CS & IT ENGINEERING

DIGITAL LOGIC

MUX

Lecture No. 04

By- CHANDAN SIR

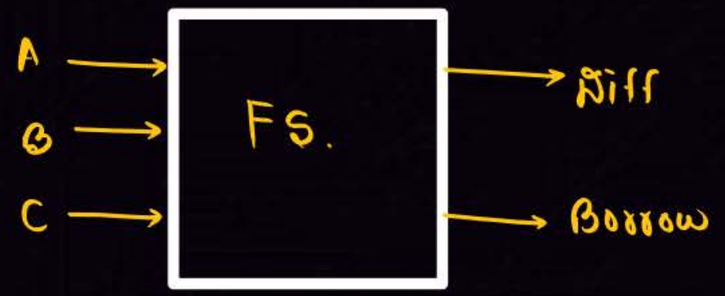
01 MULTIPLEXER

02 QUESTION PRACTICE

03 DISCUSSION

FULL SUBTRACTOR

Step 1 :-



Step 2.

	A	B	C	Diff	Borrow
0→	0	0	0	0	0
1→	0	0	1	1	1
2→	0	1	0	1	1
3→	0	1	1	0	1
4→	1	0	0	1	0
5→	1	0	1	0	0
6→	1	1	0	0	0
7→	1	1	1	1	1

Step 3. $\text{Diff} = \Sigma m(1,2,4,7) = A \oplus B \oplus C$

$$\text{Borrow} = \Sigma m(1,2,3,7)$$

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

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

$$= (B \oplus C)\bar{A} + BC$$

$$= \overline{(B \odot C)} \cdot \bar{A} + BC$$

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

$(\bar{A}B + AB)C + \bar{A}B(\bar{C} + C)$

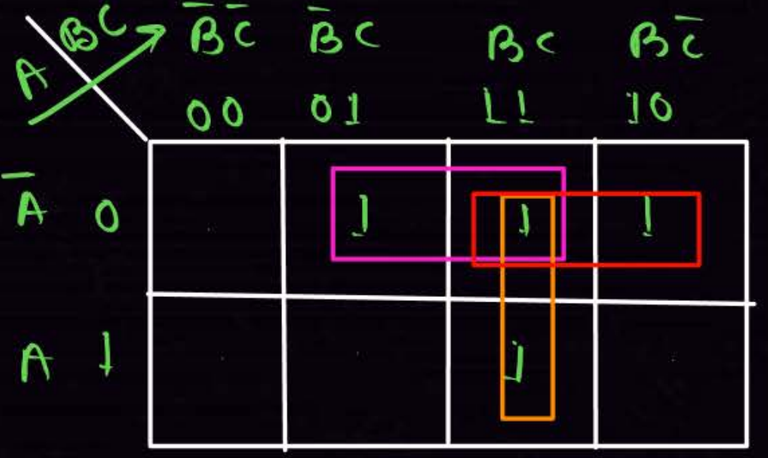
$(A \odot B)C + \bar{A}B$

$\overline{(A \oplus B)}C + \bar{A}B$ ← semiminimized

Step4: Minimization.

