



# VLSI Testing

## 積體電路測試

### *Design For Testability (DFT)*

### *Part I: Internal Scan Chains*

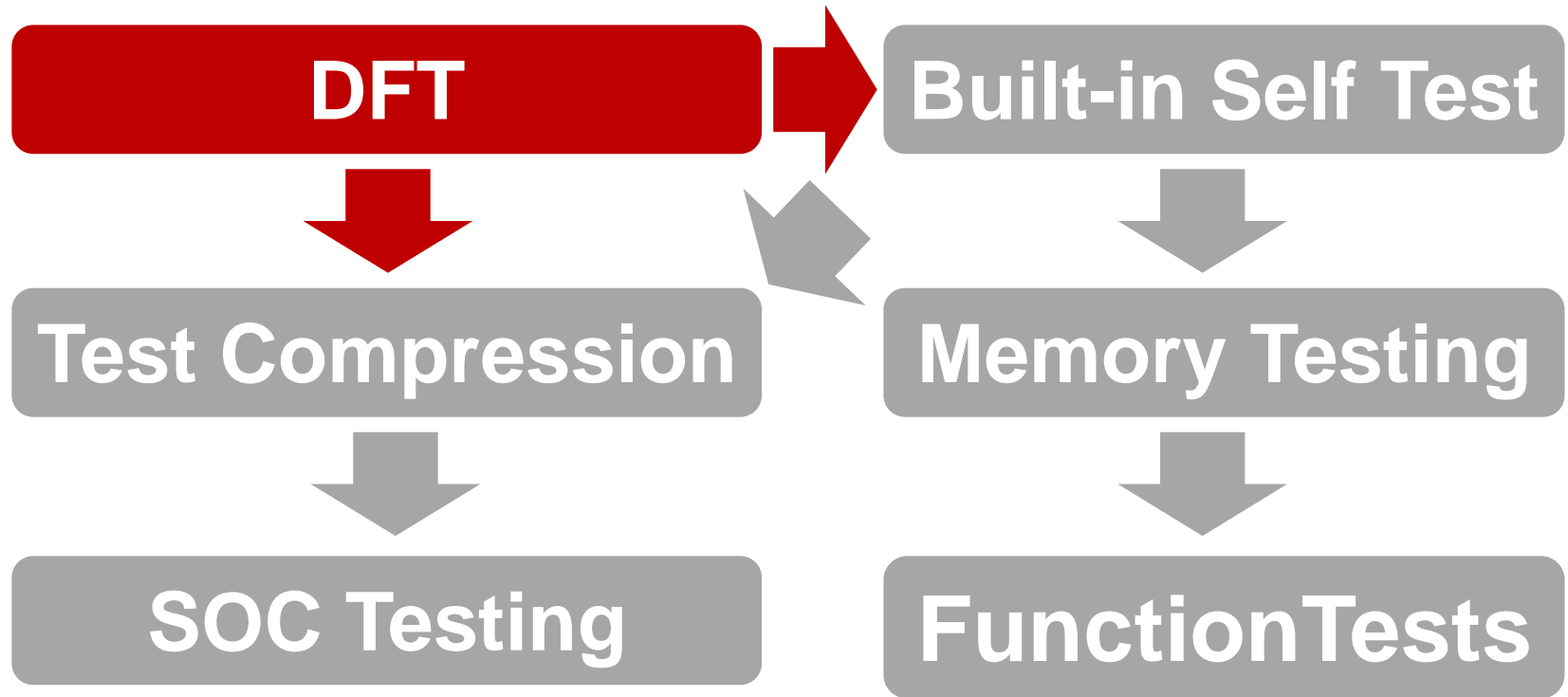
**Professor James Chien-Mo Li 李建模**

**Lab. of Dependable Systems**

**Graduate Institute of Electronics Engineering**

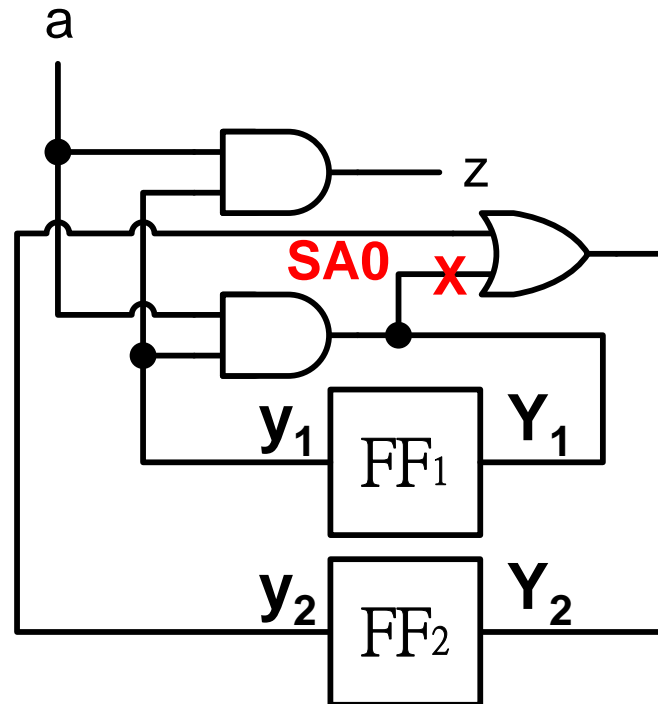
**National Taiwan University**

# Course Roadmap (Design Topics)



# 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?



# Why Am I Learning This?

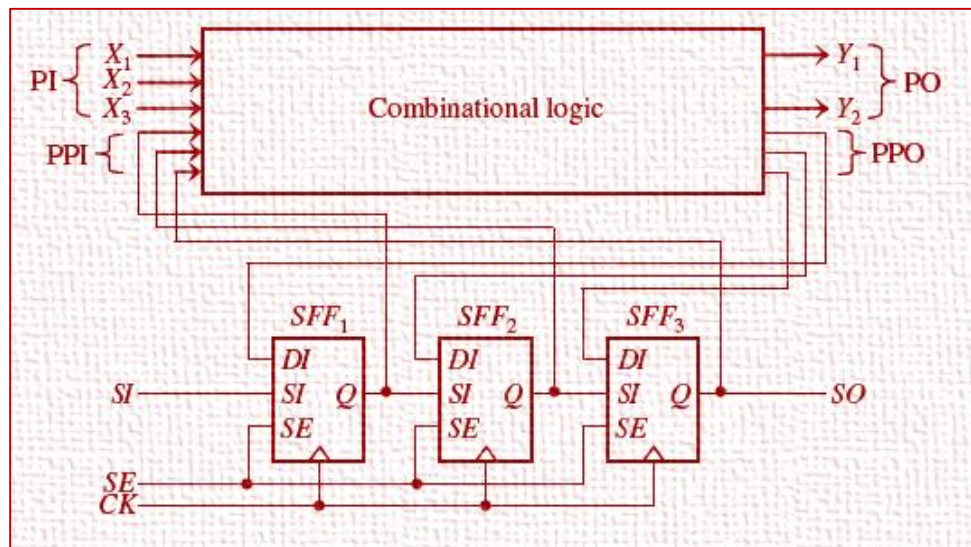
