

CSE-308: Digital System Design

Lecture 9

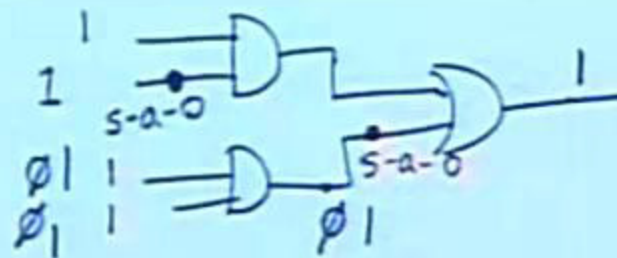
Test Generation

Basic Idea

O C E T
I . I . T . K G P

- Given the description of a circuit and a fault set under an assumed fault model, determine
 - the set of test vectors required to detect faults in the set.
 - the undetected faults.

- What is required actually?
 - Set a reverse logic value at the site of the fault.
 - Justify primary inputs accordingly.
 - Set / modify primary input values so that any change in logic value at the site of the fault may propagate to the primary output.



Test Generation Algorithms

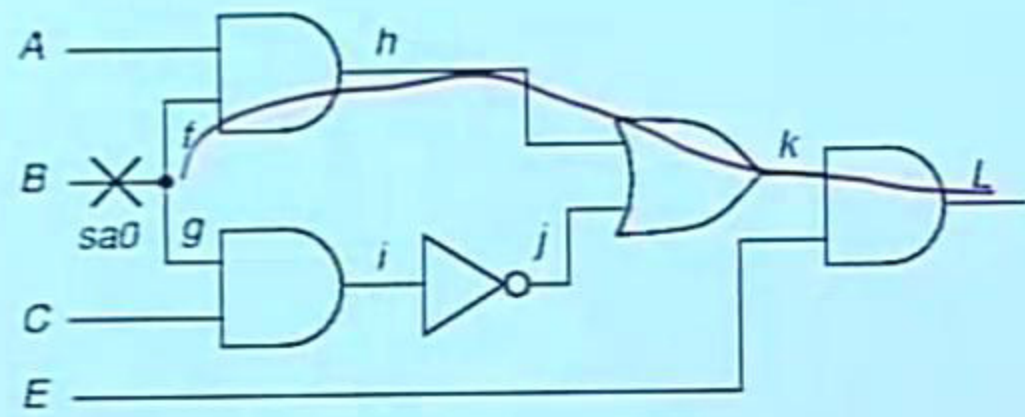
CCET
I.I.T. KGP

- Many algorithms exist:
 - Boolean difference method
 - D-algorithm
 - PODEM
 - FAN
 - etc etc etc