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

$$Borrow = \sum m(1,2,3,7)$$



$$Borrow = \bar{A}B + \bar{A}C + BC$$

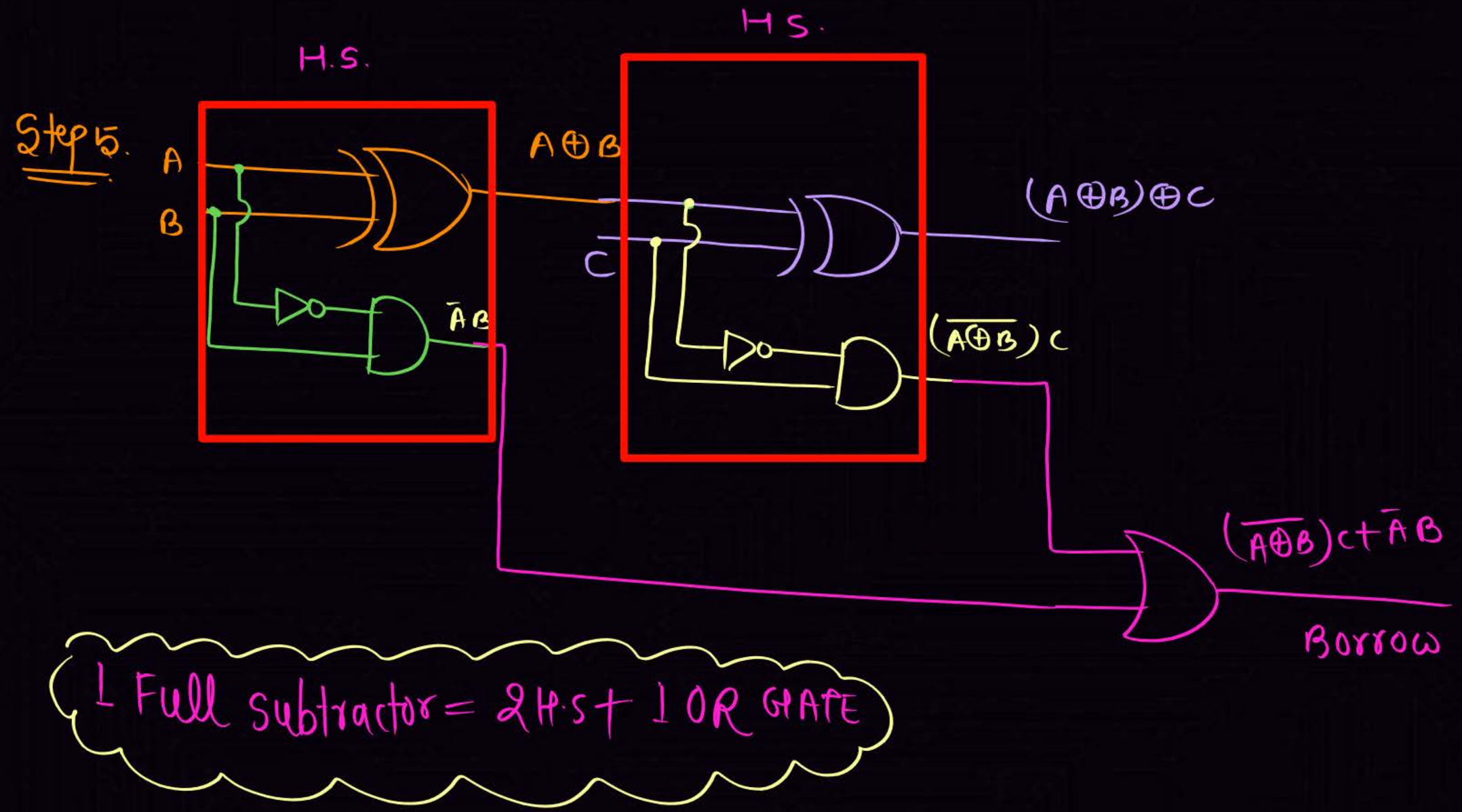
$$\text{Diff} = (A \oplus B) \oplus C$$

$$\text{Borrow} = \Sigma m(1,2,3,7)$$

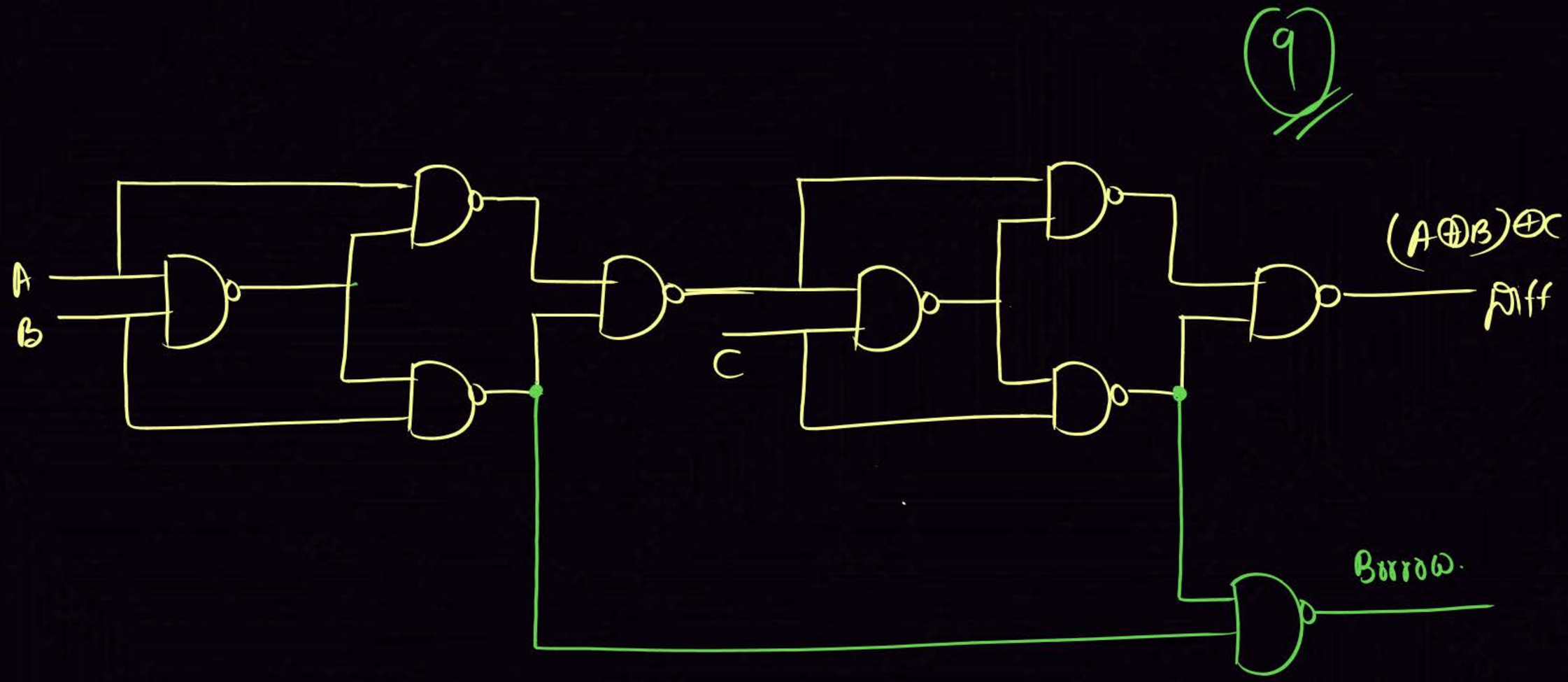
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$

$$= \overline{(A \oplus B)} \cdot C + \bar{A}B$$

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



1 Full Subtractor = 2 H.S + 1 OR GATE


9
A
B
C
(A⊕B)⊕C
Diff
Borrow.

H.A

$$\left\{\begin{array}{l} \text{Sum} = A \oplus B \\ \text{Carry} = AB \end{array}\right\}$$

NAND, NOR } = 5.

F.A

$$\text{Sum} = A \oplus B \oplus C$$

$$\text{Carry} = \Sigma m(3,5,6,7)$$

$$= (A \oplus B)C + AB$$

$$= AB + AC + BC$$

NAND, NOR } = 9

H.S.

$$\text{Diff} = A \oplus B$$

$$\text{Borrow} = \bar{A}B$$

NAND, NOR } = 5

F.S.

