Chapter 4: Memory Built-In Self-Test

Jin-Fu Li
Dept. of Electrical Engineering
National Central University
Jhongli, Taiwan

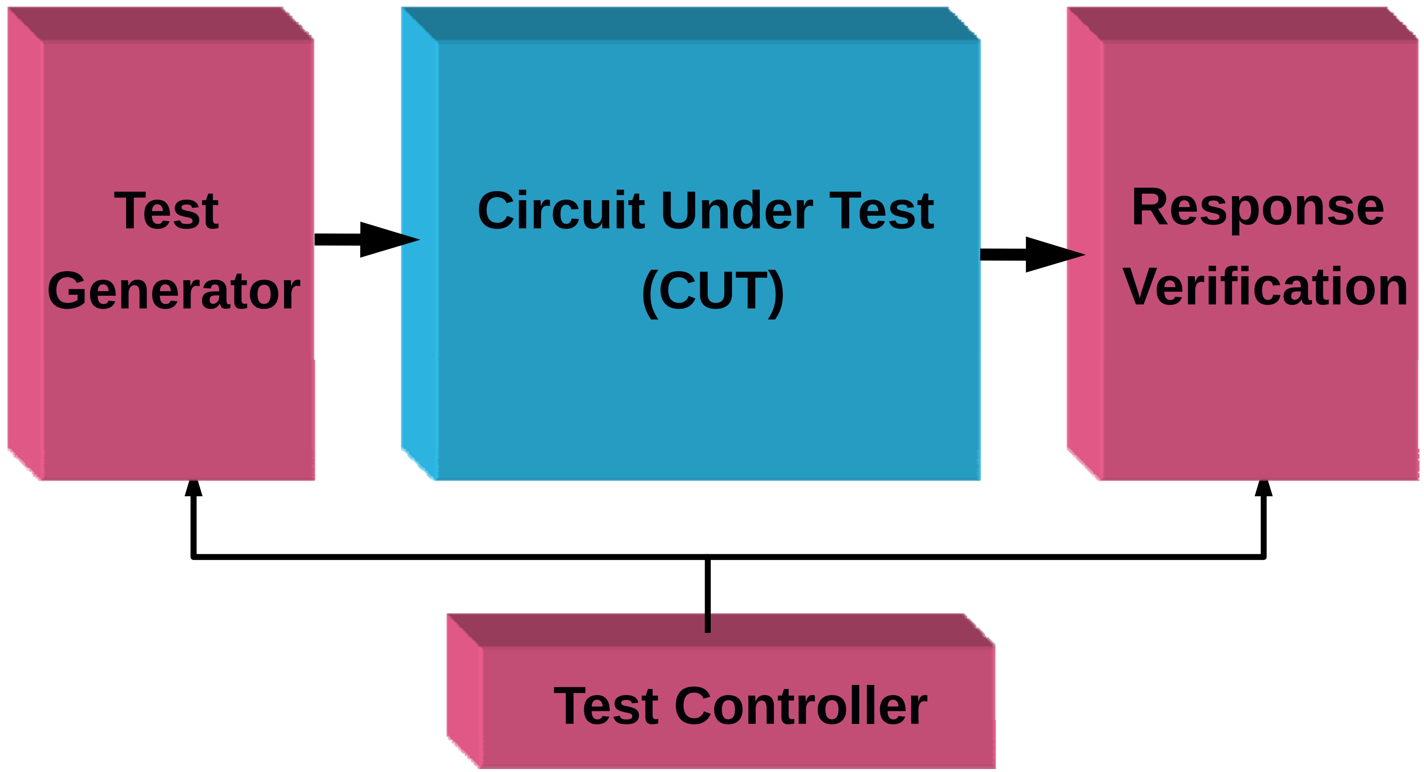
Outline

- ROM BIST
- RAM BIST
- Serial BIST for RAMs
- Processor-Based RAM BIST
- RAM BISTs in SOCs
- References

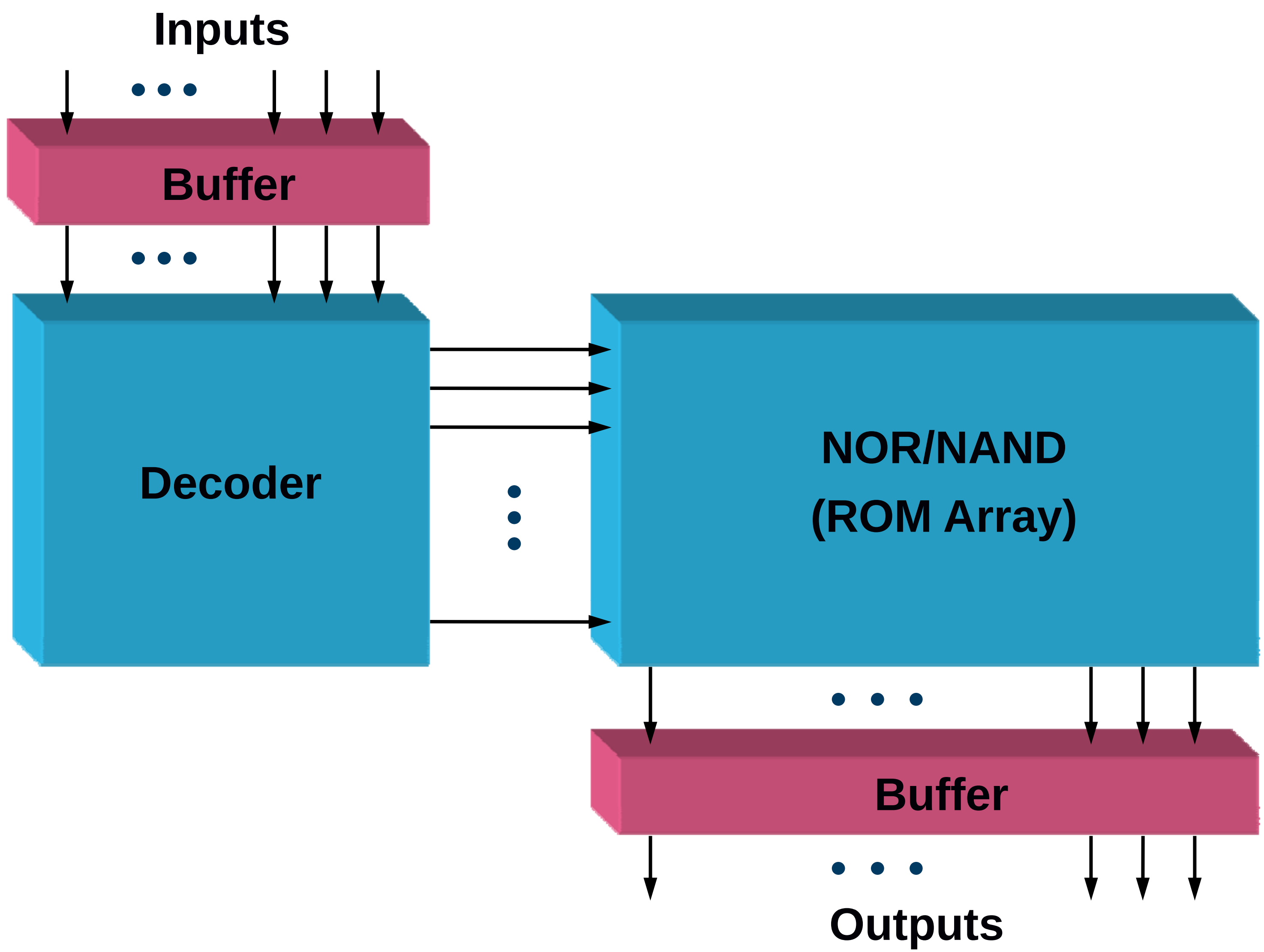
Introduction

- Characteristics of today's SOC designs
  - Typically more than 30 embedded memories on a chip
  - Memories scattered around the device rather than concentrated in one location
  - Different types and sizes of memories
  - Memories doubly embedded inside embedded cores
  - Test access to these memories from only a few chip I/O pins
- Built-in self-test (BIST) is considered the best solution for testing embedded memories within SOCs
  - It offers a simple and low-cost means without significantly impacting performance

