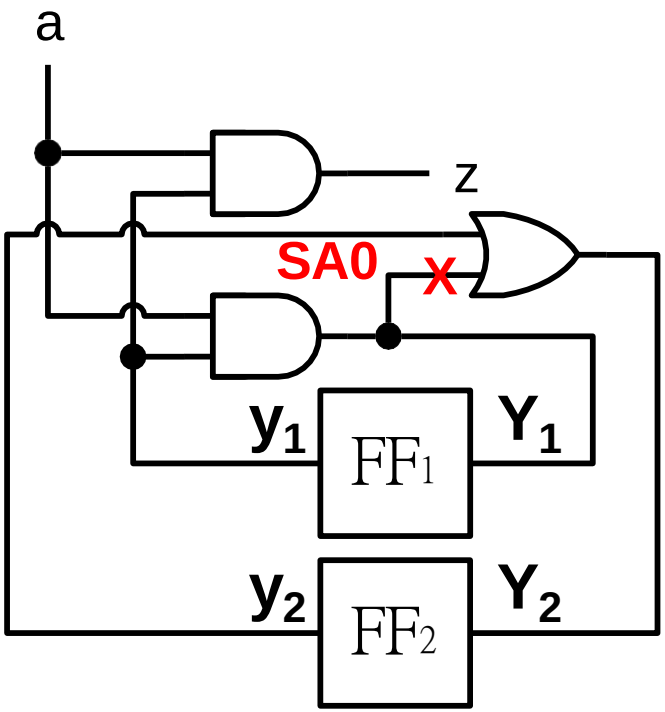
Design For Testability (DFT)
Part I: Internal Scan Chains

Motivating Problem

- **Sequential ATPG fails to generate test pattern. Your manager asked you to add circuits so that this fault can be detected.**
- **Can we change FF to improve controllability/observability?**

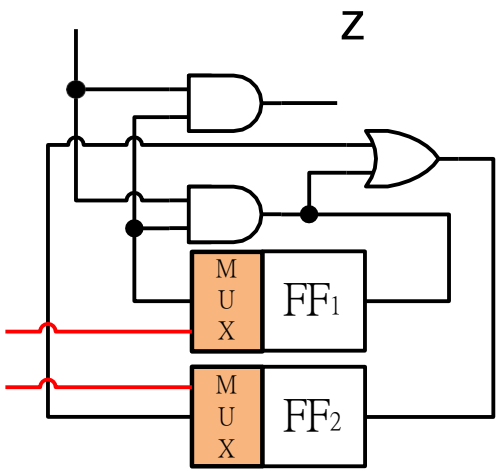


Design for Testability (DFT)

- What is DFT?
  - Insert circuitry that has little to do with function, but
  - Improves testability : controllability and observability
- Why DFT?
  - 1. Reduce ATPG effort
    - Shorten ATPG run time
  - 2. Improve test quality
    - High fault coverage
  - 3. Reduce test cost
    - Shorten test length and test time
  - 4. Reduce time to market
    - Easy debug and diagnosis

Penalty of DFT

- 1. Design effort overhead
- 2. Performance degradation
- 3. Hardware overhead
  - extra area, extra pins
- 4. Yield loss
  - Larger area means lower yield
- 5. Power overhead
  - "Power tax" of testing



- In 1970's, expensive silicon and simple design, DFT not popular
  - Now, cheaper silicon and complex design, DFT is necessary

DFT is Necessary for Modern Designs

Quiz

Q: Which of the following is NOT true about DFT?

A. DFT reduces designers' effort

B. DFT cost extra hardware and power

C. DFT reduces test cost so it is needed for complex design

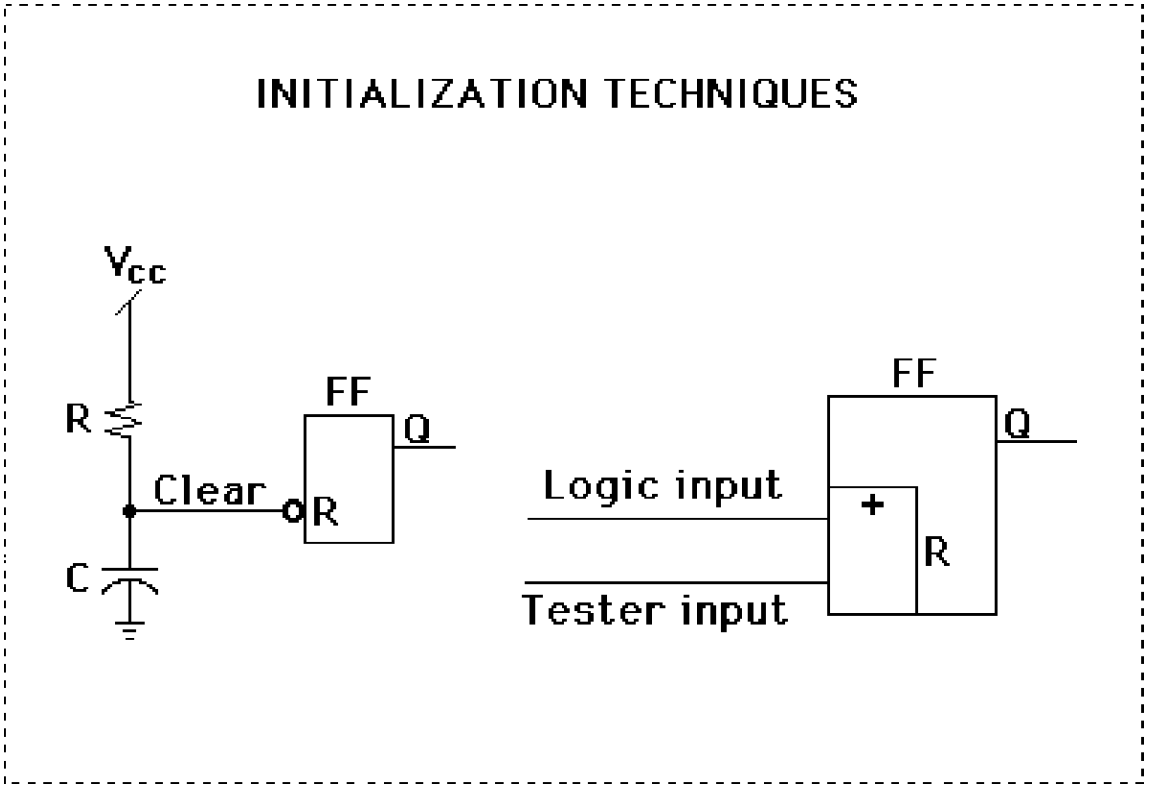
ANS:

Ad Hoc DFT

- What is “Ad hoc” ?
  - for the particular end or case at hand without consideration of wider application
- What is *Ad Hoc DFT* ?
  - DFT technique for a particular design
    - Not generally applicable to all designs
  - Relies on good design practice learned from experience
- Ad-hoc DFT examples:
  - Rule1: design circuits easily initializable
  - Rule2: disable internal clocks during test
  - Rule3: partition large circuits into small blocks
  - Rule4: insert test points into circuits of low testability
  - ….

Ad Hoc DFT Example (1)

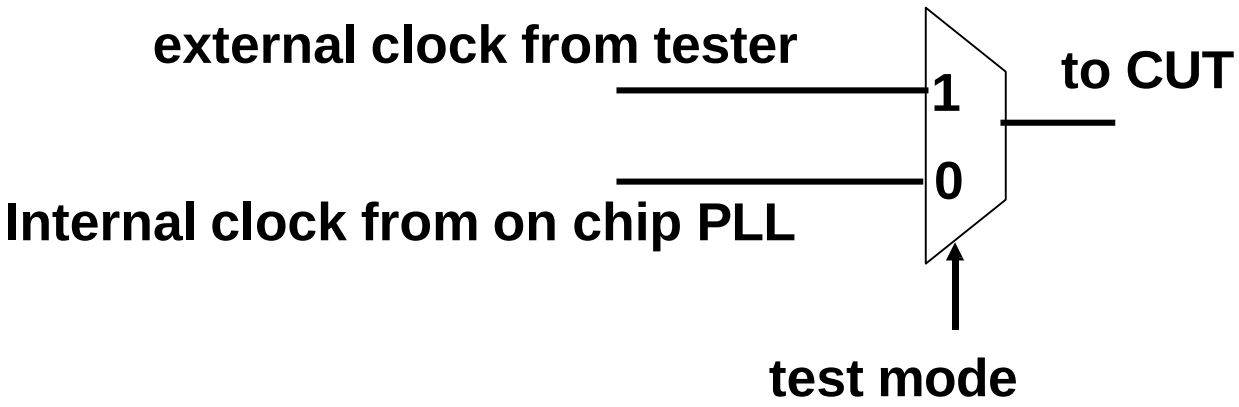
- **Rule1: design circuits easily initializable**
  - **Self-initialization, or**
  - **Initializble from tester**



Courtesy Prof. McClueksy, Stanford University

Ad Hoc DFT Example (2)

- **Rule2: disable internal clocks during test**
  - **Disable on-chip PLL in test mode**

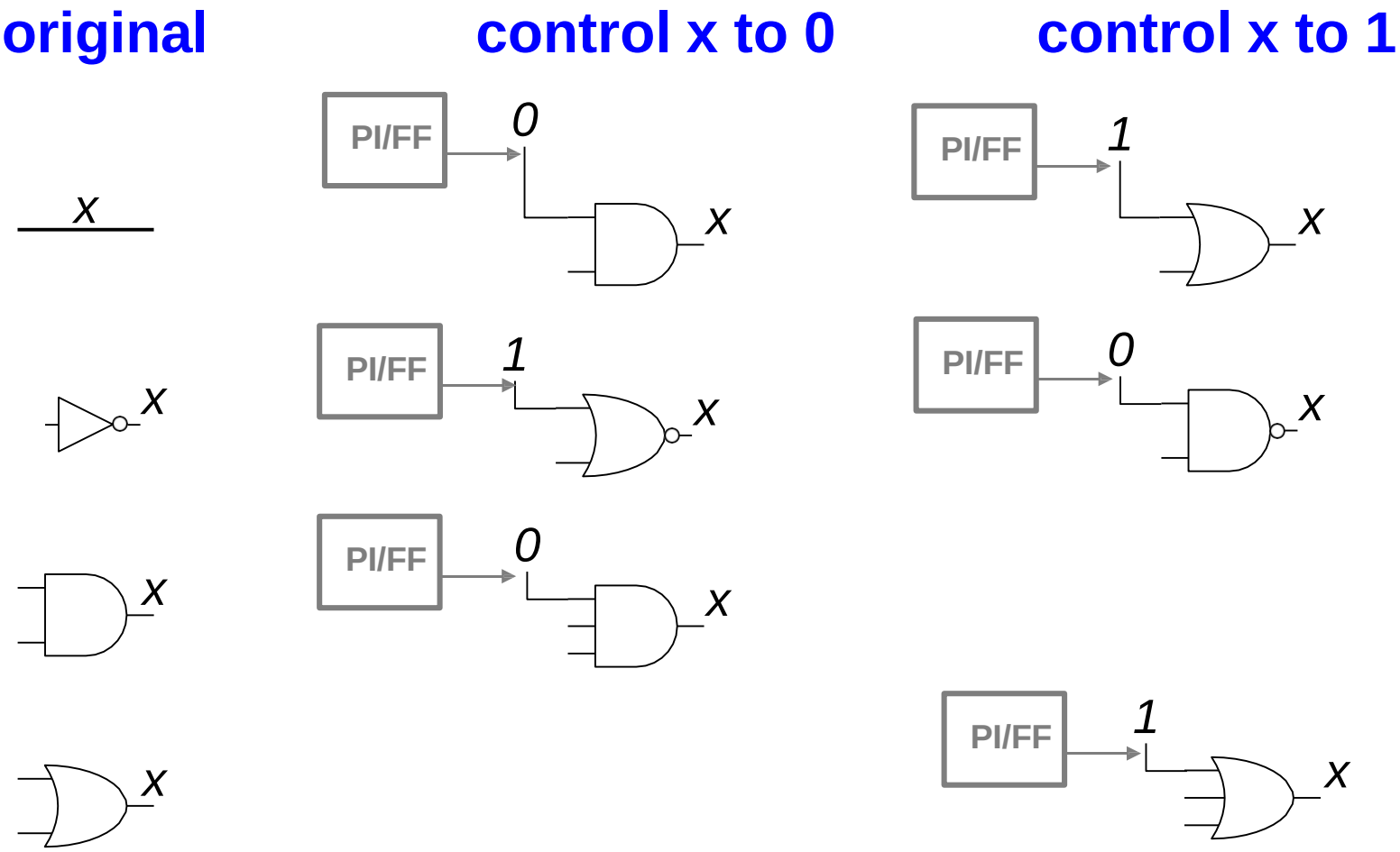


Test Point Insertion

- **Objective:**
  - **Insert test points to enhance**
    - **Controllability**
    - **Observability**
- **Two types of test points**
  - ***Control points*: enhance controllability.**
  - ***Observation points*: enhance observability**

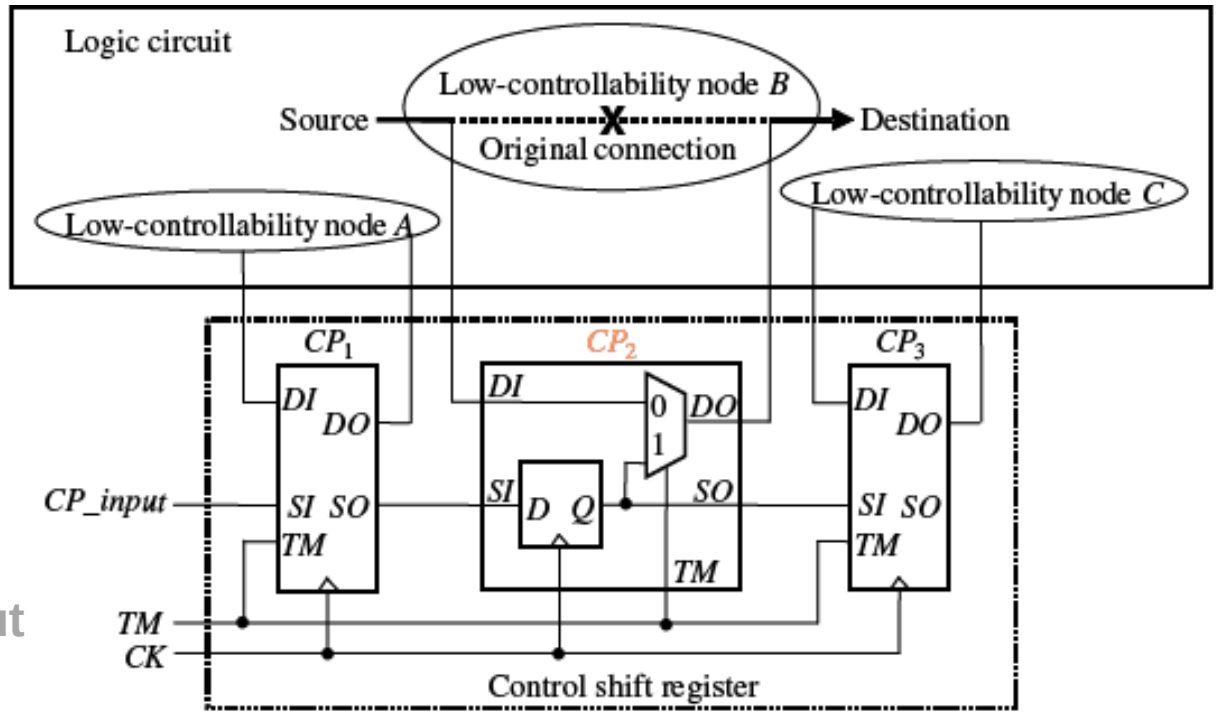
Control Points

- Control signals come from primary input (PI) or scan FF
  - Details see next video



