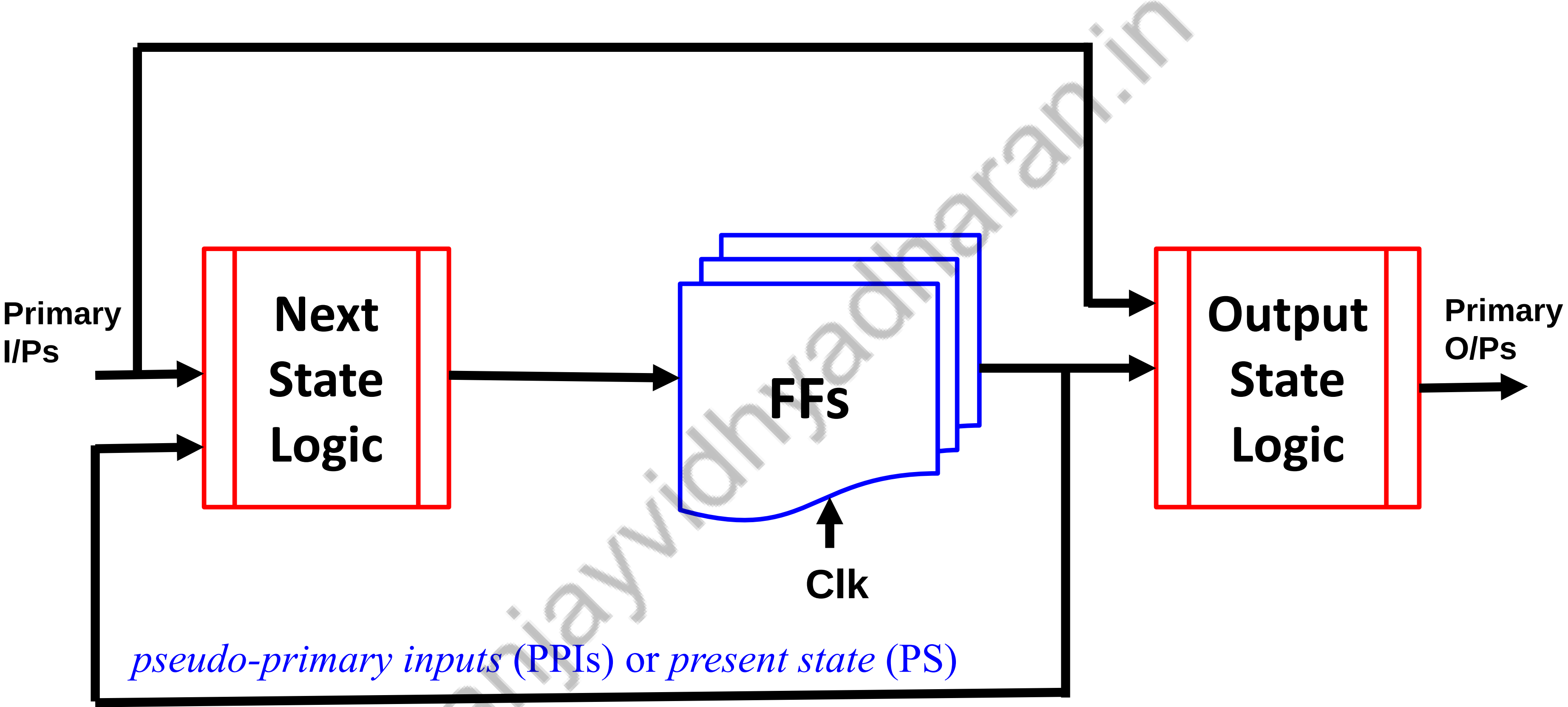
Testability of VLSI

Lecture 08:
Testing of Sequential Circuits

By Dr. Sanjay Vidhyadharan

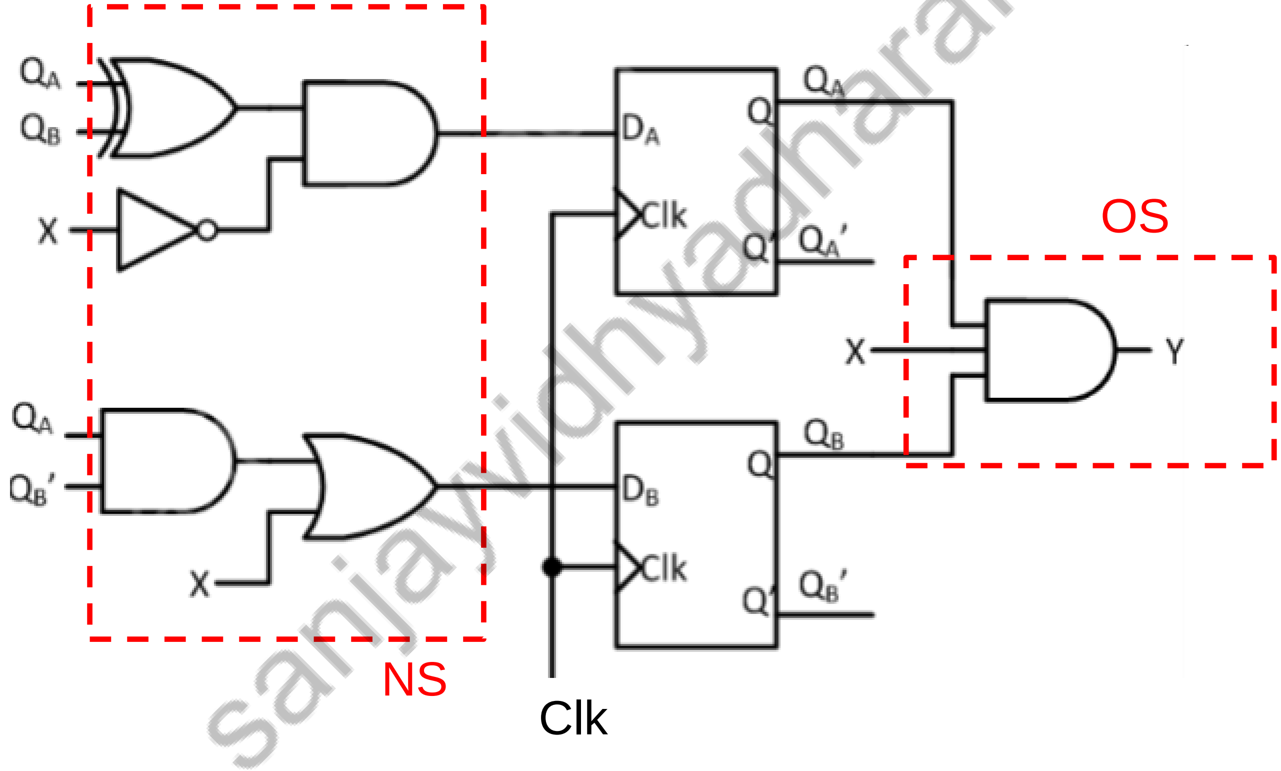
Sequential Circuit



Testing of Combination Blocks is similar to that we studied earlier
Difference being that the inputs from FFs are not directly controllable

Sequential Circuit

Design of sequence detector overlapping (1001)



Sequential Circuit Testing

1. The circuit contains internal memory whose state is not known at the beginning of the test. The test must, therefore, initialize the circuit to a known state.

2. After test inputs are applied, the final state of the internal memories must be inferred only indirectly from primary outputs. Only in special cases can the internal memory be made controllable and observable for testing, sometimes at the cost of extra hardware

3. Test for a fault in sequential logic essentially contains

   (a) initialization of the internal memory,

   (b) a combinational test to activate the fault and bring its effects to the boundary of the combinational logic,

   (c) if the fault has affected one or more memory elements, then observation of the state of one of the affected elements at a primary output.

Thus, the test of a fault may be a sequence of several vectors that must be applied in the specified order.

ATPG for Single-Clock Synchronous Circuits

Time-Frame Expansion Method

- Tests generated by a combinational ATPG method.
- Very efficient for circuits described at the Boolean gate-level.
- Its efficiency degrades significantly with cyclic structure, multiple-clocks, or asynchronous circuitry.

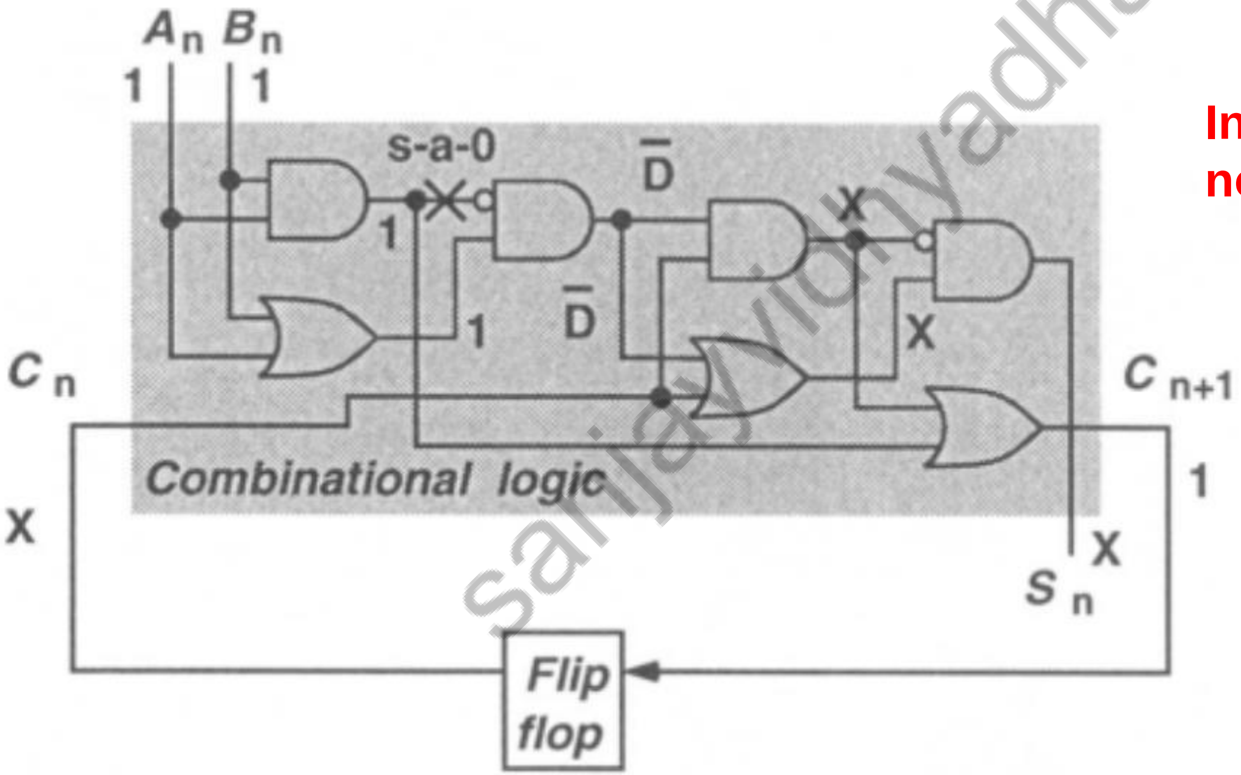
ATPG for Single-Clock Synchronous Circuits

