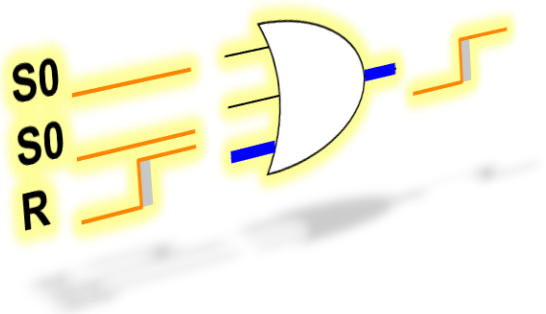
Delay Test

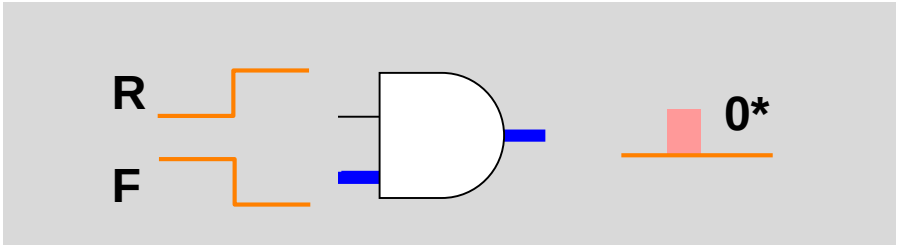
- Introduction and delay fault models
- **Path Delay Fault**
  - Path Sensitization
  - Test Generation [Lin 1987]
  - **Fault Simulation**
    - **Enumerative method [Smith 1985]**
    - **Non-enumerative method [Pomeranz 1994]**
- Transition Delay Fault
- Experimental Results* (not in exam)
- Issues of Delay Tests* (not in exam)
- Conclusions



Six-valued Logic

- *Nine-valued* logic simulation for path sensitization (see 9.2)
- *Six-valued* logic for fault simulation
  - Remove 'X'. assume all inputs are fully specified
  - Keep 0* and 1*, so non-robust sensitization considered

Value	Meaning
S0	Static 0
S1	Static 1
R	Rising
F	Falling
0*	Static-0 hazard
1*	Static-1 hazard
U1	X→1
U0	X→0
XX	X→X



AND	S0	S1	R	F	0*	1*
S0	0	0	0	0	0	0
S1	0	1	R	F	0*	1*
R	0	R	R	0*	0*	R
F	0	F	0*	F	0*	F
0*	0	0*	0*	0*	0*	0*
1*	0	1*	R	F	0*	1*

Path Delay Fault Simulation

① *Enumerative* method

- ☺ Simple algorithm
- ☹ Users need to provide fault list
  - How? Use static timing analyzer (STA) to find long paths
- ☹ Simulation fault coverage may not be representative

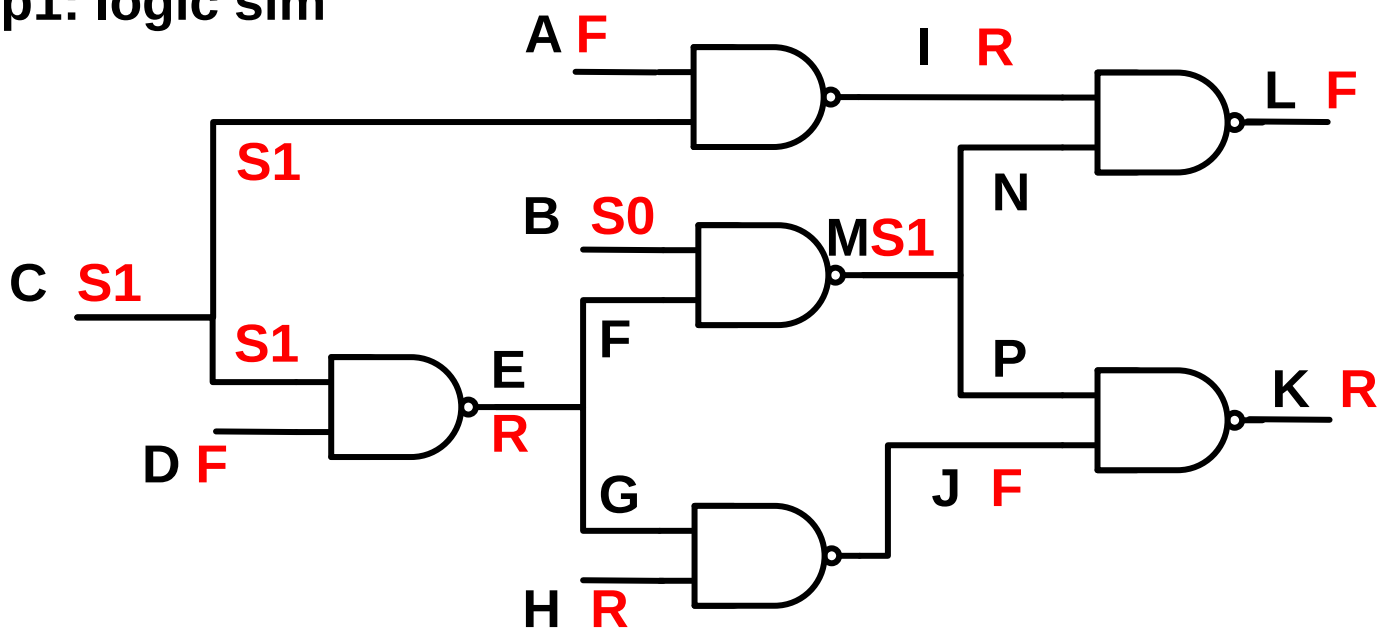
② *Non-enumerative* method

- ☺ Users don't need to provide fault list
- ☹ More complicated algorithm
- ☹ Simulation fault coverage can be pessimistic (see FFT)

