VLSI Testing
積體電路測試

Sequential ATPG

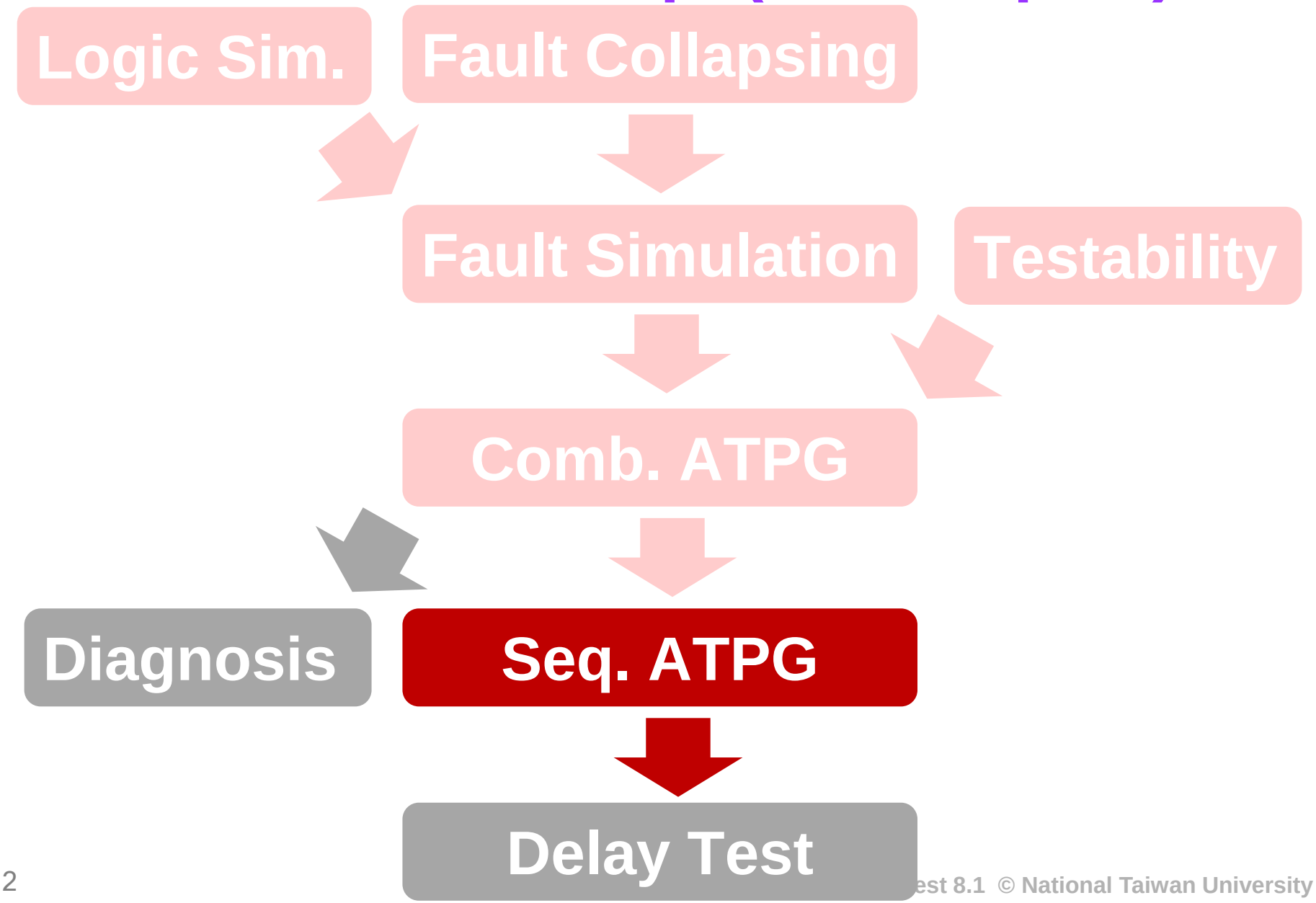
Professor James Chien-Mo Li 李建模

Lab. of Dependable Systems

Graduate Institute of Electronics Engineering
National Taiwan University

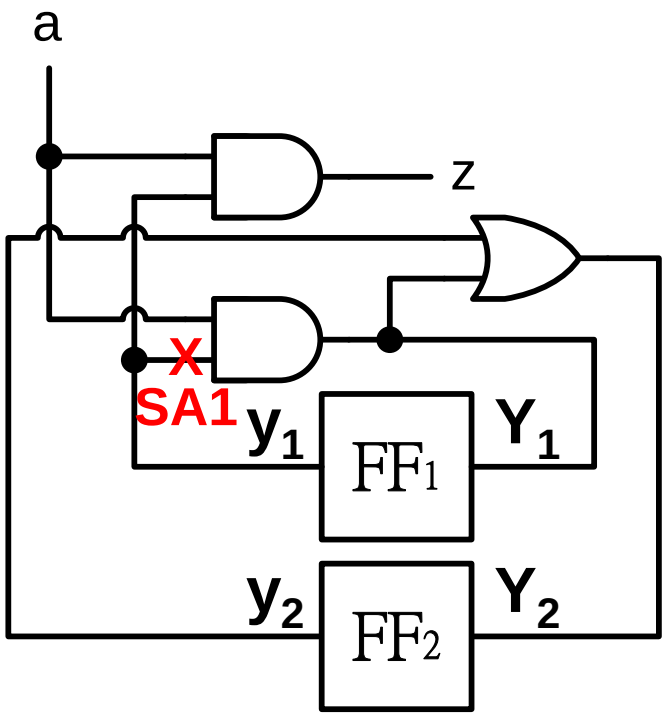
* Some picture are courtesy of Prof. Jiun-Lang Huang, NTU

Course Roadmap (EDA Topics)



Motivating Problem

- You already know ATPG for combinational circuits
- But manager asks you to generate a test for sequential circuits
  - No scan allowed in flip-flops (FF)



