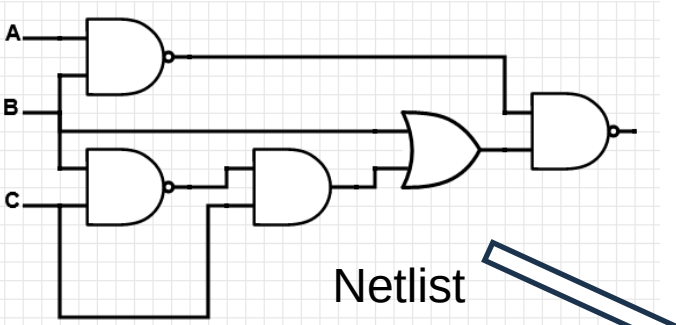
Testability of VLSI

Lecture 5: Fault Simulation

By Dr. Sanjay Vidhyadharan

Fault Simulation



Netlist

Collapsed fault list
A0, A1, B0…

Fault Simulator

Fault Statistics

Text Vectors

Automatic Test pattern generation

Fault Simulation

Steps Involved in Automatic Test pattern generation

1. Fault Sensitization : Driving a node with a stuck at fault with complementary signal by appropriately selecting the input vector.

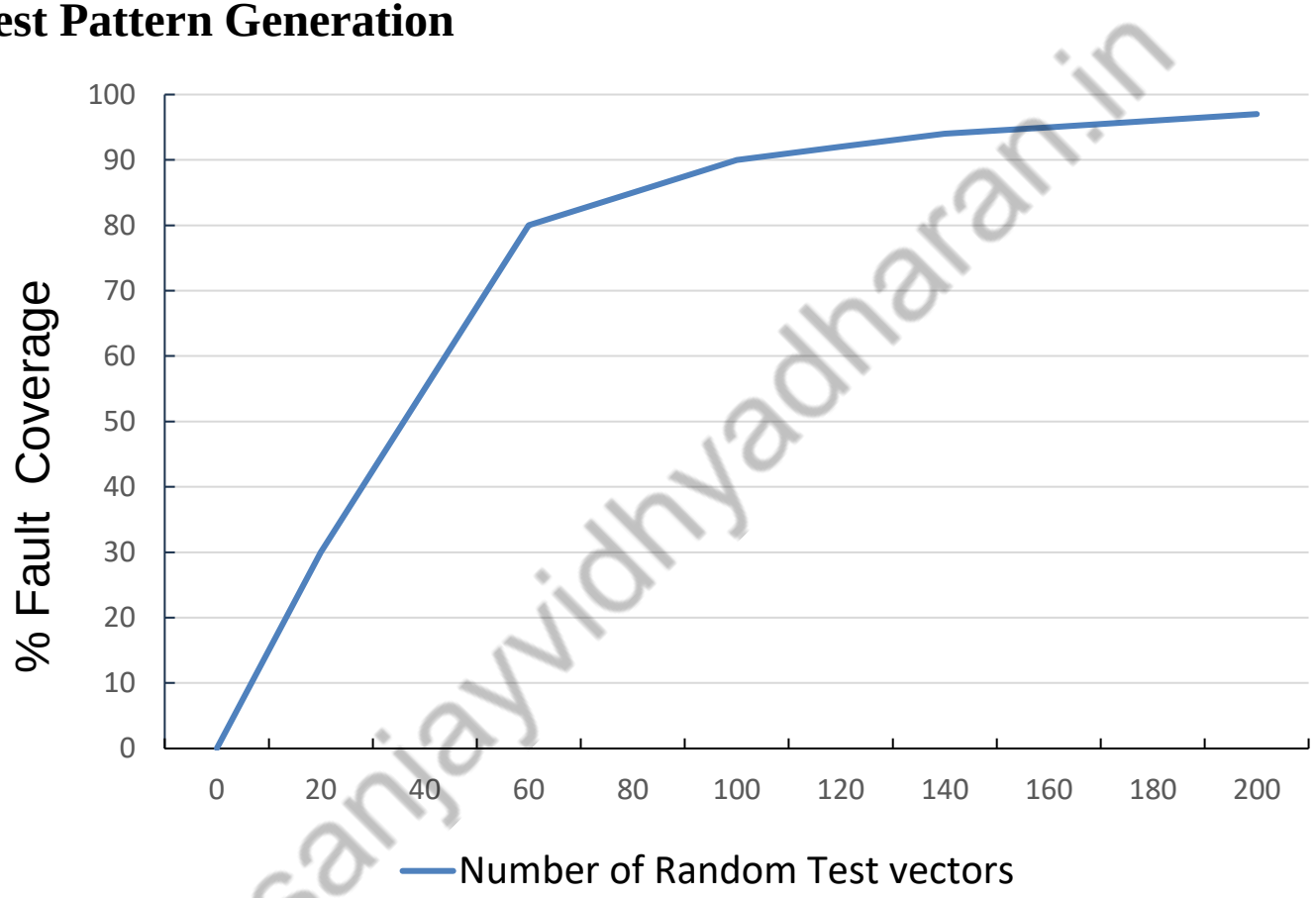
2. Fault Propagation : Affect of the fault needs to be propagated to one of the primary outputs.

3. Line Justification : Determination of the values at primary inputs so that fault sensitization and propagation are successful.

*SPJ is a lengthy process needs to be done for fault separately, but we can get **100%** fault coverage of **detectable faults.***

Fault Simulation

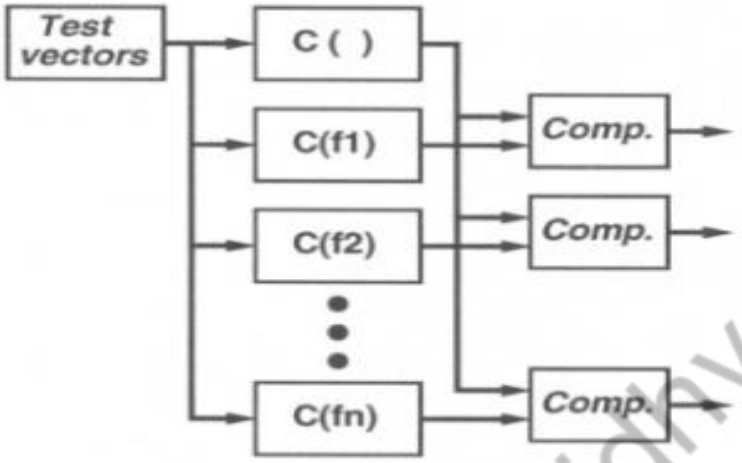
Random Test Pattern Generation



*RTP is a very fast process, detects multiple faults in each run, but we may not get **100%** fault coverage of **detectable faults.***

Algorithms for Fault Simulation

Serial Fault Simulation



(a) Fault simulation.

Fault	Detecting vector 1st	2nd	3rd
f1	5	12	19
f2			
fn	35	102	

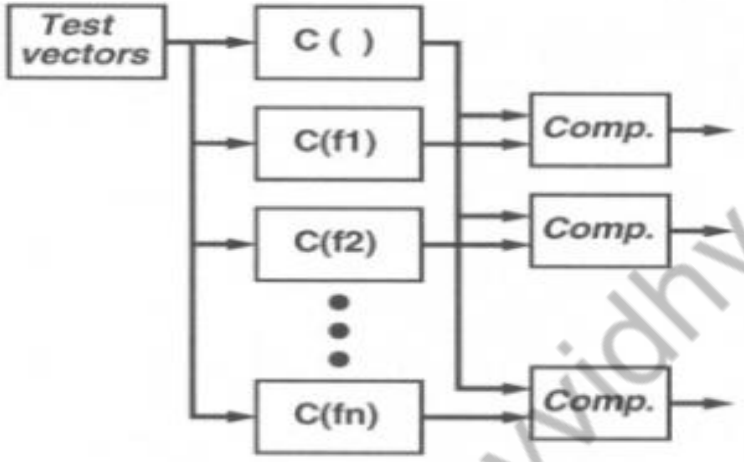
(b) Result.

The block C( ) is the fault-free circuit and blocks C(f1) through C(fn) are copies of the same circuit with faults f1 through fn . The same vectors are applied to all blocks and the outputs of the faulty circuits are compared in the comparators shown as *Comp.* *Event Driven can save time as one circuit to other not much change*

When fault fn is detected for the first time by vector 35, the simulation of block C(fn) is suspended beyond that vector. This procedure, known as ***fault dropping***, considerably speeds up the fault simulation process .

