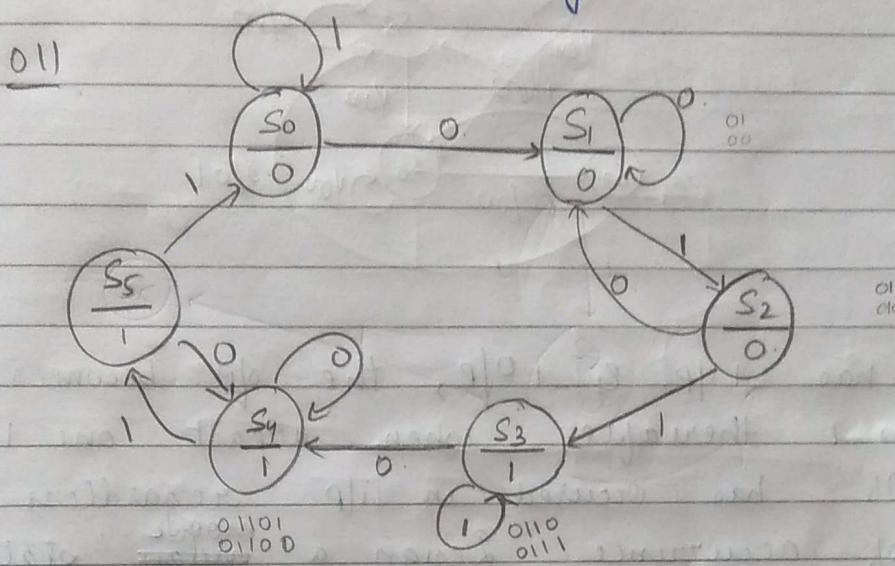
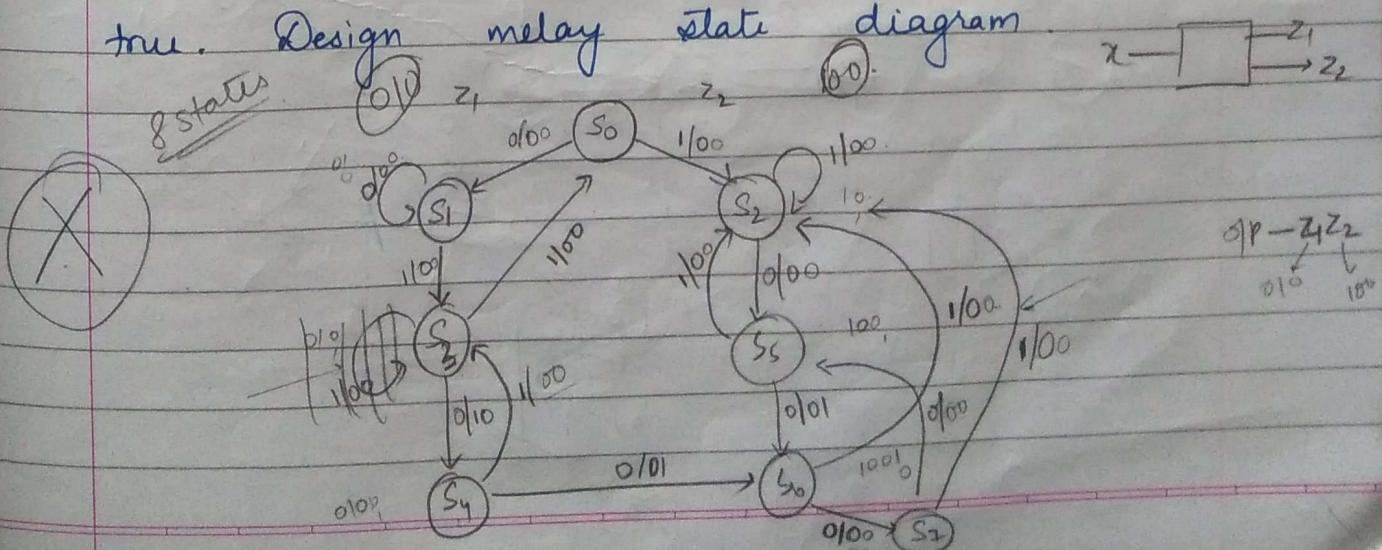
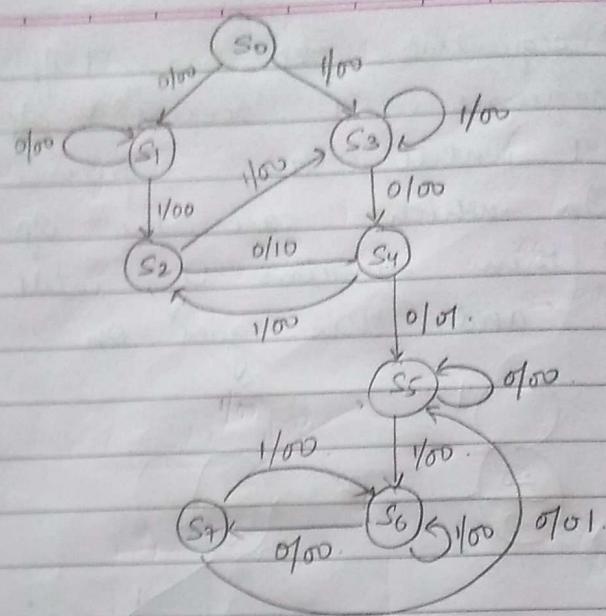


eg A moore sequential ckt has 1 i/p and 1 o/p when input sequence 011 occurs o/p becomes 1 and remains 1 until the sequence 011 occurs again. In which case the o/p returns to 0. The o/p then remains 0 until 011 occurs 3rd time. and this goes on.



eg. A sequential ckt has 1 i/p 2 o/p. O/p $z_1=1$ occurs everytime the i/p sequence 010 is completed provided that the sequence 100 has never occurred. $z_2=1$ occurs everytime the i/p sequence 100 is completed. Note that $z_2=1$ o/p has occurred when $z_1=1$ can never occur but not vice-versa is true. Design delay state diagram.