Time-Frame Expansion Method

Eg. Serial Adder

1. Test Vector for Sensitisation? **11**
2. Value of Cn for Propagation? **1 or 0**
3. Initialisation Test Vector for? **{00,01, 10,11}**

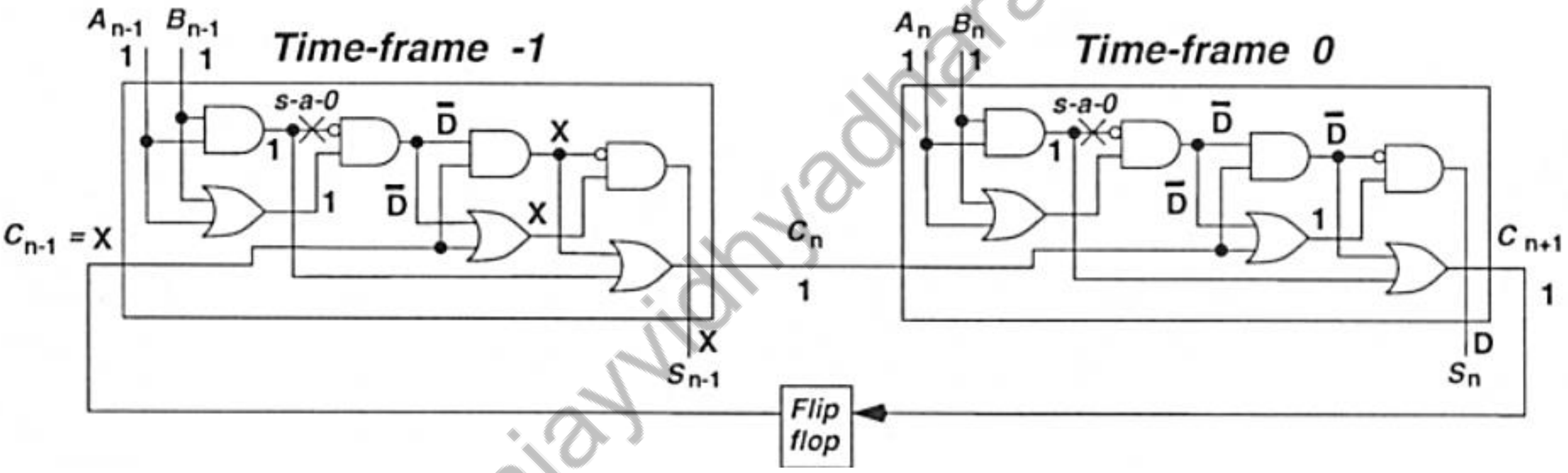
Initialisation may or may not sensitise



ATPG for Single-Clock Synchronous Circuits

Time-Frame Expansion Method

Eg. Serial Adder

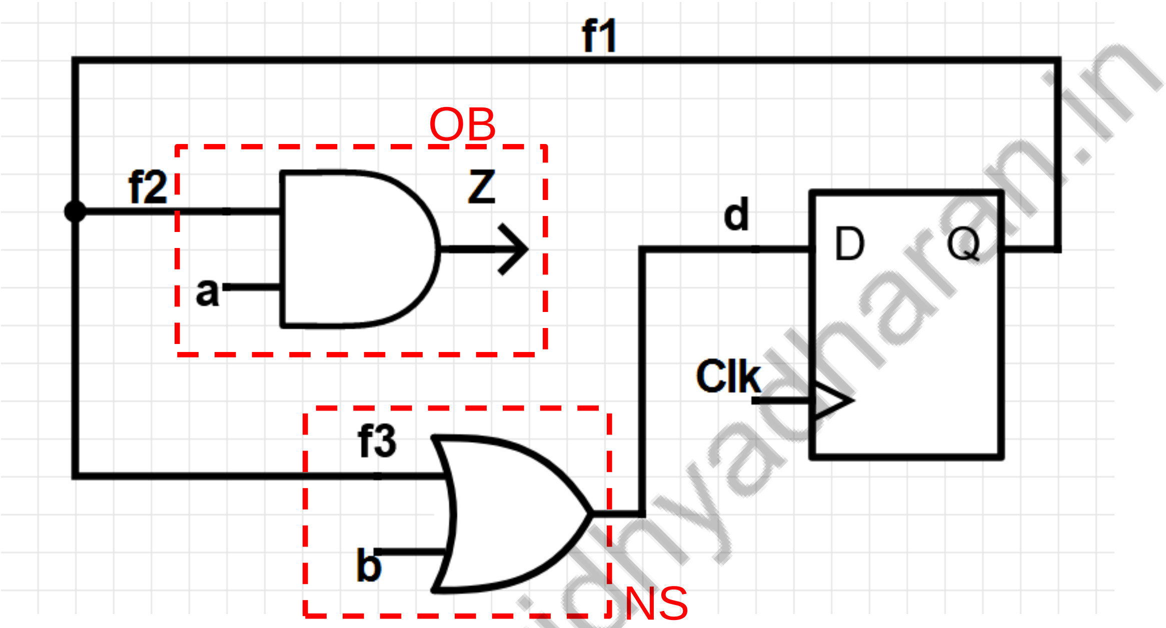


ATPG for Single-Clock Synchronous Circuits

Assumptions

- Single Synchronized Clock for all FFs
- Single Stuck-at Faults in Next Stage and Output Stage Blocks
- No faults internal to FFs
- No Faults in Clock Path

Time-Frame Expansion with D Algorithm

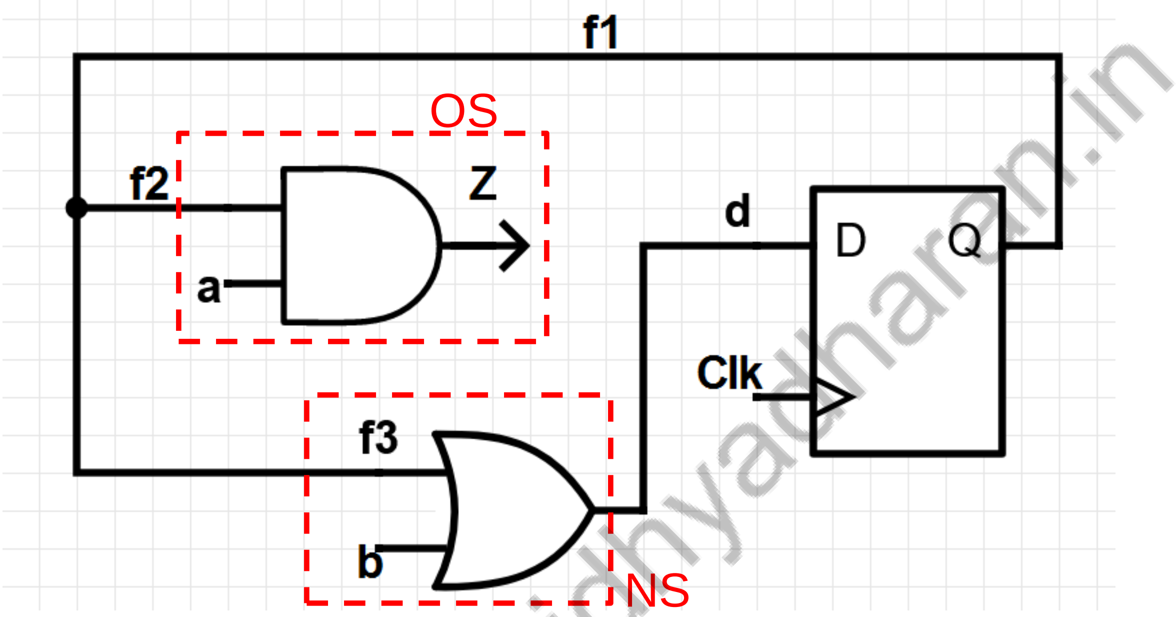


1. Which are the Next Stage and Output Stage Blocks

Test vector to detect sa0 at a ? (1,X) puts “D with f2=1 it gets propagated to Z

Test vector for initialization ? (X,1)

