

> Decimal
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

, only upto 9

ECD

Excess 3 code

* **Gray Code:** (NO transition like BCD)
 There is only one variable change in all.
 Eg: K-Map 00 01 11 $\xrightarrow{\text{Changed pos}}$ 10

* Binary to Grey Code & vice versa

$$2) \begin{array}{r} 10110 \\ \downarrow \downarrow \downarrow \downarrow \downarrow \\ 11101 \\ \downarrow \oplus \downarrow \\ 10110 \end{array} \rightarrow \begin{array}{l} \text{Binary} \\ \text{Grey} \\ \text{Binary} \end{array}$$

$$3) \begin{array}{r} 11000110 \\ \downarrow \downarrow \downarrow \downarrow \\ 10100101 \end{array} \rightarrow \begin{array}{l} \text{Binary} \\ \text{Grey} \end{array}$$

$$4) \begin{array}{r} 111010 \\ \downarrow \\ 101100 \end{array} \rightarrow \begin{array}{l} \text{Grey} \\ \text{Binary} \end{array}$$

Adders :

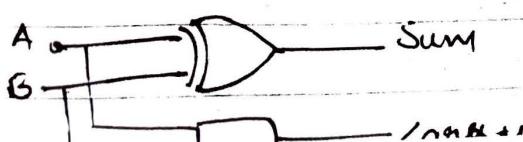
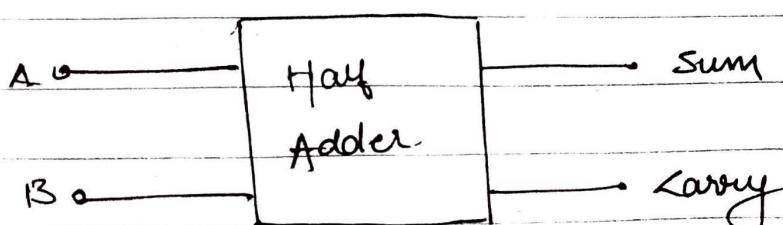
Basic Rule:

* Half Adders

		O/P	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\left. \begin{array}{l} 0+0=0 \\ 0+1=1 \\ 1+0=1 \\ 1+1=0 \end{array} \right\} \text{with carry 1}$$

↳ eq. to
XOR ↳ eq. to
AND

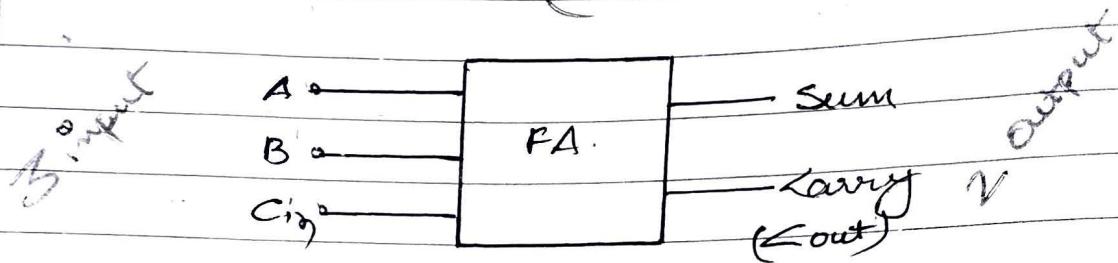


Digital circuit

~* Full Adders

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(XOR addⁿ)



From T-T build SOP expression.

$$\begin{aligned}
 \text{Sum} (S) &= \bar{A}\bar{B}\bar{C}_{in} + \bar{A}\bar{B}C_{in} + A\bar{B}\bar{C}_{in} + AC_{in} \\
 &= \bar{A}(\bar{B}C_{in} + BC_{in}) + A(\bar{B}C_{in} + BC_{in}) \\
 &= \bar{A}(B \oplus C_{in}) + A(B \oplus C_{in})
 \end{aligned}$$

$$\text{Put } x = B \oplus C_{in}$$

$$= \bar{A}x + Ax = A \oplus x$$

$$= A \oplus (B \oplus C_{in})$$

$$= A \oplus B \oplus C_{in}$$

↳ Output is X-OR addⁿ of input.

(Not binary addⁿ)