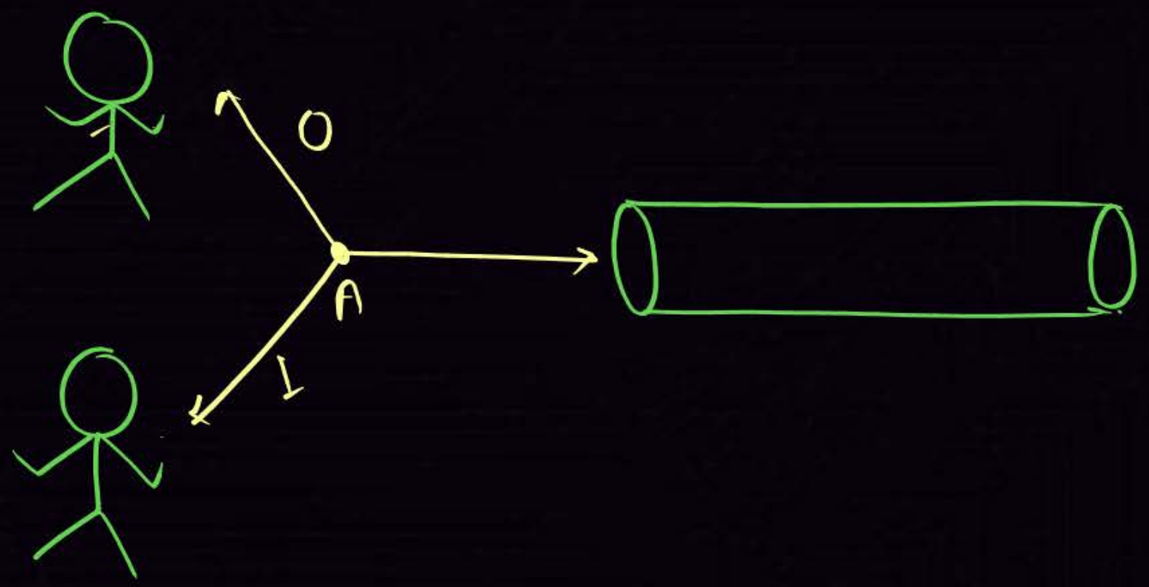
$$\text{Diff} = A \oplus B \oplus C$$

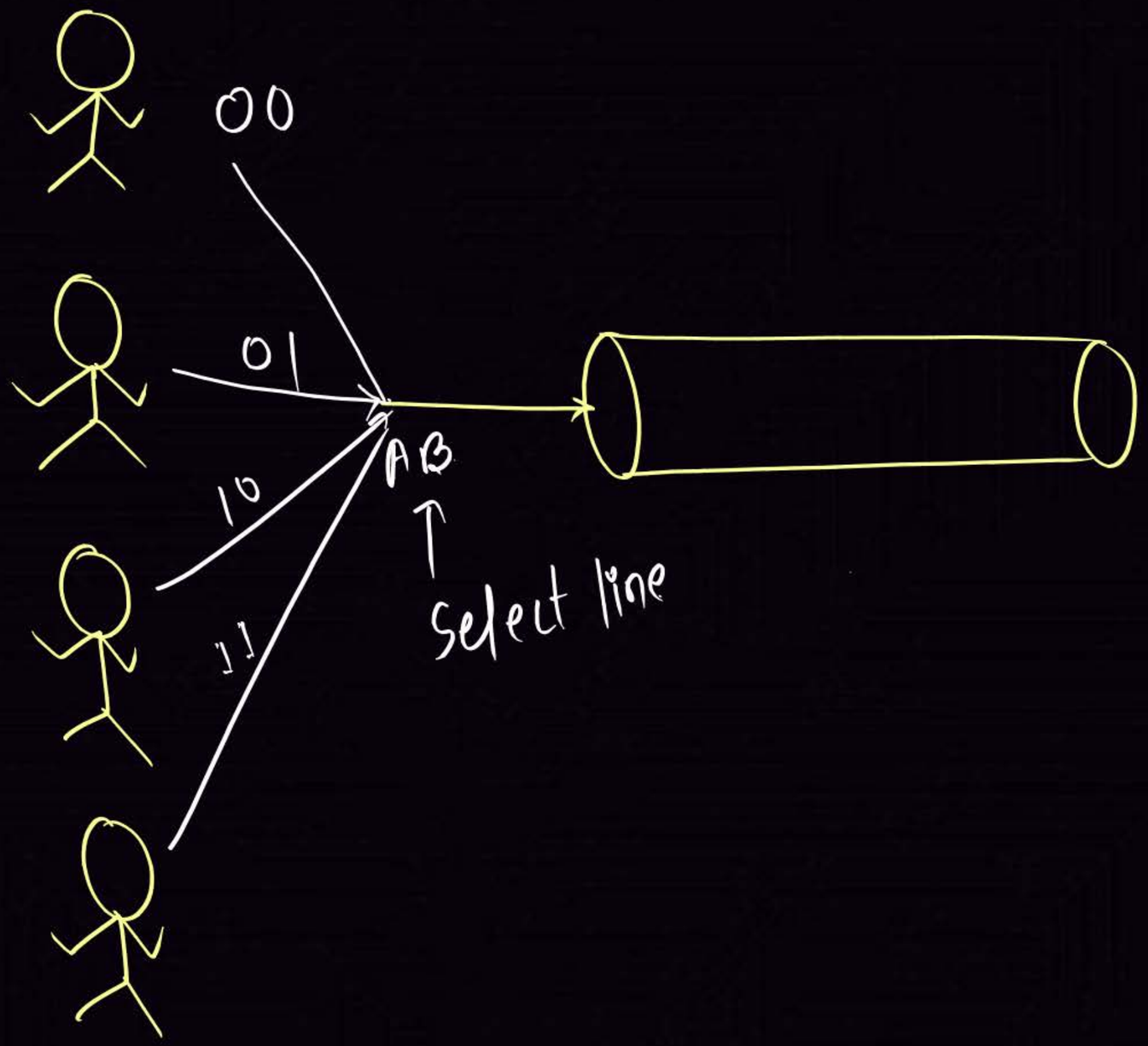
$$\text{Borrow} = \Sigma m(1,2,3,7)$$

$$= (\overline{A \oplus B})C + \bar{A}B$$

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

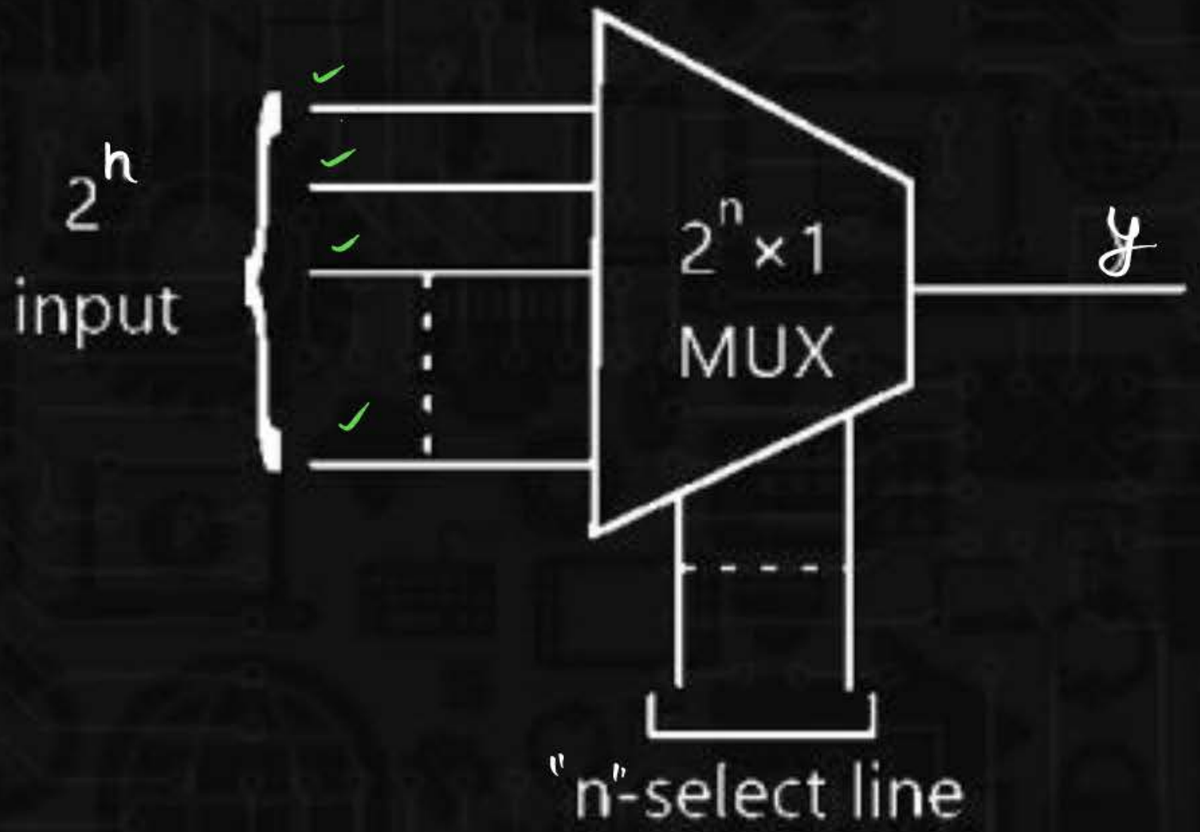
NAND, NOR } = 9


O
A
I


00
01
10
11
AB
Select line

MULTIPLEXER

1 Question



MUX

↳ input o/p

$2^n : 1$

↳ AND-OR Logic

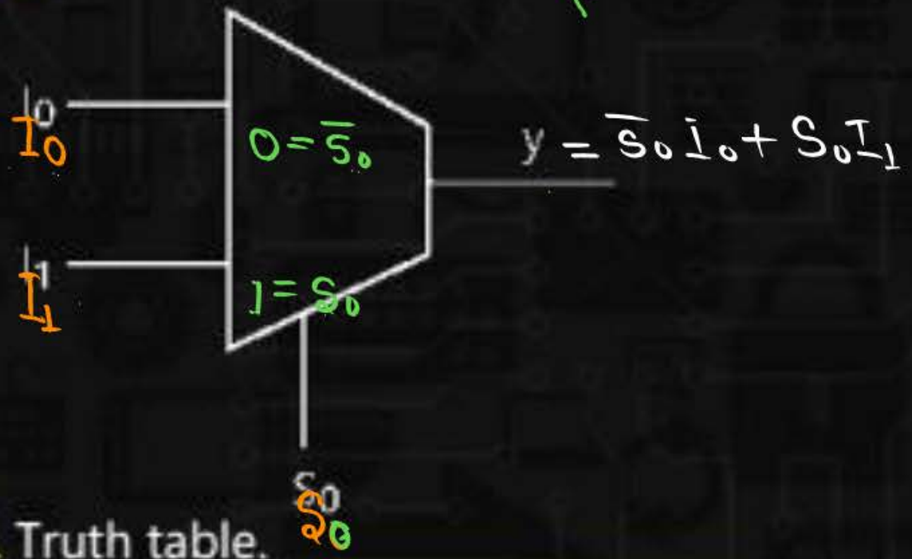
↳ Universal Logic

MULTIPLEXER

Q.1 Design a 2 × 1 MUX?

i/p o/p

Step-1. Number of input 2 outputs



$$y = \bar{S}_0 I_0 + S_0 I_1$$

Step 2. Truth table.

