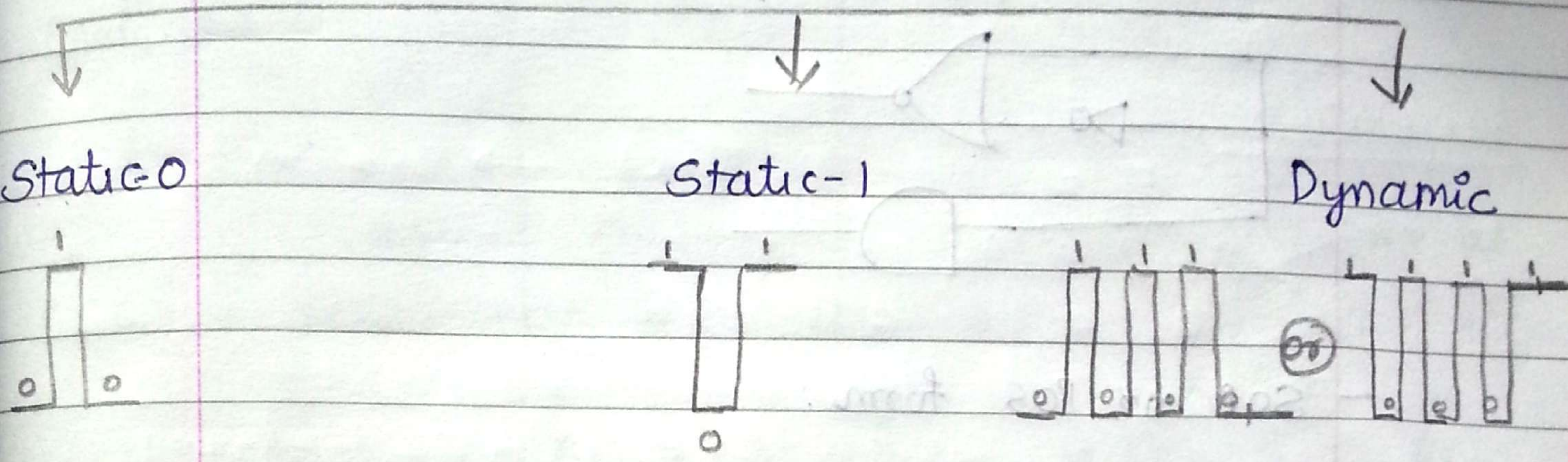
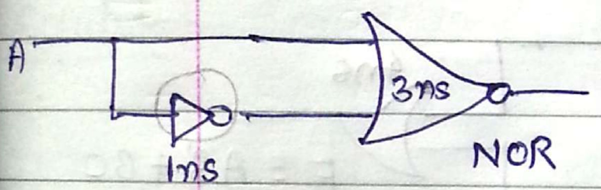
Hazards :

Unwanted transition on the o/p

HAZARDS



①

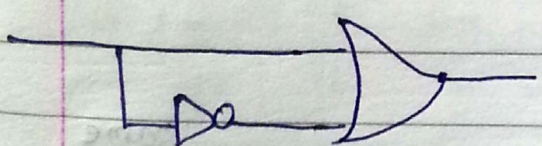


$Y = A \cdot \bar{A} = 0$

diff^r propagation path, having diff^r propagation delay → due to that hazard will occur.
as in example in 1 ns → hazard will there → we don't get right o/p ans.

- pos form

②

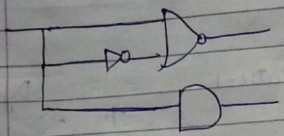


$Y = A + \bar{A} = 1$

we are expecting static o/p is one but for a time we get o/p 0 → that's known as Static-1

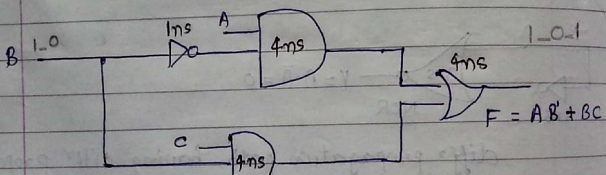
- sop form

③ flip to zero more than once ⟶ that's known as dynamic and after some time its settled to 1

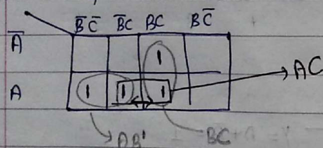


– SOP or POS form

⇒ Static-1 :



