

# Chapter 08 MSI logic circuits

MSI  $\rightarrow$  Medium Scale Integrated

SSI  $\rightarrow$  Small Scale Integrated

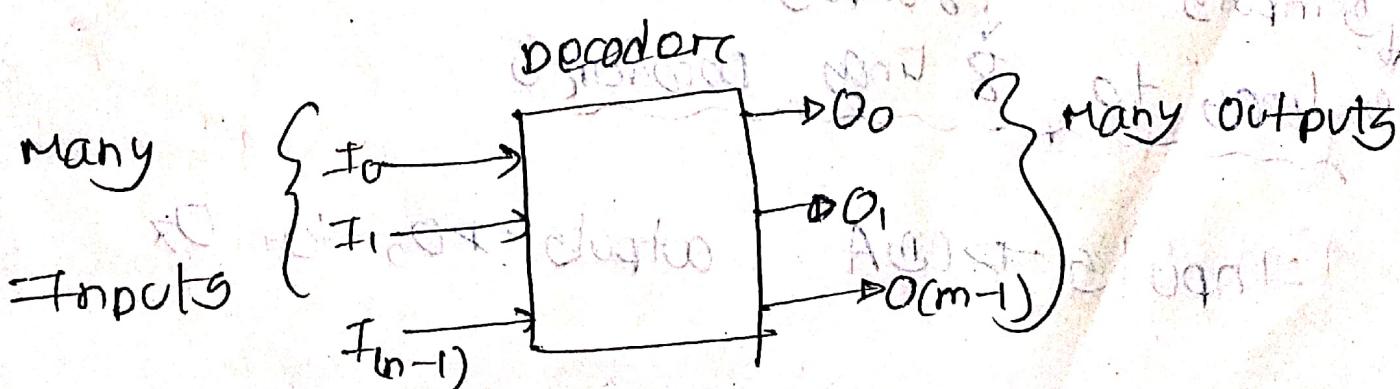
LSI  $\rightarrow$  Large Scale Integrated

VLSI  $\rightarrow$  Very Large Scale Integrated.

$\hookrightarrow$  [4-2/4-1]  $\rightarrow$  tag mark special course  $\rightarrow$  VLSI

(1) Decoder (2) Multiplexor (MUX) (3) Demultiplexor (DEMUX)

(1) Decoder



At a time only one of the output will be active according to the corresponding

combination of inputs.

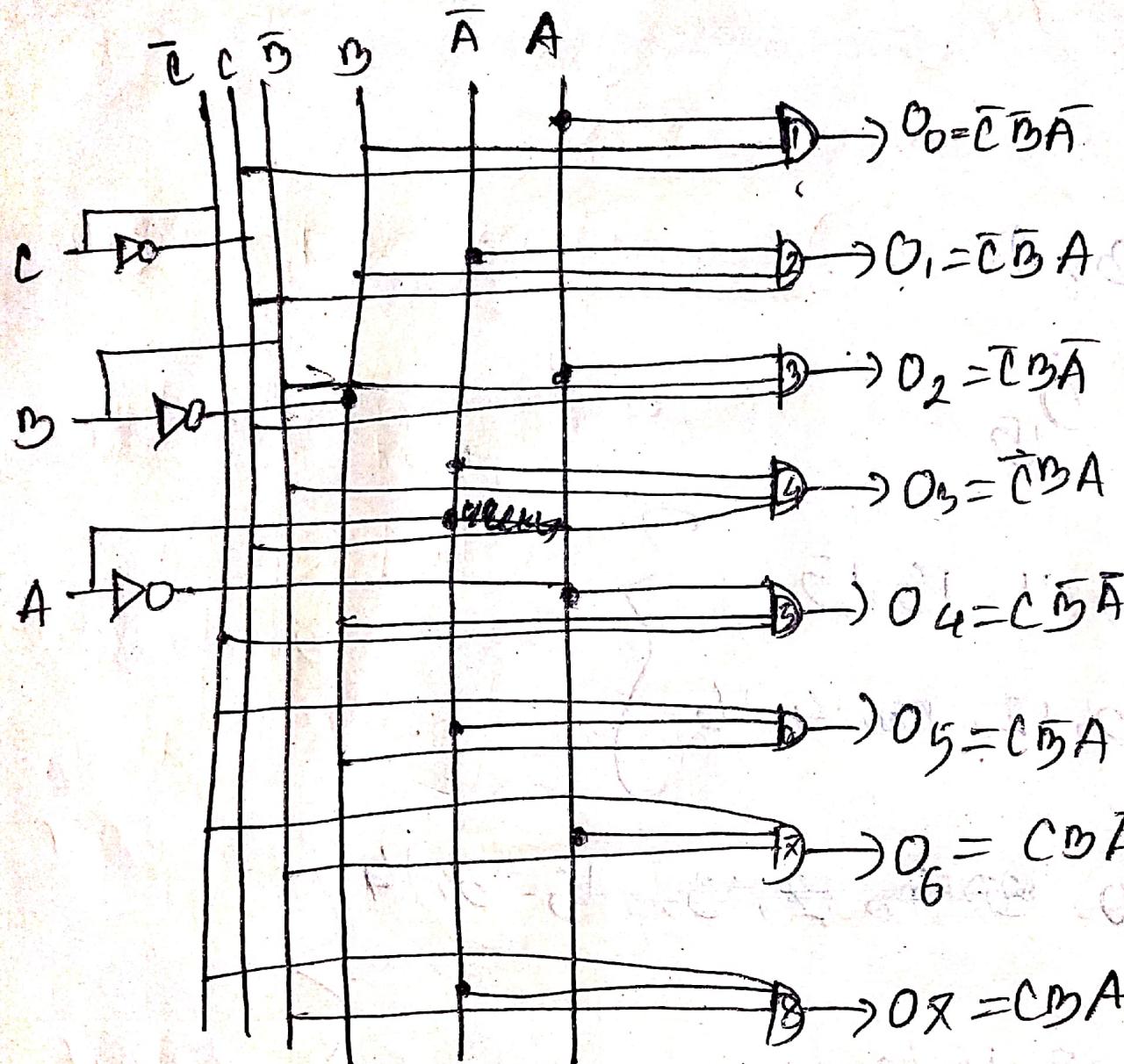
No. of Input	No. of Output
2	$4 = 2^2$
3	$8 = 2^3$
4	$16 = 2^4$
5	$32 = 2^5$

$$\text{No. of outputs} = 2^{\text{number of inputs}}$$

3 inputs      8 outputs  
3 lines to 8 lines Decoder

Inputs  $\rightarrow$  CBA, outputs  $\rightarrow$  O<sub>0</sub> to O<sub>7</sub>

Active



C	B	A	OUTPUT
0	0	0	0 <sub>0</sub>
0	0	1	0 <sub>1</sub>
0	1	0	0 <sub>2</sub>
0	1	1	0 <sub>3</sub>
1	0	0	0 <sub>4</sub>
1	0	1	0 <sub>5</sub>
1	1	0	0 <sub>6</sub>
1	1	1	0 <sub>7</sub>