Control Point (2)

- When TM=0, normal operation
- When TM=1, nodes ABC are controlled by FF 1, 2, 3, respectively



Control point insertion

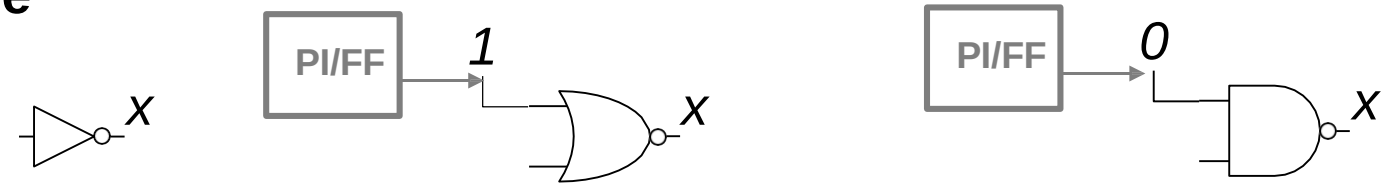
DI=date input
DO=data output
TM=test mode
SI=scan input
SO=scan output

Quiz

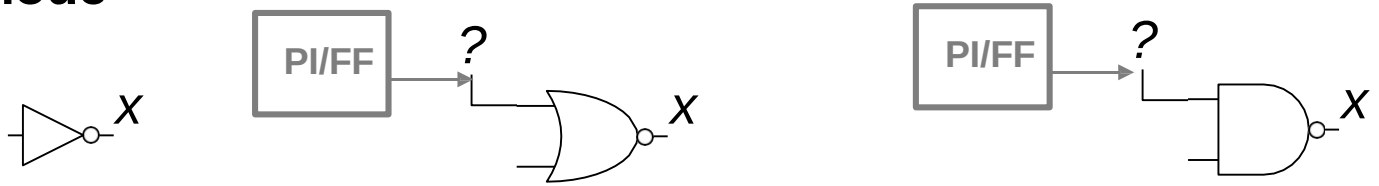
Q: What are control values in NORMAL mode?

ANS:

Test mode

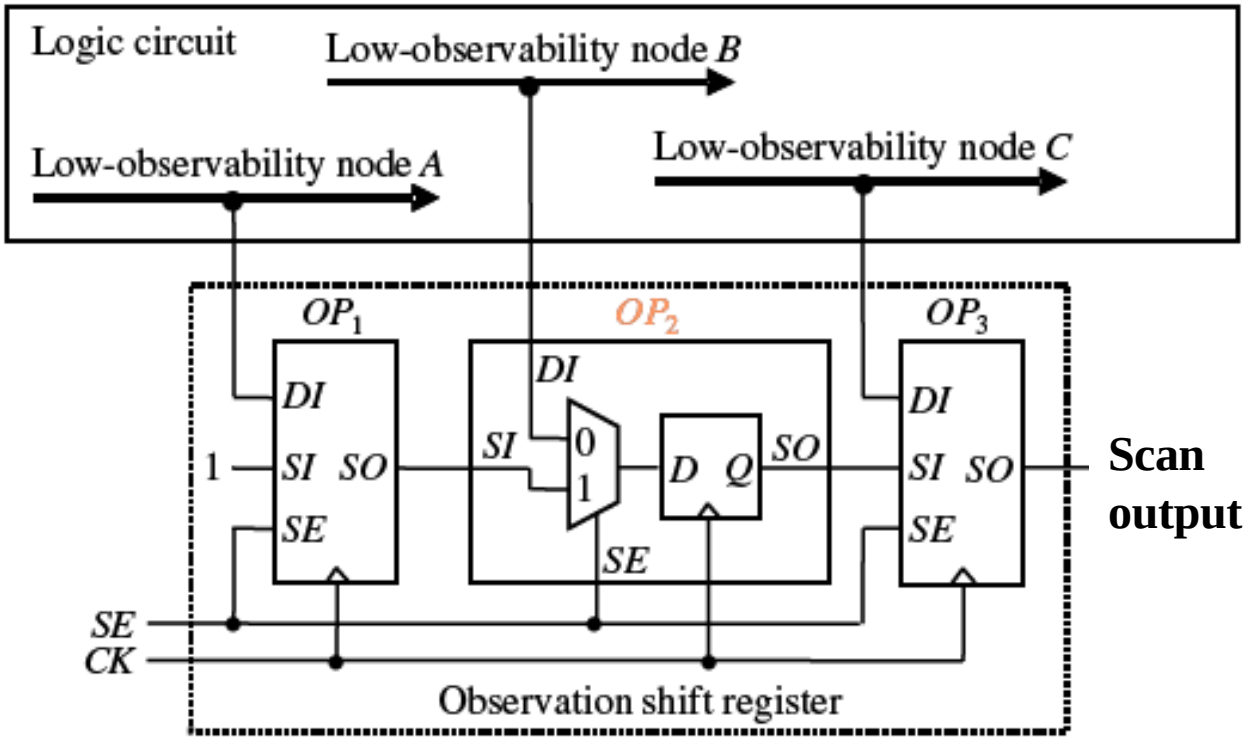


Normal mode



Observation Points

- **SE=0, ABC are captured into FF by one CK**
- **SE=1, after three CK, values are observed at scan output (SO)**

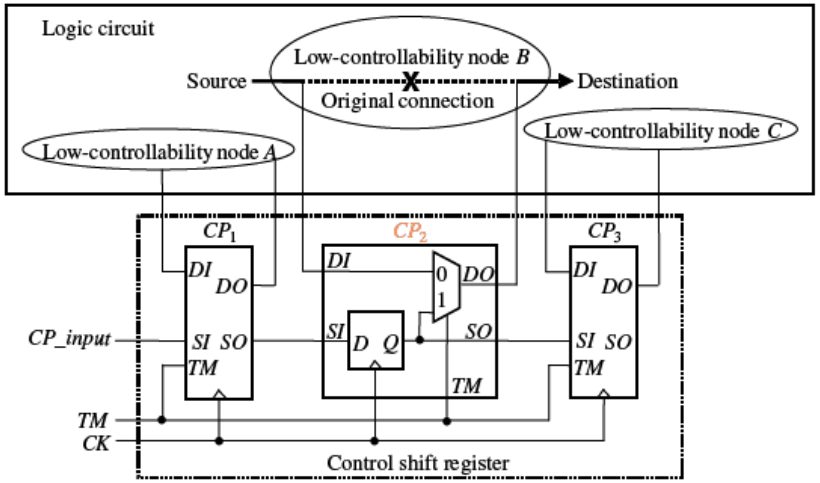


DI=date input
DO=data output
SE=scan enable
SI=scan input
SO=scan output

Observation point insertion

Issues with Test Points

- 1. Performance degradation
  - design slower with test points
- 2. Area overhead
  - About 1 test point every 1K gates
- 3. Pin overhead
- 4. Where to insert test points?
  - Need testability measure
- 5. Built-in Self Test (BIST) needs test point more than ATPG
  - Improve fault coverage , see BIST chapter



Summary

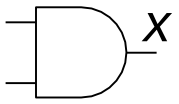
- Introduction to DFT
  - DFT helps to improve controllability/observability
  - *Ad-hoc DFT* based on experience
    - Advantage
      - Suitable for your particular design under test
    - Disadvantage
      - Not systematic → lacks EDA tool
      - As opposed to *structured DFT*
  - Test point insertion
    - control point/ observation points
    - EDA tools ready for use

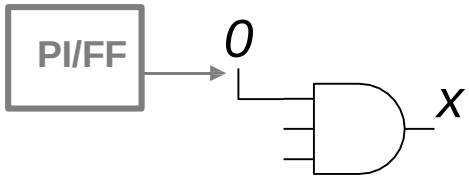
Structured DFT is Preferred than Ad Hoc DFT

FFT

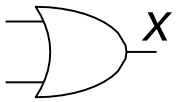
- **Q: What do we do to control AND gate output to 1?**

original **control x to 0** **control x to 1**

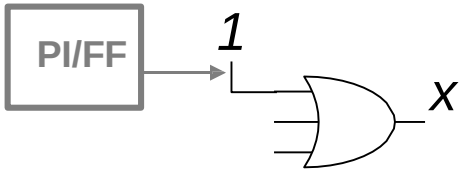




?

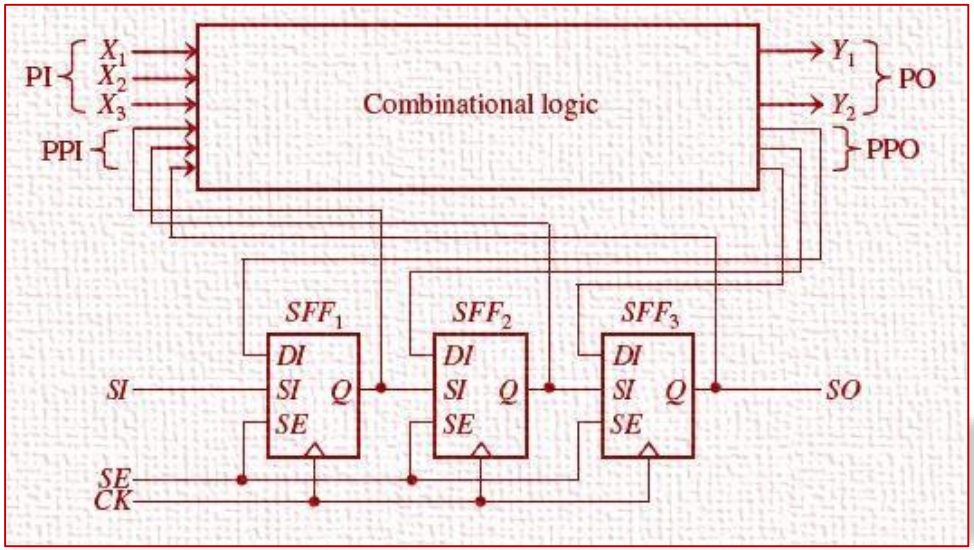


?



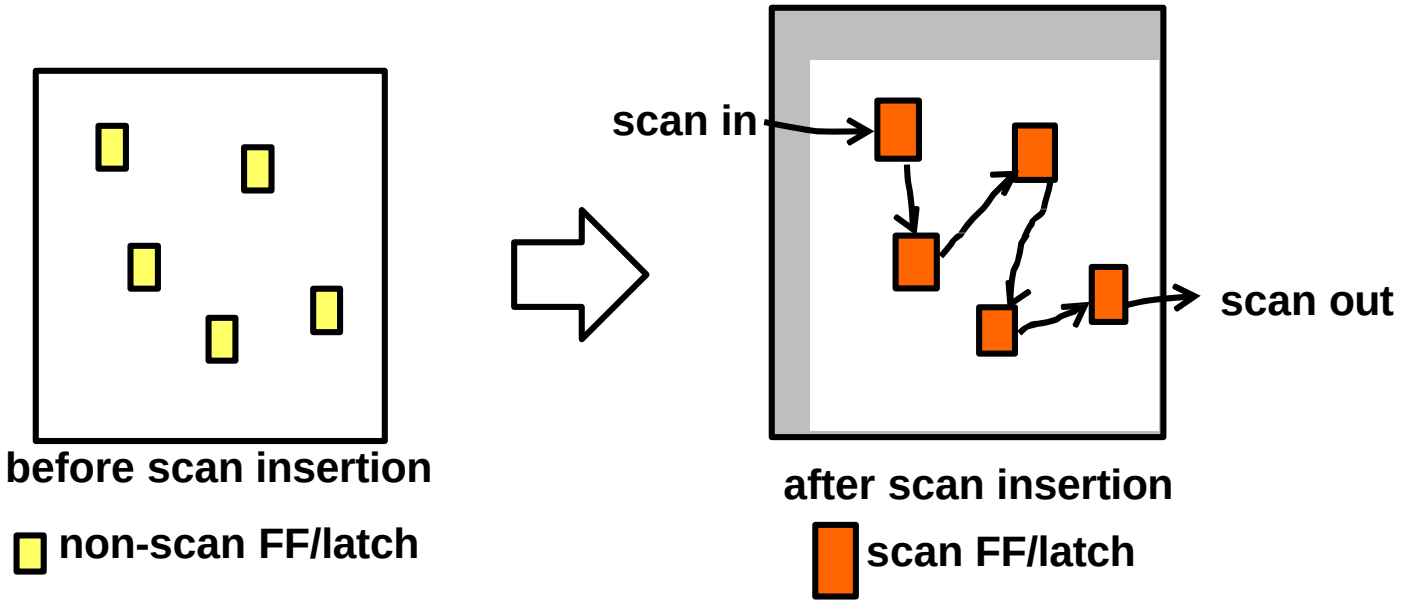
DFT - Part 1

- Introduction
- **Internal Scan**
  - FF-based
    - MUXed-D scan (1973, Stanford)
    - Clocked scan (1968, NEC)
  - Latch-based
    - LSSD (1977, IBM)
- Scan Design Flow
- Issues and Solutions
- Conclusion

