Testability of VLSI

Lecture 07:
Automatic Test Pattern Generation for Combinational Circuits

By Dr. Sanjay Vidhyadharan

ATPG Algorithm

Roth's **D-Algorithm** (D-ALG) , established the calculus and algorithms for ATPG using D-cubes.

The next development was Goel's **PODEM** algorithm. He efficiently used path propagation constraints to limit the ATPG algorithm search space, and introduced the notion of *backtrace.*

The third significant development was Fujiwara and Shimono's **FAN** algorithm . They efficiently constrained the backtrace to speed up search, and took advantage of signal information to limit the search space.

Prime Implicants

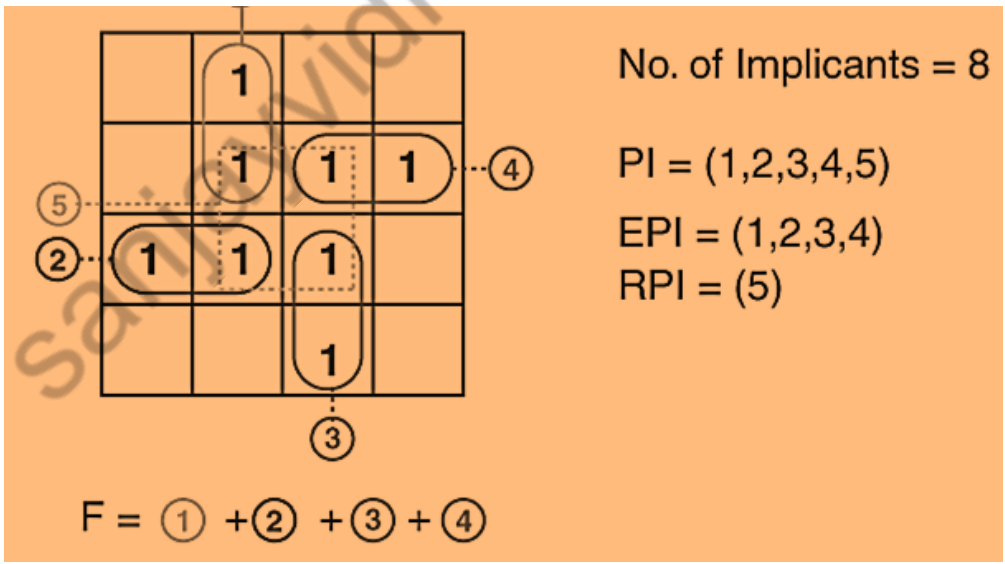
F = AB + ABC + A'C.

		BC			
		00	01	11	10
A	0	0	1	1	0
	1	0	0	1	1

Prime Implicants = AB + BC + A'C.

		BC			
		00	01	11	10
A	0	0	1	1	0
	1	0	0	1	1

Essential Prime Implicants = AB + A'C.



D-Calculus and D-Algorithm (Roth)

Definition 1. The ***singular cover*** of a logic gate is the minimal set of input signal assignments needed to represent *essential prime implicants* in the Karnaugh map of that logic gate, for both output cases of 0 and 1.

