SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB302–VLSI DESIGN

III YEAR/ V SEMESTER

UNIT 4 –VLSI TESTING

TOPIC 6 –BIST

OUTLINE

- INTRODUCTION
- BASIC CONCEPT OF TESTING
- PRINCIPLE OF TESTING
- DIFFICULTIES IN TESTING
- HOW TO DO TESTING
- CIRCUIT MODELING
- AUTOMATIC TEST PATTERN GENERATION (ATPG)
- DIFFICULTIES IN TEST GENERATION-2 TYPES
- TESTABLE DESIGN
- ACTIVITY
- BUILT-IN-SELF TEST (BIST)
- RANDOM NUMBER GENERATOR (RNG)
- SIGNATURE ANALYZER (SA)
- MEMORY BIST ARCHITECTURE
- CPU TEST CONTROL ARCHITECTURE
- TESTING METHODS
- ASSESSMENT
- SUMMARY & THANK YOU

BASIC CONCEPT OF TESTING

Testing: To tell whether a circuit is good or bad

Related fields

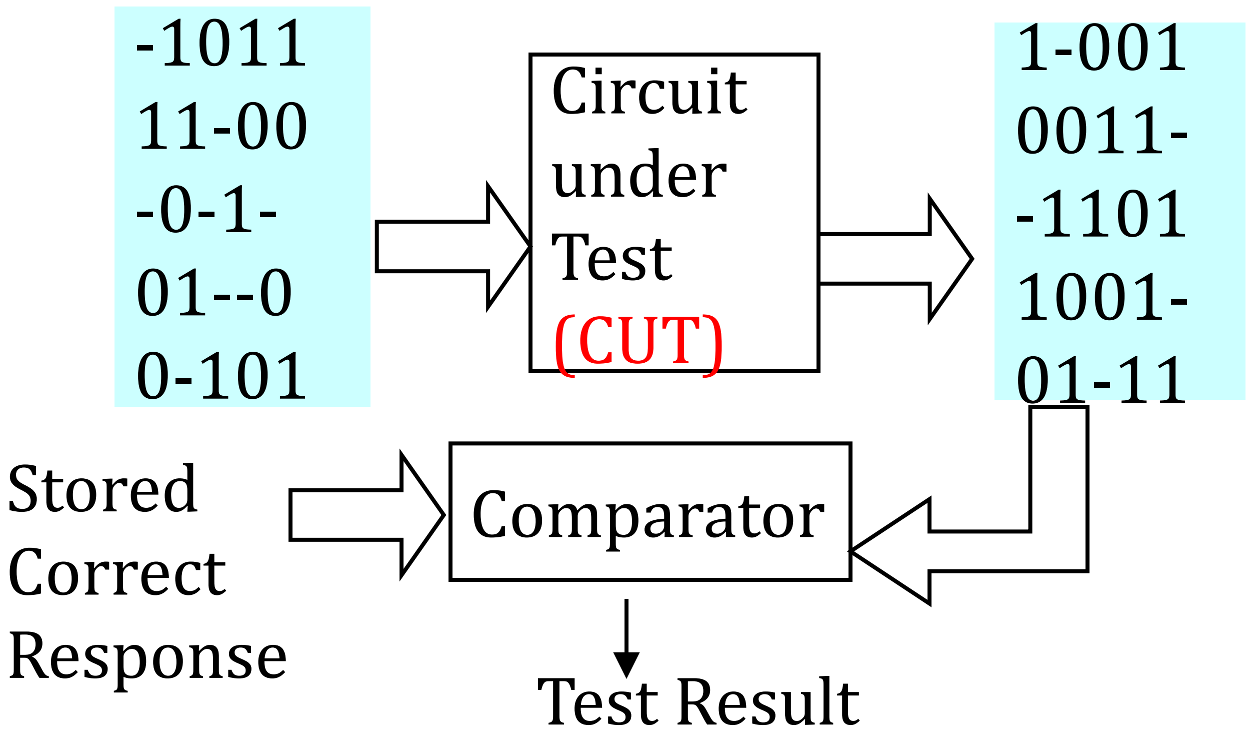
Verification: To verify the correctness of a design

Diagnosis: To tell the faulty site

Reliability: To tell whether a good system will work correctly or not after some time.