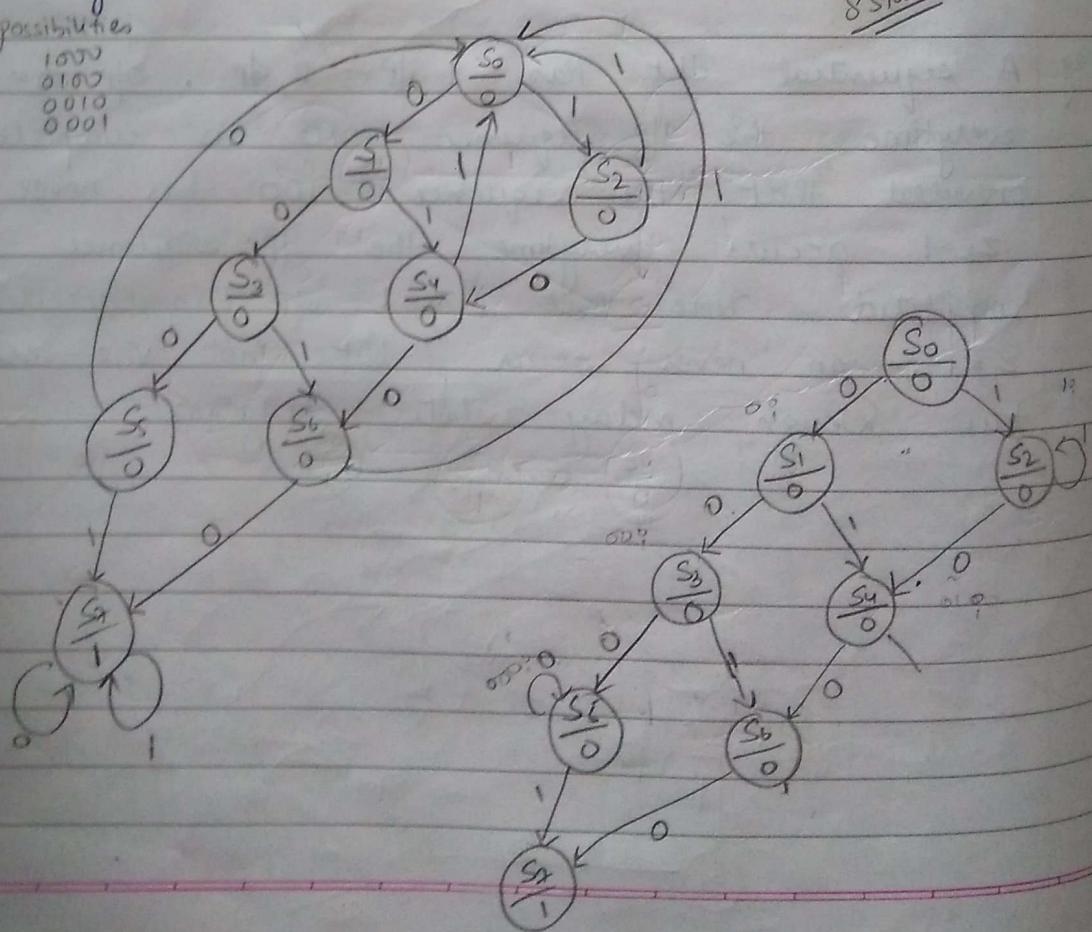


eg. A cktr has 1/I/P & 1/O/P, the O/P becomes 1 & remains 1 thereafter when atleast one 1's and three 0's has occurred in I/P regardless the order of occurrence. Design a ~~multiple~~ state diagram.

possibilities

1000
0100
0010
0001

8 states



IMPLICATION CHART

PS	NS		OP	
	X=0	X=1	X=0	X=1
a	c	f	0	0
b	d	e	0	0
c	Ka.	g	0	0
d	b	g	0	0
e	e	b	0	1
f	f	a	0	1
g	c	g	0	1
h	c	f	0	0

← can be removed

Here, $a=h$ because the NS and OP for each of them is same. \Rightarrow h state can be removed.

Equivalence Theorem \rightarrow For given IP's, if OP's are same of both ~~same~~ and NS could be made same.

eg.

$a=b$ is possible bcoz OP's are same for both a & b IP's.

But if $c=d$ & $f=e$ then only $a=b$
 $\Rightarrow c=d$ and $f=e$ ← are implied cond'n's/pair.

	a	b	c	d	e	f
b	c-a f-e					
c	x-d	x-g				
d	x-b	x-f	a-b			
e						
f					a-b	
g					x-e x-g	x-f x-g

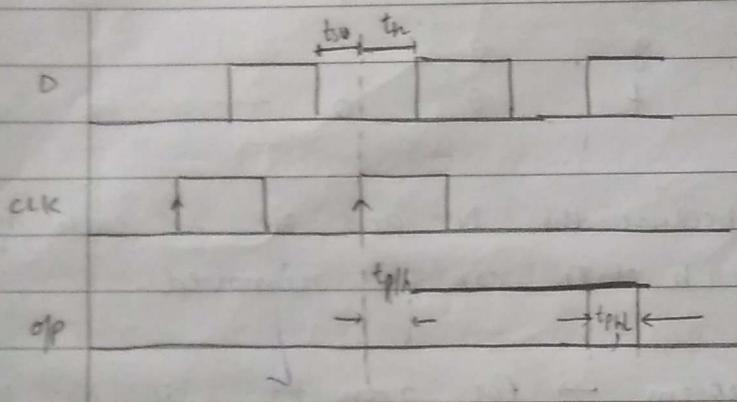
← implication chart

As the combination of
 $c=f$ is not possible
 \Rightarrow so $g=f$ is not
possible. (similarly for others)
(Start checking this
from the end)

i.e. Here, $a=b$ $c=d$ $e=f$ is possible \Rightarrow state table gets reduced

PS	NS		OP		Reduction table.
	$x=0$	$x=1$	$x=0$	$x=1$	Reduction table.
a	c	c	0	0	
c	a	g	0	0	As we get $a=b$ $c=d$
e	e	a	0	1	$e=f \Rightarrow$ we can remove
g	c	g	0	1	the rows ⇒ states get reduced

TIMING EQUATIONS



• $t_{phl} \rightarrow$ Propagation time to move from low to high.

• $t_{phl} \rightarrow$ " high to low.

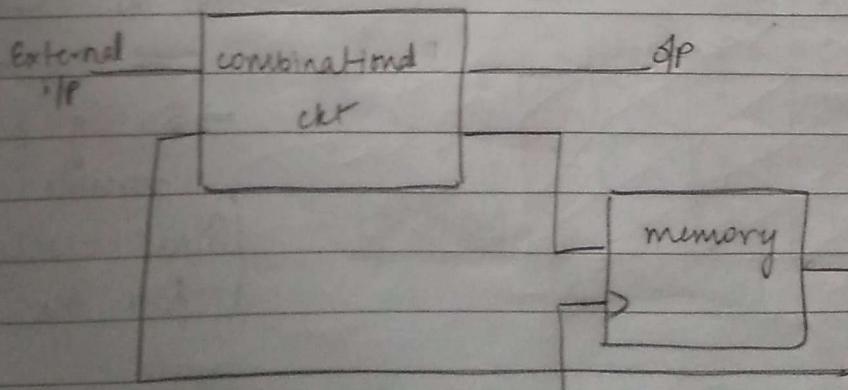
$$\max(t_{phh}, t_{phl}) = t_{pmax}$$

$$\min(t_{phh}, t_{phl}) = t_{pmin}.$$

Set-up time \rightarrow Time for which I/P must be stable before arrival of CLK.

