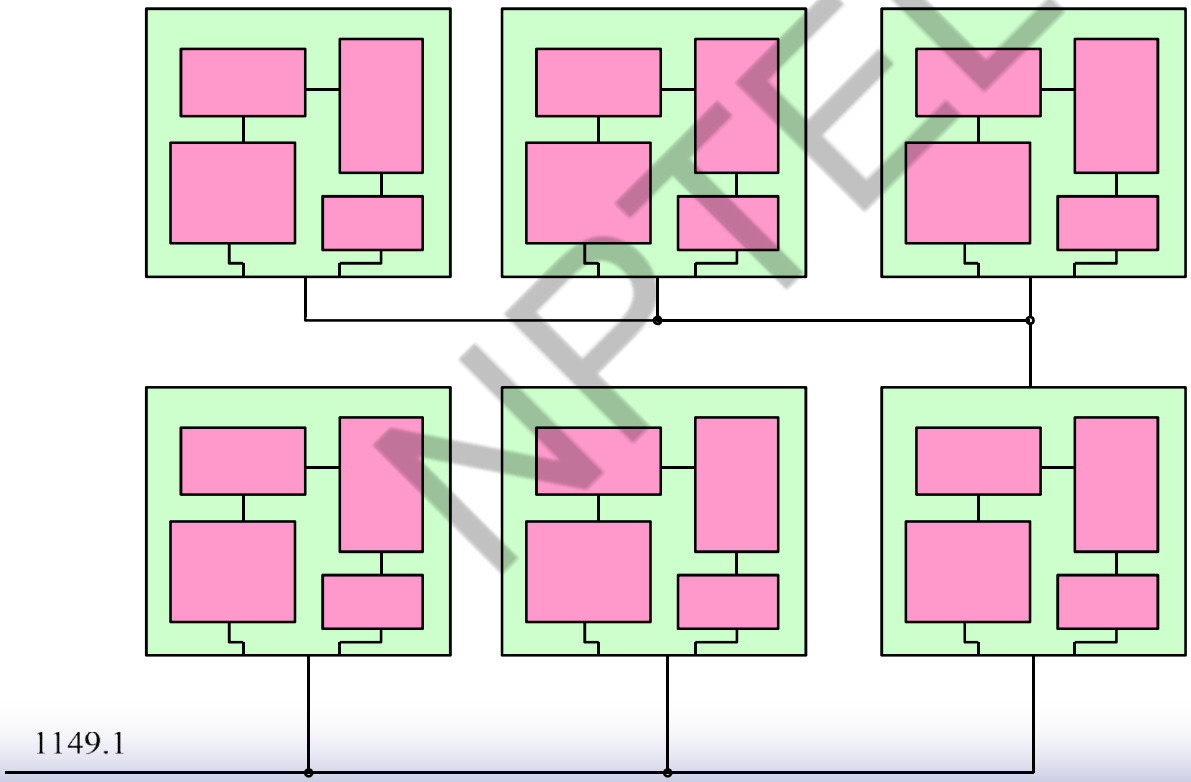
Bus Master for Chips with Boundary Scan (4/5)

- Multi-drop architecture



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

- Hierarchical architecture



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

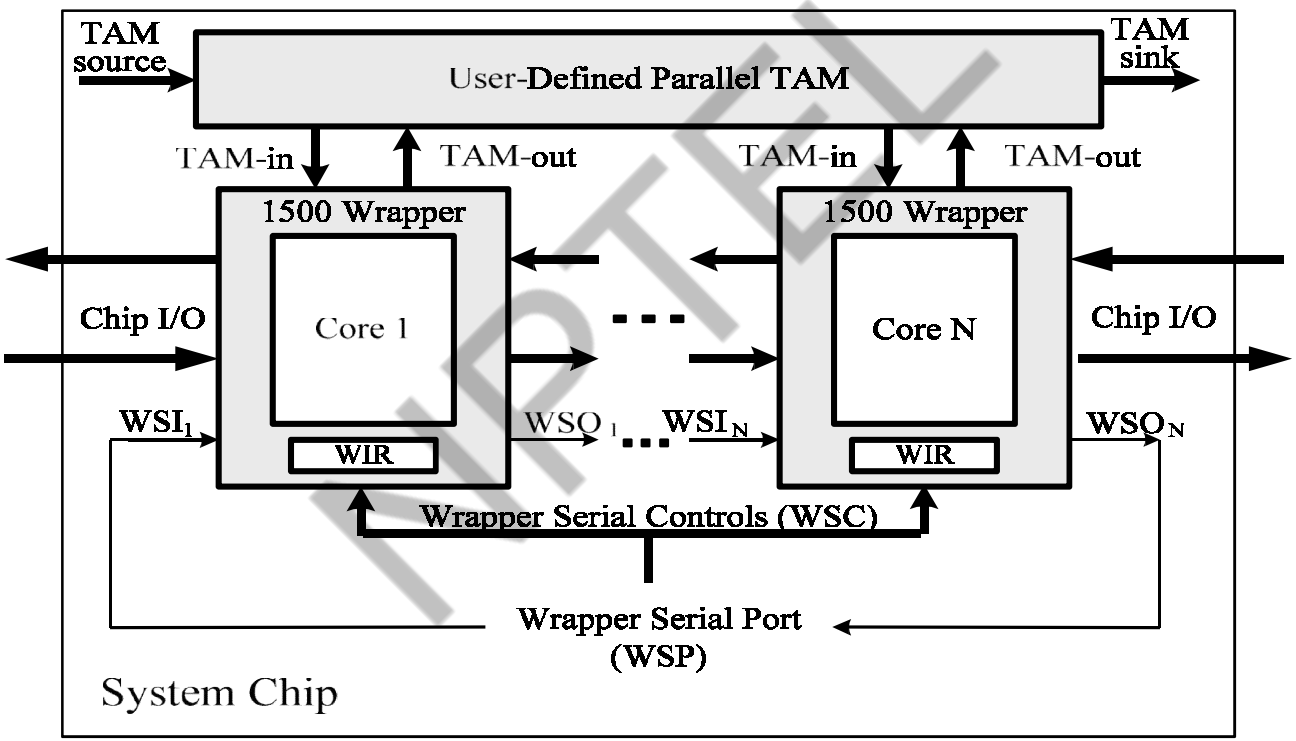
SOC Test Problems/Requirements (1/2)

- 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)

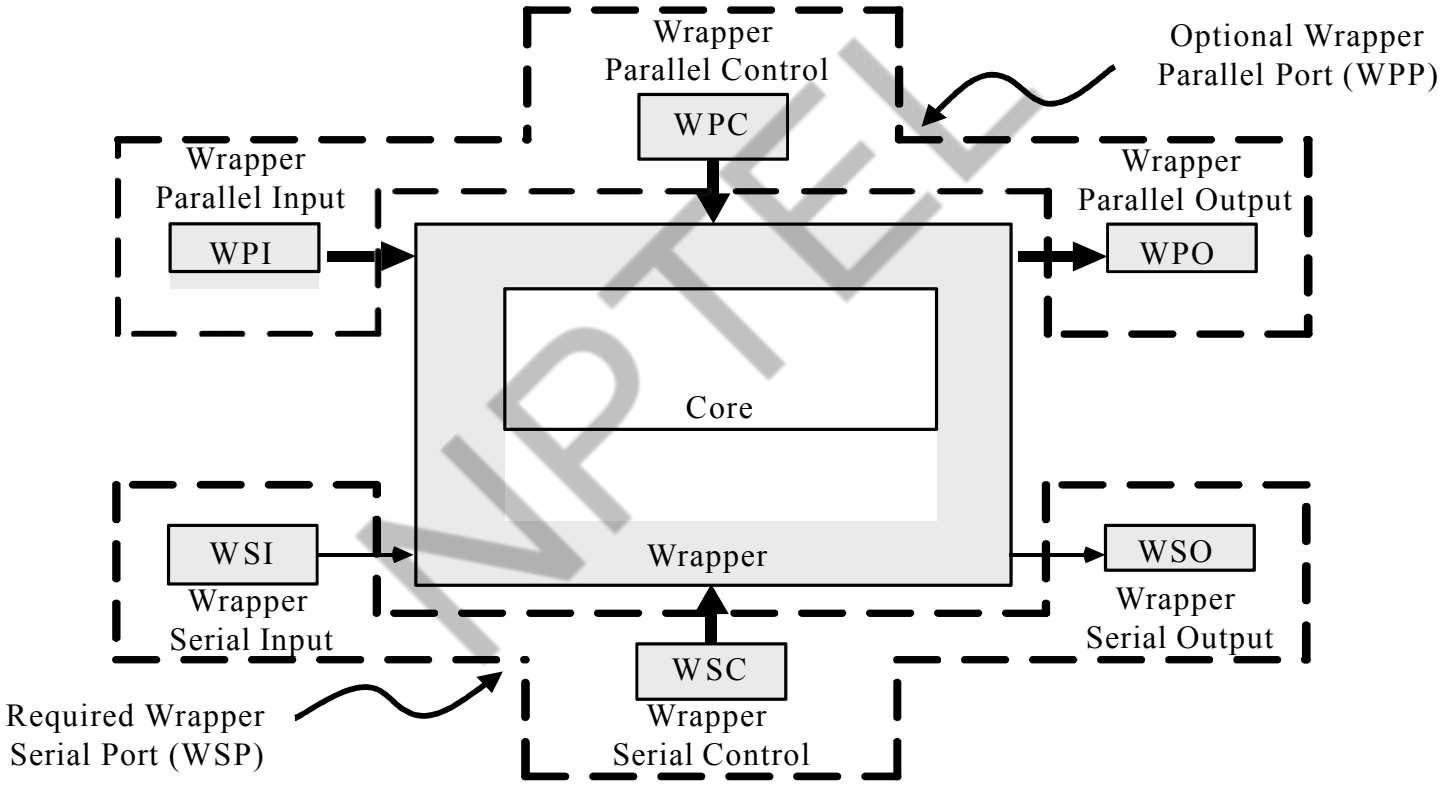
- Higher-performance core pins than SOC pins
  - Need on-chip, at-speed testing
- External ATE inefficiency
  - Need "on-chip ATE"
- Long test application time
  - Need parallel testing or test scheduling
- Test power must be considered
  - Need lower power design or test scheduling
- Testable design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