Active high  
output

28-02-2022

Inputs  $\rightarrow D, C, B, A$

Outputs  $\rightarrow O_0 \text{ to } O_{15}$

$\bar{A} \rightarrow 1, 3, 5, 7, 9, 11, 13, 15$

$A \rightarrow 2, 4, 6, 8, 10, 12, 14, 16$

Out put A

$\bar{B} \rightarrow 1, 2, 5, 6, 9, 10, 12, 13, 16, 13, 14$

$B \rightarrow 3, 4, 7, 8, 11, 12, 15, 6$

$\bar{C} \rightarrow 1 \text{ to } 4, 9 \text{ to } 12$

$C \rightarrow 5 \text{ to } 8, \text{ then } 13 \text{ to } 16$

$\bar{D} \rightarrow 1 \text{ to } 8$

$D \rightarrow 9 \text{ to } 16$

4 inputs

10 outputs

4 lines to 10 line decoder

Input  $\rightarrow$  D, C, B, A

Outputs  $\rightarrow$  O<sub>0</sub> to O<sub>9</sub>

$\bar{A} \rightarrow$  D1, 3, 5, 7, 9 }

A  $\rightarrow$  D2, 4, 6, 8, 10

$\bar{B} \rightarrow$  1, 2, 5, 6, 9, 10 }

B  $\rightarrow$  3, 4, 7, 8,

$\bar{C} \rightarrow$  1 to 4 odd, 9, 10 }

C  $\rightarrow$  5 to 8

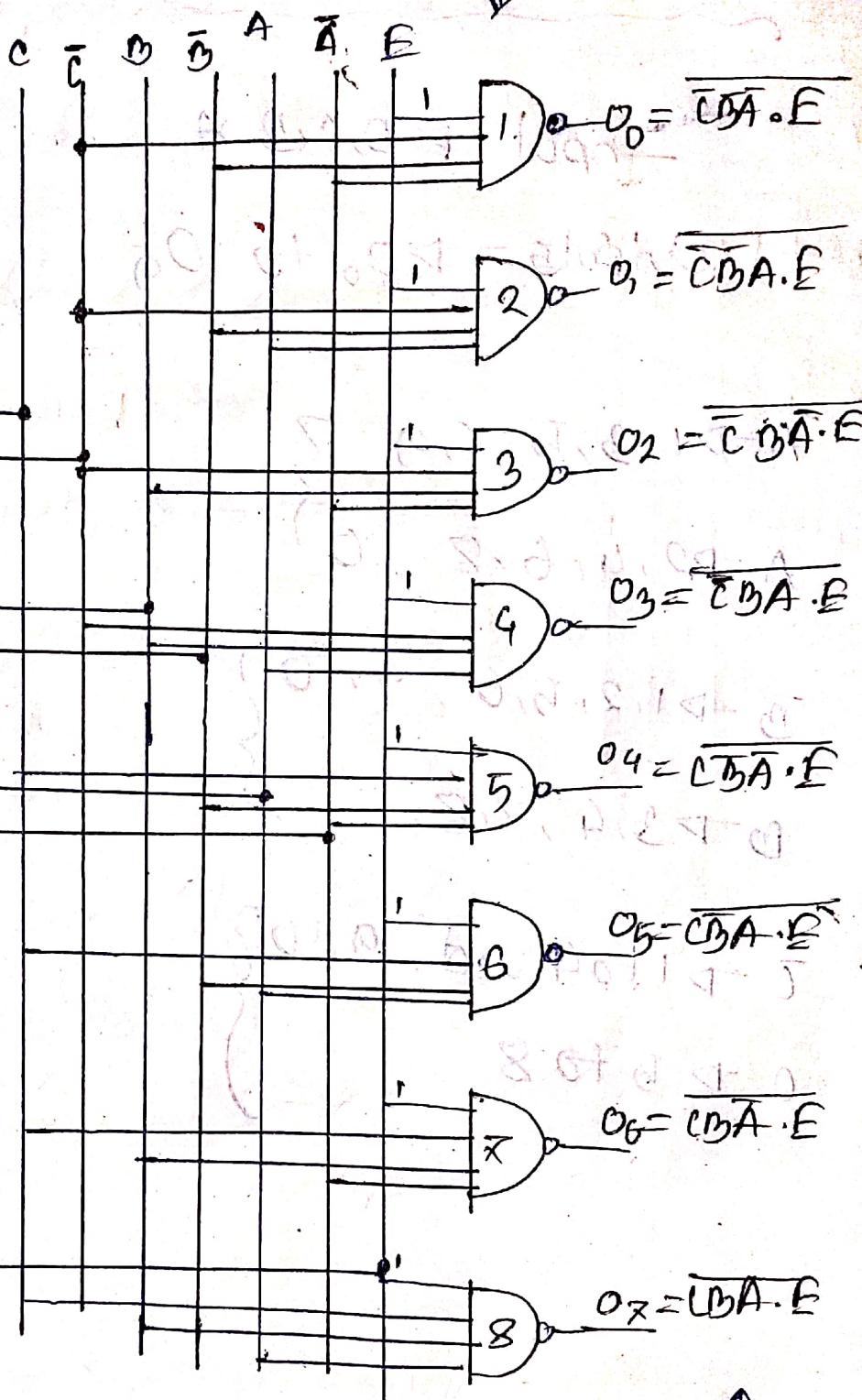
10 AND Gates

(1 to 9)

D	C	B	A	
0	0	0	0	O <sub>0</sub>
0	0	0	1	O <sub>1</sub>
0	0	1	0	O <sub>2</sub>
0	0	1	1	O <sub>3</sub>
0	1	0	0	O <sub>4</sub>
0	1	0	1	O <sub>5</sub>
0	1	1	0	O <sub>6</sub>
0	1	1	1	O <sub>7</sub>
1	0	0	0	O <sub>8</sub>
1	0	0	1	O <sub>9</sub>

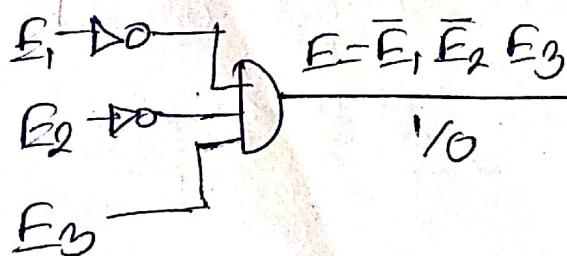
# IC #74138 (Decoder) Internal Circuit

সাইজে ক্ষেত্রফল  
NAND USE করাব



## Enable Inputs

E<sub>1</sub>, E<sub>2</sub>, E<sub>3</sub>



## Active Mode

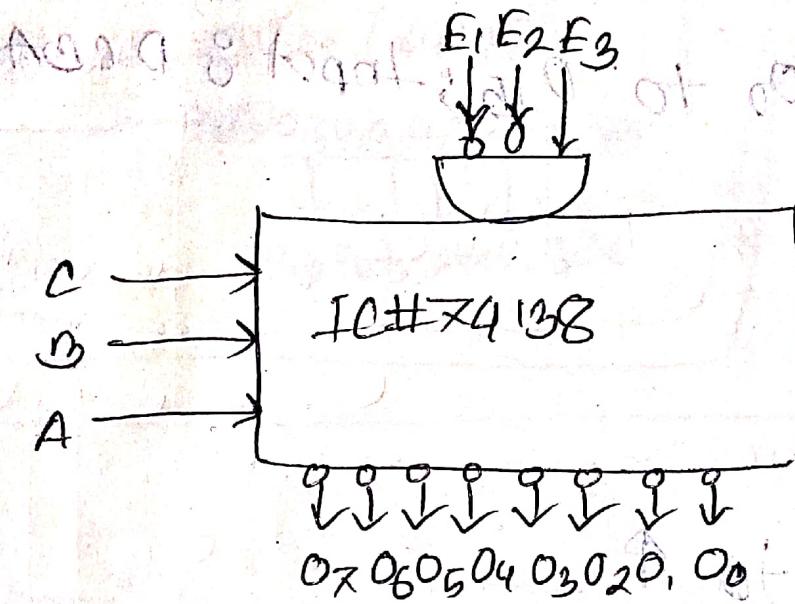
E=1, E<sub>1</sub>=0

E<sub>1</sub>=0, E<sub>2</sub>=0, E<sub>3</sub>=1

Other combination  
IC will be inactive

Active low  
output

# Block Diagram of IC#74138



Ex-1

{  $E_1 = E_2 = 0, E_3 = 1 \rightarrow$  Active Mode }

$$C = B = 1, A = 0$$

$\rightarrow$  Which output will be active?

Ans: O<sub>6</sub>

Ex-2

{  $E_1 = 0, E_2 = 1, E_3 = 1 \rightarrow$  Inactive Mode }

$$C = 1, B = 0, A = 1$$

$\rightarrow$  Which output will be active?

Ans: No output will be active.

# Design 4 lines to 16 lines Decoder Using IC #74138

You can use other logic gates, if necessary.

→ 2 IC's Output O<sub>0</sub> to O<sub>15</sub>; Input O CBA

02-02-2021

Design 4 lines to 16 lines

Truth Table

O C	B A	Active Output
0 0	0 0	O <sub>0</sub>
0 0	0 1	O <sub>1</sub>
0 0	1 0	O <sub>2</sub>
0 0	1 1	O <sub>3</sub>
0 1	0 0	O <sub>4</sub>
0 1	0 1	O <sub>5</sub>
0 1	1 0	O <sub>6</sub>
0 1	1 1	O <sub>7</sub>
1 0	0 0	O <sub>8</sub>
1 0	0 1	O <sub>9</sub>
1 0	1 0	O <sub>10</sub>
1 0	1 1	O <sub>11</sub>
1 1	0 0	O <sub>12</sub>
1 1	0 1	O <sub>13</sub>
1 1	1 0	O <sub>14</sub>
1 1	1 1	O <sub>15</sub>