- Design for Testability
  - ◆ Make ATPG easier
  - ◆ Improve fault coverage (test quality)
  - ◆ Reduce test length (test cost)

*“Good design is obvious.  
Great design is transparent.”*

*(Joe Sparanolf )*

# DFT Outline

- Part 1
  - ◆ Introduction
  - ◆ Internal Scan
    - \* FF-based: MUX-scan, clock scan
    - \* Latch-based: LSSD
- Part 2
  - ◆ External Scan
    - \* JTAG (1149.1)



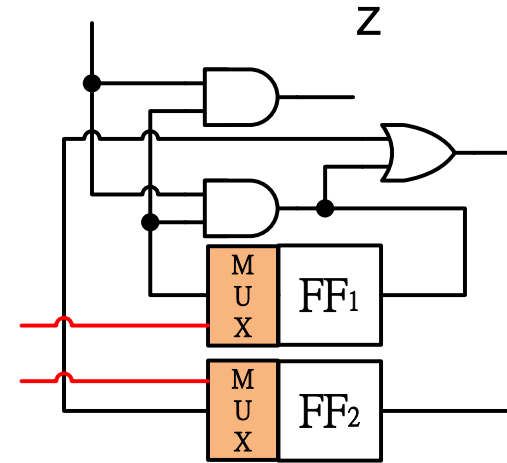
# Design for Testability (DFT)

