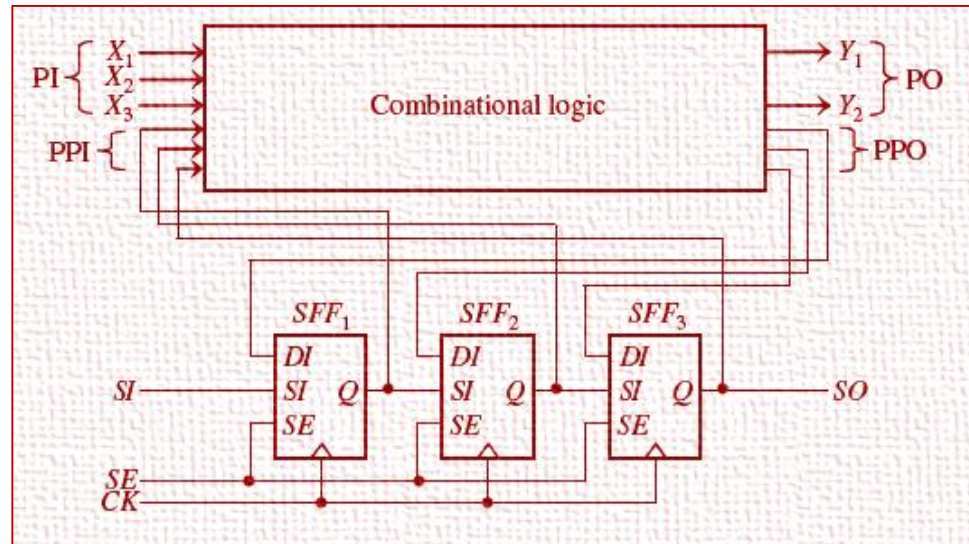
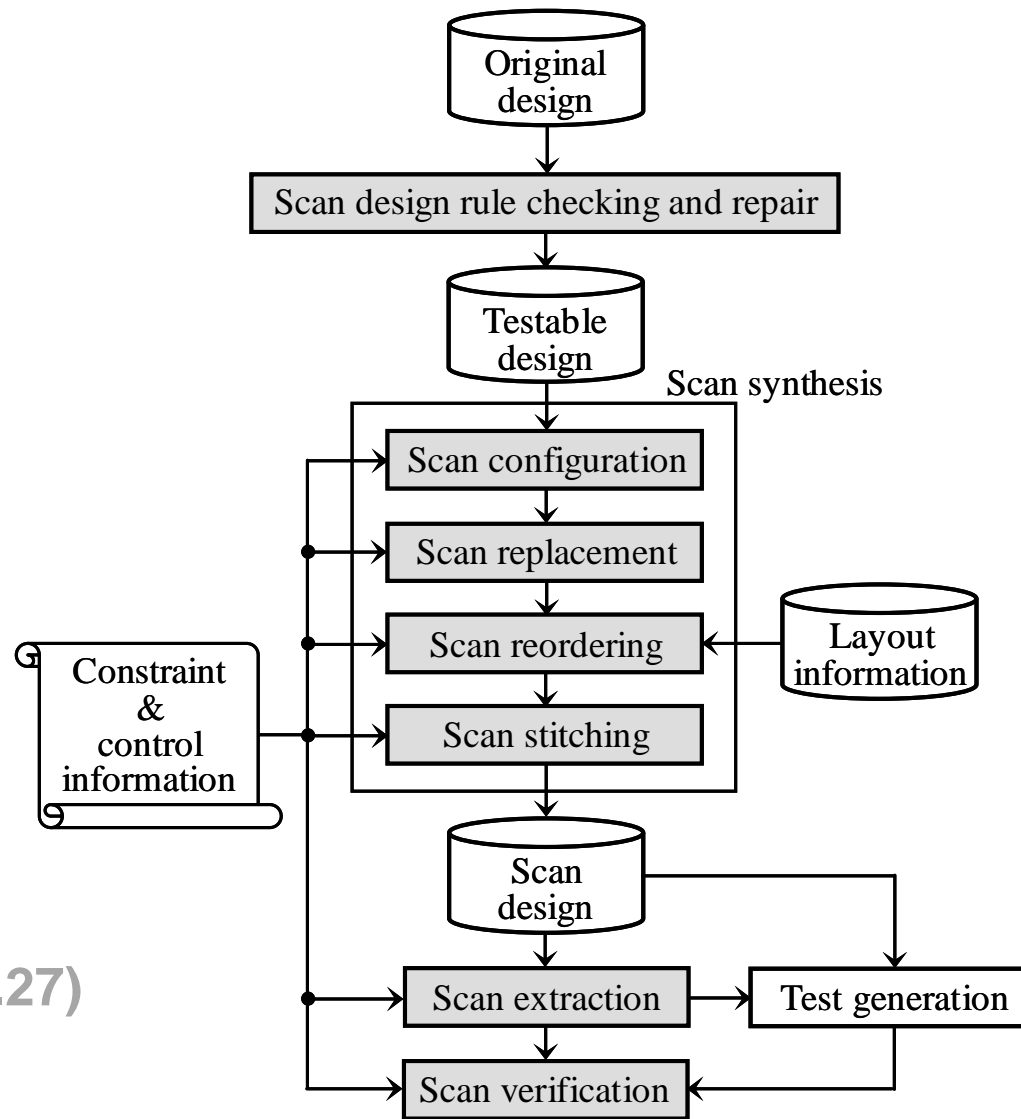


# DFT - Part 1

- Introduction
- Internal Scan
  - ♦ FF-based
  - ♦ Latch-based
- **Scan Design Flow**
- Issues and Solutions
- Conclusion



# Scan Design Flow



(WWW Fig. 2.27)

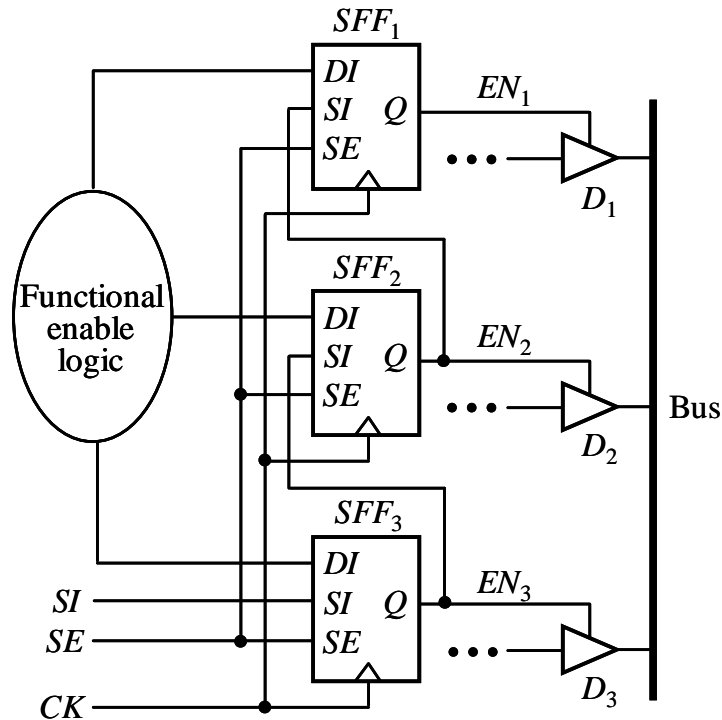
# Scan Design Rule Checking

- **DRC: Make sure designs obey rules so that scan chains can be inserted**
  - ◆ **Tools now supports automatic fix violations after DRC**
- **Typical scan design rules (WWW Table 2.7)**