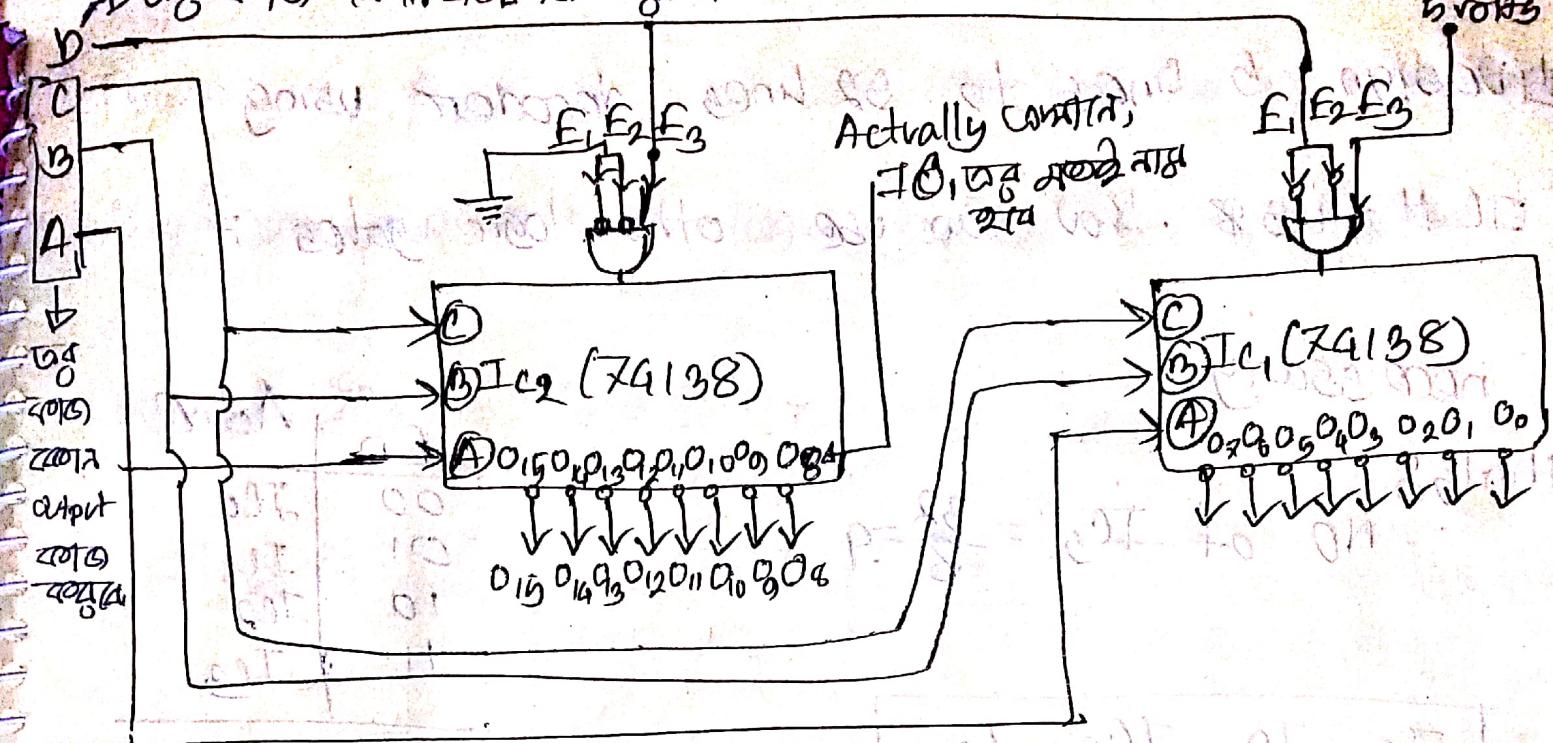
I<sub>C1</sub> → Active

I<sub>C2</sub> → Inactive

I<sub>C1</sub> → Inactive

I<sub>C2</sub> → Active

→ ପ୍ରତି କଣ୍ଟ ମୋଟ ଇନ୍‌ପୁଟ ଯଦ୍ୱାରା,



### Operation 3

If D=0,

then Port 1 = f<sub>1</sub>      E<sub>1</sub>=E<sub>2</sub>=0, E<sub>3</sub>=1

for IC<sub>2</sub>      E<sub>1</sub>=E<sub>2</sub>=E<sub>3</sub>=0

∴ IC<sub>1</sub> is active & IC<sub>2</sub> is inactive

If D=1,

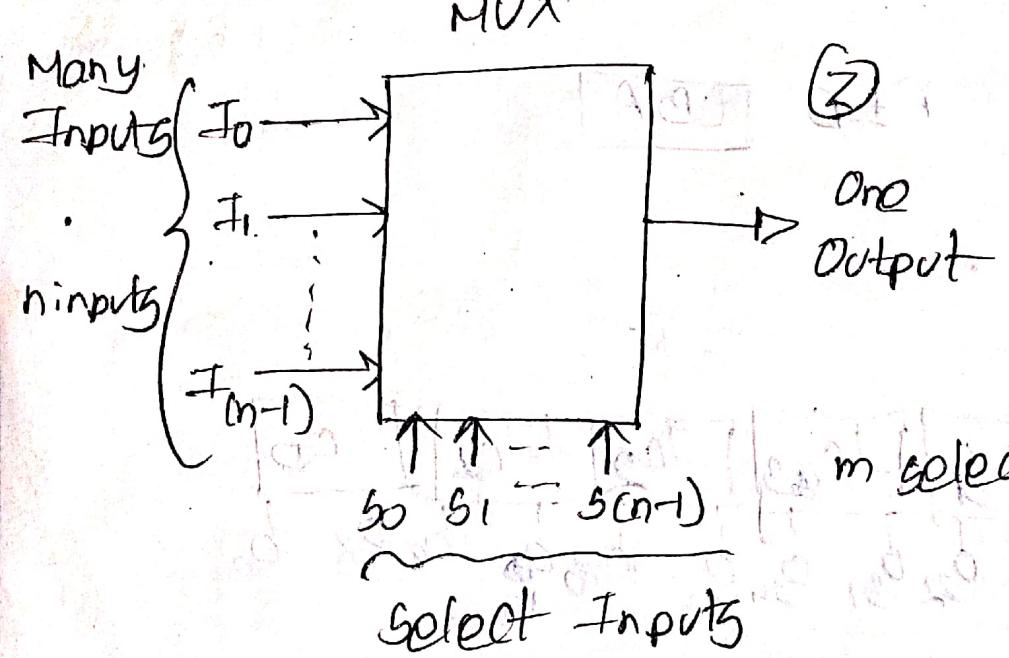
then for IC<sub>1</sub>, E<sub>1</sub>=E<sub>2</sub>=E<sub>3</sub>=1

∴ for IC<sub>2</sub>, E<sub>1</sub>=E<sub>2</sub>=0, E<sub>3</sub>=1

∴ IC<sub>1</sub> is inactive and IC<sub>2</sub> is active

Multiplexor වන් Input යෙනුයේ Output වැයි!

## (2) Multiplexor (MUX)



MUX is a device where there are many inputs one output and some select inputs. At a time one of the inputs goes to the output according to the corresponding combination of select inputs.

No of select inputs      No of normal inputs

Inputs

Inputs

2

$$4 = 2^r$$

3

$$8 = 2^3$$

4

$$16 = 2^4$$

5

$$32 = 2^5$$

No of Inputs = 2

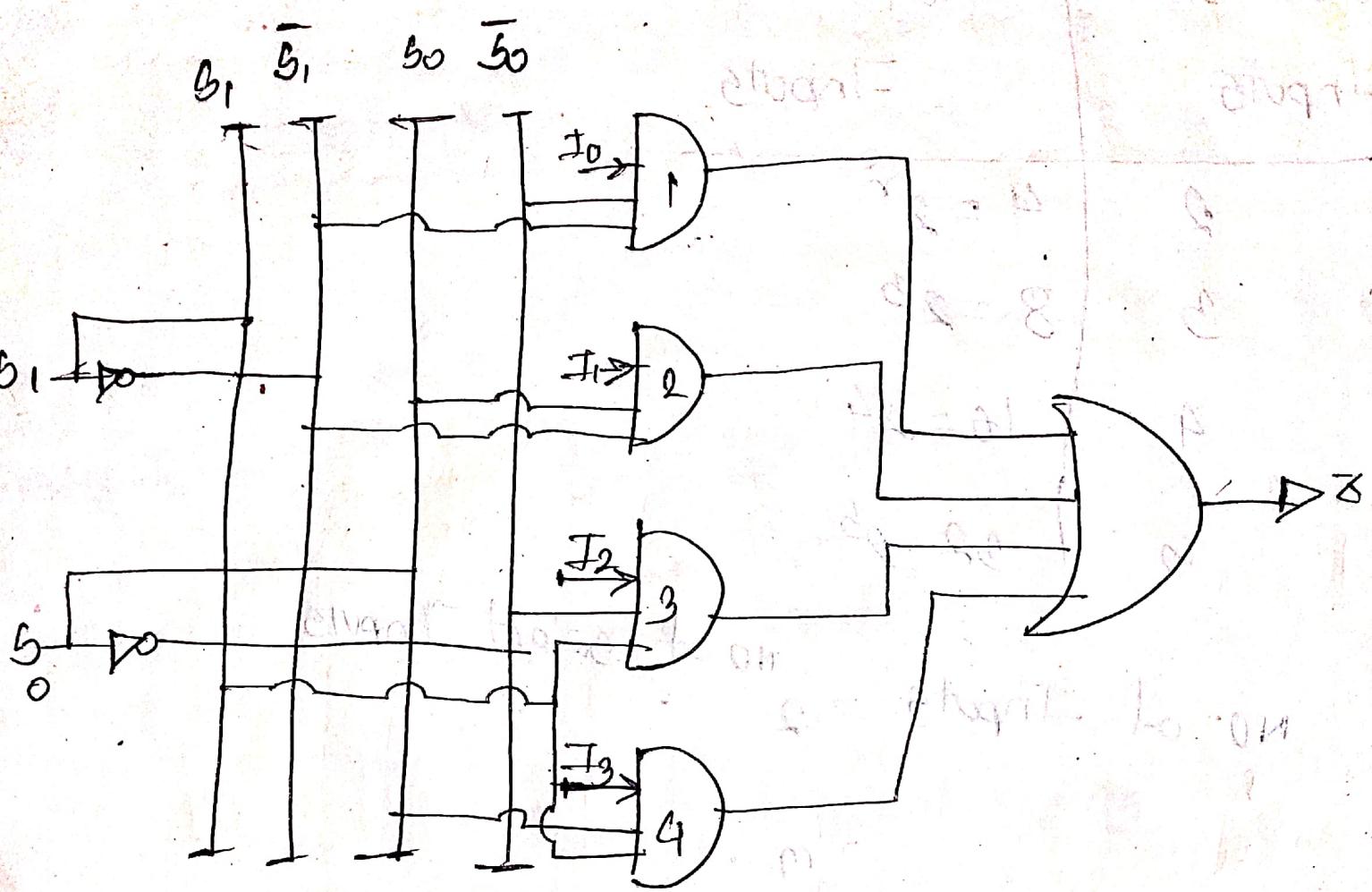
No of select Inputs

$$\leq n=2^m$$

4 inputs MUX 8

$$Z = \bar{S}_1 \bar{S}_0 \cdot I_0 + \bar{S}_1 S_0 \cdot I_1 + S_1 \bar{S}_0 \cdot I_2 + S_1 S_0 \cdot I_3$$

$S_1$	$S_0$	$Z =$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$



4 Input MUX

DLS08

Chapter [3-7] [M&M term  
Syllabus]

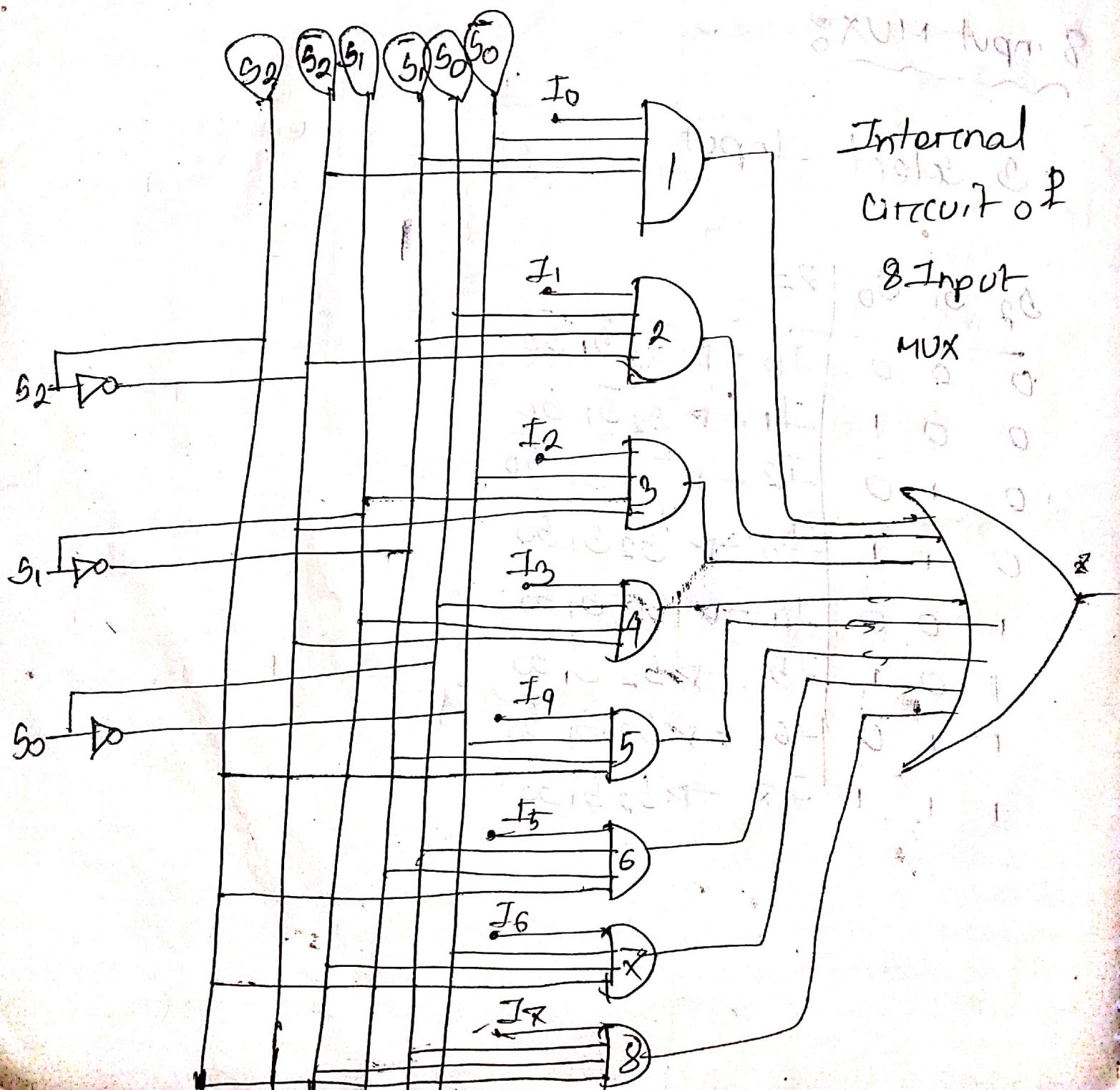
06-03-2022

8 input MUX

3 select Input

$S_2$	$S_1$	$S_0$	$I =$
0	0	0	$I_0 \rightarrow \bar{S}_2 \bar{S}_1 \bar{S}_0$
0	0	1	$I_1 \rightarrow \bar{S}_2 \bar{S}_1 S_0$
0	1	0	$I_2 \rightarrow \bar{S}_2 S_1 \bar{S}_0$
0	1	1	$I_3 \rightarrow \bar{S}_2 S_1 S_0$
1	0	0	$I_4 \rightarrow S_2 \bar{S}_1 \bar{S}_0$
1	0	1	$I_5 \rightarrow S_2 \bar{S}_1 S_0$
1	1	0	$I_6 \rightarrow S_2 S_1 \bar{S}_0$
1	1	1	$I_7 \rightarrow S_2 S_1 S_0$