Path Sensitization

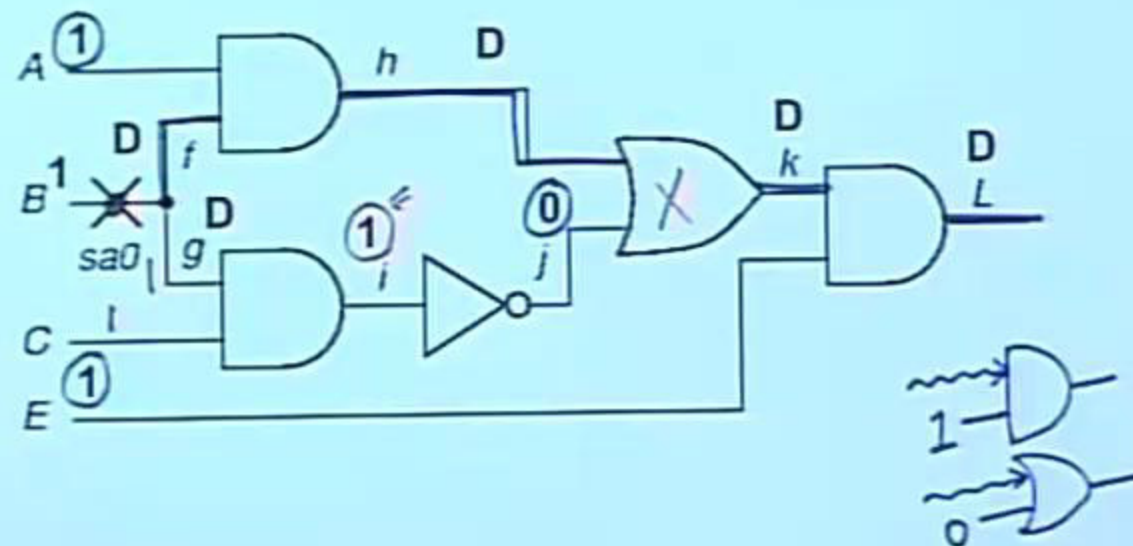
© CET
I.I.T. KGP

- Three steps:
 - Fault sensitization
 - Fault propagation
 - Line justification

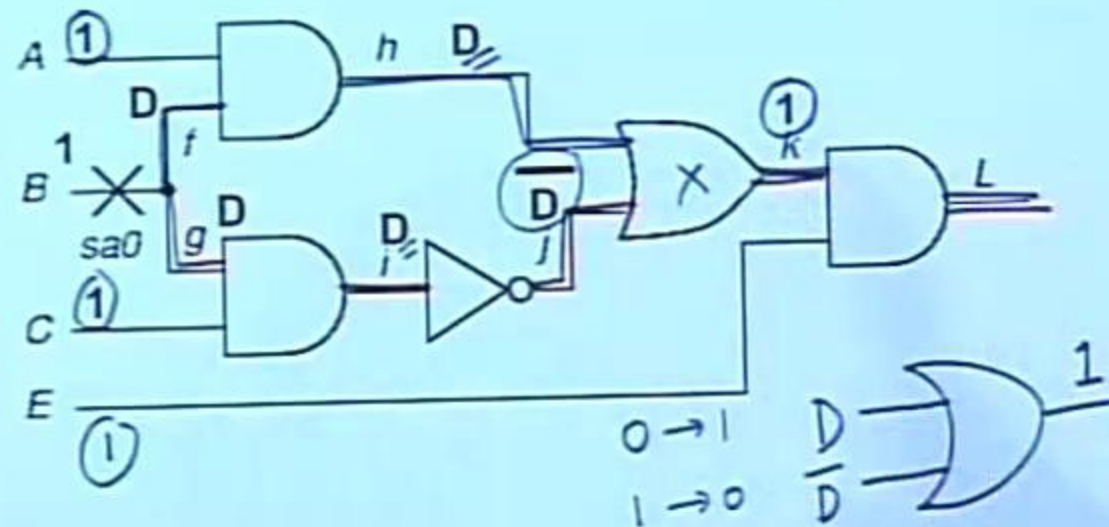


$D :: a \text{ change } (0 \rightarrow 1)$
 $\bar{D} :: 1 \rightarrow 0$

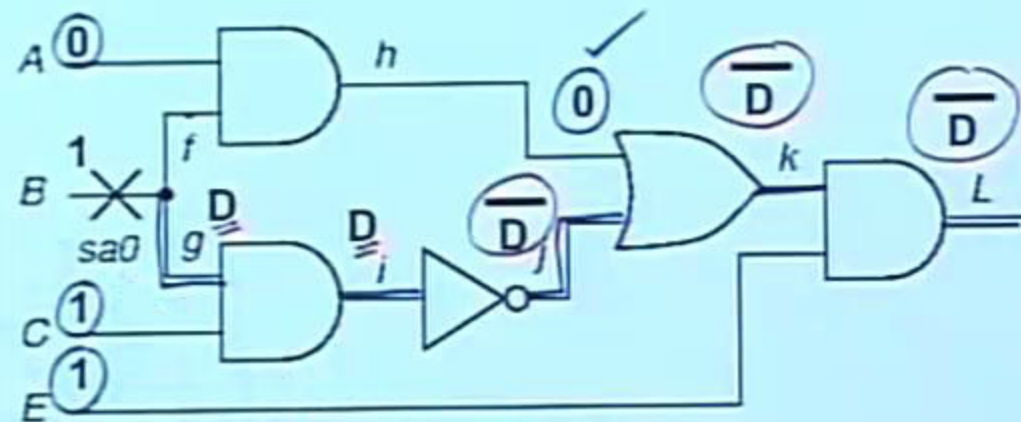
- Try path $f-h-k-L$. This path is blocked at j , since there is no way to justify the 1 on i



- Try simultaneous paths $f-h-k-L$ and $g-i-j-k-L$. These paths blocked at k because D -frontier (chain of D or \overline{D}) disappears



- Final try: path $g-i-j-k-L$ - test found!