AND

		B	
		0	1
A	0	0	0
	1	0	1

NOR

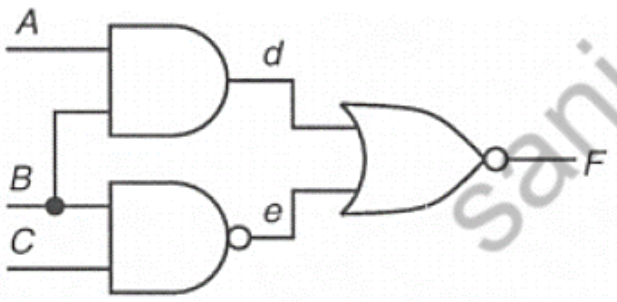
		B	
		0	1
A	0	1	0
	1	0	0

NAND

		B	
		0	1
A	0	1	1
	1	1	0

NAND

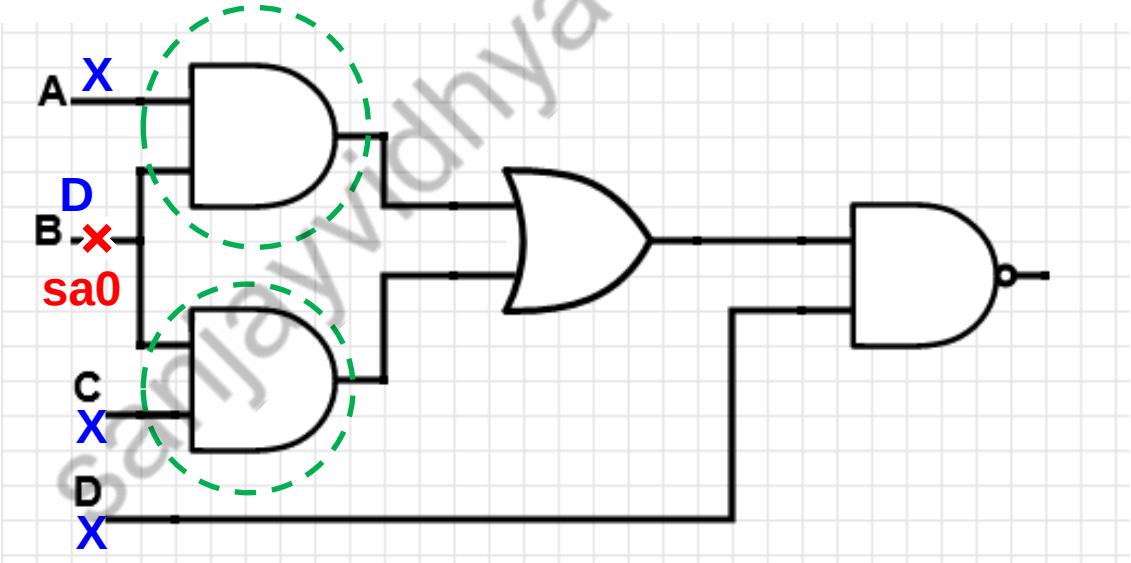
b	c	e
1	1	0
0	X	1
X	0	1



Gate type	Inputs		Output	Gate type	Inputs		Output
AND	*a*	*b*	*d*	NOR	*d*	*e*	*F*
1	0	X	0	4	1	X	0
2	X	0	0	5	X	1	0
3	1	1	1	6	0	0	1

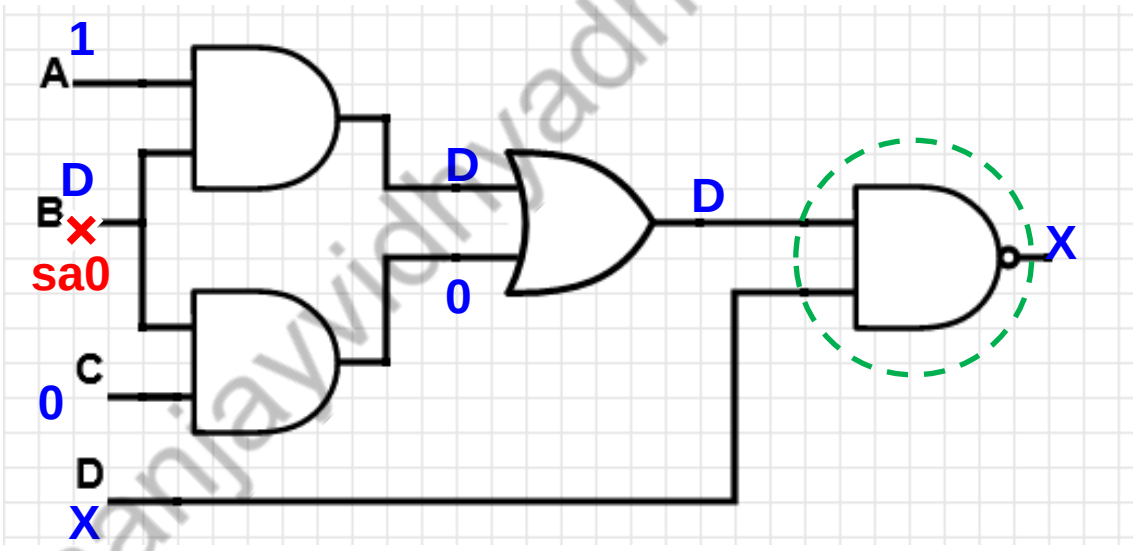
D-Calculus and D-Algorithm (Roth)

Definition 2. The ***D-frontier*** consists of all gates whose output value is currently x but have one or more error signals (either D's or D's) on their inputs. Error propagation consists of selecting one gate from the D-frontier and assigning values to the unspecified gate inputs so that the gate output becomes D or D. This procedure is also referred to as the D-drive operation. If the D-frontier becomes empty during the execution of the algorithm, then no error can be propagated to a PO. Thus an empty D-frontier shows that backtracking should occur.



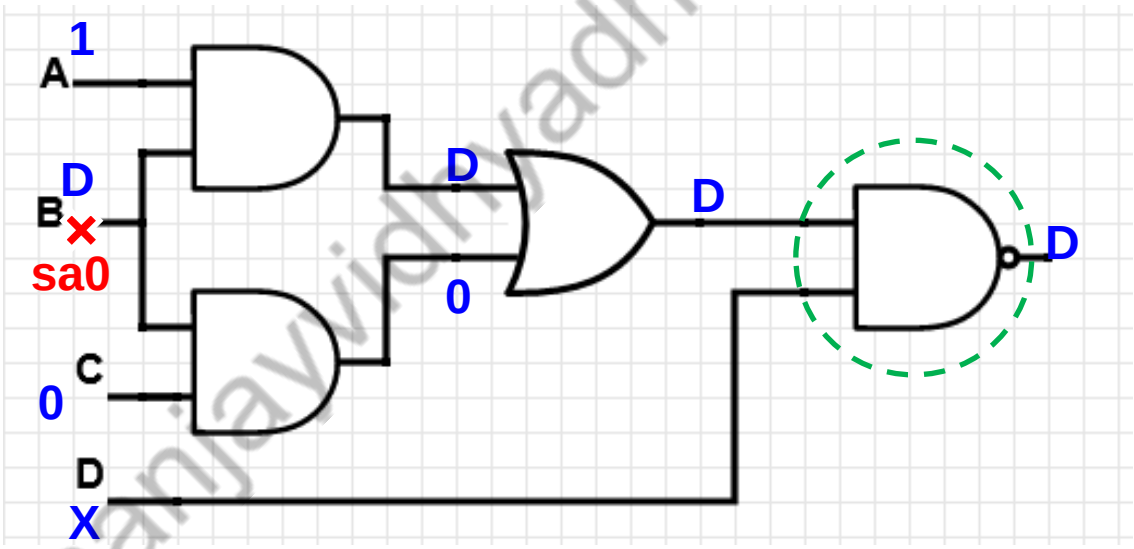
D-Calculus and D-Algorithm (Roth)

Definition 3. The ***Unique D-frontier.*** There is only one gate in the D-frontier and the fault needs to be propagated through it.



