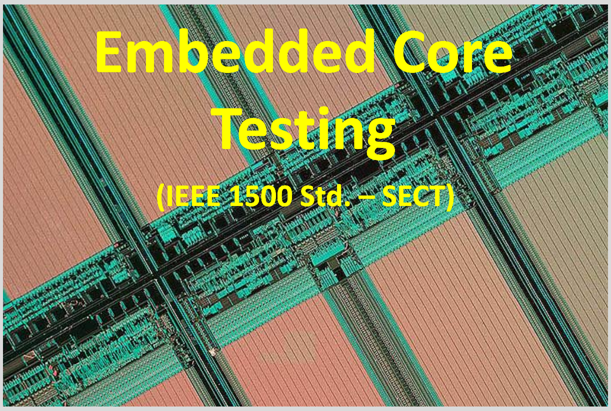
CMOS INTEGRATED CIRCUIT DESIGN TECHNIQUES

University of Ioannina



Embedded Core Testing

(IEEE 1500 Std. – SECT)

Dept. of Computer Science and Engineering

Y. Tsiatouhas

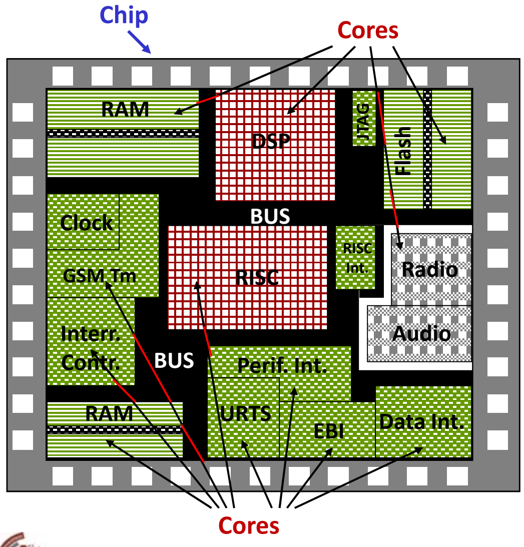
CMOS Integrated Circuit Design Techniques

Overview

1. *Basic SECT architecture*
2. *The Wrapper Interface Port (WIP)*
3. *SECT registers*
4. *Instruction set*
5. *Parallel test access mechanism*

VLSI Systems
and Computer Architecture Lab

System-on-Chip (SoC)

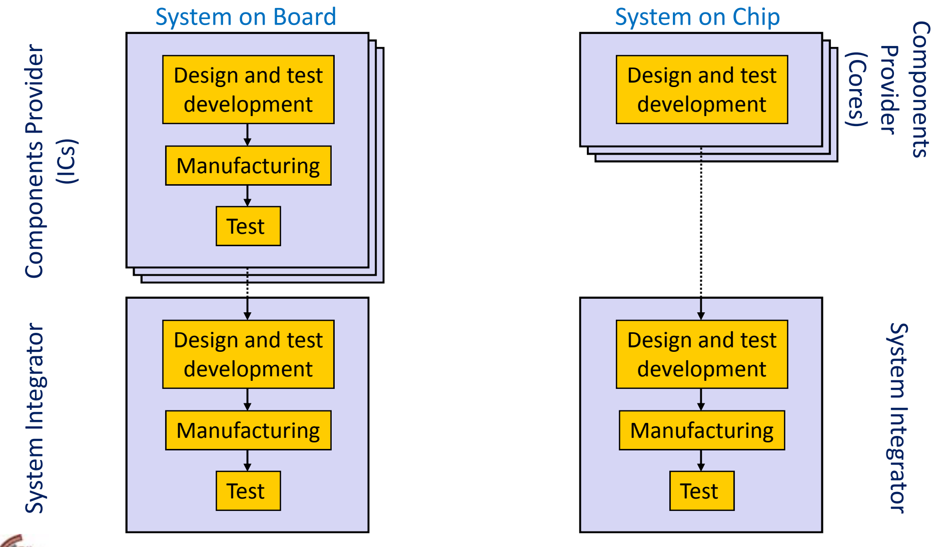


A main design paradigm in modern CMOS technologies is the development of Systems-on-a-Chip (SoCs) where reusable cores (digital, analog, memory) are embedded in a chip.

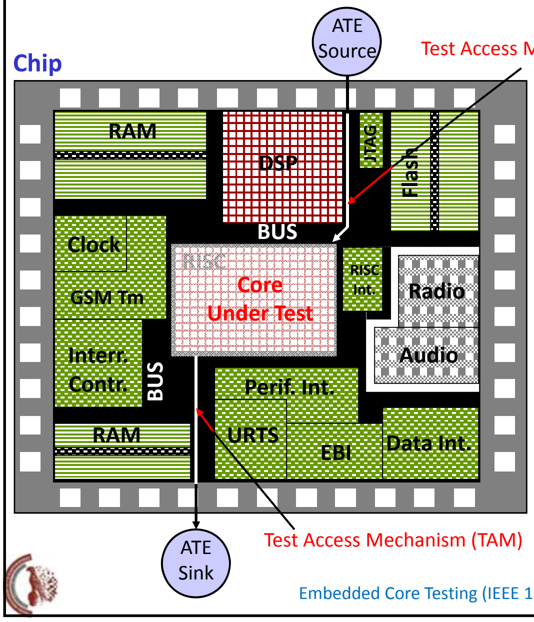
The used core designs are characterized as:

- Soft (RTL code).
- Firm (netlist).
- Hard (layout).

Design Paradigms


System on Board
System on Chip
Components Provider (ICs)
Components Provider (Cores)
Design and test development
Manufacturing
Test
Design and test development
Design and test development
Manufacturing
Test
Design and test development
Manufacturing
Test
System Integrator
System Integrator

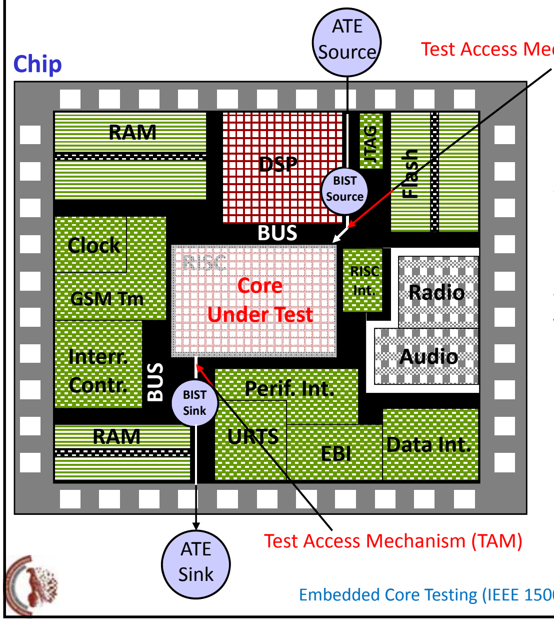
Automatic Test Equipment Support



Automatic Test Equipment: Hardware equipment for the testing and diagnosis of integrated circuits under the control of proper software. An ATE provides test data (*test source*) and monitors the test responses (*response sink*)

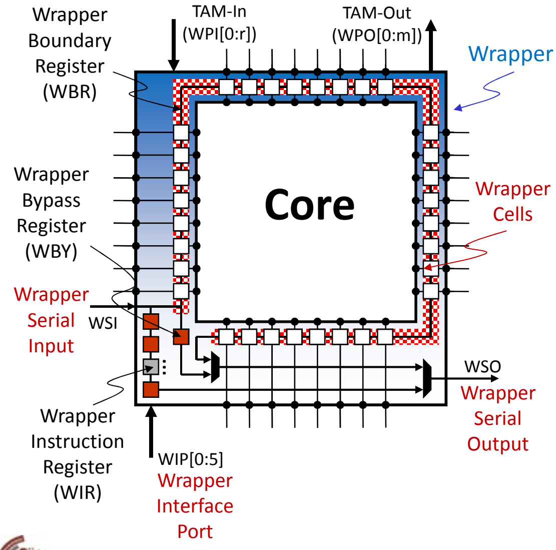
Test Access Mechanism: Circuitry embedded in the chip to support the transfer of test data between the ATE and the chip.