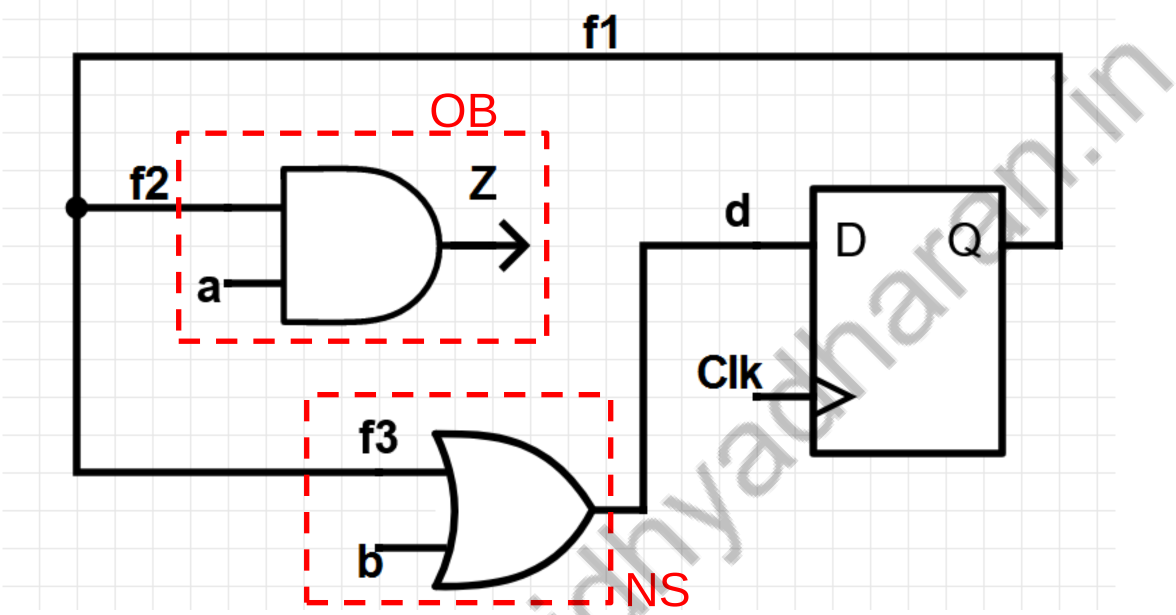
Time-Frame Expansion with D Algorithm



Test vector to detect sa0 at d ? (X,1) puts “D” at d

Next vector? (1,X) propagates “D” to output

Time-Frame Expansion with D Algorithm



Test vector to detect sa0 at f3 ? (X,1) puts "D" at f3

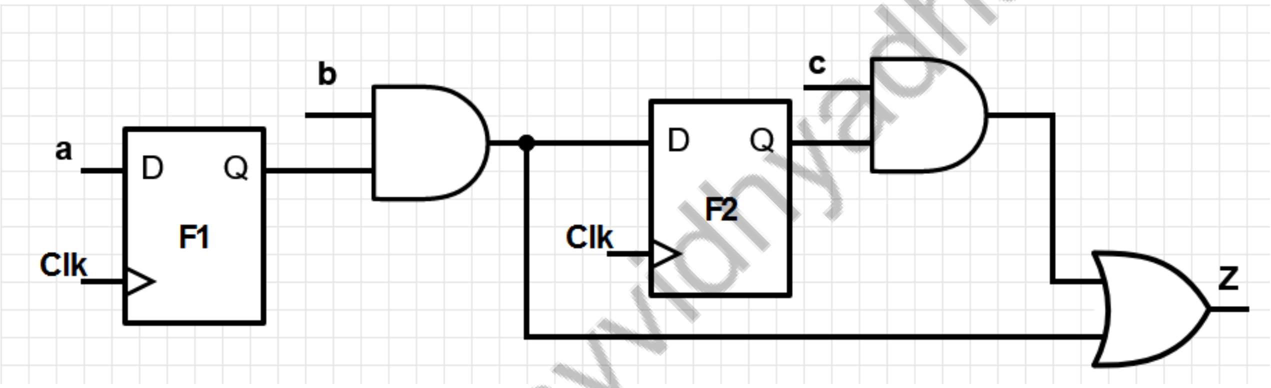
Next vector? (X,0) propagates "D" to f1

Next vector? (1,X) propagates "D" to Z

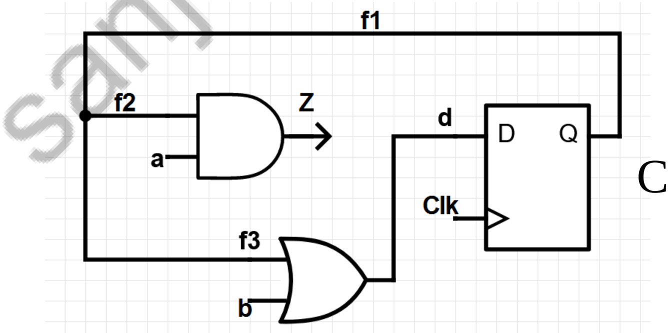
Time-Frame Expansion with D Algorithm

Definitions

1. Sequential Depth of FF
   (a) Sequential Depth is one if O/P of FF controlled by Primary I/Ps
   (b) Sequential Depth is n if O/P of FF controlled by Primary I/Ps and also by at least one Sequential Depth n-1 FF



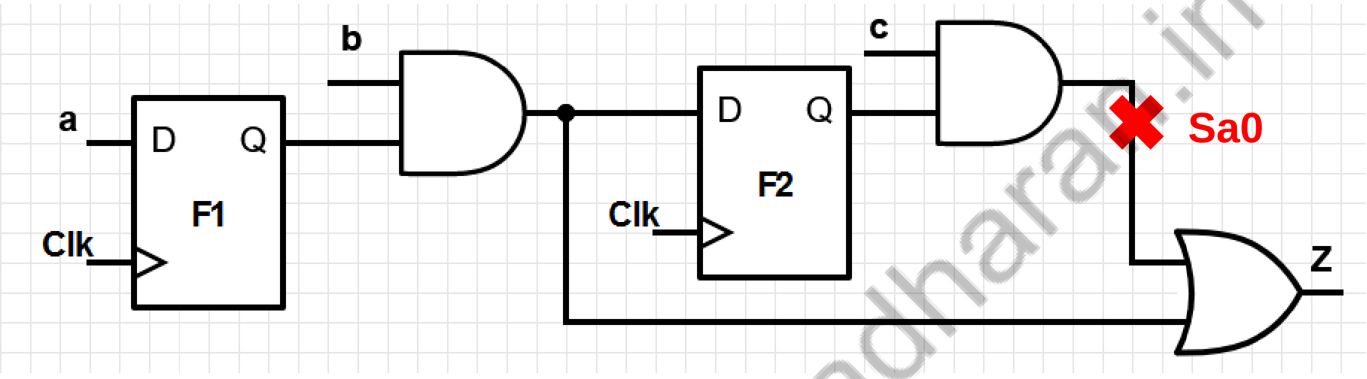
2. Sequential Circuit is Non-Cyclic there are no FFs whose I/P is dependant on its O/P