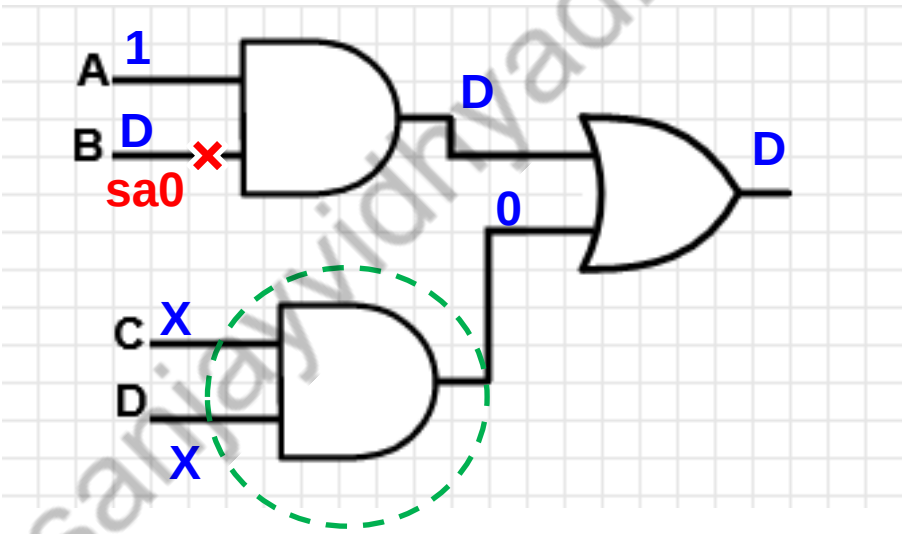
D-Calculus and D-Algorithm (Roth)

Definition 4. The ***J-frontier.*** To keep track of the currently unsolved line-justification problems, we use a set called the J-frontier, which consists of all gates whose output value is known but is not implied by its input values.



D-Calculus and D-Algorithm (Roth)

Definition 4. The ***J-frontier.*** To keep track of the currently unsolved line-justification problems, we use a set called the J-frontier, which consists of all gates whose output value is known (requirement) but is not implied by its input values.



D-Calculus and D-Algorithm (Roth)

Definition 5. A ***Propagation D-cube*** is a collapsed truth table entry that can be used to characterize an arbitrary logic block..

AND gate *propagation* D-cube D,1,D or D', 1, D' or D,D,D or D',D',D'

OR gate *propagation* D-cube D,0,D or D',0, D' or D,D,D, or D',D',D'

NOR gate *propagation* D-cube D,0,D' or D',0, D

NAND gate *propagation* D-cube D,1,D' or D',1, D

D-Calculus and D-Algorithm (Roth)

Definition 6. ***Primitive D-cubes of failure (PDCF)*** model faults in a logic circuit, and can model any (1) stuck-at-0 fault, (2) stuck-at-1 fault, (3) bridging fault (short circuit), or (4) arbitrary change in logic gate function (e.g., from AND to OR.)

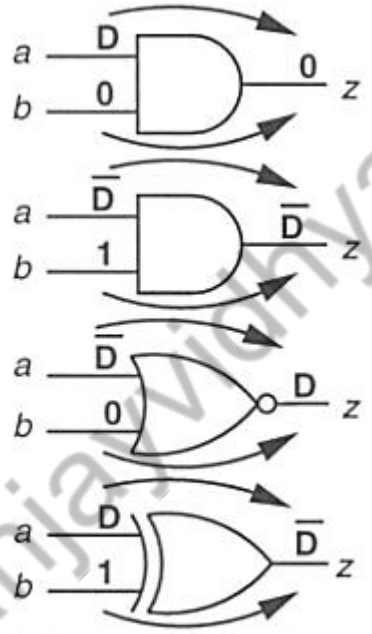
AND Sa0 PCDF 11D

OR Sa0 PCDF ? X1D or 1XD

OR Sa1 PCDF ? 00D'

D-Calculus and D-Algorithm (Roth)

Definition 7. ***Forward implication*** results when the inputs to a logic gate are significantly labeled so that the output can be uniquely determined. Gate is removed from ***D-frontier List***



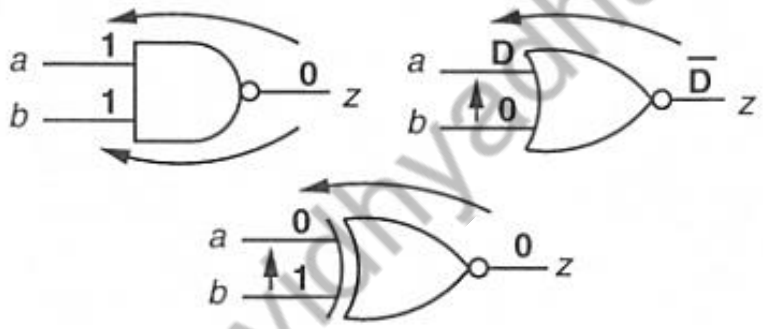
(a) Forward implications.

a \ b	0	1	X	D	D̄
0	0	0	0	0	0
1	0	1	X	D	D̄
X	0	X	X	X	X
D	0	D	X	D	0
D̄	0	D̄	X	0	D̄

(b) AND gate implication table.