Why Am I Learning This?

- **Sequential ATPG**
  - **Generate test patterns for sequential circuits**
  - **Without DFT or scan**

"If life is a test,
one should wish it to be short."

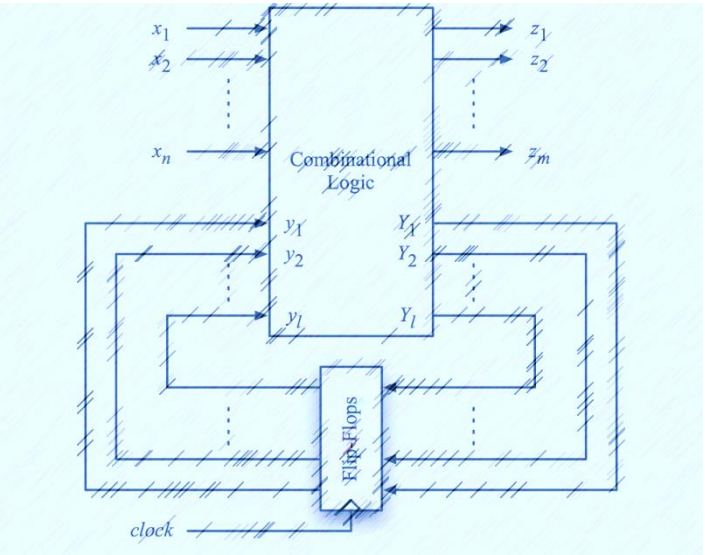
(Bernardin de Saint-Pierre)

Test Generation

Fault Models	Combinational Circuits (or Sequential ckt. with scan)	Sequential Circuits
No fault model	PET	Checking experiment
Single Stuck-at Fault Model	D PODEM FAN	Extended D 9-valued
Delay Fault Model	Path Delay Transition Delay Fault	Launch on Capture Launch on Shift