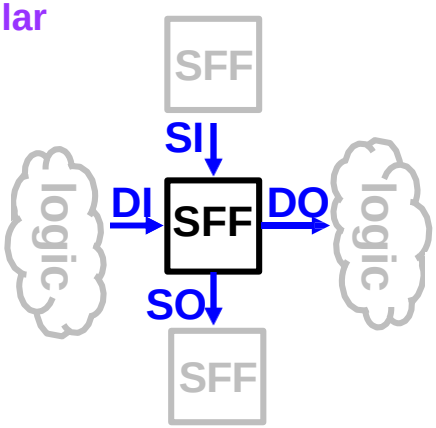


Scan Chains

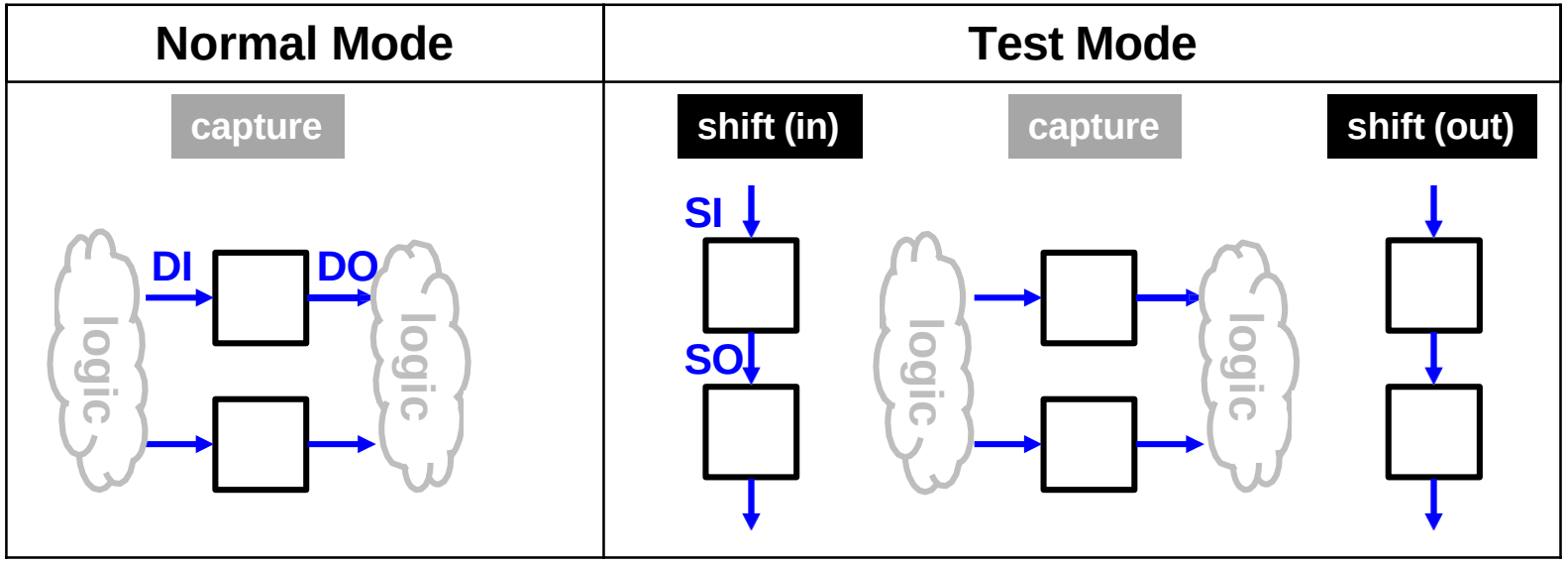
- Scan: connect (internal) FF/latches as shift register
  - Control and observe FF/latches in test mode
  - Remain original function in normal mode
- Proposed in early 1970's [Williams 73][Eichelberger 77]
  - Most important DFT for synchronous digital circuits
- *Scan chain insertion*: aka. *DFT insertion*, *DFT synthesis*
  - 1. Replace FF/latch 2. Stitch FF/latch into a chain



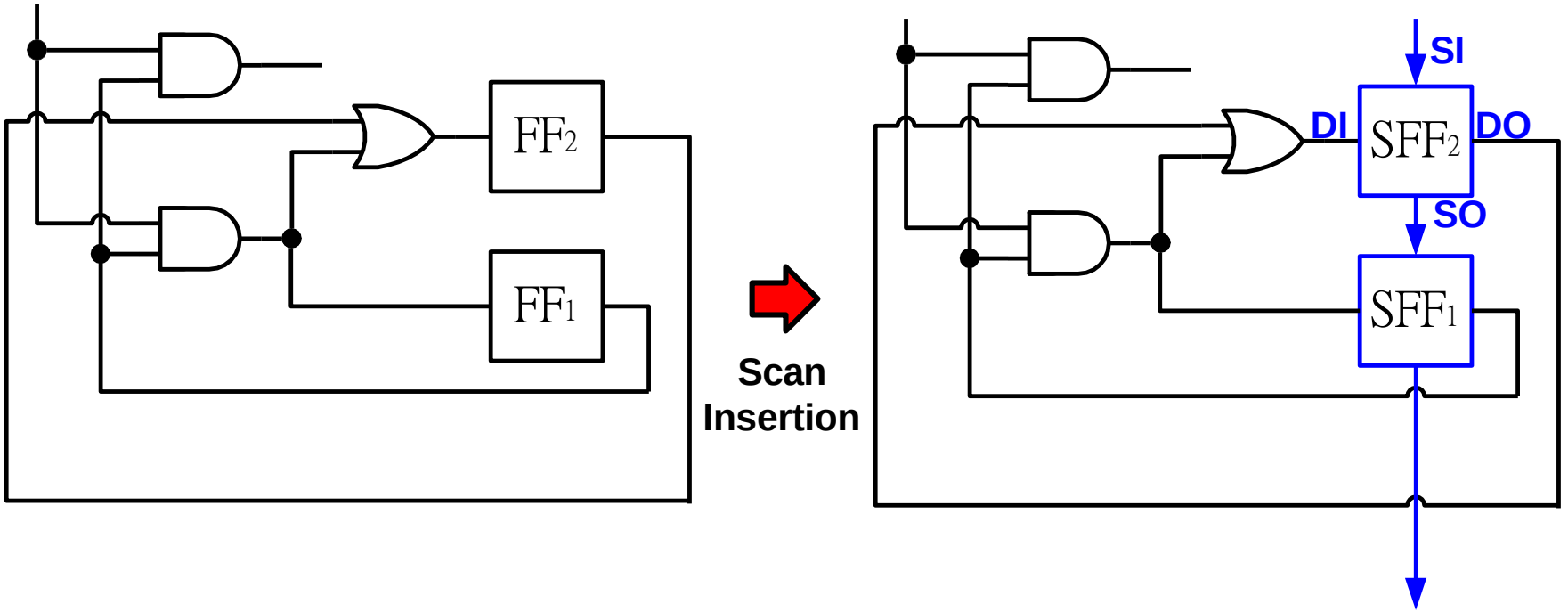
Scan Flip-Flop (SFF) *scan latch is similar



- SFF has four main pins:
  - Scan Chain: *Scan Input* (SI), *Scan output* (SO)
  - Logic: *Data Input* (DI), *Data Output* (DO)
    - DO and SO can be shared
- SFF has two functions: *shift* and *capture*
- Circuit has two operation modes: *Normal* mode and *Test* mode