Hold time \rightarrow Time for which I/P must be stable after arrival of CLK.

Metastable state \rightarrow When I/P changes b/w set-up and hold time \Rightarrow O/P ~~become~~ changes which is not required.



1) Clock pulse should be long enough than set-up time

$$t_{CK} \geq t_{Cmax} + t_{SU} + t_{Pmax}$$

↑
Propagation delay ↑
In combinational ckt Set-up
time.

2) For hold condition of FF $t_{Pmin} + t_{Cmin} \geq t_h$

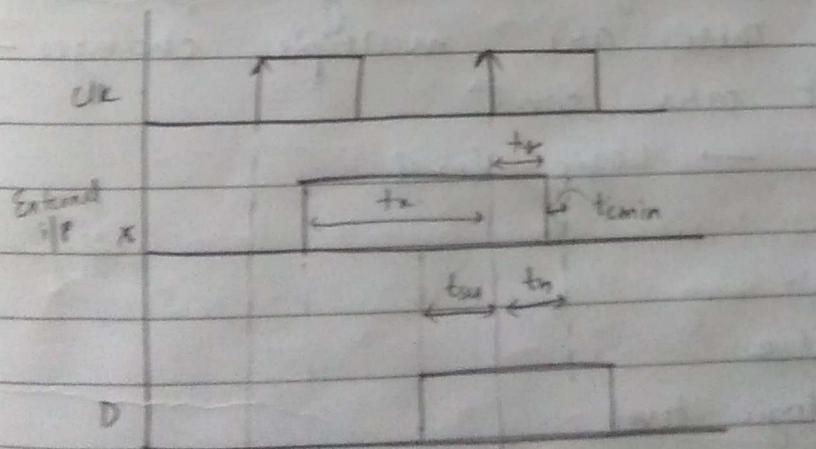
3) For external i/p \rightarrow next i/p should occur once it has propagated through combinational ckt and has passed the set-up time of FF

$$t_x \geq t_{Cmax} + t_{SU}$$

x → previous i/p
y → next i/p

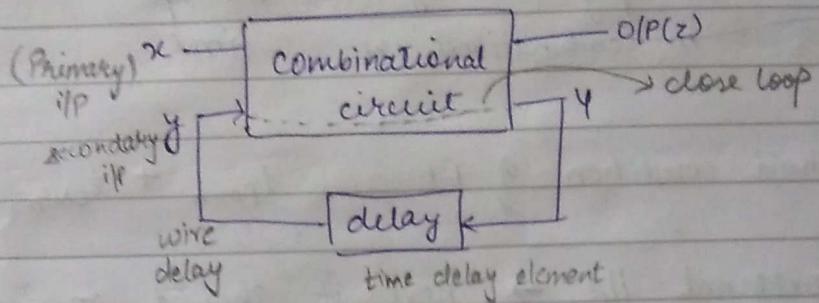
4) ~~$t_{CK} \geq t_{Pmax} + t_{Cmax}$~~ $t_y + t_{Cmin} \geq t_h$

$$\text{Set-up margin} = t_{CK} - (t_{Cmax} + t_{SO} + t_{Pmax})$$



ASYNC. SEQUENTIAL CIRCUIT

There is no memory element present in async sequential circuit.



- $y = Y \leftarrow$ stable state condition
- Once stable state condition is achieved, then only primary i/p is changed
- Doesn't depend on clock pulse
- Faster working

TYPES

(1) Fundamental mode :-

- At any point of time only 1 i/p can change else if there are multiple changes \rightarrow then its don't care condⁿ
- i/p & o/p \rightarrow logic level

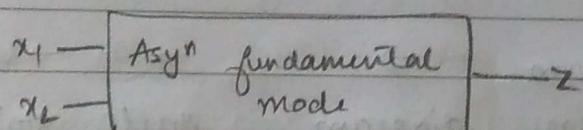
(2) Pulse mode :-

- i/p \rightarrow pulse
- o/p \rightarrow logic level

eg A certain fundamental mode design has 2 i/p x_1, x_2 and 1 o/p z . The o/p z has to change from 0 to 1 only when x_2 changes from 0 to 1 while x_1 is already 1. The o/p z has to change from 1 to 0 only when x_1 changes from 1 to 0 while x_2 is already 1. For given statement (1) find minimum row merged flow table

- (2) find valid secondary assignment
- (3) make the o/p fast and flicker free
- (4) obtain hazard free excitation & o/p func.

Step 1: Write word statement list i/p & o/p conditions



$$x_1 x_2 = 10 \rightarrow 11 \Rightarrow z = 0 \rightarrow 1$$

$$x_1 x_2 = 11 \rightarrow 01 \Rightarrow z = 1 \rightarrow 0$$

Step 2: Design a flow table

To start the design, always assume initial stable state 1 of the machine with all i/p are 0. Then we make i/p changes as per the sequences that are mentioned in the word statement. For every change in i/p introduce new stable state by opening a new row and entering a circled number indicating new stable state in the column corresponding to the changed i/p. In this process indicate transitory state in the same column of previous row. Once you exhaust all possible conditions which are listed

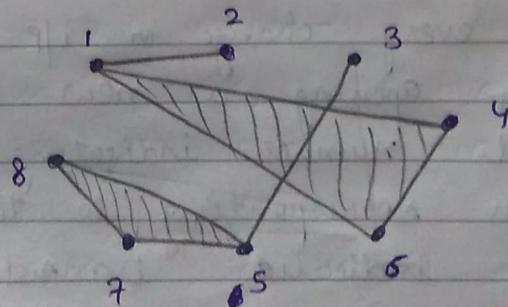
Page No. _____
Date _____

in word statement then go for the rest of the possible combinations & complete the table.

	$x_1 x_2$	00	01	11	10	
listed transition	1	(1,0)	4	-	2	Partial primitive flow table
state	2	1	-	3	(2,0)	transition state
	3	-	4	(3,1)	5	of P
	4	1	(4,0)	6	-	
	5	7	-	3	(5,1)	
	6	-	4	(6,0)	2	
	7	(7,1)	8	-	5	
	8	7	(8,1)	3	-	

Step 3 Merger diagram

- i To draw merger diagram, stable state and unstable state and '-' can be merged, and result in stable state.
 - ii A stable state and its transition state merge into stable state.
- Row 1 and 2 can be merged
 - Row 2 and 3 cannot be merged bcoz (2,0) & (5,1) do not follow any rule (i) & (ii)



larger size merger is given priority.