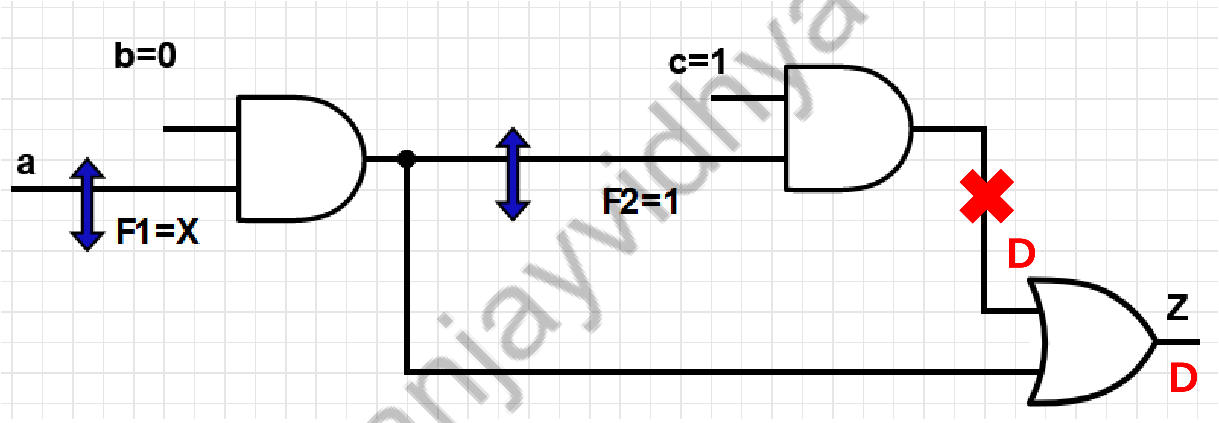


Cyclic

Time-Frame Expansion with D Algorithm

Step 1: Replace all FFs with nets and Perform D algorithm

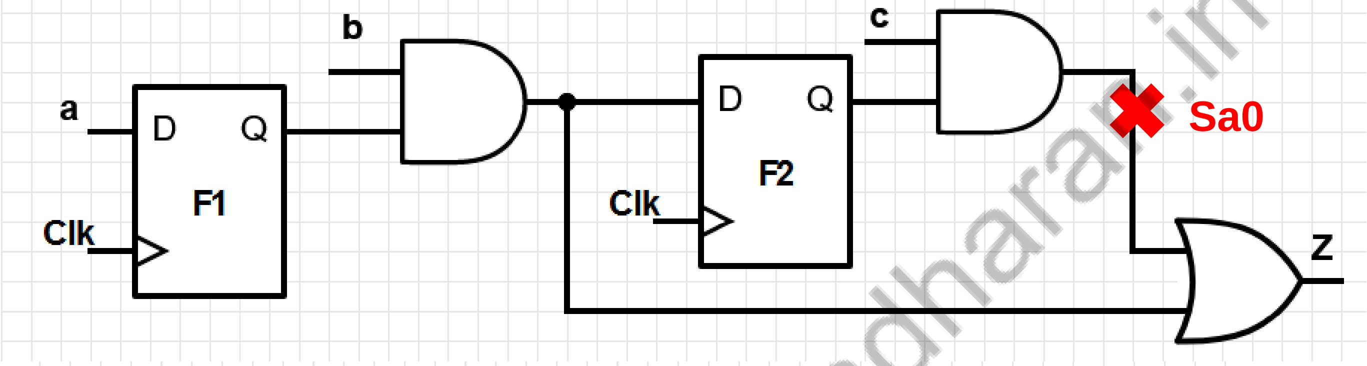


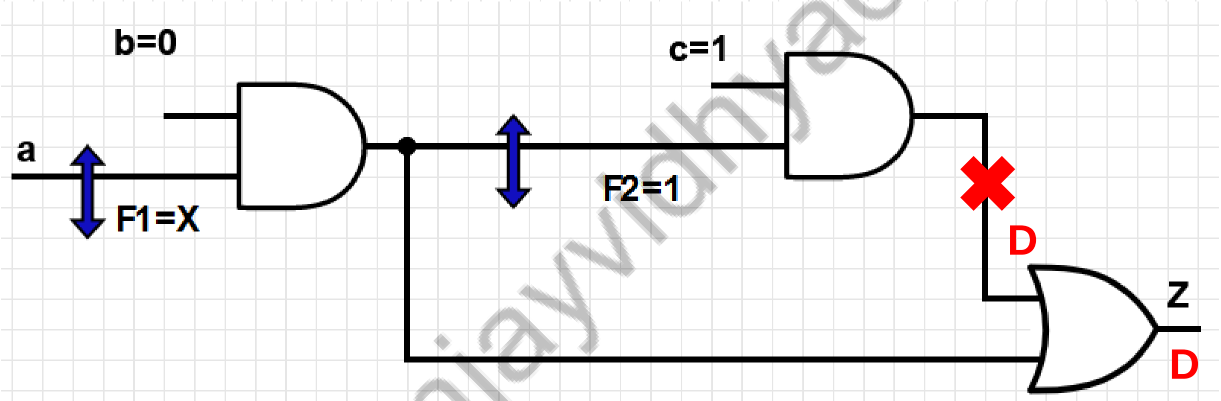


	a	b	c	F1	F2
T=0	X	0	1	X	1

Time-Frame Expansion with D Algorithm

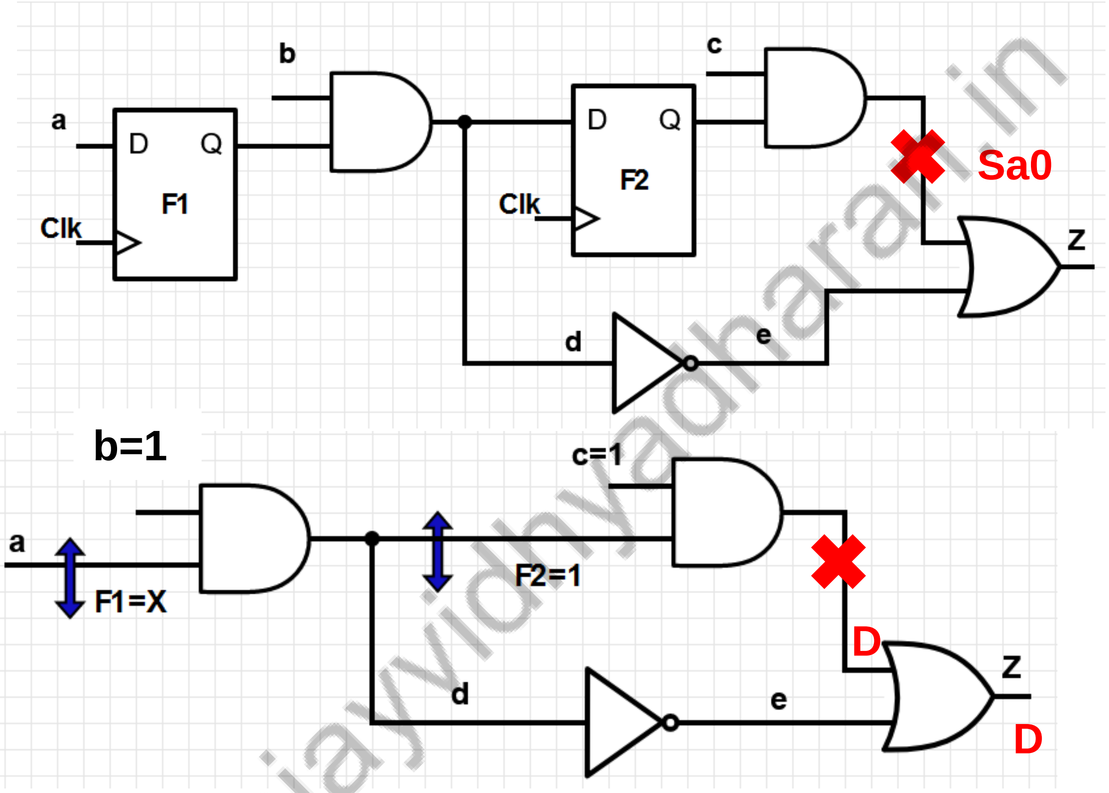
Step 2: A sequential FFs of depth n can can be set to required value in max n clock cycles





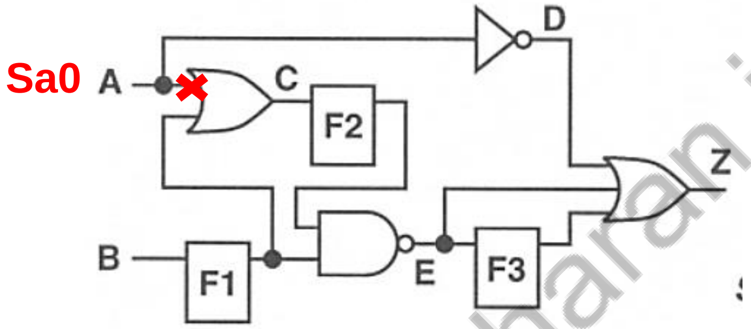
	a	b	c	F1	F2
T=0	X	0	1	X	1
T=-1	X	1	X	1	X
T=-2	1	X	X	X	X

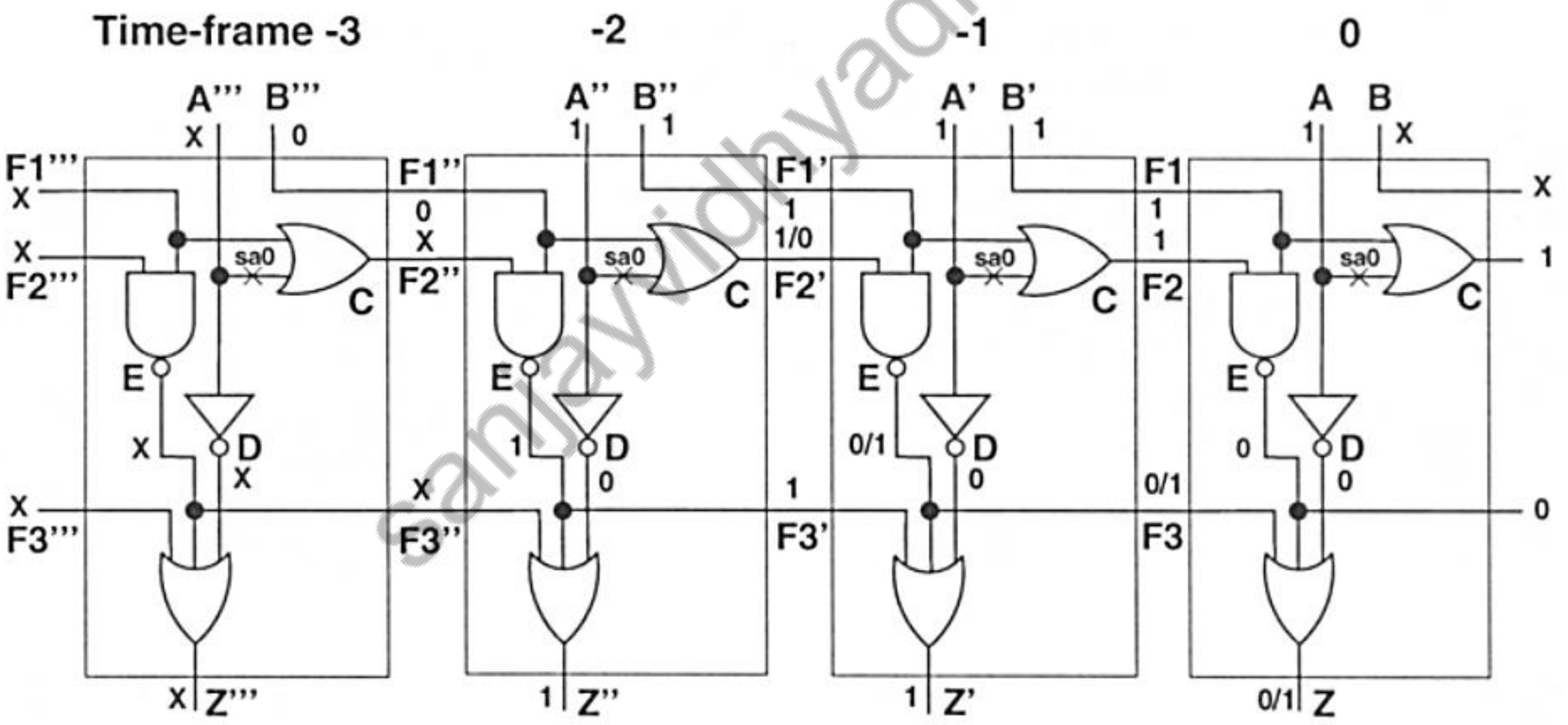
Time-Frame Expansion with D Algorithm



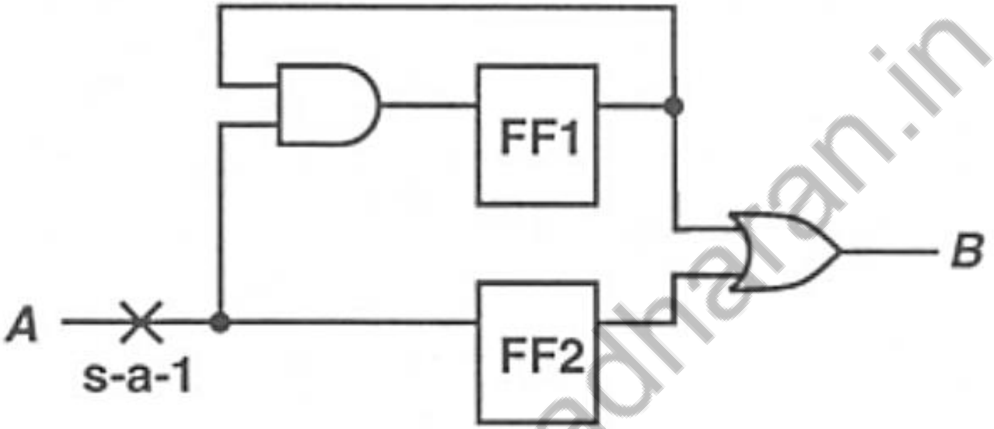
	a	b	c	d	e	F1	F2
T=0	X	1	1	1	0	1	1
T=-1	1	1	X	X	X	1	X
T=-2	1	X	X	X	X	X	X