D-Calculus and D-Algorithm (Roth)

Definition 8. ***Backward implication*** is the unique determination of all inputs of a gate for given output and possibly some of the inputs.. Gate is removed from ***J-frontier List***

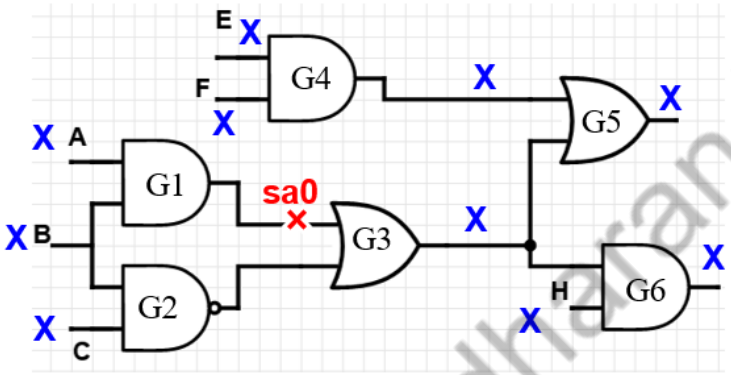


D-Calculus and D-Algorithm (Roth)

Procedure.

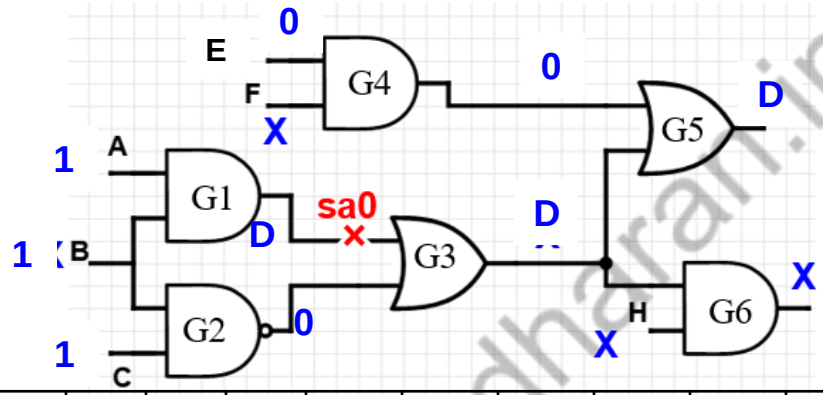
1. Pick a fault from the Fault table for a node
2. Select e PDCF for the fault.
3. D-Drive : Propagate the fault choosing from the D-frontier gates (Forward implication)
4. Back Propagate to get consistent inputs. If inconsistently encountered back track and chose alternate path.

D-Calculus and D-Algorithm (Roth)



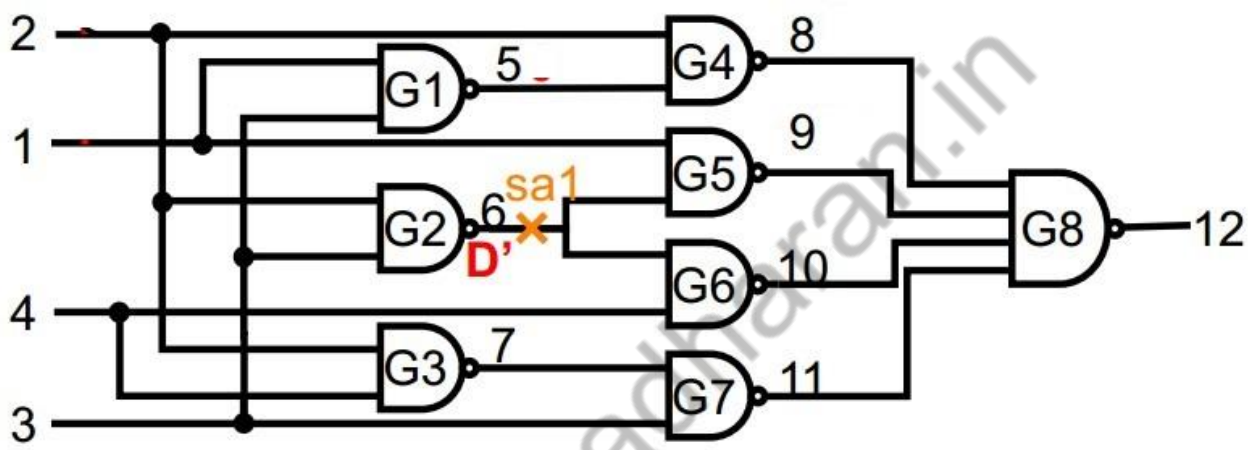
Step	A	B	C	E	F	H	G1	G2	G3	G4	G5	G6	
1. Choose a fault. Sa0 at G1.	1	1	X	X	X	X	D	X	X	X	X	X	PCDF G1 DF{G3}
2. Forward Implication	1	1	X	X	X	X	D	0	D	X	X	X	JF{G2} DF{G5. G6}
3. Forward Implication Choose G5	1	1	X	X	X	X	D	0	D	0	D	X	JF{G2, G4}
4. Backward Implication	1	1	X	0	X	X	D	0	D	0	D	X	JF{G2}
5.Backward	1	1	1	0	X	X	D	0	D	0	D	X	Done

D-Calculus and D-Algorithm (Roth)