(1, 4, 6)

2

3

(5, 7, 8)

Rows y ₁ y ₂		x ₁ x ₂ , N _s (y ₁ y ₂), z			
		00	01	11	10
a	①, 0	④, 0	⑥, 0		2
b	1	-	3		②, 0
c	-	4	③, 1		5
d	⑦, 1	⑧, 1	3		⑤, 1

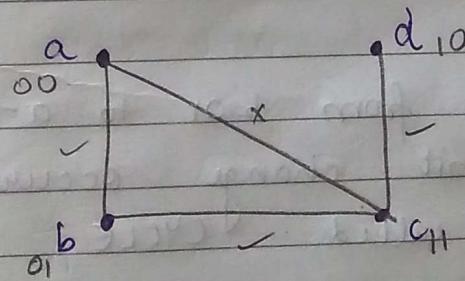
← Merge flow table.

Step 4. Adjency diagram & secondary variable assignment.

If any one of the column of 2 rows have a stable state and its unstable state then join them.

eg:- in a and b b and c

①, 0 & 1 ③, 1 & 3



States are assigned randomly, i.e., 00, 01, 10, 11 states.

y₁y₂ = y₁y₂ → for stable state.

so for a = $\frac{00}{y_1y_2}$ → then all stable states in that row would be $\frac{00}{y_1y_2}$.

$y_1 y_2$	$x_1 x_2, N(4,4_2), z$			
00	01	10	10	
a=00	00, 0	00, 0	00, 0	01
b=01	00	- (00)	11	01, 0
c=11	-	00 (01)	11, 1	10
d=10	00, 1	00, 1	11	01, 1

$$y_1 y_2 \rightarrow z = 1 \rightarrow 0.$$

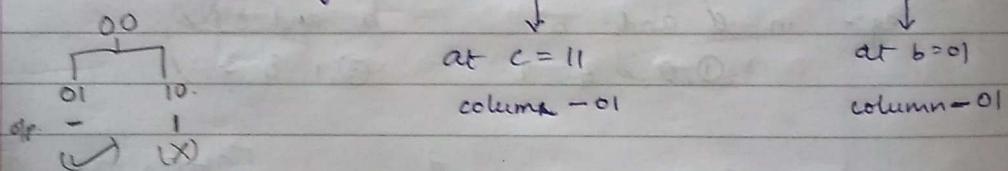
- Going from $00, 00, 00, 00$ to $01, 01, 01, 01$. 11 to 01 \rightarrow O/P change from 1 to 0 but $y_1 y_2$ changes from c to a i.e., 11 to 00 \Rightarrow 2-bit change \Rightarrow Race Condition.

- If $y_1 < y_2$ (propagation) $\Rightarrow y_1 y_2 = 11 \rightarrow 10$
 \Rightarrow o/p change 1 to 1 \Rightarrow Critical Race Condition.

- $y_1 > y_2 \Rightarrow y_1 y_2 = 11 \rightarrow 01$

so at $b = 01$ there is nothing \oplus

\Rightarrow we change $'00'$ to $'01'$ and $'-'$ to $'00'$



- So, now moving from $b=01$ to $a=00$ is possible as 1-bit change occurs.
 This process is called CYCLE.

- Assume $d=10$ & column -01 had 10 as unstable state then $y_1 < y_2$ condition would be possible as transition from $d=10$ to $a=00$ would have 1-bit change.

\Rightarrow Called as Non-Critical case

Only 1 stable state is present in column

Step 5

Step 6

$$10 \rightarrow 11 \rightarrow z=0 \rightarrow 1$$

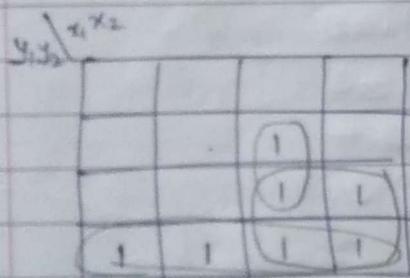
- 10 to 11 is possible as $b=01$ to $c=11$ is valid
 & q.p. changes from 0 to 1

(reg. loss)
from

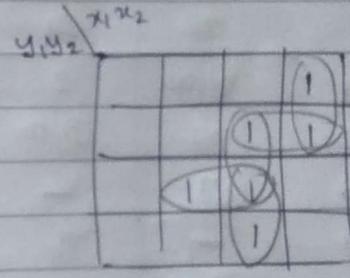
Step 5 Synthesis Equation

		$x_1x_2 + N(y_1y_2), z$			
		00	01	10	11
y ₁ y ₂		(00,0)	(00,0)	(00,0)	(01,0)
a=00					
b=01		00,0	00,0	11,1	(01,0)
c=11		-	01,0	(11,1)	10,1
d=20		(10,1)	(10,1)	11,1	(10,1)

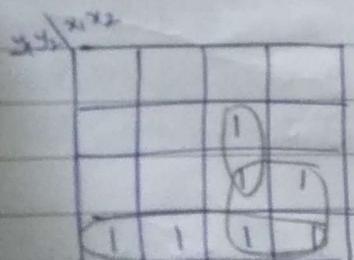
← add of pls
to unstable states



$$Y_1 = x_1y_1 + y_1y_2' + x_1x_2y_2$$



$$Y_2 = x_1x_2y_1 + x_1x_2'y_2 + x_1x_2'y_2' + x_2y_1y_2 + x_1y_1'y_2$$

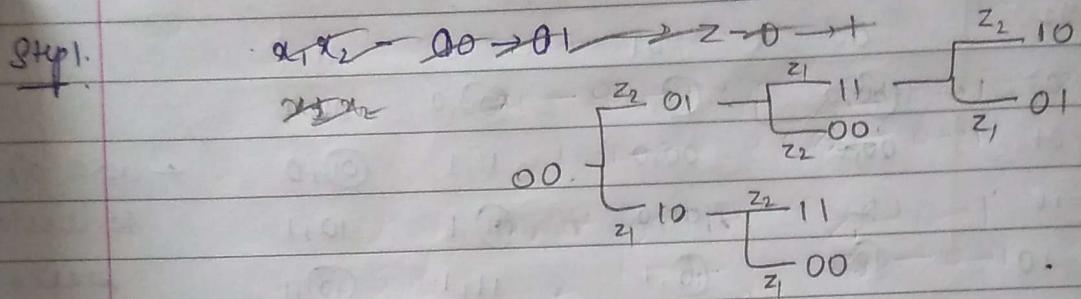


$$Z = x_1y_1 + y_1y_2' + x_1x_2y_2$$

Step 6 Draw the circuit

eg) A certain ^{async.} sequential mode of I/P x_1 x_2 and O/P z_1 z_2 . The O/P z_i where $i=1,2$ has to assume logic 1 level whenever I/P that change last was x_i ($i=1,2$) and 0 at all other times. Also identify that whether machine has power on state or not.