Algorithms for Fault Simulation

Serial Fault Simulation



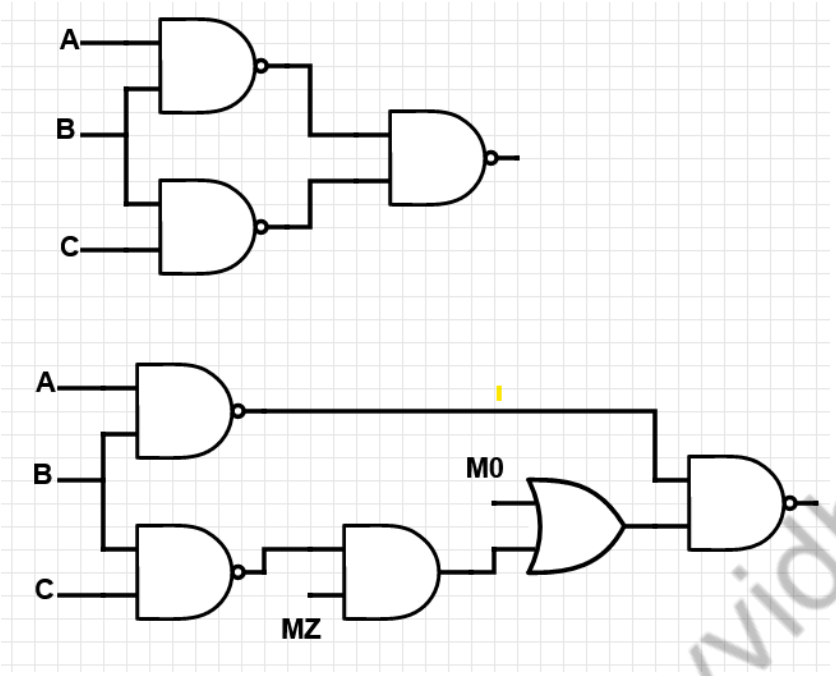
(a) Fault simulation.

Fault	Detecting vector 1st	2nd	3rd
f1	5	12	19
f2			
fn	35	102	

(b) Result.

Fault dropping considerably speeds up the fault simulation process . *Max time for Algorithm is M(n+1), M is max total test vectors of each block, n is number of faults. While testing only n vectors max required, Algorithm finds those n vectors. Could be less than n as multiple faults can be detected by a same vector.*

Inserting Faults

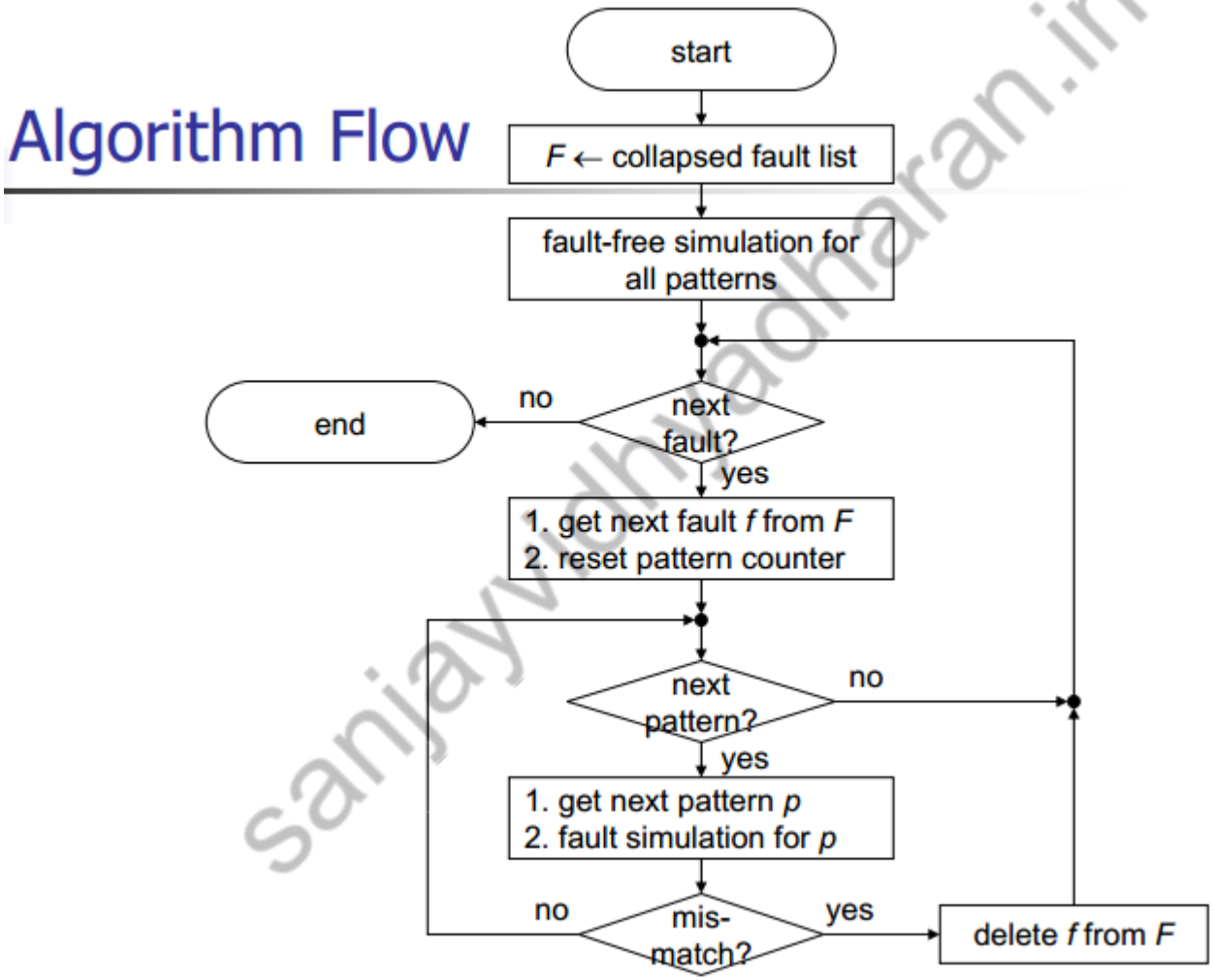


	M_Z	M_0
Fault Free	1	0
S-a-1	X	1
S-a-0	0	0

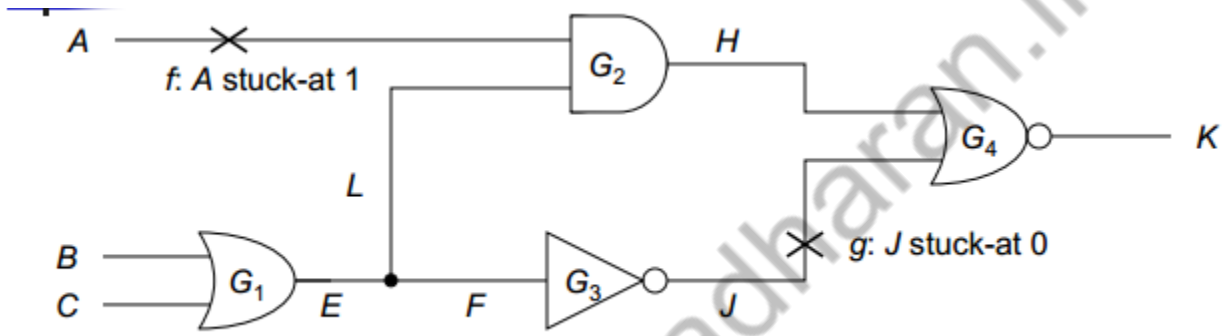
$$Z' = (Z \& M_Z) | M_0$$

Each faut site to be modeled as above and in test bench values of M_Z and M_0 are be set for different runs

Serial Fault Simulation



Serial Fault Simulation



Pat. #	Input			Internal					Output		
	A	*B*	*C*	*E*	*F*	*L*	*J*	*H*	K_{good}	K_f	K_g
P1	0	1	0	1	1	1	0	0	1	0	1
P2	0	0	1	1	1	1	0	0	1	0	1
P3	1	0	0	0	0	0	1	0	0	0	1

Serial Fault Simulation

Advantages:

Easy to implement

Ability to handle a wide range of fault models (stuck-at, delay, Br, …)