A System Overview of IEEE 1500 Standard

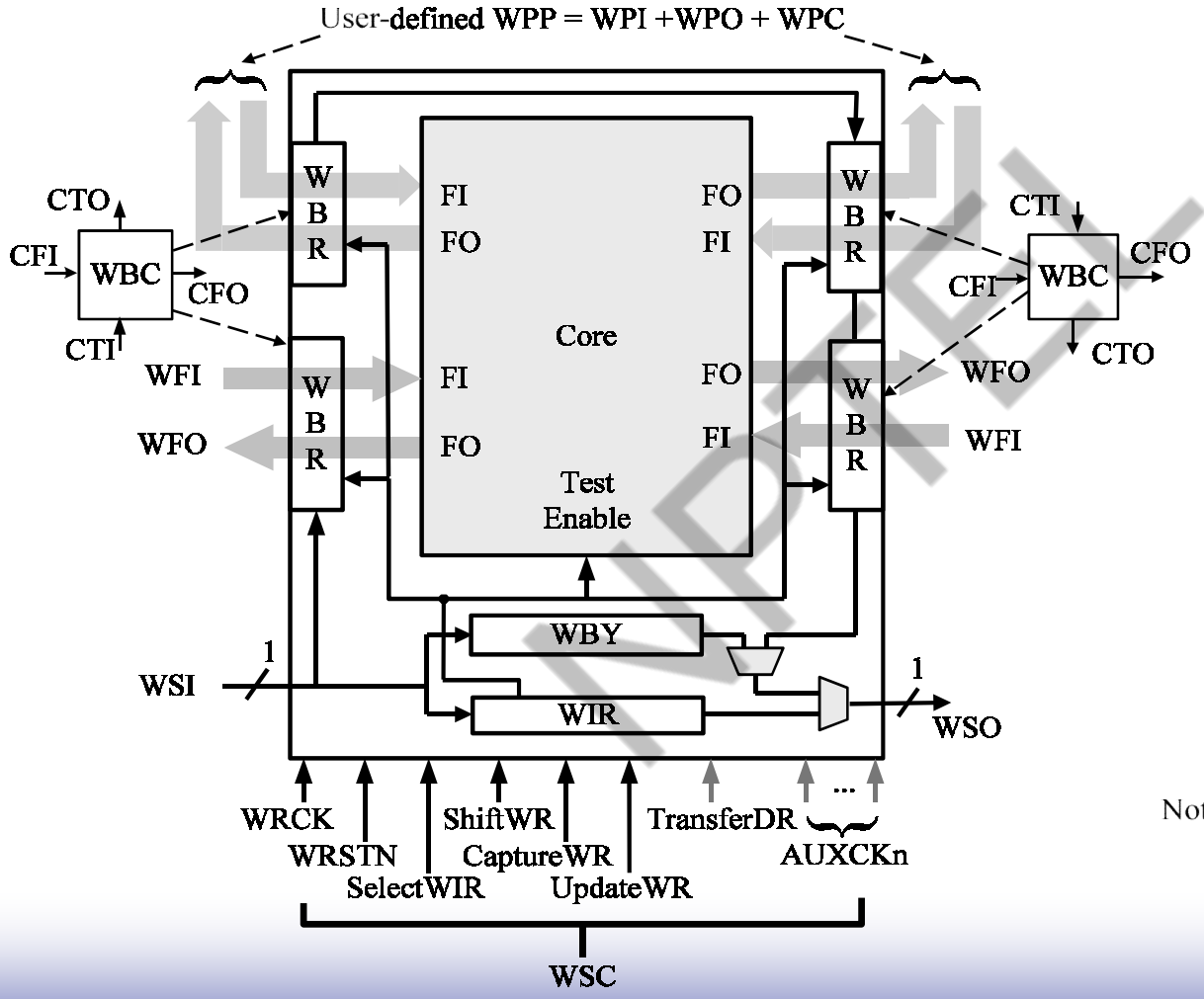


Lecture 43

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

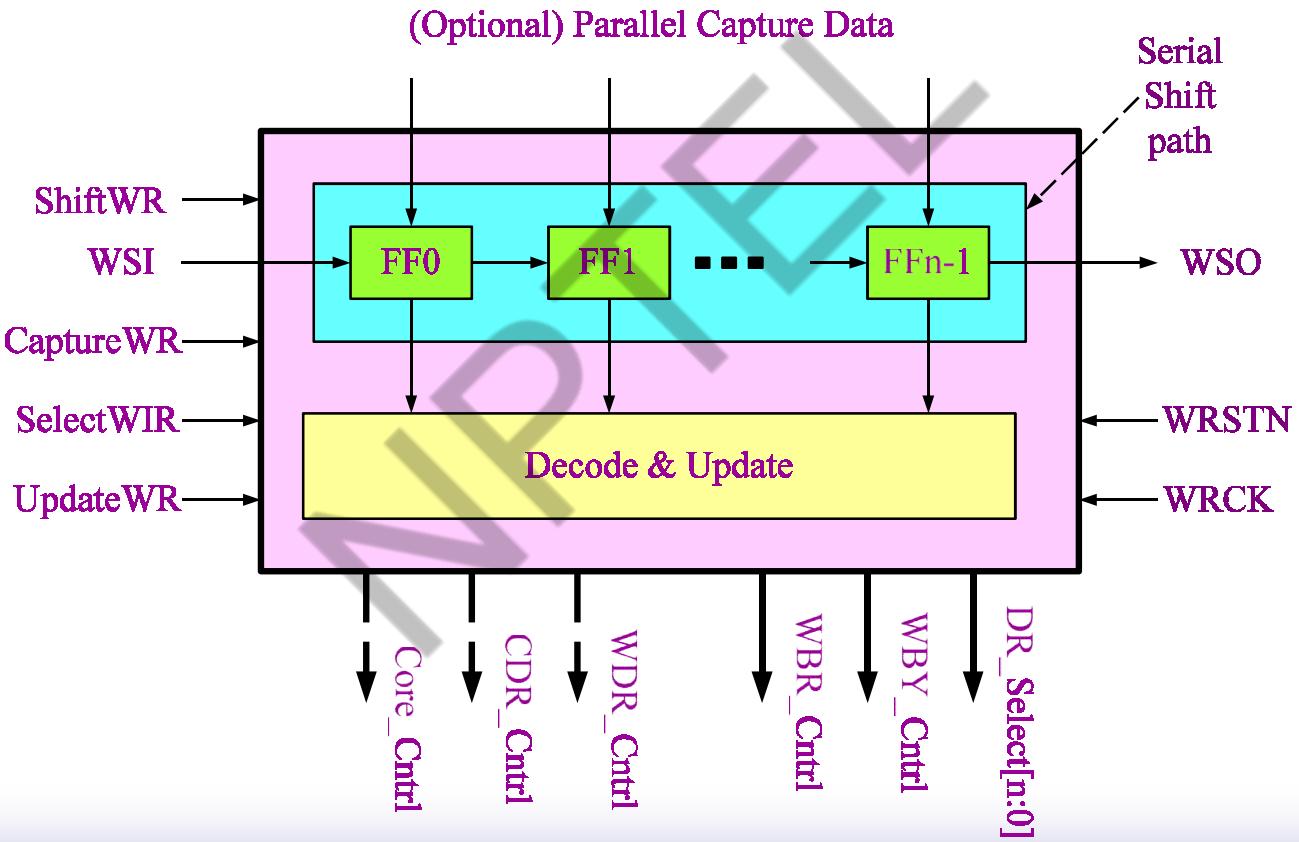
Wrapper Components

- 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



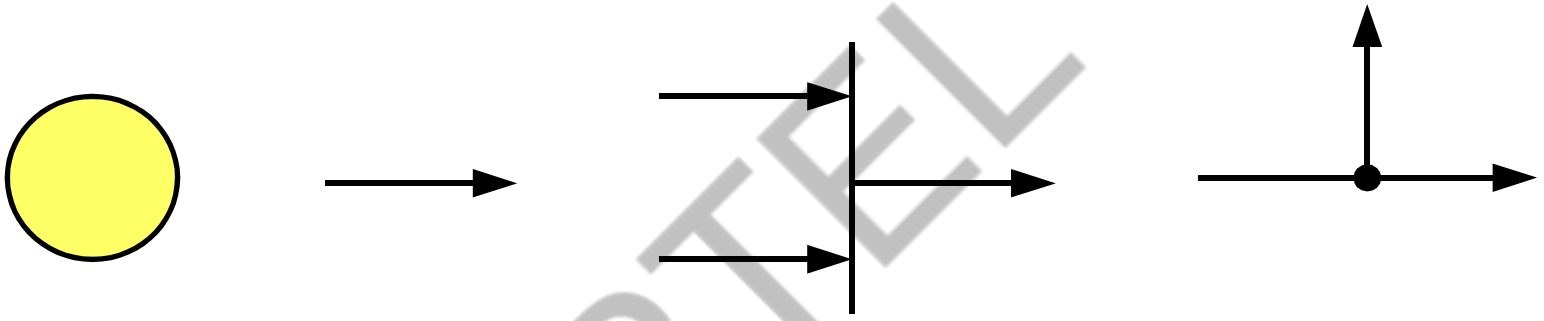
Wrapper Boundary Register (WBR)

- 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



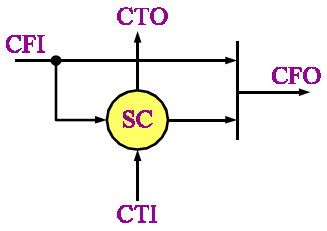
Storage element

Data path

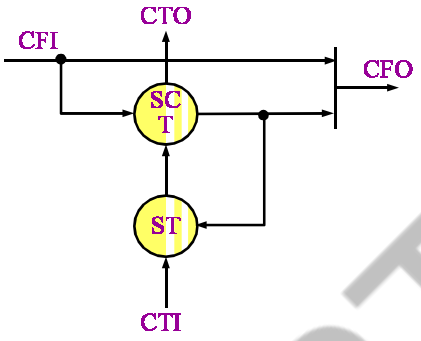
Decision point

Data paths from a source

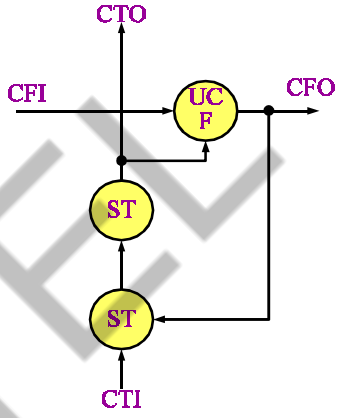
Some Typical WBC's Represented by Bubble Diagrams