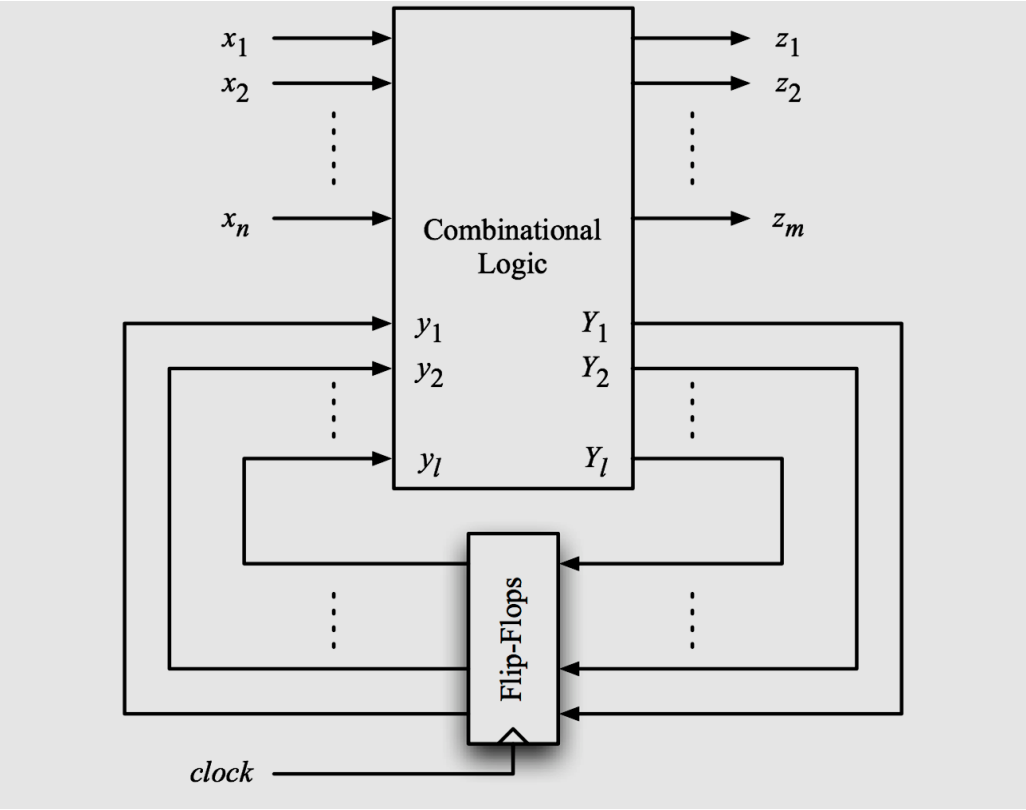
Sequential ATPG

- **Introduction**
- **Time-frame expansion methods**
- **Simulation-based methods * (not in exam)**
- **Issues of Sequential ATPG * (not in exam)**
- **Conclusions**



Huffman Model for Sequential Ckt. [Huffman 53]

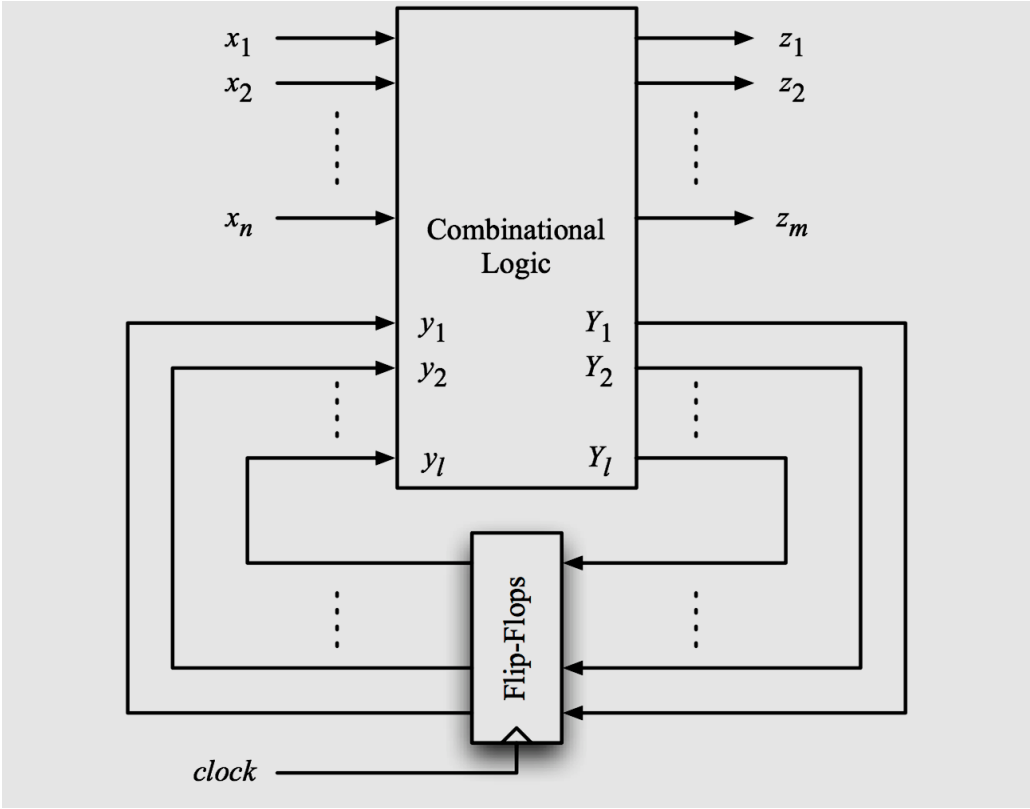
- x_i = primary inputs (PI)
- z_i = primary outputs (PO)
- y_i = FF (or latch) current states
- Y_i = FF (or latch) next states



(WWW Fig 3.3)
Can be either
Moore or Mealy

Sequential ATPG Assumptions

- **NO SCAN ALLOWED!**
  - **Control only PI**
    - **FF not controllable**
  - **Observe only PO**
    - **FF not observable**
- **Faults in CL only**
  - **NO fault in FF/latches**



Challenges of Sequential ATPG

- 1. FF/latches states uncontrollable and unobservable
  - FF/latch unknown initial states
- 2. Long run time
  - Comb. ATPG complexity (for a given fault)
    - $O(2^{number\_of\_PI})$
  - Seq. ATPG complexity (for a given fault)
    - $O(2^{number\_of\_PI} \times 9^{number\_of\_FF})$

*9-vlued logic will be covered soon

- 3. Large memory space required
  - Time frame expansion
- 4. Low fault coverage
  - Much worse than Comb. ATPG

Seq. ATPG is More Difficult Than Comb. ATPG

Quiz

Q: For a sequential circuit with 100 flip-flops, what is worst case sequential ATPG complexity?