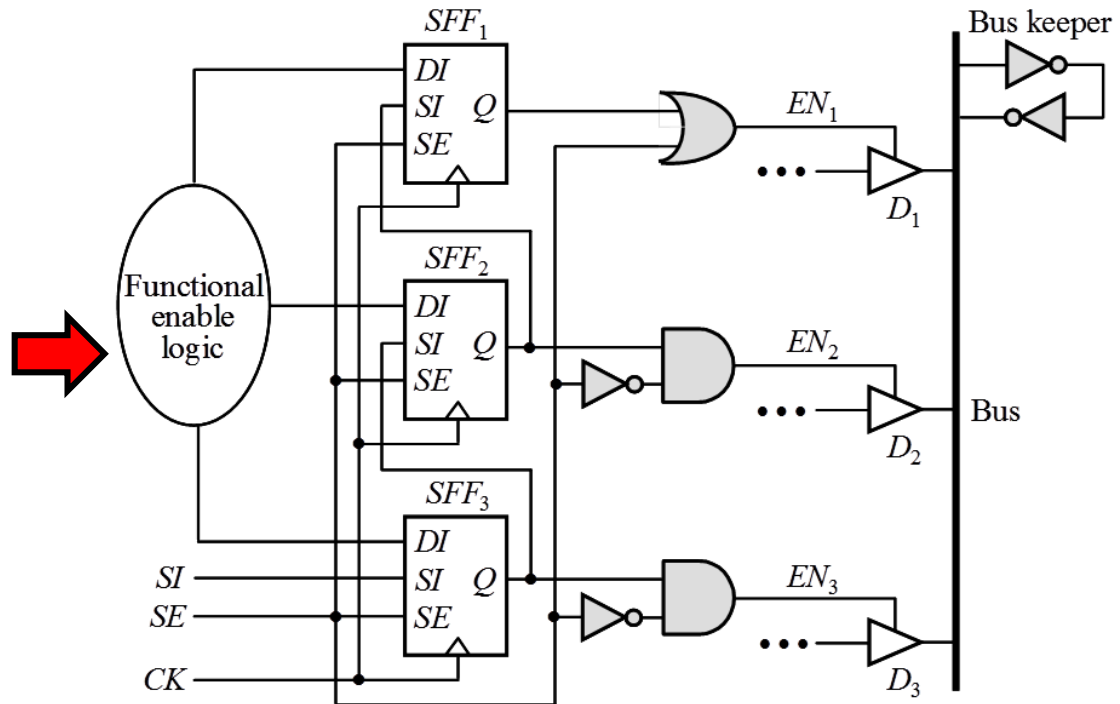
	<i>Design Style</i>	<i>Scan Design Rule</i>	<i>Recommended Solution</i>
<b>DRC#1</b>	Tri-state buses	Avoid during shift	Fix bus contention during shift
<b>DRC#2</b>	Bi-directional I/O ports	Avoid during shift	Force to input or output mode during shift
<b>DRC#3</b>	Gated clocks (muxed-D full-scan)	Avoid during shift	Enable clocks during shift
	Derived clocks (muxed-D full-scan)	Avoid	Bypass clocks
<b>DRC#4</b>	Combinational feedback loops	Avoid	Break the loops
	Asynchronous set/reset signals	Avoid	Use external pin(s)
	Clocks driving data	Avoid	Block clocks to the data portion
	Floating buses	Avoid	Add bus keepers
	Floating inputs	Not recommended	Tie to Vcc or ground
	Cross-coupled NAND/NOR gates	Not recommended	Use standard cells
	Non-scan storage elements	Not recommended for full-scan Design	Initialize to known states, bypass, or make transparent

# DRC#1: Tri-State Buses

- Problem: potential bus contention during scan shift



Original design

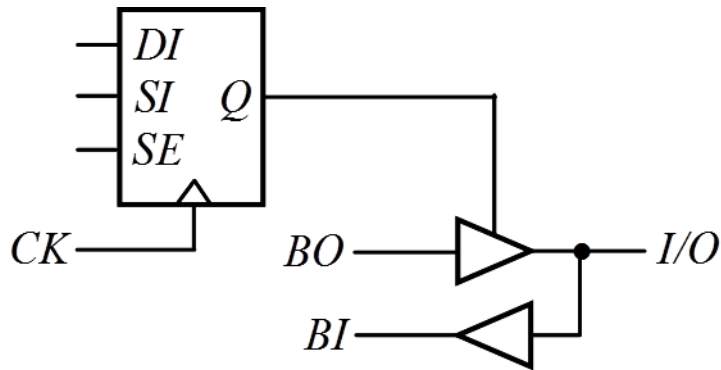


Testable design

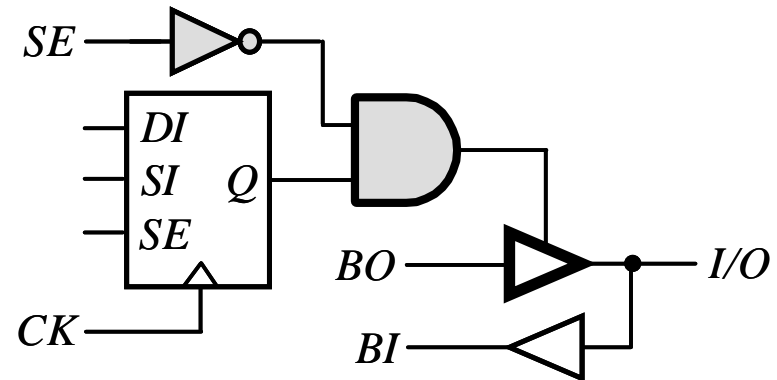
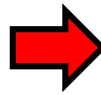
No bus contention when  $SE=1$   
No floating bus

# DRC#2: Bidirectional I/O

- Problem: random I/O direction during scan shift



Original design

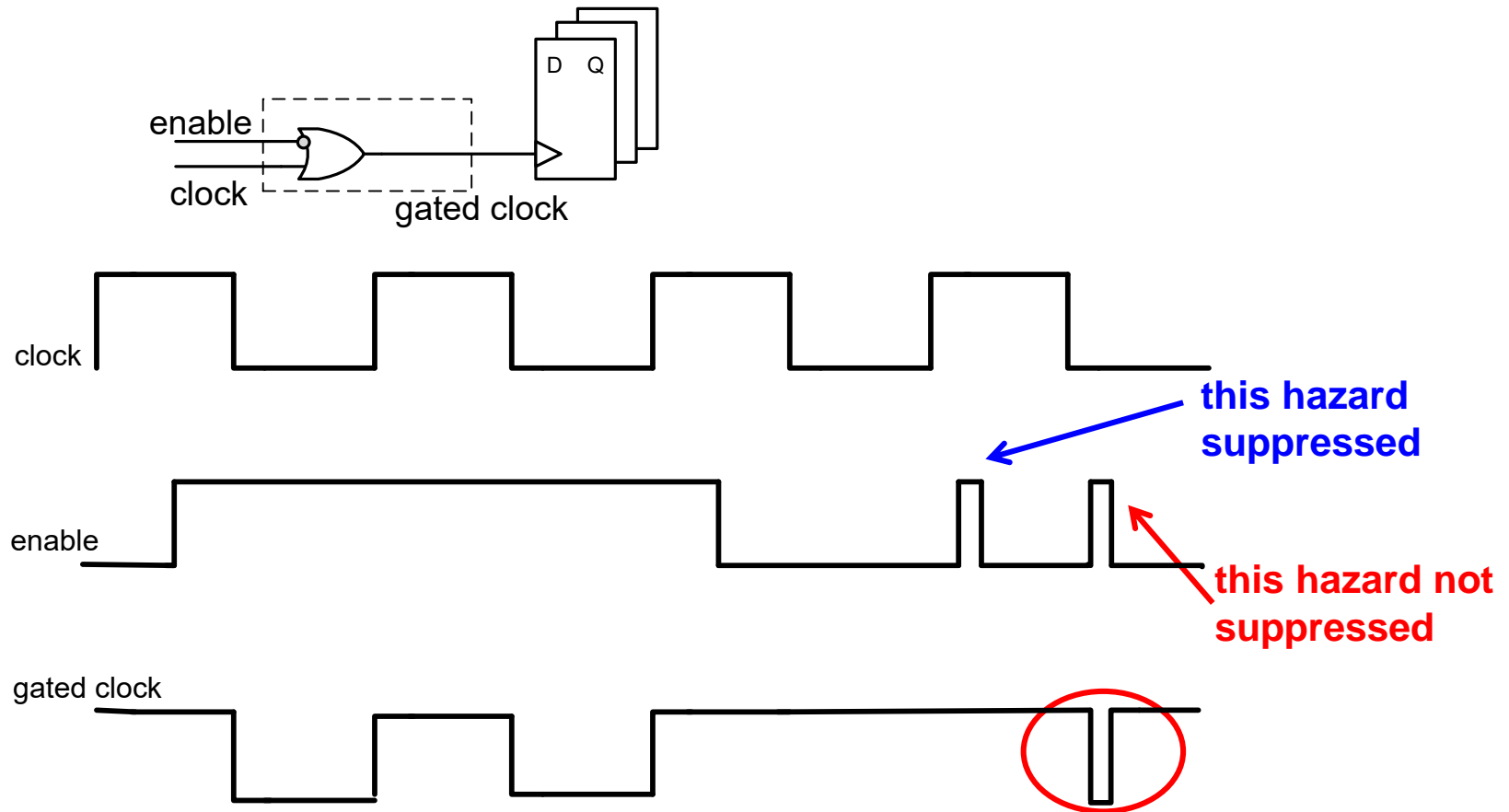


Testable design  
always input pin when SE=1

(WWW Fig 2.22)