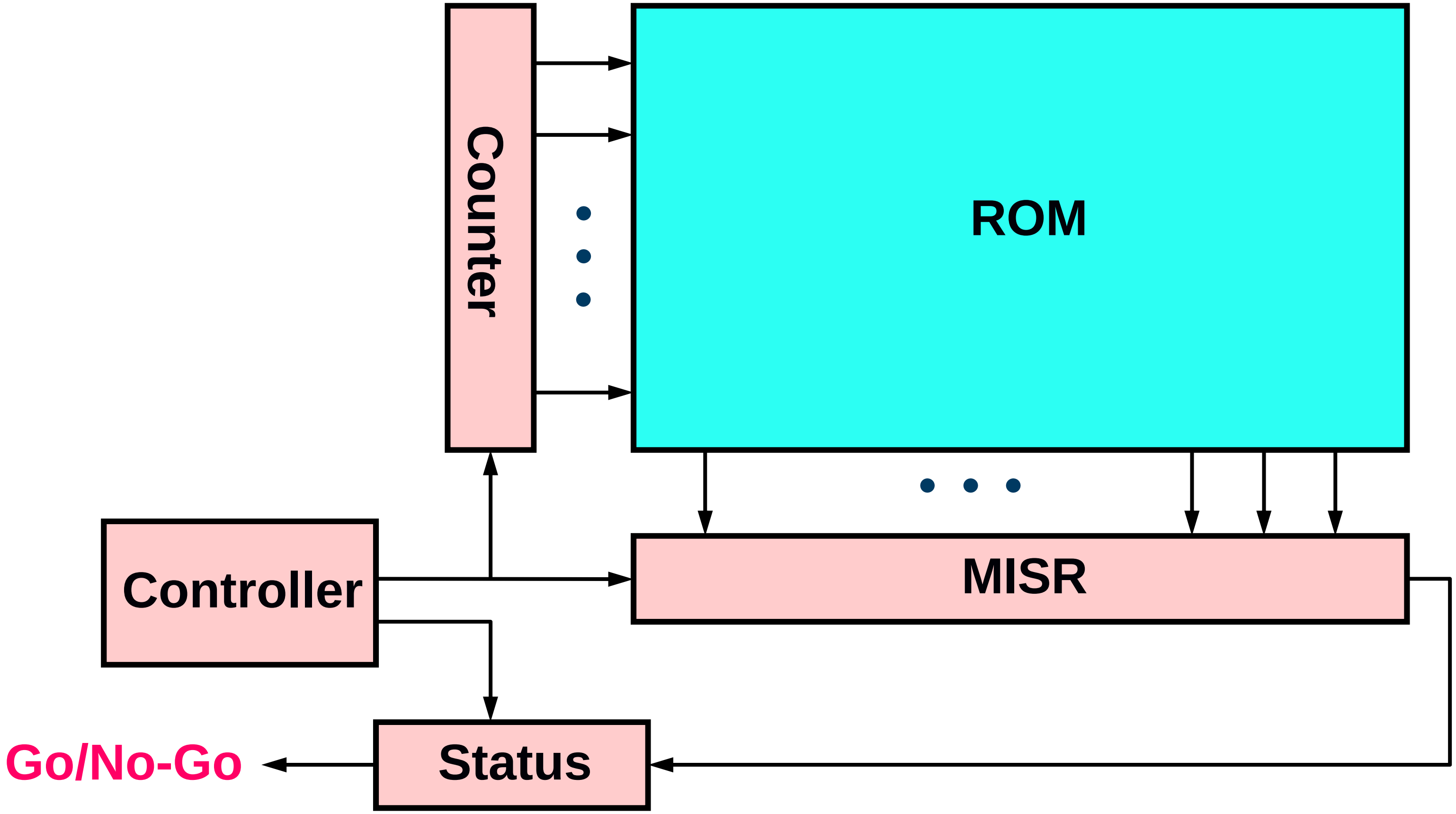
General BIST Architecture



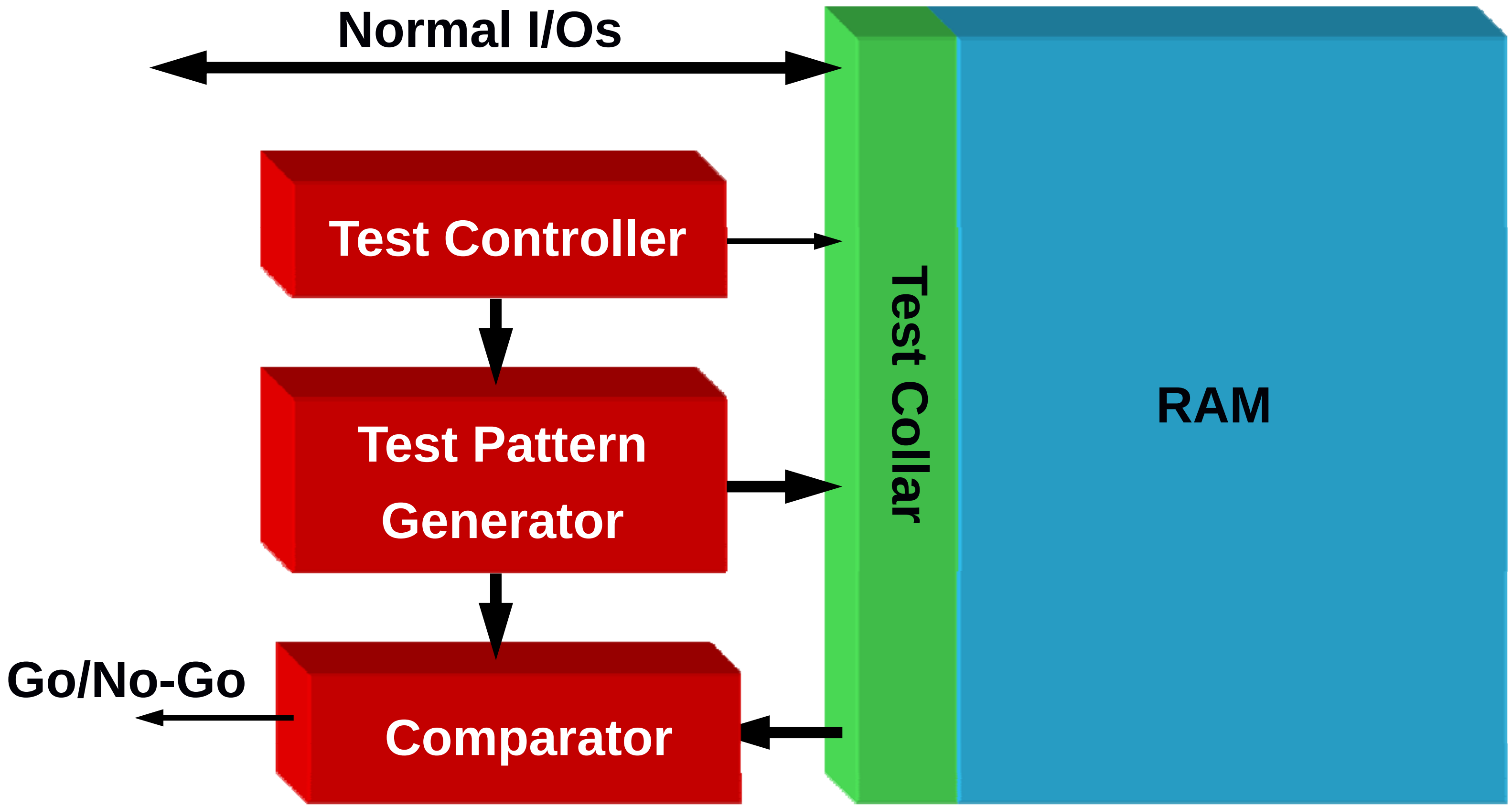
ROM Functional Block Diagram



An Example of ROM BIST



Typical RAM BIST Architecture



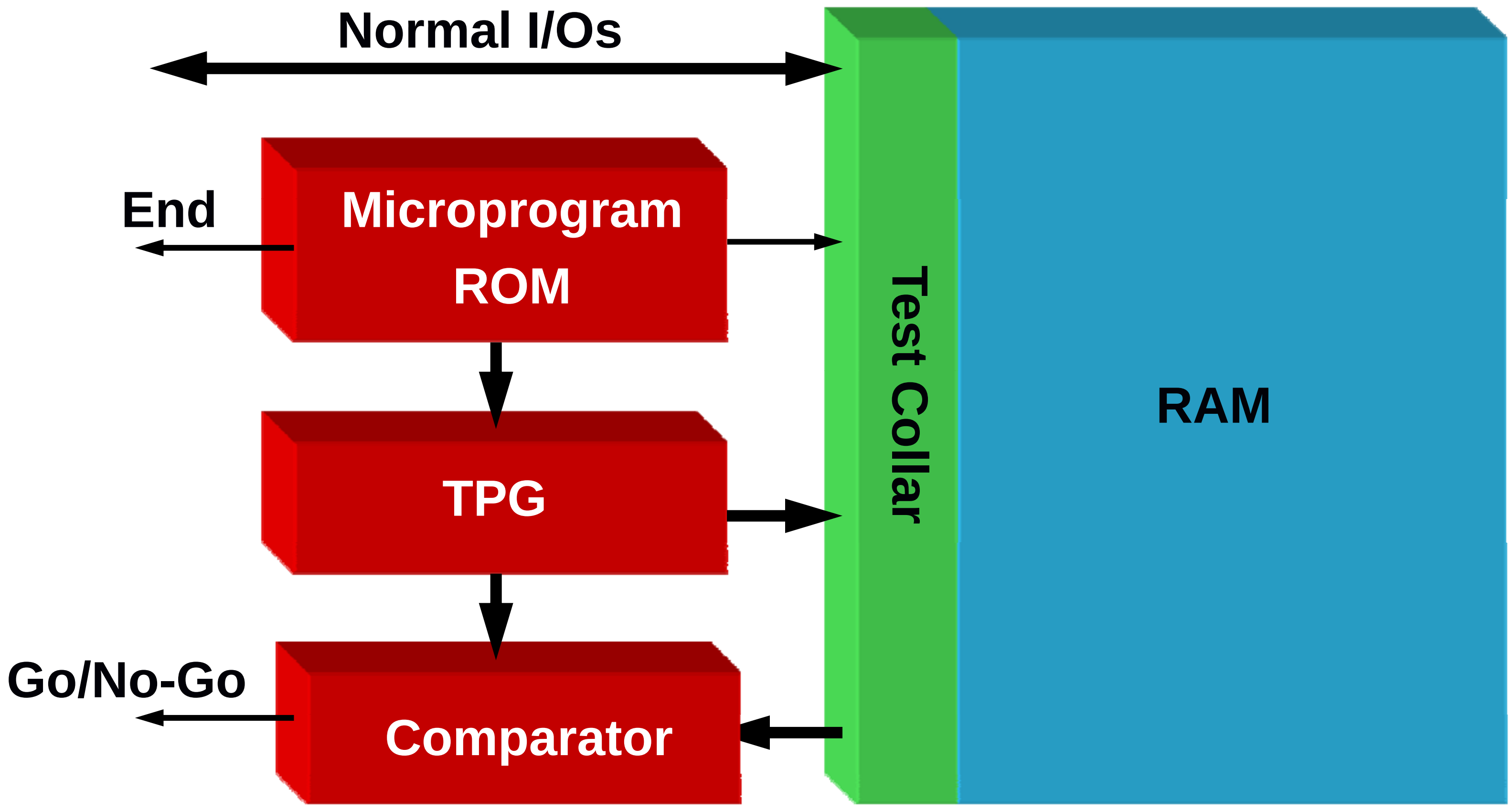
RAM BIST

- In general, two BIST approaches have been proposed for the RAMs
  - FSM-based RAM BIST
  - ROM-based RAM BIST
- Controller
  - Generate control signals to the test pattern generator & the memory under test
- Test pattern generator (TPG)
  - Generate the required test patterns and Read/Write signals
- Comparator
  - Evaluate the response

ROM-Based RAM BIST

- The features of ROM-based BIST scheme
  - The ROM stores test procedures for generating test patterns
  - Self-test is executed by using BIST circuits controlled by the microprogram ROM
  - A wide range of test capabilities due to ROM programming flexibility
- The BIST circuits consists of the following functional blocks
  - Microprogram ROM to store the test procedure
  - Program counter which controls the microprogram ROM
  - TPG
  - Comparator

ROM-Based RAM BIST Architecture



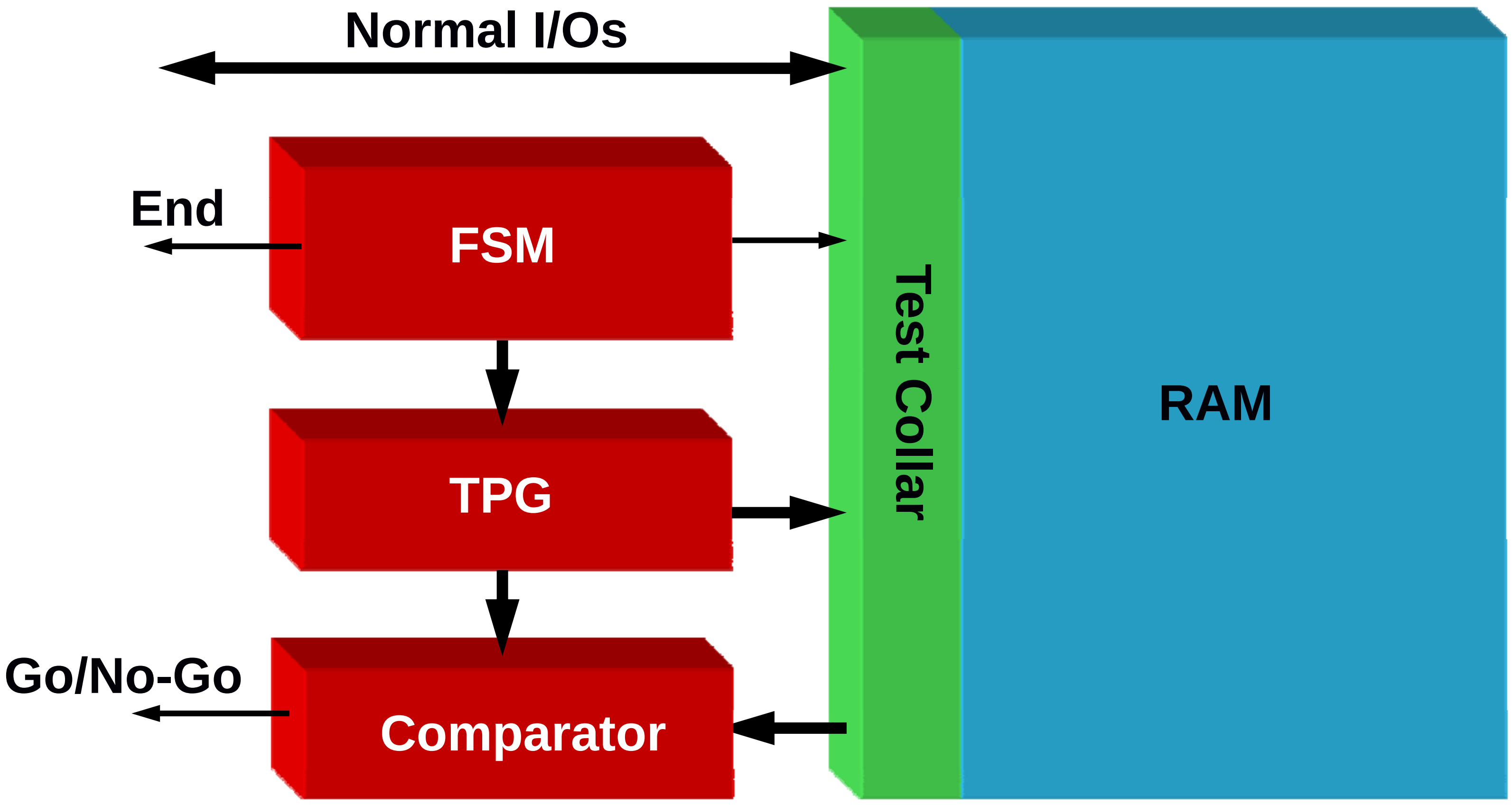
March-9N Test Procedure & Microcodes

- March-9N:

STEP	OPERATION	MICROCODE
	CLEAR	0 0 0 0 0 0 0 0 1 0
	WRITE(D), INC AC, IF AC=MAX THEN INC PC	1 0 1 0 0 0 0 1 0 0
	READ(D)	0 0 0 0 0 0 0 1 0 0
	WRITE(D’), INC AC, IF AC=MAX THEN INC PC ELSE DEC PC	1 1 1 0 0 1 0 1 0 0
	READ(D’)	0 0 0 0 0 1 1 0 0 0
	WRITE(D), INC AC, IF AC=MAX THEN INC PC ELSE DEC PC	1 1 1 0 0 0 0 1 0 0
	DEC AC	0 0 0 1 0 0 0 0 0 0
	READ(D)	0 0 0 0 0 0 1 0 0 0
	WRITE(D’), DEC AC, IF AC=0 THEN INC PC ELSE DEC PC	1 1 0 1 0 1 0 1 0 0
	READ(D’)	0 0 0 0 0 1 1 0 0 0
	WRITE(D), DEC AC, IF AC=0 THEN INC PC ELSE DEC PC	1 1 0 1 0 0 0 1 0 0
	STOP	0 0 0 0 0 0 0 0 0 1

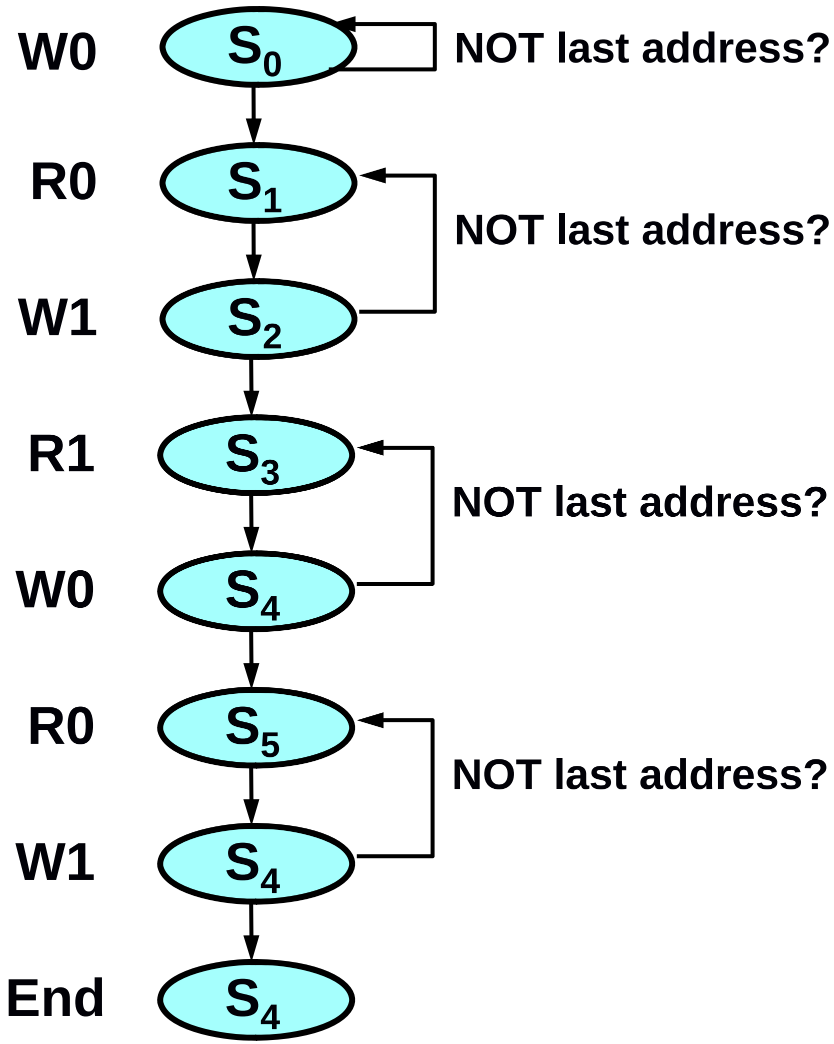
IF AC=MAX/0
ELSE DEC PC
INCREMENT AC
DECREMENT AC
EXCLUSIVE OR
INVERT/NORMAL
COMPARE/MASK
WRITE/READ
CLEAR
TEST END