Very fast combinational simulation

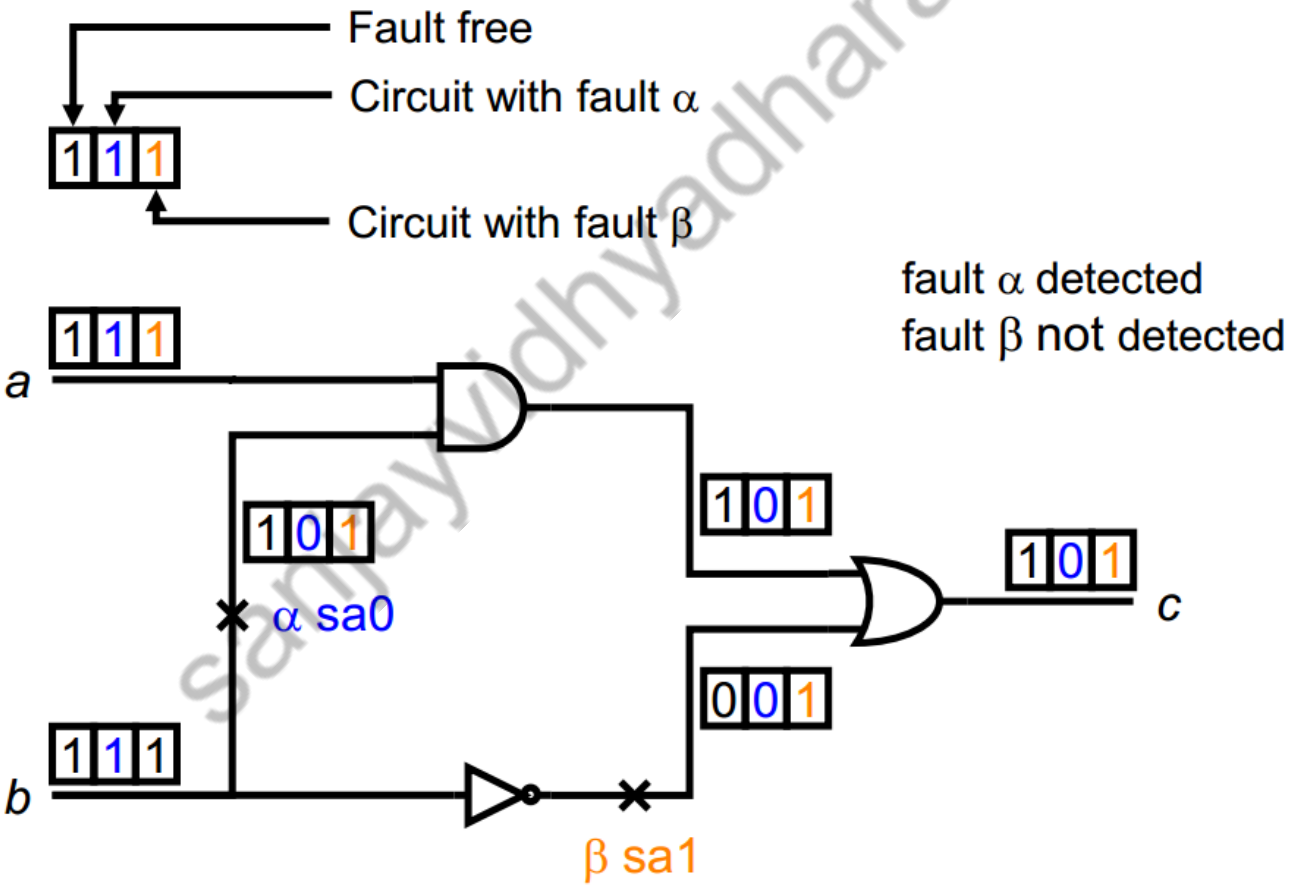
Disadvantages:

Many simulation runs required

CPU time prohibitive for VLSI circuits

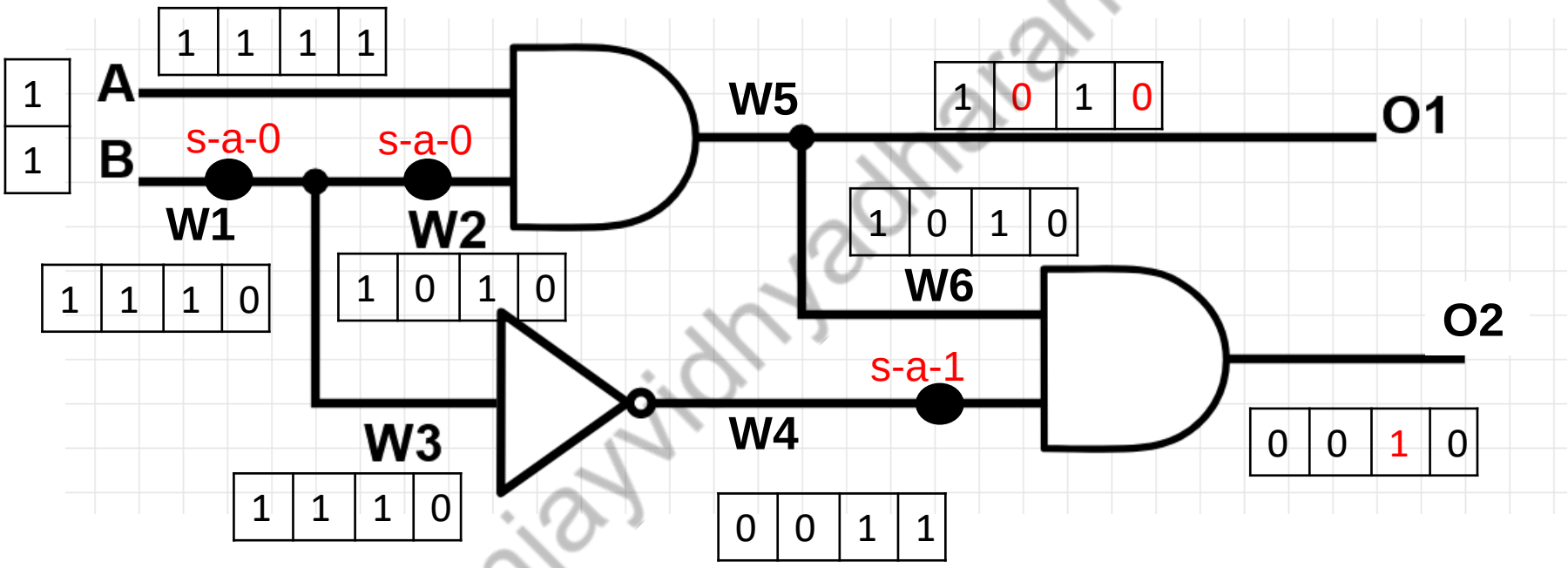
Parallel Fault Simulation

The idea of parallel fault simulation is to use the bit-parallelism of logical operations in a digital computer. For a 32-bit machine word, an integer consists of a 32-bit binary vector. A logical AND or OR operation involving two words performs simultaneous AND or OR operations on all respective pairs of bits.



Parallel Fault Simulation

N	W2 s-a-0	W4 s-a-1	W1 s-a-0



Max Number of Simulations Required

= Mn/(w-1) w is CPU word size,

We can assume the parallel process is faster than serial by approx. w times.

Parallel Fault Simulation

Advantages:

A large number of faults are detected by each pattern when simulating the beginning of test sequence.