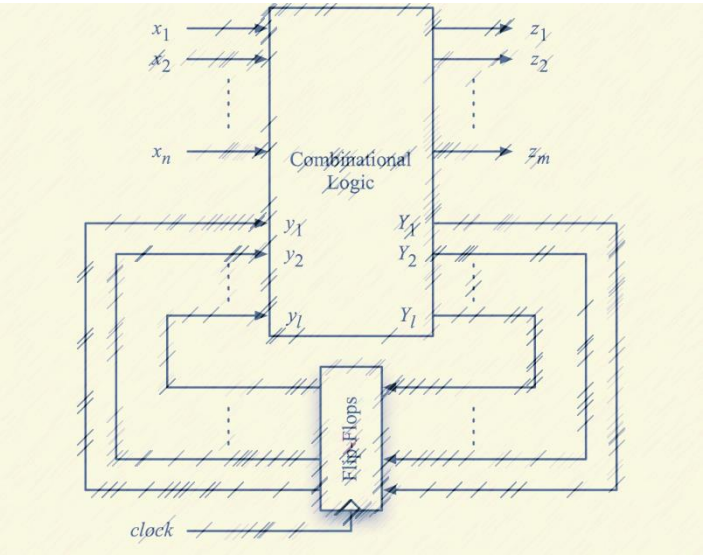
A: 9^{100}

B: 2^{100}

C: 100^2

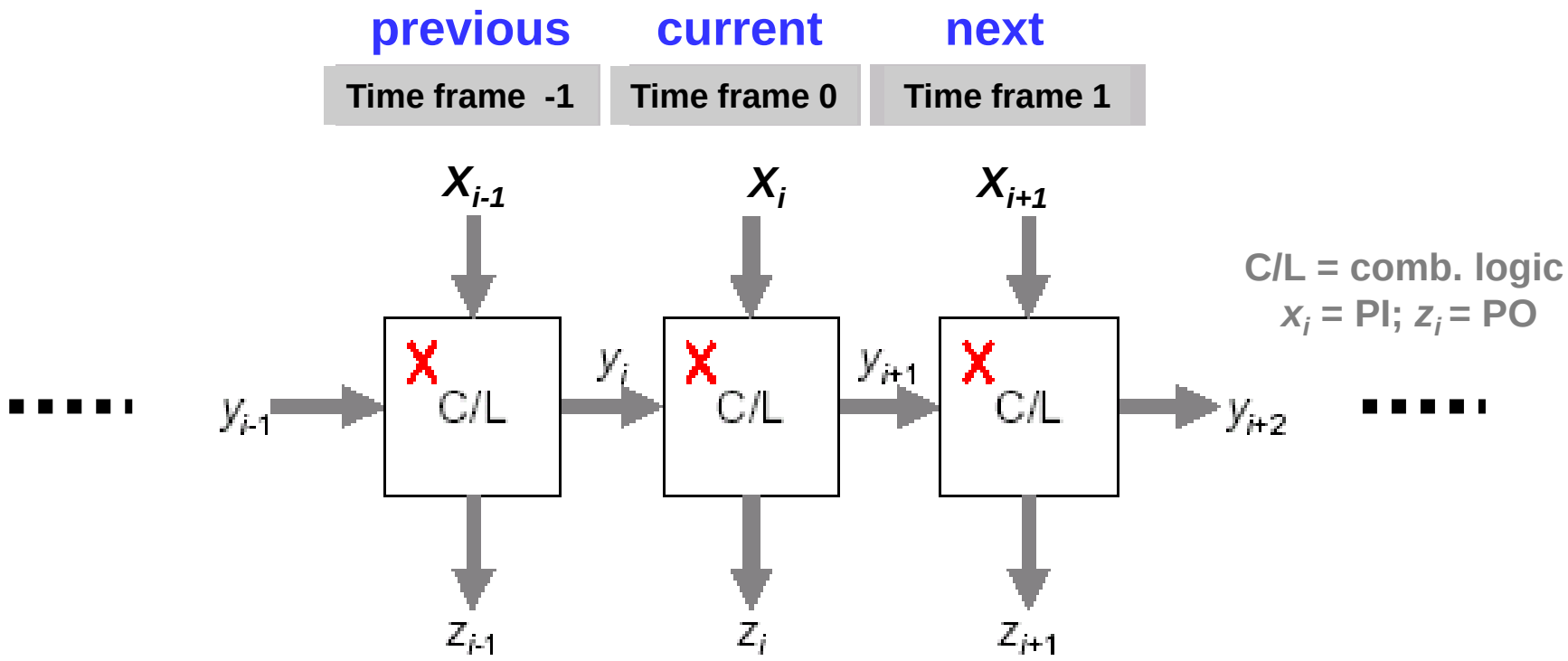
Sequential ATPG

- Introduction
- Time-frame expansion methods
  - The extended D-algorithm [Kubo 68]
  - 9-valued D algorithm [Muth 76]
  - EBT [Marlett 78], BACK [Cheng 88] *
  - Summary
- Simulation-based methods*
- Issues of Sequential ATPG*
