

Final

TOPIC NAME: Lecture 7

DAY: _____

TIME: _____

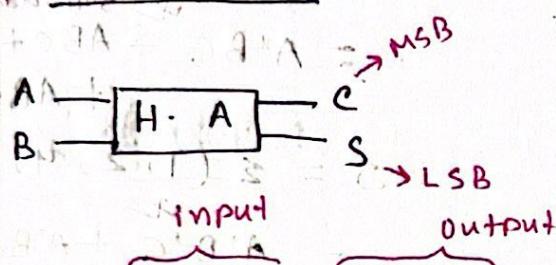
DATE: 06/4/25

Two classes of logic circuits: combinational

- combinational
- sequential

Combinational circuit:

Half Adder, (2 bit + single bit sum)



A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0

$$\begin{array}{r}
 0 \\
 + 0 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 + 0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 01 \\
 + 10 \\
 \hline
 10
 \end{array}$$

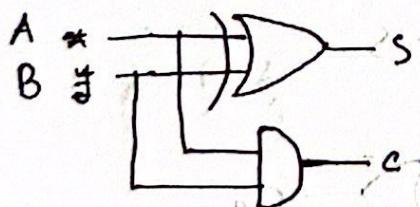
sum
carry

$$C = \Sigma(3) = AB$$

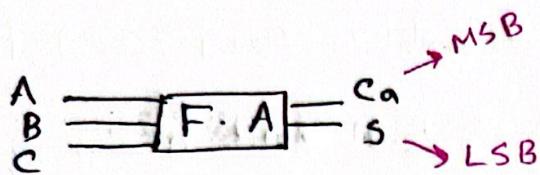
$$S = \Sigma(1, 2)$$

$$= A'B + AB'$$

$$= A \oplus B$$



Full Adder (3, 3) single bit (2nd) standard output



Output determine

1

$$\begin{array}{r} 1 \\ + 1 \\ \hline 11 \end{array}$$

	A	B	C	C_{out}	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

$C_{out} = \sum(3, 5, 6, 7)$

~~full adder~~

$$= A'B'C + AB'C + ABC'$$

$$+ ABC$$

$S = \sum(1, 2, 4, 7)$

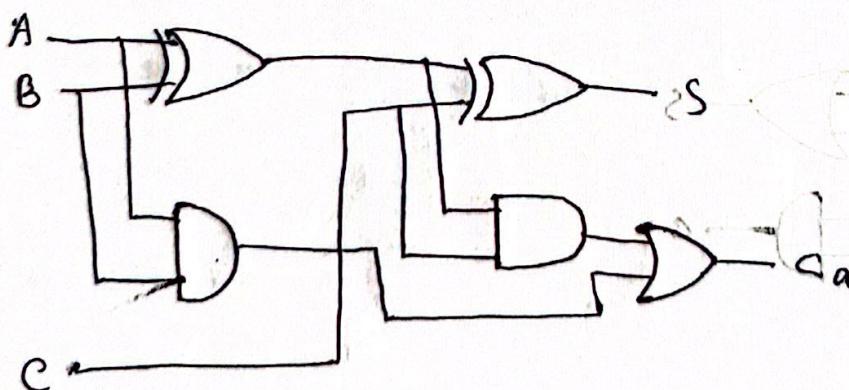
~~inverter~~

$$= A'B'C + A'BC'$$

$$+ AB'C' + ABC$$

$$= A \oplus B \oplus C$$

~~half adder use this
full adder draw~~



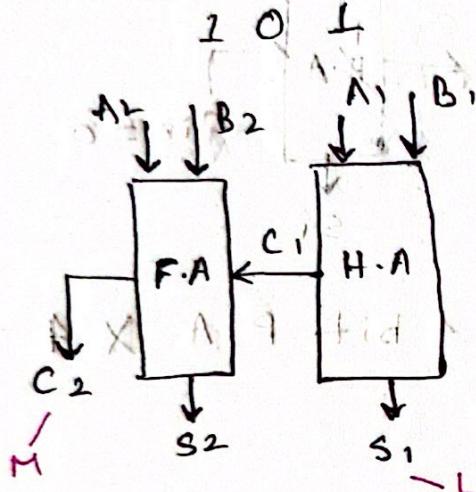
Parallel Adder

$$A = 1 \ 1$$

$$1 \ 1$$

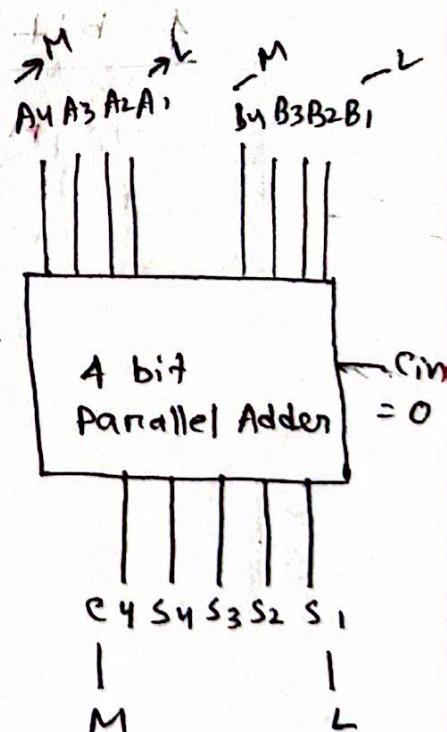
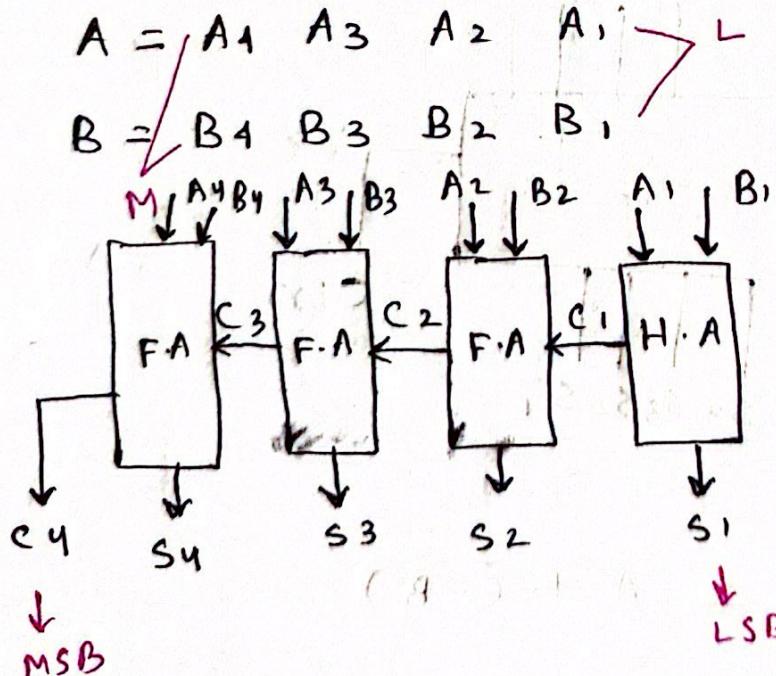
$$B = 1 \ 0$$

$$\begin{array}{r} + 0 \ 1 \\ \hline 1 \ 0 \ 0 \end{array}$$



$$A = A_2 A_1$$

$$B = B_2 B_1$$

4 bit Parallel Adder

TOPIC NAME : _____

DAY : _____

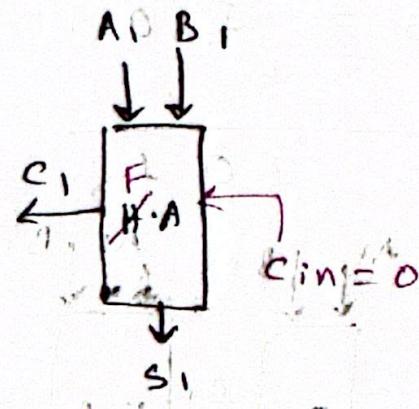
TIME : _____

DATE : / /

IC use का॒ट्टा॑ cost analysis राज्यकालीन

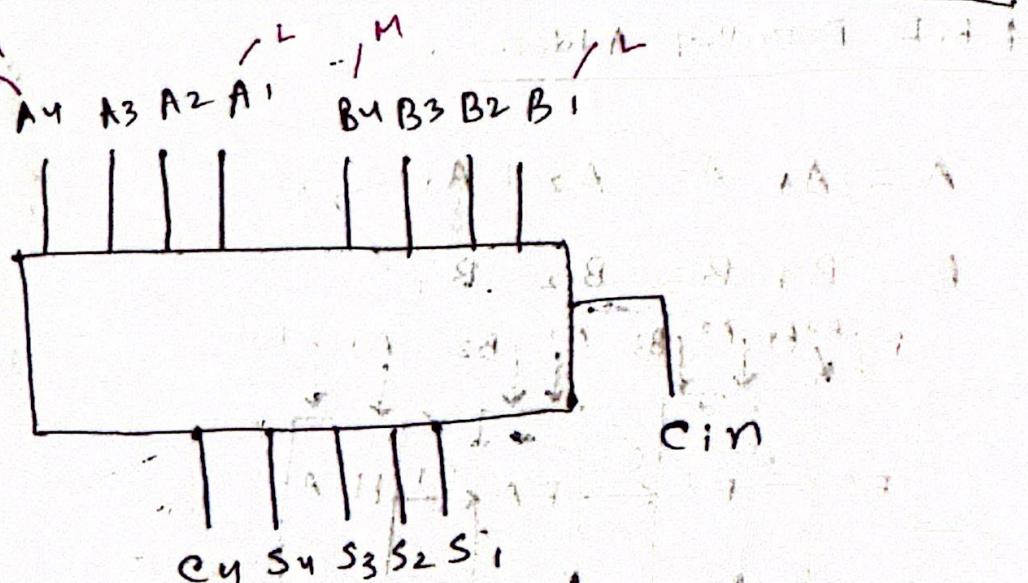
1 ट्रॉप्स Full adder use करा॒ट तय। प्राक्षेपि॑

$C_{in} = 0$ के॒रे॑ तय। तय।



16 bit Parallel Adder = 1 bit P.A X 4

4 bit Parallel Adder/eum subtraction



$$A + B$$

$$A - B \doteq A + (-B)$$

TOPIC NAME :

