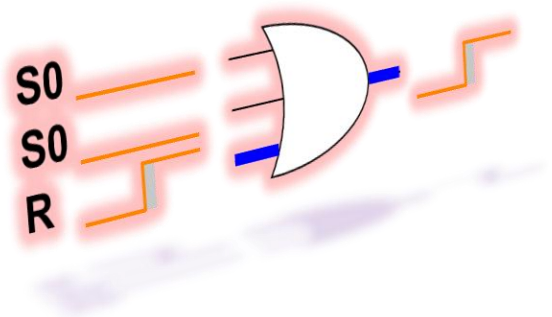
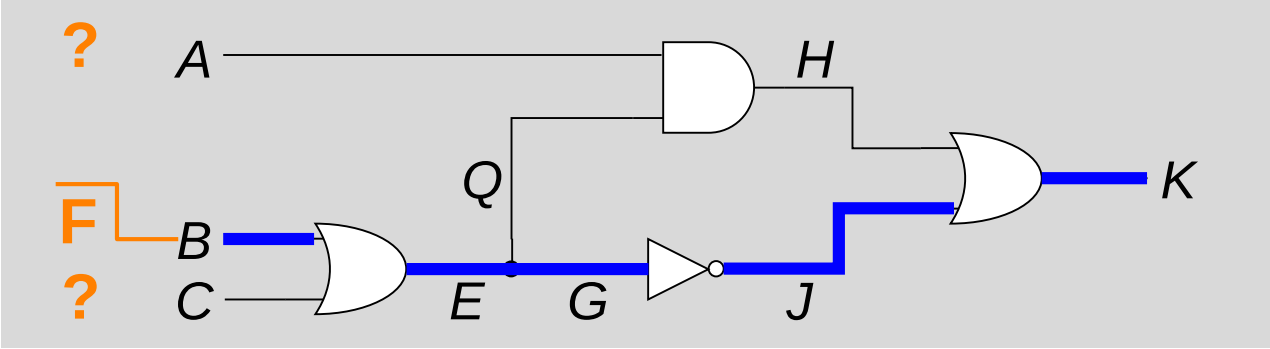
Delay Test

- Introduction and delay fault models
- **Path Delay Fault**
  - Path Sensitization
  - **Test Generation [Lin 1987]**
    - **5-valued logic**
    - **Hazard-free TPG, Robust TPG**
    - **Backtrack Examples**
  - Fault Simulation [Smith 1985] [Pomeranz 1984]
- Transition Delay Fault
- Experimental Results* (not in exam)
- Issues of Delay Tests* (not in exam)
- Conclusions



Motivating Problem

- How to generation a test pattern for PDF ↓BEGJK ?
  - Q1: What sensitization condition?
    - A: Consider hazard-free and robust tests in this lecture
    - Non-robust test NOT considered
  - Q2: What logic system?
  - Q3: How to backtrace?
  - Q4: How to backtrack?



Review: 9-valued Logic

- Used in path sensitization (see 9.2)

AND	S0	S1	R	U1	1*	F	U0	0*	XX
S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
S1	S0	S1	R	U1	1*	F	U0	0*	XX
R	S0	R	R	U1	R	0*	0*	0*	XX
U1	S0	U1	U1	U1	U1	U0	U0	0*	XX
1*	S0	1*	R	U1	1*	F	U0	0*	XX
F	S0	F	0*	U0	F	F	U0	0*	U0
U0	S0	U0	0*	U0	U0	U0	U0	0*	U0
0*	S0	0*	0*	0*	0*	0*	0*	0*	0*
XX	S0	XX	XX	XX	XX	U0	U0	0*	XX