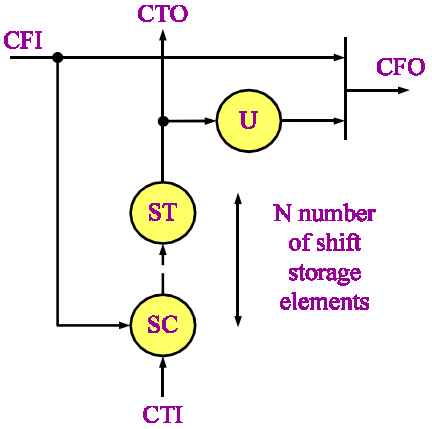
(a) WC_SD1_CII



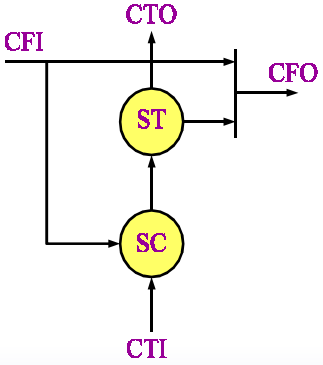
(b) WC_SD2_CIO



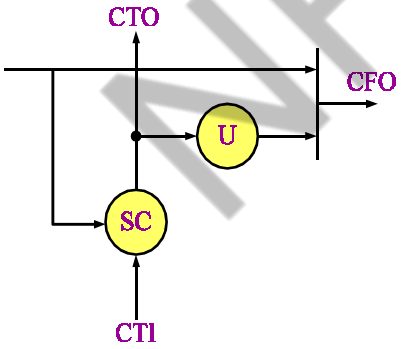
(e) WC_SD2_CIU_UF



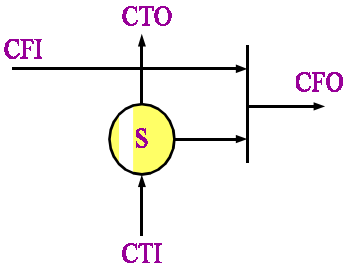
(f) WC_SDn_CII_UD



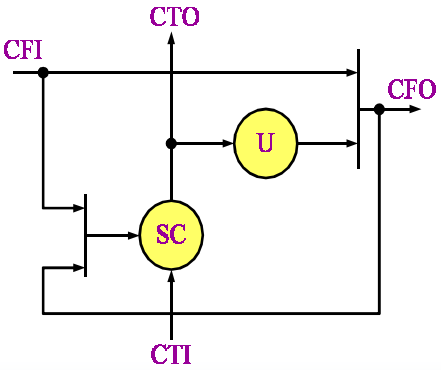
(c) WC_SD2_CII



(d) WC_SD1_CII_UD

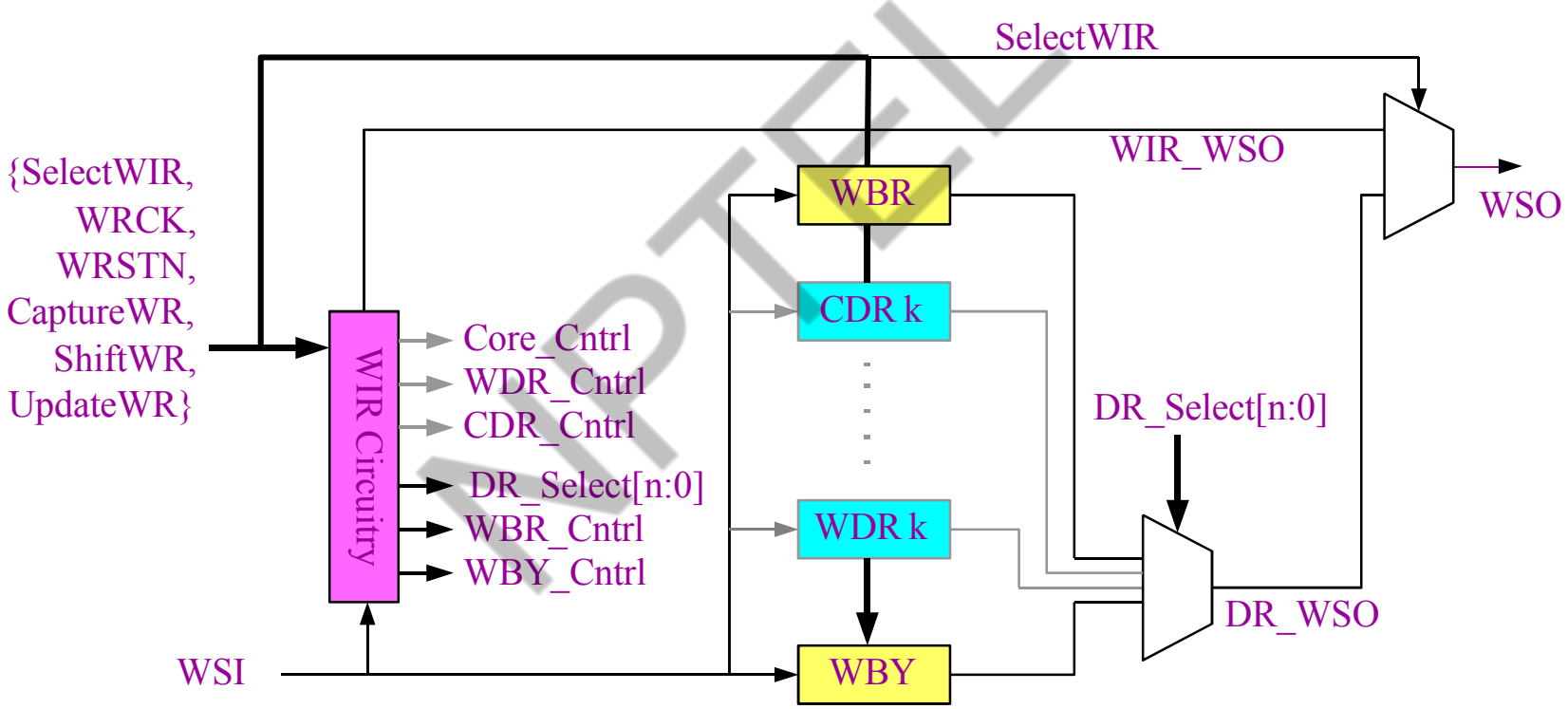


(g) WC_SD1_CN

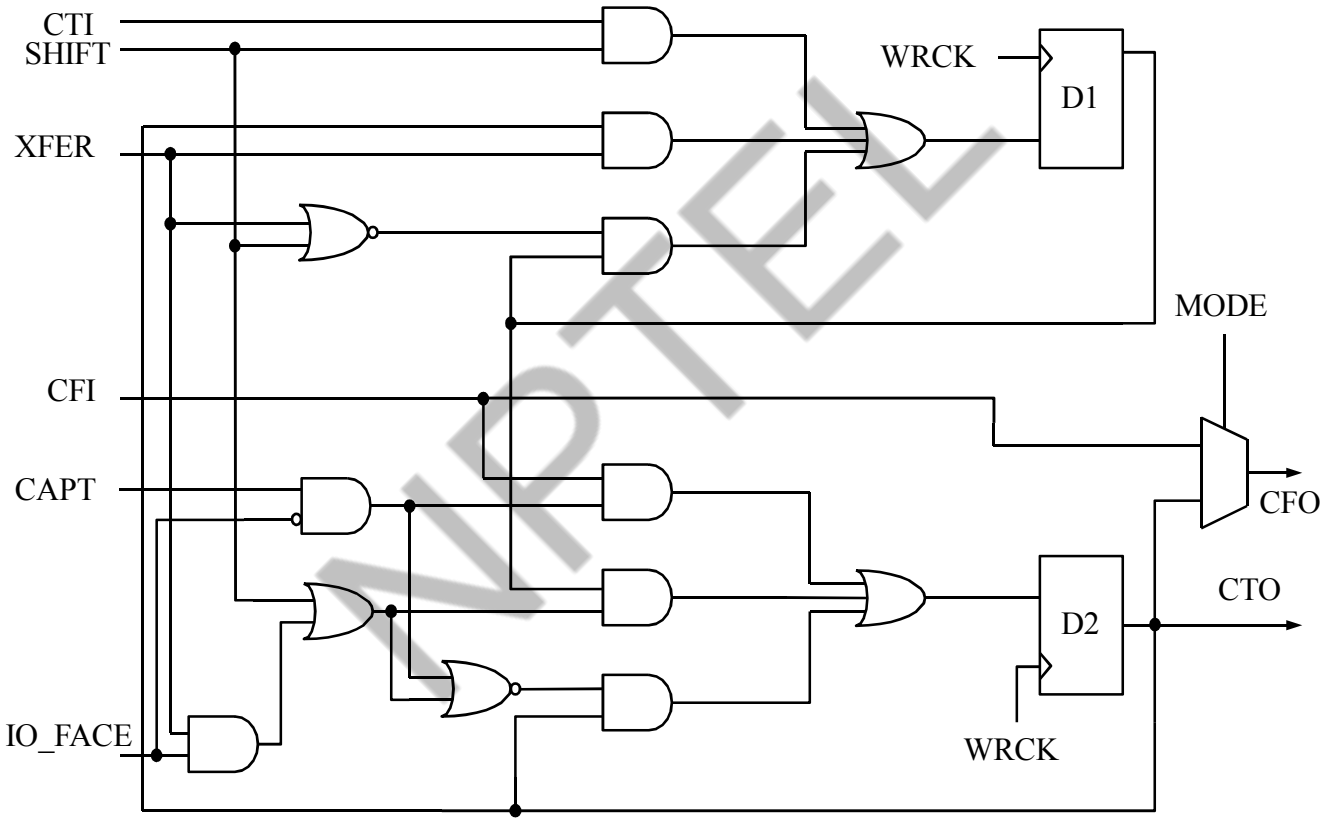


(h) WC_SD1_CBI_UD

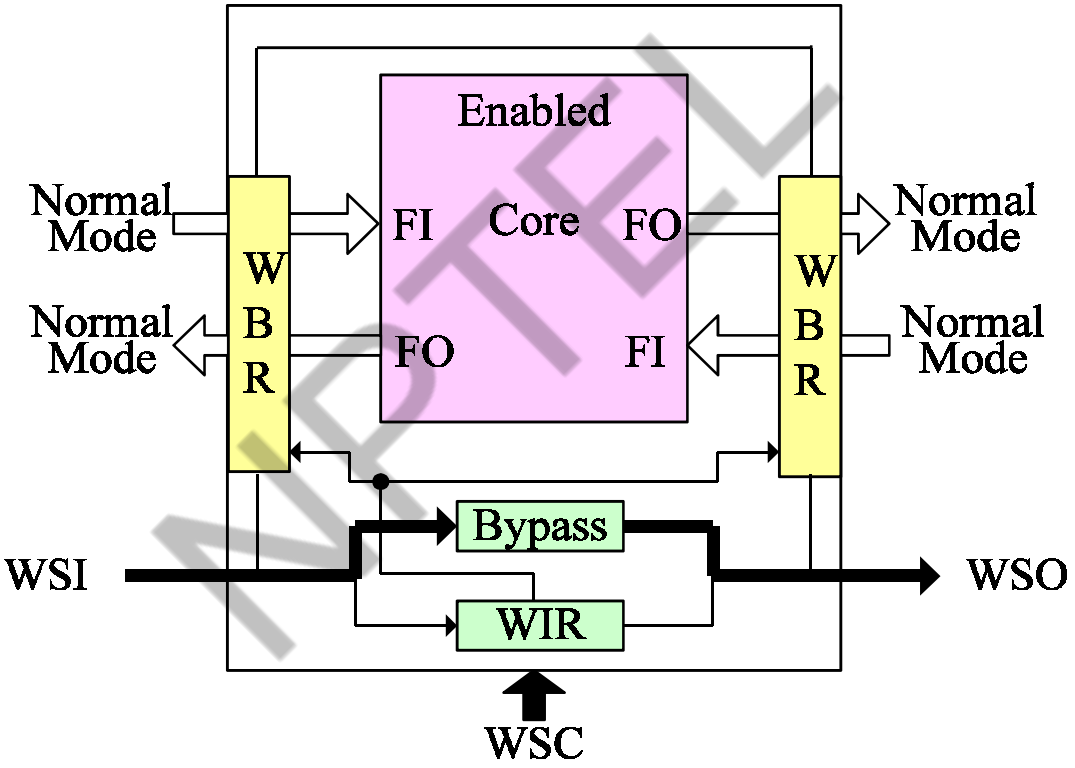
Example 10.1 - WIR Interface of WBY, WBR WDR(s) and CDR(s)



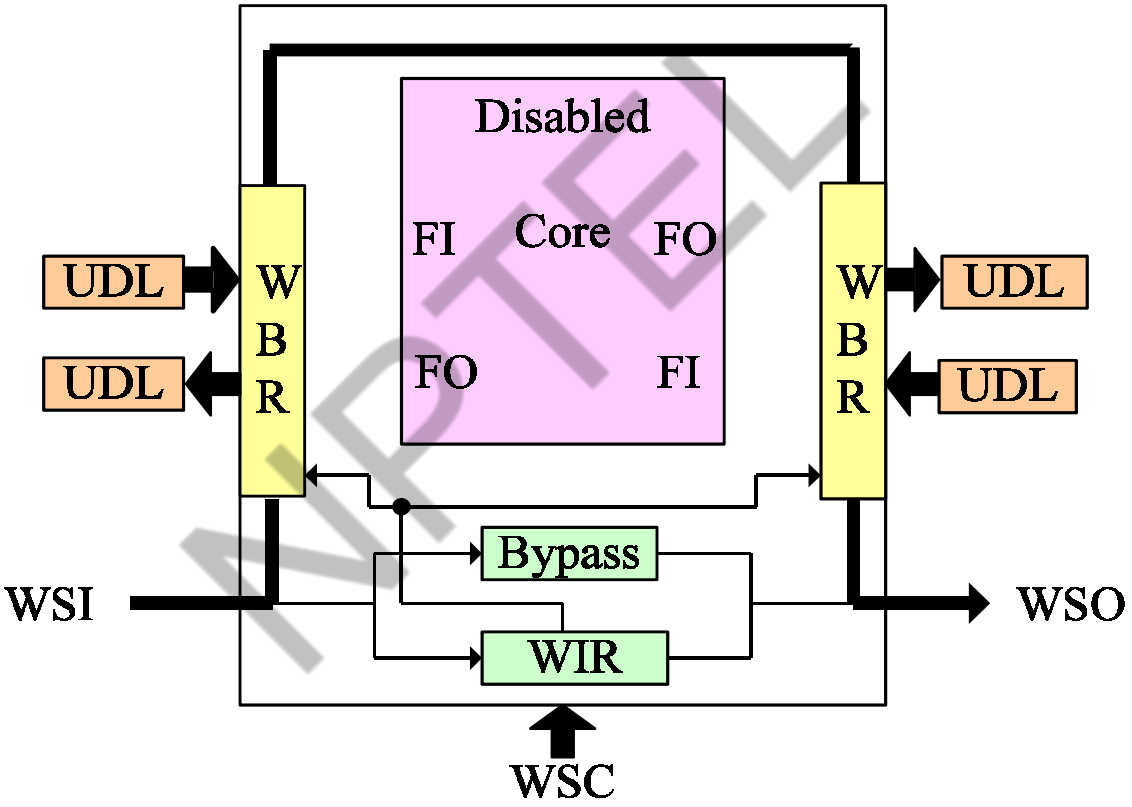
Example 10.2 - Schematic Diagram of WBC WC_SD2_CIO