DAY : _____

TIME : _____

DATE : / /

$$B = \begin{array}{l} 1 \\ 0 \\ 1 \\ 0 \end{array}$$

$$\begin{array}{r} 0 \\ 1 \\ 0 \\ 1 \\ +1 \end{array}$$

$$A_4 A_3 A_2 A_1$$

$$B_4 B_3 B_2 B_1$$

$$\begin{array}{c} 1 \\ | \\ 1 \\ | \\ 1 \end{array}$$

$$\begin{array}{c} 1 \\ | \\ 1 \\ | \\ 1 \end{array}$$

$C_4 S_4 S_3 S_2 S_1$

$C_{in}=1$

1 bit Parallel Adder
Subtractor

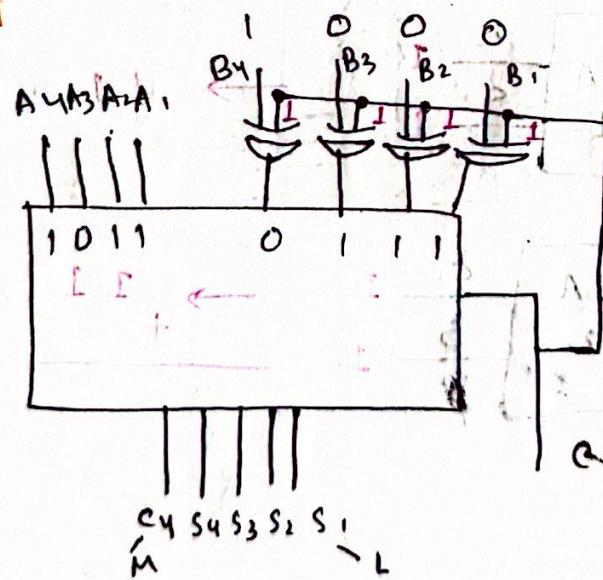
$C_{in}=0 \rightarrow \text{Add} \Rightarrow A + B$

$C_{in}=1 \rightarrow \text{sub} \Rightarrow A + (-B)$

B_4	C_{in}		
0	0	0	0
0	1	1	1
1	0	1	2
1	1	0	3

$$B_4 C_{in}' + B_4' C_{in} = B_4 \oplus C_{in}$$

4 bit Parallel
Adder cum
Subtractor



$C_{in}=0$: Add
 1 : Sub

$$\begin{array}{r} 1011 \\ 1000 \\ +1 \end{array}$$

unused input \rightarrow ground
connect

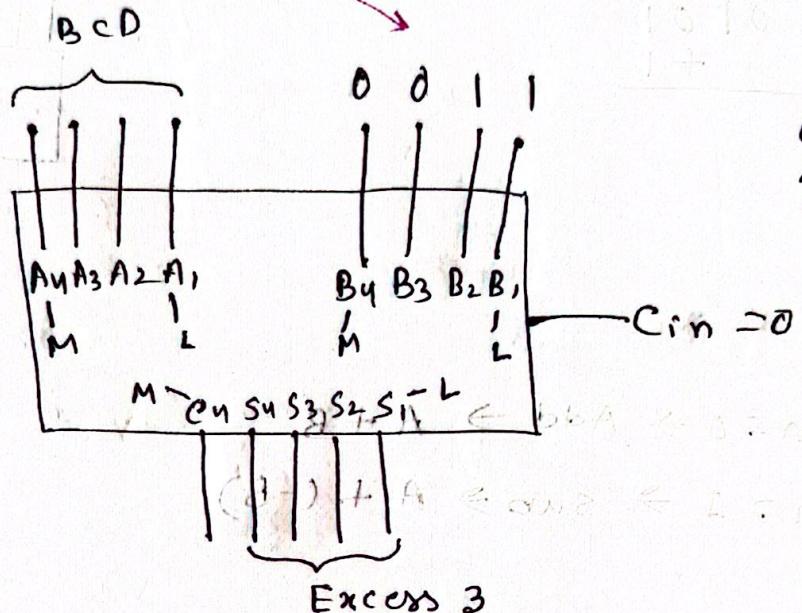
TOPIC NAME : _____

DAY : _____

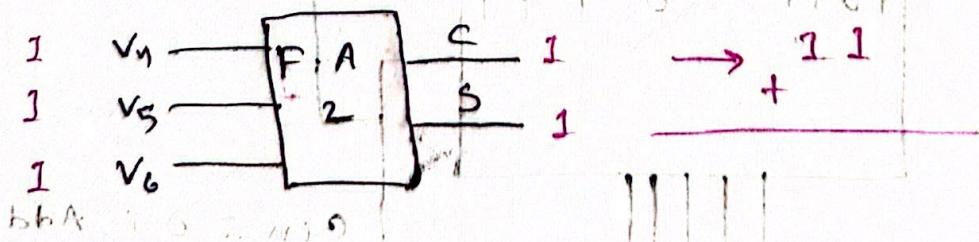
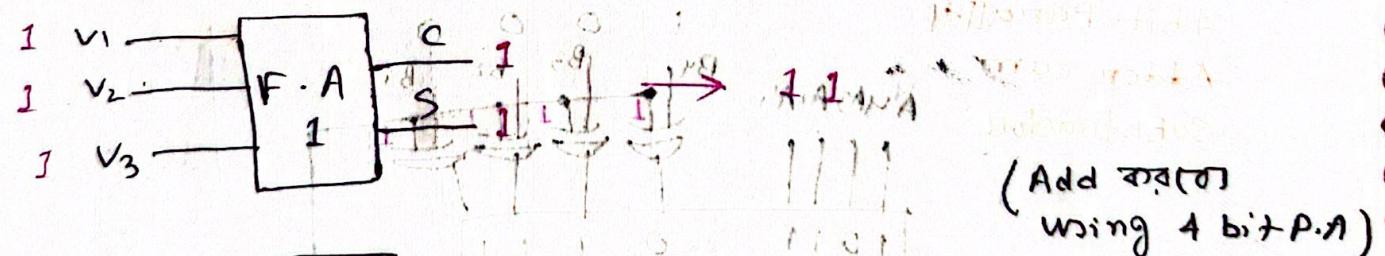
TIME : _____

DATE : / /

BCD to excess 3 (using 1 bit P.A)



Voting System
6 person



GOOD LUCK

TOPIC NAME :

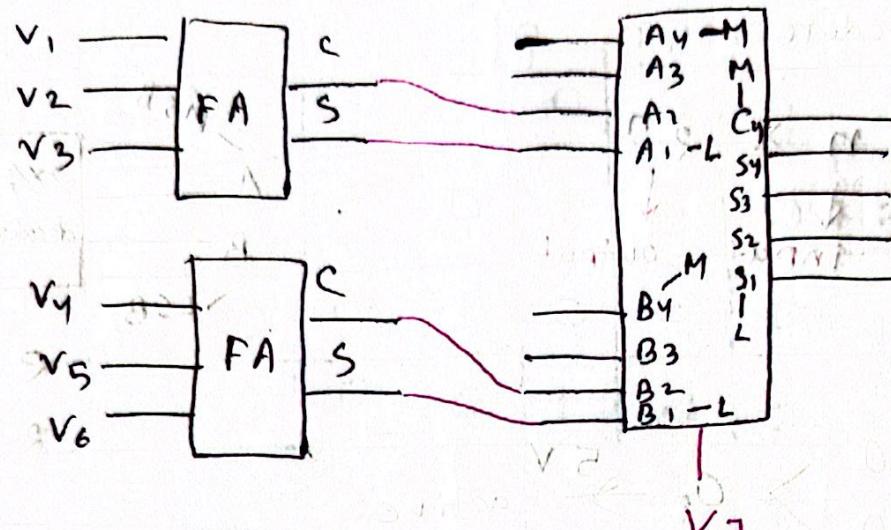
3:8 decoder \rightarrow 4:2 encoder

DAY : _____

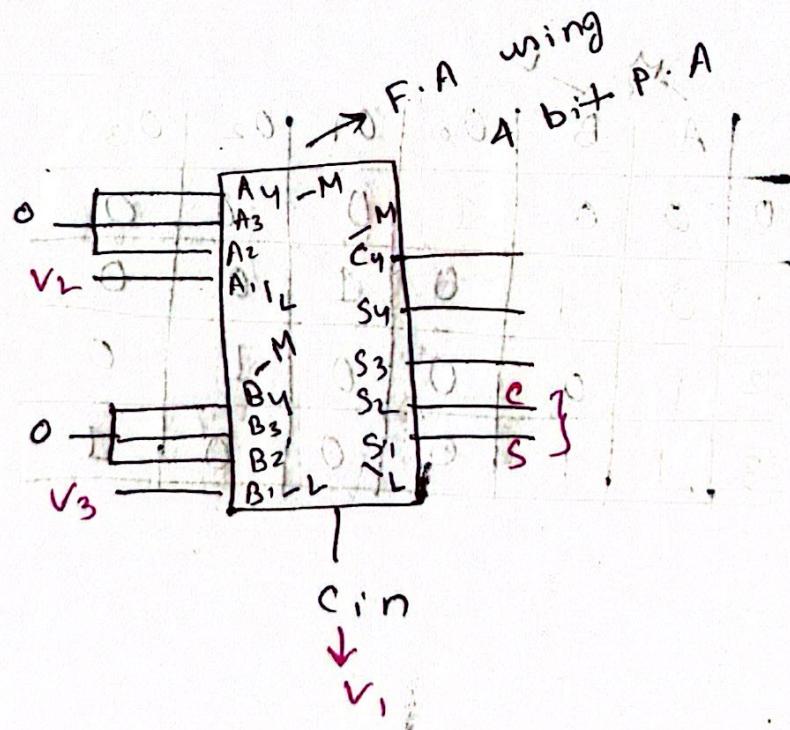
TIME : _____

DATE : / /

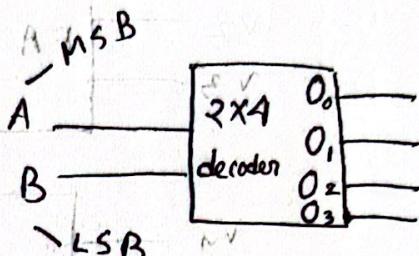
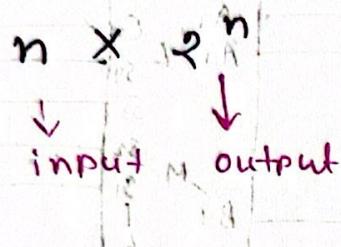
FA \rightarrow