FSM-Based RAM BIST



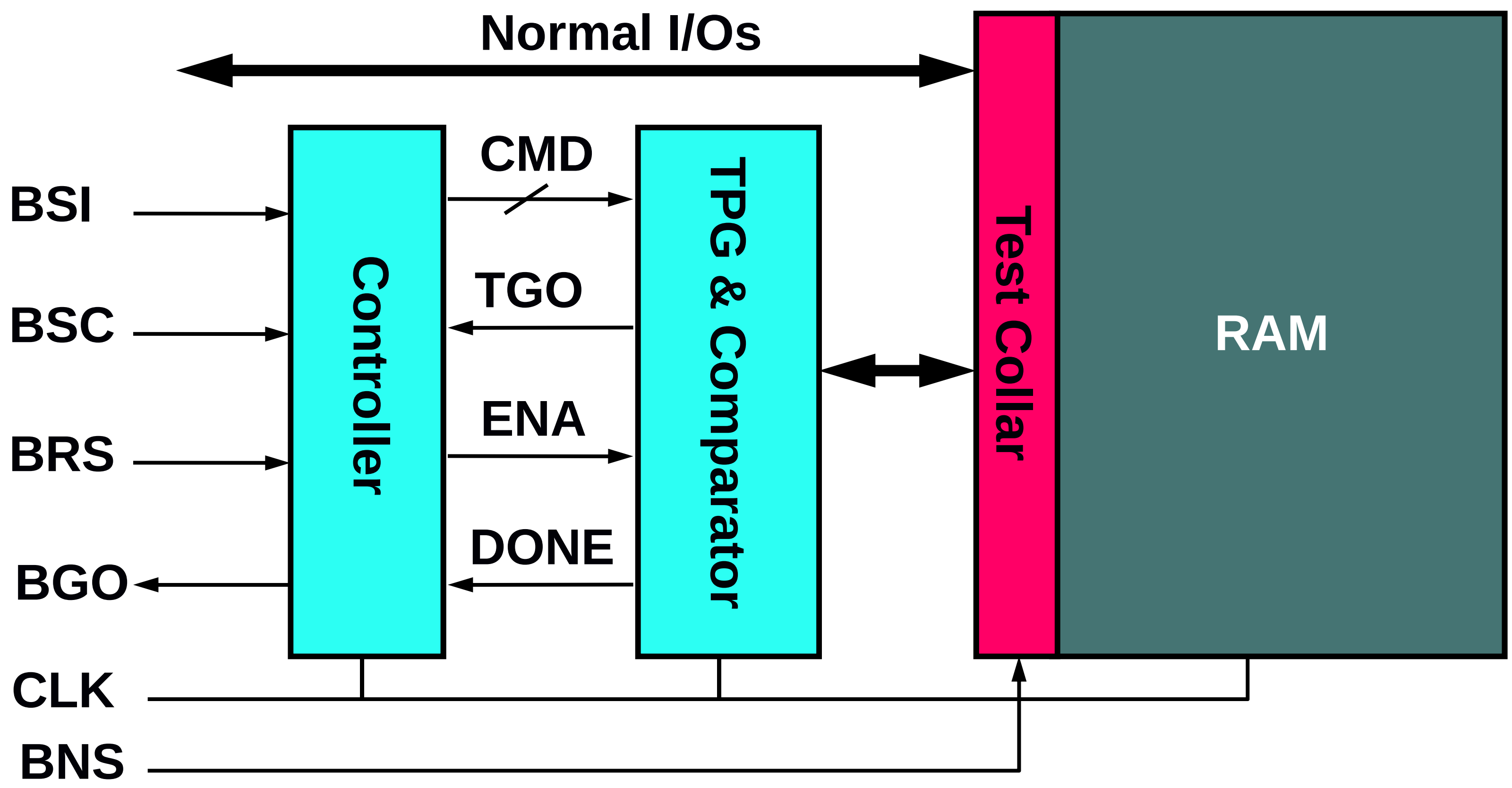
FSM-Based RAM BIST

- An example of the state diagram of controller

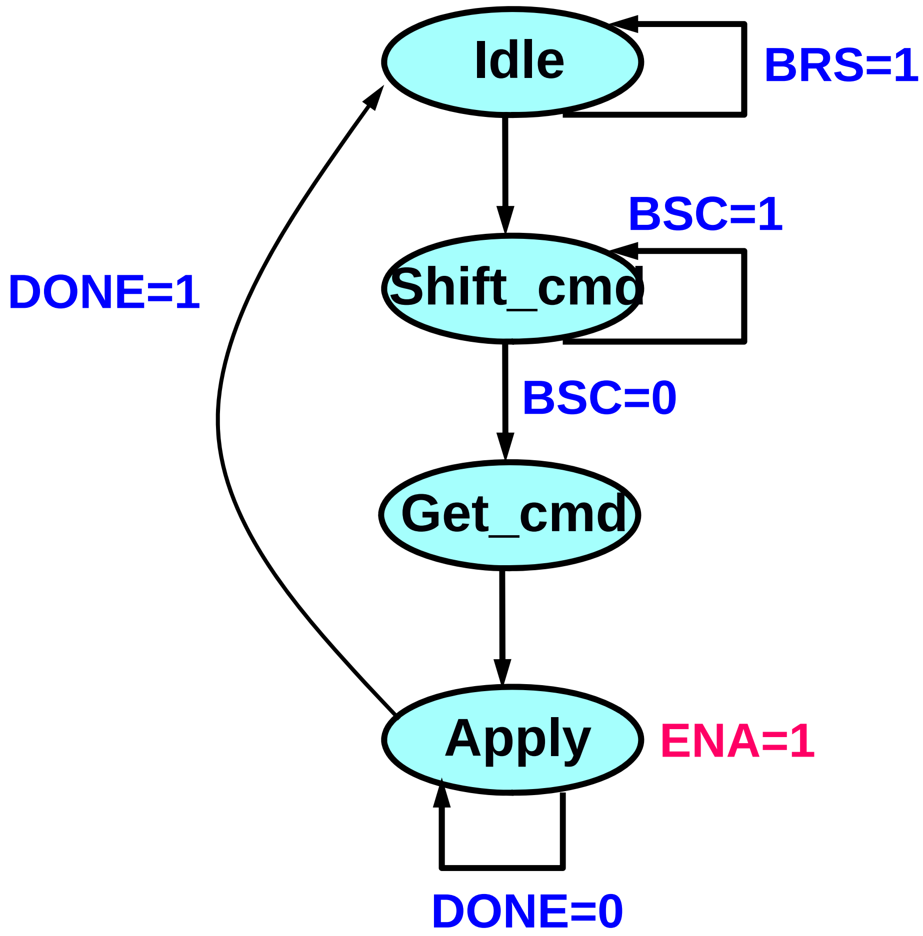


Programmable RAM BIST

- An example of the programmable RAM BIST



FSM State Diagram of the Controller



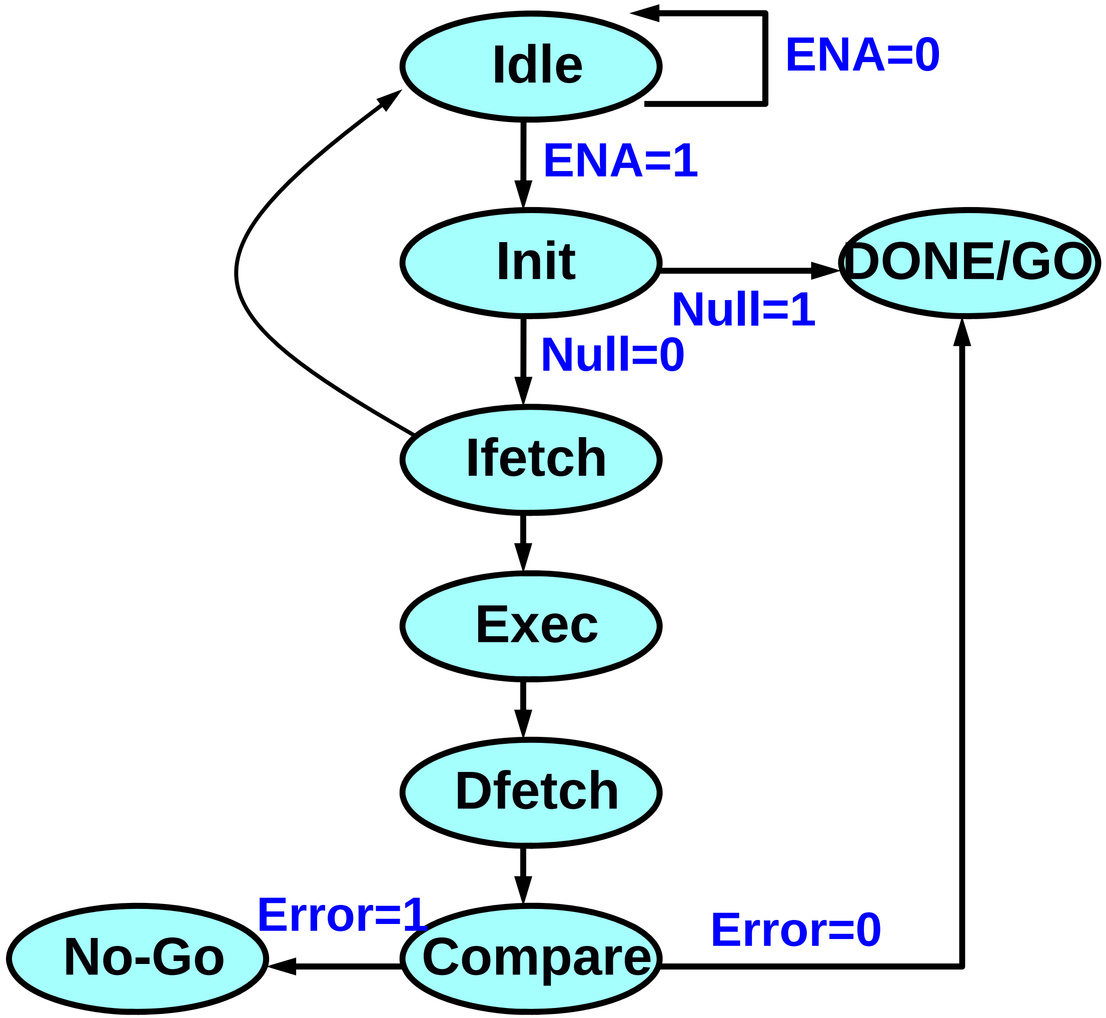
Programmability

- The programmability can be achieved by using test command
- The test command format
  -

U/D	OP	Data background

  - U/D: ascending/descending address sequence
  - OP: test operations
    - For example, wa, rawa', rawa' ra, warawa' ra', etc.
  - Data backgrounds
- The width of each field affects the programmability of the BIST design
  - For example, if 4 bits are used for OP, then only 16 possible test operations can be generated

FSM State Diagram of the TPG



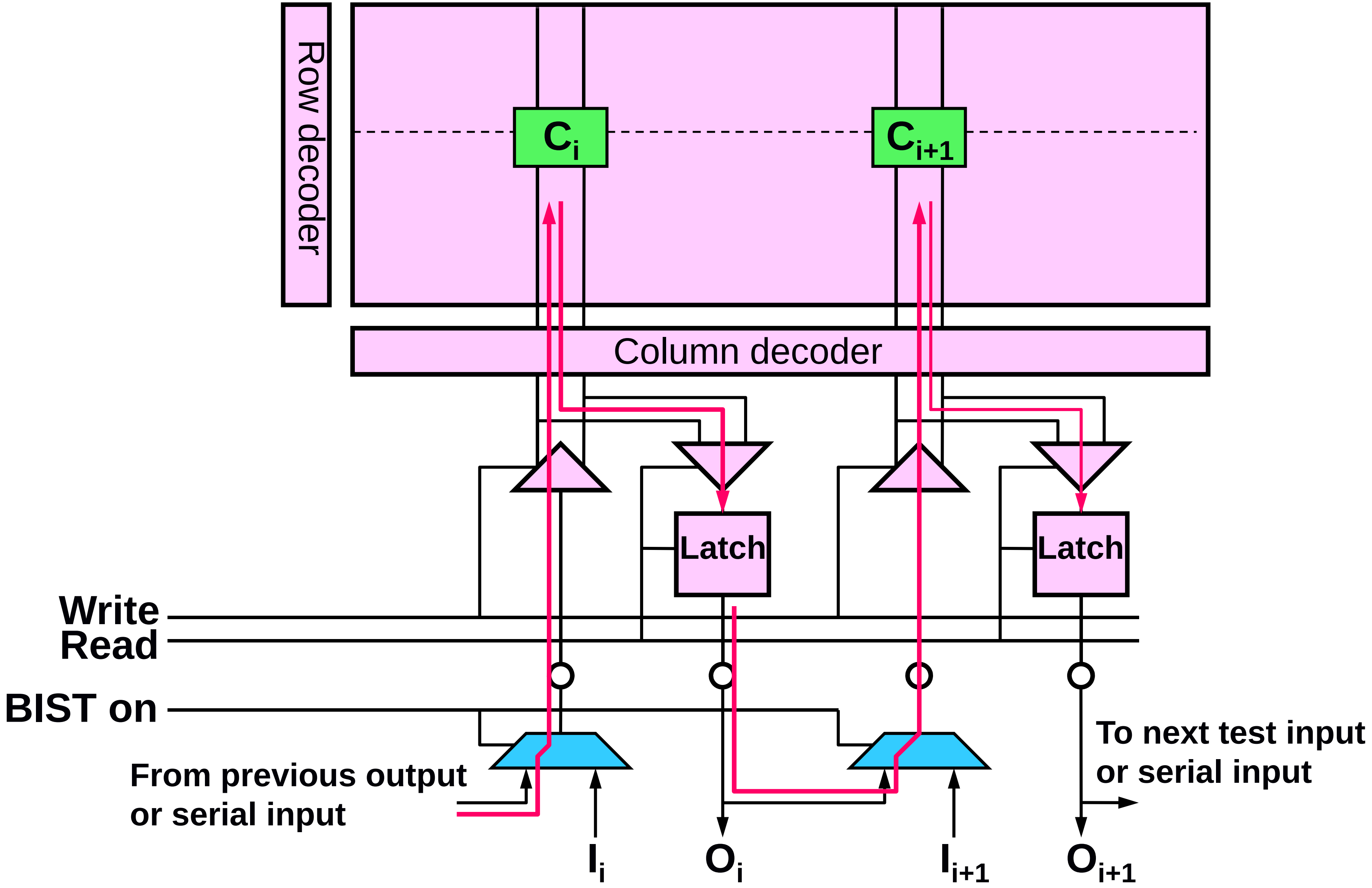
Serial BIST

- Today' s telecommunication ICs often have a variety if multiport memories on one chip
- Typical RAM BISTs evaluate all the bits of a memory word in parallel as it is read
- We can encounter significant problems when applying these BIST schemes to chips that have multiple embedded RAMs of *varying sizes* and *port configurations*
  - The area cost of these BIST designs would be unacceptably high
- One better solution is a serial BIST technique
  - To share BIST design among several RAMs

Benefits of Serial BIST

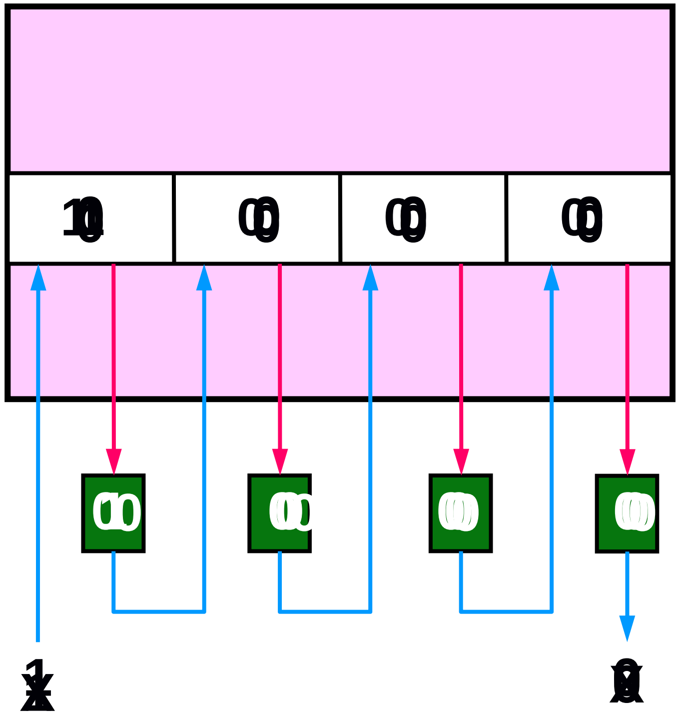
- Only a small amount of additional circuitry is required
- Only a few lines are needed to connect the RAM to the test controller
- Several RAM blocks easily share the BIST controller hardware
- The serial-access mode does not compromise the RAM cycle time
- Existing memory designs do not need any modification to use the serial interface

Serial-Data-Path Connection



Serial Shift Operation

- An example of serial shift operations for the match element (*R0W1*)



Time	Operation	Serial in	Word content				Serial out
0		X	0	0	0	0	X
1	R0	X	0	0	0	0	0
2	W1	1	1	0	0	0	0
3	R0	1	1	0	0	0	0
4	W1	1	1	1	0	0	0
5	R0	1	1	1	0	0	0
6	W1	1	1	1	1	0	0
7	R0	1	1	1	1	0	0
8	W1	1	1	1	1	1	0

Serial March (SMarch)

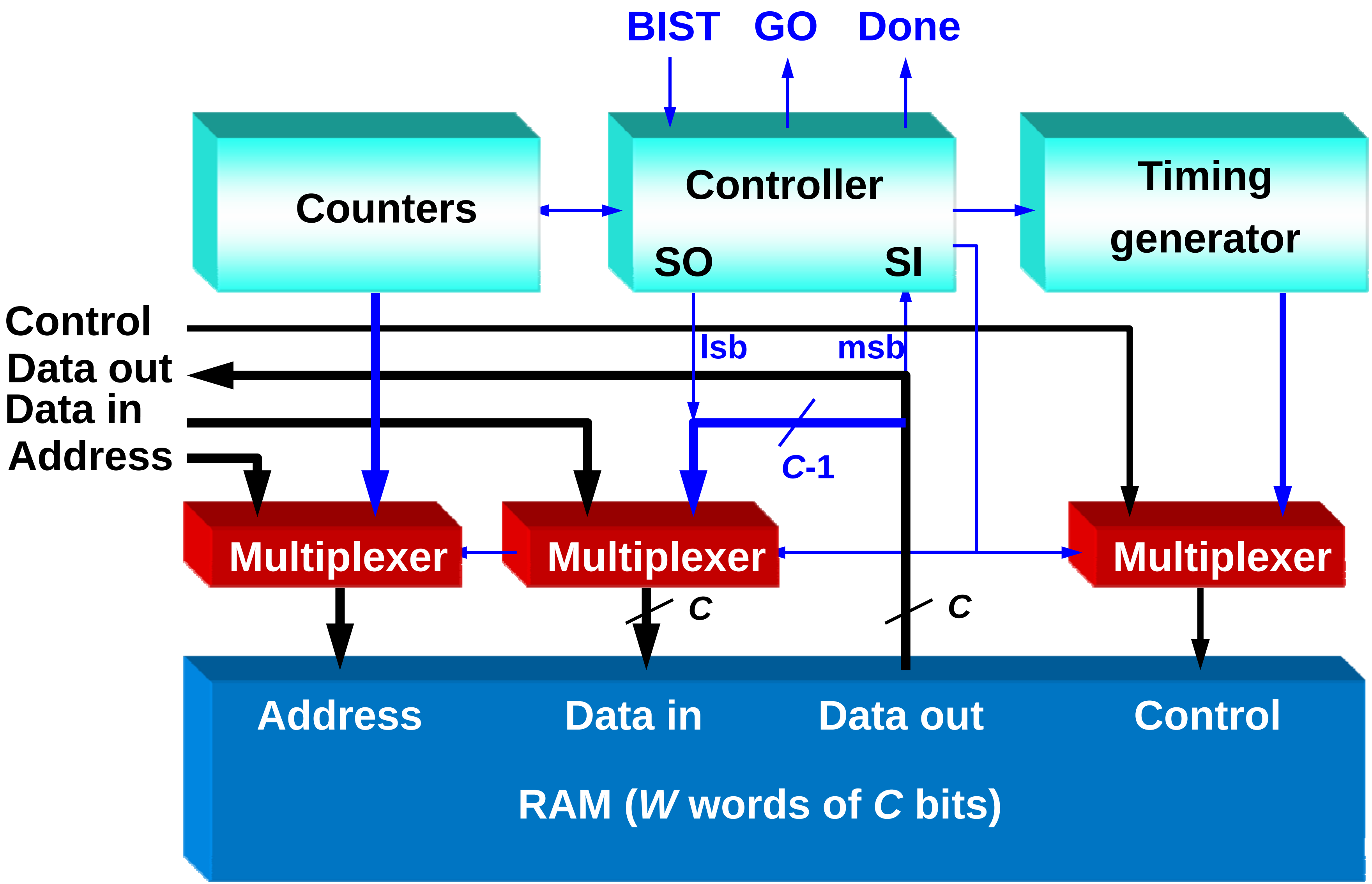
- Assume that a RAM has *W* words, and each word contains *C* bits
- A Read operation is denoted by *R0*, *R1*, or *Rx*, depending on the expected value at the serial output (*x*=do’ t care)
- For a write operation, the terms *W0* or *W1* are used and only the serial input is forced to the value indicated
- The SMarch modified from March C- is as follows