~~Step 1~~



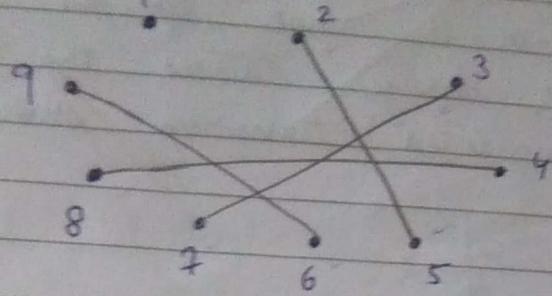
Step 2

		00	01	11	10
00	①, 00	2	-	3	
01	5	②, 01	4	-	
11	7	-	6	③, 10	
10	-	8	④, 10	9	
00	⑤, 01	2	-	3	
01	-	8	⑥, 01	9	
11	⑦, 10	2	-	3	
10	5	⑧, 10	4	-	
00	7	-	6	⑨, 01	

As '00' - x_1x_2 does not come back \Rightarrow machine has power on state.

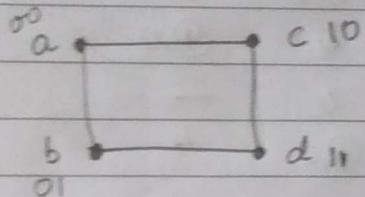
Secondly, x_1x_2 - '00' is been assumed for a start.

Step 3



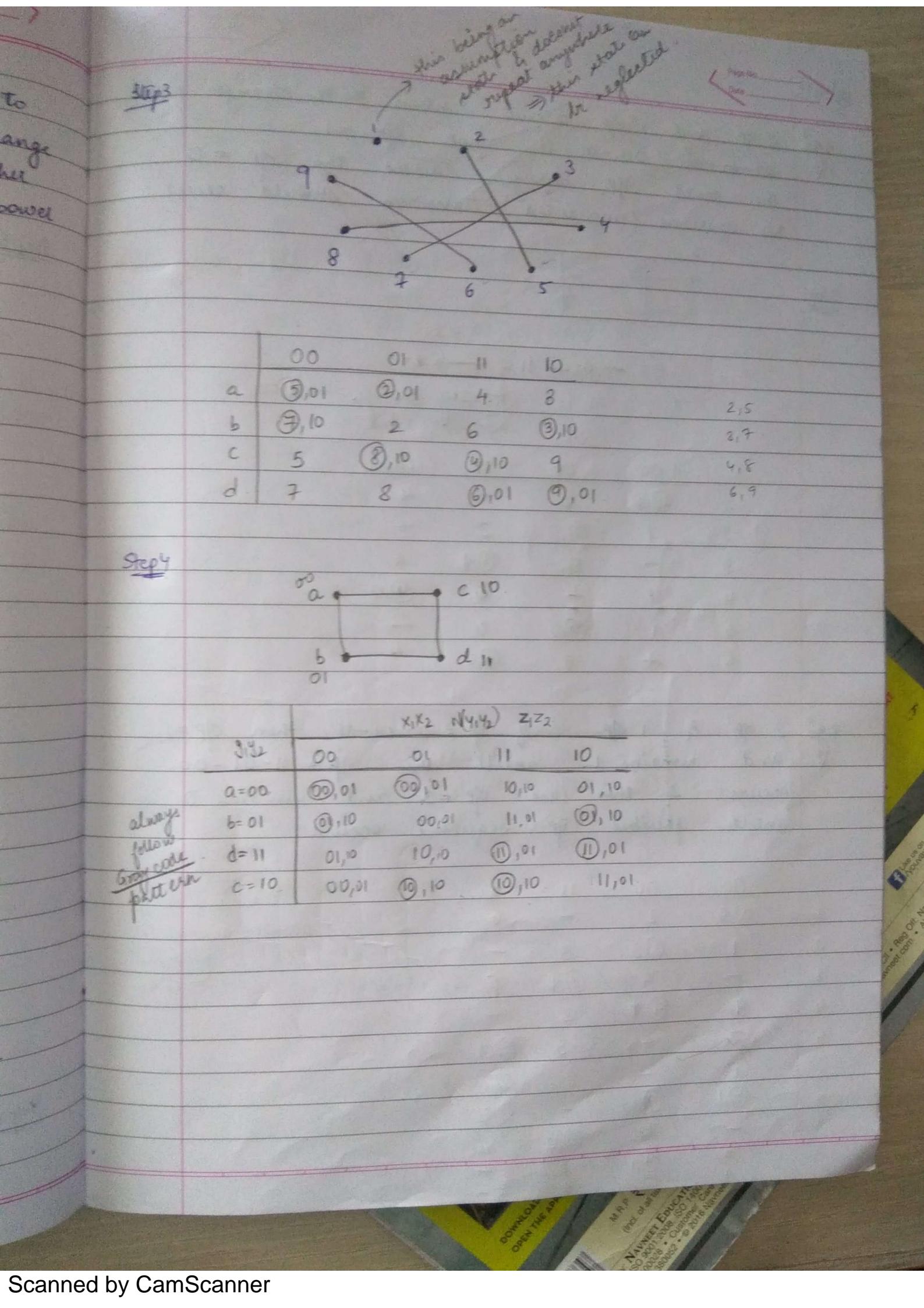
	00	01	11	10	
a	⑤,01	②,01	4	3	2,5
b	⑦,10	2	6	③,10	2,7
c	5	⑧,10	④,10	9	4,8
d	7	8	⑥,01	⑨,01	6,9

Step 4



$x_1x_2 \quad y_1y_2 \quad z_1z_2$	00	01	11	10
$a=00$	⑩,01	⑩,01	10,10	01,10
$b=01$	①,10	00,01	11,01	②,10
$d=11$	01,10	10,00	⑪,01	⑪,01
$c=10$	00,01	⑩,10	⑩,10	11,01

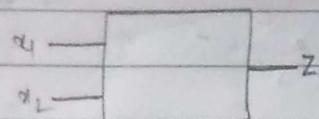
always
follows
Gray code
pattern



3/4/17

Page No.
Date

e.g. 2 i/p and 1 o/p circuit - when x_1, x_2 changes from 00 to 01 00-01-11 then o/p = 1
 The next i/p change causes the o/p = 0
 No spurious (unwanted) transition should occur