Number of PDF Can be WC Exponential

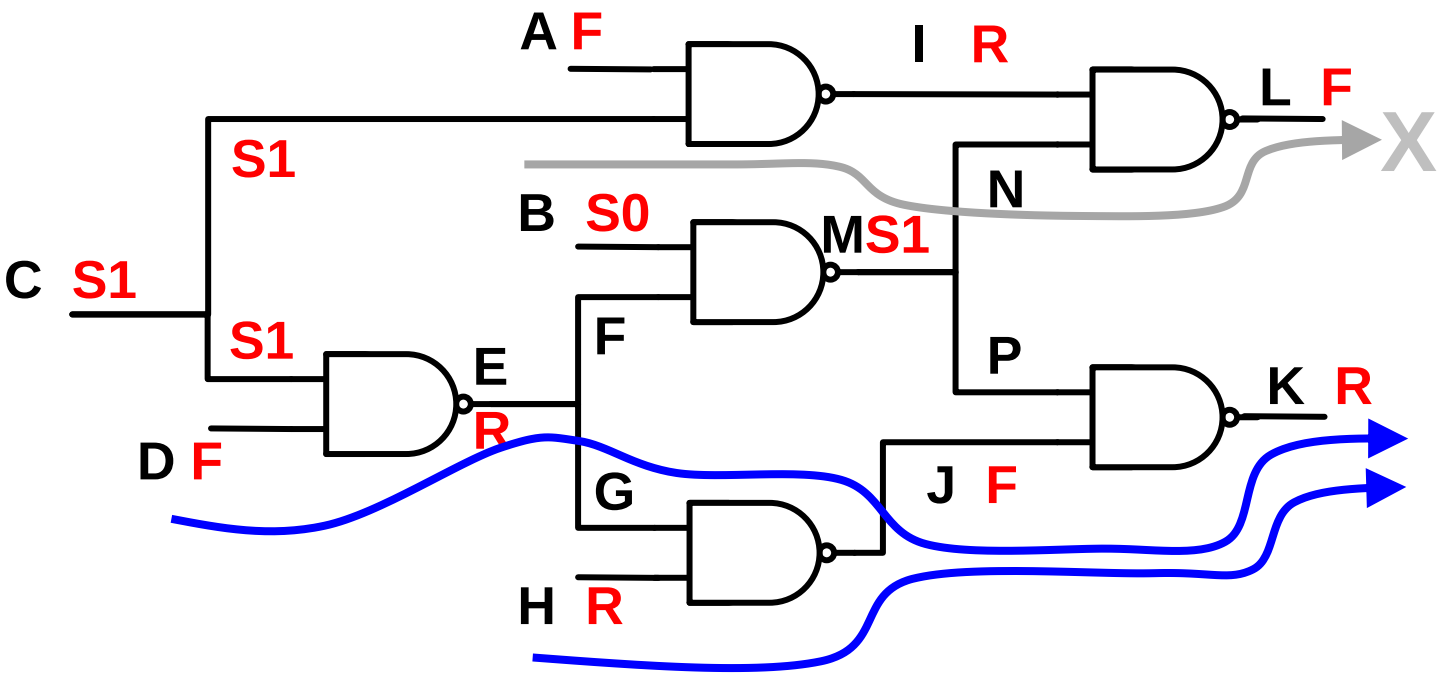
Enumerative Method [Smith 85]

- Given test set, and path delay fault list.
  1. Perform six-valued logic simulation
  2. Count number of sensitized paths in fault list
  3. Calculate fault coverage
- Example: ISCAS'85 C17 benchmark circuit.
  - Given test pattern and {↓DEGJK, ↑HJK, ↑BMNL}. Robust FC=?
  - Step1: logic sim



Enumerative Method (2)

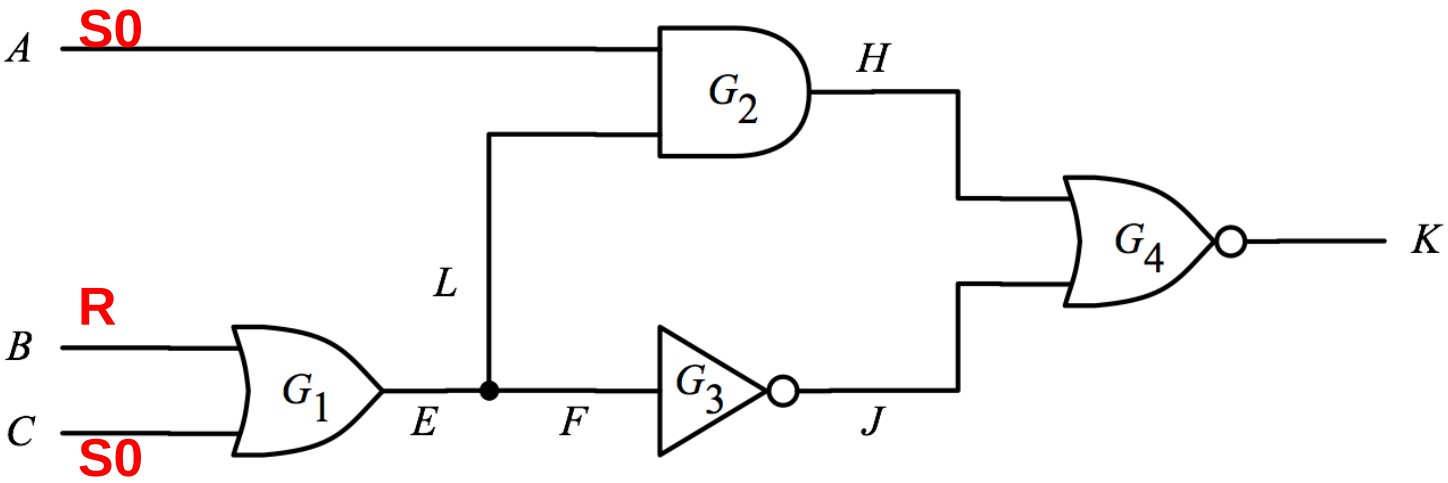
- **Example: (cont'd)**
  - **Step2. Trace every path in fault list**
  - **Step3. If every gate on path robustly sensitized, PDF is detected**
    - **Detected: ↓DEGJK, ↑HJK. Not detected: ↑BMNL**
  - **Robust fault coverage = 2/3 = 66.7%**



Quiz

Q: Apply test pattern ABC=S0,R,S0.
Given fault list {↑BEFJK, ↑BELHK}
What is robust fault coverage?