$$Z = \bar{S}_2 \bar{S}_1 \bar{S}_0 \cdot I_0 + \bar{S}_2 S_1 \bar{S}_0 \cdot I_1 + \bar{S}_2 S_1 S_0 \cdot I_2 + \bar{S}_2 S_1 S_0 \cdot I_3 \\ + S_2 \bar{S}_1 \bar{S}_0 \cdot I_4 + S_2 \bar{S}_1 S_0 \cdot I_5 + S_2 S_1 \bar{S}_0 \cdot I_6 \\ + S_2 S_1 S_0 \cdot I_8$$

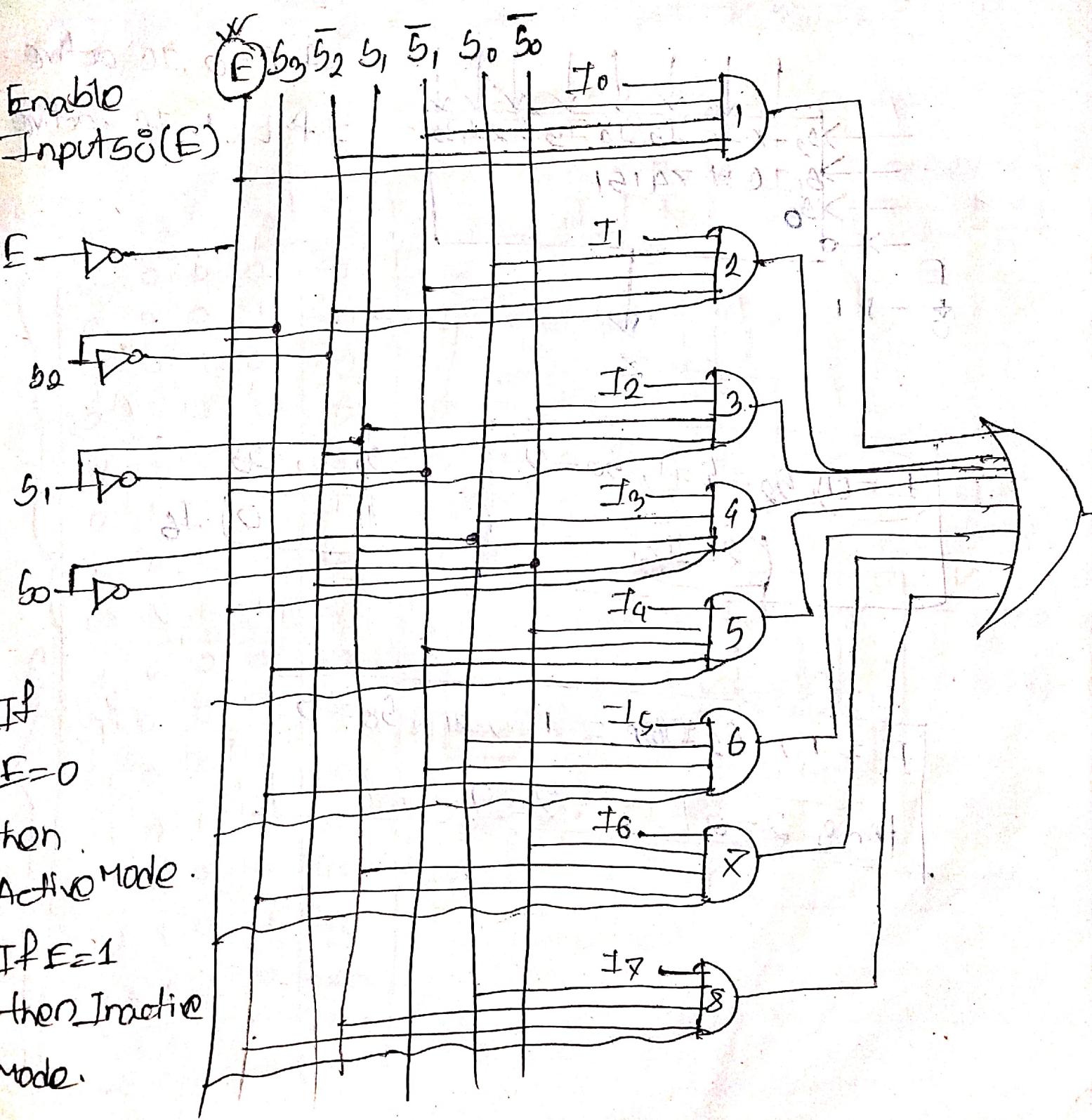


IC#X4151

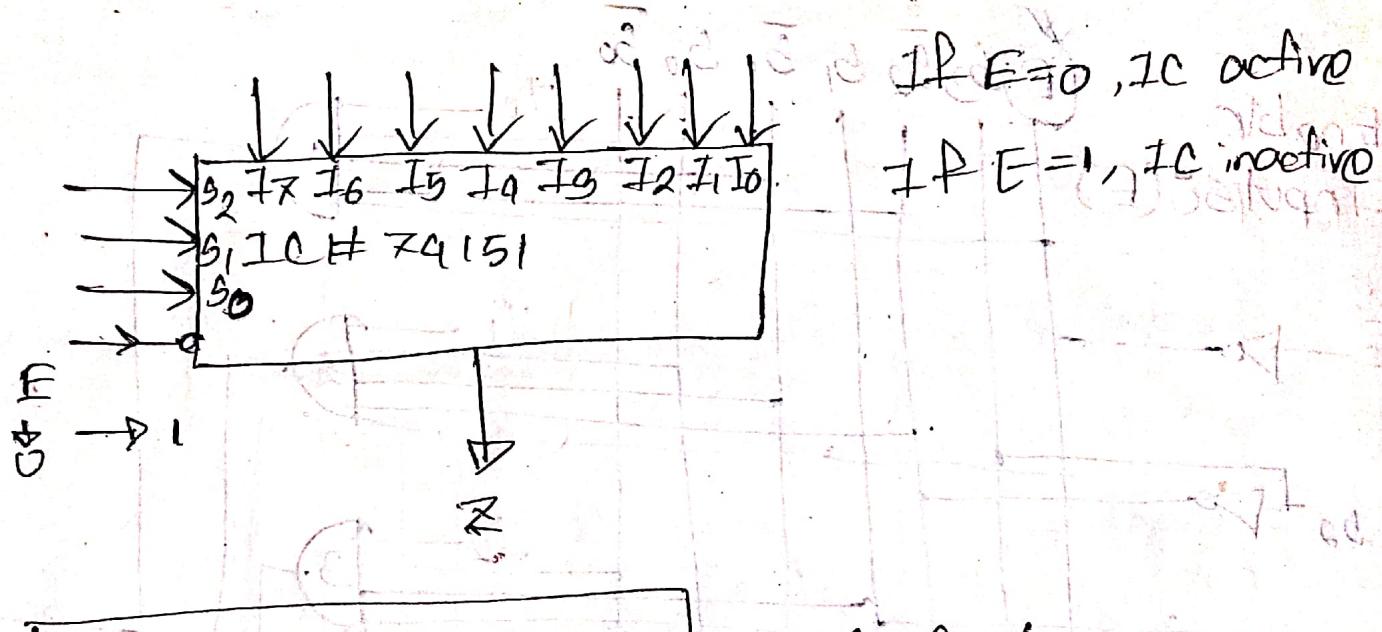
চিত্র,

মীকেন কুণ্ডা Internal Circuit বিবরণ।

#Internal circuit of IC#X4151#



Block Diagram : (किसी Input पर Output किसे बनाए)



