

DSD:-

Unit - III

Fault Generation:- Fault diagnosis and testing

Def:-

The task of determining whether a fault is present

(or) not is called fault detection.

\* The task of isolating the fault is fault location

\* Combined task of fault detection & fault location is called fault diagnosis.

\* Testing is nothing but technique adopted to diagnosis fault is testing.

Testing Process:-

circuit disruption



Fault modelling

Test generation



Fault simulation



Fault Coverage Evaluation



Test application

Fault modelling:- which consists of developing a fault  
diagram (a) List of possible faults.

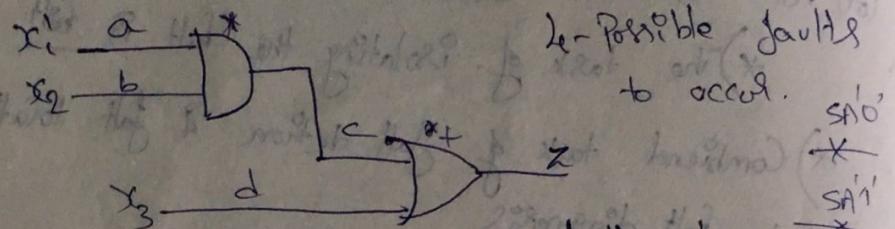
Test vector: In a combination which in the presence of a fault produces an o/p different from the fault free o/p known as test vector.

Fault simulation: Process of verifying the test set.

Fault Coverage: It refers to Percentage of faults that can be detected by test set.

Fault table method:

for And Gate - SA'0 Strong fault  
for OR Gate - SA'1 Strong "



$x_1$	$x_2$	$x_3$	fault free o/p	faulty o/p
0	0	0	0 0 0 a <sub>0</sub> b <sub>0</sub> c <sub>0</sub> d <sub>0</sub>	0 1 1 a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
0	0	1	1 1 1 a <sub>0</sub> b <sub>0</sub> c <sub>1</sub> d <sub>0</sub>	1 1 1 a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
0	1	0	0 0 0 a <sub>0</sub> b <sub>1</sub> c <sub>0</sub> d <sub>0</sub>	1 1 1 a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
0	0	1	1 1 1 a <sub>0</sub> b <sub>0</sub> c <sub>1</sub> d <sub>1</sub>	1 1 1 a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
1	0	0	0 0 0 a <sub>1</sub> b <sub>0</sub> c <sub>0</sub> d <sub>0</sub>	0 0 0 a <sub>0</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
1	0	1	1 1 1 a <sub>1</sub> b <sub>0</sub> c <sub>1</sub> d <sub>0</sub>	1 1 1 a <sub>0</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
1	1	0	0 0 0 a <sub>1</sub> b <sub>1</sub> c <sub>0</sub> d <sub>0</sub>	1 0 1 a <sub>0</sub> b <sub>0</sub> c <sub>1</sub> d <sub>1</sub>
1	1	1	1 1 1 a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>	1 1 1 a <sub>0</sub> b <sub>0</sub> c <sub>0</sub> d <sub>1</sub>

Fault Cover table:

$x_1$	$x_2$	$x_3$	$(a_0 b_0 c_0)$	$d_0$	$a_1$	$b_1$	$(c_1 d_1)$
*	0	0	0			X	X
*	0	0	1		X		
*	0	1	0	X			
*	0	1	1				
*	1	0	0				
*	1	0	1		X		
*	1	1	0			X	X
*	1	1	1	X			

\* There are 3-types of text vector

1) Essential text vector

2) Redundant "

3) Selective "

1) Essential:- If a fault is detected by one and only one text vector then text vector is an essential text vector by observing ~~the~~ Column with

{000, 010, 110}

2) Row wise  $\rightarrow$  Selective

{001, ~~000~~, 100, 101, 111}

2) Redundant :- no false

~~Selective~~ {011}

steering

$E = (0+1)1$	$S = 1$	$R = 0$
$E = (1+1)0$	$S = 1$	$R = 1$
$E = (3+4)0$	$S = 1$	$R = 1$
$E = (2+1)1$	$S = 1$	$R = 1$

Def:

## Review of Random Variables

A group of outcome in the random experiment  
is called sample space.

### Random variable:-

Real chance of getting an exact value of a  
sample space.  $(x, y) \leftarrow$

$\downarrow$   
 $(x, y)$

- ① Discrete Random variables  $\rightarrow$  finit no:-
- ② Continuous " "  $\rightarrow$  infinite

Sample Space = { HHH, HH $\bar{H}$ , H $\bar{H}$ H,  $\bar{H}$ HH, TH $\bar{H}$ , T $\bar{H}$ H, HT $\bar{H}$ , H $\bar{T}$ H }  
getting of Sample's  $\rightarrow$  3 2 2 1 1 0 1 2  
chances.

### ① Discrete R.V:-

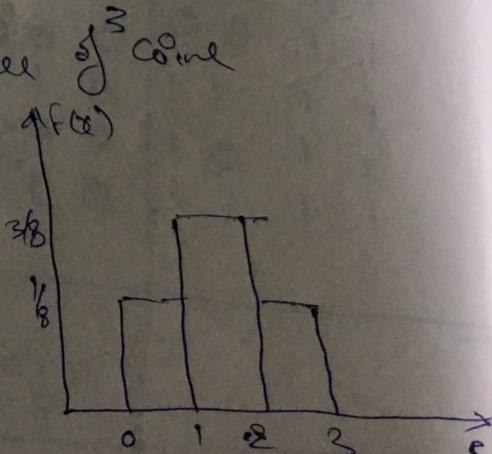
Eg:- Probability of getting 3 chances of 3 coin

$$A = \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \quad P(H=0) = \frac{1}{8}$$

$$H = \begin{matrix} 1 & 3 \\ 3 & 1 \end{matrix} \quad P(H=1) = \frac{3}{8}$$

$$H = \begin{matrix} 2 & 3 \\ 3 & 2 \end{matrix} \quad P(H=2) = \frac{3}{8}$$

$$H = \begin{matrix} 3 & 1 \\ 1 & 3 \end{matrix} \quad P(H=3) = \frac{1}{8}$$



### ② Continuous R.V:- $\rightarrow (0, \infty)$

## 2-D Effects in threads

softer threshold rule of

super hollow inserts

descenders of lowerell

Lab:- DSD

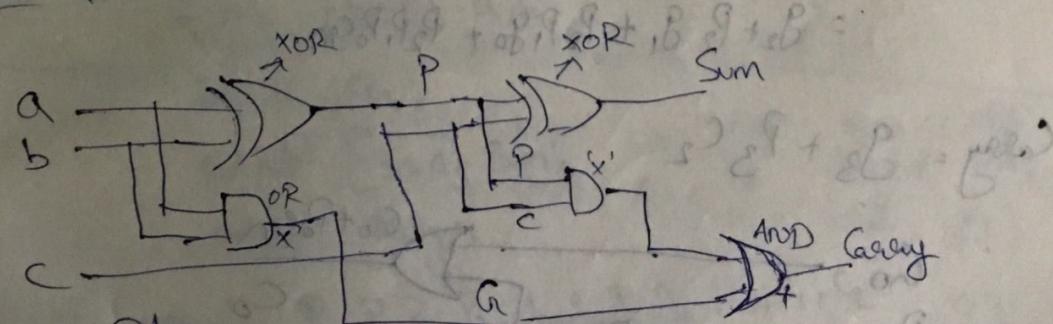
S-Program

CLA (Carry look a head adder)

a b c	Sum	Carry.	$C_i = 1$
0 0 0	0	0	0
0 0 1	1	0	0
0 1 0	1	1	1
0 1 1	0	0	0
1 0 0	1	0	1
1 0 1	0	1	1
1 1 0	0	1	1
1 1 1	1	1	1

if  $a=b=$  Carry  
 if  $a \neq b \Rightarrow C =$  Carry

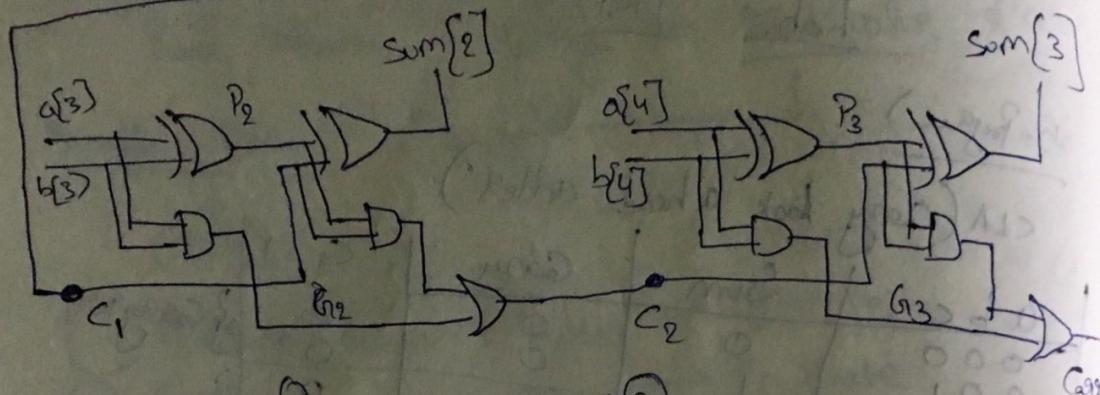
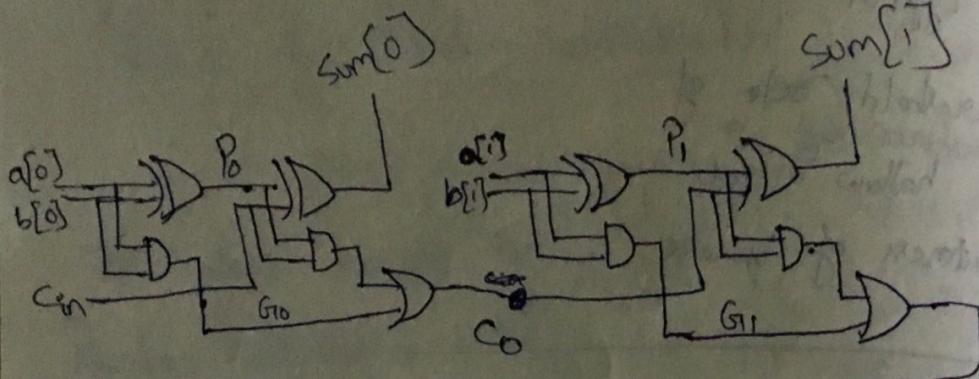
full adder logic diagram:



$$\text{Carry} = G + PC$$

$$\text{Sum} = P + C$$

4-bit



$$C_0 = g_0 + p_0 C_{in} \rightarrow ①; \quad C_1 = g_1 + p_1 C_0 \rightarrow ②; \quad C_2 = g_2 + p_2 C_1, \quad C_3 = g_3 + p_3 C_2,$$

$C_3 = \text{Substitute } ① \text{ in } ②$

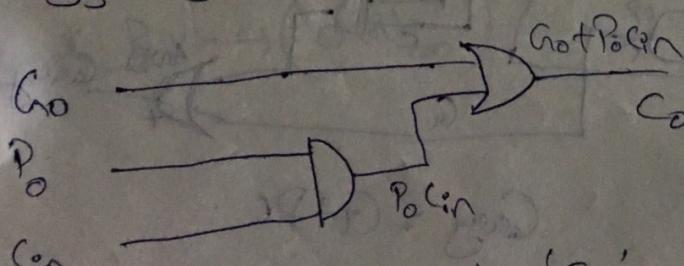
$$\begin{aligned} C_1 &= g_1 + p_1 (g_0 + p_0 C_{in}) \\ &= g_1 + p_1 g_0 + p_1 p_0 C_{in} \end{aligned}$$

$$C_2 = g_2 + p_2 C_1 \rightarrow ③$$

$$\text{now:- } C_2 = g_2 + p_2 (g_1 + p_1 g_0 + p_1 p_0 C_{in})$$

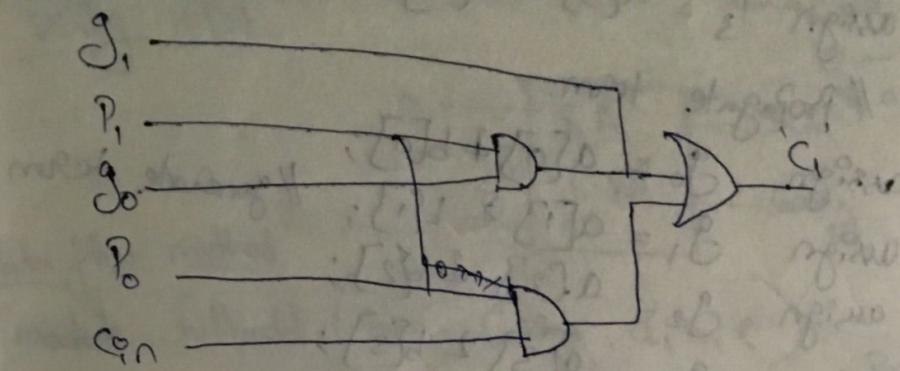
$$= g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_{in}$$

$$C_{out} = g_3 + p_3 C_2$$

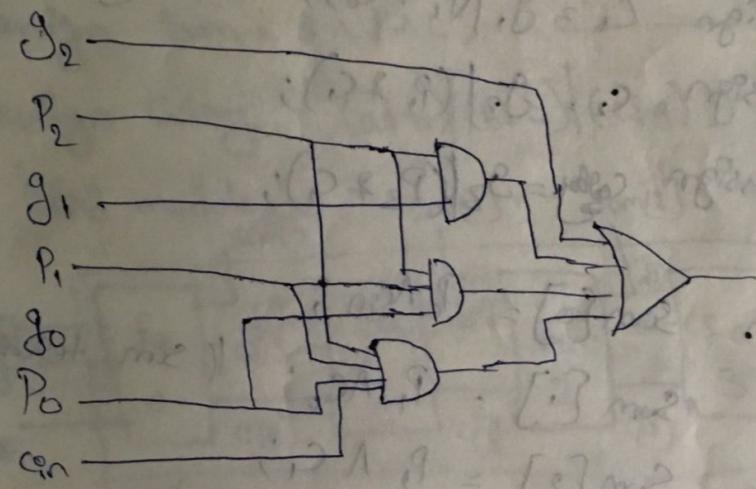


logic diagram of ' $C$ '

Logic diagram of  $S_1$



Logic diagram of  $S_2$



Coding for CLA module.  $CLA\_4(Sum, carry, a, b, c_{in})$

input  $[3:0] a, b;$

input  $c_{in};$

output  $[3:0] Sum;$

output  $carry;$

wire  $P_0, P_1, P_2, P_3, g_0, g_1, g_2, g_3, S_0, S_1, S_2$

(1)

// data flow modeling

assign  $P_0 = a[0] \wedge b[0];$

assign  $P_1 = a[1] \wedge b[1];$

assign  $P_2 = a[2] \wedge b[2];$

assign  $P_3 = a[3] \wedge b[3];$

// Propagate term.

assign  $g_0 = a[0] \wedge b[0];$

assign  $g_1 = a[1] \wedge b[1];$  // generate term

assign  $g_2 = a[2] \wedge b[2];$

assign  $g_3 = a[3] \wedge b[3];$

assign  $c_0 = g_0 | (P_0 \wedge C_{in});$

assign  $c_1 = g_1 | (P_1 \wedge C_0);$  // Carry term.

assign  $c_2 = g_2 | (P_2 \wedge C_1);$

assign  $C_{out} = g_3 | (P_3 \wedge C_2);$

assign  $Sum[0] = P_0 \wedge C_{in};$

$Sum[1] = P_1 \wedge C_0;$  // sum terms

$Sum[2] = P_2 \wedge C_1;$

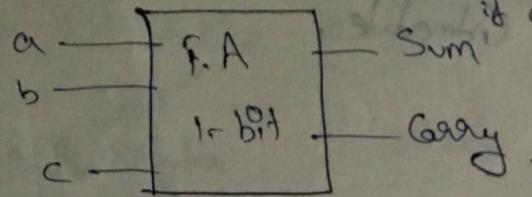
$Sum[3] = P_3 \wedge C_2;$

end module

OR AND NOR XOR XNOR NOT AND  
(|, &, ~, |~, |~, &, ~, ~), &  
(:)

# DSP Lab

VHDL HDL



Knock' 10

0 - 1

if  $a=0$  then  $a^1$

Sum

Carry

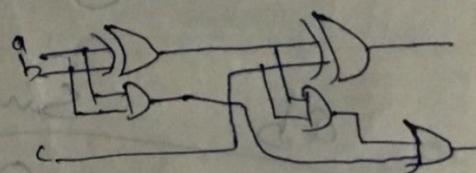
Data flow method

module fulladd (Sum, Carry, a, b, c);

Bottom up approach

input a, b, c;

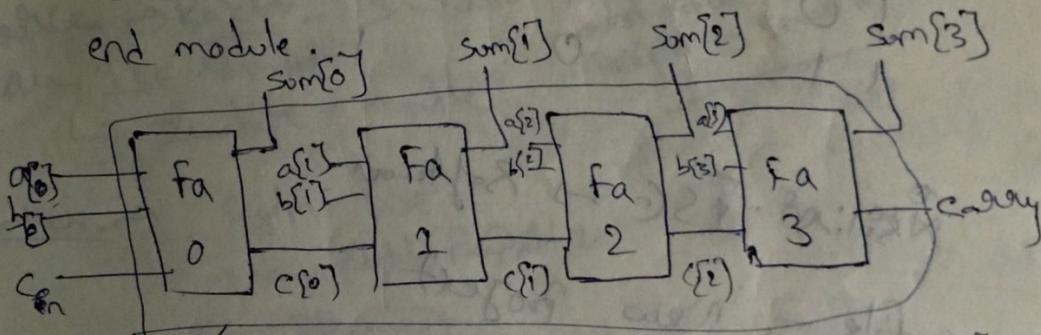
output Sum, Carry;



assign Sum = a  $\wedge$  b  $\wedge$  c;

assign Carry = (a  $\wedge$  b)  $\vee$  (b  $\wedge$  c)  $\vee$  (c  $\wedge$  a);

end module.