- final eqⁿ in the SOP form, for the static-1
- K-map



- two adjacent one, which are not in same pair

$A\bar{B}C$ ABC

transition of B is there (bcoz it's goes on two diffʳ path)

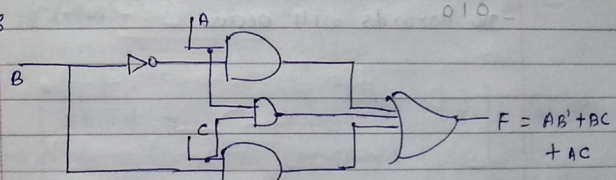
- for 1 ns, we are getting 0 at the O/P side ⟶ which causes the hazards.

- Analysis :

	m7 ABC(111)	m5 $A\bar{B}C$ (101)
BC	1	0
AB'	0	1 (after some time at that time we get 0)
	1	0 1

BC > AB' ⟶ speed is faster in BC

- Solution :



- we added one term that's → AC

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

without adding AC

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

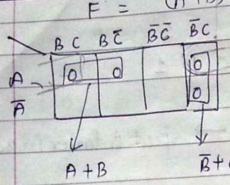
with adding AC

$F = AB' + BC$ HW

POS "static 0"

for static 0 ;

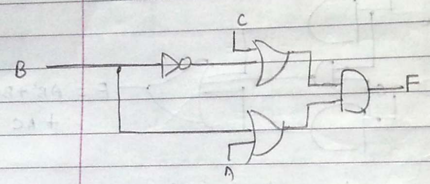
$F = (A+B)\cdot(\bar{B}+C) \longrightarrow$ POS form



$A+C$ → To remove the hazards

$\boxed{x\bar{y} + yz = (x+y)(\bar{y}+z)}$

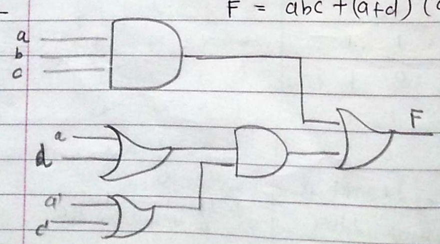
- no Hazards will occurs.



$F' = (A+B)(\bar{B}+C)(A+C)$

Ex:-

$F = abc + (a+d)(a'+c')$

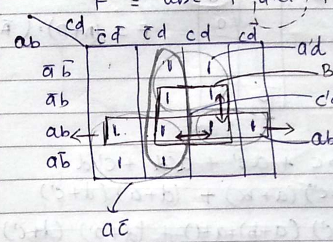


7 - feb

in SOP form:

$F = abc + a\cdot a' + ac' + a'd + c'd$

Pair of 4



- three adjacent pairs

1) (13-15) 1111 - 1101
2) (15-7) 1111 - 0111
3) (12-14) 1100 - 1110

1)

	1111 abcd	1101 abcd
$a\bar{c}$	0	1
abc	1	0

2)

	1111 abcd	0111 abcd
a'd	0	1
abc	1	0
	1	0 1

a'd > abc

Hazard will there

3)

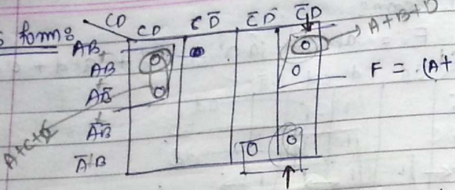
	1100 abcd	1110 abcd
$a\bar{c}$	1	0
abc	0	1

$abc > a\bar{c}$

Hazard will there

By adding B·D, we remove the effect of Hazard of due to 1) & 2) and for 3) we added the term of AB

POS form:



$F = (A+D)\cdot(\bar{A}+B+\bar{C})$

$F = abc + ac' + a'd + c'd$

$= \cancel{(a+c')(a+bc) + (d+a')(d+c')}$

$= \cancel{(a+c')(a+b)(a+c) + (d+a')(d+c')}$

$x+yz=(x+y)(x+z)$

$= abc + (a+d)(a'+c')$

$= [abc + (a+d)][abc + (a'+c')]$

$= (a+(a+d))[bc+(a+d)] \cdot [a+(a'+c')] \cdot [bc + (a'+c')]$

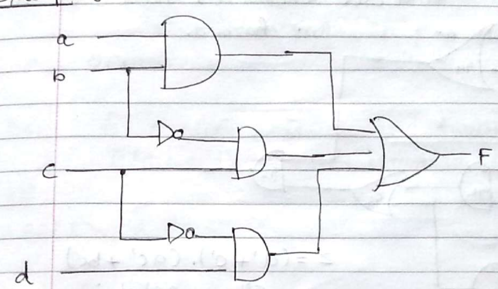
a+d

$= (a+d)(b+a+d)(c+a+d)\cdot(a'+b+c')$

$= (a+d)(a'+b+c')(a+b+d)(a+c+d)$

	0 0 1 0 A B C D	1 0 1 0 A B C D
$\bar{A}+B+\bar{C}$	1	0
$A+D$	0	1

Example:



$F = a\cdot b + \bar{b}c + \bar{c}d$

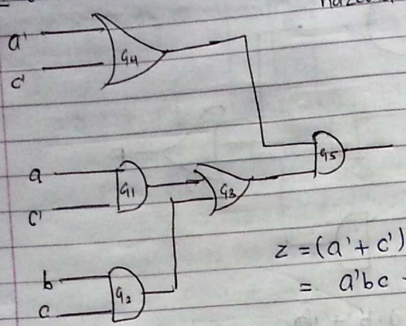
	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$		1	1	1
$\bar{a}b$		1		
ab	1	1	1	1
$a\bar{b}$		1	1	1

ad

Date: 20-feb.

Dynamic hazard

Example: Find out whether it's static 1 or static 0 hazard.



$z = (a'+c').(ac'+bc)$

$= a'bc + ac' + \cdot$

in SOP form

Sol:

a \ bc	b'c'	b'c	bc	bc'
a'			1	
a	1			1

No static '1'

in POS $z = \cancel{(a'+c'+b)(a'+c'+b')} \cancel{(ac'+bc)}$

$(a'+c')(ac'+b)(ac'+c)$

$= (a'+c')(a+b)(c'+b)(a+c)(c+c')$

$= (a'+c')(a+b)(a+c)(b+c')$

	bc	bc'	b'c'	b'c
a	0	0		0
a'		0	0	

a+b

→a+c

a'+c'

b+c'

No static '0' Hazard

here neither static-0 or staic-1 hazard will occur.

Dynamic hazard: bcoz c is travelling through the diff^r three paths. so may be hazard will occurs.

Ex: for a given ckt, design hazard free ckt.

$F(a,b,c,d) = \Sigma m(1,5,7,14,15)$

Sol:

in SOP:

	c'd'	c'd	cd	cd'
a'b'		1		
a'b		1	1	
ab			1	1
ab'				

$f = a'c'd + a'bd + bcd + abc$

in POS:

0		0	0
0			0
0	0		
0	0	0	0

$f =$

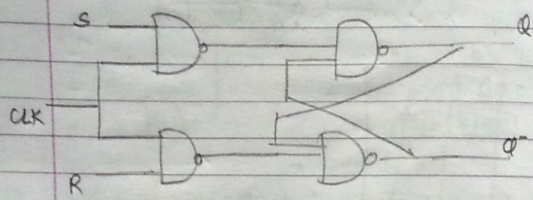
Date: 20 -feb.

Sequential Circuit

O/P depends on previous O/P and present input.

flip flops:

① **S-R flip flop:** (Set-Reset F/F)



Reset = 1, O/P = 0
Set = 0, O/P = 1

Clock → control the ckt.

NAND → where there is no supply, then O/P = 1, initial state of NAND gate

latch → in latch when we change the I/P, then it goes first into the initial state and then it changes the input and gives the output.

Flip-flop → by providing clock, we can change the inputs, we don't requires the inputs goes to the initial state

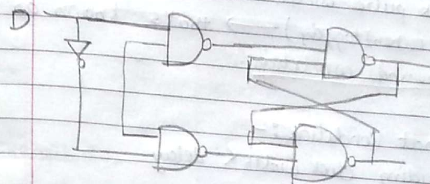
Char^c table

S	R	Q (Q_{n+1})	$\bar{Q}$ ($\bar{Q}_{n+1}$)
0	0	memory	
0	1	0	1
1	0	1	0
1	1	invalid	

Excitation table

Q_n (Ps)	Q_{n+1} (Ns)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	0	X	0

② **D-flip flop:**
(Data F/F)