- Conclusions



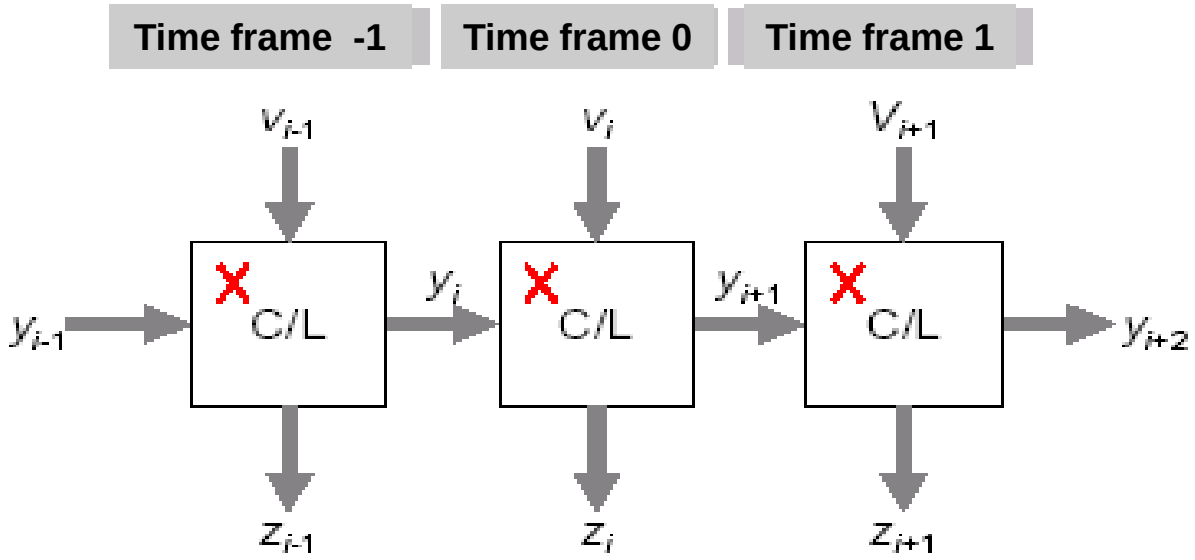
Time Frame Expansion

- IDEA: Replicate circuits and connect time frames by wires
  - y_i = "states"; No FF!
  - Replace clock cycles by space
- Becomes combinational ATPG problem
  - NOTE: Target fault appears in every time frame



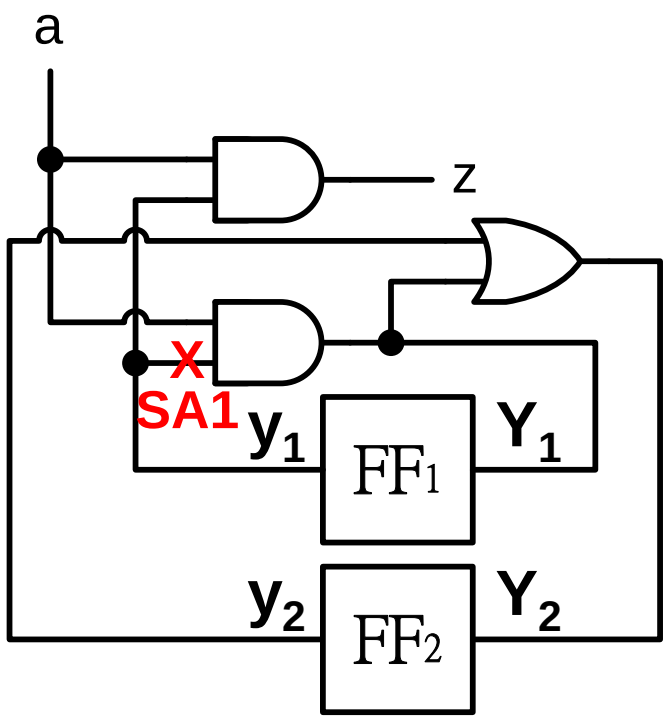
Extended D-Algorithm [Kubo 68]