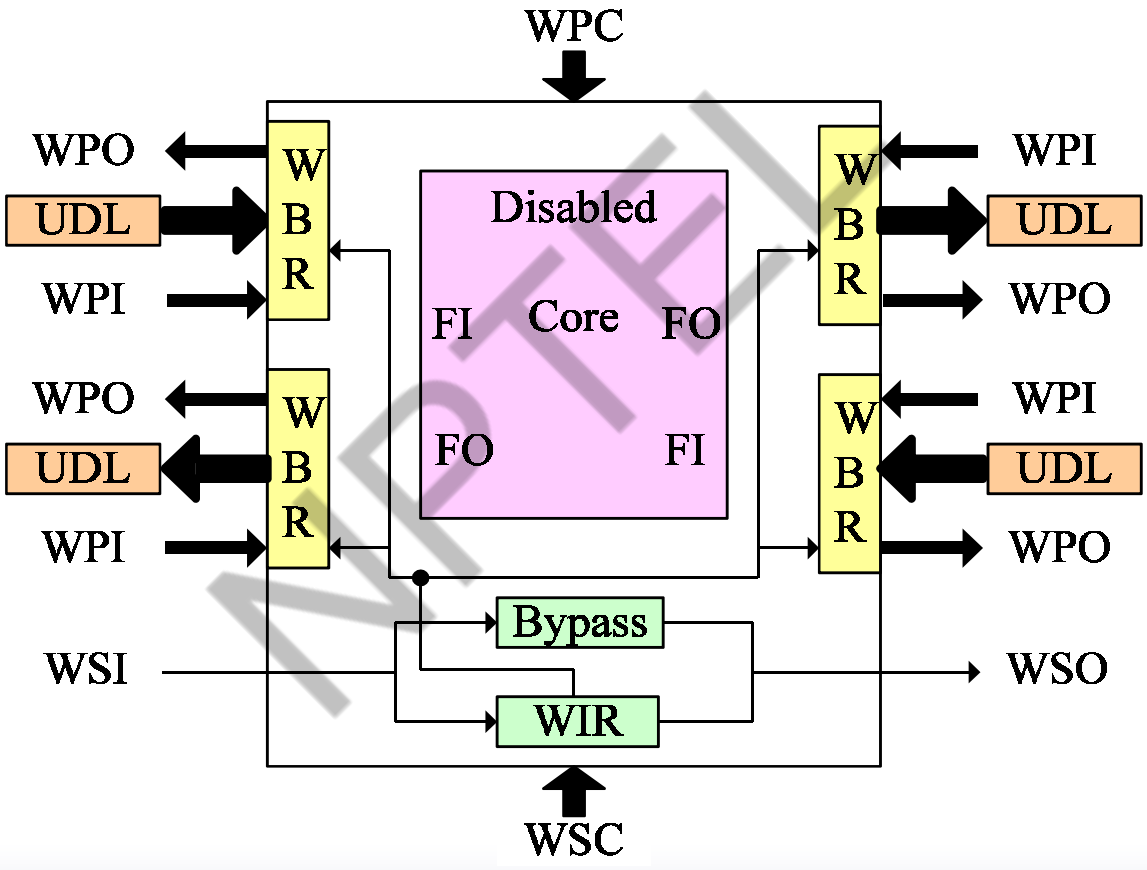
WS_BYPASS Instruction



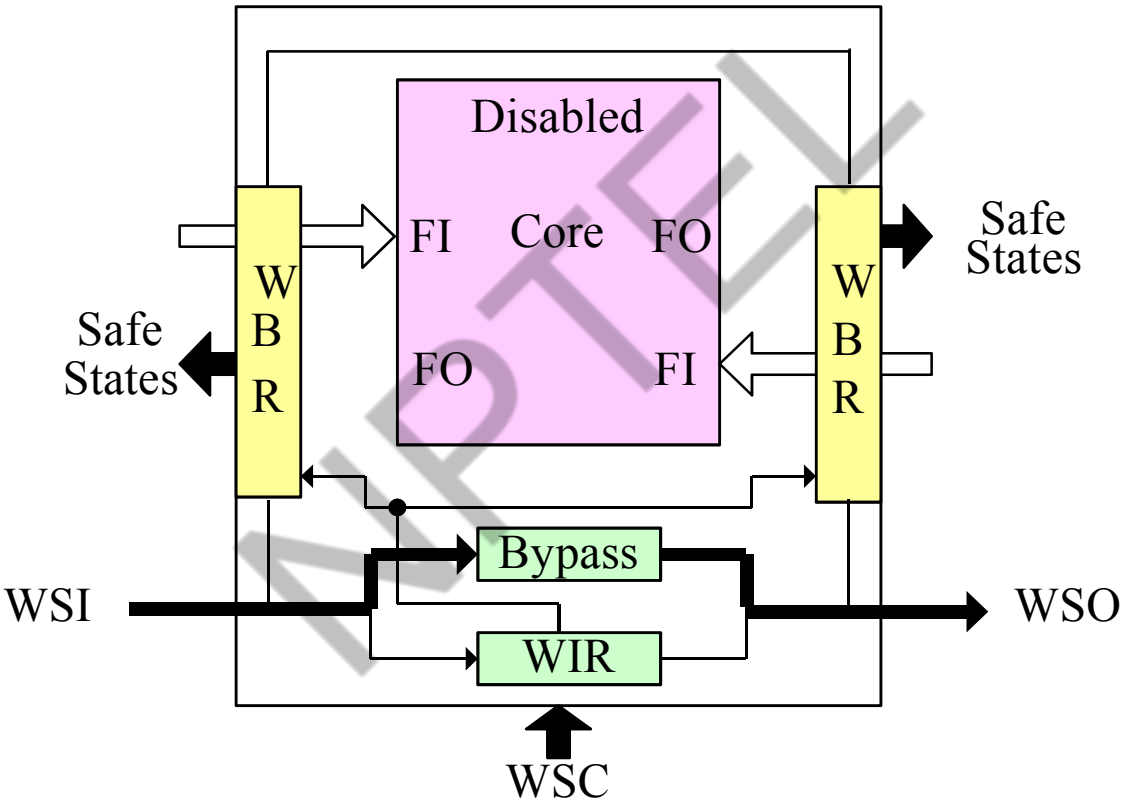
WS_EXTEST Instruction



WP_EXTEXT Instruction

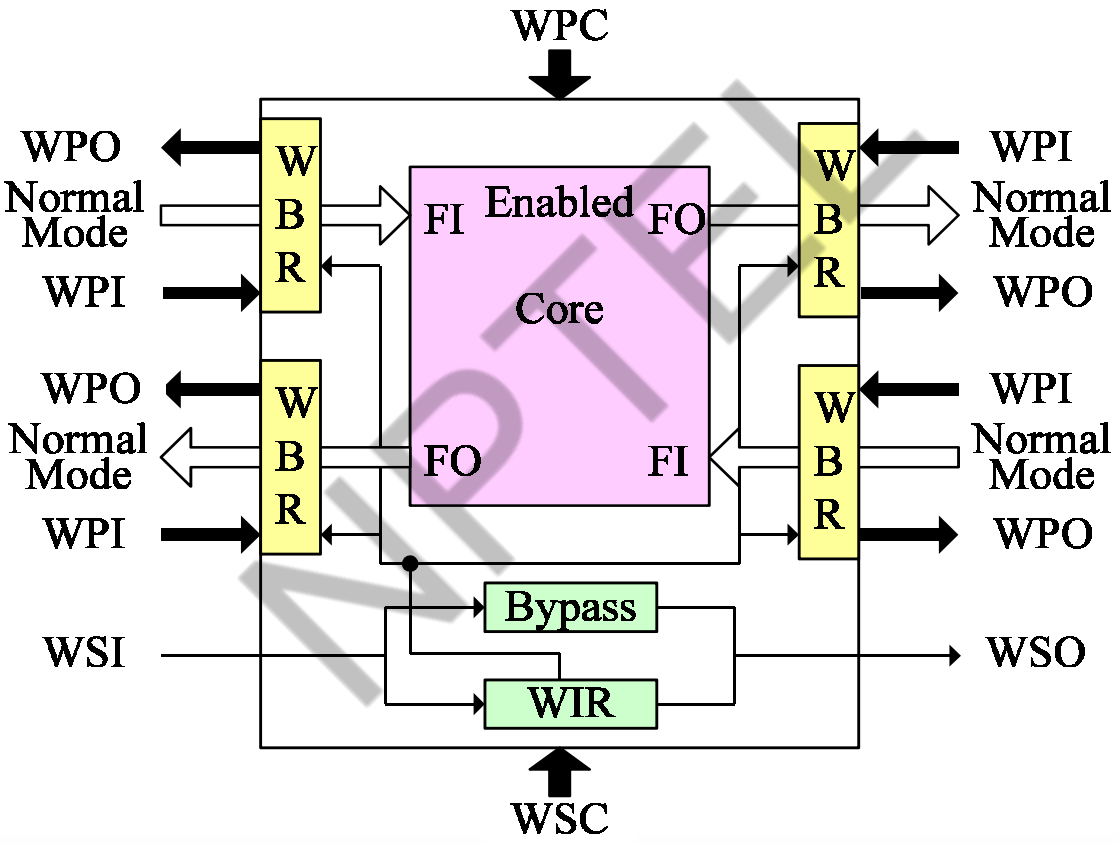


WS_SAFE Instruction

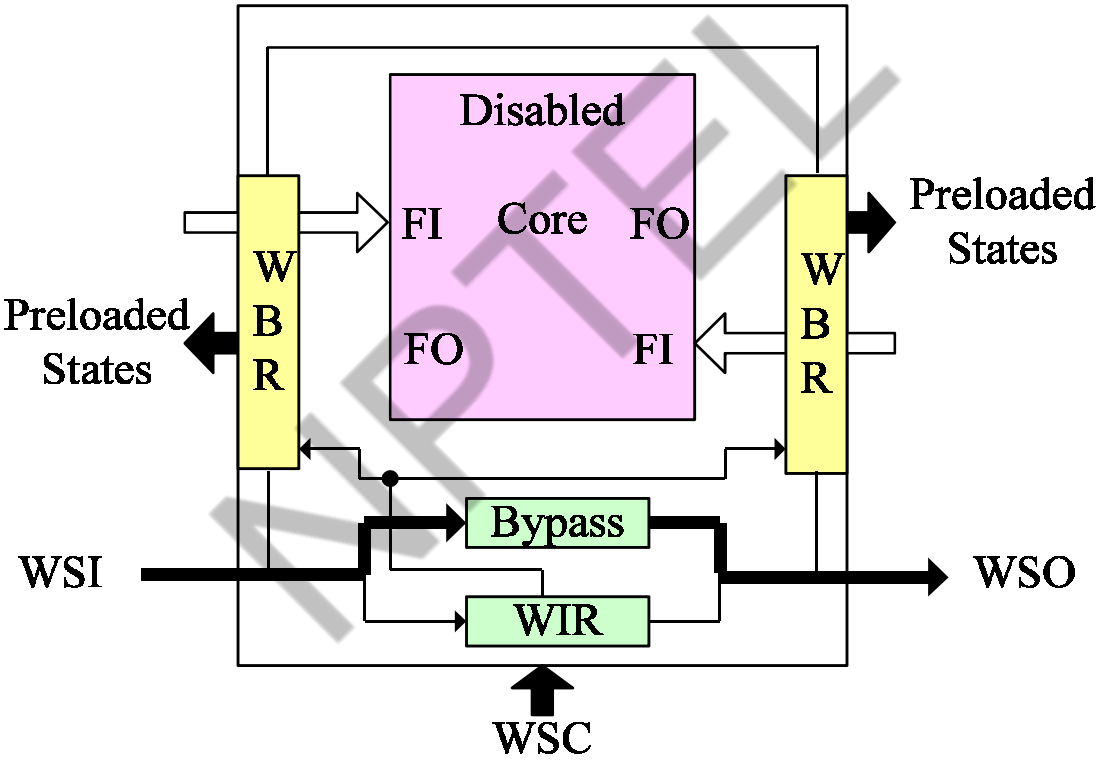


Lecture 44

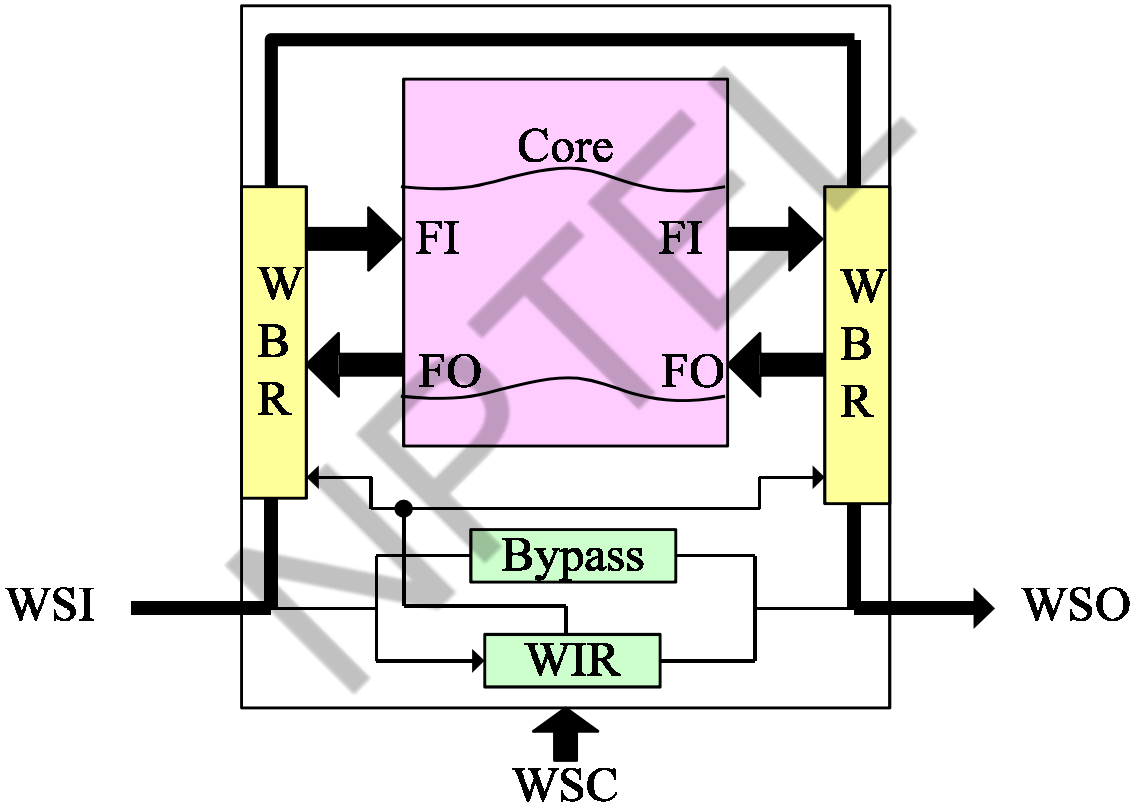
WS_PRELOAD Instruction



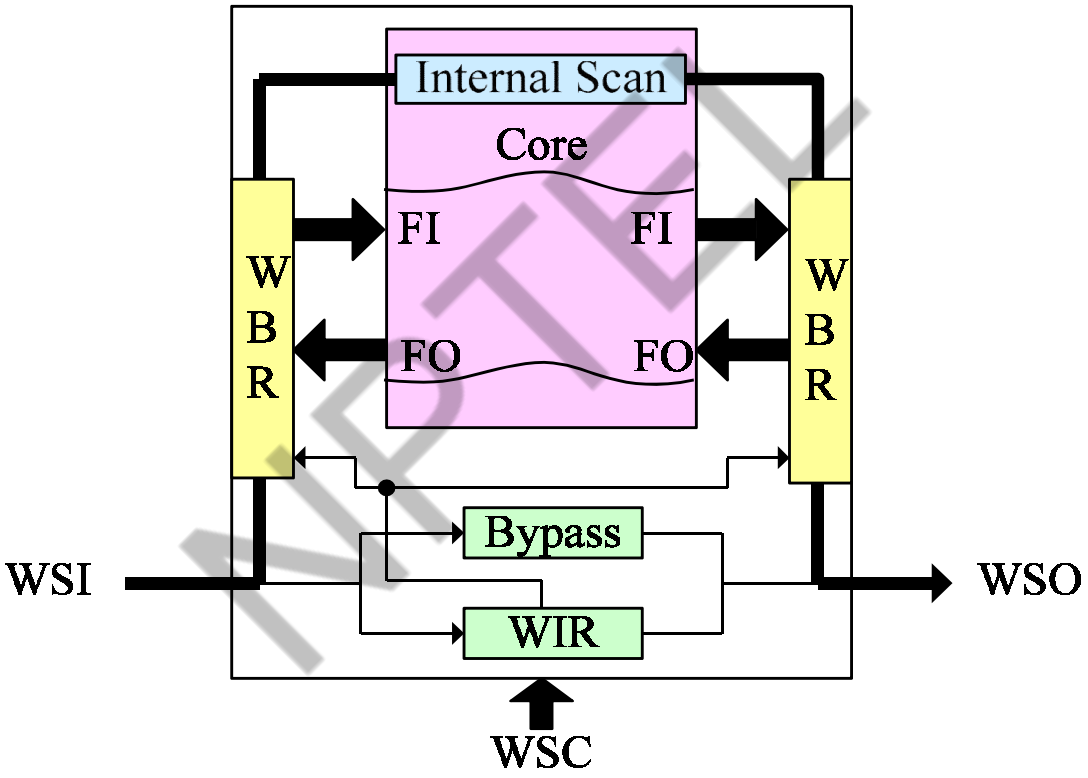
WP_PRELOAD Instruction



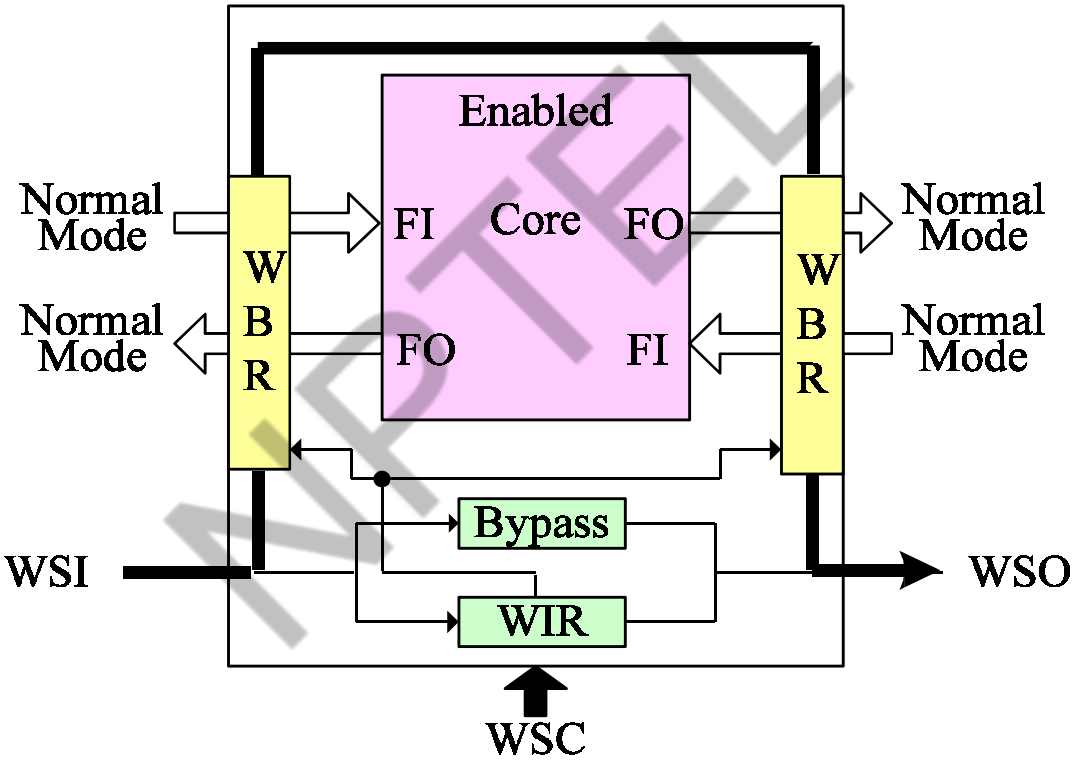
WS_CLAMP Instruction



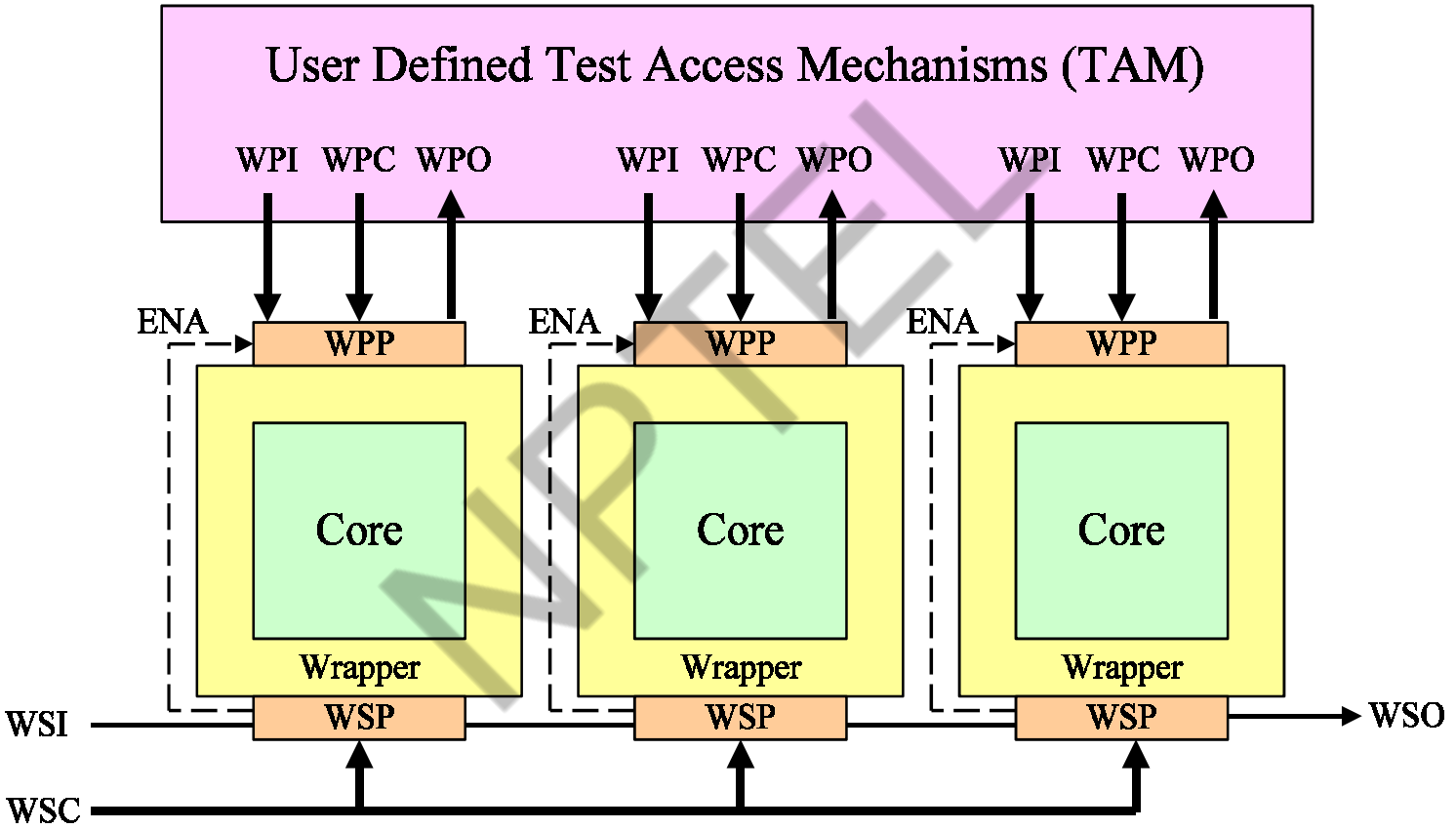