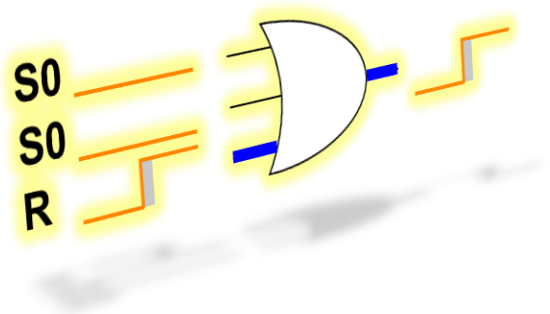
A:



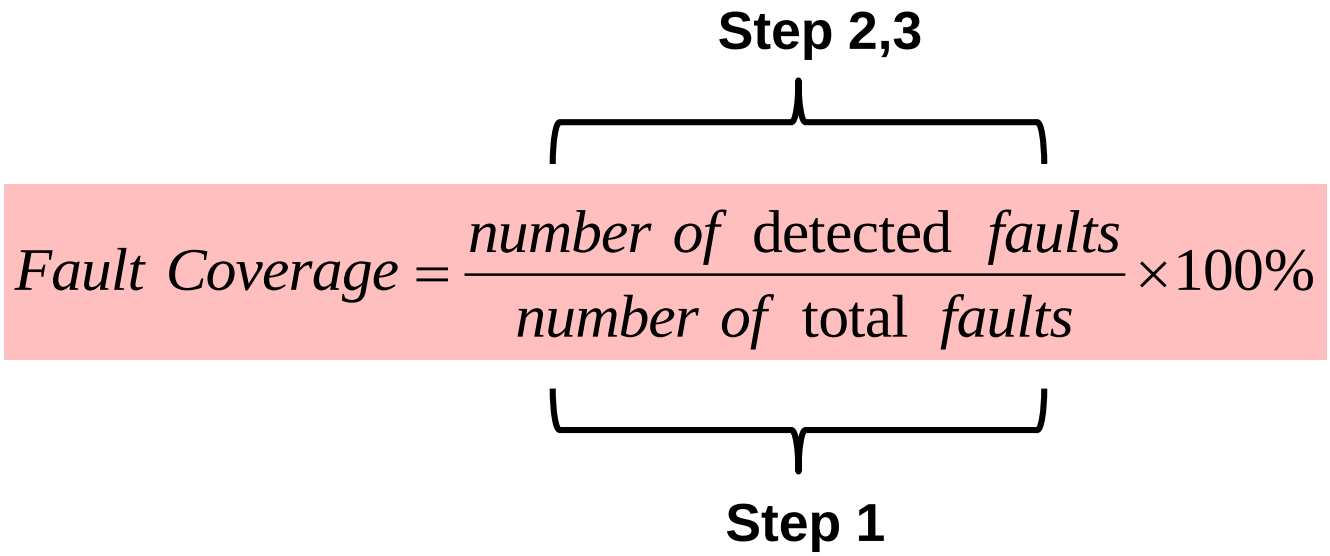
But…this fault coverage is NOT representative, since there are more than 2 PDF in this circuit.

Delay Test

- Introduction and delay fault models
- **Path Delay Fault**
  - Path Sensitization
  - Test Generation [Lin 1987]
  - **Fault Simulation**
    - **Enumerative method [Smith 1985]**
    - **Non-enumerative method [Pomeranz 1984]**
- Transition Delay Fault
- Delay Test Application
- Circuit Model for Delay Test ATPG
- Experimental Results* (not in exam)
- Issues of Delay Tests* (not in exam)
- Conclusions



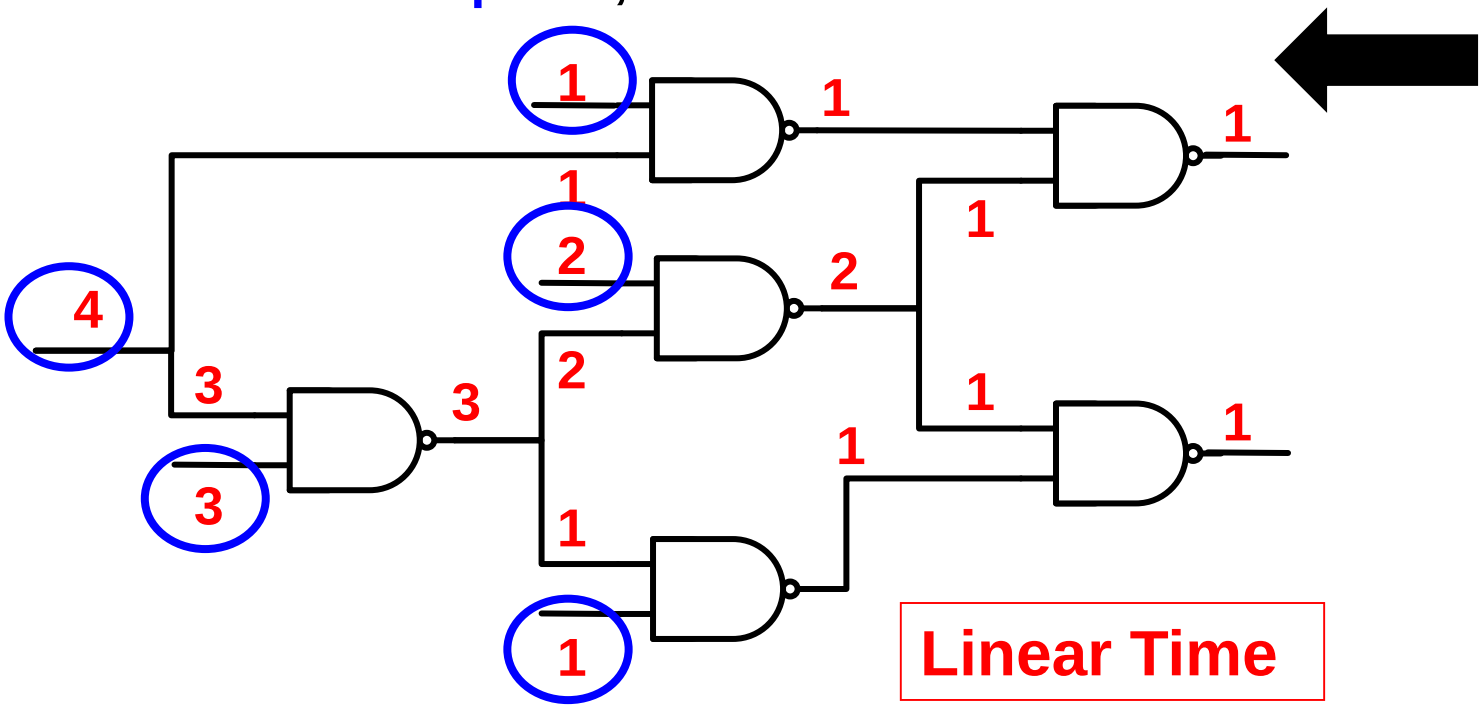
Non-enumerative Method [Pomeranz 94]



- **Three steps**
  1. **Count number of total PDF**
  2. **Count detected PDF for a test pattern**
  3. **Count detected PDF for a test set**