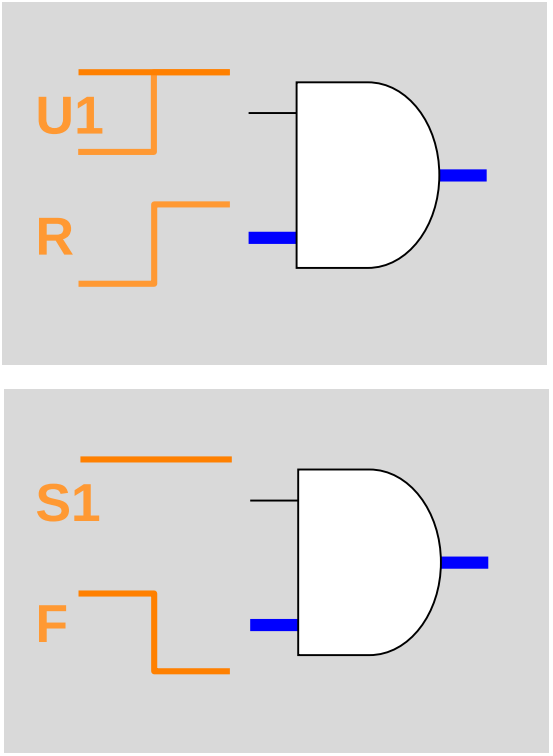
Value	Meaning
S0	Static 0
S1	Static 1
R	Rising
F	Falling
0*	Static-0 hazard
1*	Static-1 hazard
U1	X→1
U0	X→0
XX	X→X

Too Many States. Can We Reduce ?

Remove 0*, 1*

- **0*** and **1*** are NOT needed for Hazard-free/Robust test generation

AND	S0	S1	R	U1	1*	F	U0	0*	XX
S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
S1	S0	S1	R	U1	1*	F	U0	0*	XX
R	S0	R	R	U1	R	0*	0*	0*	XX
U1	S0	U1	U1	U1	U1	U0	U0	0*	XX
1*	S0	1*	R	U1	1*	F	U0	0*	XX
F	S0	F	0*	U0	F	F	U0	0*	U0
U0	S0	U0	0*	U0	U0	U0	U0	0*	U0
0*	S0	0*	0*	0*	0*	0*	0*	0*	0*
XX	S0	XX	XX	XX	XX	U0	U0	0*	XX



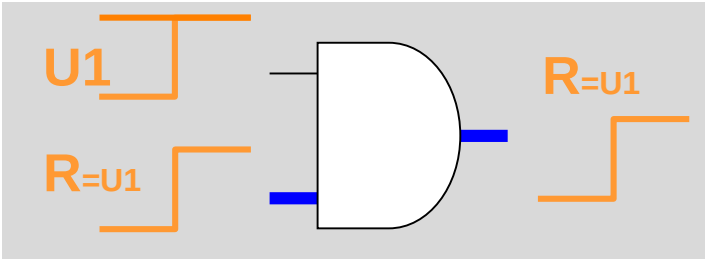
robust sensitization

Reduced to 7-valued Logic

Merge: R≡U1 F≡U0

- R=01 for on-path signals. U1 = X1 for off-path signals
- F=10 for on-path signals. U0 = X0 for off-path signals

AND	S0	S1	R	U1	F	U0	XX
S0	S0	S0	S0	S0	S0	S0	S0
S1	S0	S1	R	U1	F	U0	XX
R	S0	R	R	U1	U0	U0	XX
U1	S0	U1	U1	U1	U0	U0	XX
F	S0	F	U0	U0	F	U0	U0
U0	S0	U0	U0	U0	U0	U0	U0
XX	S0	XX	XX	XX	U0	U0	XX



AND	S0	S1	(R) U1	(F) U0	XX
S0	S0	S0	S0	S0	S0
S1	S0	S1	U1	U0	XX
U1	S0	U1	U1	U0	XX
U0	S0	U0	U0	U0	U0
XX	S0	XX	XX	U0	XX

Reduced to 5-valued Logic

Quiz

Q: Use 5-valued logic. Please fill in truth tables for OR, NOT.

