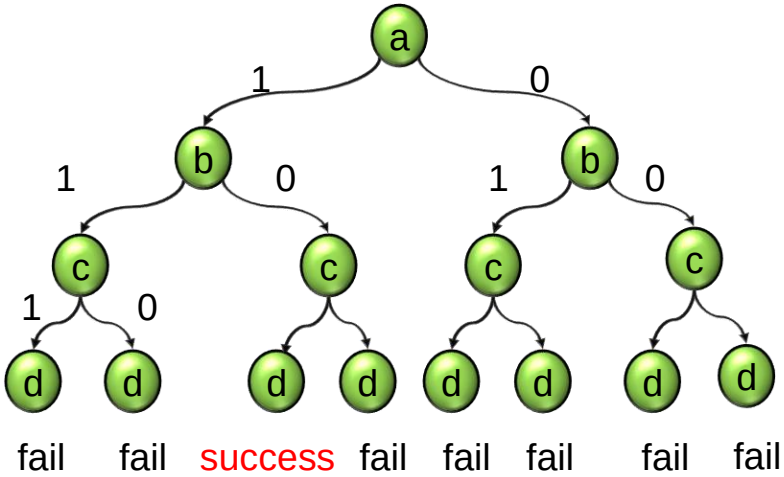
Combinational ATPG

- Introduction
- **Deterministic Test Pattern Generation**
  - Boolean difference *
  - **Path sensitization ****
  - D-Algorithm**
  - PODEM**
  - FAN**
  - SAT-based *
- Acceleration Techniques
- Concluding Remarks

Two ATPG categories:

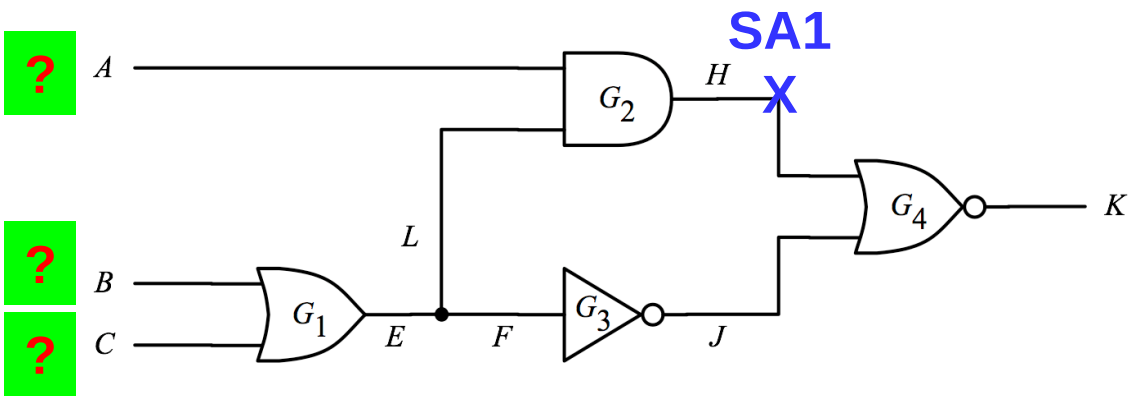
***Boolean-based methods**

****Path-based methods**



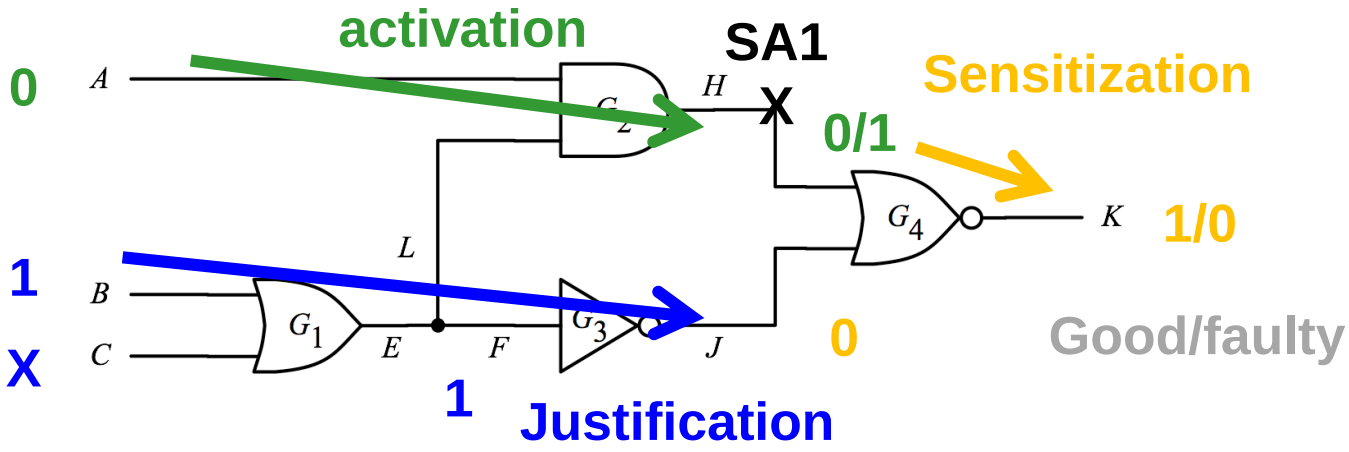
Motivating Problem

- We do not need to know Boolean expression
  - we can just find a test from circuit netlist



Let's Analyze What We Did

- ***Fault activation:*** **Assign gate inputs to generate appropriate value at fault site (*H*)**
  - ***A=0***
- ***Sensitization:*** **Assign side-inputs to non-controlling value to propagate fault effect forward**
  - ***J=0***
- ***Justification:*** **Assign primary inputs to achieve desired values**
  - ***B=1***



Single Path Sensitization

- **Single path sensitization (SPS) Algorithm:**

① ***Fault activation*** **(aka.** ***Fault excitation*****)**

Assign gate inputs to generate value at fault site

Desired value opposite to the faulty value (*e.g.* 0 for SA1)

② ***Fault effect propagation*****:**

Select one single path from fault site to an output

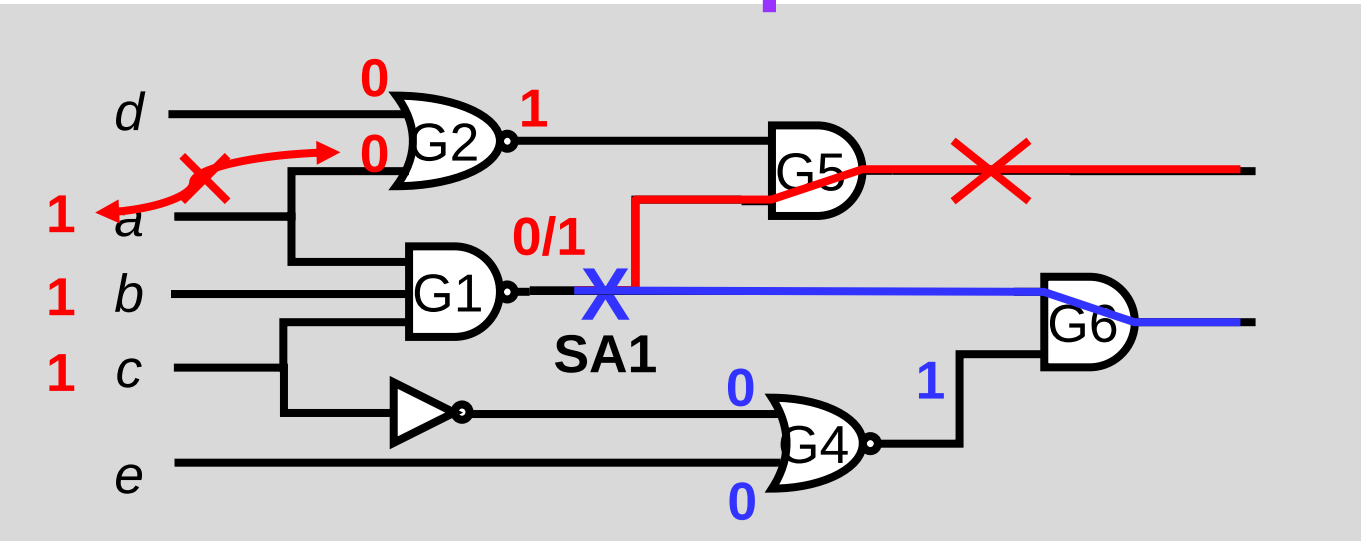
Assign side inputs to sensitize fault effect along the path

