Design for Testability

- Introduction
- Testability Measurement
- DFT Basics
- DFT Techniques
  - ad hoc
  - Scan design
  - Boundary scan

Introduction

- Improve Testability (Controllability and Observability)
- Reduce ATPG Efforts

  - ATPG Development More Efficient
- Improve Test Quality

  - High Fault Coverage
- Reduce Test Cost

  - Decrease Test Pattern Length and Test Time
- Reduce Time to Market

  - Easy Debug and Diagnosis

Testability

- Controllability: The ability to set some circuit nodes to a certain states or logic values.
- Observability: The ability to observe the state or logic values of internal nodes.

Usage of Testability Measures

- Speed up test generation
- Improve the design testability
- Guide the DFT insertion

Testability Measurement

- TMEAS [Stephenson & Grason, 1976]
- SCOAP [Goldstein, 1979]
- TESTSCREEN [Kovijanic 1979]
- CAMELOT [Bennetts *et al.*, 1980]
- VICTOR [Ratiu *et al.*, 1982]

SCOAP

- Sandia Controllability Observability Analysis Program.
- Using integers to reflect the difficulty of controlling and observing the internal nodes.
- Higher numbers indicate more difficult to control or observe.
- Applicable to both combinational & sequential circuits.

Measurement in SCOAP

- For a node A:

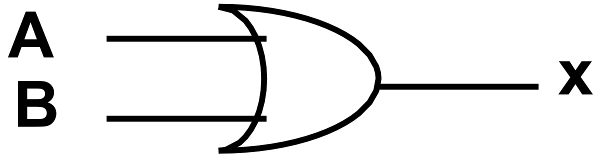
$CC^0(A)$: Combinational 0-controllability
$CC^1(A)$: Combinational 1-controllability

$SC^0(A)$: Sequential 0-controllability
$SC^1(A)$: Sequential 1-controllability

$CO(A)$: Combinational observability
$SO(A)$: Sequential observability

Combinational Components in SCOAP

Ex:



$CC^0(x) = CC^0(A) + CC^0(B) + 1;$

$CC^1(x) = \min\{CC^1(A), CC^1(B)\} + 1.$

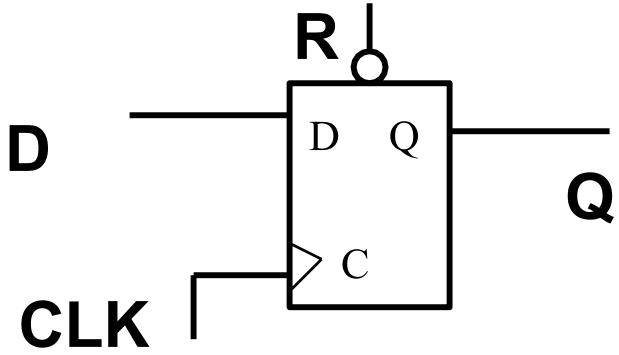
$SC^0(x) = SC^0(A) + SC^0(B)\ ;$

$SC^1(x) = \min\{SC^1(A), SC^1(B)\}.$

- CC implies the distance from PI
- SC implies the number of time frames needed to provide a 0 or 1.

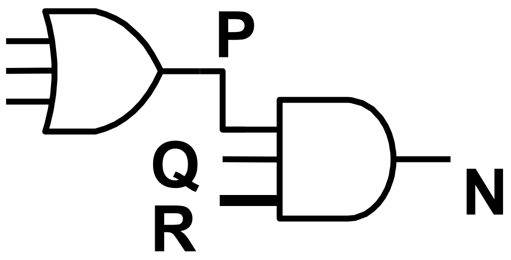
Sequential Components in SCOAP

Ex:

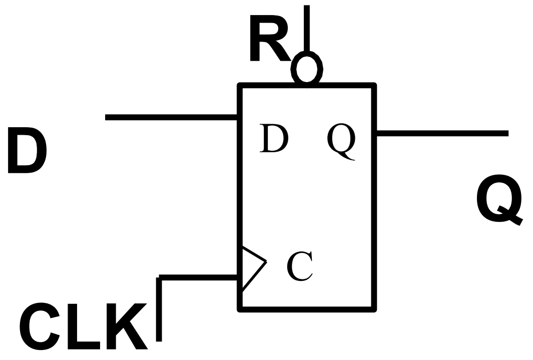


- $CC^0(Q)$ = min{$CC^0(R)$, $CC1(R) + CC^0(D) + CC^0(C) + CC^1(C)$}
- $CC^1(Q) = CC^1(R) + CC^1(D) + CC^0(C) + CC^1(C)$
- $SC^0(Q)$ = min{$SC^0(R)$, $SC^1(R) + SC^0(D) + SC^0(C) + SC^1(C)$} + 1
- $SC^1(Q) = SC^1(R) + SC^1(D) + SC^0(C) + SC^1(C)$ + 1

Observalibity in SCOAP



- $CO(P) = CO(N) + CC^1(Q) + CC^1(R) + 1$
- $SO(P) = SO(N) + SC^1(Q) + SC^1(R)$



- $CO(R) = CO(Q) + CC^1(Q) + CC^0(R)$
- $SO(R) = SO(Q) + SC1(Q) + SC^0(R) +1$

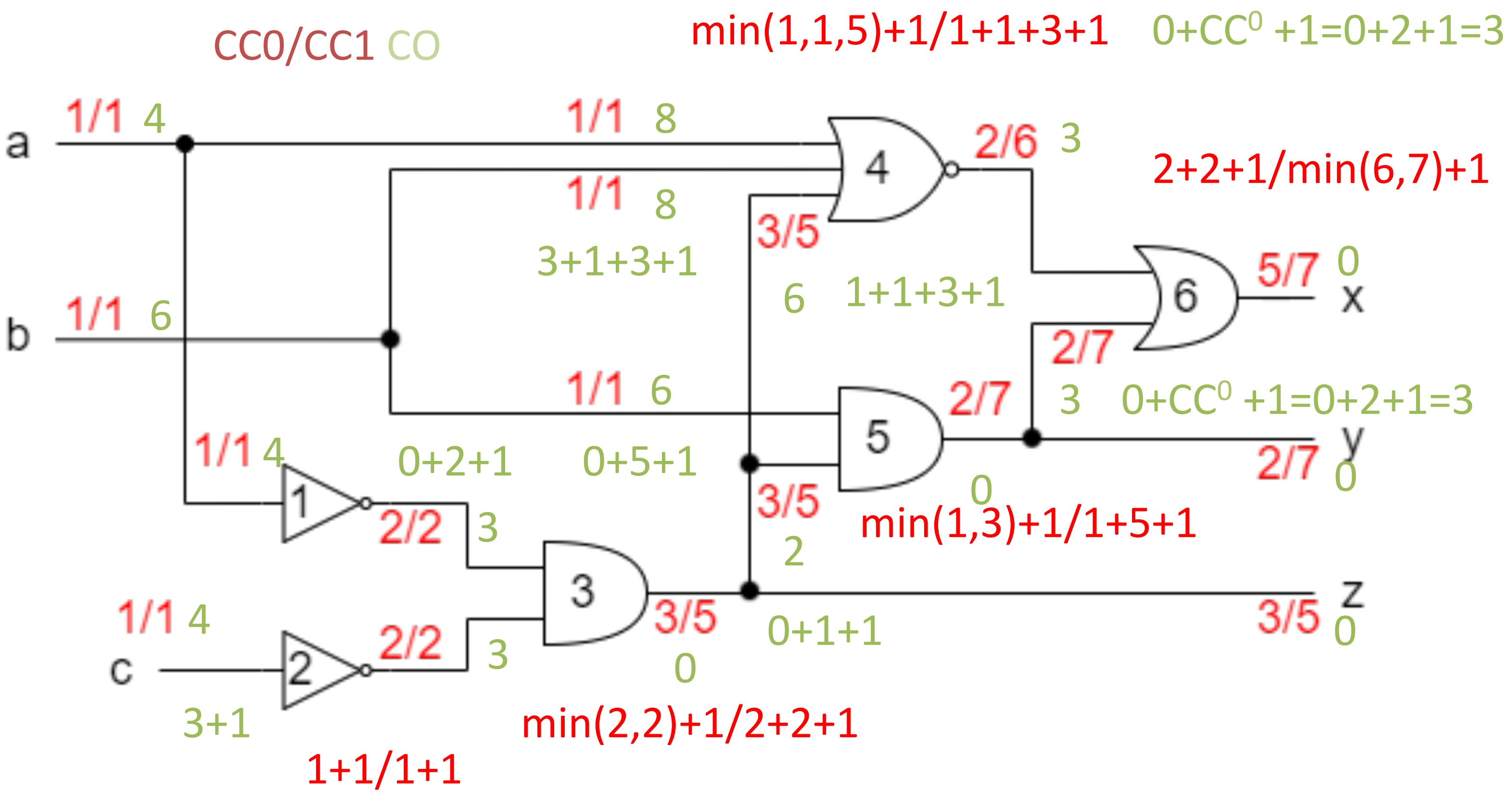
Initial States

- PI: $CC^0 = CC^1 = SC^0 = SC^1 = 1$
- PO: $CO = SO = 0$
- All other numbers are initially set to ∞

Importance of Testability Measures

- They can guide the designers to improve the testability of their circuits.
- Test generation algorithms using heuristics usually apply some kind of testability measures to their heuristic operations (e.g., in making search decisions), which greatly speed up the test generation process.

Testability Measures Example (combinational)



Design for Testability (DFT)

- **Test Costs:**
  - Test Generation
  - Fault Simulation
  - Fault Location
  - Test Equipment
  - Test Application Time
- **Test Difficulties:**
  - Sequential > Combinational
  - Control Logic > Data Path
  - Random Logic > Structured Logic
  - Asynchronous > Synchronous
- **Testability**
  - Controllability
  - Observability

Design for Testability (DFT)

- DFT techniques are design efforts specifically employed to ensure that a device in testable.
- In general, DFT is achieved by employing extra H/W.
  - ⇒Conflict between design engineers and test engineers.
  - ⇒ Balanced between amount of DFT and gain achieved.
- Examples:
  - DFT
    - ⇒Area ↑ & Logic complexity ↑
    - ⇒Yield ↓
    - ⇒For fixed fault coverage, defect level ↑
  - Therefore, DFT must guarantee to increase fault coverage.

Benefits of DFT

- In general, DFT has the following benefits:
  - Fault coverage ↑
  - Test generation (development) time ↓
  - Test length
  - Test Memory
    hope ↓
  - Test application time
  - Support a test hierarchy
    - Chips
    - Boards
    - Subsystems
    - Systems
  - Concurrent engineering
  - Reduce life-cycle costs

⇒ Pay less now and pay more later without DFT!

Costs Associated with DFTs

- Pin Overhead
- Area / Yield
- Performance degradation
- Design Time

⇒ There is no free lunch !

DFT Techniques

- ad hoc DFT technology
- Scan-based design
- Built-In Self-Test (BIST)
- Boundary Scan

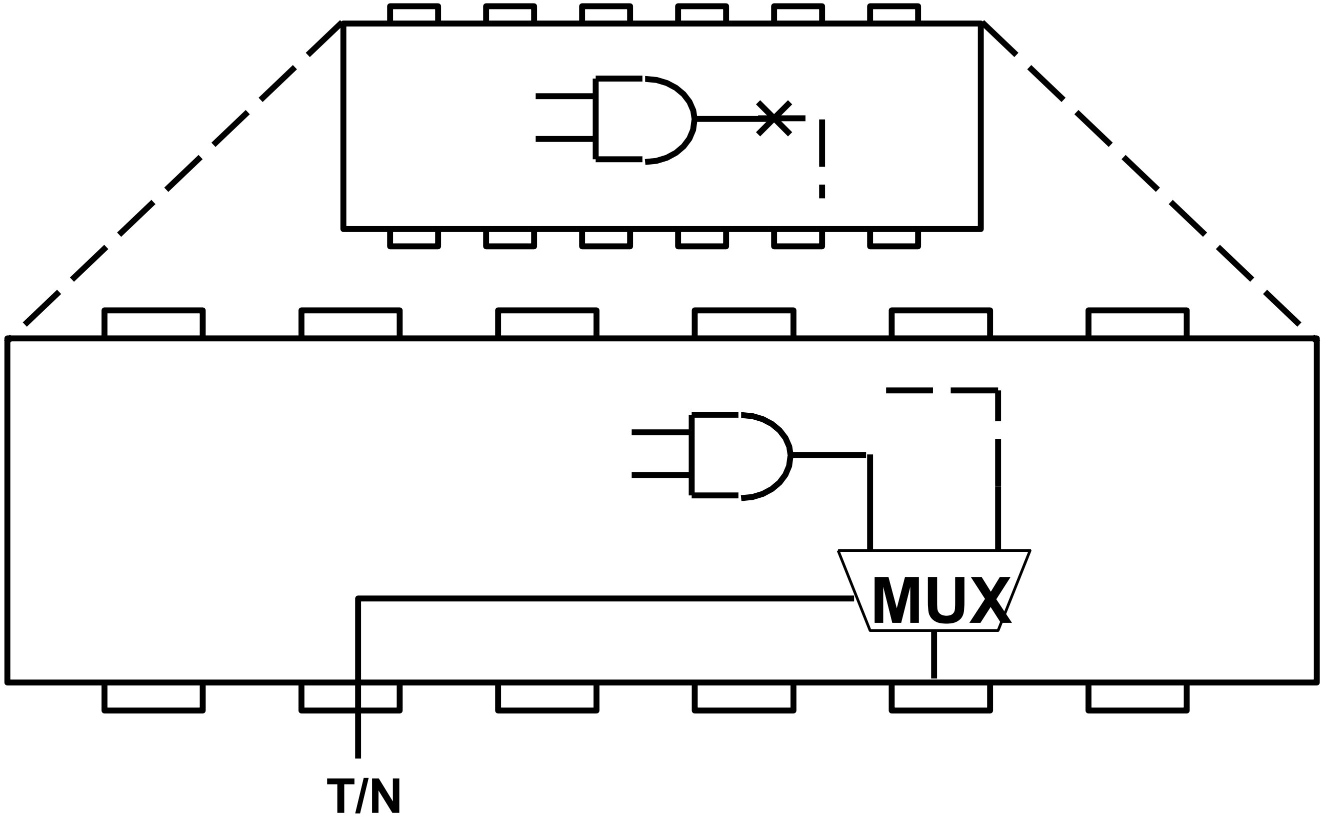
Ad hoc Techniques

- Test points
- Initialization
- Monostable multivibrators (one shot)
- Oscillators and clocks
- Counter / Shift registers
- Partitioning large circuits
- Logic redundancy
- Break global feedback paths

⋮

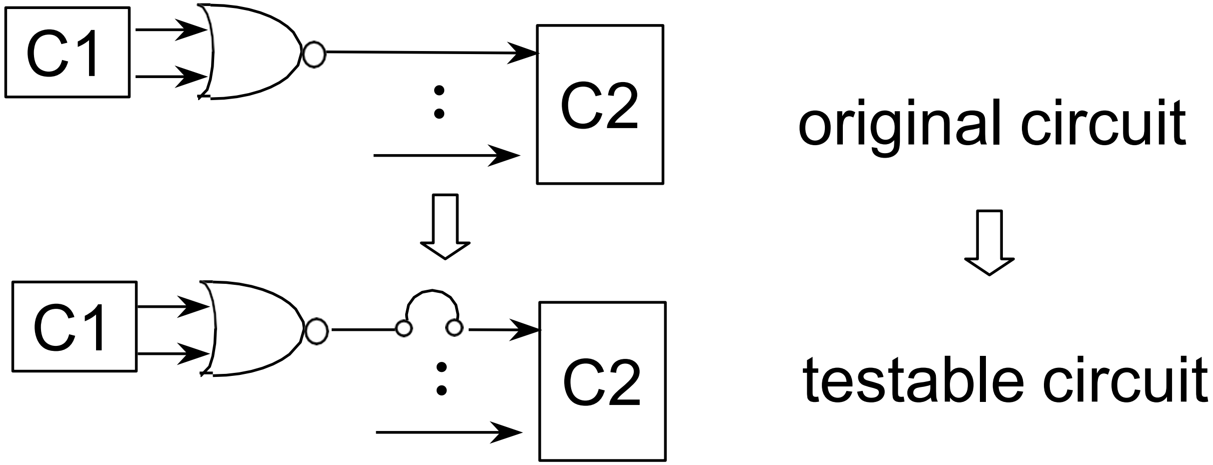
Example of *ad hoc* Techniques

- Insert test point



Test Points

- **Rule : to enhance controllability and observability by inserting control points (cp) and observation points (op), respectively.**
- **Ex:**



original circuit

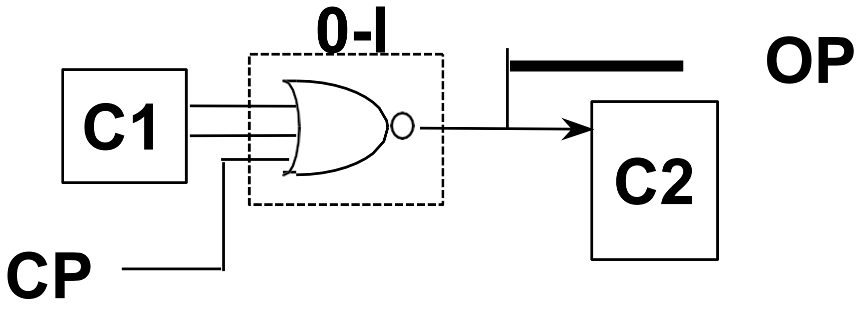
⇩

testable circuit

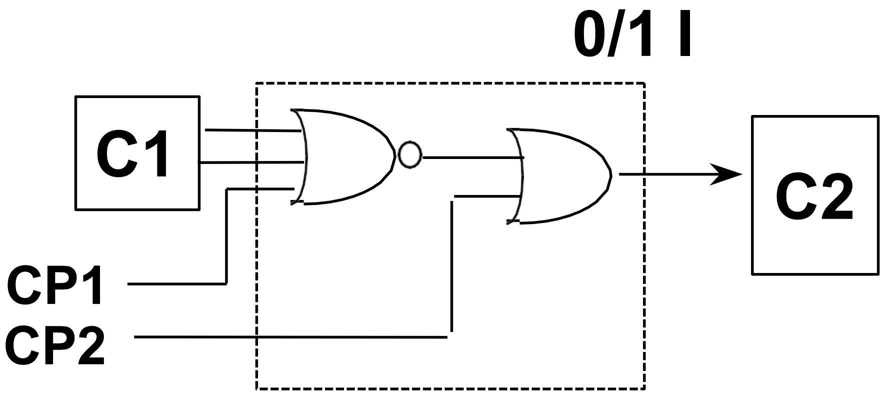
⇨ **can be done only for board**

Test Points (Cont.)

- Using a CP for 0-injection and an OP for observability:

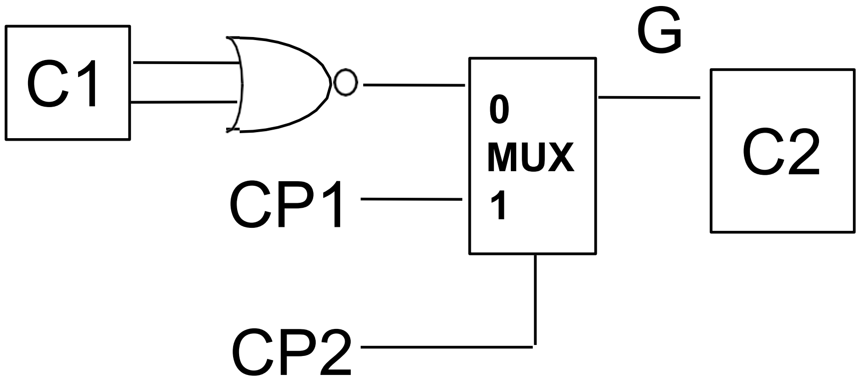


- 0/1 Injection:



Test Points (Cont.)

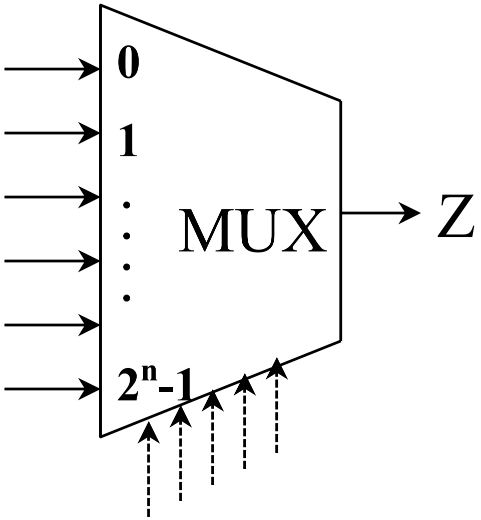
- **Using a MUX**



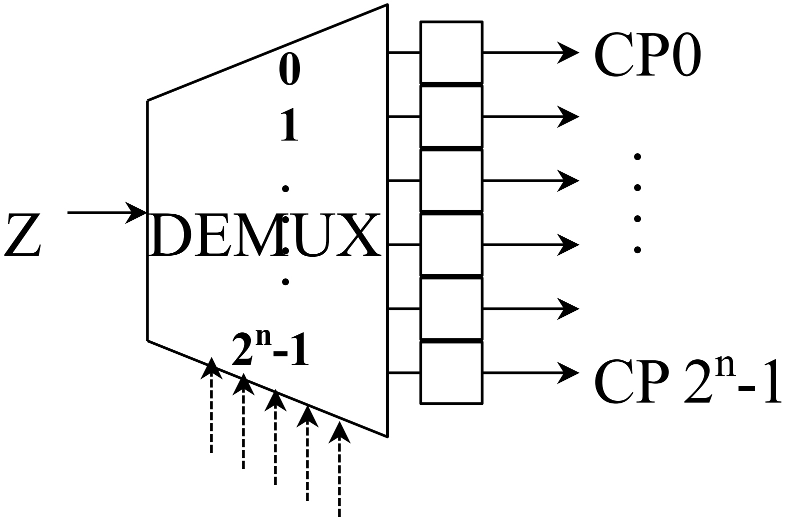
CP2=0 : normal
CP2=1: G=1 if CP1=1
G=0 if CP2=0

Test Points (Cont.)