Time-Frame Expansion with D Algorithm



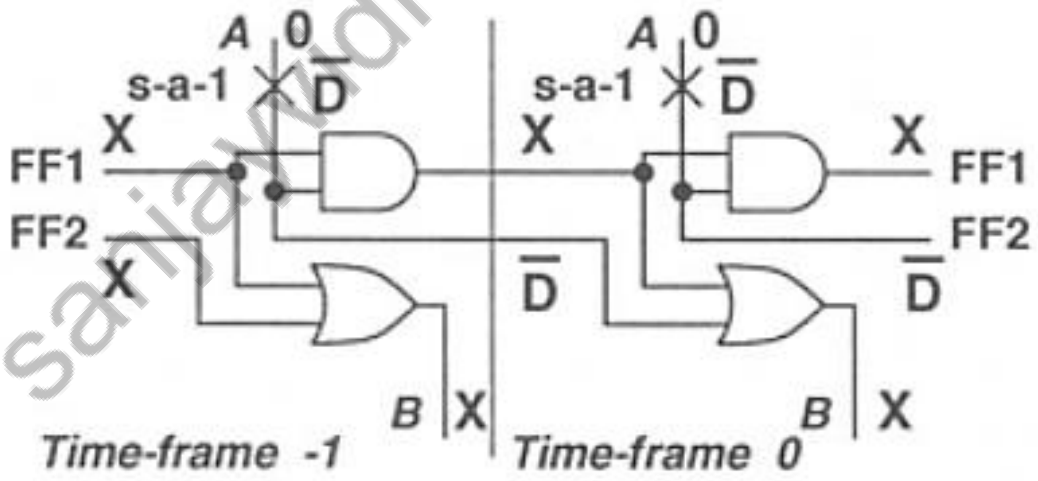


Time-Frame Expansion with D Algorithm



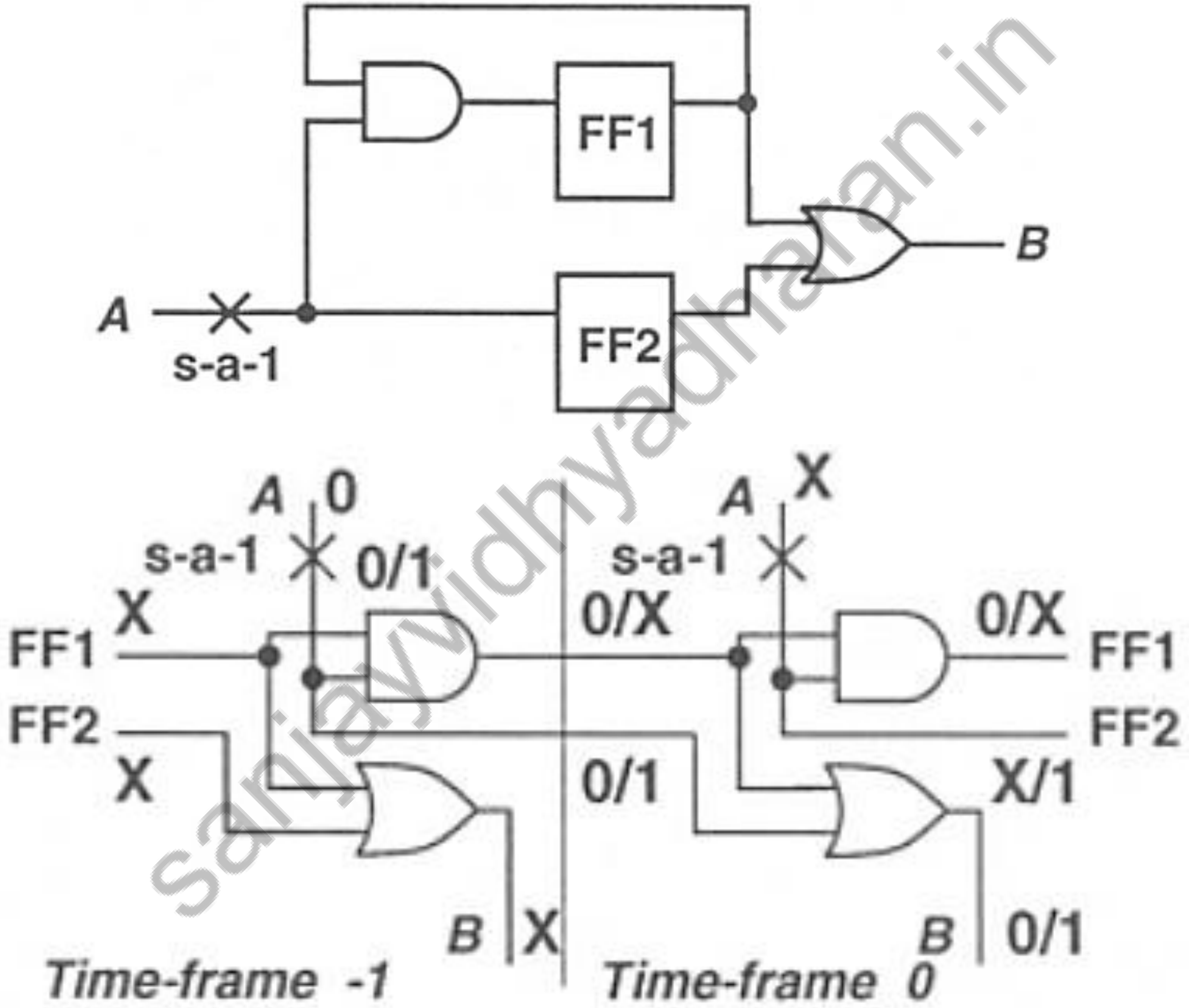
Test vector to detect sa1 at A ?