A:

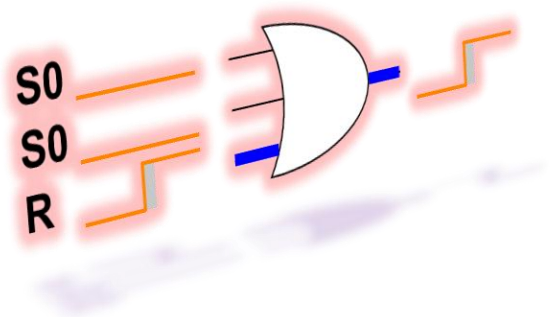
OR	S0	S1	U1	U0	XX
S0					
S1					
U1					
U0					
XX					

AND	S0	S1	U1	U0	XX
S0	S0	S0	S0	S0	S0
S1	S0	S1	U1	U0	XX
U1	S0	U1	U1	U0	XX
U0	S0	U0	U0	U0	U0
XX	S0	XX	XX	U0	XX

NOT	S0	S1	U1	U0	XX

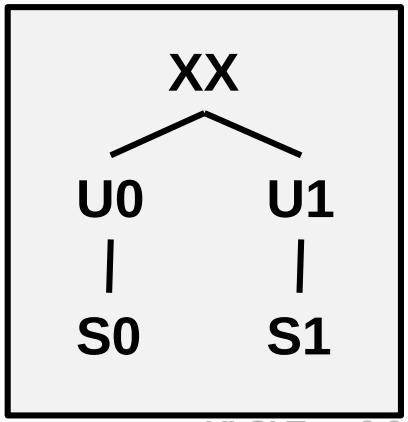
Delay Test

- Introduction and delay fault models
- **Path Delay Fault**
  - Path Sensitization
  - **Test Generation [Lin 1987]**
    - **5-valued logic**
    - **Hazard-free TPG, Robust TPG**
    - **Backtrack examples**
  - Fault Simulation [Smith 1985] [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



Test Gen. Using 5-valued Logic [Lin 87]

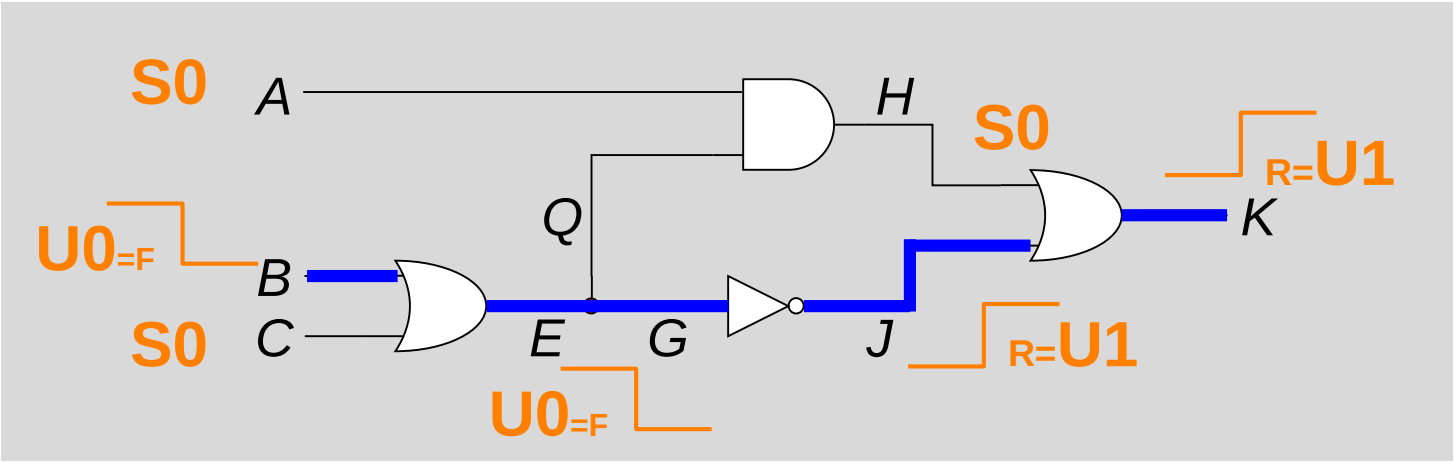
- Test gen for PDF is similar to SSF, except
  1. X-path selection is not needed since path is already decided
  2. Two different zeros and ones: U0 and S0. U1 and S1
- How to backtrace?
  - Backtrace U0 if objective is U0
  - Backtrace S0 if objective is S0
- Covering Tree
  - U0 *covers* S0, which means
    - OK if objective is U0 and simulated value is S0
    - Conflict if objective is S0 but simulated value is U0
  - U1 covers S1
  - XX covers both U0 and U1