$$\Updownarrow (Rx, W0)^C (R0, W0)^C; \Uparrow (R0, W1)^C (R1, W1)^C$$

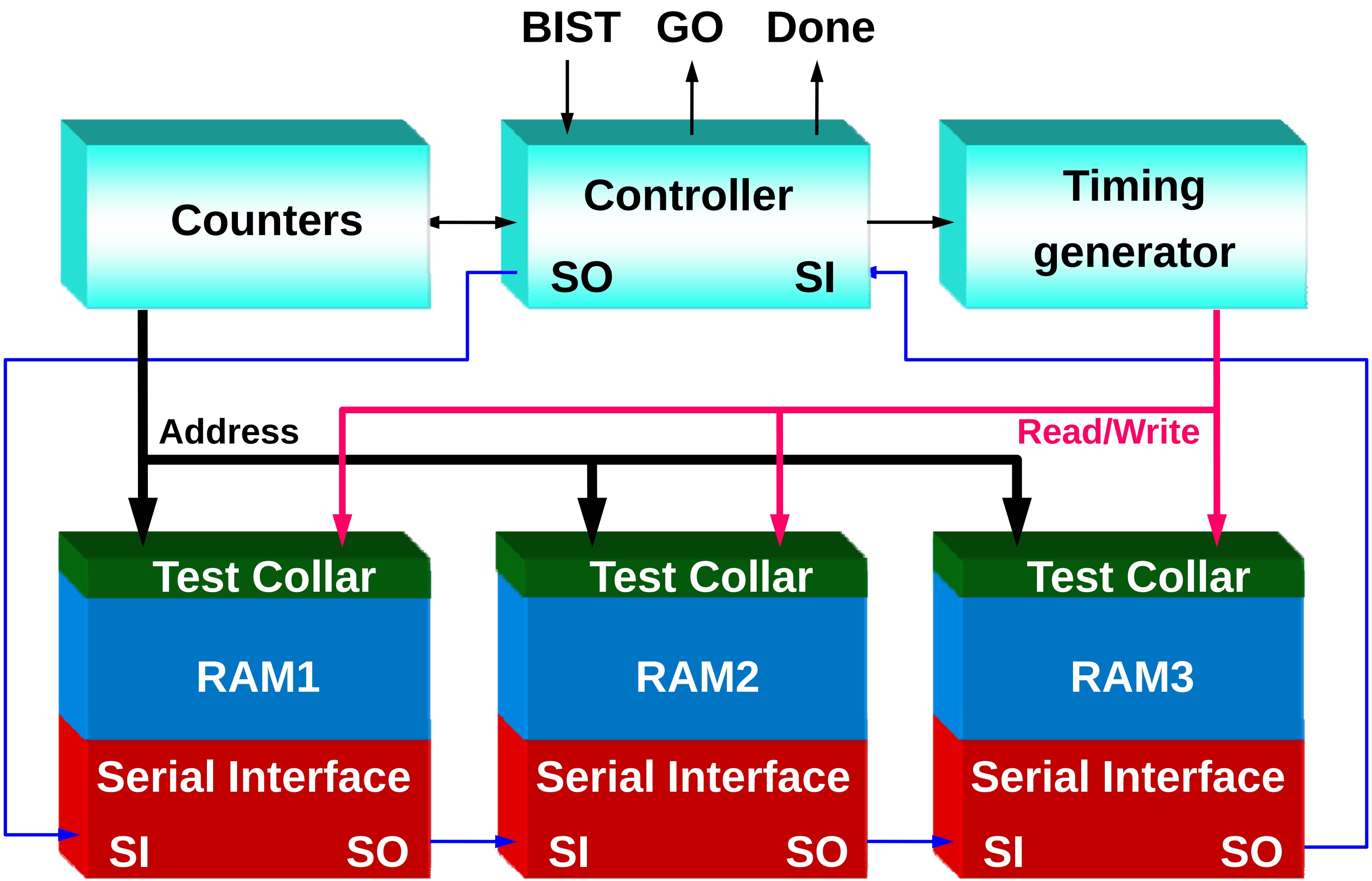
  - $\Uparrow (R1, W0)^C (R0, W0)^C; \Downarrow (R0, W1)^C (R1, W1)^C$