A=0 puts D’ at A, But cannot initialise FF1 to 0

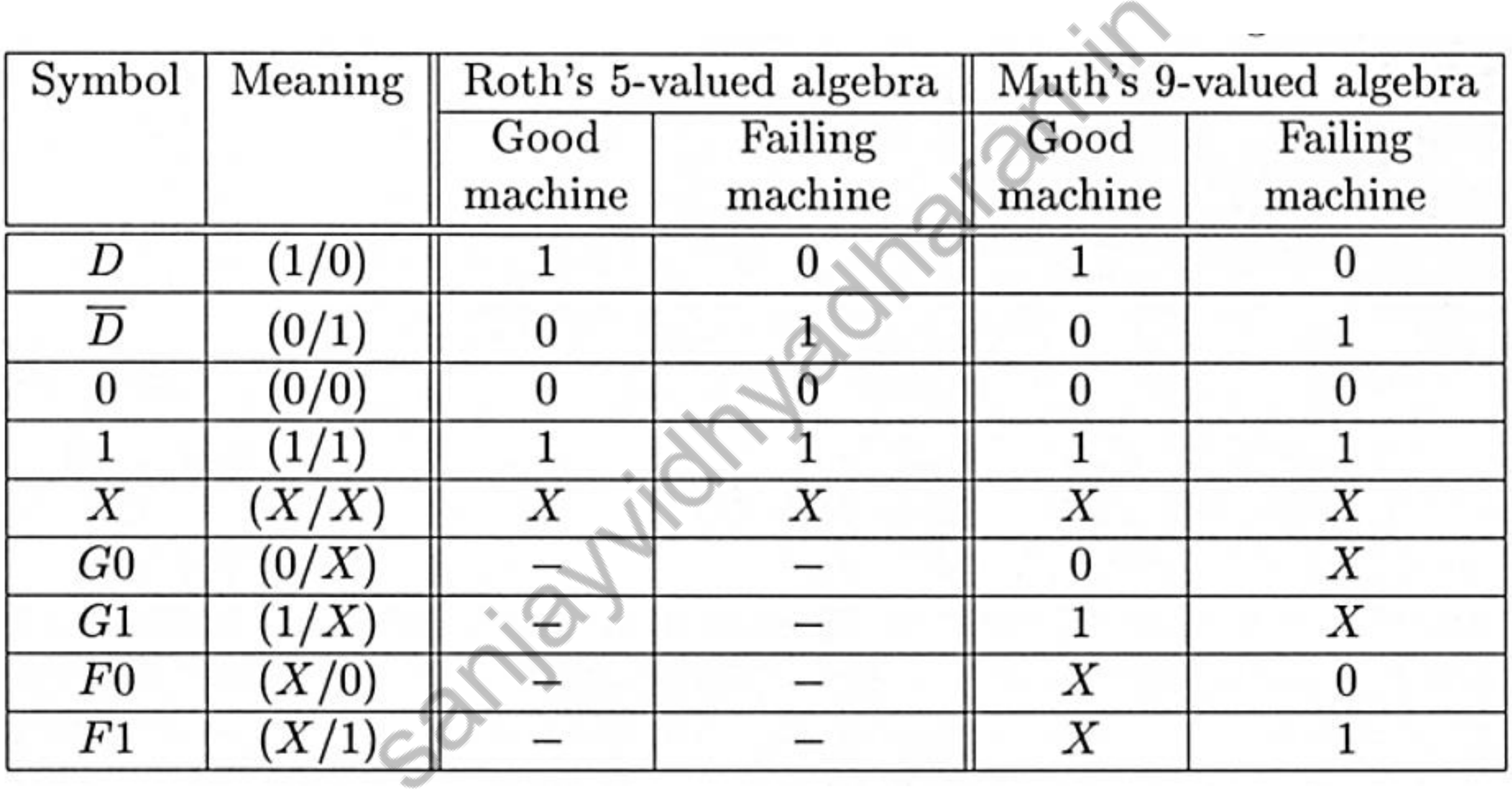


D algorithm not best or sequential circuits

Time-Frame Expansion with Muth Algorithm



Time-Frame Expansion with 9-Valued Algorithm

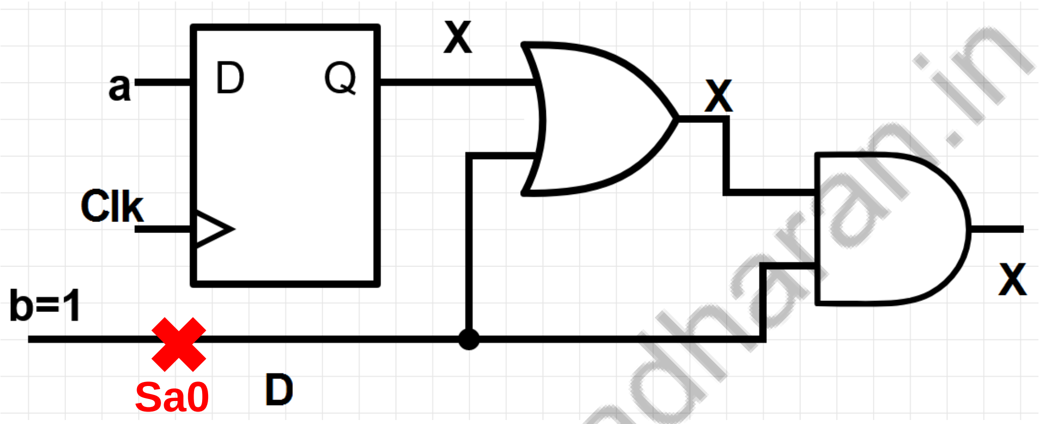


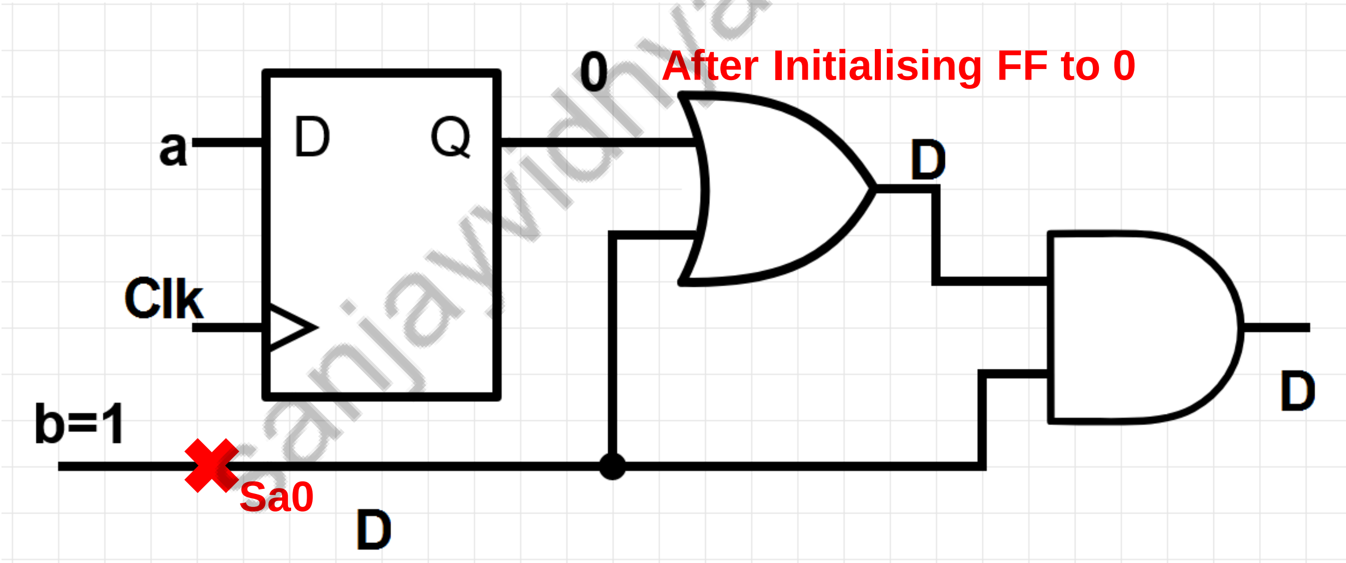
Symbol	Meaning	Roth's 5-valued algebra		Muth's 9-valued algebra	
		Good machine	Failing machine	Good machine	Failing machine
D	$(1/0)$	1	0	1	0
$\overline{D}$	$(0/1)$	0	1	0	1
0	$(0/0)$	0	0	0	0
1	$(1/1)$	1	1	1	1
X	(X/X)	X	X	X	X
$G0$	$(0/X)$	–	–	0	X
$G1$	$(1/X)$	–	–	1	X
$F0$	$(X/0)$	–	–	X	0
$F1$	$(X/1)$	–	–	X	1

Can 9-Valued Algorithm Be used for Combinational Circuit?

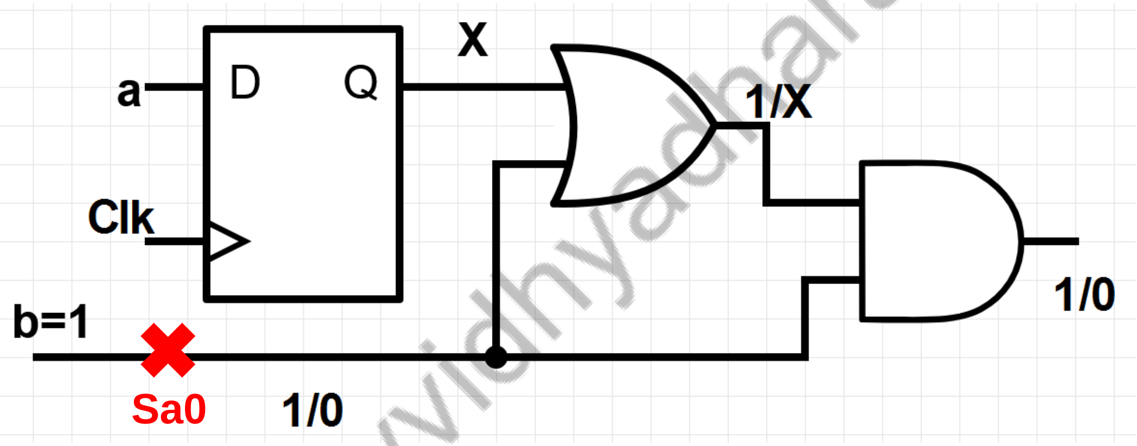
Yes, but not recommended as complexity increases without significant gain

Time-Frame Expansion with D Algorithm





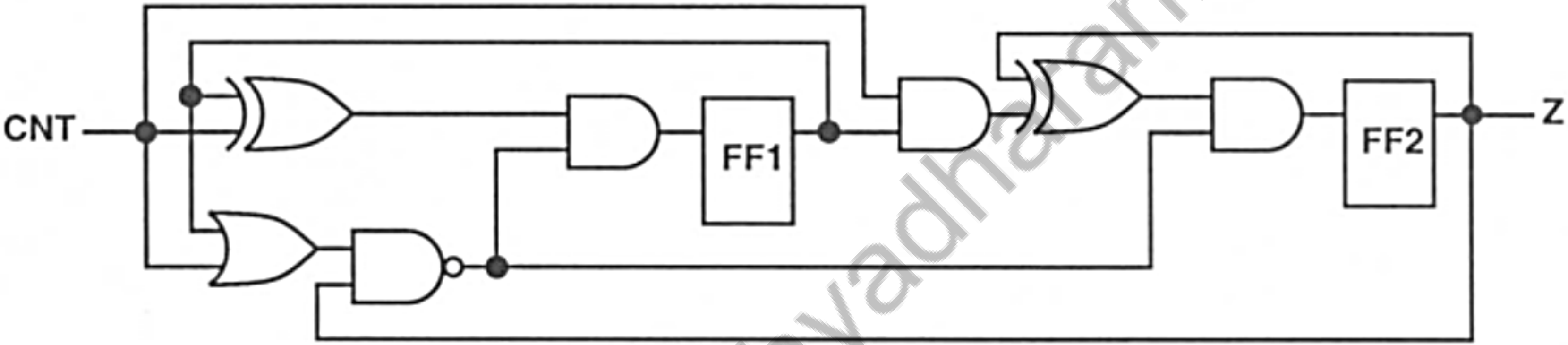
Time-Frame Expansion with Muth Algorithm



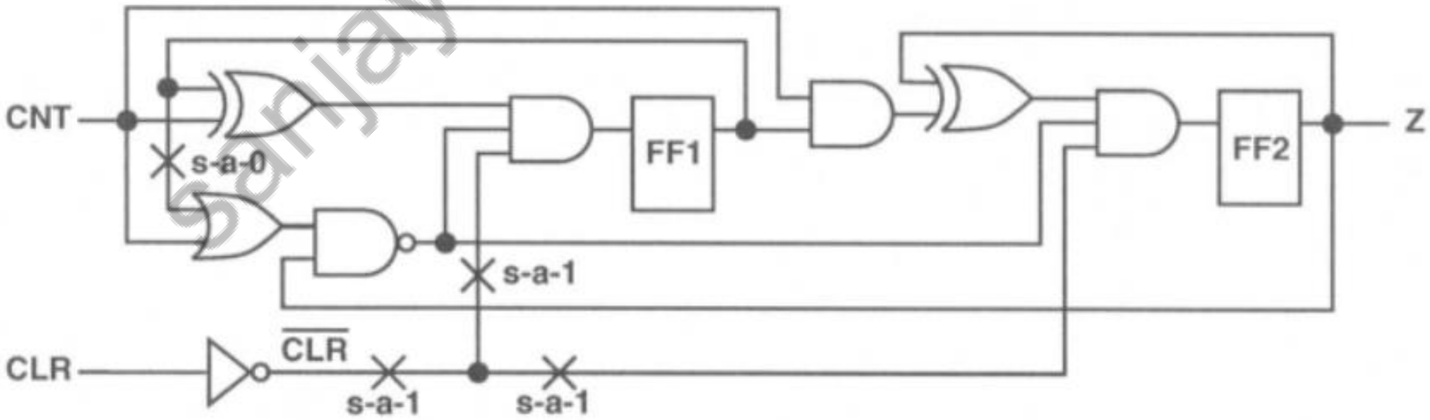
Time-Frame Expansion for Cyclic Circuits

A modulo-3 counter without initialization input

00->01->10->00



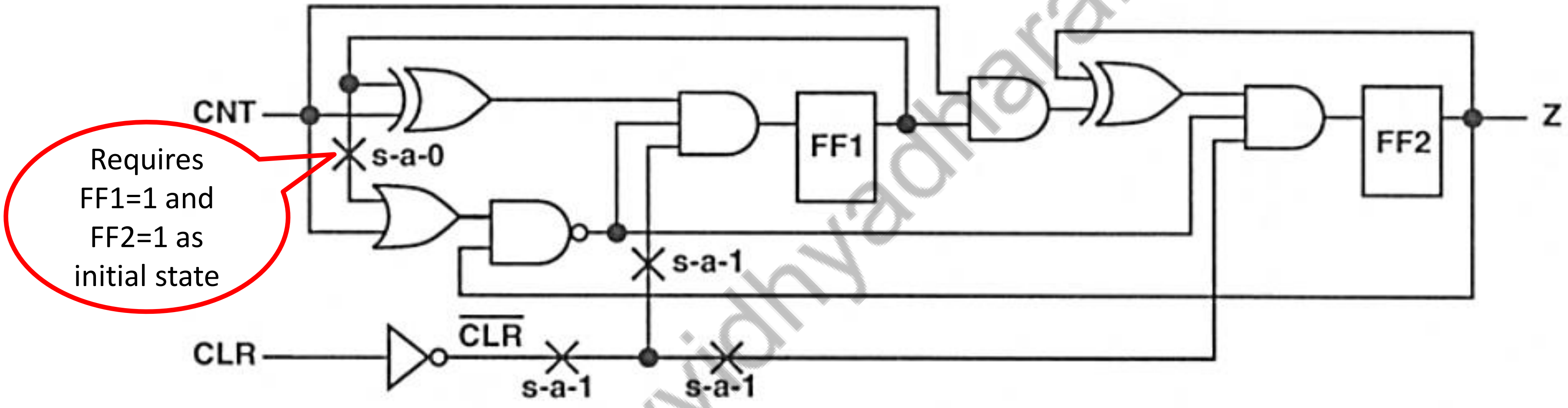
Consider the fault *Z* s-a-0. For any input, the output will be X/0.