- 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:**



# DFT - Part 1

- Part 1

- ◆ Introduction

- \* Ad-hoc DFT

- \* Test point insertion

- ◆ Internal Scan

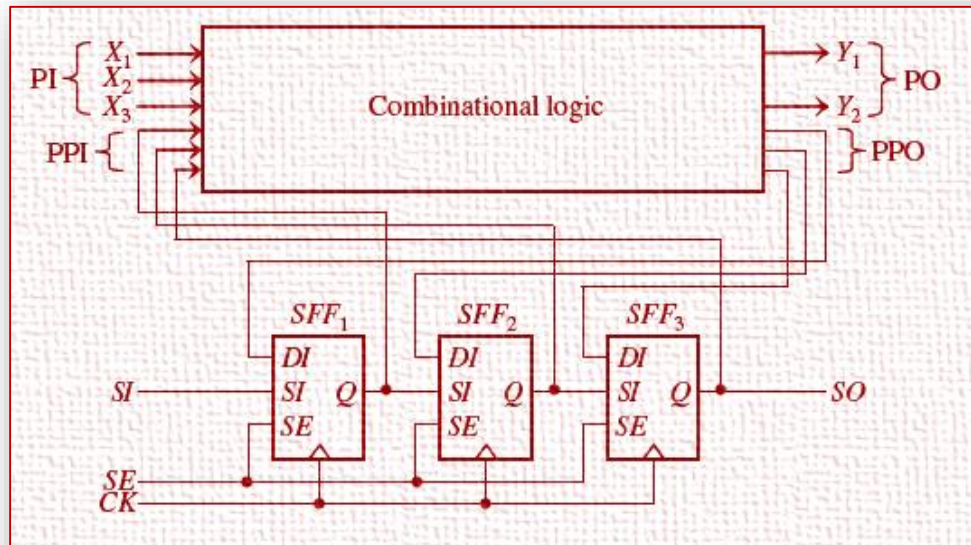
- \* FF-based

- \* Latch-based

- Part 2

- ◆ External Scan

- \* JTAG (1149.1)