Verify with K. Map

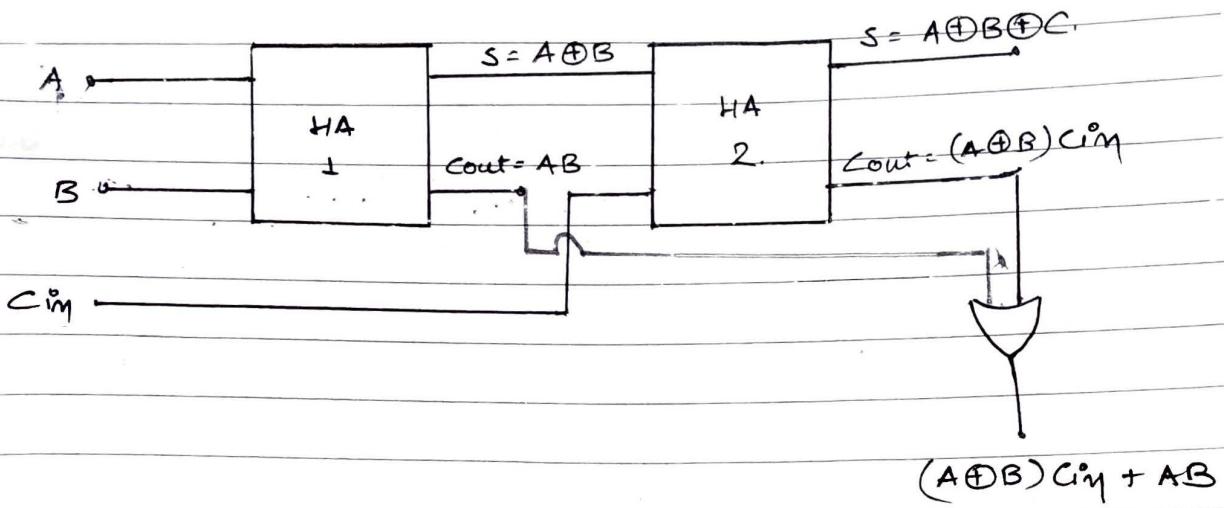
classmate

Date _____
Page _____

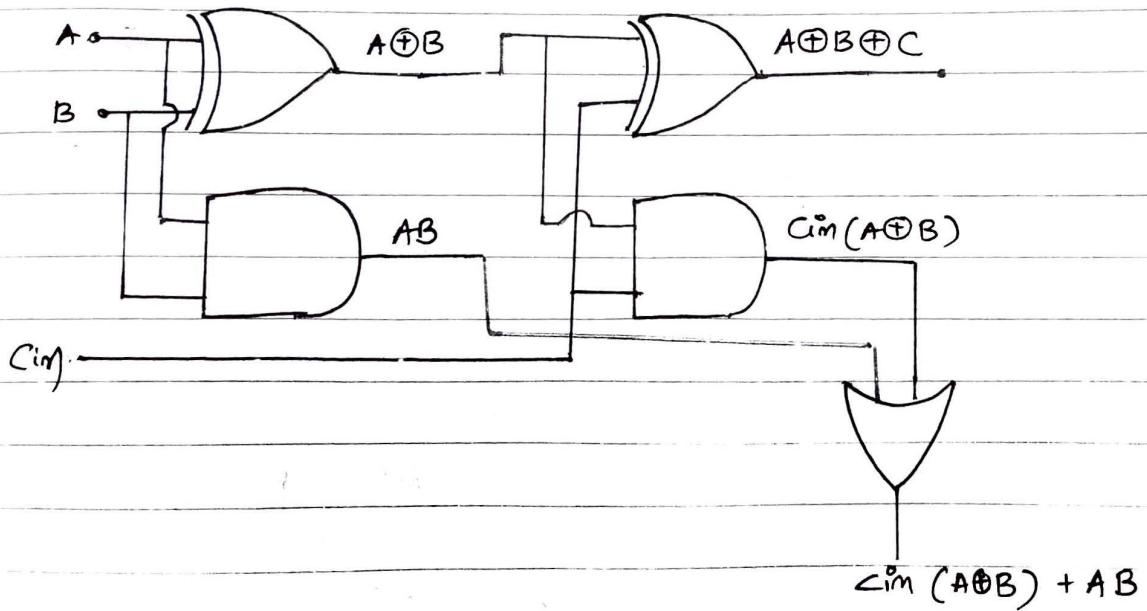
From TT

$$\begin{aligned}
 C_{out} &= \bar{A}\bar{B}C_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} \\
 &= C_{in}(\bar{A}\bar{B} + A\bar{B}) + AB(C_{in} + \bar{C}_{in}) \\
 &= \underline{\underline{C_{in}(A \oplus B)}} + AB
 \end{aligned}$$

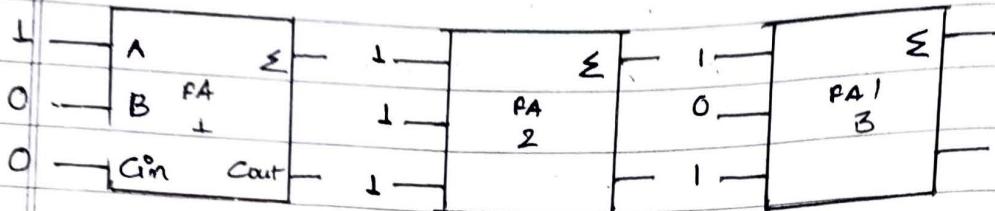
* Full Adder using 2 Half Adder.



* Full Adder using logic gate:



Ques For each of the 3 FA given below determine O/P for I/P



$$\text{For FA } \text{Sum} = A \oplus B \oplus C_{in}$$

$$\begin{array}{lll} \text{i)} & 1 \oplus 0 \oplus 0 & \text{ii)} S = 1 \\ & = 1 \oplus 0 & C_{out} = 1 \\ & & = 1. \\ & C_{out} = 1. & \end{array} \quad \begin{array}{ll} \text{iii)} S = 0 \\ C_{out} = 1 \end{array}$$

Ques Determine sum and carry for each half adder for each set of inputs.

$$\text{a)} \quad 0 \ 1 \quad \Sigma = 1 \quad \text{c)} \quad 1 \ 0 \quad \Sigma = 1 \\ C_{out} = 0 \quad \quad \quad C_{out} = 0$$

$$\text{b)} \quad 0 \ 0 \quad \Sigma = 0 \quad \text{d)} \quad 1 \ 1 \quad \Sigma = 0 \\ C_{out} = 0 \quad \quad \quad C_{out} = 1.$$

$$\text{For HA } \text{Sum} = A \oplus B$$

$$C_{out} = AB$$

Mention each and every thing like $A = 0$, $B = 1$.