$a[3]a[2]a[1]a[0]$

$b[3]b[2]b[1]b[0]$

Carry

1

1111

module ~~fulladd~~ (Sum, Carry, a, b, c);

input [3:0] a, b;

input c;

output [3:0] sum;

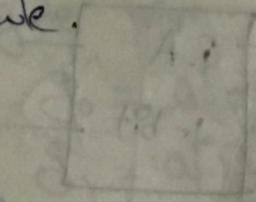
output carry;

wire (C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>);

fulladd fa0 (Sum[0], C<sub>0</sub>, a[0], b[0], c);

fulladd fa1 (Sum[1], C<sub>1</sub>, a[1], b[1], C<sub>0</sub>);

fulladd  $f_{a2}(\text{sum}[2], \text{cf}[2], a[2], b[2], c[2]);$   
 fulladd  $f_{a3}(\text{sum}[3], \text{cf}[3], a[3], b[3], c[3]);$   
 Counter



### Half adder

[www.xilinx.com/support/](http://www.xilinx.com/support/)  
[download/index.htm](http://www.xilinx.com/support/download/index.htm)

a	b	sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Xilinx ISE → Software  
 file - new project

Project name → Rajesh  
 next

Family Spartan 3E  
 IP Simulation → ISE Simulator  
 → VHDL / Verilog

Verilog language → Verilog  
 next → new source → Verilog module  
 next → file name - Rajesh  
 next → finish → yes

next → next

→ finish

$M \rightarrow$  Initiated - deactivate  
module Raftel .l

End module

Save ✓  
Source file  $\rightarrow$  Behavior Source

Raftel  $\rightarrow$  Text Syntax

Create new Source  $\rightarrow$  ✓  
Text bench wave form ✓

Raftel - TB  $\rightarrow$  next  
 $\rightarrow$  next  
 $\rightarrow$  finished.

$\rightarrow$  Combinational Sheet  $\rightarrow$  finished

Source  $\rightarrow$  Raftel - TB  $\rightarrow$  allot the PIP values to waveforms  
then use Stimulus  
 $\hookrightarrow$  simulate module

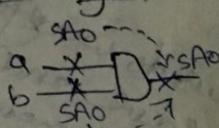
and then save and  
ocket to simulate module.

Editor  $\rightarrow$  Systl list  
 $\rightarrow$  new Systl Report ✓  
View Pts map

Fault Collapsing Ratio:-

It is nothing but a ratio of no. of Collapsing faults to the total no. of faults.

$$FCR = \frac{\text{no. of collapsing fault}}{\text{Total no. of faults}}$$

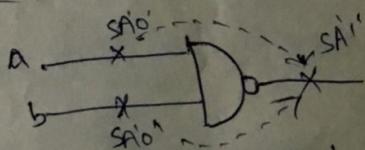
Fault Equivalence:-For Gates:-AND gates:-

T Table		out
a	b	
0	0	0
0	1	0
1	0	0
1	1	1

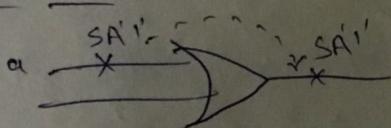
(multiply - AND)

(if AND Gate  
Stuck at '0' off)

- \* If stuck at '0' of op 'a' is zero then op is zero
- \* " " " op 'b' " " " op " "

For NAND:-

		out
a	b	
0	0	1
0	1	1
1	0	1
1	1	0

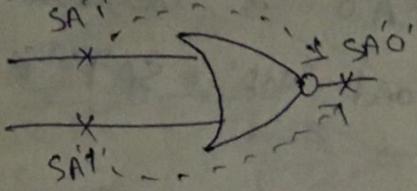
For 'OR':-

(Add - OR)

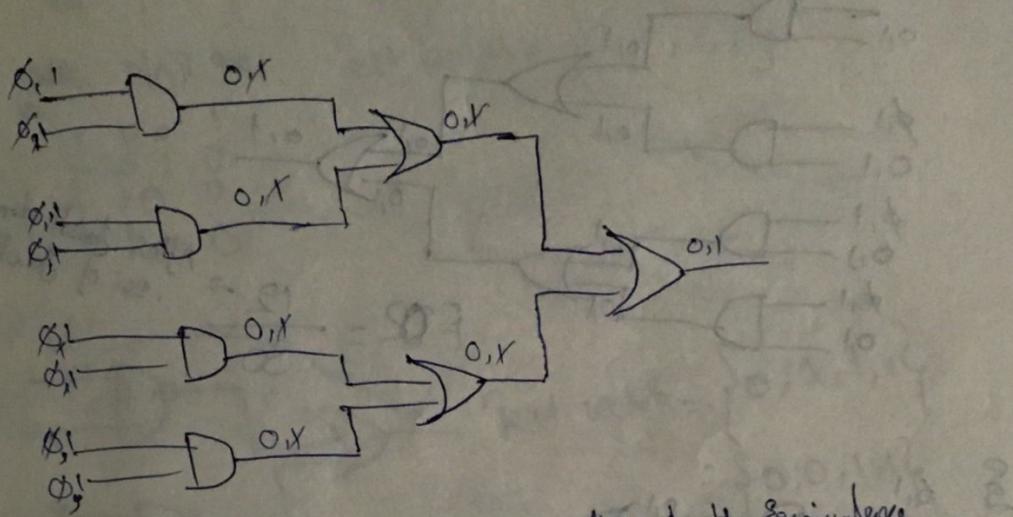
		out
a	b	
0	0	0
0	1	1
1	0	1
1	1	1

(For 'OR' Gate  
stuck at '1' see)

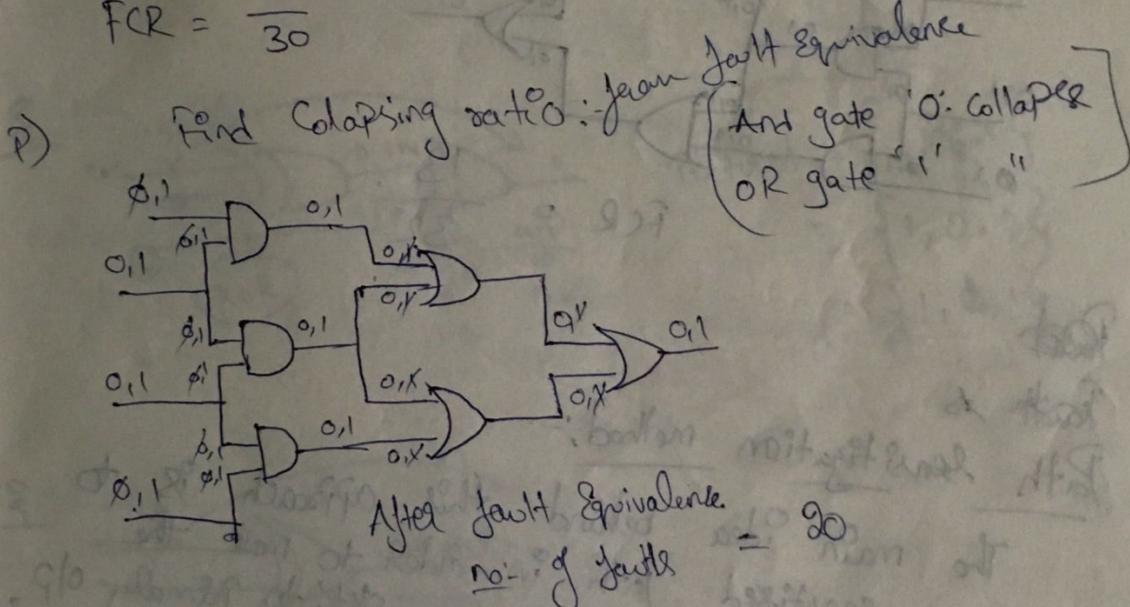
for 'NOR' :-



ab	out
00	1
01	0
10	0
11	0



$\rightarrow$  The no. of faults after fault equivalence  
 $FCR = \frac{16}{30}$



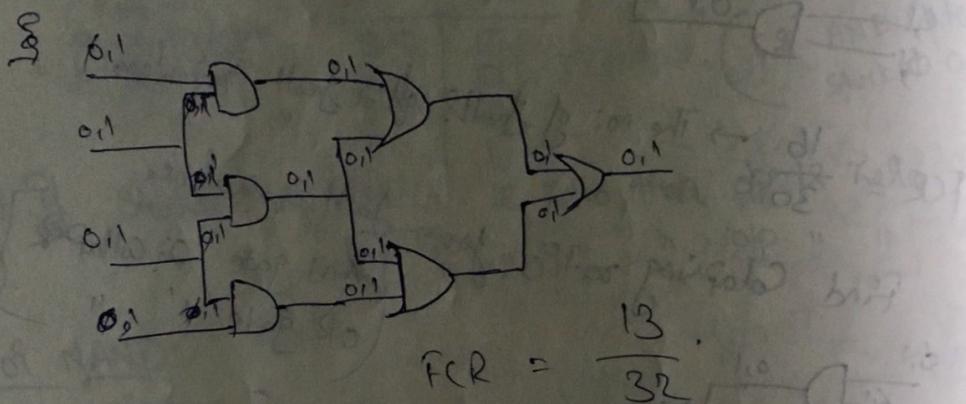
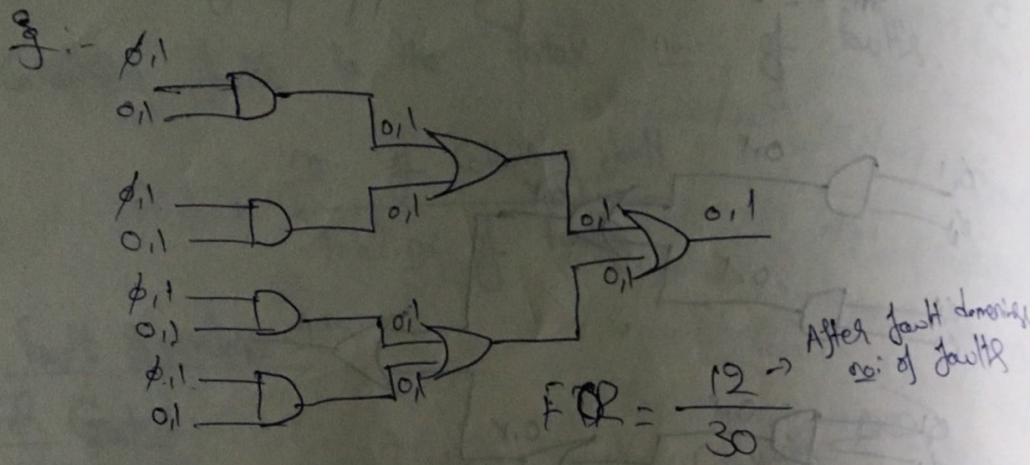
After fault Equivalence.  
no. of faults = 20

$$FCR = \frac{20}{32}$$

Fault Dominance:-

Let  $f \cup g$  be the set of all test's that detect's the fault 'g', the fault 'f' dominated the fault 'g' if  $f \cup g$  are functionally equivalent

- \* In fault dominance 'n' no. of OIP gates for AND, NAND Gate SA'0' is Considered.
- \* Similarly for OR and NOR SA'1' is Considered.



~~Path~~

~~fault~~  $\rightarrow$

Path sensitization method:-

The main idea behind this approach is to create a sensitized Path, in order to move the effect of faults and noise in the circuit to primary OIP.

\* The Path sensitization approach to test generation.

following :-

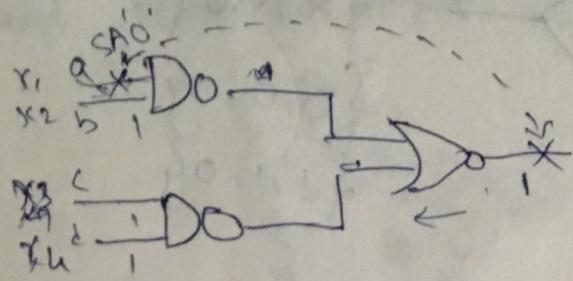
\* A Path is found material to OIP.

\* The OIP signal necessary to probe approach,

\* The Path from 'g' to the OIP is sensitized and

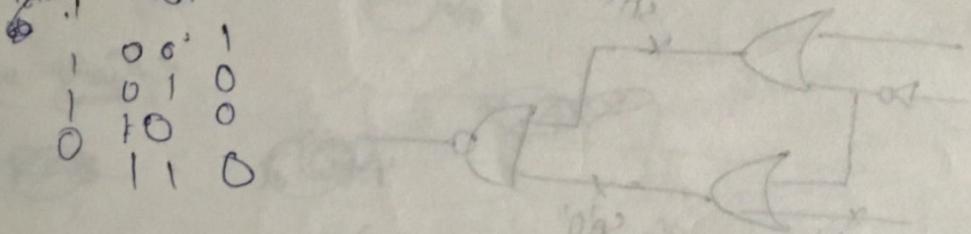
other o/p are determined by back propagation

Q:-

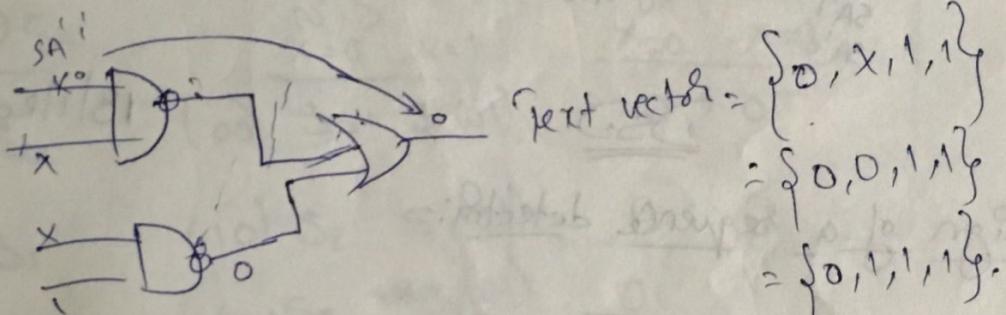


NOR gate. Test vector = {1, 1, 1, 1}

0	0	1
1	0	0
1	1	0
0	1	0
1	1	0

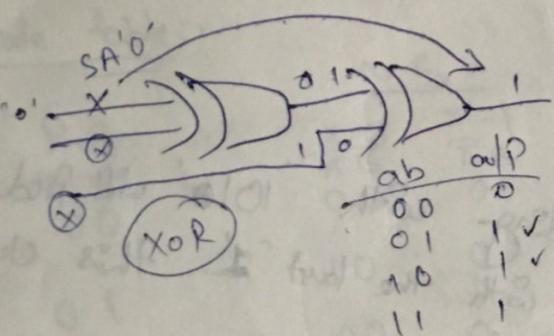


Q:-



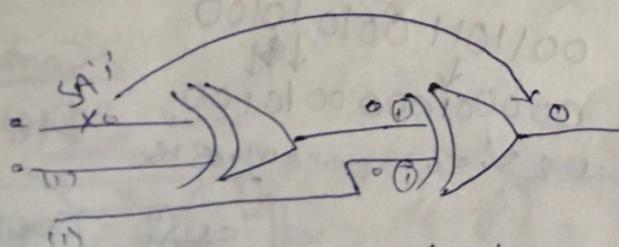
$$\begin{aligned} \text{Test vector} &= \{0, 1, 1, 1\} \\ &= \{0, 0, 1, 1\} \\ &= \{0, 1, 1, 1\}. \end{aligned}$$

Q

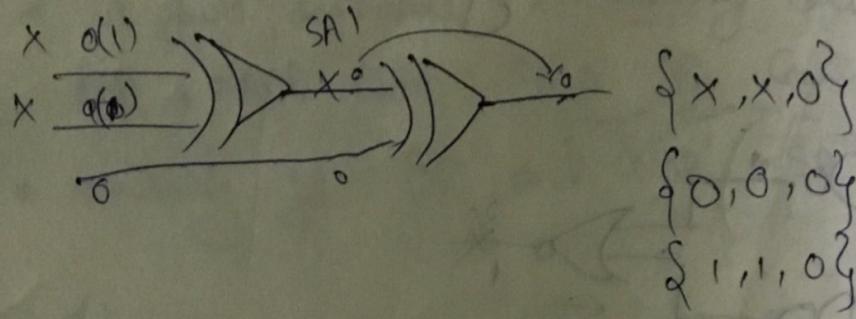


$$\begin{aligned} \text{Test vector} &= \{1, 0, 0\} \\ &= \{1, 0, 0\} \\ &= \{1, 0, 1\}. \end{aligned}$$

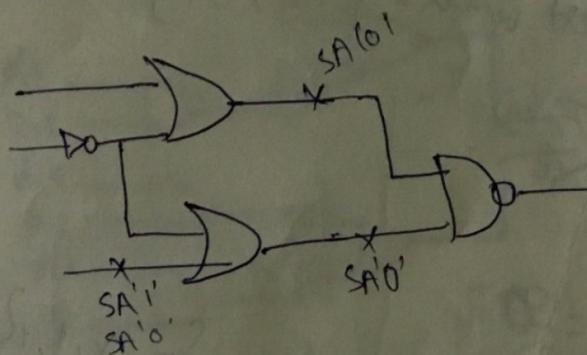
Q



ab	o/p
00	0
01	1
10	1
11	1



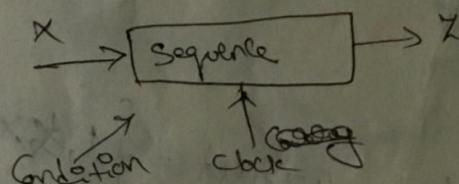
E..