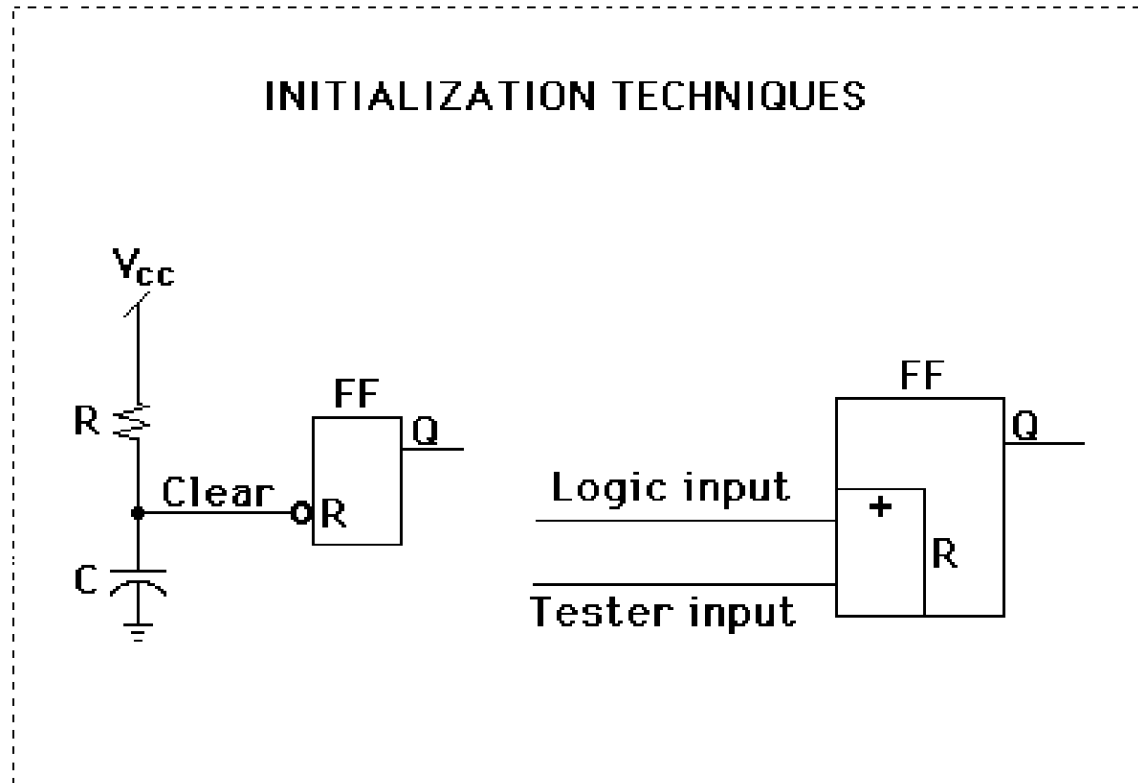


# Ad Hoc DFT

- What is “Ad hoc” ?
  - ◆ for the particular end or case at hand without consideration of wider application [Webster dictionary]
- 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)

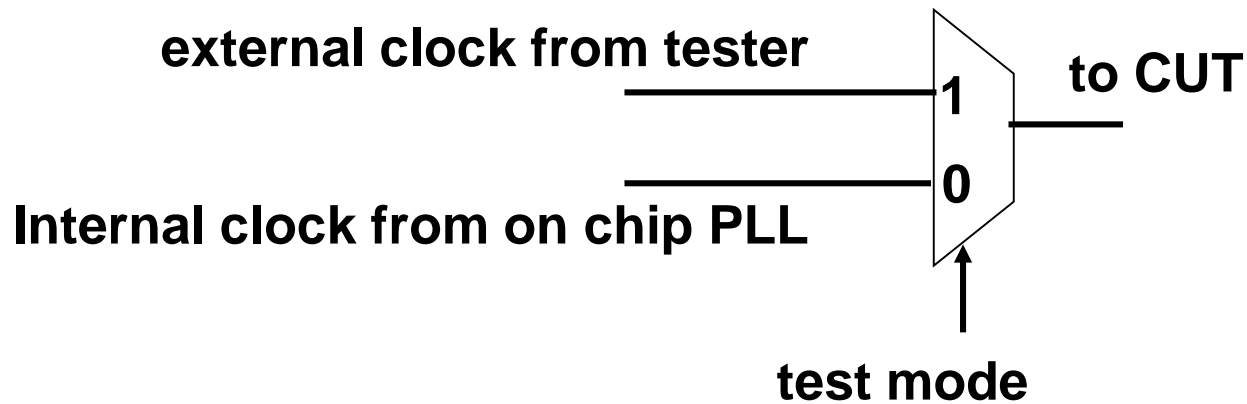
- Rule1: design circuits easily initializable
  - ◆ Self-initialization, or
  - ◆ Initializable from tester



Courtesy Prof. McClueksy, Stanford University

# Ad Hoc DFT Example (2)

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



# DFT - Part 1

- Part 1

- ◆ Introduction

- \* Ad-hoc DFT

- \* Test point insertion

- ◆ Internal Scan

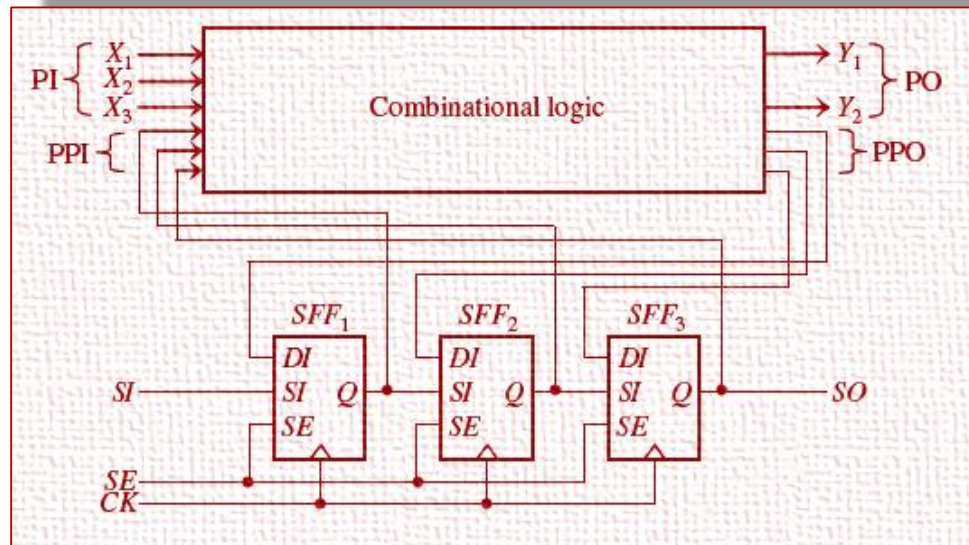
- \* FF-based

- \* Latch-based

- Part 2

- ◆ External Scan

- \* JTAG (1149.1)