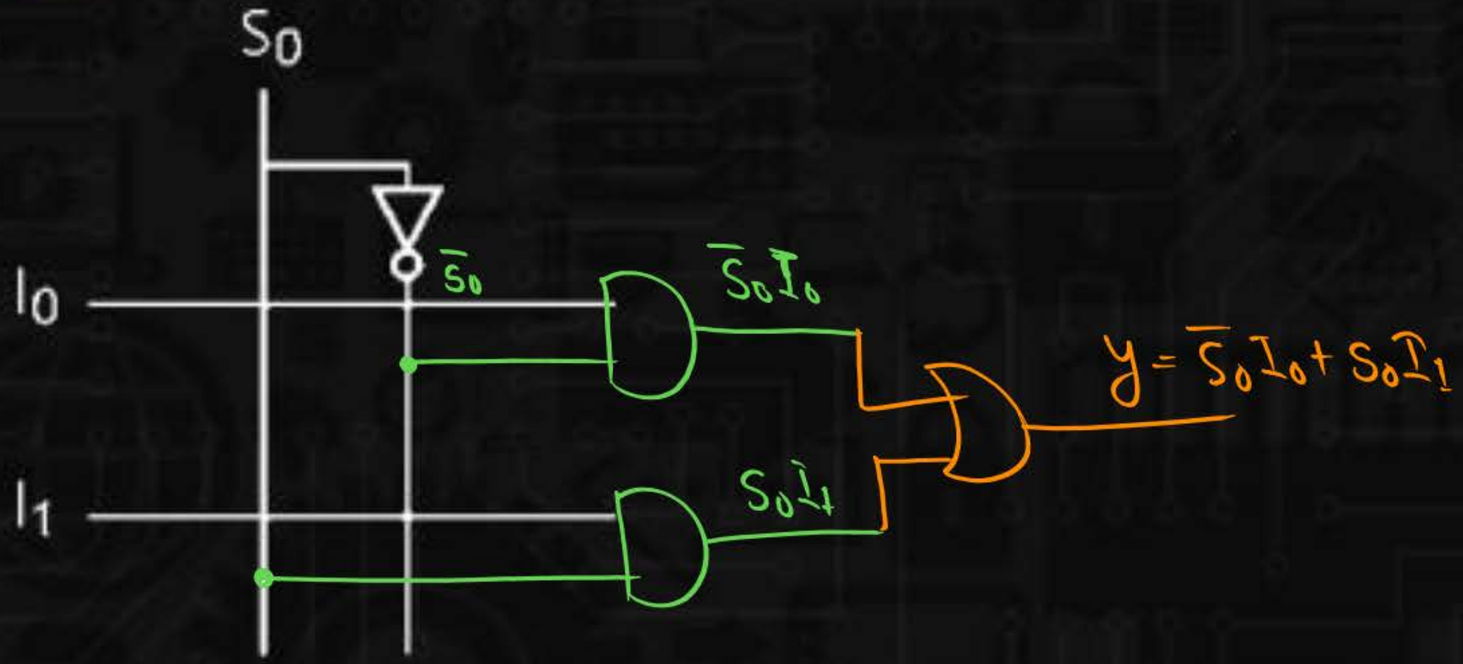
S_0	Y
0	I_0
1	I_1

MULTIPLEXER

Step 3. Logical expression $y = \bar{S}_0 I_0 + S_0 I_1$

Step 4. Minimization ✓

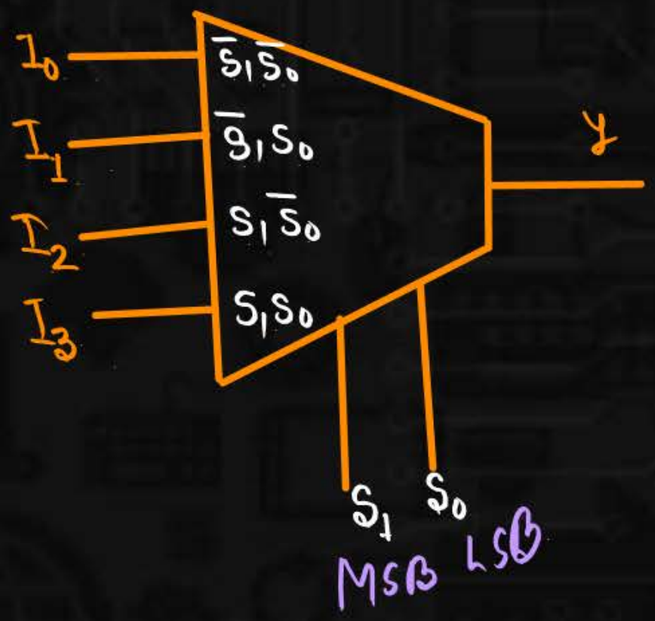
Step 5. Hardware implementation



MULTIPLEXER

Q.2 Design a 4 × 1 MUX?