$$E = 0, S_2 = S_1 = 1, S_0 = 0$$

$$Z = I_6$$

$$\begin{matrix} S_2 & S_1 & S_0 \\ 1 & 1 & 0 \end{matrix} \rightarrow I_6$$

$$E = 1, S_2 = 0, S_1 = S_0 = 0$$

$$Ans \ Z = 0$$

$$Z = ?$$

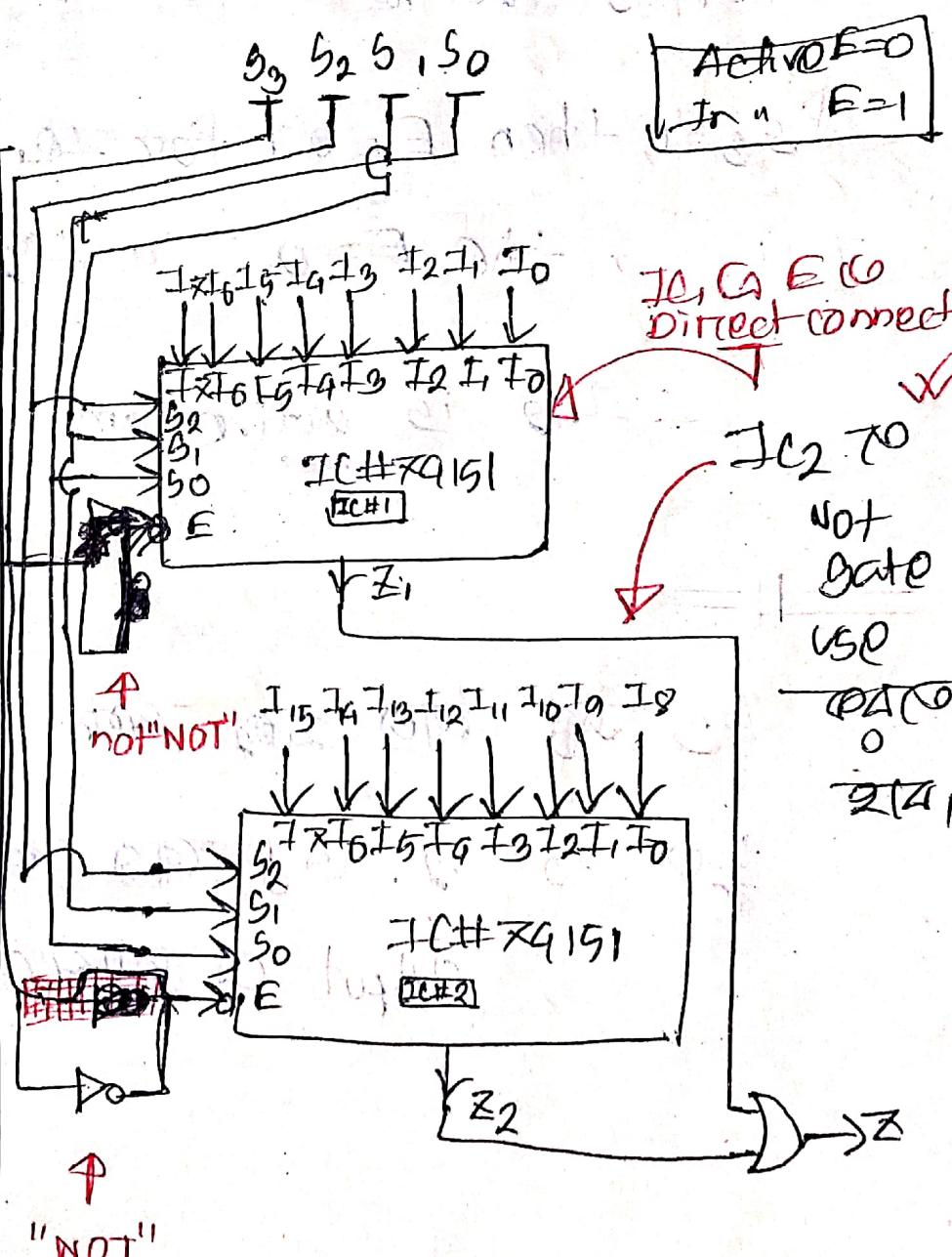
123-03-23

Design 16 input MUX using IC#74151. You can use other logic gates, if necessary.

$\rightarrow$  No of IO =  $\frac{16}{2} = 2$ ; Select input  $\rightarrow S_3, S_2, S_1, S_0$

Inputs  $\rightarrow I_0 \text{ to } I_{15}$

$S_3$	$S_2$	$S_1$	$S_0$	$Z$
0	0	0	0	$I_0$
0	0	0	1	$I_1$
0	0	1	0	$I_2$
0	0	1	1	$I_3$
0	1	0	0	$I_4$
0	1	0	1	$I_5$
0	1	1	0	$I_6$
0	1	1	1	$I_7$
1	0	0	0	$I_8$
1	0	0	1	$I_9$
1	0	1	0	$I_{10}$
1	0	1	1	$I_{11}$
1	1	0	0	$I_{12}$
1	1	0	1	$I_{13}$
1	1	1	0	$I_{14}$
1	1	1	1	$I_{15}$



operations

If  $s_3=0$ , then  $E=0$  for  $T_{C1}$

if  $s_3=1$ , the  $E=1$  for  $T_{C2}$

$\therefore T_{C1}$  is active and  $T_{C2}$  is inactive.

If  $s_3=1$ , then  $E=0$  for  $T_{C1}$

if  $s_3=0$ , the  $E=0$  for  $T_{C2}$ ,

$\therefore T_{C2}$  is active and  $T_{C1}$  is inactive

$s_3$  এর বিন্দু  $\rightarrow$   $T_{C2}$  কারণ  $T_{C1}$  active

$s_2 \& s_0 - s_2$  এর বিন্দু যাতে নির্মাণ Input

Output  $\Rightarrow$  যোদ্ধা

27-03-2022

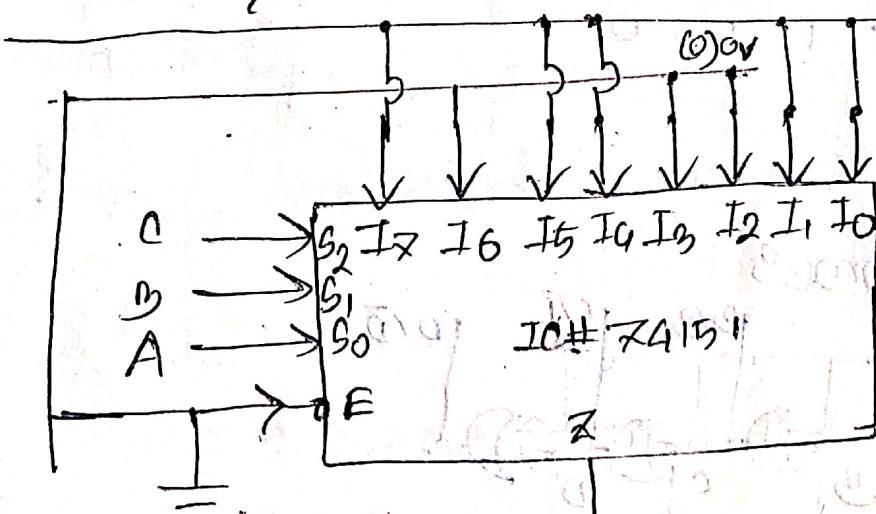
## #Application of MUX

### (i) Logic function generation

$$y = \overline{ABC} + \overline{ABC} + ABC_1 + A\overline{BC} + A\overline{B}\overline{C}$$

IC# X4151

5vols(C1)



C(B)A

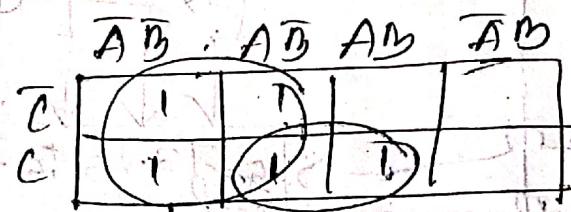
<u>S<sub>2</sub> S<sub>1</sub> S<sub>0</sub></u>	<u>Z</u>
0 0 0	I <sub>0</sub>
0 0 1	I <sub>1</sub>
0 1 0	I <sub>2</sub>
0 1 1	I <sub>3</sub>
1 0 0	I <sub>4</sub>
1 0 1	I <sub>5</sub>
1 1 0	I <sub>6</sub>
1 1 1	I <sub>7</sub>