$$\Downarrow (R1, W0)^C (R0, W0)^C; \Downarrow (R0, W0)^C (R0, W0)^C$$

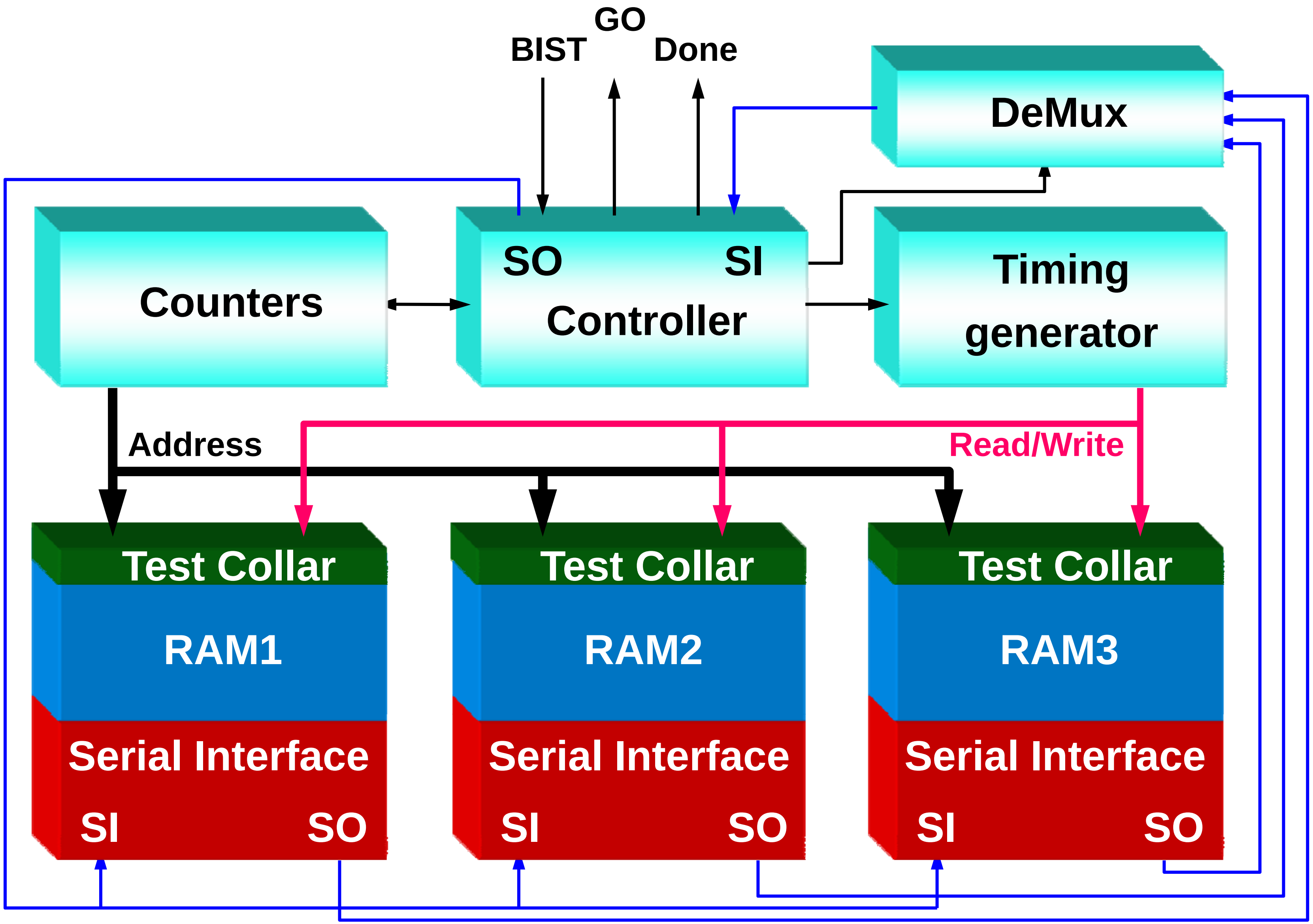
Serial BIST Architecture



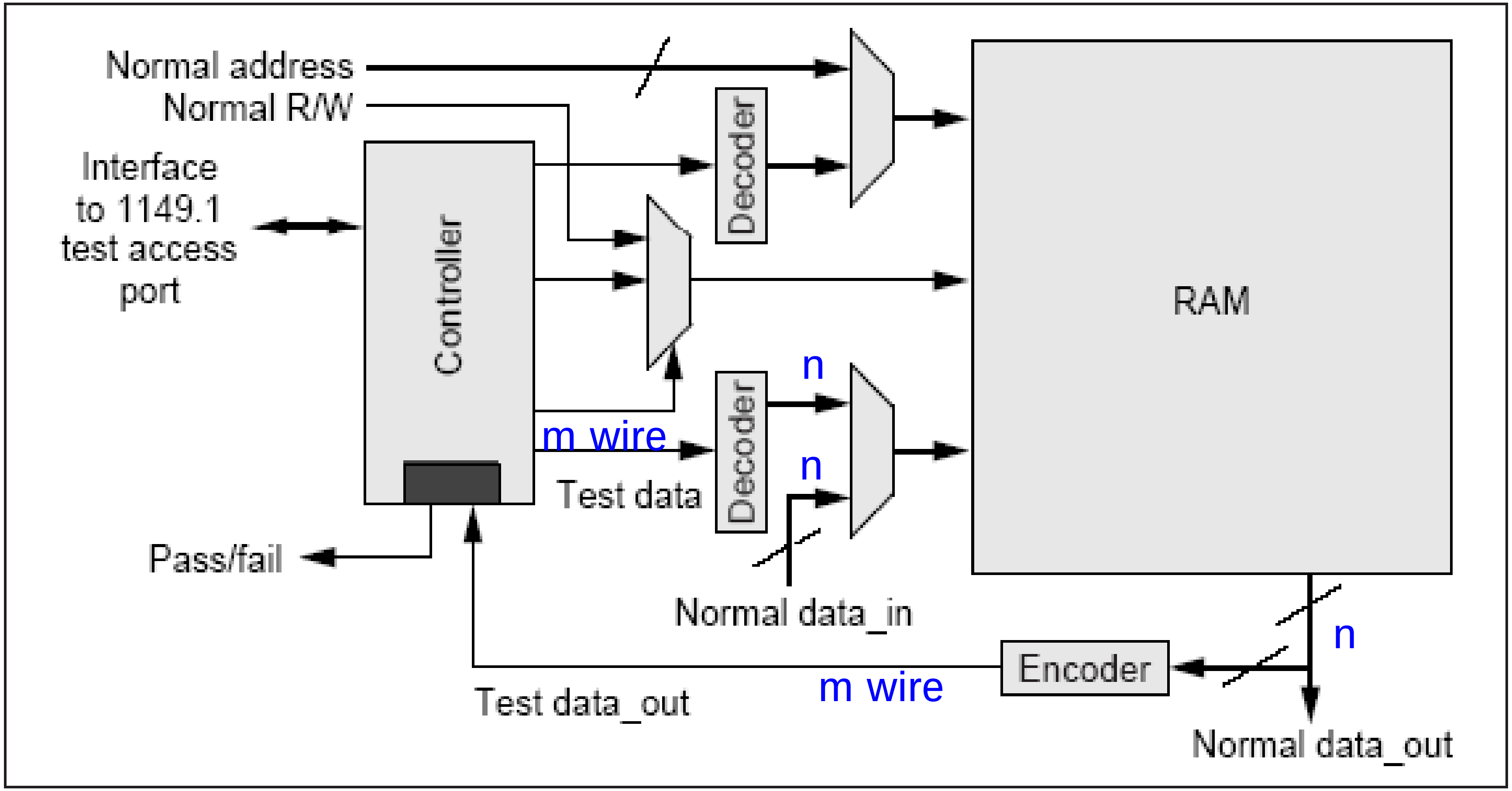
Sharing BIST in Daisy-Chain Style



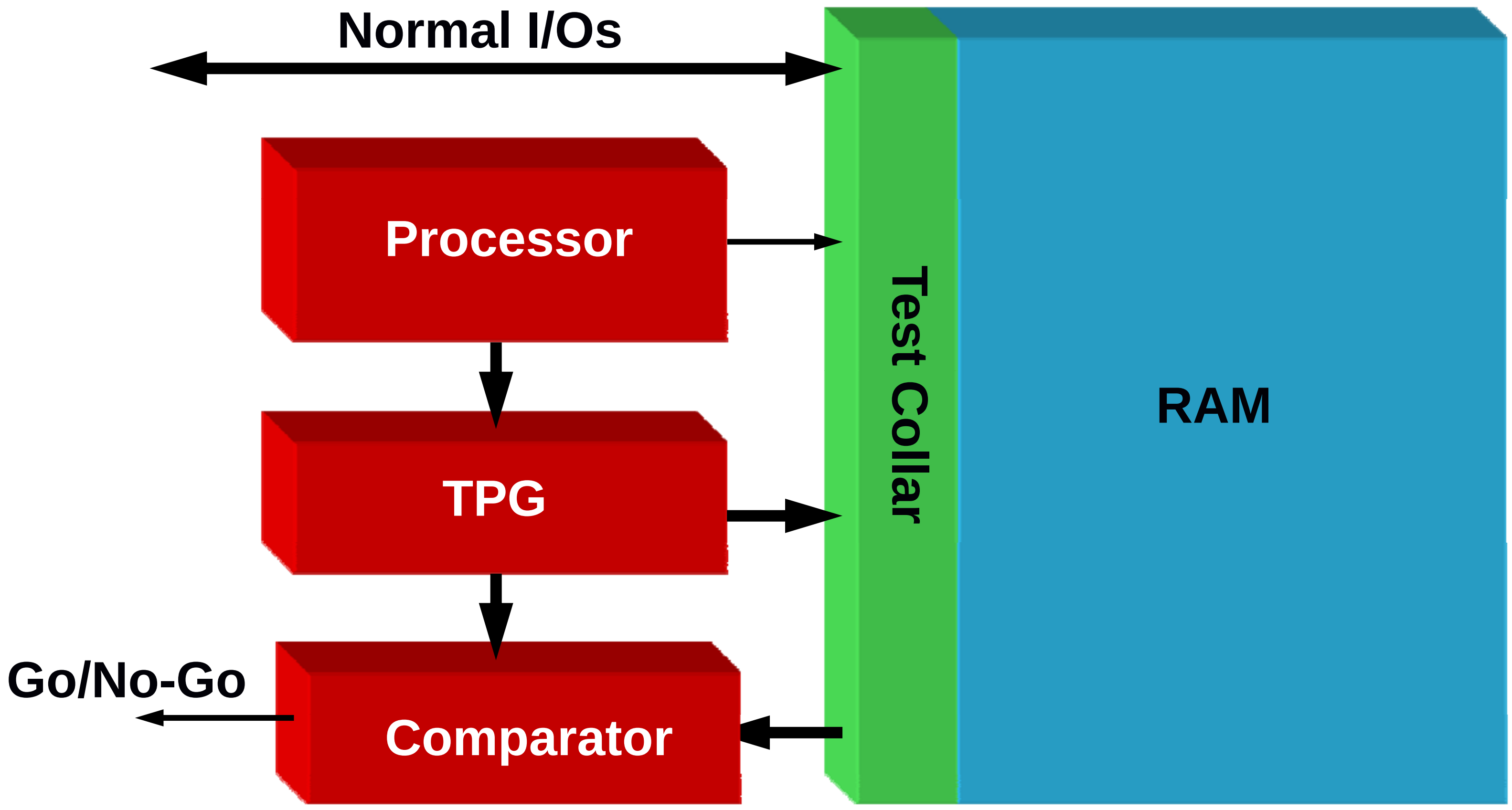
Sharing BIST in Parallel Style



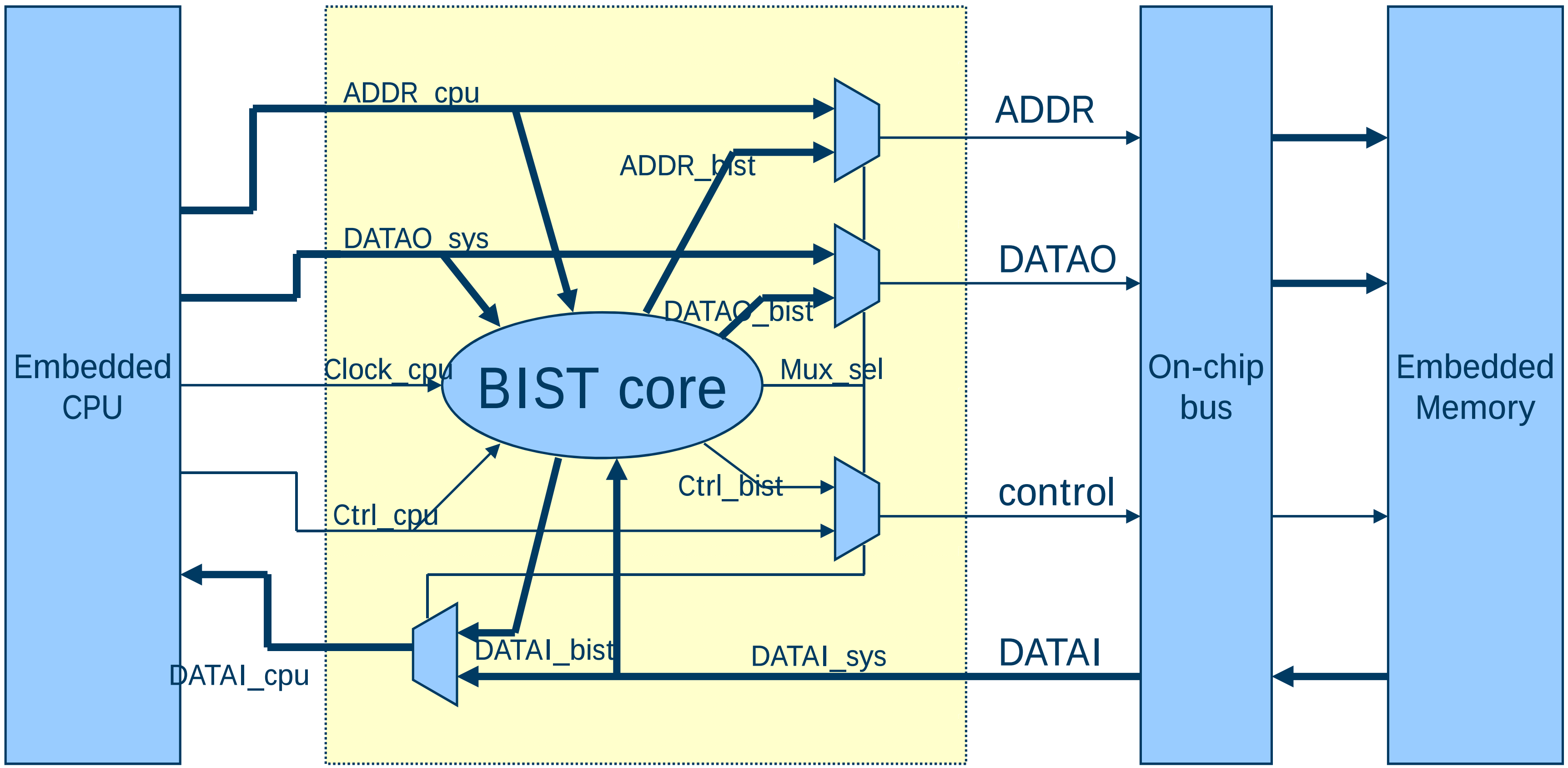
BIST Using Data Decoding/Encoding



Processor-Based RAM BIST



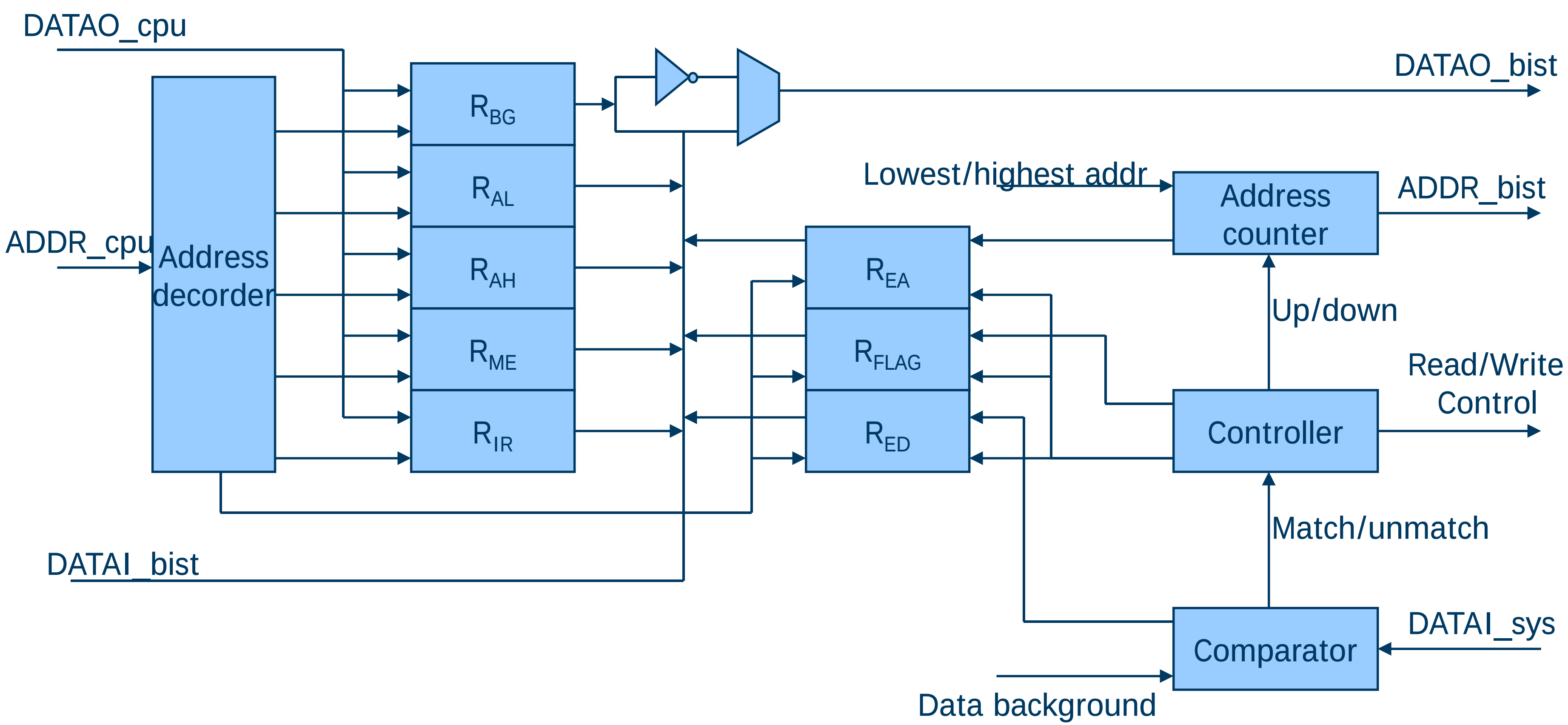
NTHU Processor-Programmable BIST



Source: Prof. C. W. Wu, NTHU

NTHU Processor-Programmable BIST

- BIST core



Source: Prof. C. W. Wu, NTHU

NTHU Processor-Programmable BIST

- Data registers

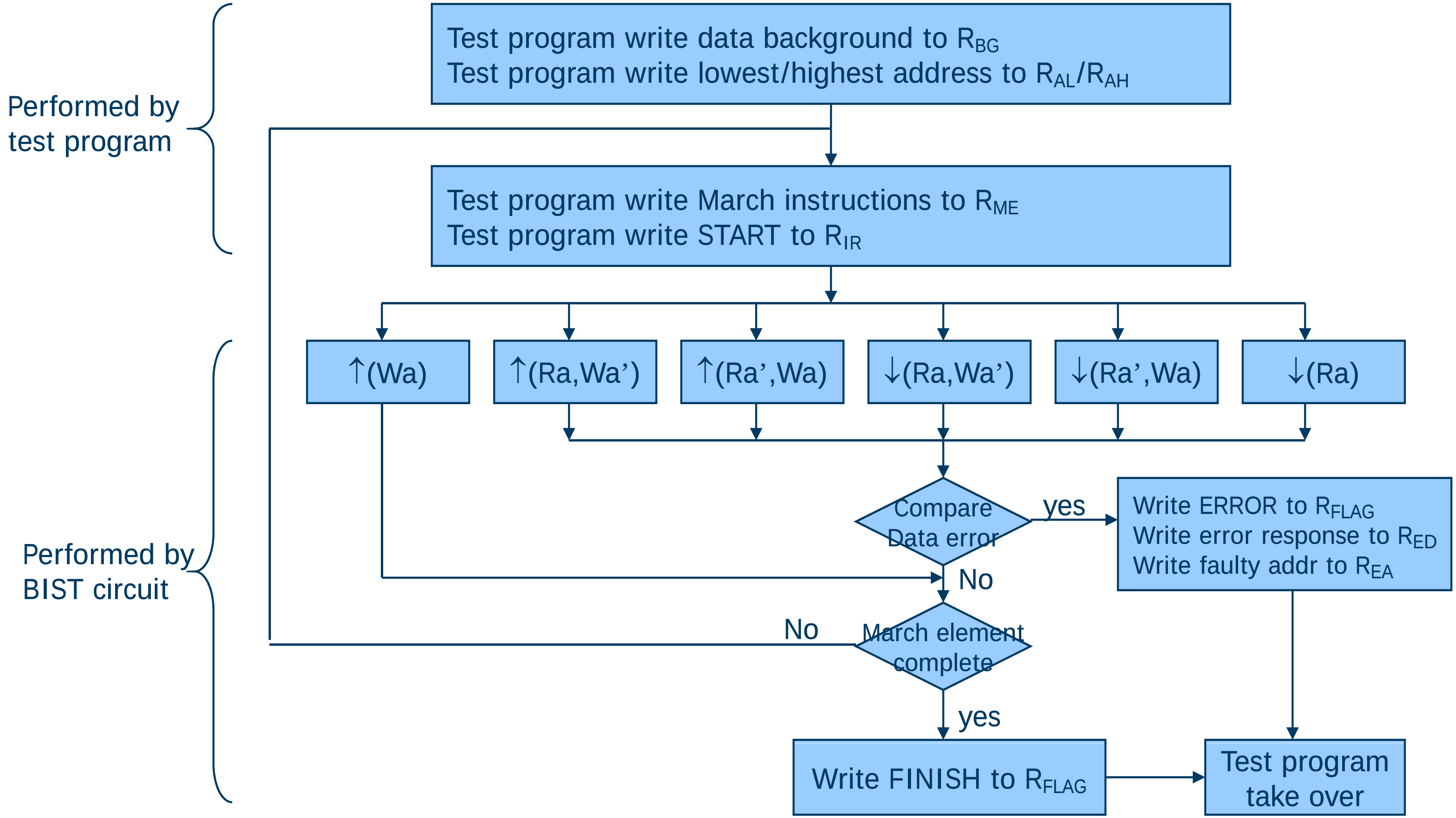
Register	Function
R_{BG}	Store background data
R_{AL}	Store lowest address
R_{AH}	Store highest address
R_{ME}	Store current March element
R_{IR}	Instruction register of BIST circuit
R_{FLAG}	Status register of BIST circuit
R_{ED}	Erroneous response of defective cell
R_{EA}	Address of defective cell