Automatic Test Equipment + BIST



Built-In Self Test (BIST) circuits can be exploited for embedded test data generation and test response monitoring. Thus, the ATE complexity is drastically reduced and the same stands for the required bandwidth to transfer test data in/out the chip.
The ATE activates the BIST circuitry and receives the final test decision.

Basic IEEE 1500 Architecture (I)

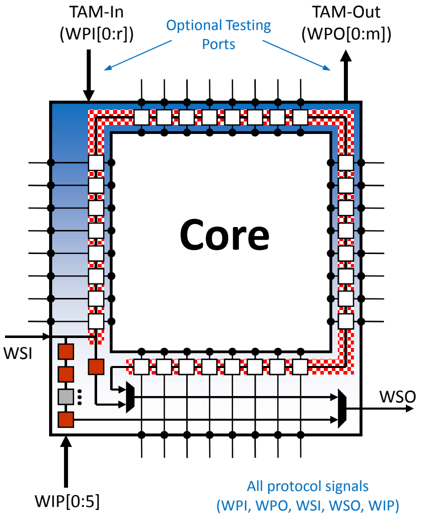


The IEEE 1500 Embedded Core Testing standard supports the testing process of embedded cores in a chip, according to a common DFT and test application methodology.

It consists of a *test wrapper* around a core and provides the following facilities:

- A *Wrapper Interface Port - WIP* with 6 inputs for control signals plus 1 serial test data input and 1 serial test data output.
- A set of registers.

Basic IEEE 1500 Architecture (II)



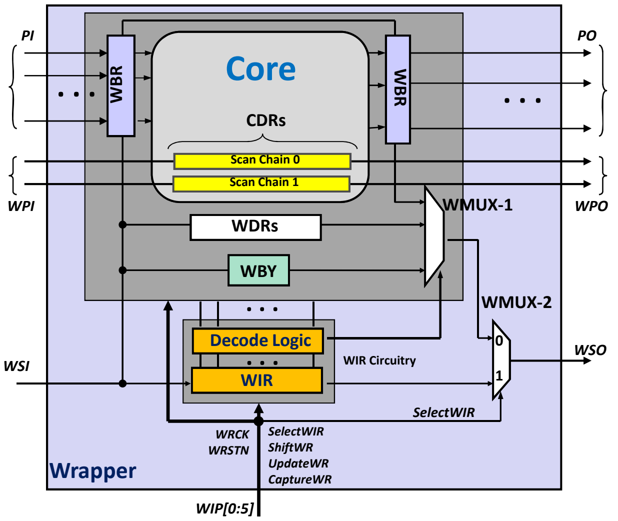
All protocol signals (WPI, WPO, WSI, WSO, WIP) are managed either by the BIST or the ATE

In the normal mode of operation the IEEE 1500 std. circuitry is "transparent".

There are three modes of operation:

- Normal Operation: where the wrapper is transparent.
- Core Internal Testing: where the internal core logic is under test.
- Core External Testing: where the interconnects between the cores and the glue logic are tested.

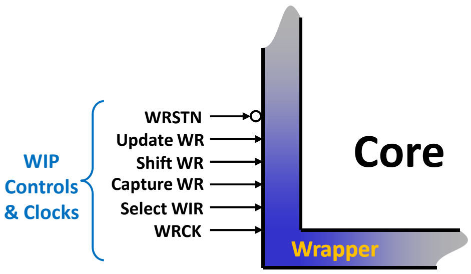
Wrapper Registers



The IEEE 1500 protocol supports the following registers:

- Wrapper Bypass Register (WBY).
- Wrapper Instruction Register (WIR).
- Wrapper Boundary register (WBR).
- Various Data Registers (WDR/CDR).

Wrapper Interface Port – WIP (I)

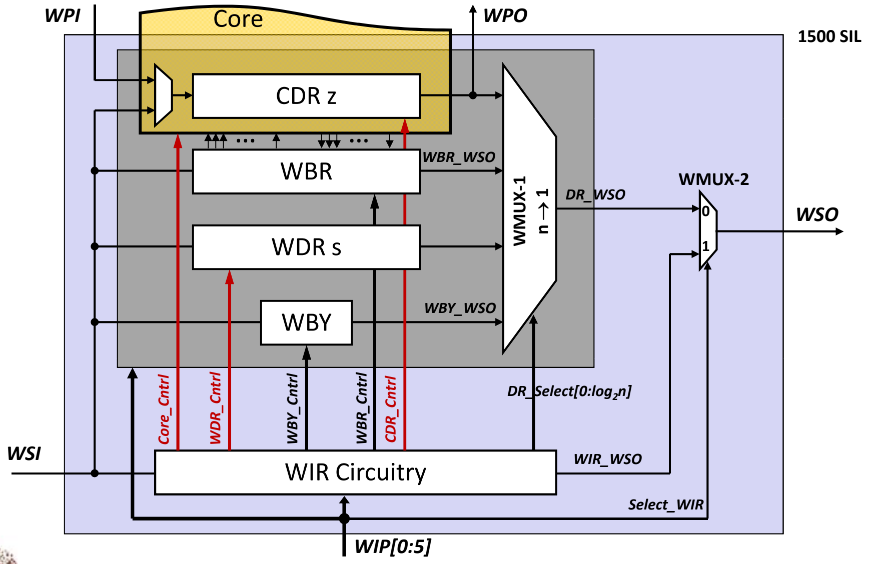


- SelectWIR (*Select Wrapper Instruction Register*): Signal which controls the operation of the WIP port. When active (high) the WIR register is connected between the WSI and WSO pins else another register (WBY, WBR, WDR, CDR …) is connected according to the instruction in the WIR register.

Wrapper Interface Port – WIP (II)

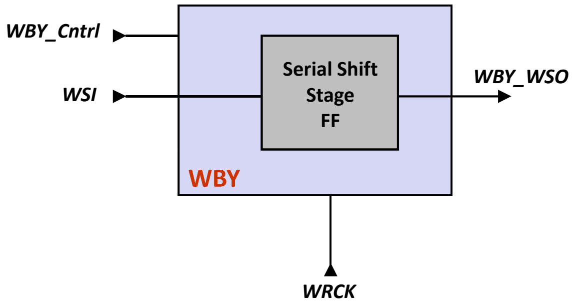
- ShiftWR (*Wrapper Shift*): Enable signal for the shift operations on the WIR register (when *SelectWIR*=high) or another register (when *SelectWIR*=low) according to the instruction in the WIR register.
- CaptureWR (*Wrapper Capture*): Enable signal for the parallel load of data in the WIR register (when *SelectWIR*=high) or another register (when *SelectWIR*=low) according to the instruction in the WIR register.
- UpdateWR (*Wrapper Update*): Enable signal for the update of the WIR register (when *SelectWIR*=high) or another register (when *SelectWIR*=low) according to the instruction in the WIR register.
- WRCK (*Wrapper Clock*): Clock signal for the operation of the WIR and WBY registers or other registers of the wrapper (e.g. WBR) or even the core.
- WRSTN (*Wrapper Reset*): Asynchronous *reset* signal (active-low) of the WIR register or other registers of the wrapper.