- **Multiplexing Observation Points:**



- **Demultiplexing and Latching Control Points:**



Selection of CP

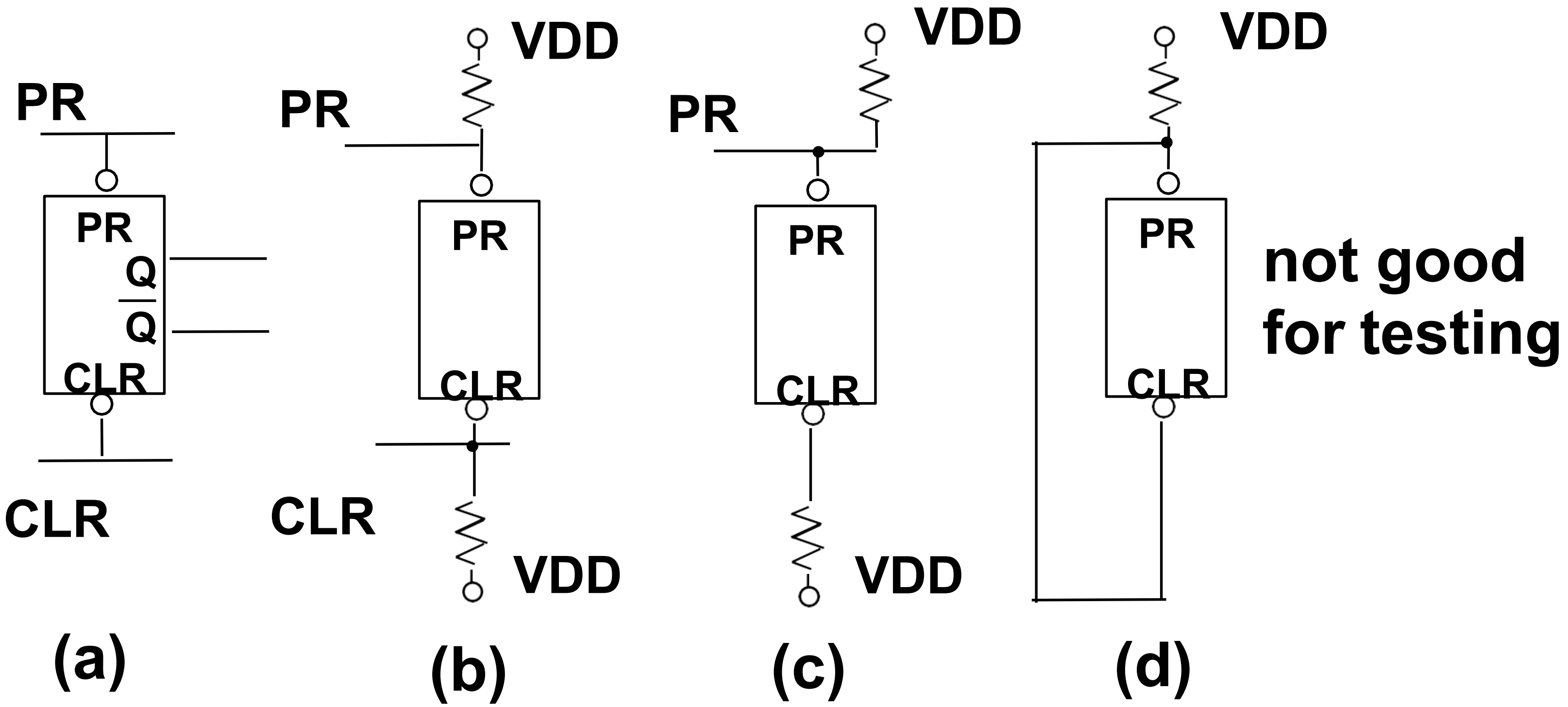
- Control, address and data bus lines on bus-structured designs.
- Enable/hold inputs to microprocessors.
- Enable and Read/write inputs to memory.
- Clock and preset/reset inputs to F/Fs, counters, shift registers, etc.
- Data select inputs to multiplexers and demultiplexers.
- Control lines on tri-state devices.

Selection of OP

- Stem lines with high fanout.
- Global feedback path
- Redundant signal lines
- Outputs of devices with many inputs, e.g., multiplexers and parity generators.
- Outputs from state devices.
- Address, control, data buses

Initialization

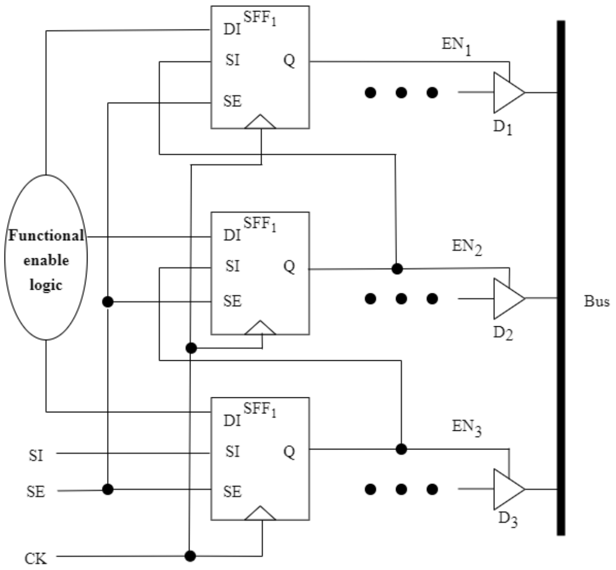
- Rule: Design circuits to be easily initialized
  - Don’t disable preset and clear lines



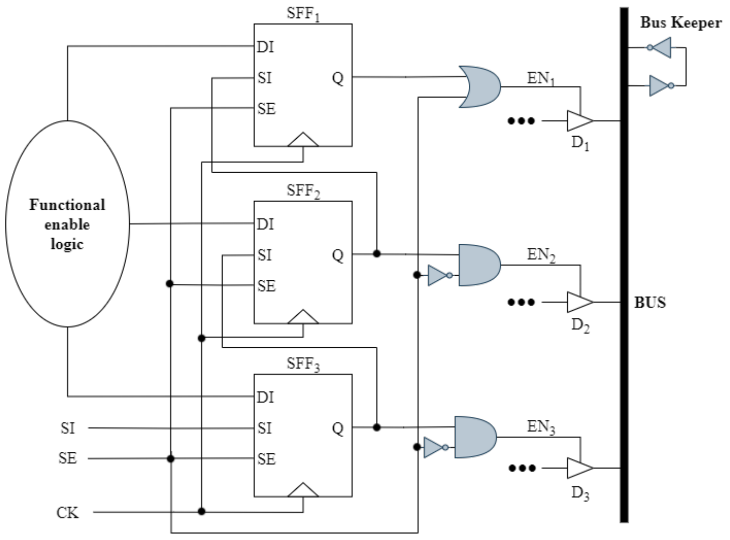
not good for testing

How to avoid ‘x’

Bus contention while doing scan shift or undriven bus



Original design

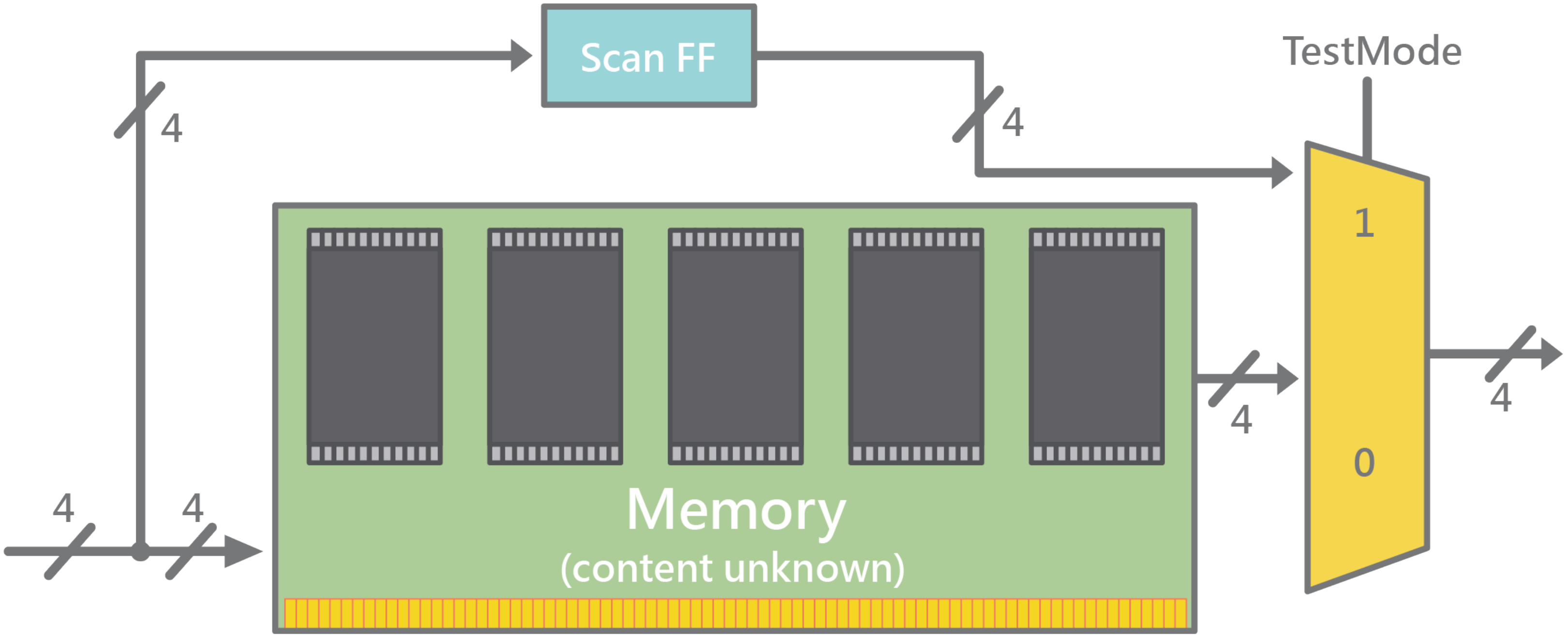


Testable design

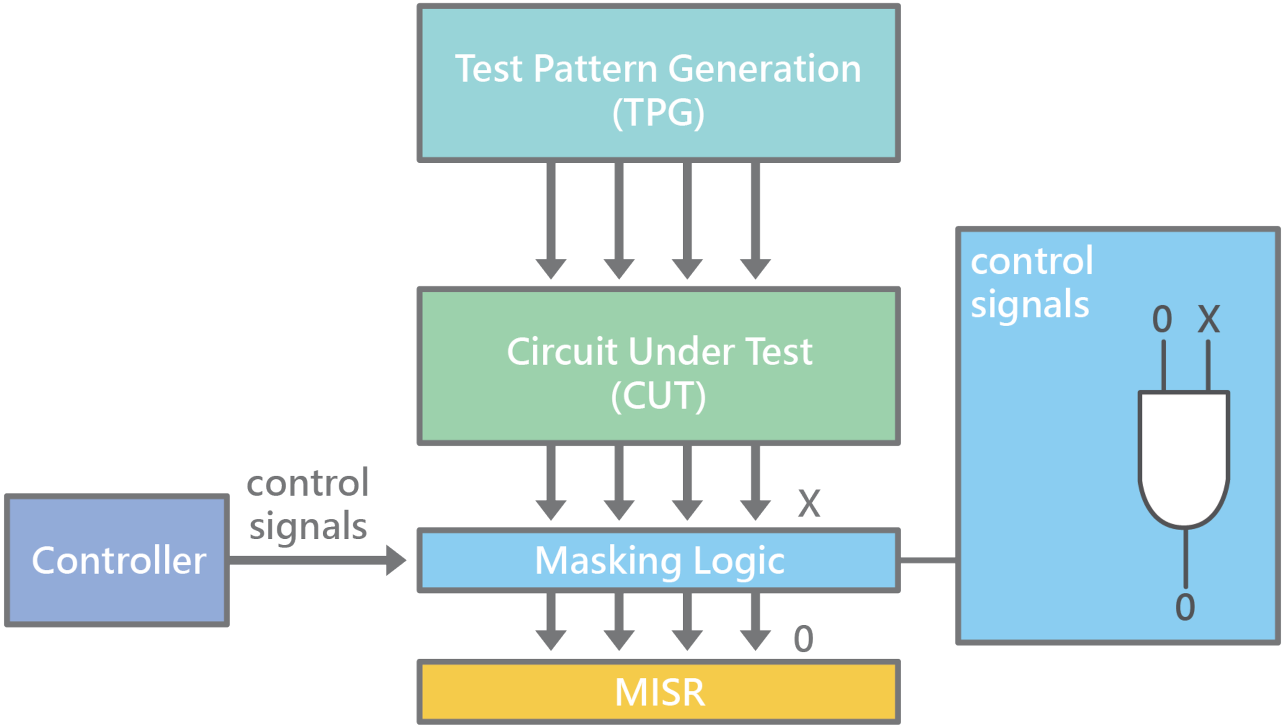
No bus contention when SE=1

No floation bus

How to avoid ‘x’

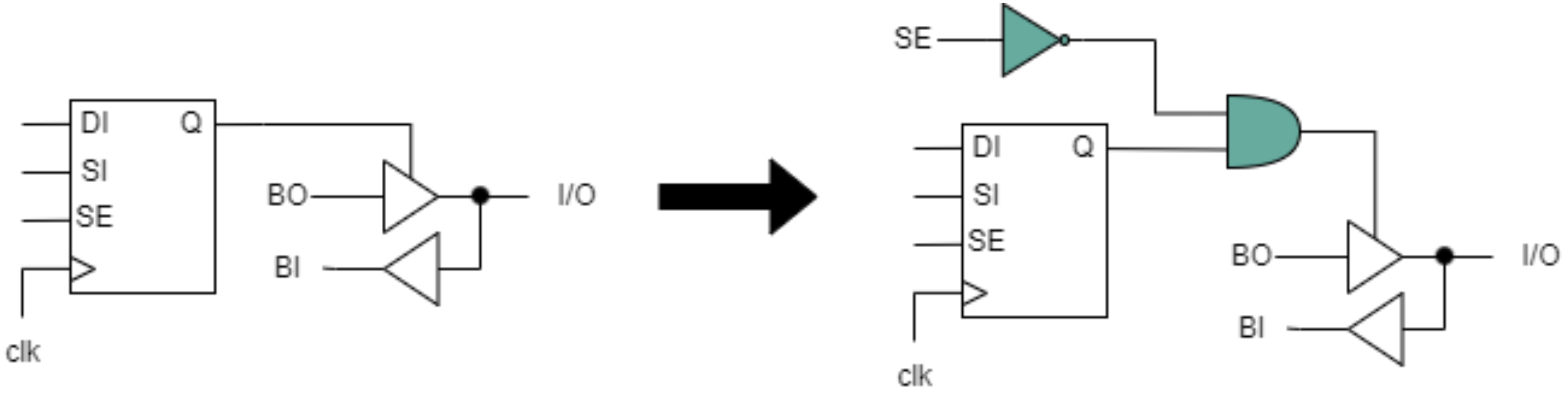


Mask Logic before MISR



Control Random I/O Direction

Random I/O direction during scan shift

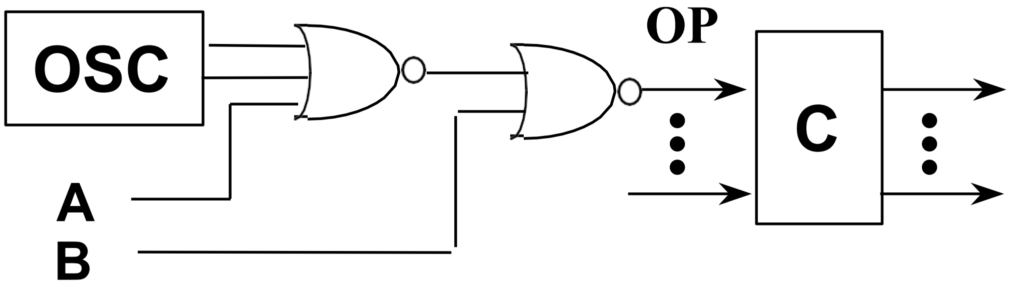


Original Design

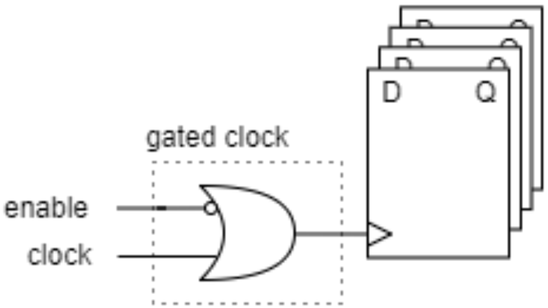
Testable design

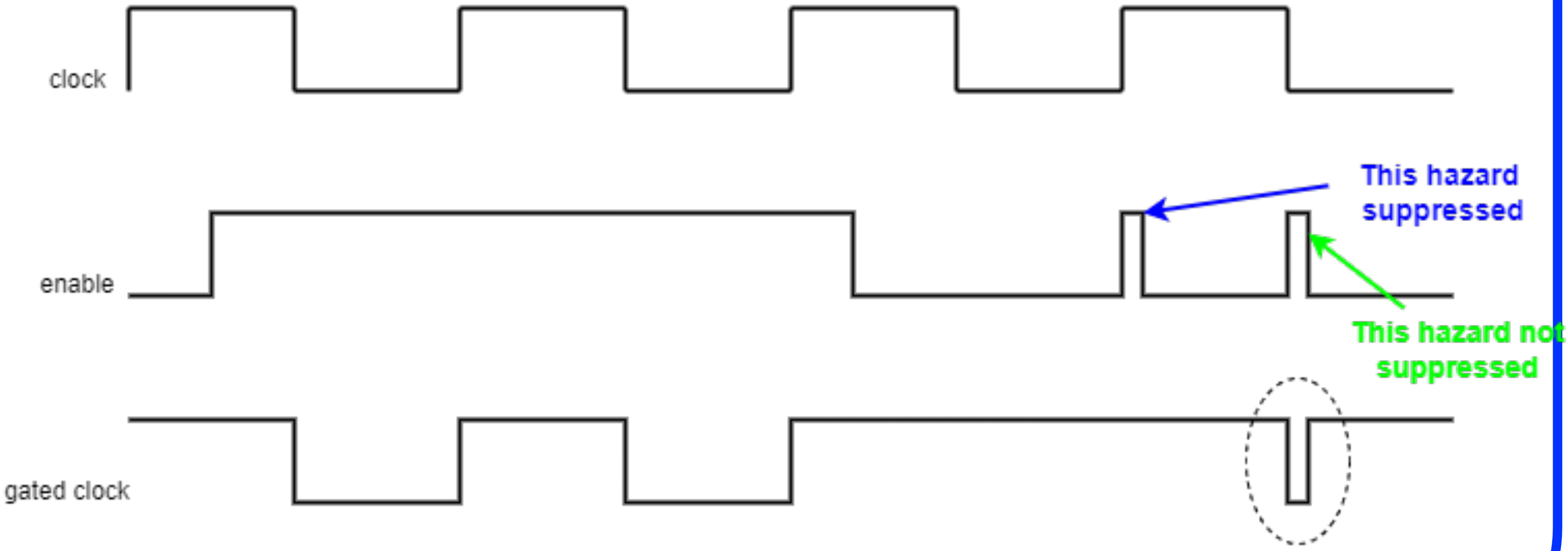
Monostable Multivibrator, Oscillators and Clocks

- Rule: Disable internal one shot, OSC and clocks
  - inserting CP and/or OP while disabling these devices
- Example:



Gated Clock Enable Signal Noise



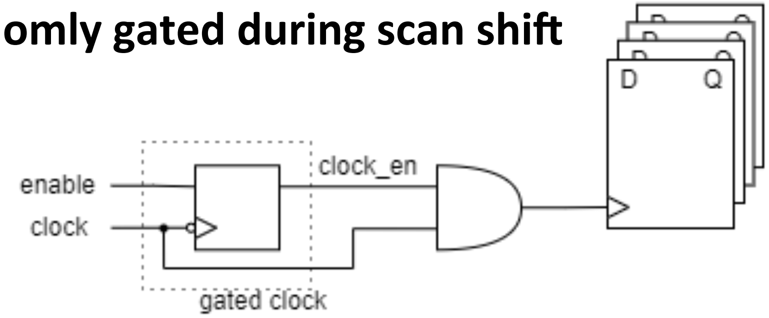


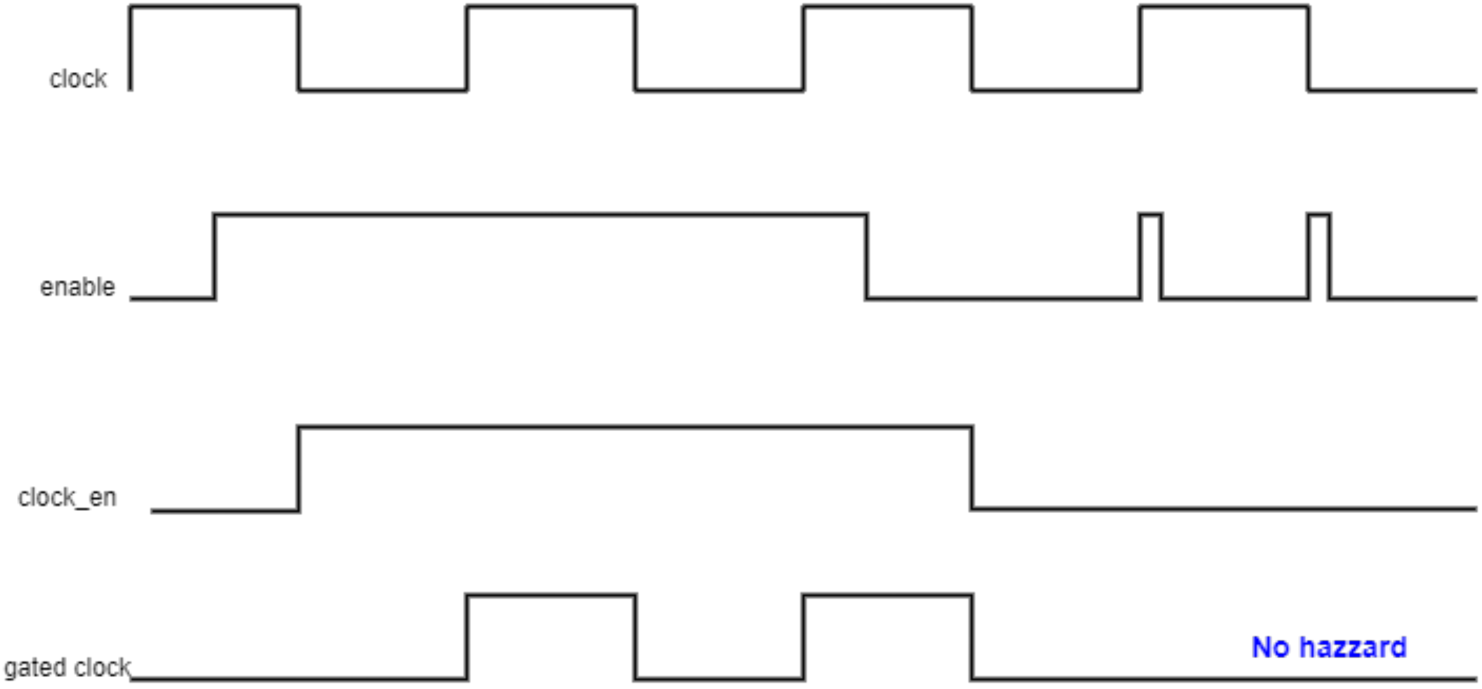
This hazard suppressed

This hazard not suppressed

Cope With Enable Signal Noise(1)