Ques FA has $C_{in} = 1$ what are sum and C_{out} when $A = 1, B = 1$

$$\text{Sum} = 1 \quad (A \oplus B \oplus C)$$

$$C_{out} = 1 \quad (C_{in} (A \oplus B) + AB)$$

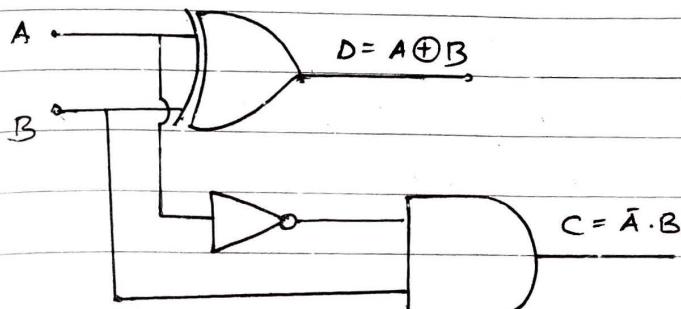
Subtractor

* Half subtractor?

A logic circuit which will perform B (subtrahend) from A (minuend) is known HS.

I/P		O/P		$\left. \begin{array}{l} 0-0=0 \\ 0-1=1 \\ 1-0=1 \\ 1-1=0 \end{array} \right\}$ <small>with borrow</small>
A	B	D (Difference)	C (borrow)	
0	0	0	0	
0	1	1	1	
1	0	1	0	
1	1	0	0	

\hookrightarrow eq. to $A \oplus B$ \hookrightarrow eq. to $\bar{A} \cdot B$



~~HS~~ HS using Logic Gates.

* Full subtractor: Designed with 3 inputs and used for multibit subtraction

A (minuend) B (subtrahend) C_{n-1} (Borrow)

O/P: D (Difference)
 C_n (Borrow).

T.T

I/P		C_{n-1}	D	O/P	C_n (Behavior)
A	B				
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1

 \rightarrow Eq. to

$$A \oplus B \oplus C$$

For C_n

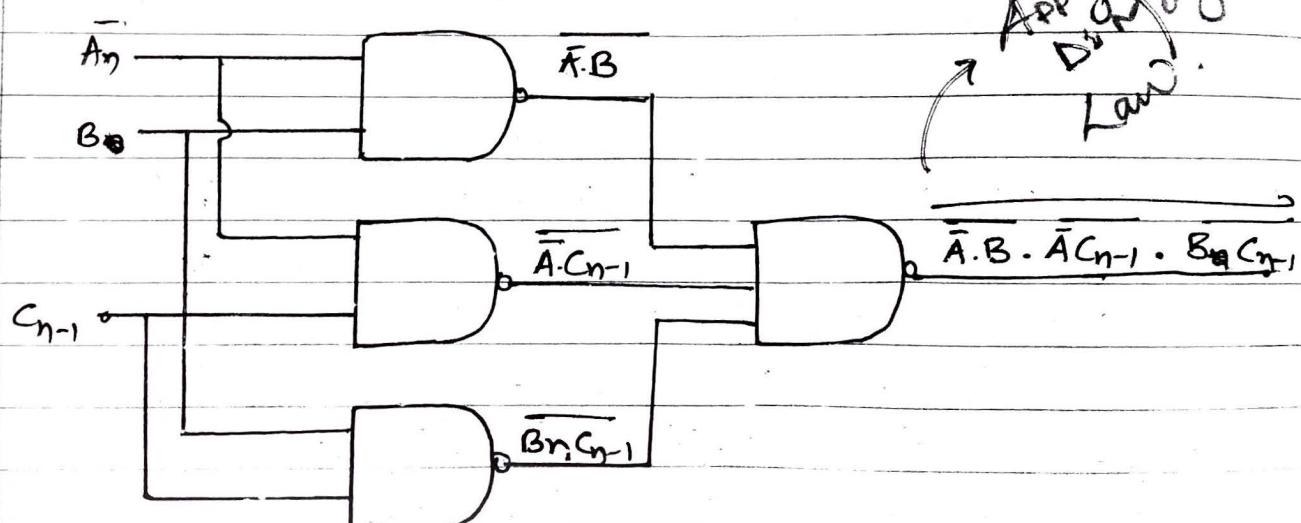
K-map.