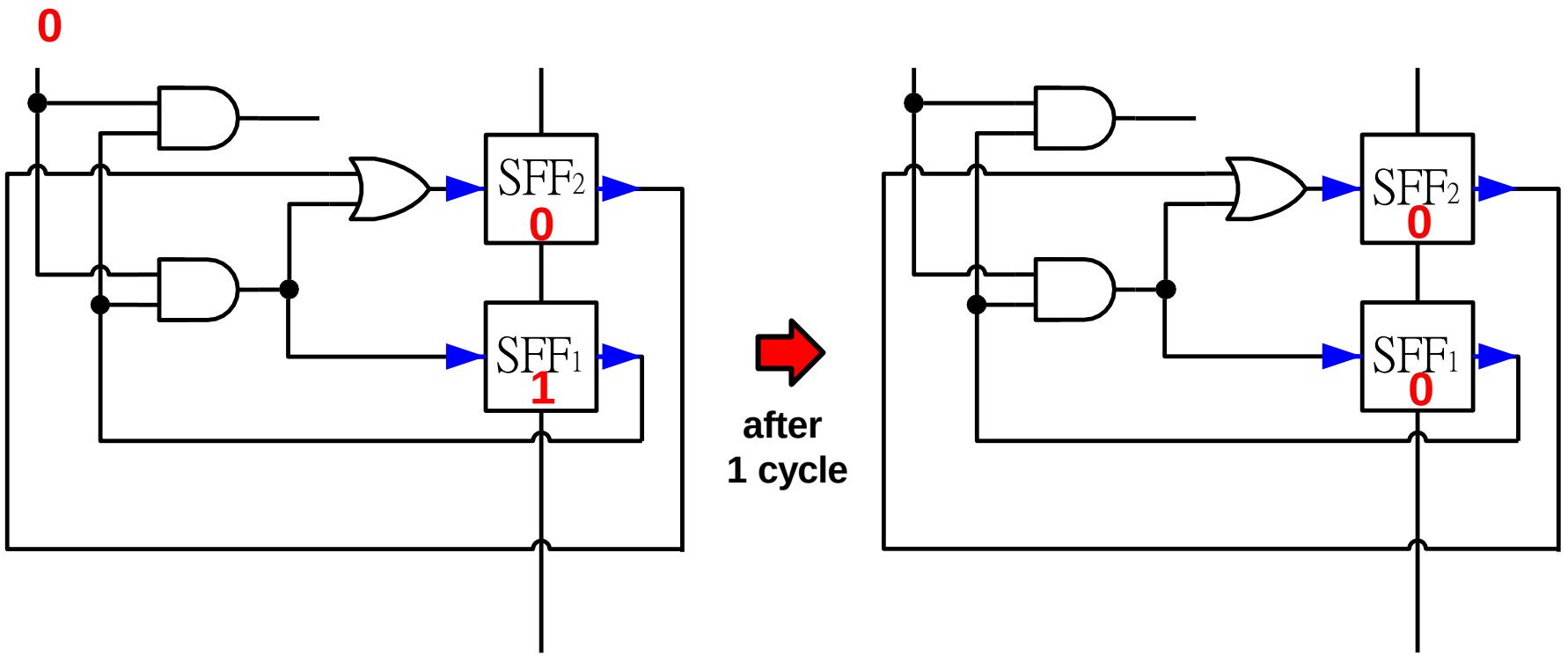


Example: Scan Insertion

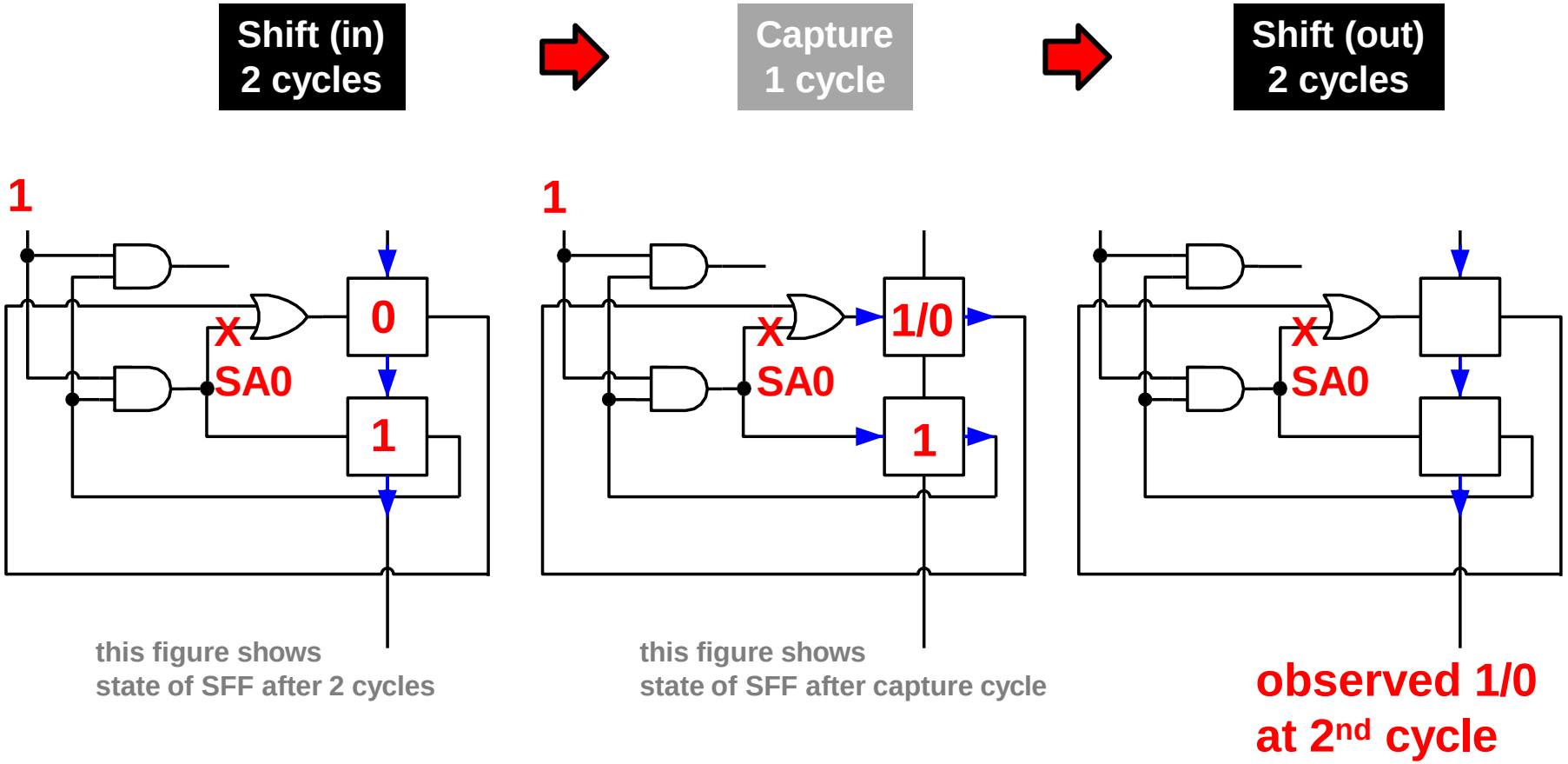


Example: Normal Mode

- **Scan-FF same as non-scan FF**



Example: Test Mode

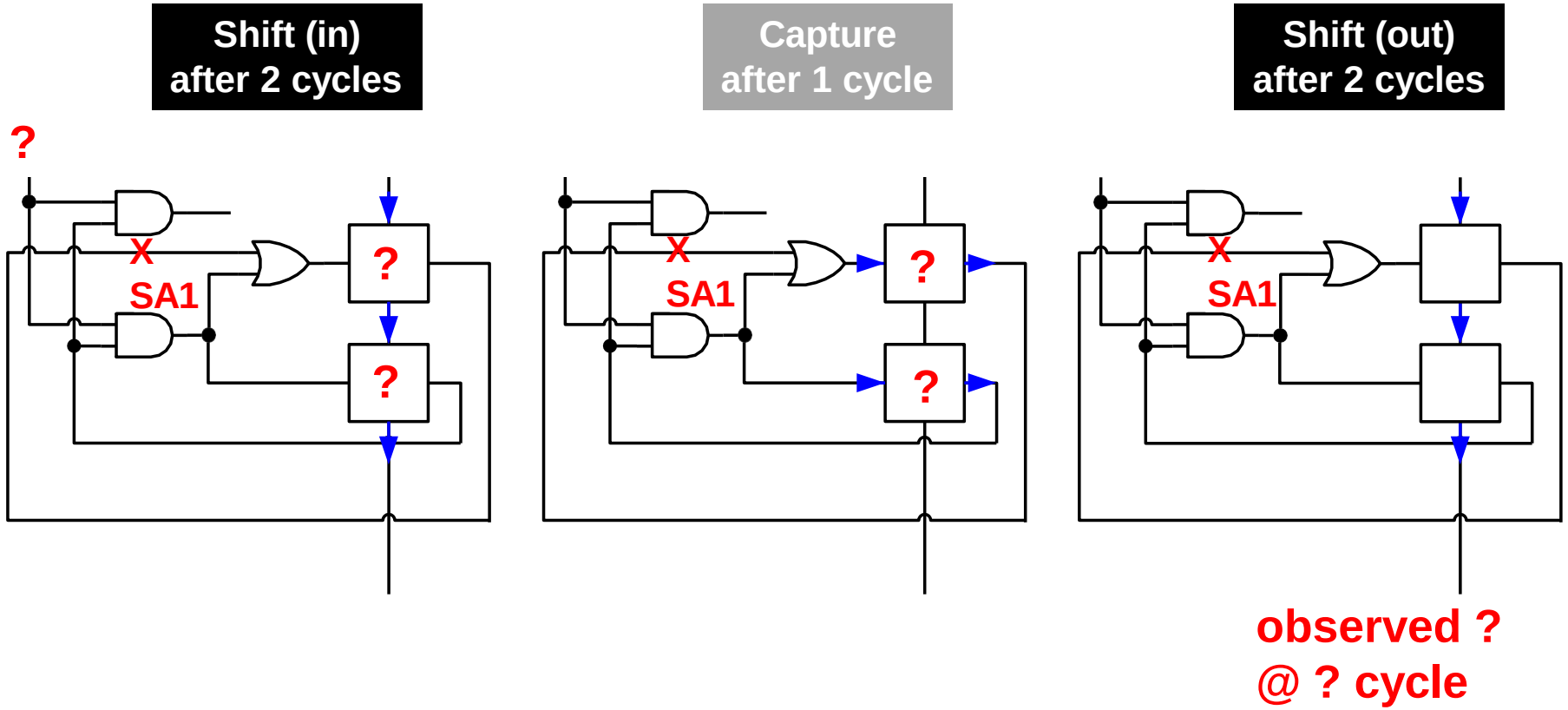


Fault Detected in Test Mode

NOTE: this fault untestable in Normal Mode. why?

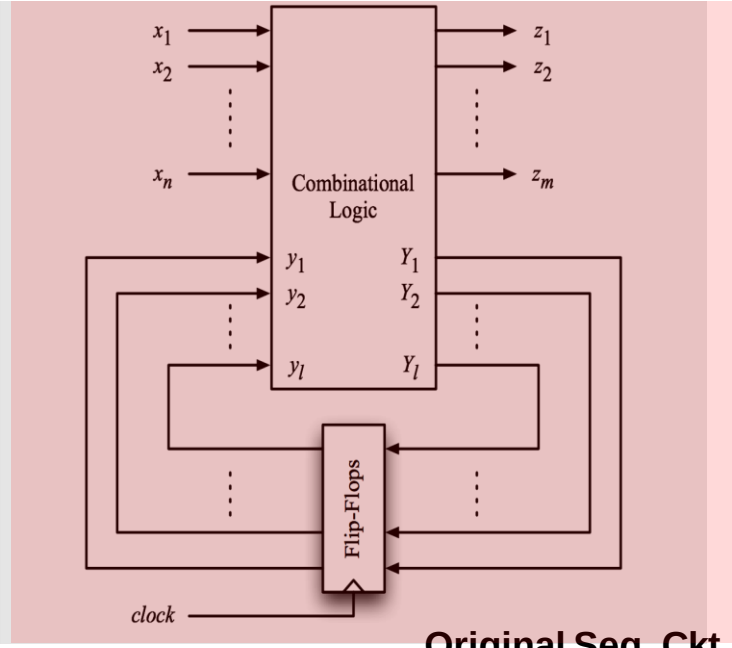
Quiz

Q: Consider SA1 fault, please fill in values of ?

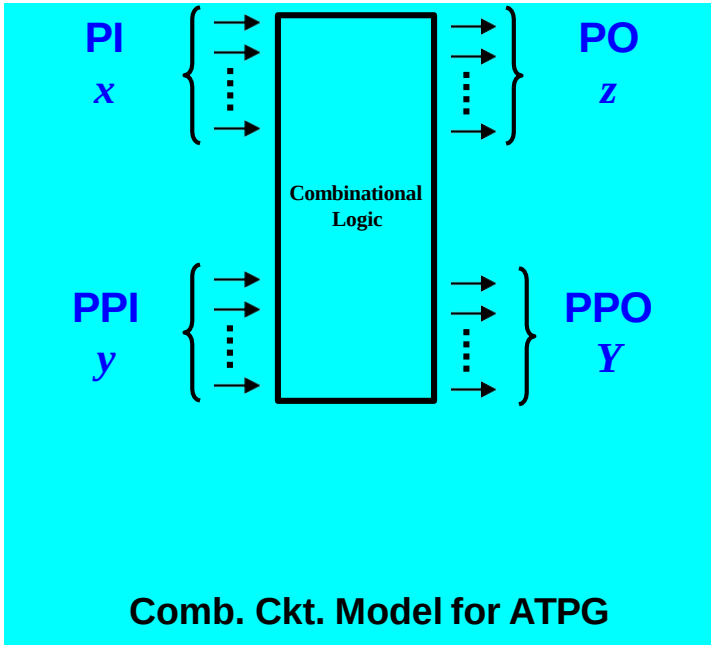


Scan Turns Seq. Ckt. to Comb. Ckt.

- Scan turns sequential ckt into combinational ckt in test mode
  - ATPG sees only comb. ckt. model
- SFF DO become *Pseudo Primary Input* (PPI), fully controllable
- SFF DI become *Pseudo Primary Output* (PPO), fully observable



Original Seq. Ckt.



Comb. Ckt. Model for ATPG

Scan Makes ATPG Easier!

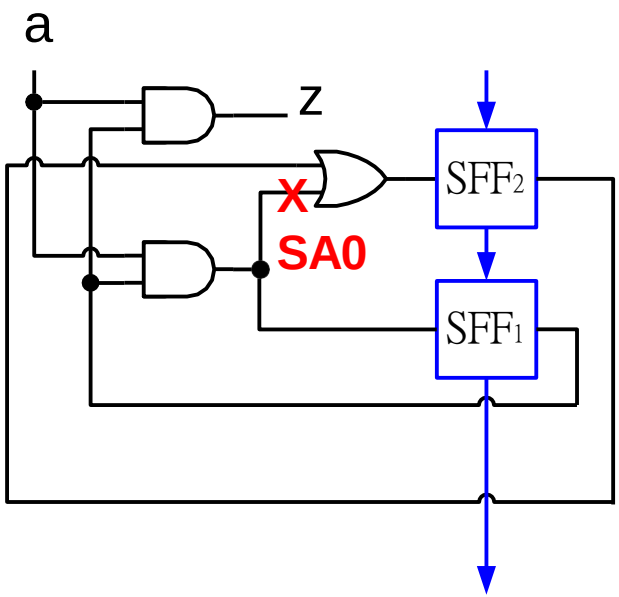
QUIZ

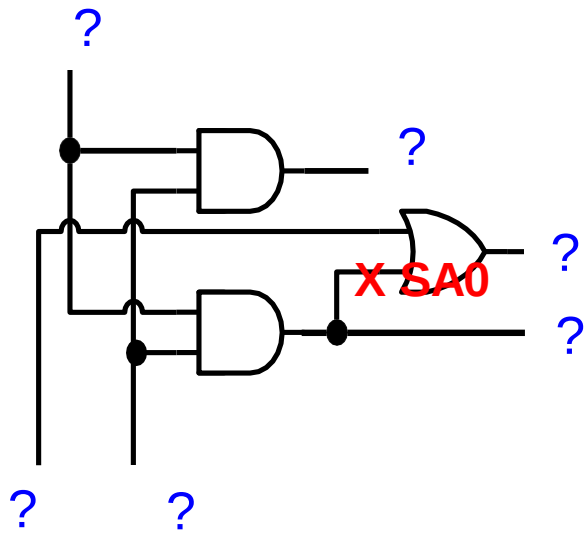
We insert scan into this circuit so we can remove FF in ATPG model.

Q1: Which pins are PI? PO? PPI? PPO?

Q2: If we want to detect SA0 fault, what are their values?

ANS:





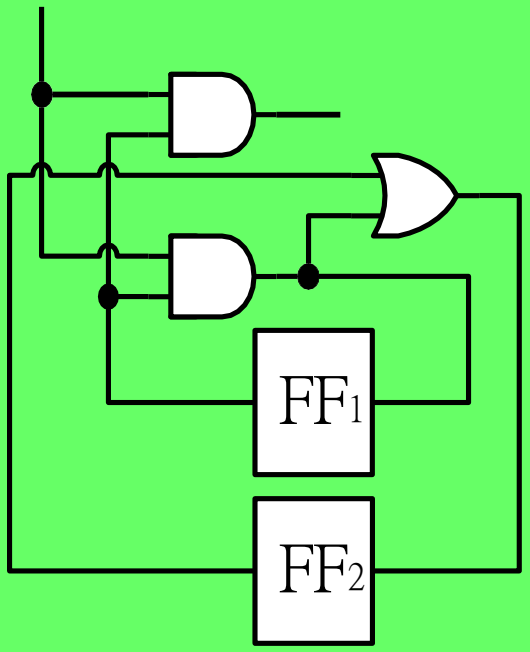
Pros and Cons of Scan

- Advantages of scan
  - Systematic DFT, not *ad hoc.* Many automatic tools.
  - Easy ATPG: faster run time and higher fault coverage
  - Easy *silicon debugging* or *diagnosis*
- Disadvantages of scan
  - Area overhead (typically, 5~10% OH acceptable)
    - larger SFF area + routing for scan chains
  - Performance overhead
    - SFF has longer setup time, hold time
  - Pin overhead
    - scan in, scan out, *scan enable*, *scan clocks* …
  - Power overhead
  - Extra design efforts
    - scan insertion, verification

QUIZ

We insert scan into this circuit and replace non-scan FF by scan FF.

Q: Given the gate area table, what is area overhead of scan DFT?



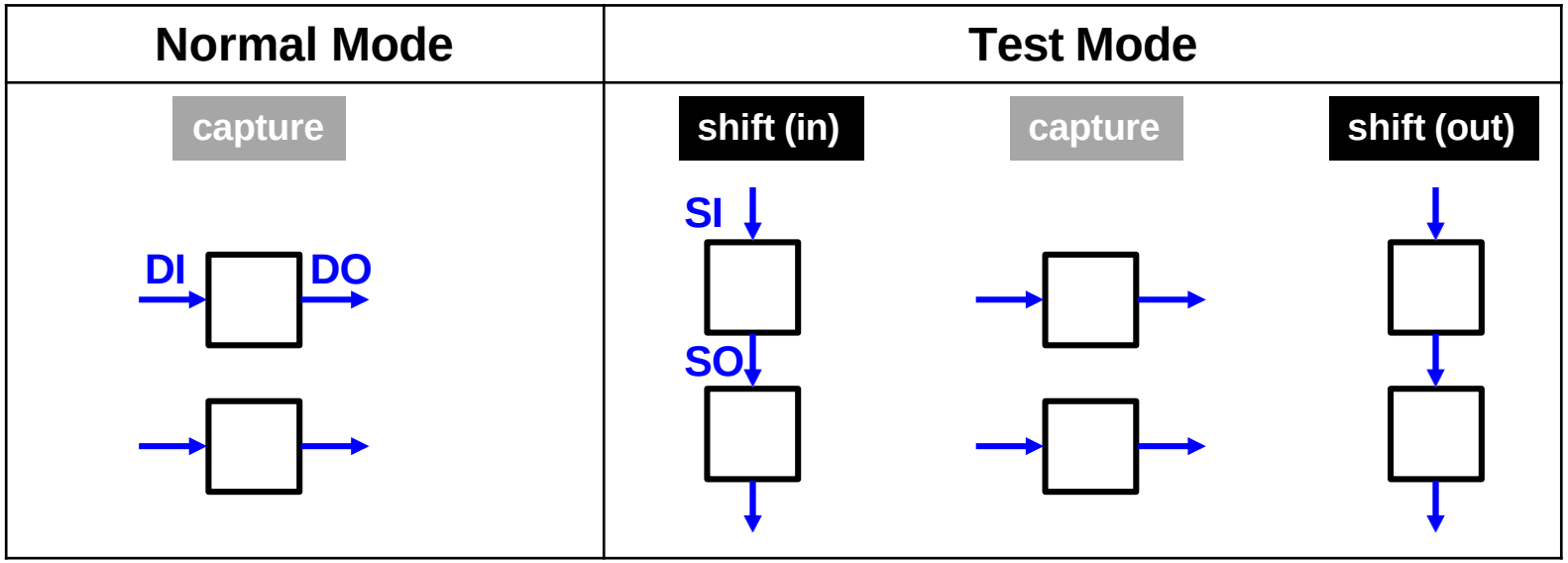
Gate	area
OR	3
AND	3
non-scan FF	5
scan FF	6

ANS:

$$\text{area O.H.} = \frac{\text{area}_{afterDFT} - \text{area}_{beforeDFT}}{\text{area}_{beforeDFT}} =$$

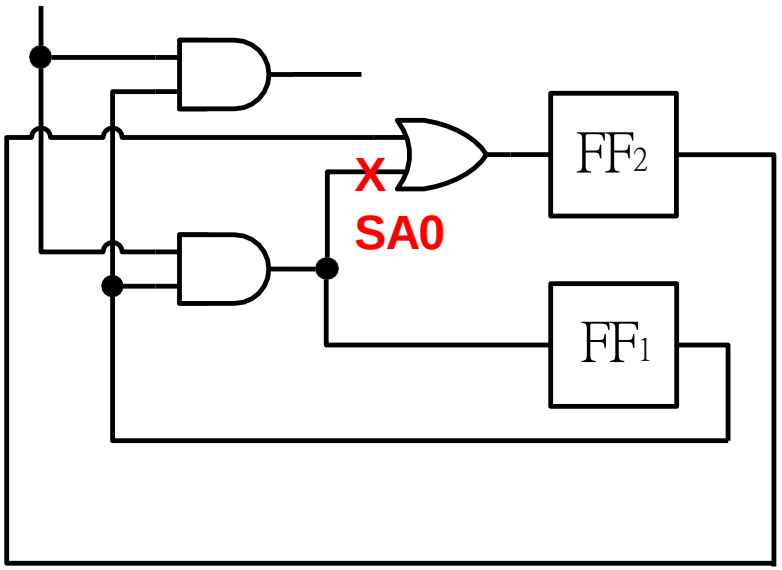
Summary

- Scan is most popular DFT for digital VLSI
- Scan FF has four pins: SI, SO, DI, DO
- Circuit has two operation modes: normal mode, test mode
  - Test mode: shift (in), capture, shift (out)
- Scan makes ATPG easier at cost of overhead (area/performance …)