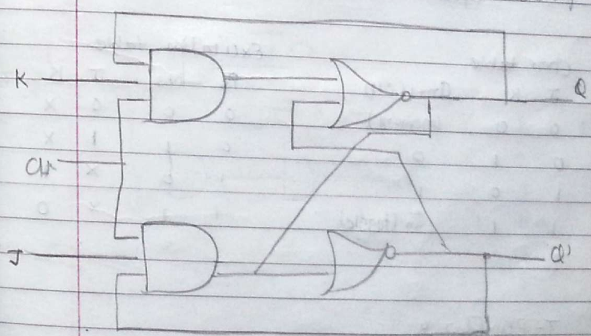
Char^c table

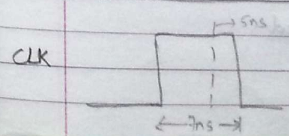
D	Q_{n+1} (O/P)
0	0
1	[illegible]

Excitation

Ps	Ns	D
0	0	0
0	1	1
1	0	0
1	1	1

③ **J-K flip flop:**





Why char^c → truth table

When J=K=1, then propagation delay of F/F is lesser than then clock pulse width, then O/P continues changing (when toggling) → that's known as "Race around condition".

To remove that condition :-
- By providing clock width < delay
- By using master-slave
- By doing edge triggering

Characteristic table → for all inputs, outputs are defined. it's tells about the characteristic of the flip flop.

Excitation table → By present and next state, we can defined the inputs.

Char table

J	K	Q_{n+1} (O/P)
0	0	(memory)
0	1	0
1	0	1
1	1	$\bar{Q}_n$ (toggle)

Excitation table

Ps	Ns	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

④ T- Flip Flop :
J & K are shorted

Date: 21-feb

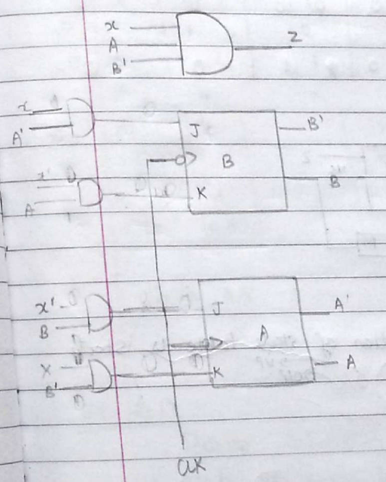
Char^c table

T	Q(n+1)
0	Q(n)
1	$\bar{Q}$(n)

excitation table

Ps	Ns	T
0	0	0
0	1	1
1	0	1
1	1	0

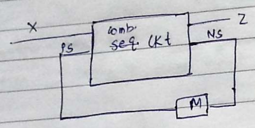
Example :



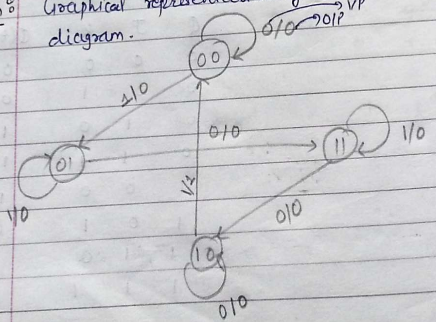
Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

State table:

PS A	PS B	NS x=0 A^+	NS x=0 B^+	NS x=1 A^+	NS x=1 B^+	O/P x=0	O/P x=1
0	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0
1	0	1	0	0	0	0	1
1	1	1	0	0	1	0	0



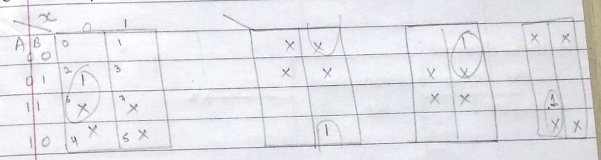
State Diagram: Graphical representation of state table is state diagram.



State equation:

O/P	PS (A B)	x	NS (A^+ B^+)	J_A	K_A	J_B	K_B
0	0 0	0	0 0	0	X	0	X
0	0 0	1	0 1	0	X	1	X
0	0 1	0	1 1	1	X	X	0
0	0 1	1	0 1	0	X	X	0
0	1 0	0	1 0	X	0	0	X
1	1 0	1	0 0	X	1	0	X
0	1 1	0	1 0	X	0	X	1
0	1 1	1	1 1	X	0	X	0