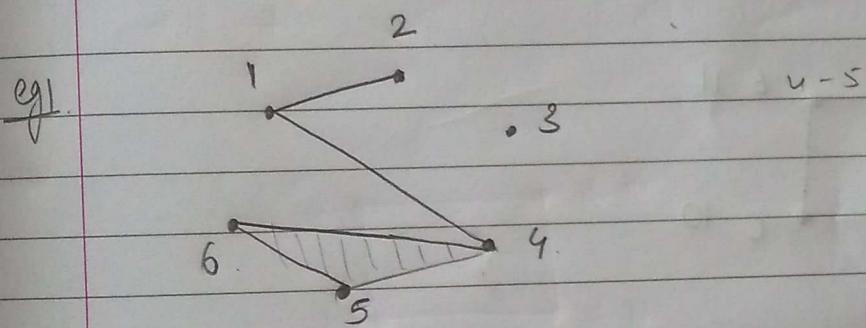
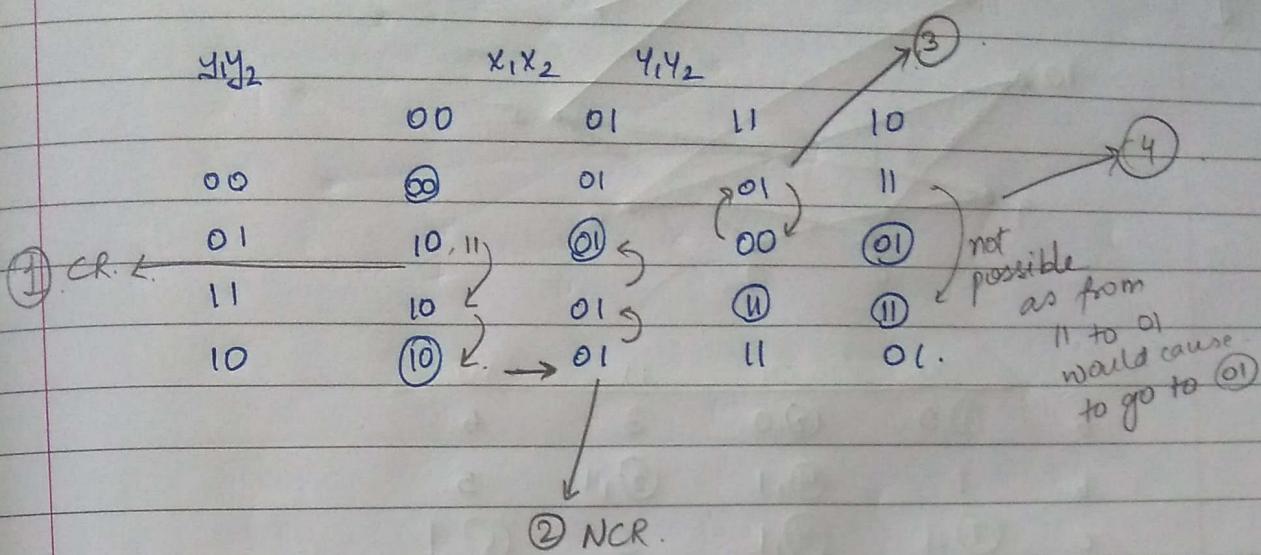
Step 100 - 01 - 11 $\rightarrow z = 1$.any change $\rightarrow z = 0$.

	00	01	11	10
1	①,0	2	-	4
2	1	②,0	3	-
3	-	5	③,1	4
4	1	-	6	④,0
5	1	⑤,0	6	-
6	-	5	⑥,0	4

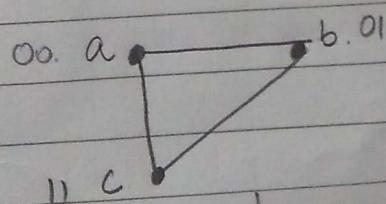
e.g. 2 i/p & 1 o/p If i/p = 00-01-11 then o/p = 1 and remains 1 until the i/p = 11-01-00 occurs. In this case o/p = 0 and remains 0 until previous i/p sequence occurs.

	00	01	11	10	11	01
1	①,0	2	-	6	11	10
2	1	②,0	3	-	00-01-11	$\rightarrow z=1$
3	-	4	③,1	5	01	$\rightarrow z=0$
4	1	④,1	3	-	10	$\rightarrow z=1$
5	7	-	3	⑤,1	10	$\rightarrow z=0$
6	1	-	8	⑥,0	10	$\rightarrow z=1$
7	⑦,1	9	-	5	10	$\rightarrow z=0$
8	-	10	⑧,0	6	10	$\rightarrow z=1$
9	7	⑨,1	3	-	10	$\rightarrow z=0$
10	1	⑩,0	8	-	10	$\rightarrow z=1$

Rare conditions

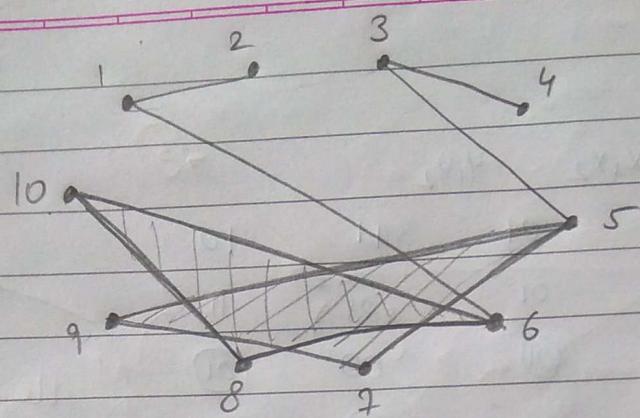


	00	01	11	10
a	0, 1	0, 1	3	4
b	1	5, 0	6, 0	4, 0
c	-	5	3, 1	4

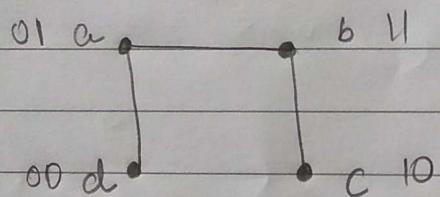


$y_1 y_2$	$x_1 x_2 \quad N(y_1 y_2) =$			
	00	01	11	10
$a = 00$	0, 0	0, 0	11	0, 1
$b = 01$	0, 0	0, 1	0, 0	0, 1
$c = 11$	-	0, 1	11, 1	0, 1

eg 2.