Step 1 :-



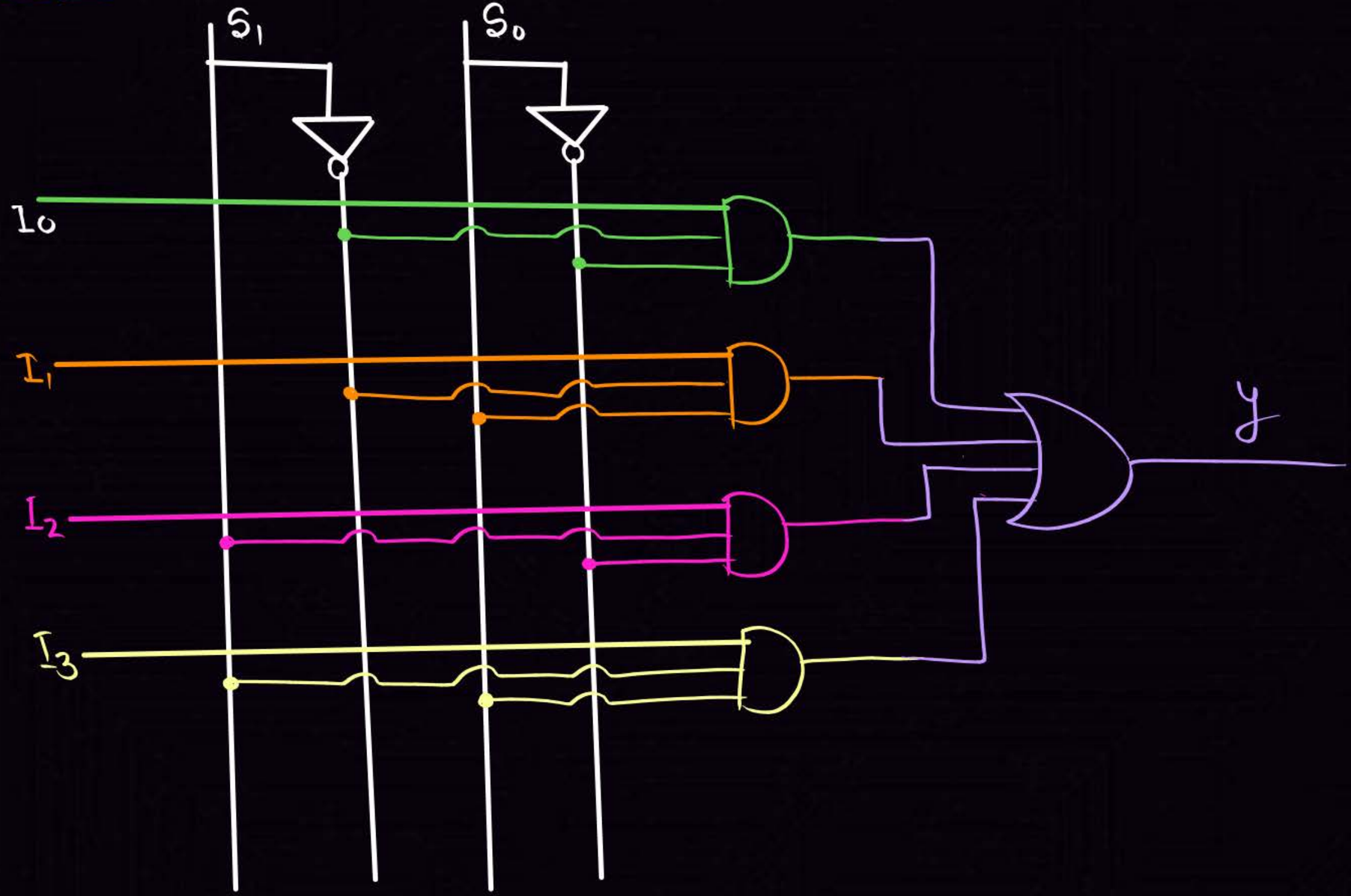
Step 2 :-

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Step 3. $Y = \bar{S}_1\bar{S}_0I_0 + \bar{S}_1S_0I_1 + S_1\bar{S}_0I_2 + S_1S_0I_3$

Step 4 :→ Minimization.

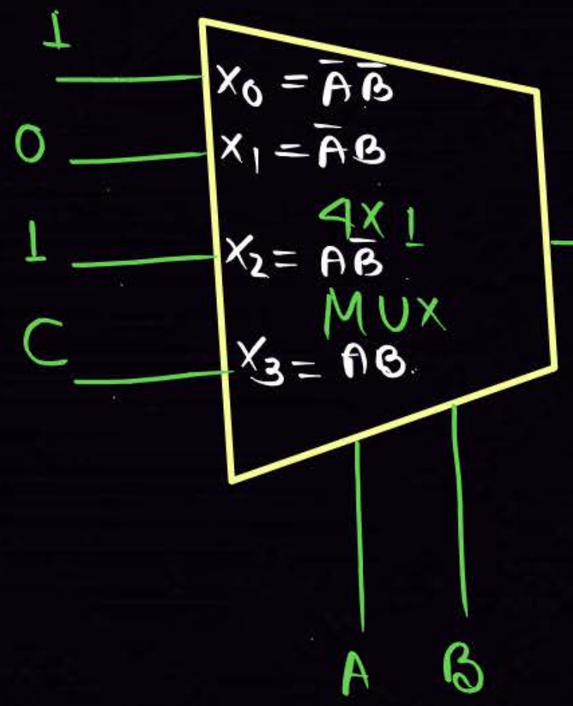
Step 5. $y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$



Q Design a 8x1 MUX ?

Q Design a 16x1 MUX ?

Ex



$y = \bar{A}\bar{B}\cdot 1 + \bar{A}B\cdot 0 + A\bar{B}\cdot 1 + AB\cdot C$

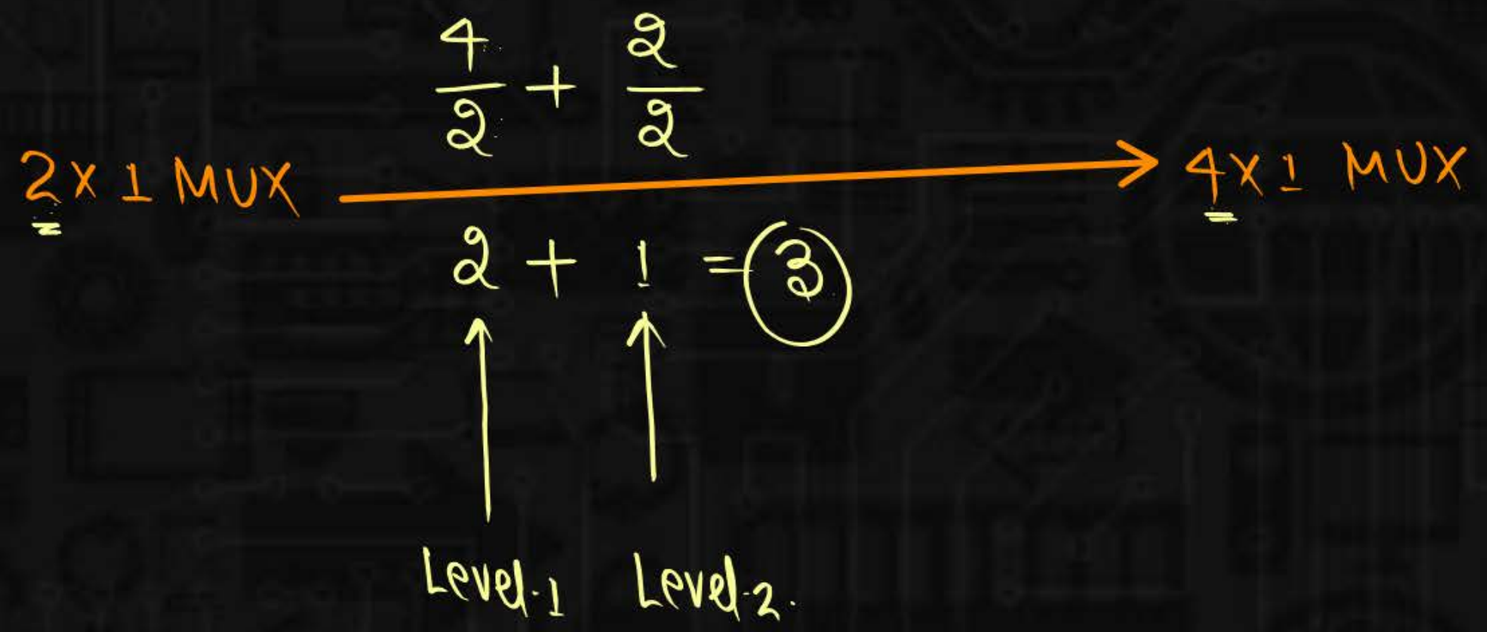
$= \bar{A}\bar{B} + A\bar{B} + ABC$

TYPE 1 . Desgining of Higher Order Mux By Lower Order Mux

Q.3 Design 4×1 MUX using 2×1 MUX.