C	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Implement  $F(A, B, C)$

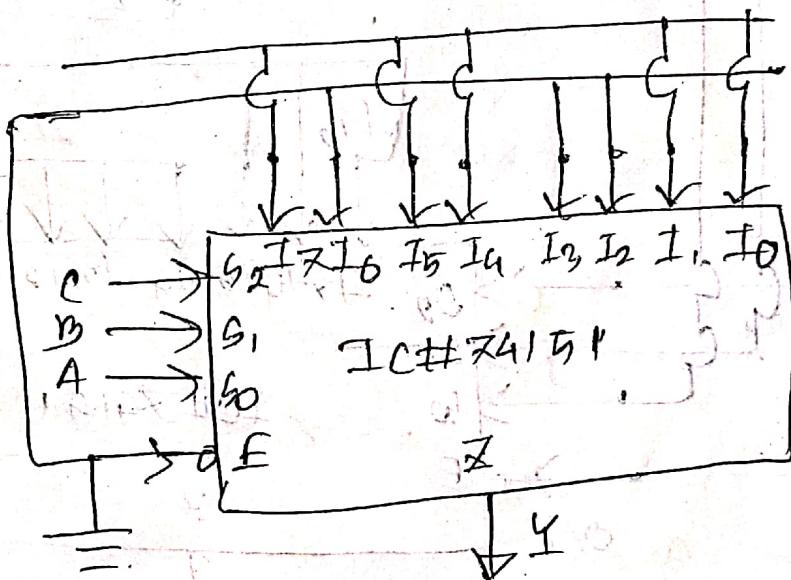
=  $C(0,1,4,5,2)$  using

IC#74151



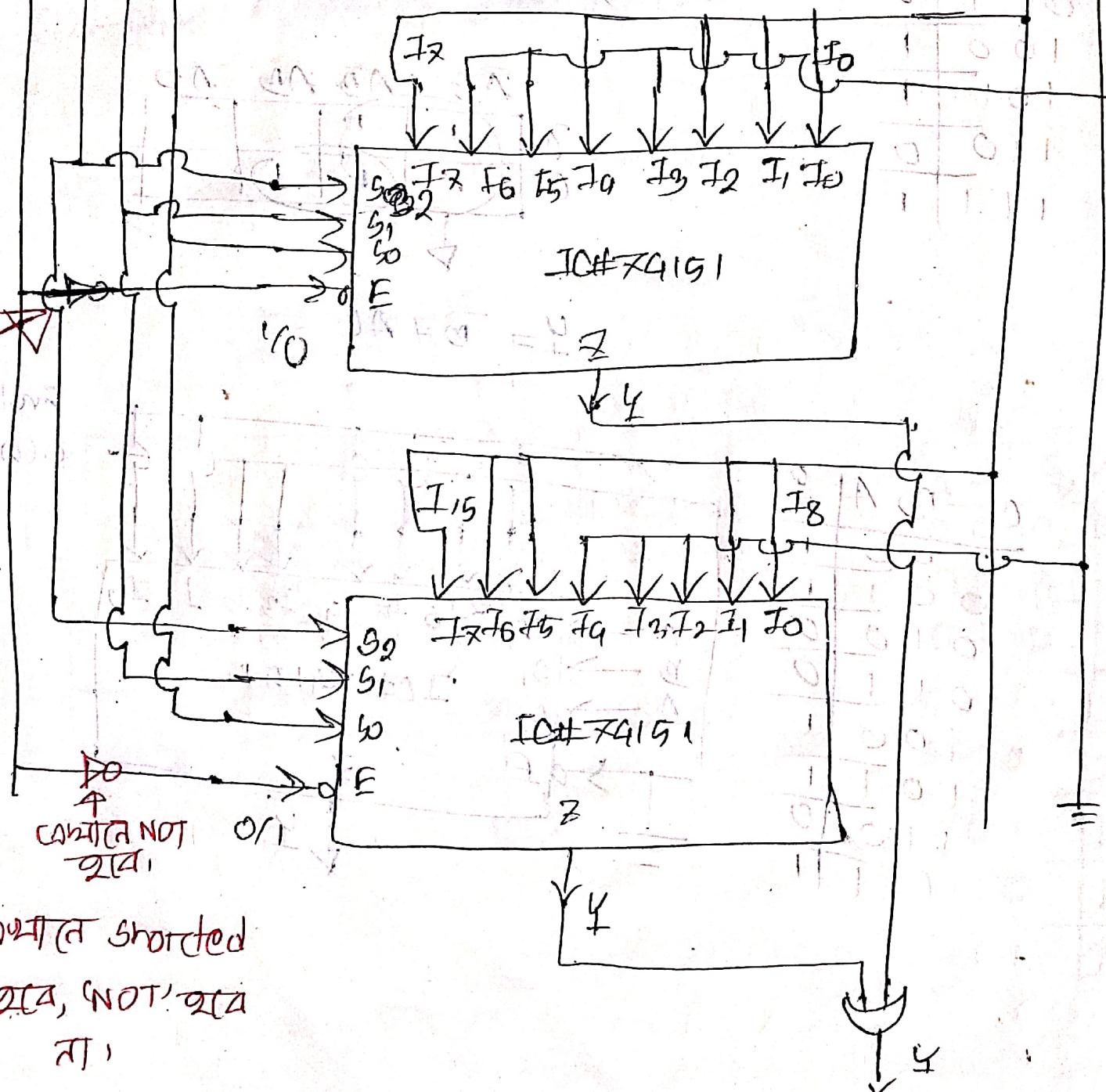
$$Y = \overline{B} + AC$$

C	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



$$\#PA(B, C, D) = \Sigma(0, 1, 2, 4, 7, 8, 9, 13, 14, 15)$$

D C B A  
S3 S2 S1 S0



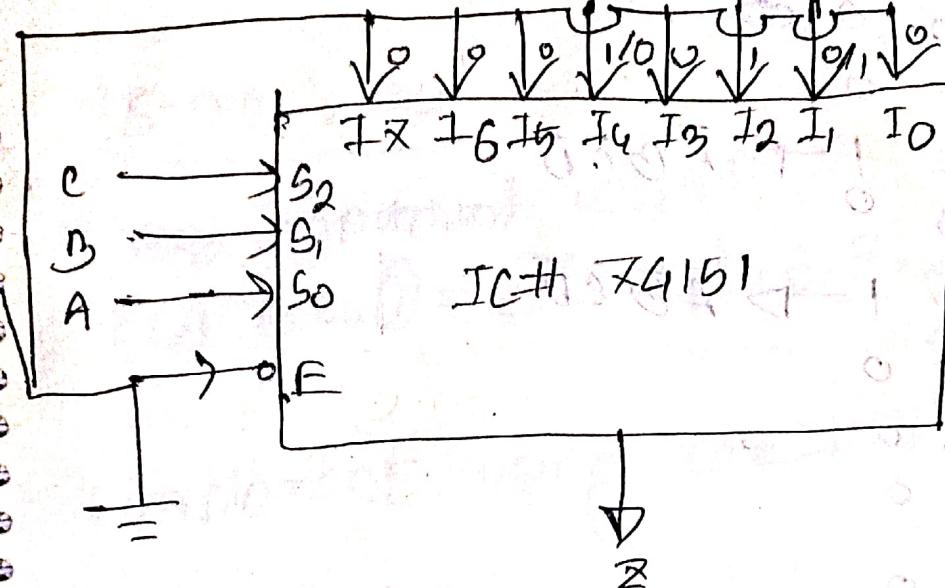
प्रथम shorted

-DCA, NOT(DCA)

AT

P.28 Q1

5 Volts(1)



(a) Draw truth table

(b) Simplify using K-Map or otherwise

(c)  $\bar{D} \oplus 1$

D	C	B	A	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>
0	0	0	0	10				
0	0	0	1		11			
0	0	1	0			12		
0	0	1	1				13	
0	1	0	0					14
0	1	0	1					15
0	1	1	0					16
0	1	1	1					17

(d)  $\bar{A} \oplus B$

D	C	B	A	Z
0	0	00	I <sub>0</sub>	0
0	0	01	I <sub>1</sub>	0
0	0	10	I <sub>2</sub>	1 → $\bar{A}B\bar{C}\bar{D}$
0	0	11	I <sub>3</sub>	0
01	00	I <sub>4</sub>	1	→ $\bar{A}B\bar{C}\bar{D}$
01	01	I <sub>5</sub>	0	
01	10	I <sub>6</sub>	0	
01	11	I <sub>7</sub>	0	
10	00	I <sub>8</sub>	0	
10	01	I <sub>9</sub>	1	→ $A\bar{B}\bar{C}\bar{D}$
10	10	I <sub>10</sub>	1	→ $A\bar{B}\bar{C}\bar{D}$
10	11	I <sub>11</sub>	0	
11	00	I <sub>12</sub>	0	→ $\bar{A}\bar{B}\bar{C}\bar{D}$
11	01	I <sub>13</sub>	0	
11	10	I <sub>14</sub>	0	
11	11	I <sub>15</sub>	0	

when D=0,

(b)

$$Z = \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + \cancel{\bar{A}\bar{B}CD}$$

$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$
$\bar{C}\bar{D}$	0	0	0
$\bar{C}D$	1	0	0
$C\bar{D}$	0	0	0
$CD$	0	1	0

$$= \overline{AB}CD + A\overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D}$$

28-03-2022

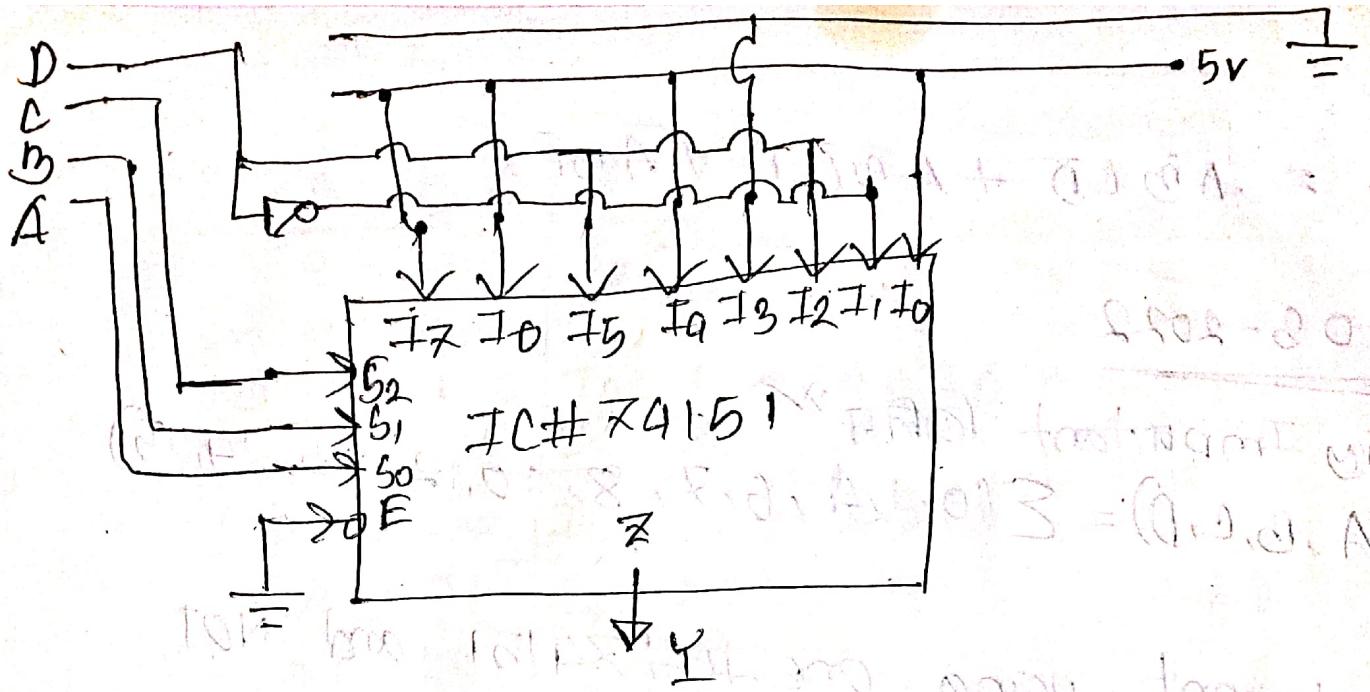
# very important.  $\text{for } \text{Expt. } 1$  (1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 15)

# FCA, B, C, D) = \{0, 1, 4, 6, 7, 8, 10, 12, 15, 14, 15\}

Implement using one IC#74151 and NOT.

gates, if necessary.

				portion		common	
D	C	B	A	Y			
$D=0$	0	(0 0 0)	1	$I_0 \checkmark$			
	0	0 0 1	1	$I_1$			
	0	0 1 0	0	$I_2$			
	0	0 1 1 1	0	$I_3$			
	0	1 0 0	1	$I_4$			
	0	1 0 1	0	$I_5$			
	0	1 1 0	1	$I_6$			
	0	1 1 1	1	$I_7$			
$D=1$	1	(0 0 0)	1	$I_0$			
	1	0 0 1	0	$I_1 \rightarrow I_1$			
	1	0 1 0	1	$I_2 \rightarrow I_2$			
	1	0 1 1 0	1	$I_3 \rightarrow I_3$			
	1	1 0 0	1	$I_4 \rightarrow I_4$			
	1	1 0 1	1	$I_5 \rightarrow I_5$			
	1	1 1 0	1	$I_6 \rightarrow I_6$			
	1	1 1 1	1	$I_7 \rightarrow I_7$			