Source: Prof. C. W. Wu, NTHU

NTHU Processor-Programmable BIST

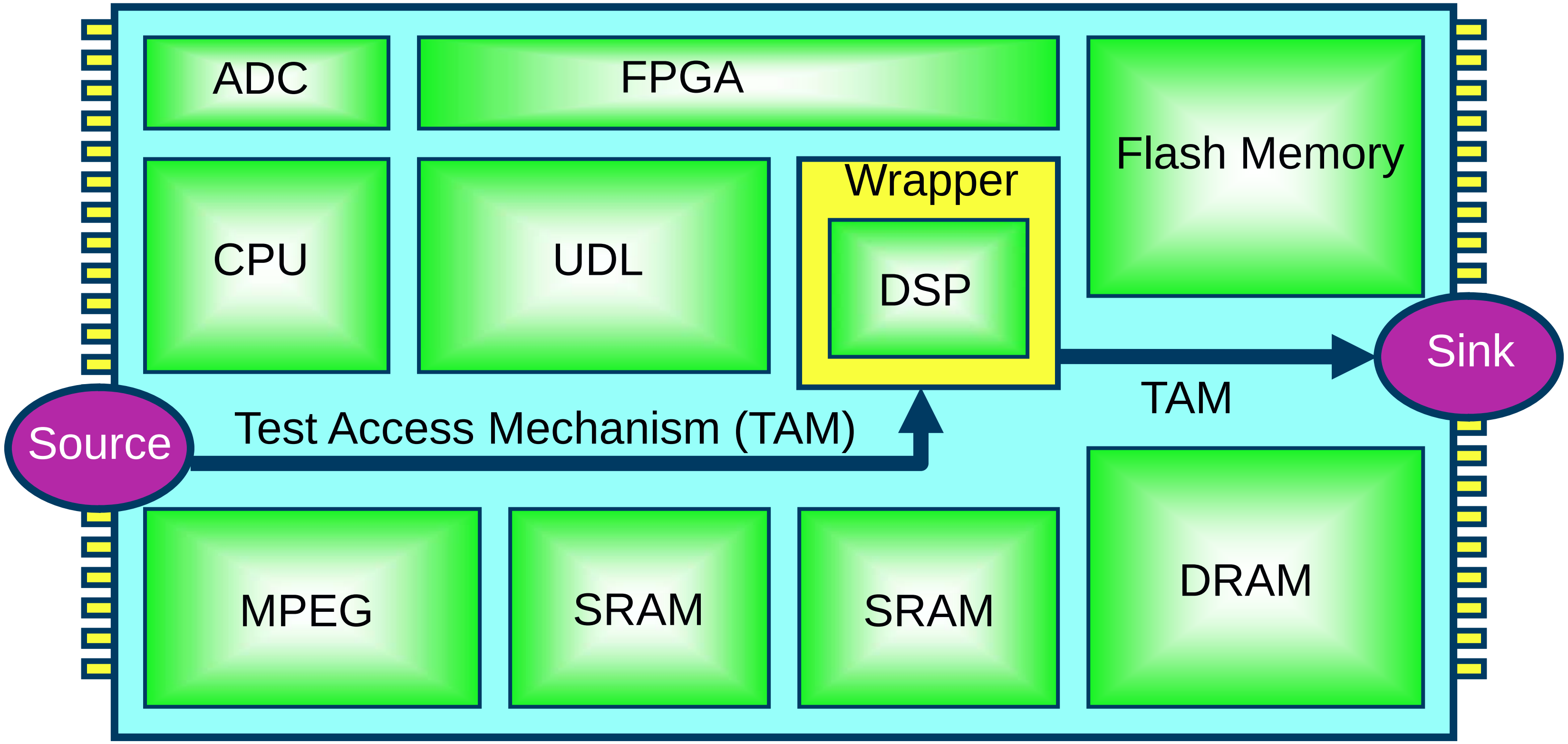
- BIST procedure

Source: Prof. C. W. Wu, NTHU



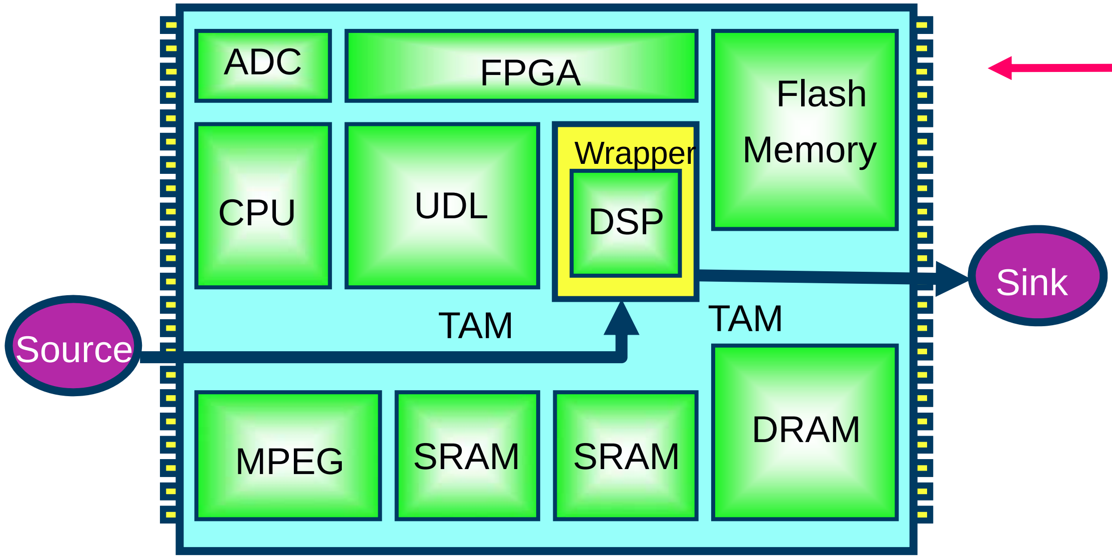
SOC Testing

- A typical SOC chip



Source: Y. Zorian, et al.-ITC98

SOC Test Access

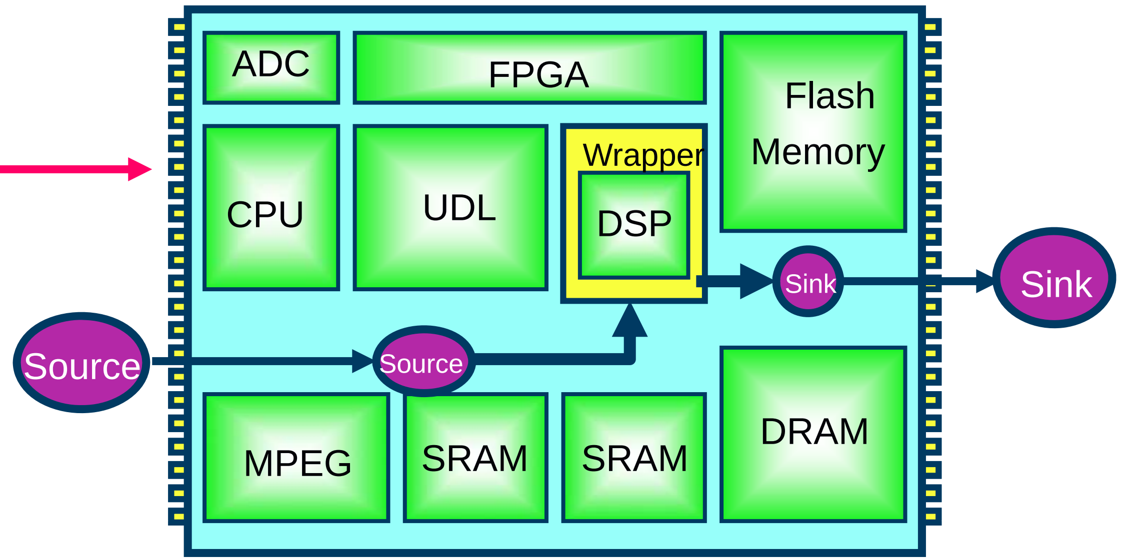


Off-chip Source/Sink

1. Pins determine bandwidth
2. More TAM area
3. Requires expensive ATE

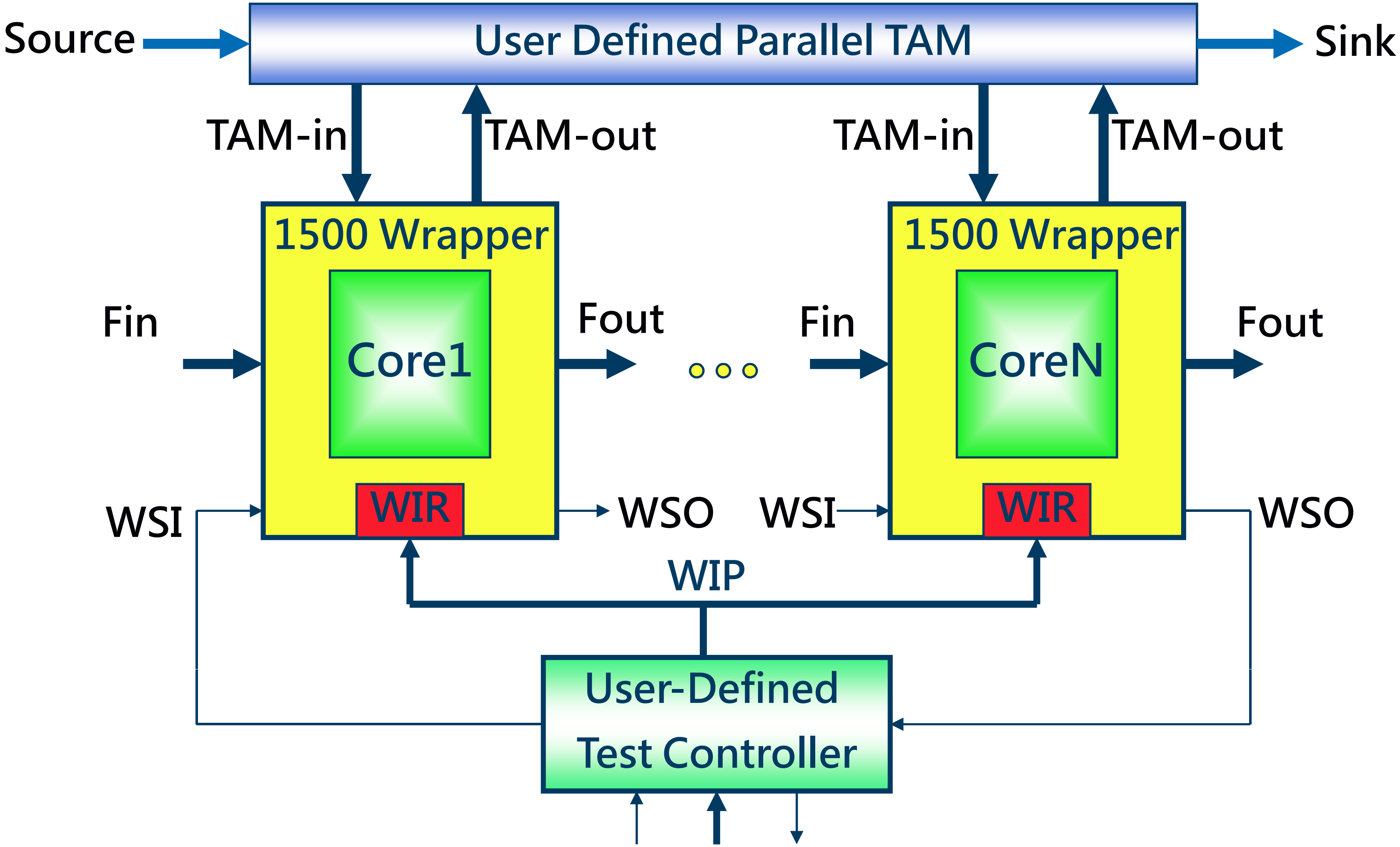
On-chip Source/Sink

1. Close to Core-under-test
2. Less TAM area
3. BIST IP area
4. Requires lightweight ATE

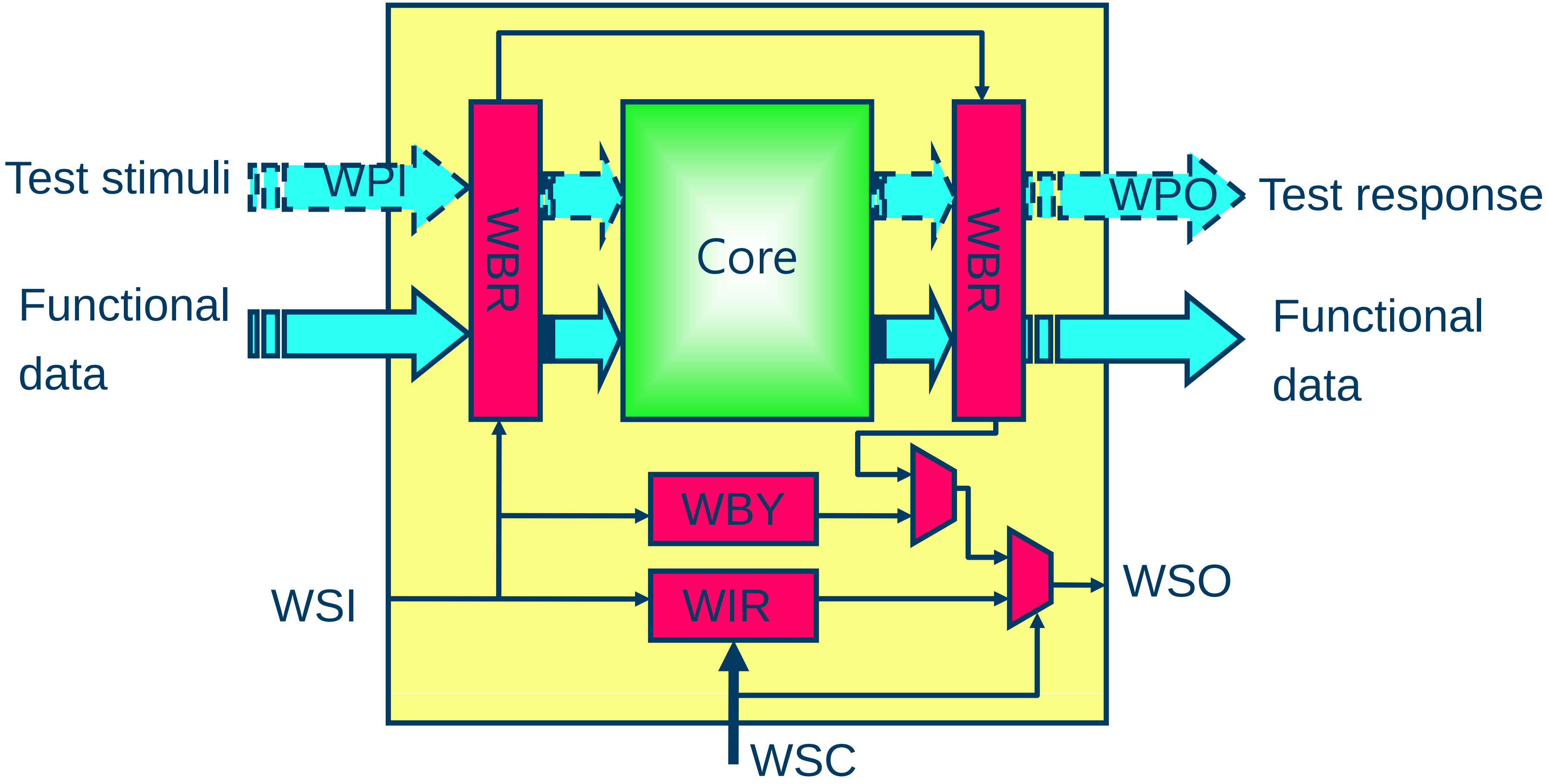


Source: Y. Zorian and E.J. Marinissen

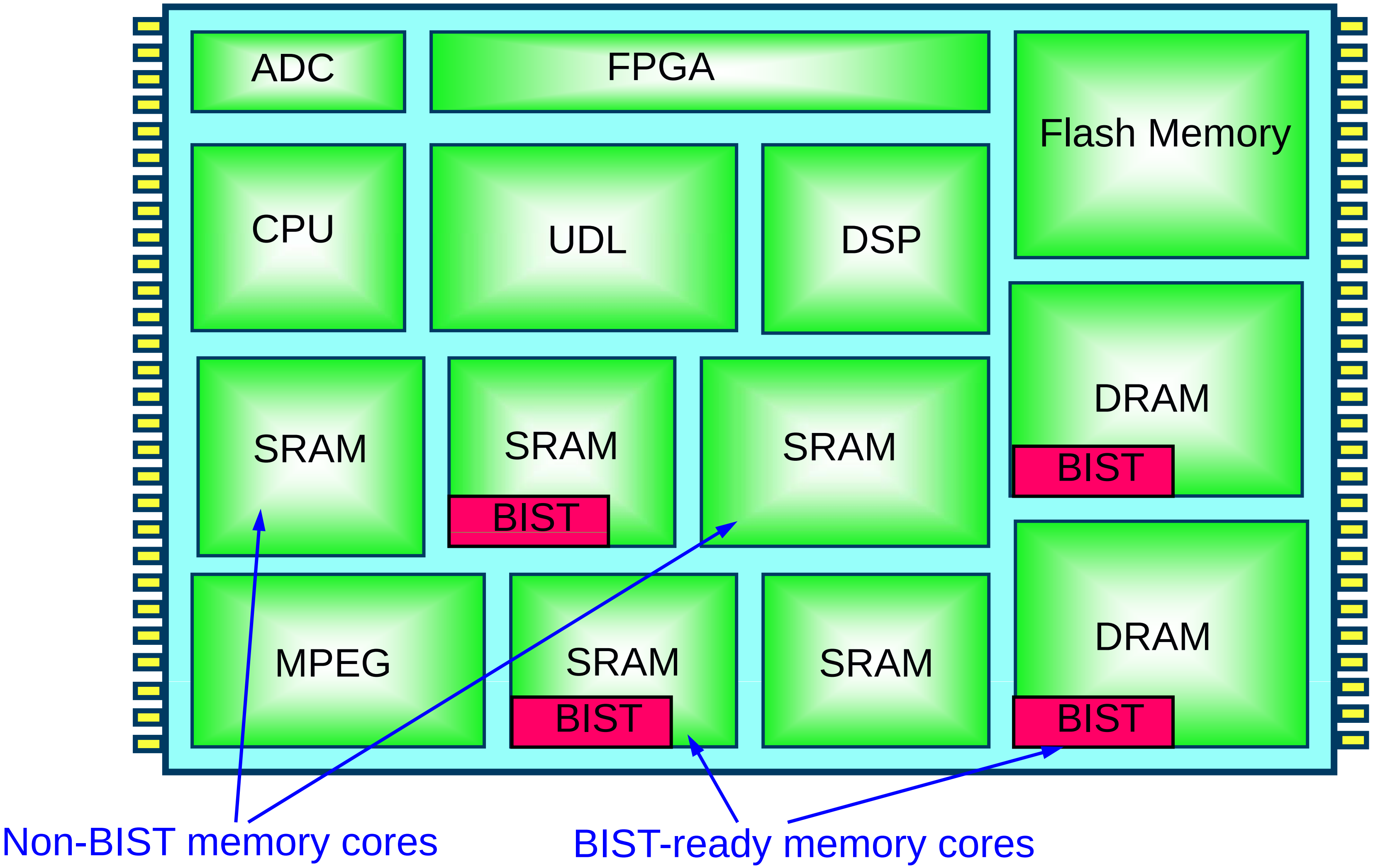
1500 Scalable Test Architecture



1500 Test Wrapper



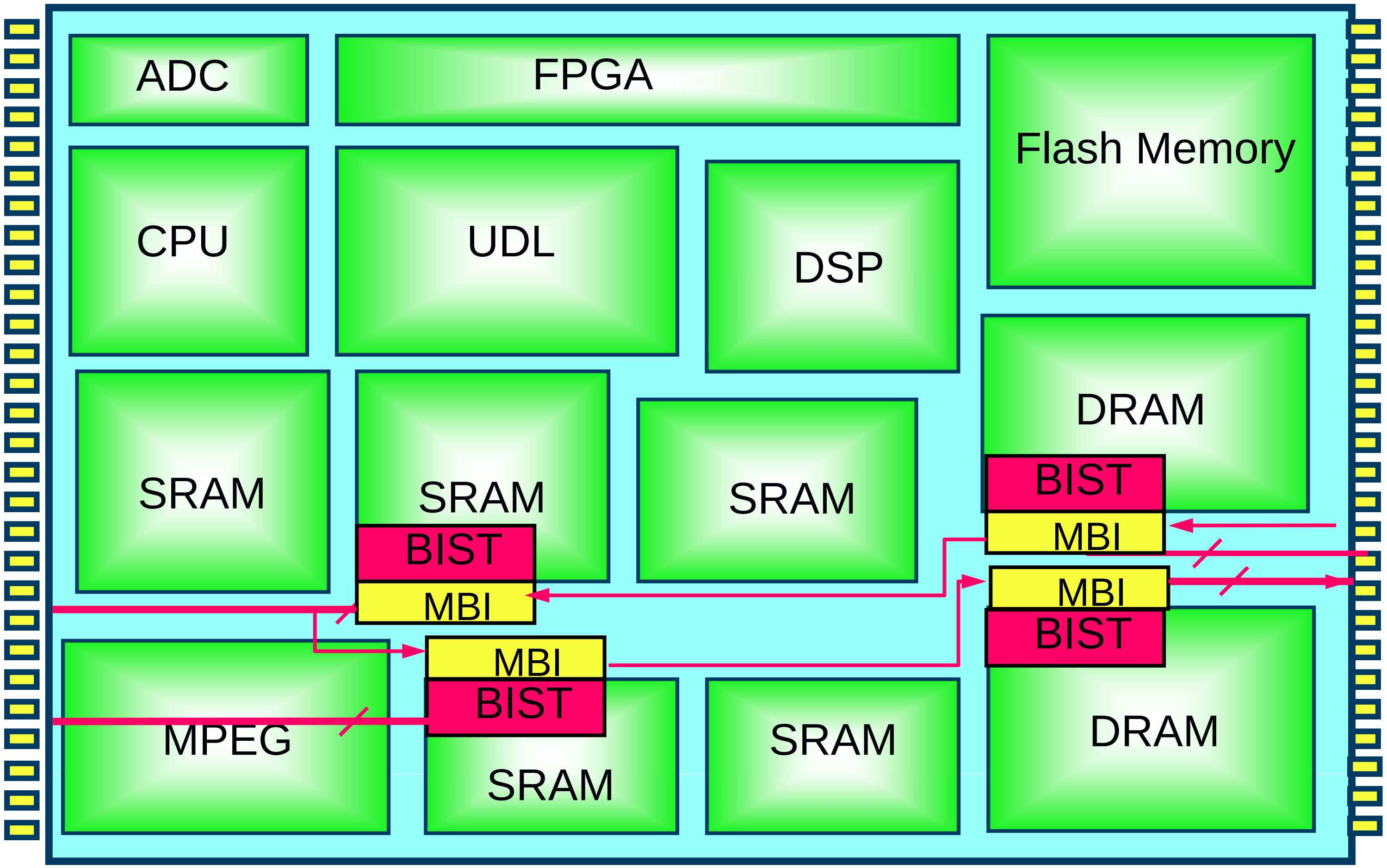
Memories in SOCs



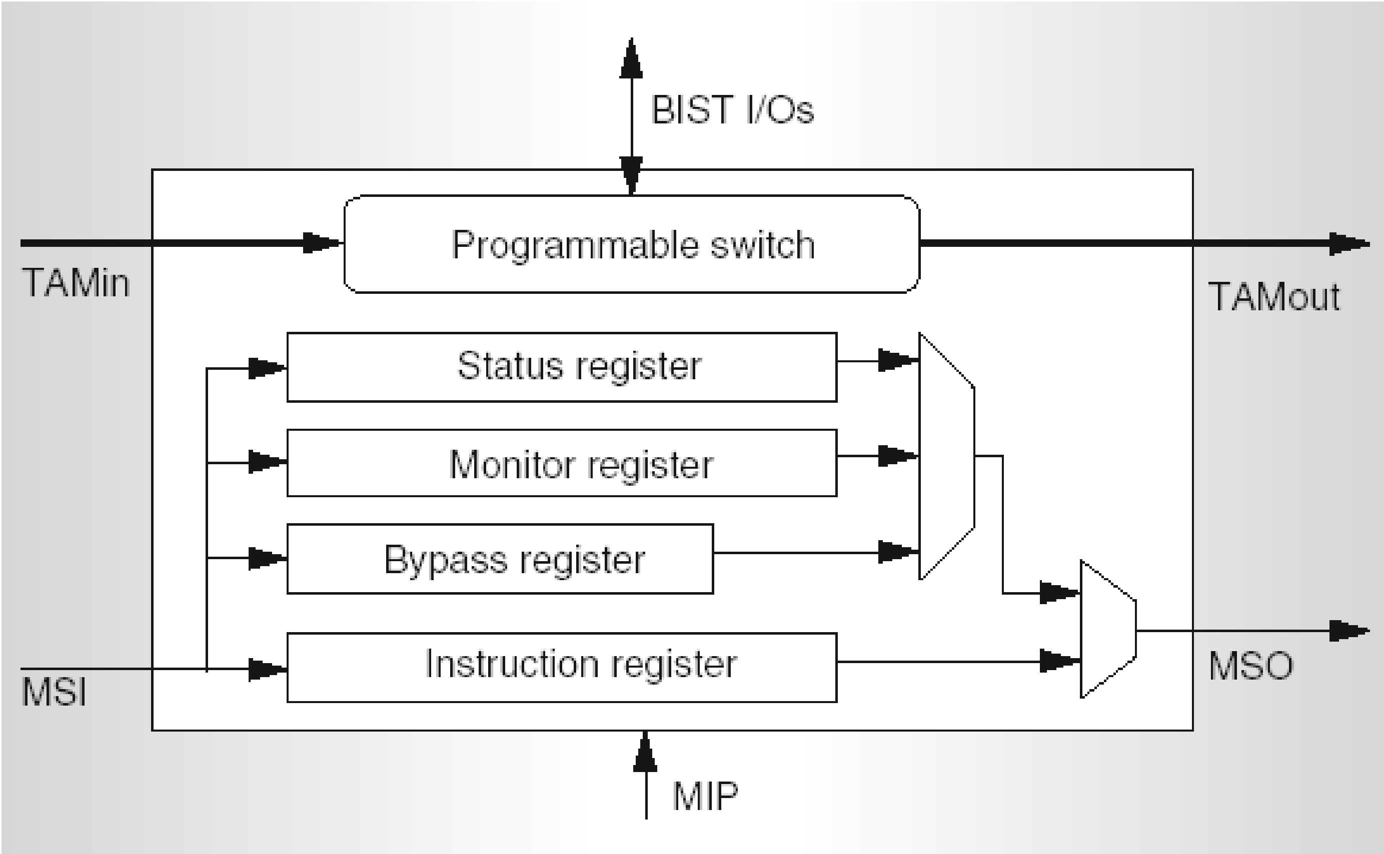
RAM BIST in SOCs

- Memory cores in an SOC can be categorized into two types in term of testability
  - BIST-ready memory cores
  - Non-BIST memory cores
- An SOC can contain tens or even hundreds of memory cores
  - Although a BIST usually have only about 8 controlling pins