Debug: To find the faulty site and try to eliminate the fault

PRINCIPLE OF TESTING

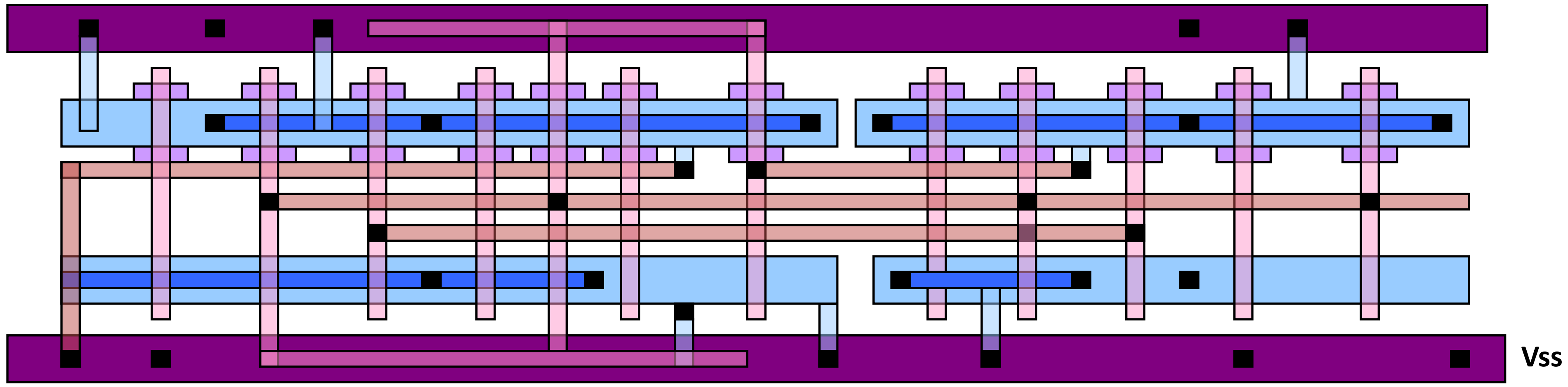


- Testing typically consists of
  - Applying set of test stimuli (input patterns, test vectors) to inputs of circuit under test (CUT), and
  - Analyzing output responses
- The quality of the tested circuits will depend upon the thoroughness of the test vectors

DIFFICULTIES IN TESTING

- Fault may occur anytime
  - Design
  - Process
  - Package
  - Field

- Fault may occur at any place



- **VLSI circuit are large**
  - Most problems encountered in testing are NP-complete
- **I/O access is limited**

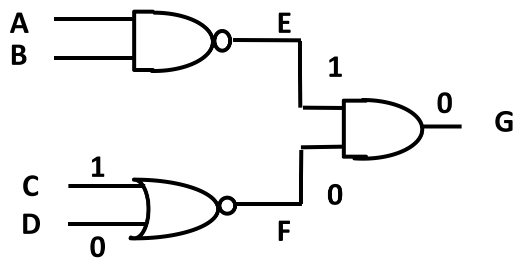
HOW TO DO TESTING

From designer’s point of view:

• Circuit modeling • Fault modeling	Modeling
• Logic simulation • Fault simulation • Test generation	ATPG
• Design for test • Built-in self test	Testable design
• Synthesis for testability	

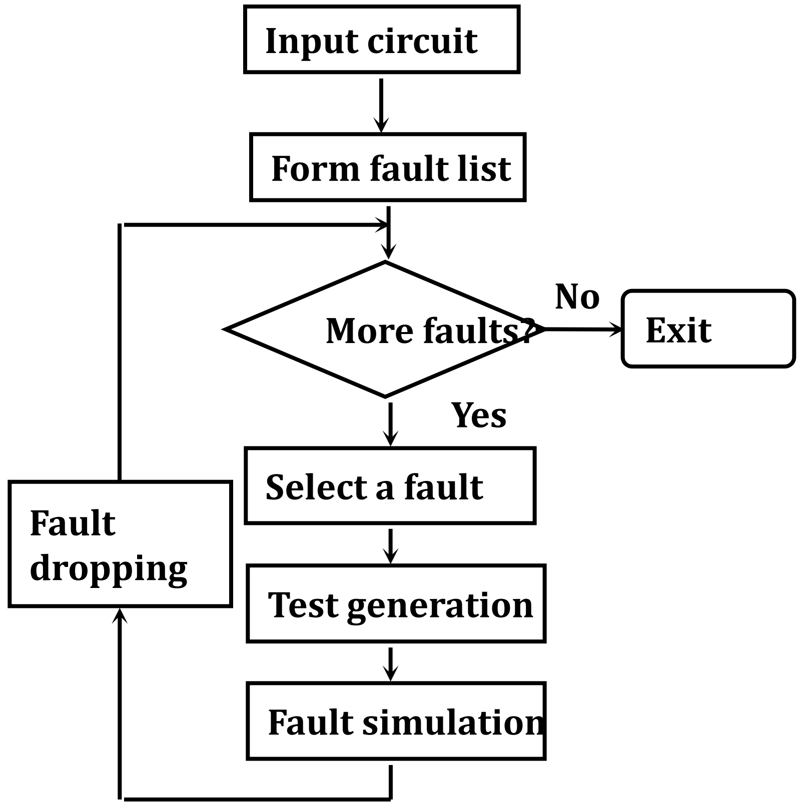
CIRCUIT MODELING

- **Functional model---** logic function
  - **f(x1,x2,...)=...**
  - **Truth table**
- **Behavioral model--- functional + timing**
  - **f(x1,x2,...)=... , Delay = 10**
- **Structural model--- collection of interconnected components or elements**



AUTOMATIC TEST PATTERN GENERATION

Ÿ **ATPG:** Given a circuit, identify a set of test vectors to detect all faults under consideration.



Input circuit

Form fault list

More faults?