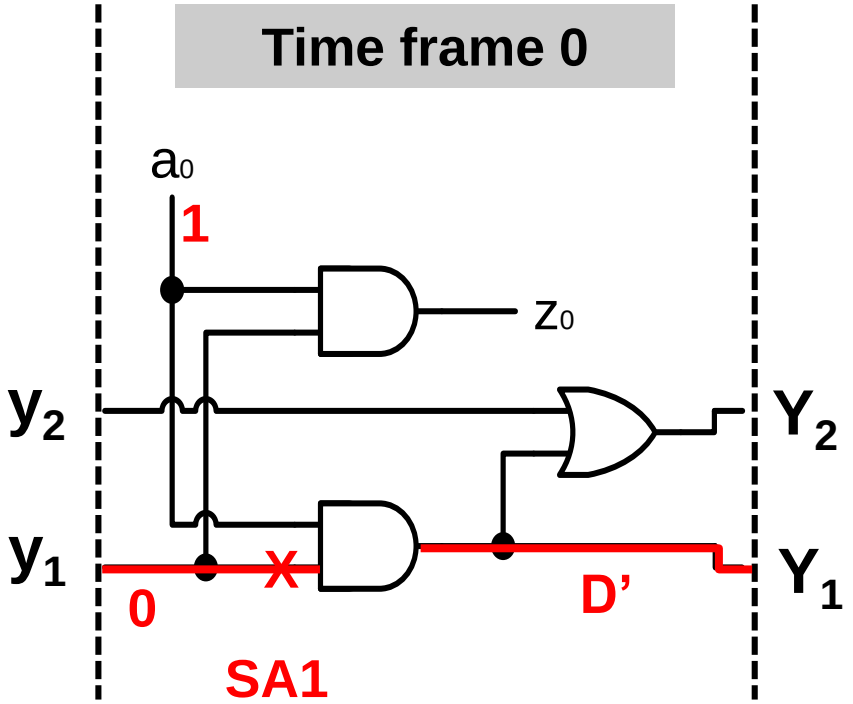
1. Select a target fault *f*
2. Create a copy of the combinational logic, set it to time frame 0
3. Generate a test for *f* for time frame 0 using D-algorithm
4. If the fault effect is propagated to the FF's, continue fault effect *propagation* in the *next* time frame
5. If there are values required in the FF outputs, continue the *justification* in the *previous* time frame



Example (1)

- STEP 2: create Time frame 0
- STEP 3: generate a test
  - a_0=1; y_1=0 ; Y_1=D'

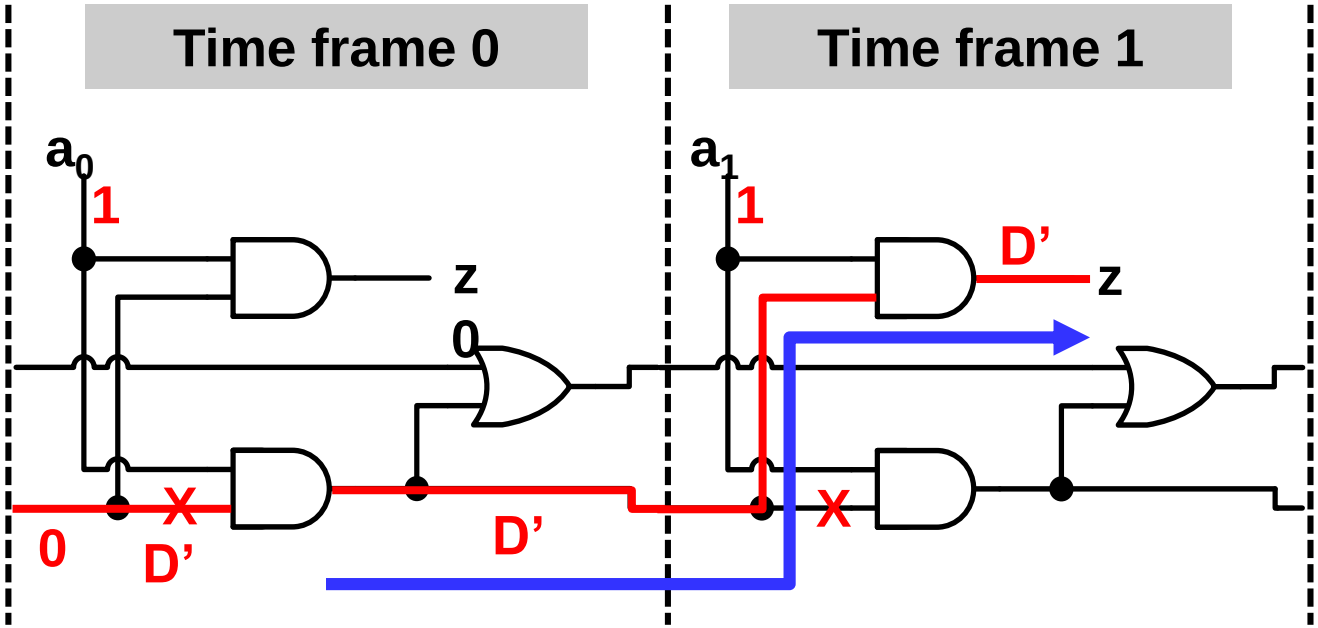




Still Need Propagation

Example (2)