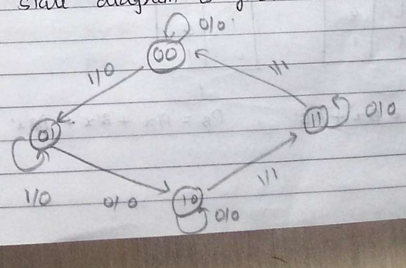
Excitation table



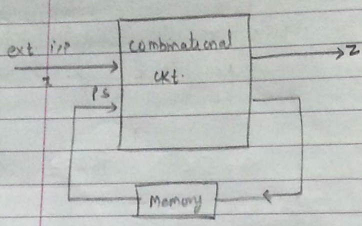
$J_A = B\bar{x}$ $K_A = \bar{B}x$ $J_B = \bar{A}x$ $K_B = A\bar{x}$

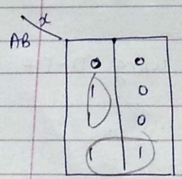
27-2-17

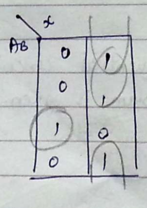
Design sequential circuit using D-F/F state diagram is given.



PS			NS		O/P		
A	B	x	A^+	B^+	z	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
1	0	0	1	0	0	1	0
1	0	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	1	1	0	0	1	0	0







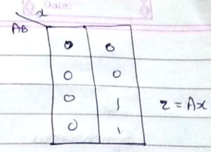
$D_A = Bx' + AB'$ $D_B = A'x + B'x + ABx'$

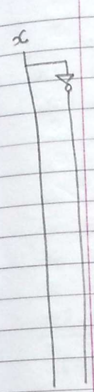
∴ state equation

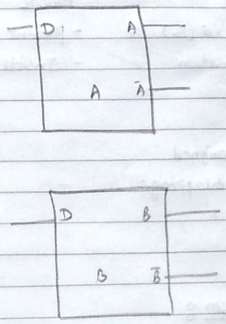
$D_A = Bx' + AB'$

$D_B = A'x + B'x + ABx'$

$z = Ax$







finite state machine :- irrespective of how many states are used, the total no of states present is finite.

Sequential Circuit

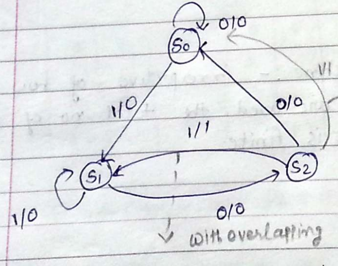
- Synchronous
  - Melay
    - present i/p (Ps)
    - o/p transition arrow
    - can be designed
      - with overlapping
      - without "
  - Moore
    - present state
    - Inside state

    } o/p depends on
- Asynchronous

Melay FSM :

$X = 0\,0\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,1\,0\,0$ → overlapping occurs

$Z = 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0$

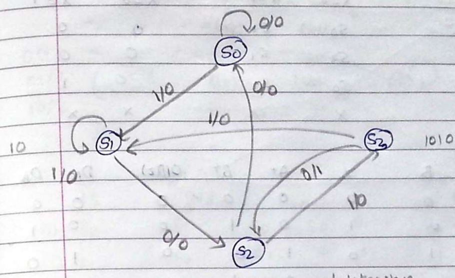


→ without overlapping (this means we come back to the initial state)

↓ with overlapping

(here, we come back to the state where 1st bit of sequence 101 was detected)

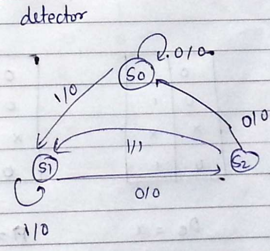
1010 pattern



100 → reset takes place

28-feb :

101 Sequence : detector



	PS	NS $x=0$	NS $x=1$	O/P $x=0$	O/P $x=1$
(00)	S_0	S_0 (00)	S_1 (01)	0	0
(01)	S_1	S_2 (10)	S_1 (01)	0	0
(10)	S_2	S_0 (00)	S_1 (01)	0	1
(11)		X	X	X	X

A	B	x	A^+	B^+	O/P(Z)	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

AB \ x

0	0
1	0
X	X
0	0

$D_A = Bx'$

0	1
0	1
X	X
0	1

$D_B = x$

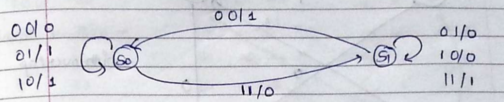
0	0
0	0
X	X
0	1

$Ax = Z$

Ex: Design a 1-bit serial adder using Mealy FSM.