No

Exit

Yes

Select a fault

Test generation

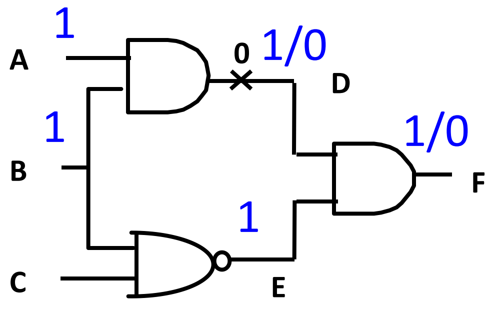
Fault simulation

Fault dropping

TEST GENERATION

- **Given a fault, identify a test to detect this fault**

Example:

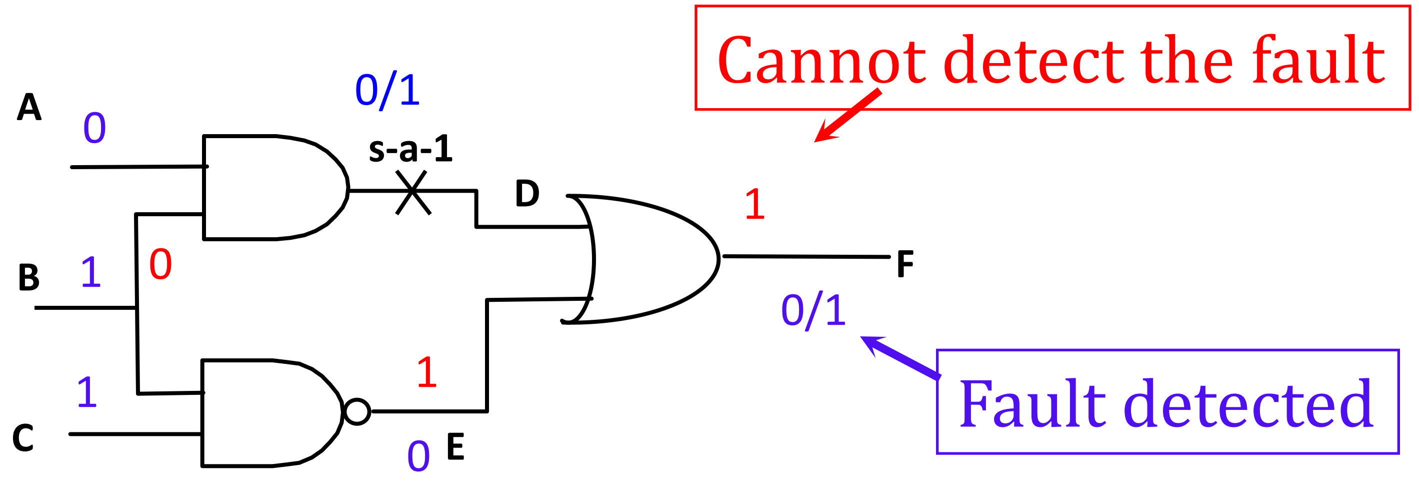


To detect D s-a-0, D must be set to 1.
Thus A=B=1.

To propagate fault effect to the primary output
E must be 1. Thus C must be 0.
Test vector: A=1, B=1, C=0

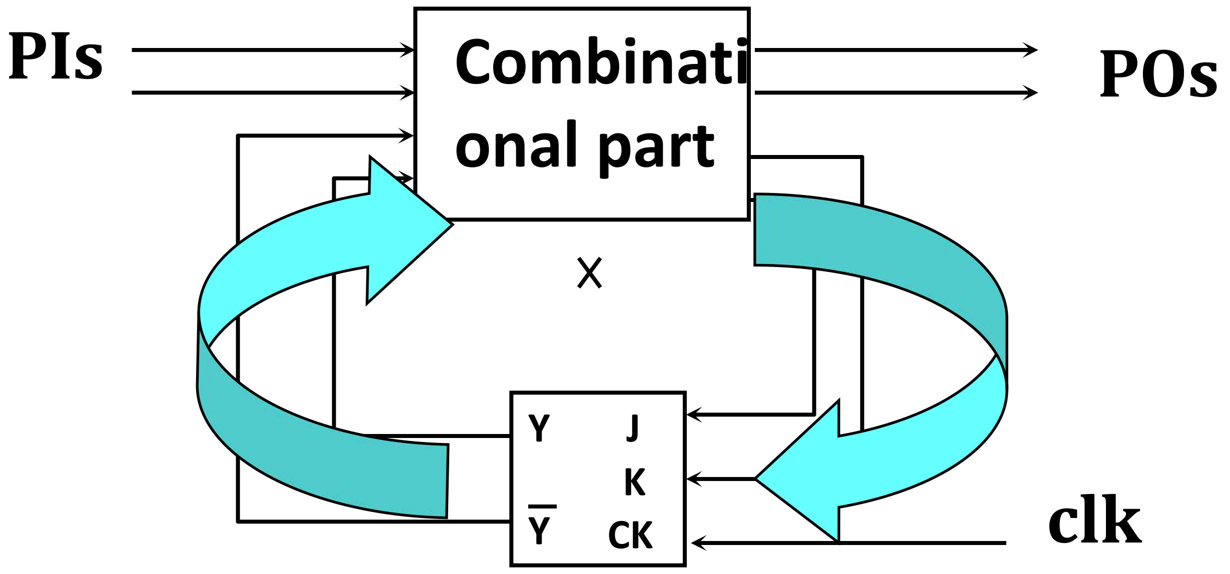
DIFFICULTIES IN TEST GENERATION

1. Reconvergent fan-out



DIFFICULTIES IN TEST GENERATION (CONT.)