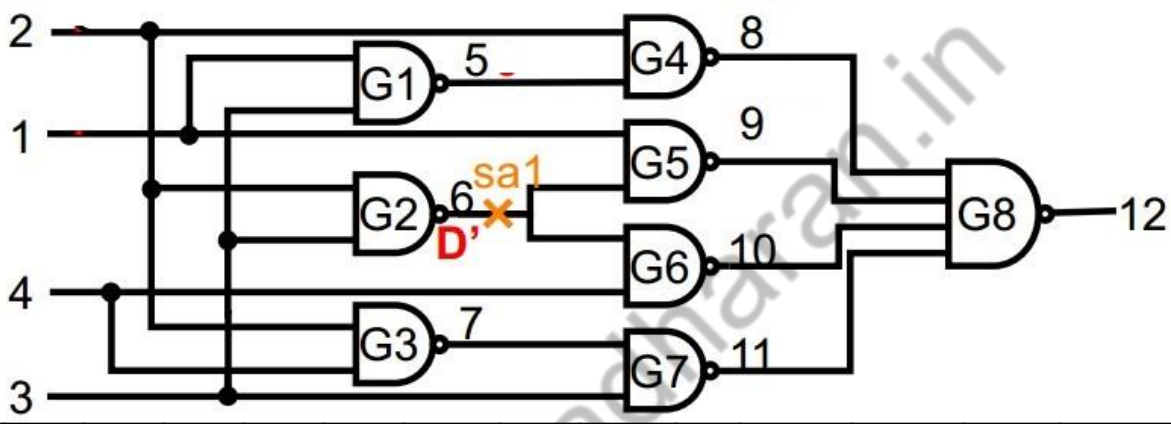
Step	A	B	C	E	F	H	G1	G2	G3	G4	G5	G6	
1. Choose a fault. Sa0 at G1.	1	1	X	X	X	X	D	X	X	X	X	X	PCDF G1 DF{G3}
2. Forward Implication	1	1	X	X	X	X	D	0	D	X	X	X	JF{G2} DF{G5. G6}
3. Forward Implication Choose G5	1	1	X	X	X	X	D	0	D	1	D	X	JF{G2, G4}
4. Backward Implication	1	1	X	0	X	X	D	0	D	0	D	X	JF{G2}
5.Backward	1	1	1	0	X	X	D	0	D	1	D	X	Done

D-Calculus and D-Algorithm (Roth)



Step	1	2	3	4	5	6	7	8	9	10	11	12	
1. Choose a fault. Sa0 at G2.	X	**1**	**1**	X	X	**D'**	X	X	X	X	X	X	PCDF G2 DF{G5,G6}
2. Choose G5. Forward Implication	**1**	1	1	X	X	D'	X	X	**D**	X	X	X	DF{G8}
3. Forward Implication.	1	1	**1**	**X**	X	D'	X	**1**	D	**1**	**1**	**D'**	JF{G4, G6,G7}
4.Backward Implication	1	1	**1**	**0**	**0**	D'	**0**	1	D	1	1	D'	JF{G1, G3}
5.Backward Implication	1	1	1	0 **1**	0	D'	0	1	D	D	1	X	Contention for G7

D-Calculus and D-Algorithm (Roth)



Step	1	2	3	4	5	6	7	8	9	10	11	12	
1. Choose a fault. Sa0 at G2.	X	**1**	**1**	X	X	**D'**	X	X	X	X	X	X	PCDF G2 DF{G5,G6}
6. Choose G5 & G6	**1**	1	1	1	X	D'	X	X	**D**	**D**	X	X	DF{G8}
7. Forward Implication.	1	1	1	**1**	X	D'	X	**1**	D	D	**1**	**D'**	JF{G4, G7}
8.Backward Implication	1	1	1	1	**0**	D'	**0**	1	D	D	1	D'	

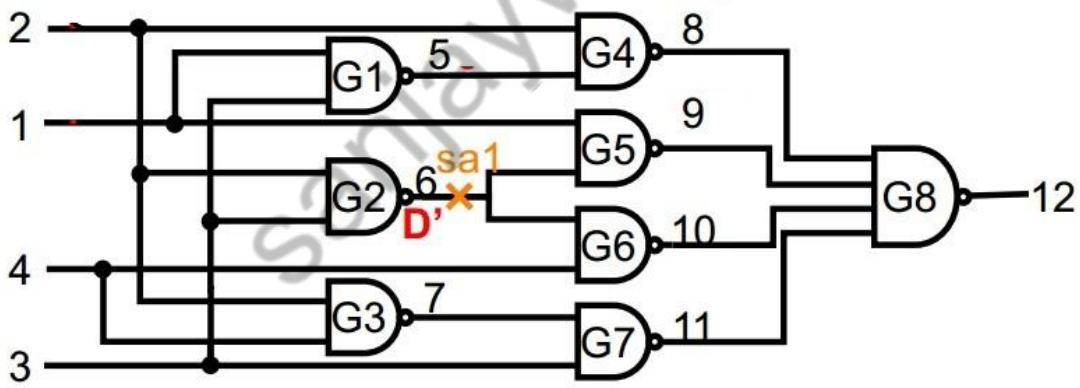
D-Calculus and D-Algorithm (Roth)

Advantage

1. D algorithm is *complete ATPG* Guarantee to generate a pattern for a testable fault

Disadvantage

1. Internal nodes are also assigned values hence the search space is large
2. Does not help in choosing best D-Frontier and relies on back tracking