Wrapper Registers' Architecture



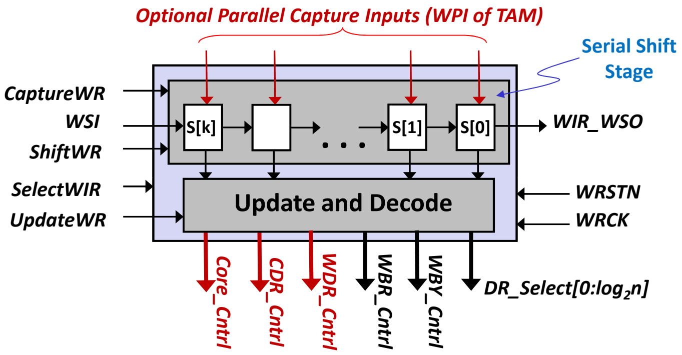
Wrapper Bypass Register – WBY

- The *Wrapper Bypass Register – WBY* is an one stage register. When this register is enabled, the signal at the WSI input port bypasses every other register and directly feeds the WSO output port.

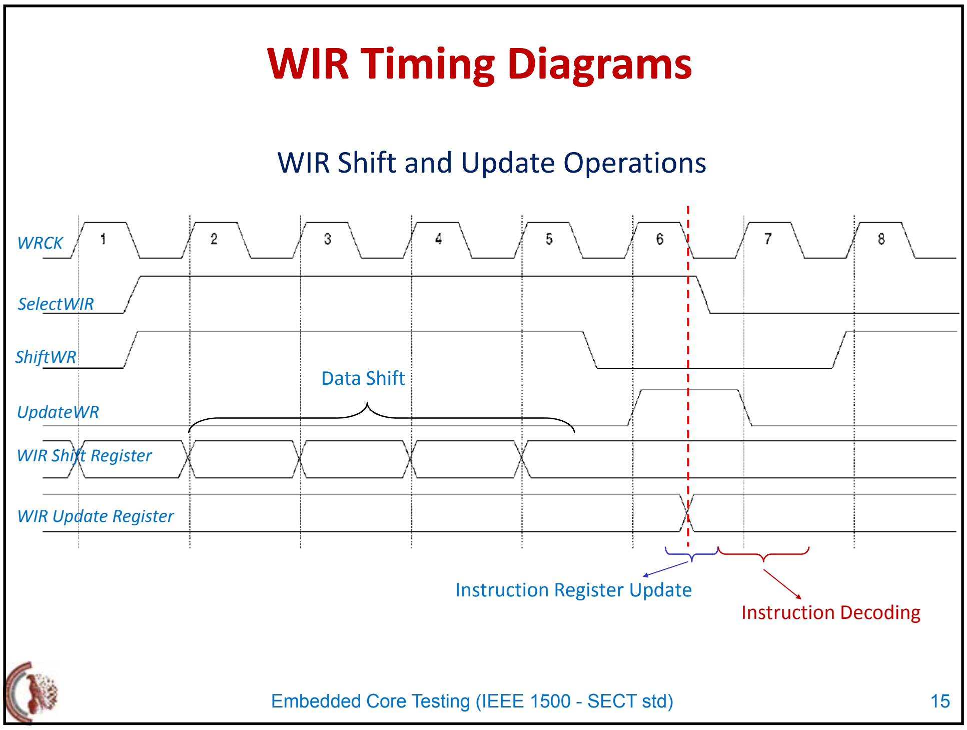


Wrapper Instruction Register – WIR

- The *Wrapper Instruction Register – WIR* is a serial/parallel input and output register. It consists of two stages: a flip-flop based serial shift stage and a latch based update/decode stage. Every flip-flop drives a latch. The latch retains the current instruction when the register is fed with new data (instructions).

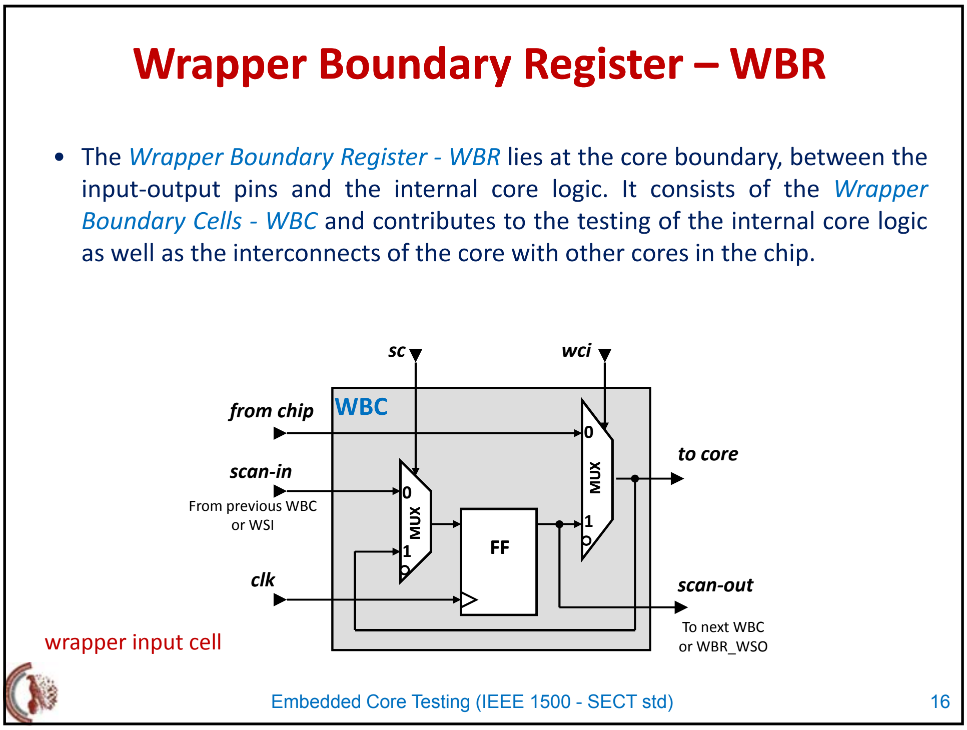


k ≥ 3



WIR Timing Diagrams

WIR Shift and Update Operations



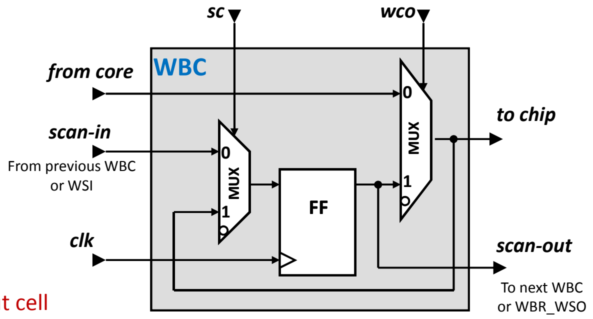
Wrapper Boundary Register – WBR

- The *Wrapper Boundary Register - WBR* lies at the core boundary, between the input-output pins and the internal core logic. It consists of the *Wrapper Boundary Cells - WBC* and contributes to the testing of the internal core logic as well as the interconnects of the core with other cores in the chip.

wrapper input cell

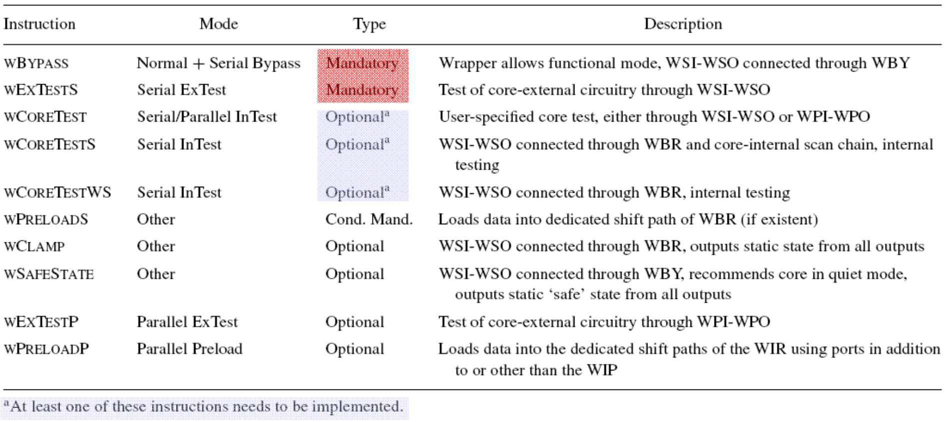
Wrapper Boundary Register – WBR

The multiplexers of the WBR cells are controlled by the *sc* (*scan*) signal, which is also used to control the core's scan chains, and the *wci* (*wco*) signal for the input cell (output cell). The value of the *wci* (*wco*) signal is defined by the decoding of the instruction inside the WIR. In addition, the clock *clk* signal of the core is used to drive the flip-flop of the cell.