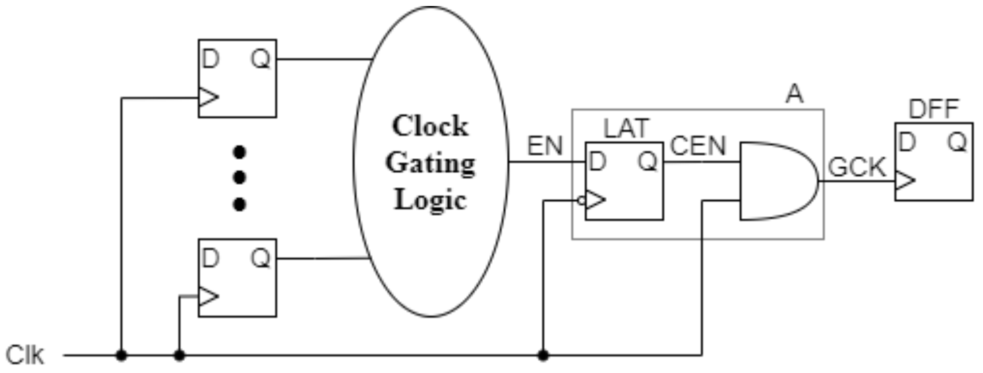
- Clock is randomly gated during scan shift



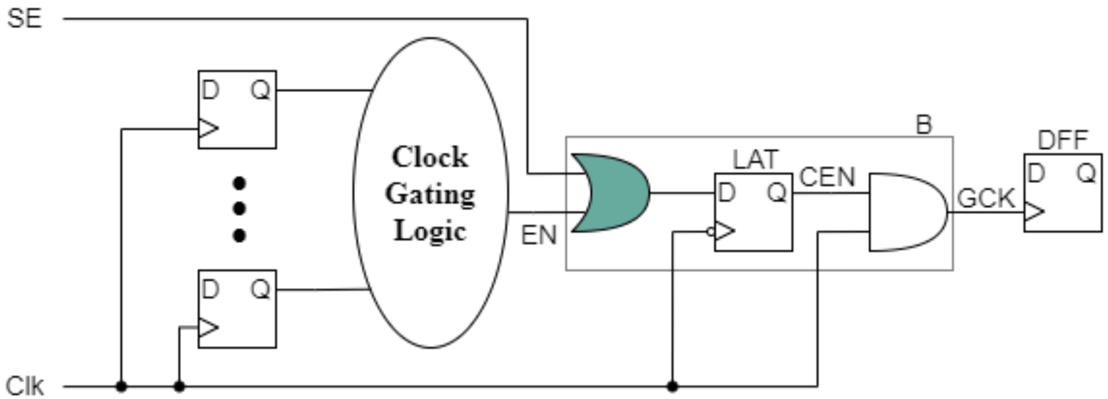


No hazzard

Cope With Enable Signal Noise(2)



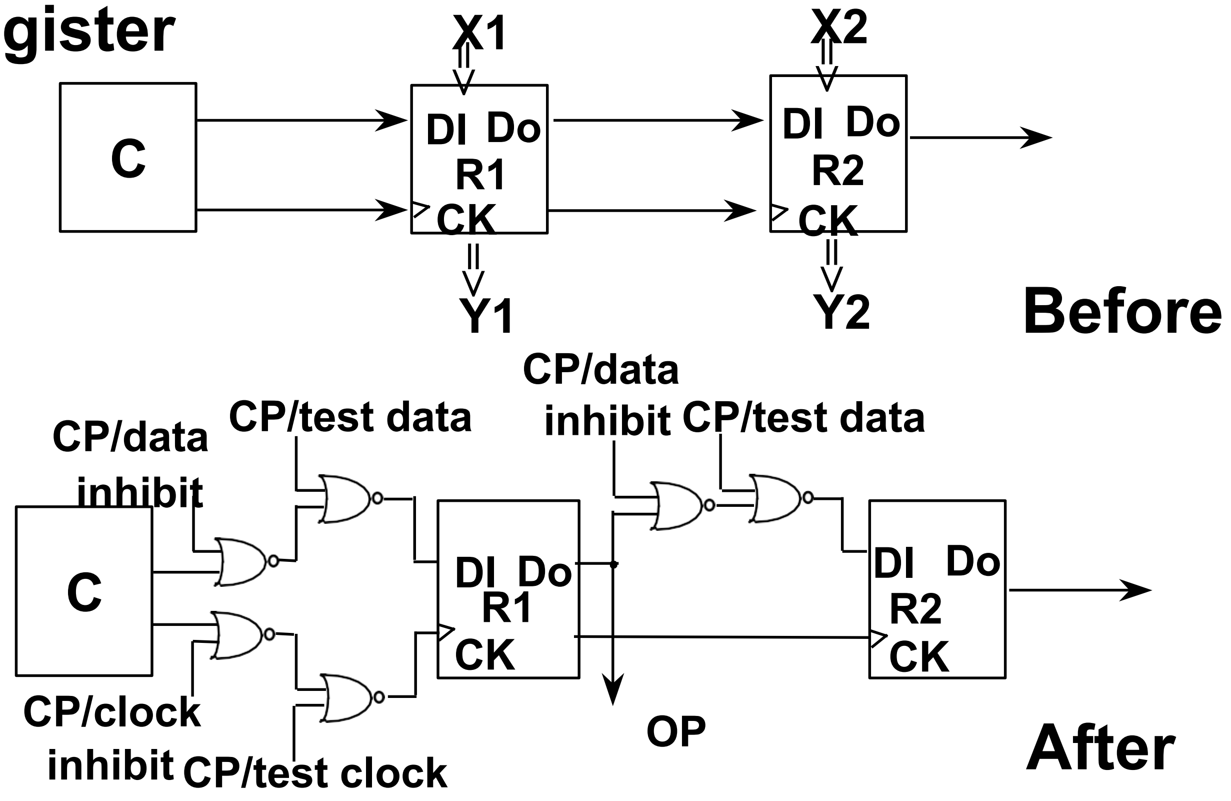
Original Design



Testable design

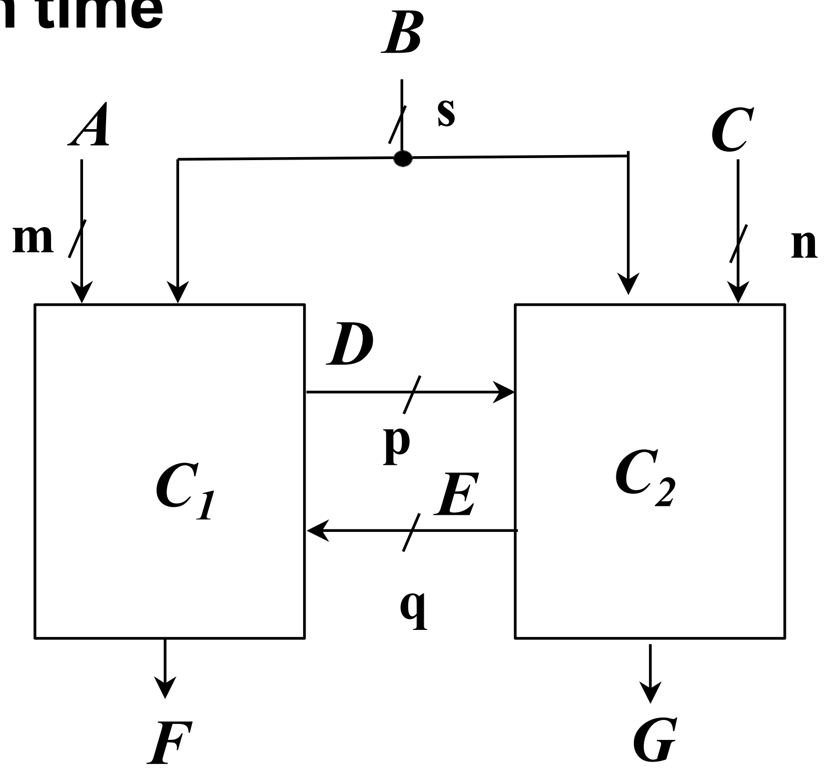
Partitioning Counters and Shift Registers

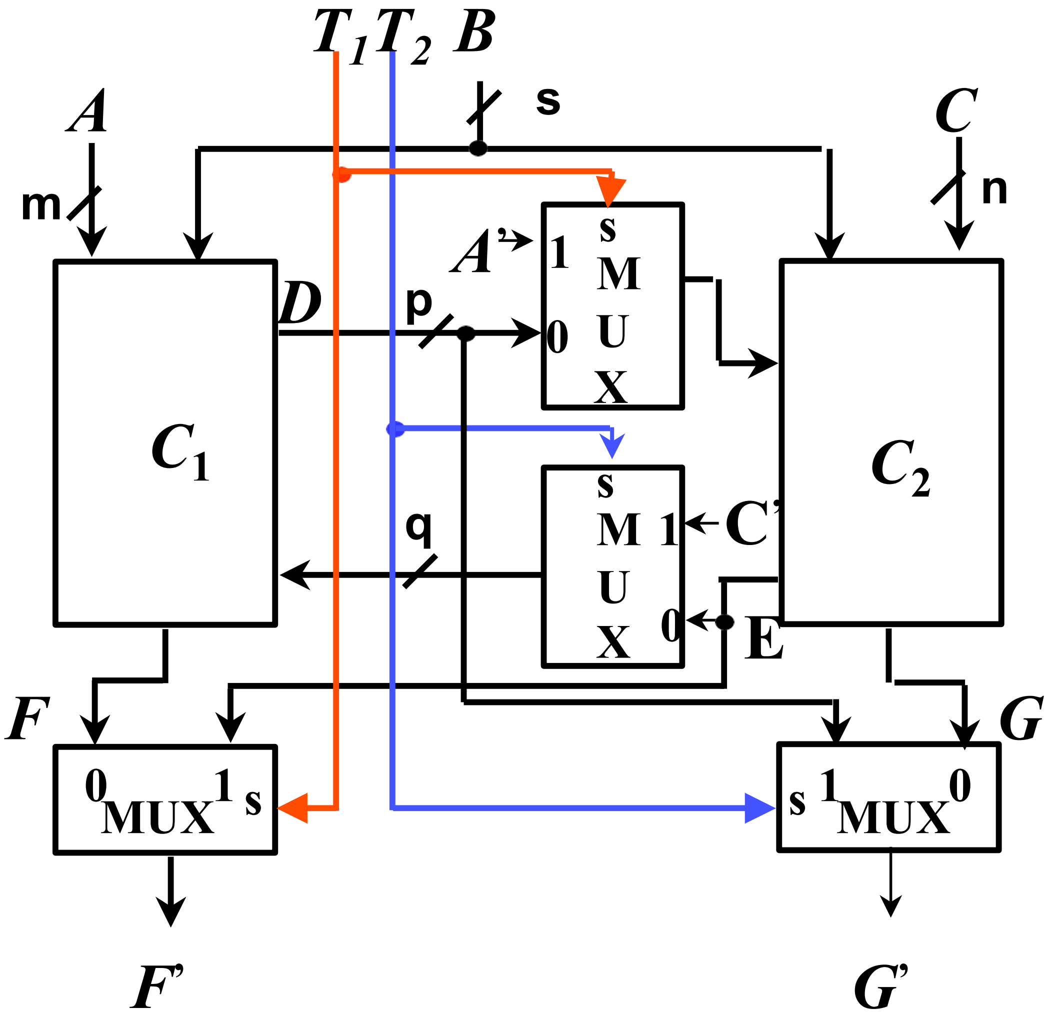
- Rule: Partition into small units
- Ex: Register



Partition of Large Combinational Circuits

- Rule : To reduce test generation costs and/or test application time





T_1	T_2	Mode
0	0	normal
0	1	test C_1
1	0	test C_2

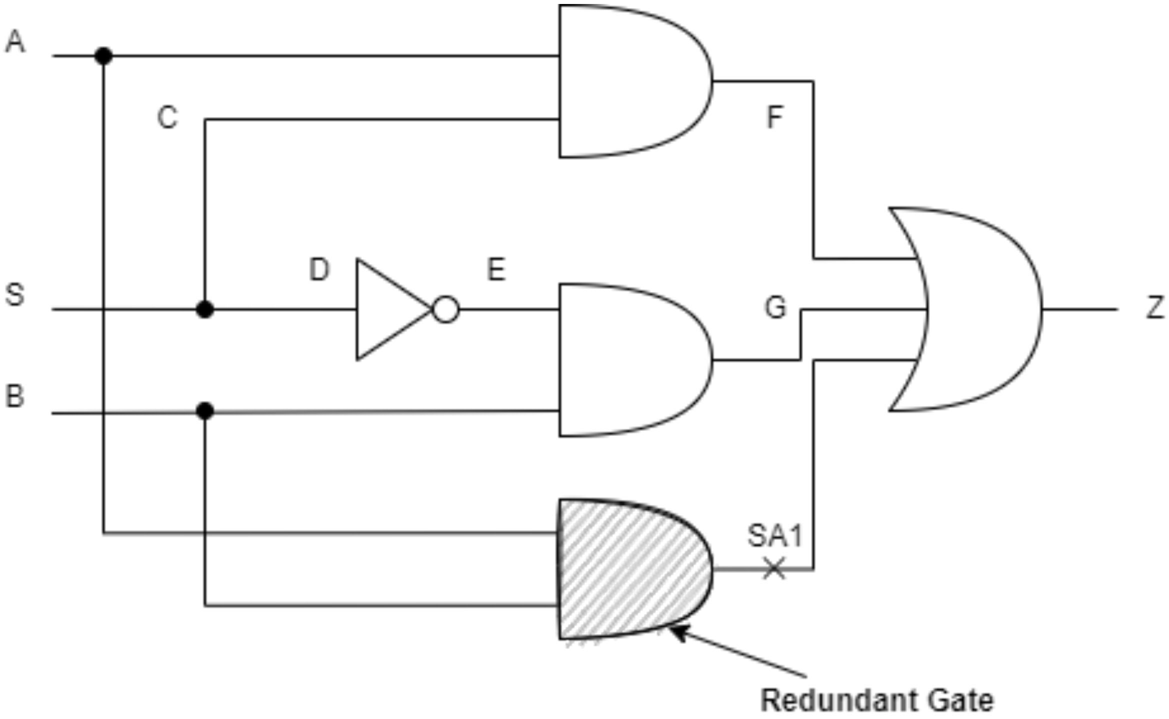
If $2^{p+n} + 2^{q+m} < 2^{m+n}$ then test time can be reduced

Logic Redundancy

Rule: Avoid or eliminate redundancy ckt.

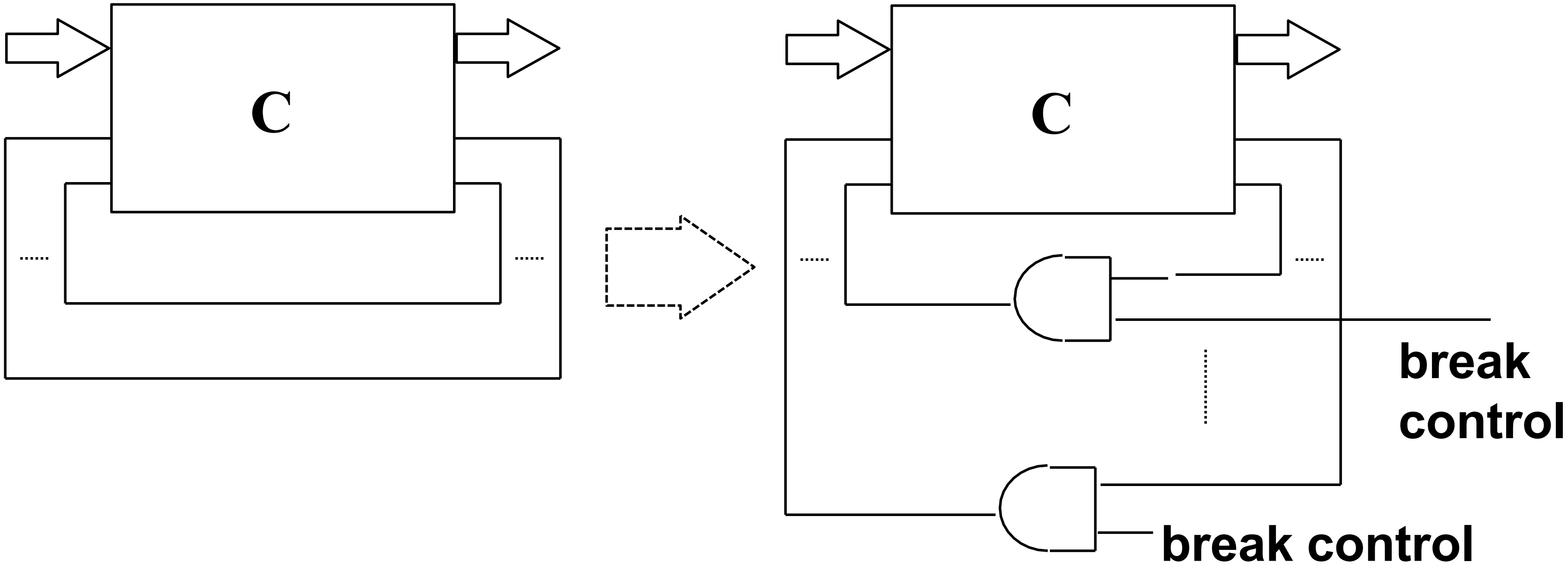
- Design errors
- Undetectable faults
- Invalidation of some tests
- Bias fault coverage