DSD

18/1/2011

Design of a sequence detector:-



If any  $\oplus$  sequence ending with '1010' will produce ant o/p  $Z=1$  coincide with the last '1'. This clt does not reset when '1' occurs.

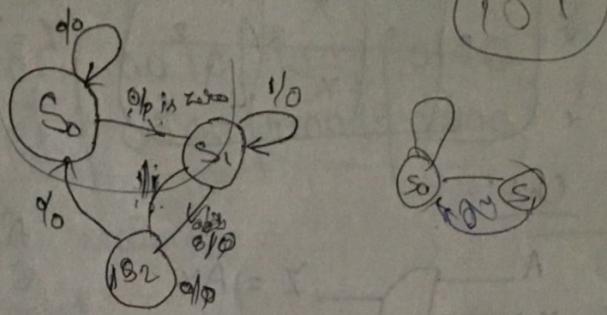
$\oplus$  Sequence  $X = 0011011\ 0010\ 10100$   
 $Z = 000001000\ 0010100$   
 time = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Mazy clt:-

The o/p of mazy sequen clt depends on Present state of flip flop.

State	Seq Regit
S <sub>0</sub>	reset
S <sub>1</sub>	01
S <sub>2</sub>	0
S <sub>3</sub>	1

Mealy Oct: The O/P of Mealy CKT depends on  
Present State of flip flop and IP sequence



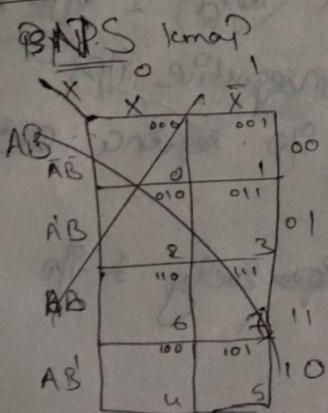
State graph:

<u>P.S</u>	<u>N.S</u> $x=0$ $x=1$	<u>O/P</u> $x=0 ; x=1$
(00) $S_0$	(00) $S_0$ $S_1(01)$	0    0
(01) $S_1$	(10) $S_2$ $S_1(01)$	0    0
(10) $S_2$	(00) $S_0$ <del><math>S_1(01)</math></del>	0    1

State table:

<u>P.S</u>	<u>N.S</u>		<u>O/P</u>	
	$x=0$	$x=1$	$x=0$	$x=1$
$\overline{AB}$	$\overline{A}\overline{B}$	$\overline{A}B$	0	0
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1
	$A^+$	$B^+$		

K-maps



$AB$	$\overline{AB}$	$A\overline{B}$	$\overline{A}B$
00	0	1	X
01	0	0	X
11	X	X	X
10	X	X	X

$\overline{XB}$  for  $A^+$  term

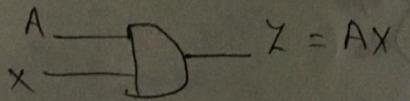
$A\overline{B}$	$\overline{A}B$	$AB$	$\overline{AB}$
X	0	0	X
X	1	1	X
X	1	X	1
X	1	1	1

Karnaugh Map for  $Z = XA$

AB		00	01	10	11
0	X	0	1	X	1
1	X	0	1	X	1
		0	5	7	6

$$Z = XA$$

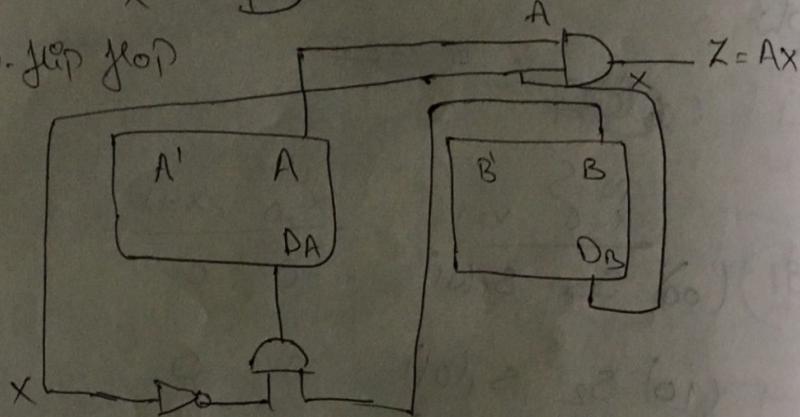
Sequence Ckt:-



$$A^+ = RA$$

$$B^+ = RB$$

D-Flip Flop



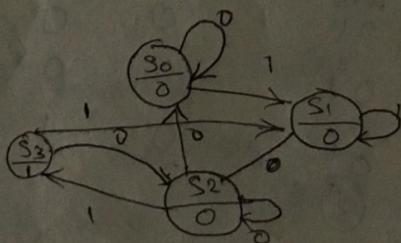
Moorre Ckt:-

Condition

101

$$X = 0011011001010100$$

$$Z = 00000 \downarrow 00000 \downarrow 0 \downarrow 00$$



A sequential ckt has 1 o/p 'X' and 1 o/p 'Z'  
the ckt checks groups of 4-consecutive PPs  
and produces an o/p  $Z=1$  if the group sequence 0101  
occurs. ~~if 1001 occurs.~~

The ckt resets after ~~every~~ every 4-o/p

find the mlay state graph

$$X = 0101 \quad 0010 \quad 1001 \quad 0100$$

Condition 0101 (a) 1001

the CLK needs for every 4-bits

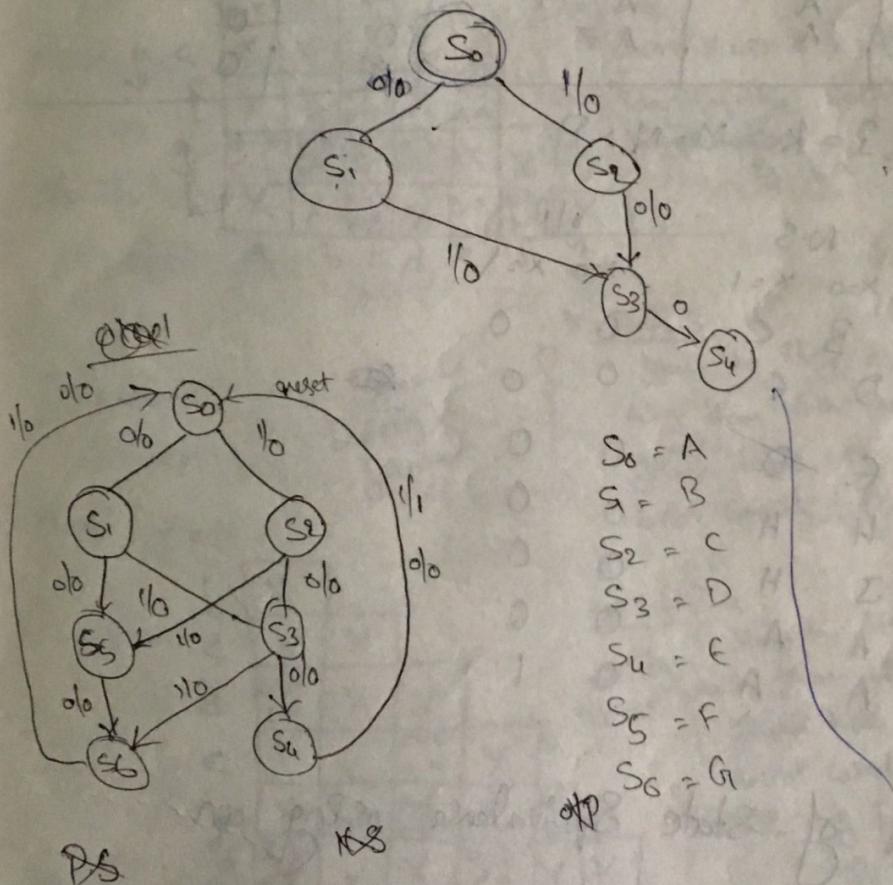
$$\begin{array}{c} \text{sd} \\ x = 0101 \\ \downarrow \\ 2 = 0001 \end{array} \quad \left| \begin{array}{c} 0010 \\ \downarrow \\ 0000 \end{array} \right. \quad \left| \begin{array}{c} 1001 \\ \downarrow \\ 0001 \end{array} \right. \quad \left| \begin{array}{c} 0100 \\ \downarrow \\ 0000 \end{array} \right.$$

State Q/P sequence received.

$S_0$	-	reset
$S_1$	-	0
$S_2$	-	1
$S_3$	$\Rightarrow$	01 (s) 10
$S_4$	$\Leftarrow$	010 (s) 100

metaling CLK

$S_0 \rightarrow$   
 $S_1 \rightarrow$



$S_0 = A$   
 $S_1 = B$   
 $S_2 = C$   
 $S_3 = D$   
 $S_4 = E$   
 $S_5 = F$   
 $S_6 = G$

Reset	N.S			O.P	
	x=0	x=1	x=0	x=1	
0	A	B	D	C	O
1	C	F		G	O
00	D		H	I (H)	O
01	E		J	K (H)	O
10	F		L (J)	M (H)	O
11	G		N (H)	P (H)	O
000	H		A	A	O
001			A	A	O
010			A	A	O
011			A	A	-1
100			A	A	0
101			A	A	-1
110			A	A	-1
111	P		A	A	O

$$J=2 ; H=I=K=M=N>P.$$

P.S	N.S		O.P	
	x=0	x=1	x=0	x=1
A	B	C	O	O
B	D	E	O	O
C	E	D	O	O
D	H	H	O	O
E	J	H	O	O
H	A	A	O	O
J	A	A	O	1

Determination of State Equivalence using an implication table:-

Q:-

S.	Present State	Next State		Present Opt
		$x=0$	$x=1$	
	a	d	c	0
	b	f	h	0
	c	e	d	1
-	d	a	e	0
	e	c	a	1
	f	f	b	1
	g	b	h	0
	h	c	g	1

Implementation table:

	b	$x=0$		$x=1$	
		$x=0$	$x=1$	$x=0$	$x=1$
	c	X	X		
	d	a	a	c	c
	e	X	X	ce	da
	f	X	X	ef	fb
	g	ab	fb	X	ab
	h	X	X	ec	ga
		a	b	c	d
		e	f	g	

Step-1

\* When ab opt are same leave it,  
when ac opt are different of if 'X'

Leave same as for every block

Step-2

\* from  $x=0$  &  $x=1$  compare and  
identify the blocks entries by table  
implementation.

Eg. ba  $\rightarrow$  df

Step-3

From the Step-2 entered blocks  
vary them with Present State in  
given table. If the opt are same  
don't cancel; if not same cancel.

b	X					
c	X	X				
d	da	X	X			
e	c-e					
f	X	X	ef	X	fa	
g	X	X	ab	X	ab	X
h	X	X	ec	X	ga	fg
	a	b	c	d	e	f
						g

Step-4

From the Step-3' entered blocks  
abcdegf vary from with Present State in  
given table. If the opt are same  
don't cancel; if not same cancel  
here:- If F-c is same and  
bg is verified from  $x=0$  &  
 $x=1$   
& different then it is cancelled.

$\therefore a=d$ ;  $c=e$ ; ~~etc~~

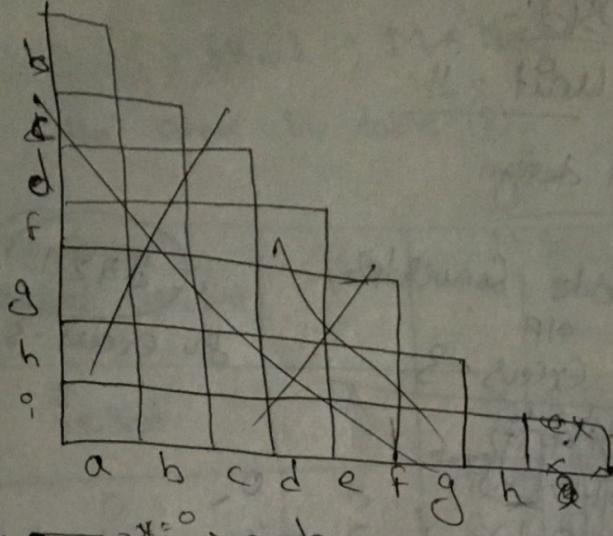
## Redundant table:-

Present State	next state		d.p
	x=0	x=1	
a	a	c	0
b	f	c	0
c	c	a	1
f	<del>f</del>	b	1
g	b	c	0

→ Steps for Construction of implementation table:-

- 1) Construct a chart which contains a square for pair of states.
- 2) Compare each pair of rows in the ~~step~~ state if the d.p's associated with states 'i' and 'j' are different place and "X". in square box  $i-j$  to that  $i \neq j$  if the d.p are same place the implied pair in square  $i-j$
- 3) Go through the table square by square if square contains the implied pair m-n, and square m-n contains an 'X' the  $i-j$  and an 'X' mark & be placed in square  $i-j$

Ex:-	P.S.	N.S		d.p
		x=0	x=1	
a	e	e	1	
b	c	e	1	
c	g	h	0	
d	h	a	1	
e	g	f	0	
f	e	g	0	
g	h	b	1	
h	c	d	0	
o	f	b	1	



b	$e-c \Rightarrow x=0$	$e-e \Rightarrow x=1 \Rightarrow a=b$					
c	X X						
d	<del>a</del> <del>b</del> <del>c</del> <del>d</del> <del>e</del> <del>f</del> <del>g</del> <del>h</del>	X					
e	X X	$i-i-f$					
f	X X	<del>a</del> <del>b</del> <del>c</del> <del>d</del> <del>e</del> <del>f</del> <del>g</del> <del>h</del>					
g	<del>a</del> <del>b</del> <del>c</del> <del>d</del> <del>e</del> <del>f</del> <del>g</del> <del>h</del>	X X					
h	X X	<del>a</del> <del>b</del> <del>c</del> <del>d</del> <del>e</del> <del>f</del> <del>g</del> <del>h</del>					
i	<del>a</del> <del>b</del> <del>c</del> <del>d</del> <del>e</del> <del>f</del> <del>g</del> <del>h</del>	X X					
a	b	c	d	e	f	g	h

$\Rightarrow$  Condition

$a=b$  only when  $e=c; e=e$

$e=c \Rightarrow i=i$  ( $e=c \neq i$ )

$\Rightarrow$  If  $x=0$  is verified then OLP are different then it is cancelled.

$\Rightarrow$  If  $x=0$  is verified then OLP are different then it is cancelled.

$\Rightarrow a=b; c=e; f=h; d=g; i$

Redundant table:-

P.S	N.S		OLP
	$x=0$	$x=1$	
a	c	c	1
b	d	f	0
c	f	a	1
d	f	d	0
e	c	a	1
f	c	a	1
g			0
h			1
i			0

DSD:-

Unit - II

Sequential Circ design.

Design eg - Code Converter

	IP BCD	to Excess-3
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

8421

for Excess-3 add 3 to  
number

time	IP received (LSB first)	P.S.	NS $x_2   x=1$	Postant $x=0   x=1$
$t_0$	select	A	-B	C
$t_1$	0	B	D	E
$t_2$	00 01 10 11	D G F G	H I J K L M N P	I O P Q R S T U V W X Y Z
$t_3$	000 001 010 011 100 101 110 111	H I J K L M N P	A A A A A A A A	O O O O O O O O
		H J M	A A A	I -

Reduction

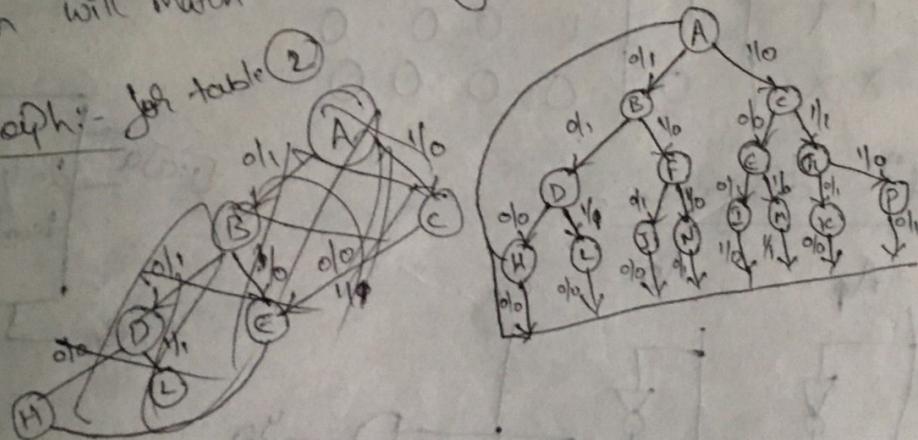
$H = 3$ ;  $JK = 2$ ;  $M = N = P$

Substitute the above in table ②.

time	SIP Sequence Received (MSB first)	PS	N.S		Present	
			$x_0$	$x_1$	$x=0$	$x=1$
$t_0$	reset	A	B	C	I	O
	0	B C	D E	(F) E	I O	O I
$t_1$	1	D E	H I	(J) H	O I	I O
$t_2$	00 01 10	(F) X	G H	(J) H X	I X	X X
$t_3$	000 001 010 011 100 101 110 111	H J K L M N P	A A A A	O O -	O O -	I -

Note: A dash will match with any state ( $s_i$ ) any o/p value

State graph - for table ②



non " "  
Feedback tracing faults:-

1st M.D Questions

DSD :-

- 1(a) Explain the basic design blocks of ASM chart  
(b) Discuss in detail about reduction of state diagrams with examples?