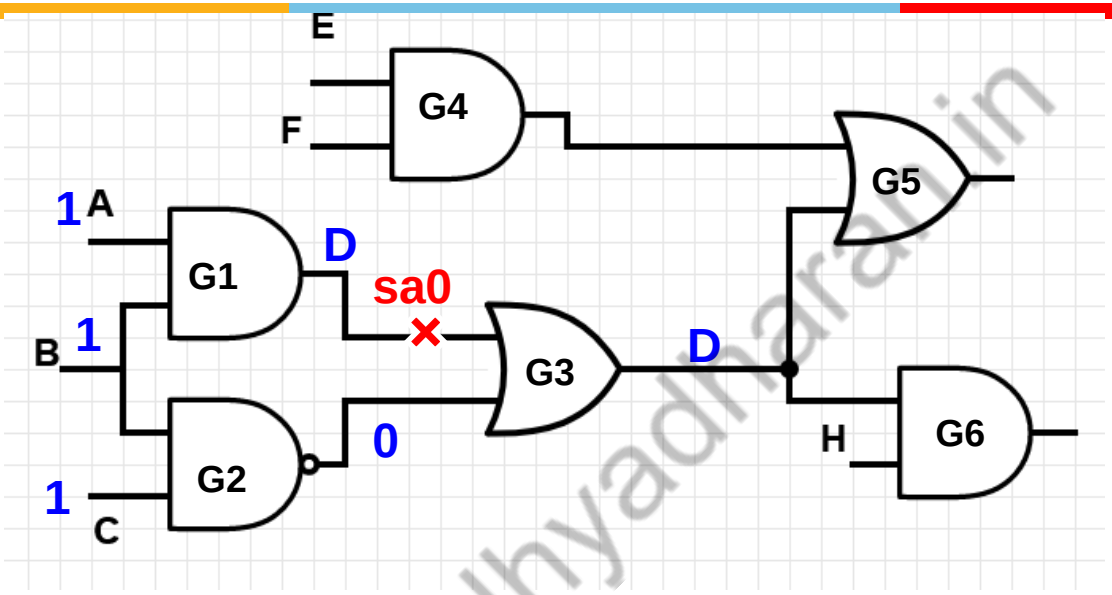
PODEM

1. **Only allow assignments to *PI only***
   Doesn't assign internal nodes
   Greatly reduces search tree

2. **Assigned PI are then forward implication**
   No justification needed

3. Flip last PI assignment when two conditions:
   A. Fault not activated
   B. No propagation path to any output

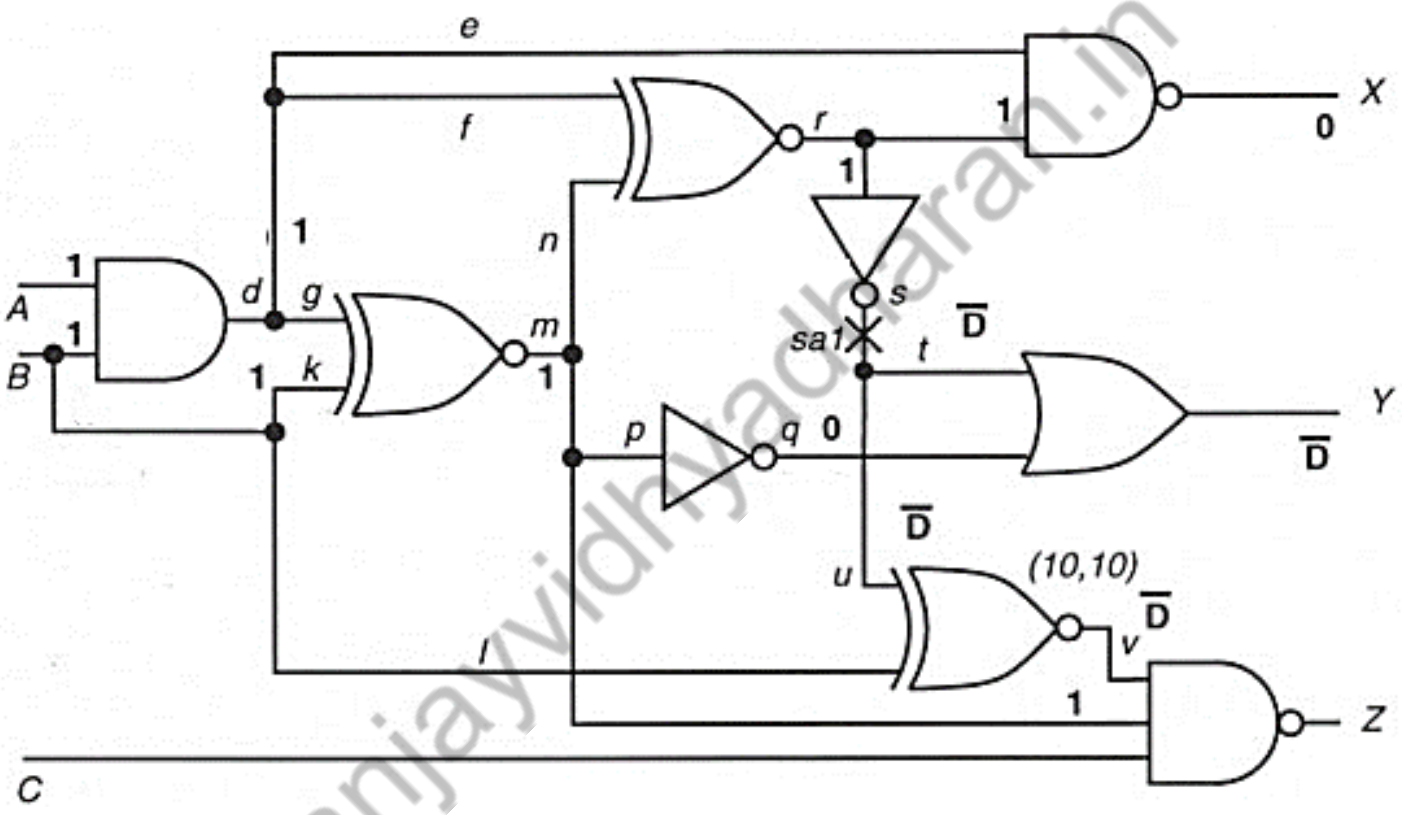
PODEM



1. Choose A=1 B=1 to det D at the desired fault location
2. Use G3 to Propagate. Single Option. C=1
3. Choose G6 easiest path to propagate . Level or SCOAP analysis will give the easiest path.
4.H=1. C=1 Depth First

Minimum number of logic gates between the start of the path and any PO. Objectives were selected by level to pick the *easiest* objective to achieve. After objectives were selected, backtracing determined PI assignments to justify these objectives.