2. Sequential test generation



TESTABLE DESIGN

- Design for testability (DFT)
  - ad hoc techniques
  - Scan design
  - Boundary Scan
- **Built-In Self Test (BIST)**
  - **Random number generator (RNG)**
  - **Signature Analyzer (SA)**
- Synthesis for Testability

CLASS ROOM ACTIVITY

HOW CAN YOU DO YOUR INTERVIEW PREPARATION ????

Tell about yourself

Resume/CV –short & Neat

Aptitude,GD,Technical skill,HR interview

Tell about your final year project

Co & Extra curricular activities

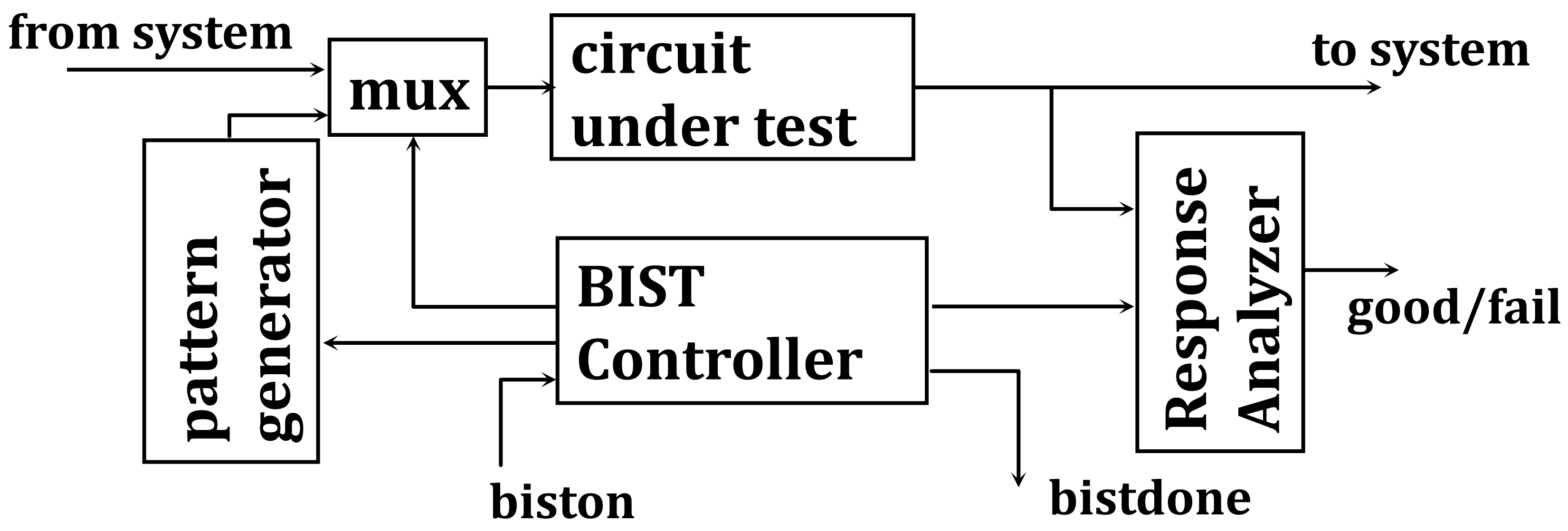
Know about your company applying & Your job profile-Skill matching

Self confidence ,Body language

Knowledge ,Skill,Attitude,Team work, Adaptability etc…

BUILT-IN-SELF TEST (BIST)

- Places the job of device testing inside the device itself
- Generates its own stimulus and analyzes its own response



BASIC CONCEPTS

- We add extra hardware to the chip for test generation and response evaluation
  - Done on chip INSIDE
  - Additional hardware overload
- External control pins
- Input pin-Test control(TC)
- Output pin-Good/Bad