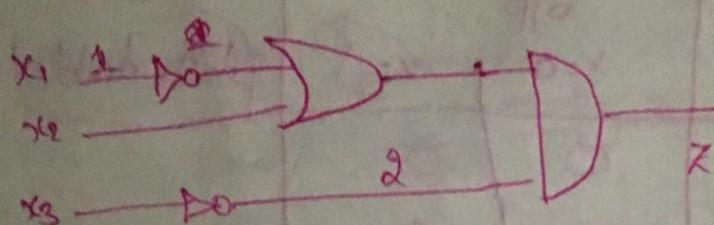
2(a) using an example describe how an sequential circuit is designed using CPLDs?

⑤ Describe the steps involved in D-algorithm?

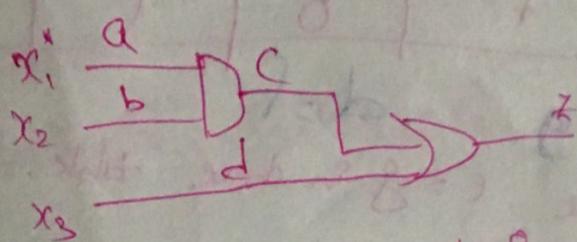
3) using the Path sensitization method and Boolean differentiation find the test vector for SR flip flop on I/P line '1' and SR 2 fault on internal line

(b) Apply Karnaugh  
the given logic

1(b):-



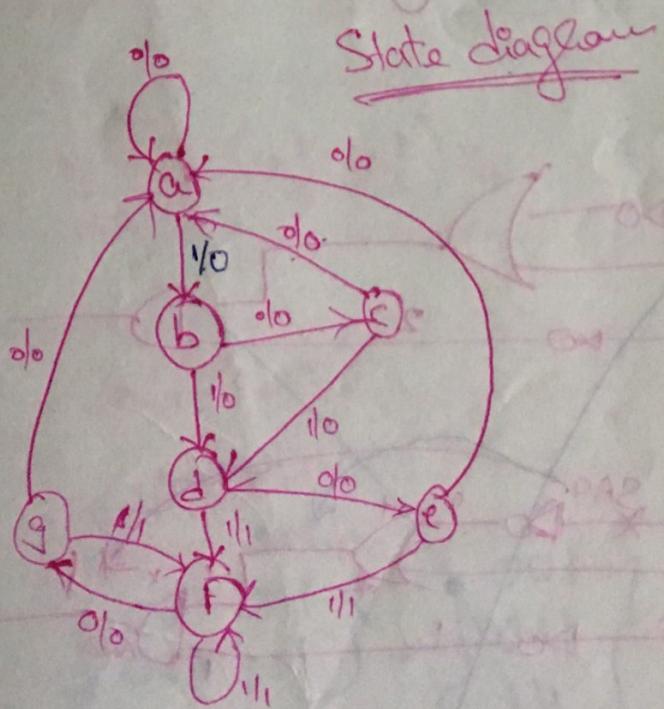
3(b) Draw the table giving the set of all possible single stuck at faults and the faulty and fault free responses and also construct the fault cover table for the circuit shown below.



4(a) what are different faults found in combination of circuits? how can they be categorized?

Karnaugh's algorithm to detect different faults in boolean function?  $f = x_1x_2 + x_1x_3x_4 + x_2x_4$

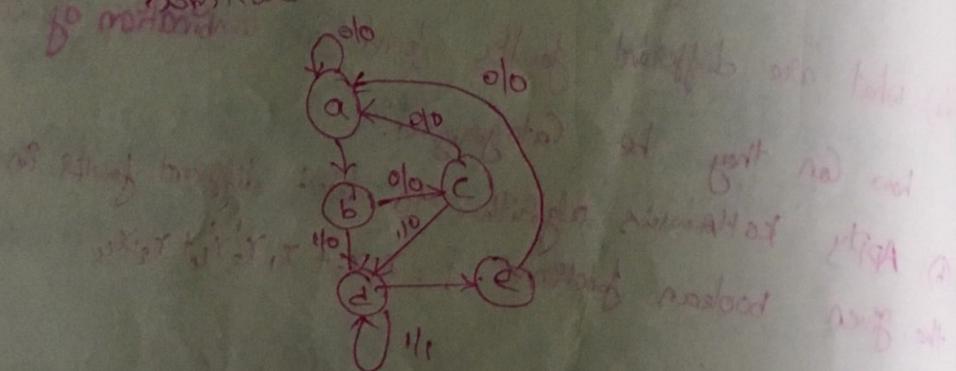
7(b):-



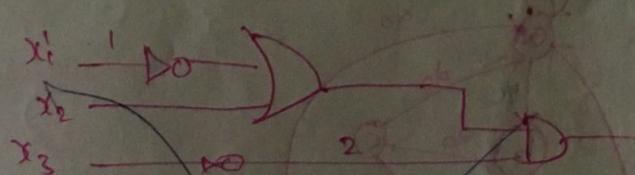
PS	NS		O/P		state table
	$x=0$	$x=1$	$x=0$	$x=1$	
a	a	b	0	0	
b	c	d	0	0	
c	a	d	0	0	
d	e	f	0	1	
e	a	f	0	1	
f	g	f	0	1	
g	a	f	0	1	

$d = f$   
 $c = g$        $e = g$  in above table.

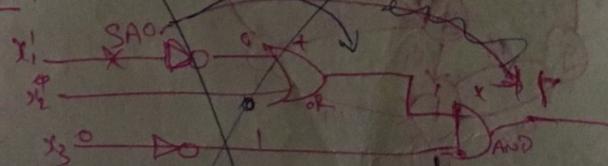
Substitute  
for more



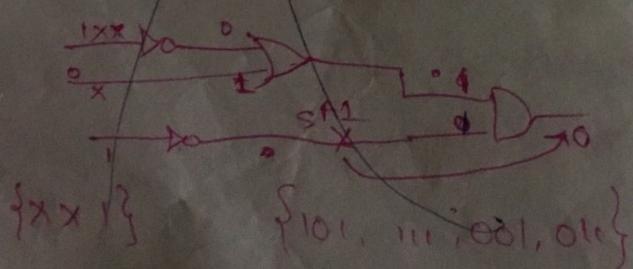
3(a) (a)

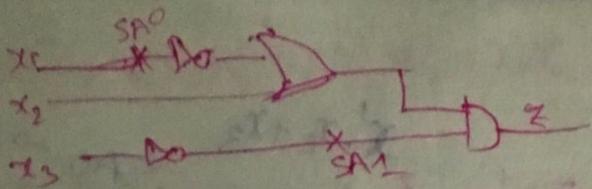


Line 1 :-



now :- Line 2  $\rightarrow$  SA1 fault





Boolean difference

$$f(x) = f_i(0) \oplus f_i(1) = x_1' x_3 + x_2 x_3$$

$$\text{SAI}^0: - x_i \cdot \frac{df(x)}{dx_i} \quad \text{SAI}^1: x_i' \cdot \frac{df(x)}{dx_i}$$

at IP  $x_i$

when  $x_i = 0$

$$f_i(0) = x_3' + x_2 \cdot x_3' = x_3'(1+x_2) = x_3'$$

$x_i = 1$

$$f_i(1) = x_2 \cdot x_3'$$

$$f(x) = f_i(0) \oplus f_i(1)$$

$$\frac{df(x)}{dx_i} = x_3' \oplus x_2 x_3'$$

$$\frac{df(x)}{dx_i} = x_3'(x_2 \cdot x_3')' + x_3 \cdot x_2 \cdot x_3'$$

$$\Rightarrow x_3'(x_2' + x_3')$$

$$\Rightarrow x_2' \cdot x_3'$$

$$\text{SAI}^0: - x_i \cdot \frac{df(x)}{dx_i} = (x_2' \cdot x_3') x_i$$

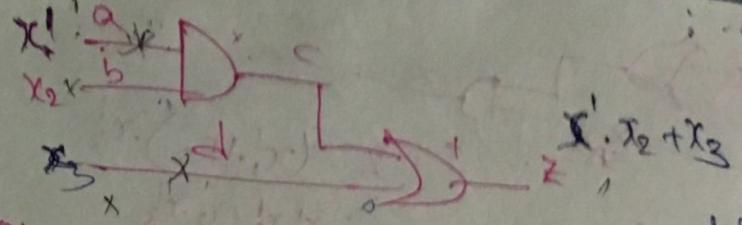
$$\text{SAI}^1: x_i \cdot \frac{df(x)}{dx_i} = (x_3')' (x_1 + x_2)$$

$$= x_1' x_3 + x_2 x_3$$

$$\{x_1, x_2\}$$

$$\begin{cases} 001 & 011 \\ 011 & 111 \end{cases}$$

$$\begin{cases} 001 \\ 011 \\ 111 \end{cases}$$



$X_1$	$X_2$	$X_3$	fault free O/Ps generated	faulty responses	$a_0, b_0, c_0, d_0$	$a_1, b_1, c_1, d_1$
0	0	0	0	000	0 0 0 0	1 1 1 1
0	0	1	1	001	1 1 0 0	0 0 1 1
0	1	0	01	010	0 0 0 0	1 1 1 1
0	1	1	1	011	1 1 1 1	0 0 0 0
1	0	0	0	100	0 0 0 0	1 1 1 1
1	0	1	1	101	1 1 1 1	0 0 0 0
1	1	0	0	110	0 0 0 0	1 1 1 1
1	1	1	1	111	1 1 1 0	1 1 1 1

Fault Cover table:-

Test vector states			Faults			
$X_1$	$X_2$	$X_3$	$(a_0, b_0, c_0)$	$d_0$	$a_1, b_1$	$(c_1, d_1)$
0	0	0			x	x ✓
0	0	1			x	
0	1	0			x	
0	1	1				
1	0	0				x ✓
1	0	1			x	
1	1	0			x	x ✓
1	1	1			x	

redundant  $\rightarrow$  no fault  $\{011\}$

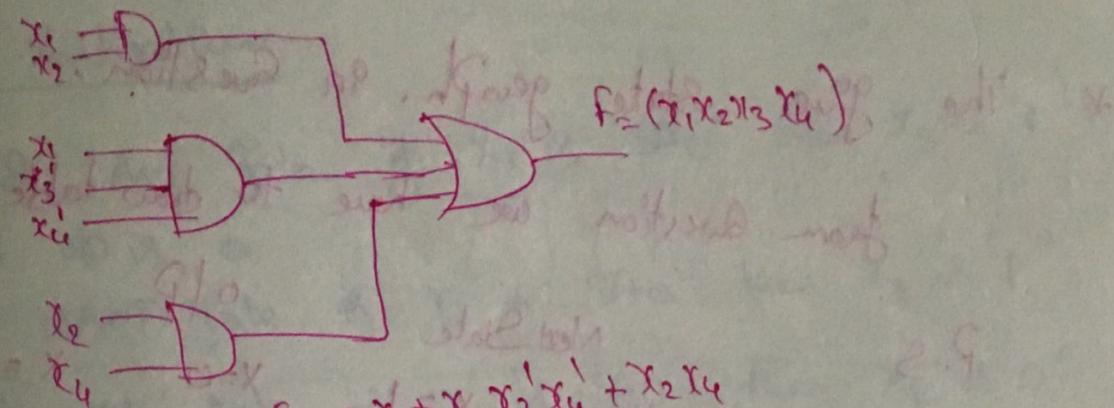
Selective  $\rightarrow \{001, 100, 101, 111\}$

Essential  $\rightarrow \{000, 010, 110\}$  Single column one faults

Based on the fault free O/P the fault response is determined as if O/P is '0' then the response fault '1' is denoted as 'x' (don't care)

'1' then " " " " " " " " " "

$$f(n) \therefore f = x_1x_2 + x_1x_3'x_4' + x_2x_4$$



$$f = x_1x_2 + x_1x_3'x_4' + x_2x_4$$

$$(1xx, 1x00, x1x1)$$

lives of blocks  
 (12) 11010  
 (13) 11011  
 (14) 11110  
 (15) 11111

$$\frac{1000(8)}{1100(12)} \frac{0101(5)}{0111(7)}$$

$$\frac{101(13)}{111(15)}$$

a-test:-

$$f(x) = \{14, 8, 5(8), 7\}$$

b-test:- K-map.

$x_1x_2$	00	01	11	10
$x_3x_4$	00	01	11	10
00	0	1	1	0
01	4	15	1	6
11	12	13	15	14
10	1	2	9	11

$$* f = \bar{a}bc + b\bar{c}c$$

$$= \bar{a}cx + b\bar{c}x + a\bar{c}x$$

$$= \bar{a}1x + b1x + a1x$$

which is state false for all states except (0,0,0)

## Reduction state table:-

- \* The given state graph. In question.  
From question we have to draw table

P.S	Next State		O/P	
	$x=0$	$x=1$	$x=0$	$x=1$

- \* In table N.S & O/P states should be equal
- \* In table N.S & O/P states should be equal  
then they can be replaced in next table.
- \* In the last reduction table we have to draw the state diagram graph.

Procedure:-

Mode Ckt.:-

single output

- 1) Determine the O/P & O/P of flip flops.
- 2) Then find K-map for flip flops of DA & DB  
and also for outputs
- 3) Then draw table Present, next state, O/P  
 $x=0 \quad x=1$
- 4) Then again take K-map for  $x=0 A^+ \& x=1 A^+$  and  
again  $x=1 A^+ \& x=1 B^+$   
 $DA$
- 5) Then draw state table.. and plot states at S<sub>0, S<sub>1</sub>, S<sub>2</sub></sub>
- 6) Then draw state graph for above table.

nearby :-  $\alpha p$  is as  $x=0$  &  $x=1$

## Path Segmentation:-

For And gate if  $x_1$  is SAO' then  $x_2$  is 1)

and old is o

op is 0  
NAND gate by  $x_1$  is SA0 then  $x_2$  is 1

For 'OR' gate if  $x_1$  is '0' then  $x_2$  is '0'  
 NOR " "  $x_1$  is " "  $x_2$  is '0'

for And sake  $x_1$  is SA'T' then  $x_2$  is 'I'

NAND  $x_1 \times x_2 = 1$  (1)

OR " "

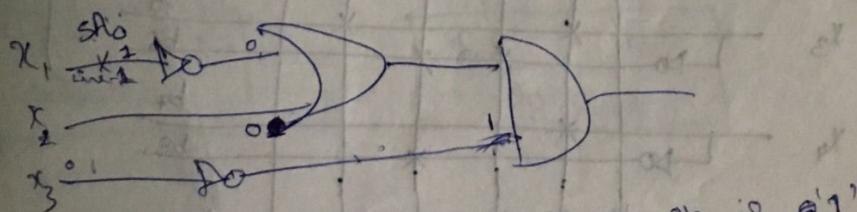
NOR " " SA1 " "

If 'Sao' then 'X' if 'O' then 'O'

$$\text{AND (NAND)} = \textcircled{6}^*$$

$$OR, (NOR) = 0$$

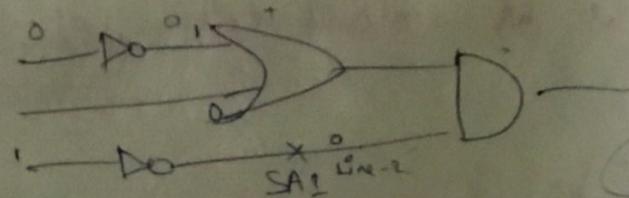
386



For Path Sensitization if SA<sub>0</sub> is there then  $\eta_0$  is 1

where the AND Gate is '1' and for OR gate is '0' at Line 1

Test vector is  $\begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$



$$\left\{ \begin{matrix} x \\ 0 \\ 1 \end{matrix} \right\} \Rightarrow \left\{ \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \right\} \left\{ \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \right\}$$

## DSD:- Unit - 7

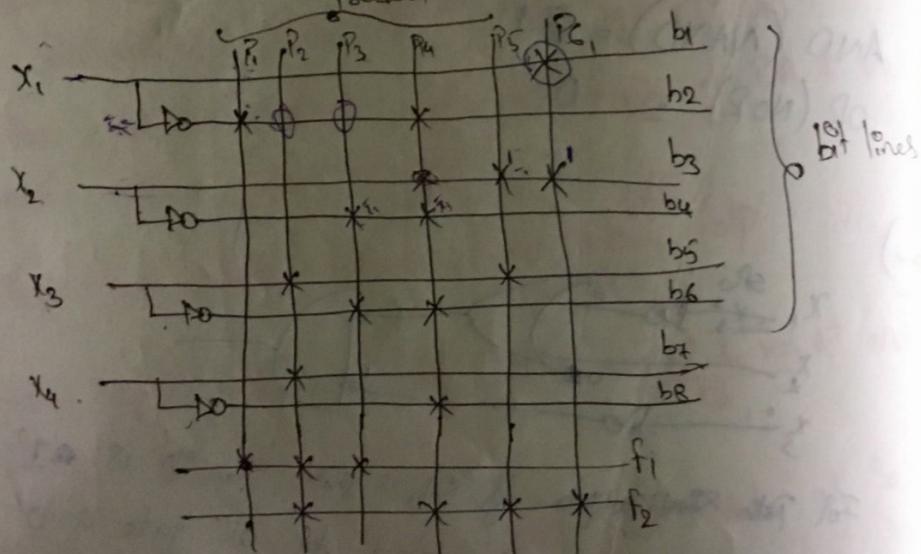
- 1) Fault models  $\rightarrow 4$
- 2) Test generation  $\rightarrow 2$
- 3) Test PLA Design  $\rightarrow 5$

### 1) Fault models:- . Introduction

$$f_1 = \overline{x_1} + \overline{x_3}x_4 + \overline{x_2}x_3$$

$$f_2 = x_3x_4 + \overline{x_1}x_2\overline{x_3}x_4 + \overline{x_2}x_3 + x_1x_2$$

Product terms



PLA can be described by matrix.