using 4 bit parallel Adder



MSI circuits

Decoder

0 > O₀ → 5V active

1 > O₂ → 5V active

	A	B	O ₀	O ₁	O ₂	O ₃
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

$$O_0 = A'B'$$

$$O_1 = A'B$$

$$O_2 = AB'$$

$$O_3 = AB$$

3 : 8

	A'	B'	C'	0 ₀	0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

$$0_0 = A'B'C'$$

$$0_1 = A'B'C$$

$$0_2 = A'BC'$$

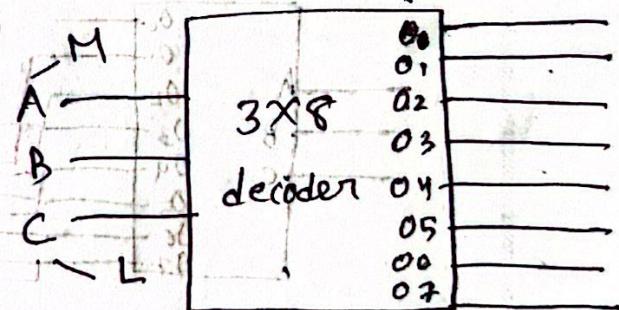
$$0_3 = A'BC$$

$$0_4 = AB'C'$$

$$0_5 = AB'C$$

$$0_6 = ABC'$$

$$0_7 = ABC$$



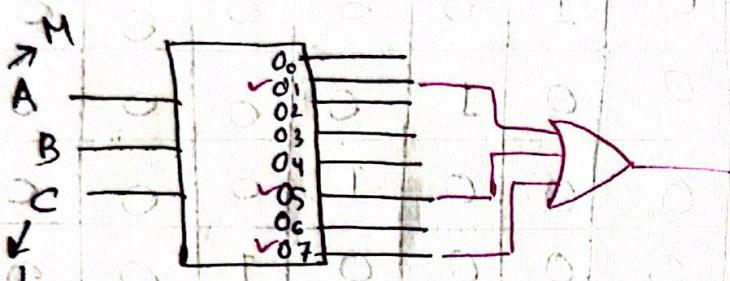
Implementing a Function using a Decoder

$$F(A, B, C) = \Sigma(1, 5, 7)$$

→ Input 3

So draw 3×8

{ sind minterm,
then OR gate



3×8

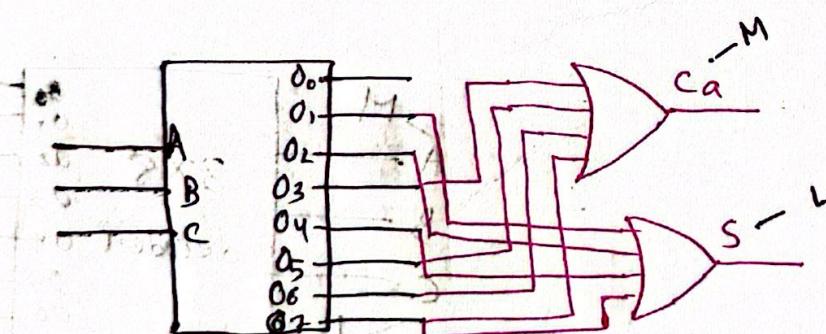
Implement a Full Adder using Decoder

i) Decoder size

ii) Draw decoder

iii) Output \Rightarrow OR gate
minterms for full adder

iv) OR gate



TOPIC NAME : _____

DAY : _____

TIME : _____ DATE : / /

	A	B	C	C_a	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

$$C_a = \Sigma (3, 5, 6, 7)$$

~~$$S = \Sigma (10, 23, 4, 7)$$~~

$$S = \Sigma (1, 2, 4, 7)$$

	A(2)	B(2)	C(2)	D(2)
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

3:8

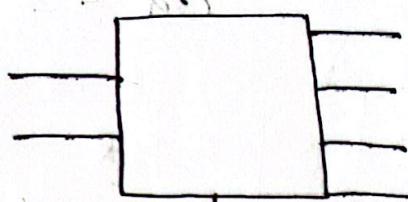
2:4

TOPIC NAME : _____

DAY : _____

TIME : _____

DATE : / /



\hookrightarrow IC on/off switch

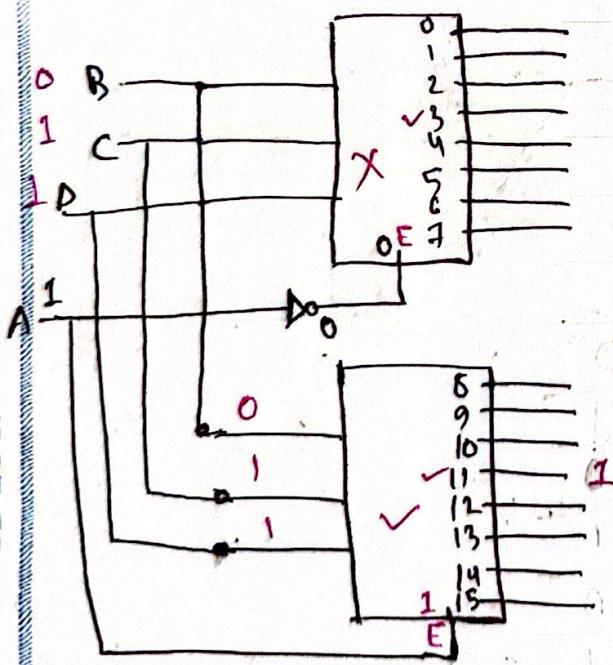
→ active
low
(off position)

Active high \rightarrow 1 from active

Active low \rightarrow 0 from active

4:16 Decoder using 3:8 decoder(s)

$$\frac{3:8}{4:16} \rightarrow \frac{1C}{8} = 2 \text{ decoders}$$

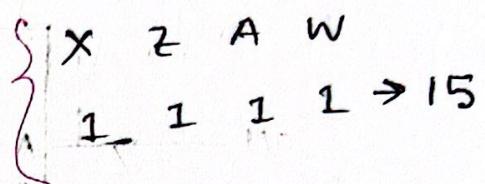
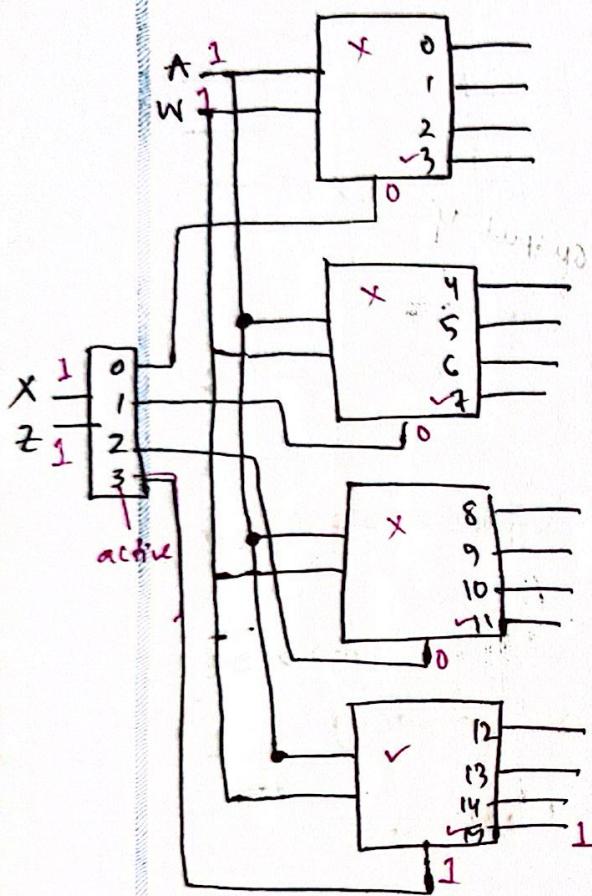


A	B	C	D	Output
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
1	1	0	0	3
0	0	1	0	4
1	0	1	0	5
0	1	1	0	6
1	1	1	0	7
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
1	1	0	1	11
0	0	1	1	12
1	0	1	1	13
0	1	1	1	14
1	1	1	1	15

4:16 Decoder using 3 2:4 decoder(s)

$$F(X, Z, A, W) = \Sigma(1, 3, 12, 15)$$

$\frac{16}{4} \rightarrow 4$ decoder



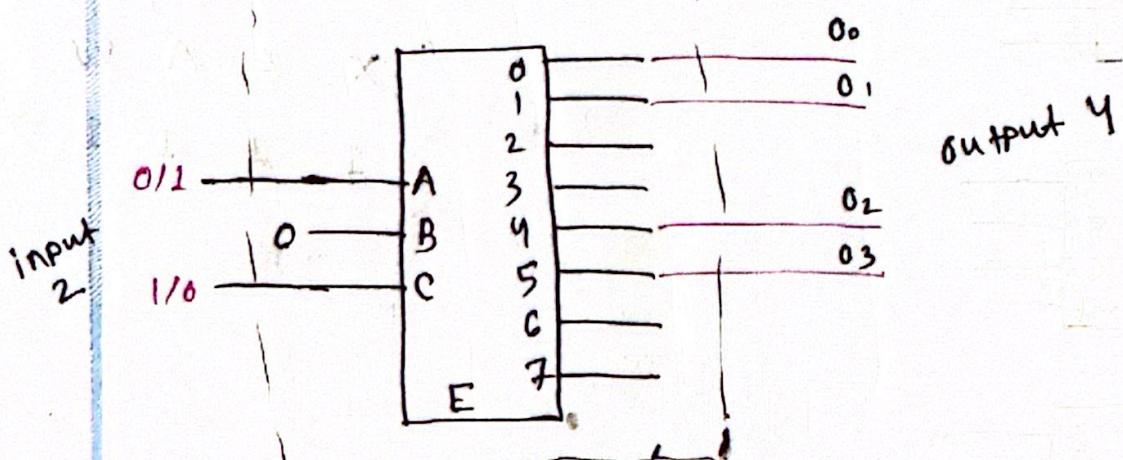
TOPIC NAME : _____

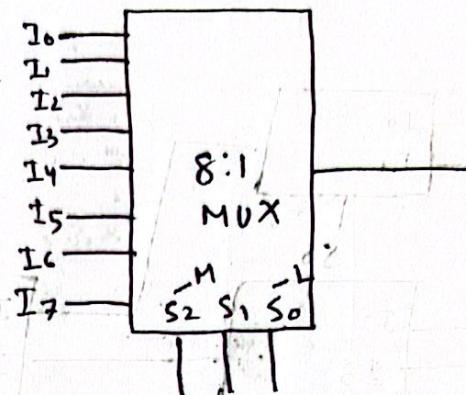
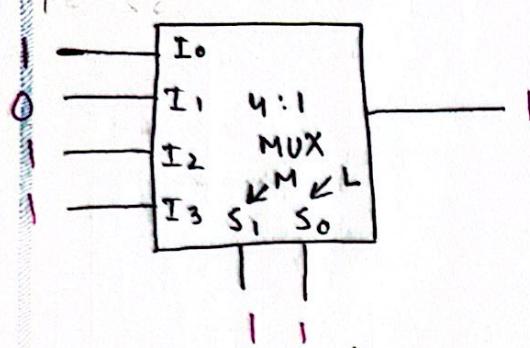
DAY : _____

TIME : _____

DATE : / /

2:4 Decoder using 3:8 Decoder(s)



MUX

$2^n \rightarrow$ input lines

1 \rightarrow output line (fixed)

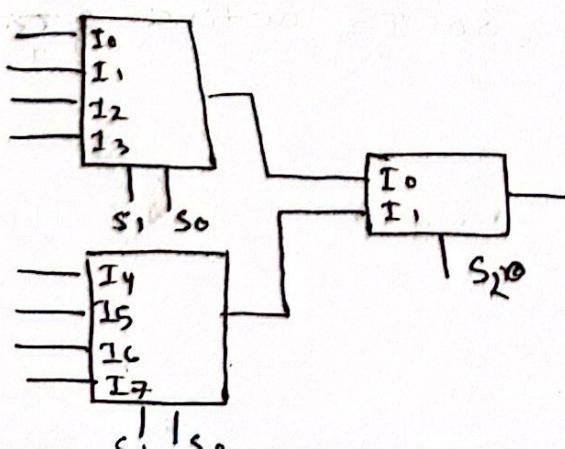
$n \rightarrow$ selector (S)

selector \Rightarrow 00 for I_0 select 2⁰, output 0
1, (when Input 1011)

01 for output 0.