  - The total BIST controlling pins is huge if each BIST-ready memory cores has its own BIST controlling pins
- The BIST controlling pins should be shared
  - One solution is using memory BIST interface

Sharing BIST Controlling Pins



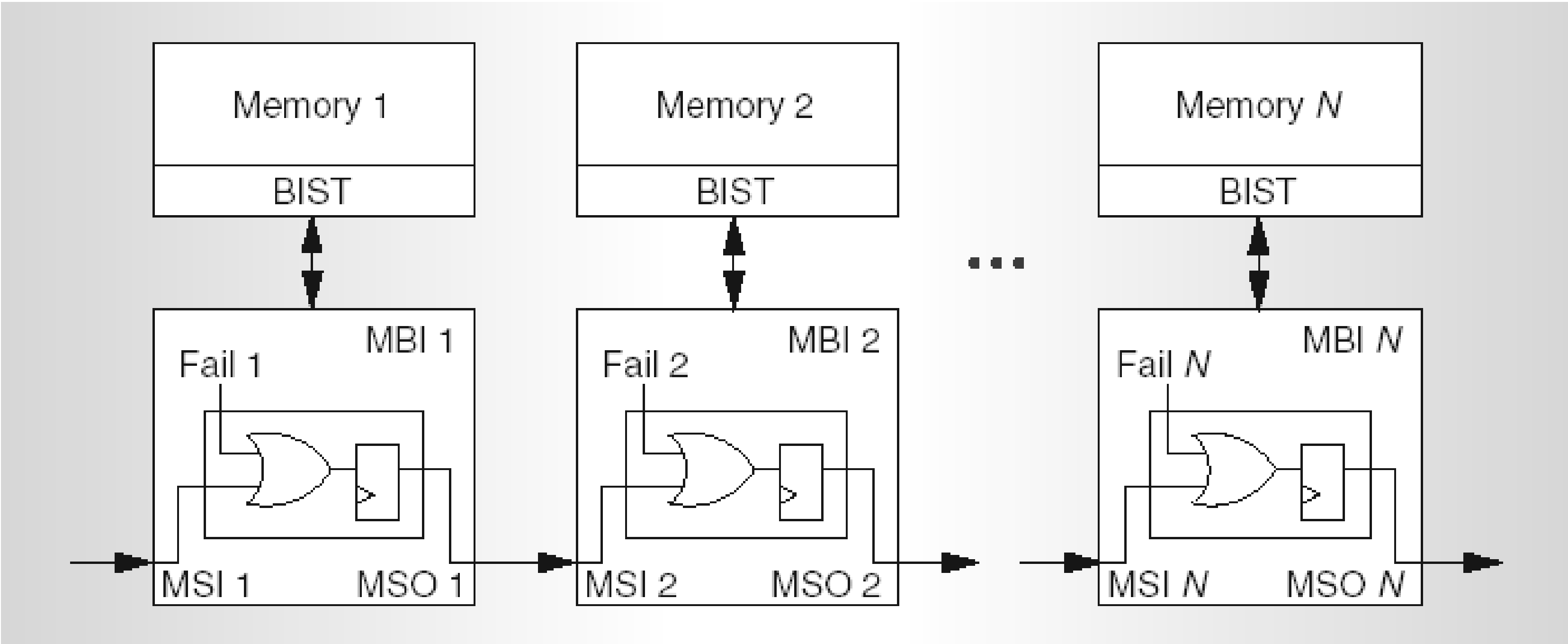
Memory BIST Interface (MBI)



Memory BIST Interface (MBI)

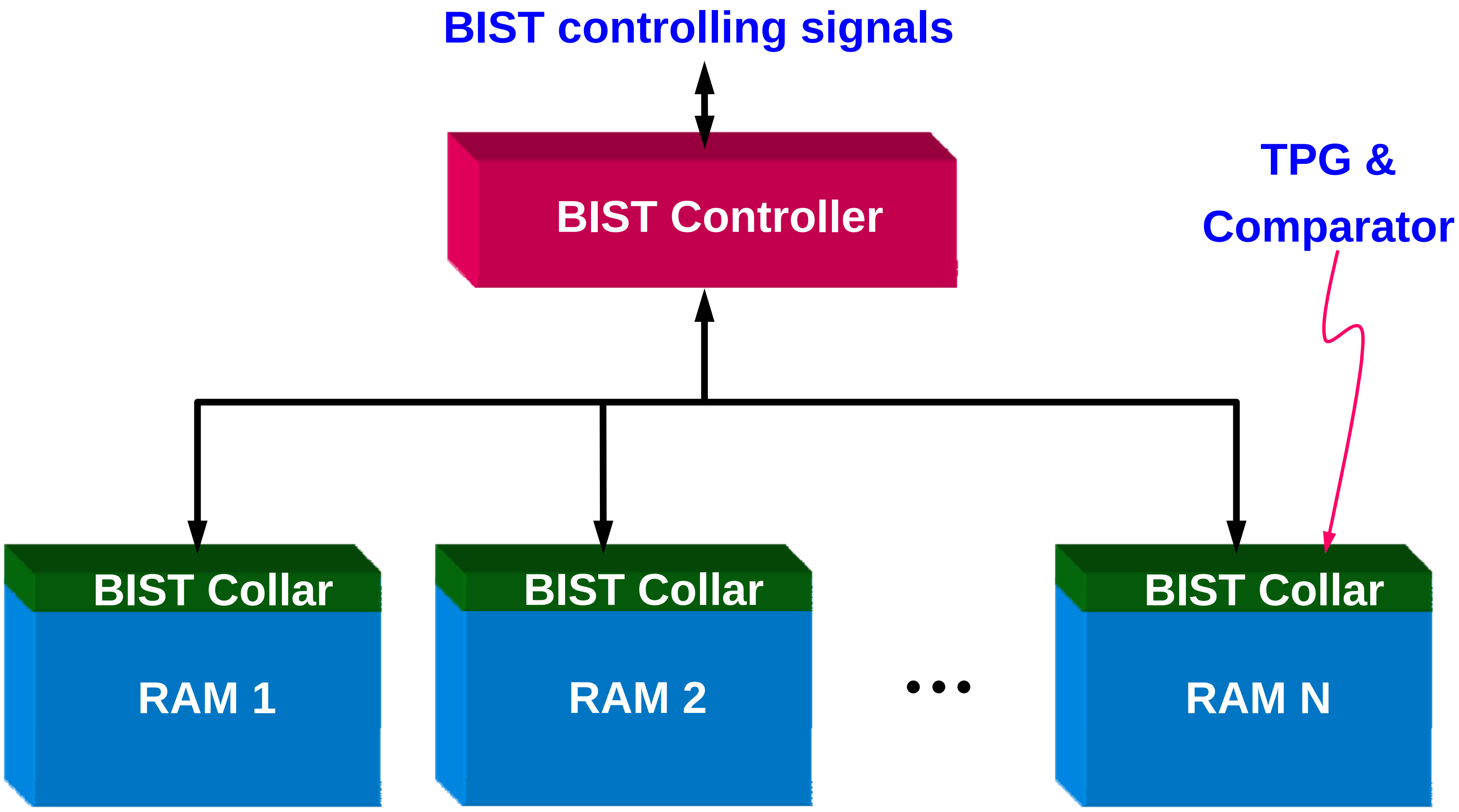
- Instruction Register
  - Store the instructions
    - RUN_BIST, RUN_DIAGN, EXPORT_STATUS, TAM_CONTROL
- Bypass Register
  - It is selected if the corresponding memory core is not tested
- Monitor Register
  - Monitor the error flag (indicating whether a memory fault is detected or not)
- Status Register
  - Record the key status values, such as Fail output from the BIST

Testing Multiple Memories with MBI

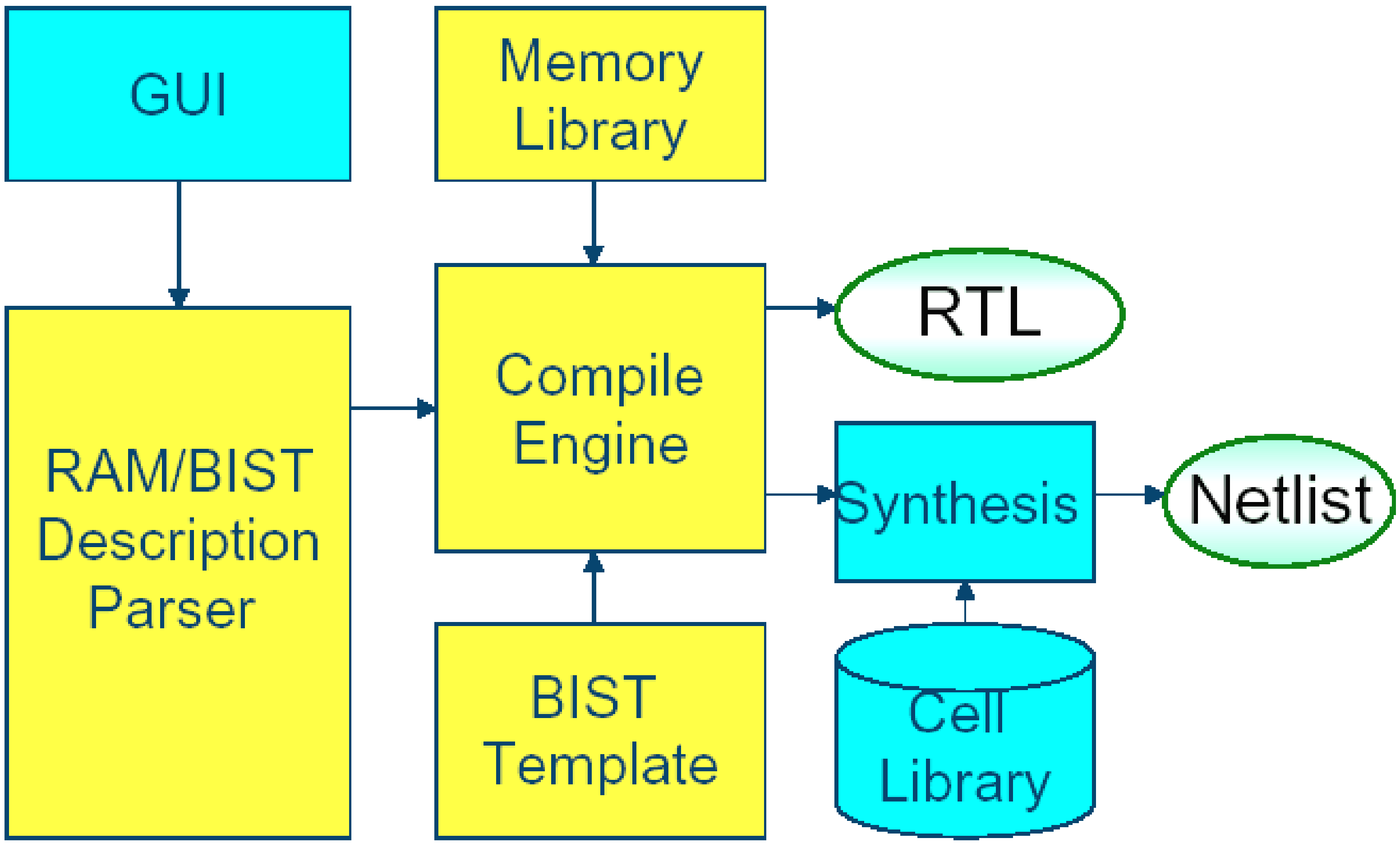


- Using RUN_BIST instruction, we can test multiple memory cores concurrently
  - When one or more BIST circuits detect faults, the primary MSO_N will be high after N-K clock cycles if the concurrent output of the (K+1) through the N memory cores are fault free