Example

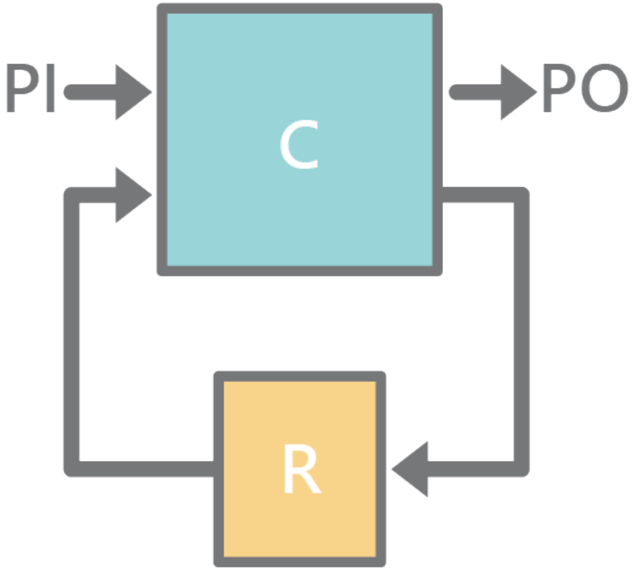


Global Feedback Paths

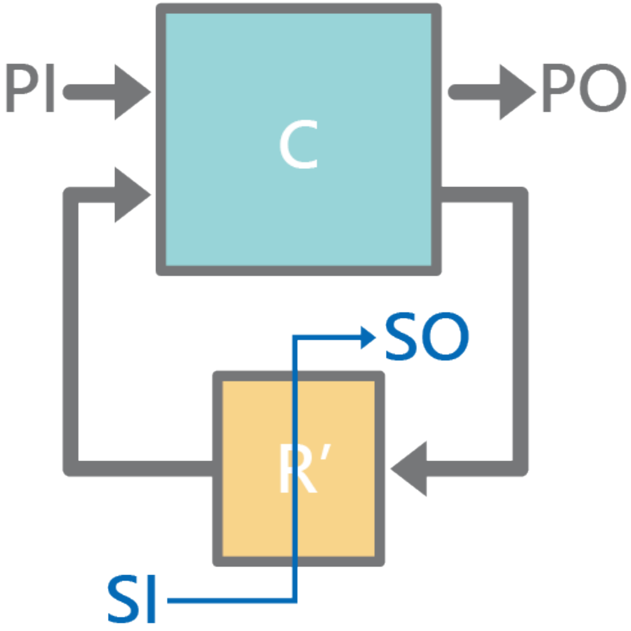
Rule: break global feedback



Scan System



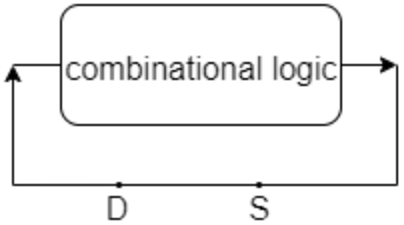
Original design



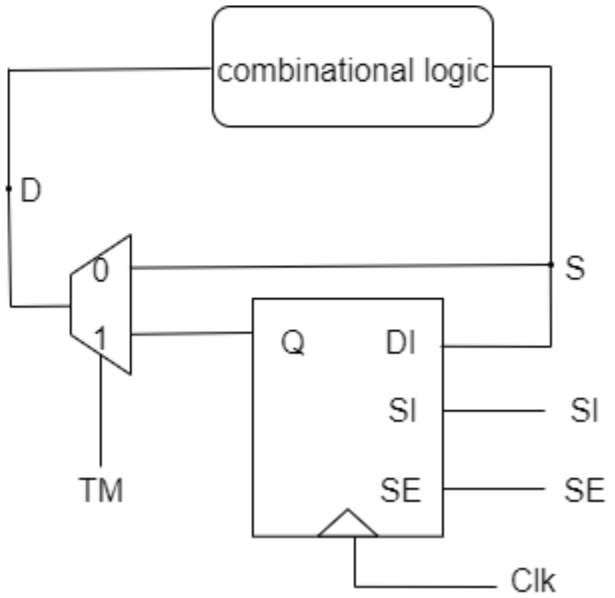
SI

Modified circuit

Scan System



Original Design



Testable Design

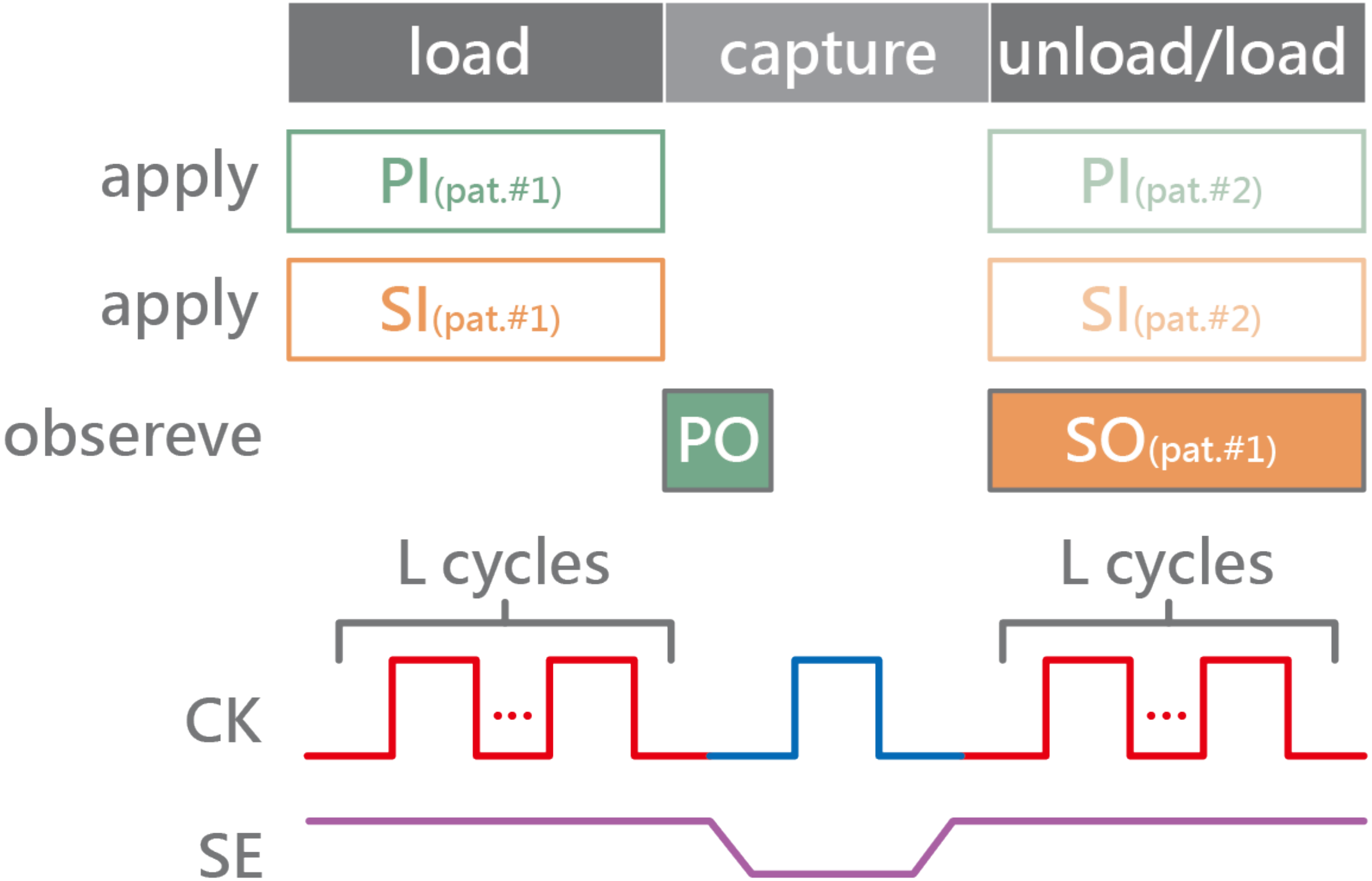
Original design

Modified circuit

Scan System

- Long test time
- Large test data
- Too much overhead
- How to test DFT circuit itself

Scan Operation

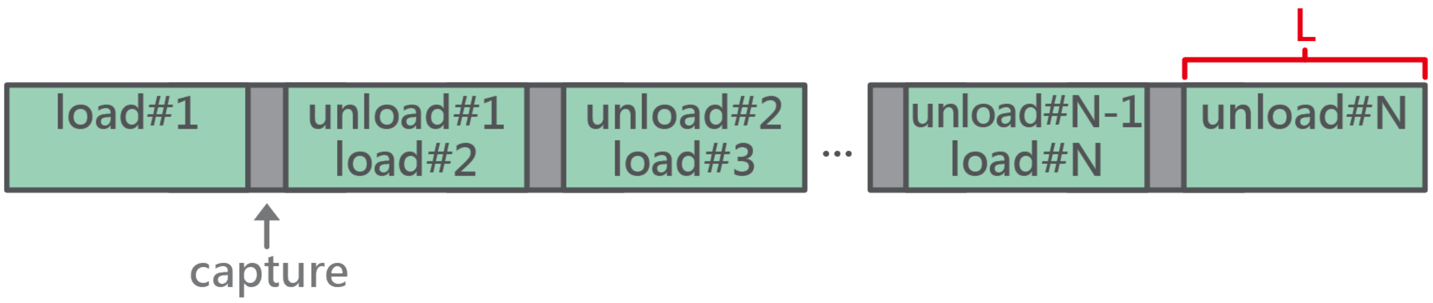


Scan System Performance

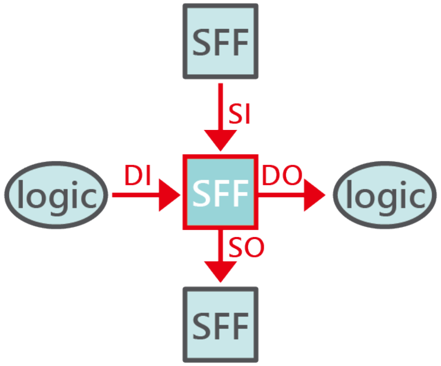
- How to calculate the test cycle:
  - L: Length of scan chain, Npattern: Number of test patterns

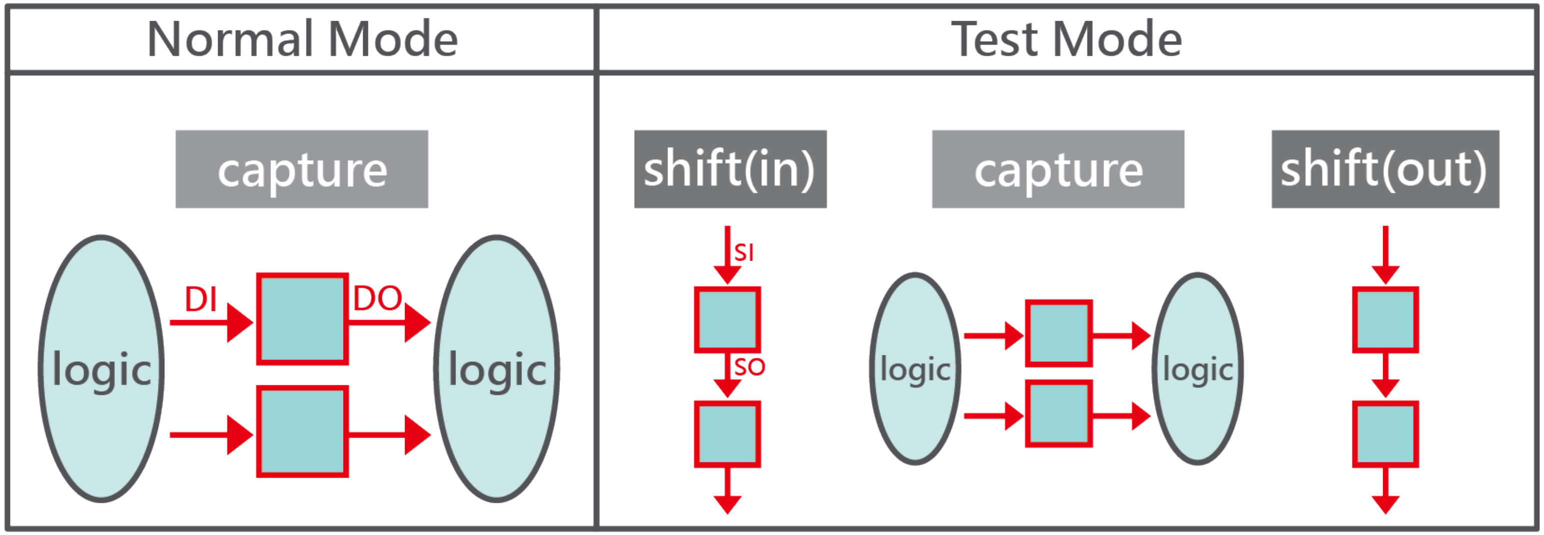
$$\underbrace{(N_{pattern}+1) \times L}_{\text{shift}} + \underbrace{N_{pattern}}_{\text{capture}}$$

- Example: Apply 5000 test patterns to a CUT with 10000 SFF
  - Total cycles = 50015000 cycles
    (5000+1)x10000 shift cycles + 5000 capture cycles

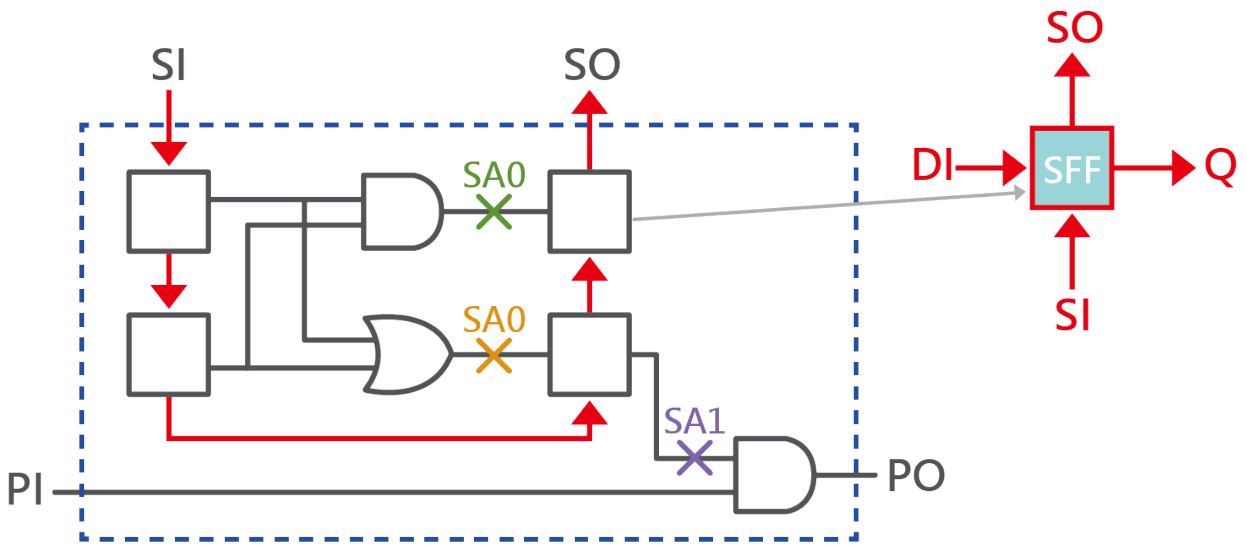


Example


SFF
SI
logic
DI
SFF
DO
logic
SO
SFF


Normal Mode
Test Mode
capture
logic
DI
DO
logic
shift(in)
SI
SO
capture
logic
logic
shift(out)

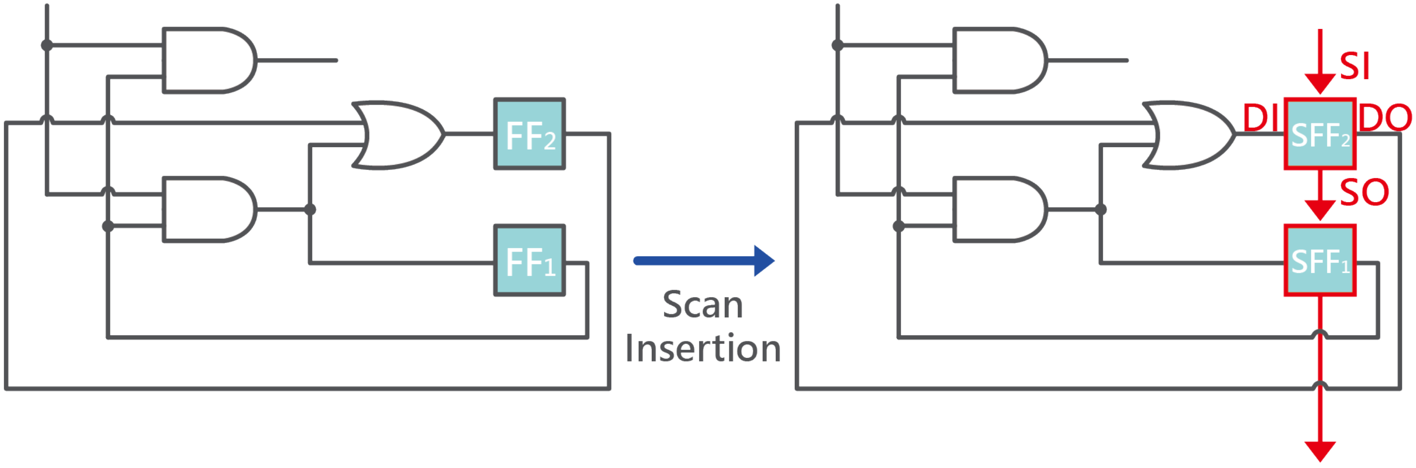
Example



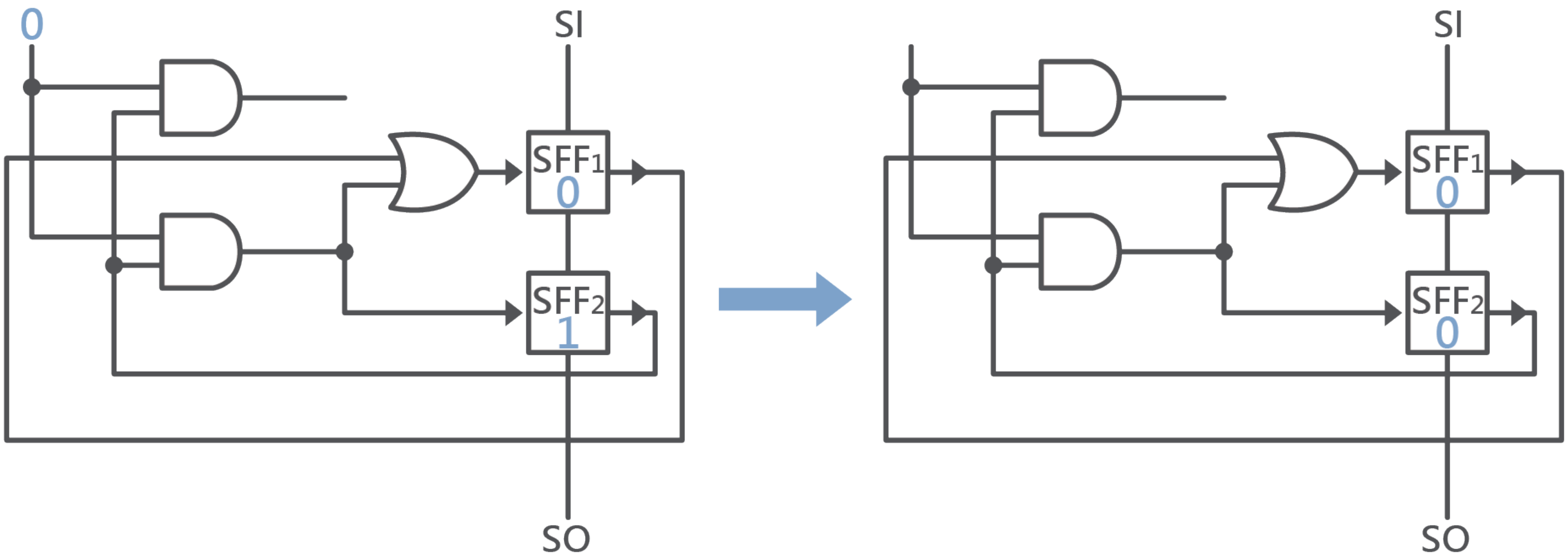
load		capture		unload
CK=PPPP		CK=P		CK=PPPP
SE=1111	SE 1→0	SE=0	SE 0→1	SE=1111
SI=X011				SO=HHXX
PI=1	PO=L			

Scan DFT Overhead

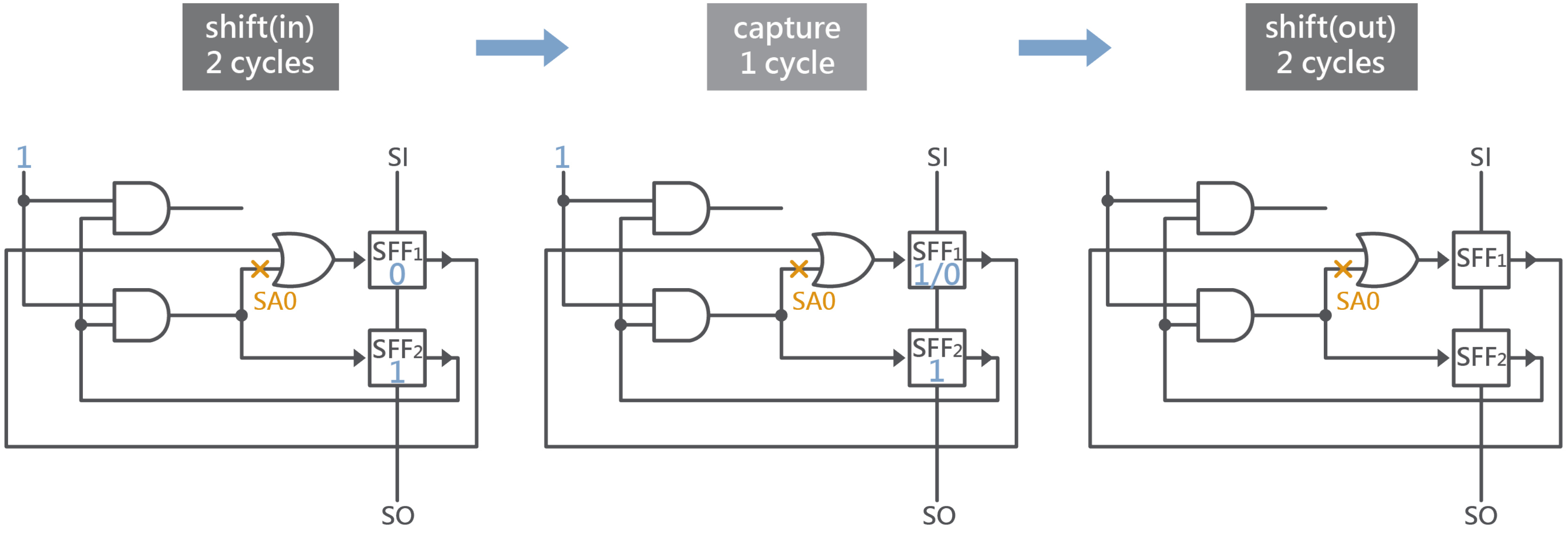
- Performance Degradation
- Design overhead
- Hardware overhead
- Yield loss
- Power overhead