	AB	00	01	11	10
0		0	1	0	0
1		1	1	1	0

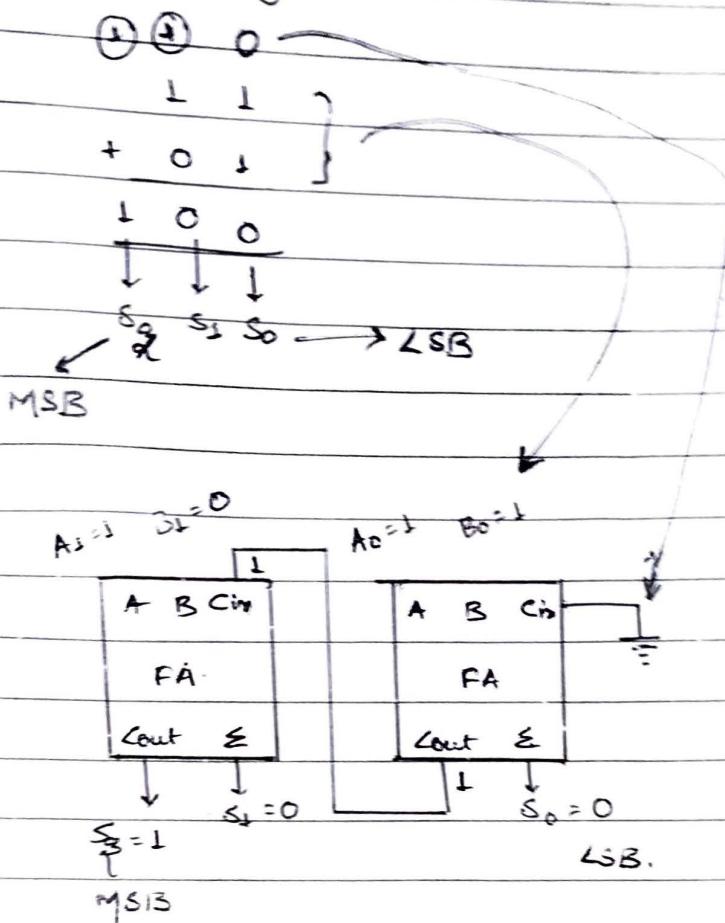
$$C_n = \bar{A}B + \bar{A}C_{n-1} + B(C_{n-2}) \quad \checkmark$$

$$C_n = \underline{\underline{C_{n-1}(\bar{A}B)}} + \bar{A}B$$

~* FS using NAND gate



Parallel Binary Adder.



* Code Converters

→ Binary to BCD cc.

	Binary				BCD				
cc ⁰	A	B	C	D	B ₄	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	Normal	0	0	0	0
9	1	0	0	1		0	1	0	0
10	1	0	1	0		1	0	0	0
11	1	0	1	1		1	0	0	1
12	1	1	0	0		1	0	0	0
13	1	1	0	1		1	0	0	1
14	1	1	1	0	D	1	0	1	0

	A	B	C	D	B_4	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1
2	0	0	1	0	0	0	0	1	0
3	0	0	1	1	0	0	0	1	1
4	0	1	0	0	0	0	1	0	0
5	0	1	0	1	0	0	1	0	1
6	0	1	1	0	0	0	1	1	0
7	0	1	1	1	0	0	1	1	1
8	1	0	0	0	0	1	0	0	0
9	1	0	0	1	0	1	0	0	1
10	1	0	1	0	1	0	0	0	0
11	1	0	1	1	1	0	0	0	1
12	1	1	0	0	1	0	0	1	0
13	1	1	0	1	1	0	0	1	1
14	1	1	1	0	1	0	1	0	0
15	1	1	1	1	1	0	1	0	1

K MAP for B_4

(SOP eqn)

		AB CD	00	01	11	10
		CD	00			
		AB	00			
		00			1	
		01			1	X
		11		X	1	1
		10			1	1

 $\rightarrow B_4$ $AB + AC$

		AB CD	00	01	11	10
		CD	00			
		AB	00			
		00				1
		01				1
		11				
		10				

 $\rightarrow B_3$ $AB\bar{C}$

B2

B1.

=AB	00	01	11	10
CD	1			
00				
01	1			
11	1	1		
10	1	1		

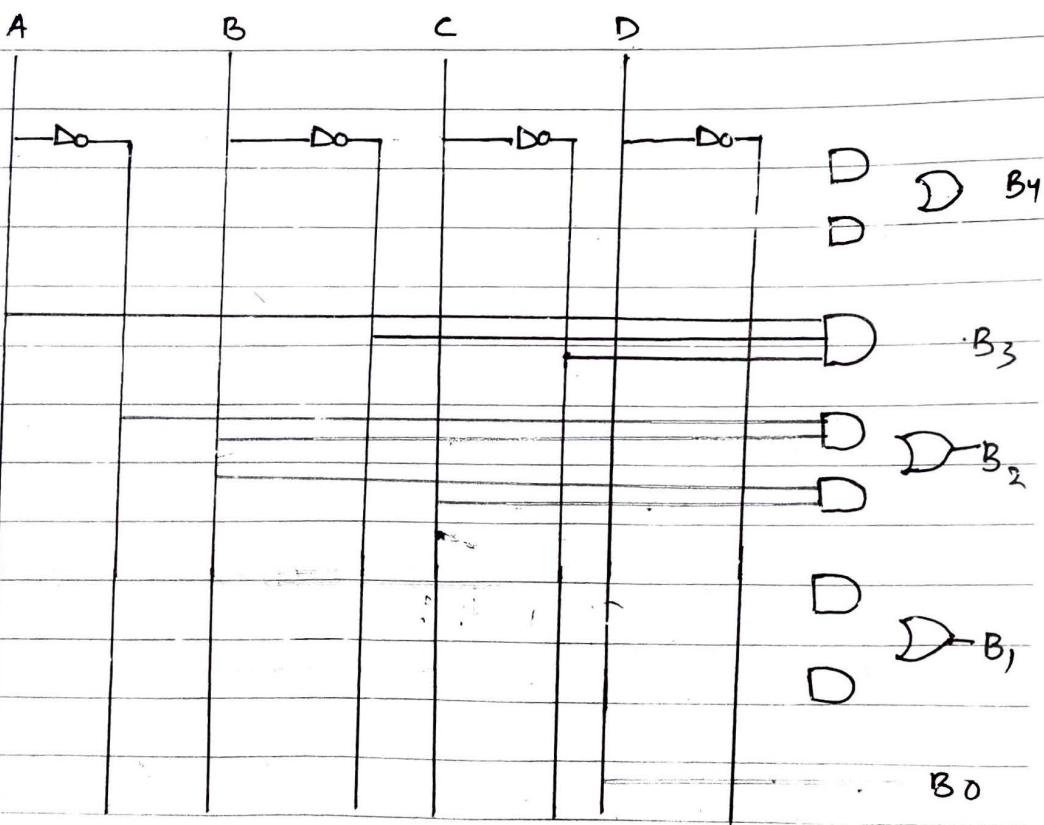
AB	00	01	11	10
CD	1			
00				
01				
11	1	1	1	
10	1	1	1	

