Design For Testability

2011.06.03

Sungho Kang

COMPUTER SYSTEMS &
RELIABLE SOC LAB.

Agenda

- Introduction
- Test Methodology
- Design for Testability
- IEEE P1500
- Case Study
- Conclusion

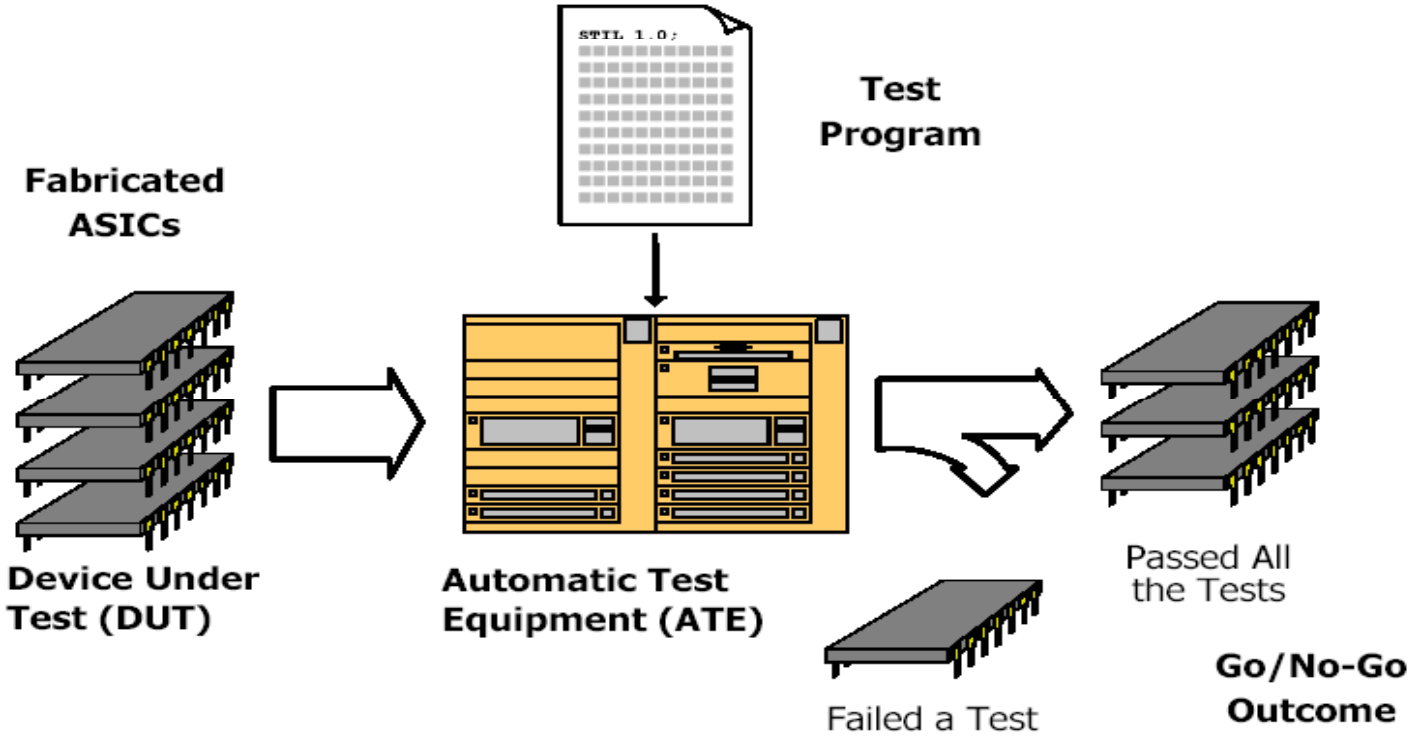
VLSI Implementation

Design Constraints

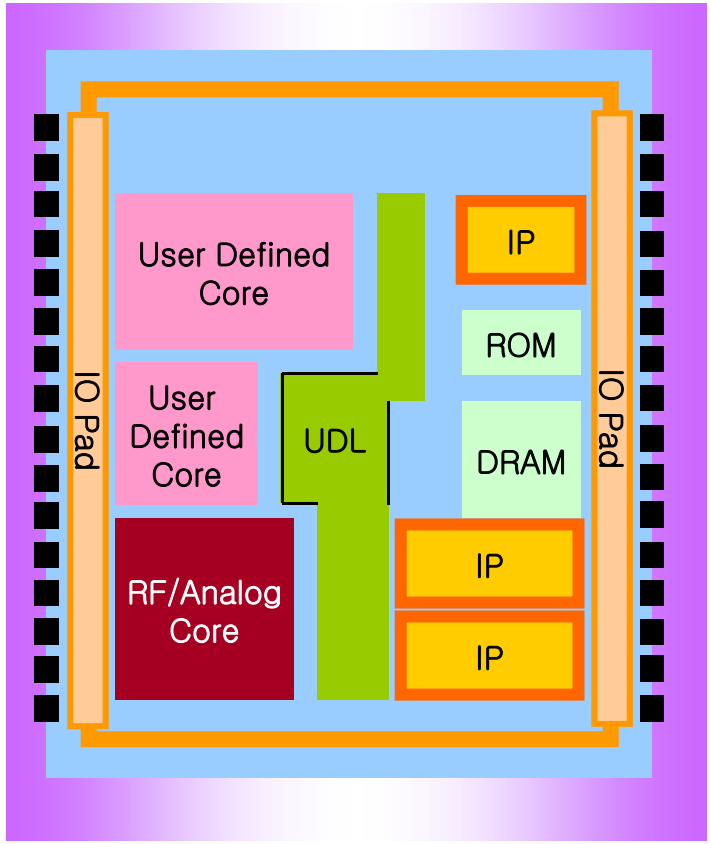
- **Area**
- **Speed**
- **Power**
- **Testability**

Testing Process

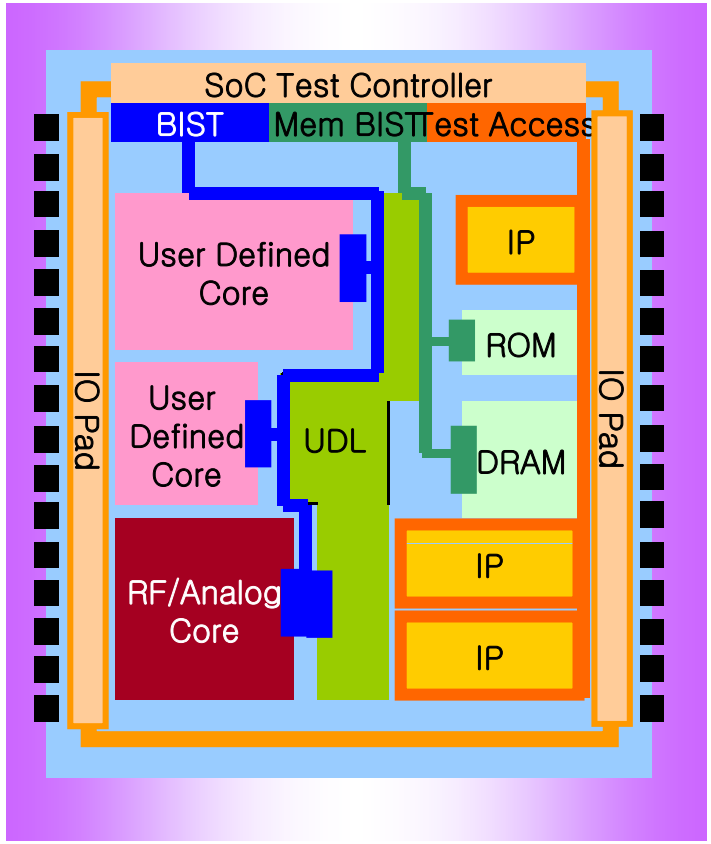
- **Rule of thumb: spend 5-10% die area on DFT (10% of die is state and scan is 50% overhead per state!)**



SOC Design

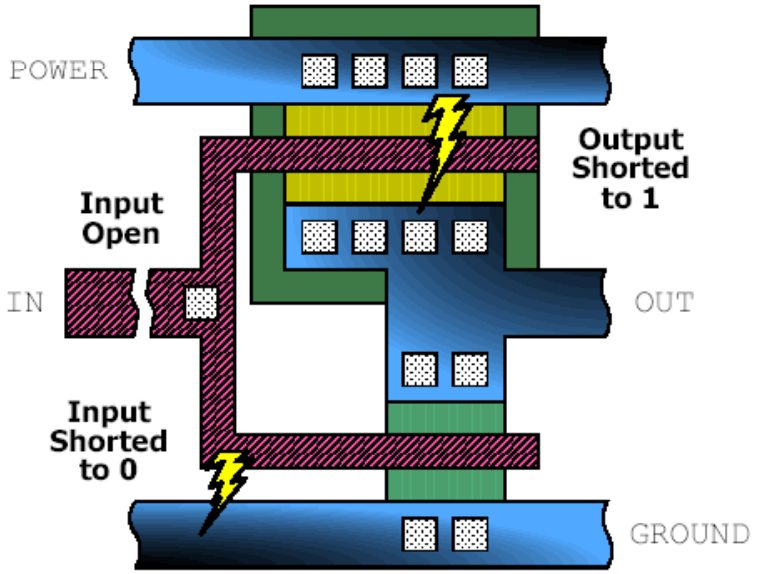

User Defined Core
IP
ROM
User Defined Core
UDL
DRAM
IO Pad
IO Pad
RF/Analog Core
IP
IP

DFT for SOC


SoC Test Controller
BIST
Mem BIST
Test Access
User Defined Core
User Defined Core
RF/Analog Core
UDL
IP
ROM
DRAM
IP
IP
IO Pad
IO Pad

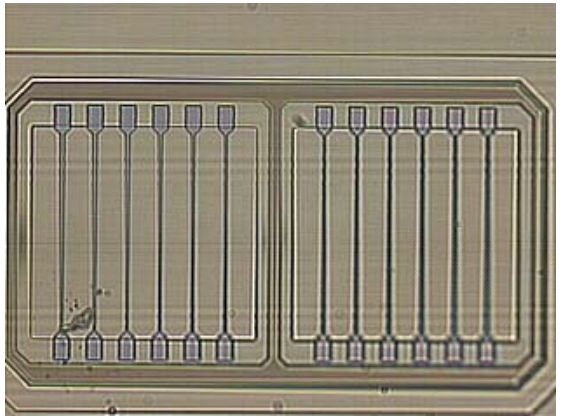
Manufacturing Defects

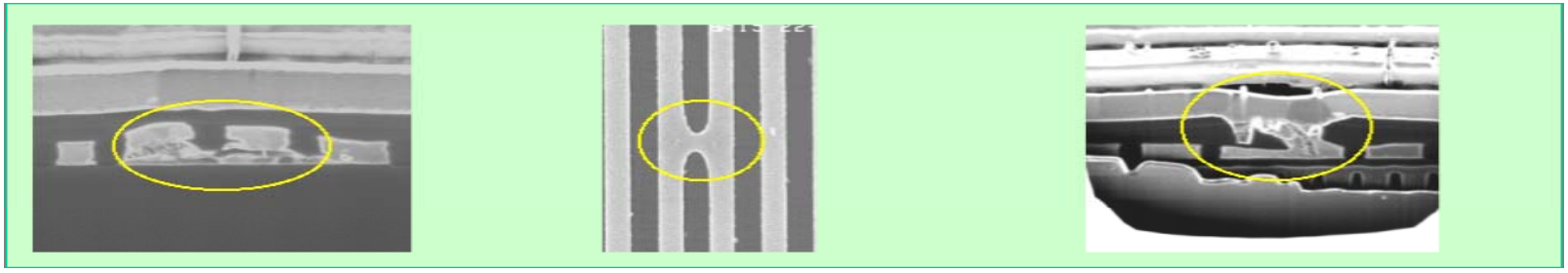
- In fabrication, defects get introduced from many sources:
  - **Contamination**
  - **Metalization Defect**
  - **Implant Defect**
  - **Wafer Defect**
  - **Oxide Defect**
  - **Interconnect Defect**