PODEM

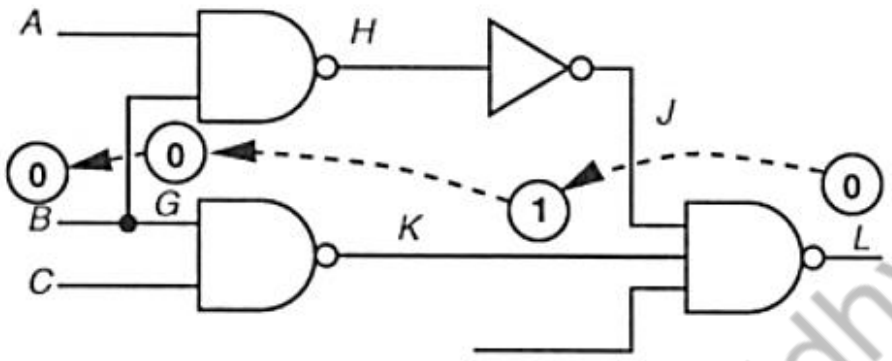


Test Vector is "11X" *and response* "0DX"

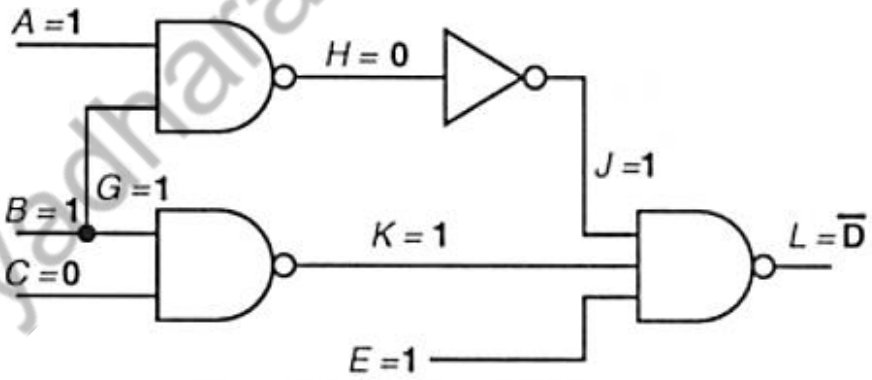
The basic idea of PODEM is to limit the search space to primary inputs without compromising the completeness. That is done by using the backtrace

FAN

Test Vector for sa1 at L



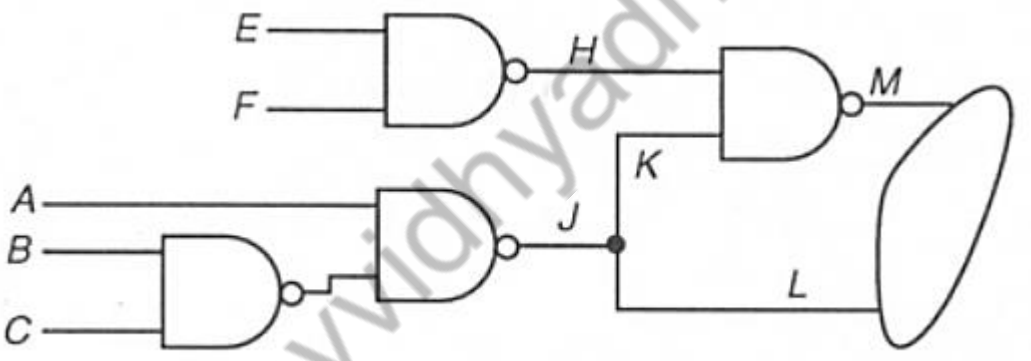
(a) Behavior of PODEM.



(b) Behavior of FAN.