WS_INTEST Instruction



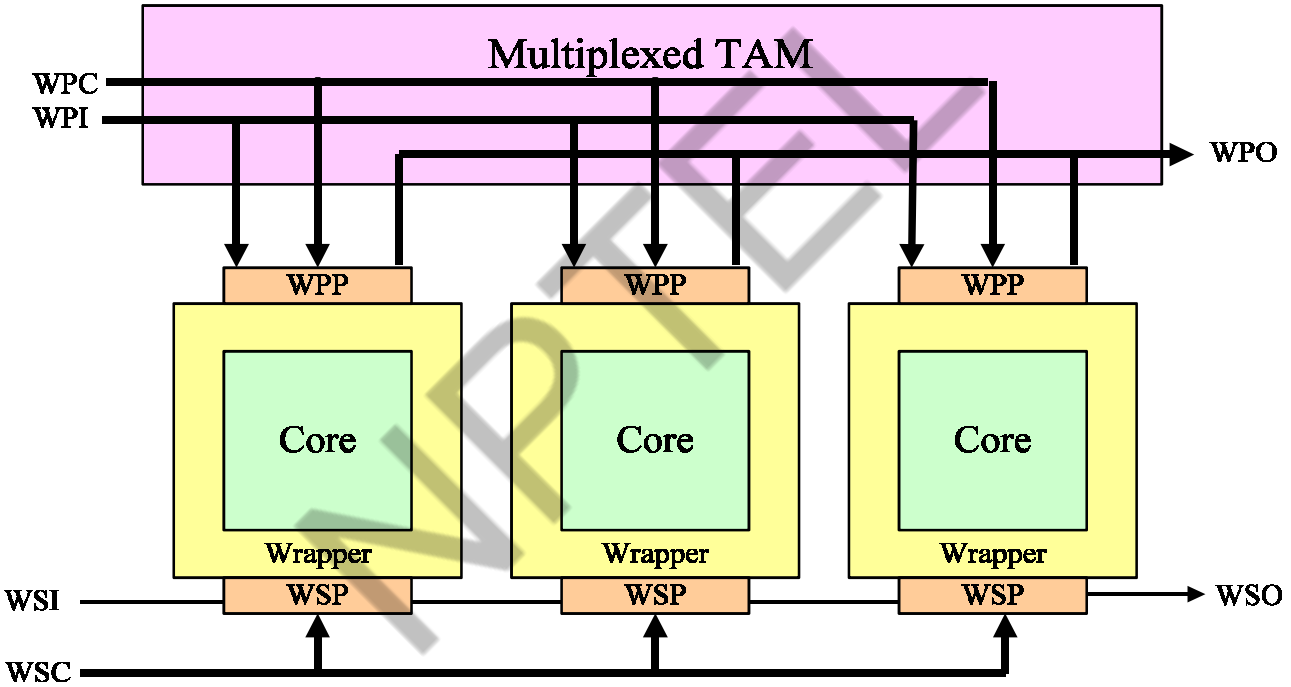
WS_INTEST_SCAN Instruction



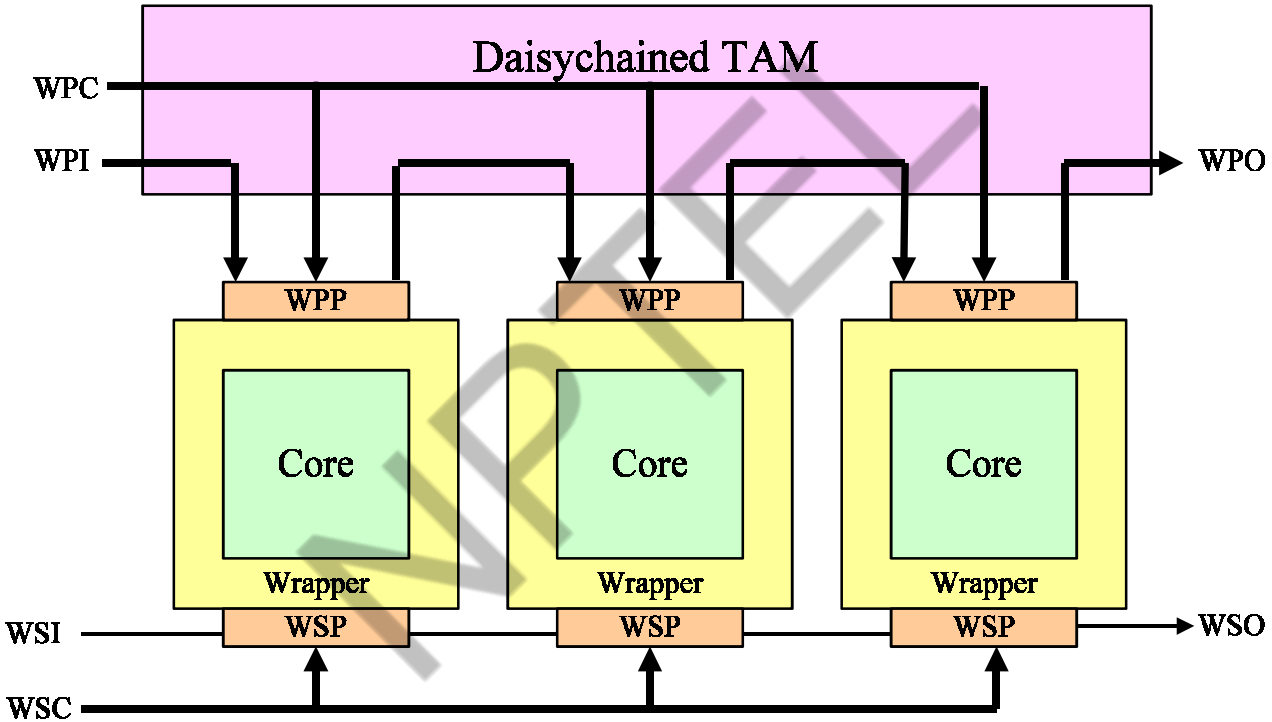
General Parallel TAM Structure



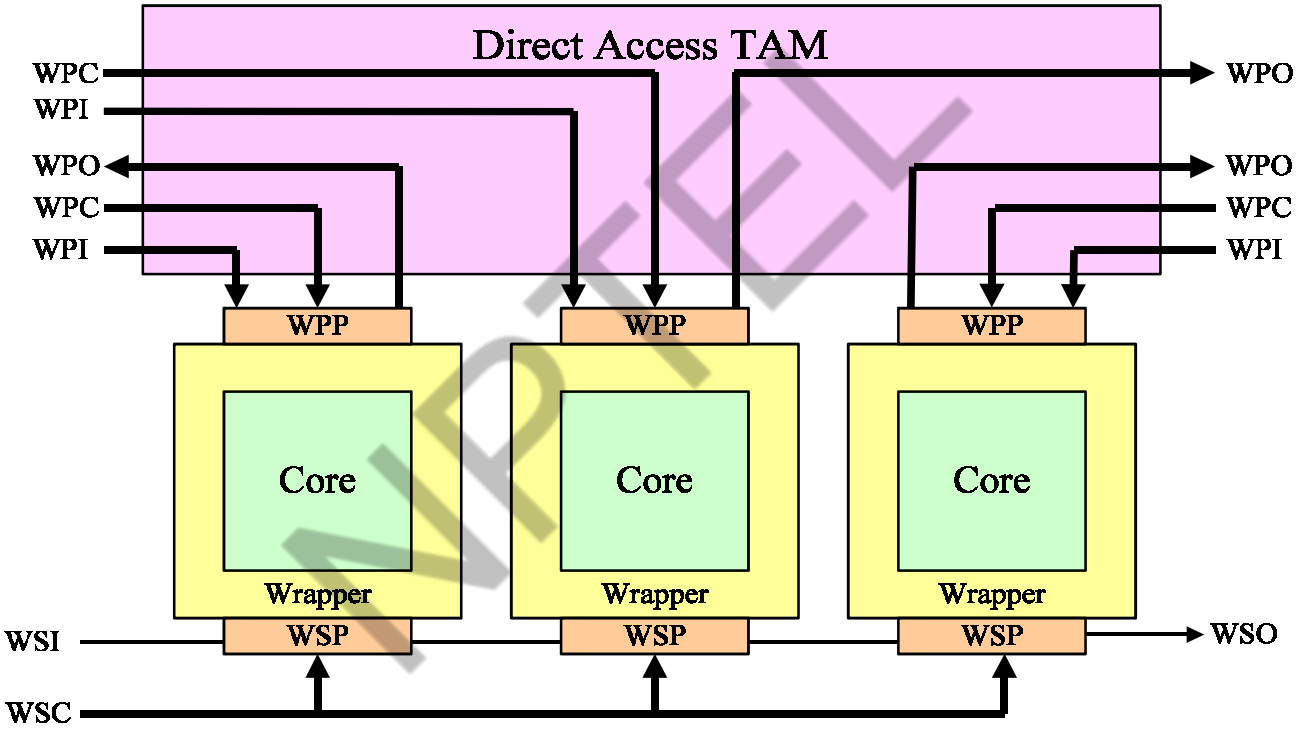
Multiplexed TAM Architectures



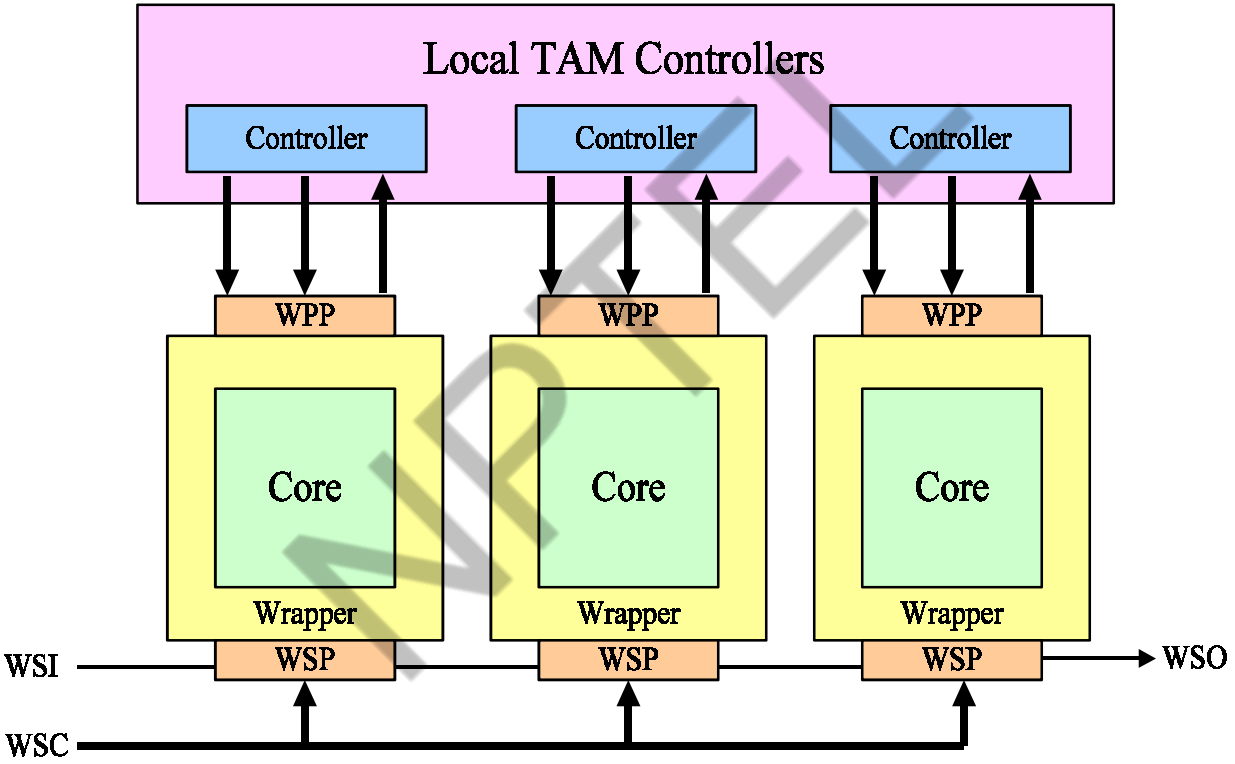
Daisy chained TAM Architecture



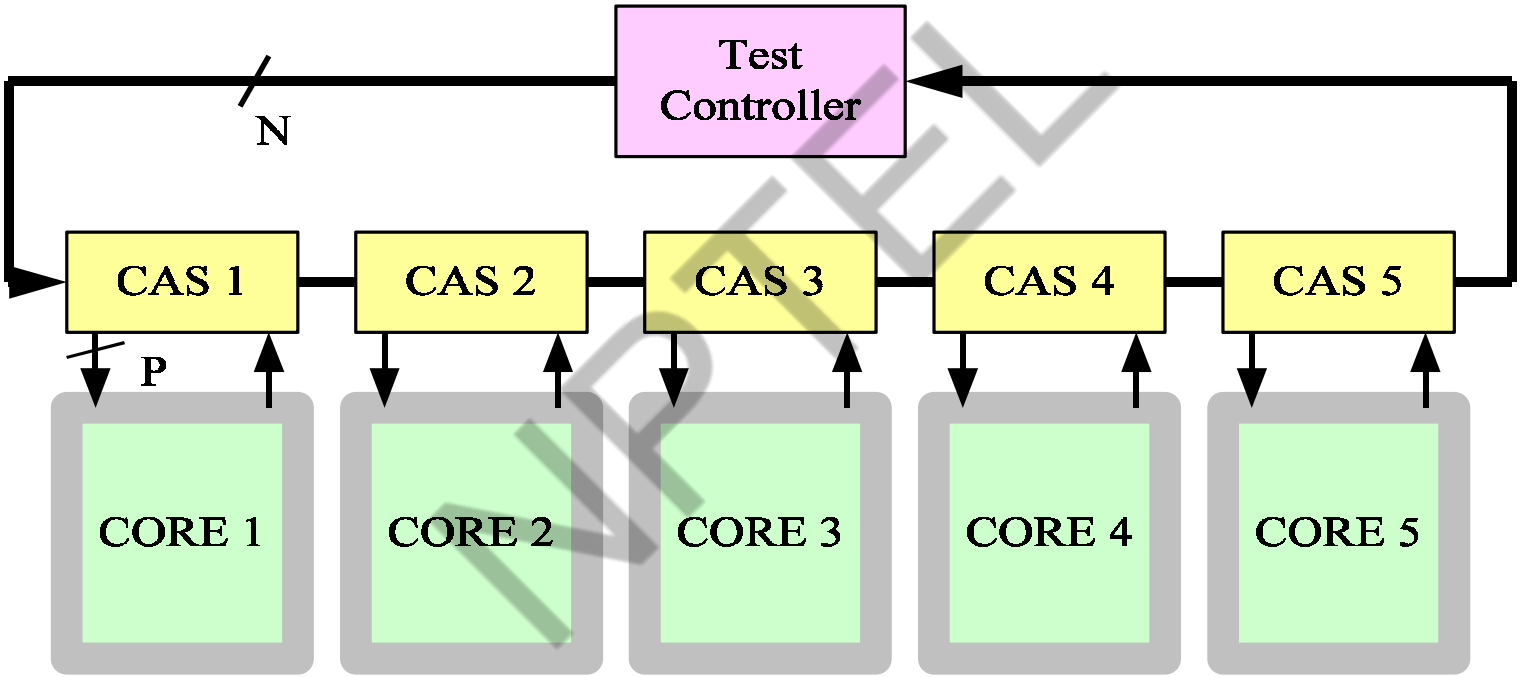
Direct Access TAM Architectures



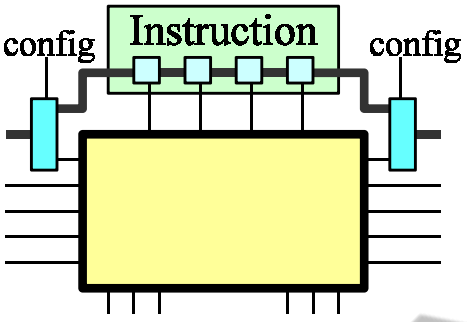
Local Controller TAM Architectures