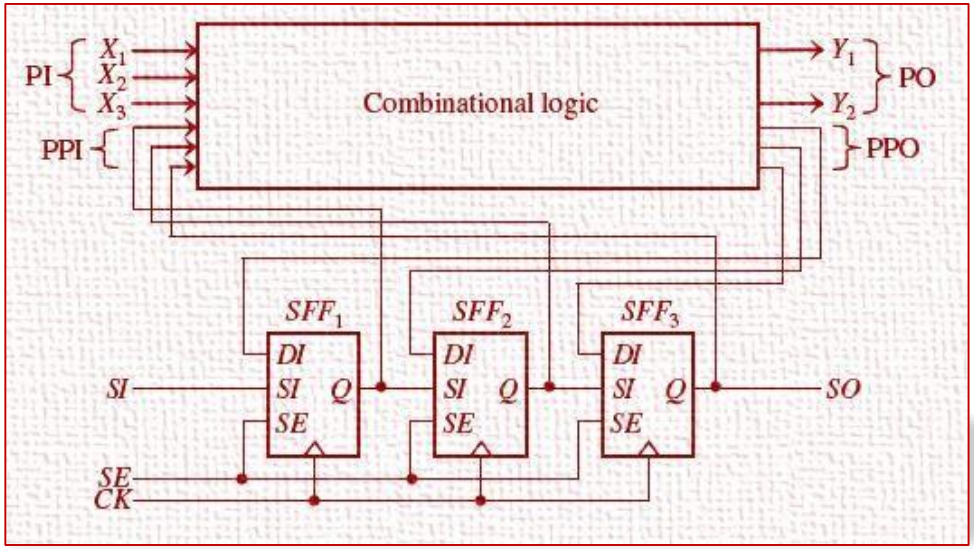
FFT

- **Q1: This stuck-at zero fault is untestable in normal mode, why?**
- **Q2: Since it is untestable in normal mode, why do we care about it?**



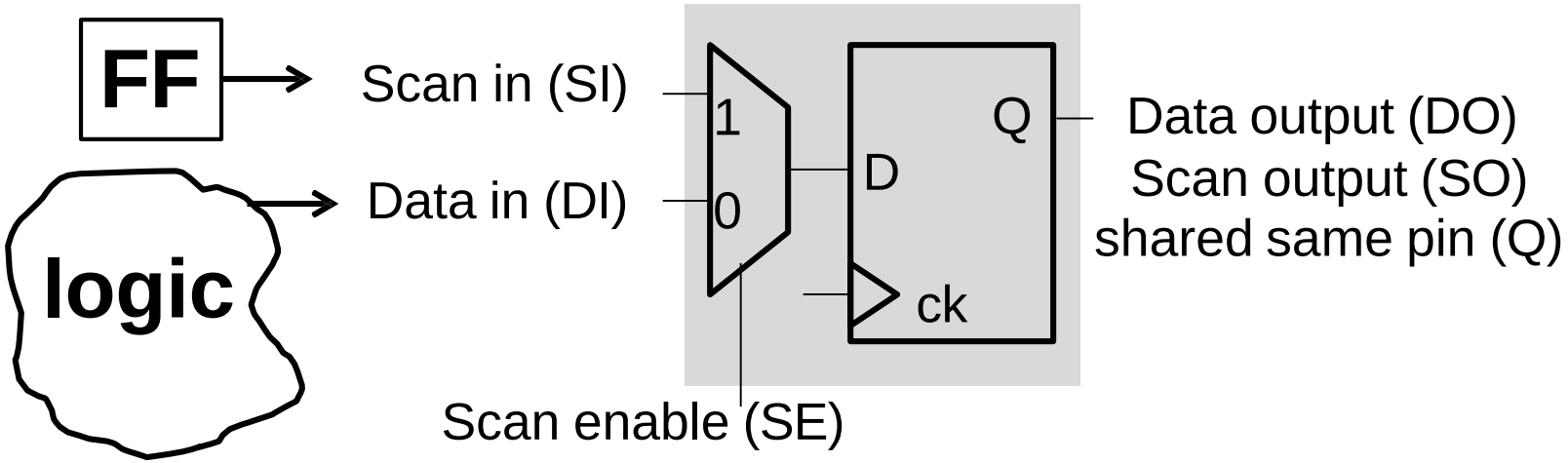
DFT - Part 1

- Introduction
- **Internal Scan**
  - **FF-based**
    - **MUXed-D scan (1973, Stanford)**
      - **MUXed-D scan flip-flop**
      - **Test Mode Operation**
    - Clocked scan (1968, NEC)
    - Other scan
  - Latch-based
    - LSSD (1977, IBM)
- Scan Design Flow
- Issues and Solutions
- Conclusion



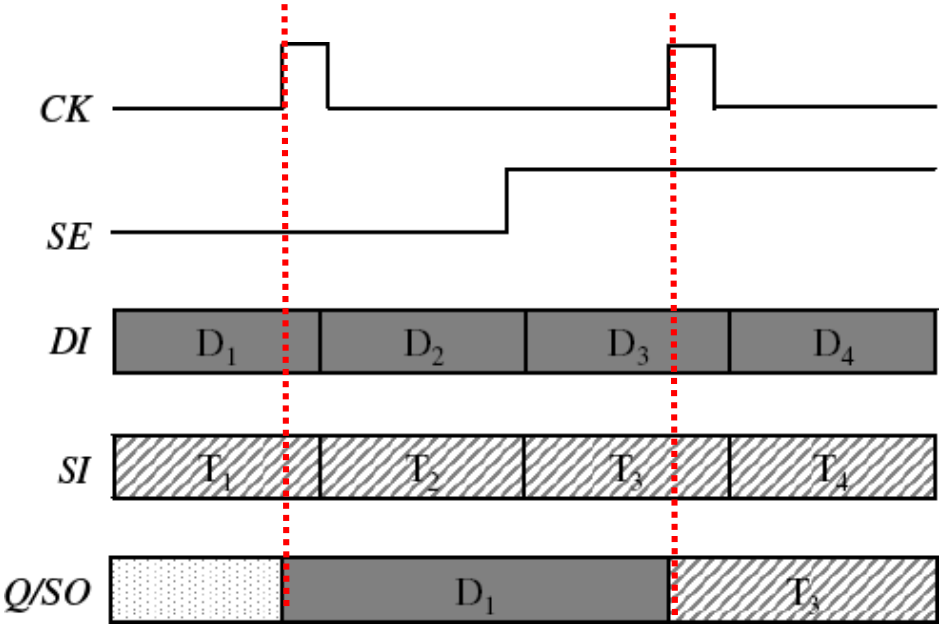
MUXed-D Scan Flip-flop [Williams 73]

- **Often used in flip-flop based, standard cell design**
- **3 input, 1 output, 1 clock pin**
- ***Scan Enable* (SE) pin selects inputs**
  - ***Data in* (DI): from logic**
  - ***Scan in* (SI): from previous scan FF**
- ***Data output* (DO) and *Scan output* (SO) often share same pin (Q)**
- **One multiplier (MUX) slower and larger than regular non-scan FF**

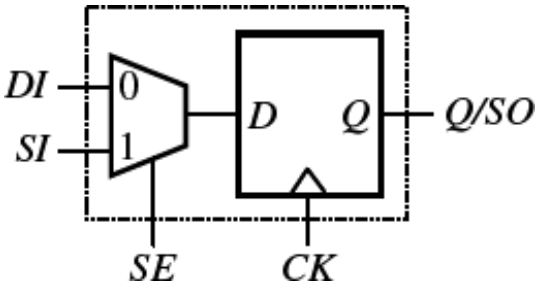


MUXed-D Scan Flip-flop

- **Positive edge triggered MUXed-D scan FF**
  - **SE=1, SI→ Q at positive clock edge**
  - **SE=0, DI→ Q at positive clock edge**



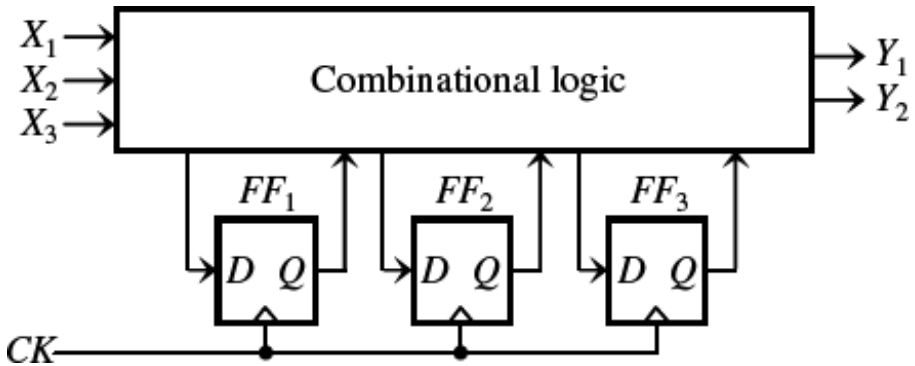
	operation
SE=0	Capture (DI→Q)
SE=1	Shift (SI→SO)



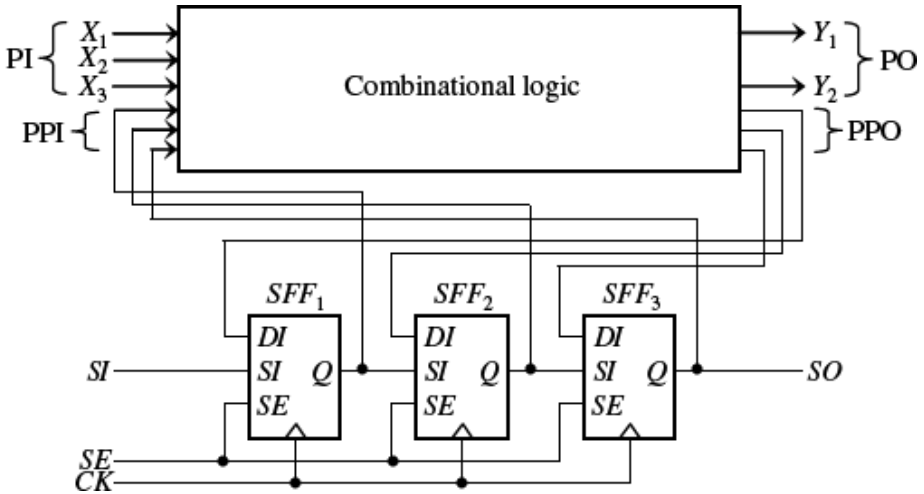
SE = 1 enables scan

Muxed-D Scan Architecture

- Original circuit
  - Single clock
  - Positive edge triggered
  - Non-scan DFF



- After scan insertion
  - Three extra pins
    - SI, SE, SO

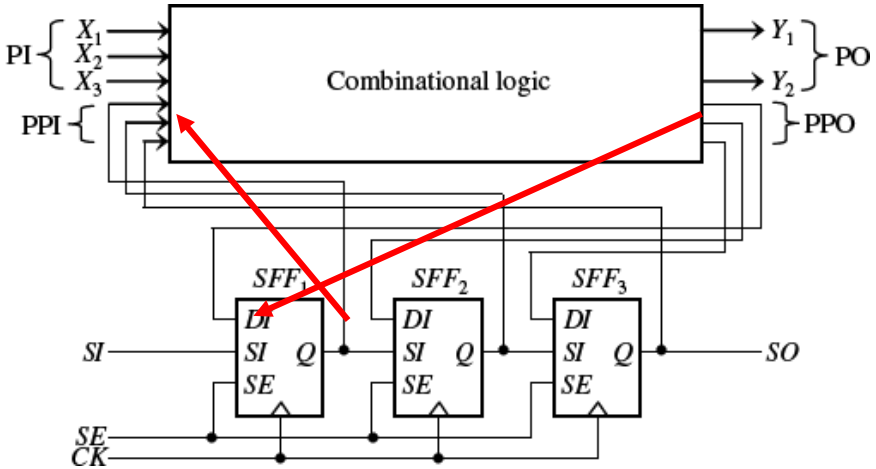


SFF = scan FF

Normal Mode Operation

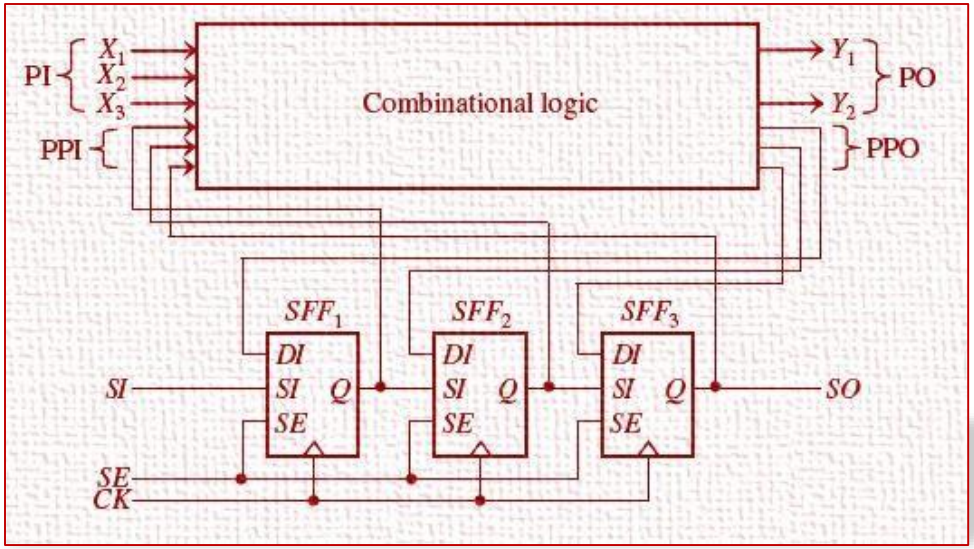
- **Always SE = 0**
- **Same as original circuit**

	operation
SE=0	**Capture (DI→Q)**
SE=1	Shift (SI→SO)