2 select line

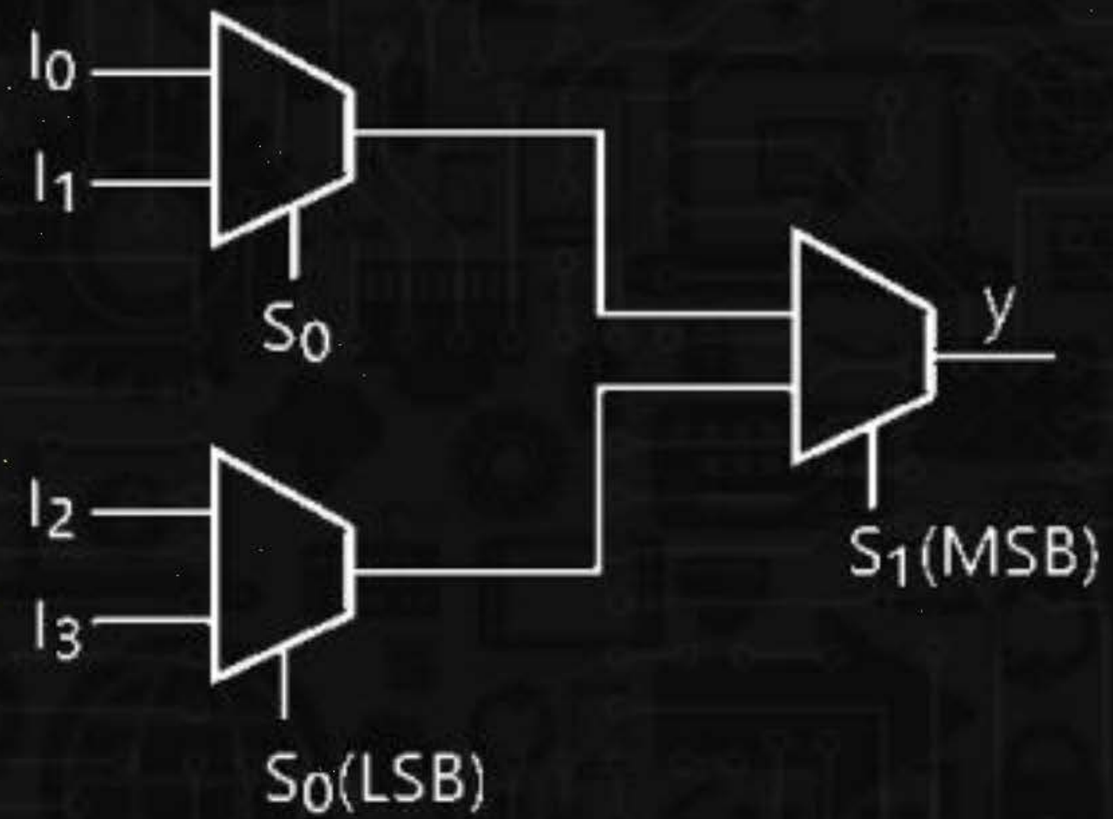
1 select line



MULTIPLEXER

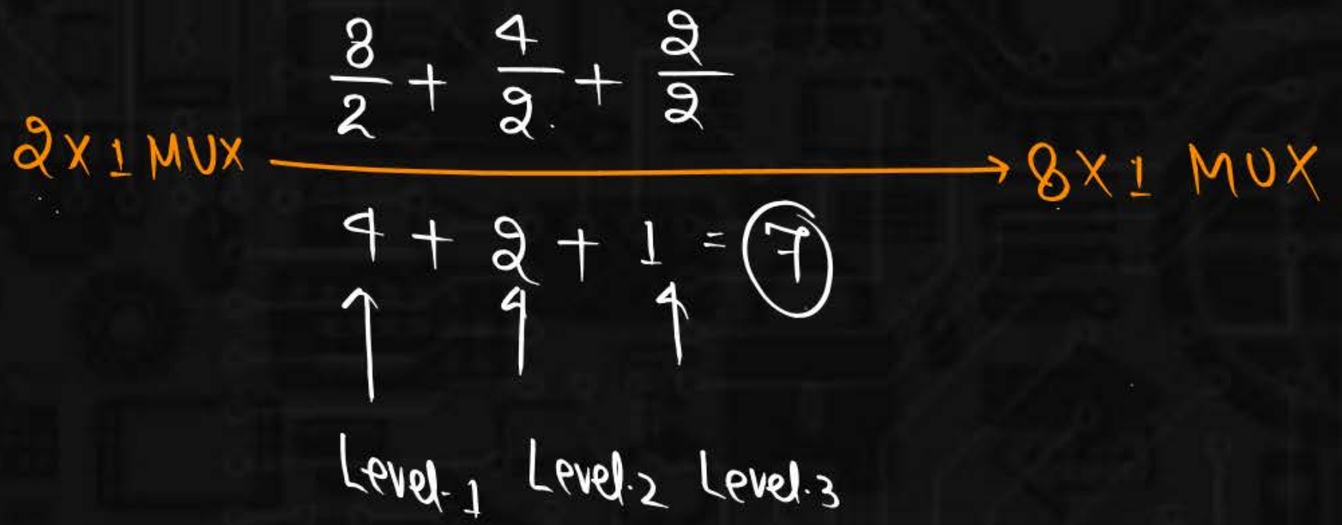
$2\times1 \xrightarrow{\frac{4}{2}+\frac{2}{2}} 4\times1$