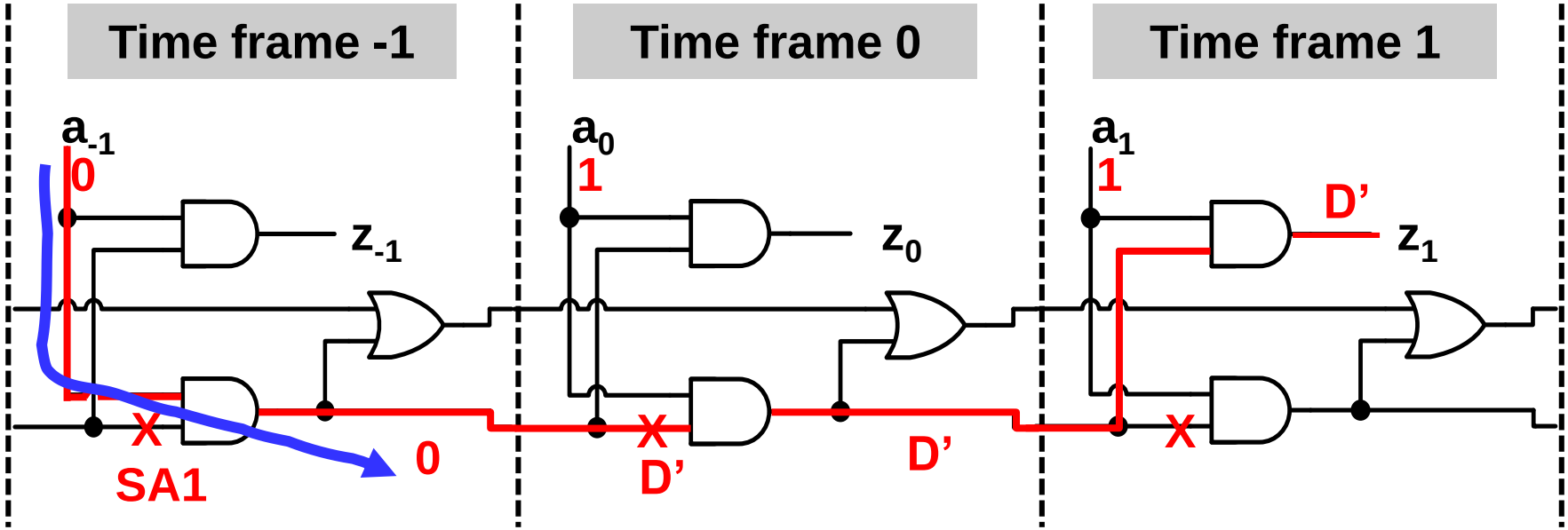
- **STEP4: Fault effect propagation to time frame 1**
  - **a_1=1**



Still Need Activation

Example (3)

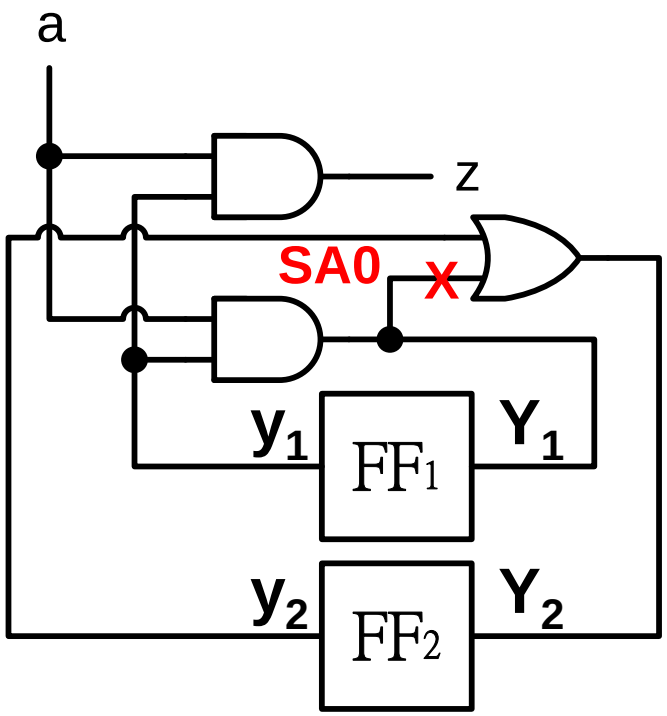
- **STEP 5: Fault activation back to time frame -1**
  - $a_{-1}=0$