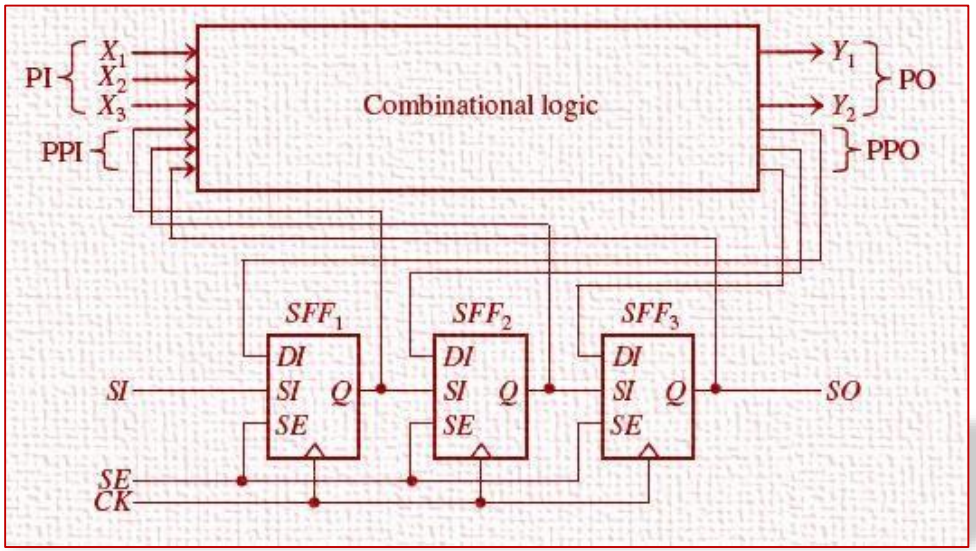
Test Mode Operation

- **Stuck-at fault testing**
  - One pattern per scan
- Delay fault testing
  - Two-patterns per scan
    - Launch-on-Shift
    - Lunch-on-Capture

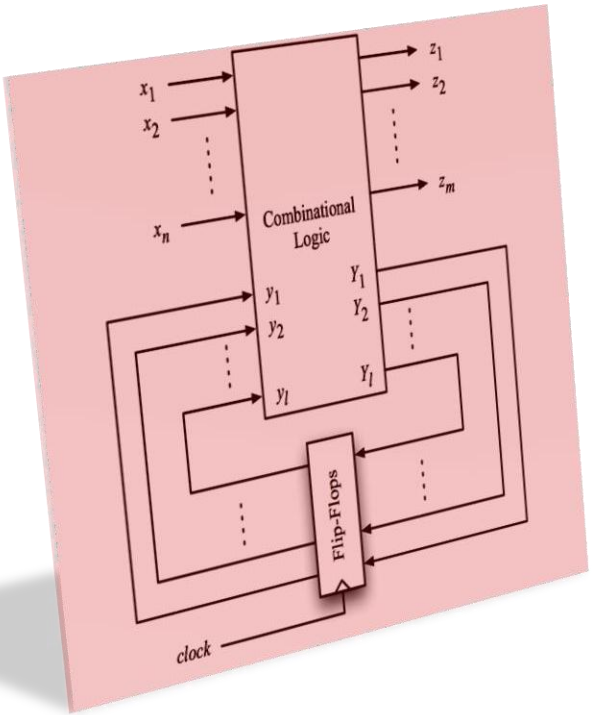


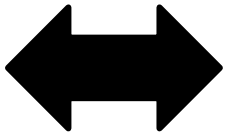
DFT - Part 1

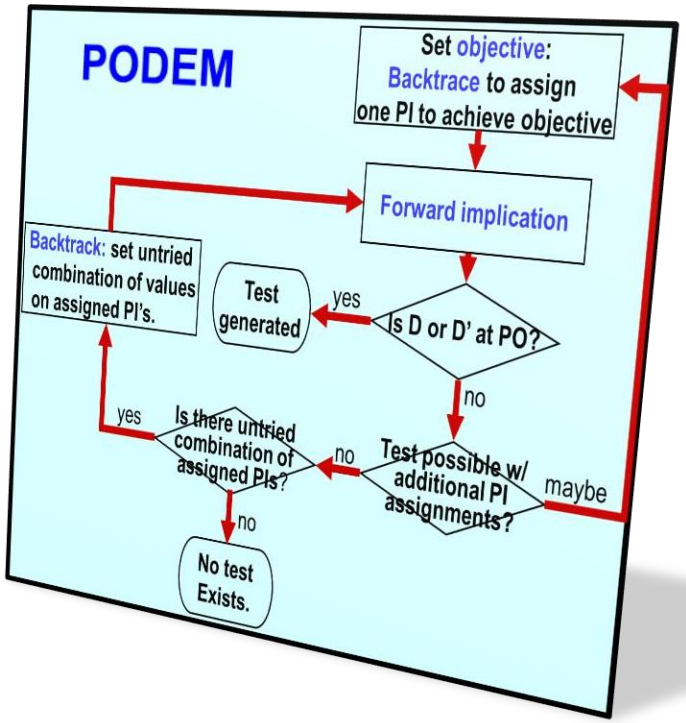
- Introduction
- **Internal Scan**
  - **FF-based**
    - **MUXed-D scan (1973, Stanford)**
      - MUXed-D scan flip-flop
      - Test Mode Operation
      - **Ckt. Model for ATPG**
        - **SSF**
        - **LOS**
        - **LOC**
    - Clocked scan
    - Other scan
  - Latch-based



How to Run Comb. ATPG on Seq. Ckt?

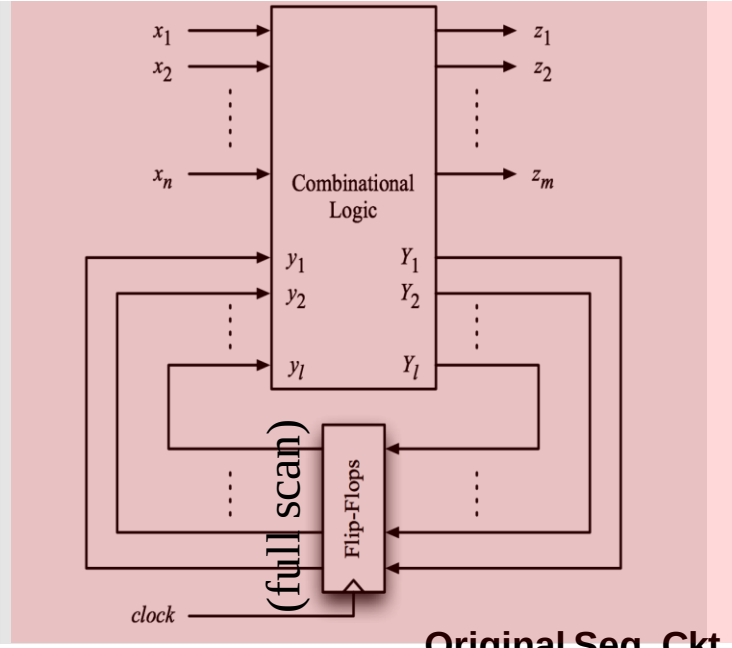




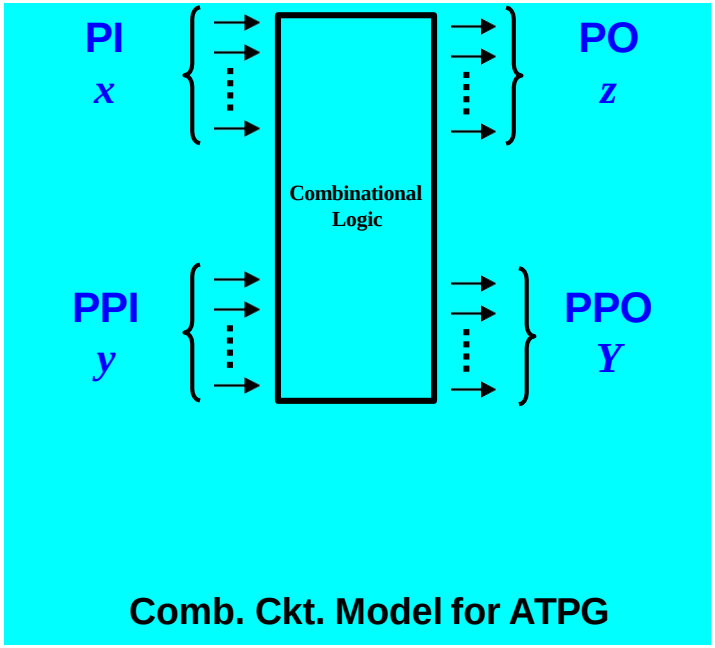


DFT Turns Seq. Ckt. to Comb. Ckt.

- DFT turns sequential ckt into combinational ckt in test mode
  - ATPG sees only comb. ckt. model
- Scan FF become *Pseudo Primary Input* (PPI), fully controllable
- Scan FF become *Pseudo Primary Output* (PPO), fully observable



Original Seq. Ckt.

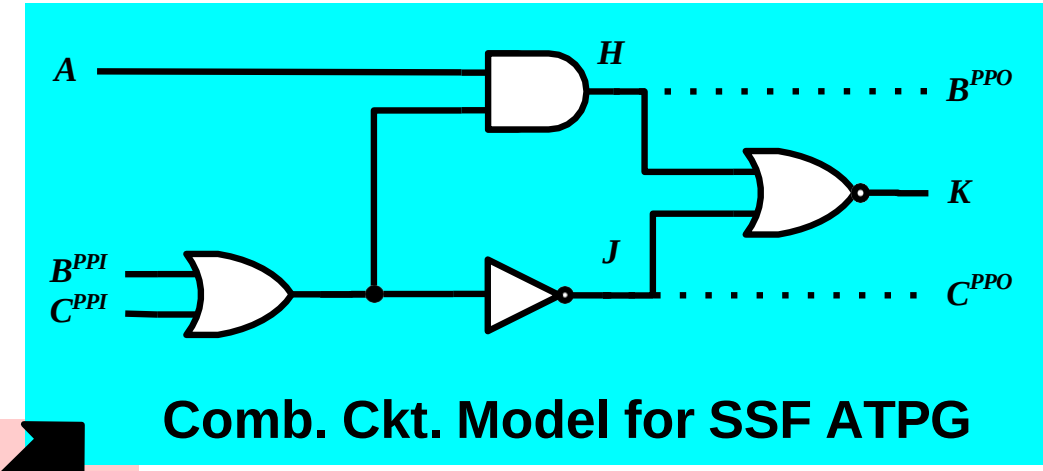


Comb. Ckt. Model for ATPG

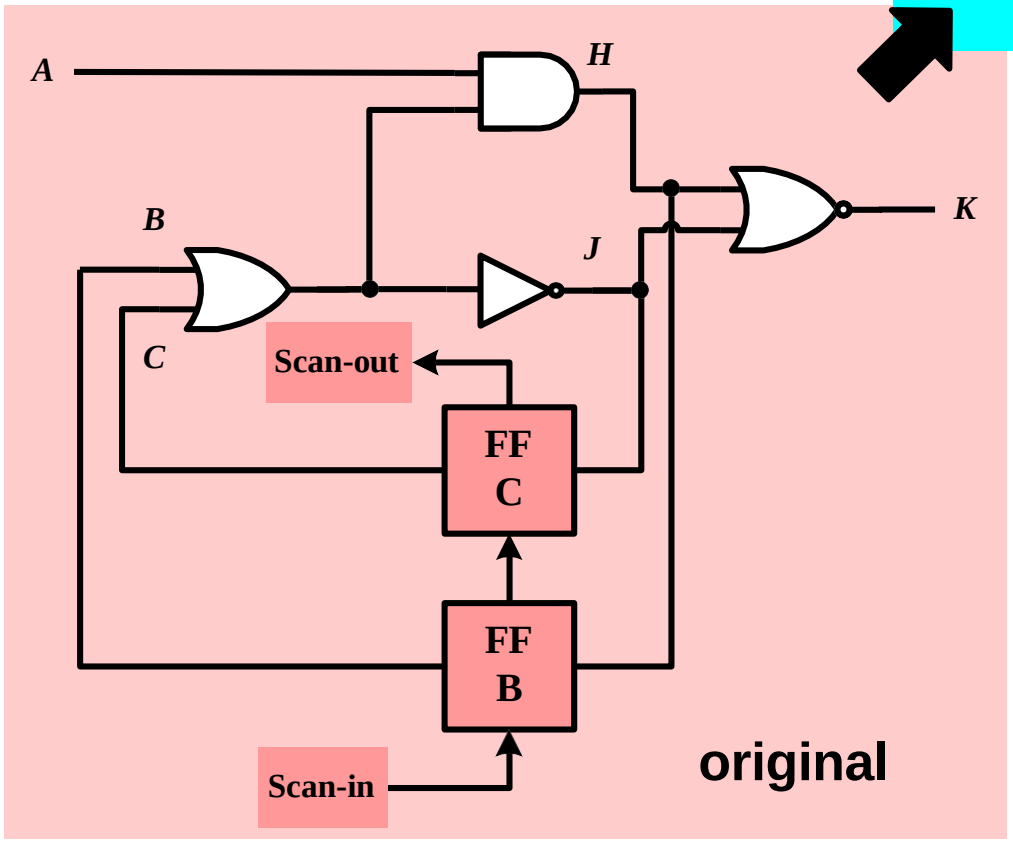
Comb. ATPG Much Faster than Seq. ATPG

Example (1/2)