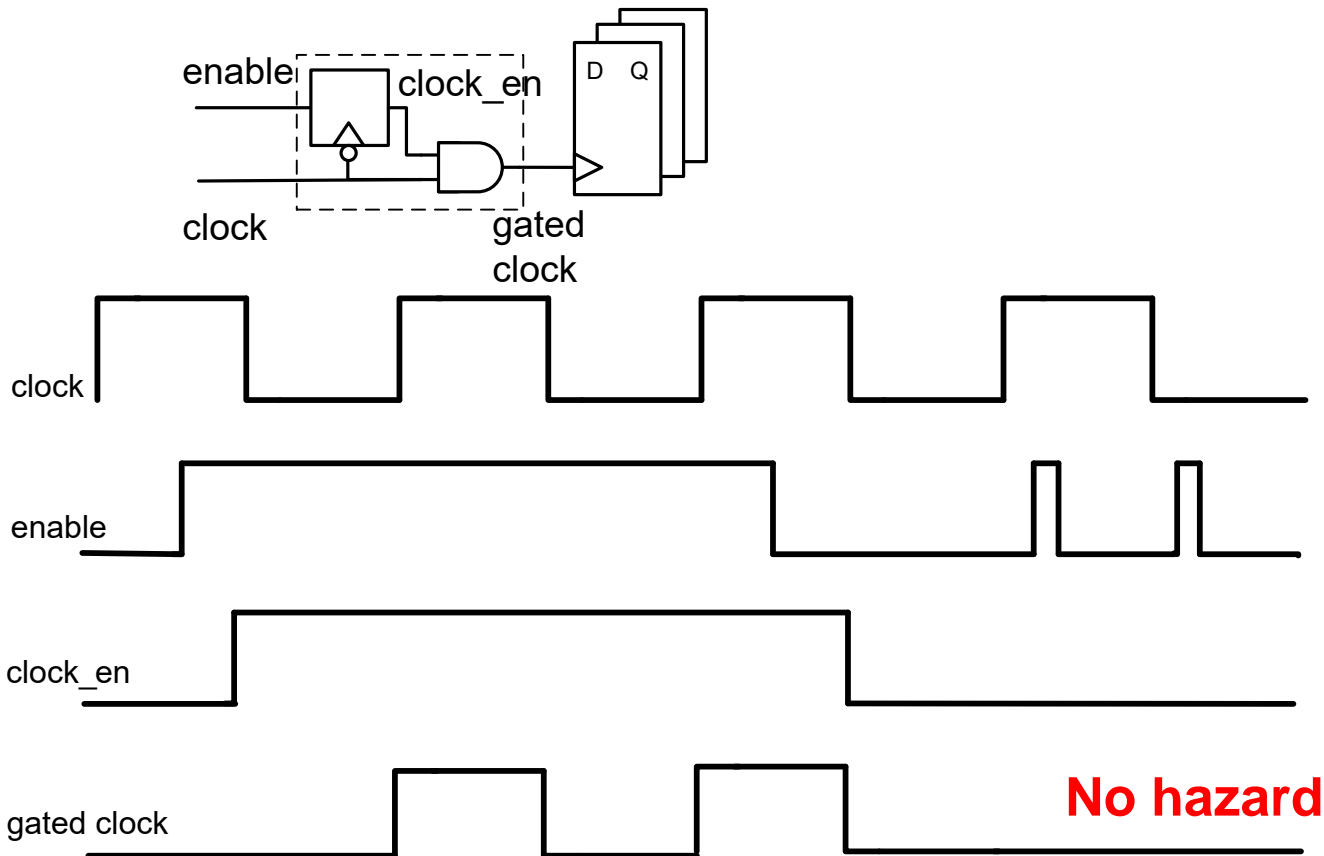
# Gated Clock (1)

- Clock is gated (OFF) when enable = 0
- This is not good design. Cannot handle hazards



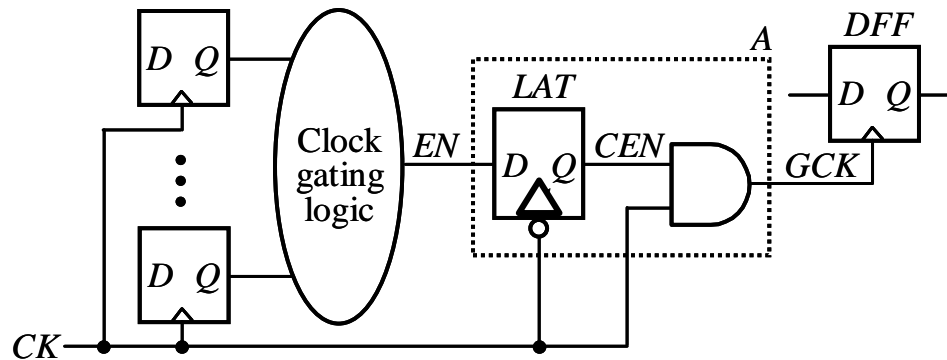
# Gated Clocks (2)

- This is a better design that handle hazards
- Problem: clock is **randomly gated** during scan shift

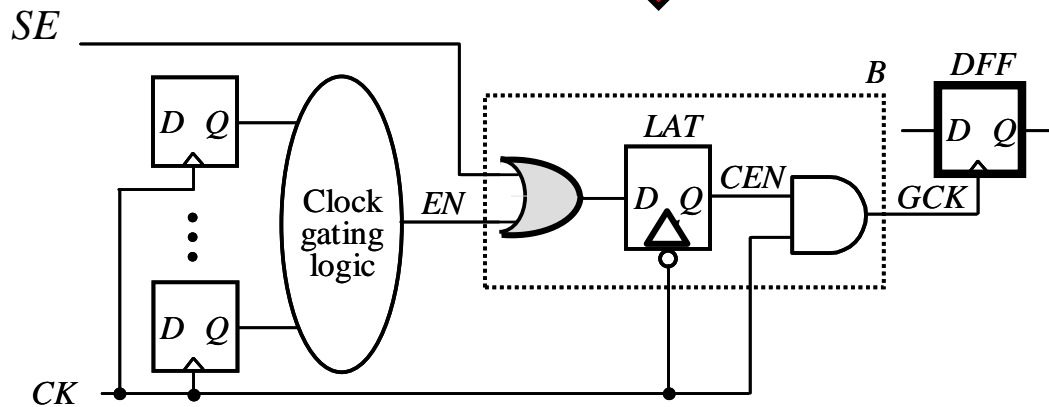


# DRC#3: Gated Clock

- Problem repaired: When  $SE=1$ , clock always ON



Original design



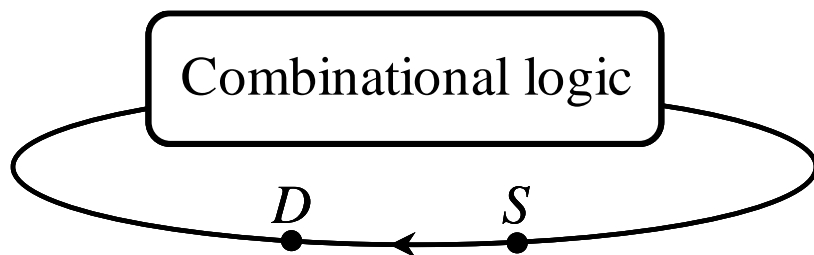
Testable design  
clock always ON when  $SE=1$

(WWW Fig. 2.23)



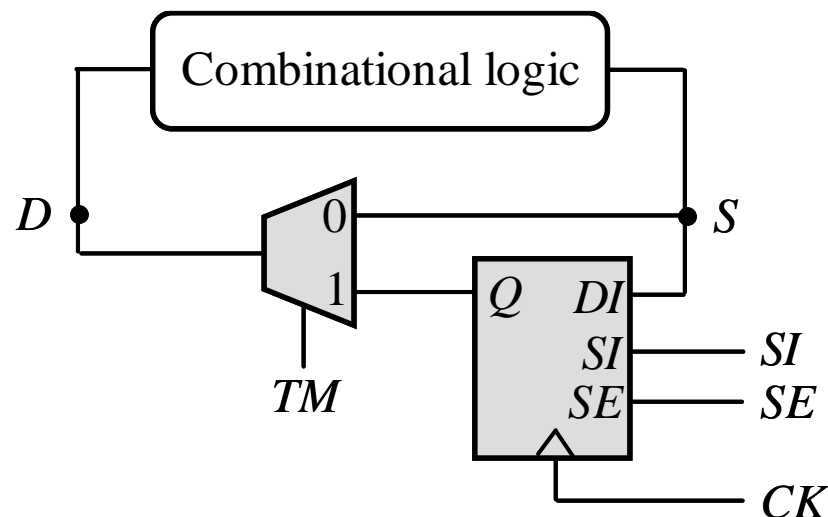
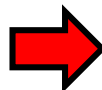
# DRC#4: Combinational Feedback Loop

- Problem: Sequential behavior in test mode
  - ◆ Sequential ATPG is difficult!



(WWW Fig. 2.25)