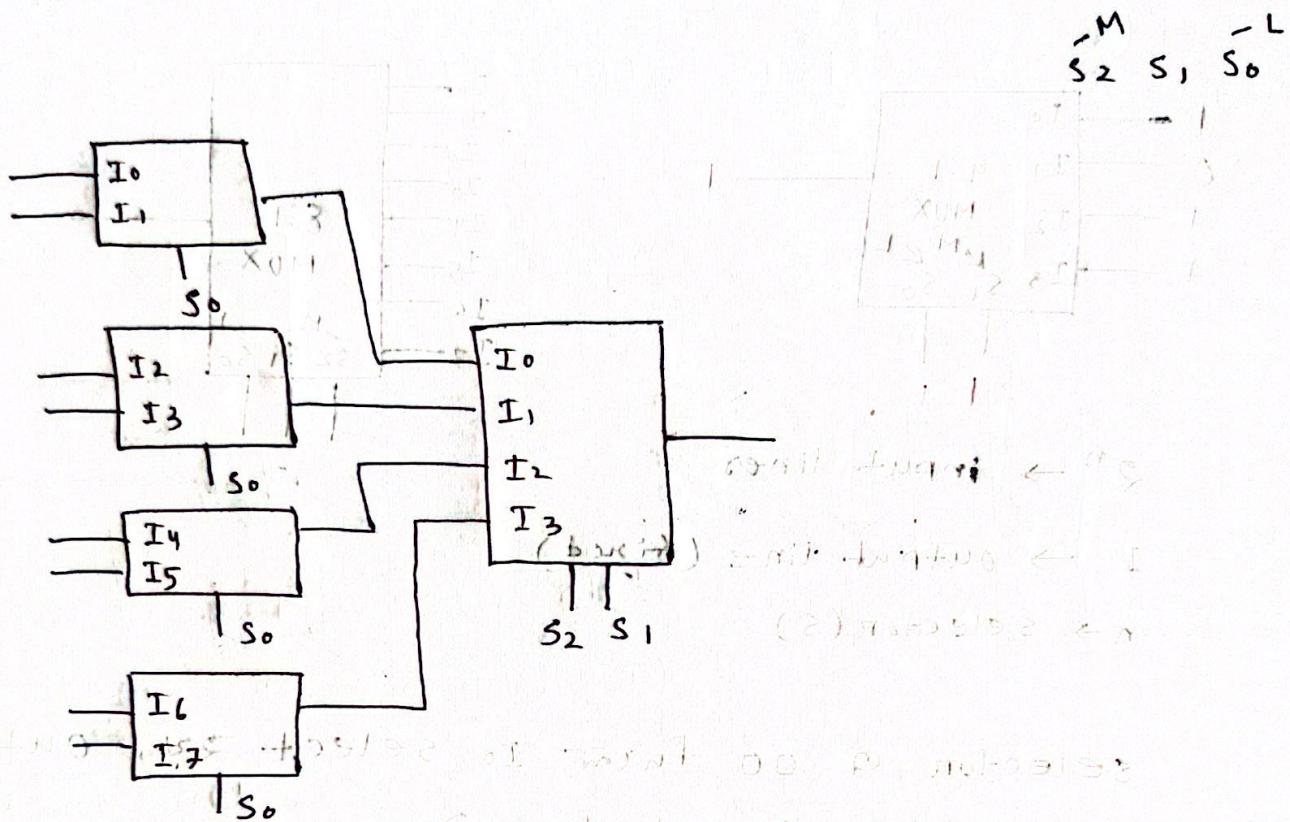
Large MUX:

Build 8:1 mux using two 4:1 and one 2:1 mux



M L same level
(2) MUX 2⁰,
others selector
short 2⁰ 1
उपर फैला (2)
निकल आजे और
assign कराए

Build 8:1 mux using four 2:1 and one 4:1 mux



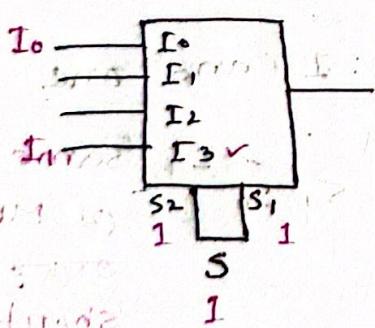
Build 2:1 Mux using 4:1 MUX

$$S=0 \text{ यानी } S_1, S_2 = 0$$

S₀: I₀ active 2⁰

$$S=1 \text{ यानी } S_1, S_2 = 1$$

S₀: I₃ active 2⁰



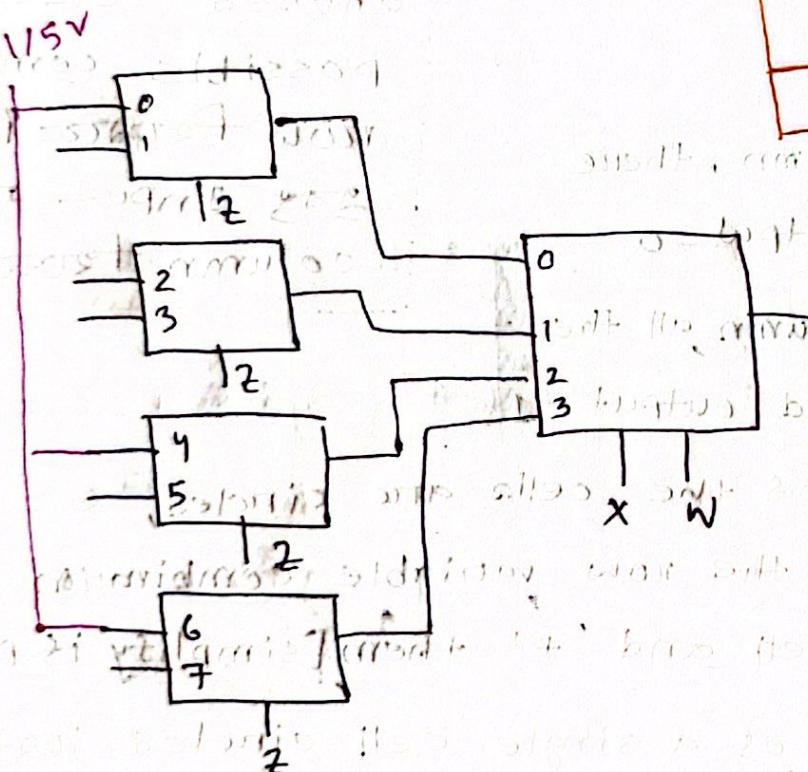
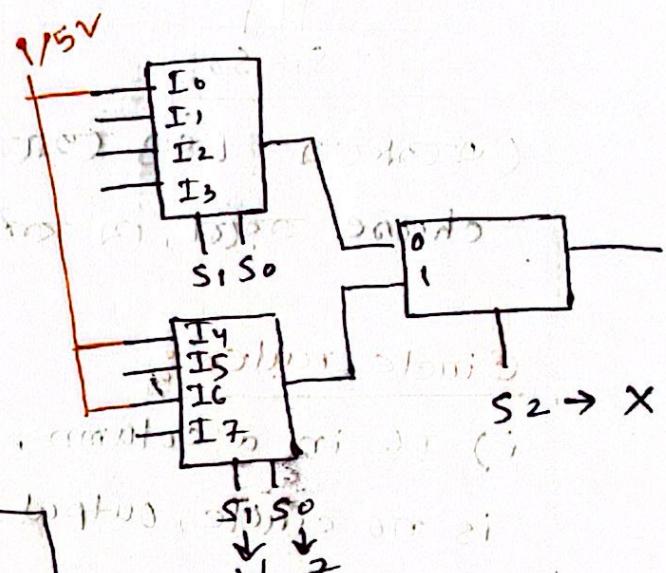
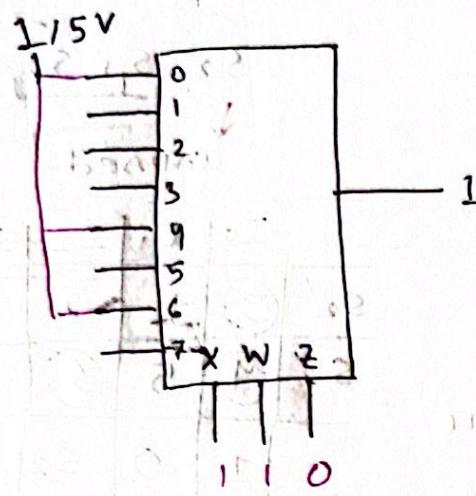
$F = \Sigma(0, 4, 6)$; implement the given function

using 8:1 mux.

$$F(x, w, z) = \Sigma(0, 4, 6)$$

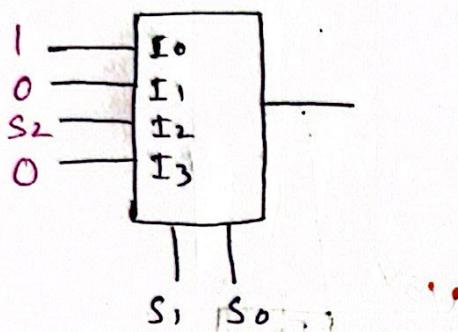
$$= \Sigma(0, 4, 6)$$

↓
selector



$$F(S_2, S_1, S_0) = \sum(0, 4, 8)$$

Ans: 1 mux using single 4:1 mux



(प्रारंभिक LSB तूरन सेलेक्टर
choose करती, (2) करते नाही)

Circle rules:

- i) If in a column, there is no circle, output = 0
- ii) If in a column, all the cells are circled, output = 1
- iii) If some of the cells are circled, then take the row variable combination of each cell and '+' them [simplify if possible]
- iv) In case of a single cell circled, just take the row variable combination of that cell.