③ ***Justification*****:**

Assign primary inputs to justify desired values assigned in ①&②

If justification fails, backtrack.

Example



Consider stuck-at-1 fault

① **Fault activation**

- ***a* = *b* = *c* = 1**

② **Fault effect propagation: two propagation paths**

- **Choose path {G5}. Want G2 = 1**

③ ***a* = *d* = 0 → justification fails!**

② **Backtrack! Choose another path {G6}. Want G4 = 1**

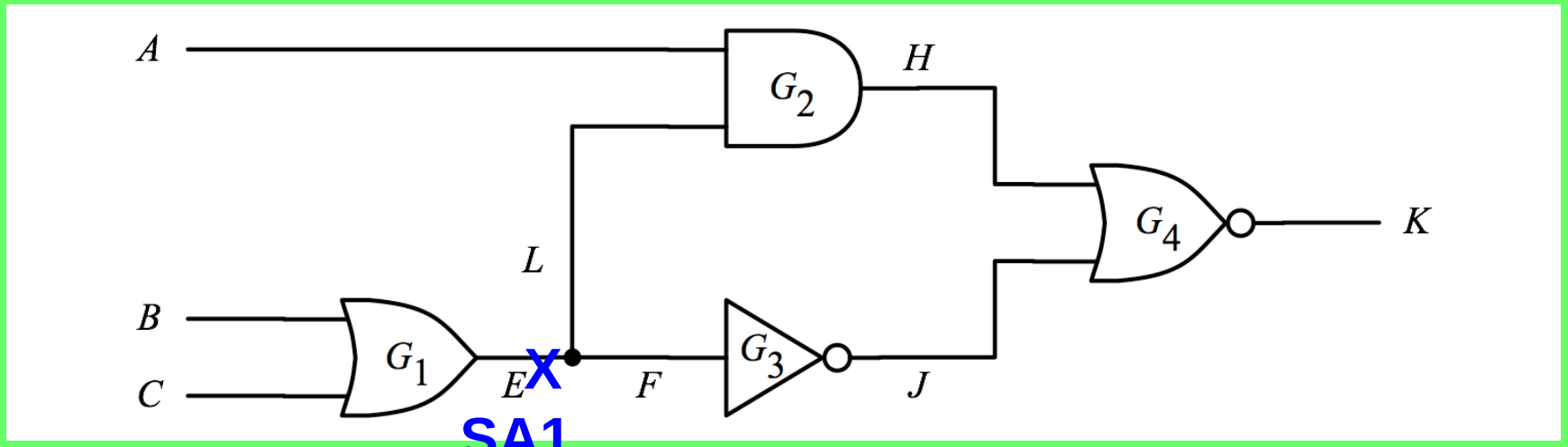
③ ***c* = 1, *e* = 0 → justification succeeds**

- **test pattern: *abce'* generated**

Quiz

Q1: Generate a test pattern for *E* SA1 fault. Choose path *ELHK*.
A:

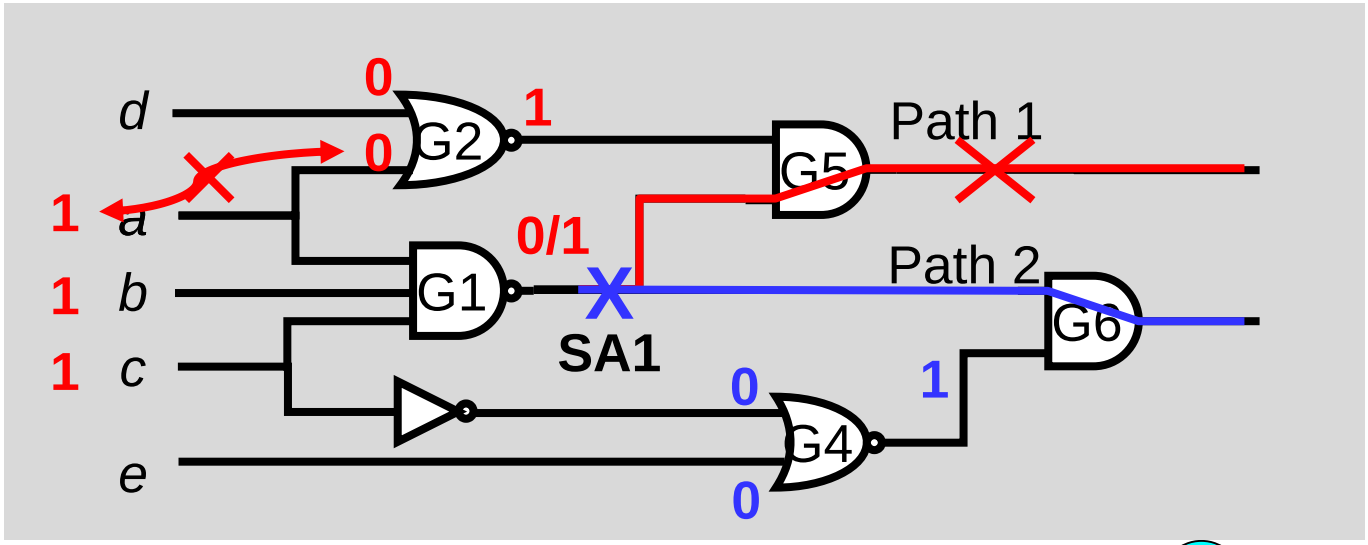
Q2: (Cont'd) Backtrack to another path *EFJK*.
A:



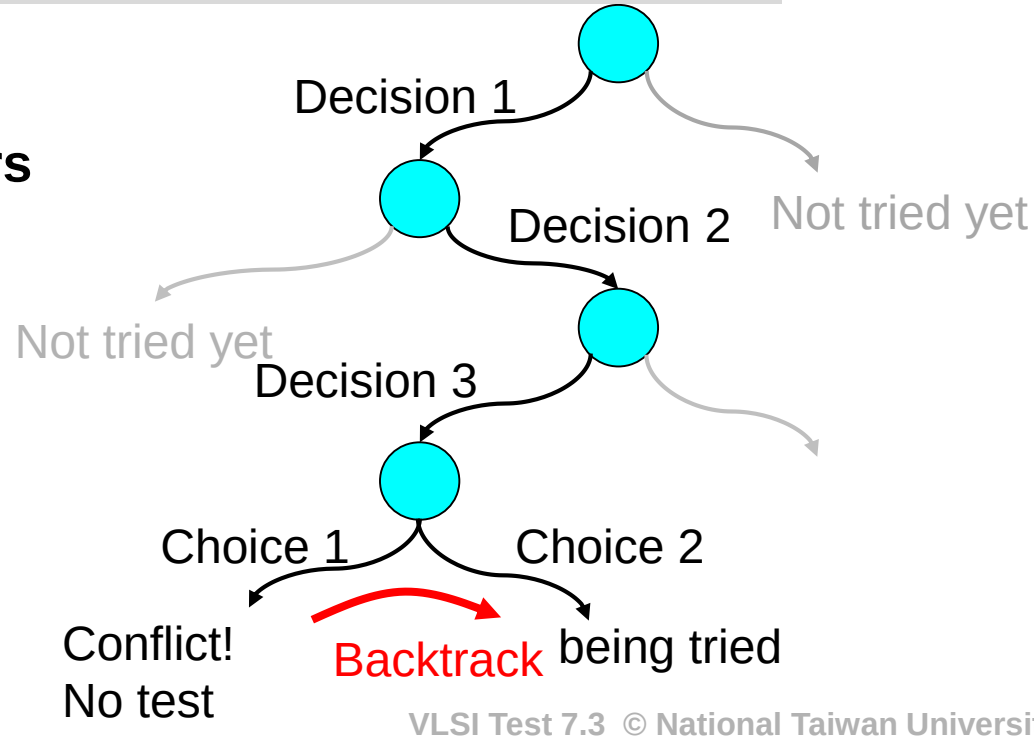
Important to Choose Correct Path

But it is difficult…

Backtrack



- **When we made a mistake in decision tree, conflict occurs**
  - **Go back to a previous decision point**
  - **Change decision**
  - **Redo the rest**



Pros and Cons

☺ **Pros**

- **Easy to implement**
- **No Boolean equation needed**

☹ **Cons**

- **Q1: Too many paths to choose, which one is correct?**
- **Q2: Single-path sensitization not enough to detect all faults**

- **Single path sensitization (SPS) Algorithm:**

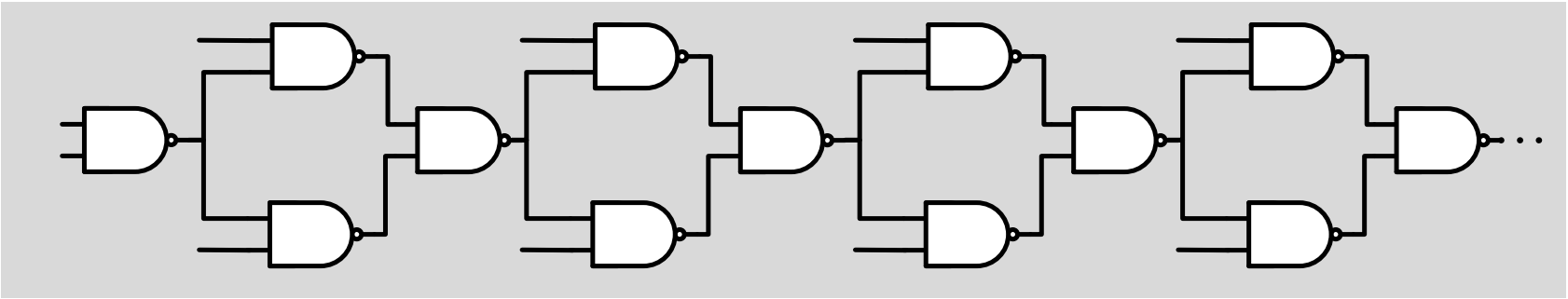
① ***Fault activation*** **(aka.** ***Fault excitation*****)**
Assign internal signals to generate value at fault site
Desired value opposite to the faulty value

② ***Fault effect propagation*****:**
Select **one single path** from fault site to an output
Assign internal signals to sensitize fault effect along the path

③ ***Justification*****:**
Assign primary inputs to justify signal values assigned in ①&②
If justification fails, **backtrack**.

Q1: Too Many Paths !