$P = (A, O)$  where 'A' is an  $m \times N$  matrix representing the AND array and 'O' is  $m \times k$  matrix representing the ~~values~~ OR array

	$x_1$	$x_2$	$x_3$	$x_4$	$\dots$	$f_1, f_2$
$P_1$	0	X	X	X		1 0
$P_2$	X	X	1	1		1 1
$P_3$	X	0	0	X		1 0
$P_4$	0	0	0	0		0 1
$P_5$	X	1	1	X		0 1
$P_6$	1	1	X	X		0 1

AND connection :-

$$1 - X^0 \\ 0 \rightarrow \overline{X}^P \checkmark$$

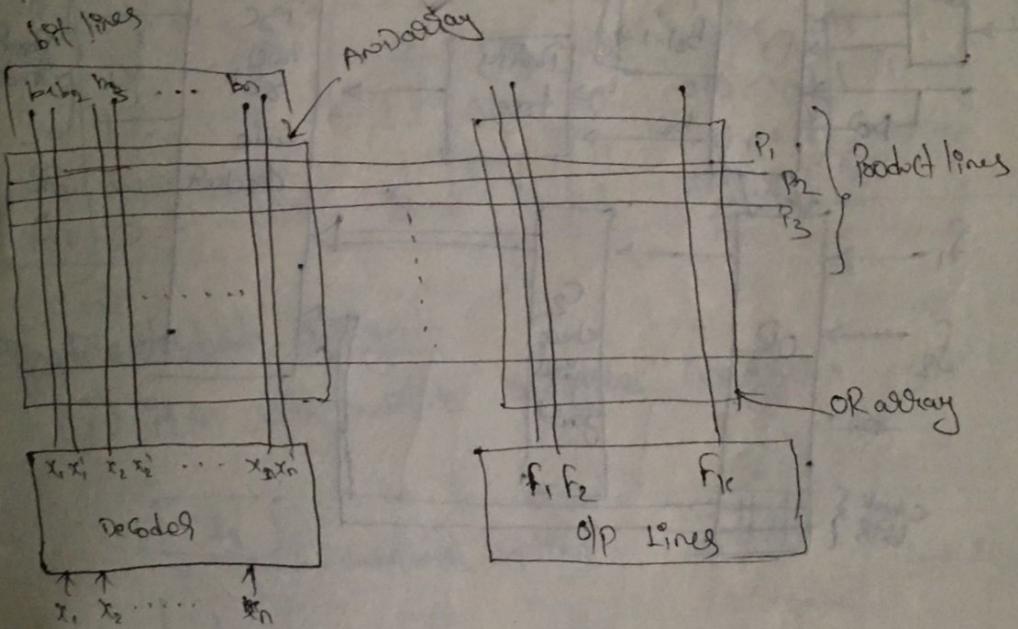
no-connection X

OR  $f_1, f_2$

$$1 \rightarrow \text{Connection} \\ 0 \rightarrow \text{no-Connection}$$

Fault models :- 2 types

- 1) shrinkage faults
- 2) Growth
- 3)Appeal
- 4) Dis-appeal



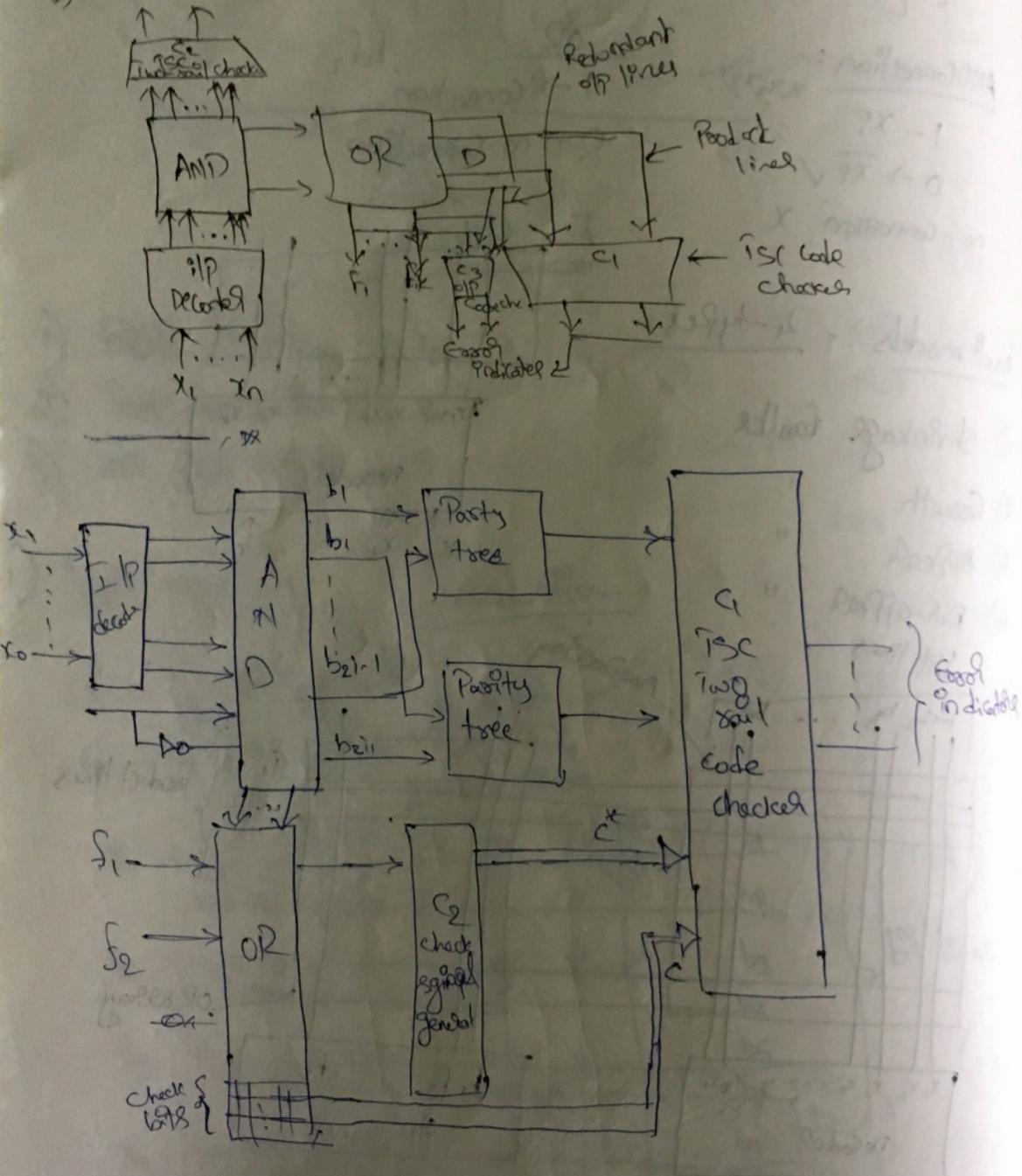
$$n \rightarrow 2^n$$

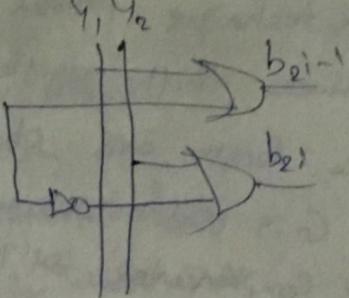
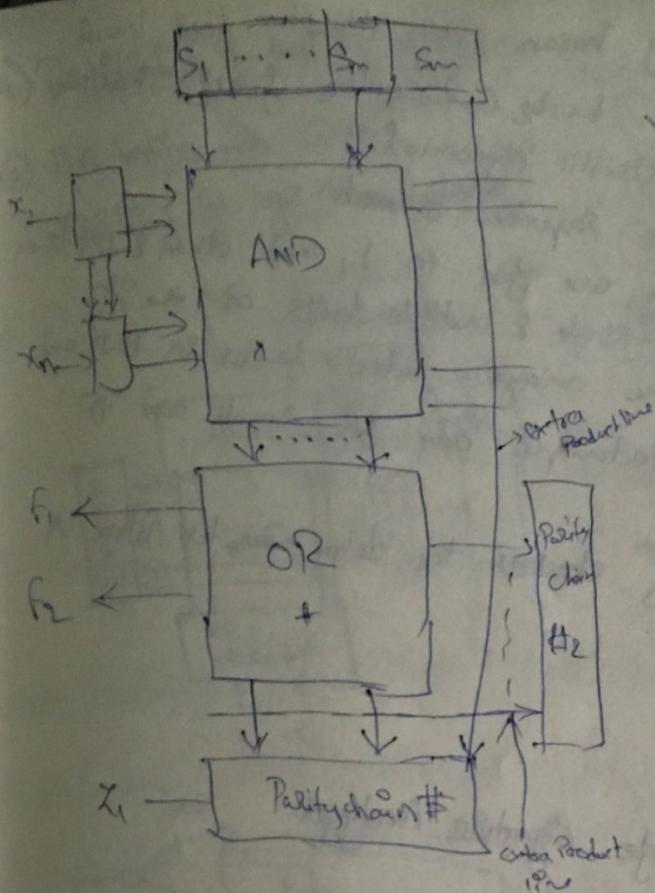
- 1) shrinkage faults :- An extra connection b/w bit line and Product line in the AND array where the missing connection b/w bit line and Product line in an AND array is Growth faults.

Appearance faults :- An Extra Connection, blue Product and open in OR array.

Testable PLA Design:-

1) Concurrent testable PLA's with Special Coding :-



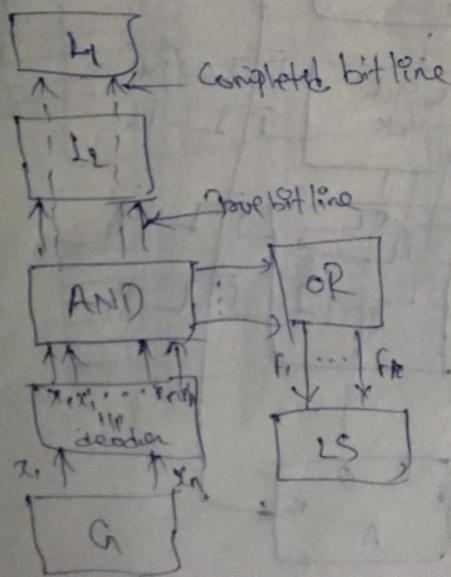


$x_0$	$y_1$	$y_2$	$b_{2i}$	$\bar{b}_{2i}$
a	0	0	a	$\bar{a}$
a	0	1	a	$\bar{a}$
a	1	0	$\bar{a}$	a
-	1	1	$\bar{a}$	a

$x_1 \dots x_i \dots x_n$	$y_1 y_2$	$S_1 \dots S_i \dots S_n$	$y_1 y_2$
$I_1$	...	0 ... 0 ... 0	0
$I_{50}$	0 ... 0 ... 0 ... 0	0 ... 010 ... 0	1
$I_{31}$	1 ... 1 ... 1 ... 1	0 ... 010 ... 1	1
$I_{40}$	1 ... 101 ... 1	1 ... 111 ... 1	$G_1$
$I_{11}$	0 ... 010 ... 0	1 ... 111 ... 1	$G_2$

### ③ Signature testable design:-

(1) PLA with multiple signature Analyzers



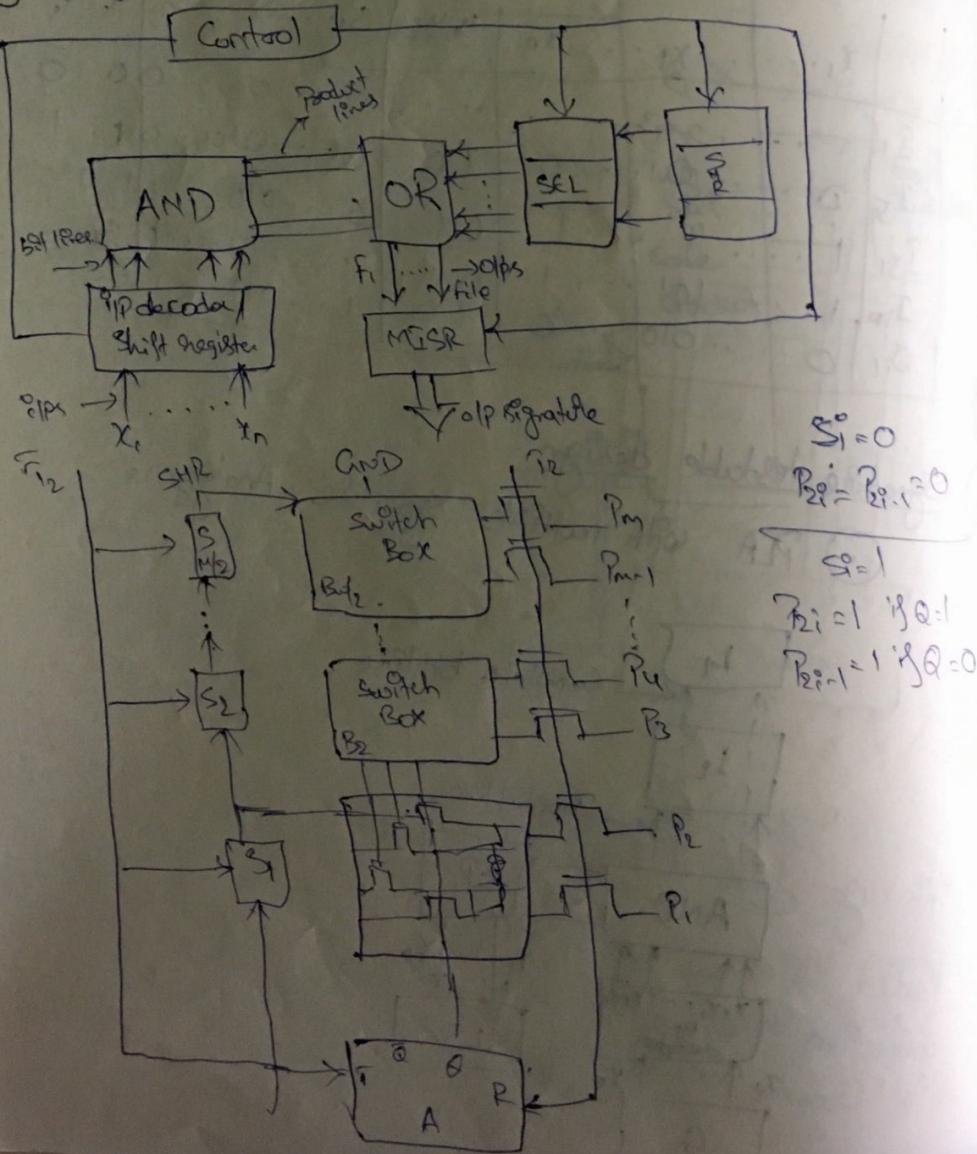
This technique is based by using 8 bits at least for linear feedback shift registers.  $L_1, L_2, L_3$  &  $L_4$  is the first 3 having zero length and characteristic polynomial.

$G_i$  is used maximum sequence generator. The complemented bit lines are fed to  $L_1$  and other 2-bit lines are fed to  $L_2$  to check all single & multiple faults at the QP and bit lines. The signature analyzer detect faults in AND and OR array. Then this technique only for small and QP large.

Disadv: - Lack of Parity checker the delay per test pattern is very small.

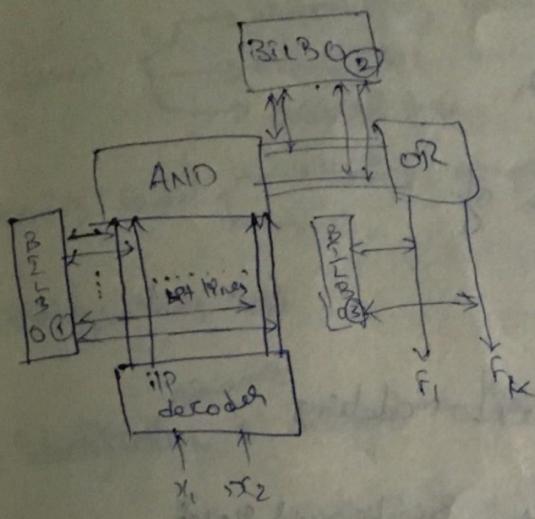
## DSP

Sel testable PLA with single signature Analyser.