1. Count Number of Paths

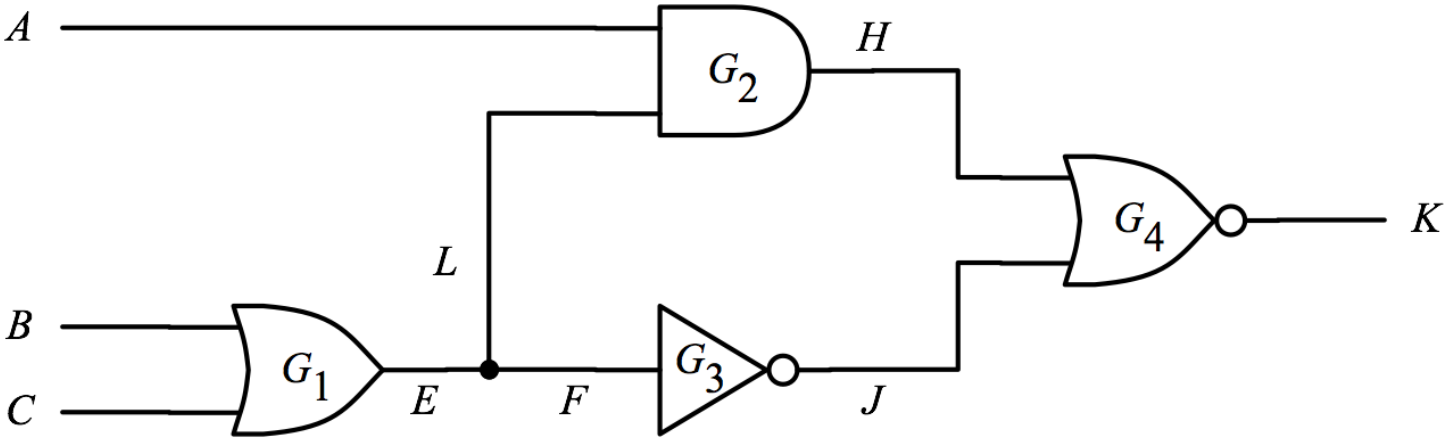
- **Algorithm: Count number of paths backward, from PO to PI**
  - **Each PO = 1. Propagate numbers from gate output to gate inputs**
  - **Fanout stem = sum of branches**
  - **Number at PI = number of paths starting from this PI**
- **Example: ISCAS C17 circuit**
  - **Total 1+4+2+3+1=11 paths, 22 PDF**



Quiz

Q: Please count number of paths. How many path delay faults?

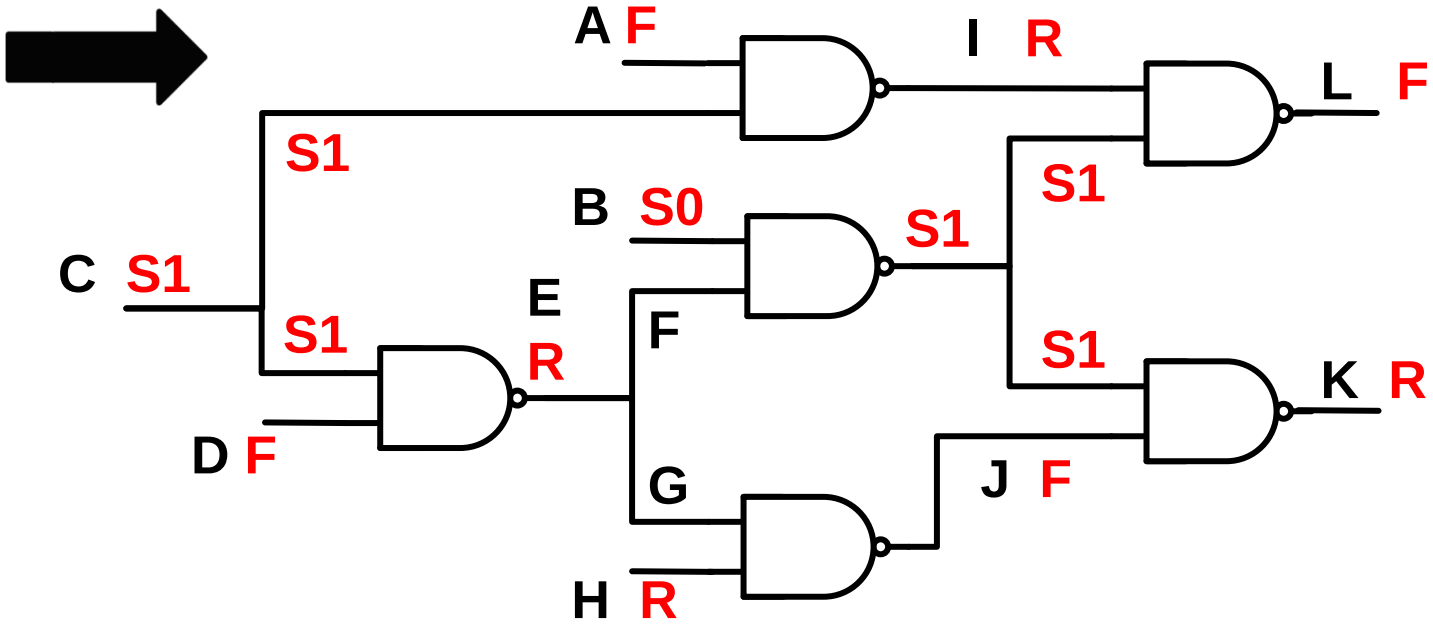
A:



2. Count Detected PDF for a Test (1)

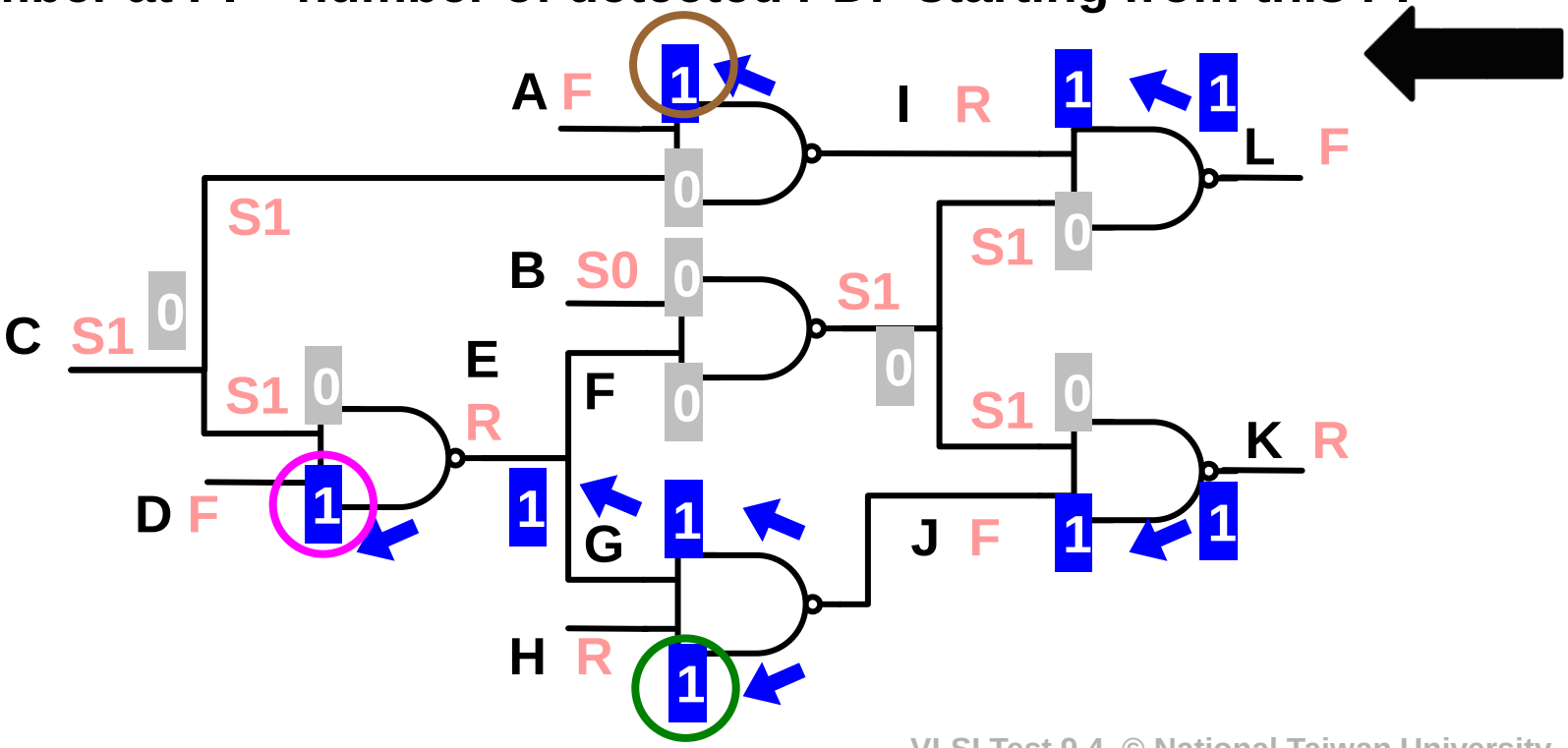
- **Six-valued logic simulation, from PI to PO**
- **Example:**
  - **Apply test pattern T_1. A=F, B=S0, C=S1, D=F, H=R**
  - **How many PDF robustly detected?**

* we demo robust example here.
this technique also applicable to NR.



2. Count Detected PDF for a Test (2)

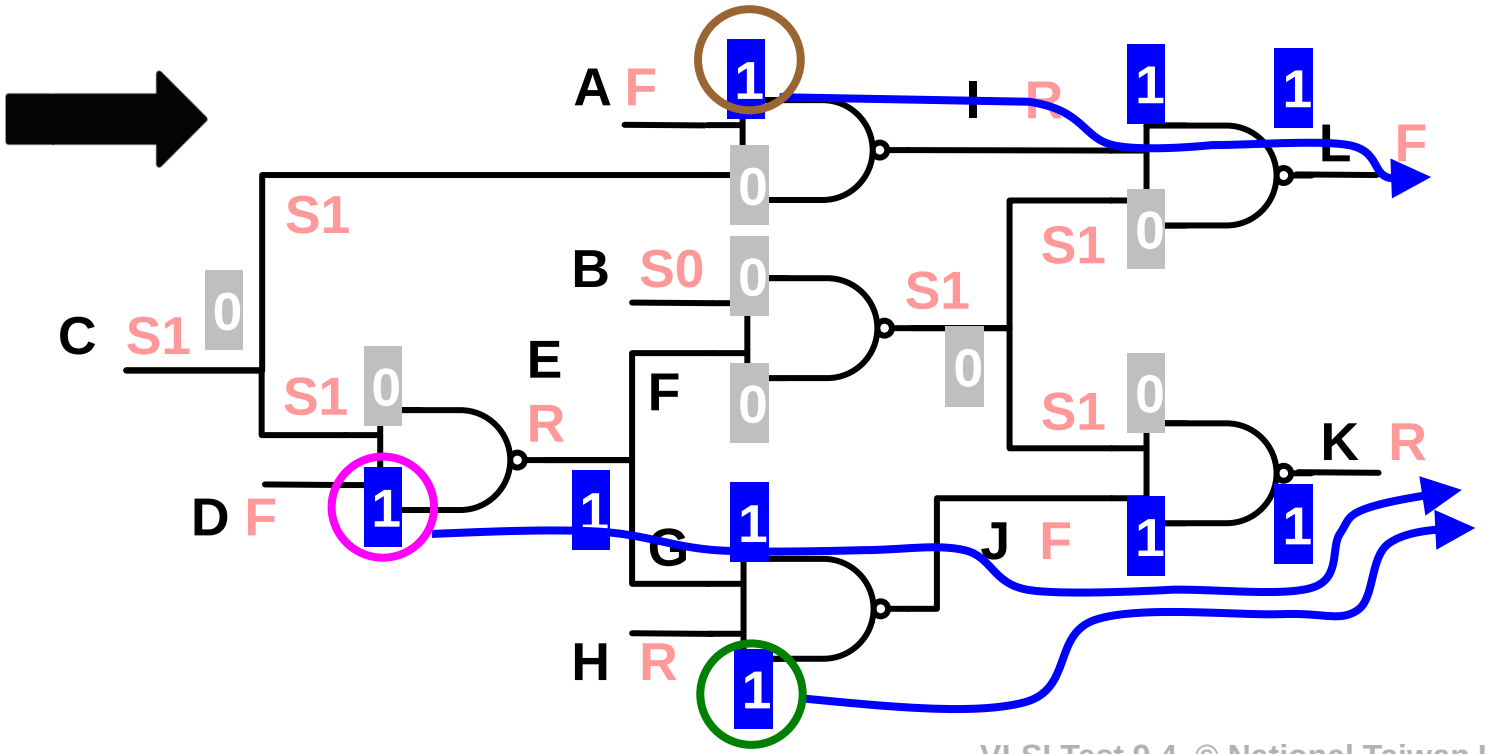
- **Algorithm: Count robustly sensitized paths backward**
  - **If output gate is robustly sensitized, PO=1**
  - **Backward propagate numbers on robustly sensitized gate input**
    - **Otherwise, number is zero**
  - **Fanout stem = sum of branches**
  - **Number at PI = number of detected PDF starting from this PI**



2. Count Detected PDF for a Test (3)

- Trace non-zero numbers forward
- Test pattern T_1 robustly detects three PDF
  - ↓DEGJK, ↑HJK, ↓AIL

* we demo robust example here.
this technique also applicable to NR.