M
S2 S1 S0
↓ unused

I ₀	I ₁	I ₂	I ₃
S _{2'} (0)	1	-2	3
S ₂ (1)	5	6	7
0	1	0	S ₂ 0

unused selector as
possible combination
now find the first
row Input from
column first or later,

TOPIC NAME : _____

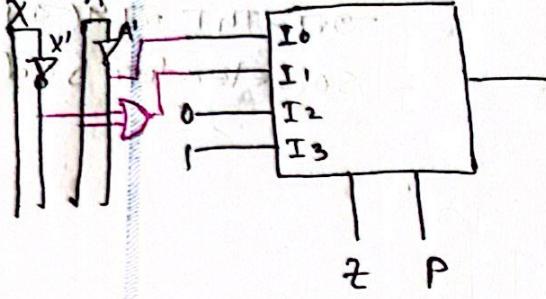
DAY : _____

TIME : _____

DATE : / /

Implement the following function using a single
4:1 mux with enable and select inputs.

$$F = (x^M, A, z, p) = \sum (0, 1, 3, 5, 7, 18, 11, 13, 15)$$



	I ₀	I ₁	I ₂	I ₃
X'A'	0	1	2	3
X'A	4	5	6	7
XA'	8	9	10	11
XA	12	13	14	15
	A'	X'+A	0	1

$$x'A' + xA'$$

$$= A'(x' + x)$$

$$= A'$$

$$x'A' + x'A + xA$$

$$= x'(A + A') + xA$$

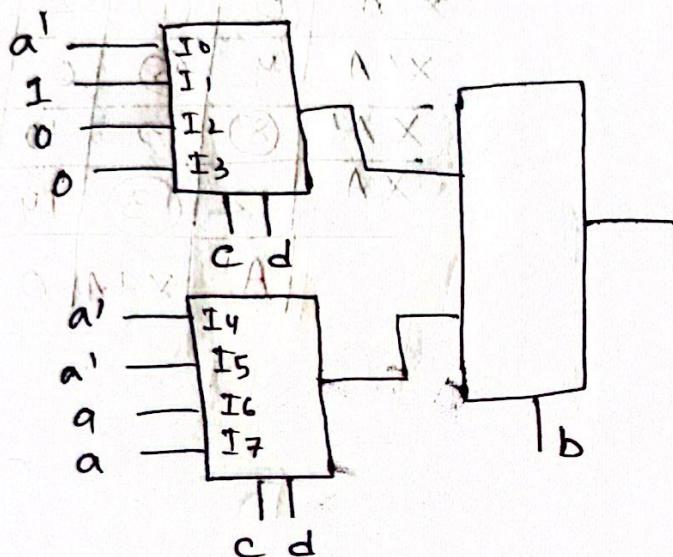
$$= x' + xA$$

$$= (x' + x)(x' + A)$$

$$= (x' + A)$$

$$* F(a, b, c, d) = \sum(0, 1, 4, 5, 9, 14, 15)$$

Implement the above boolean function using two 4:1 MUX(S), and one 2:1 MUX.



	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇
a'	0	1	2	3	4	5	6	7
a	8	9	10	11	12	13	14	15
a'	1	0	0	a'	a'	a	a	a

ट्रैक्टर 8:1 MUX

नियंत्र, फॉर्म

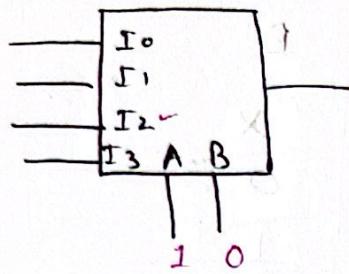
उत्पादन न. 8(3)

SO selector bed

TOPIC NAME : _____

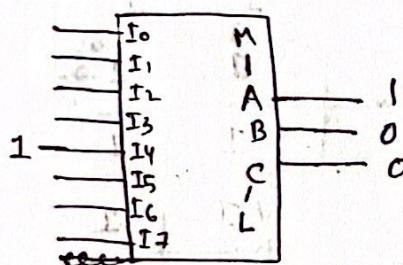
DAY : _____

TIME : _____ DATE : / /



$$(A'B')I_0 + (A'B)I_1 + (AB')I_2 + (AB)I_3$$

Encoder

 $2^n = \text{input}$ $n = \text{output}$

Encoder 6 → 3
line active when 1

Octal-binary encoder

Output

D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	x	y	z
1	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	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Priority encoder

	I ₃	I ₂	I ₁	I ₀	A	B
0	0	0	0	0	X	X
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	0	1
4	0	1	0	0	1	0
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	0
8	1	0	0	0	1	1
9	0	0	1	1	1	1
10	0	1	0	1	1	1
11	1	0	1	0	1	1
12	0	1	0	0	1	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	1	1

Priority logic : If more than one input is active, give priority to the highest position input.

	I ₀	I ₁	I ₂	I ₃	A	B
0	1	0	0	0	0	0
1	0	1	0	0	0	1
2	0	0	1	0	1	0

TOPIC NAME : _____

DAY : _____

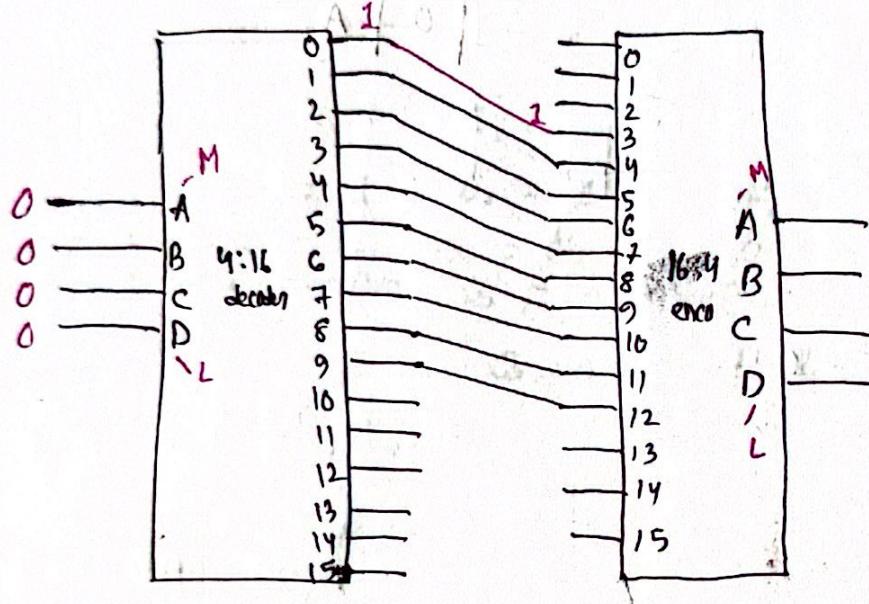
TIME : _____ DATE : / /

Designing a BCD to excess 3 code converter using
 a '4x16' decoder and '16x4' encoder.

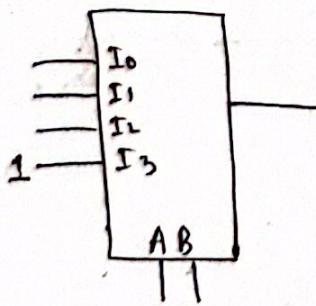
BCD(0-9)

excess 3

$$9+3=12$$



$F(\overline{A}, \overline{B}) = \sum (3)$; Implement the function using
 A : 1 MUX .



A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

AND Gate

TOPIC NAME : _____

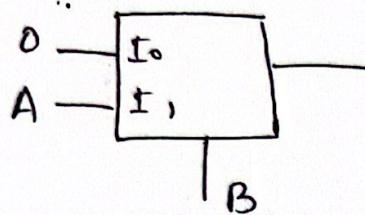
DAY : _____

TIME : _____

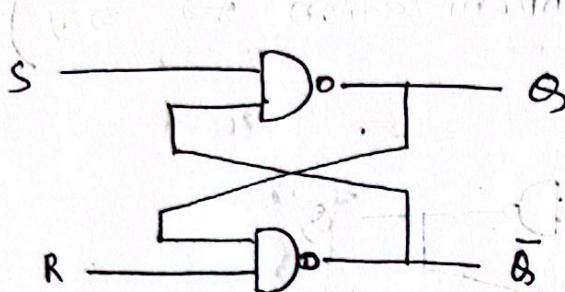
DATE : / /

$F(A, B) = \Sigma(3)$; Implement the function using

2: 1 mux



	I ₀	I ₁
A'	0	1
A	2	3
	0	A

Sequential circuit

NAND		
A	B	(AB)'
0	0	1
0	1	1
1	0	1
1	1	0

$$S = 0; R = 1; Q = 1, \bar{Q} = 0$$

$$S = 1; R = 1; Q = 1, \bar{Q} = 0$$

$$S = 1; R = 0; Q = 0, \bar{Q} = 1$$

$$S = 1; R = 1; Q = 0, \bar{Q} = 1$$

$$S = 0; R = 0; Q = 1, \bar{Q} = 1 \times$$

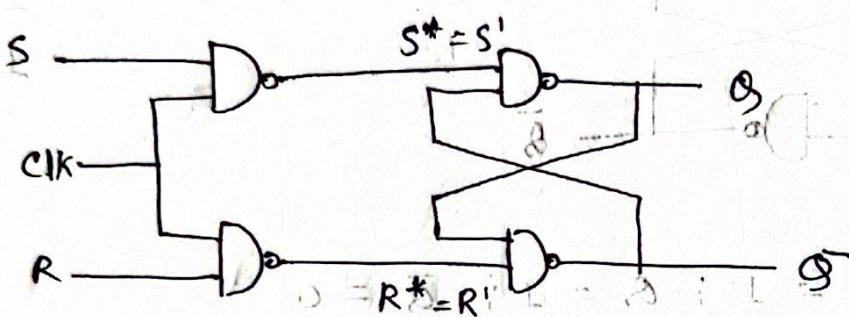
SR Latch is 0,0 state unused.

SR Latch truth table

S	R	Q	\bar{Q}
0	0	not used	
0	1	1	0
1	0	0	1
1	1	memory	next state = present state

S,R Flip-Flop

\downarrow clock edge (synchronization वा कंप)



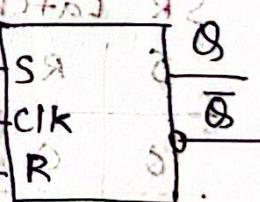
$$\begin{aligned} S^* &= (S \cdot \bar{R} \cdot \bar{k})' \\ &= S' + \bar{R}' + k \end{aligned}$$

$$R^* = (R \cdot \bar{S} \cdot \bar{k})' = R' + S' + \bar{k}$$

$$= R' + \bar{k}$$

$$= R' + 1' = R'$$

Block diagram



Truth table

	R	$S^* = S'$	$R^* = R'$	Q	\bar{Q}
0	0	1	1	memory	\rightarrow no change
0	1	1	0	0	1
1	0	0	1	1	0
1	1	0	0	not used	

For each Flip-Flop A (any one of them) :

- Block Diagram
- Characteristic table
- Excitation table
- Truth table

4 FLIP-FLOPS

i) S-R

ii) D

iii) J-K

iv) T

Characteristic table: Depending on the flip-flop's

inputs and present state what would be the next state.

$Q(t)$	S	R	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	not used
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	not used

$$S=R=0 \Rightarrow Q(t+1)=Q(t)$$

$$Q(t)=Q(t+1)$$

$$S=R=1 \Rightarrow Q(t+1)=\text{not } Q(t)$$

$$Q(t+1)=\text{not used}$$

$$S=0, R=1$$

$$Q(t+1)=0$$

$$S=1, R=0$$

$$Q(t+1)=1$$

Excitation table: Analyzing the flip flops

present and next state we try to determine what might be the input(s) was.

		flip flop input	
$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

But, 0, 0 is combination

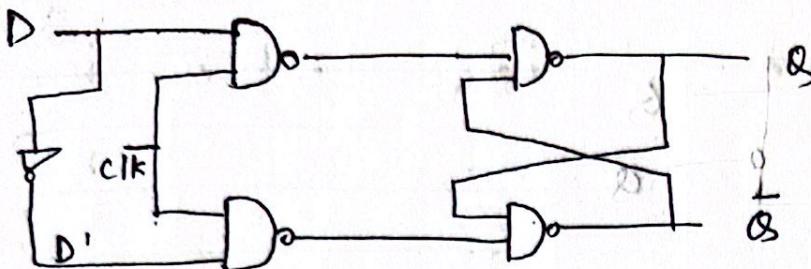
now, check S and R not combination

S same, so keep the same value,

is R not matched, R = don't care (X)

Don't count (not used) as a combination.