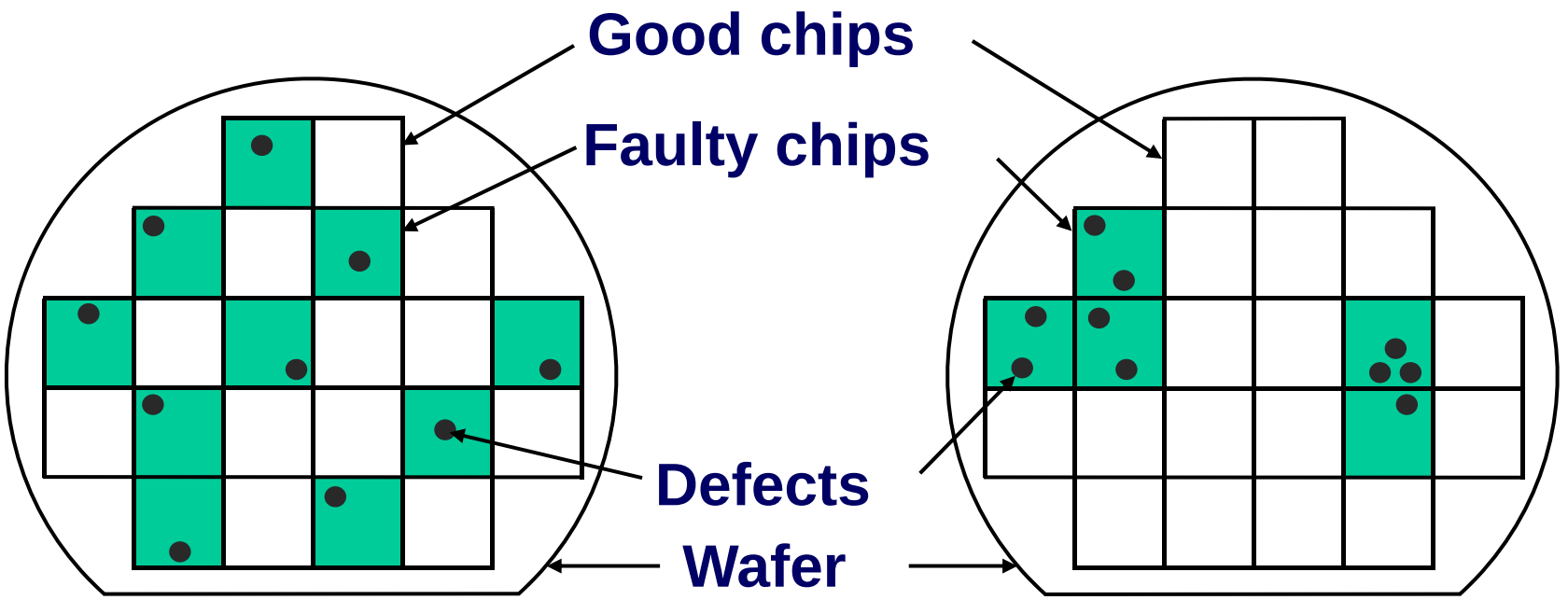
Defect & Fault

- **Permanent**
- **Intermittent**
- **Temporary**





Clustered VLSI Defects



Unclustered defects
Wafer yield = 12/22 = 0.55

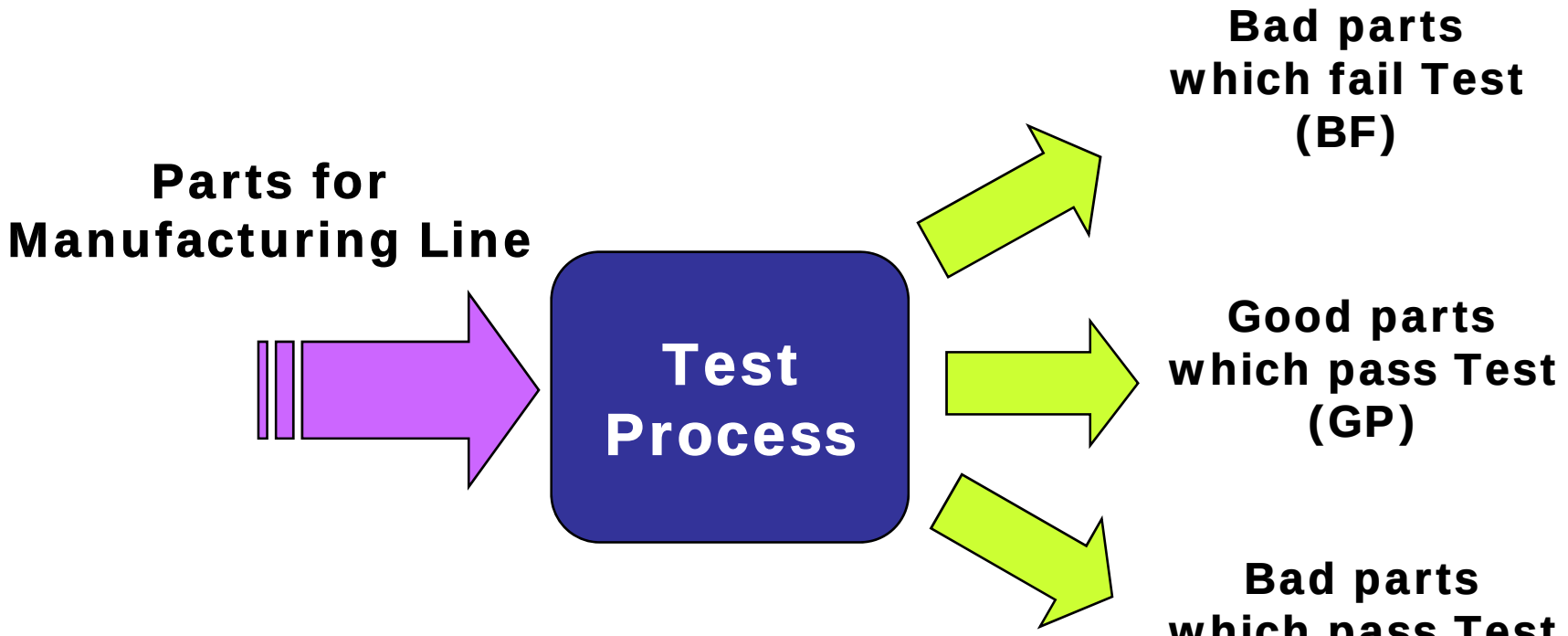
Clustered defects (VLSI)
Wafer yield = 17/22 = 0.77

Verification vs Test

- **Verification**
  - Correctness of design
  - Simulation, Emulation and Formal verification
  - Performed once before manufacturing
  - Responsible for quality of design

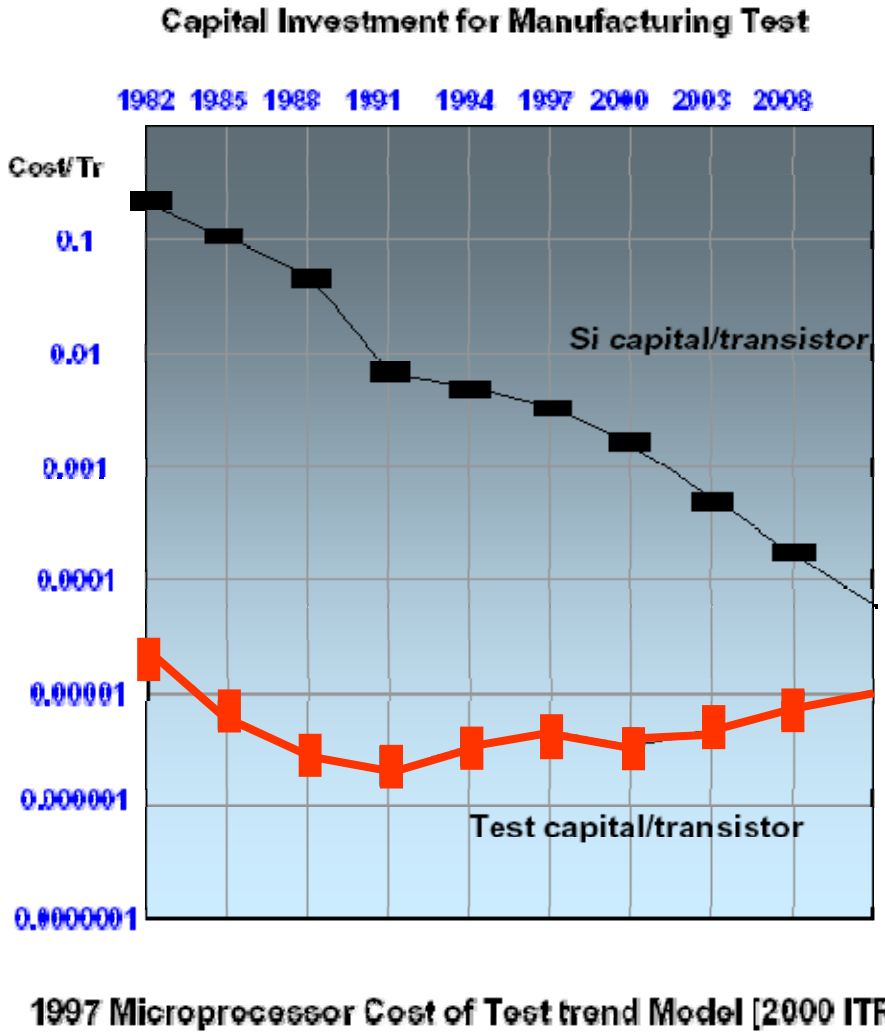
- **Test**
  - Correctness of manufactured hardware
  - Test generation and test application
  - Performed every manufactured devices
  - Responsible for quality of devices

Test Process



$$\text{Defect Level} = \frac{BP}{GP + BP}$$

Test Cost vs Manufacturing Cost



1997 Microprocessor Cost of Test trend Model [2000 ITRS]

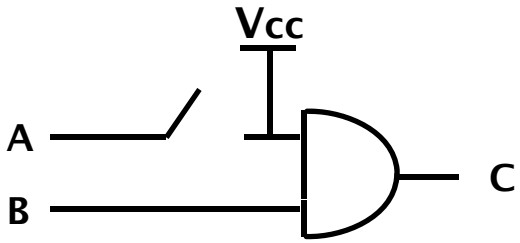
Single Stuck-at Fault (SSF)

- Only one line in the circuit is faulty at a time
- The fault is permanent (as opposed to transient)
- The effect of the fault is as if the faulty node is tied to either Vcc (s-a-1), or Gnd (s-a-0)
- The function of the gates in the circuit is unaffected by the fault

Fault: A s-a-1

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Fault-Free Gate

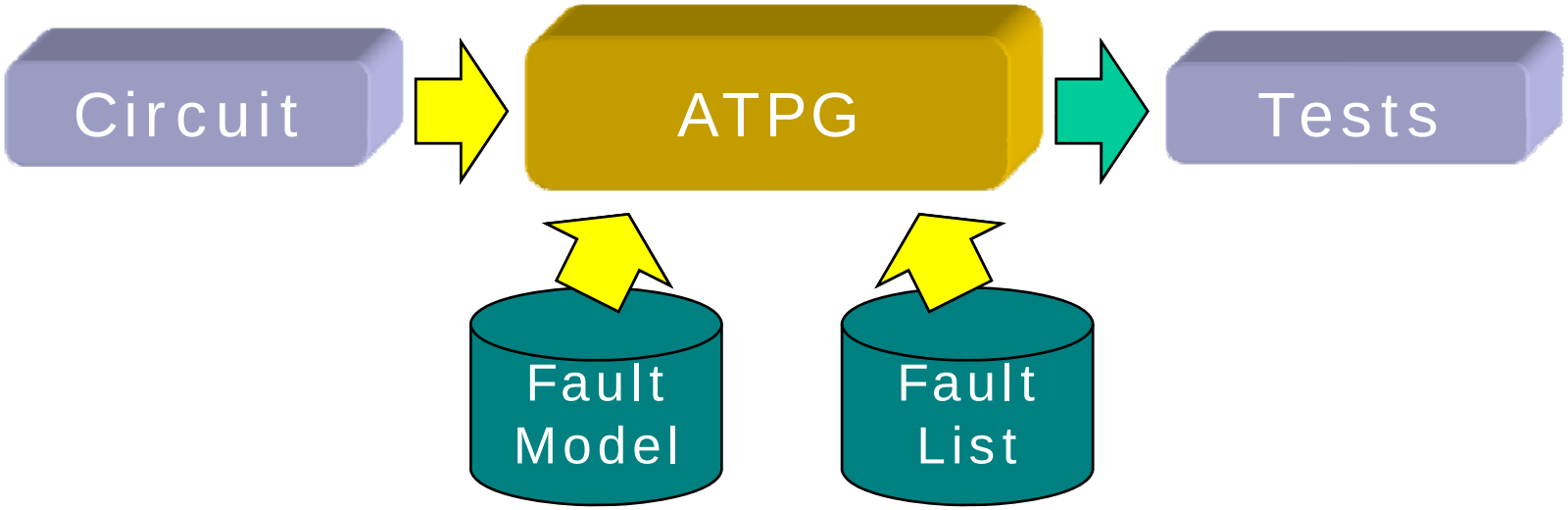


A	B	C
0	0	0
0	1	1
1	0	0
1	1	1

Faulty Gate

ATPG

- **Automatic Test Pattern Generation (ATPG)**
  - **Calculate the set of test patterns from a description of the logic network and a set of assumptions called fault models**

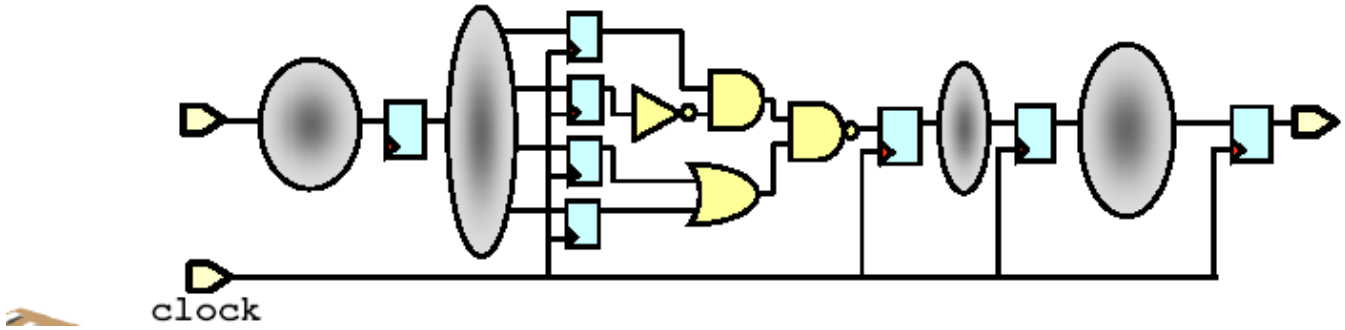


Sequential ATPG

To detect a stuck-at fault in synchronous **sequential** logic, we can still use the familiar D algorithm, but it'll take….

- One or more clock cycles to **activate** the fault.
- One or more clock cycles to **propagate** the fault effect.