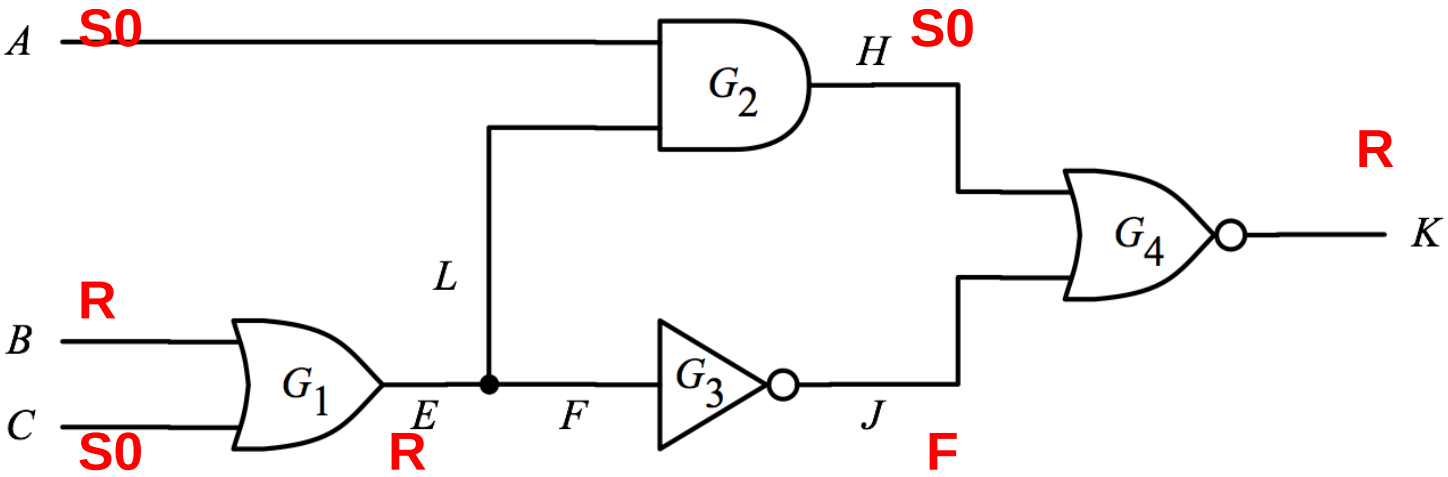
Quiz

Q: Counting backward from PO to PI. How many PDF robustly detected by this test pattern?

A:

Q: What is fault coverage of this test pattern? (Totally 10 PDF.)

A:

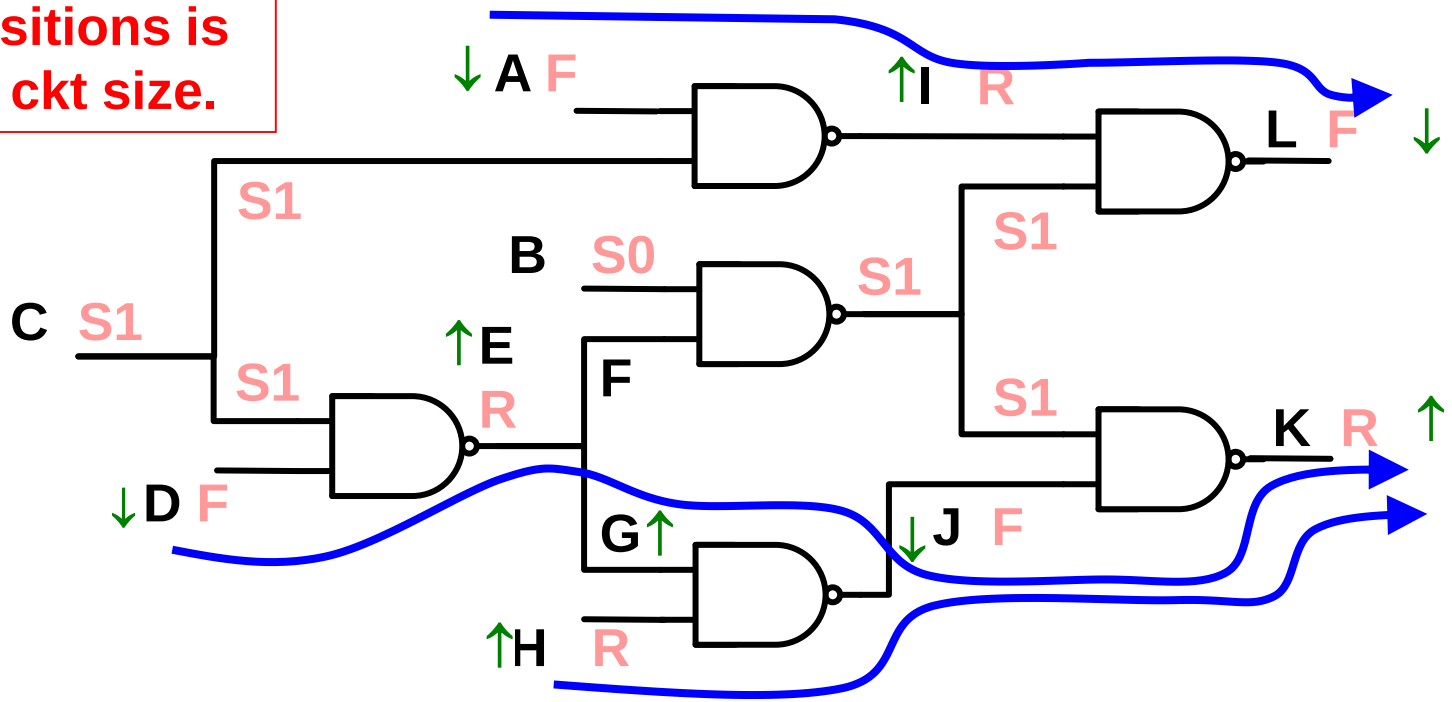


This FC is More Representative than Before.
But What If More than One Test Pattern?