Disadvantages:

Only applicable to the unit or zero delay models

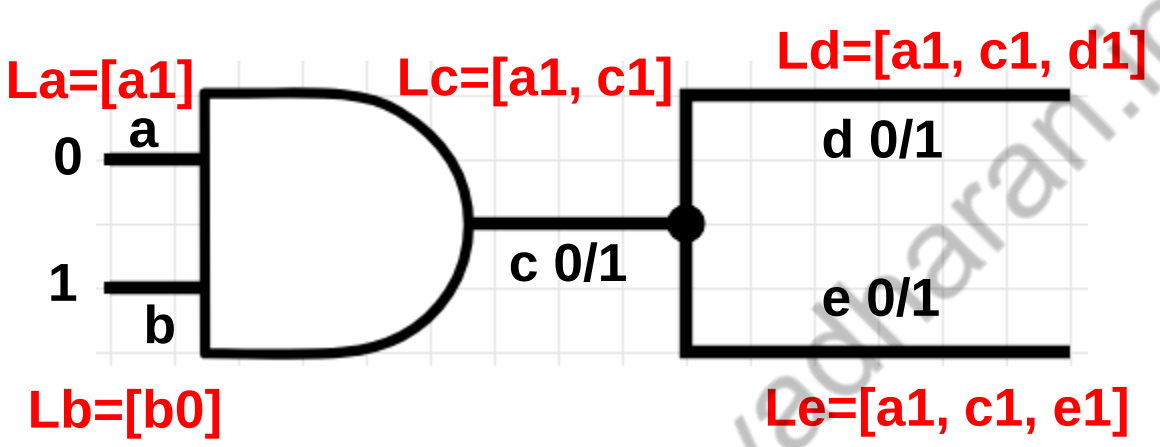
Faults cannot be dropped unless all (w-1) faults are detected

Deductive Fault Simulation

All signal values in each faulty circuit are deduced from the fault-free circuit values and the circuit structure. Since the circuit structure is the same for all faulty circuits, all deductions are carried out simultaneously. Thus, a deductive fault simulator processes all faults in a single pass of true-value simulation augmented with the deductive procedures. This gives the deductive simulators a tremendous speed, but only when the modeling conditions can be satisfied.

https://www.youtube.com/watch?v=zTLI2i69tKQ

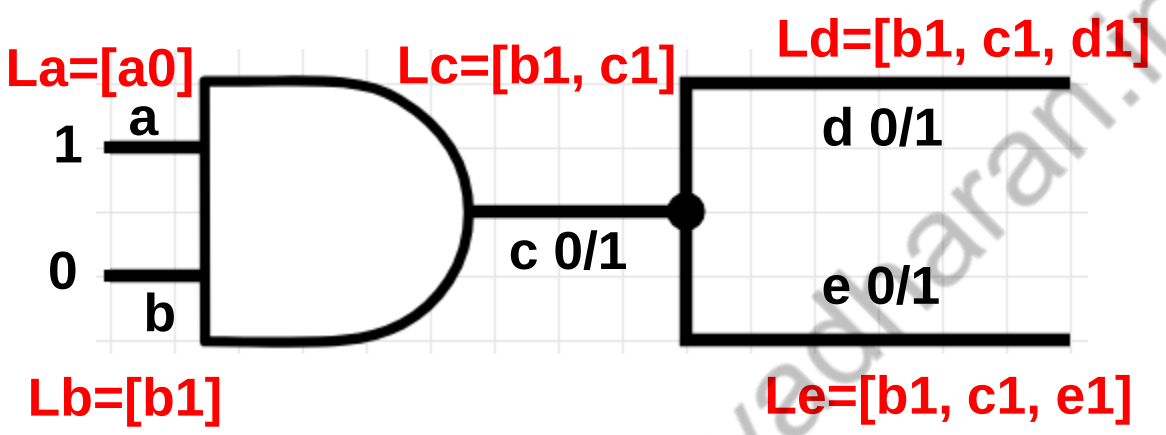
Deductive Fault Simulation



Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$

https://www.youtube.com/watch?v=zTLI2i69tKQ

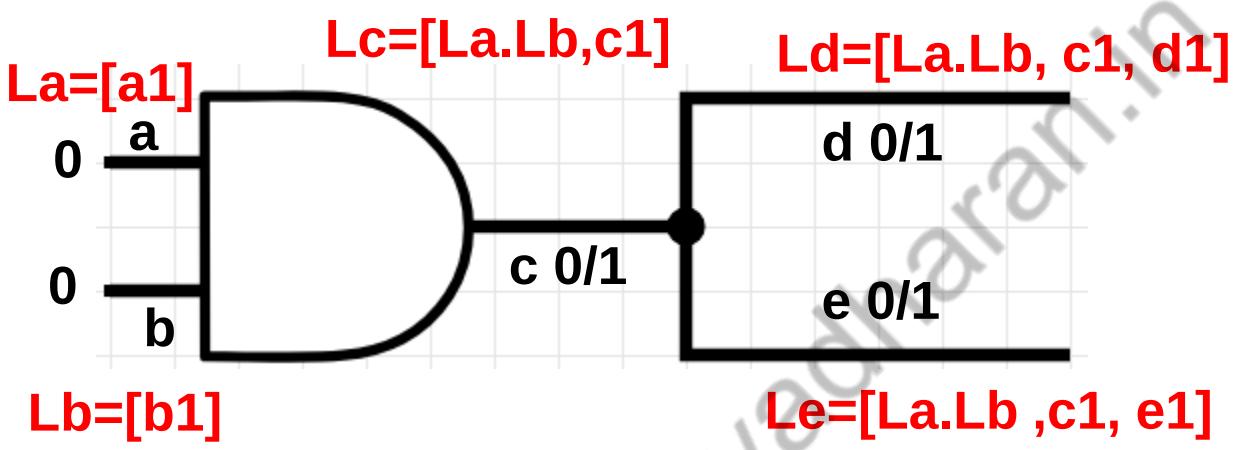
Deductive Fault Simulation



Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$
AND	1	0	0	$(Lb \cap \overline{La}) \cup c1$

https://www.youtube.com/watch?v=zTLI2i69tKQ

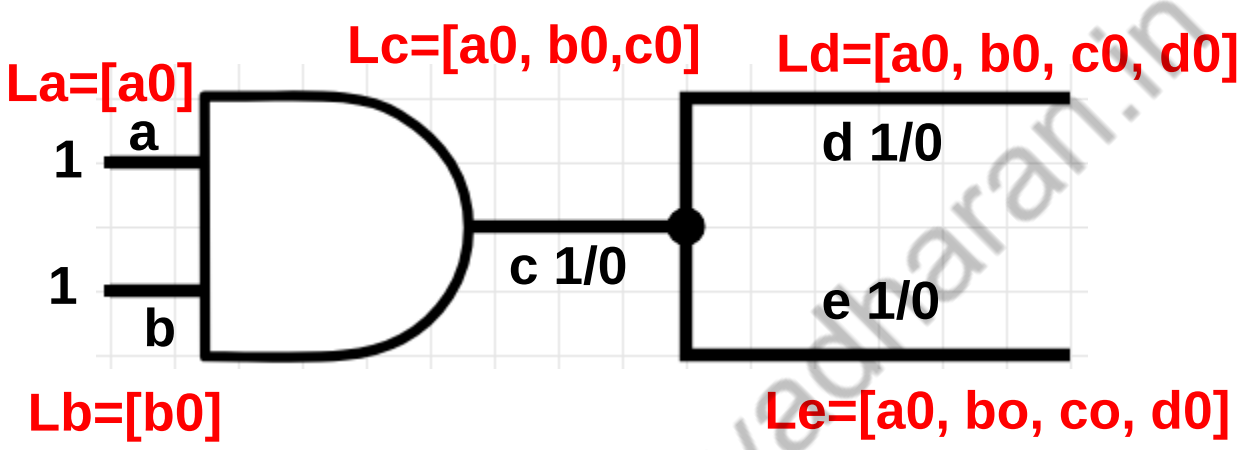
Deductive Fault Simulation



Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$
AND	1	0	0	$(Lb \cap \overline{La}) \cup c1$
AND	0	0	0	$(La \cap Lb) \cup c1$

https://www.youtube.com/watch?v=zTLI2i69tKQ

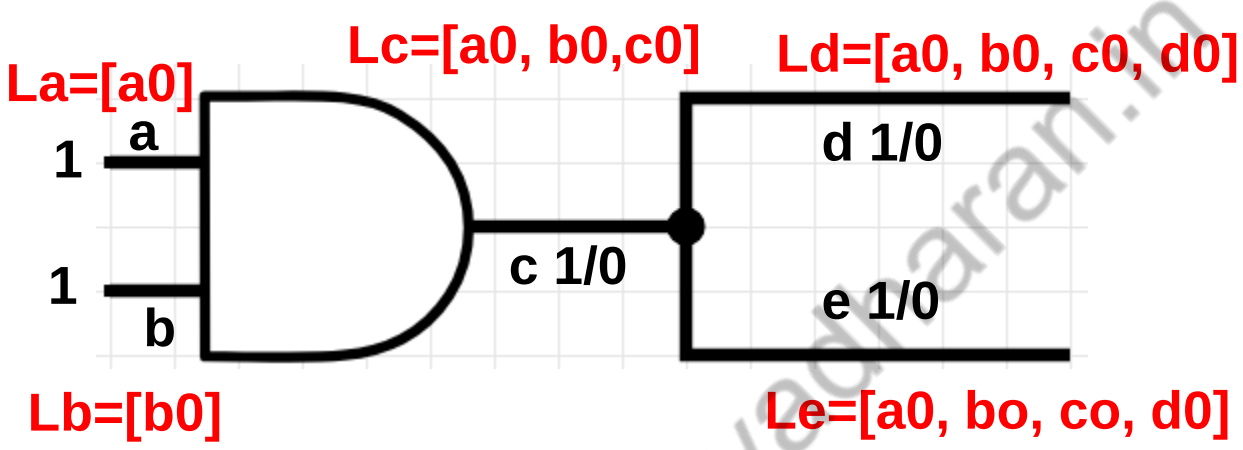
Deductive Fault Simulation



Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$
AND	1	0	0	$(Lb \cap \overline{La}) \cup c1$
AND	0	0	0	$(La \cap Lb) \cup c1$
AND	1	1	1	$(La \cup Lb) \cup c0$