wrapper output cell

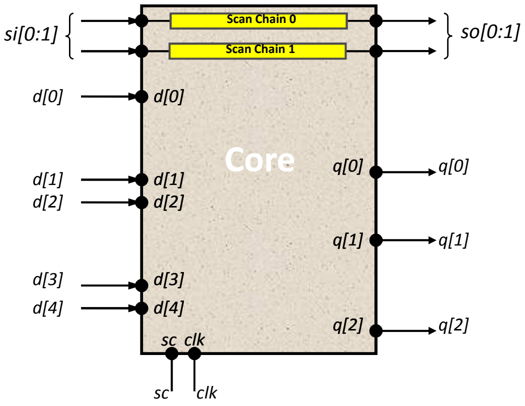
IEEE 1500 std. Instructions



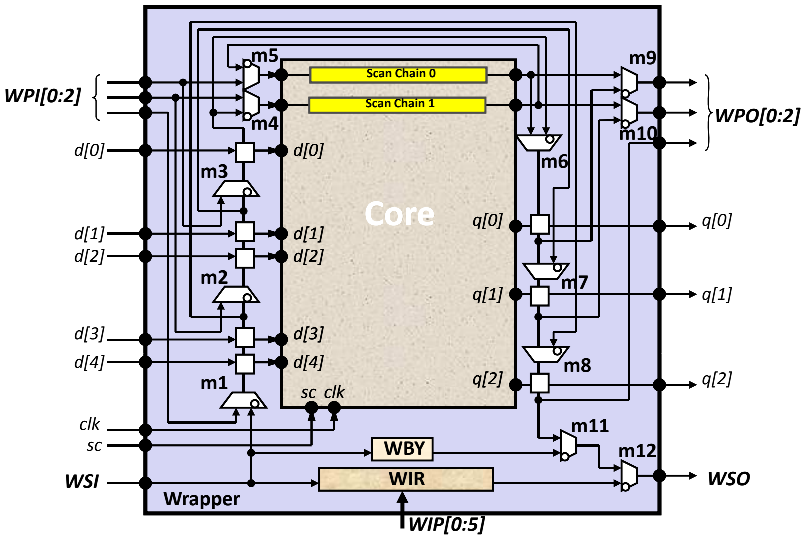
Instruction	Mode	Type	Description
WBYPASS	Normal + Serial Bypass	Mandatory	Wrapper allows functional mode, WSI-WSO connected through WBY
WEXTESTS	Serial ExTest	Mandatory	Test of core-external circuitry through WSI-WSO
WCORETEST	Serial/Parallel InTest	Optional[a]	User-specified core test, either through WSI-WSO or WPI-WPO
WCORETESTS	Serial InTest	Optional[a]	WSI-WSO connected through WBR and core-internal scan chain, internal testing
WCORETESTWS	Serial InTest	Optional[a]	WSI-WSO connected through WBR, internal testing
WPRELOADS	Other	Cond. Mand.	Loads data into dedicated shift path of WBR (if existent)
WCLAMP	Other	Optional	WSI-WSO connected through WBR, outputs static state from all outputs
WSAFESTATE	Other	Optional	WSI-WSO connected through WBY, recommends core in quiet mode, outputs static 'safe' state from all outputs
WEXTESTP	Parallel ExTest	Optional	Test of core-external circuitry through WPI-WPO
WPRELOADP	Parallel Preload	Optional	Loads data into the dedicated shift paths of the WIR using ports in addition to or other than the WIP

[a] At least one of these instructions needs to be implemented.

The Core



IEEE 1500 Wrapper Compliant Core

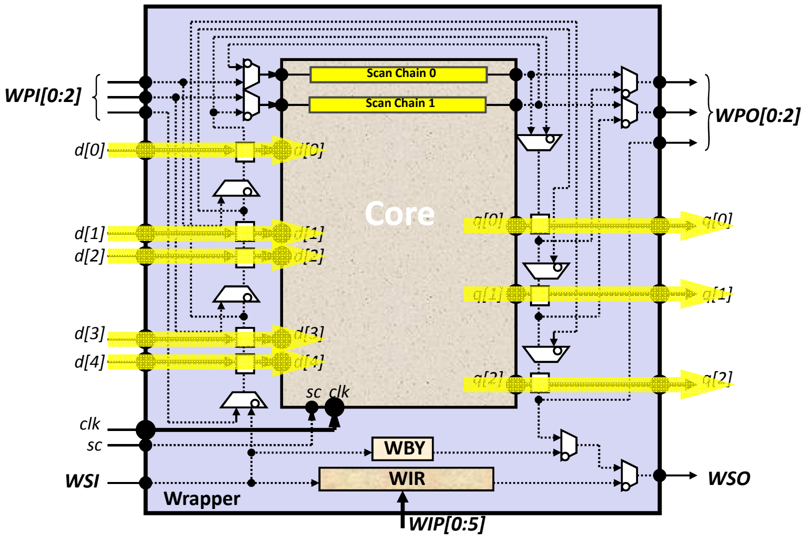


The WBYPASS Instruction

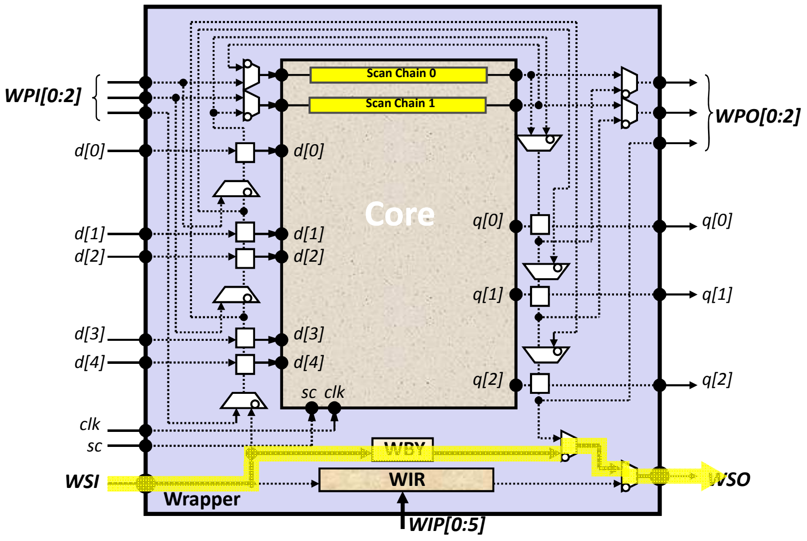
The mandatory WBYPASS instruction provides two non-conflicting modes of operation: the normal mode and the bypass mode.

This instruction configures the WBCs for the normal mode of operation and in parallel connects the WBY register between the WSI and WSO ports so that during test operations the core is bypassed to provide fast test data access to other cores in the chip.