**Original design**



**Testable design**

break loop when  $TM=1$

$TM=0$ , normal operation

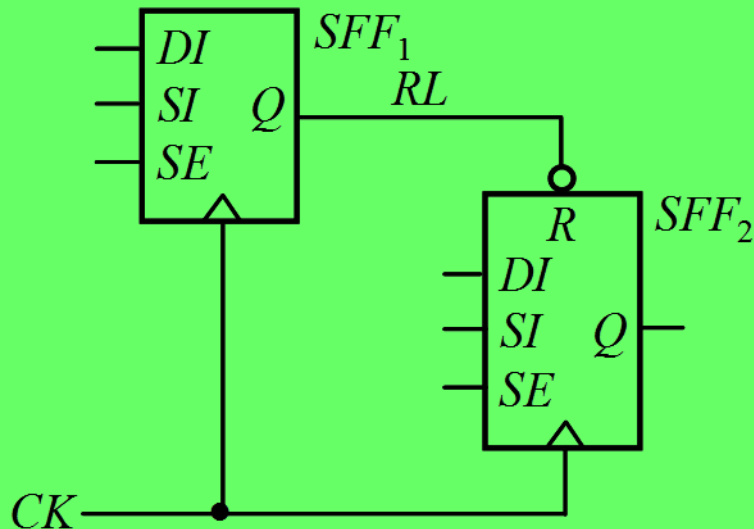
$TM=1, SE=1$ : scan shift

$TM=1, SE=0$ : scan capture

# Quiz

**Problem:**  $SFF_2$  randomly reset during scan shift

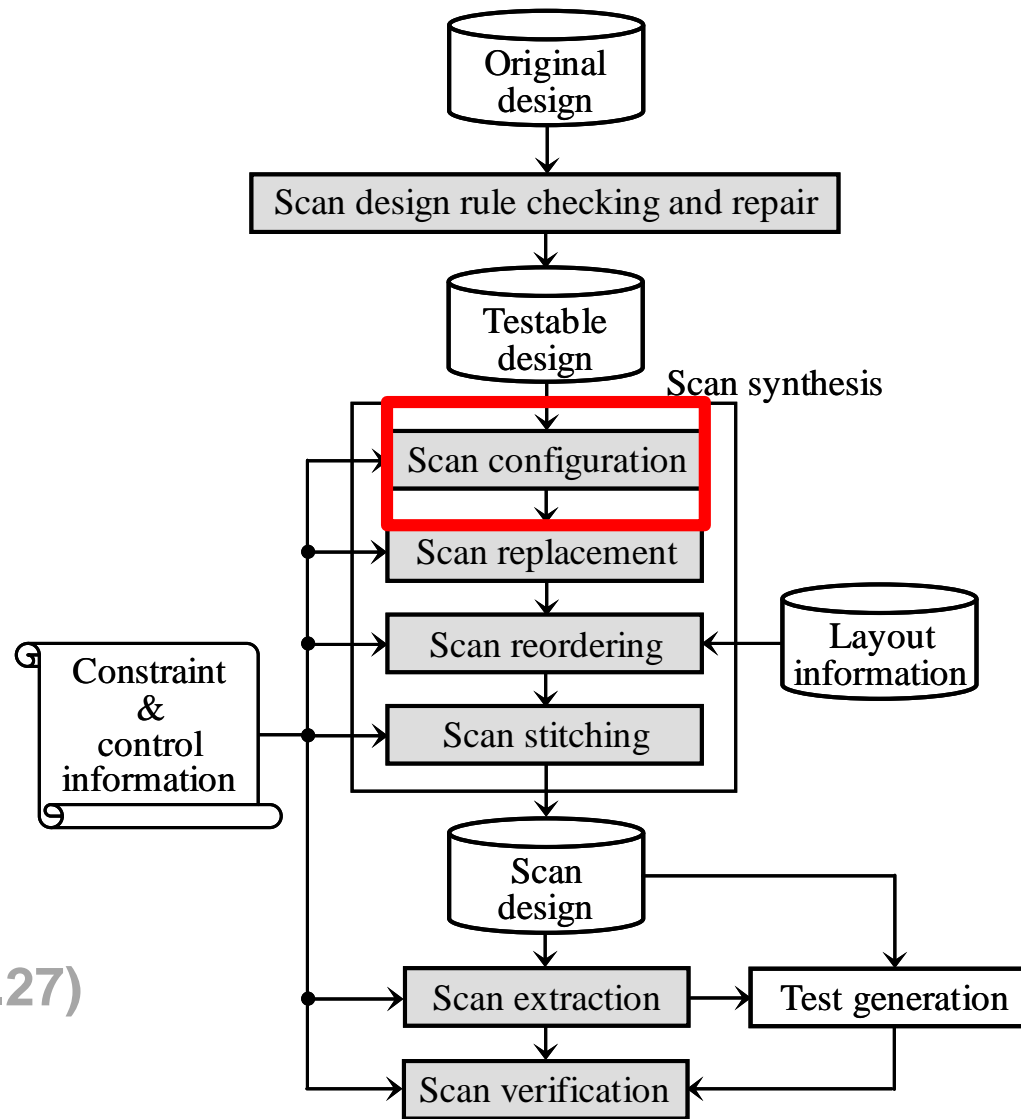
**Q:** Please repair this circuit so that no reset occurs when  $SE=1$



**Original design**

**Testable design**

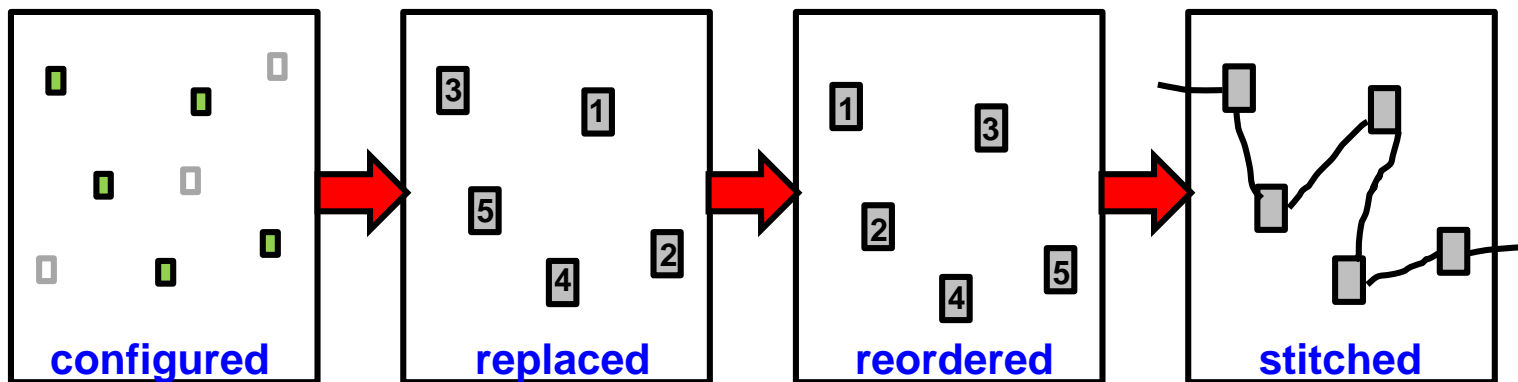
# Scan Design Flow



(WWW Fig. 2.27)

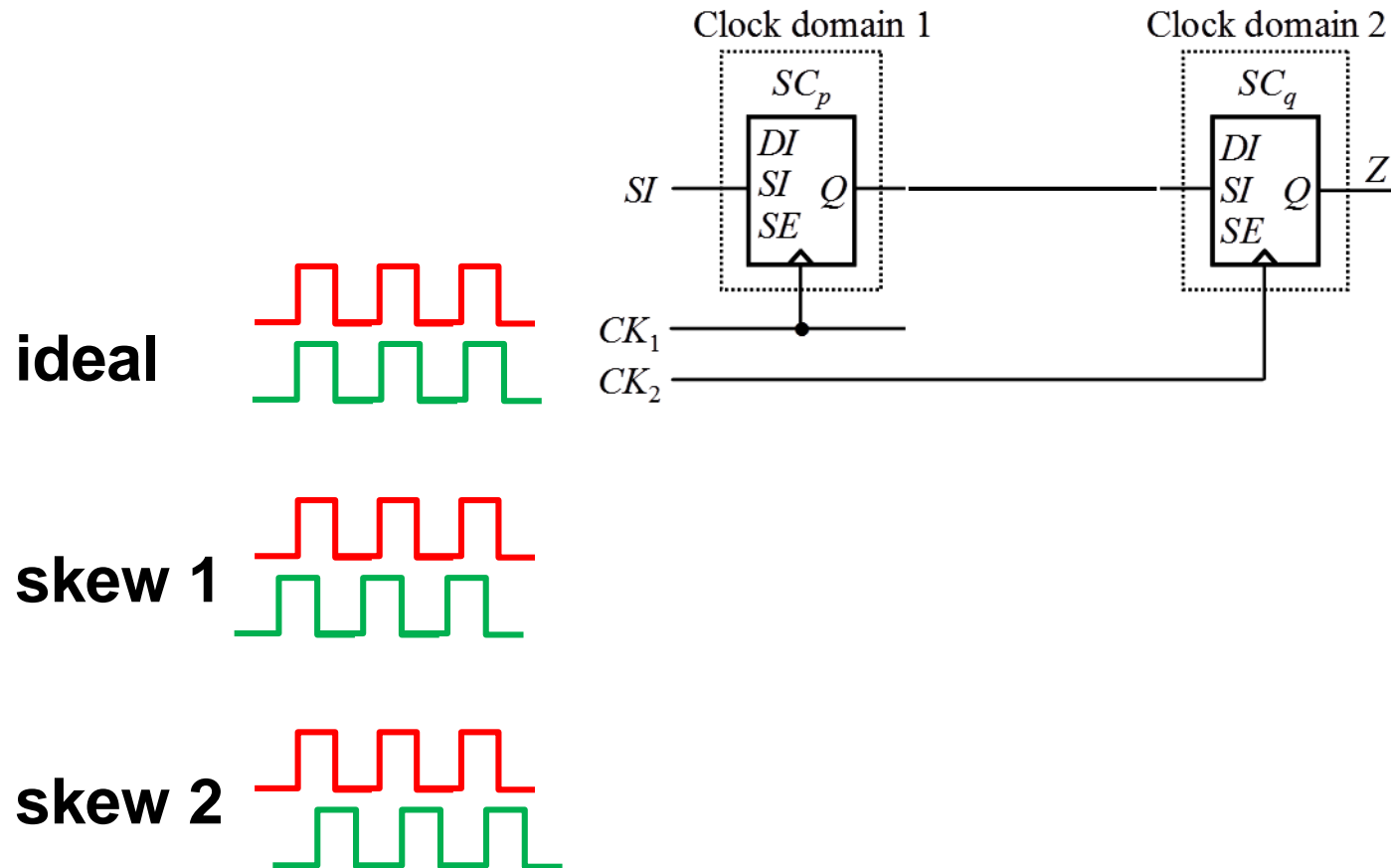
# Scan Synthesis

- aka. *Scan insertion, DFT insertion*
- Four steps:
  - ♦ 1. *Scan Configuration*
    - \* Decide number of scan chains, FF to exclude, ...
  - ♦ 2. *Scan Replacement*
    - \* Replaces original FF/latch in testable design with their functionally-equivalent scan cells
  - ♦ 3. *Scan Reordering*
    - \* Reorder scan cells to minimize routing wires
  - ♦ 4. *Scan Stitching*
    - \* Stitch all scan cells together to form scan chains