- Two scan FF in a scan chain
- SI→FF-B →FF-C →SO



Comb. Ckt. Model for SSF ATPG



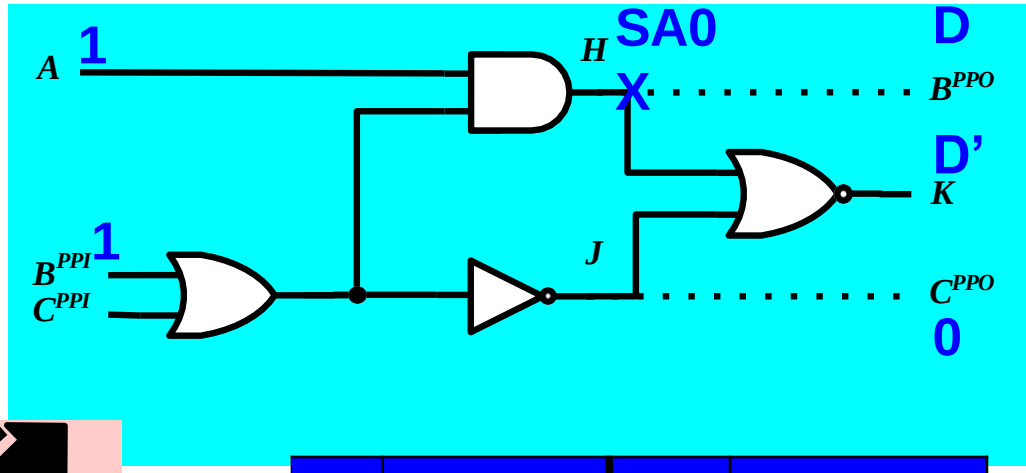
original

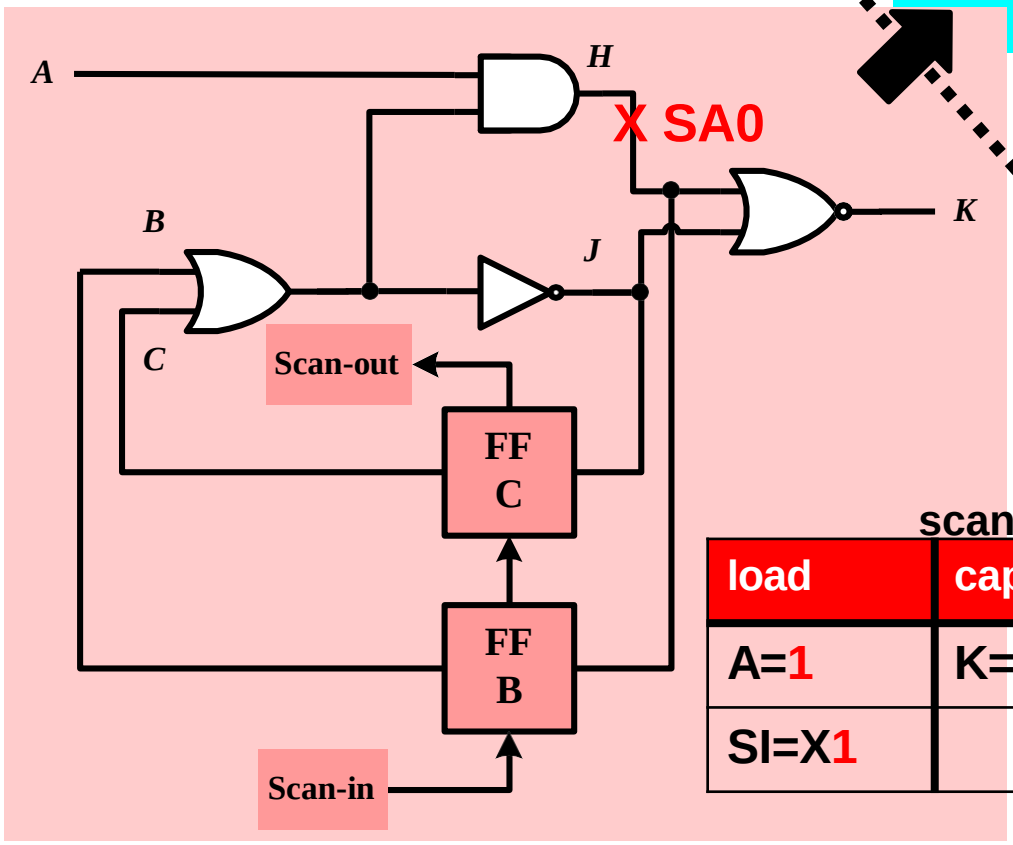
- A is PI
- B,C are PPI
- K is PO
- HJ are PPO
  - $H=B^{PPO}$, $J=C^{PPO}$

NOTE: this model assume no fault in FF. (see FFT)

Example (2/2)

- Generate SSF ATPG pattern
  - H SA0 fault





PI	PPI	PO	PPO
A	$B^{PPI}C^{PPI}$	K	$B^{PPO}C^{PPO}$
1	1 X	D'	D 0

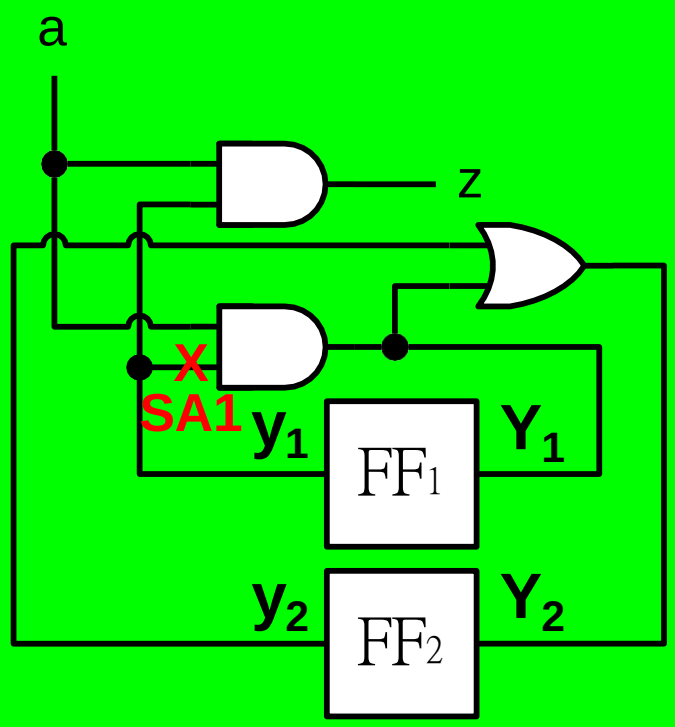
comb. pattern

scan pattern

load	capture	unload
A=1	K=L	
SI=X1		SO=LH

Quiz

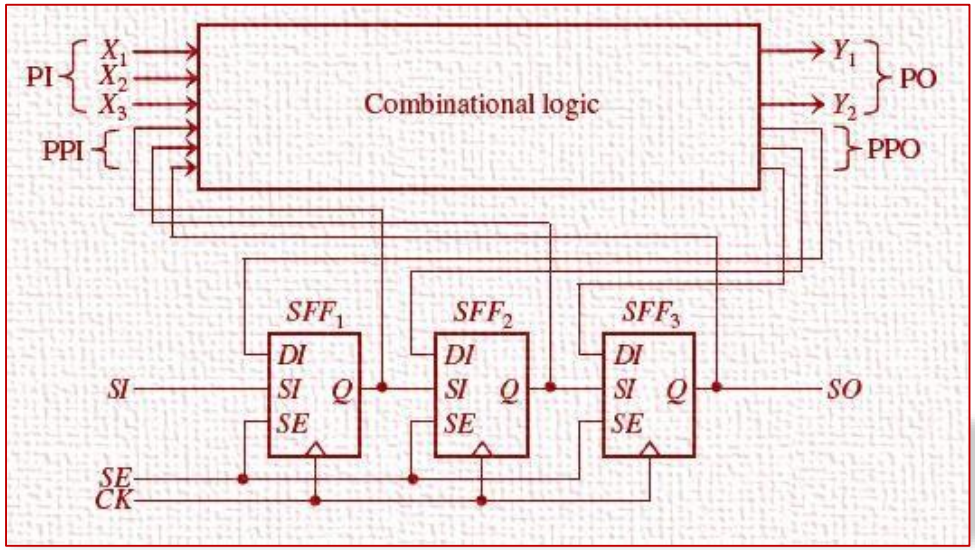
Q1: Convert seq. ckt. into comb. ckt. model for SSF ATPG.
Q2: Generate a test pattern for SA1 fault.
(show comb. pattern)



PI	PPI	PO	PPO
a	y_1 y_2	z	Y_1 Y_2

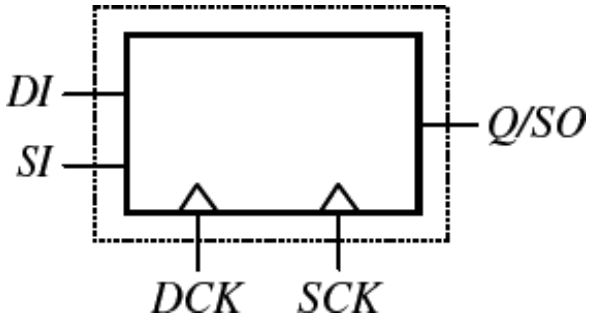
DFT - Part 1

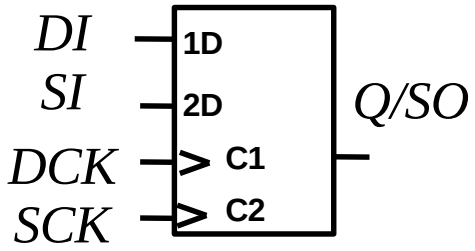
- Introduction
- **Internal Scan**
  - **FF-based**
    - MUXed-D scan (1973, Stanford)
    - **Clocked scan (1968, 1975 NEC)**
    - **Other scan**
  - Latch-based
    - LSSD (1977, IBM)
- Scan Design Flow
- Issues and Solutions
- Conclusion



Clocked Scan FF [Kobayashi 68][Funatsu 75]

- 2 inputs
  - Data in (DI): from logic
  - Scan in (SI): from previous scan FF
- 2 independent clocks
  - Scan clock (SCK): capture SI
  - Data clock (DCK): capture DI
- Data output (Q) and Scan Output (SO) share same pin

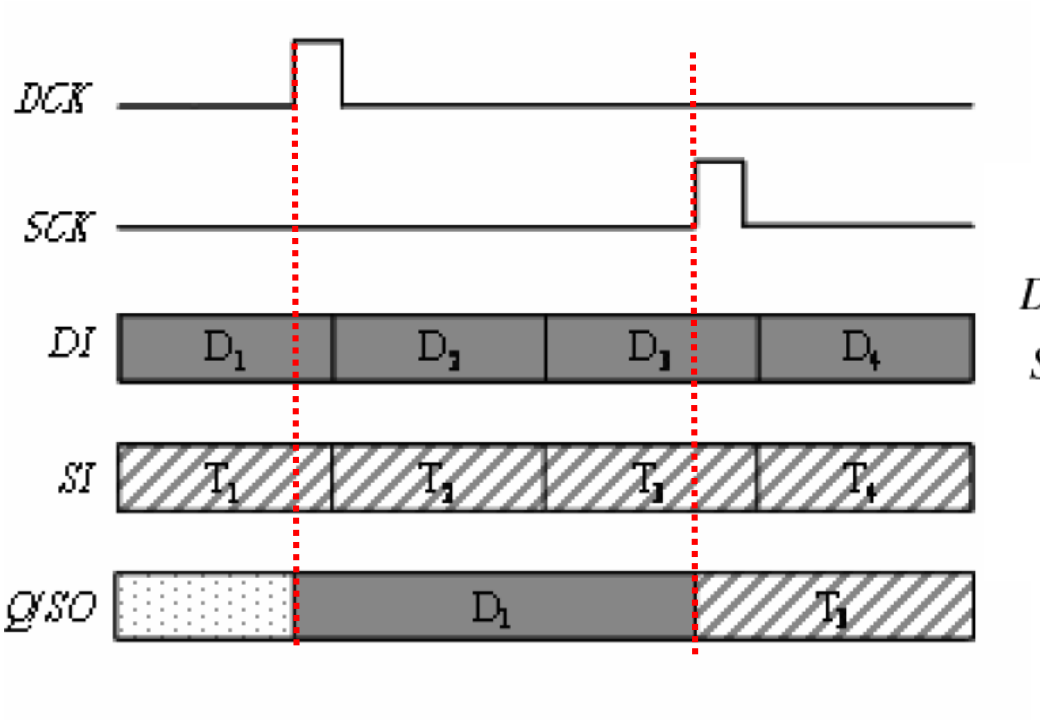


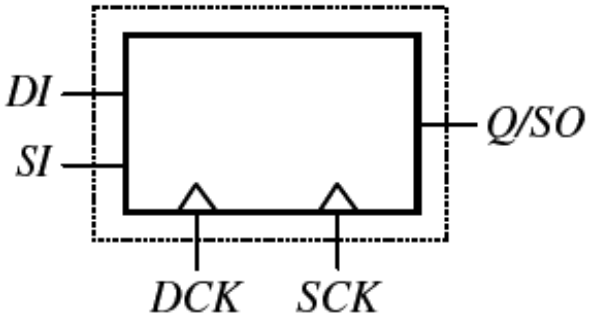


IEEE symbol

Clocked Scan FF (2)

- DI → Q at positive DCK edge
- SI → Q at positive SCK edge



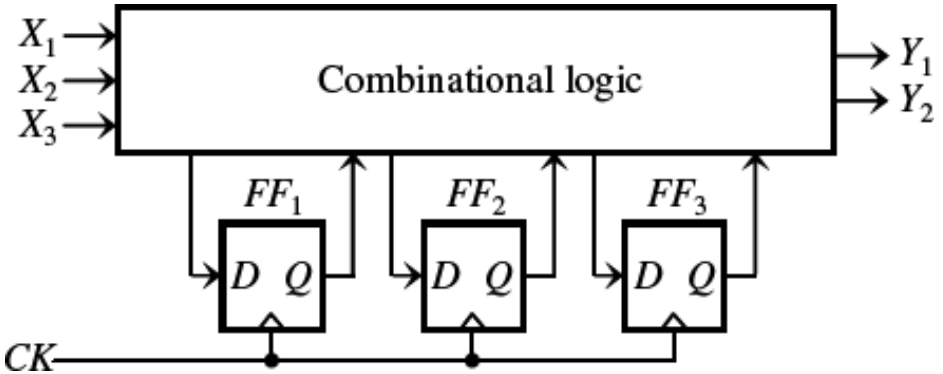


(WWW. Fig 2.11)

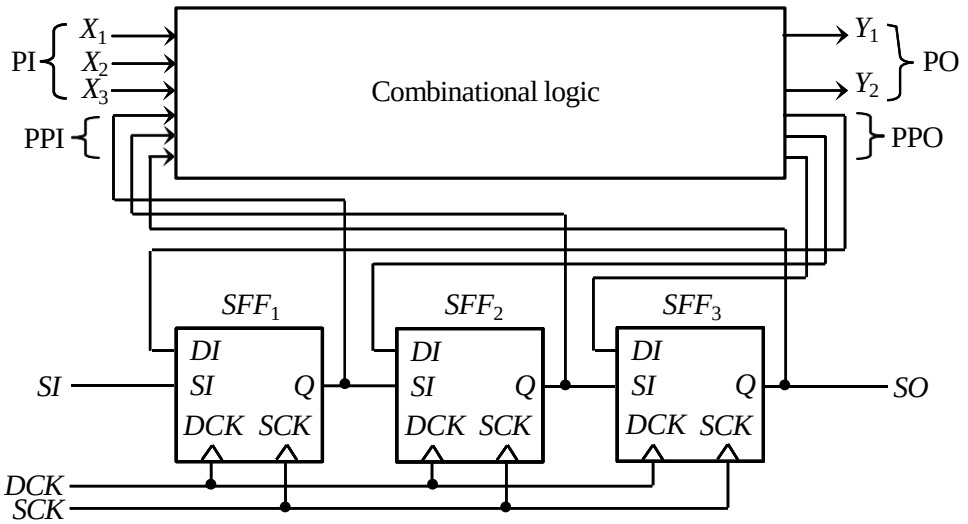
SCK/DCK Do NOT Overlap

Clocked Scan Architecture

- **Original circuit**
  - **Single clock: CK**
  - **Regular D-FF**

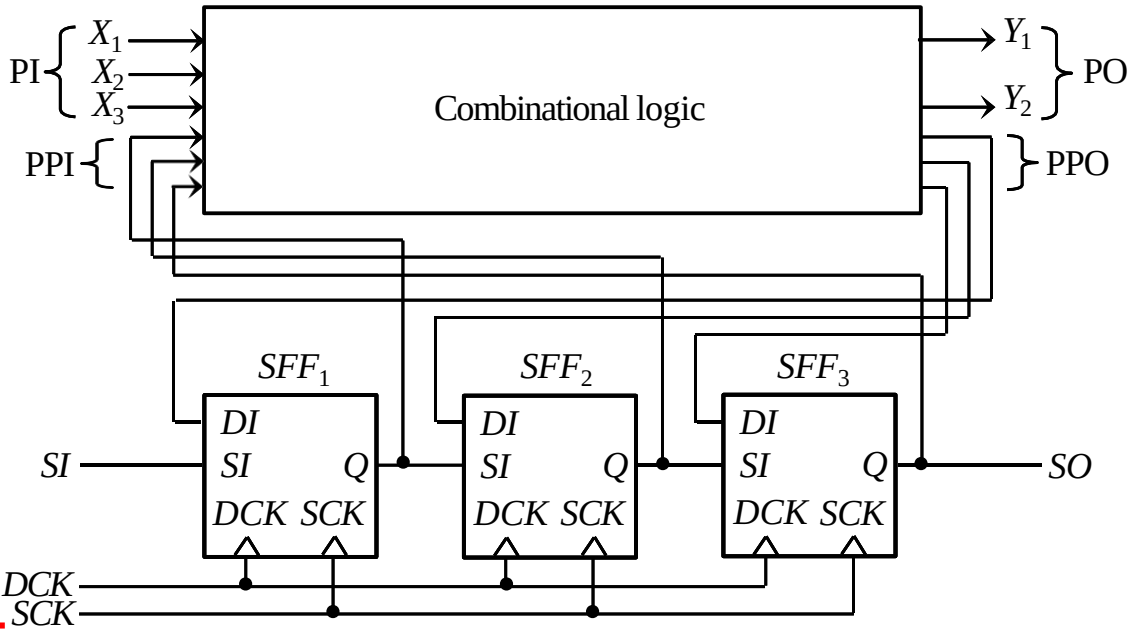


- **After scan insertion**
  - **Two extra I/O pins**
    - **SI, scan input**
    - **SO, scan output**
  - **One extra clock**
    - **SCK**