Time-Frame Expansion for Cyclic Circuits

A modulo-3 counter without initialization input

00->01->10->00



the CLR input will set the circuit in state. Since the state is set on the application of the clock after *CLR* becomes 1, this operation is called *synchronous initialization*

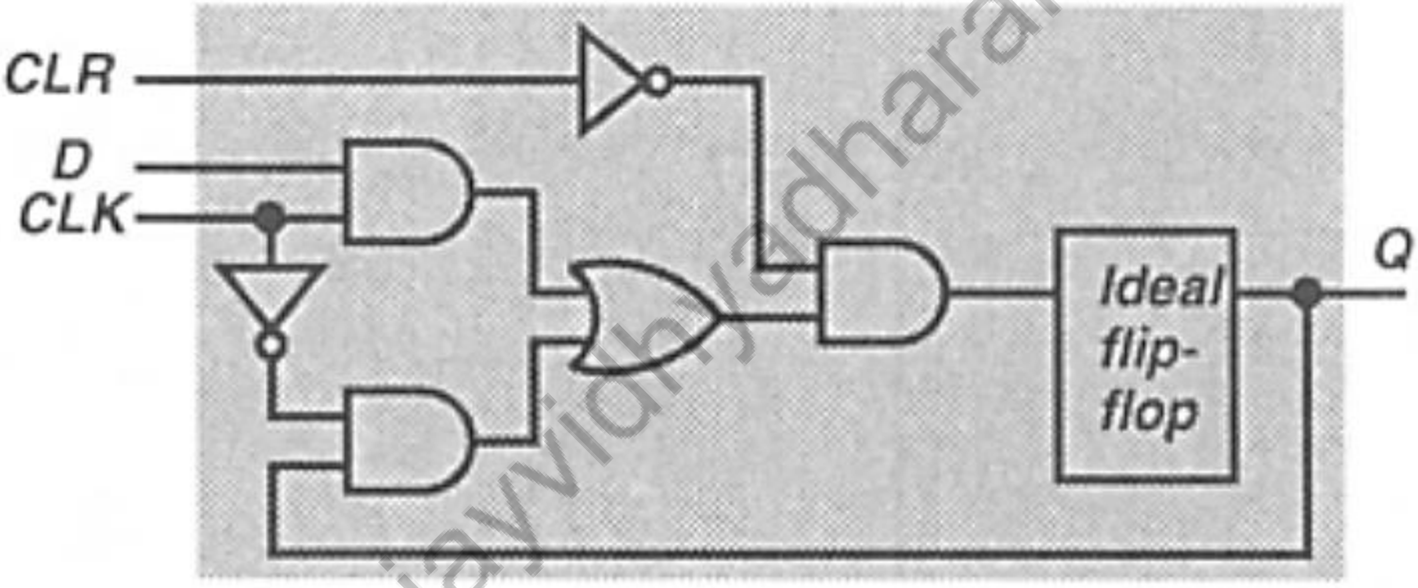
***Asynchronous clear* and *preset* signals are also effective.**

Clock Faults and Multiple-Clock Circuits

Our discussion so far has focused on single-clock circuits. All flip-flops were controlled by one clock, which was a primary input to the circuit. For test generation this clock was modeled only implicitly. That is why many of our circuit diagrams show flip-flops without clock signals. It was assumed that one input vector is applied per clock cycle This approach provides simplicity to test generation. However, there is a loss of generality.

Clock Faults and Multiple-Clock Circuits

An explicitly clocked flip-flop with asynchronous clear.



The logic in the shaded region in Figure is used for modeling the function of the flipflop. Faults inside this logic are usually not modeled.

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

In Phase 1 initialization vectors are generated. The purpose of these vectors is to bring flipflops in the circuit to known states irrespective of their starting state.

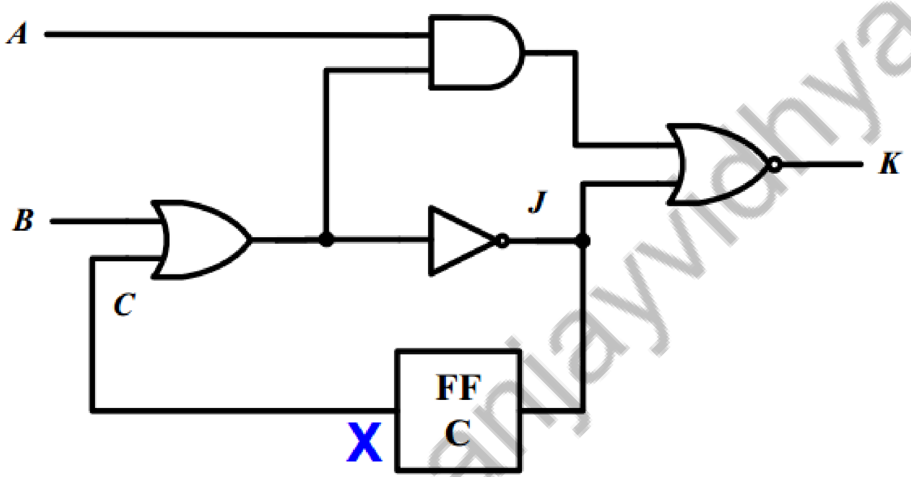
Phase 2 begins with vectors that are either supplied by the designer or generated in Phase 1. A fault list is generated in the conventional manner. For example, this list may contain all single stuck faults or a subset of such faults. These faults are simulated using a fault simulator. If the coverage is adequate, the test generation would stop. Otherwise, tests are generated with all undetected faults as targets. In the initial stages of test generation, the fault list is usually long and the objective of this phase is to generate tests by concurrently targeting all undetected faults.

Phase 2, if the fault coverage has not reached the required level then **Phase 3** is initiated. In this phase, test vectors are generated for single faults targeted one at a time.

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

Phase 1: Initialization. Here, the cost is defined simply as the number of flipflops that are in the unknown state. Initially, the cost may be equal to the number of flip-flops in the circuit. The goal in the initialization phase is to reduce this cost to 0. This cost function is derived only from good circuit simulation and is not related to the faulty circuit behavior. If the circuit is hard to initialize, one may relax the criterion for exiting to the next phase by allowing a small number of flip-flops, say 10%, to remain uninitialized.



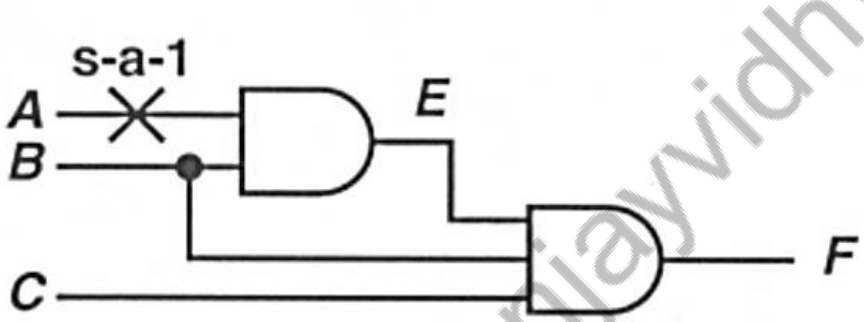
AB	cost = number of unknown FF
00	1
01	0
10	1

After simulation of a trial vector, the “trial cost” is computed as the number of flip-flops that are in the unknown state. If the trial cost is lower than the current cost, then the trial vector is saved. If the trial cost is zero, then the initialization phase is complete

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

Phase 2: Concurrent fault detection. The initialization vectors may already have detected some faults. Some others may have been activated but not detected. As a result, effects of active faults will be present at internal nodes of the circuit. a suitable cost function is the shortest distance to a primary output from any fault effect caused by the fault.

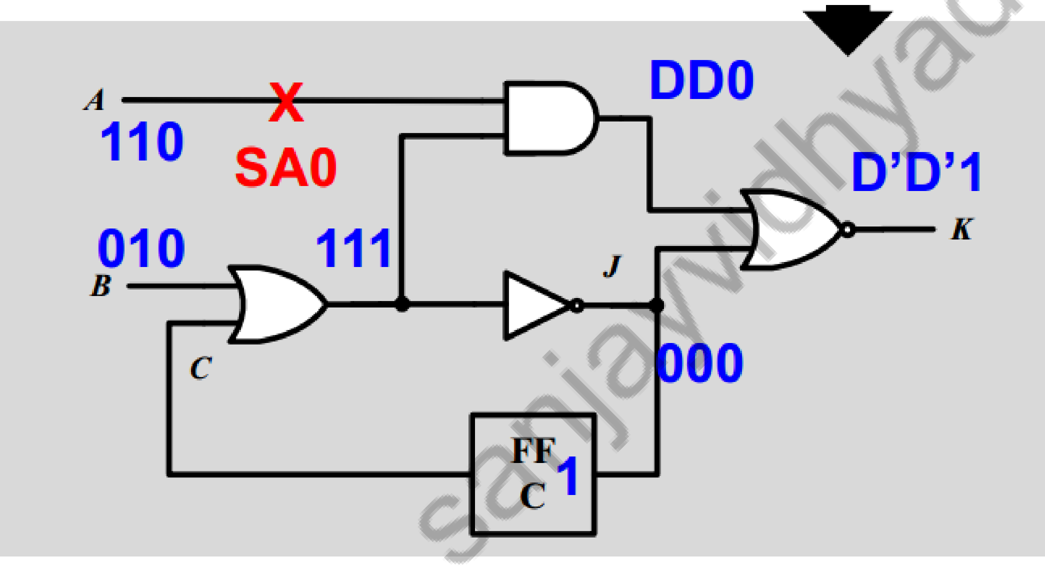


Trial vectors	0 0 0	1 0 0	0 1 0	0 1 1
Faulty signal	A	-	E	F
Cost	2	∞	1	0

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

Phase 2: Concurrent fault detection. The initialization vectors may already have detected some faults. Some others may have been activated but not detected. As a result, effects of active faults will be present at internal nodes of the circuit. a suitable cost function is the shortest distance to a primary output from any fault effect caused by the fault.



AB	cost = distance of D or D' to PO
10	0
11	0
00	∞

Video lectures by Professor James Chien-Mo Li

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

Phase 2: Concurrent fault detection. The initialization vectors may already have detected some faults. Some others may have been activated but not detected. As a result, effects of active faults will be present at internal nodes of the circuit. a suitable cost function is the shortest distance to a primary output from any fault effect caused by the fault.