1 GT combination

D FLIP FLOPTruth table

D	Q_s	\bar{Q}_s
0	0	1
1	1	0

Characteristic table

$Q(t)$	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1

D \Rightarrow value 2T $Q(t+1) \Rightarrow$ value

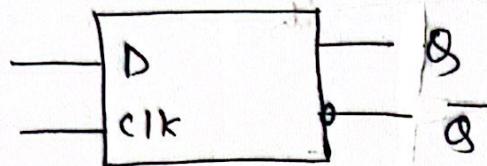
3 same.

(D flip flop is memory
operation करता
जाता नहीं ।)Excitation table

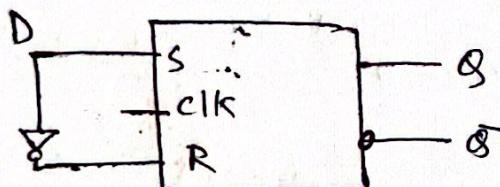
$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

$$[Q(t+1) = D]$$

normal D flip flop



S-R flip flop \Rightarrow D flip flop



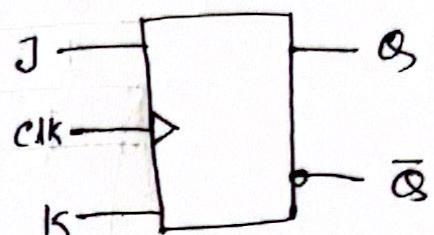
J-K flip flop

Truth table

J	K	Q	Q'
0	0	memory	\rightarrow no change
0	1	0	1
1	0	1	0 (Q1 & Q2) \rightarrow Q3
1	1	Toggle	\rightarrow next state = present state'

Characteristic table

$Q(t)$	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

Excitation table

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

- i) combination
- ii) if match,
as it's
is not match,
don't care (X)

T flip flopTruth table

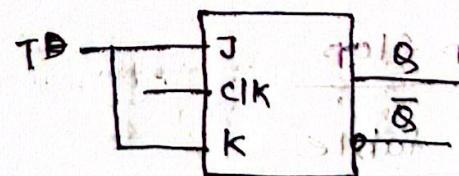
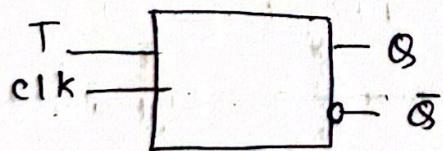
T	Q	\bar{Q}
0	memory	
1	Toggle	

Characteristic table

$Q(t)$	T	$Q(t+1)$	
0	0	0	→ memory $\because T=0$
0	1	1	→ Toggle $\because T=1 \rightarrow Q(t+1) = Q(t)$
1	0	1	→ memory
1	1	0	→ Toggle $\rightarrow Q(t+1) = Q(t)$

Excitation table

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0



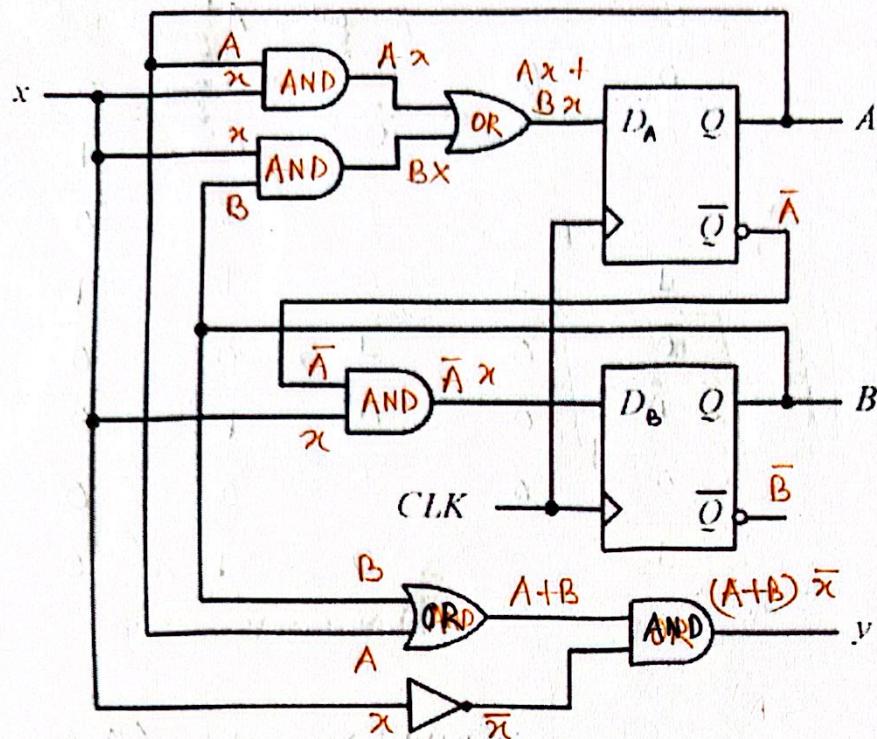
using J-K flip flop

with 2 identical inputs

to build T FF

Draw the state diagram of the given circuit: →

Characteristic
table



i. How many flip-flops are there? =>

ii.

Present State(s)	A, B or $(A(t), B(t))$
Next State(s)	$A+, B+$ or $(A(t+1), B(t+1))$
Input(s)	D_A, D_B, x
Output(s)	y

iii. Equations:

D_A	$Ax + B\bar{x}$
D_B	$\bar{A}x$
y	$(A+B)\bar{x}$

$$A^+ \rightarrow A, D_a$$

$$B^+ \rightarrow B, D_b$$

iv + v. (state table)

present state of S.S.1

present state of S.S.2
get input

A	B	x in	Da	Db	y out	A ⁺	B ⁺
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	1	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	1	0	0
1	1	1	1	0	0	1	0

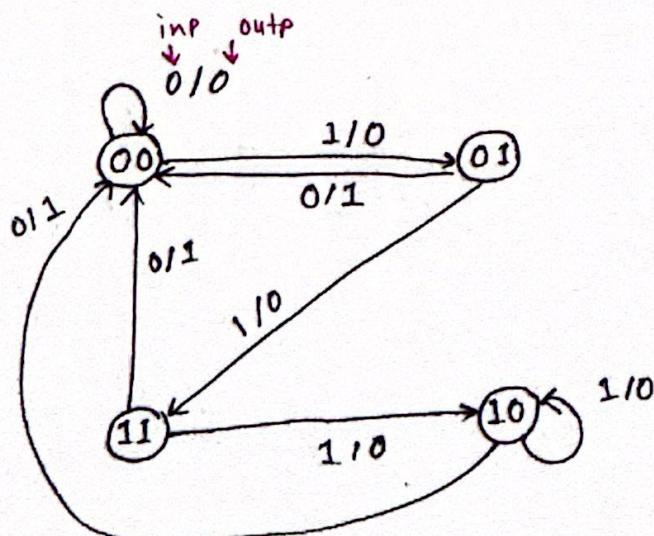
D Flip-Flop Characteristic Table:

Q(t) [Present State]	D	Q(t+1) [Next State]
0	0	0
0	1	1
1	0	0
1	1	1

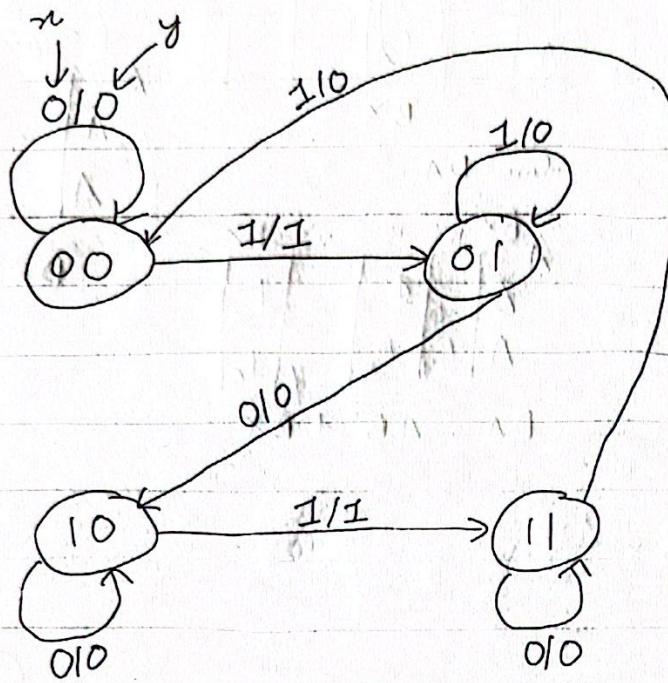
Q input variable (A, B)

Output = A.

vi. State Diagram:



Draw the sequential circuit of the given state diagram using D flip flop: → excitation table



input $\rightarrow A, B, X$
 $Kmap \rightarrow 2^3 = 8$

i+ii.

Kmap

$DA \rightarrow A, A^+$
 $DB \rightarrow B, B^+$

DA, DB
 identified
 from
 excitation
 table.

A	B	x	I	A^+	B^+	y	DA	DB
0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0	1
0	1	0	1	0	0	0	1	0
0	1	1	0	1	0	0	0	1
1	0	0	1	0	0	0	1	0
1	0	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1	1
1	1	1	0	0	0	0	0	0

DA and DB as minterm to implement

$$DA = \Sigma (2, 4, 5, 6)$$

$$DB = \Sigma (1, 3, 5, 6)$$

$$y = \Sigma (1, 5)$$

$Q(t)$ [Present State]	$Q(t+1)$ [Next State]	D_t
0	0	0
0	1	1
-1	0	0
1	1	1

Son y,

We need 3
k-map here.