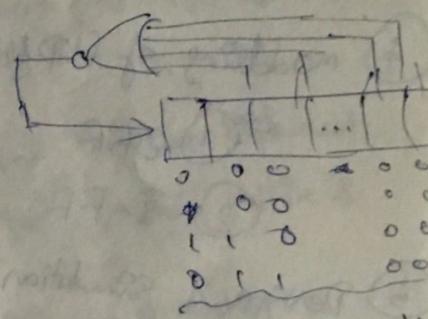
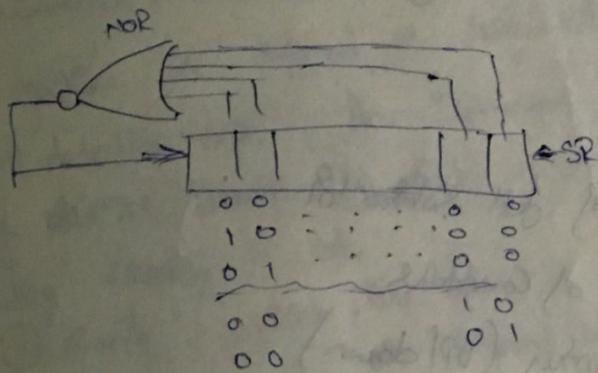
## 2) Partitioning & testing of PLA

(a) PLA testing with BELBO's  
(Building In logic Blocks obseva)

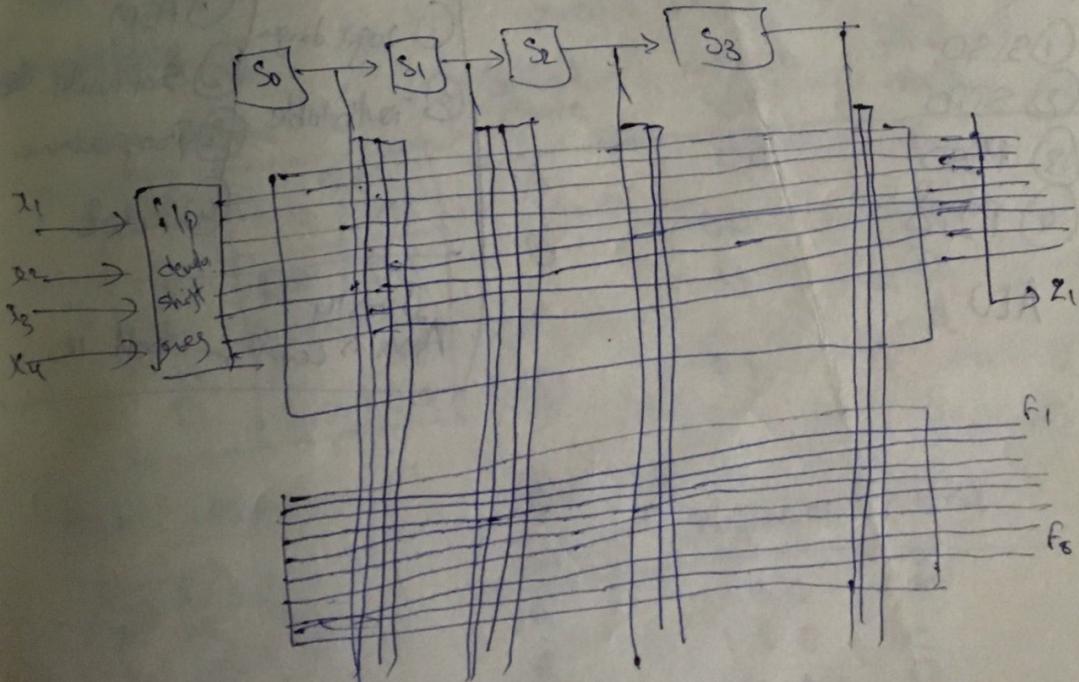


- ① Concurrent
- ② Parity
- ③ Signature Analysis
- ④ Fully Testable PLA

iIP  $\rightarrow$  Test generation  
dIP  $\rightarrow$  Signature Analysis



## (b) Parallel testable PLA (PTPLA)



# 1) Simulation & verification of logic gates

## ② Design simulation of

- ① Half adder
- ② Full  $\rightarrow$
- ③ Carry look ahead adder

## ③ Simulation and verification of

- ④ Decoder
- ⑤ ROM
- ⑥ Encoder using behavioral modeling

## ④ modeling of flip-flops with synchronous reset

and

- ① DFF

- ② T-FF

## ⑤ Design & simulation of JK subtractor

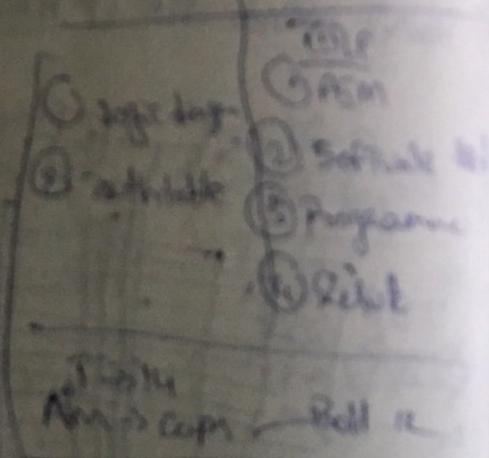
## ⑥ Design & simulation of Counter

- ① Gray scale Counter (up/down)

## ⑦ Design of shift registers

- ① DRSR
- ② SRSR
- ③ PRSR
- ④ PCSR

## ⑧ ALU



# DSD

## Unit - 5

State Identification of fault detection experiments

→ Introductory Example

1) Experiment → uncertainties

→ success tree

② Homing Expert → Homing Sequence

③ Distinguishing Expt

State identification experiments is designed to identify initial state and final state.

Experiments :- The application of IP sequence to IP terminals of a machine is referred as experiment.

Fault detection exp :- An experiment designed to take Experiment through all its transition in such a way that whether machine is

operating correctly or not.

① Simple → which are formed on a single copy of machine

② multiple → " are performed on two or more identical

Copies.

① Introductory Eg:-

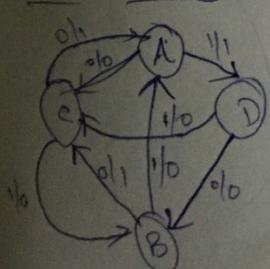
$x = 01, 11$

Machine M<sub>1</sub>

P-S	X=0	X=1
A	00	D <sub>1</sub>
B	C, 0	A, 1
C	A, 1	B, 0
D	B, 0	C, 0

Initial State	Response to 01	final state
A	00	B
B	00	B
C	11	D
D	01	A

State diagram



for 11

Initial State	Response to 11	final state
A	100	B
B	110	C
C	011	D
D	000	A

(2) uncertainties:-

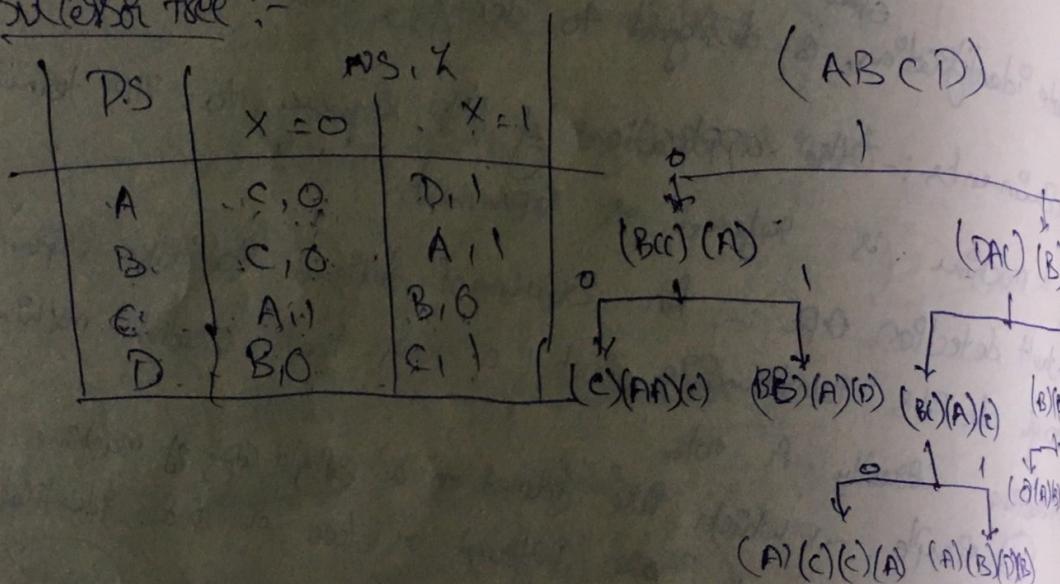
An uncertainty vector whose components contain a single state each is said to be a trivial uncertainty vector.

Eg:- (A) (c) (A) (c)

An uncertainty vector whose components contain either single states or identical repeated states, is said to be homogeneous uncertainty vector.

Eg:- ~~(A)(A)~~ (c) (c) ; (AA)(c)(c)

(3) successor tree :-



O

Homing Experiment:-

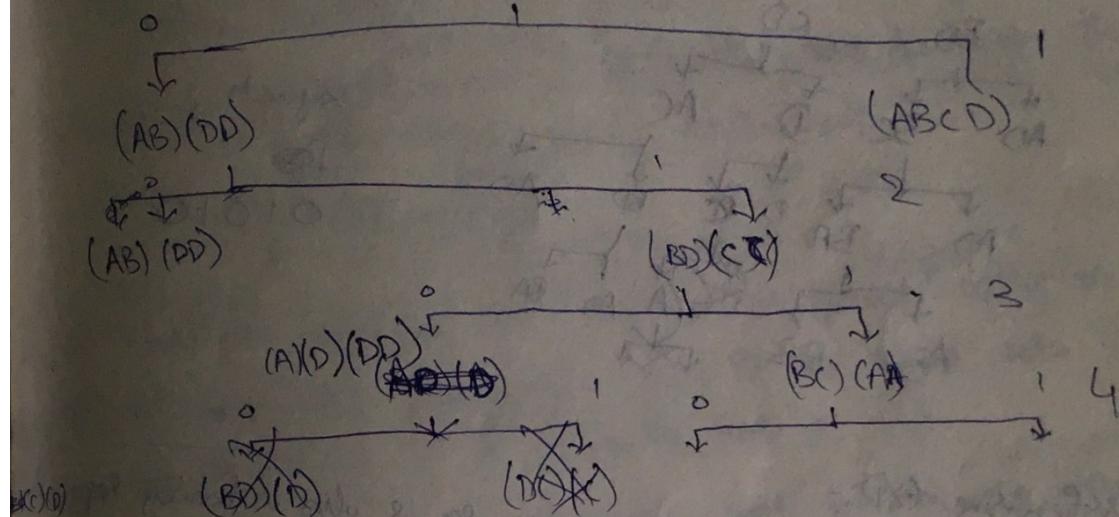
① A homing tree is a success tree in which J-th level node become terminal. when any of the following occur the node is associated with an uncertainty vector whose non homogeneous components are associated with some node in a preceding level.

② Some node in the j-th level is associated with a ~~trivial~~ homogeneous vector.

Homing Seq - (Top)

P.S	NS, Z	
	X=0	X=1
A	B, 0	D, 0
B	A, 0	B, 0
C	<del>B, 1</del>	A, 0
D	D, 1	C, 0

(A B C D)

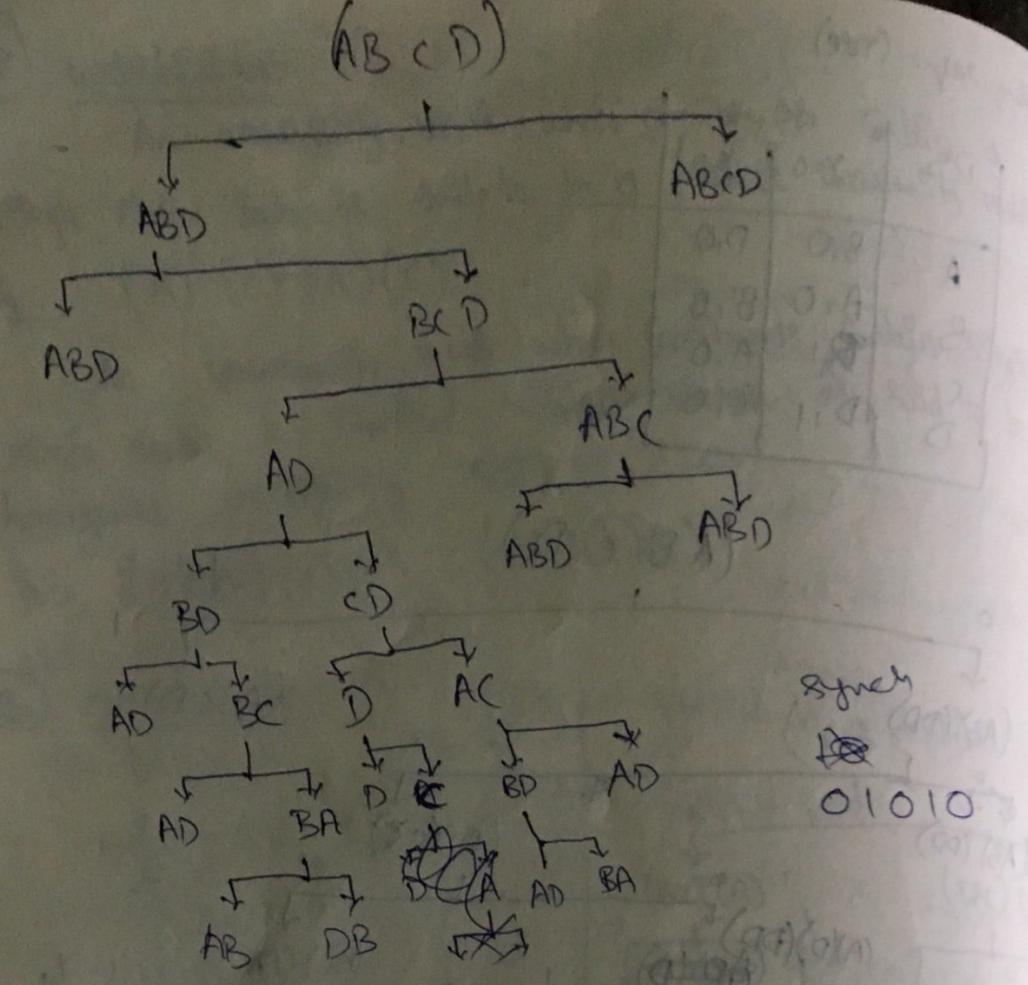


Homing Seq - 010 4<sup>th</sup> level

Initial State	Response to 010	Final State
A	000	D
B	00 1	D
C	10 1	D
D	10 1	D

Synchronising Exp:-

P.S	NS, Z	
	X=0	X=1
A	B, 0	D, 0
B	A, 0	B, 0
C	<del>D, 1</del>	A, 0
D	D, 1	C, 0



syntax  
B  
01010

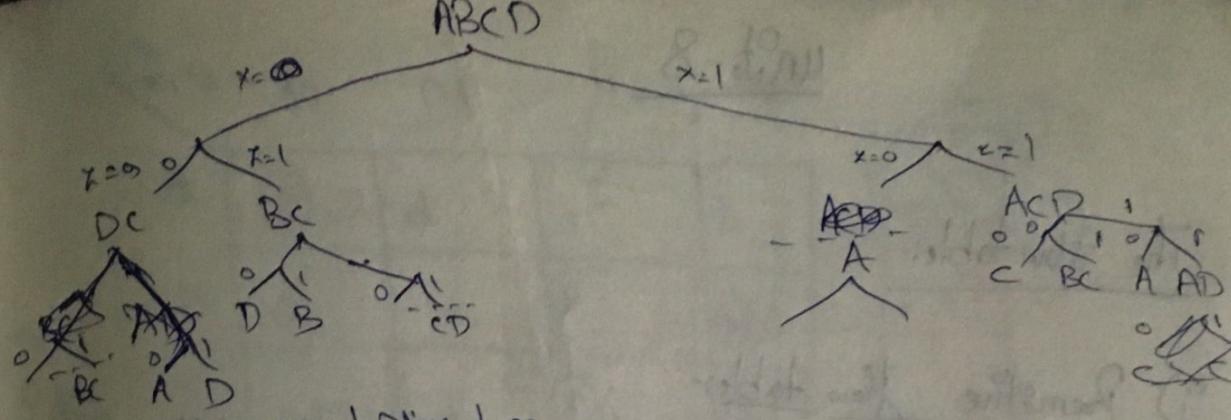
### Distinguishing EXP:-

Every distinguish seq is also a having seq

- ① Reset disting seq → Same as success tree
- ② Distingu Tree → Same as success tree
- ③ The shortest distinguish Prefix →
- ④ Adaptive distinguish Experiments ✓

### ③ The shortest distinguish Prefix:-

PS	NS, Y	
	X=0	X=1
A	C, 0	A, 1
B	D, 0	C, 1
C	B, 1	D, 1
D	C, 1	A, 0



Partial adaptive tree

single active stop the separation in tree

~~Step 1~~

ABCD

$x=0$

$z=0$

$z=1$

DC

BC

$x=1$

$x=0$

$z=0$

$z=1$

A

D

B

C

A Partial tree describing the experiments.

A Complete Experiment is Summarized as follows apply an op '0' and inspect colouring of p, if the op is zero go to Step-2' of if '1' go to Step '3'.

Step-2' apply an op '1' the initial state can now be determined uniquely by observing response if it is zero the initial state was 'B'. if it is '1' the initial state was 'A'

Step 3: Apply an op '0' & the response to the op is zero, the initial state was 'C' if it is '1' the initial state was 'D'.

P.S	NS, Y	
	$x=0$	$x=1$
A	A, 1	E, 0
B	A, 0	C, 0
C	B, 0	D, 1
D	C, 1	C, 0
E	C, 0	D, 0

## The flow table:-

### Q) Premitive flow table:-

A Premitive flow table is a flow table  
is a flow table having only one stable state for  
now with outputs specified only for stable states.