ATPG - Algorithmic



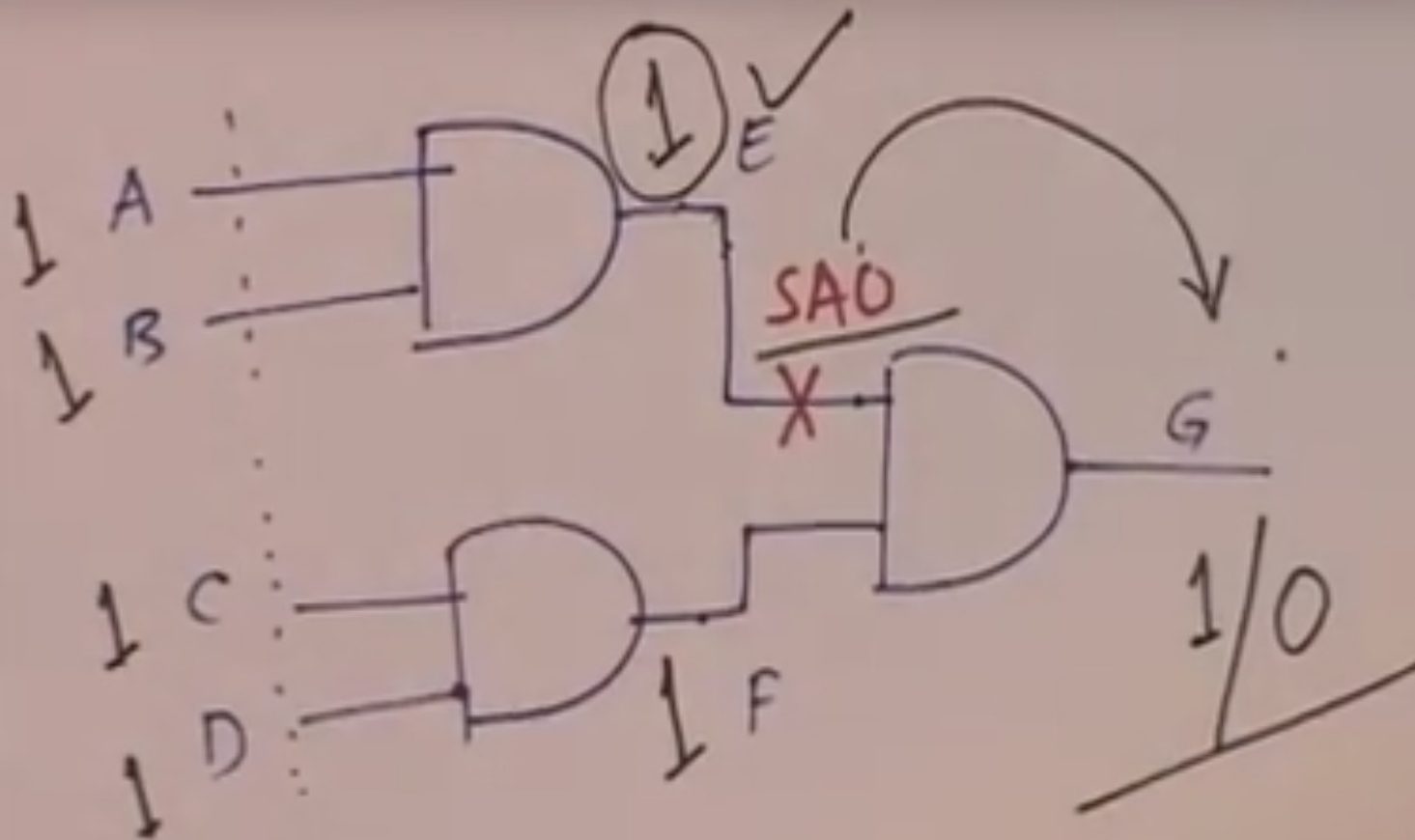
❖ Path Sensitization Method

- Fault Sensitization
- Fault Propagation
- Line Justification

❖ Path Sensitization Algorithms

- **D-Algorithm** (Roth)
- PODEM (P. Goel)
- FAN (Fujiwara)
- SOCRATES (Schultz)
- SPIRIT (Emil & Fujiwara)