Hazard-free TPG Example

- PODEM-like algorithm as an example
- Generate hazard-free test pattern for PDF: ↓BEGJK
  - Assign B = U0
  - Objective: C = S0, backtrace S0, assign C = S0
    - Simulate: E = G = Q = U0, J = U1
  - Objective: H = S0, backtrace S0, assign A = S0
    - Simulate: K = U1, reach PO
  - Test generated: (A,B,C) = (S0, U0, S0)
    - V_1 = 010, V_2 = 000

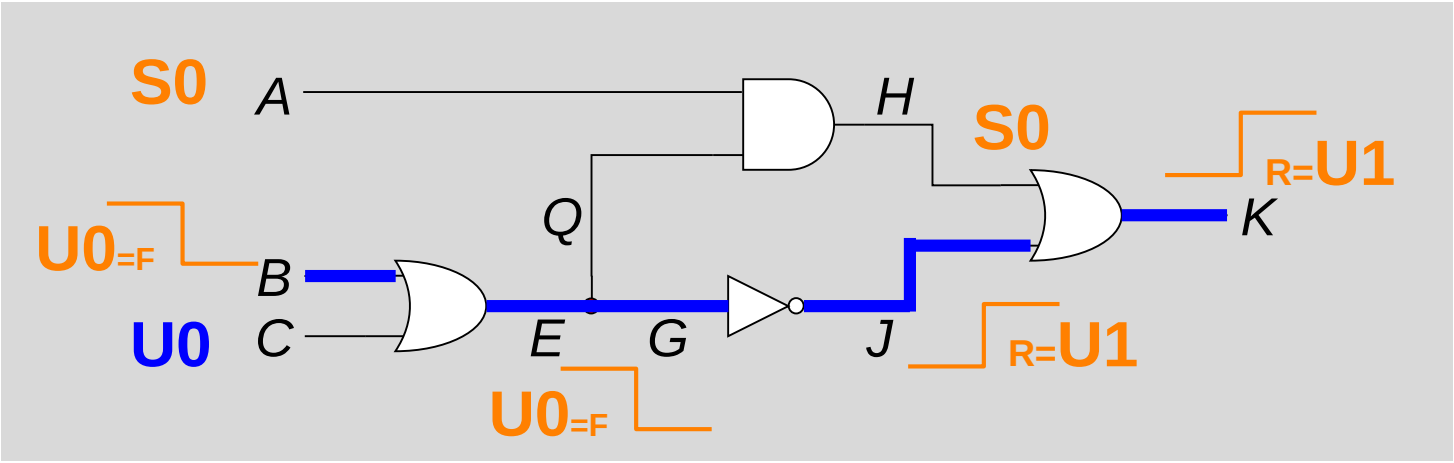
OR	S0	S1	U1	U0	XX
S0	S0	S1	U1	U0	XX
S1	S1	S1	S1	S1	S1
U1	U1	S1	U1	U1	U1
U0	U0	S1	U1	U0	XX
XX	XX	S1	U1	XX	XX



Robust TPG Example

- Generate robust test pattern for PDF: ↓BEGJK
  - Assign B = U0
  - Objective: C = U0, backtrace U0, assign C=U0
    - Simulate E = G = Q = U0,J = U1
  - Objective: H = S0, backtrace S0, Assign A=S0
    - Simulate: K = U1, reach PO
  - Test generated: (A,B,C) = (S0, U0, U0)
    - V_1 = 01X, V_2 = 000