https://www.youtube.com/watch?v=zTLI2i69tKQ

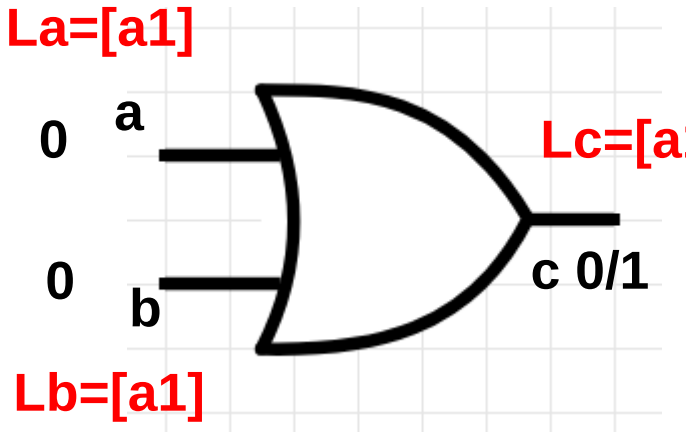
Deductive Fault Simulation

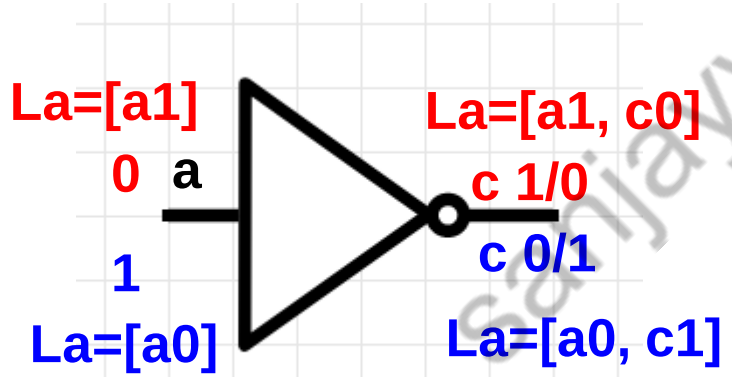


Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	a1, c1
AND	1	0	0	b1, c1
AND	0	0	0	c1
AND	1	1	1	a0, b0, c0

https://www.youtube.com/watch?v=zTLI2i69tKQ

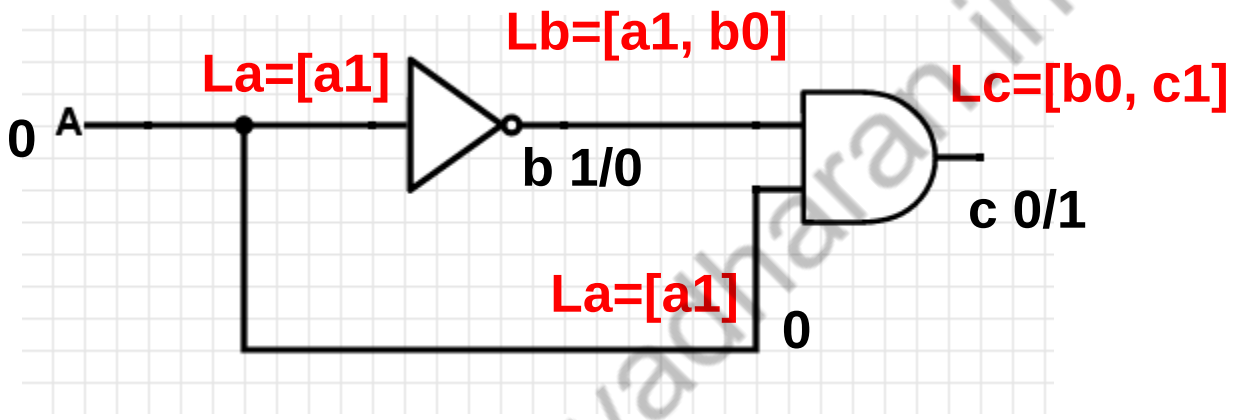
Deductive Fault Simulation





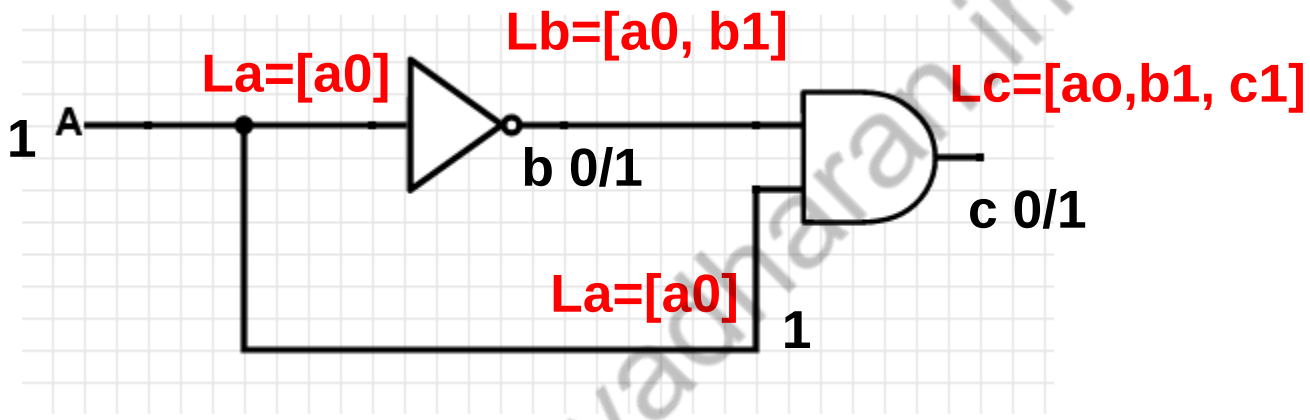
Gate type	Inputs a	Inputs b	Output c	Output fault list L_c
AND	0	0	0	$[L_a \cap L_b] \cup c_1$
	0	1	0	$[L_a \cap \overline{L_b}] \cup c_1$
	1	0	0	$[\overline{L_a} \cap L_b] \cup c_1$
	1	1	1	$[L_a \cup L_b] \cup c_0$
OR	0	0	0	$[L_a \cup L_b] \cup c_1$
	0	1	1	$[\overline{L_a} \cap L_b] \cup c_0$
	1	0	1	$[L_a \cap \overline{L_b}] \cup c_0$
	1	1	1	$[L_a \cap L_b] \cup c_0$
NOT	0	-	1	$L_a \cup c_0$
	1	-	0	$L_a \cup c_1$