Path Sensitization

General Structure of TG Algorithm

begin

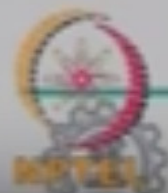
 set all values to x

Justify (l, v)

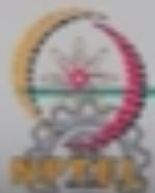
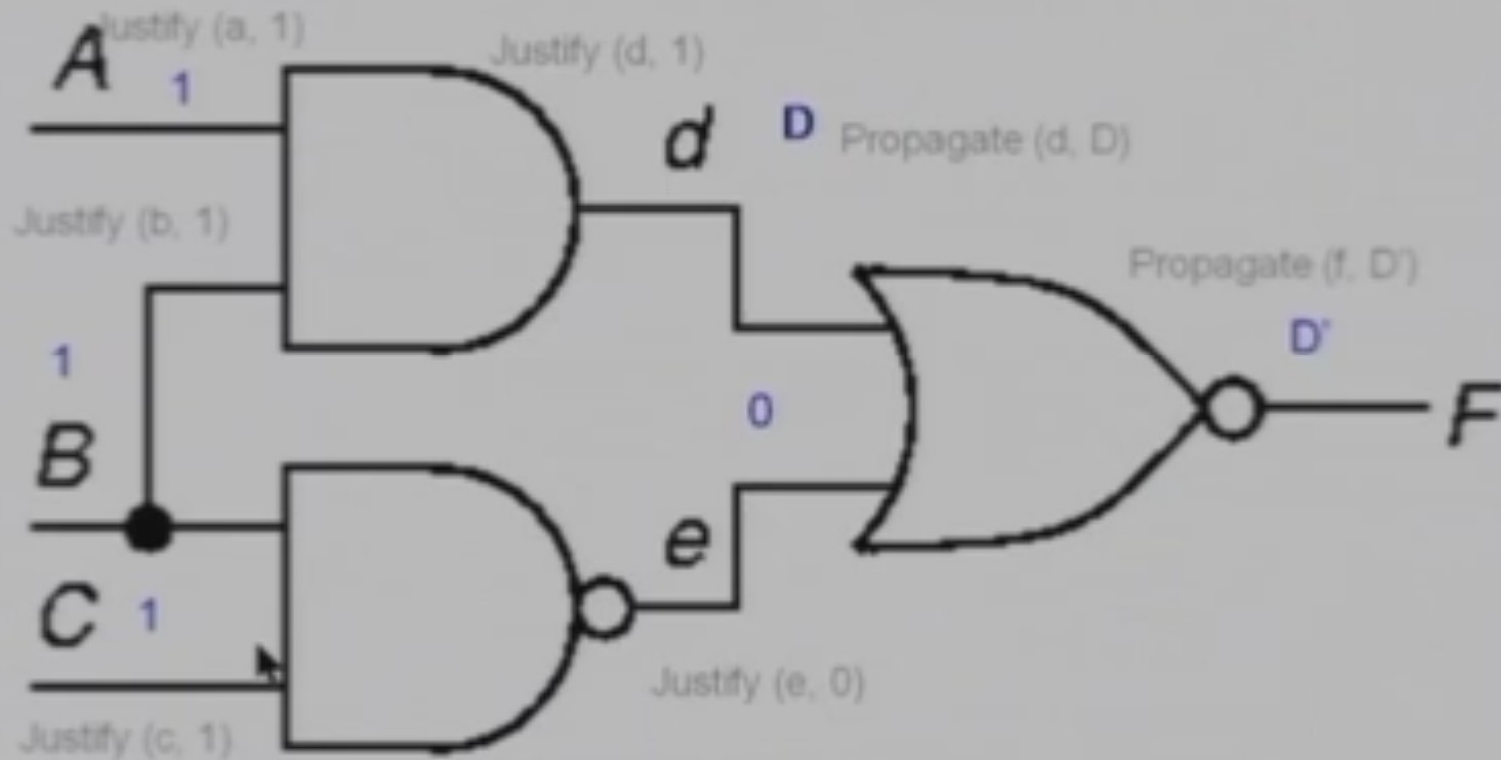
 if ($v = 0$) then *Propagate* (l, D)

 else *Propagate* (l, D')

end



Path Sensitization



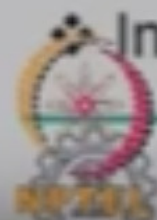
TG: Common Concept

- ❖ Fault Activation problem \rightarrow a LJ Problem
- ❖ The Fault Propagation problem \rightarrow
 1. Select a FP path to PO \rightarrow Decision
 2. Once the path is selected \rightarrow a set of LJ problems
- ❖ The LJ Problems \rightarrow Decisions or Implications