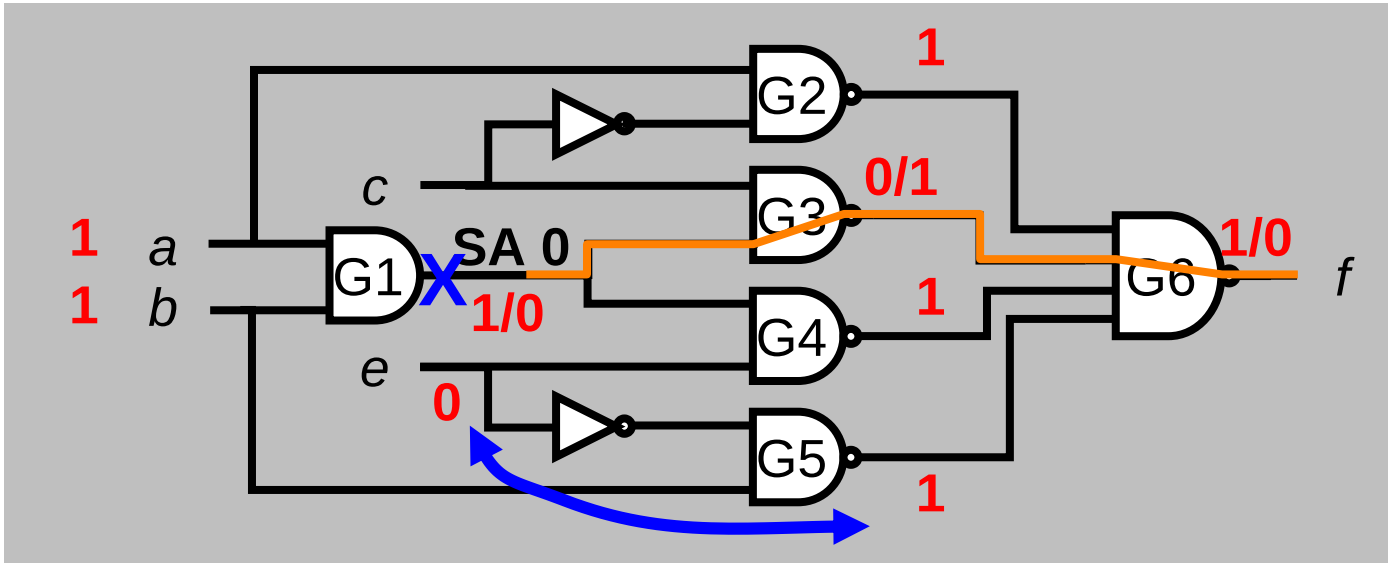
- How to choose correct path?
  - No smart algorithm
  - Simple idea: exhaustively try all paths
- How many paths in a circuit?
  - Worst case example: $3n$ gates, 2^n paths! (n = # of stages)



# of Paths is Exponential to Circuit Size!
Impossible to Try All

Q2: Single Path Not Enough

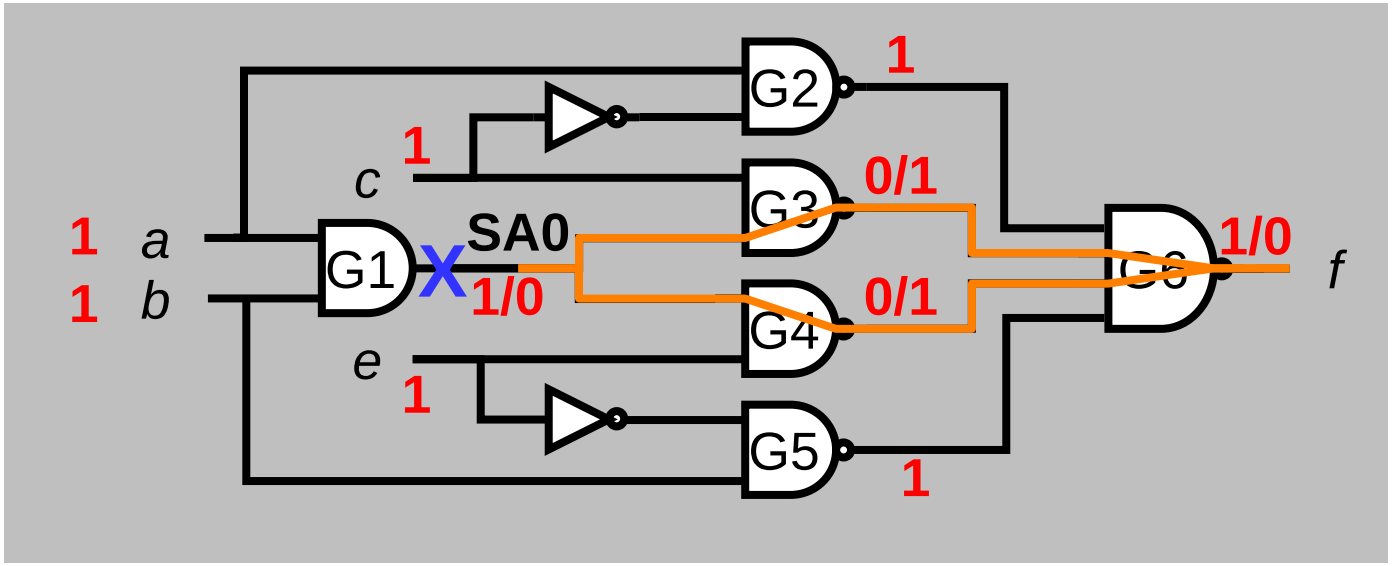
- **Example**
  - ① **Fault activation: *a* = *b* = 1**
  - ② **Fault propagation: Choose path {G3-G6}. want G2 = G4 = G5 = 1**
  - ③ **G4 = 1 → *e* = 0 → G5 = 0 → justification fails**
  - ② **Choose another path {G4-G6}. Justification also fails.**
  - ♦ **SPS algorithm fails**