\bar{A}	BX	$\bar{B}X$	$\bar{B}X$	BX	BX
\bar{A}	0	1		3	2
A	4	1	5	7	6

$\bar{B}X$

fun DA,

B X	B X	B X	BX	B X
A				
A	0	1	3	2
A	1 _a	1 _b	3 _c	2 _d

$$DA = A\bar{B} + B\bar{X}$$

iii.

$$y = \bar{B}x$$

$$Da = A\bar{g} + B\bar{x}$$

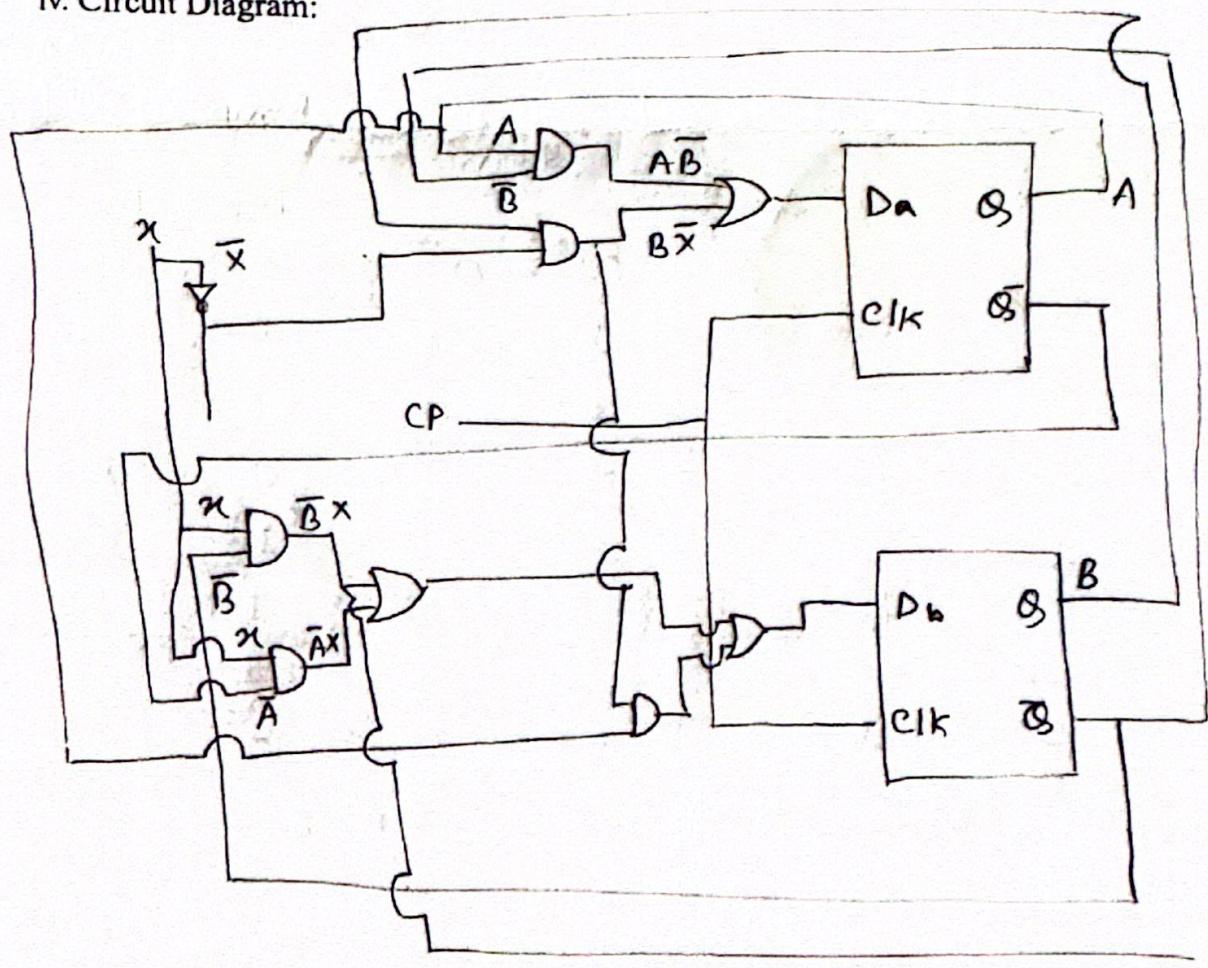
$$Db = \bar{B}x + \bar{A}x + A\bar{B}\bar{x}$$

for DB,

Bx	$\bar{B}\bar{x}$	$\bar{B}x$	Bx	$\bar{B}\bar{x}$
A		1	1	
\bar{A}				
A		1		1

$$DB = \bar{B}X + \bar{A}X + AB\bar{X}$$

iv. Circuit Diagram:



Imp for
signal

TOPIC NAME : _____

DAY : _____

TIME : _____

DATE : 20/04/25

1 bit slip flop एवं 1 bit storage वाला बिट स्लिप फॉलोप

Inputs : external inputs (slip flop input + internal input)
Outputs : external output (हरा)

Slip flop वाले प्रेसेन्ट स्टेट डॉट

Circuit to state Diagram → characteristic table

Circuit तथा state diagram वाले सभी characteristic table लिखो।

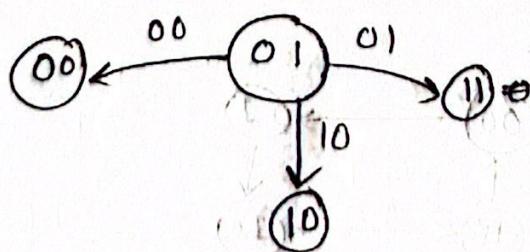
State → circuit Diagram → excitation table

i) Draw state table

present state 2 bit \rightarrow so slip flop 2 टे

external input लिखो (01, 10, 00, 11)

and 3 टे यानि external input 2 bit लिखो ।



Counter

start করে তার স্ট্যাট একই হবে। Same state

(start) এ ফিল আসলে, তারে counter circuit

বৈন।

Counter

- ↳ synchronous
- ↳ asynchronous

Synchronous: কোটি clock একে অবচলনের connection

Asynchronous: এই flip-flop এর output এর পরামর্শে

clock হিতবেকাত করে।

এই গুলো দেখে এই state এর মধ্যে

Synchronous counter:

* Design a 2-bit Synchronous binary counter

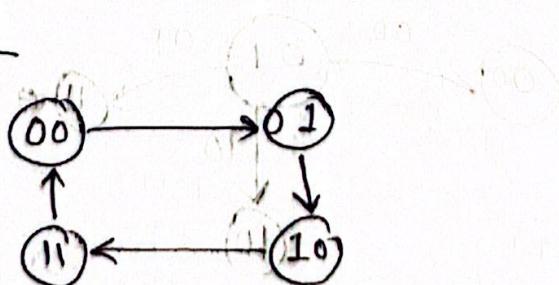
(using T flip-flop)

2 bit \rightarrow 4
combinations

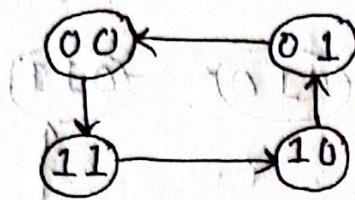
যদি নির্দিষ্ট না রয়েল by default up

counter draw করত হবে,

Up counter



Down counter: ~~most significant bit is 1~~



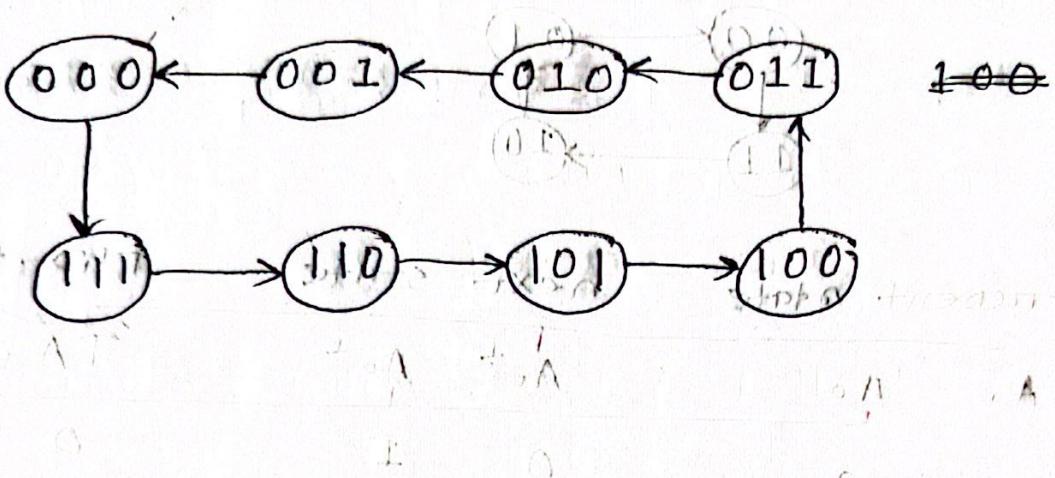
present state		next state		flip-flop input using excitation table	
A	A ₀	A ₁ +	A ₀ +	T _{A1}	T _{A0}
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

Q _n	Q _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

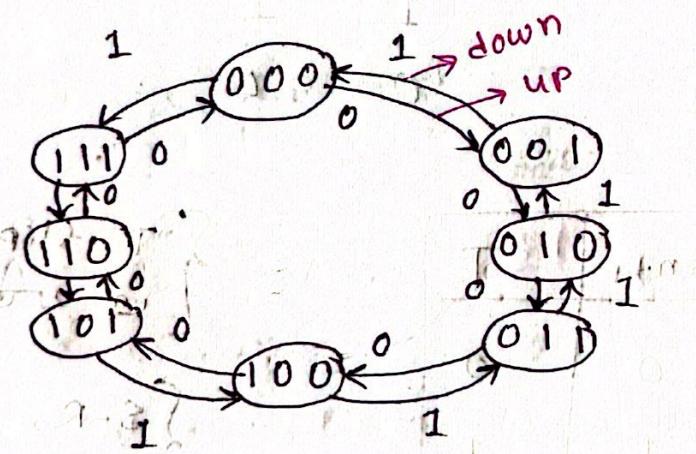
{ T flip-flop, D flip flop
or zero input to K-map

{ S-R, J-K or zero input
to first K-map.

* 3 bit synchronous Down counter



3 bit synchronous up/down counter

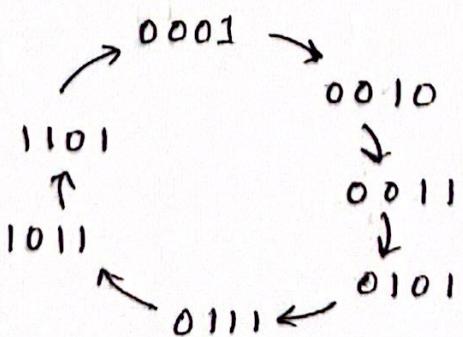


anc & CCR, so
external bit
1 bit.