Normal Mode of Operation – WBYPASS



Serial Bypass – WBYPASS

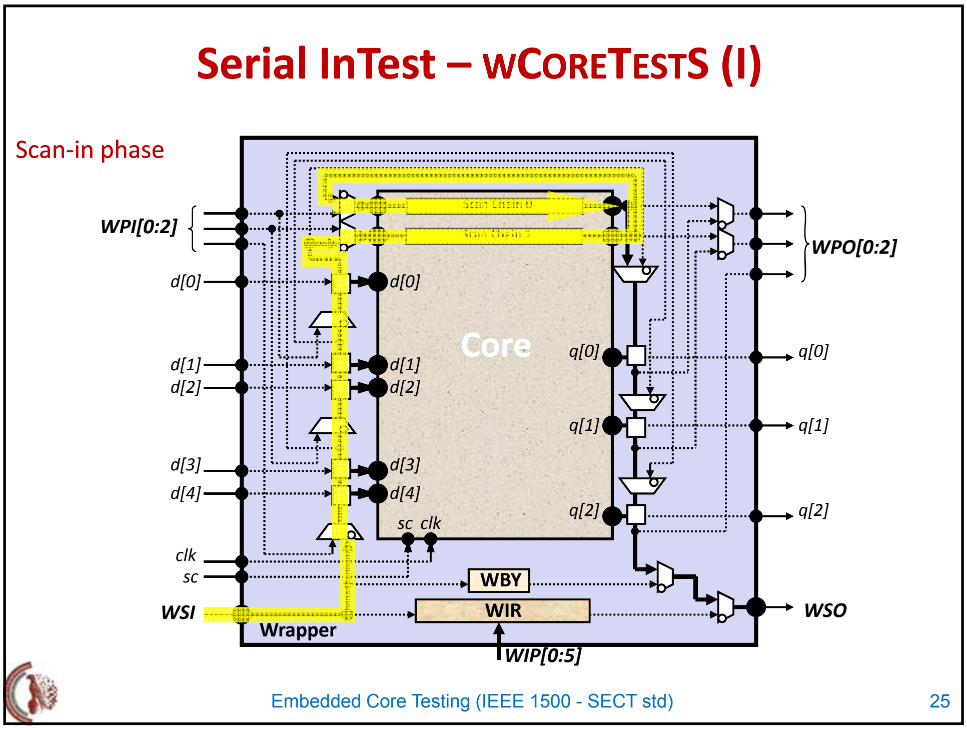


The WCORETESTS Instruction

The WCORETESTS instruction is utilized for the serial internal core testing.

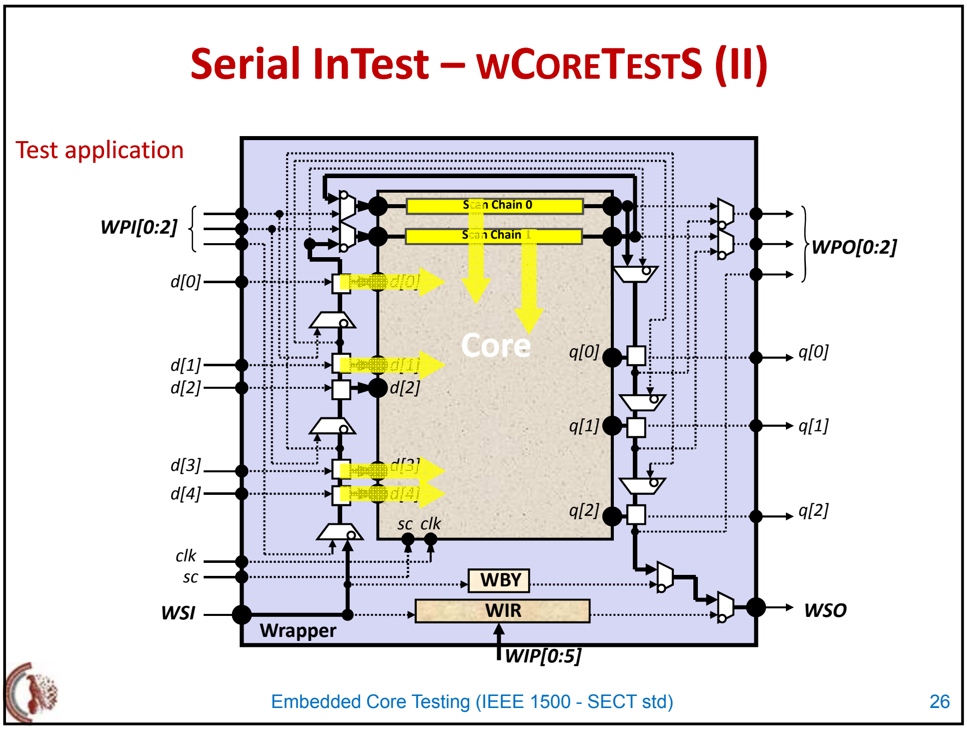
This instruction connects the WBR register and the internal core scan chains to a unified scan chain between the WSI and WSO ports.

Test data, from the test control unit (BIST or ATE), are scanned-in to the wrapper input boundary cells of the WBR and the scan chains and they are applied to the core’s logic. Next, the core’s responses are captured at the scan chains and the wrapper output boundary cells of the WBR and they are scanned-out to the test control unit (BIST or ATE) for processing.



Serial InTest – WCORETESTS (I)

Scan-in phase

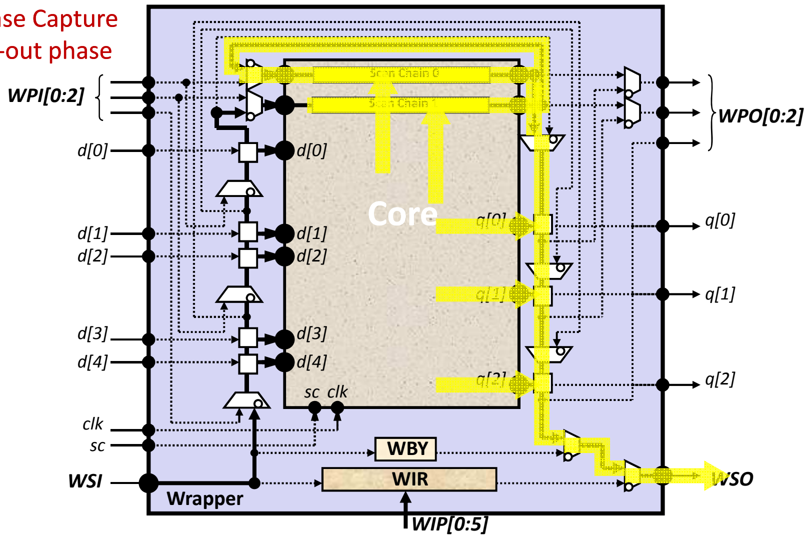


Serial InTest – WCORETESTS (II)

Test application

Serial InTest – WCORETESTS (III)

Response Capture & Scan-out phase



The WCORETESTWS Instruction

The WCORETESTWS instruction is almost identical to the WCORETESTS instruction. The difference is that the core scan chains are not utilized and only the WBR is exploited for the scan operations.

The WEXTESTS Instruction

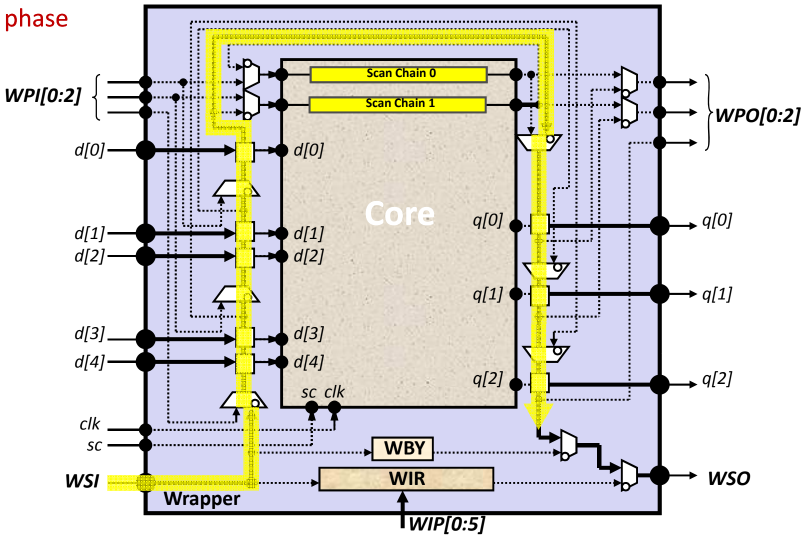
The mandatory WEXTESTS instruction is utilized during the serial external testing of the core interconnects with other cores in the chip.