What Information to Store?

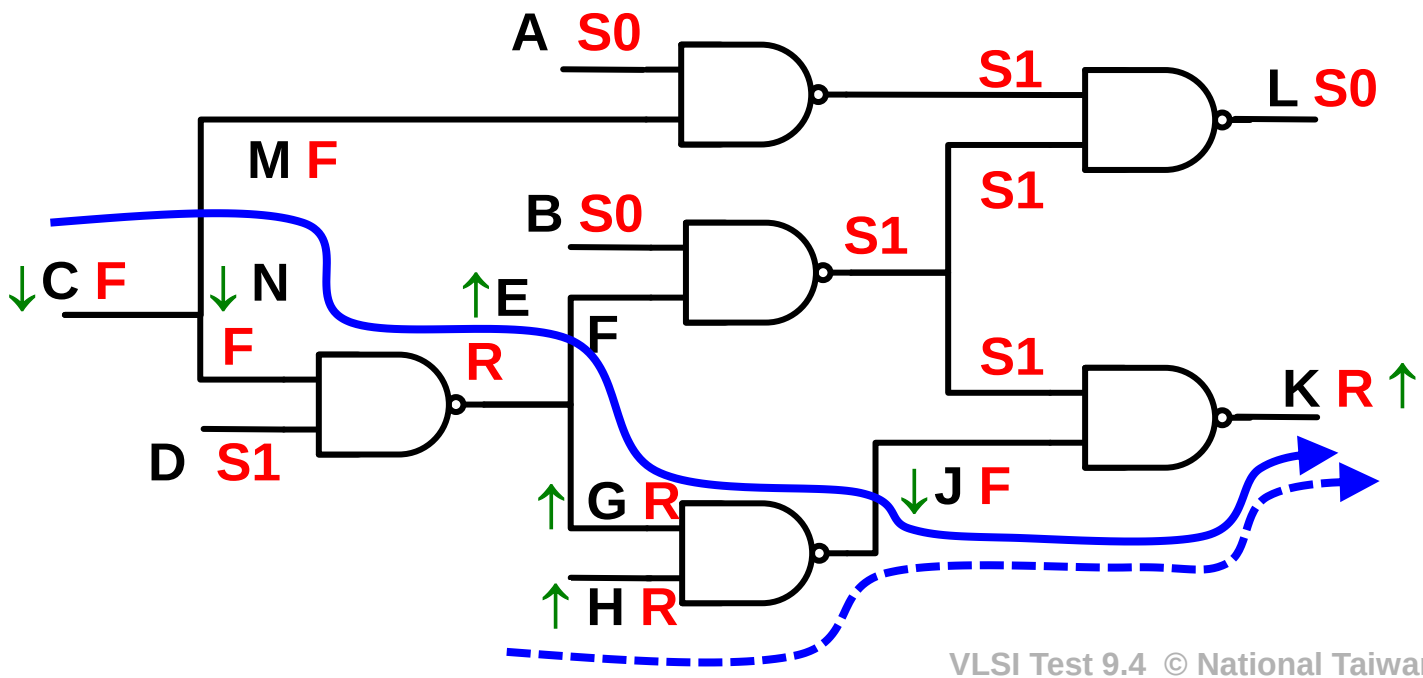
- Q: Can we store detected PDF? {↓AIL, ↓DEGJK, ↑HJK }
  - A: No, because too many paths. Memory can explode!
- Alternative: store transitions of signals on detected paths
- Example: Store 9 transitions for T_1
  - (A,↓) (D,↓) (E,↑)(G,↑)(H,↑)(I,↑) (J,↓)(K,↑)(L,↓)

# of transitions is linear to ckt size.



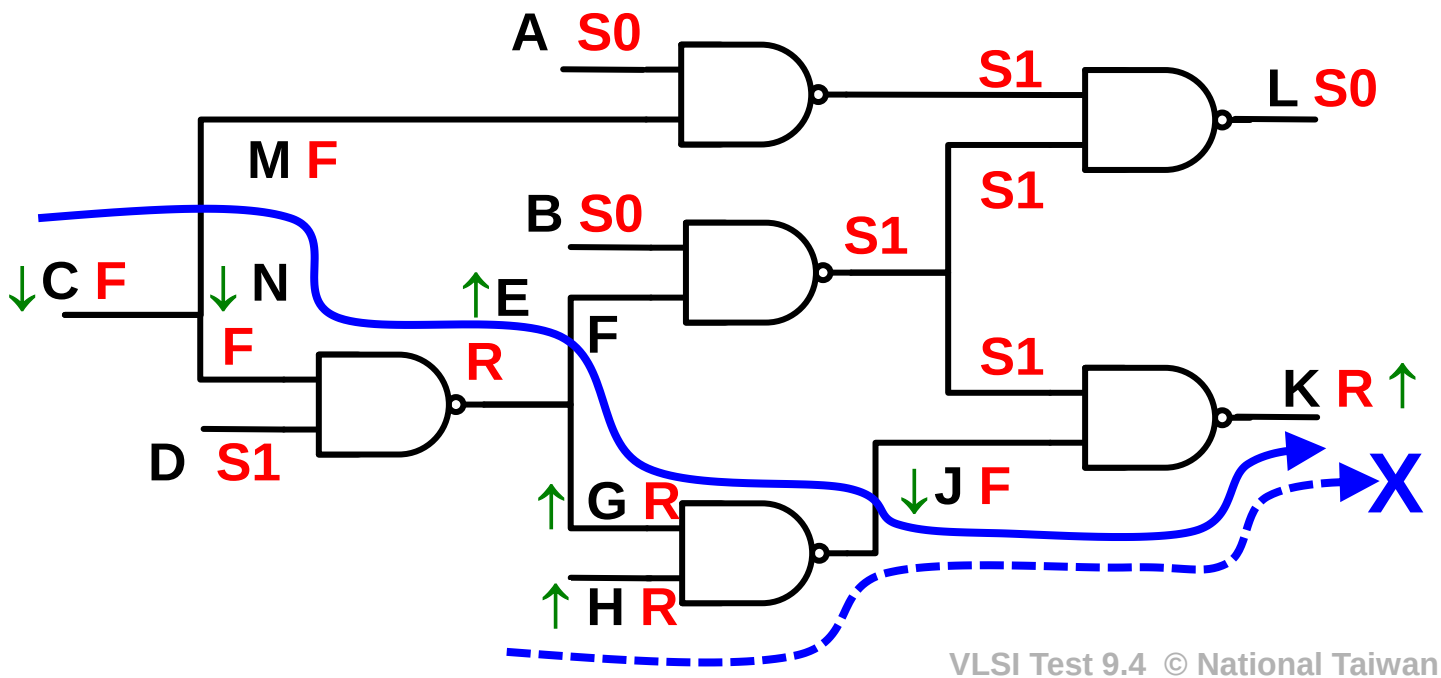
3. Count Detected PDF for a Test Set (1)