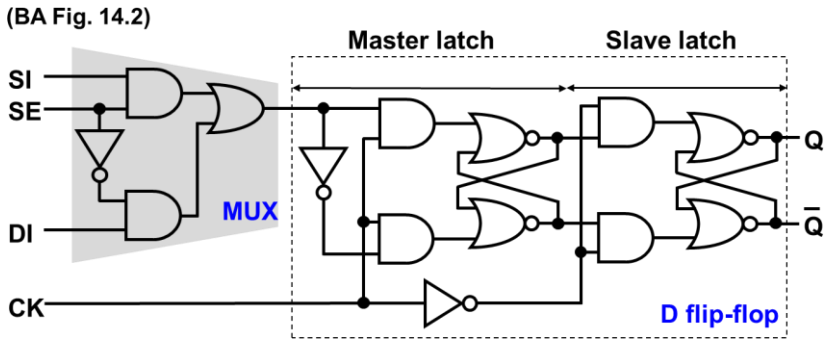
Clocked Scan Operation

- **Normal Mode**
  - **DCK, DCK …**
- **Test Mode**
  - **Shift: SCK, SCK, SCK (load scan chain)**
  - **Capture: DCK**
  - **Shift: SCK, SCK, SCK (unload scan chain)**

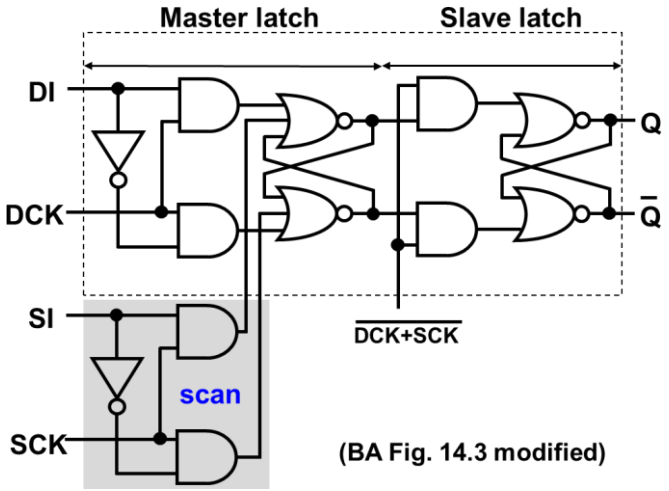


Pros/Cons of Clocked Scan

- Advantage
  - Faster than MUXed-D scan
    - Less delay overhead
- Disadvantage
  - Larger routing overhead than MUXed-D scan
    - Needs one extra clock distribution (SCK)



MUXed-D scan



Clocked scan

CS Faster but Larger

Quiz

Q: Which of the following is NOT true about clocked scan?

A. Clocked scan is better than MUXed-D scan

B. Clocked scan has two clocks

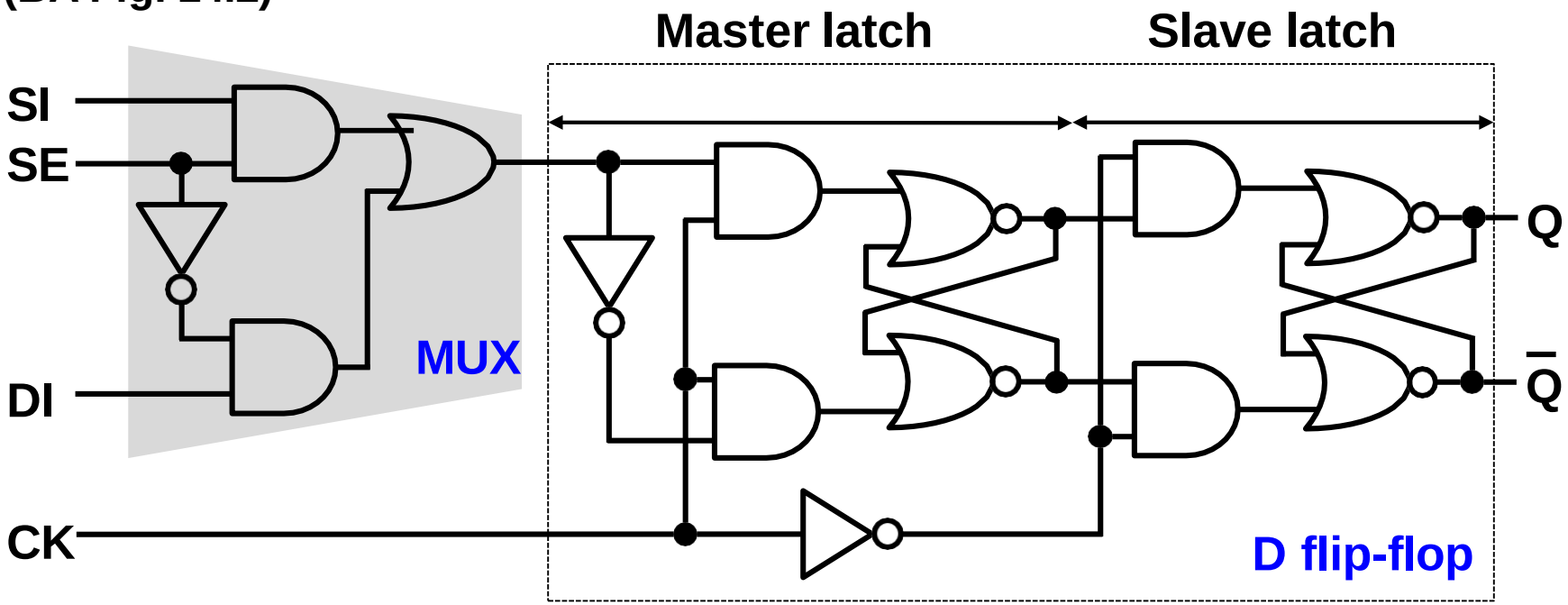
C. Clocked scan is faster than MUXed-D scan

D. Clocked scan is useful for high speed circuits

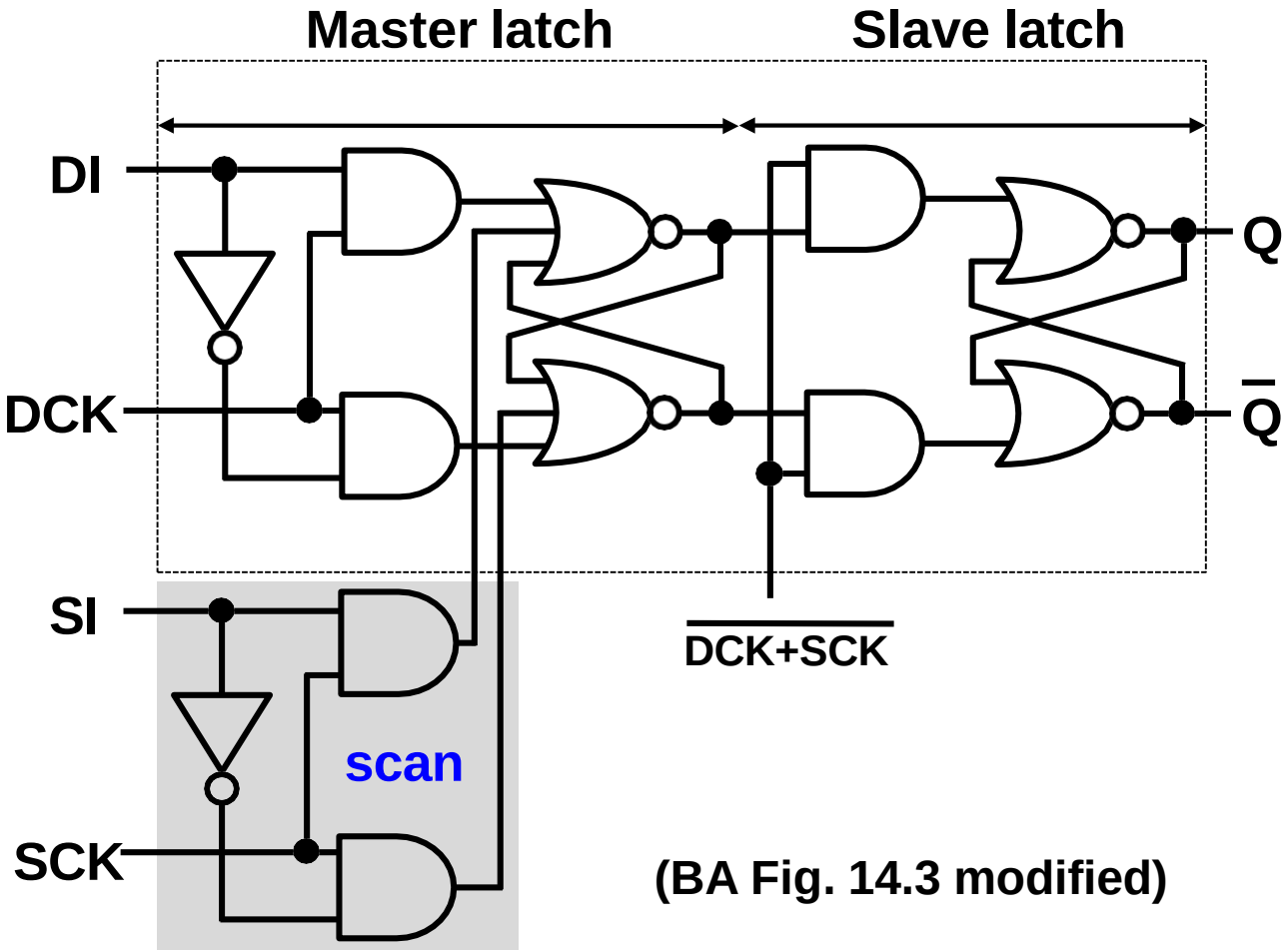
ANS:

APPENDIX: Schematic of MUXed-D Scan

(BA Fig. 14.2)



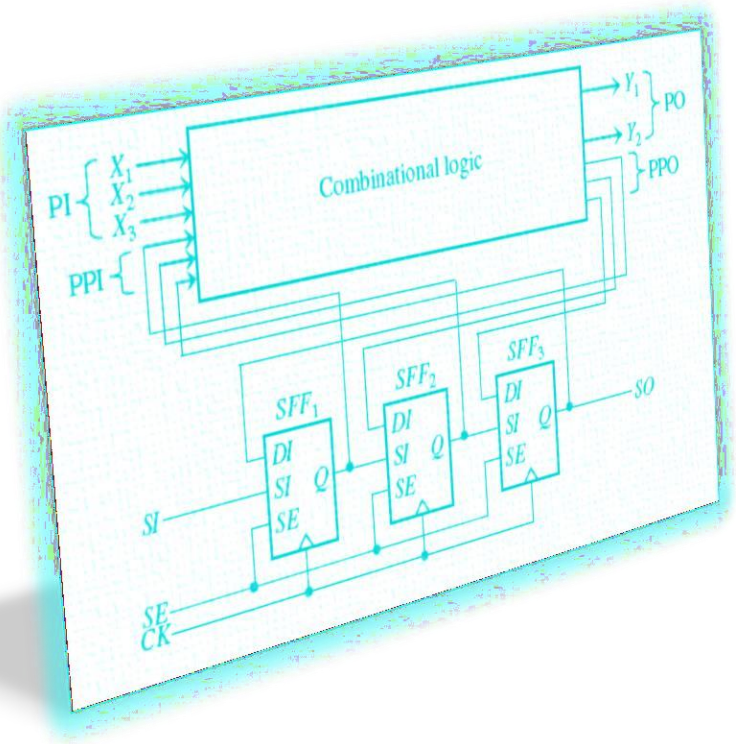
APPENDIX: Schematic of Clocked Scan



(BA Fig. 14.3 modified)

DFT - Part 1

- Introduction
- **Internal Scan**
  - FF-based
    - MUXed-D scan (1973, Stanford)
    - Clocked scan (1968, NEC)
    - Other scan
  - **Latch-based**
    - **LSSD (1977, IBM)**
- Scan Design Flow
- Issues and Solutions
- Conclusion



Reference

- [Eichelberger 77] E.B. Eichelberger, and T.W. Williams, "A logic design structure for LSI testability." Proc. of Design Auto. Conf., 1977. (IBM)
- [Funatsau 75] S. Funatsu, N. Wakatsuki, and T. Arima, "Test generation systems in Japan," Proc. Design Auto. Symp., 1975. (NEC)
- [Kobayashi 68] A. Kobayashi, S Matsue, H Shiba “Flip-flop circuit with FLT capability,” Proc. of IECEO conf., 1968. (NEC)
- [McCluskey 86] E. J. McCluskey , *Logic Design Principles*, 1986
- [Williams 73] M.J.Y. Williams, J. B. Angell, “Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic,” IEEE Trans. on Comput., Vol. C-22, Issue: 1, 1973. (Stanford)