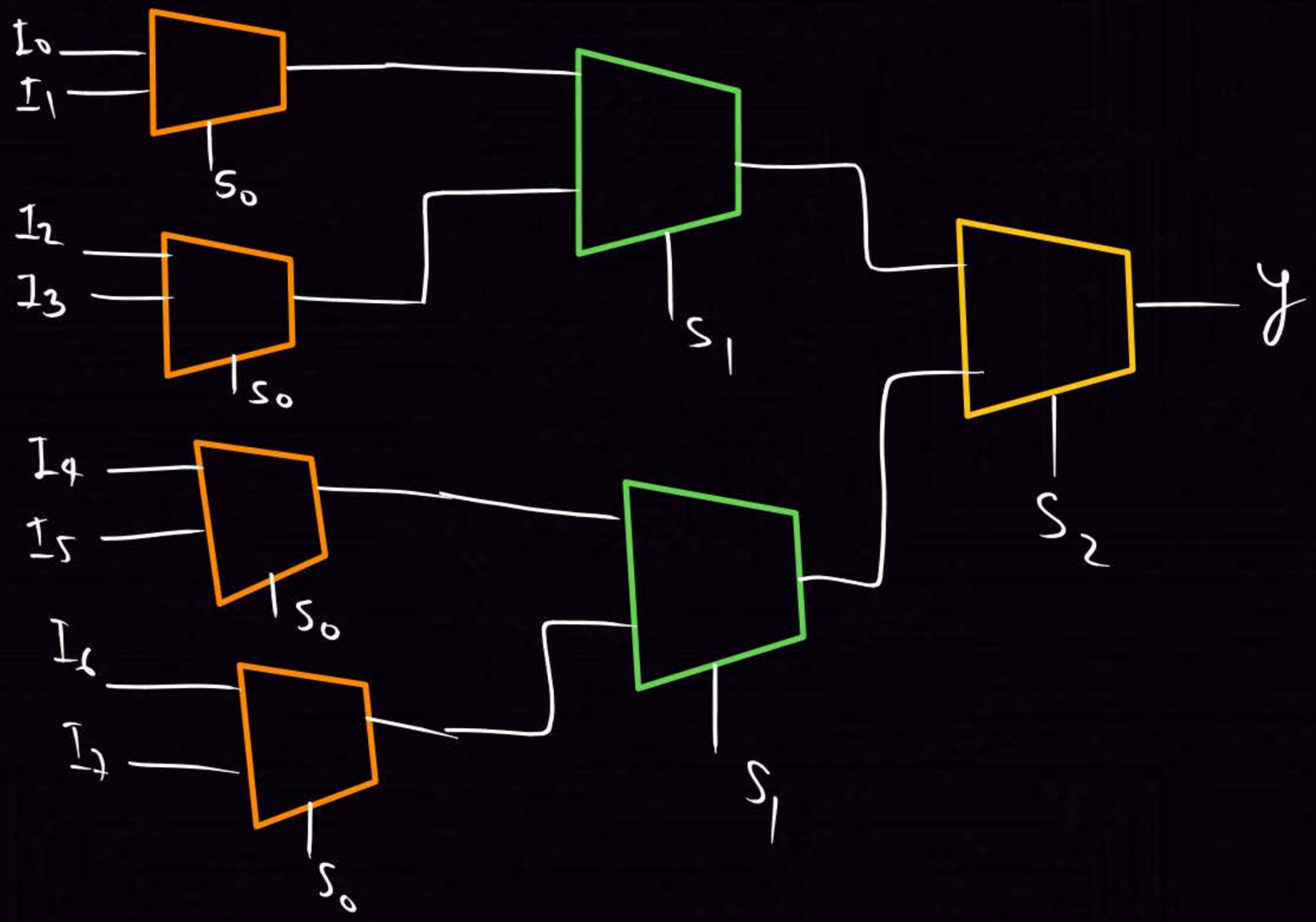
$2+1=3$



MULTIPLEXER

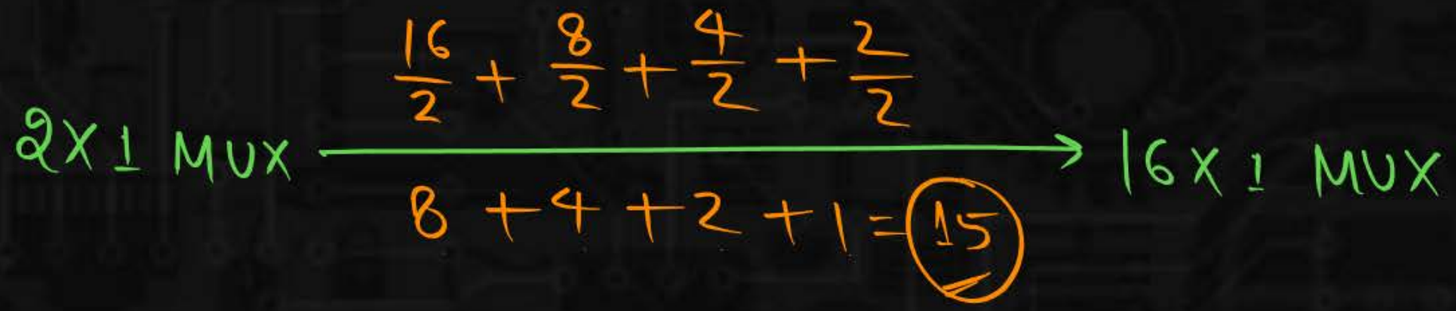
Q.4 Design 8 × 1 MUX using 2 × 1 MUX.




I0
I1
S0
I2
I3
S0
S1
I4
I5
S0
I6
I7
S0
S1
S2
y

MULTIPLEXER

Q.5 Design 16 × 1 using 2 × 1 MUX.



MULTIPLEXER

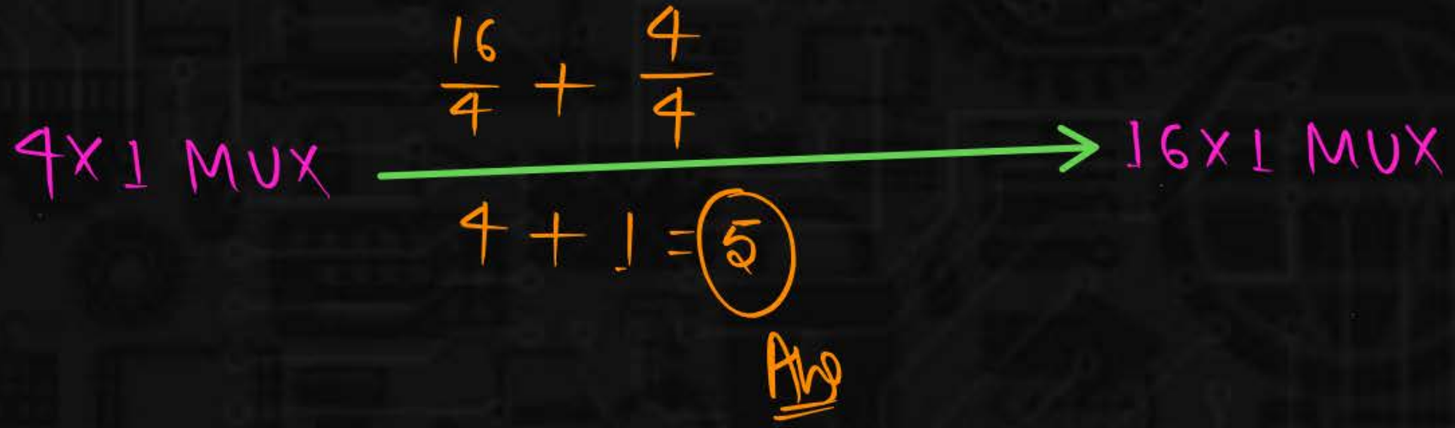
Q.6 Design 64 × 1 MUX using 2 × 1 MUX.

MULTIPLEXER

Q.7 Design $2^n \times 1$ MUX using 2×1 MUX.

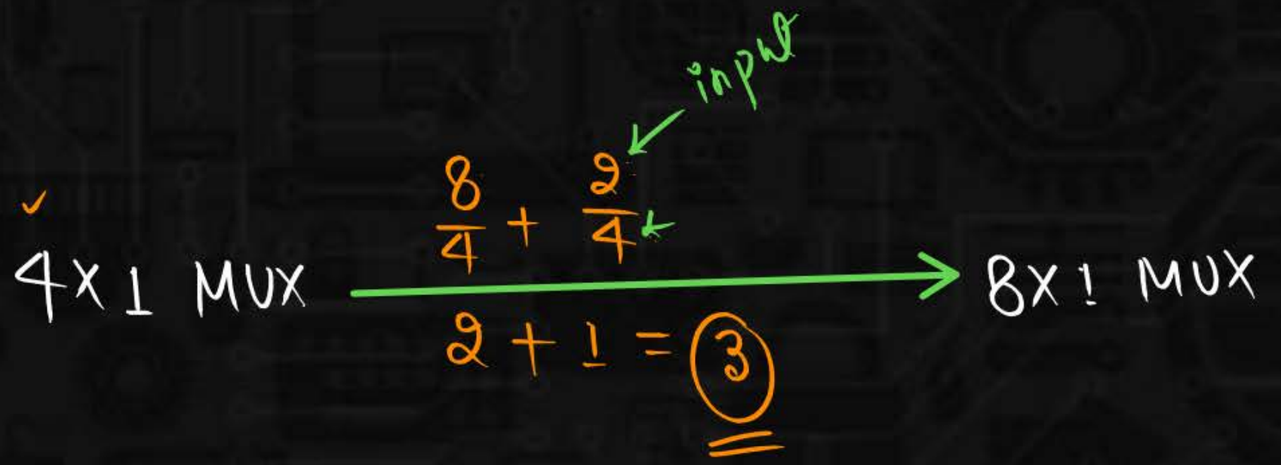
MULTIPLEXER

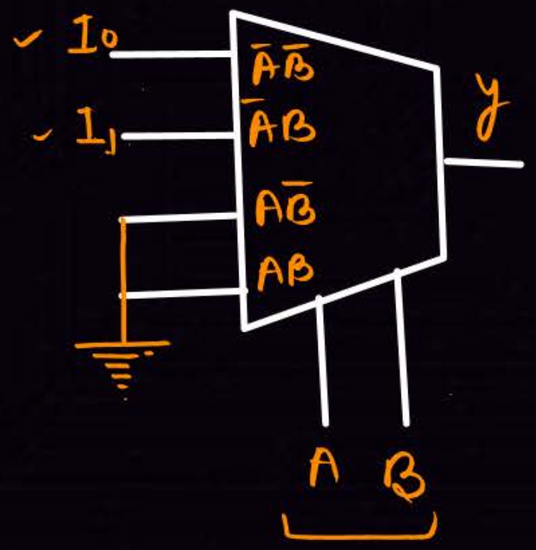
Q.8 Design 16 × 1 MUX using 4 × 1 MUX.