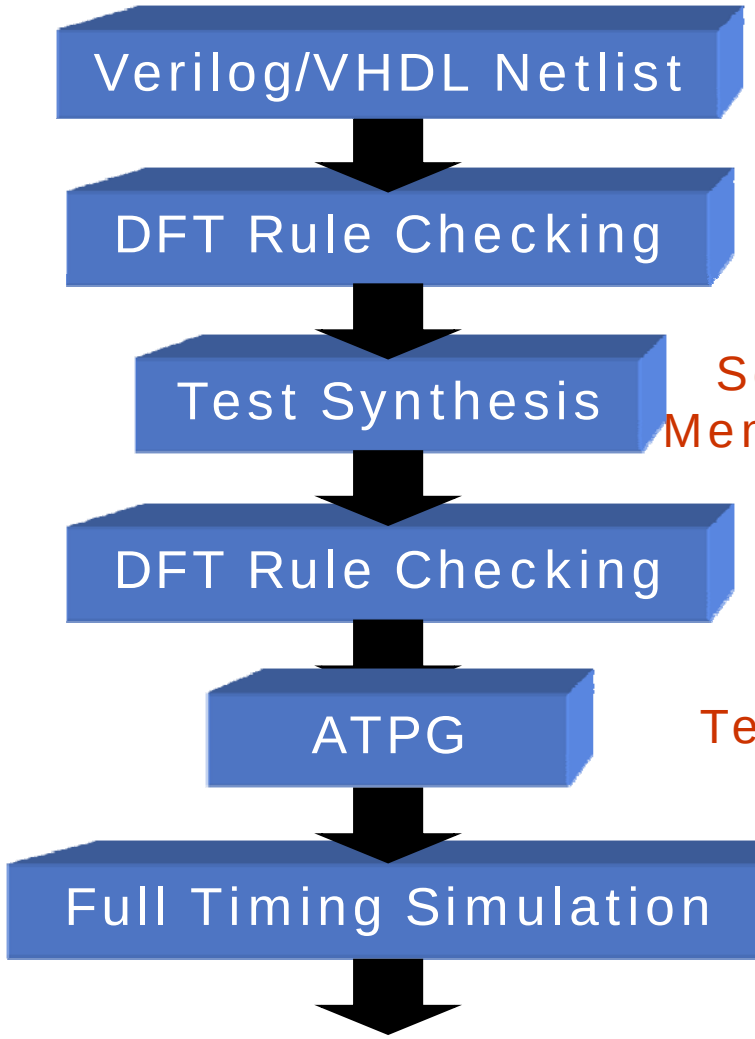
In general, we'll need a **sequence** of patterns to detect a fault!



Design-for-Testability (DFT)

- **Definition**
  - **Any design effort to reduce test costs**
  - **The process of including special features to make a device easily testable**
- **Objective**
  - **To reduce in overall design cycle times and test costs without sacrificing the quality of the product**

DFT Flow



DFT Rule Checking

Scan Design, Boundary Scan Design
Memory BIST Design, Logic BIST Design

DFT Rule Checking
Static Timing Analysis

Test Pattern Generation

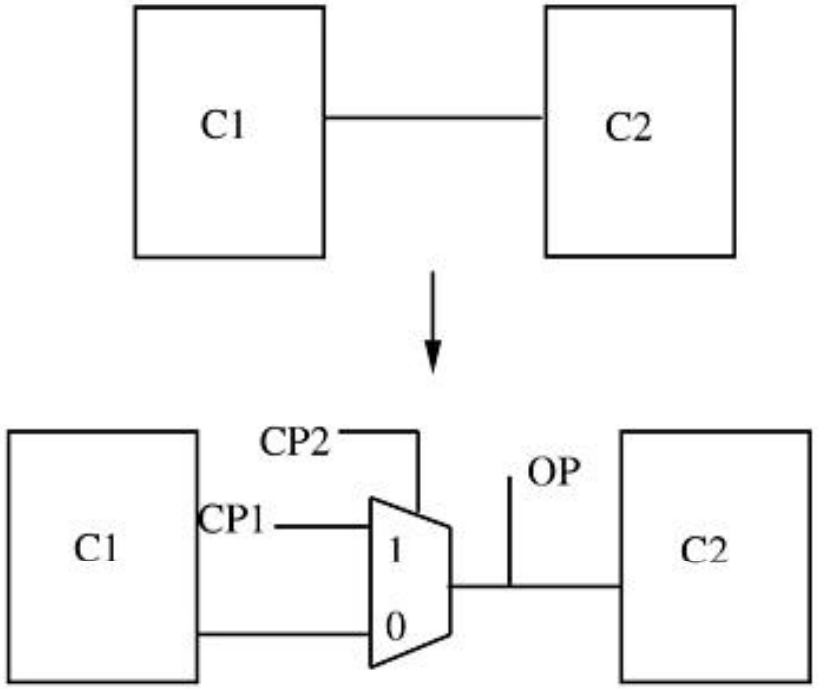
Logic Simulation
Test Vector Translation

Ad-Hoc DFT Methods

- **Good design practices learnt through experience are used as guidelines:**
  - ✓ Avoid asynchronous (unclocked) feedback.
  - ✓ Make flip-flops initializable.
  - ✓ Avoid redundant gates. Avoid large fanin gates.
  - ✓ Provide test control for difficult-to-control signals.
  - ✓ Avoid gated clocks.
  - ✓ Consider ATE requirements (tristates, etc.)
- **Design reviews conducted by experts or design auditing tools.**
- **Disadvantages of ad-hoc DFT methods:**
  - ✓ Experts and tools not always available.
  - ✓ Test generation is often manual with no guarantee of high fault coverage.
  - ✓ Design iterations may be necessary.

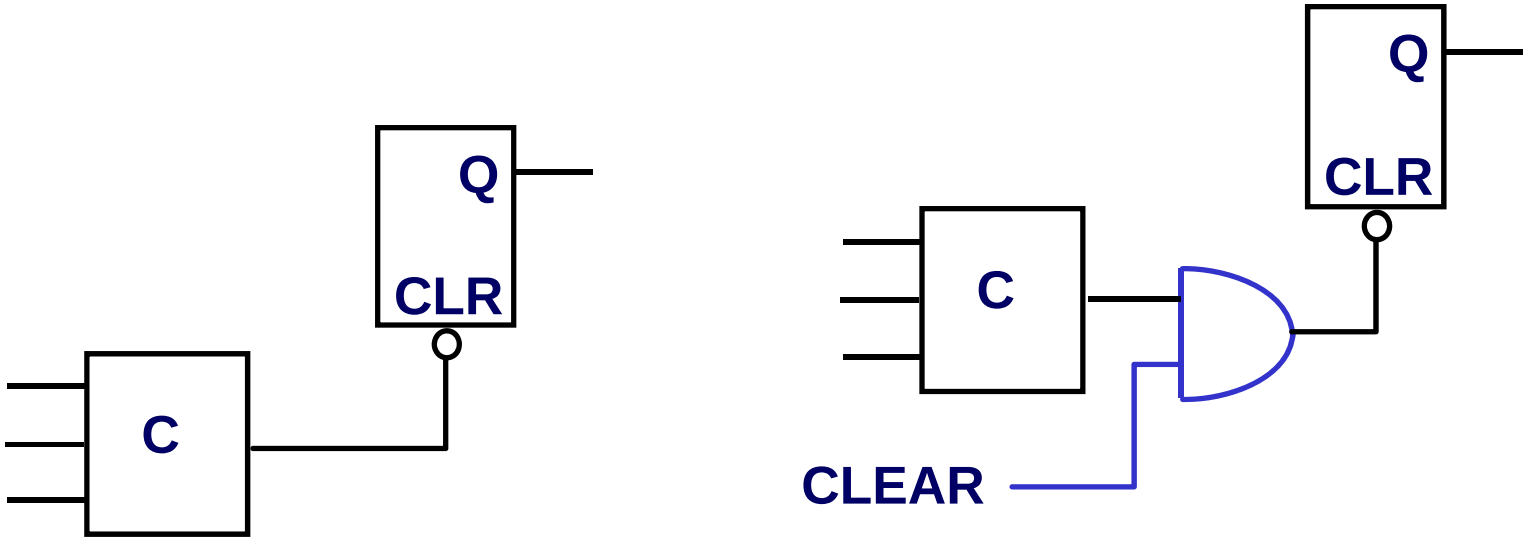
Test Points

- **Employ test points to enhance controllability and observability**
- **Large demand on extra I/O pins**
- **Example**