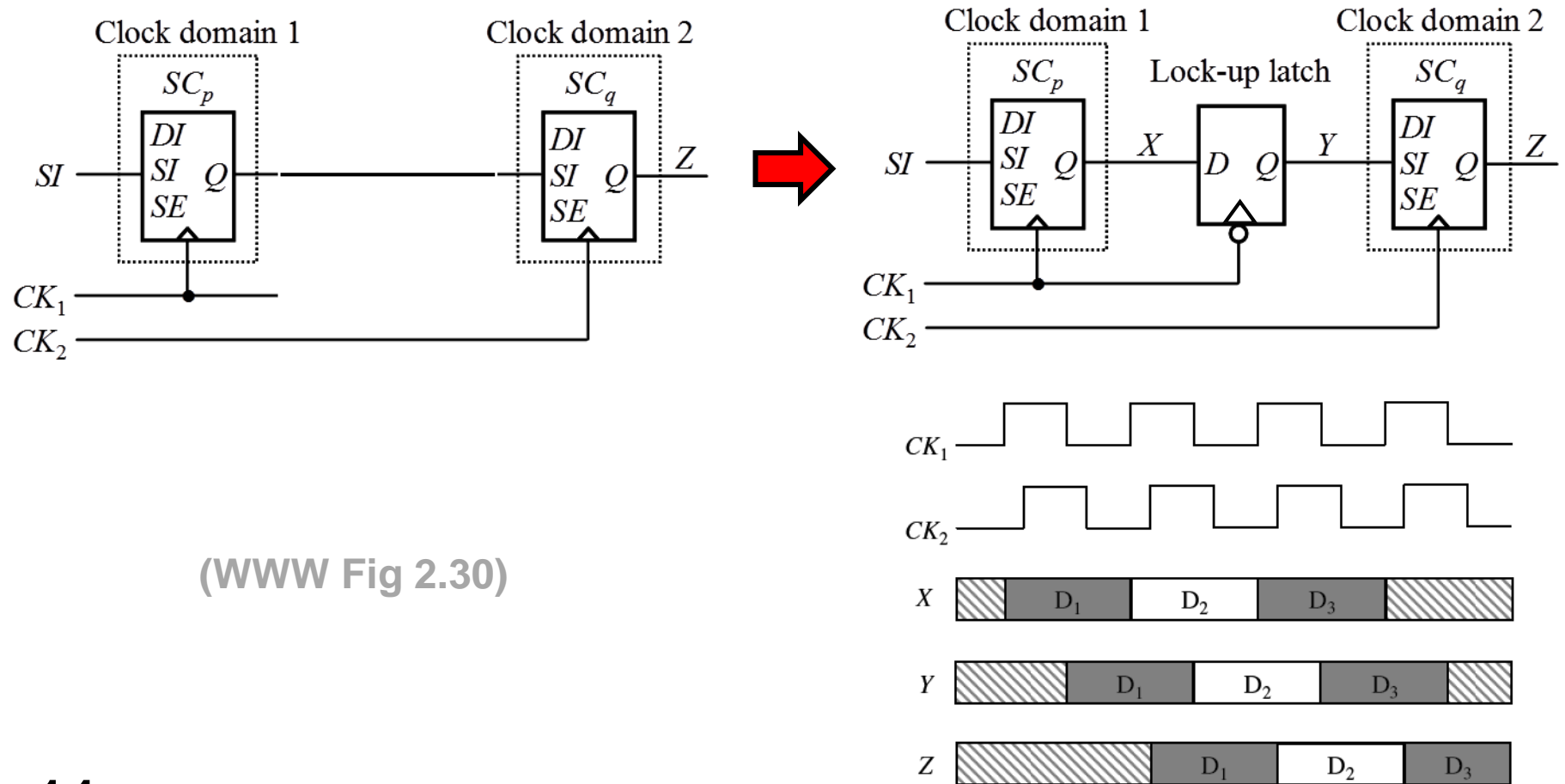
# Scan Chain Cross Clock Domains

- What is wrong with this scan chain?
  - ♦ HINT: consider clock skew among two clock domains



# Lock-up Latch

- Insert **lock-up latch** between two clock domains
  - ◆ Avoid clock skew

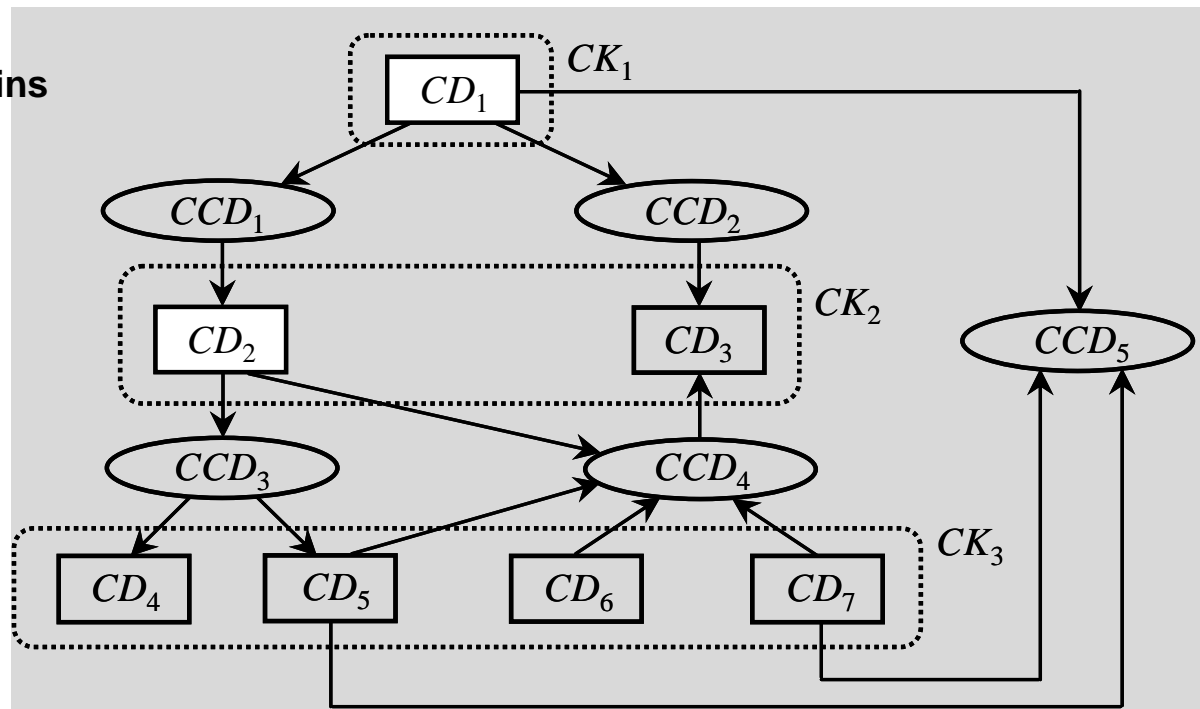


(WWW Fig 2.30)