 $\bar{A}B + BC$ $ABC + \bar{A}C$

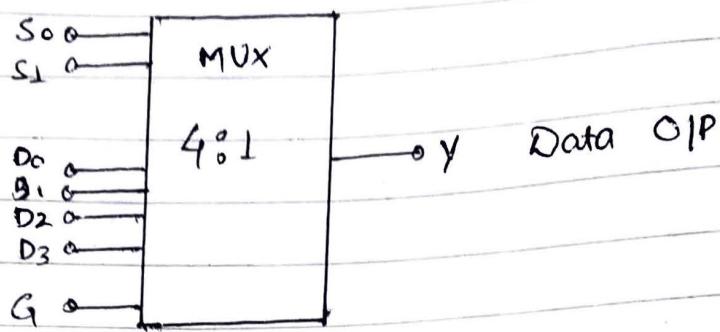
B0

AB	00	01	11	10
CD	1			
00				
01	1	1	1	1
11	1	1	1	1
10		X		

D



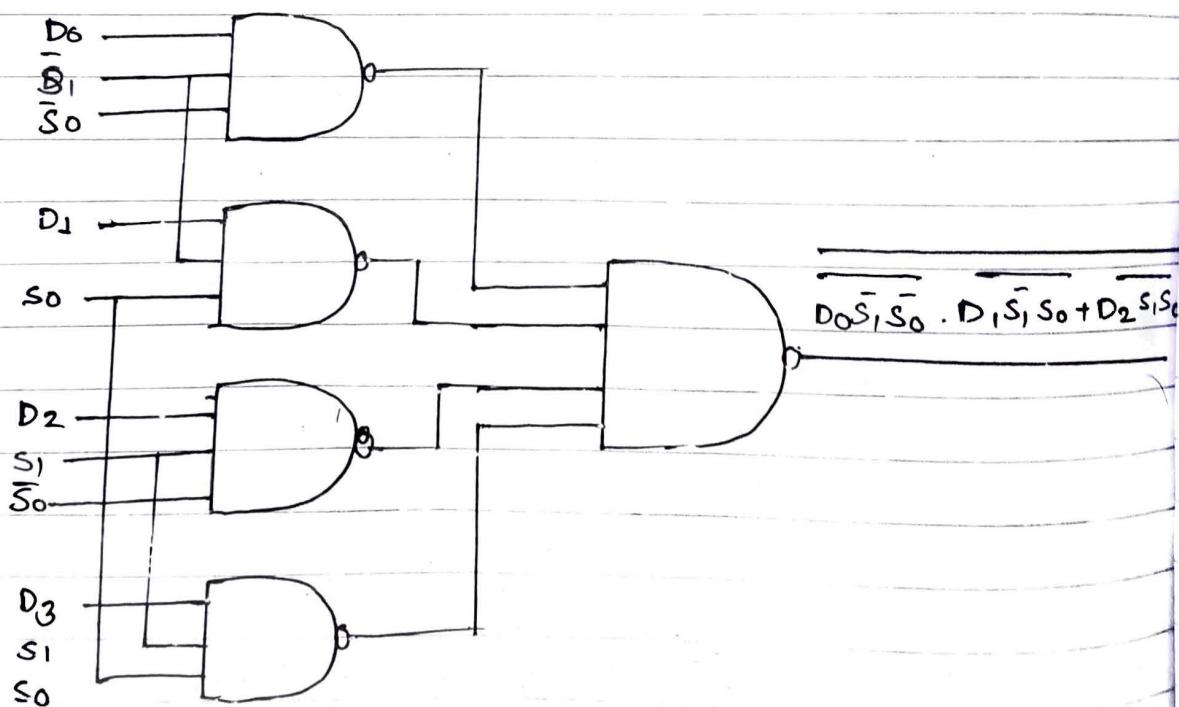
$$2^2 = 4$$



Data IP		O/P
$S_1 \quad S_0$		
0 0		D_0
0 1		D_1
1 0		D_2
1 1		D_3

$$Y = (D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0) \bar{G}$$

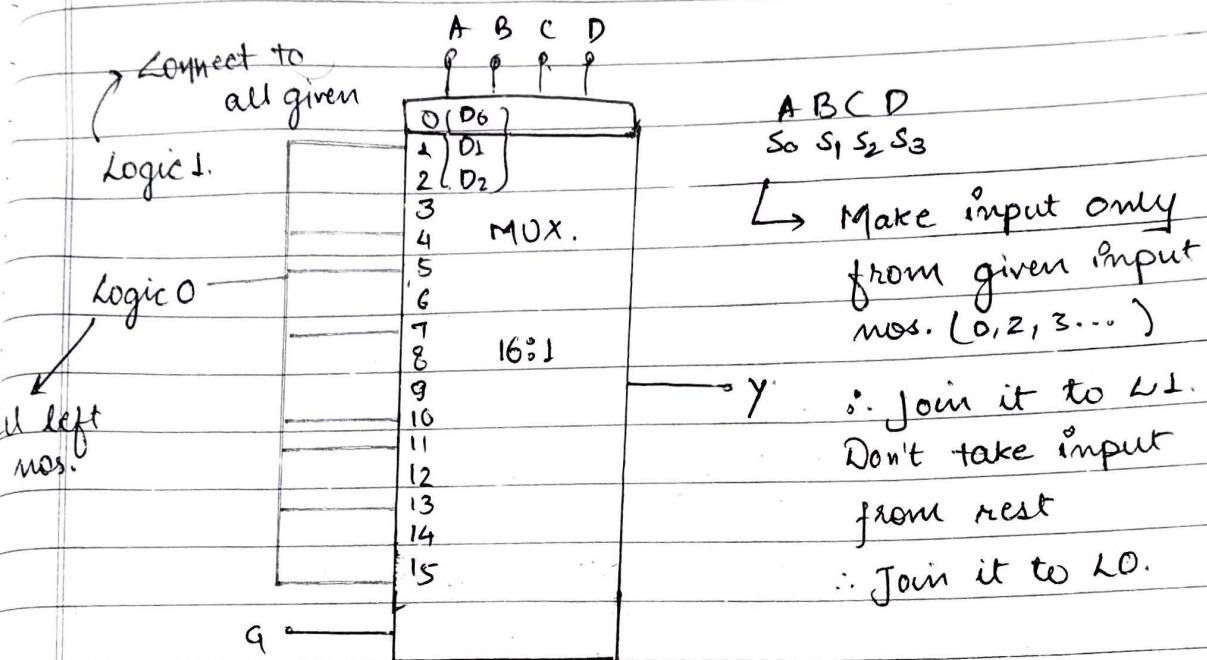
Ques Implement 4:1 MUX using NAN D gate



Implement MUX expression

$$f(A, B, C, D) = \sum m(0, 2, 3, 6, 8, 9, 12, 14)$$

multiplexer to be used.



Ques Realize the logic funcⁿ of T.T

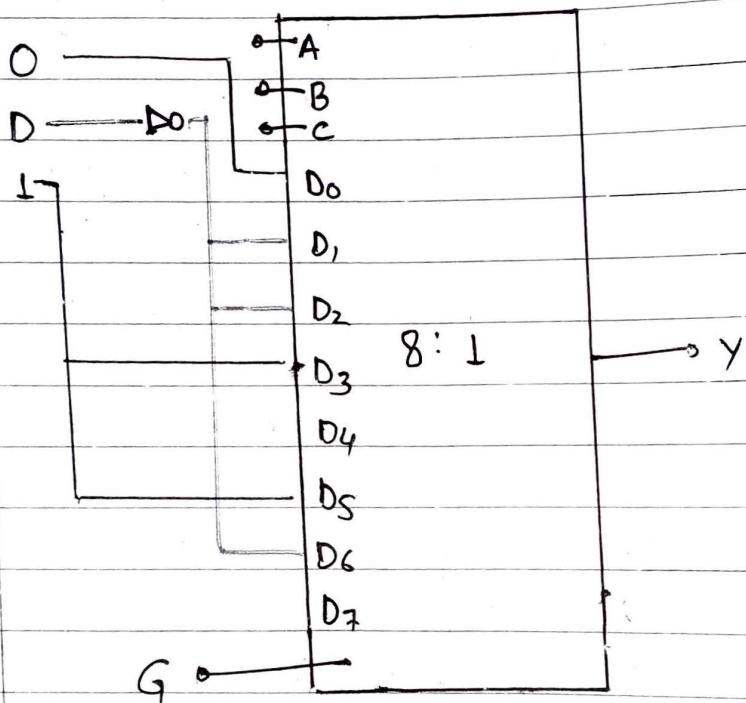
Ques Implement 16:1 MUX using 8:1 MUX.

A	B	C	D	O/P X	OP	→ All given ↩	ABCD	Y
0	0	0	0	0	0			
0	0	0	1	0	0			
0	0	1	0	1	1			
0	0	1	1	0	0			
0	1	0	0	1	1			
0	1	0	1	0	0			
0	1	1	0	1	1			
0	1	1	1	0	0			
1	0	0	0	1	1			
1	0	0	1	0	0			
1	0	1	0	1	1			
1	0	1	1	0	0			
1	1	0	0	1	1			
1	1	0	1	0	0			
1	1	1	0	0	1			
1	1	1	1	1	0			

X

✓

A	B	C	O/P
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



Ques Use 74HC151 to implement the following expression

$$Y = \bar{A}_2 \bar{A}_1 \bar{A}_0 + A_2 \bar{A}_1 \bar{A}_0 + \bar{A}_2 A_1 \bar{A}_0$$

or Implement above expression using 4:1 MUX

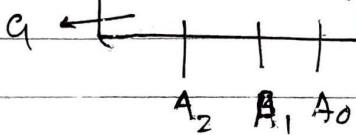
$$f(A_2, A_1 A_0) = \sum m(0, 2, 4)$$

$A_2 \quad A_1 \quad A_0 \quad 0/10$

0	0	0	1	$\left\{ \begin{array}{c} 0 \\ 4 \\ 2 \end{array} \right\}$
1	0	0	1	
0	1	0	1	

Logic 1

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



to 4:1 Convert

Back

* BCD Adder Using logic gates & K-map.