OR	S0	S1	U1	U0	XX
S0	S0	S1	U1	U0	XX
S1	S1	S1	S1	S1	S1
U1	U1	S1	U1	U1	U1
U0	U0	S1	U1	U0	XX
XX	XX	S1	U1	XX	XX

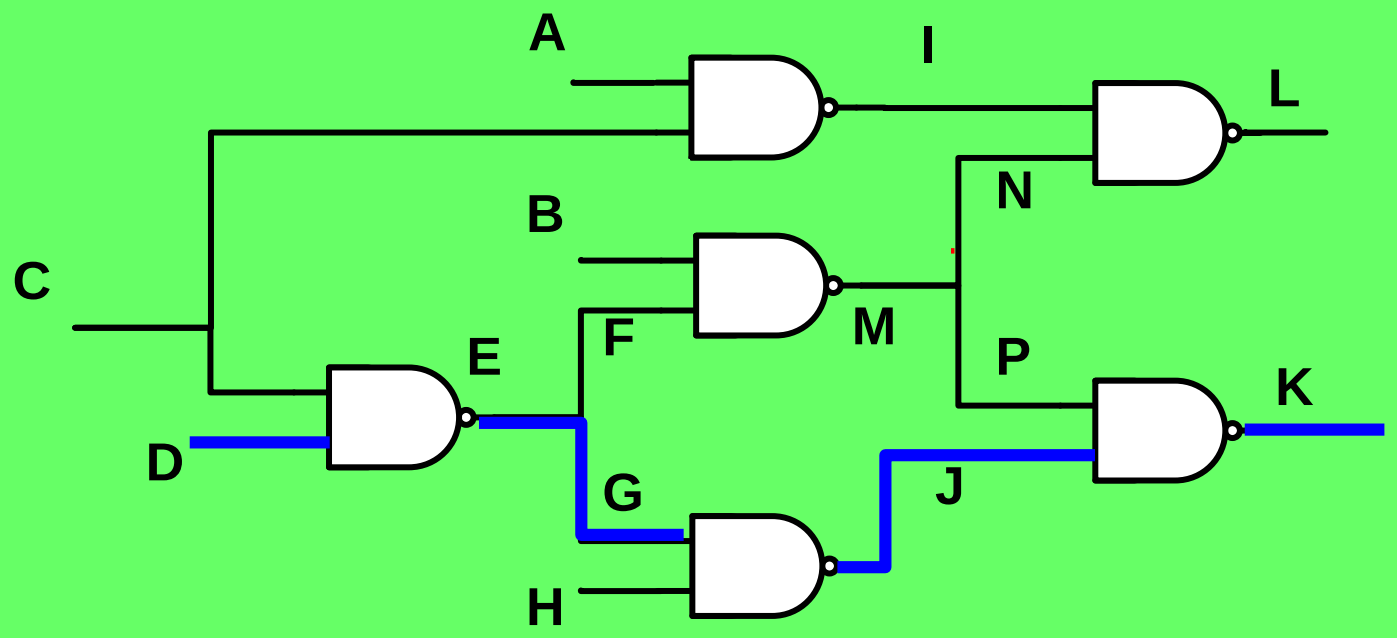


Quiz

Q: Please generate a robust test for PDF ↓DEGJK

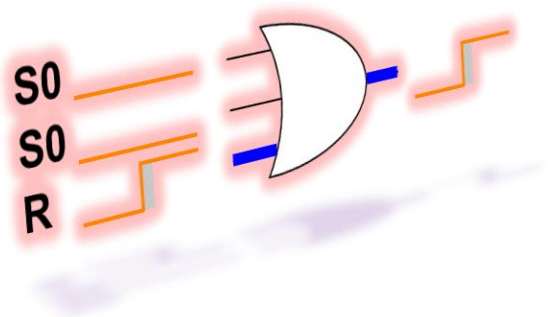
A:

NA / ND	S0	S1	U1	U0	XX
S0	S1	S1	S1	S1	S1
S1	S1	S0	U0	U1	XX
U1	S1	U0	U0	U1	XX
U0	S1	U1	U1	U1	U1
XX	S1	XX	XX	U1	XX



Delay Test

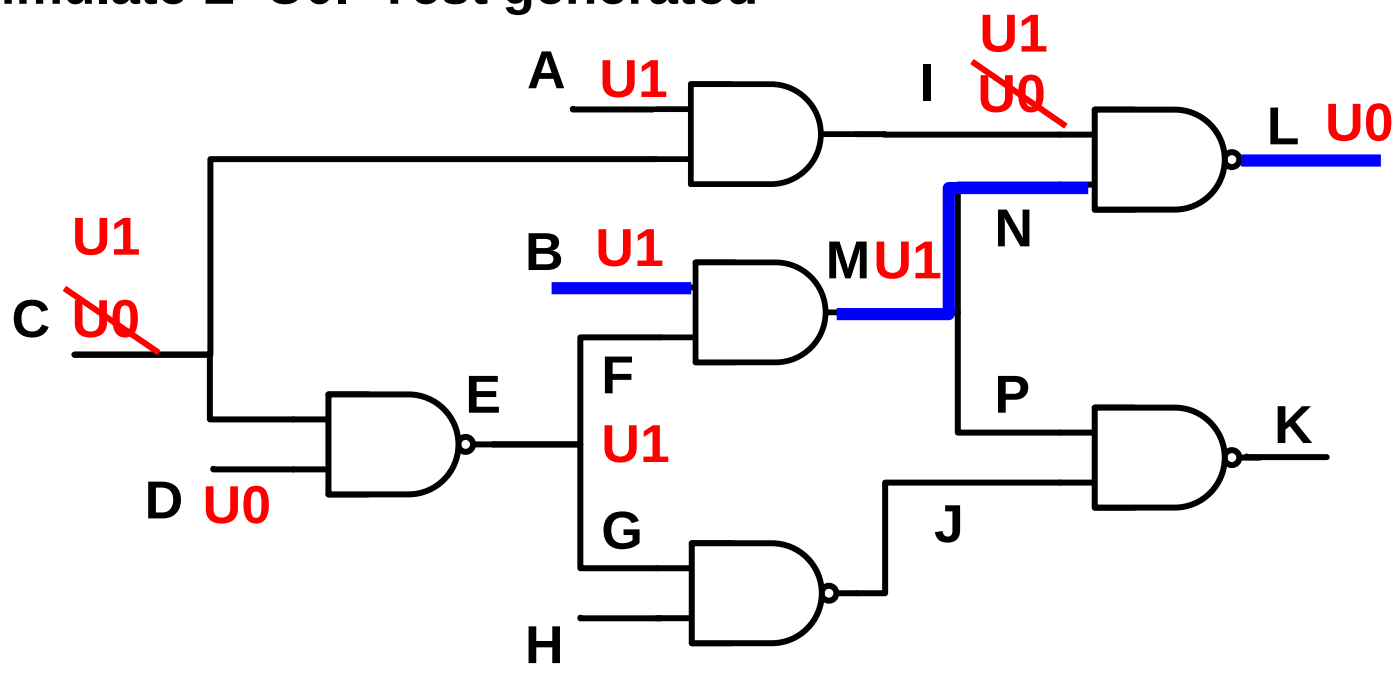
- Introduction and delay fault models
- **Path Delay Fault**
  - Path Sensitization
  - **Test Generation [Lin 1987]**
    - **5-valued logic**
    - **Hazard-free TPG, Robust TPG**
    - **Backtrack Examples**
  - Fault Simulation [Smith 1985] [Pomeranz 1984]
- Transition Delay Fault
- Delay Test Application
- Circuit Model for Delay Test ATPG
- Experimental Results* (not in exam)
- Issues of Delay Tests* (not in exam)
- Conclusions



Backtrack Example

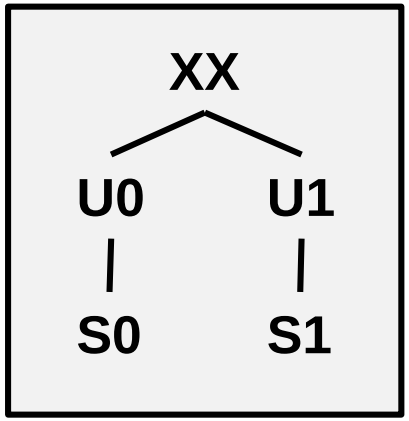
- Generate a robust test pattern for PDF ↑BMNL
- Assign B=U1
- Objective: F=U1. Backtrace C=U0
  - Simulate I=U0, block PDF. Backtrack C=U1
- Objective: F=U1. Backtrace D=U0
- Objective: I=U1. Backtrace A=U1
  - Simulate L=U0. Test generated