$b=0$

	0	1	2	3	4	5	6	$I_2$
$I_0$	1	0	1	1	0	1	0	1
$I_1$	0	1	0	1	0	1	0	1

	0	1	2	3	4	5	6	$I_2$
$I_0$	1	0	1	0	1	0	1	1
$I_1$	0	1	0	1	0	1	0	1

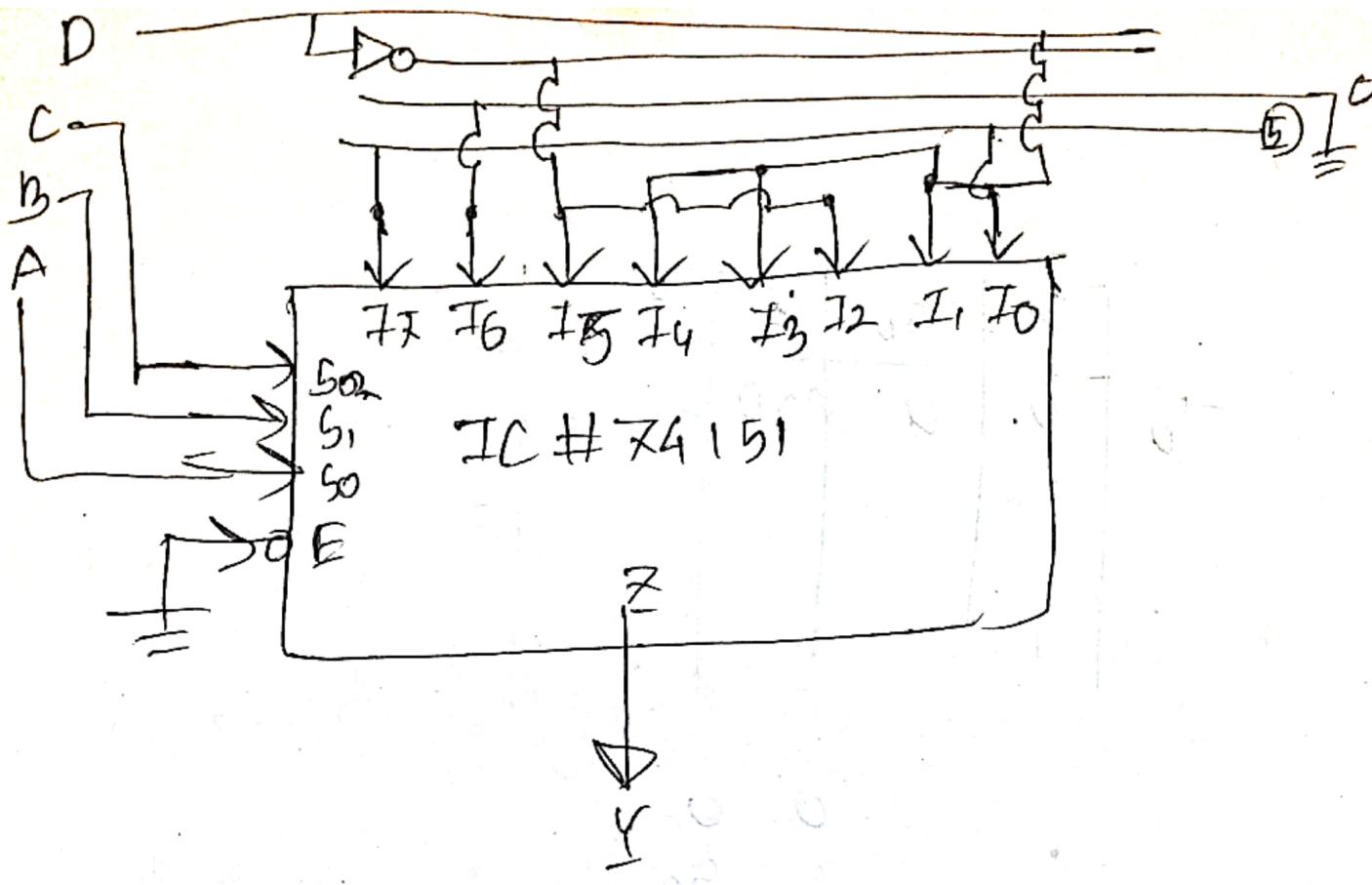
$$f(A, B, C, D) = \sum (0, 2, 5, 7, 8, 9, 11, 12, 15)$$

$$1 = \overline{I_0} \overline{I_1} I_2 I_0$$

$$0 = I_6$$

$$D = \overline{I_0}, I_1, I_3, I_4$$

$$\bar{D} = \overline{I_2}, I_5$$



→ SAP4 → next class

↳ simple AS possible computer

↳ uux — IC#74153

12:50 PM | 30-03-2022

## DESIGN AND IMPLEMENTATION OF GAP-1

GAP → Simple As possible Computer

→ 2 operation (Logical and Arithmetic)

→ Addition and Subtraction (Arithmetic)

• GAP-1 ◉ Logical possible at

→ Programming (Assembly Language)

Machine Language

MATHEMATICS

(GAP-1) - [Slide Page-3]

# program counter

→ currently 0000000000000000

→ AT ZER Line Gump

size 4 so maximum 16 lines can be made.

→ Only counts OP.

## #Accumulator

LID താഴെ പറയുന്ന value മാറ്റുമ്പോൾ, സാക്ഷാത്  
എടാക്കുന്ന അടിസ്ഥാന ഫലം അടിസ്ഥാനം. Address കുറ  
ചിലേ മാറ്റാക്കുന്നതും Register എന്നാണ്.  
Subtractors കുല ഫലം വിശ്വാസിക്കുന്നതും അംഗീകാരം.

→ ID കുറതി ഉപയോഗിച്ച് operation കുമ്പാടു നൽകപ്പെട്ട  
result "Accumulator" കുമ്പാടു നൽകപ്പെട്ട  
ചില ഫലം കുറച്ചു കുറച്ചു കുറച്ചു കുറച്ചു  
ചില ഫലം കുറച്ചു കുറച്ചു കുറച്ചു

• Add/Sub Unit is 8-bit numbers.

Accumulator - "A"

Registers - "B"

Adder/Subtractor

↳ "A+B" OR "A-B"

MAO → Memory Address Register

LDA → Load Accumulator  $\leftrightarrow$  RAM [MAR]

ADD → Addition → Add ram data to accumulator

$\text{SUB} \rightarrow \text{Subtraction} \rightarrow \text{Subtract RAM data} \quad \text{Acc} \leftarrow \text{Acc} - B$

~~OUT-ID OUT-ACC~~

HLT → CLR A → 0      HLT → Halt

↑ Capital letter G മലയാളം ഒരു

LDA 8H → Hexadecimal

↳ means 8 नम्बर location को 8 मी valve तक

उत्तरी लोड वार्षा,

LDA fit  $\rightarrow$  ~~first~~ load the accumulation with

contents of memory location F1.

~~10010001~~ 10010000000000000000000000000000 Hexa decimal 0

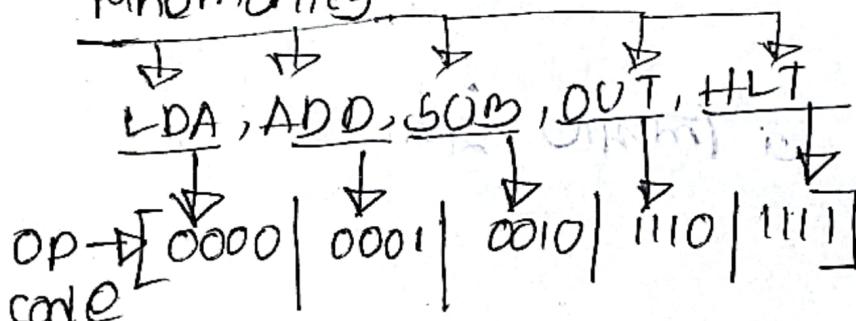
LDA stands for Load the Accumulator.

LDA, ADD, SUBTRACT called ~~DATA~~ instructions

memory reference instructions

because they use data stored in the memory.

Mnemonics



Op code stands for operation code.

Program in Assembly

Address

0FF to FFF

1. ACC ← RAM[0FH]

2. ACC ← 10H

3. ACC ← 0H + 18H

4. ACC ← 10H + 18H + 20H

5. ACC ← 10H + 18H + 20H - 14H

6. OUT ← ACC

0H + 18H + 20H - 14H

Maximum 7 operation

Why?  $① + ①$   
OUT, HLT

$$16 - 2 = 14 \text{ op}$$

Add/Sub  
total possible operations  $\approx 2^7$  total operation

total op -  $\frac{14}{2} = 7$  operations possible.

blank memory operation 7th of code

After first 27, [ ]

03-04-22

SAP1 → Only Addition and Subtraction

1. LDA → Load to the Accumulator

LDA 8H | ACCA → RAM[8H] | OPP code  
0000

2. ADD → Addition

ADD AH | ACCA → ACC + RAM[AH] | OPP code  
0001

3. SUB → Subtraction

SUB 9H | ACCA → ACC - RAM[9H] | OPP code  
0010

4. OUT → Output

OUTA → ACC | 1110 0010  
OPP code

5. HLT → Halt | Stop the program

\* SAP1 T27CO 25 Marks Q1 Question 21/20/21

and machine lang.

#Write the assembly language program that performs the following operations:  $10H + 18H \rightarrow 20H$   
 $-14H$

### Assembly language

### Machine Language

Optional

Address	Contents	Address	Content (In Binary)	Contents in Hex
0H	LDA FH	0000	0000 1111	OFH
1H	ADD EH	0001	0001 1110	1EH
2H	ADD DH	0010	0001 1101	1DH
3H	SUB CH	0011	0010 1100	2CH
4H	OUT FA	0100	1110 1111	EFH
5H	HLT FH	0101	1111 FFFF	FFH
6H	FFH	0110	1111 1111	FFH
7H	FFH	0111	1111 1111	FFH
8H	FFH	1000	1111 1111	FFH
9H	FFH	1001	1111 1111	FFH
AH	FFH	1010	1111 1111	FFH
BH	FFH	1011	1111 1111	FFH
CH	14H	1100	0001 0100	14H
DH	20H	1101	0010 0000	20H
EH	18H	1110	0001 1000	18H
FH	20H	1111	0001 0000	20H

→ ID Hex to Binary (or) Addressing per digit as  
BCD form format.

# Create a "SAPI assembly program" and then  
generate the machine code for following expression  
 $52+28-38+72-12$ . These numbers is in  
decimal form.

# Decimal to Hexa → 20A0 21A 21B

D : 52+28-38+72-12

i zero  
↓ 111

H : 34H + 1CH - 26H + 48H 0CH - 0CH

# Assembly Language

# Machine Language

Address	Contents	Address	Content (In Binary)	Content (In Hex)
0 H	LDA FH	0 000	0000 1111	0 FH
1 H	ADD EH	0 001	0001 1110	1 EH
2 H	SUB DH	0 010	0010 1101	2 DH
3 H	ADD CH	0 011	0001 1100	1 CH
4 H	SUB BH	0 100	0010 1011	2 BH
5 H	OUT FH	0 101	1110 1111	E FH
6 H	HLT FH	0 110	1111 1111	FFH
7 H	FFH	0 111	1111 1111	FFH
8 H	FFH	1 000	1111 1111	FFH
9 H	FFH	1 001	1111 1111	FFH
A H	FFH	1 010	1111 1111	FFH
B H	FFH	1 011	0000 1100	0 CH
C H	0 CH	1 100	0100 1000	48 H
D H	48 H	1 101	0010 0110	26 H
E H	26 H	1 110	0001 1100	1 CH
F H	1 CH	1 111	0011 0100	34 H
	34 H			

Create a SAP1 assembly language program and then generate the machine code for the expression of  $18 - 20 + 35 + 37 - 8$ . These numbers is in decimal form.