To justify $c = 1 \rightarrow a = 1, b = 1$ (Implication)

To justify $c = 0 \rightarrow a = 0$ or $b = 0$ (Decision)



Incorrect decision \rightarrow Backtrack \rightarrow Another decision

D-Algorithm

Roth (IBM) - 1966

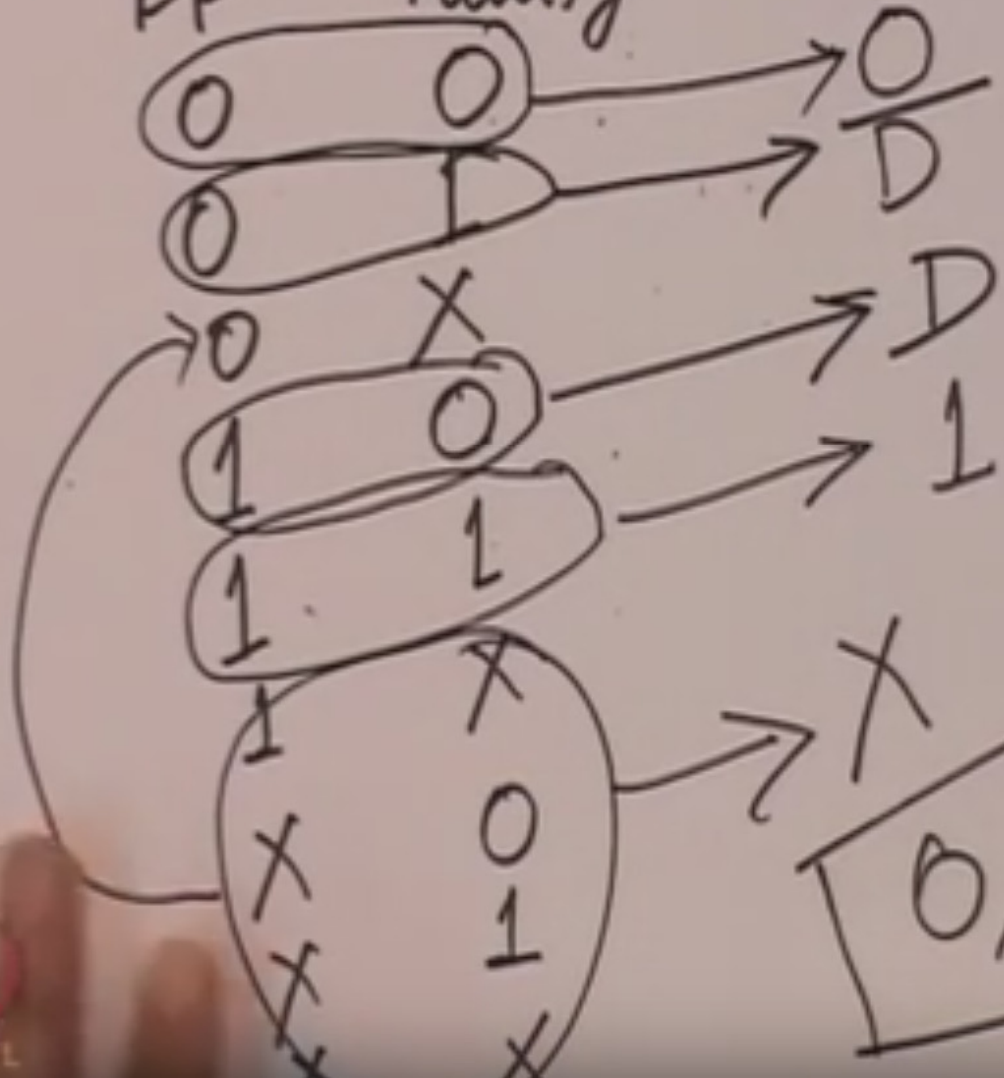
- Fundamental concepts invented:
 - First complete ATPG algorithm
 - **D-Calculus** (5 valued logic) – Composite value
 - **0** (0/0)
 - **1** (1/1)
 - **D** (1/0)
 - **D'** (0/1)
 - **X** (x/0, x/1, 1/x, 0/x, x/x)
 - Implications – forward and backward
 - Implication stack
 - **Backtrack**
 - Test Search Space