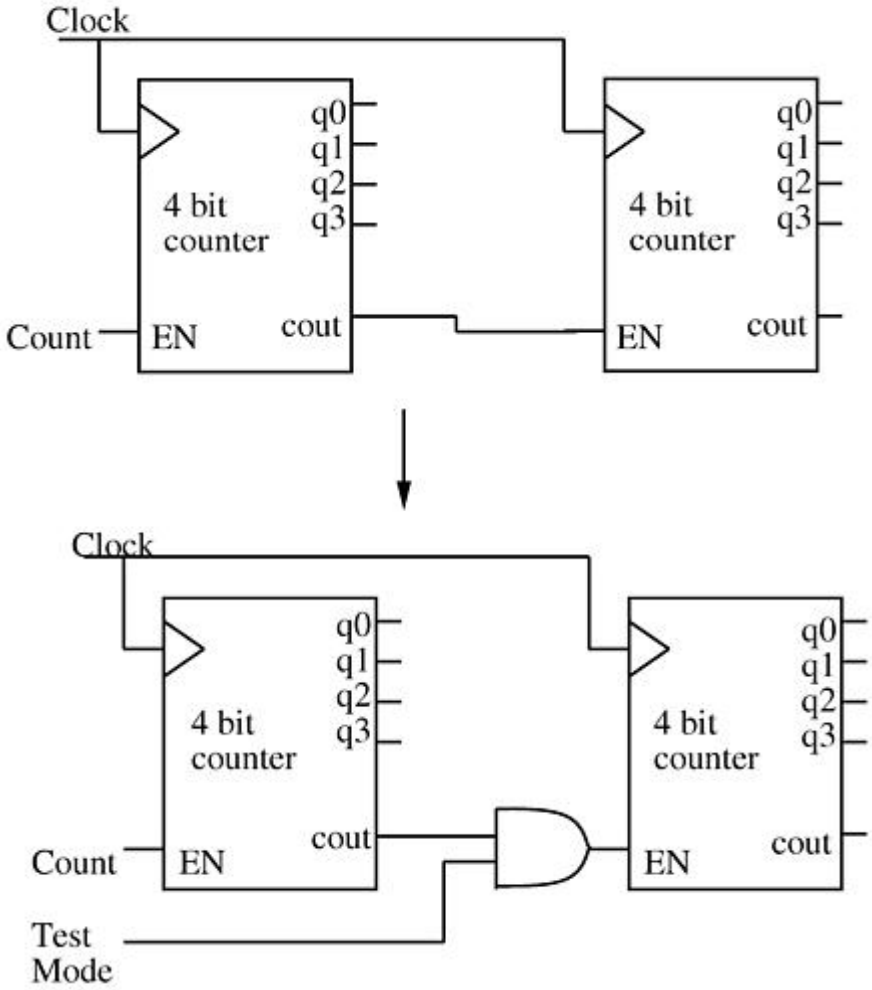
Initialization

- **Flip-flop with explicit clear**
  - **Use explicit clear to all FFs**



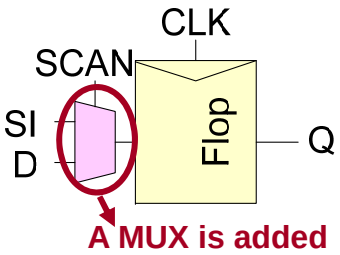
Partitioning

- **Split large counters**

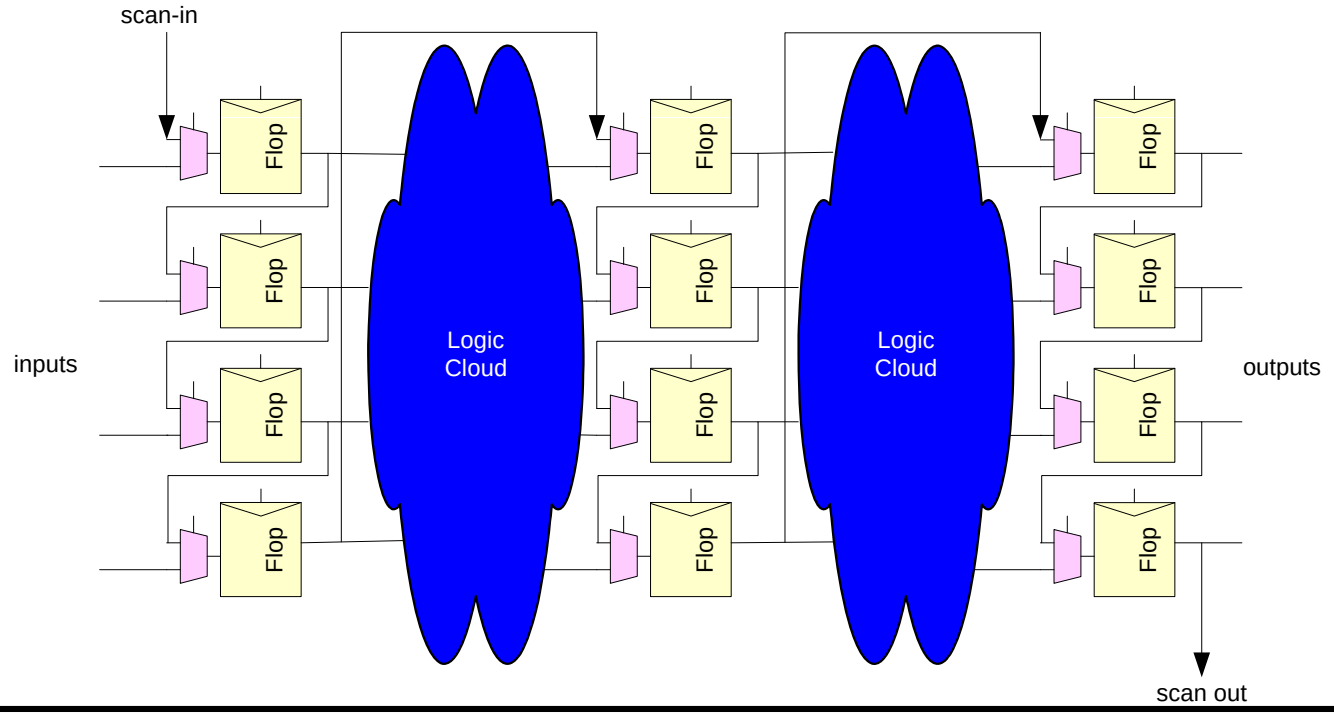


Scan Test

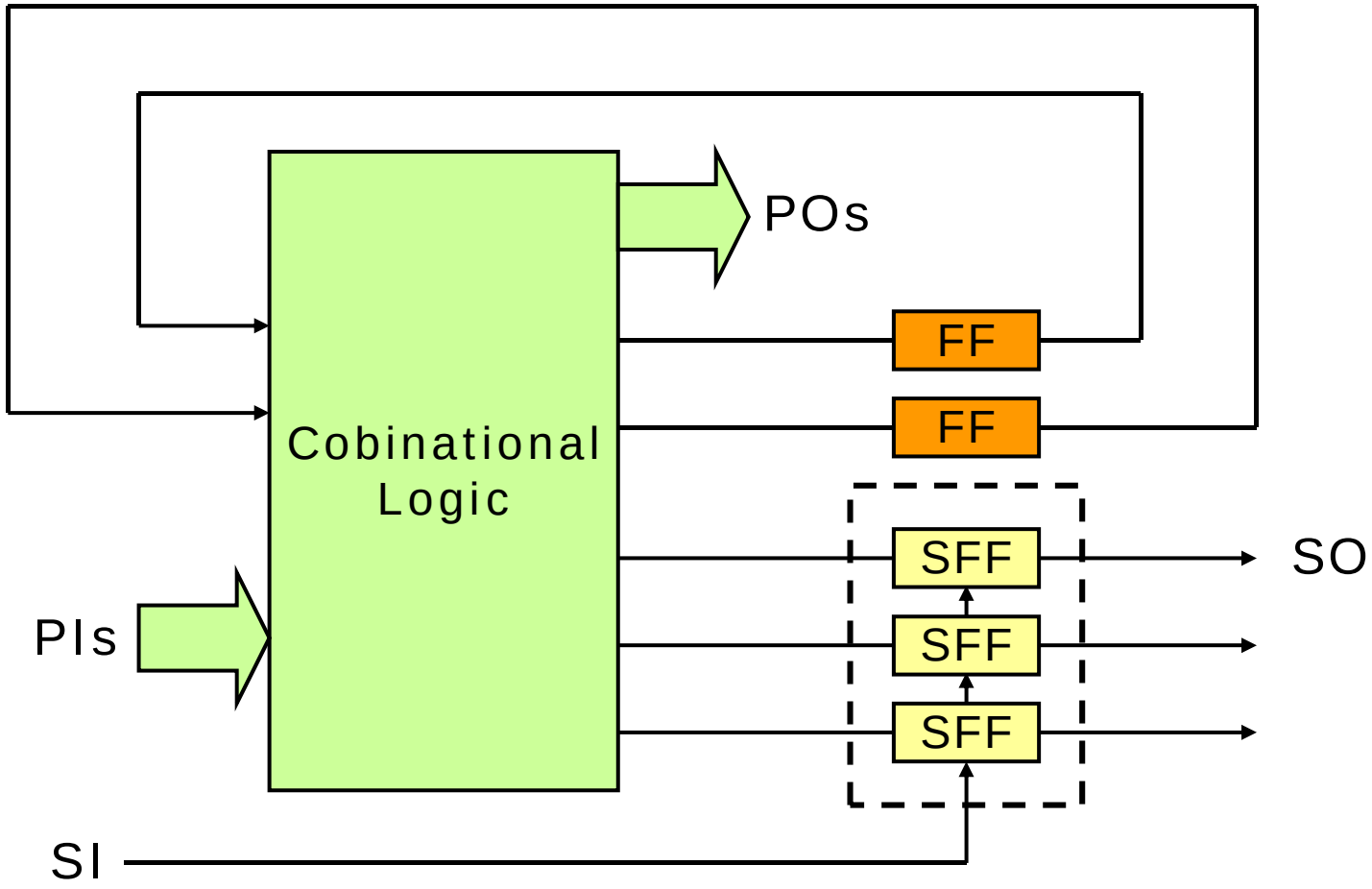
- **Convert each flip-flop to a scan register**
  - **Only costs one extra multiplexer**
  - **Normal mode: flip-flops behave as usual**
  - **Scan mode: flip-flops behave as shift register**



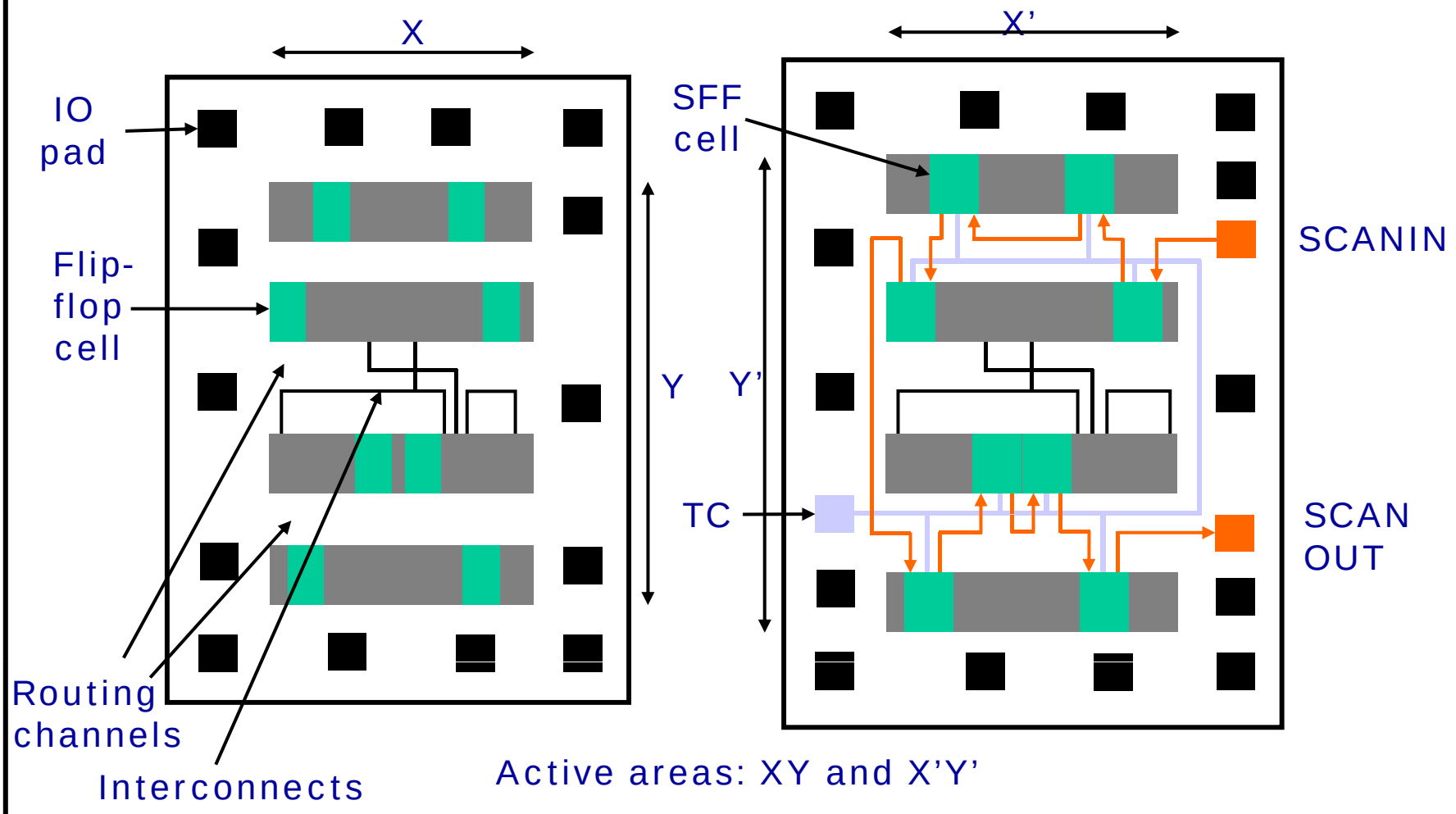
- **Contents of FFs can be scanned out and new values scanned in**



Partial Scan



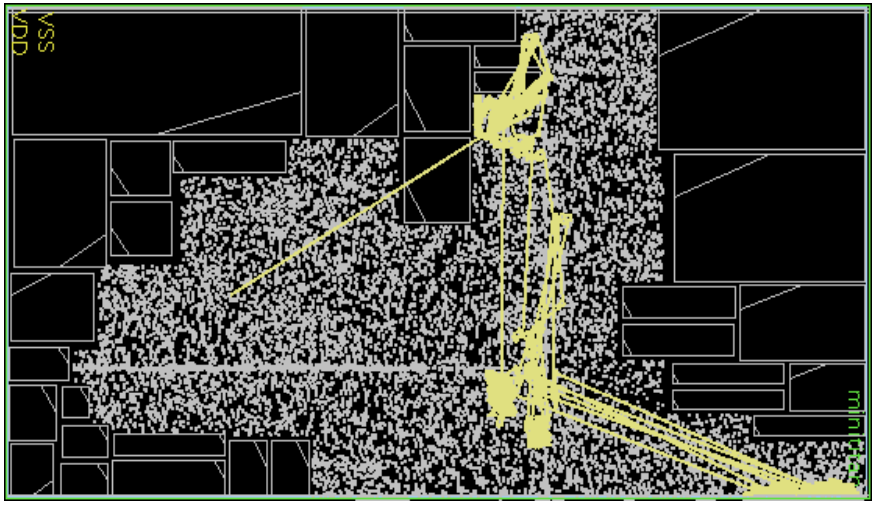
Optimum Scan Layout



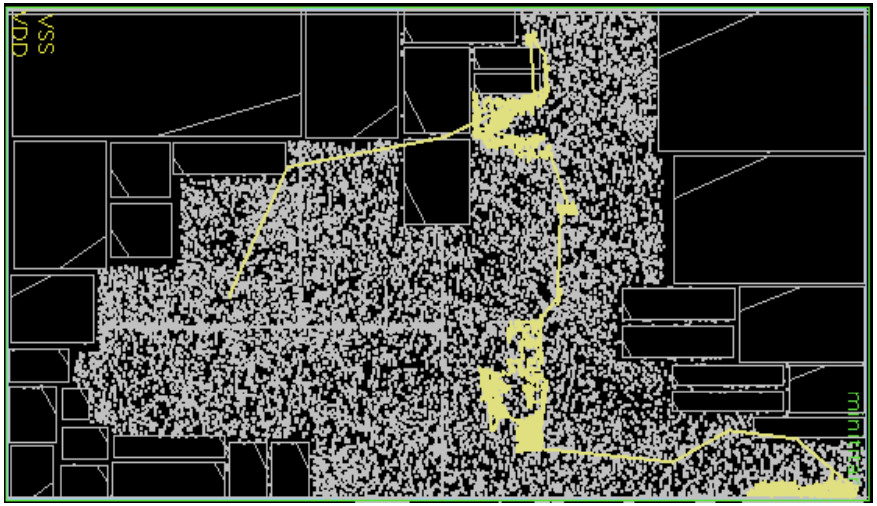
Scan Chain Reordering

- To reduce the routing congestions
- To reduce the hold-buffer insertion during placement
  - May need skew-based optimization