This Fault Is Actually Testable. Why SPS Fail?

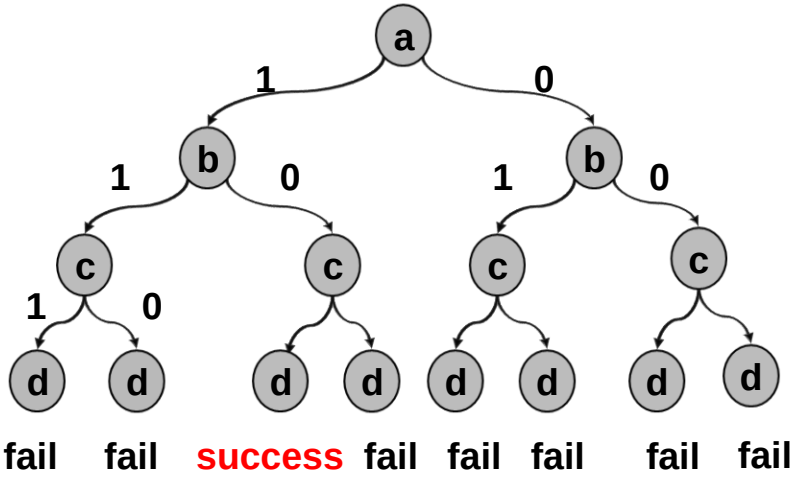
Solution: Multiple Path Sensitization

- This fault requires *multiple path sensitization*
- Both two paths {G3-G6} and {G4-G6} are sensitized
  - Error is propagated along both paths simultaneously
  - G2 = G5 = 1 , *c* = *e* = 1
  - Test generated successfully
- FFT: can we extend 1-path to 2, 3, 4, 5…path sensitization?



Complete ATPG Algorithm

- A *complete ATPG* exhausts the whole input space (2^n)
  - If a test pattern exists, complete ATPG will find it for sure
- An *incomplete ATPG* does NOT exhaust the whole input space
  - ATPG may fail even though a test pattern DOES exist



Complete ATPG Guarantees to Find Solution
if it exists

Quiz

Q: Is Single Path Sensitization a complete ATPG algorithm?