- Example: After T_1, apply another test pattern T_2
  - Repeat step 2, T_2 detects PDF: {↓CNEGJK, ↑HJK}
  - Transitions of T_2: (C,↓) (N,↓) (G,↑)(H,↑)(J,↓)(K,↑)
- For test set {T_1 T_2}, fault coverage = (3+2)/22 = 22.7%?
  - No! ↑HJK was detected by T_1. Cannot count twice
  - But we didn't store PDF. How do we know ↑HJK was detected?



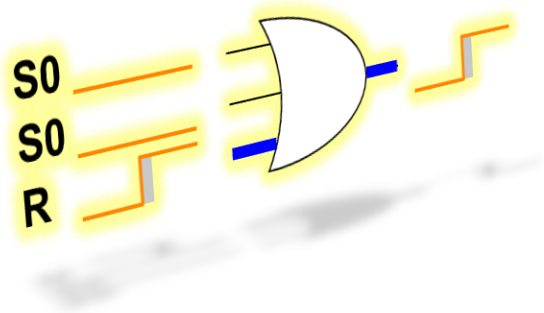
3. Count Detected PDF for a Test Set (2)

- Criterion: To count PDF as newly detected,
  - Require at least one transition did NOT appear previously
- ↓CNEGJK is newly detected by T_2
- ↑HJK is NOT newly detected because all transition appeared in T_1
  - Transitions of T_1: (A,↓) (D,↓) (E,↑)(G,↑)(H,↑)(I,↑)(J,↓)(K,↑)(L,↓)
  - Transitions of T_2: (C,↓) (N,↓) (G,↑)(H,↑)(J,↓)(K,↑)
- For test set {T_1 T_2}, robust fault coverage = (3+1) /22 = 18.1%



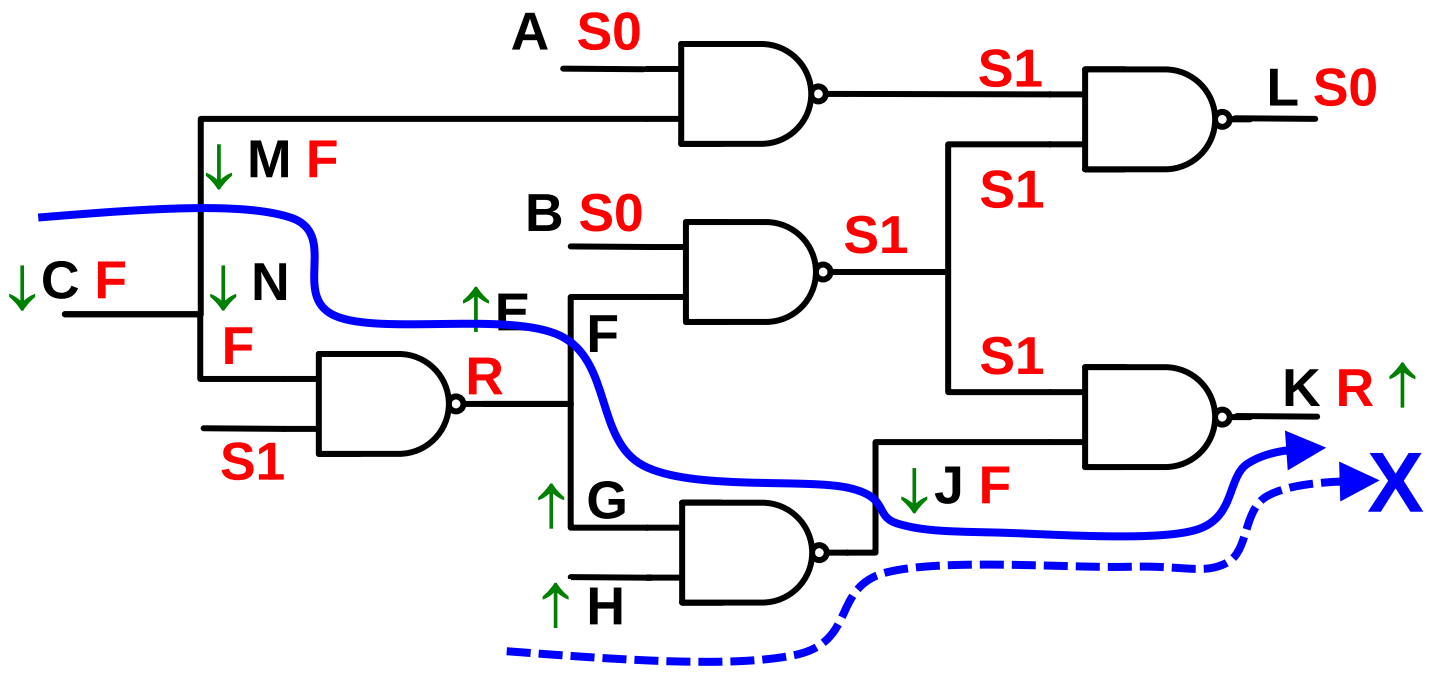
Summary

- Logic system depends on requirement
  - Five-valued logic for robust test pattern generation
  - Six-valued logic for fault simulation (without unknowns)
- Enumerative method
  - Users need to provide fault list
- Non-enumerative method
  - Users don’t need to provide fault list
    1. Counting number of total PDF
    2. Counting number of detected PDF for a test pattern
    3. Counting number of detected PDF for a test set
  - Store transitions, instead of PDF
    - Memory usage won’t go exponential



FFT

- Q: Storing transitions, instead of PDF, can result in pessimistic FC
  - Simulation FC can be lower than actual FC
  - Why? please show example



FFT2

- Q: In this video, we use robust test as examples.
  - Can we apply 6-valued logic to simulate non-robust path delay fault coverage? (given fully specified test patterns)