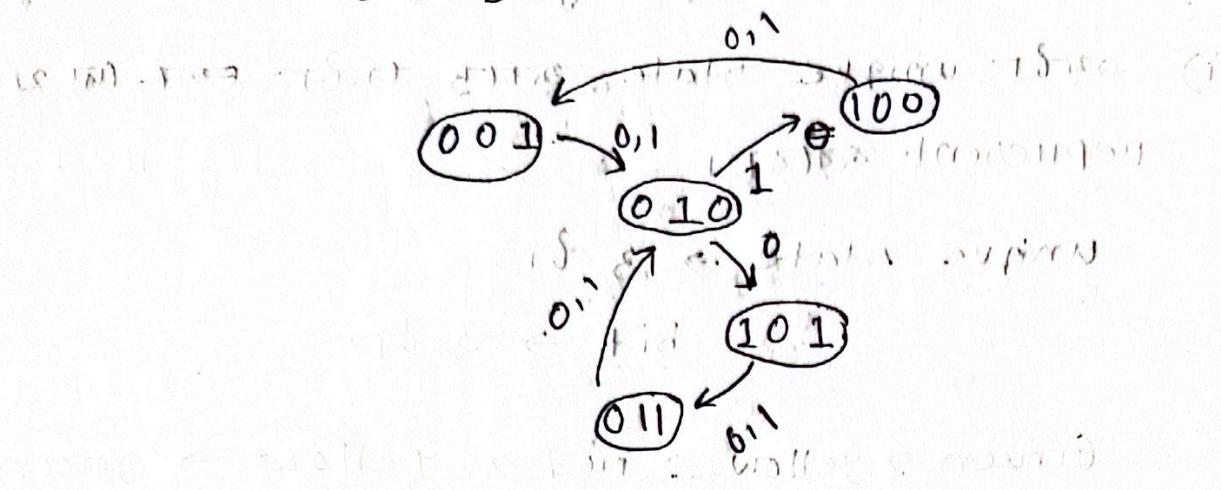
up → 0
down → 1

Practice Problem 1

Implement the following counter using TFF: $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 11 \rightarrow 13 \rightarrow 1$



* $1 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$ (don't care)



x	A	B	C	A^+	B^+	C^+
0	0	0	1	0	1	0
1	0	0	1	0	1	0
0	0	0	0	0	0	1
1	0	0	0	0	0	1

for all the unused states refer to the starting state

(is not mentioned then don't care (x))

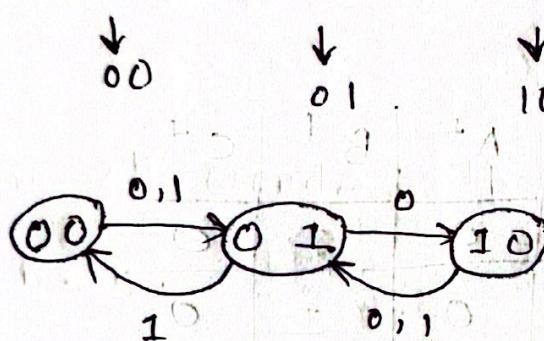
Practice Problem 2 (Using D FF)

i) কয়েটা unique state আছে তাৰ প্ৰতি এক দিন
represent কৰলো।

unique state \rightarrow 3 bit

bit \rightarrow 2 bit

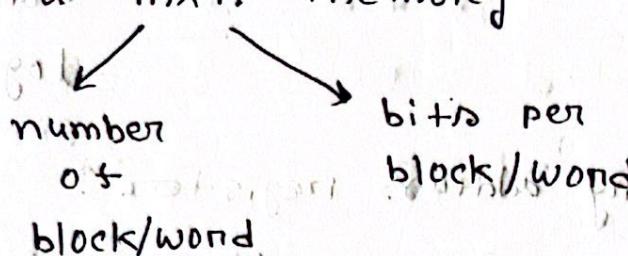
Green \rightarrow yellow \rightarrow red \rightarrow yellow \rightarrow green



X	A	B	A +	B +	DA	DB
0	0	0	0	1	0	1
0	0	1	1	0	1	0
0	1	0	0	1	0	1
0	1	1	x	x	x	x
1	0	0	x	x	0	1
1	0	1	0	0	0	0
1	1	0	0	1	0	1
1	1	1	x	x	x	x

Memory

Designing $m \times n$ memory



Read operation:

memory address register : 010

? no word active কার্যত

read or write অনুসূচিত নথি এবং এটি active কার্যত

এবং store কার্যত !

এখন আজ নিব

Steps:

1. CPU sends the memory address from where the data will be read into the MAR.
2. The memory unit activates the corresponding memory location.
3. The read control signal will be activated. ($\text{Read} = 1$)
4. Data at the address is fetched and placed in the MBR.

Write operation:

Steps:

1. CPU sends the memory address where data will be written into the MAR.
2. Places the data to be written in the MBR.
3. The write control signal will be activated ($\text{Write} = 1$).
4. Memory locates the given address and stores the data in it.

TOPIC NAME : _____

DAY: _____

TIME: _____

no. of words / /

what do you understand by a 1024×16 bit memory

how many bits can be stored in a word

$$\log_2 1024 = 10$$

size of memory address register

$$= \log_2 (\text{No of words})$$

= Address line

for 2^k words; the address register size will be k .

The control lines - Read & write determine the direction of data transfer.

You can not perform read and write on a single location at the same time.

Using a Flip-Flop 1 bit data can be stored.

For 1024×16 bit memory.

a) How many flip flop are needed to build the address reg.

b) How many FFS are needed to build the buffer a address reg.

SOLN:

$$2^{10} = 1024$$

address register size $\Rightarrow 10$; So 10 FFSbuffer register size $\Rightarrow 16$; So 16 FFS

* How many address lines are needed for a 64 MB RAM with 32 bit word?

Capacity = no. of words \times bit/word

$$\text{No. } 64\text{ MB} = Y \times 32 \text{ bits}$$

$$\Rightarrow 64 \times 1024 \times 1024 \times 8 = 32Y$$

bits

$$\Rightarrow Y = 16777216$$

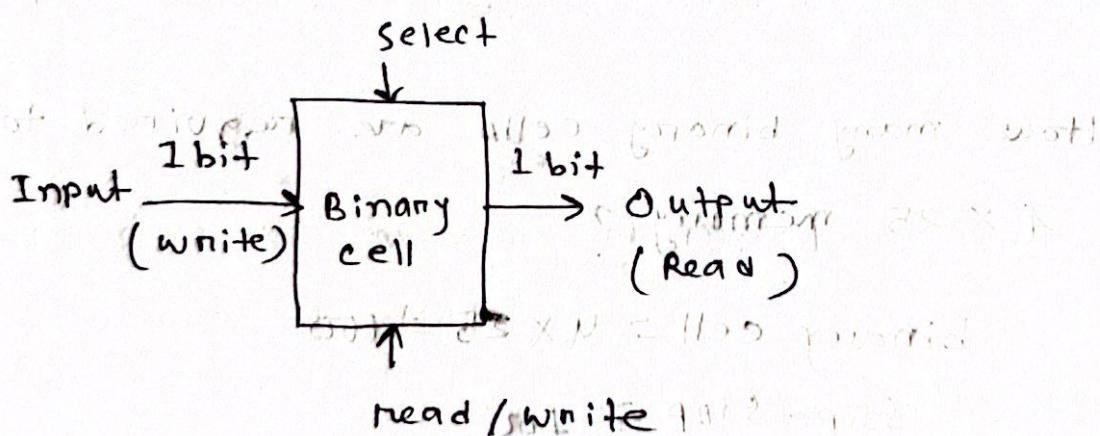
$$\log_2(16777216) = 24$$

Address line = 24

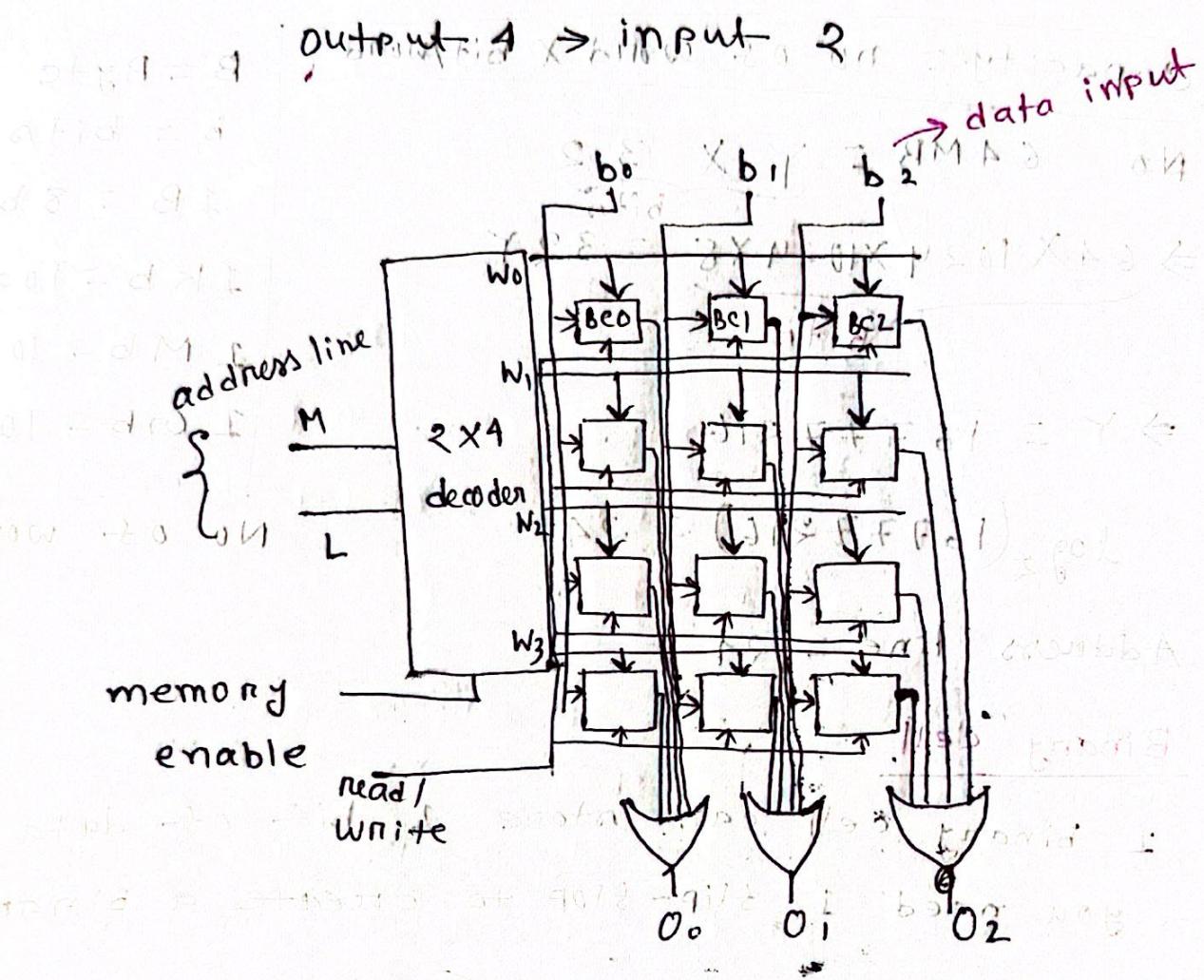
Binary cell

1 binary cell can store 1 bit of data.

You need 1 flip-flop to create a binary cell.



Design a 4×3 memory / RAM with 3 bit decoder \Rightarrow output



How many binary cells are required to build a 4×25 memory?

$$\text{binary cell} = 4 \times 2^5 = 100$$

$$100 = 100$$

Design a 2×3 RAM

↓
decoder output
(1:2)