D<sub>0</sub>: 18 - 20 + 35 + 37 - 8

H<sub>0</sub>: 12H - 14H + 23H + 25H - 8H

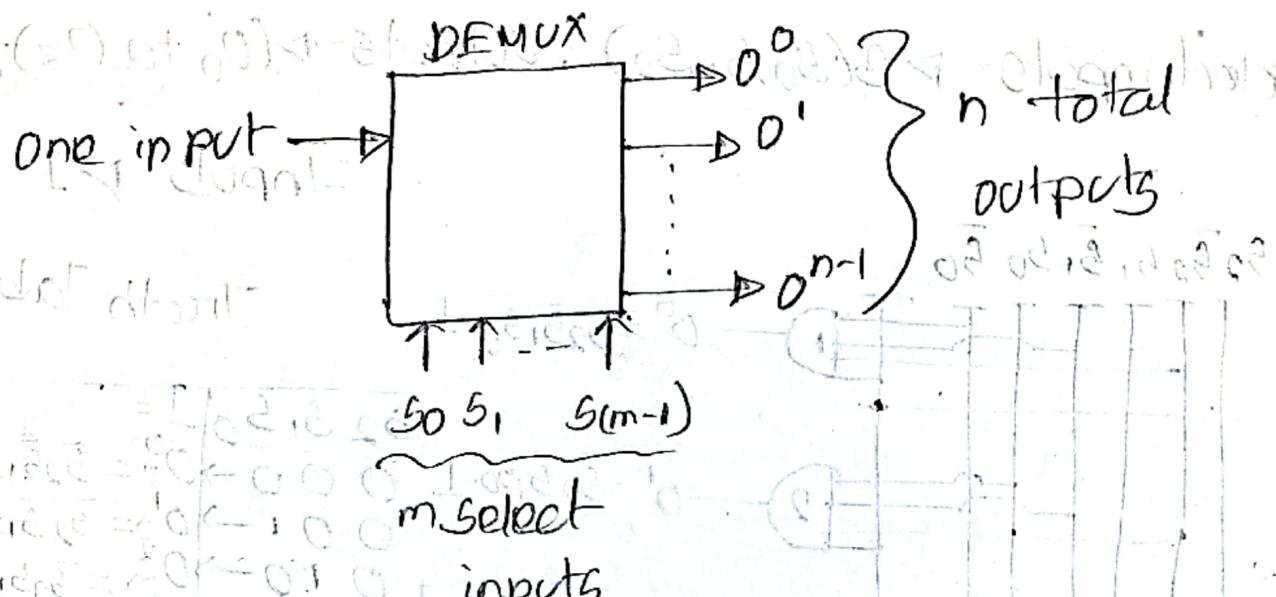
Assembly Language : Machine Language

Address	Contents	Address	Contents (In Binary)	Contents (Int'l)
0H	LDA F1H	0000	0000 1111	0F1H
1H	SUB E1H	0001	0010 1110	2DEH
2H	ADD D1H	0010	0001 1101	1DH
3H	ADD C1H	0011	0001 1100	1CH
4H	SUB B1H	0100	0010 1011	2BH
5H	OUT F1H	0101	1110 1111	EFH
6H	HLT F1H	0110	1111 1111	FFF1H
7H	FF1H	0111	1111 1111	FF1H
8H	FF1H	1000	1111 1111	FF1H
9H	FF1H	1001	1111 1111	FF1H
AH	FF1H	1010	1111 1111	FF1H
BH	08H	1011	0000 1000	08H
CH	25H	1100	0010 0101	25H
DH	23H	1101	0010 0011	23H
EH	14H	1110	0001 0100	14H
FH	12H	1111	0001 0010	12H

06-03-2022

### ③ Demuxi pleser (DEMUX)

what is DEMUX? (Ans 3 ch 0, 1, 2)



After a time input goes to the one of the outputs according to corresponding combination of select inputs.

No of select inputs	No of outputs
2	4
3	8
4	16
5	32

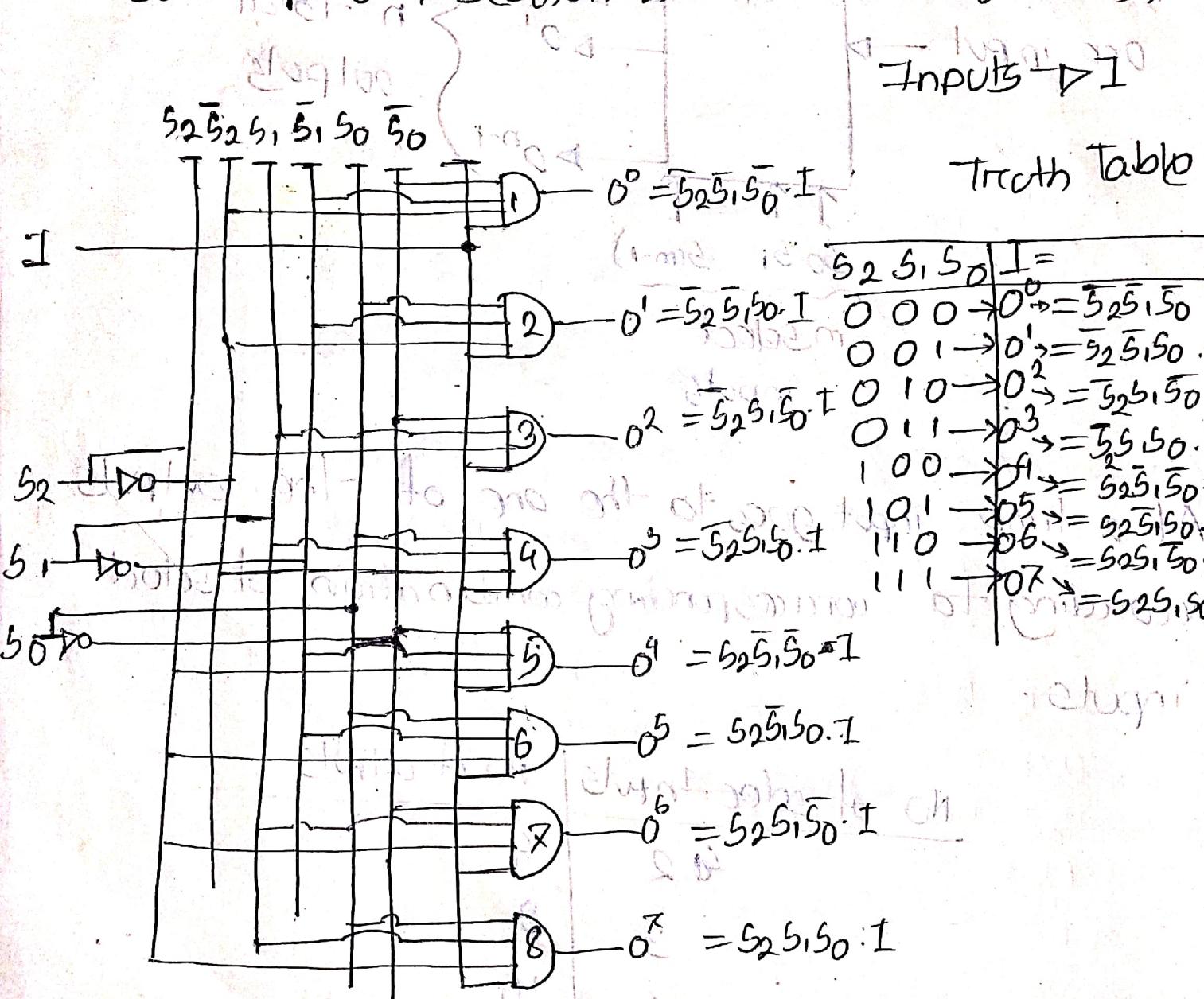
$\therefore \text{No of outputs} = 2^{(\text{No of select inputs})}$

$$\therefore n = 2^m$$

# 1 OF 8 DEMUX-ID DEMUX 8

(1 line to 8 line DEMUX)

Select inputs  $\rightarrow S_2(S_0, S_1, S_2)$ ; Outputs  $\rightarrow O_0 \text{ to } O_7$ :



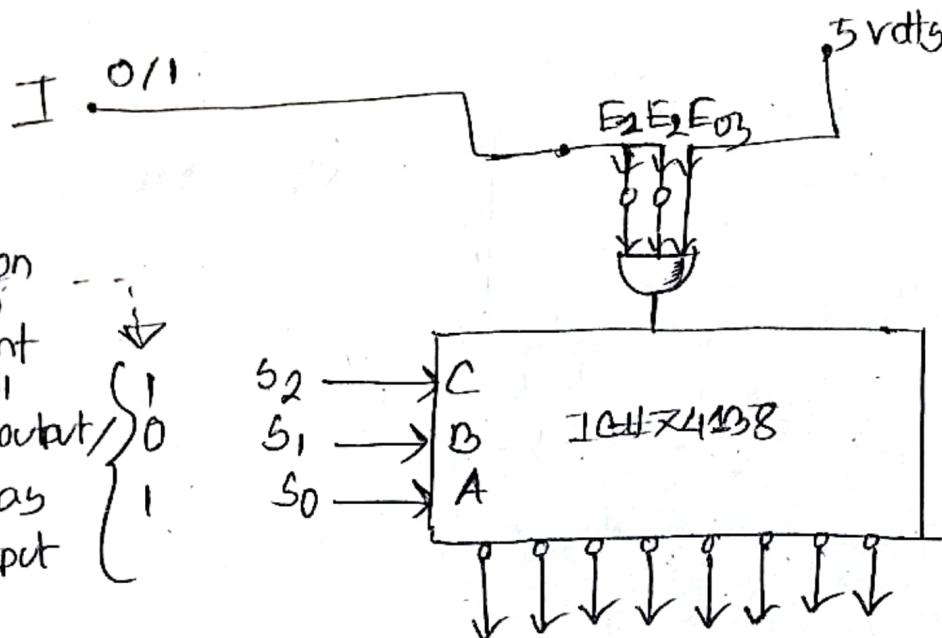
74138 IC Decoder / Demux (1 of 8)

সংস্কৃতি = সংগৃহী হো ও

# Design 1 of 8 (1 line to 8 lines) DEMUX using

IC 74138.

when  
we  
want  
101  
as output / 0  
0 has  
output



Active low  
Output means

0 → Active  
1 → Inactive

0 → Active  
1 → Inactive  
2 → Inactive

IC Active 0

Active 0:  $E_1 = E_2 = 0$ ;  
 $E_0 = 1$ .

# When  $I=0, O_5=0$

, ,  $I=1, O_5=1$

when  $I=0$  then IC is Active

when  $I=1$  then IC is Inactive.

# END of Chapter 9

MSI logic circuits #