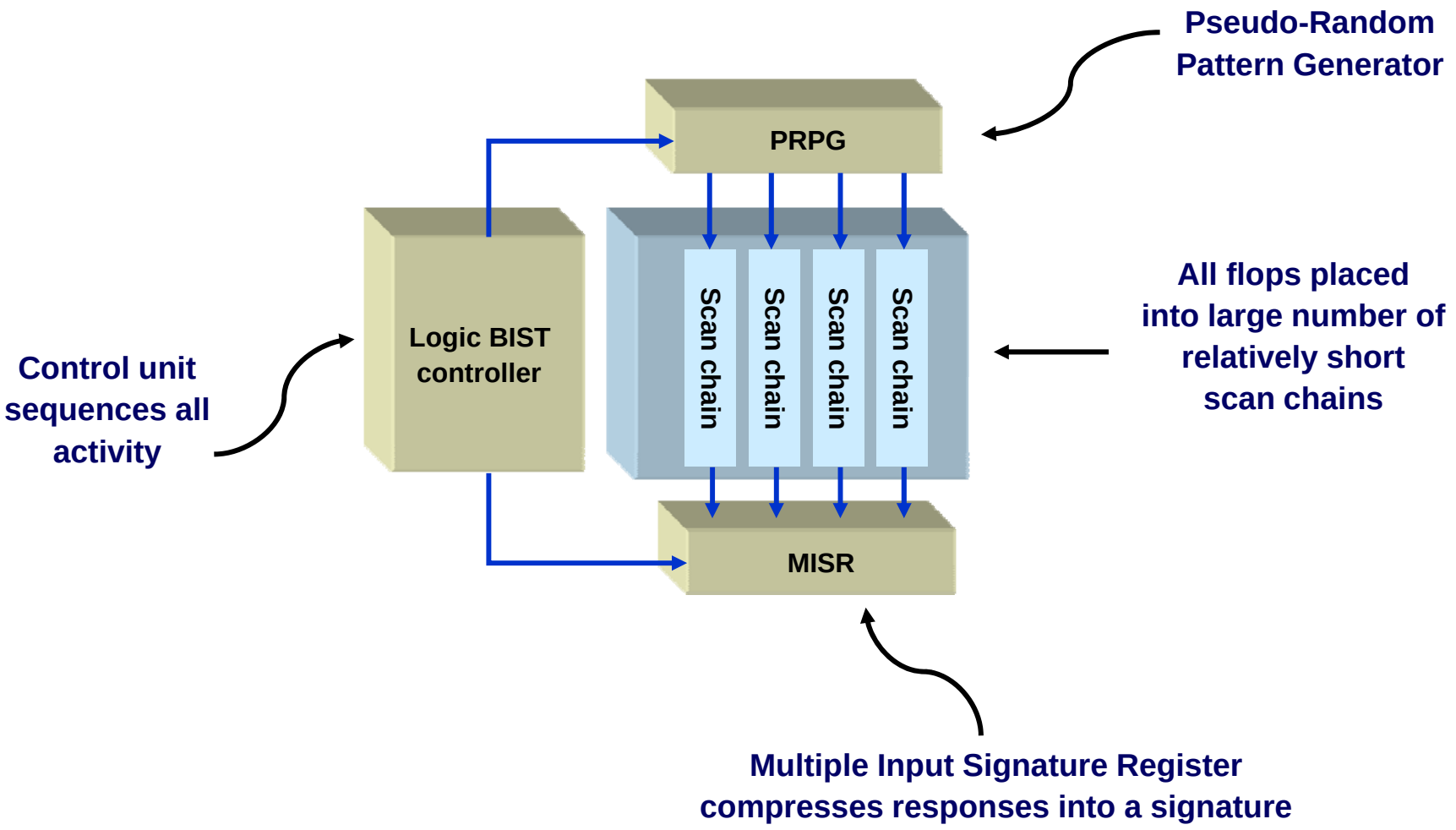
Before



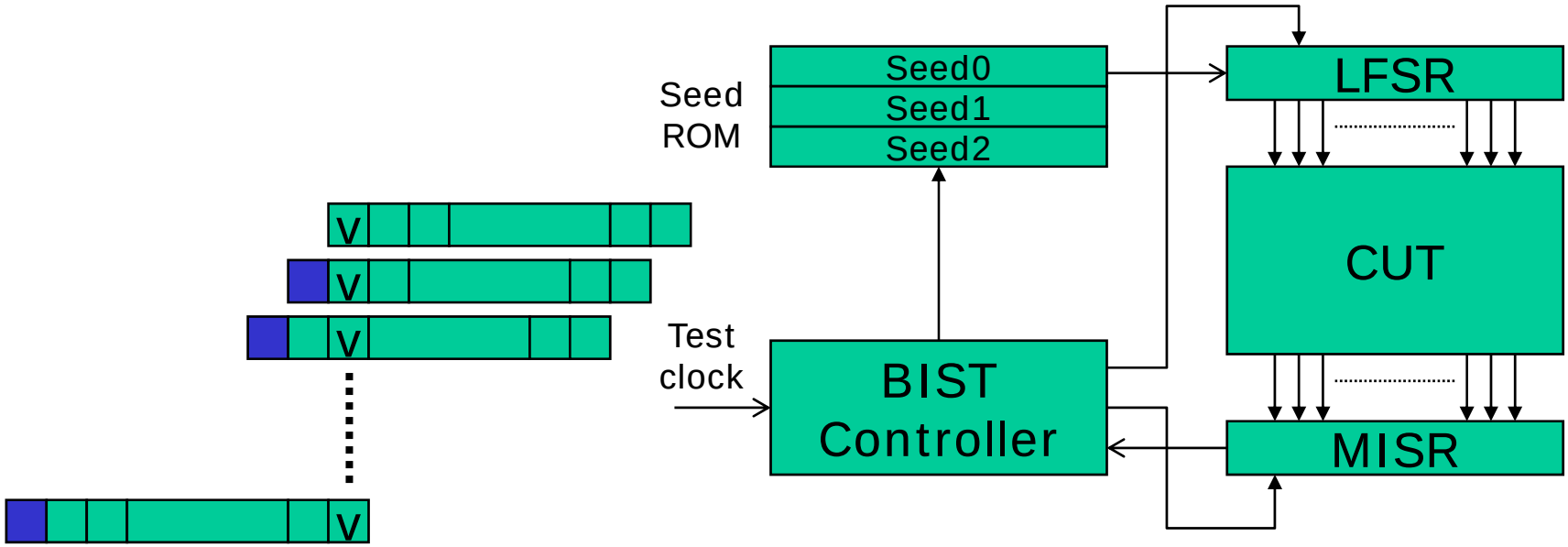
After



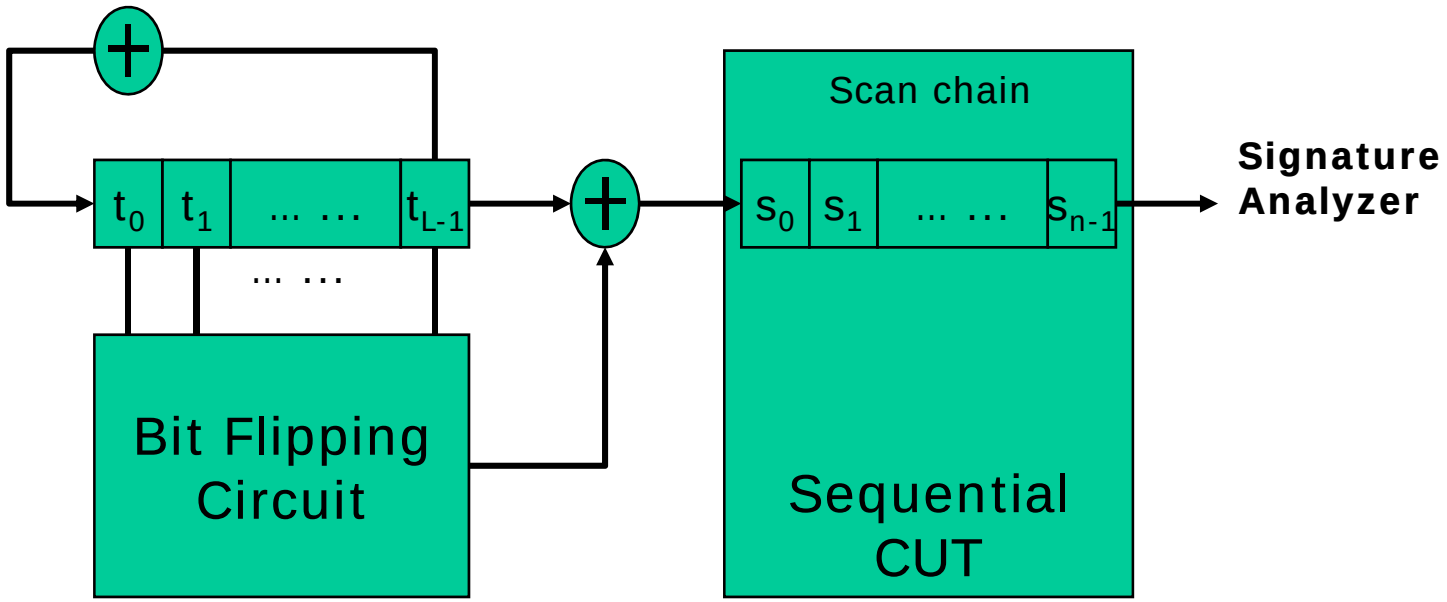
Logic BIST


Pseudo-Random
Pattern Generator
PRPG
Logic BIST
controller
Scan chain
Scan chain
Scan chain
Scan chain
All flops placed
into large number of
relatively short
scan chains
Control unit
sequences all
activity
MISR
Multiple Input Signature Register
compresses responses into a signature