	00	01	11	10
a	①,0	②,0	③	6
b	1	⑨,1	③,1	5
c	⑦,1	⑨,1	3	⑤,1
d	1	⑩,0	⑧,0	⑥,0



	00	01	11	10
d = 00	01,0	00,0	00,0	00,0
a = 01	①,0	①,0	11,1	00,0
b = 11	01,0	11,1	11,1	10,1
c = 10	⑩,1	⑩,1	11,1	⑩,1

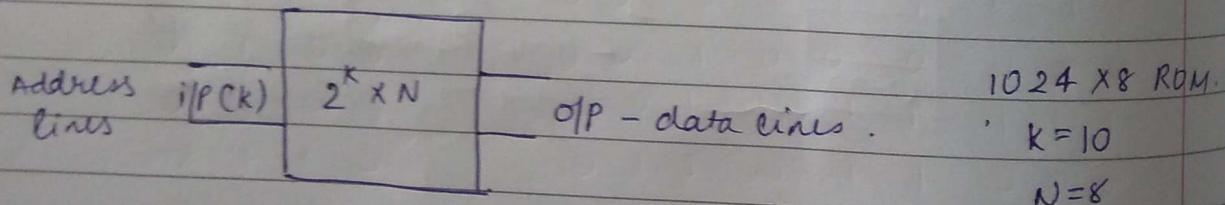
10/4/17

PLD

- PC \rightarrow hardware is software controllable
 \rightarrow hardware cannot be changed
 - In PLD, hardware can be changed
 - Types
 - ① Field Programmable.
 - SPLD
 - CPLD
 - FPGA
 - ② Factory Programmable.
 - PROM
 - PAL
 - PLA
 - MGA (Mask Gate Array)
 - ROM.
- Both are one time programmable device.
- SPLD \rightarrow Simple PLD
 - CPLD \rightarrow Complex PLD
 - FPGA \rightarrow Field programmable Gate Array.
 - PROM \rightarrow can be programmed many times.

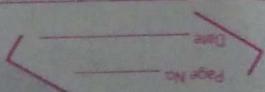
ROM

- does not have MUX, flip flop, shift register. In fact all these are present in RAM!
- RAM \rightarrow cannot change the hardware

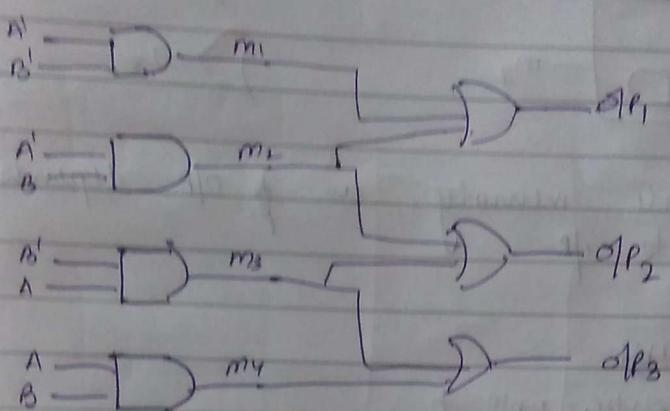


• Used to store data & design circuit
Steps to design ROM -

- ① How many i/p & o/p
- ② Truth table
- ③ Connections



- k I/P lines $\rightarrow 2^k$ AND gates are present/required.
 \downarrow
 used as 2^k address lines are to be accessed.
- $N \rightarrow m$: 2^m OR gates required.
 $2^m \times 2^k$: ROM



~~if~~ 3 bit I/P \Rightarrow (I/P^2)

8 I/P $\rightarrow 2^3 = 8 \rightarrow$ AND gates

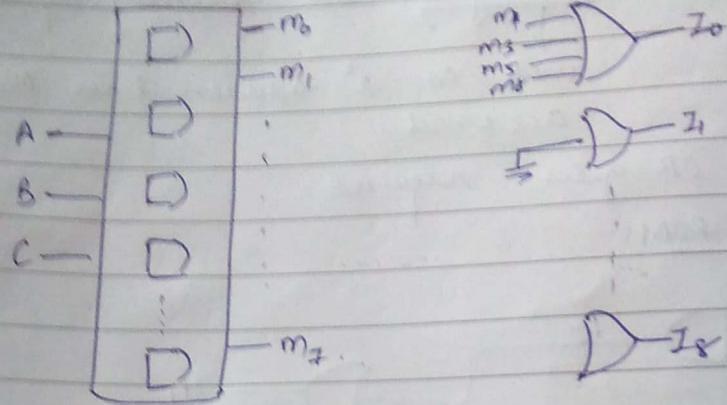
\hookrightarrow max no. possible = 111 \rightarrow i.e. 7.

$$2^7 = 49$$

$\therefore 6$ O/P $\rightarrow 6$ OR gates.

A	B	C	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

\uparrow
 data generated according
 to I/P.



- Non-rotatable memory as O/P depends on power supplied provided to i/p

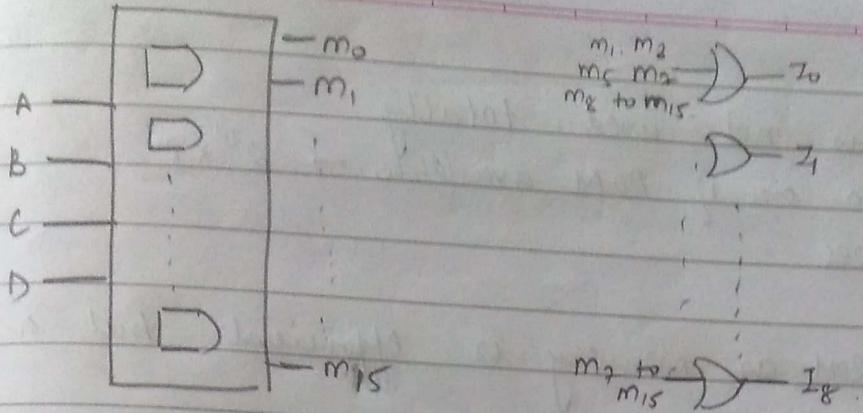
e.g. i/p → 4 bit
o/p → cube upto 7
otherwise all 1

4 bit → 2⁴ → 16 AND

7³ = 343 → 9 OR gates.