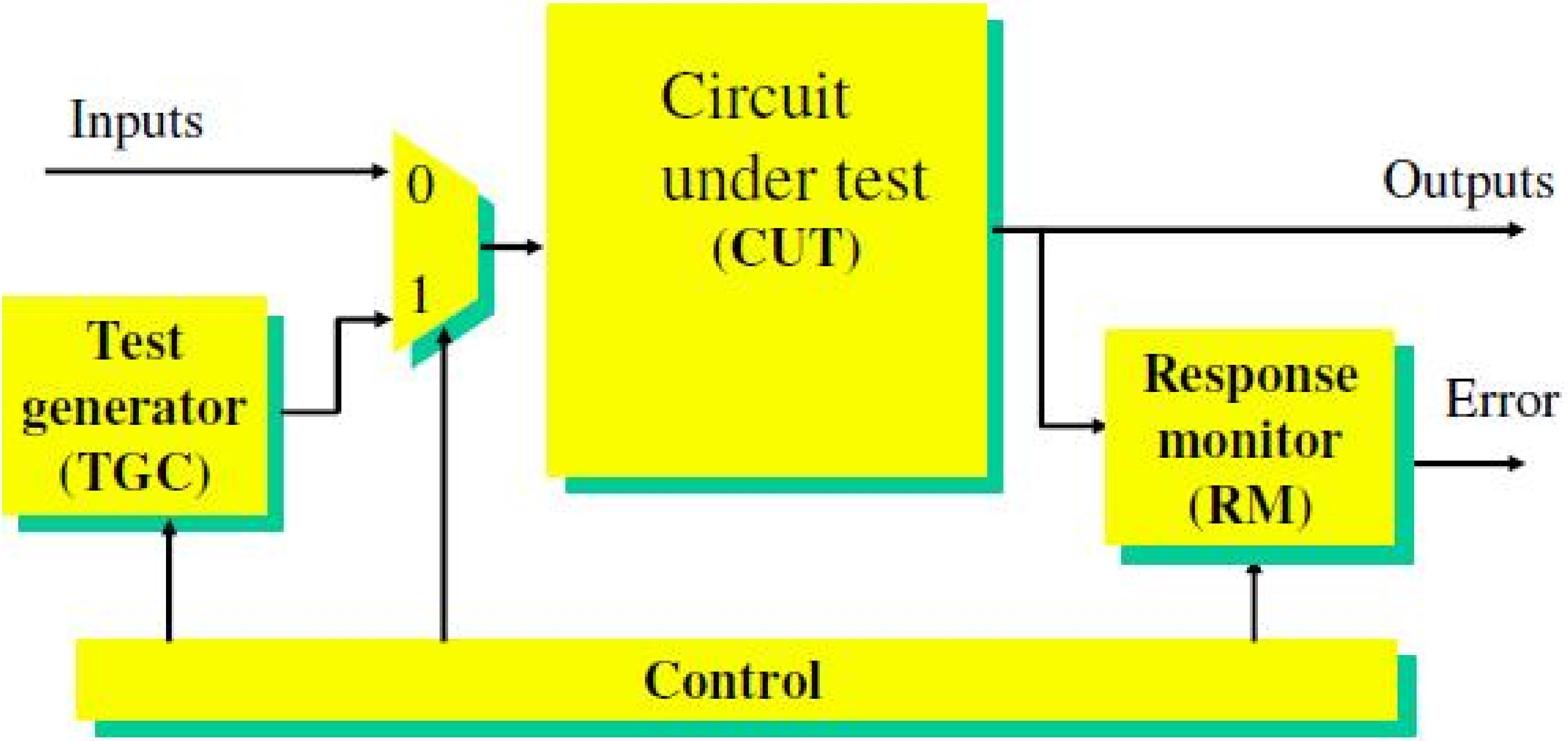
BUILT-IN-SELF TEST (BIST)