FF

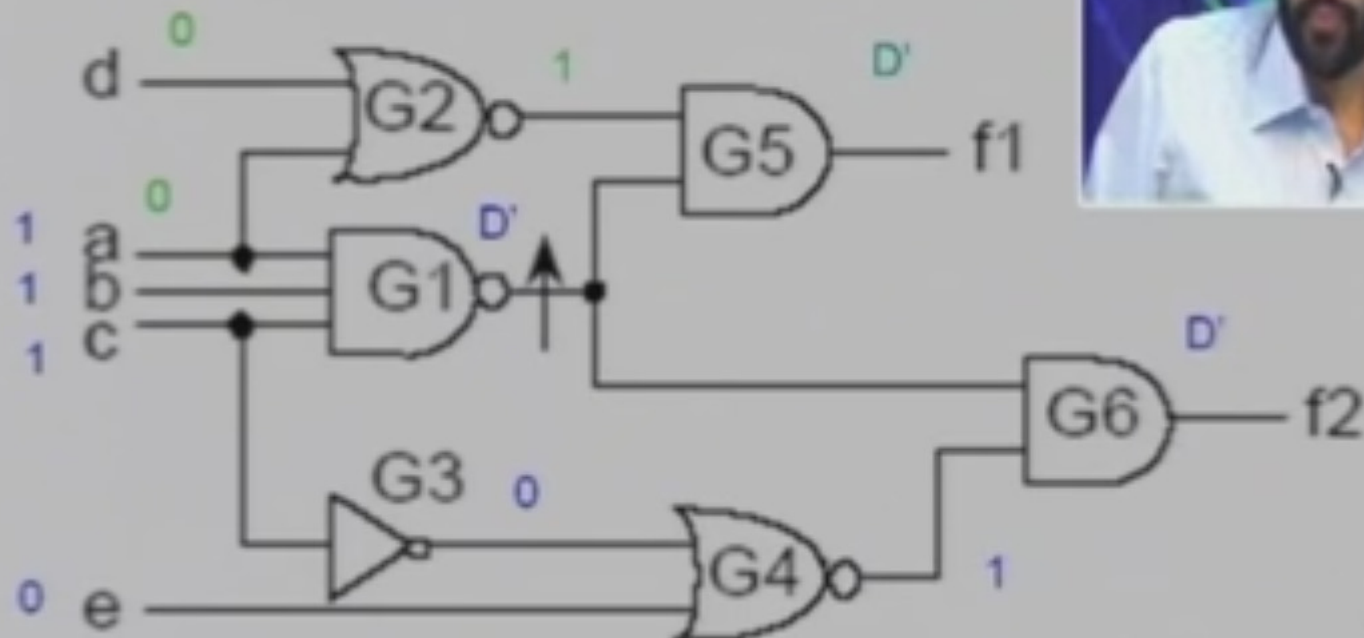
Faulty

0, 1, X

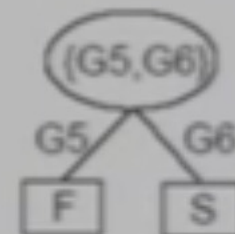


0, 1, D, \bar{D} , X

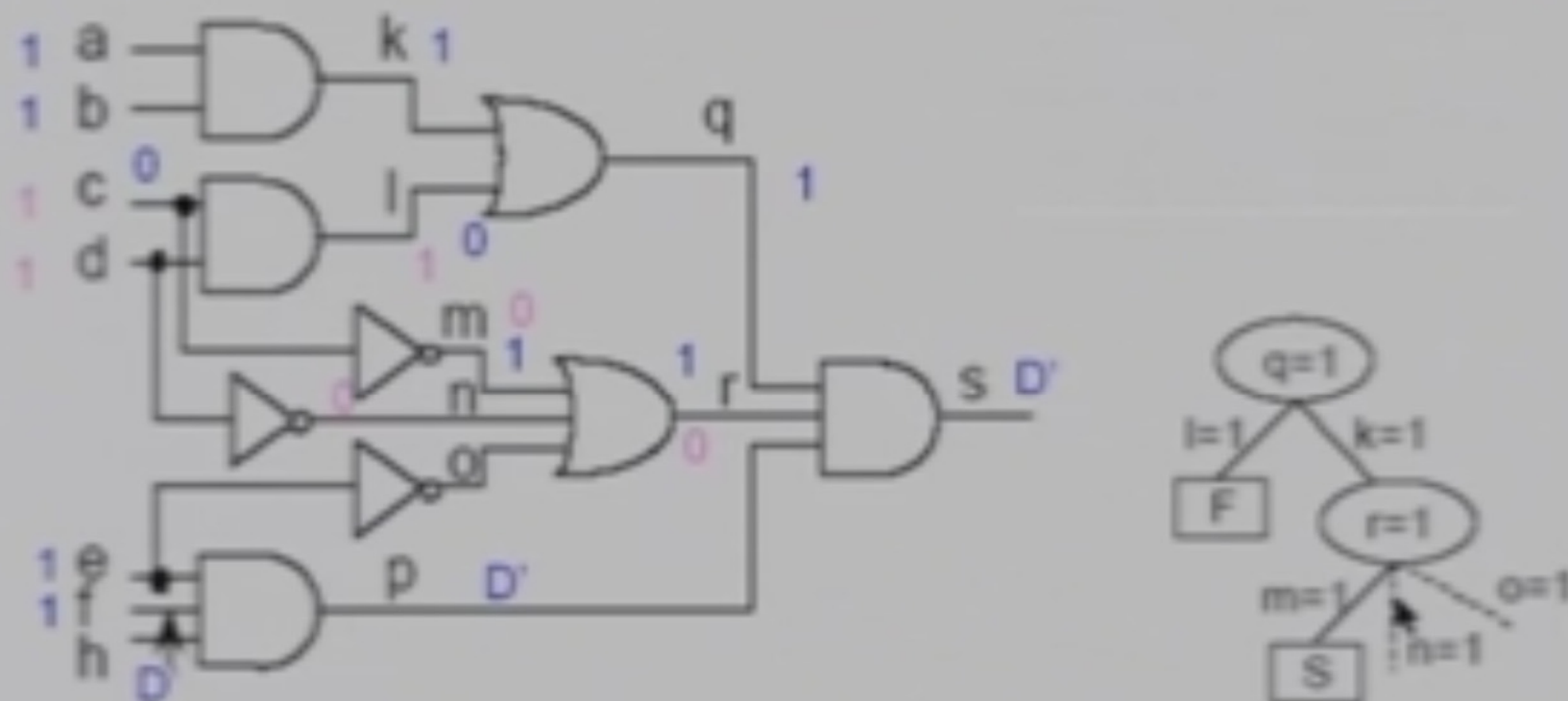
Decisions during FP



D – frontier: The set of all gates whose output value is currently x but have one or more **fault** signals on their inputs