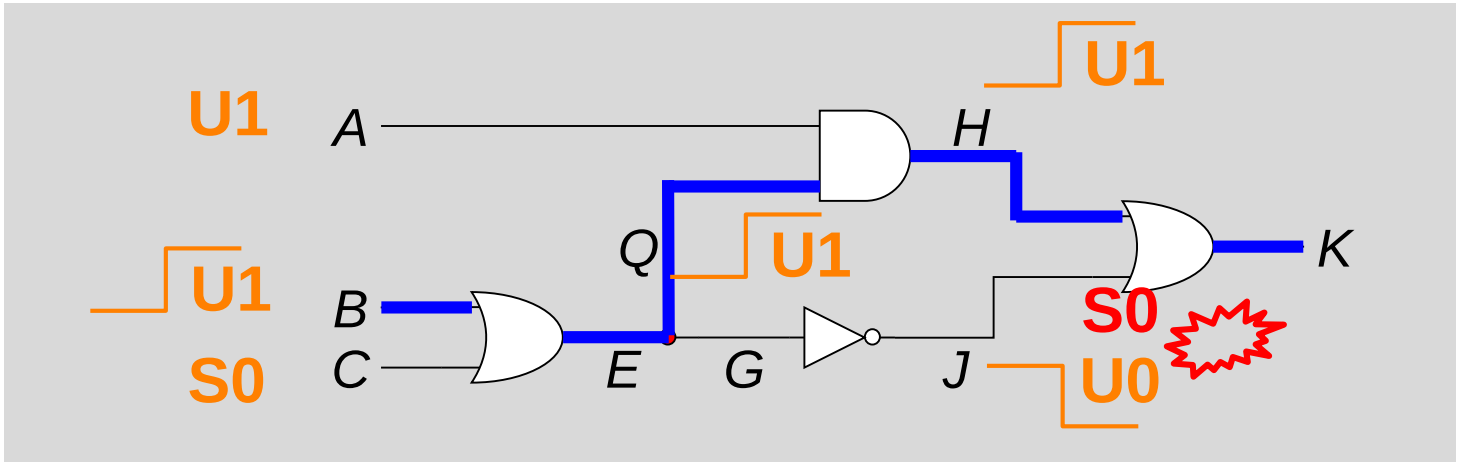
Backtrack U0 to U1, not to S1 or S0.
Backtrack S0 to S1, not to U0 or U1.
Keep decision tree always 2 branches.



Robustly Untestable Example

- Generate a robust test pattern for PDF ↑BEQHK
  - Assign B = U1
  - Objective: C = S0 , backtrace S0, assign C=S0
    - Simulate: E = U1, G = U1, J = U0
  - Objective: A = U1, backtrace U1, assign A=U1
    - Simulate: H = U1
  - Objective value: J=S0 but simulated value J = U0. conflict!
  - Backtrack. A=U0. Backtrack C=S1. Test gen Fail.
  - This PDF is *robustly untestable*