Example-Normal Mode



Example-Test Mode



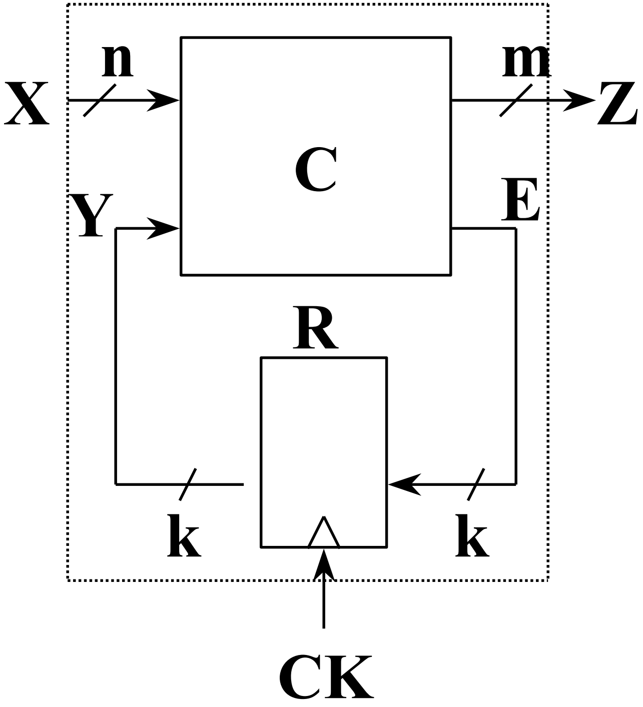
shift(in)
2 cycles

capture
1 cycle

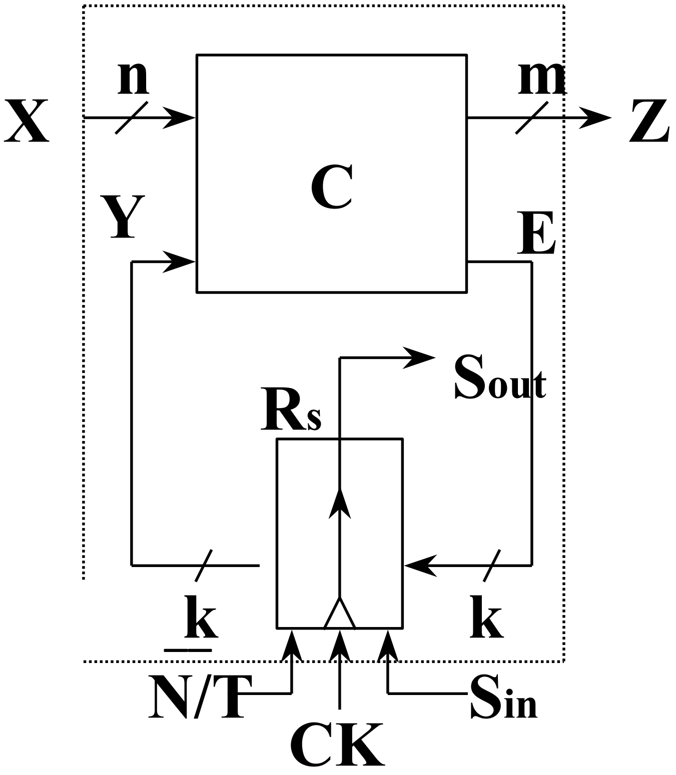
shift(out)
2 cycles

1/0 is observed at 2nd cycle

Full Serial Integrated Scan



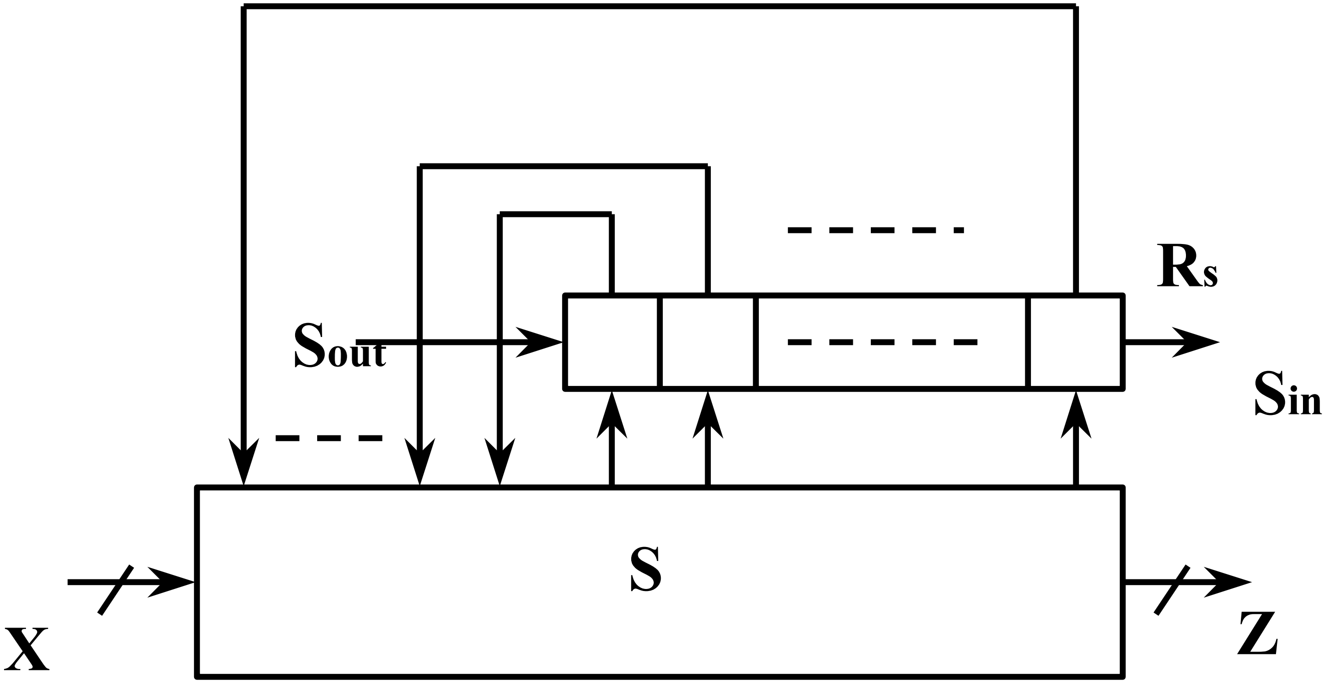
Normal



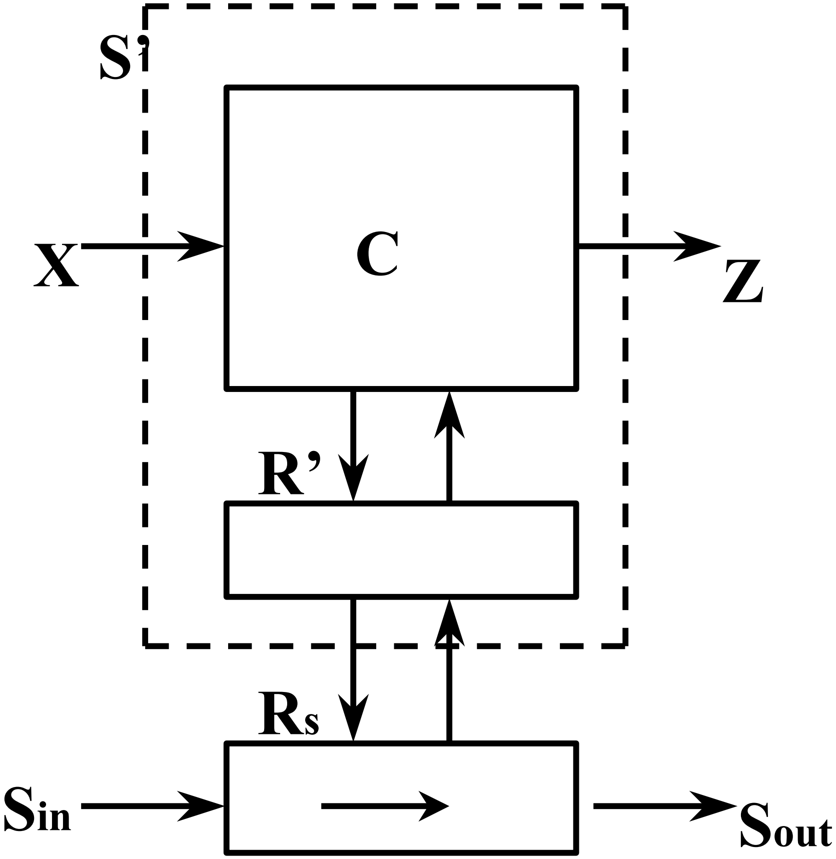
Scanned

Sequential ATPG → Combinational ATPG

Isolated Serial Scan (Scan/set)

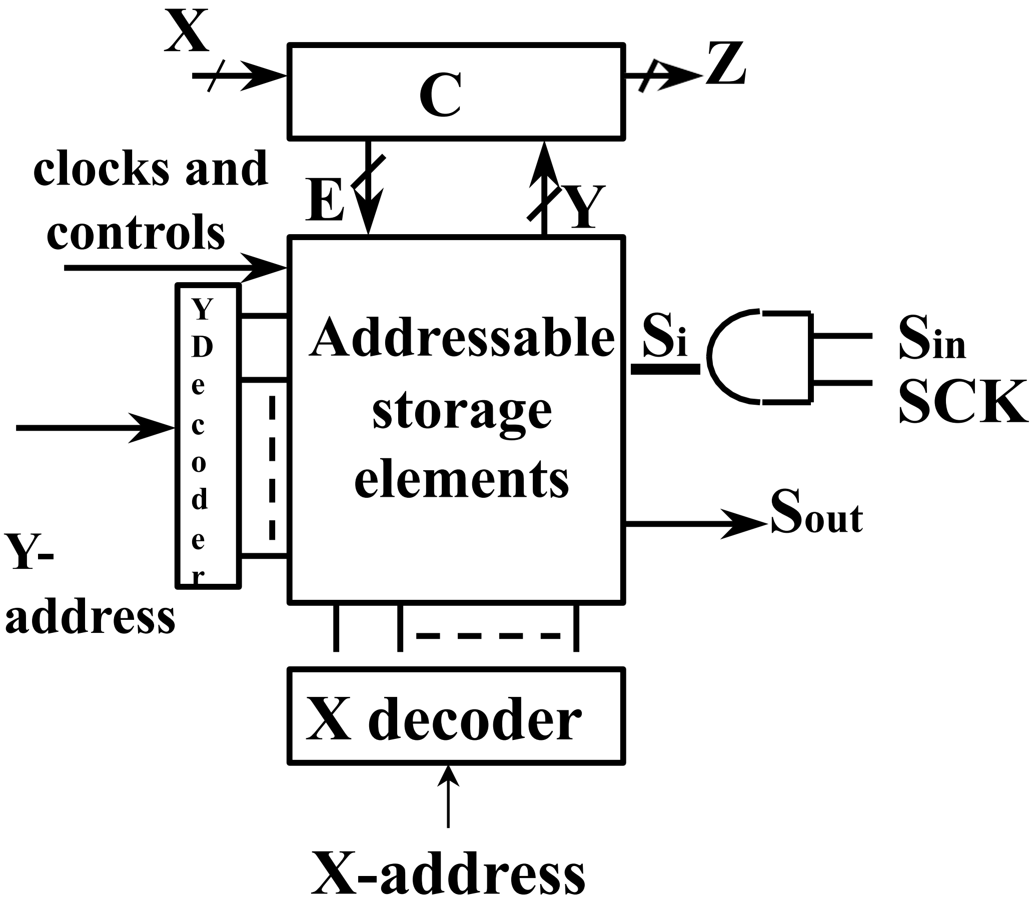


Full Isolated Scan (Structured)



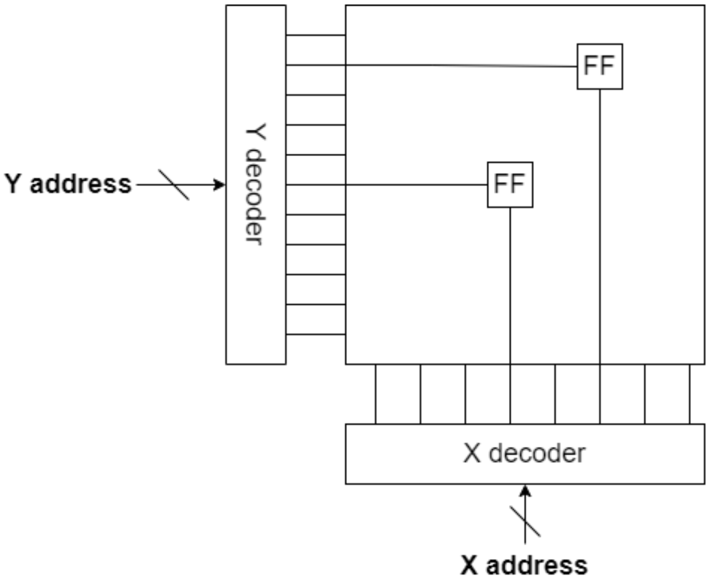
- Shadow register
- Real-time test support
- snapshot

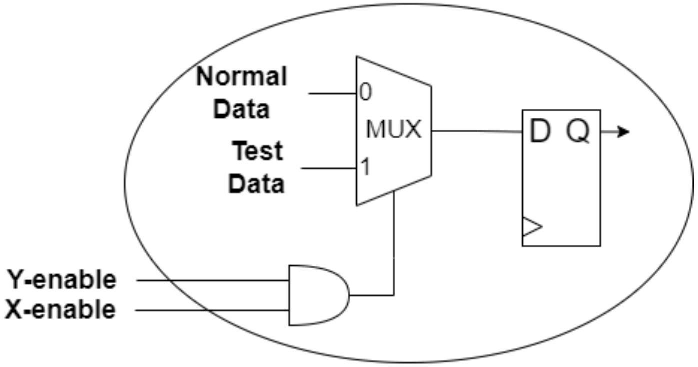
Random-Access Scan (Non serial-structured)



- High area overhead
- Faster test application: only bit change
- Concept of crosscheck

Random-Access Scan (Non serial-structured)





Scan Cell Design

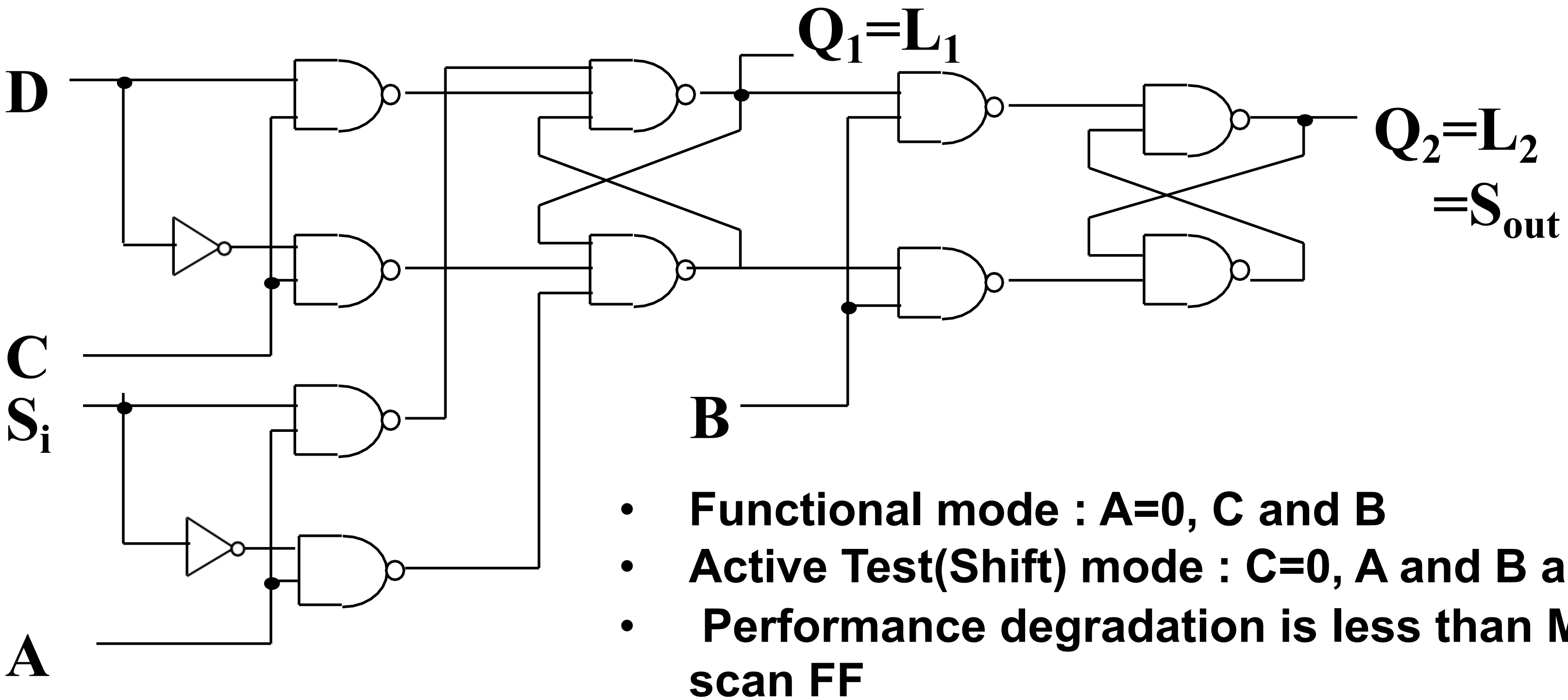
- Static / Dynamic
- Single / Double stages
- Latch / Flip-flop (Clocking Scheme)

Usually Two Operation Modes

- Functional mode
- Shift mode

IBM LSSD Scan Cell

- Gate Level



- Functional mode : A=0, C and B
- Active Test(Shift) mode : C=0, A and B active
- Performance degradation is less than Mux-scan FF
- Non-overlap clock scheme

IBM LSSD Scan Cell

Advantages

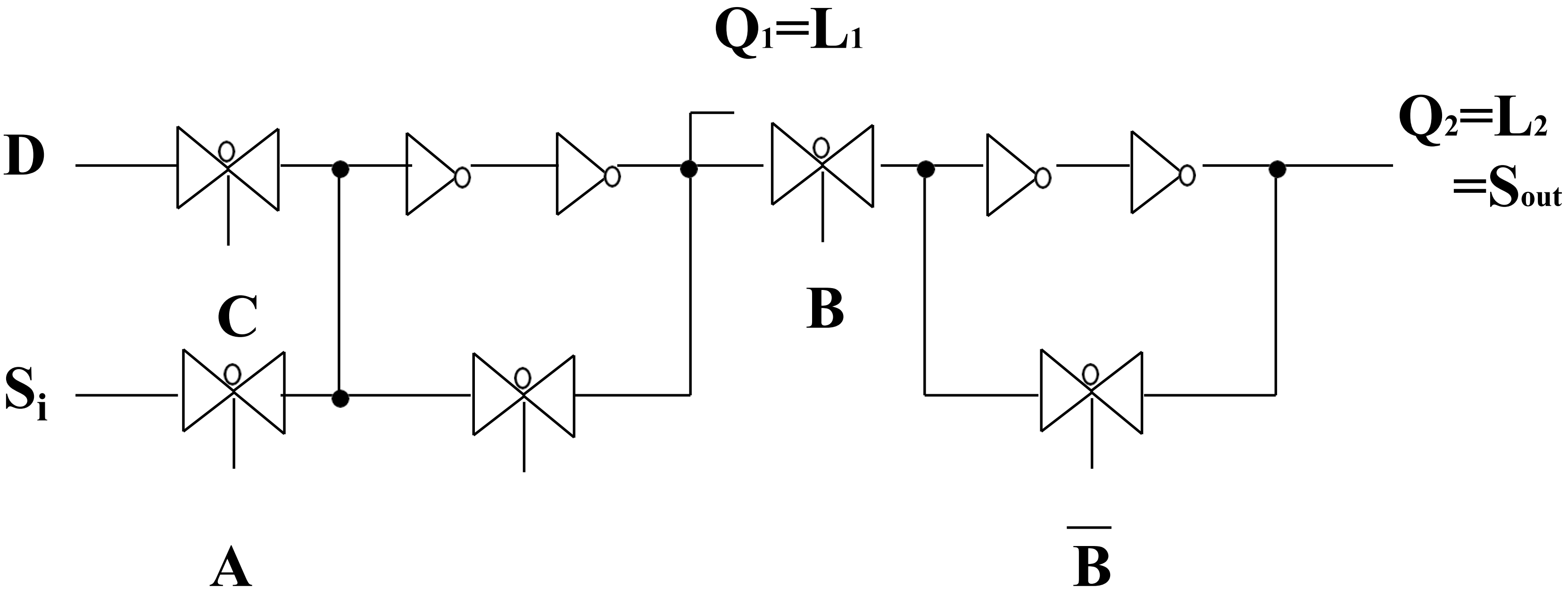
- It allows us to insert scan into a latch-based design.
- LSSD is guaranteed to be race free, which not case for the Muxed-D and Clocked-scan design.

Disadvantages

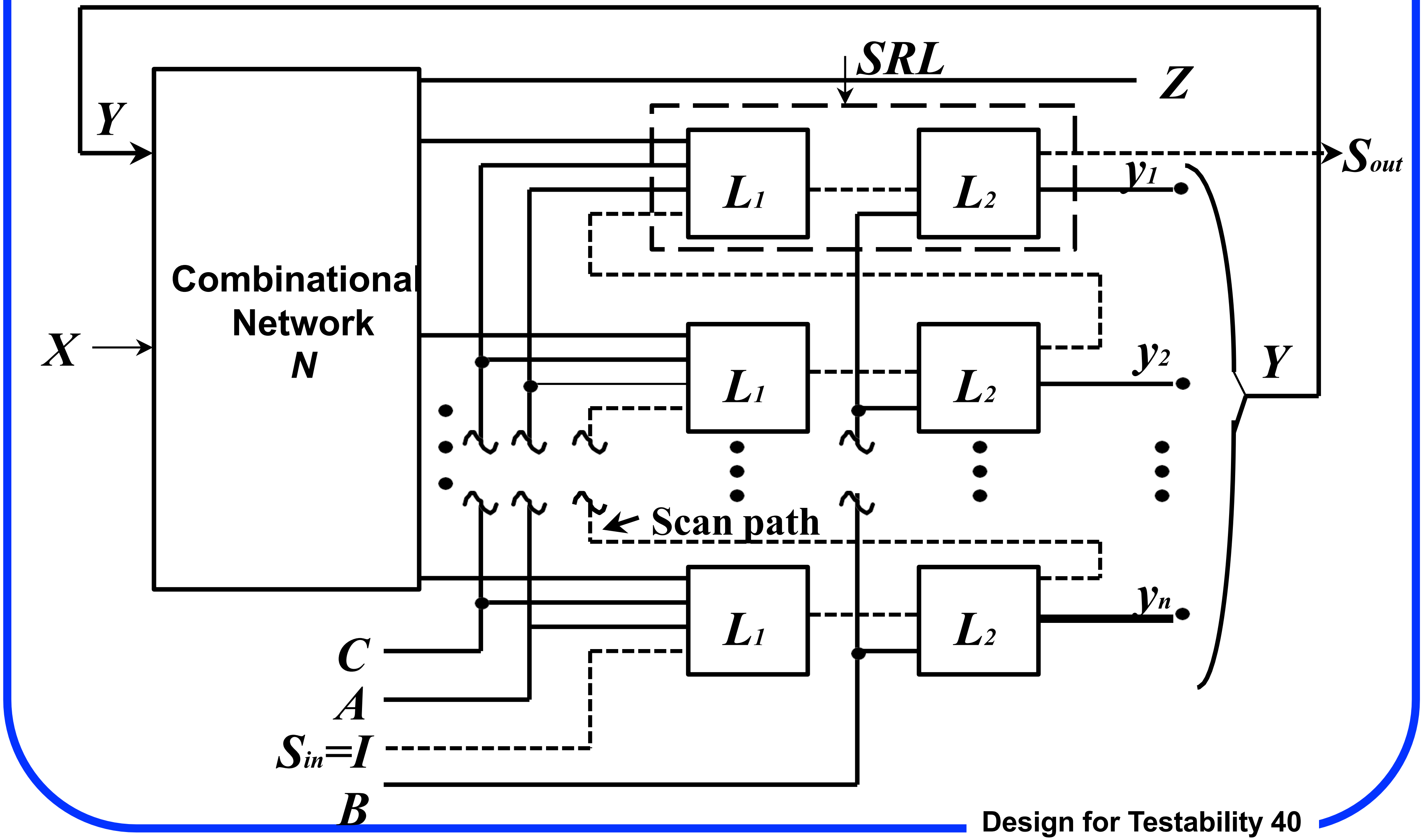
- It requires routing for the additional clocks, which increases routing complexity.

IBM LSSD Scan Cell (Cont.)