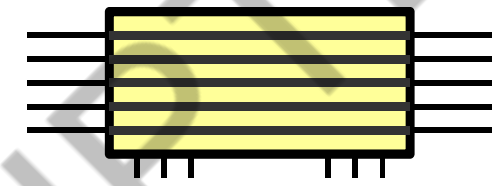
Core Access Switch (CAS) Architecture



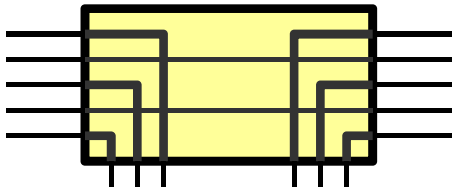
Different Functional Modes of CAS



Configuration mode

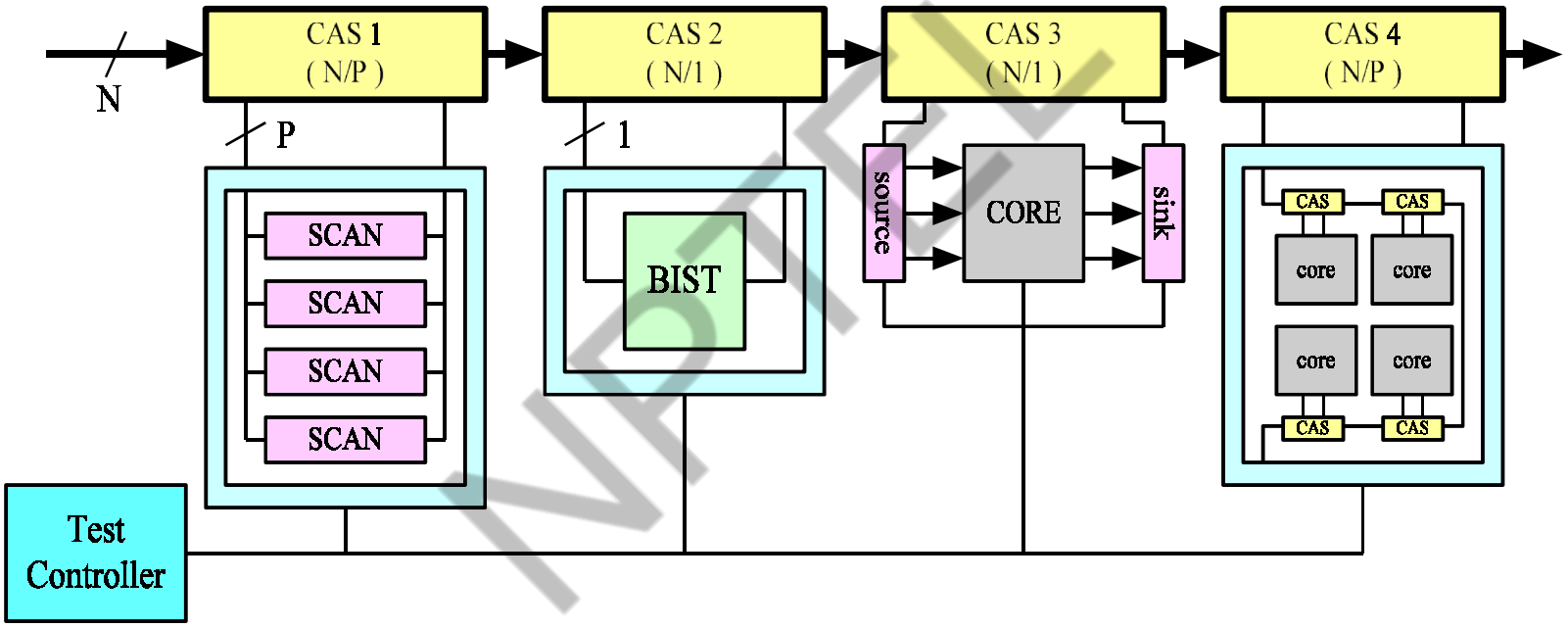


Bypass mode

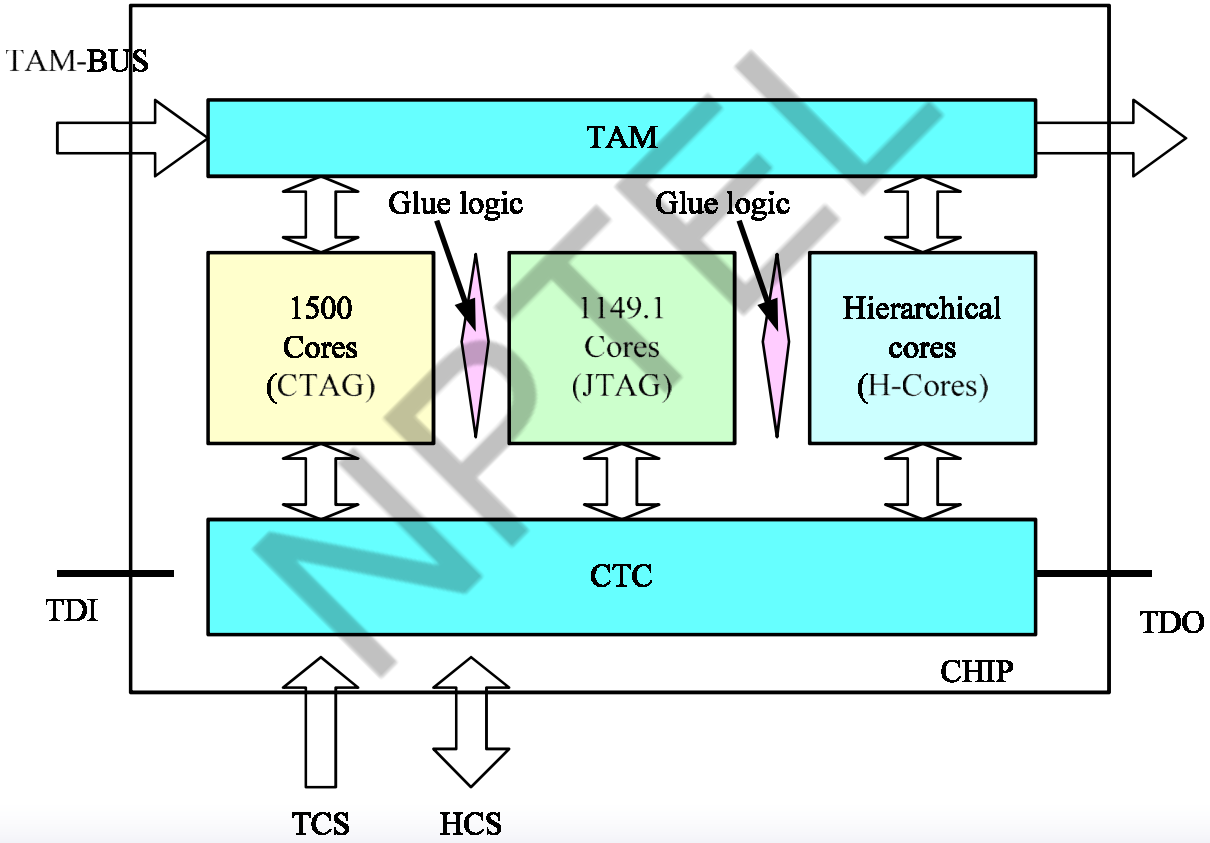


Test mode

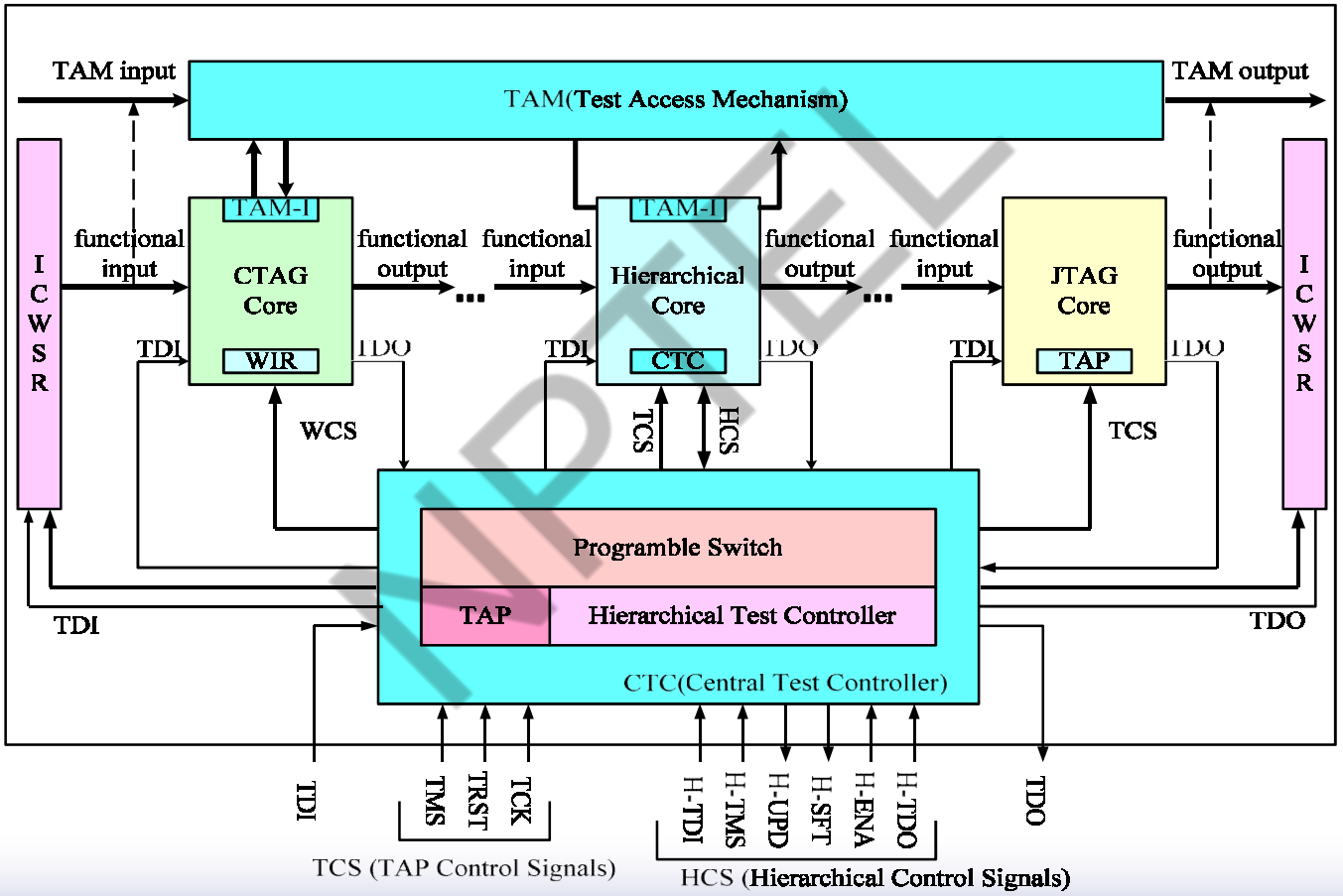
Various Types of Test Supporting Using CAS Structure



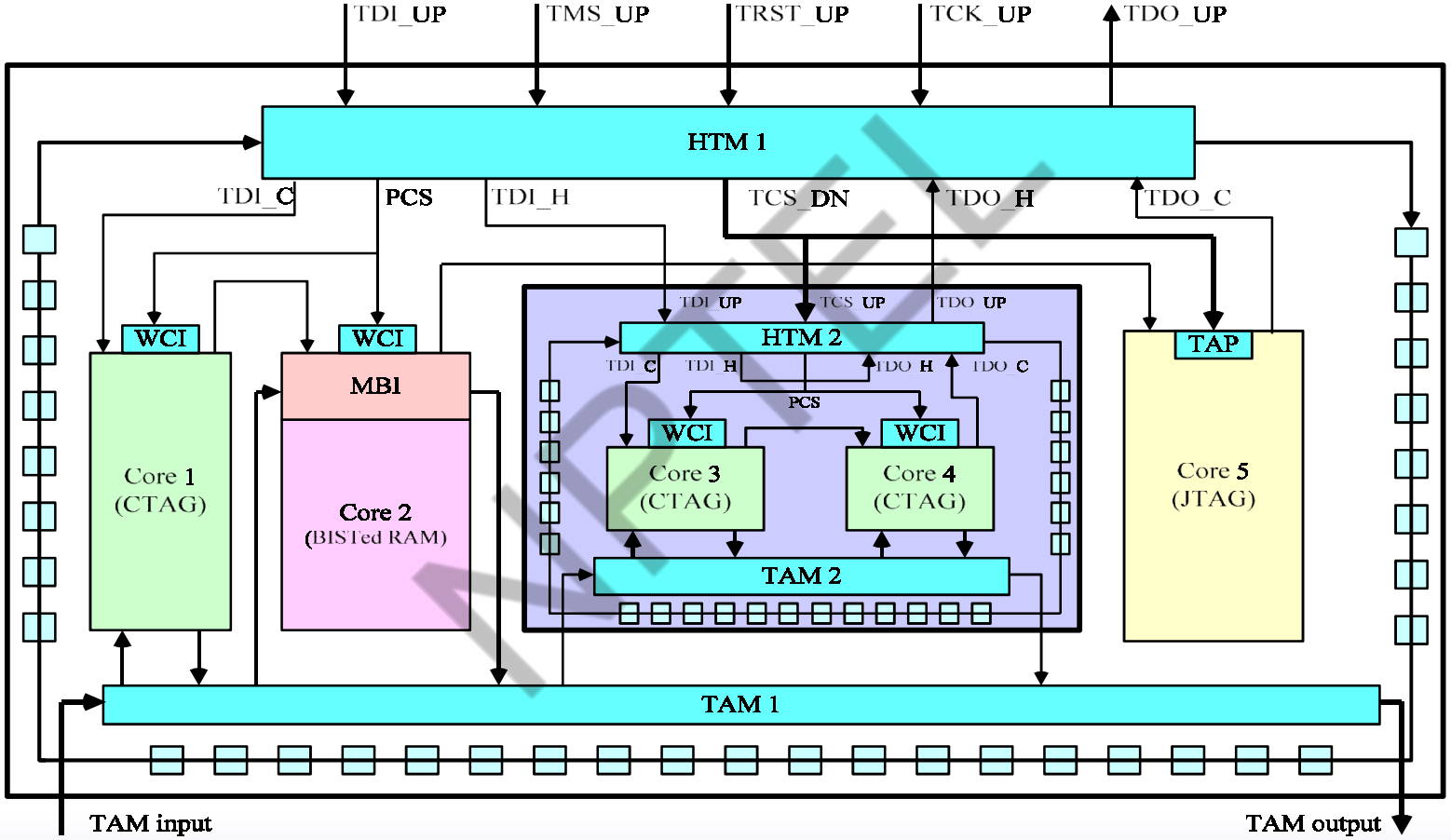
A Hierarchical Test Architecture Supporting Plug & Play Feature