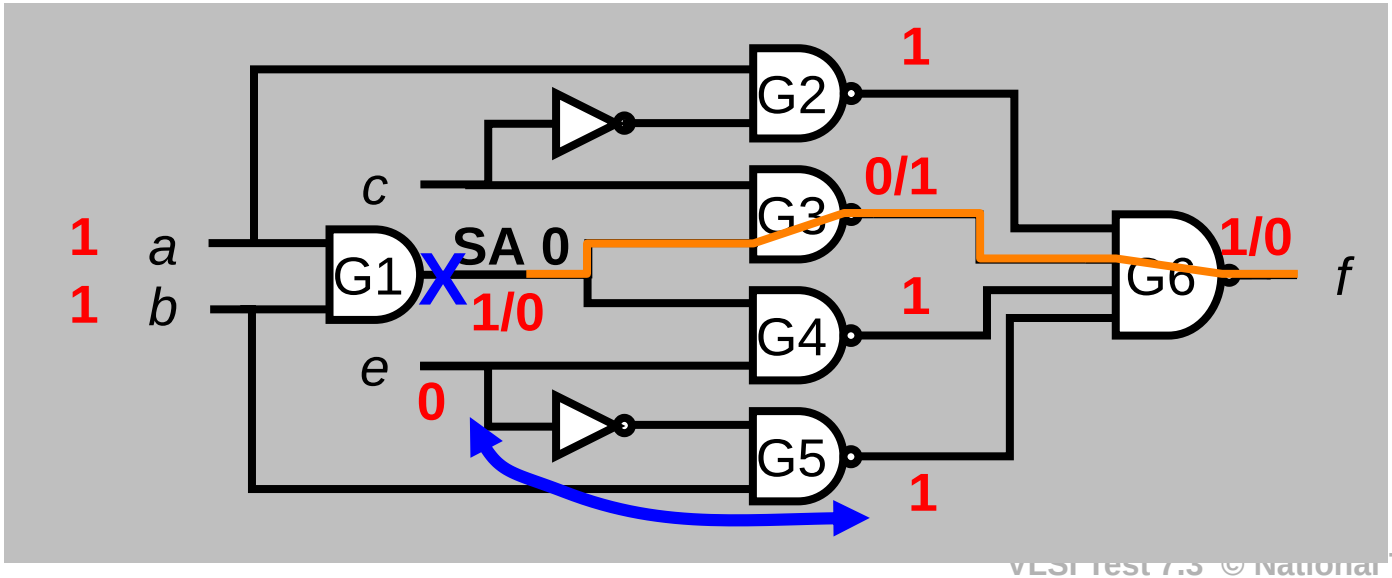
Hint: recall this example

- **Single path sensitization (SPS) Algorithm:**

① *Fault activation* (aka. *Fault excitation*)
  Assign internal signals to generate value at fault site
  Desired value opposite to the faulty value

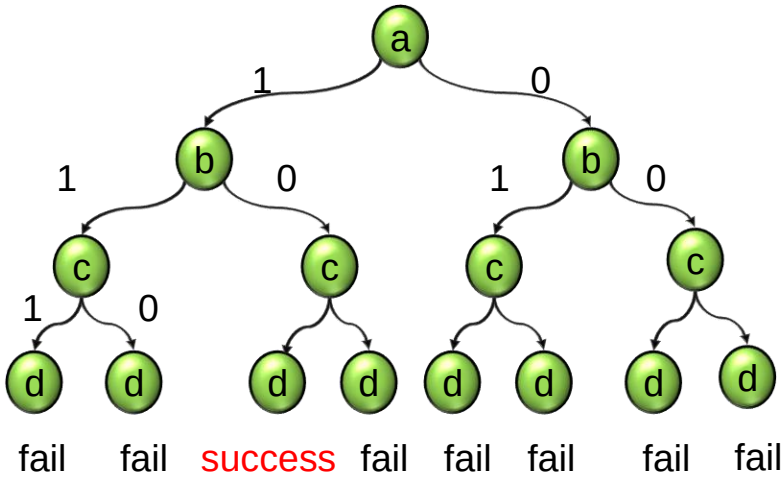
② *Fault effect propagation*:
  Select one single path from fault site to an output
  Assign internal signals to sensitize fault effect along the path

③ *Justification*:
  Assign primary inputs to justify signal values assigned in ①&②
  If justification fails, backtrack.



Summary

- **Single Path Sensitization Algorithm**
  - ① **Fault activation**
  - ② **Fault effect propagation**
  - ③ **Justification**
- **Complete ATPG: guarantee to find solution if one exists**
- **Disadvantages of SPS algorithm**
  - **Number of paths is exponential to circuit size. Too many!**
  - **Single path sensitization is incomplete ATPG**



FFT

- Q1: If SPS fails, can we extend 1-path to 2, 3, 4, 5… paths?
  - What is the problem for multiple path sensitization algorithm?
- Q2: Pros and Cons of complete ATPG algorithm?
  - in terms of run time and fault coverage