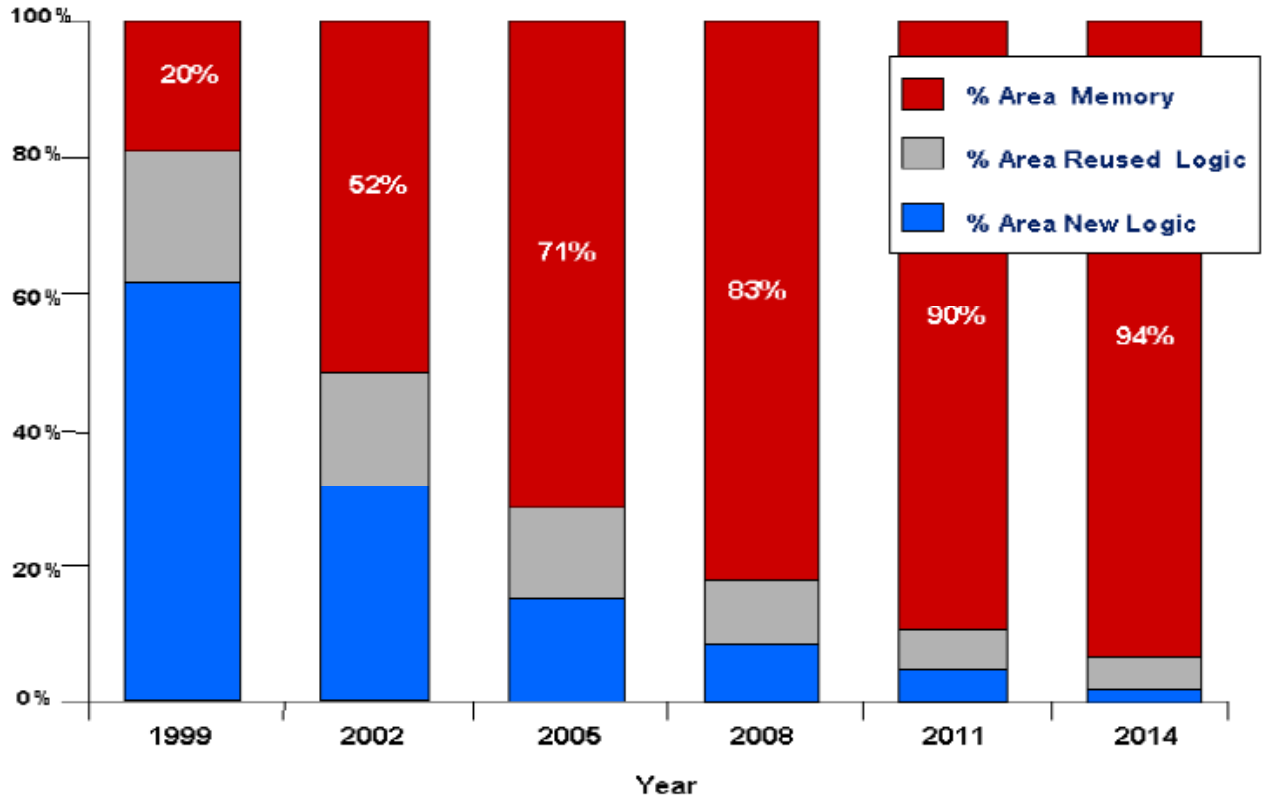
Multiple Seed / Reseeding



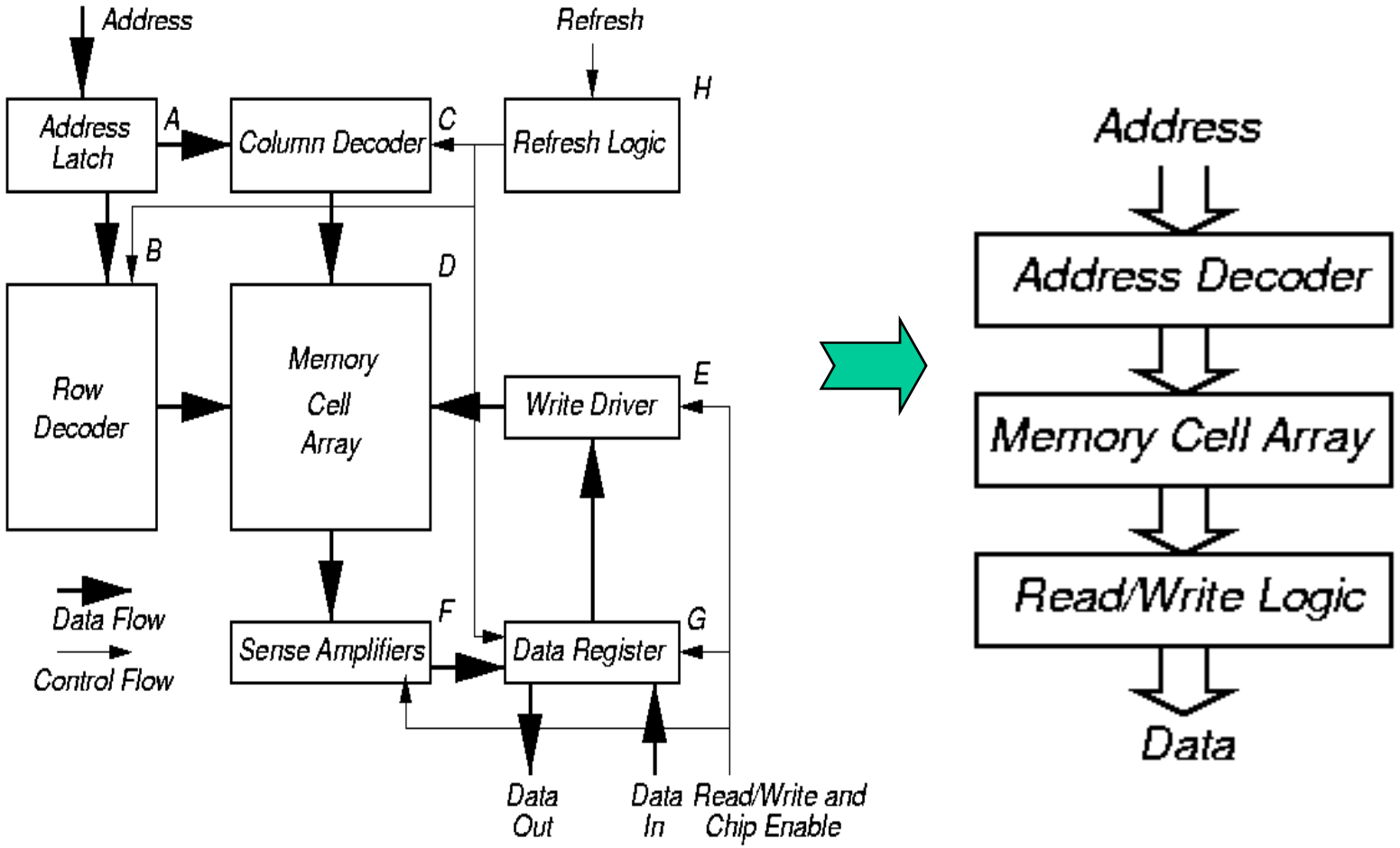
Bit Fixing/Bit Flipping Method



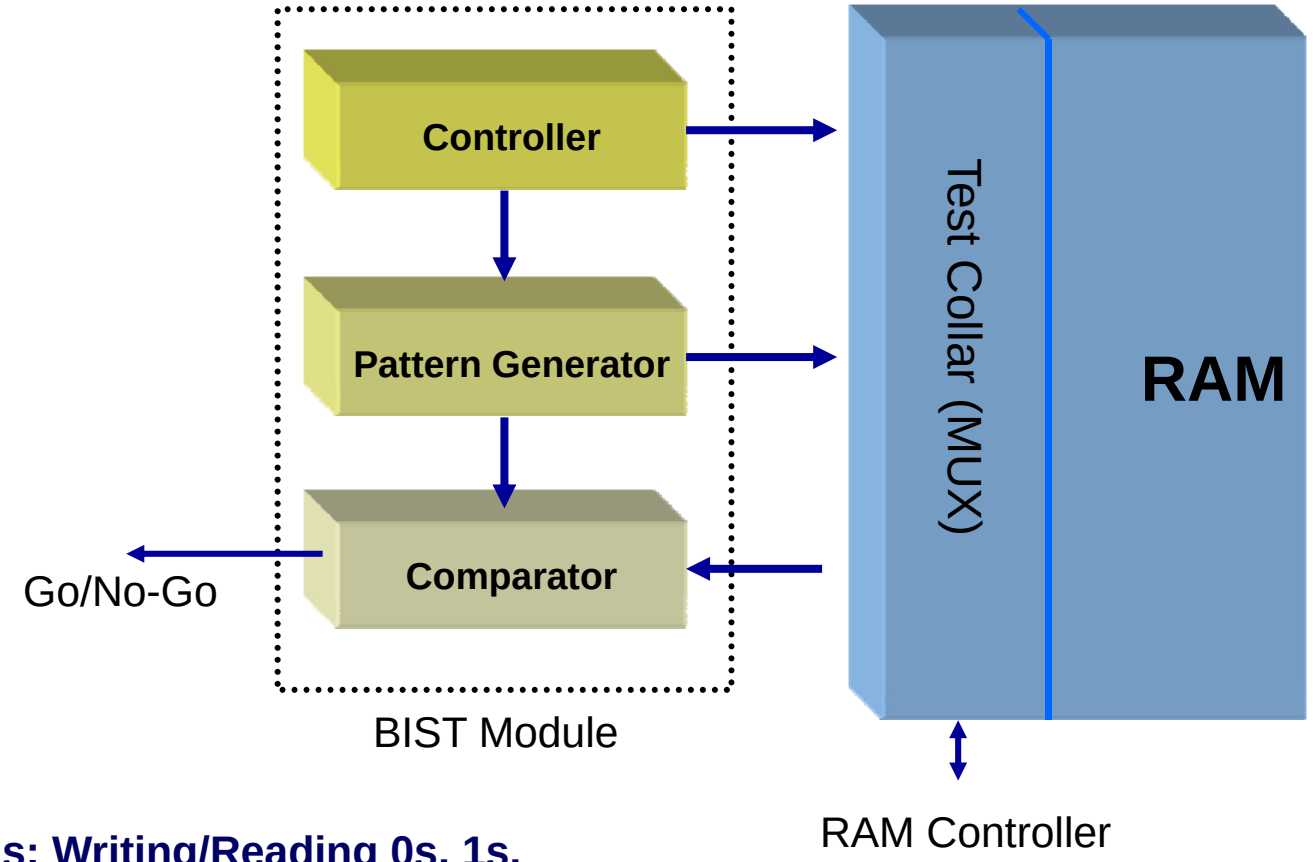
Embedded Memory in SOC



Memory Functional Model



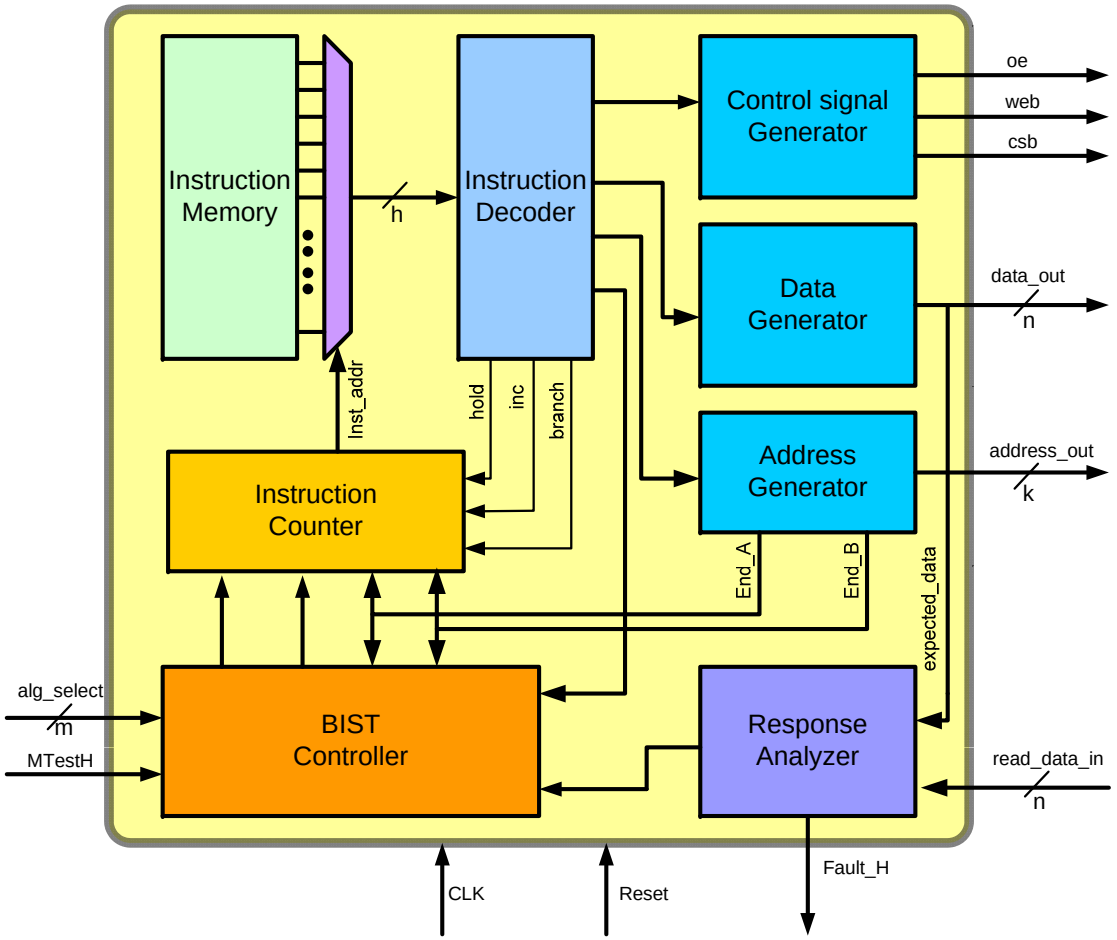
Memory BIST Concept



Patterns: Writing/Reading 0s, 1s,
Walking 0s, 1s
Galloping 0s, 1s

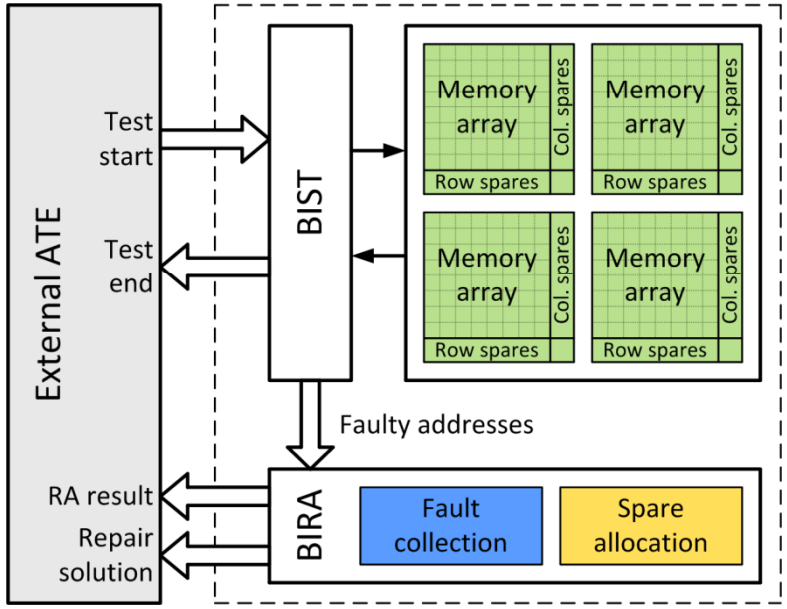
PMBIST

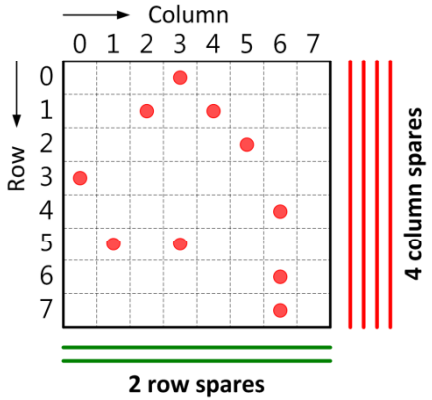
◆ Architecture



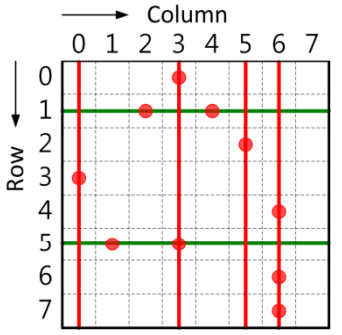
Built-In Redundancy Analysis (BIRA)

- 메모리 셀의 고장을 분석하고 여분의 셀을 이용해 고장을 수리하기 위한 효율적인 방법을 찾고자 함
- 수리 비용 및 시간의 단축을 위하여 **redundancy analysis** 방식을 **built-in**하여 메모리 셀을 수리 → **Hardware overhead** 및 **repair rate**의 고려가 필요
- 성능평가 기준**: repair rate, area overhead, rate of overused spares, analysis sped**





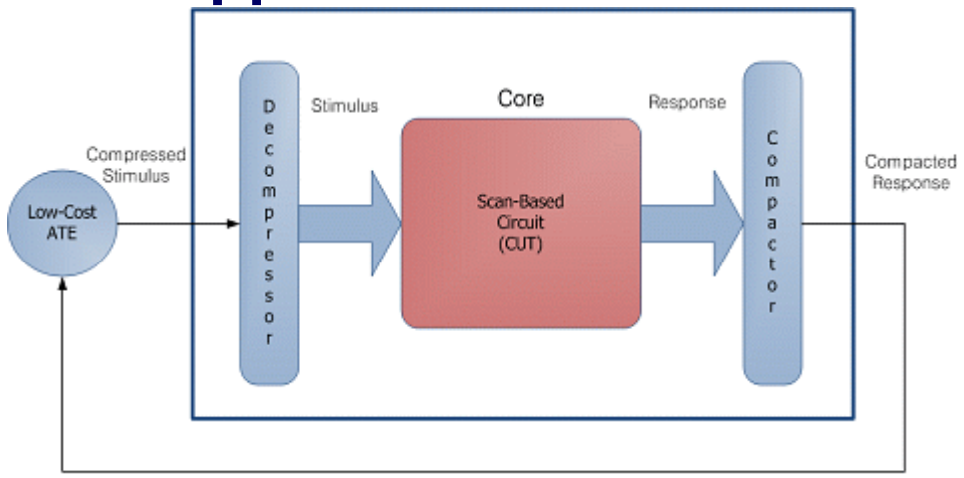
고장 수리 전 메모리



고장 수리 후 메모리

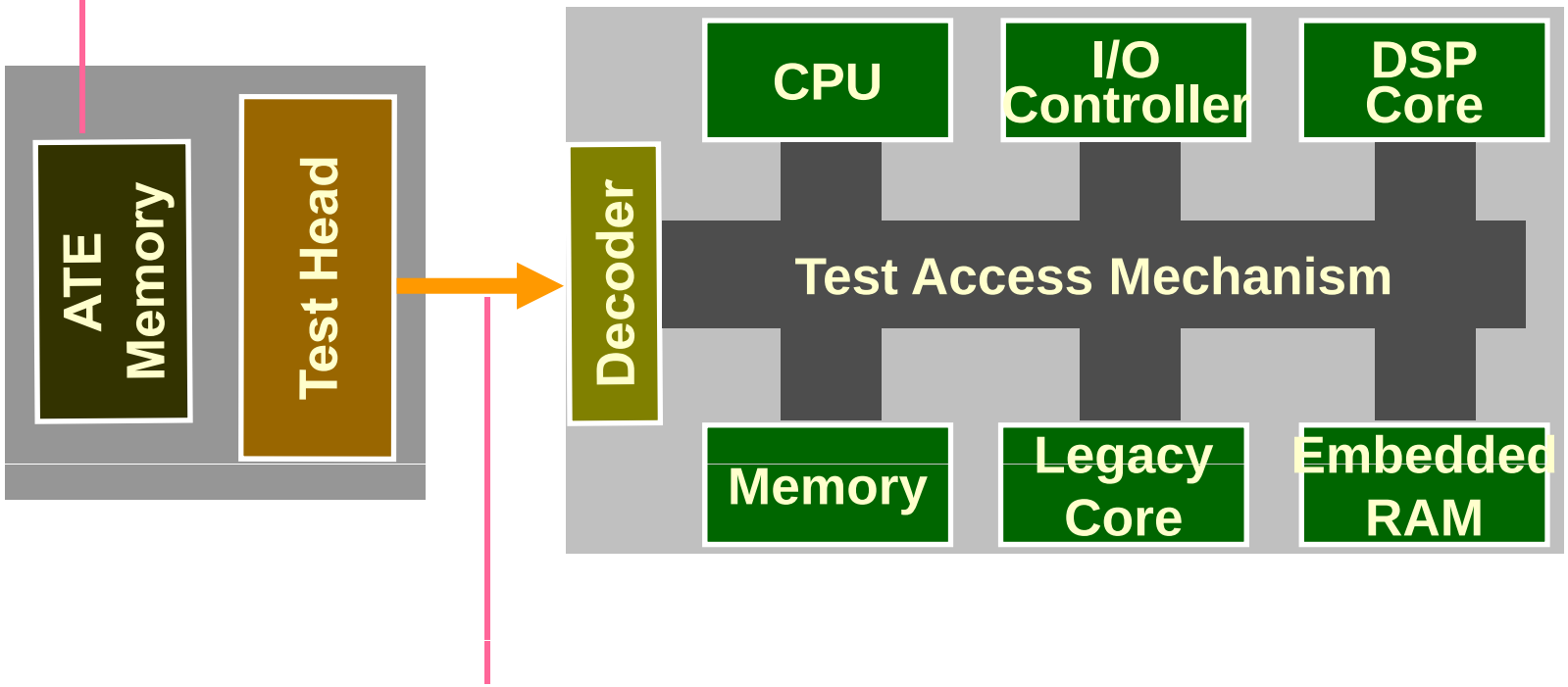
Test Compression

- **Compress the test input sequences**
- **Need a decompression units to make original test sequences**
- **Can be reduced for both limitations of ATE**
  - **The size of ATE memory**
  - **The width of ATE channel**
- **Can be reduced test application time**