# Clock Grouping\* not in exam

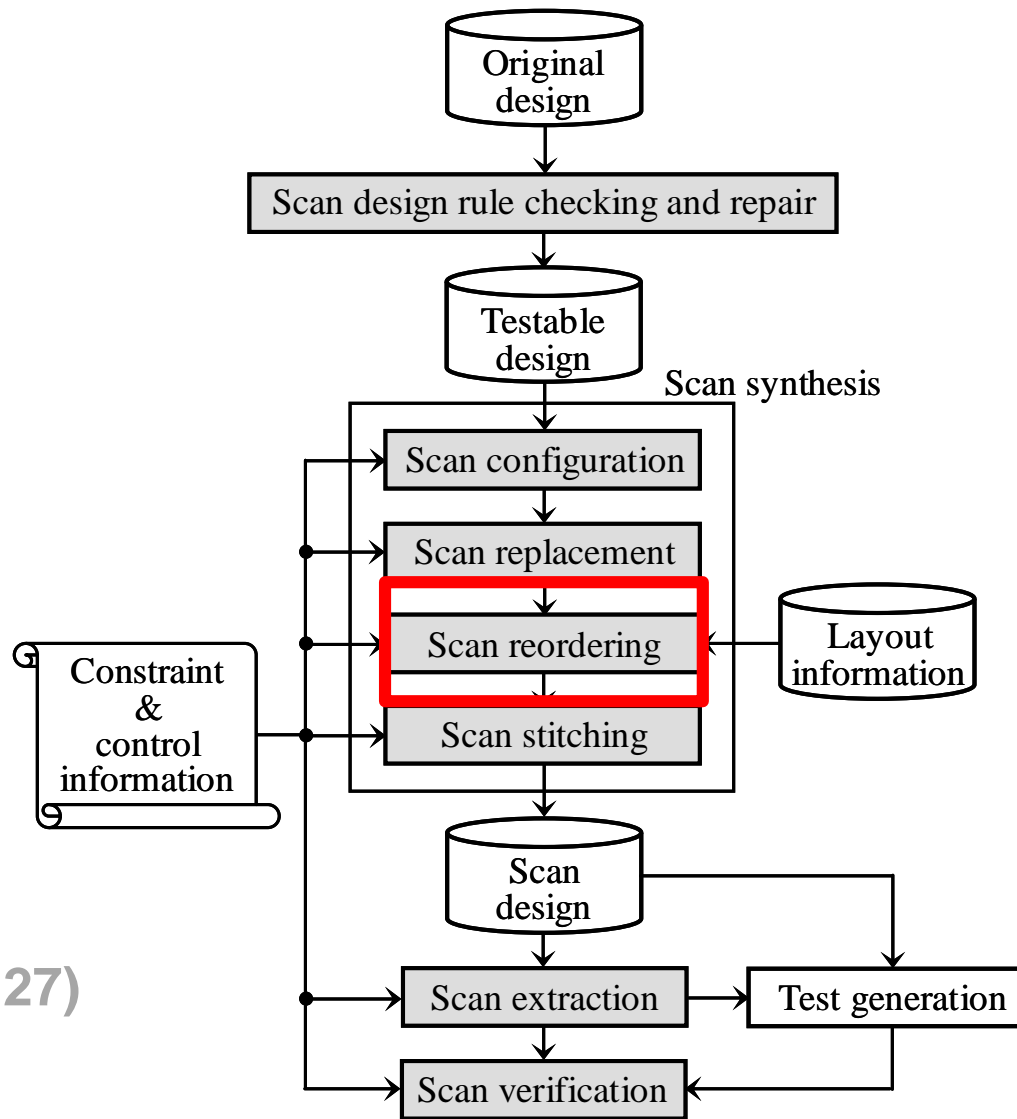
- Due to large clock skew, data from different clock domains **CANNOT** be captured together
  - ♦  $CD_1$  and  $CD_2$  cannot form a scan chain.
- We can assign independent CD as one **clock group**
  - ♦  $CD_2$  and  $CD_3$  has NO interaction. Can form a scan chain.
  - ♦ Totally **three** clock groups needed in this design:  $CK_1 \sim CK_3$

CD: (functional) clock domains  
CCD: cross clock domains  
edge indicates data transfer  
CK: clock groups



(WWW Fig. 2.28)

# Scan Design Flow

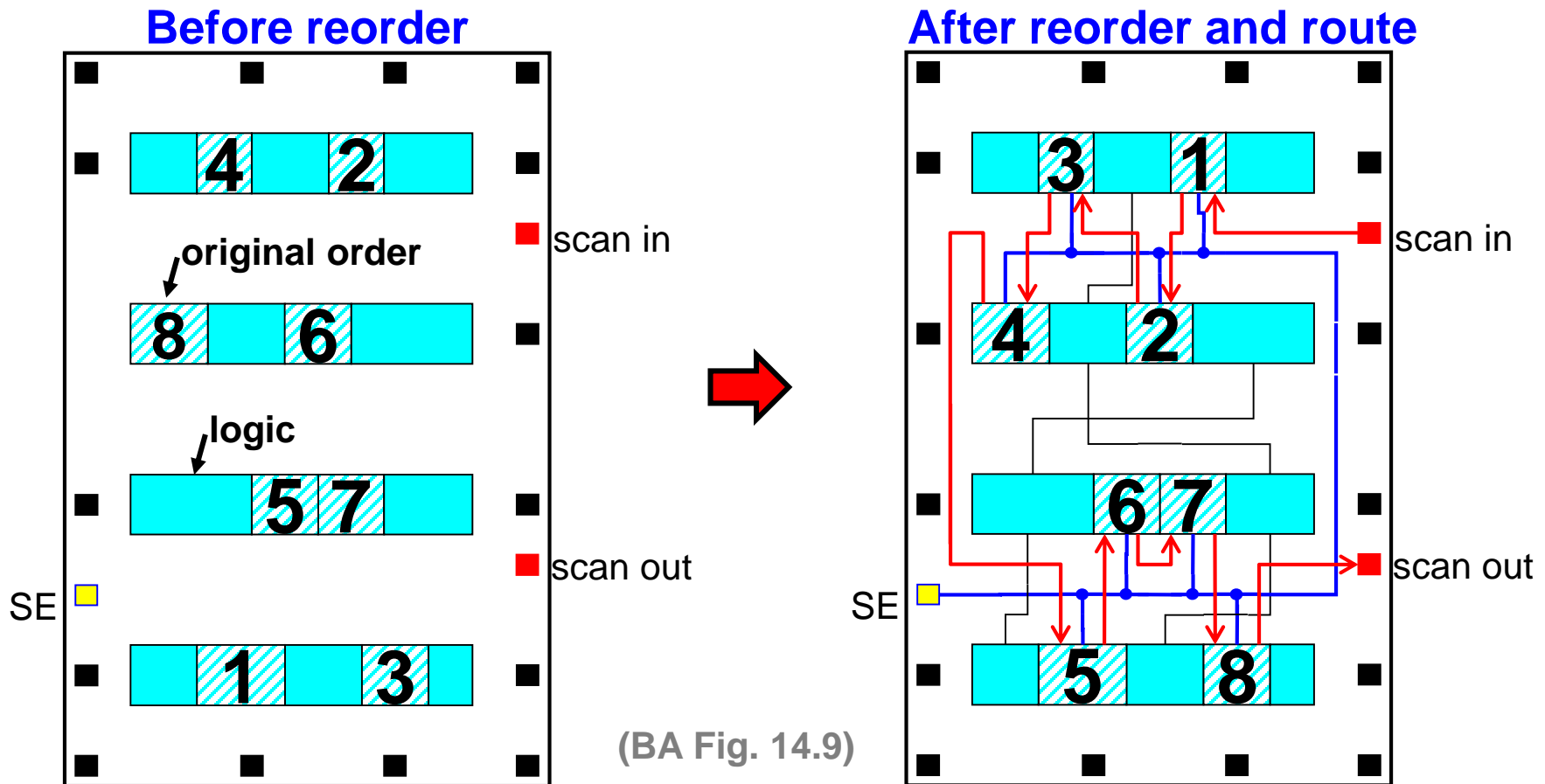


(WWW Fig. 2.27)



# Scan Chain Reordering

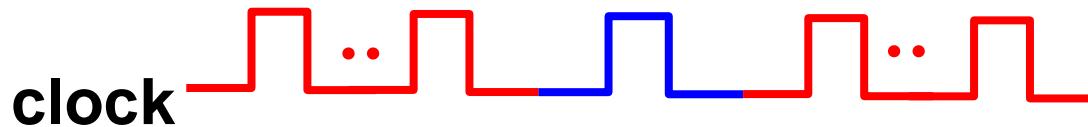
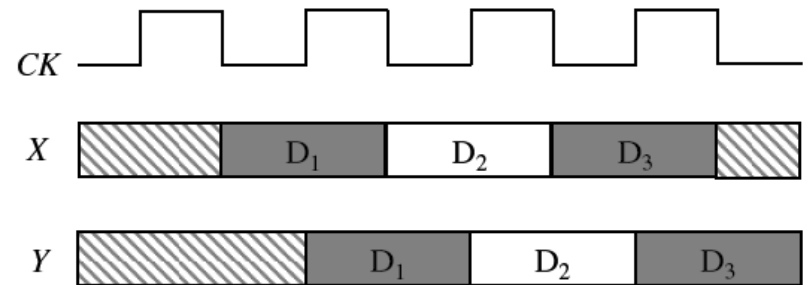
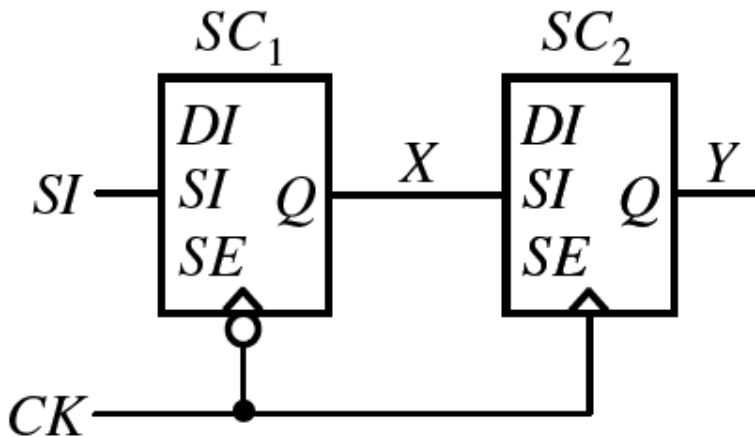
- 1. Place logic gates and FF based on functional timing
- 2. Reorder scan cells to save routing wires
- 3. Route function signals and scan chains