Test Generated = 0, 1, 1

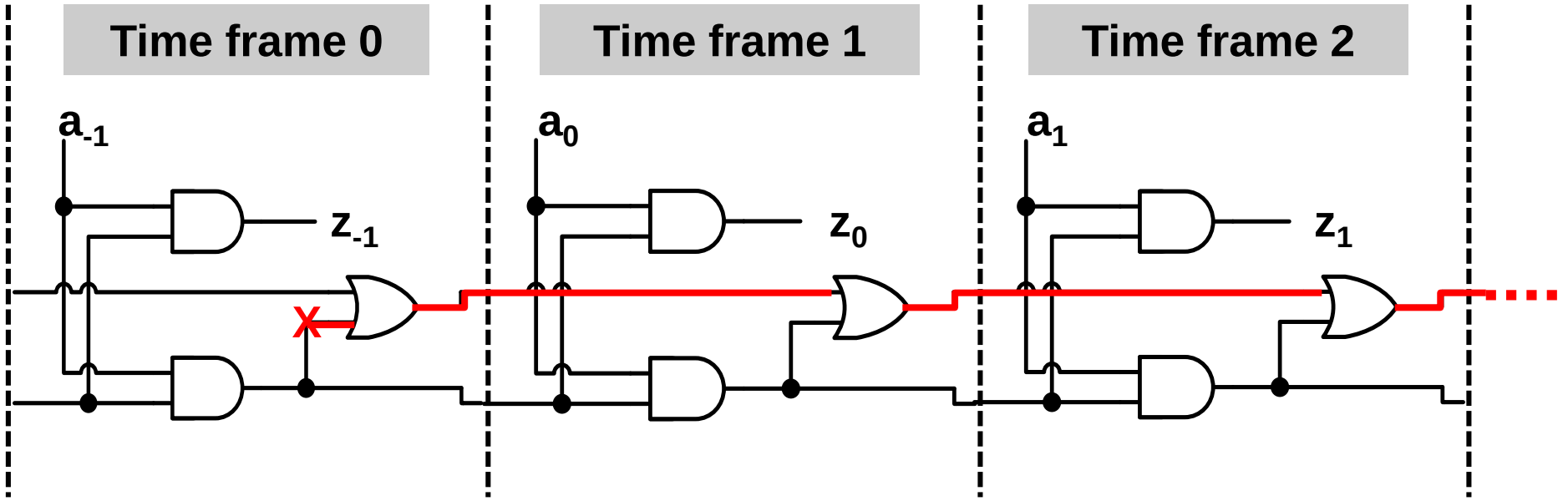
Quiz

Q: Generate a test for stuck-at zero fault in sequential circuit.



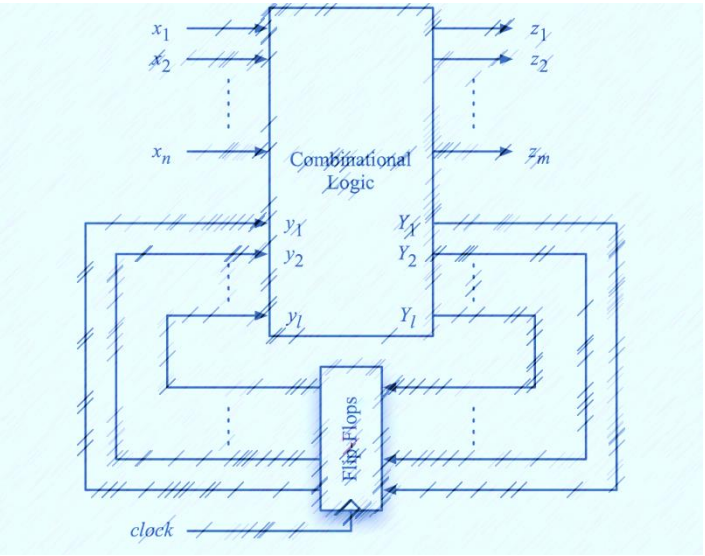
ANS

- No way to propagate
  - This fault is ***untestable*** by sequential ATPG
- Endless timeframe expansion …
  - Memory explosion!



Summary

- **Sequential ATPG**
  - **Time and space consuming**
  - **Low fault coverage**
- **Time-frame expansion methods - extended D-algorithm**
  - **Replicate circuits into many time frames**
  - **Propagate forward, the then activated backward**
  - **create new time frame if needed**



FFT

- In comb. ATPG, first fault activation, then propagation
- In seq. ATPG, first propagation, then activation
  - why?