Sharing BIST Hardware

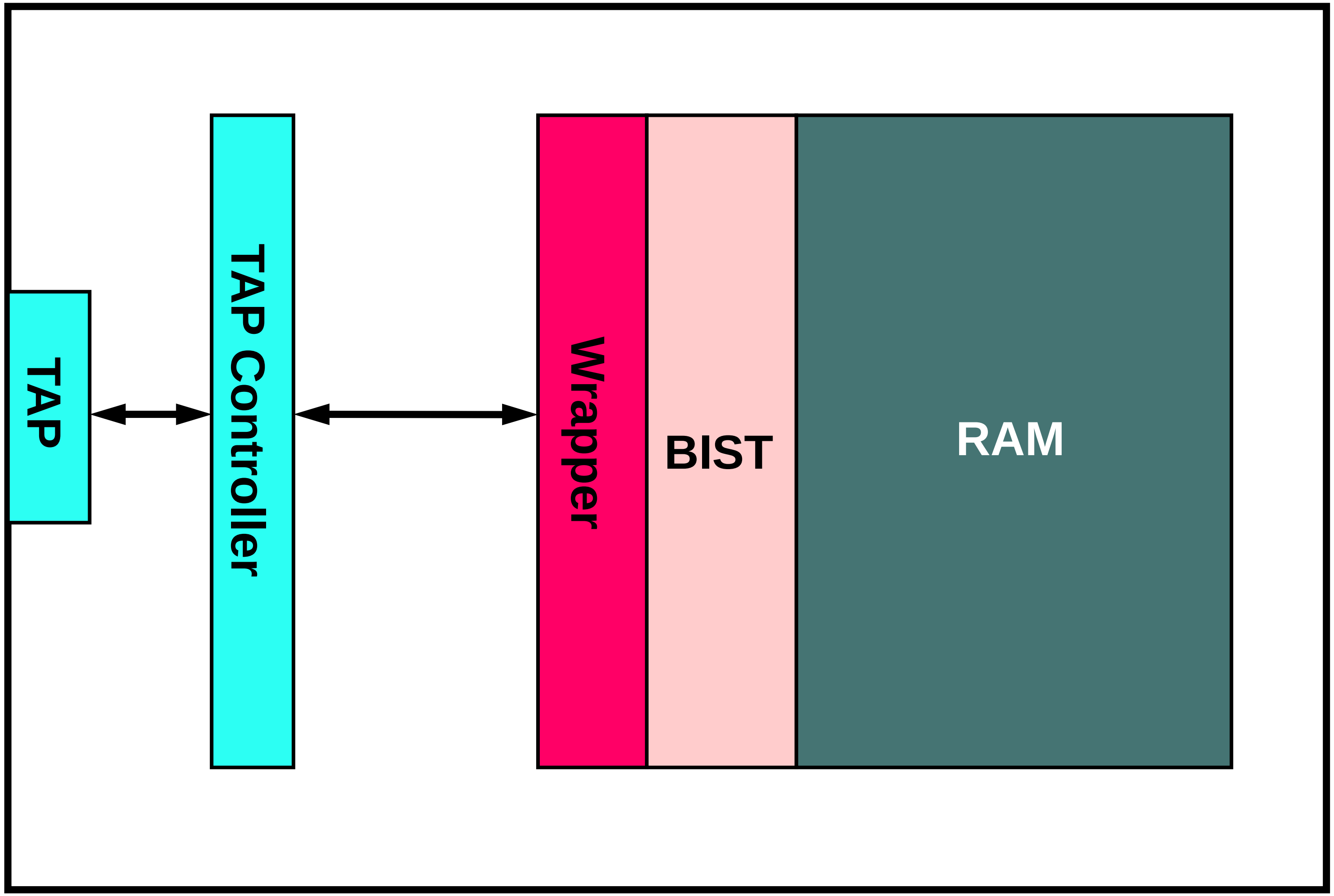


RAM BIST Compiler



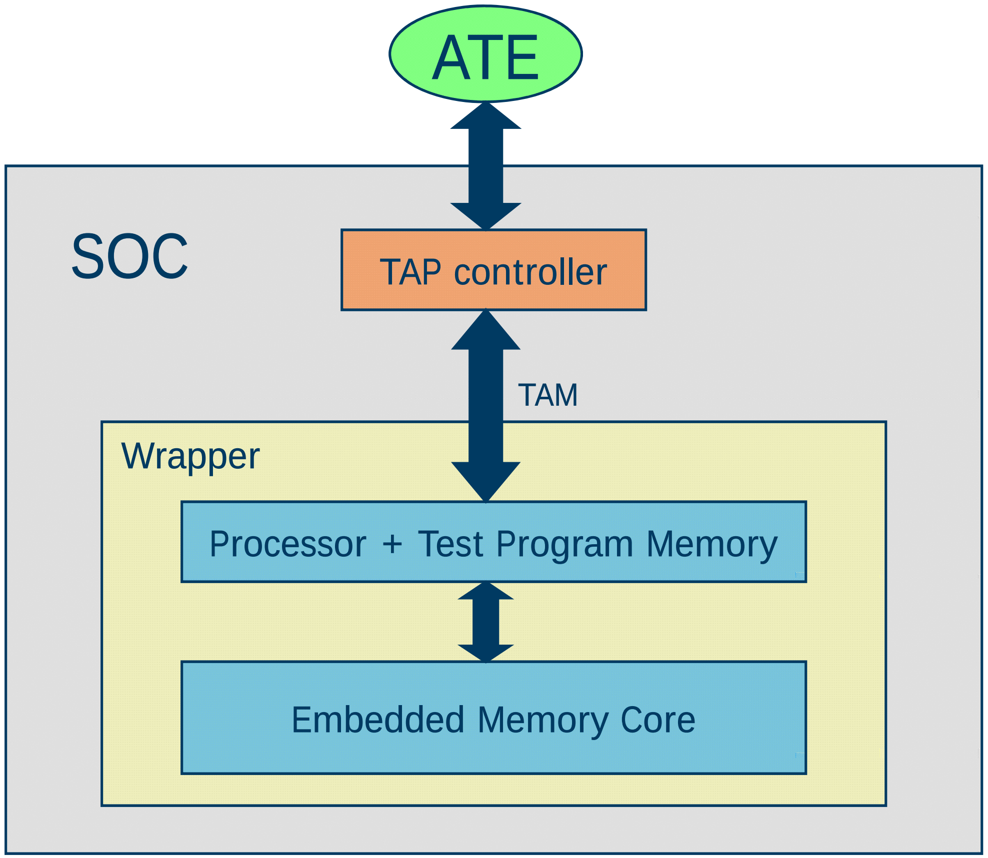
Source: Prof. C. W. Wu, NTHU

1500-Compilant BIST



Programmable BISD for RAMs in SOCs

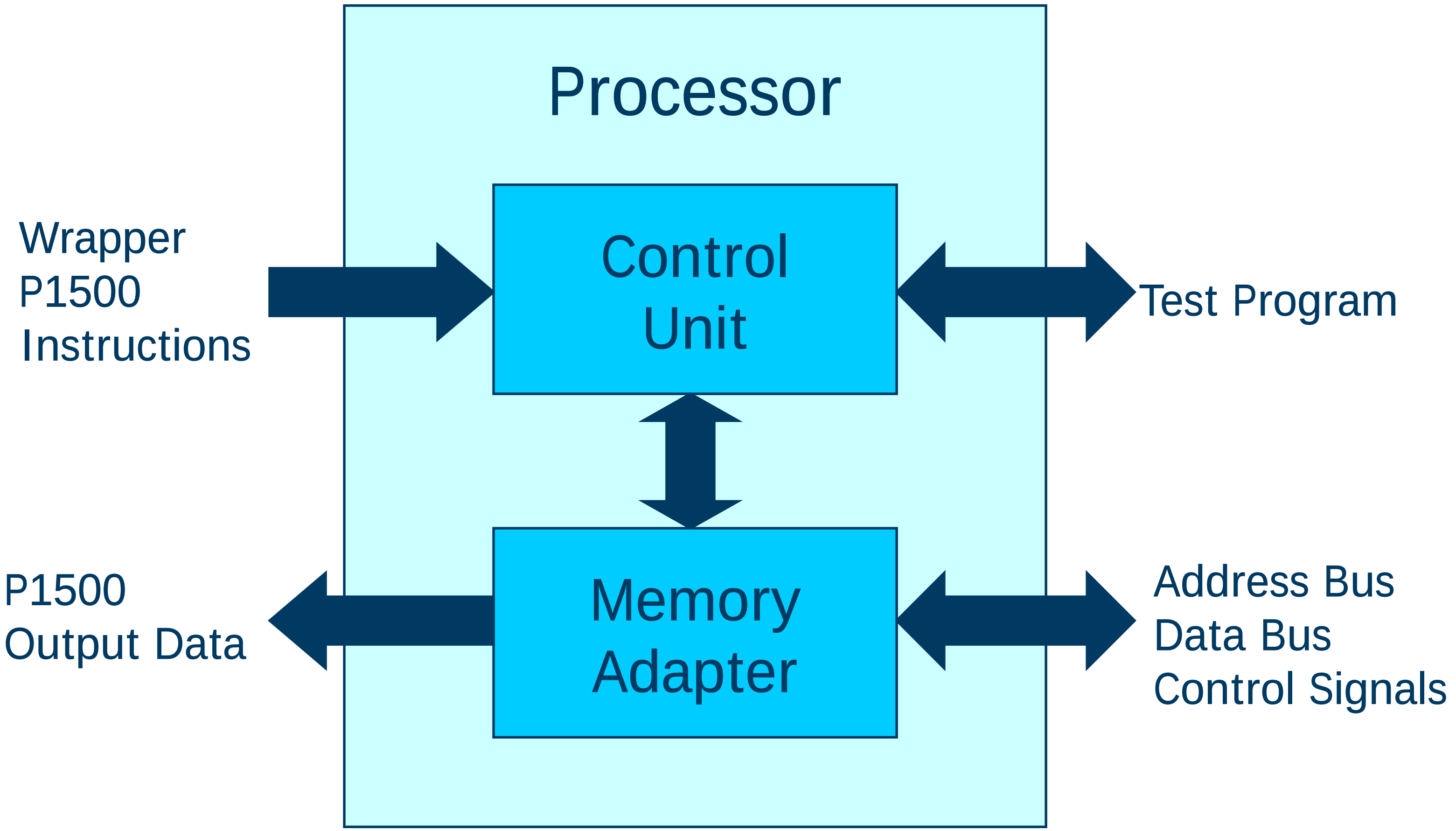
- General test architecture



Source: Appello D., et. al. ITC03

Programmable BISD for RAMs in SOCs

- Test Processor



Source: Appello D., et. al. ITC03

Programmable BISD for RAMs in SOCs

- Control Unit
  - Manage the test program execution
  - Include an Instruction Register (IR) and Program Counter (PC)
  - The control unit allows the correct update of some registers located in Memory Adapter
  - This part simplifies the processor reuse in different applications without the need for any re-design

Programmable BISD for RAMs in SOCs

- Memory Adapter
  - Control Address registers (Current_address)
  - Control Memory registers
    - Current_data
    - Received_data
  - Control Test registers
    - Dbg_index, Step, Direction flag, and Timer registers
  - Result registers
    - Status, Error, and Result registers
  - Some constant value registers
    - Add_Max, Add_Min, DataBackGround, Dbg_max

Programmable BISD for RAMs in SOCs

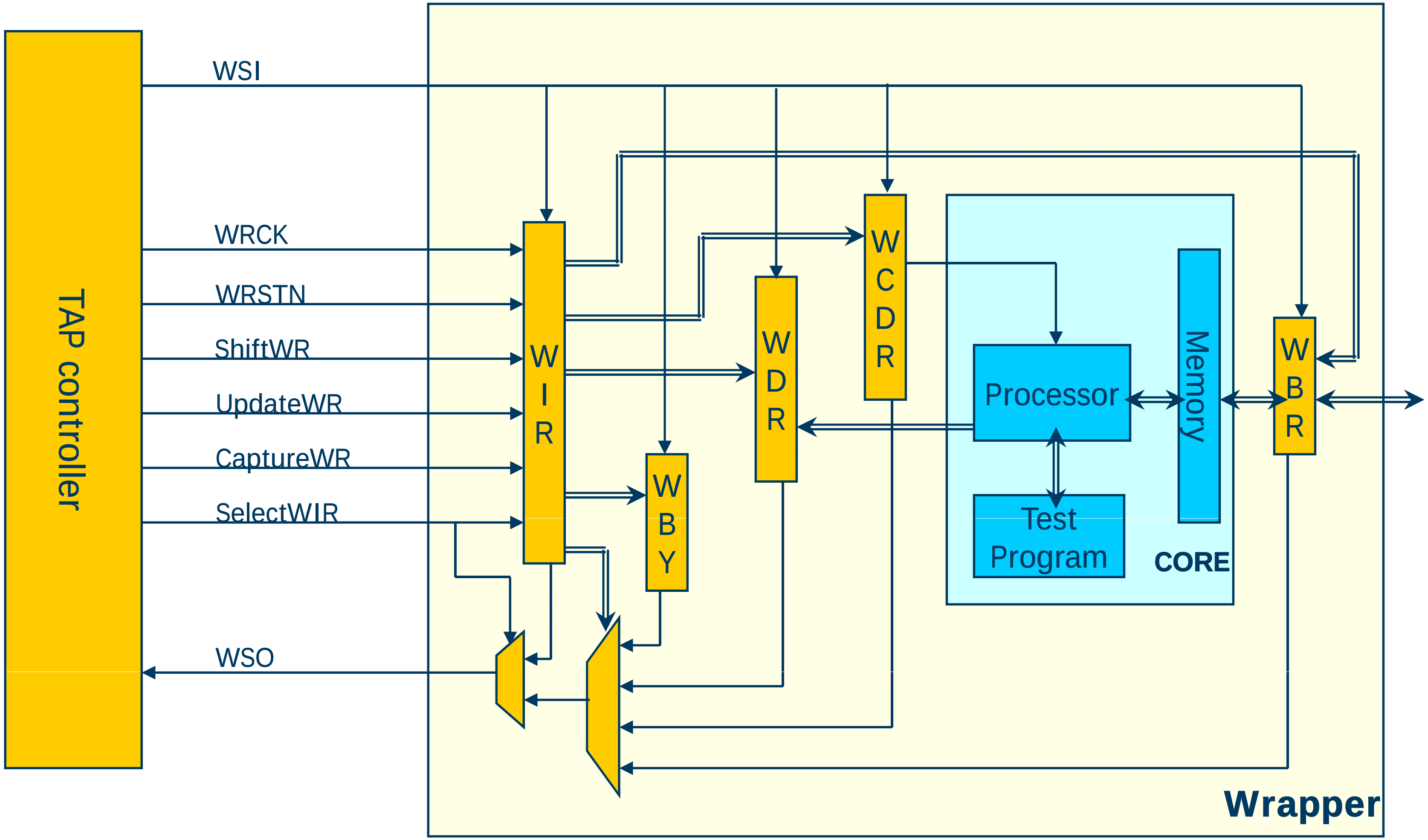
- Processor instruction set

Instruction	*Meaning*
SET_ADD	$Current\_address \Leftarrow Add\_Max$ $Direction\ flag \Leftarrow BACKWARD$
RST_ADD	$Current\_address \Leftarrow Add\_Min$ $Direction\ flag \Leftarrow FORWARD$
STORE_DBG	$Current\_data \Leftarrow DataBackGround[Dbg\_index]$ $Dbg\_index \Leftarrow Dbg\_index+1$
INV_DBG	$Current\_data \Leftarrow$ NOT $(Current\_data)$
READ	$Current\_data \Leftarrow Memory[Current\_address]$
WRITE	$Memory[Current\_address] \Leftarrow Current\_data$

Source: Appello D., et. al. ITC03

Programmable BISD for RAMs in SOCs

- Wrapper



Source: Appello D., et. al. ITC03

Summary

- ROM BIST has been presented
- ROM-based and FSM-based RAM BIST have been introduced
- Serial BIST methodology for embedded memories has also presented
- BIST approaches for testing multiple RAMs in an SOC have also been addressed

References

- [1] A. K. Sharma,"Semiconductor memories – technology, testing, and reliability", IEEE Press, 1997.
- [2] B. Nadeau-Dostie, A. Silburt, and V. K. Agarwal,"Serial interfacing for embedded-memory testing", IEEE D&T, pp.52-63, apr. 1990.
- [3] C. W. Wu,"VLSI testing & design for testability II: Memory built-in self-test", http://larc.ee.nthu.edu.tw/~cww/
- [4]C.-H. Tsai and C.-W. Wu, ``Processor-programmable memory BIST for bus-connected embedded memories'', in Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC), Yokohama, Jan. 2001, pp. 325-330
- [5]C.-W. Wang, C.-F. Wu, J.-F. Li, C.-W. Wu, T. Teng, K. Chiu, and H.-P. Lin, ``A built-in self-test and self-diagnosis scheme for embedded SRAM'', in Proc. 9th IEEE Asian Test Symp. (ATS),Taipei, Dec. 2000, pp. 45-50
- [6]D. Appello, F. Corno, M. Giovinetto, M. Rebaudengo, and M. S. Reorda,"A P1500 compliant BIST-Based approach ro embedded RAM diagnosis", pp.97-102, MTDT, 2001.
- [7]J. F. Li, H. J. Huang, J. B. Chen, C. P. Su, C. W. Wu, C. Cheng, S. I. Chen, C. Y. Hwang, and H. P. Lin,"A hierarchical test methodology for systems on chip", IEEE Micro, pp. 69-81, Sep./Oct., 2002.
- [8]S. Mourad and Y. Zorian,"Principles of Testing Electronic Systems", John Wiley & Sons, 2000.