Deductive Fault Simulation



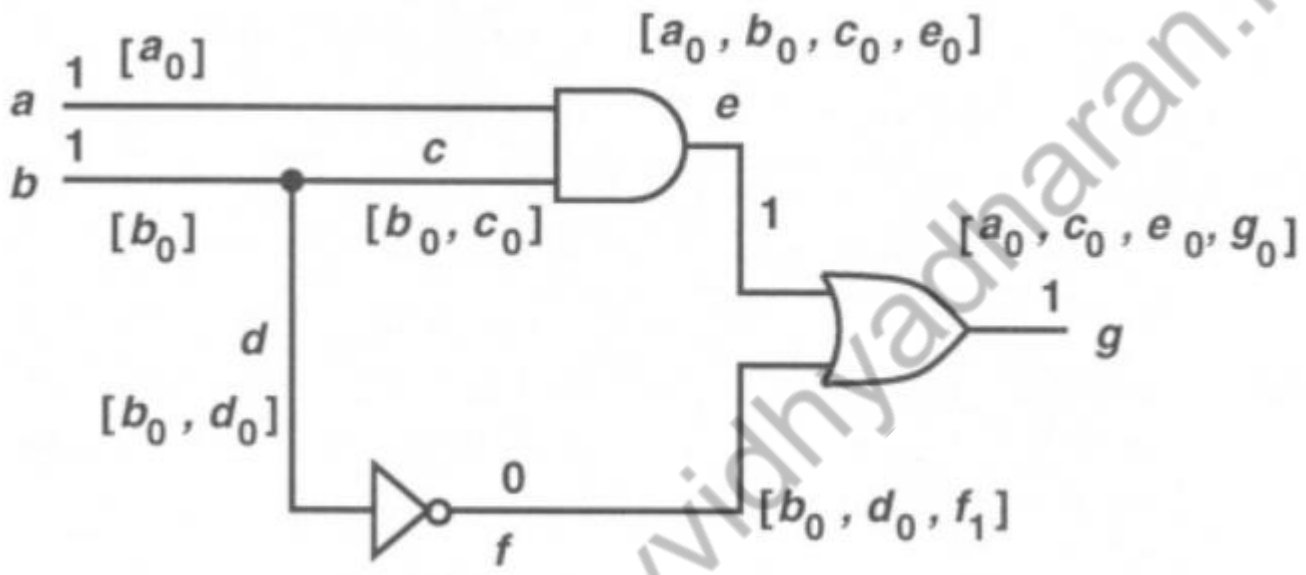
Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$
AND	1	0	0	$(Lb \cap \overline{La}) \cup c1$
AND	0	0	0	$(La \cap Lb) \cup c1$
AND	1	1	1	$(La \cup Lb) \cup c0$

Deductive Fault Simulation



Gate Type	Inputs		Output	O/P Fault List
	a	b	c	
AND	0	1	0	$(La \cap \overline{Lb}) \cup c1$
AND	1	0	0	$(Lb \cap \overline{La}) \cup c1$
AND	0	0	0	$(La \cap Lb) \cup c1$
AND	1	1	1	$(La \cup Lb) \cup c0$

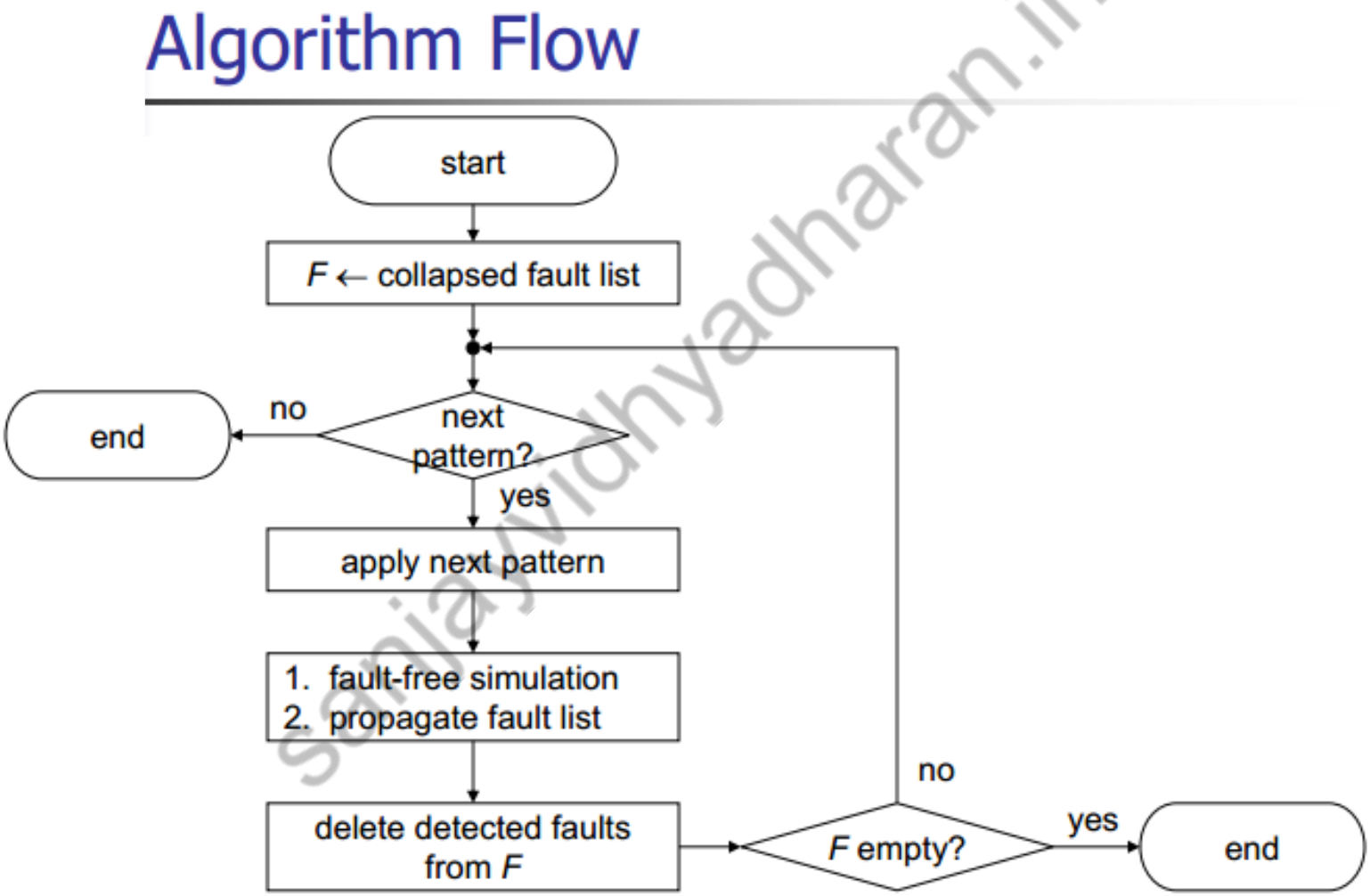
Deductive Fault Simulation



OR	0	0	0	$[L_a \cup L_b] \cup c_1$
	0	1	1	$[\overline{L_a} \cap L_b] \cup c_0$
	1	0	1	$[L_a \cap \overline{L_b}] \cup c_0$

Single Vector Simulation will give what are the faults which can be detected and what should be the correct expected result.

Deductive Fault Simulation



Deductive Fault Simulation

Advantages:

Very efficient

Simulate all faults in one pass

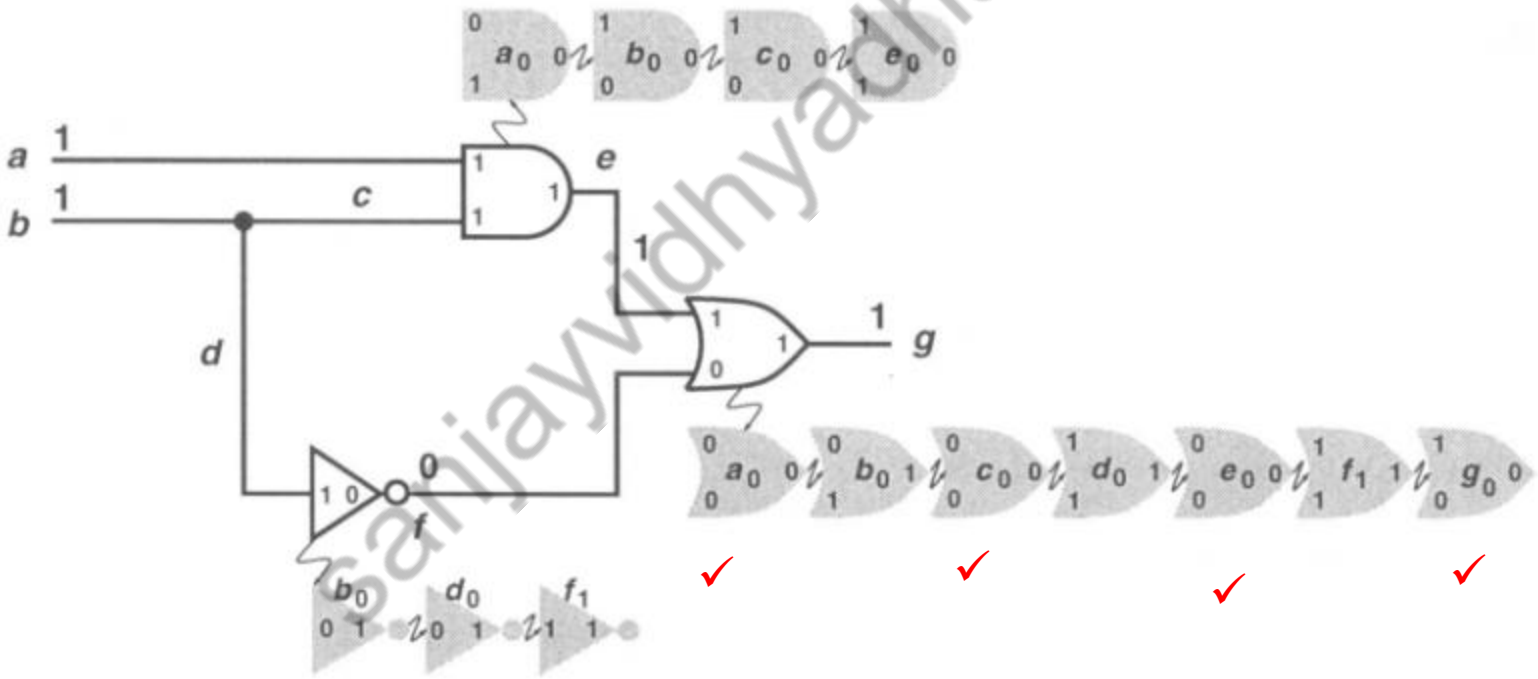
Disadvantages:

Not easy to handle unknowns

Only for zero-delay timing model Potential memory management problem

Concurrent Fault Simulation

It can handle various types of circuit models, faults, signal states, and timing models. It basically extends the event-driven simulation method to the simulation of faults in the most efficient way and faster. Data from previous simulation is retained.



Also gives information of Faults not detected. They need to be carried forward.

Concurrent Fault Simulation

Simulate only differential parts of whole circuit
Event-driven simulation with fault-free and faulty circuits simulated altogether
Concurrent fault list for each gate
Consist of a set of bad gates
Fault index & associated gate I/O values
Initially only contains local faults
Fault propagate from previous stage

Concurrent Fault Simulation

Good event

Events that happen in good circuit

Affect both good gates and bad gates

Bad event

Events that occur in the faulty circuit of corresponding fault

Affect only bad gates

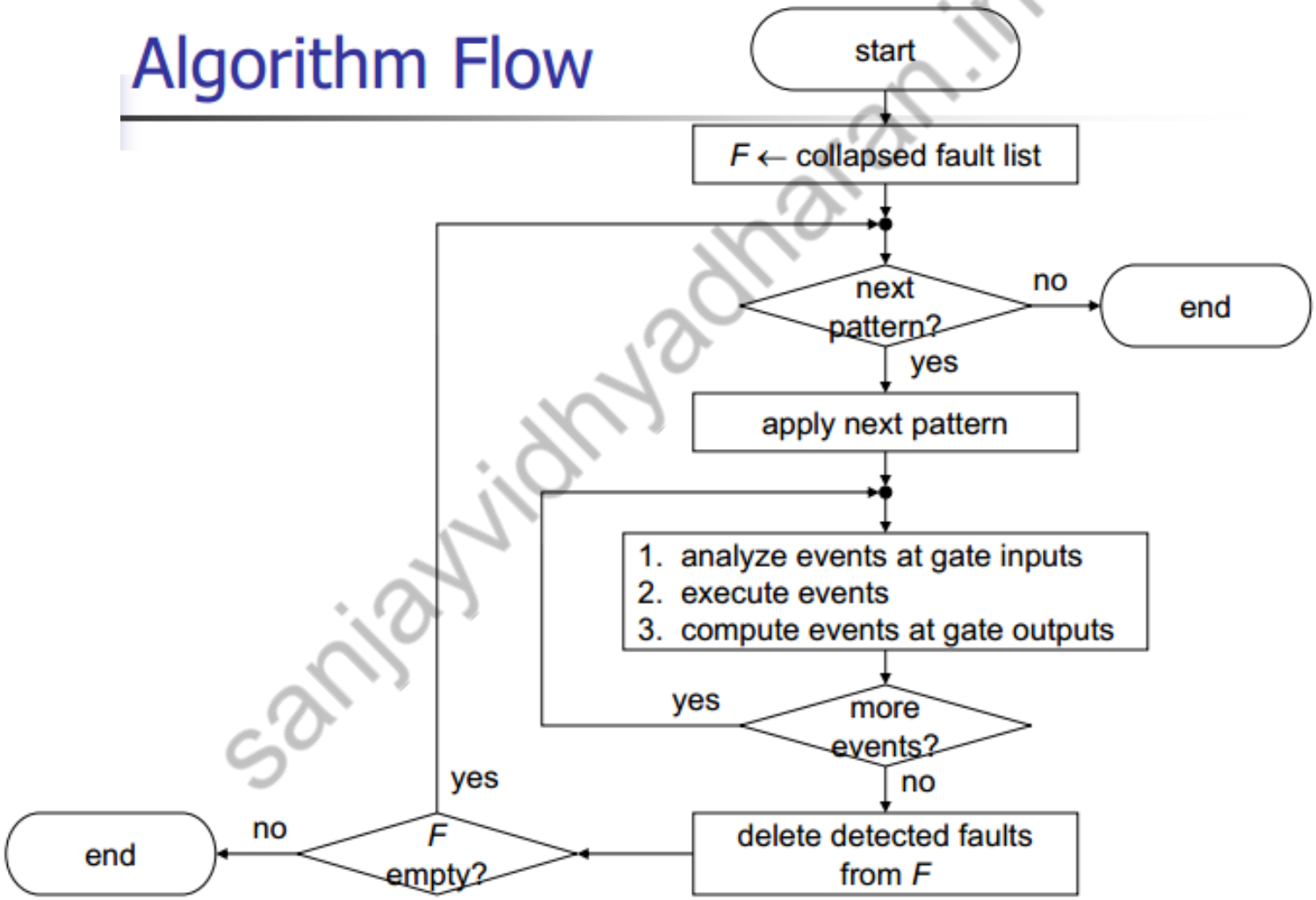
Diverge

Addition of new bad gates

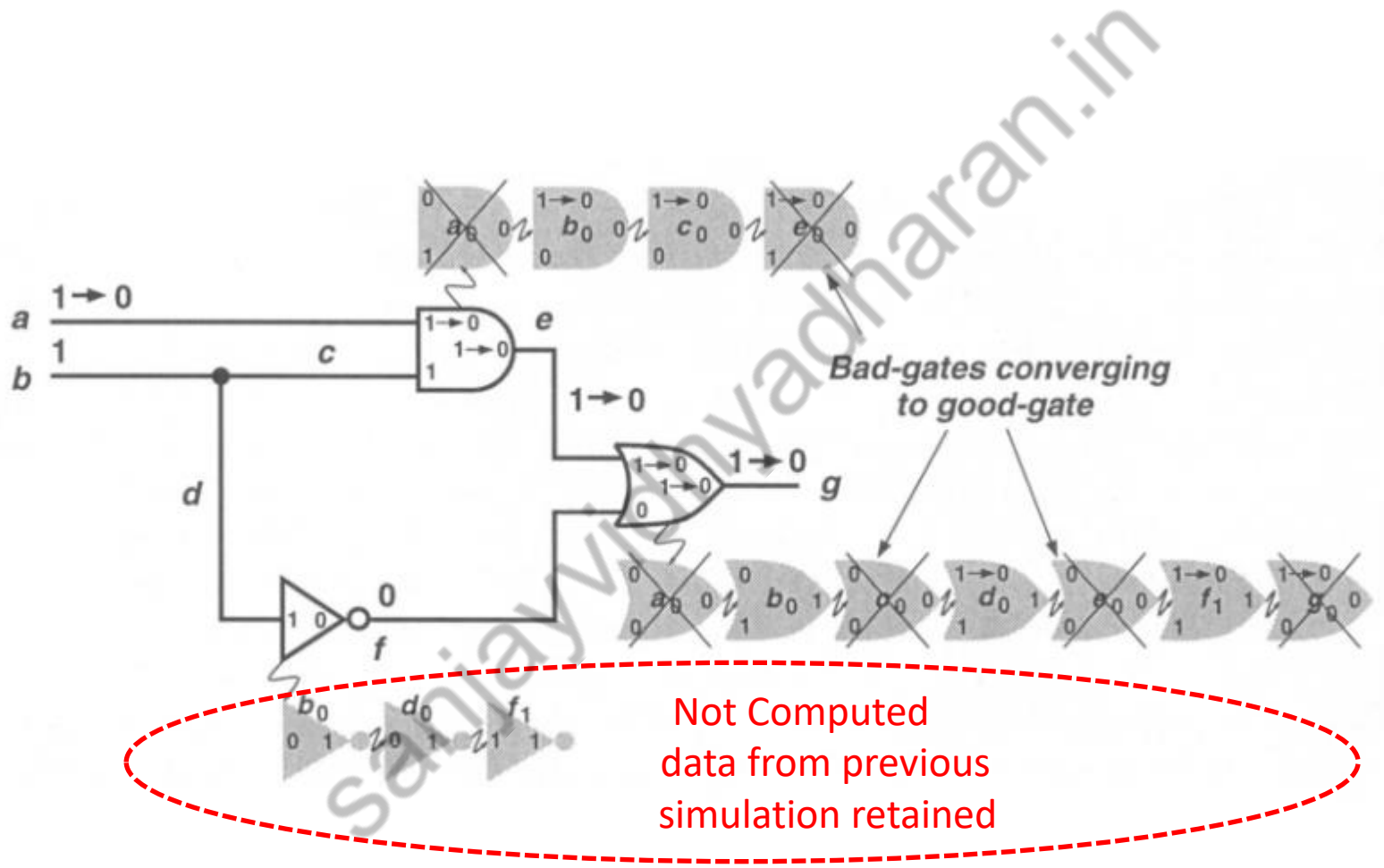
Converge

Removal of bad gates whose I/O signals are the same as corresponding good gates

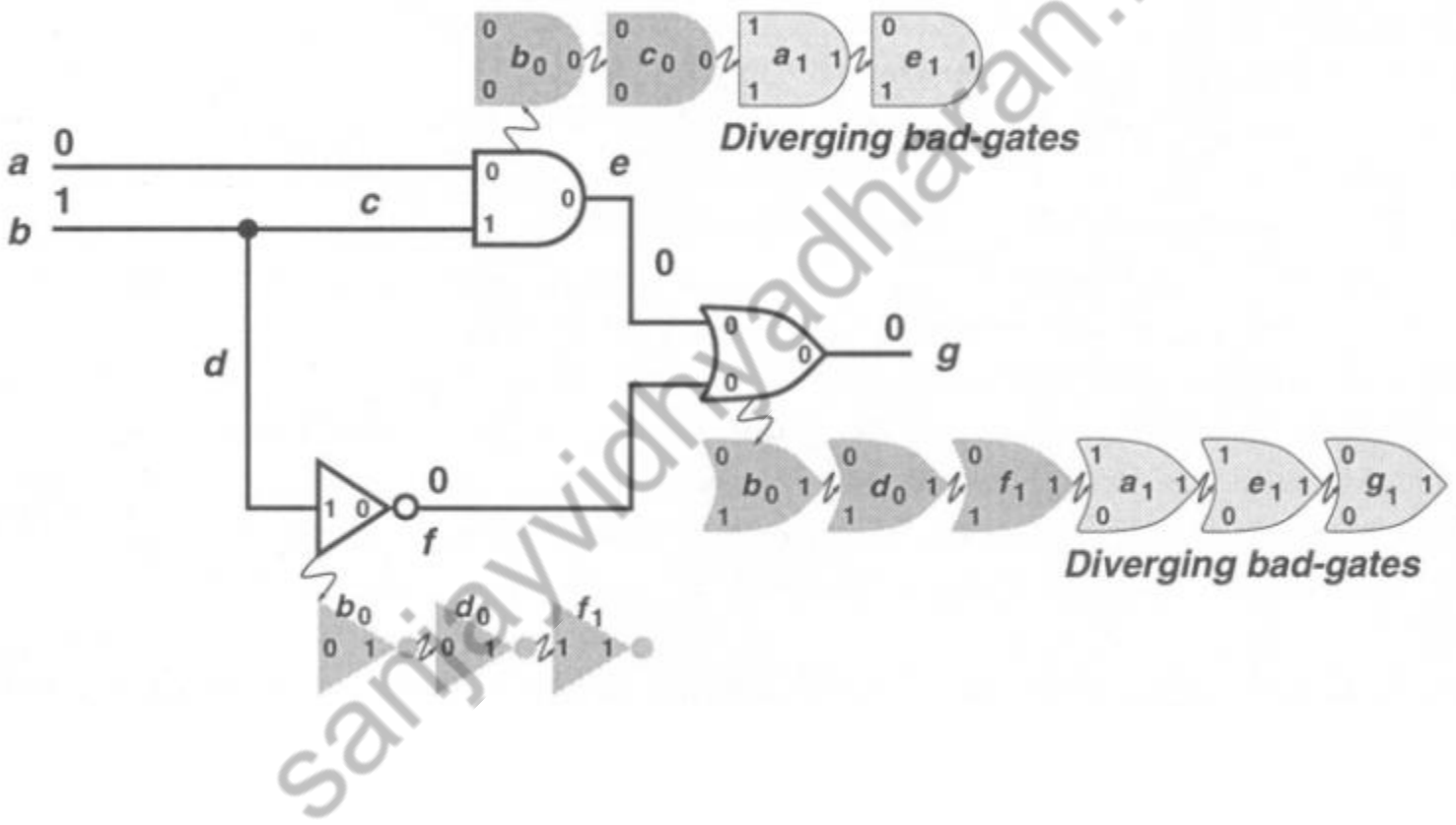
Concurrent Fault Simulation



Concurrent Fault Simulation



Concurrent Fault Simulation