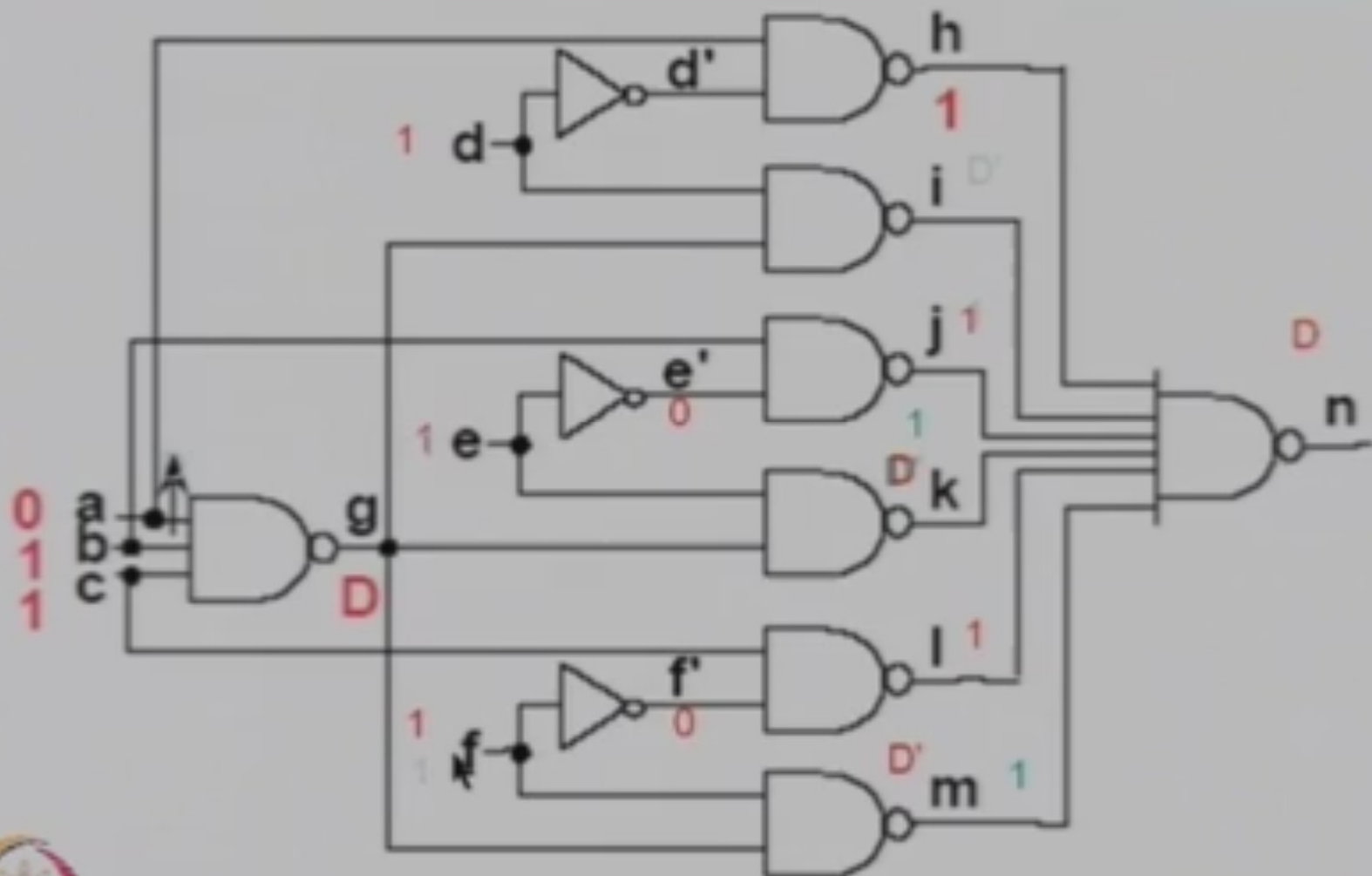
Decisions during LJ



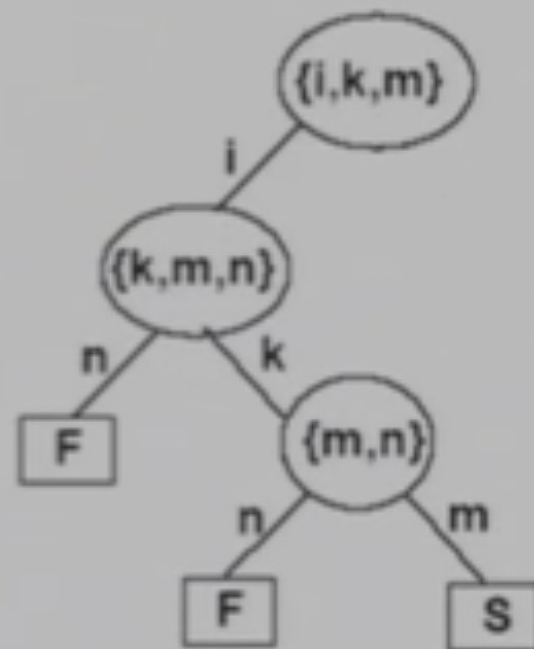
J – Frontier : A set of all gates whose output value is known but not implied by its input value



D-Algorithm : Example



Decision Tree



Two times of backtracking!!