$S_0 \rightarrow$ stands for no carry

$S_1 \rightarrow$ " " carry



	PS	NS 00	NS 01	NS 10	NS 11	O/P 00	O/P 01	O/P 10	O/P 11
(0)	S_0	S_0	S_0	S_0	S_1	0	1	1	0
(1)	S_1	S_0	S_1	S_1	S_1	1	0	0	1

PS	I/P A	I/P B	NS	O/P
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$\boxed{\text{Moore}}$

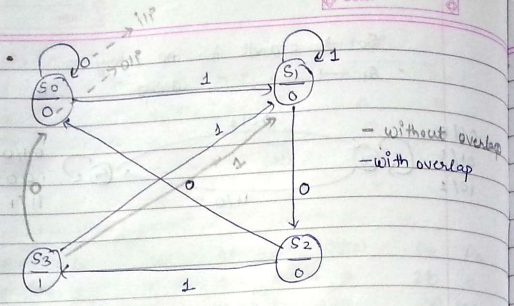
1010 – Mealy

1010 – Moore

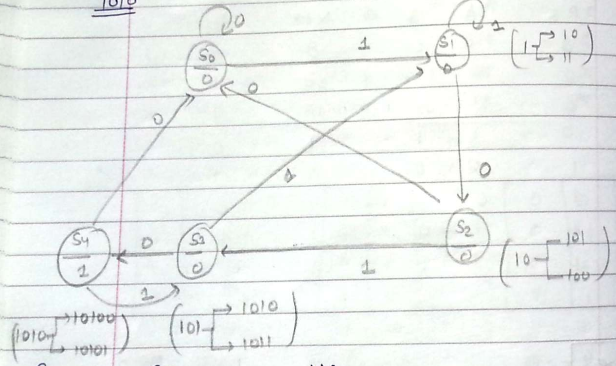
101 – moore

J. Roth → dyn code conversion

Manchester to NRZ



1010



for 101

PS		NS x=0	NS x=1	O/P
S_0	00	S_0	S_1	0
S_1	01	S_2	S_1	0
S_2	10	S_0	S_3	0
S_3	11	S_2	S_1	1

for 1010

	PS	NS x=0	NS x=1	O/P
000	S_0	S_0	S_1	0
001	S_1	S_2	S_1	0
010	S_2	S_0	S_3	0
011	S_3	S_4	S_1	0
100	S_4	S_0	S_3	1

PS A	B	x	NS A^+	B^+	O/P	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	0	1	1	0
1	1	1	0	1	1	0	1

for 1010

A	B	C	x	A^+	B^+	C^+	O/P	D_A	D_B	D_C
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	1
0	0	1	0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	1	0	0	1	1
0	1	1	0	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0	1
1	0	0	0	0	0	0	1	0	0	0

							1	0	1	1
1	0	0	1	0	1	1	x	x	x	x
1	0	1	0	x	x	x				
1	0	1	1	x	x	x				
1	1	0	0	x	x	x				
1	1	0	1	x	x	x				
1	1	1	0	x	x	x				
1	1	1	1	x	x	x				

Melay 1010

PS	NS x=0	NS x=1	O/P x=0	O/P x=1
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_3	0	0
S_3	S_2	S_1	1	0

PS A	PS B	i/p	NS A+	NS B+	O/P	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	0	1	1	0
1	1	1	0	1	0	0	1

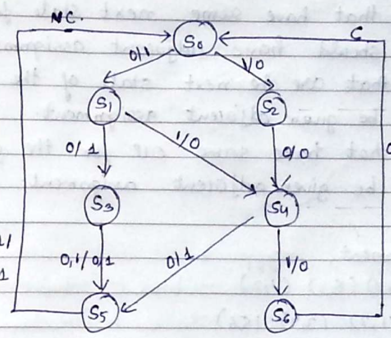
Date: 6-March

BCD - Excess3 Conversion:

$\left.\begin{matrix} 0000 \\ \vdots \\ 1111 \end{matrix}\right\}$ BCD

To convert in XS-3 form add 0011 to BCD no.

0/1
0011
0/1 → carry
0/1 → i/p
0011
0/1 → carry
0/1 → i/p
0011
0/1 → carry
0/1 → i/p
0011



1st bit of seq. detected

2nd bit of seq. detected

3rd bit detected

There is no possibility of taking i/p = 1 in carry side as the i/p BCD no. would be illegal.