Built-in self-test lets blocks test themselves

– Generate pseudo-random inputs to comb. logic

– Combine outputs into a *syndrome*

– With high probability, block is fault-free if it produces



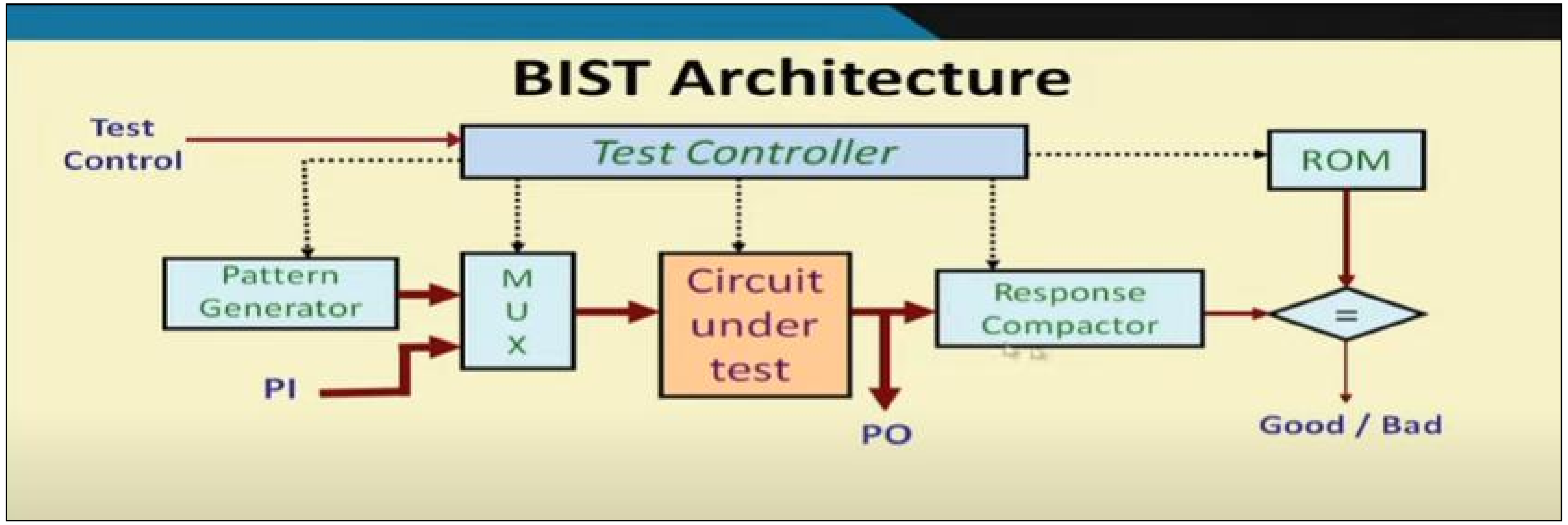
BUILT-IN-SELF TEST (BIST) ARCHITECTURE

- Two major tasks

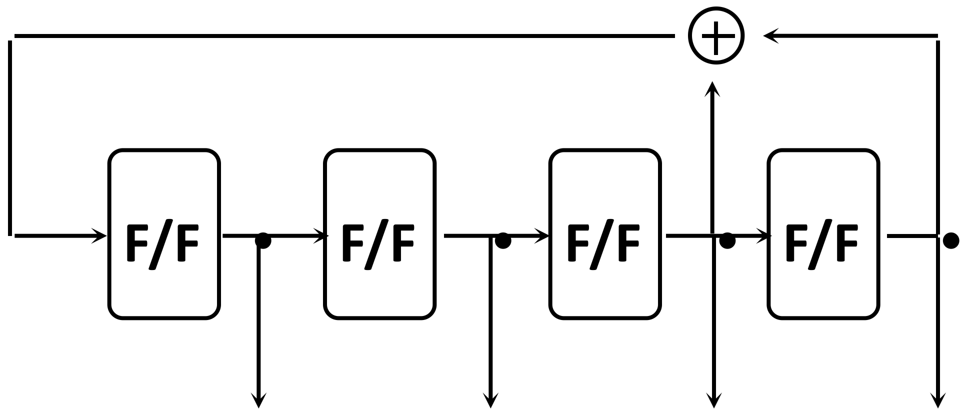
  - Test pattern generation

  - Test result compaction

- Usually implemented by linear feedback shift register
- NETLIST -----Test Generation -------Test Vectors
- Error input----CUT-----Error output
- Automated Test Equipment ATE (Loaded Test Pattern) ---CUT----Output given to ATE



RANDOM NUMBER GENERATOR (RNG)



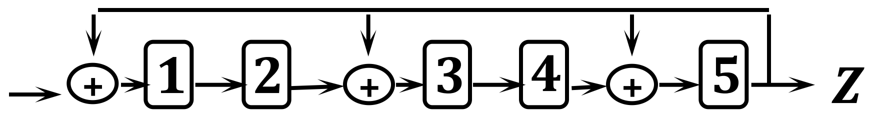
0001
1000
0100
0010
1001
1100
0110
1011
0101
1010
1101
1110
1111
0111
0011
0001
(repeat)

1. Generate “pseudo” random patterns

2. Period is $2^n - 1$
3.Pseudo Random pattern is an input test vectors
4.Facult coverage done by Fault simulation
    –Test length is large
    --much faster test generation
    --Continue until fault coverage 60-80%then switch to ATPG

SIGNATURE ANALYZER (SA)

Input sequence 10101111 (8 bits)



$$G(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^7$$

$$P(x) = 1 + x^2 + x^4 + x^5$$

Time	Input stream	Register contents	Output stream
0	1 0 1 0 1 1 1 1	0 0 0 0 0	← Initial state
1	1 0 1 0 1 1 1	1 0 0 0 0	
.	.	.	
.	.	.	
5	1 0 1	0 1 1 1 1	
6	1 0	0 0 0 1 0	1
7	1	0 0 0 0 1	0 1
8		0 0 1 0 1	1 0 1

Remainder

$$R(x) = x^2 + x^4$$

Quotient

$$1 + x^2$$

SIGNATURE ANALYZER (SA) (CONT.)