Concurrent Fault Simulation

Advantages

Efficient

Faults can be simulated in any modeling style or detail supported in true-value simulation (offers most flexibility.)

Faster than other methods

Disadvantages

Potential memory problem

Size of the concurrent fault list changes at run time

Comparison of Fault Simulation Techniques

Speed

Serial fault simulation: slowest

Parallel fault simulation: O(n3), n: num of gates

Deductive fault simulation: O(n2)

Concurrent fault is faster than deductive fault simulation

Memory usage

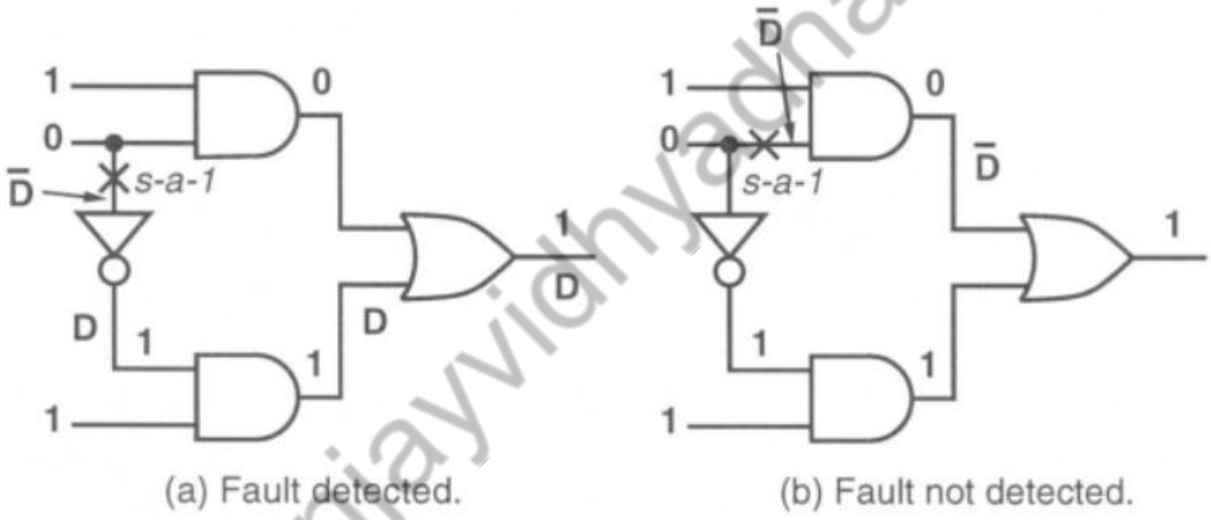
Serial fault simulation, parallel fault simulation: no problem

Deductive fault simulation: dynamic allocate memory and hard to predict size

Concurrent fault simulation: more severe than deductive fault simulation

Roth's TEST-DETECT Algorithm

The circuit is simulated for a vector in the true-value mode. This determines the states of all lines. Next, faults are analyzed one at a time to determine which faults are detected by the presently simulated vector. The analysis is based on Roth's *D*-calculus that allows a composite representation of a signal in the fault-free and faulty circuits.



(a) Fault detected. (b) Fault not detected.

In Roth's *D*-calculus D = (1,0) and D'=(0,1). D algebra will be covered in subsequent classes.

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