Ques Implement two 4 bit parallel BCD adder using K-map and logic Gates.

Decimal

C S₃ S₂ S₁ S₀

Y

0 → Valid
BCD

0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	0
10	0	1	0	1	0	1
11	0	1	0	1	1	1
12	0	1	1	0	0	1
13	0	1	1	0	1	1
14	0	1	1	1	0	1
15	0	1	1	1	1	1
16	1	0	0	0	0	1
17	1	0	0	0	1	1
18	1	0	0	1	0	1

1 → Invalid
BCD

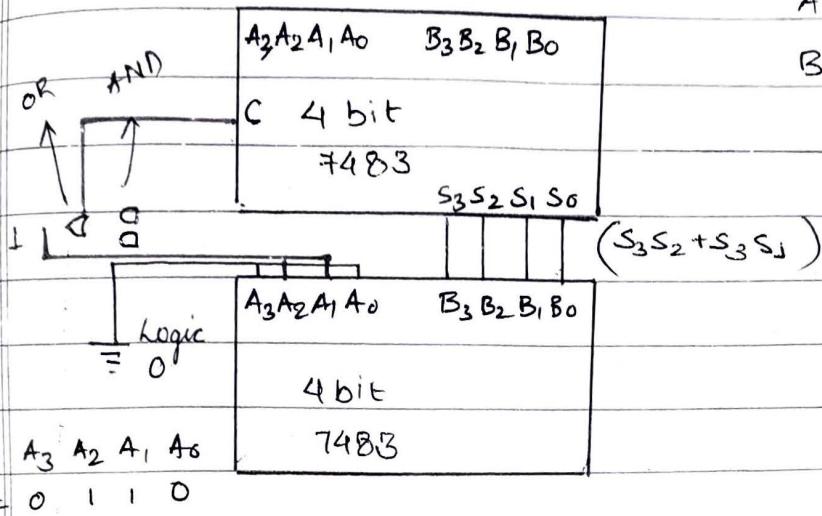
0 → 15 Case 1. (4 input bits used)

16 → 18 Case 2 (5 — do —)

		00	01	11	10
		S ₃ S ₂			
S ₁ S ₀	S ₁ S ₀
00	00
01	01
11	11	1	1	1	1
10	10	.	.	1	1

Y =

$$S_3S_2 + S_3S_1$$



$$A = 0111$$

$$B = 0011$$

$$1010 \rightarrow S$$

* BCD Subtraction using Logic Gates and K. Map.

→ BCD sub become BCD add^m when we take 9's complement of substrahend.

→ ~~Case 1:~~ 9's C of BCD no can be taken as
9 - x (x = that no)

Eg: 9's complement of 7
= 9 - 7 = 2

→ ~~Case 1:~~

If sum of minuend and 9's C of substrahend gives an invalid BCD code.

Eg:

9 - minuend
- 5 - substrahend

$$9's C of 5 = 9 - 5 = 4 \rightarrow 0100$$

$$9 \Rightarrow 1001$$

$$+ 0100$$

$$\hline 1101$$

Invalid

 $\therefore \text{Add } 6 \ (0110)$

$$\begin{array}{r}
 1101 \\
 + 0110 \\
 \hline
 10011
 \end{array}$$

END
 Around
Larry

→ Case 2: In above sum EAC is generated so ADD that carry to LSB!

$$\begin{array}{r}
 0011 \\
 + 0001 \\
 \hline
 100 \quad \rightarrow 4 \leftarrow \text{Ans}
 \end{array}$$

→ Case 3: $\frac{1}{2}$:

If MSB bit generated carry

$$\begin{array}{r}
 8 \\
 - 1 \\
 \hline
 \end{array}$$

$$9'sC \text{ of } 1 = 9-1 = 8 \rightarrow 1000$$

$$\begin{array}{r}
 8 \Rightarrow 1000 \\
 + 1000 \\
 \hline
 10000
 \end{array}$$

10000 (Invalid)

$$+ 0110$$

$$\begin{array}{r}
 10110
 \end{array}$$

$$0111 \rightarrow 7 \leftarrow \text{Ans.}$$

→ Case 3 :

If sum is valid BCD

{ When sum is valid BCD (< 9) }

{ it is always in 9'sC form }

$$\text{Eg: } \begin{array}{r} 4 \\ - 8 \\ \hline \end{array}$$

$$9'sC \text{ of } 8 = 9 - 8 = 1 \rightarrow 0001.$$

$$4 \Rightarrow 0100$$

$$+ 0001$$

$$\hline 0101 \Rightarrow 5 < 9 \text{ (Valid)} \quad \text{9-5}$$

↳ 9'sC of 4.

∴ It is in 9'sC form ans is -ve.