# Test Point Insertion

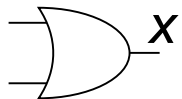
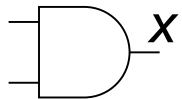
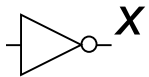
- Objective:
  - ◆ Insert test points to enhance
    - \* Controllability
    - \* Observability
- Two types of test points
  - ◆ *Control points*: enhance controllability.
  - ◆ *Observation points*: enhance observability

# Control Points

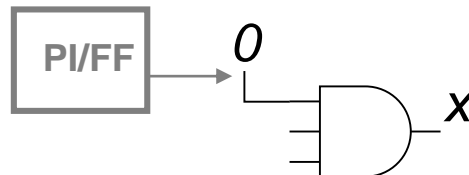
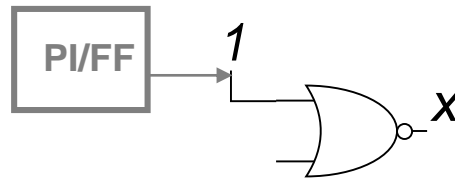
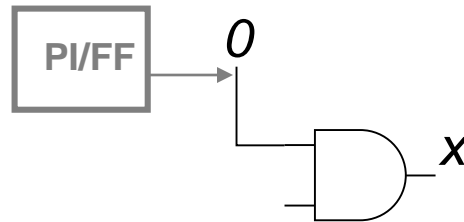
- Control signals come from primary input (PI) or scan FF
  - Details see next video

original

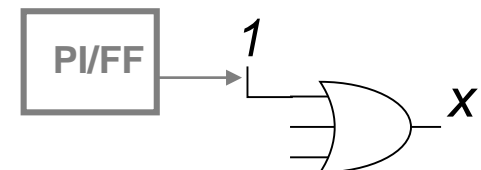
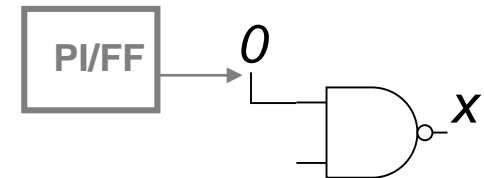
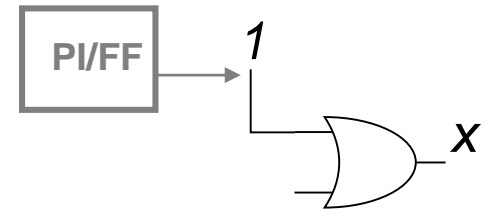
$x$