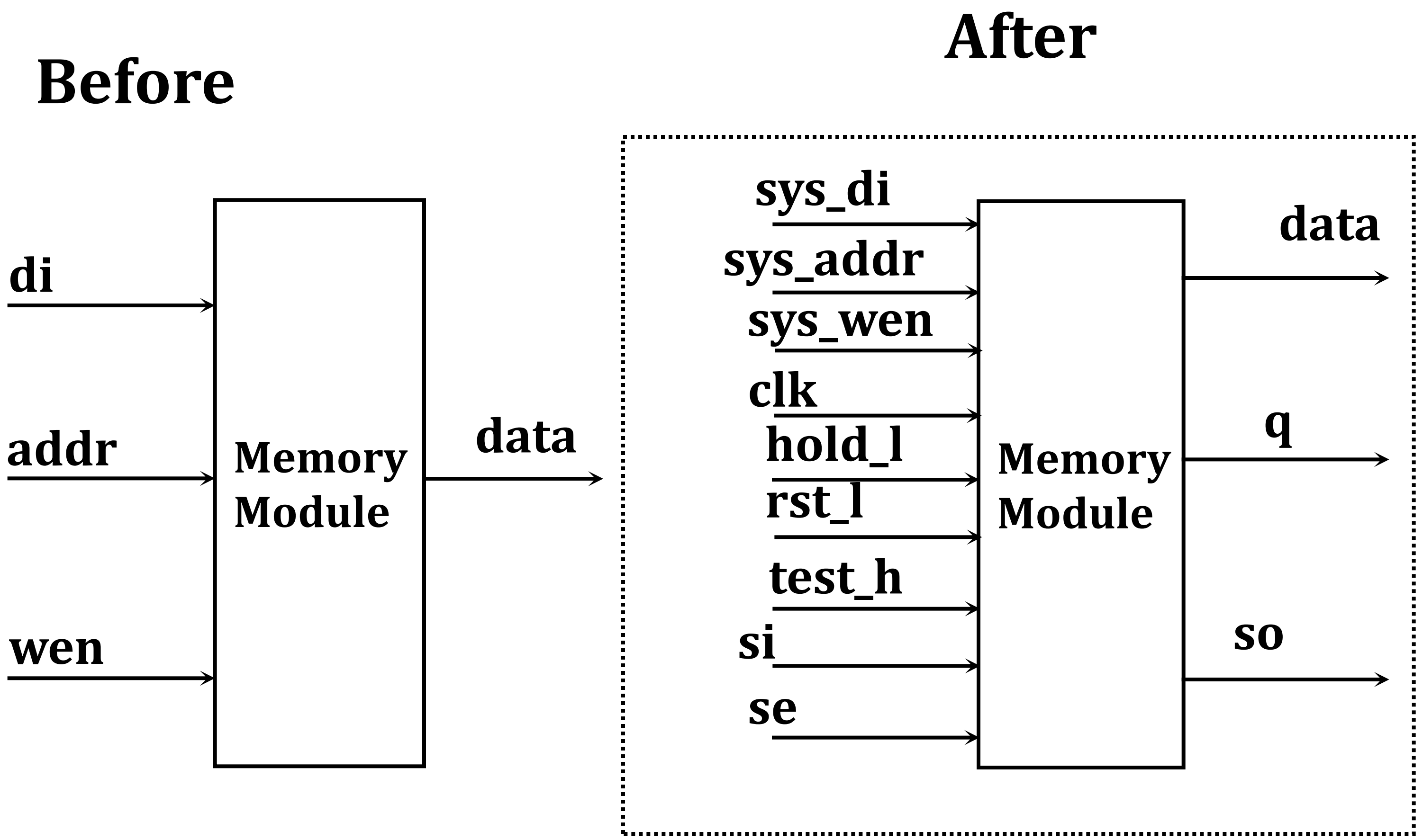
- A LFSR performs polynomial division

$$\begin{array}{rl} & P(x): x^5+x^4+x^2+1 \\ \times & Q(x): x^2+1 \\ \hline & x^7+x^6+x^4+x^2+x^5+x^4+x^2+1 \\ & =x^7+x^6+x^5+1 \end{array}$$

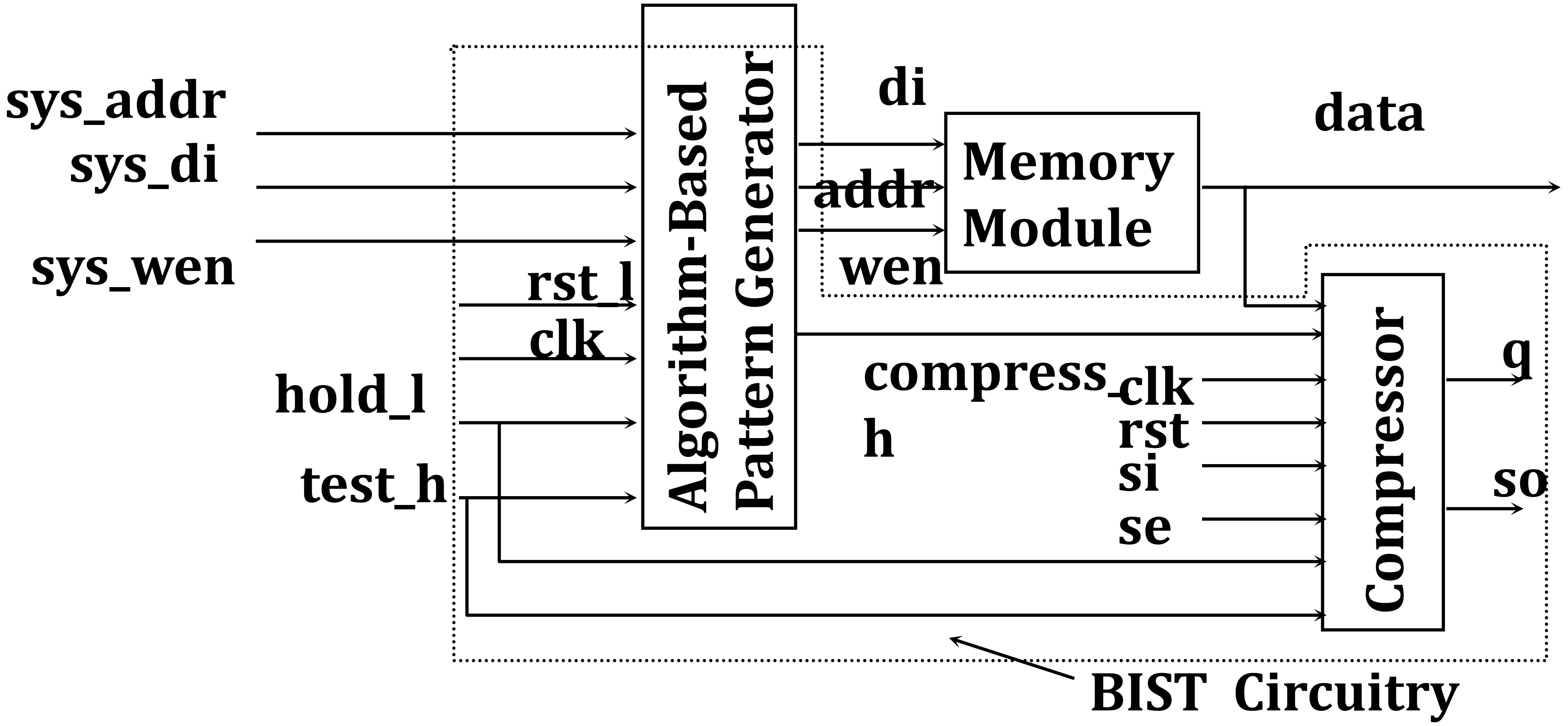
$$P(x)Q(x)+R(x)=x^7+x^6+x^5+x^4+x^2+1=G(x)$$