This instruction provides controllability and observability over the glue logic surrounding the core.

The instruction exclusively connects the WBR between the WSI and WSO ports. The wrapper output boundary cells of the WBR are exploited to provide test data, while the wrapper input boundary cells of the WBR are used to capture the test responses. The test data are serially transferred between the test controller (BIST or ATE) and the WBR.

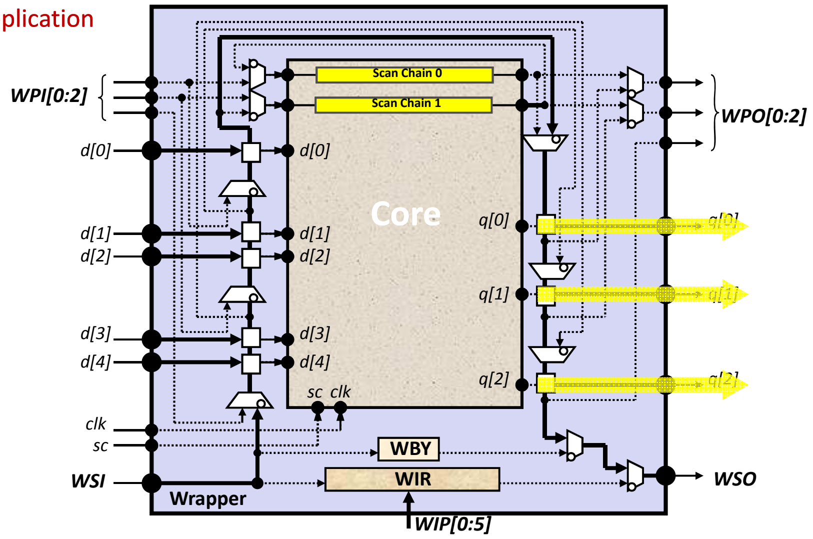
Serial ExTest – WEXTESTS (I)

Scan-in phase



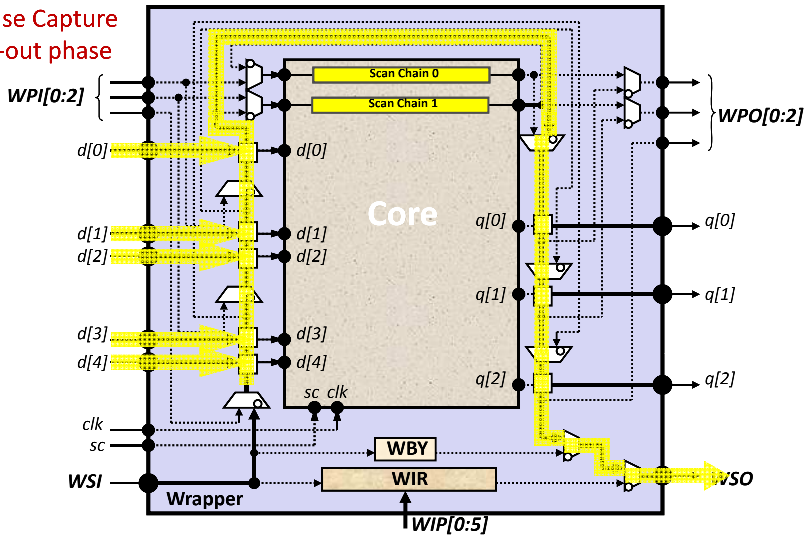
Serial ExTest – WExTestS (II)

Test application



Serial ExTest – WExTestS (III)

Response Capture & Scan-out phase



The WCORETEST Instruction

The WCORETEST instruction is utilized for the parallel internal core testing.

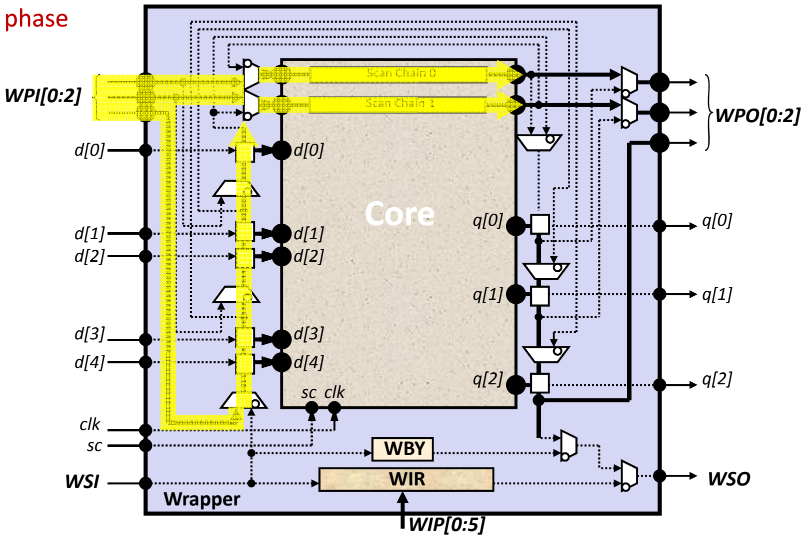
This instruction exploits the parallel (user defined) TAM to scan-in/out in parallel test data to/from the WBR and the scan chains.

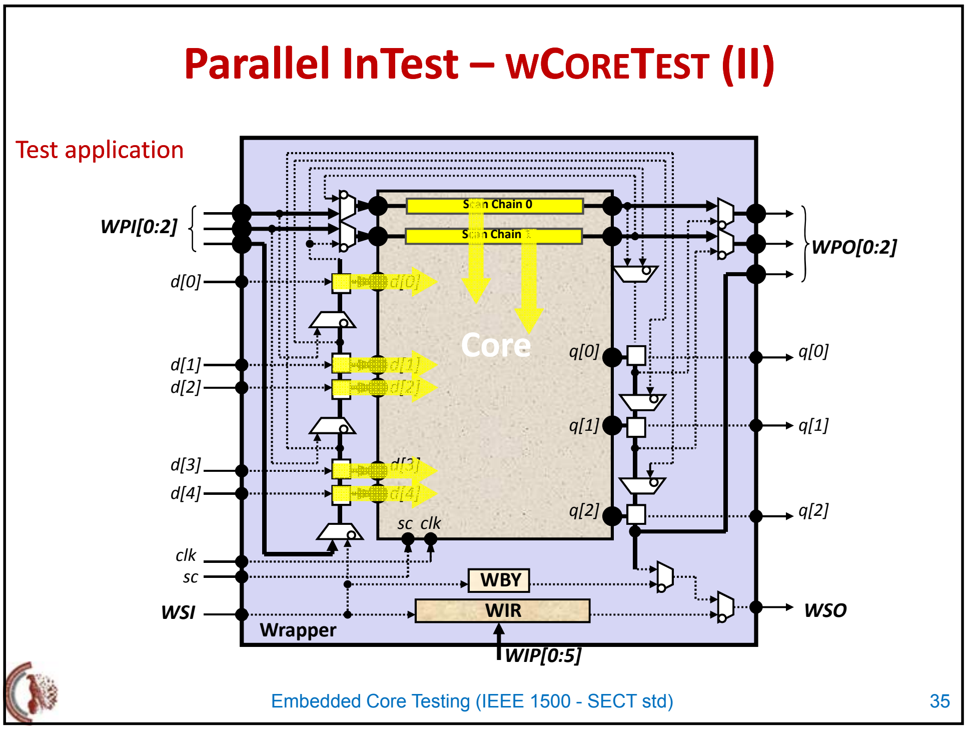
The instruction connects the WBR between a dedicated input of the WPI port and a dedicated output of the WPO port. Similarly, connects the scan chain (if it is required) between dedicated inputs of the WPI port and dedicated outputs of the WPO port.