* Demultiplexer :

$M = O/P \text{ lines}$

$m = \text{selector}$

$$M = 2^m$$

74138 $\Rightarrow 1 : 8 \text{ DMUX}$

or

3 lines to 8 lines

Decoder.

74154 $\Rightarrow 1 : 16 \text{ DMUX}$

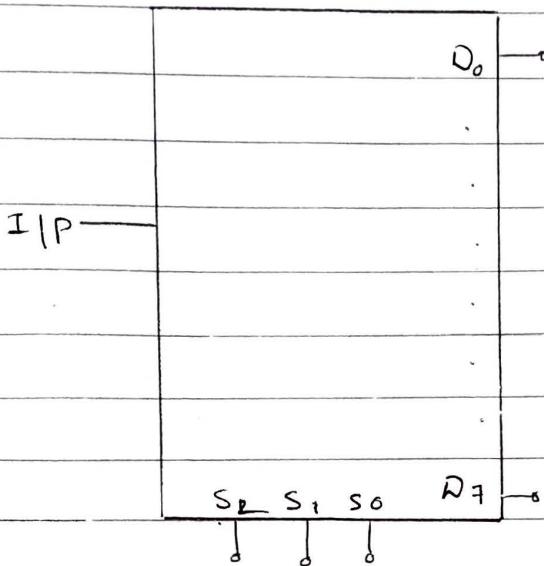
or

4 lines to 16 lines

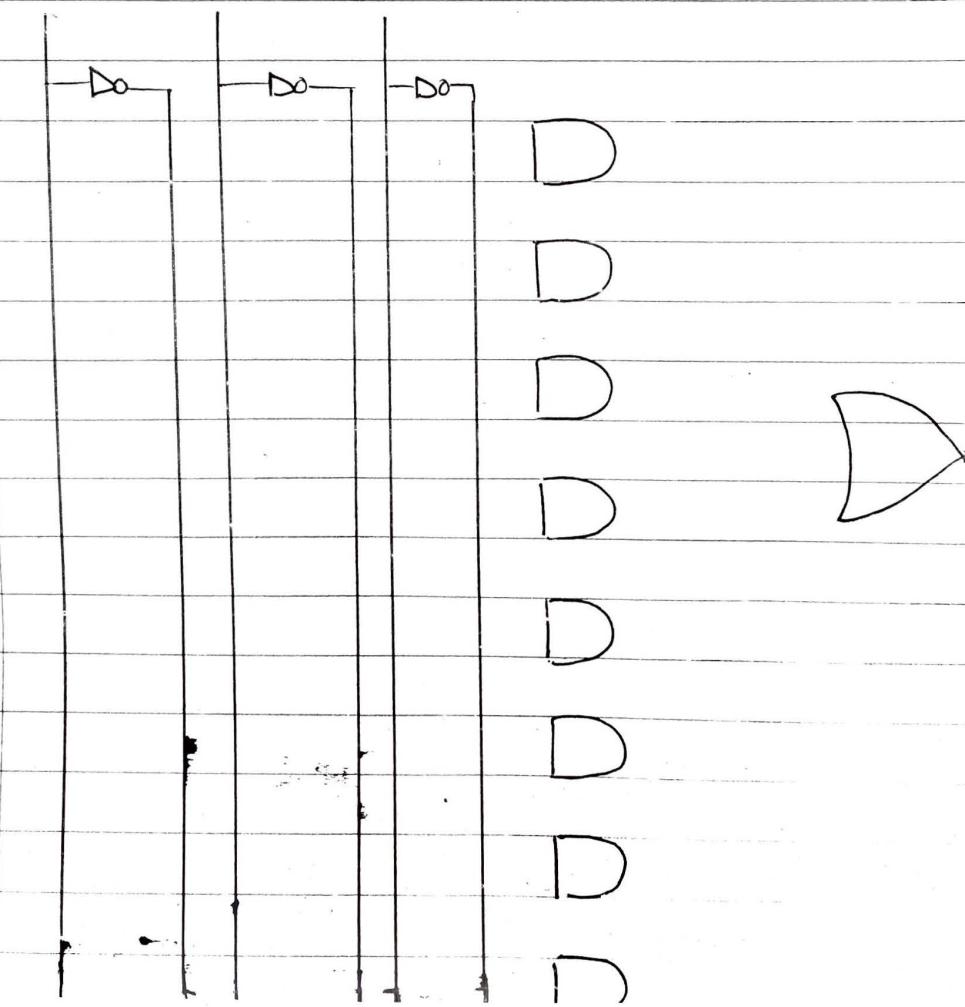
Decoder

for all lines having 1.

$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 + \bar{S}_2 \bar{S}_1 S_0 + \bar{S}_2 S_1 \bar{S}_0 + \bar{S}_2 \bar{S}_1 S_0 + S_2 \bar{S}_1 \bar{S}_0 \\ + S_2 \bar{S}_1 S_0 + S_2 S_1 \bar{S}_0 + S_2 S_1 S_0$$



$S_2 \quad S_1 \quad S_0$



Q4 Implement full adder using 3 lines to 8 lines Decoder.

I/P			O/P		Cout
A	B	Cin	Σ		
0	0	0	0		0
0	0	1	1		0
0	1	0	1		0
0	1	1	0		1
1	0	0	1		0
1	0	1	0		1
1	1	0	0		1
1	1	1	1		1

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

$$\Sigma_{\text{cout}} = \Sigma(3, 5, 6, 7)$$