- **Switch / Inverter level**

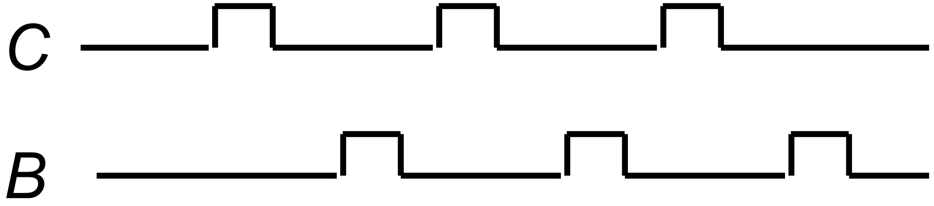


LSSD Double-Latch Design

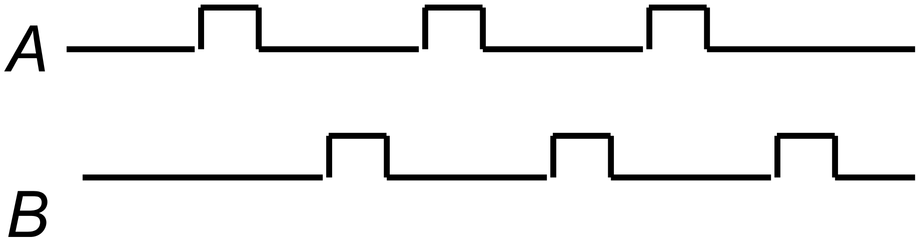


Clocking Scheme of LSSD Double-Latch

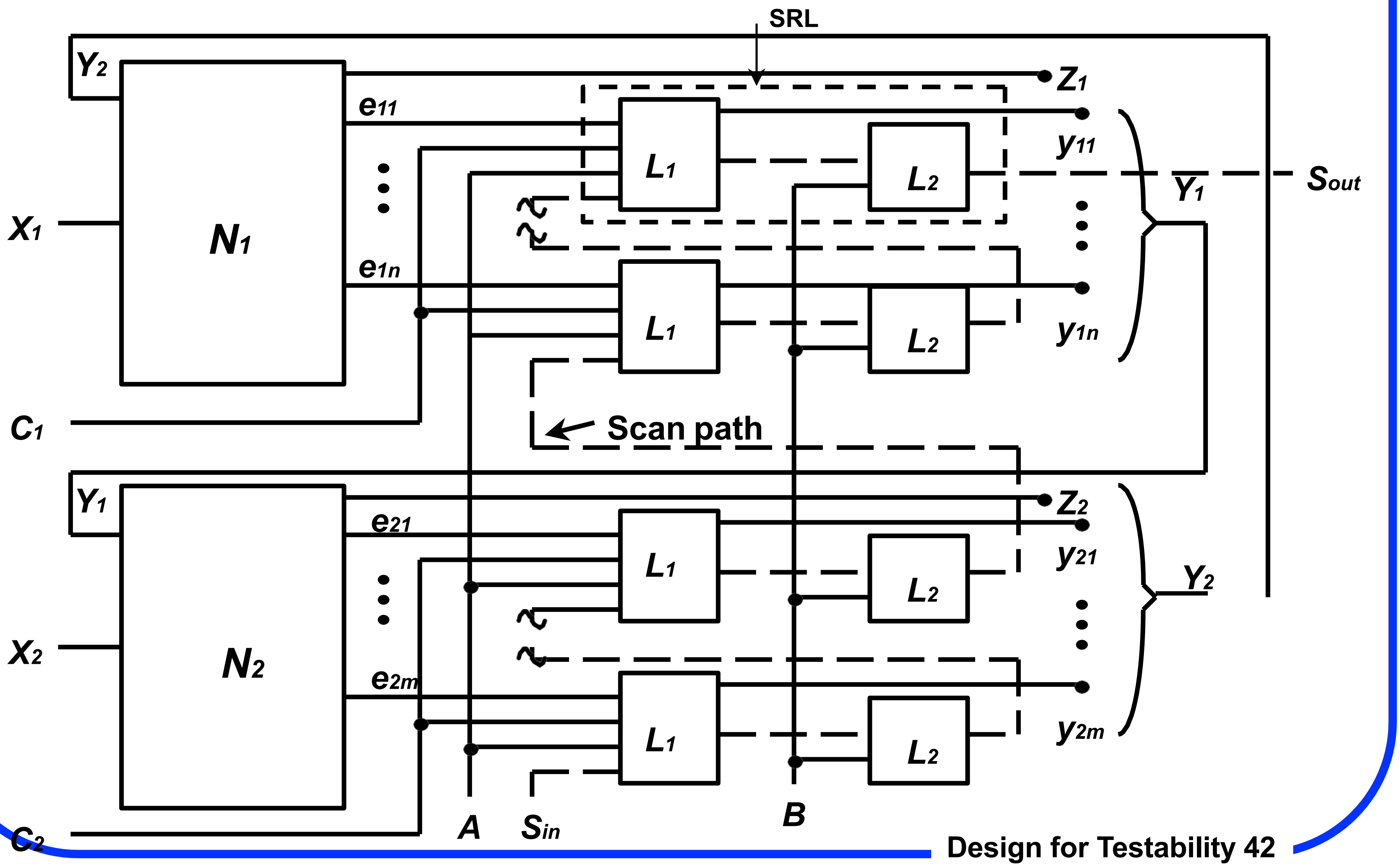
Normal mode:



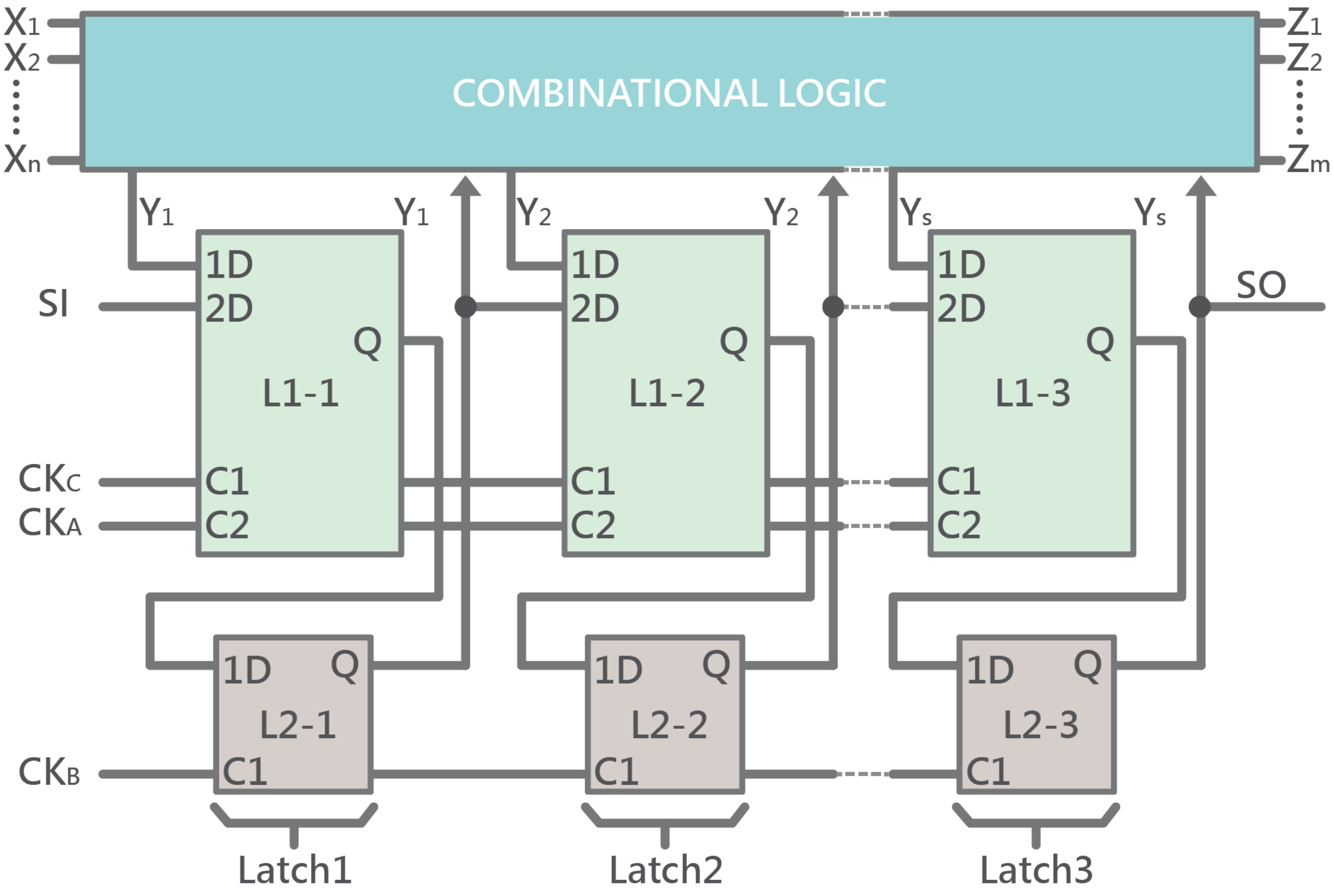
Test mode:



LSSD Single-Latch Design



Example

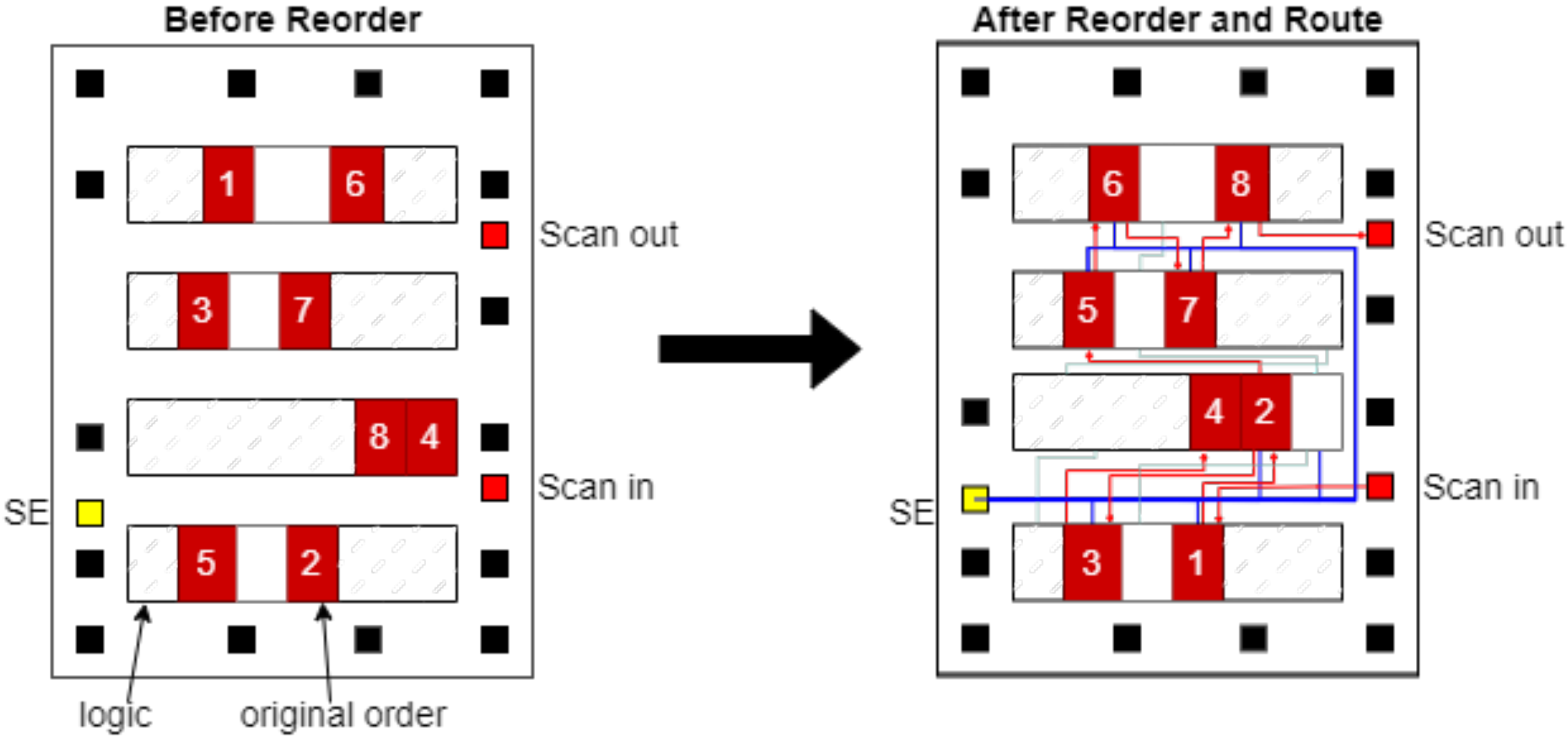


COMBINATIONAL LOGIC
X1
X2
Xn
Z1
Z2
Zm
Y1
Y1
Y2
Y2
Ys
Ys
SI
SO
CKC
CKA
CKB
1D
2D
Q
L1-1
C1
C2
1D
2D
Q
L1-2
C1
C2
1D
2D
Q
L1-3
C1
C2
1D
Q
L2-1
C1
1D
Q
L2-2
C1
1D
Q
L2-3
C1
Latch1
Latch2
Latch3

Scan Design Costs

- Area overhead
- Possible performance degradation
- Extra pins
- High test time
- Extra clock control

More Area OverHead

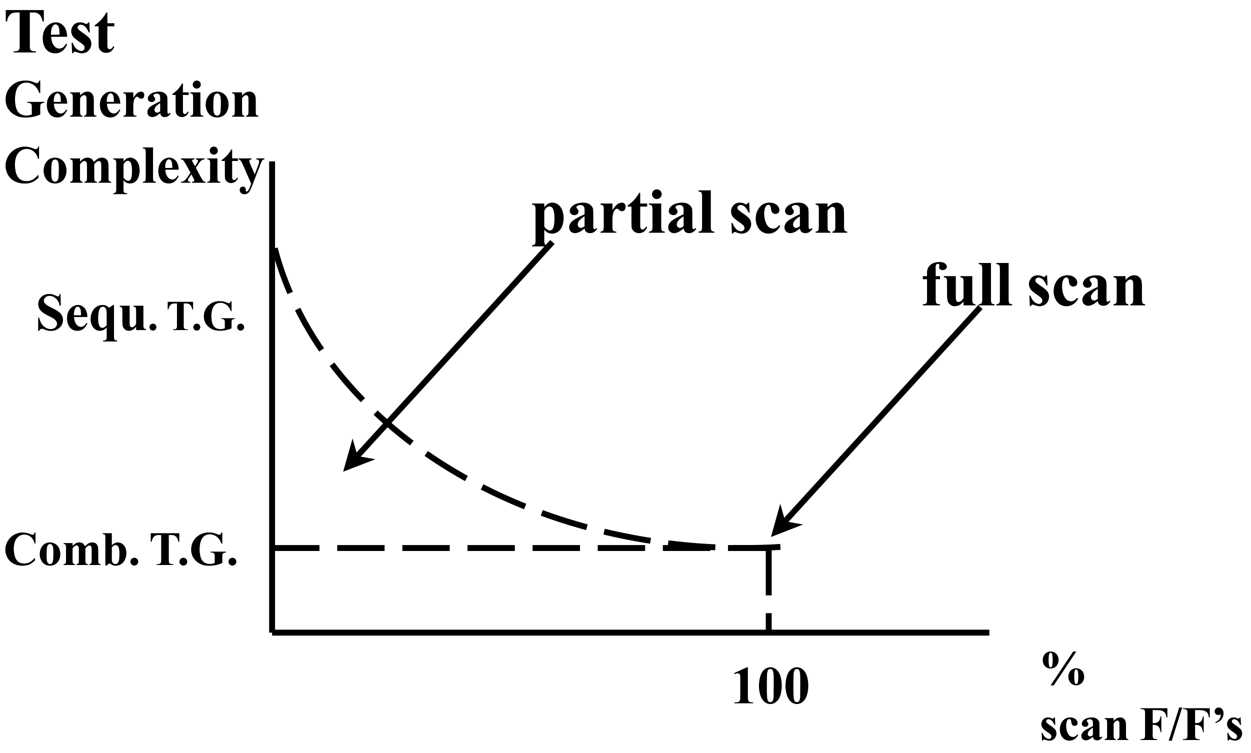


Advanced Scan Concepts

- Partial scan (P.S.)
- Multiple test session (M.T.S.)
- Multiple scan chains (M.S.C.)
- Broadcast scan chains (B.S.C)

Method	P.S.	M.T.S.	M.S.C.	B.S.C.
Area overhead	↓	same	same or ↑	same
Performance Degradation	↓	same	same	same
Extra pins	same	same	same or ↑	↓
Extra clock control	same	same	same	same
Test application time	↓ or ↑	↓	↓	↓

Partial Scan: Only a subset of all flip-flops are scanned

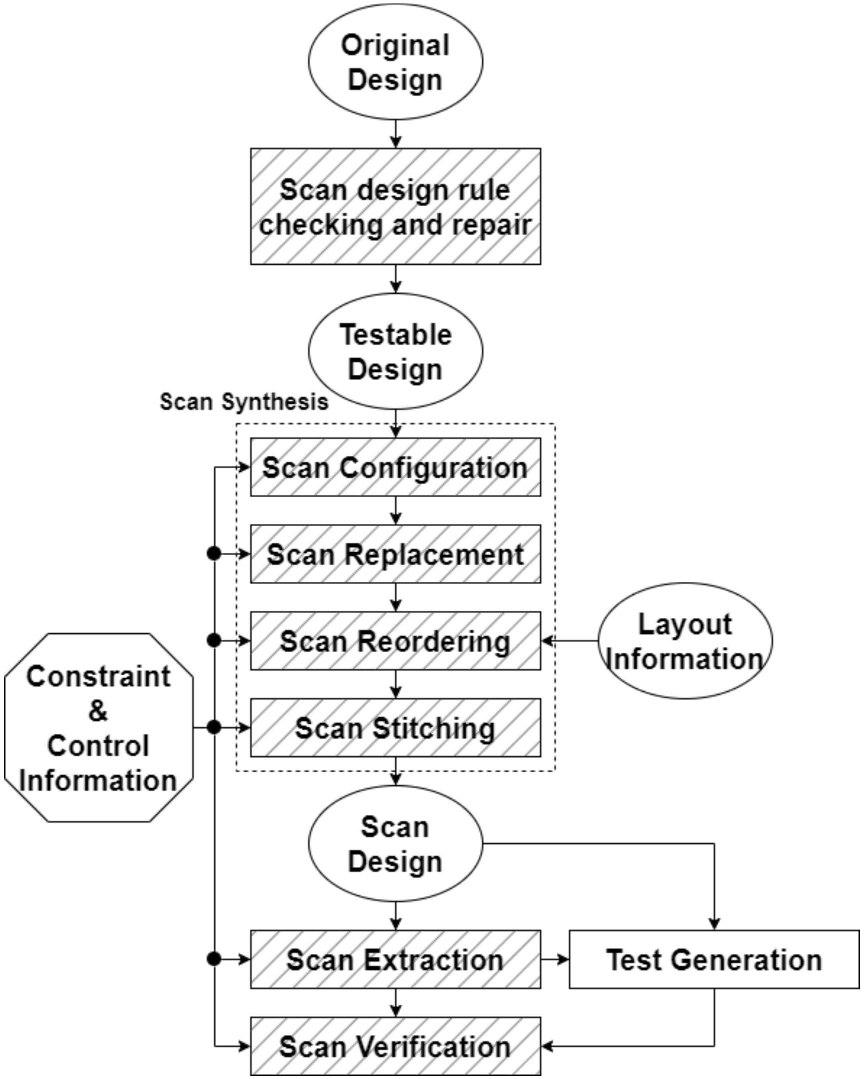


- Trade-off between
  - –Area overhead
  - –Test generation complexity

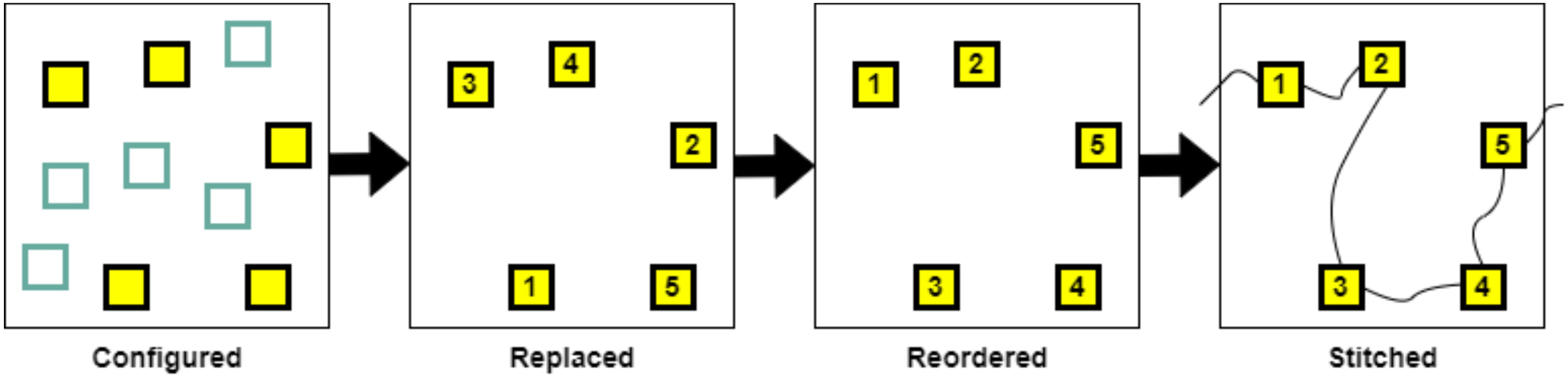
Partial Scan and Full Scan

- Full scan
  - -- Every flip-flop is scanable
  - -- More hardware overhead
  - -- Higher fault coverage
  - -- Longer test time
  - -- Shorter ATPG time required
- Partial scan
  - -- Not every flip-flop is scanable
  - -- Less hardware overhead
  - -- Less fault coverage
  - -- Shorter test time
  - -- Longer ATPG time required