The test data are concurrently scanned-in from the test controller (BIST or ATE) to the wrapper input boundary cells of the WBR and the scan chains and they are applied to the core logic. Next, the test responses are captured at the scan chains and the wrapper output boundary cells of the WBR and then concurrently scanned-out to the test controller (BIST or ATE) for processing.

Parallel InTest – WCORETEST (I)

Scan-in phase

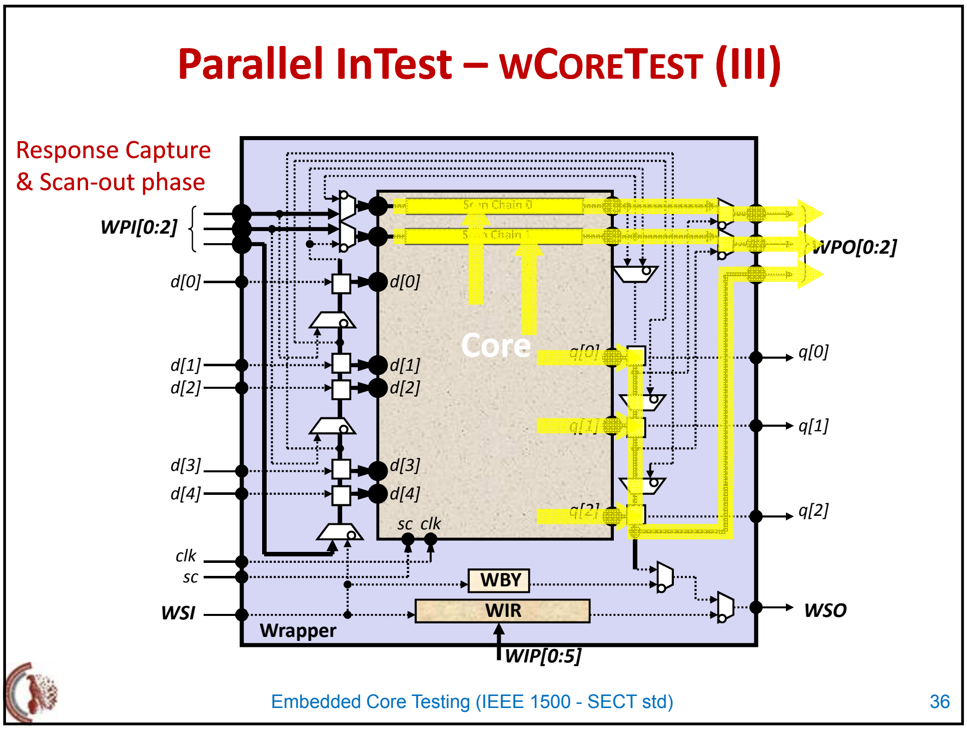




Parallel InTest – wCoreTest (II)

Test application

Embedded Core Testing (IEEE 1500 - SECT std)



Parallel InTest – wCoreTest (III)

Response Capture
& Scan-out phase

Embedded Core Testing (IEEE 1500 - SECT std)

The WEXTESTP Instruction

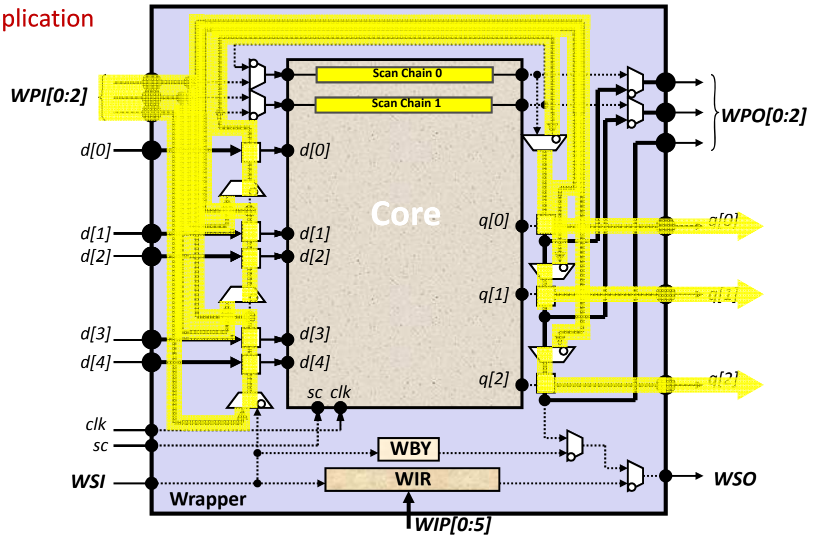
The WEXTESTP instruction is utilized for the parallel external testing of the core interconnects with other cores in the chip.

The instruction provides controllability and observability of the glue logic surrounding the core.

The test data are scanned-in in parallel from the test controller (BIST or ATE) to the wrapper output boundary cells of the WBR. The wrapper input boundary cells of the WBR are used to capture the test responses which are scanned-out in parallel to the test controller (BIST or ATE) for processing.

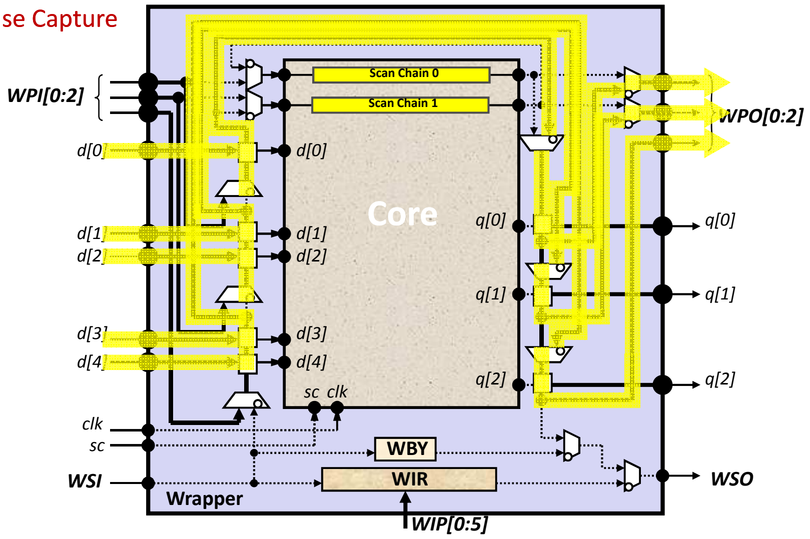
Parallel ExTest – WEXTESTP (I)

Test application

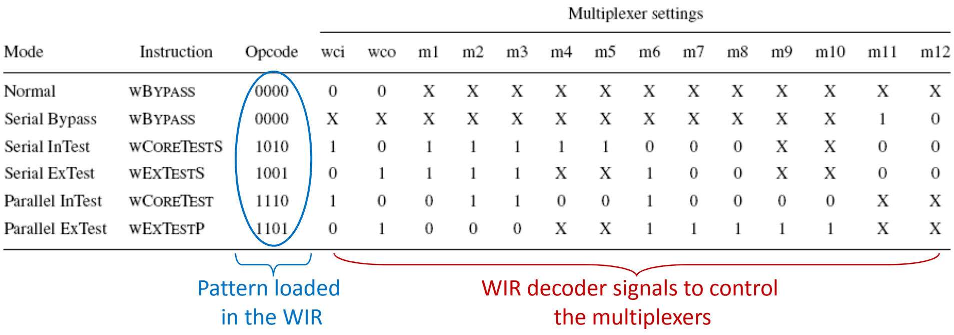


Parallel ExTest – wExTestP (II)