Condition :- The o/p is to be zero until the i/p becomes  
 $x_1, x_2 = \text{b} \cancel{\text{0}} \text{0} 11$  but only if it follows an i/p of

10

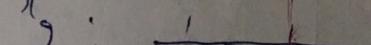
10 11

[after 10 11 o/p then o/p is]

$x_1, x_2 :$	00	01	11	10	11	10	00	10	11	01	11	✓
$z :$	00	0	0	0	1	0	0	0	1	0	0	

Note :-

\* :

 $x_1 :$  $x_2 :$  $z :$ 

State	i/p	o/p
a	00	0
b	10	0
c	11	1
d	01	0

table  
State for  
States -

the  $\oplus P$  becomes  
in  $\oplus P$  of  
active, then  $\oplus P = 1$

01 11 ✓  
00

	00	01	11	10
a	(@, 0)	d	-	b
b	a	-	c	(b, 0)
c	-	d	(c, 1)	b
d	a	(d, 0)	e	-
e	-	d	(e, 0)	b

Any 1 bit has to be changed while allotting the states

## Unit - 6

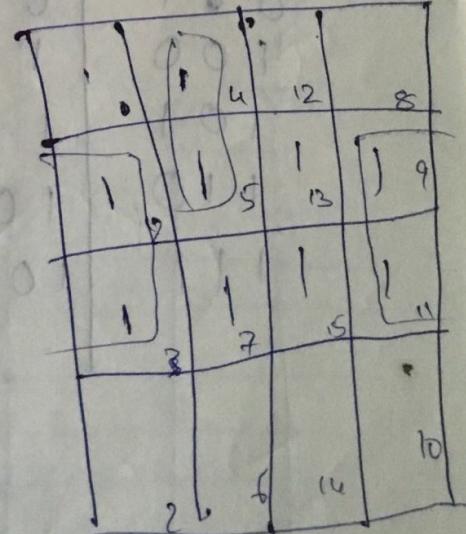
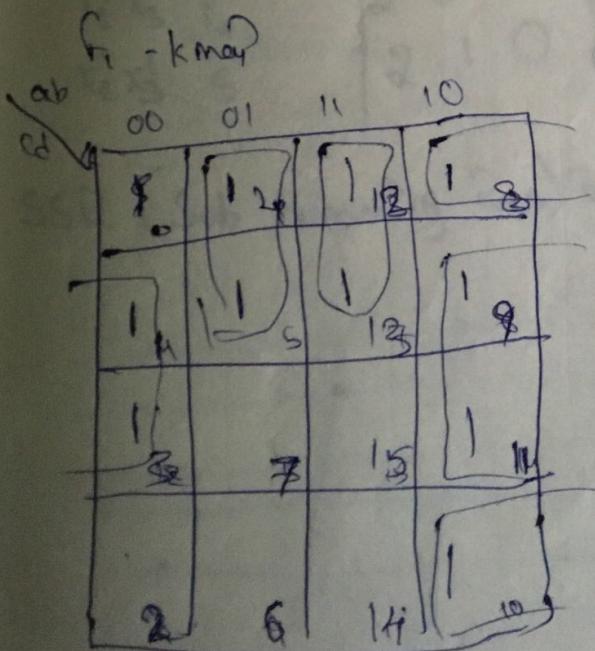
Programmable logic Arrays:-

PLA minimization:-

$$f_1 = \Sigma(1, 3, 4, 5, 8, 9, 10, 11, 12, 13)$$

$$f_2 = \Sigma(1, 3, 4, 5, 7, 9, 11, 13, 15)$$

$$f_3 = \Sigma(5, 7, 8, 10, 12, 13, 15)$$



	0	4	11	18
a				
b	1	5	12	9
c	3	7	15	14
d	2	6	14	10

PLA Holdings

Convert 3-bit BCD to Gray Code and draw PLA and check whether folding is possible or not

PLA minimization is nothing but reducing the no. of Redundant terms.

$B \oplus D$	Gray Code
000	$\begin{smallmatrix} x_1 & x_2 & x_3 \\ 0 & 0 & 0 \end{smallmatrix}$
001	$\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}$
010	$\begin{smallmatrix} 0 & 1 & 1 \end{smallmatrix}$
011	$\begin{smallmatrix} 0 & 1 & 0 \end{smallmatrix}$
100	$\begin{smallmatrix} 1 & 1 & 0 \end{smallmatrix}$
101	$\begin{smallmatrix} 1 & 1 & 1 \end{smallmatrix}$
110	$\begin{smallmatrix} 1 & 0 & 1 \end{smallmatrix}$
111	$\begin{smallmatrix} 1 & 0 & 0 \end{smallmatrix}$

$x_3$	00	01	11	10
0	0	1	3	2
1	1	1	1	1
	4	5	7	6

(1)  $Z_1 = X_1$

$x_3$	00	01	11	10
0	.	0	1	1
1	1	1	1	1

(3) (2)

$$Z_2 = X_1' \cdot X_2 + X_1 X_2'$$

(4) (5)

$$Z_3 = X_2' X_3 + X_2 X_3'$$

$x_3$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

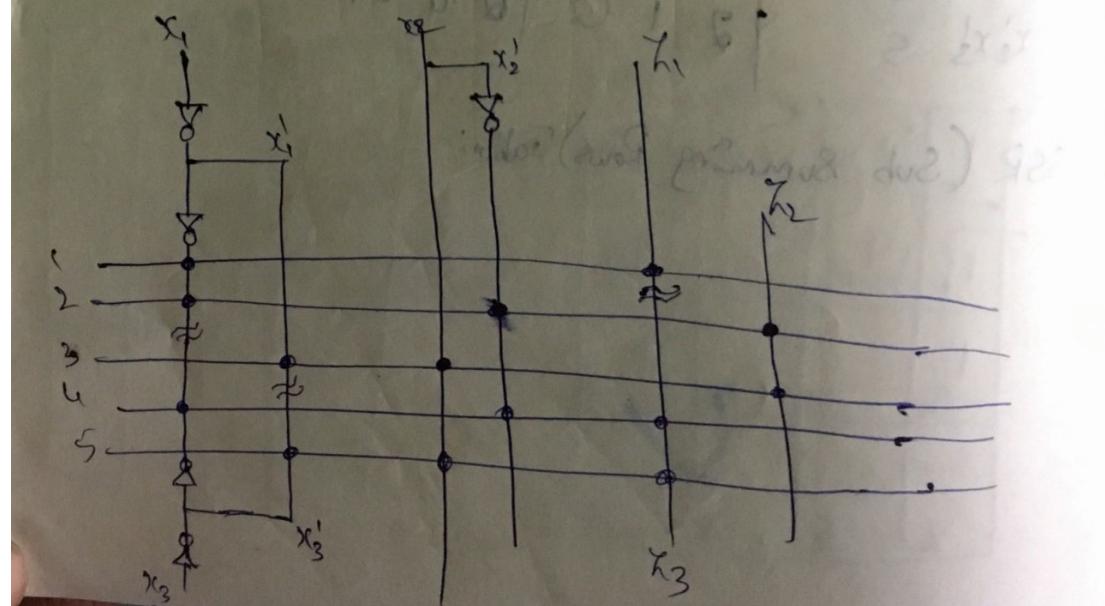
Product term	00P	01P	11P	00P
$X_1$ 1	1	2	2	1 0 0
$X_1 X_2$ 2	1	0	2	0 1 0
$X_1' X_2$ 3	0	1	2	0 1 0 Q
$X_2 X_3$ 4	2	0	1	Q 0 1
$X_2' X_3$ 5	2	1	0	Q 0 0 1

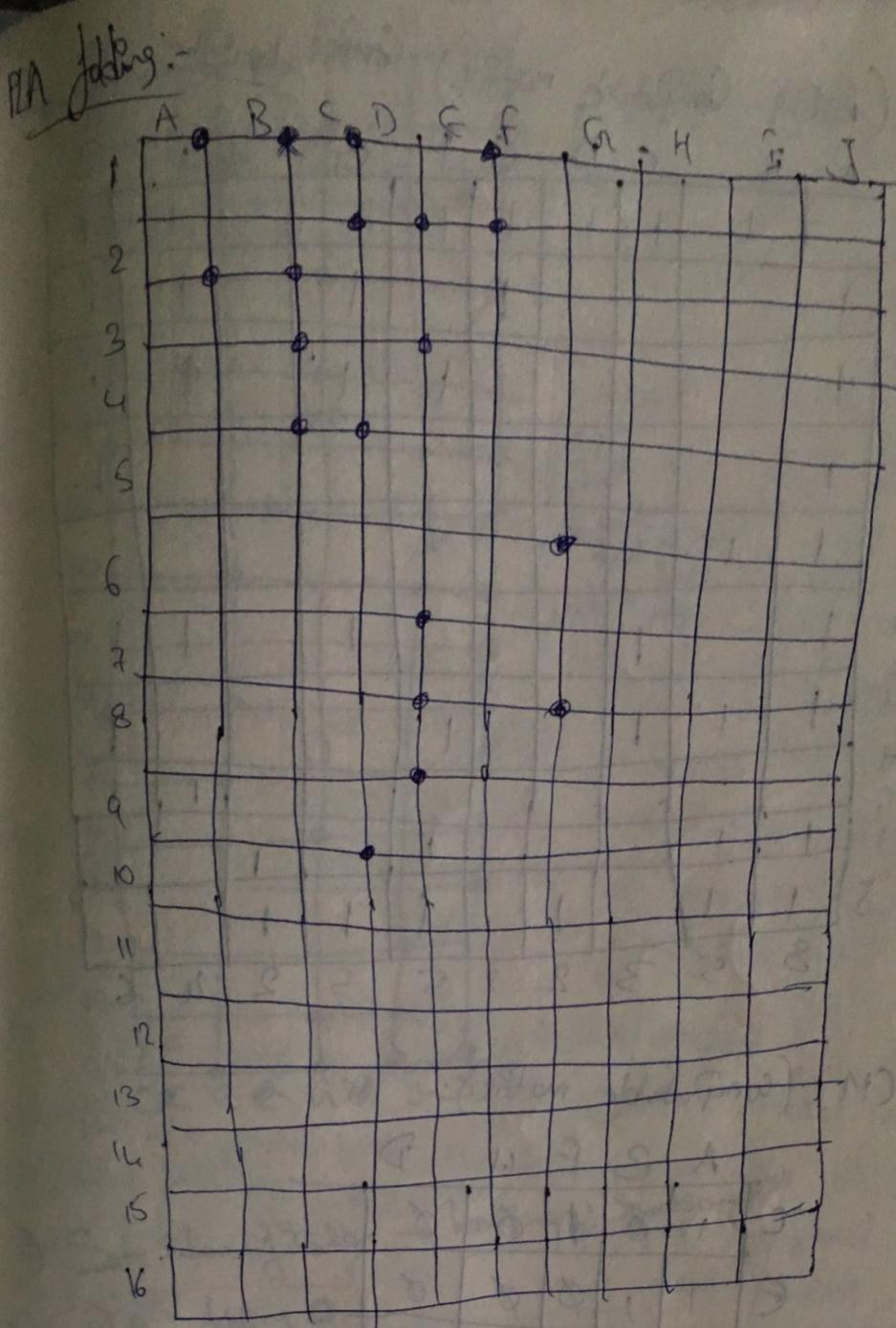
0 → no connection  
1 → True Connection  
0 → Complement

SSR (Sub Summing Rows) Table:

Column	SSP's
$x_1$	1, 2
$x'_1$	3
$x_2$	3, 5
$x'_2$	2, 4
$x_3$	4
$x'_3$	5
$x_1$	1
$x_2$	2, 3
$x_3$	4, 5

Columns	SSP's
$(x_1, x_1)$	1, 2, 3
$(x_2, x'_2)$	3, 5, 2, 4
$(x_3, x'_3)$	4, 5
$x_1$	1
$x_2$	2, 3
$x_3$	4, 5





SSRs of PLA

Column	SSR	Row
A	1, 4 1, 3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
B	3, 4, 5, 9, 12	
C	1, 2, 5, 10, 11, 15	
D	2, 4, 7, 8, 10, 15, 16	
E	1, 2, 11, 15	
F	6, 8, 11, 13	
G	3, 5, 6, 14, 15	
H	7, 10, 12, 13, 16	
I	3, 9, 16	
J		

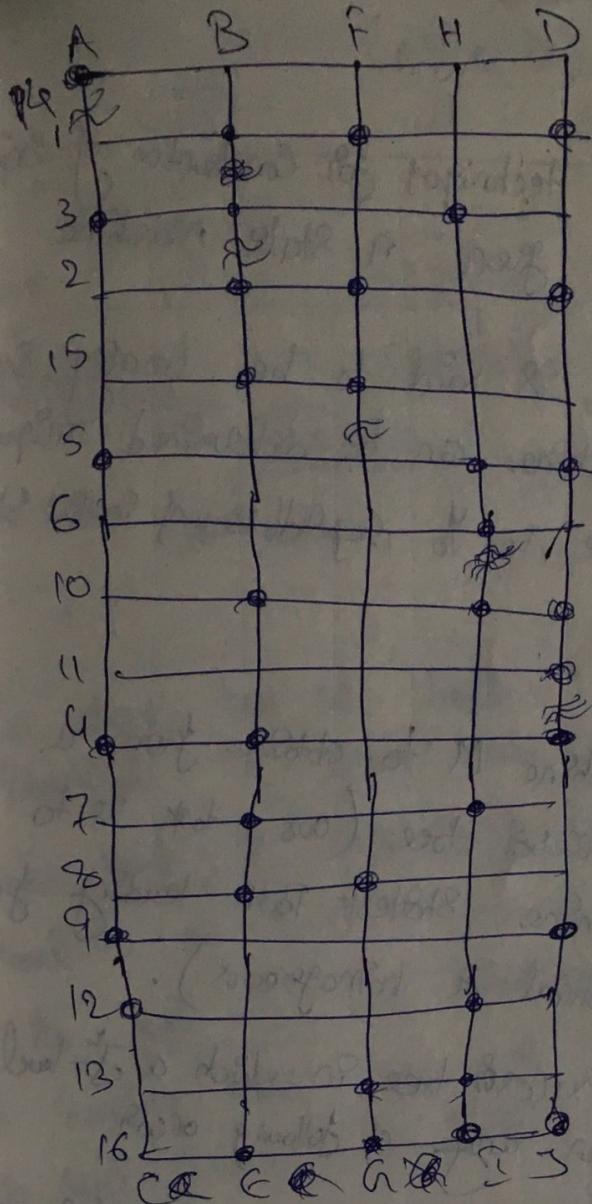
FCM (folding Compatible matrix) Compact algorithm

	A	B	C	D	E	F	G	H	I	J
A		1	1	1	1	1	1	1	1	1
B	1				1	1	1	1	1	1
C	1					1	1	1	∅	1
D	1						1			1
E	1	1								.
F	1		1				1		1	1
G	1	1	1				1			1
H	1	1	1				1			1
I	1	1	1				1		1	1
J	1	1	1	1			1	1	1	1
	8	5	3	2	2	5	5	2	4	6

CM (Compatible matrix) :-

	A	B	F	H	D
C	1	0	∅	∅	∅
E	1	1	∅	∅	∅
G	1	1	1	0	0
I	1	1	1	1	∅
J	1	1	1	1	1

folds      Pairs  
 $(A, C)(B, E)(F, G), (H, I)(D, J)$  FCM.



5 pair  
max colour  
joining in  
using compact  
PLA

- 1) State identification  $\rightarrow$  homing; Synchronous
- 2) PLA folding
- 3) Flow table  $\rightarrow$  fault; hazards
- 4) Fault  $\rightarrow$  signature

## Homing Experiments:-

used to develop techniques for construction of  
to identify final state of given  $n$ -state machine

Def:- An I.P sequence  $y_0$  is said to be homing seq if the final state of machine can be determined uniquely from the machine response to  $y_0$ , regardless of initial state.

### → The homing tree:-

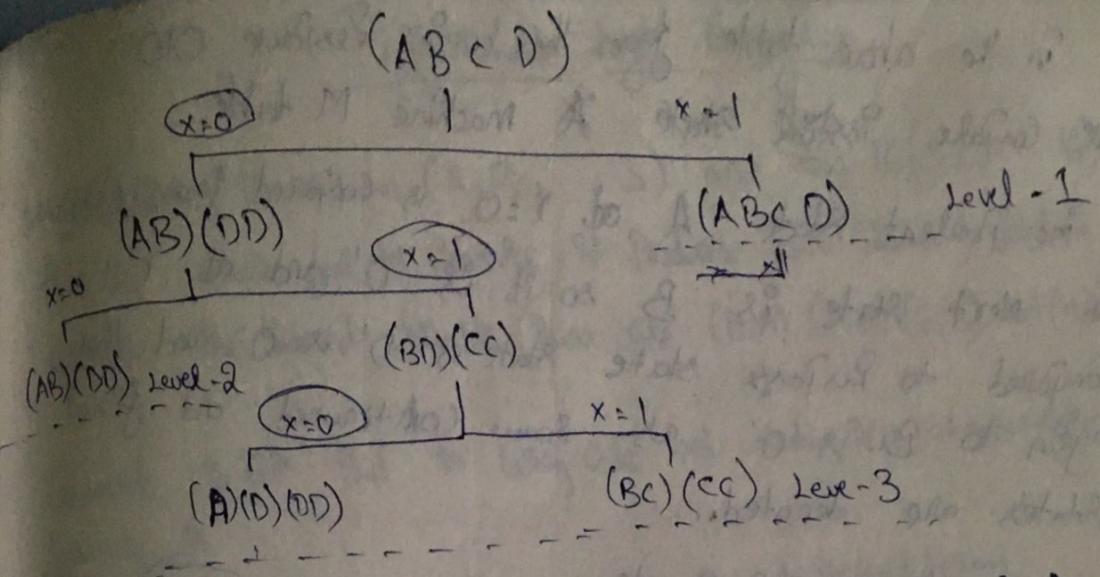
Given by machine  $M$  to obtain from a truncated version of its success tree (our task is to construct the tree to determine shortest path leading from the initial uncertainty to trivial & homogeneous).