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)

Lecture 45

System/Network-on-Chip Test Architectures

Introduction

Introduce basic and advanced architectures for:

- System-on-Chip (SOC) Testing
- Network-on-Chip (NOC) Testing

Further focus on:

- Testing on On-Chip Networks
- Design and Test Practices in Industry

Introduction to SoC Testing

SoC testing is a composite test comprised of individual tests for each core, user-defined logic (UDL) tests, and interconnect tests.

To avoid cumbersome format translation for IP cores, SoC and core development working groups such as *virtual socket interface alliance (VSIA)* have been formed to propose standards.

IEEE 1500 standard has been announced to facilitate SoC testing.

IEEE 1500 specifies interface standard which allows cores to fit quickly into *virtual sockets* on SoC.

Core vendors produce cores with an uniform set of interface features. SoC integration is simplified by plugging cores into standardized sockets.

Challenges of SoC Testing

Generally, core users cannot access core net-lists and insert design-for-testability circuits. Core users rely on test patterns supplied by core vendors.

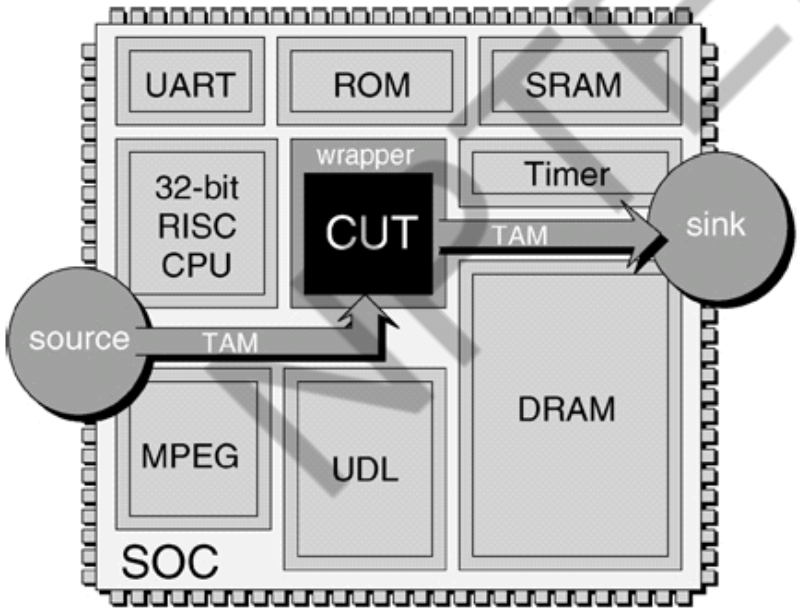
Care must be taken to make sure that undesirable test patterns and clock skews are not introduced into test streams.

Cores are often embedded in several layers of user-defined or other core-based logic, and are not always directly accessible from Chip I/Os.

Test data at I/Os of an embedded core might need to be translated into a format for application to the core.

Conceptual Architecture of Embedded Core-Based SoC Testing

Mainly, three structural elements are required. They are test pattern source and sink, test access mechanism (TAM), and core test wrapper.



More Test Challenges

Once test data transport mechanism (TAM) and test translation mechanism (test wrapper) are determined, major challenge for system integrator is *test scheduling*.

Test scheduling must consider several conflicting factors: (a) SoC test time minimization, (b) resource conflicts due to sharing of TAMs and on-chip BIST engines, (c) precedence constraints among tests, and (d) power constraints.

Finally, analog and mixed-signal core testing must be dealt with. Testing analog and mixed-signal cores is challenging because their failure mechanisms and test requirements are less known than digital cores.

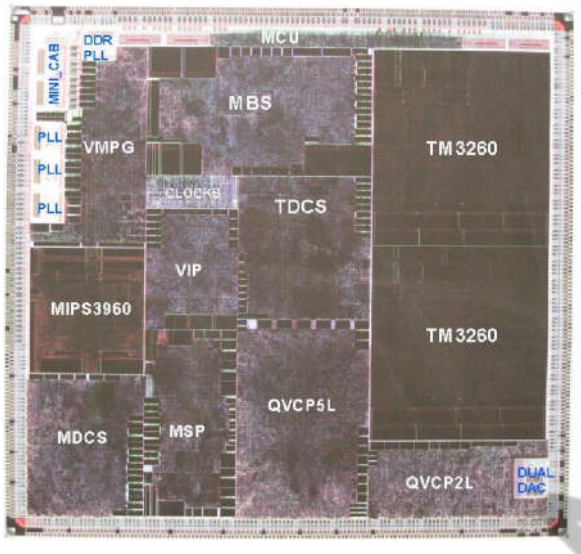
Outline for SoC Testing

- Introduction to testing
  - Motivation for modular testing of SOCs
- Wrapper design
  - IEEE 1500 standard, optimization
- Test access mechanism design and optimization
- Test scheduling
- Exploiting port scalability to test embedded cores at multiple data rates
  - Virtual TAMs
  - Matching ATE data rates to scan frequencies of embedded cores
- Conclusions

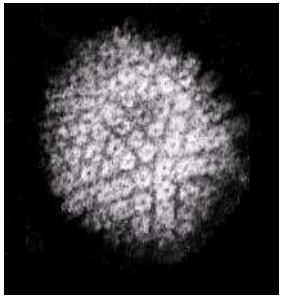
System Chips

~50 million transistors

1 cm



1 cm



Viruses: Size 300 nm

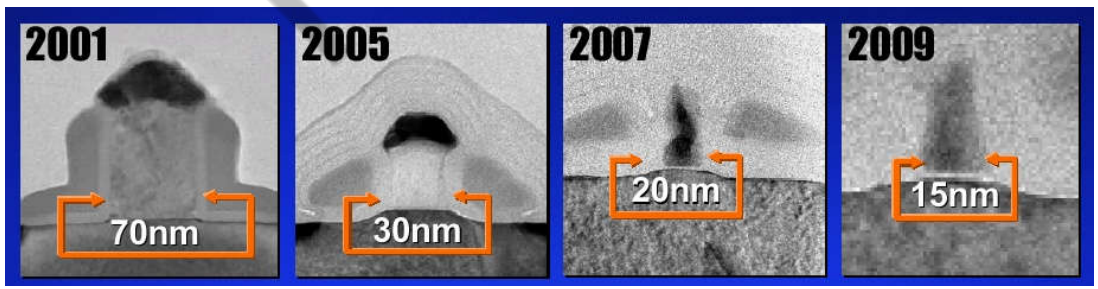
Intel Itanium (2006): 1.7 billion transistors

EE Times:

Intel crafts transistor with 20-nm gate length

David Lammers, David Lammers

(06/11/2001)



Motivation for Testing: XBox 360 Technical Problems

The "Red Ring of Death": Three red lights on the Xbox 360 indicator, representing "general hardware failure" (http://en.wikipedia.org/wiki/3_Red_Lights_of_Death)
The Xbox 360 can be subject to a number of possible technical problems. Since the Xbox 360 console was released in 2005 the console gained reputation in the press in articles portraying poor reliability and relatively high failure rates.
On 5 July 2007, Peter Moore published an open letter recognizing the problem and announcing 3 years warranty expansion for every Xbox 360 console that experiences the general hardware failure indicated by the three flashing red lights on the console.

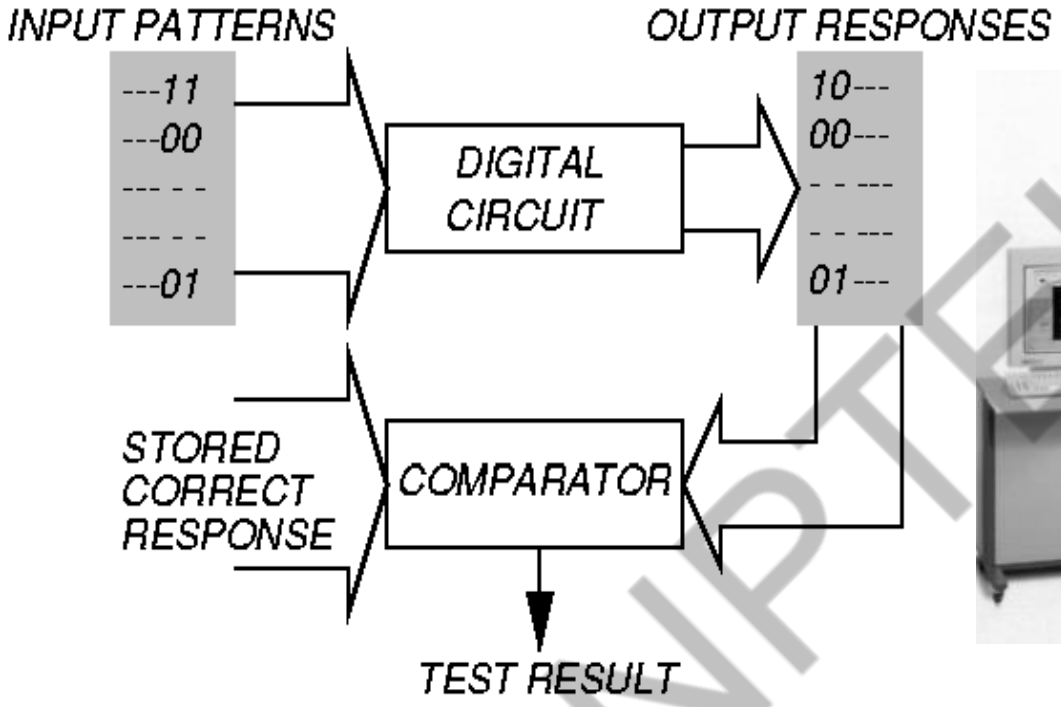
XBox 360 Technical Problems (Cont'd)

July 5, 2007, Xbox issues to cost Microsoft $1 billion-plus. Unacceptable number of repairs leads to company extending warranties.

Matt Rosoff, an analyst at the independent research group Directions on Microsoft, estimates that Microsoft's entertainment and devices division has lost more than $6 billion since 2002.

Lecture 46

Testing Principles



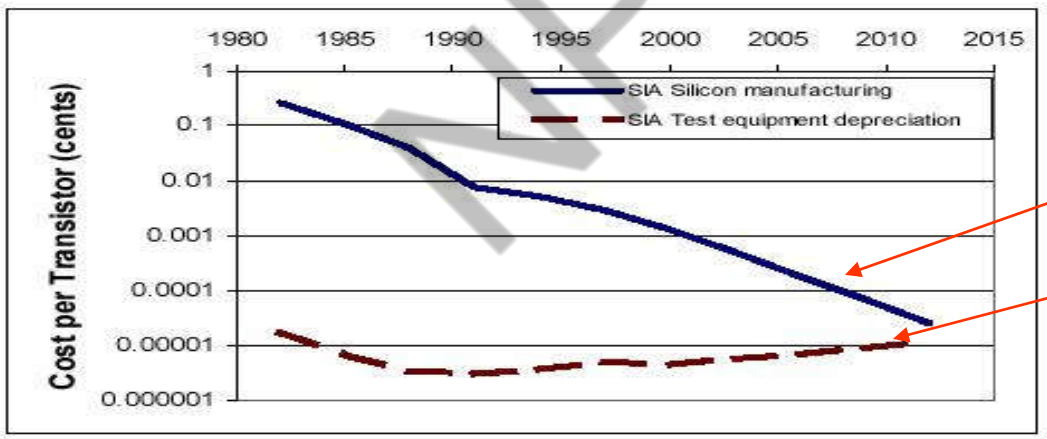
- Screen defective chips (wafer, package)
- Stress test (burn-in)
- Diagnosis: Locate defects, yield learning

- Speed binning
- Design-for-testability (DFT) typically used
- Test generation, scan design

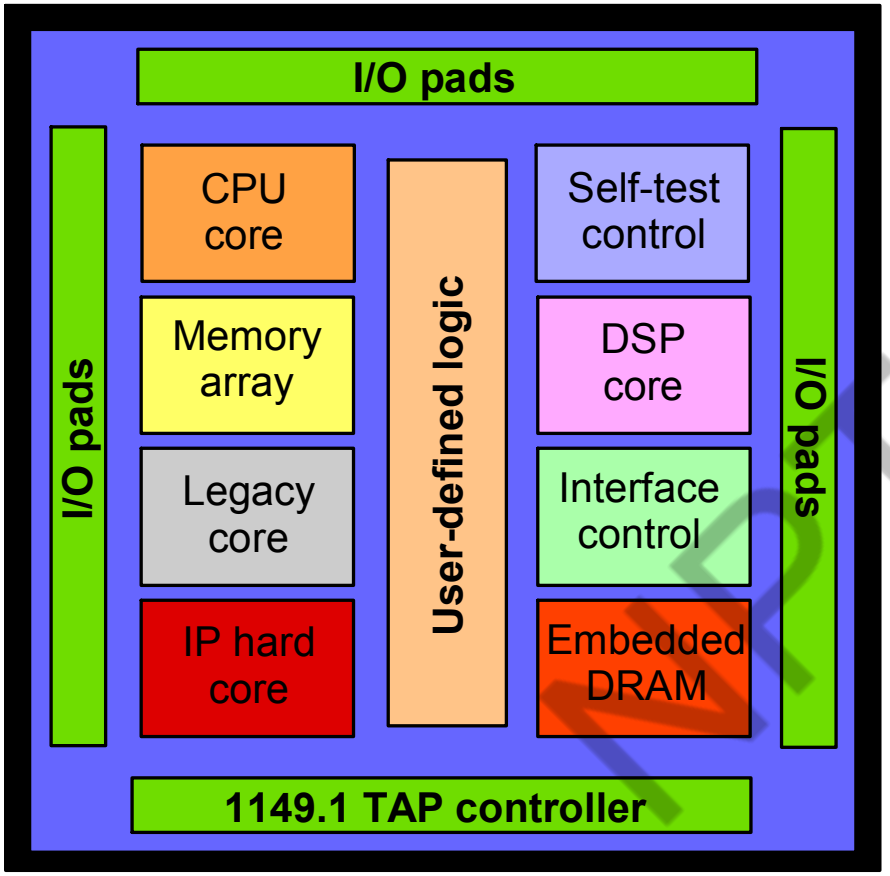
Motivation for Core-Based SOC Testing

System-on-chip (SOC) integrated circuits based on embedded intellectual property (IP) cores are now commonplace