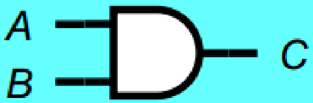
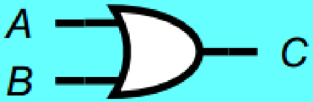
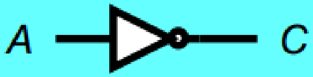
When there are several undetected faults, cost is computed for each fault *i* for some input vector and internal state. Similarly, the cost is obtained for a candidate trial vector. A comparison of and determines whether to accept the candidate vector or reject it. Since there can be several undetected faults, there are two *lists* of cost functions instead of just two numbers. The search for tests should be guided by a group of faults instead of a single target fault. One can devise simple rules to determine the acceptance of a vector. For example, if the combined cost of 10% of the lowest-cost undetected faults is found to decrease, then the new vector may be accepted.

Simulation-Based Sequential Circuit ATPG

CONTEST Algorithm

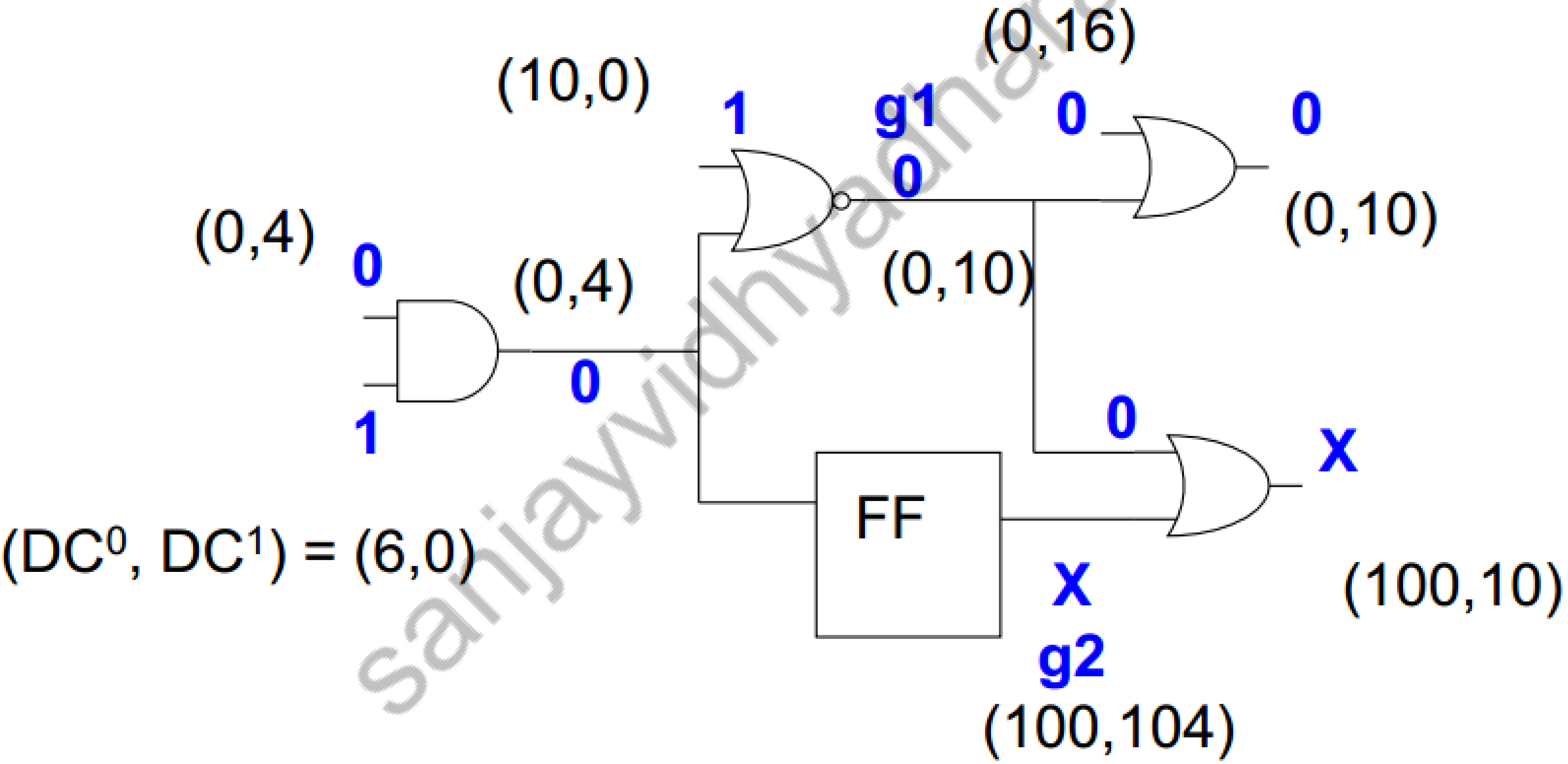
Phase 3: Phase 3: Single fault detection. The cost function in this phase is based on a SCOAP-like testability measure.

Dynamic Controllability

	$DC^0(C)$	$DC^1(C)$
	min[$DC^0(A)$,$DC^0(B)$] , if C=1 or x 0 , if C=0	$DC^1(A)$ + $DC^1(B)$, if C=0 or x 0 , if C=1
	$DC^0(A)$ + $DC^0(B)$, if C=1 or x 0 , if C=0	min[$DC^1(A)$,$DC^1(B)$] , if C=0 or x 0 , if A=1
	$DC^1(A)$, if C=1 or x 0 , if C=0	$DC^0(A)$, if C=0 or x 0 , if C=1
Primary inputs	1 , if C=1 or x 0 , if C=0	1 , if C=0 or x 0 , if C=1
C = FF(A)	$DC^0(A)+K$, if C=1 or x 0 , if C=0	$DC^1(A)+K^*$, if C=0 or x 0 , if C=1

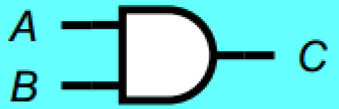
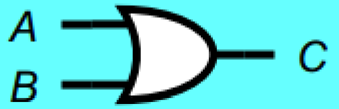
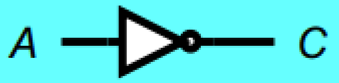
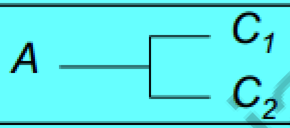
Simulation-Based Sequential Circuit ATPG

Dynamic Controllability

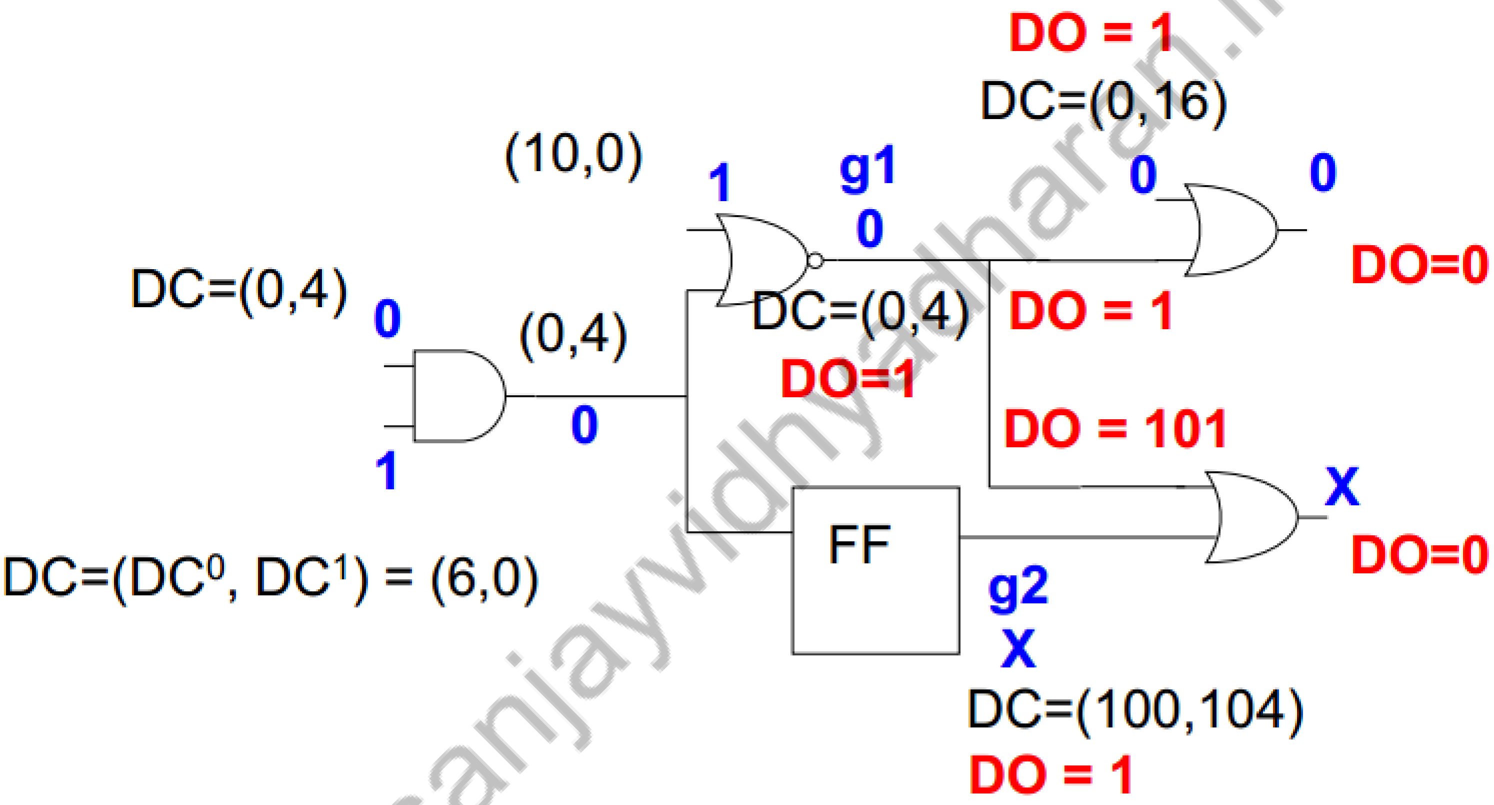


Simulation-Based Sequential Circuit ATPG

Dynamic Observability

	DO(*A*)
	$\mathrm{DO}(C) + \mathrm{DC}^1(B) + 1$
	$\mathrm{DO}(C) + \mathrm{DC}^0(B) + 1$
	$\mathrm{DO}(C) + 1$
	$\min[\mathrm{DO}(C_1),\mathrm{DO}(C_2)]$
Primary outputs	0

Simulation-Based Sequential Circuit ATPG



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