- Probability of aliasing error = $1/2^n$ (n: # of FFs)

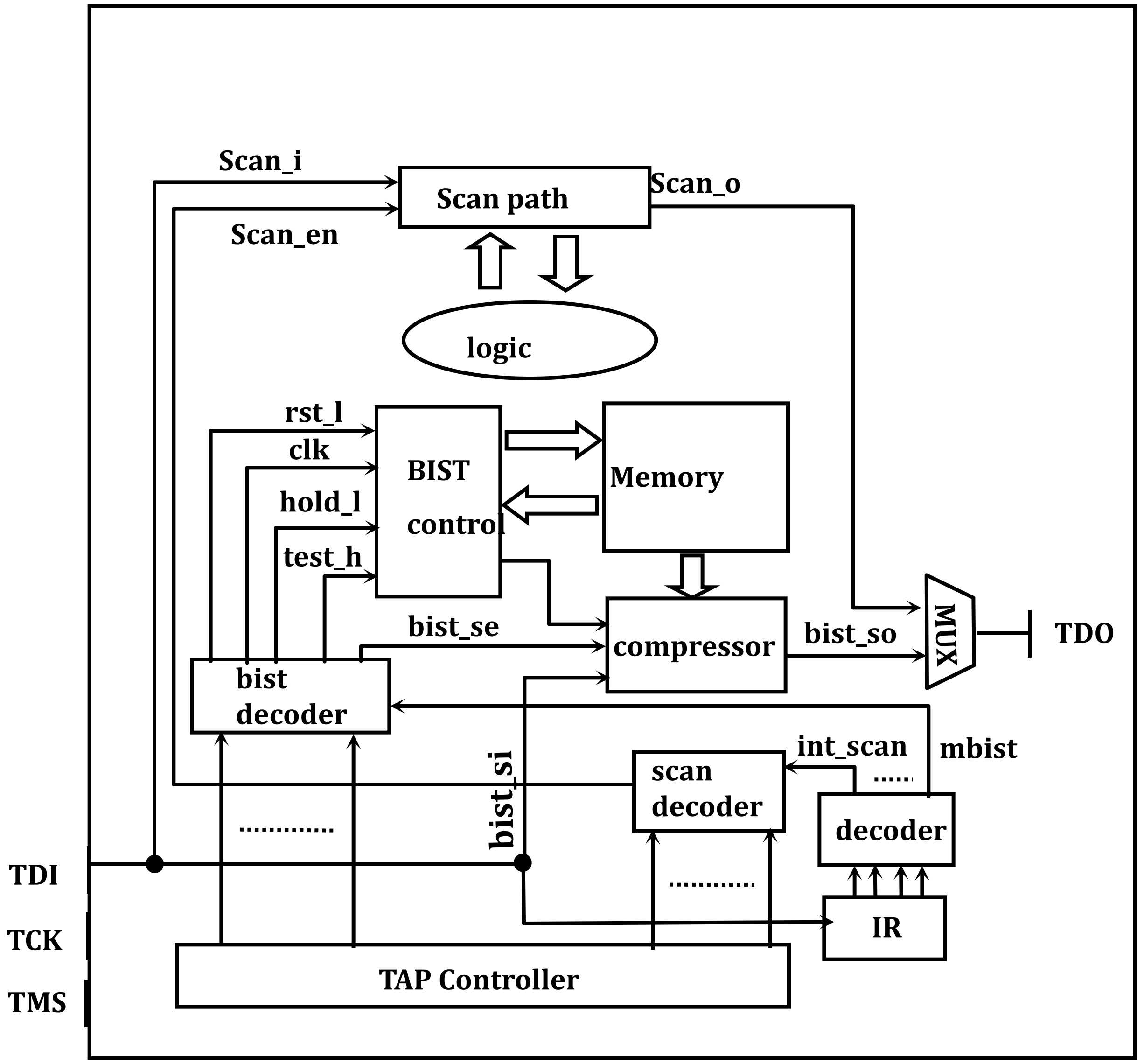
MEMORY BIST ARCHITECTURE



MEMORY BIST ARCHITECTURE (CONT.)



CPU TEST CONTROL ARCHITECTURE



NEEDS OF BIST

- Field Test & Diagnosis(Software Test)-not needed expensive ATE
    - Low hardware fault coverage
    - Poor diagnostic resolution
    - Time consuming
- In Hardware –Lower system test effort & better diagnosis
    - Improve system maintenance & repair

TESTING METHODS

- **A 32-bit adder --- ATPG**
- **A 32-bit counter --- Design for testability + ATPG**
- **A 32MB Cache memory --- BIST**
- **A 10^7-transistor CPU --- All test techniques**
- **An SOC**

ASSESSMENT

1. **How can you make test generation?**
2. **How can you generate random number?**
3. **Why we use Signature Analyser in BIST?**
4. **List out the basic concepts of BIST**
5. **Draw the architecture of BIST.**
6. **Match all correctly**

A 32-bit adder --- BIST

A 32-bit counter --- All test techniques

A 32MB Cache memory --- ATPG

A 10^7-transistor CPU --- Design for testability + ATPG

SUMMARY & THANK YOU