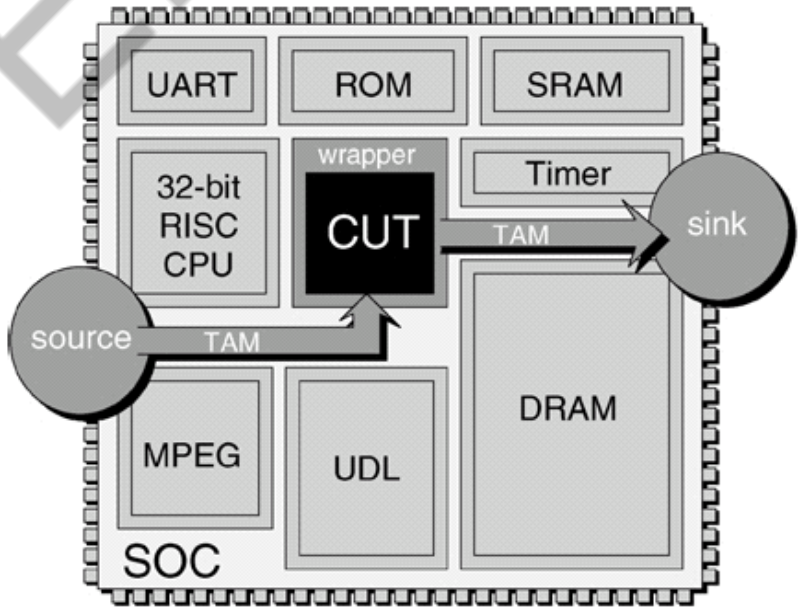
- SOCs include processors, memories, peripheral devices, IP cores, analog cores
- Low cost, fast time-to-market, high performance, low power
- Manufacturing test needed to detect manufacturing defects



System-on-Chip (SOC)



- Test access is limited
- Test sets must be transported to embedded logic
- High test data volume & test time



NXP Nexperia™ PNX8550 SOC: 338,839 flip-flops, 274 embedded cores, 10M logic gates, 40M logic transistors!

Cost of Test

"The emergence of more advanced ICs and SOC semiconductor devices is causing test costs to escalate to as much as 50 percent of the total manufacturing cost." [Kondrat 2002]

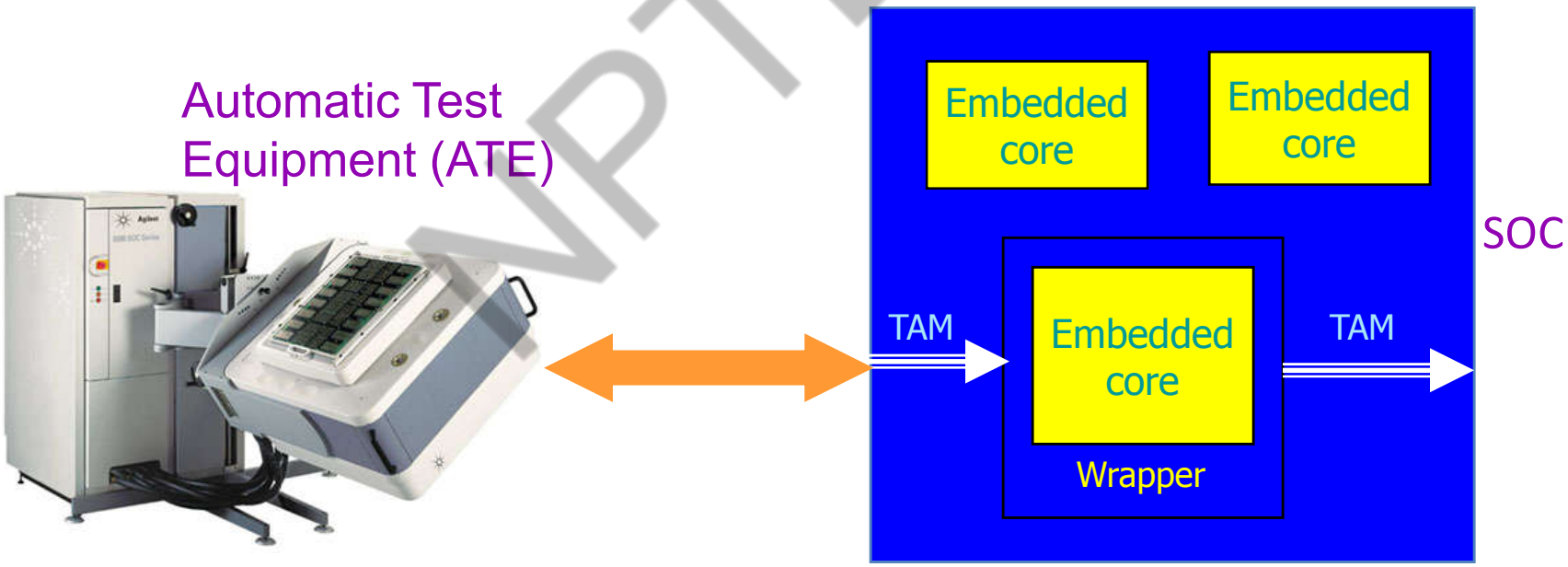
"As a result, semiconductor test cost continues to increase in spite of the introduction of DFT, and can account for up to 25-50% of total manufacturing cost". [Cooper 2001]

"Test may account for more than 70% of the total manufacturing cost - test cost does not directly scale with transistor count, dies size, device pin count, or process technology." [ITRS'03]

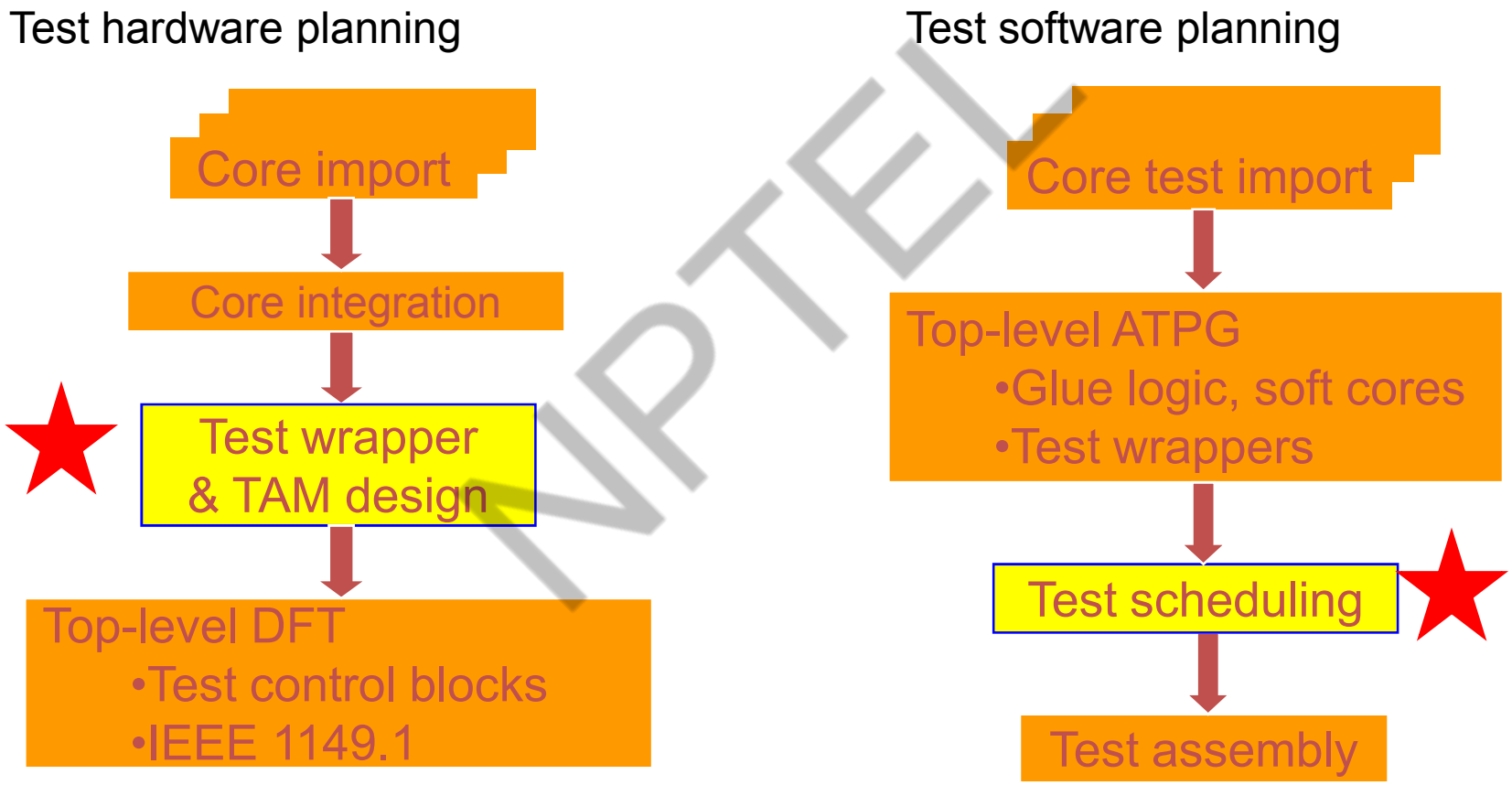
Modular Testing

- Test embedded cores using patterns provided by core vendor (test reuse)
- Test access mechanisms (TAMs) needed for test data transport: TAMs impact test time and test cost
- Test wrappers translate test data supplied by TAMs
- TAM optimization, test scheduling, and test compression are critical

  Test data volume and testing time in 2010 will 30X that for today's chips [ITRS'05]



Test Planning
Optimizing Test Access to Cores and Scheduling Test Hardware



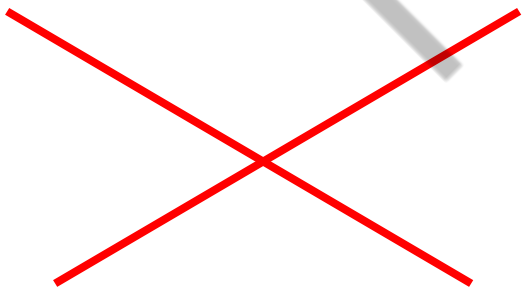
IEEE 1500 Core Test Standard

Goals

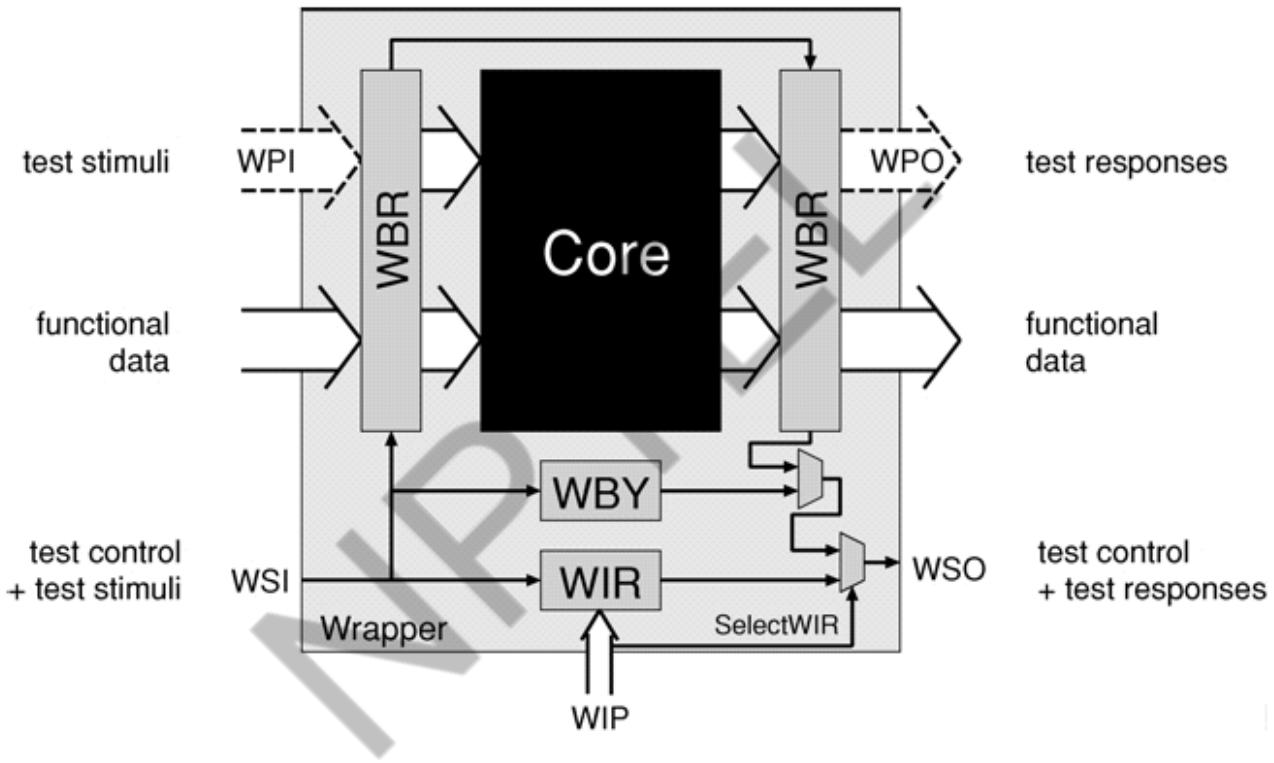
- Define test interface between core and SOC
- Core isolation
- Plug-and-play protocols

Scope

- Standardize core isolation protocols and test modes
- TAM design
- Type of test to be applied
- Test scheduling



IEEE 1500 Wrapper



Wrapper Modes: (1) Normal; (2) Serial Test; (3) 1-N Test; (4) Bypass; (5) Isolation; (6) Extest [Marinissen 2002]

Wrapper Boundary Cells