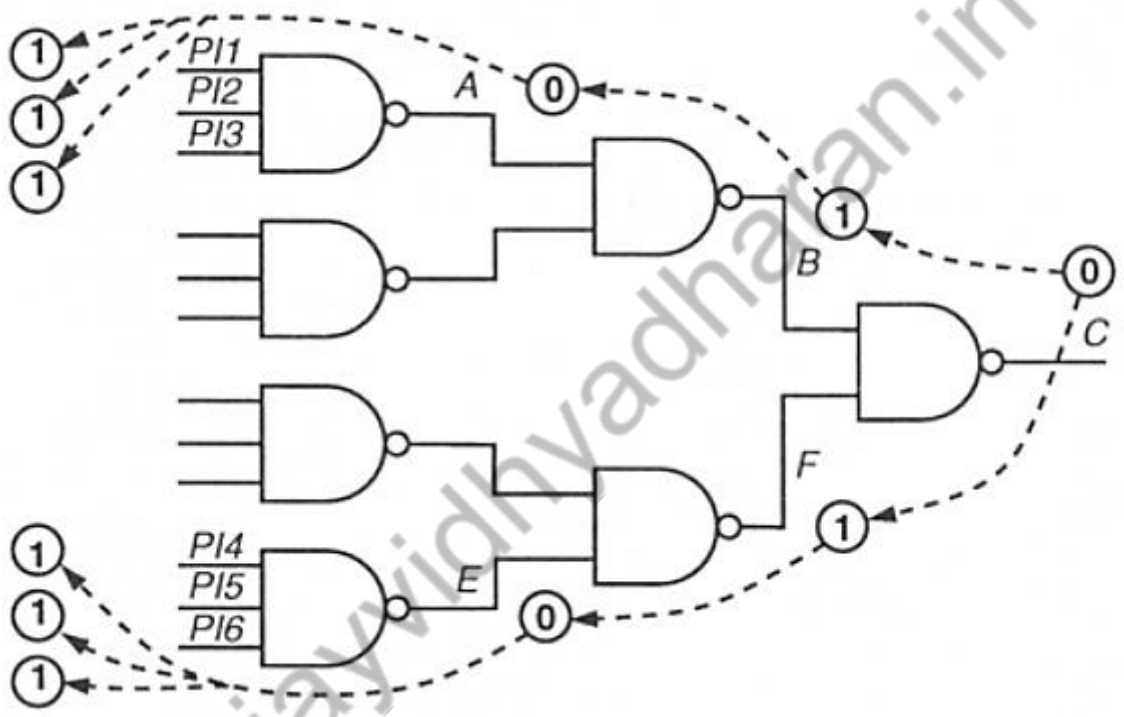
FAN

Headlines. Fujiwara and Shimono developed the notion of *headlines*, which are points where the circuit can be partitioned such that a cone of logic driven by PIs can be isolated from the rest of the circuit by cutting a single line, called the *headline.* This means that either a logic 0 or a logic 1 can be justified from the headline back to the circuit PIs.



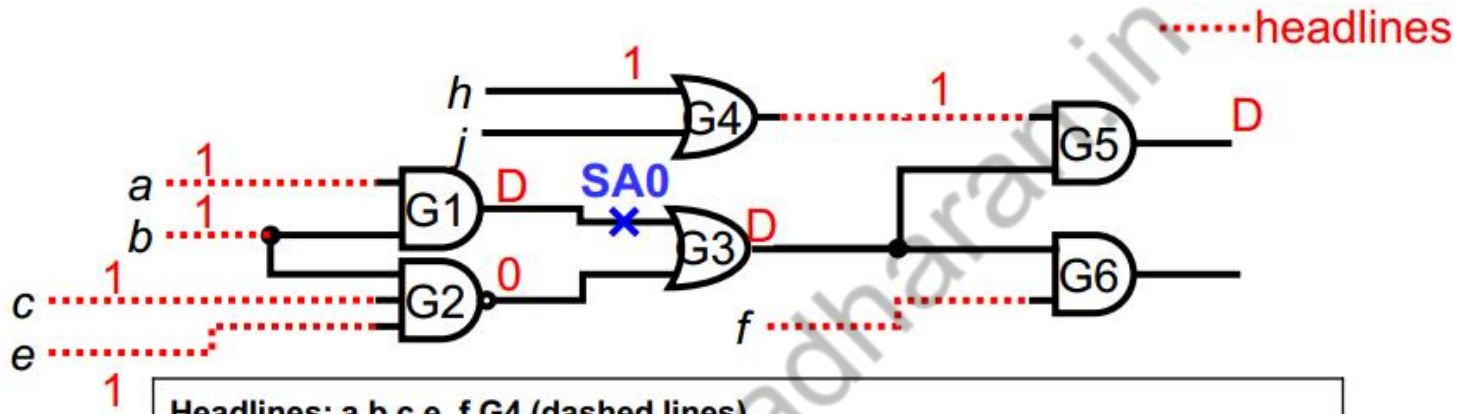
H and *J* are headlines

FAN



PODEM will make six backtraces to justify C=0. The first backtrace sets objectives of B=1 and A=0 and finally assigns PI1 as 1. This process is laboriously repeated five more times until we have PI2 =1, PI3 =1, PI4=1 PI5=1 and PI4=1 as internal node are not assigned any value. This is happening because PODEM backtraces in a depth-first fashion.

FAN



Headlines: a b c e f G4 (dashed lines)
Initial objective: G1 output =1
Implication: assign *a* = 1; *b*=1
Objective: propagate through G3, objective G2=0
Implication: assign *c* = 1; *e*=1
Propagate through G5, objective G4=1
Assign headline G4 = 1
G5=D, Objective achieved.
Justify head line G4 = 1 → h=1
Test generated abcehjf = 11111xx

Implication

Make decision at head line

Justify head lines at end

References

1. “Essentials of Electronic Testing, for Digital, Memory and Mixed-Signal VLSI Circuits”, Michael L. Bushnell and Vishwani D. Agrawal, – Kluwer Academic Publishers (2000).

2. Video lectures by Professor James Chien-Mo Li
Lab. of Dependable Systems Graduate Institute of Electronics Engineering
National Taiwan University
https://www.youtube.com/watch?v=yfcoKOUV5DM&list=PLvd8d-SyI7hjk_Ci0zpTqImAtpEjdK5JF&index=1

3. NPTEL Lectures
https://www.youtube.com/watch?v=M8VEEaYwlQ&list=PLbMVogVj5nJTClnafWQ9FK2nt3cGG8kCF&index=31

Thankyou