A homing tree is success tree in which a  $j^{\text{th}}$  level node becomes terminal when any of following occur

- 1) The node is associated with an uncertainty vector whose non-homogeneous components are associated with some nodes in a preceding level.
- 2) Some node in  $j^{\text{th}}$  level is associated with a trivial or a homogeneous vector.

The homing tree of "Machine M" shown below table  
the node associated with vector  $(AB)(DD)$

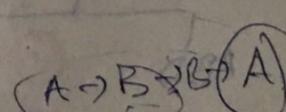
P. S	N. S	
	$X=0$	$X=1$
A	B, 0	D, 0
B	A, 0	B, 0
C	D, 1	A, 0
D	D, 1	C, 0



- \* In above tree at level -1  $(ABCD)$  is repeated as it is identical to given node  $(ABCD)$ .
  - \* In level -2  $(AB)(DD)$  node at  $x=0$  is identical to level -1 at  $x=0$ .
  - \* At level -3  $x=0$   $(A)(D)(DD)$  is successor node as it is homing term and at  $x=1$   $(B'C')(CC)$  node is identical to level -2  $x=1$ . The response of homing tree is "010".
- homing sequence:- whose length is at most  $(n-1)$  exists for every reduced  $n$ -state machine M.

From table machine M

PS	N.S	
	$x=0$	$x=1$
A	B, 0	D, 0
B	A, 0	B, 0
C	D, 1	A, 0
D	D, 1	C, 0



In the above ~~tree~~ from initial state A  
from the response of the homing tree is 010

Initial state	Response to 010	Final state
A	000	A ✓
B	001	D
C	101	D
D	101	D

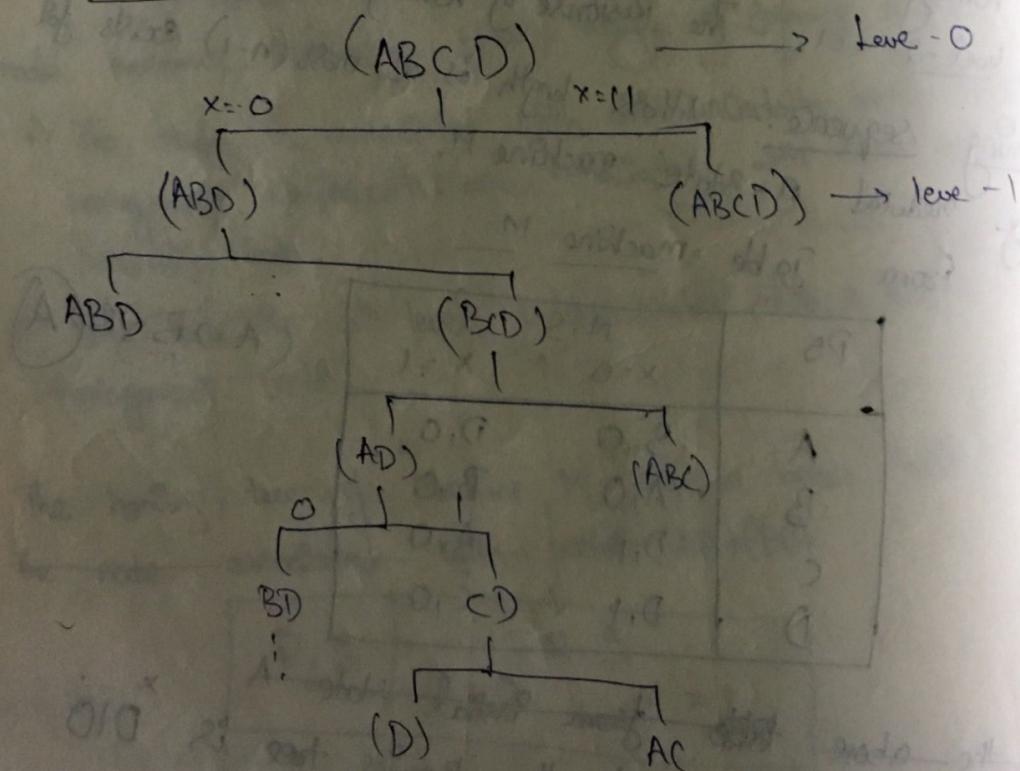
In the above table from the binary Residue 010  
we compare initial state A machine M table

The Present State A at  $x=0$  is compared from Response  
as Next state is B so it is '0' and at  $x=1$  is  
compared to Response state state is '0' is 0 and then  
again to B is '0' as same continued and final  
states are denoted....

Synchronizing Experiments:-

DD  $\rightarrow$  D

From machine 'M' table		N.S
P.S		
	$x=0$	$x=1$
A.	B, 0	D, 0
B	A, 0	B, 0
C	D, 1	A, 0
"D	D, 1	C, 0



In above tree it is defined until the single terms  
of tree arrived.

\* not need to write repeated entries e.g. (ABDD) is  
written as (ABD).

## Programmable Logic Array:

Consider 10-column ( $A, B, \dots, J$ ) and row-16 ( $1, 2, \dots, 16$ ) PLA. When fed to PLA, it produces four pairs of foldable columns. The ordered foldable pairs are  $(D, A)$ ,  $(F, C)$ ,  $(G, B)$  and  $(H, J)$ .

Convert 3-bit BCD to Gray code and draw PLA chart without folding.

PLA minimisation is nothing but reducing no. of product terms.

PLA folding is " " " " " *where*

BCD			Gray Code			$\bar{a}b+\bar{b}a$
$x_1$	$x_2$	$x_3$	$z_1$	$z_2$	$z_3$	
0 0 0	0	0	0	0	0	
0 0 1	0	0	0	0	1	
0 1 0	0	1	0	1	1	
0 1 1	0	1	0	1	0	
1 0 0	1	0	1	1	0	
1 0 1	1	0	1	1	1	
1 1 0	1	1	0	1	0	
1 1 1	1	1	1	0	0	

$$\begin{aligned} z_2 &= \bar{x}_1 x_2 + \bar{x}_2 x_3 \\ z_3 &= x_2 x_3 + \bar{x}_2 x_2 \end{aligned}$$

K-map for  $Z_1$ :

$x_3$	00	01	11	10
$x_1$	0	0	1	1
$x_2$	0	1	1	1
	4	5	6	7

$$\Rightarrow Z_1 = X_1 \quad \textcircled{1}$$

K-map for  $Z_2$ :

$x_3$	00	01	11	10
$\bar{x}_1$	0	0	1	1
$x_1$	0	1	1	1
	4	5	7	6

$$\begin{array}{c} \textcircled{2} \quad \textcircled{3} \\ Z_2 = \bar{x}_1 \cdot \bar{x}_2 + x_2 \cdot \bar{x}_1 \end{array}$$

### K-map for $Z_3$

$x_1$	$x_2$	$x_3$	$Z_3$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$Z_3 = \bar{x}_2 x_3 + x_2 \bar{x}_3$$

(4)      (5)

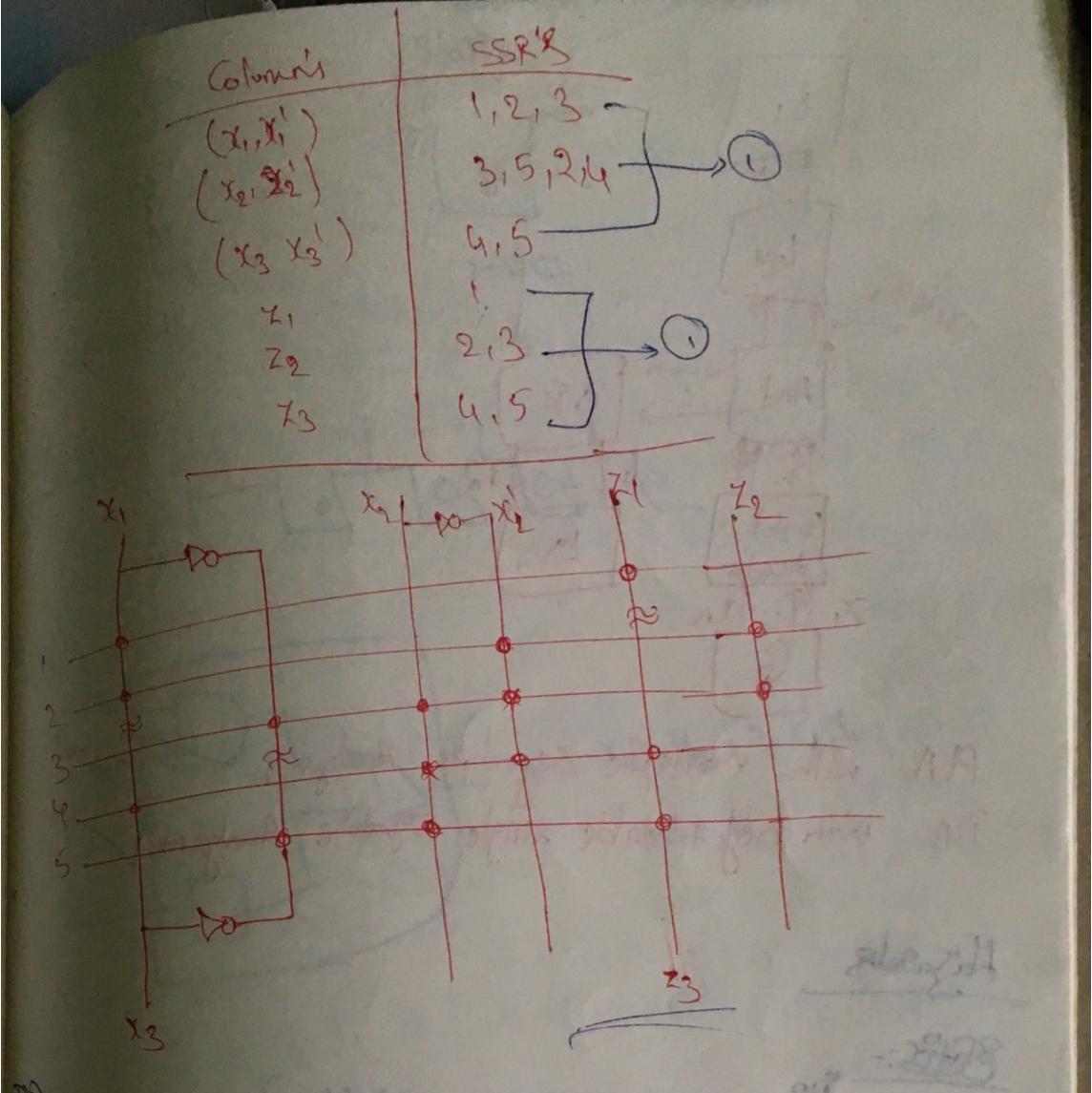
Product terms	PIP			OP		
	$x_1$	$x_2$	$x_3$	$x_1$	$x_2$	$x_3$
$x_1$ 1	1	-	-	1	-	-
$\bar{x}_1 x_2$ 2	1	0	-	-	1	-
$\bar{x}_1 x_2 x_3$ 3	0	1	-	-	1	-
$x_2 x_3$ 4	-	0	1	-	-	1
$x_2 \bar{x}_3$ 5	-	1	0	-	-	1

1 denotes True Connection

0 - denotes Complement (8) OFF state

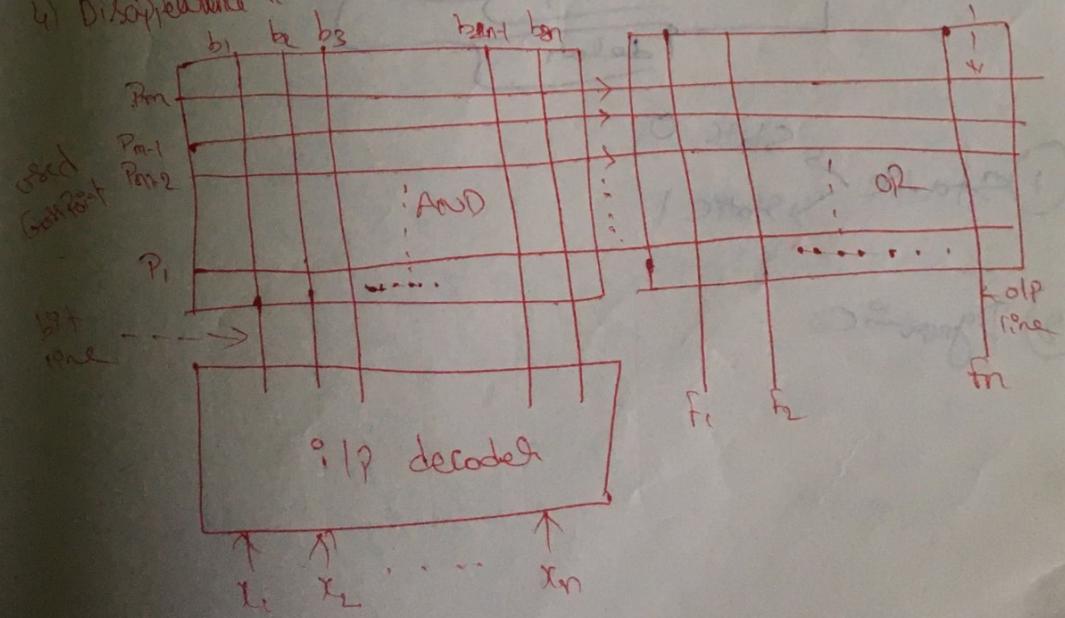
- denotes no connection

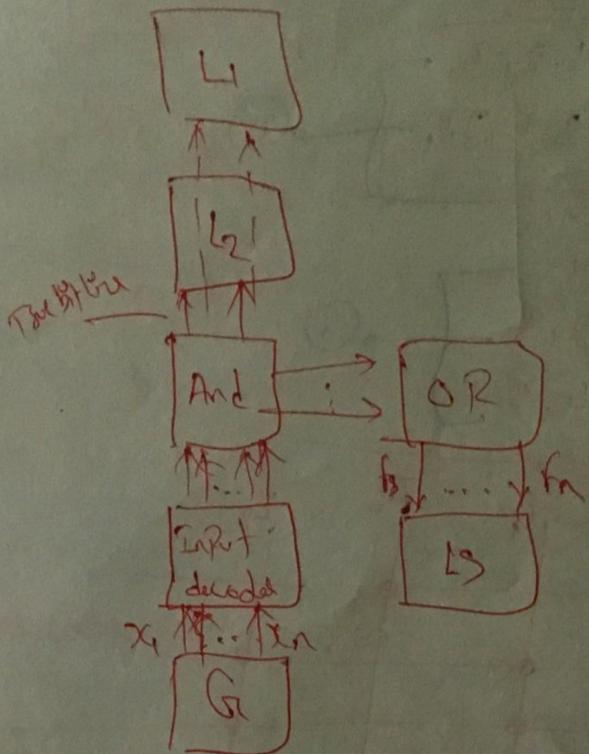
Columns	SSR's
$x_1$	1, 2
$\bar{x}_1$	3
$x_2$	3, 5
$\bar{x}_2$	2, 4
$x_3$	4,
$\bar{x}_3$	5
$x_1$	1
$x_2$	2, 3
$x_3$	4, 5



### PA fault models:-

- 1) Shrinkage fault:-
- 2) Growth fault
- 3) Appearance
- 4) Disappearance

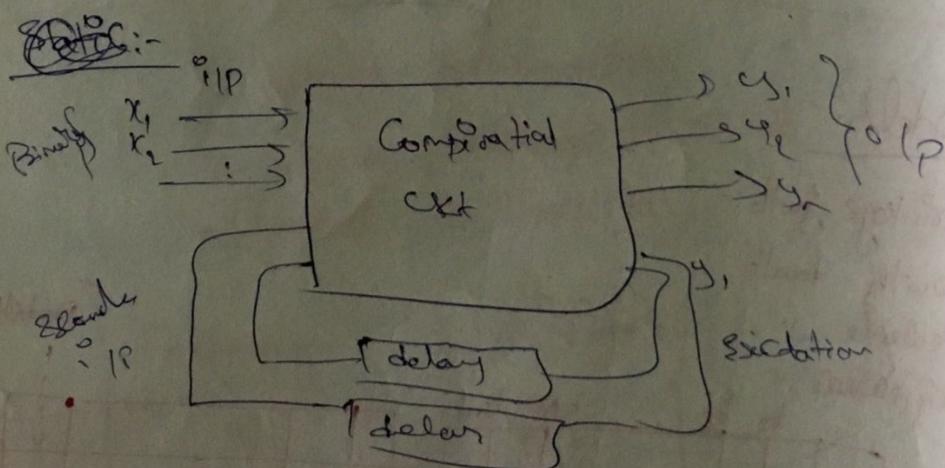




PLA with multiple signature Analysis ✓

PLA with self testable single signature Analyzer ✓

### Hazards



1) Static  $\rightarrow$  State 0  
 $\rightarrow$  State 1

Dynamic