control  $x$  to 0

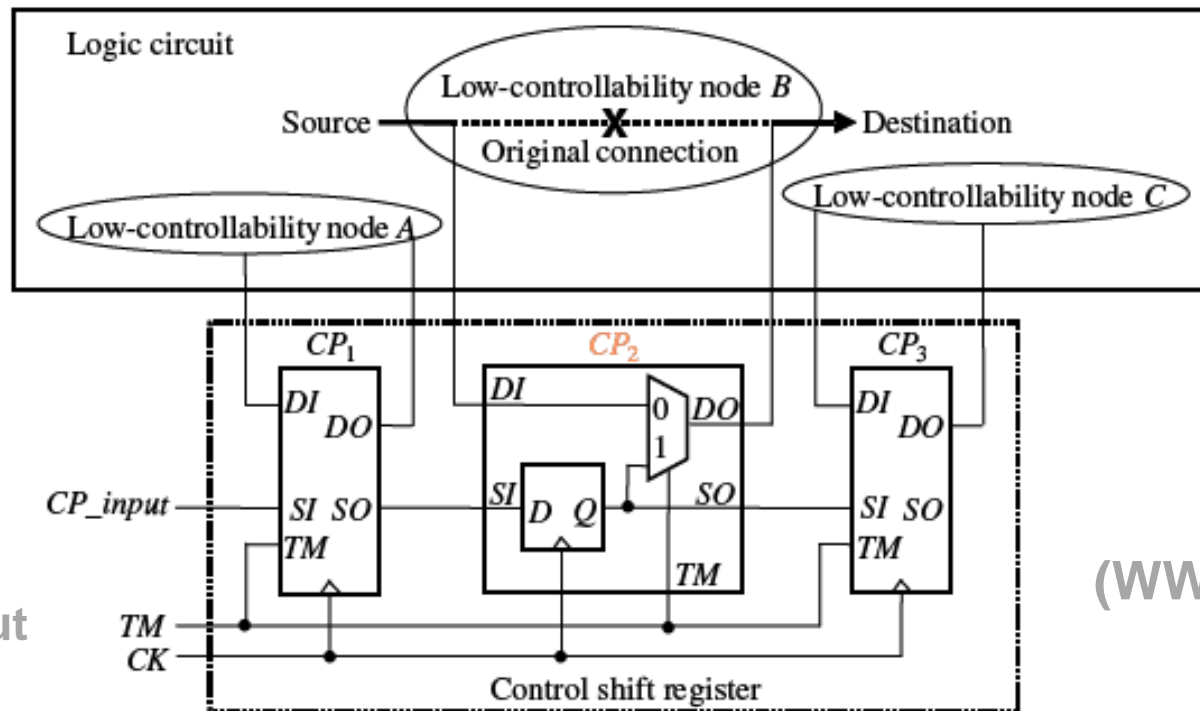


control  $x$  to 1



# Control Point (2)

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



(WWW Fig. 2.5)

Control point insertion

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

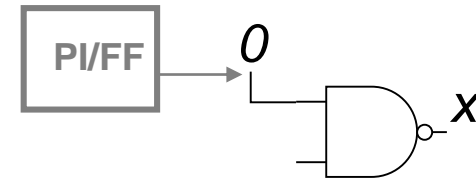
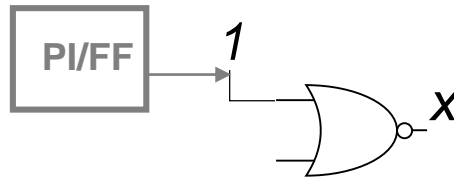
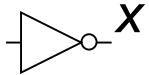


# Quiz

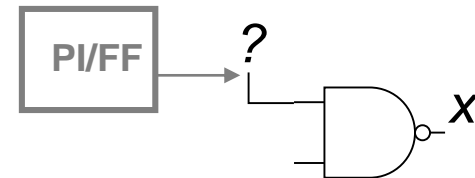
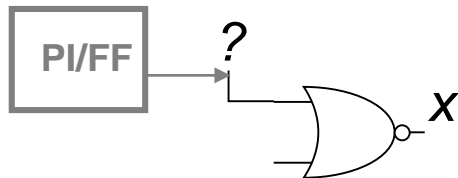
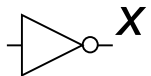
**Q: What are control values in NORMAL mode?**