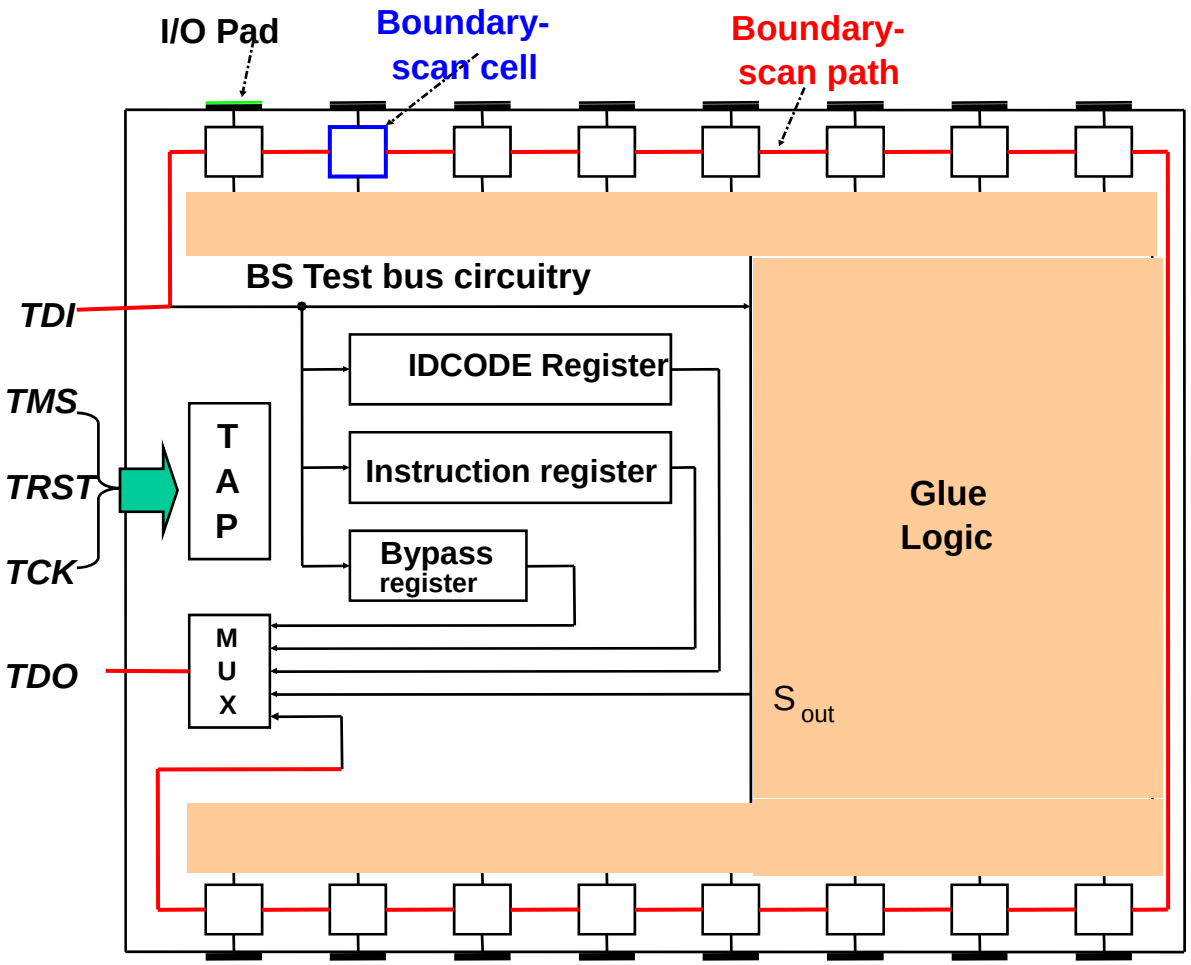
Test Compression Concepts

Smaller Memory (storing encoded test data)

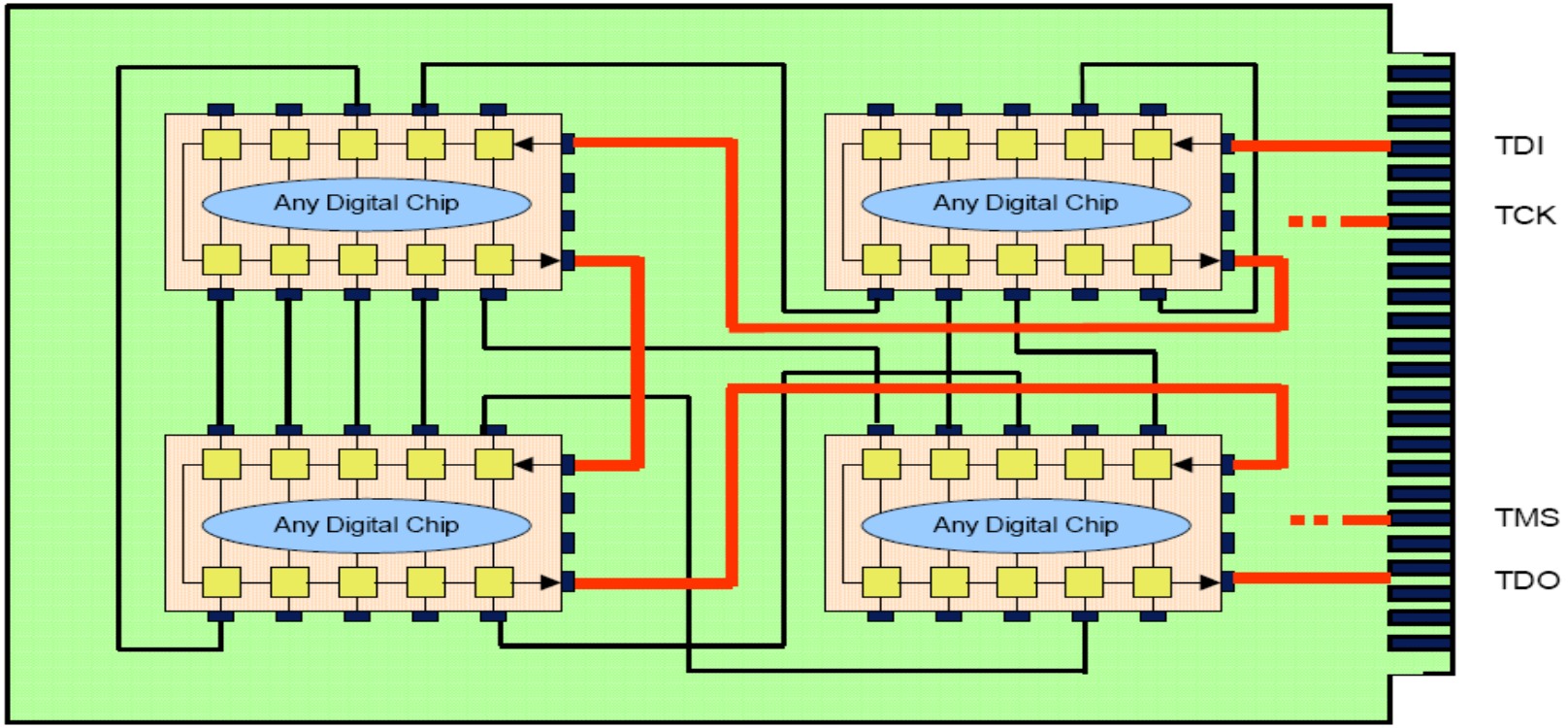


Lower bandwidth

IEEE 1149.1 Device Architecture

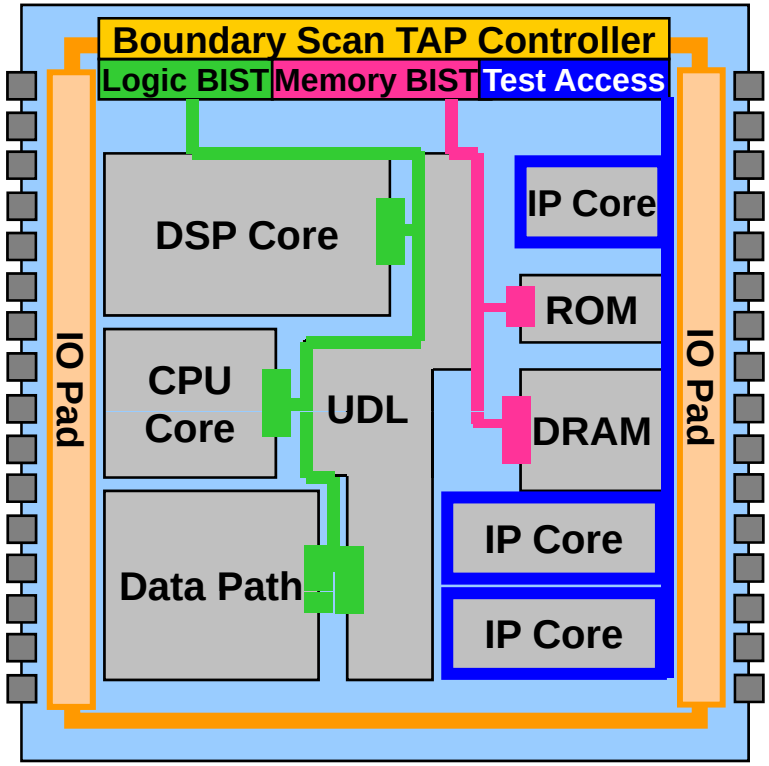


The Principle of BS Architecture

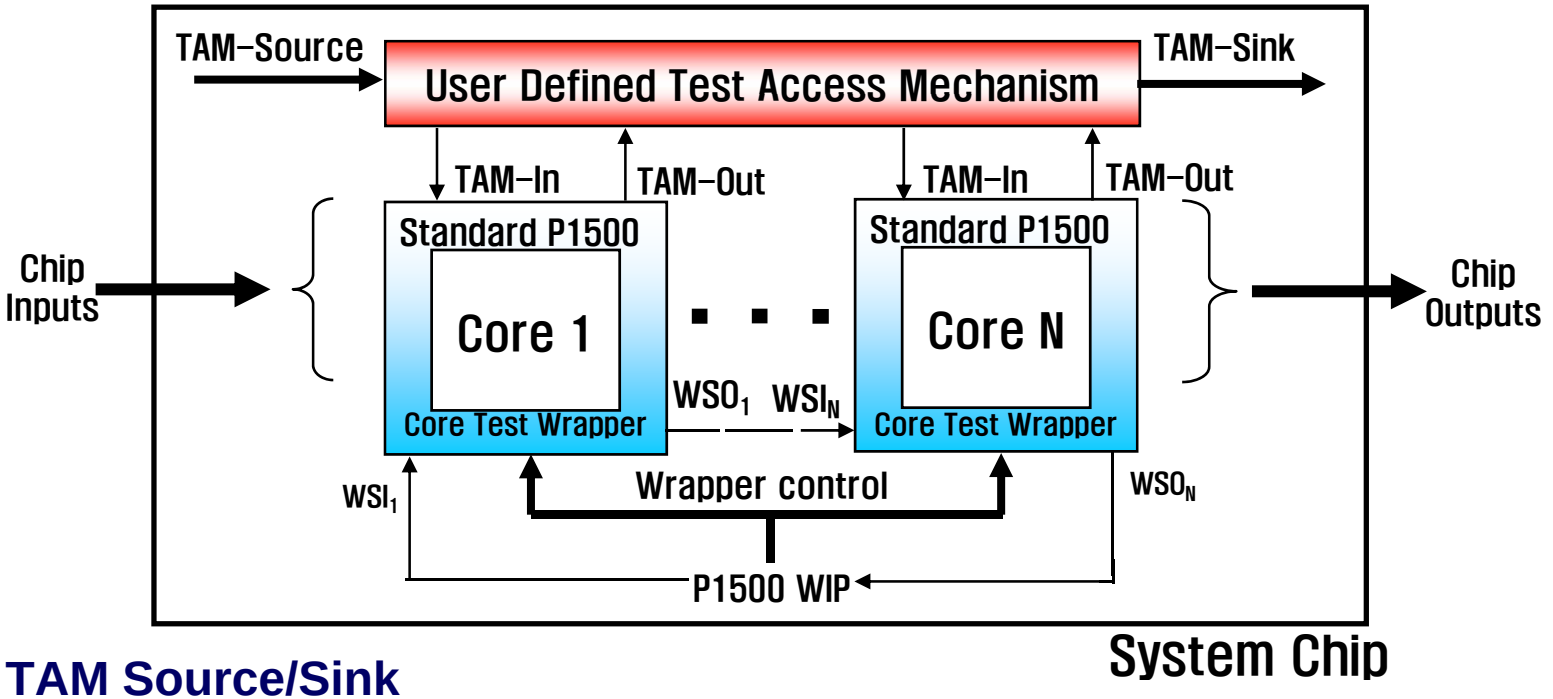


SOC Design Evolution

- Emergence of very large transistor counts on a single chip
- Mixed technologies on the same chip
- Creation of Intellectual Property (IP)
- Reusable IP-based design