	8	7	6	5	4	3	2	1	0
0 0 0 0	0	0	0	0	0	0	0	0	0
0 0 0 1	0	0	0	0	0	0	0	0	1
0 0 1 0	0	0	0	0	0	0	1	0	0
0 0 1 1	0	0	0	0	1	1	0	1	1
0 1 0 0	0	0	1	0	0	0	0	0	0
0 1 0 1	0	0	1	1	1	1	1	0	1
0 1 1 0	0	1	1	0	1	1	0	0	0
0 1 1 1	1	0	1	0	1	0	1	1	1
1 0 0 0	1	0	0	1	0	1	0	1	0
1 0 0 1	0	1	1	1	1	1	0	0	1
1 0 1 0	1	0	1	0	1	0	1	1	0
1 0 1 1	1	1	1	1	1	1	1	1	1
1 1 0 0	1	1	0	0	0	0	0	0	0
1 1 0 1	1	1	0	1	1	1	1	1	1
1 1 1 0	1	1	1	0	1	1	1	0	0
1 1 1 1	1	1	1	1	0	1	1	1	1

Page No. Date



$$F_1 = \sum (0, 1, 2, 3, 4)$$

$$F_2 = \sum (1, 2, 3, 6, 8)$$

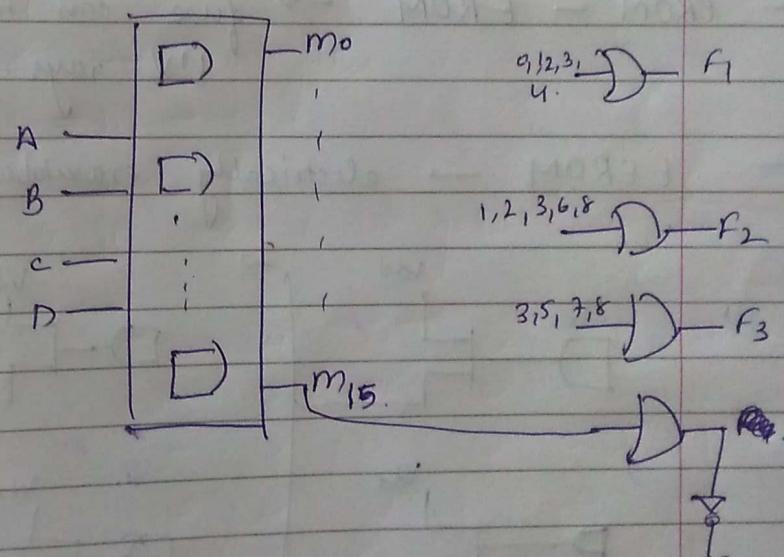
$$F_3 = \sum (3, 5, 7, 8)$$

$$F_4 = \sum (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)$$

$14 \rightarrow$ near to $2^4 \Rightarrow 16$ AND gates. $\rightarrow 4$ I/Ps.

$F_1, F_2, F_3, F_4 \Rightarrow 4$ O/Ps gates.

0	0	/	0	0
0	0	/	0	1
0	0	/	1	0
0	0	/	1	1
0	1	/	0	0
0	1	/	0	1
0	1	/	1	0
0	1	/	1	1
1	0	/	0	0
1	0	/	0	1
1	0	/	1	0
1	0	/	1	1
1	1	/	0	0
1	1	/	0	1
1	1	/	1	0
1	1	/	1	1



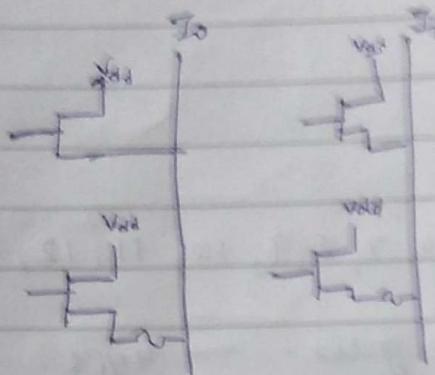
\Rightarrow minterms required $\rightarrow m_0 \text{ to } m_8, m_{15}$
 $\Rightarrow 10$ AND gates

This can be used if only
AND & OR gates are used
 \rightarrow not ROM.

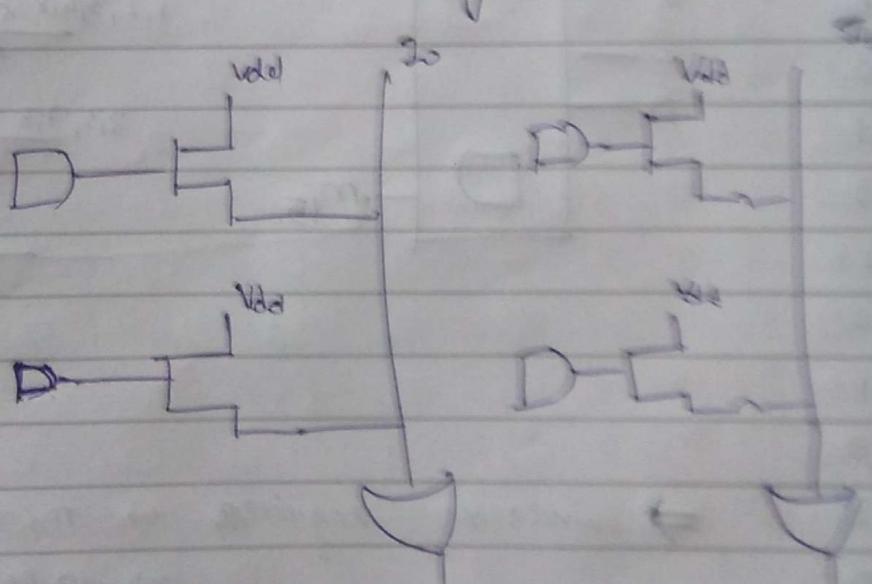
Disadvantages

- hardware is not used totally
minimum size of ROM available $\rightarrow 2^{12} \times 8$
- Uses power

\therefore Therefore, ROM as memory is efficient but when
it's not.



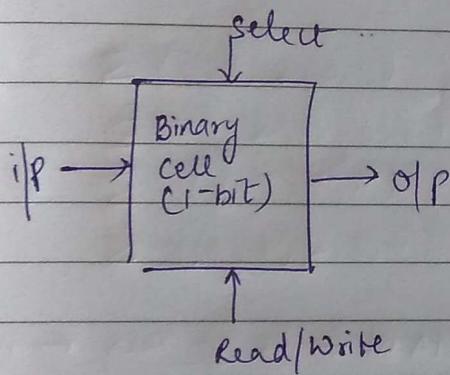
- PROM - EROM \rightarrow fuse can be joined again by UV rays.
- EEPROM \rightarrow electrically erasable



RAM

Read/Write = 1 } → then ckt works (assumed)
chip select>Select = 1 }
 } → Read operation takes place.

Read/Write = 0. } → Write Operation takes place.
Select = 1 }



DRAM → Refresh cycle is req. to discharge
capacitor totally.
less power consumption.