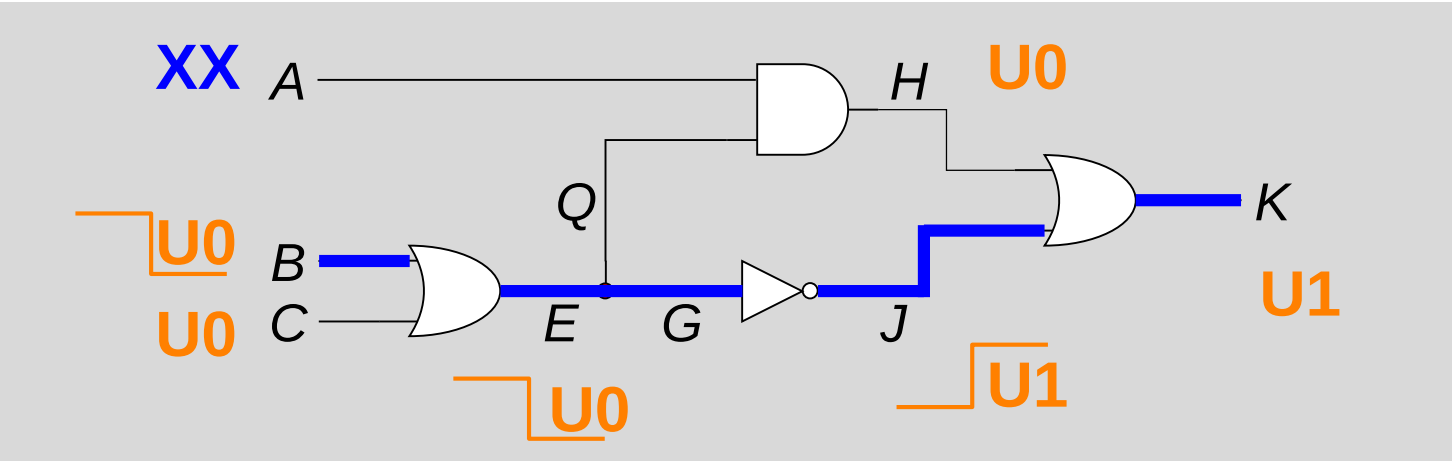




Non-robust TPG Example

- Generate a non-robust test pattern for PDF ↓BEGJK
  - Assign B = U0
  - Objective: C = U0, backtrace U0, Assign C=U0
    - Simulate: E = G = Q = U0, J = U1, H=U0, K=U1
  - Test generated: (A,B,C) = (XX, U0, U0)
    - V_1 = X1X, V_2 = X00

* Generally, 5-vlued logic is NOT sufficient to generate non-robust tests (see FFT).



Summary of PDF ↓BEGJK

- Total 8 test patterns to detect PDF ↓BEGJK

6 Non-robust tests

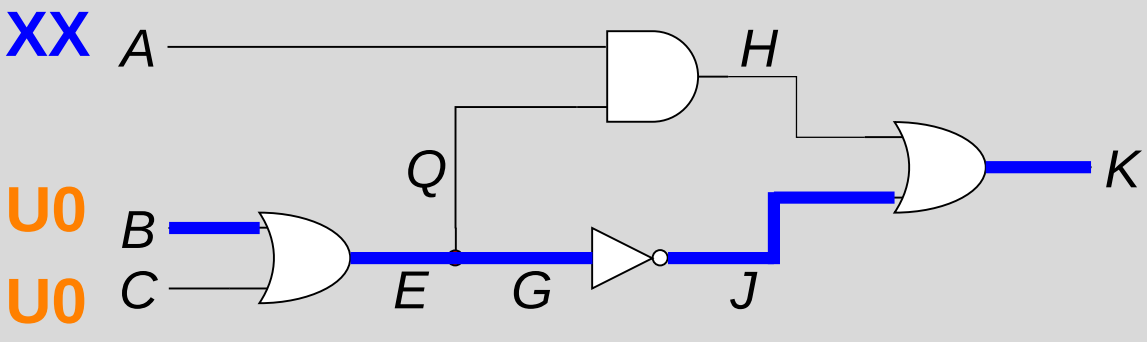
V1 = 010, V2 = 100
V1 = 011, V2 = 100
V1 = 111, V2 = 100
V1 = 111, V2 = 000
V1 = 110, V2 = 000
V1 = 110, V2 = 100

2 Robust tests

V1 = 011, V2 = 000

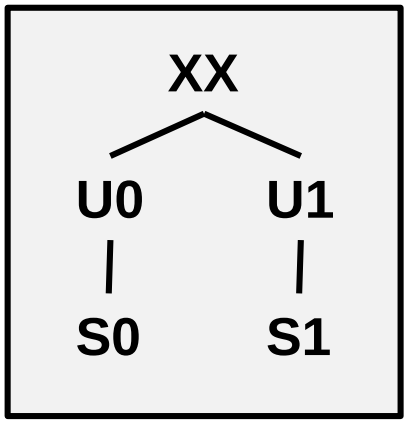
$V_1 = 010, V_2 = 000$

1 Hazard-free test (also robust test)



Summary

- 5-valued logic used to generate hazard-free/robust test patterns
  - S0, S1, U0(=F), U1(=R), XX
  - U0 covers S0, but S0 does not cover U0
- Backtrace/Backtrack same type of zero/ones
  - Backtrace U0 if objective is U0. Backtrace S0 if objective is S0.
  - Backtrack to U1 if original is U0. Backtrack to S1 if original is S0.
- Many PDF are robustly untestable
  - For testable PDF, many more non-robust tests than robust test



FFT

- **Q: 5-valued logic not sufficient to generate NR tests, why?**
  - **HINT: K can be static-1 hazard, or Rising**