# Mixed N/P-edge FF

- OK when **negative-edge BEFORE positive-edge** FF
- Both scan shift (SE=1) and scan capture (SE=0) are correct

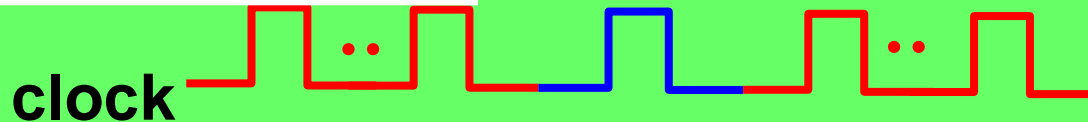
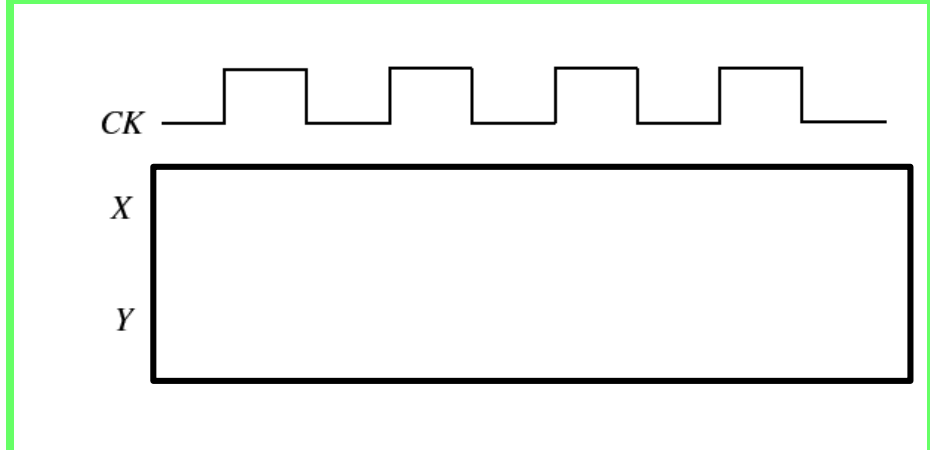
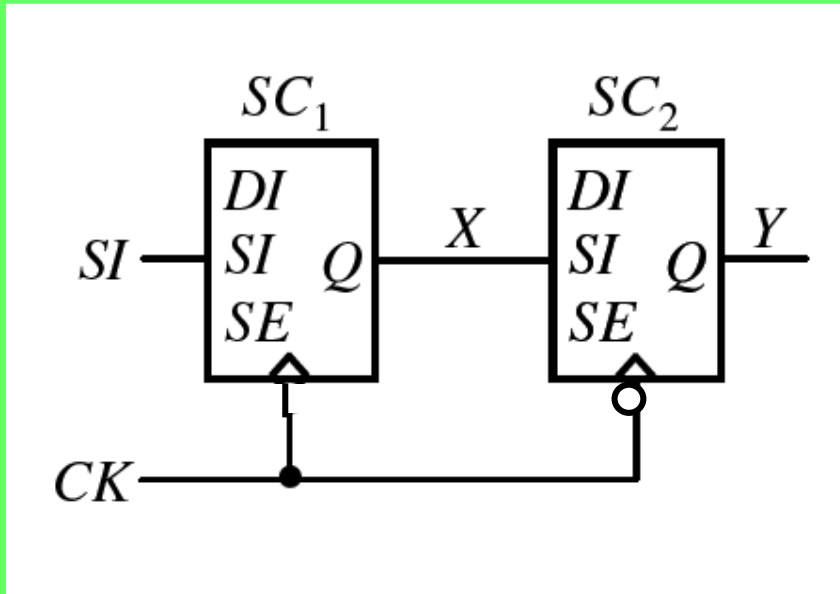
(WWW Fig 2.29)



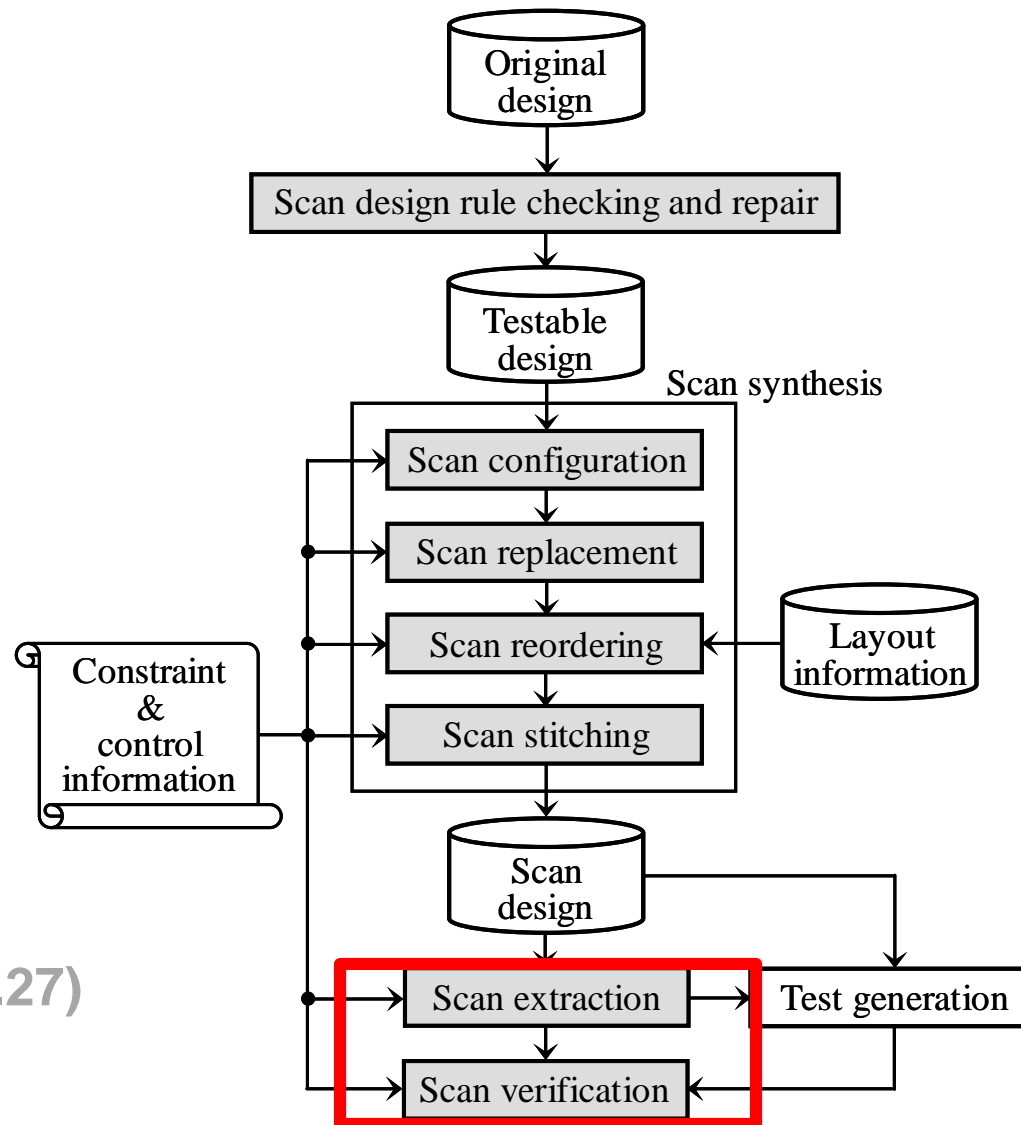
Scan\_Enable

# Quiz

**WRONG** if **positive-edge BEFORE** negative-edge FF  
**WHY?** Hint: consider scan capture when  $SE=0$



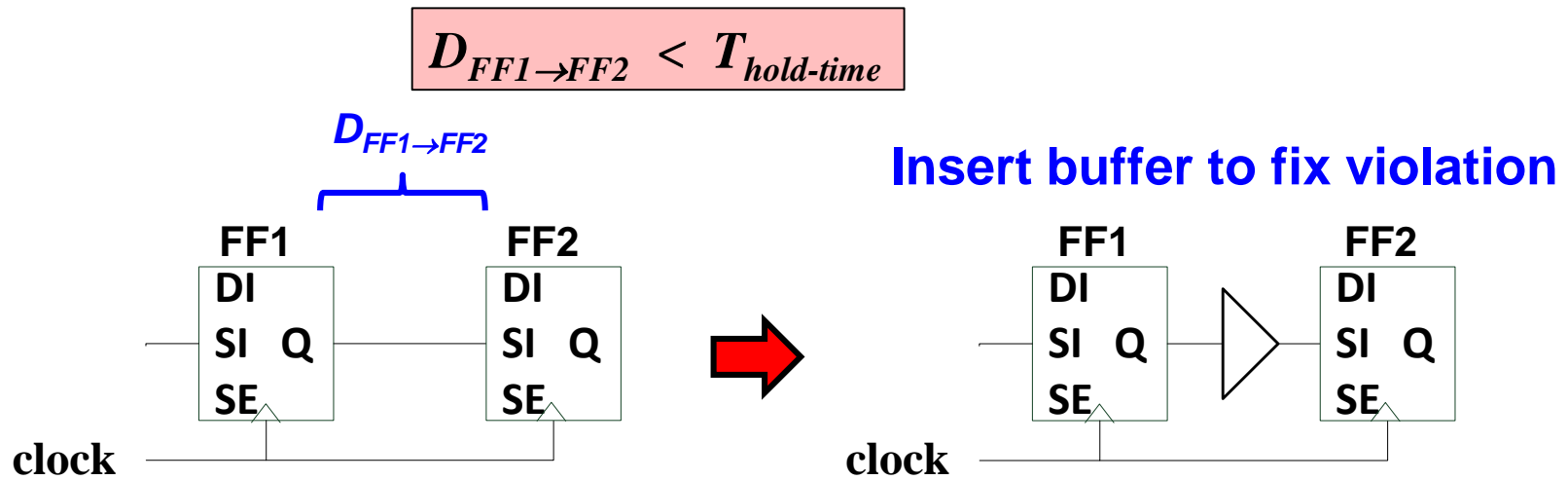
# Scan Design Flow



(WWW Fig. 2.27)