**ANS:**

**Test mode**

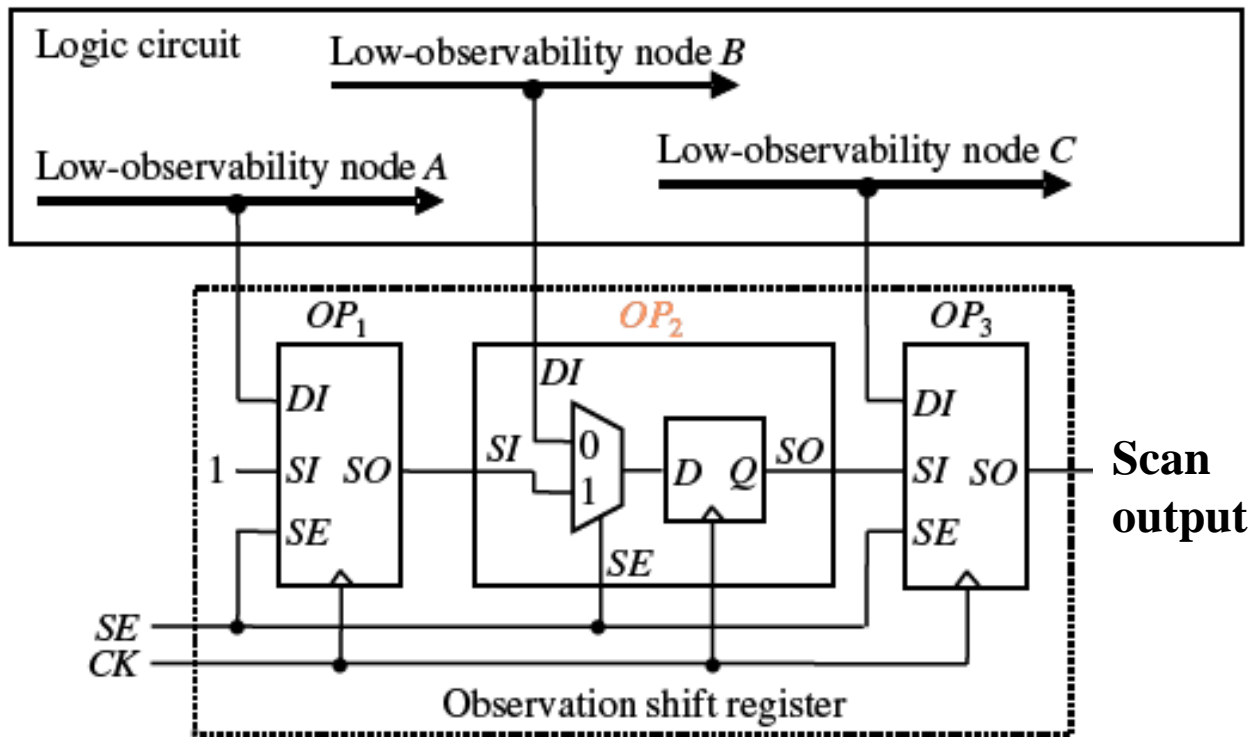


**Normal mode**



# Observation Points

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

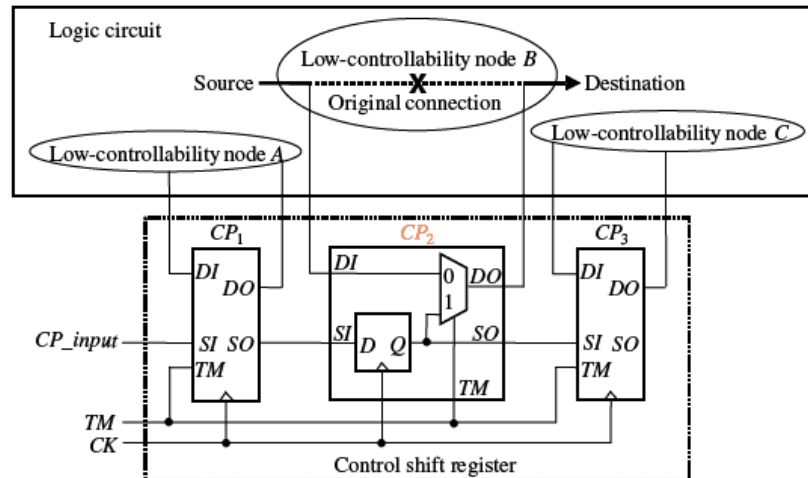


DI=date input  
DO=data output  
SE=scan enable  
SI=scan input  
SO=scan output  
(WWW Fig. 2.6)

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

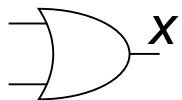
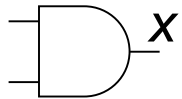
- Introduction to DFT
  - ◆ DFT helps to improve **controllability/observability**
  - ◆ **Ad-hoc DFT** based on experience
    - \* Advantage
      - Suitable for your particular design under test
    - \* Disadvantage
      - Not systematic → **lacks EDA tool**
      - As opposed to **structured DFT**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

**Structured DFT is Preferred than Ad Hoc DFT**

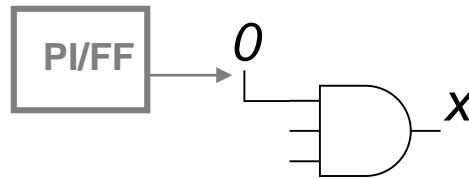
# FFT

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

original



control x to 0



?

control x to 1

?