- assume LSB bit enters first.
- 4th bit is detected, the circuit restarts, i.e., starts from S_0.

PS	NS x=0	NS x=1	O/P x=0	O/P x=1
S_0	S_1	S_2	1	0
S_1	S_3	S_4	1	0

S_2	S_4	S_4	0	1
S_3	S_5	S_5	0	1
S_4	S_5	S_6	1	0
S_5	S_0	S_0	0	1
S_6	S_0	X	1	X

Rule 1: states that have same next state for given input should have adjacent assignment.

Rule 2: States that are the next state of the same state should be given adjacent assignment.

Rule 3: States that have same O/P for the given i/p should be given adjacent assignment.

State Assignment

Rule 1 $\longrightarrow$ $\underset{x=0}{(3,4)}$ $\underset{x=0}{(5,6)}$ $\underset{x=1}{(1,2)}$ → look in colum format in NS

Rule 2 $\longrightarrow$ (1,2) (3,4) (5,6) → " Row " " NS

Rule 3 $\longrightarrow$ $\underset{x=0}{(0,1,4,6)}$ $\underset{x=1}{(2,3,5)}$ → look in column format in O/P

S_1	S_2
	S_3
	S_4
S_6	S_5

Rule - 1, 2

S_0	S_1
	S_2
S_3	S_5
S_4	S_6

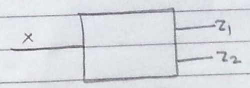
Rule-3

Ex: A moore seq. ckt has 1 i/p and 1 o/p when i/p seq. 011 occurs o/p becomes 1 and remains 1 until the seq. 011 occurs again. In which case the o/p returns to 0. The o/p then remains 0 until 011 occurs 3rd time and this goes on.

Sol:

Ex:- A seq. ckt has 1 i/p 2 o/p. O/P $Z_1 = 1$ occurs everytime the i/p seq. 010 is completed provided that the seq. 100 has never occured. $Z_2 = 1$ occurs everytime the i/p seq. 100 is completed. Note that $Z_2 = 1$ o/p has occured then $Z_1 = 1$ can never occur but not vice-versa is true. Design melay state Diagram.

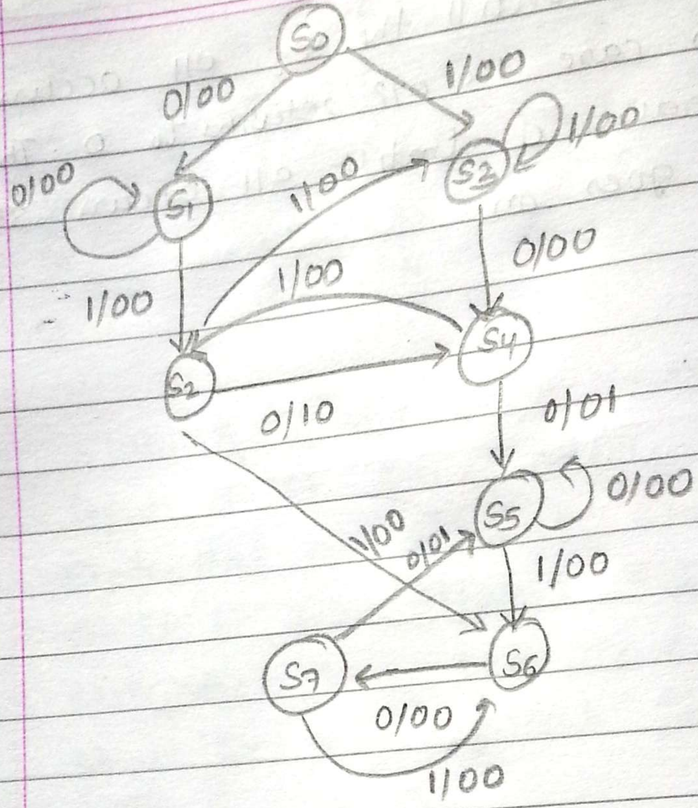
Sol:



$Z_1 = 1 \longrightarrow 010$

$Z_2 = 1 \longrightarrow 100$

Date: 10-March



Example: A syn. ckt. having 1 i/p and 1 o/p
output 1 and remains 1 . There after
atleast once 1 and three 0 has occurs
regardless the order of occurance. for this problem
Design moore state diagram

Sol:

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