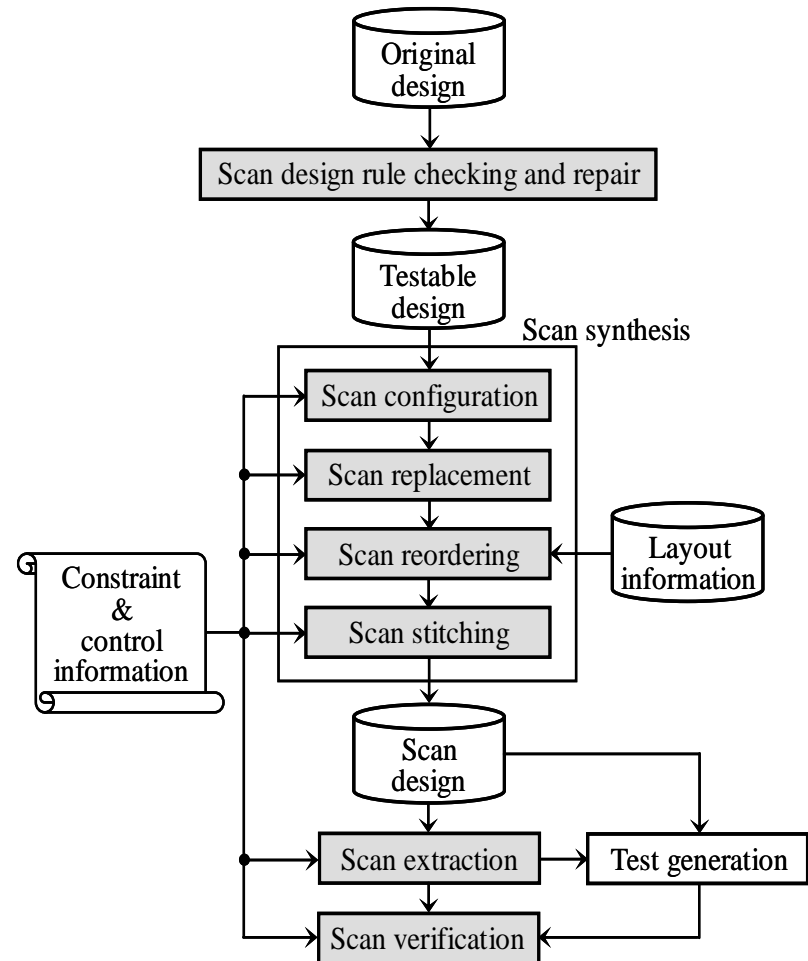
# Scan Extraction and Verification

- **Scan Extraction** (*ATPG* and *Scan Synthesis* are separate tools)
  - ♦ 1. Extract order of scan cells for ATPG
  - ♦ 2. Make sure all chains are correctly connected
- **Scan Verification**
  - ♦ Make sure ATPG test patterns correctly applied
    - \* both scan shift and scan capture
  - ♦ Example: timed logic simulation can catch *hold-time violation*



# Summary

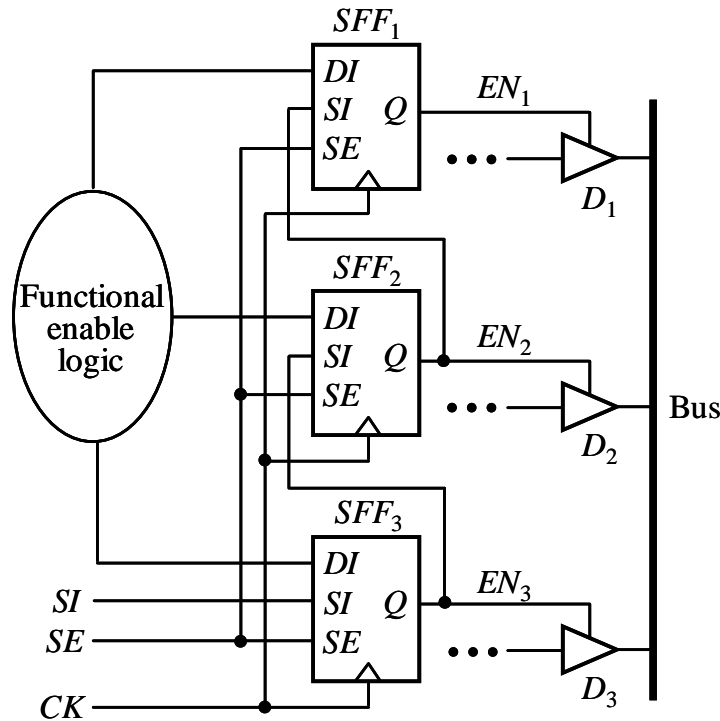
- **Scan Design Rule Checking (DRC)**
  - ♦ Tristate buses, bi-directional I/O, gated clocks, feedback loops...
  - ♦ Violations must be repaired
- **Scan Synthesis**
  - ♦ Scan configuration, replacement, reorder, stitching
  - ♦ Insert lock-up latch, buffer
- **Scan Extraction and Verification**
  - ♦ Test patterns work correctly



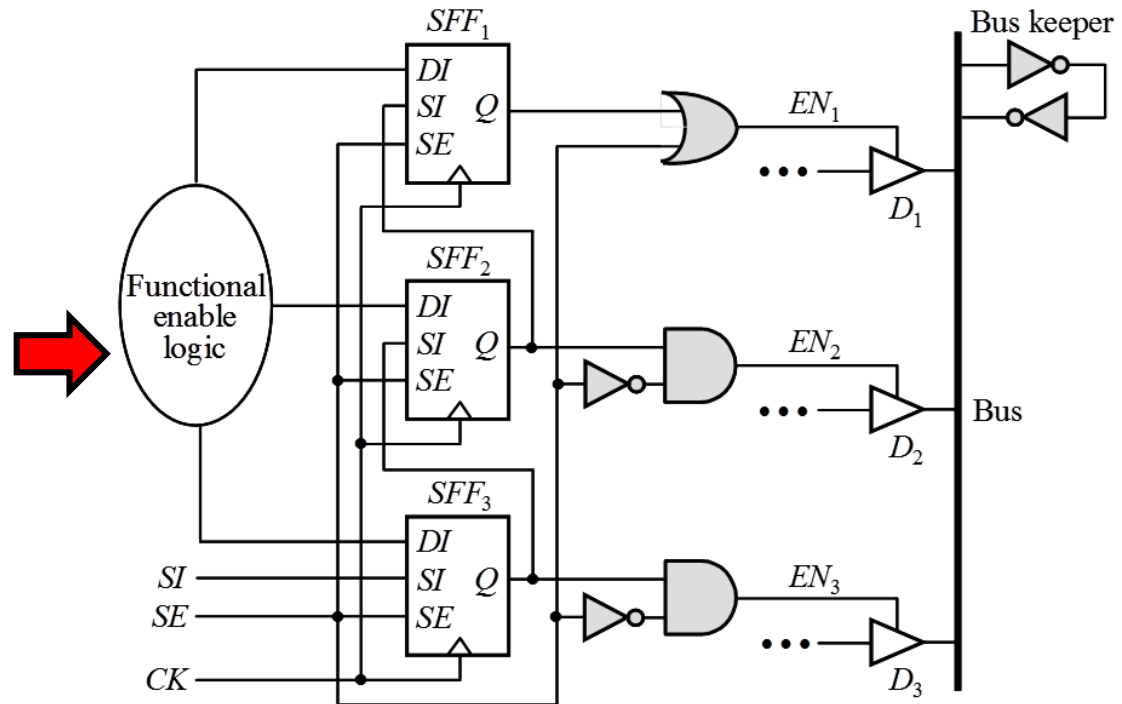
**Successful Scan Design is Non-trivial**

# FFT

- Q: We make sure no contention when SE=1. How about SE=0?  
 ♦ *i.e.* How to avoid contention during scan capture?



Original design



Testable design

No bus contention when SE=1  
 No floating bus

# FFT

- Q: What is wrong with this scan chain?
  - ♦ HINT: consider time skew