Response Capture



Instructions and MUX Settings



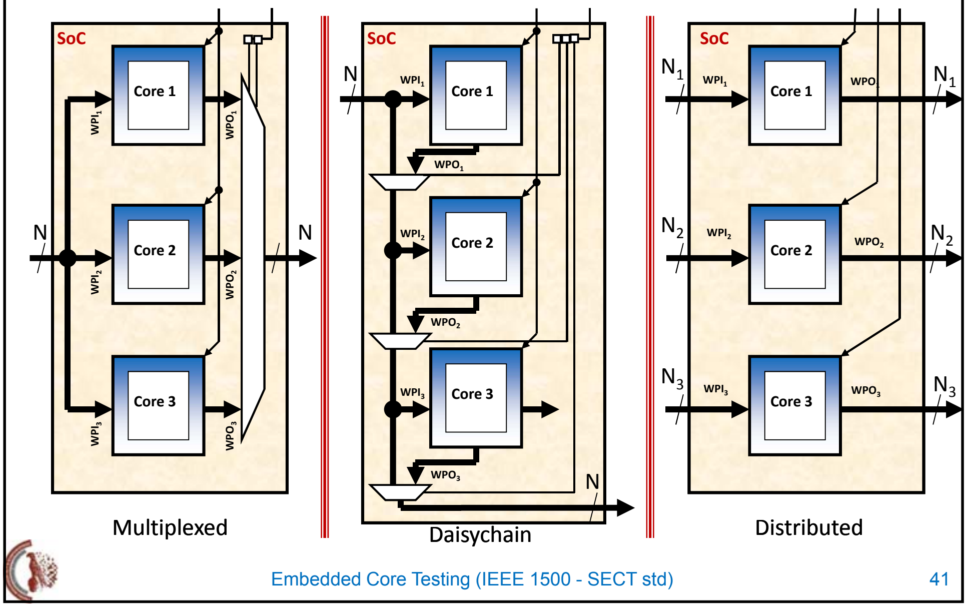
Mode	Instruction	Opcode	Multiplexer settings													
			wci	wco	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12
Normal	wBypass	0000	0	0	X	X	X	X	X	X	X	X	X	X	X	X
Serial Bypass	wBypass	0000	X	X	X	X	X	X	X	X	X	X	X	X	1	0
Serial InTest	wCoreTestS	1010	1	0	1	1	1	1	1	0	0	0	X	X	0	0
Serial ExTest	wExTestS	1001	0	1	1	1	1	X	X	1	0	0	X	X	0	0
Parallel InTest	wCoreTest	1110	1	0	0	1	1	0	0	1	0	0	0	0	X	X
Parallel ExTest	wExTestP	1101	0	1	0	0	0	X	X	1	1	1	1	1	X	X

Pattern loaded in the WIR

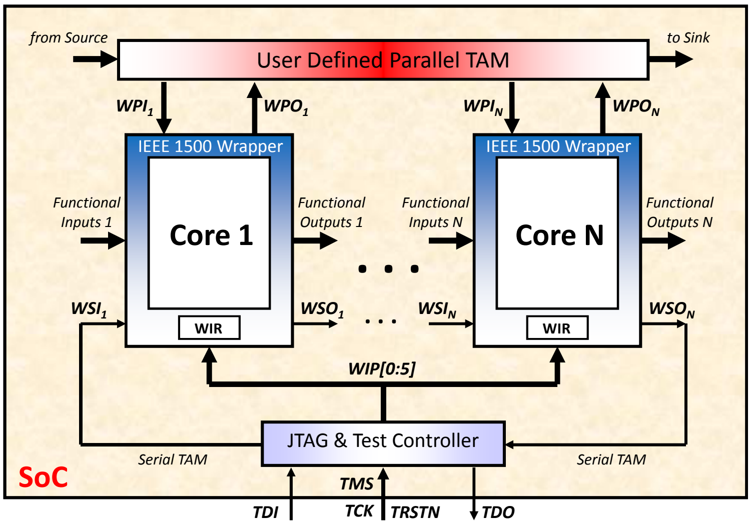
WIR decoder signals to control the multiplexers

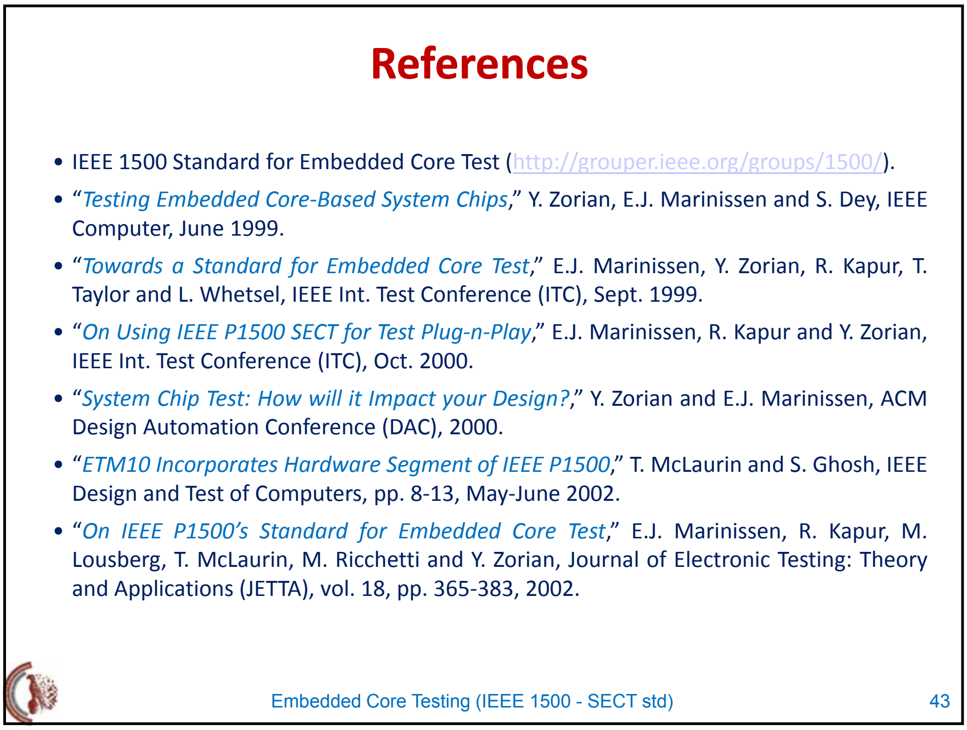
X = don't care value

Parallel TAM Configurations



IEEE 1500 Chip Architecture





References

- IEEE 1500 Standard for Embedded Core Test (http://grouper.ieee.org/groups/1500/).
- "*Testing Embedded Core-Based System Chips*," Y. Zorian, E.J. Marinissen and S. Dey, IEEE Computer, June 1999.
- "*Towards a Standard for Embedded Core Test*," E.J. Marinissen, Y. Zorian, R. Kapur, T. Taylor and L. Whetsel, IEEE Int. Test Conference (ITC), Sept. 1999.
- "*On Using IEEE P1500 SECT for Test Plug-n-Play*," E.J. Marinissen, R. Kapur and Y. Zorian, IEEE Int. Test Conference (ITC), Oct. 2000.
- "*System Chip Test: How will it Impact your Design?*," Y. Zorian and E.J. Marinissen, ACM Design Automation Conference (DAC), 2000.
- "*ETM10 Incorporates Hardware Segment of IEEE P1500*," T. McLaurin and S. Ghosh, IEEE Design and Test of Computers, pp. 8-13, May-June 2002.
- "*On IEEE P1500's Standard for Embedded Core Test*," E.J. Marinissen, R. Kapur, M. Lousberg, T. McLaurin, M. Ricchetti and Y. Zorian, Journal of Electronic Testing: Theory and Applications (JETTA), vol. 18, pp. 365-383, 2002.