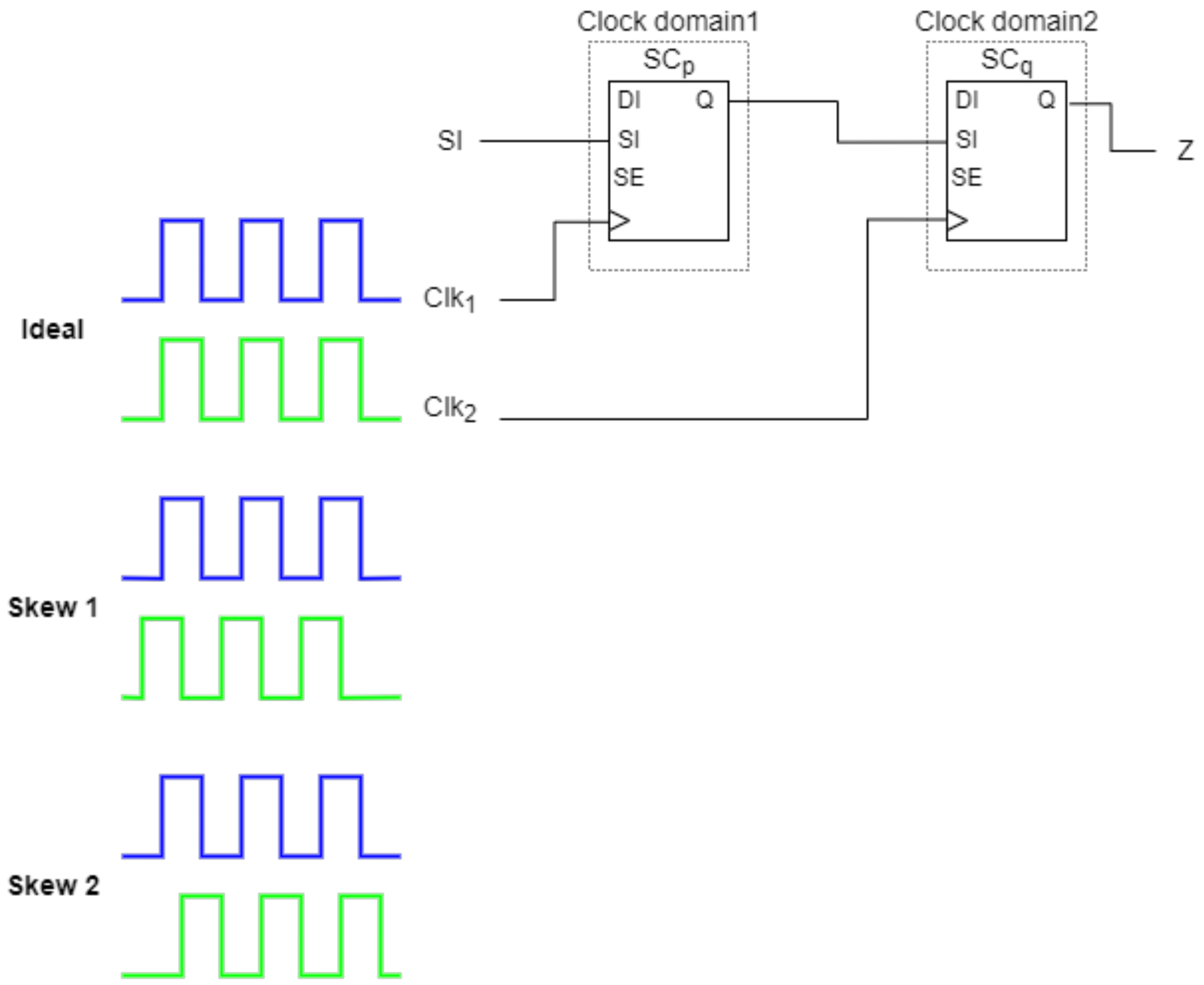
Scan Design Flow



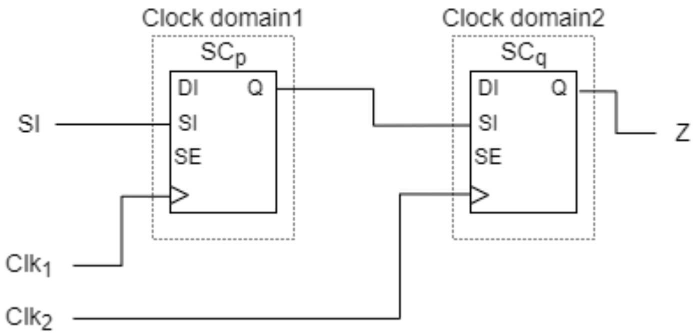
Scan Register Selection

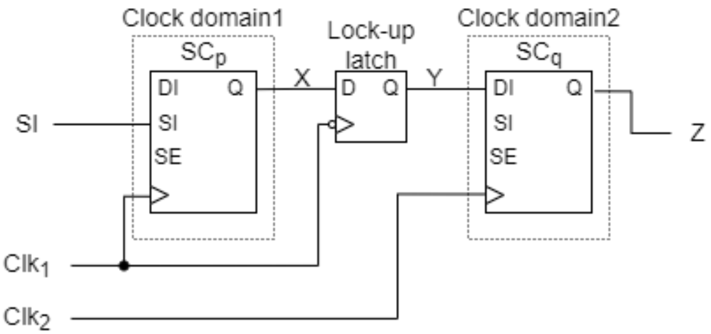


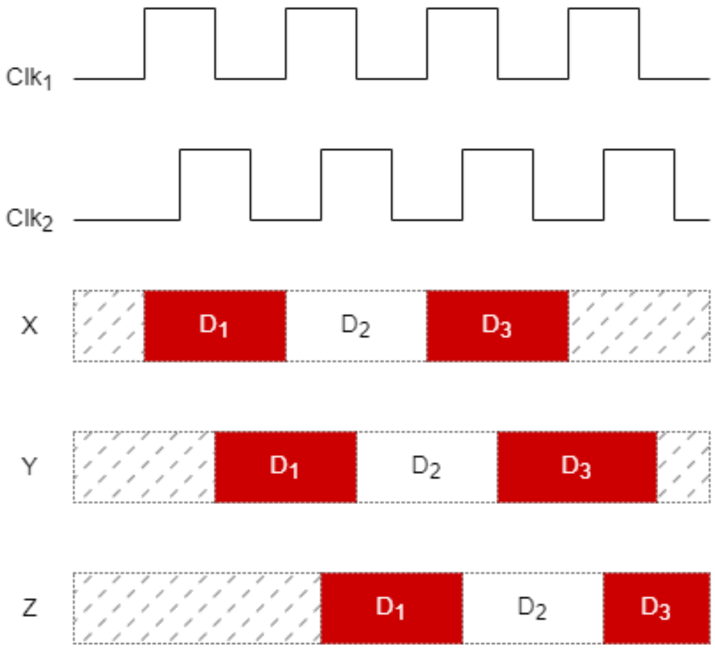
Clock Skew Between Two Clock Domains



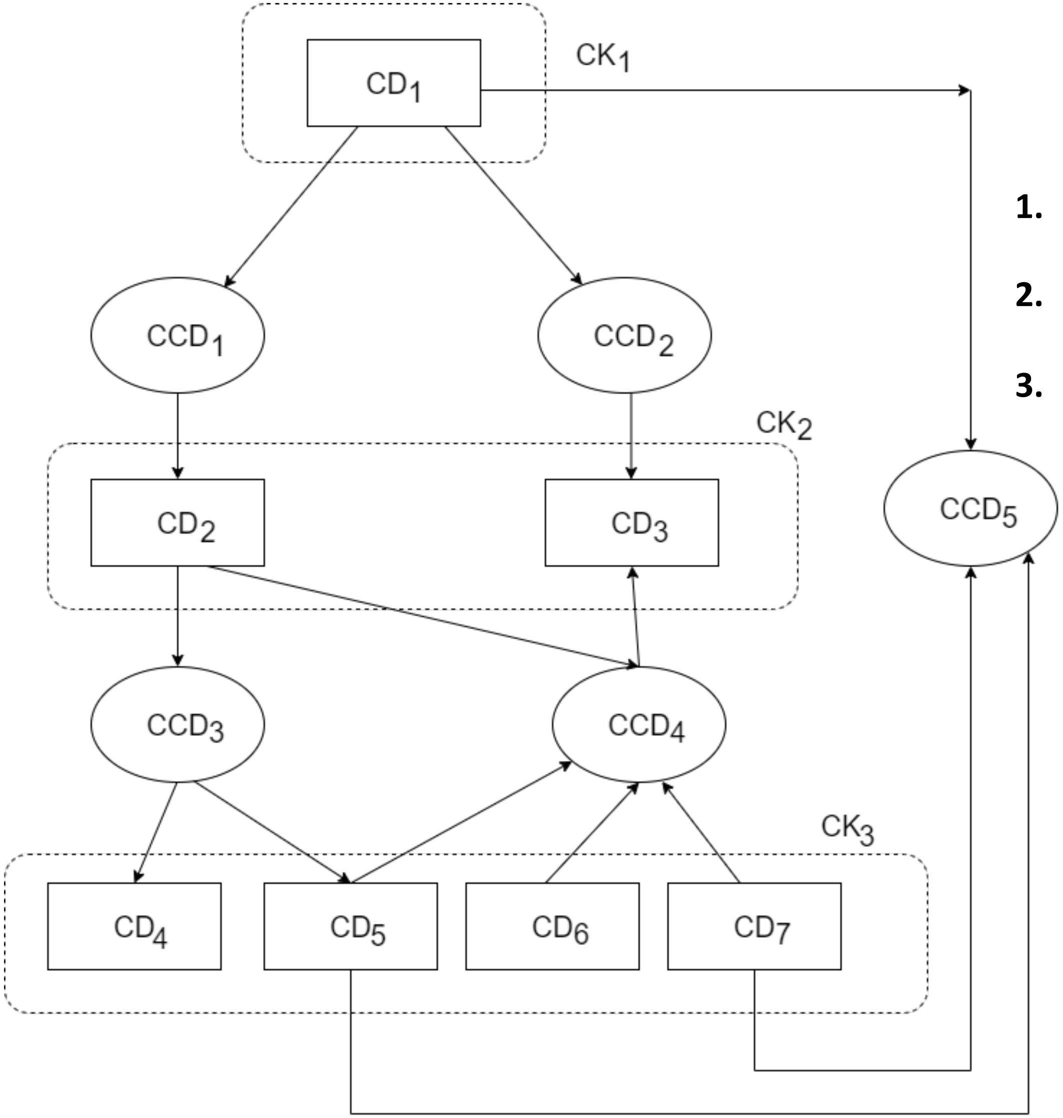
Lock-up Latch







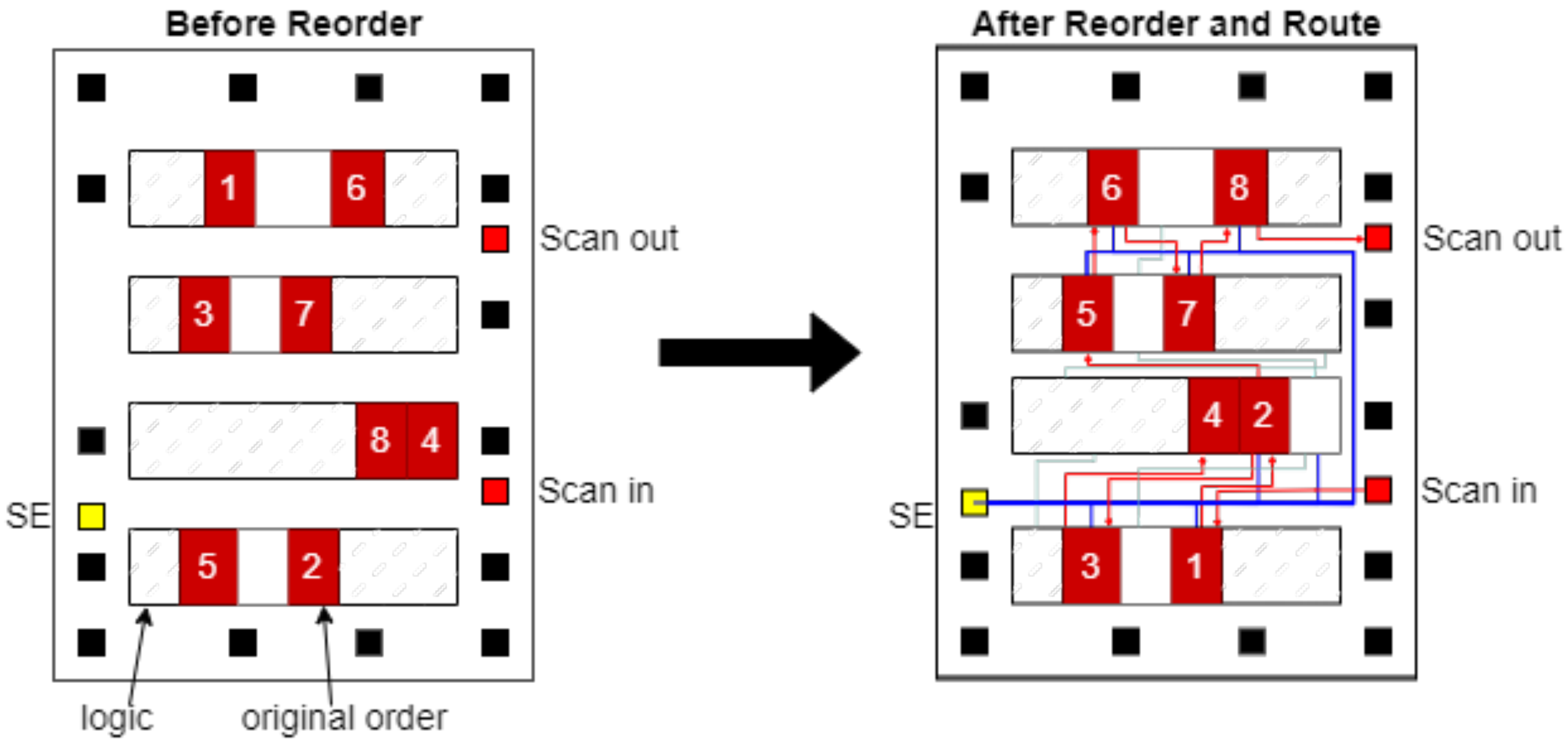
Clock Group



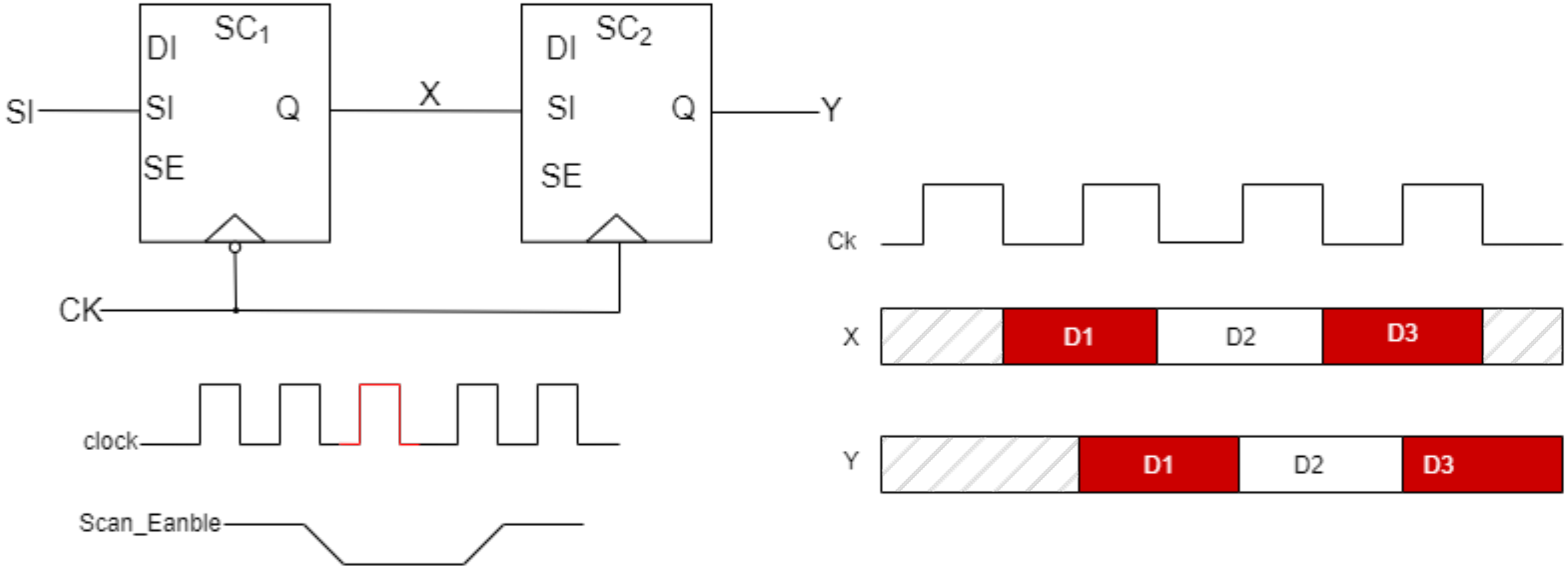
1. **If there is clock skew between CD1 and CD2, then CD1 and CD2 can not form a scan chain.**
2. **CD2 and CD3 can form a scan chain if there is no interaction among them.**
3. **Three clock group required in this design**

CD: Clock Domain
CCD: Cross Clock Domain
CK: Clock Group

Reorder and Route

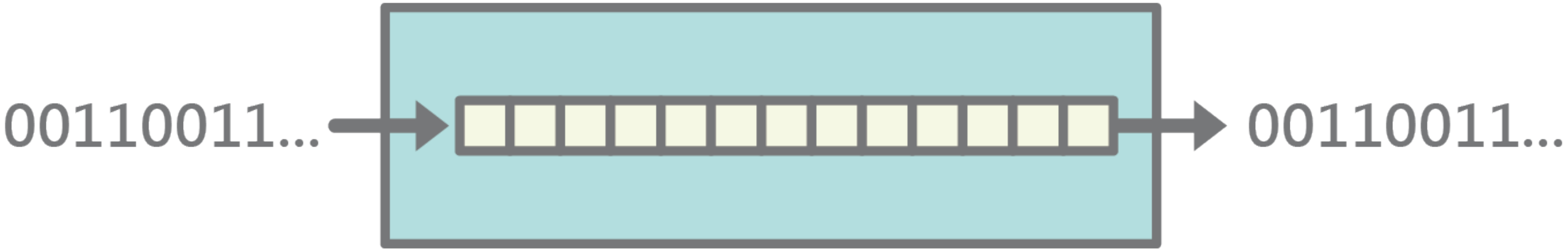


Negative/Positive edge



Scan Chain Integrity Test

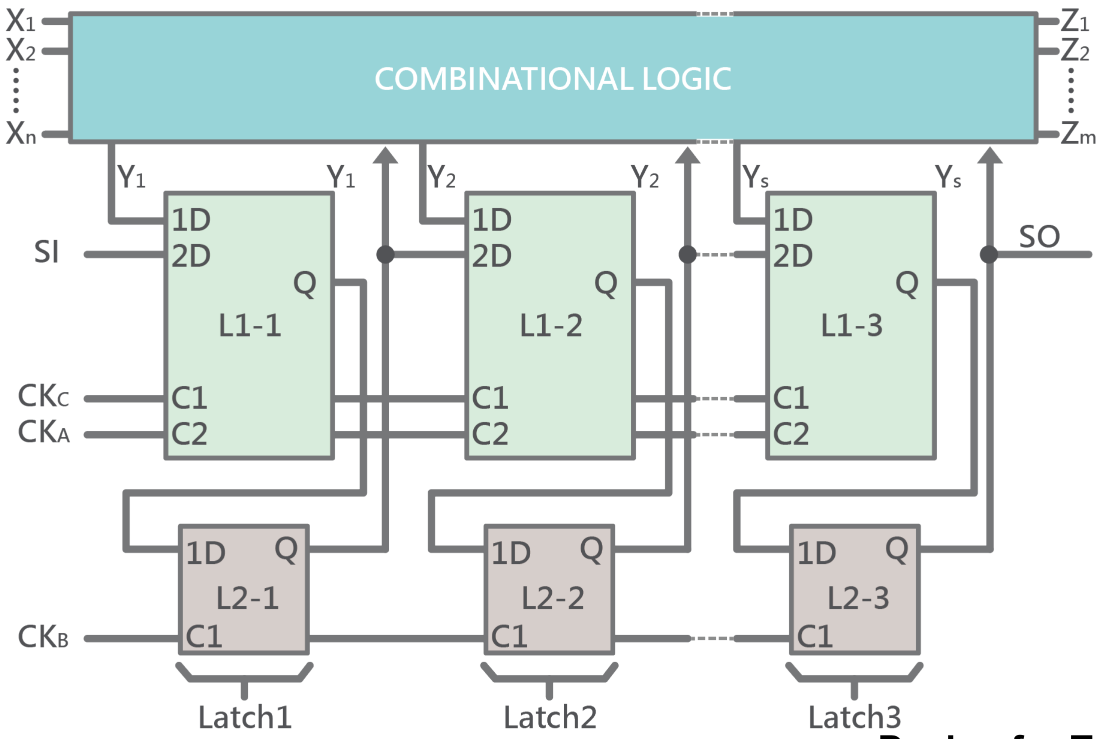
- Scan chain requires to test itself
- Sequence 00110011…is shifted in and shifted out
- There is no capture operation
- Each scan FF undergo four transitions 0→0, 0→1, 1→0, 1→1
- Detect all stuck-at faults and transition faults in the scan chain



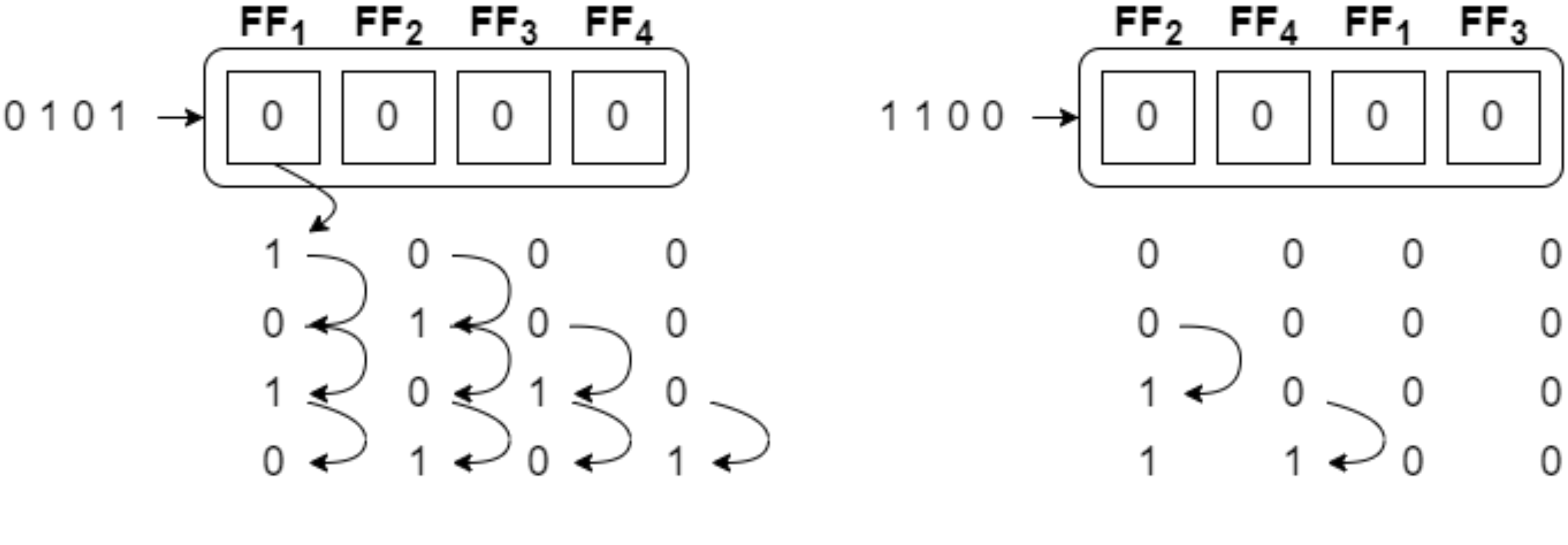
$$\underbrace{(N_{pattern}+1) \times L}_{\text{shift}} + \underbrace{N_{pattern}}_{\text{capture}} + \underbrace{2L}_{\text{scan integrity}}$$