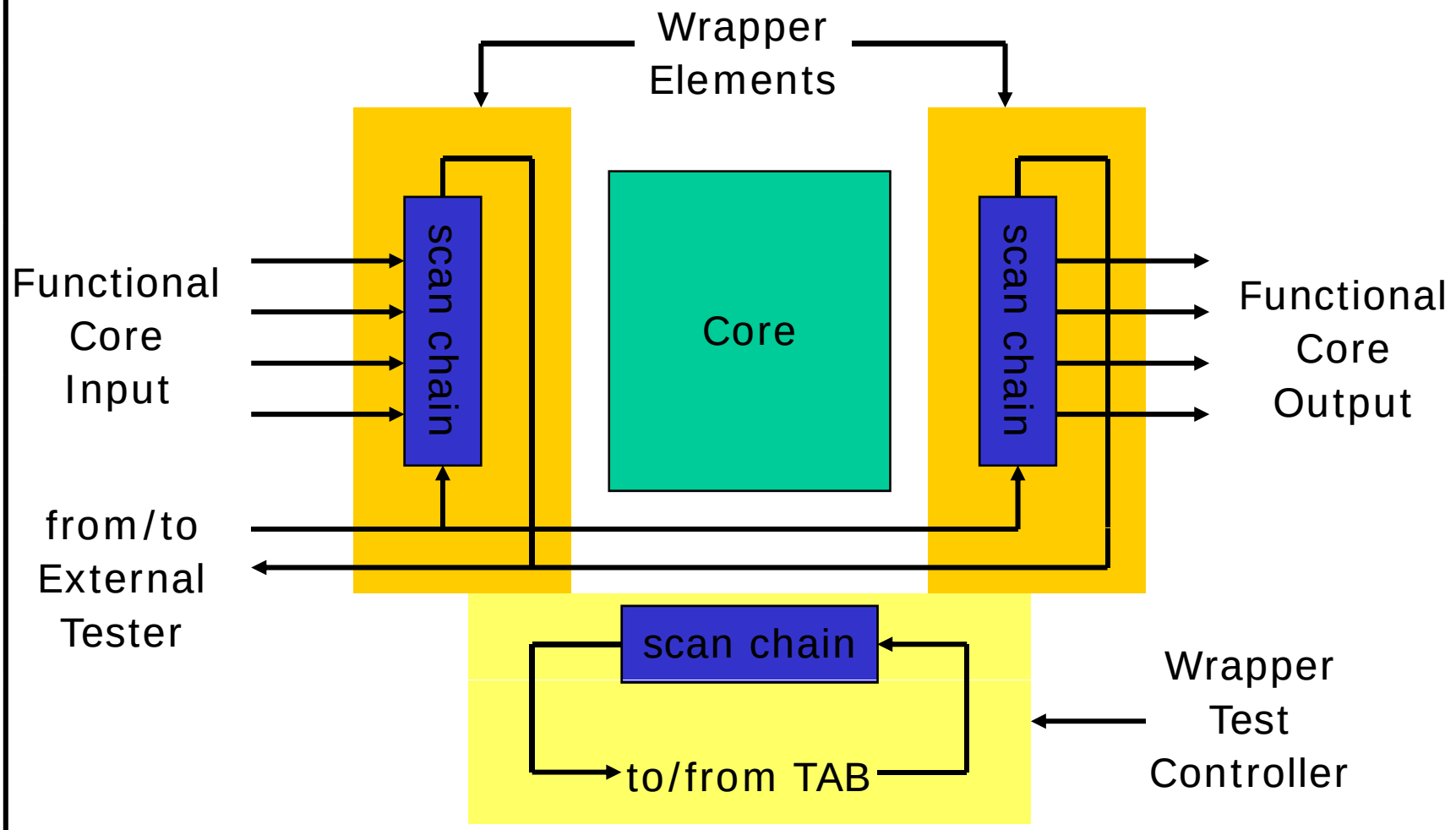


SoC with P1500 Wrapped Cores



- **TAM Source/Sink**
  - From chip I/O, test bus/rail/port, BIST, etc..
- **TAM In/Out**
  - 0 to n lines for parallel and/or serial test data, or test control
- **P1500 Wrapper Interface Port (WIP)**
  - From chip-level TAP controller, chip I/0, …

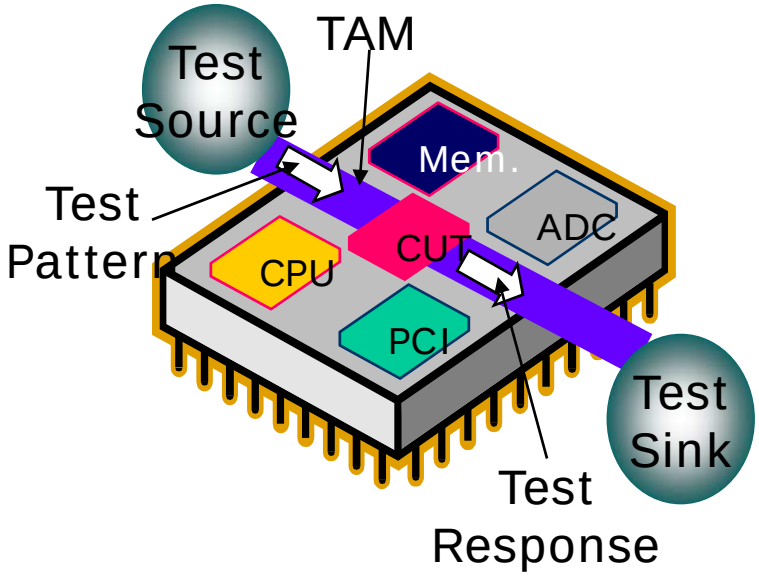
Test Wrapper



TAM (Test Access Mechanism)

P1500

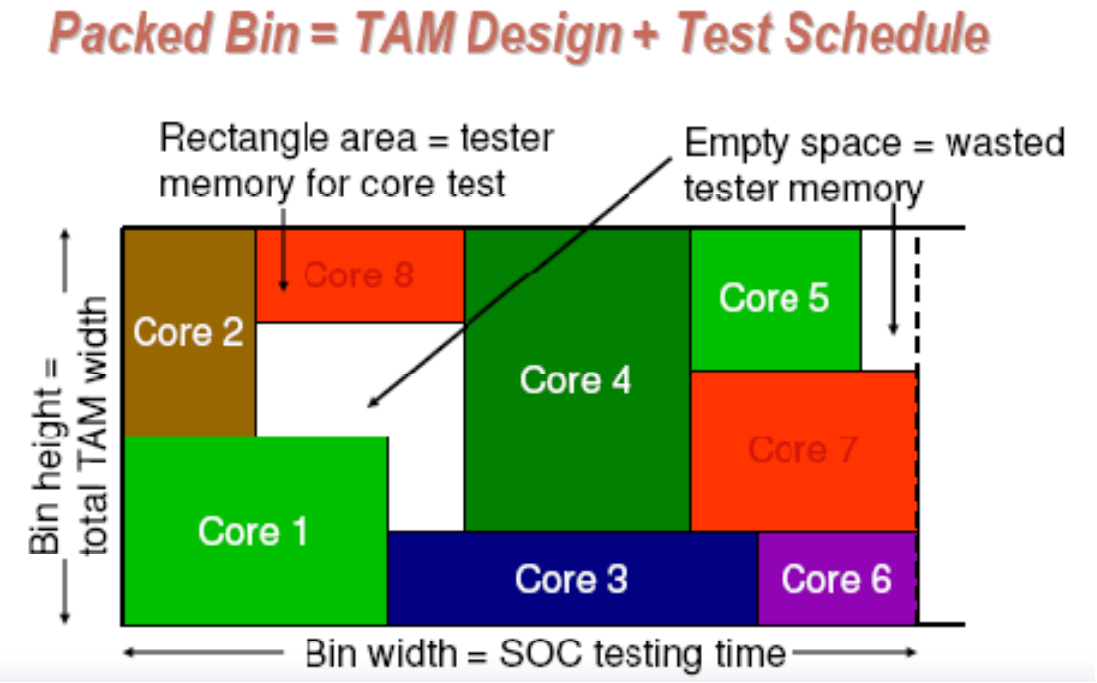
- **Test access mechanism (TAM)**
  - **Defined as user-defined test data communication structure**
  - **Carries test signals from source to module, and module to sink**
  - **Tests module interconnects via test-wrappers**
  - **May contain bus, boundary-scan and analog test bus components.**



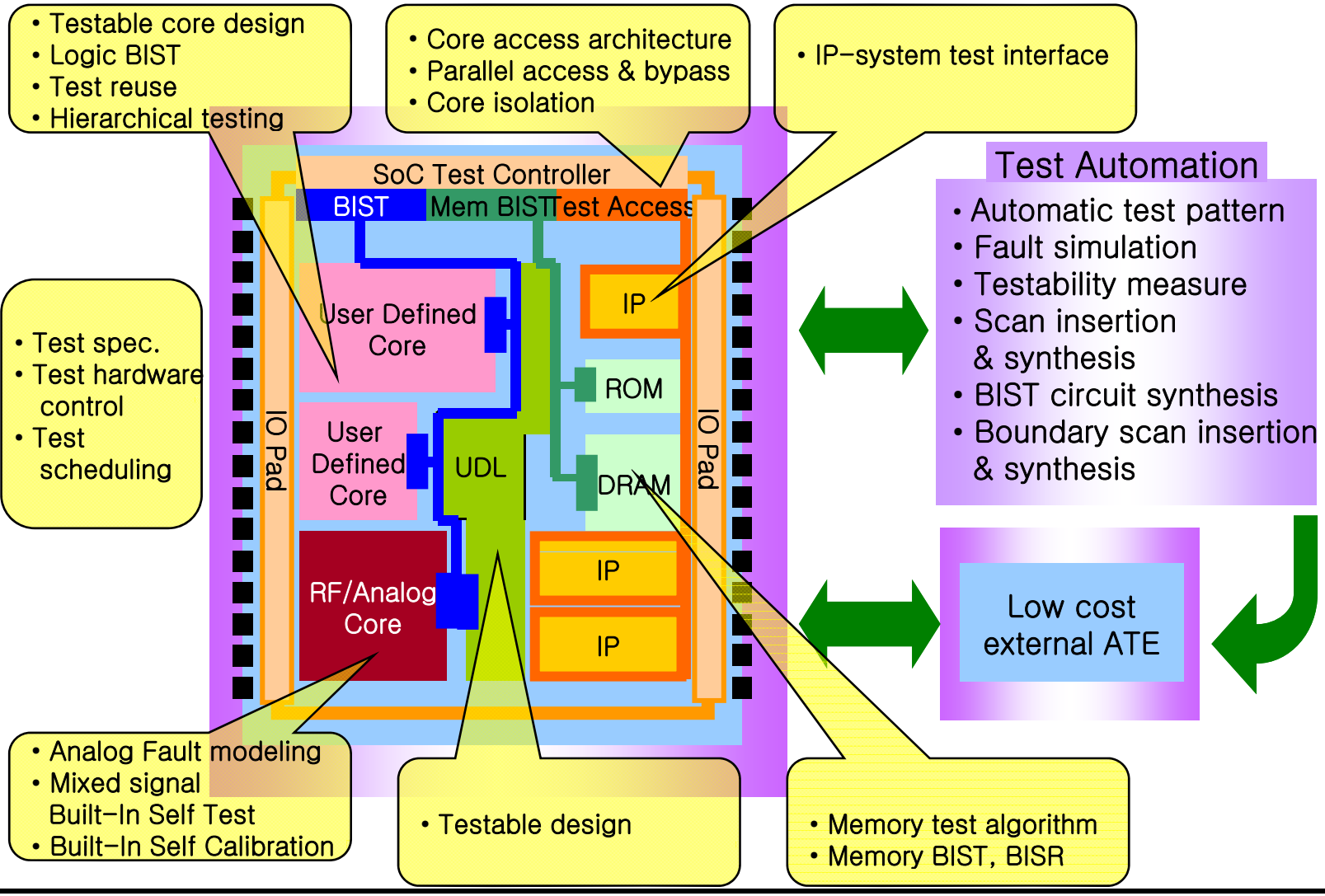
Test Scheduling

P1500

- **Test scheduling problem**
  - **Determine the minimal number of test sessions required to test all IPs**
  - **Determine the minimal colors that can be assigned to the nodes of a graph such that no edge connects two nodes of the same color**
- **More complex when test time and power are considered**



Conclusion


• Testable core design
• Logic BIST
• Test reuse
• Hierarchical testing
• Core access architecture
• Parallel access & bypass
• Core isolation
• IP–system test interface
SoC Test Controller
BIST
Mem BIST
Test Access
User Defined Core
IP
ROM
User Defined Core
UDL
DRAM
IO Pad
IO Pad
RF/Analog Core
IP
IP
• Test spec.
• Test hardware control
• Test scheduling
Test Automation
• Automatic test pattern
• Fault simulation
• Testability measure
• Scan insertion & synthesis
• BIST circuit synthesis
• Boundary scan insertion & synthesis
Low cost external ATE
• Analog Fault modeling
• Mixed signal Built-In Self Test
• Built-In Self Calibration
• Testable design
• Memory test algorithm
• Memory BIST, BISR