Scan Chain Flush Test

- LSSD Scan chain requires to test itself
- Apply CK=0 , CA=1 and CB=1 to form the path from SI to SO
  -- Apply SI=0→1 at SI
  -- Measure delay time form SI to SO



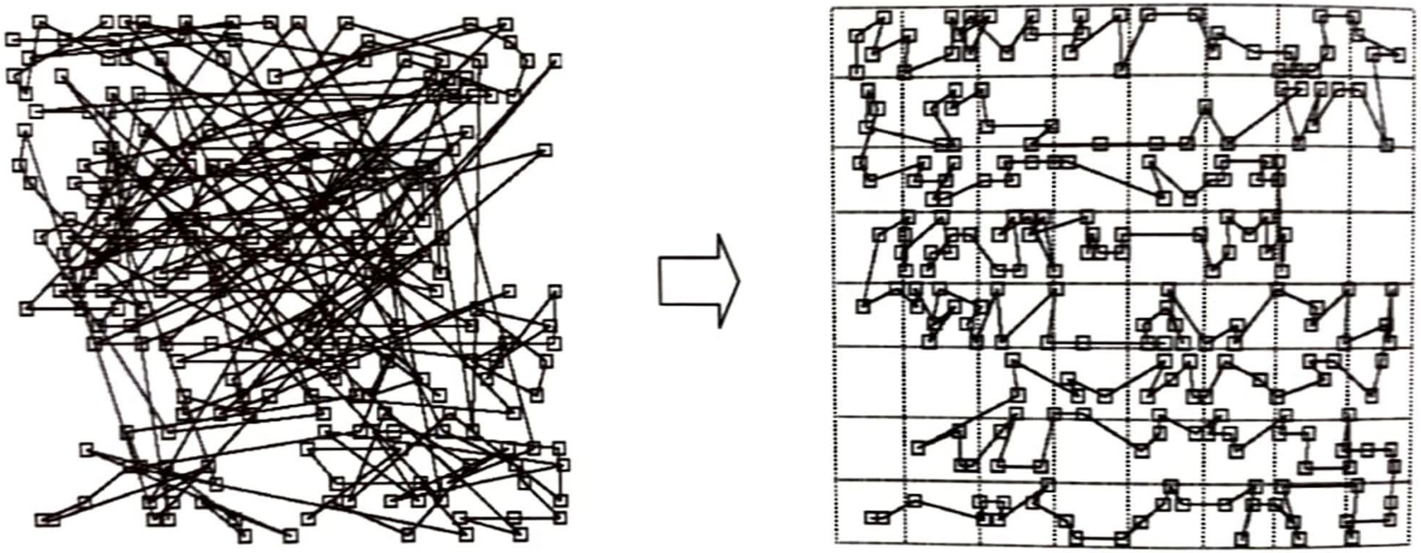
Scan Chain Low Power



(a)

(b)

Scan Chain Low Power

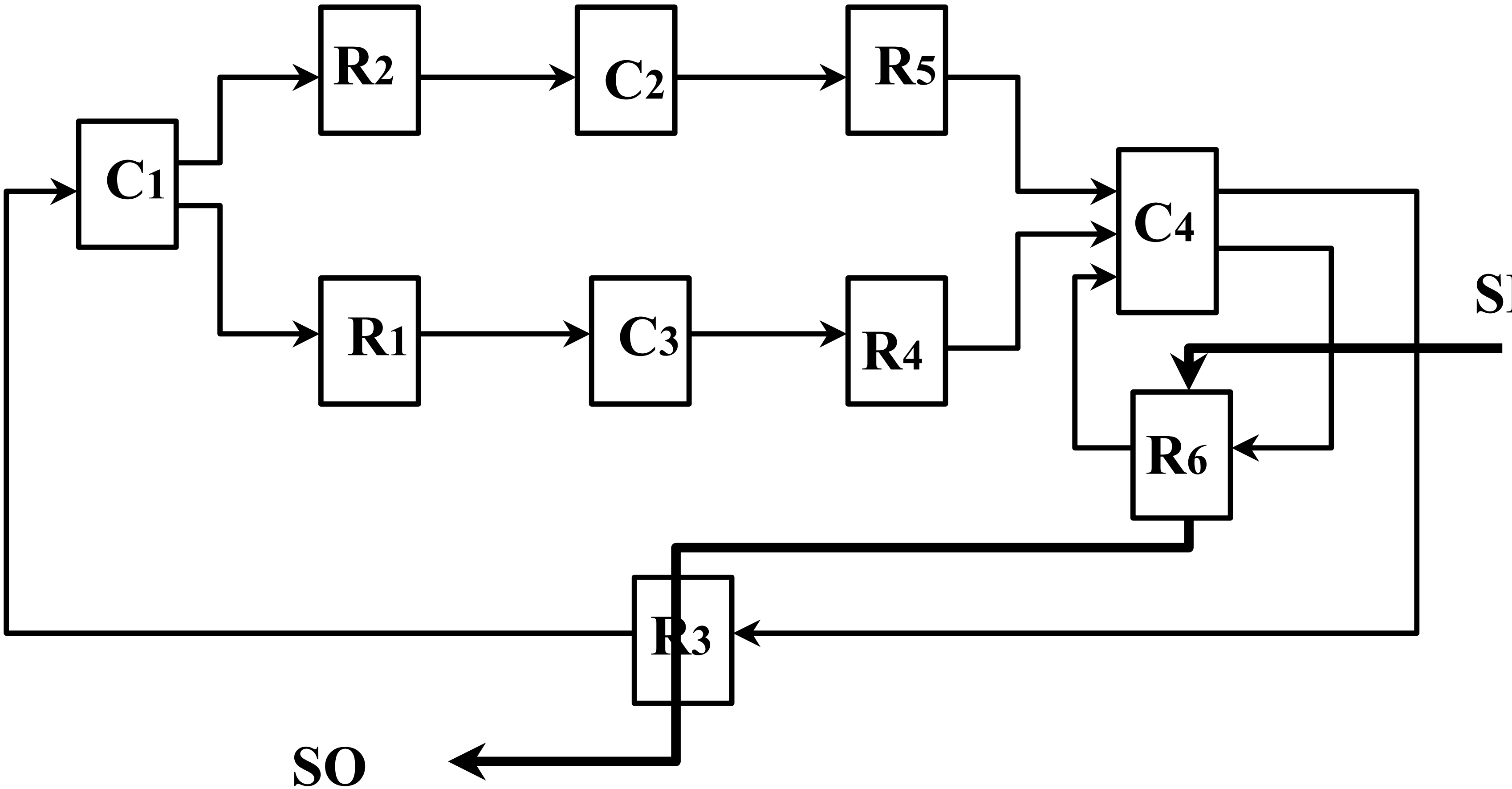


(a)

(b)

Example

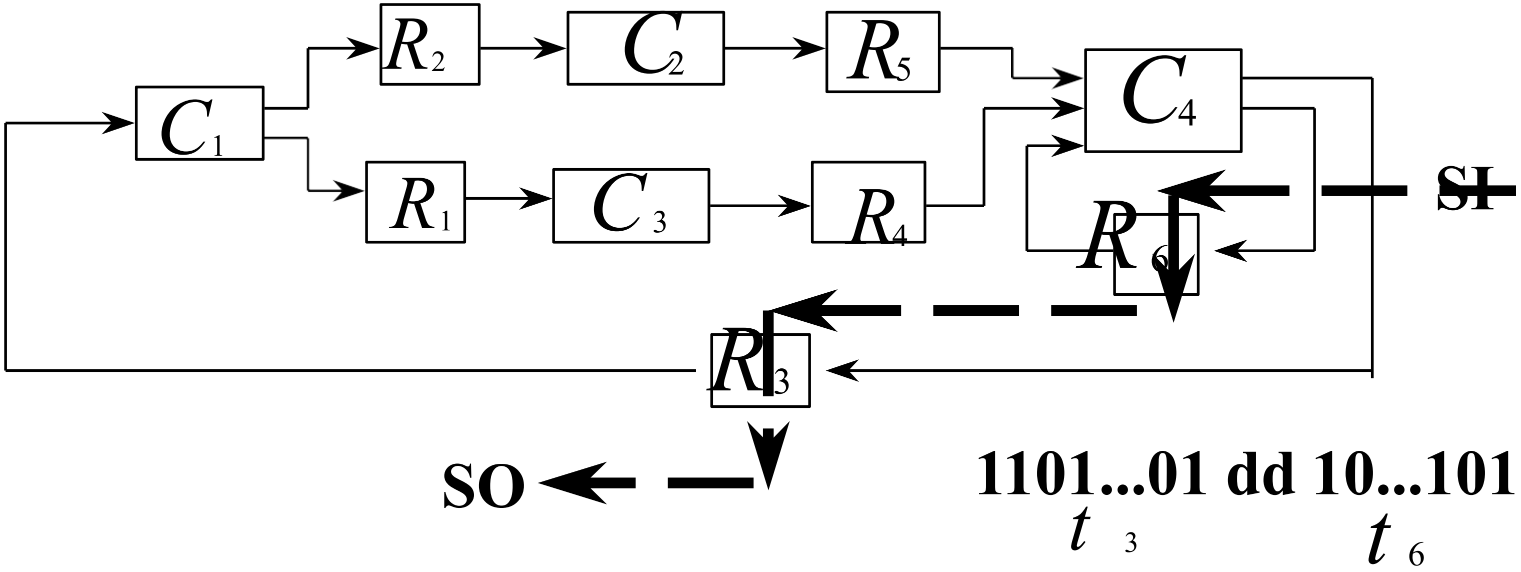
Fig.1



Example (Cont.)

- Test procedure:
  - Scan in a test pattern to R3 and R6 .
  - Hold test pattern in R3, R6 for two clock cycles such that test response appears in R5 and R4.
  - Load data (from R5, R4) to R6, R3 and shift out

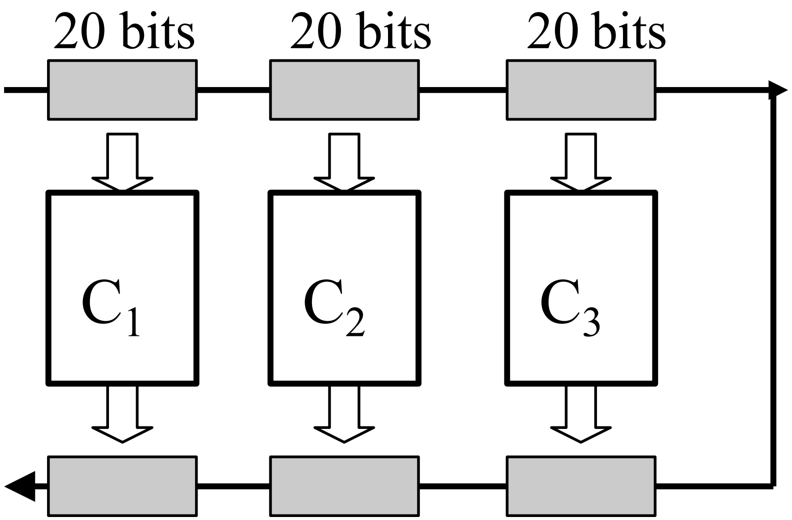
Eliminate on HOLD mode of Scan Registers

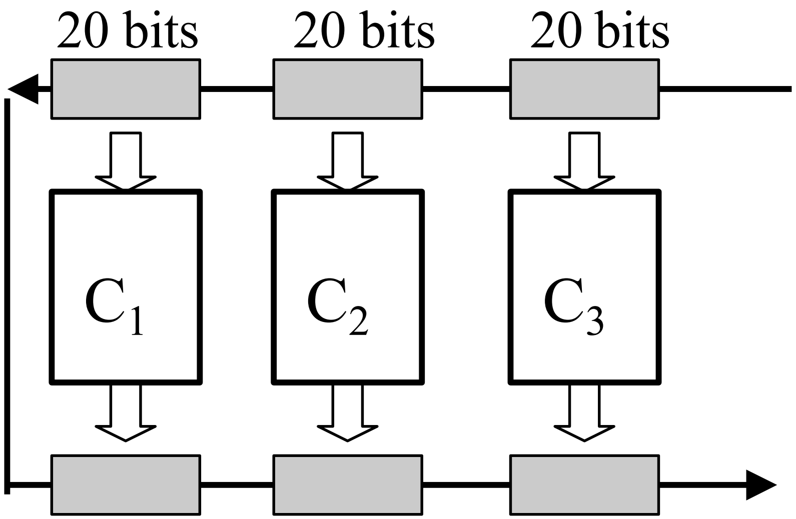


- By adding two dummy bits between the patterns to be scanned to R_3 and R_6 ,the HOLD mode can be eliminated

Multiple Test Session

- # patterns: C_1 :100, C_2: 200 and C_3: 300

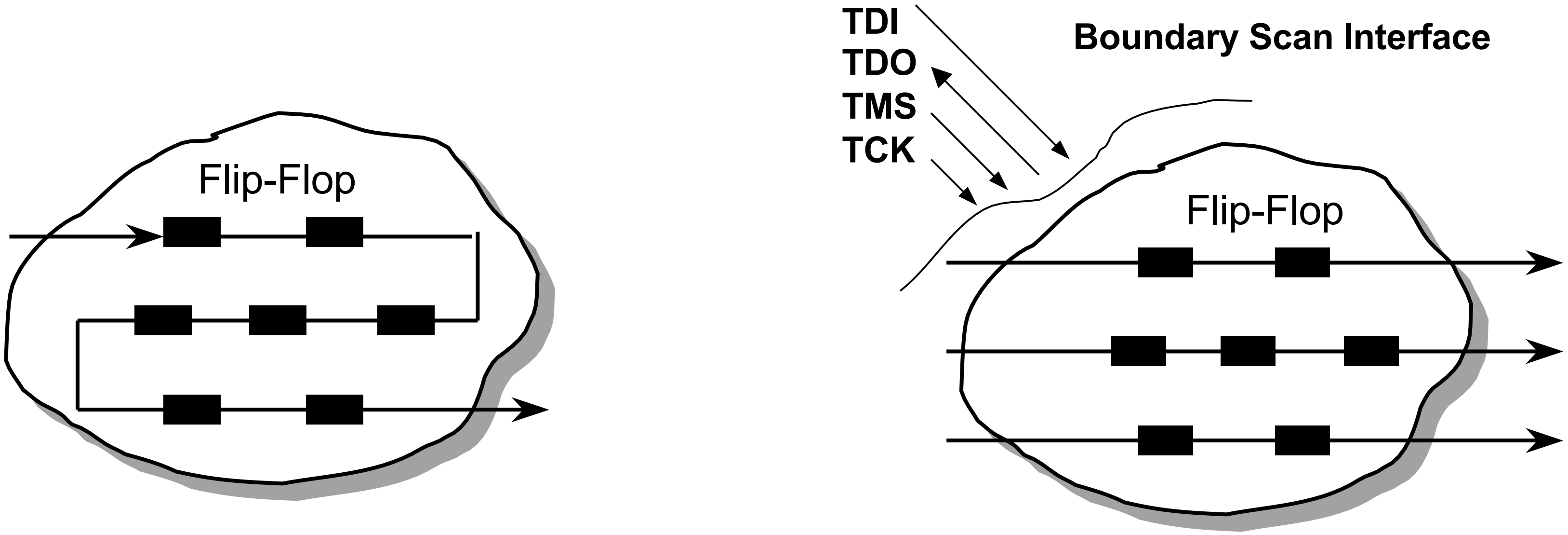




Test Time
= 60 *300
=18,000 (cycles)

Test Time
= 60 *100+40*100+20*100
=12,000 (cycles)

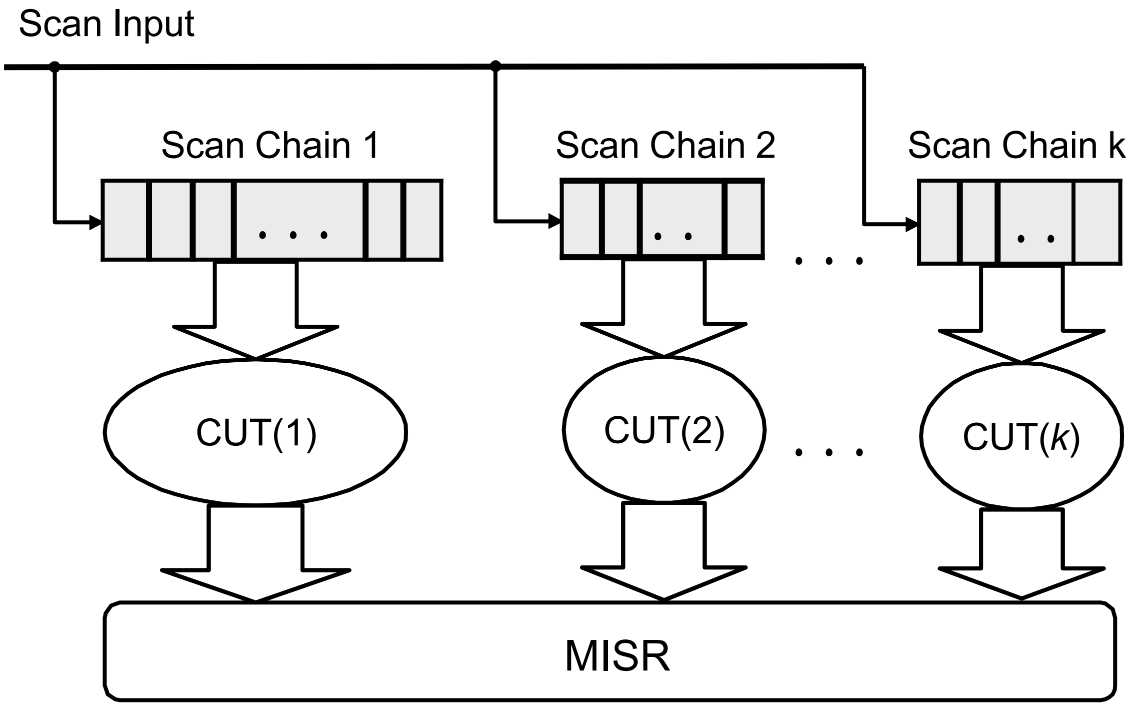
Multiple Scan Chains



- Reduce test application time
- Usually test I/O will share the normal I/O

Broadcast Scan Chains-
General Hardware Architecture

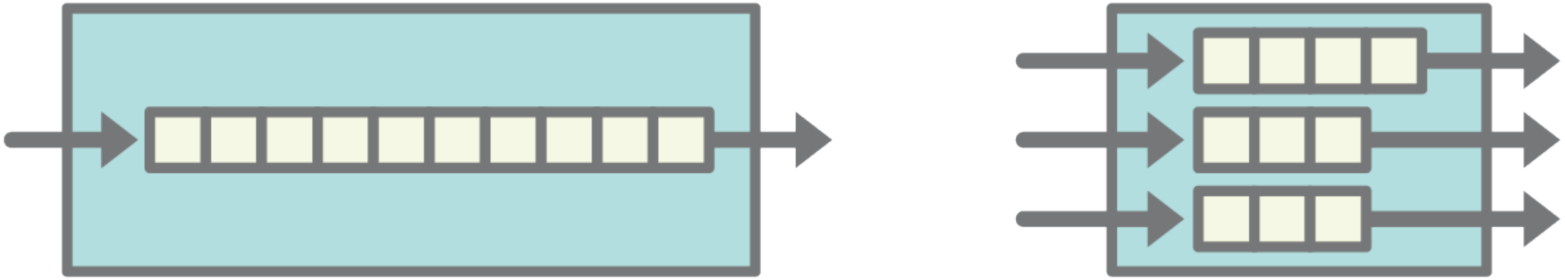
- **Using a single data input to support multiple scan chains**



Multiple Scan Chain

Example : 10k SSF partitioned into three sacn chains

- **L=Max{ 3K, 3K, 4K}= 4K**
- **More I/O pin required**
- **60% reduction in test time**



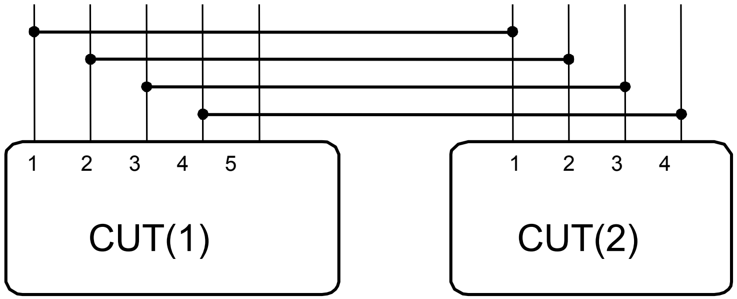
Single scan chain
n=1, N_{SFF}=10K, L=10K

Three scan chains
n=3, N_{SFF}=1K, L=4K

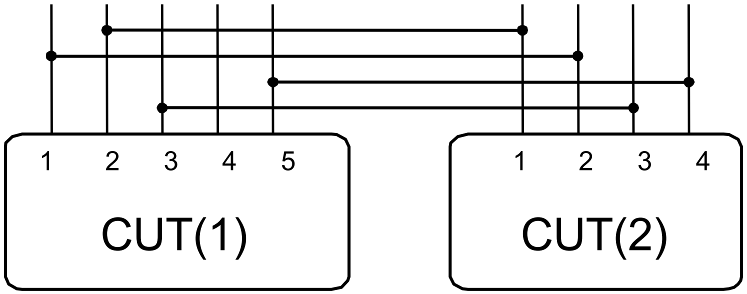
Virtual Circuits

- The inputs of CUTs are connected in a 1-to-1 manner.

  Example :



(a) i-to-i connection



(b) random connection

➡ The whole virtual circuit is considered as one circuit during ATPG.

➡ The resulting test patterns can be shared by all CUTs.