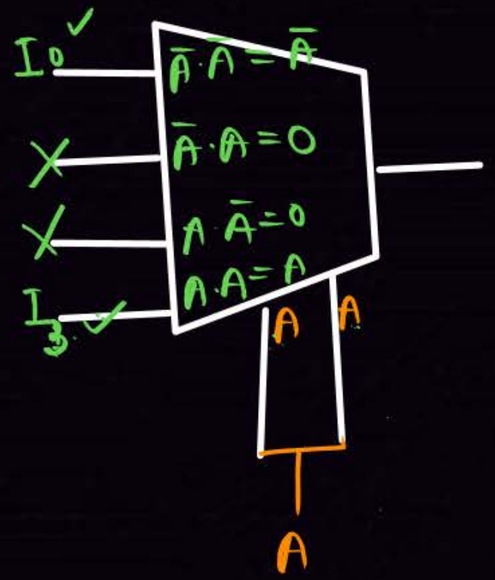
MULTIPLEXER

Q.9 Design 8×1 using 4×1 MUX.

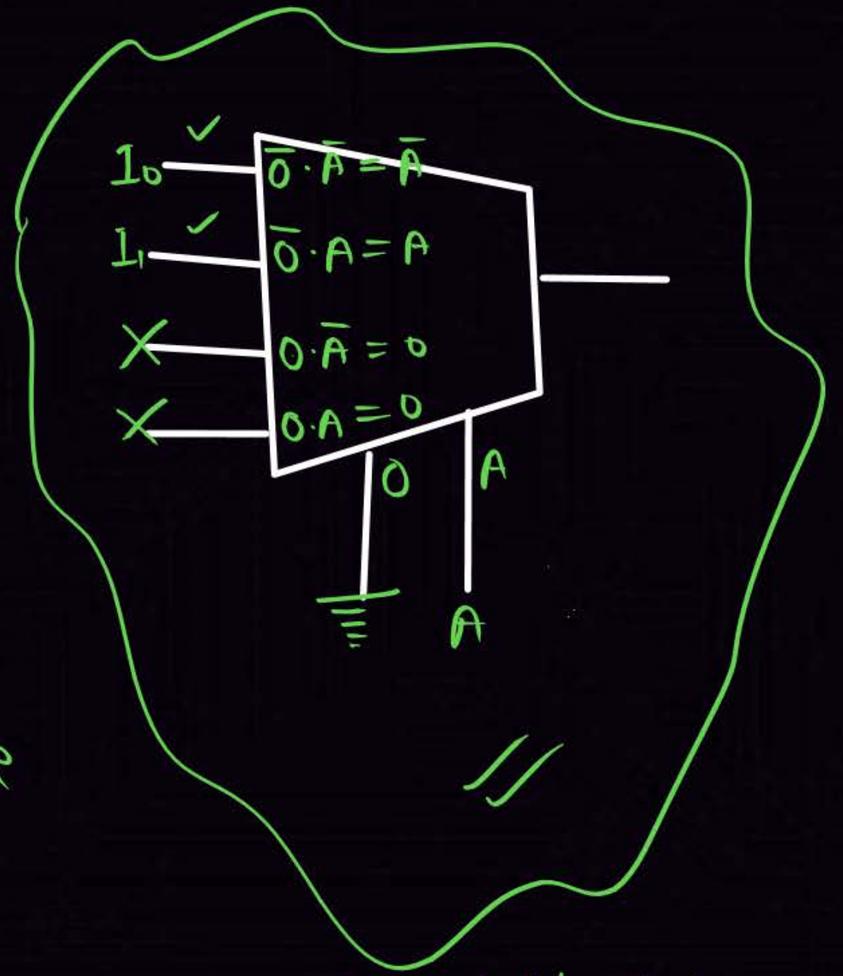




Select Variable = 2.



Select Variable = 1

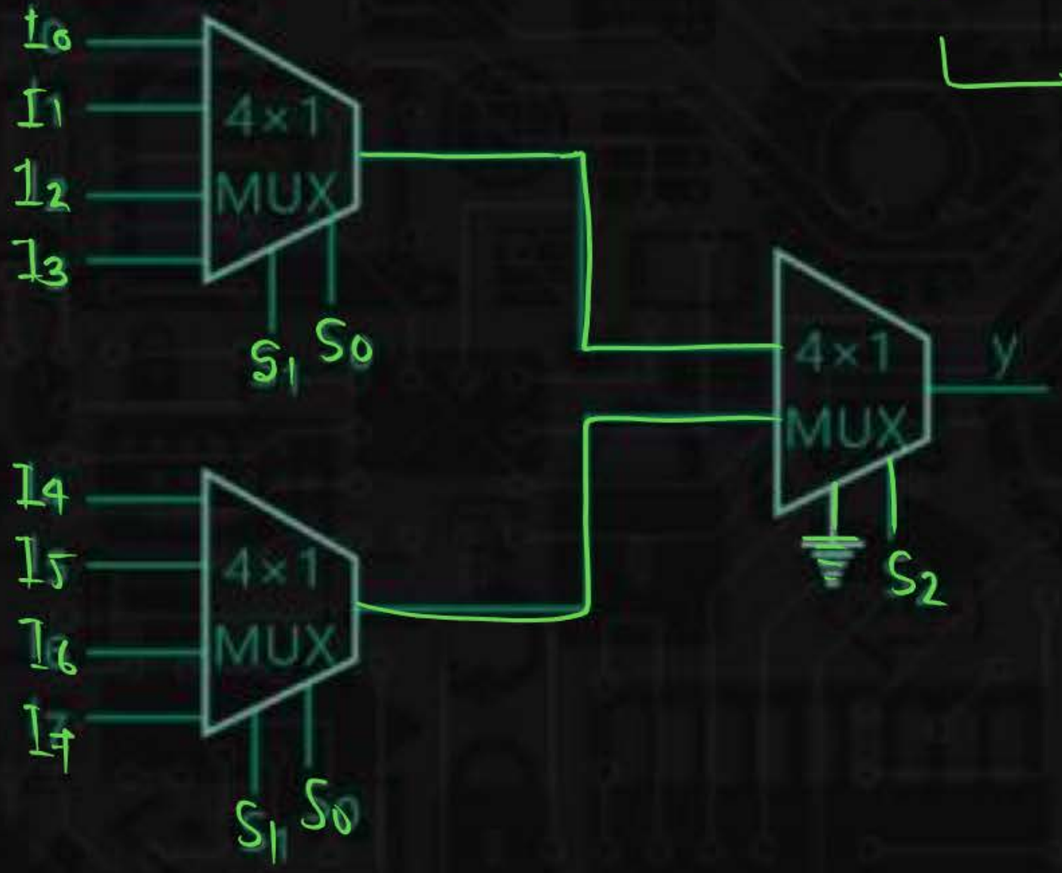


Select Variable = 1

MULTIPLEXER

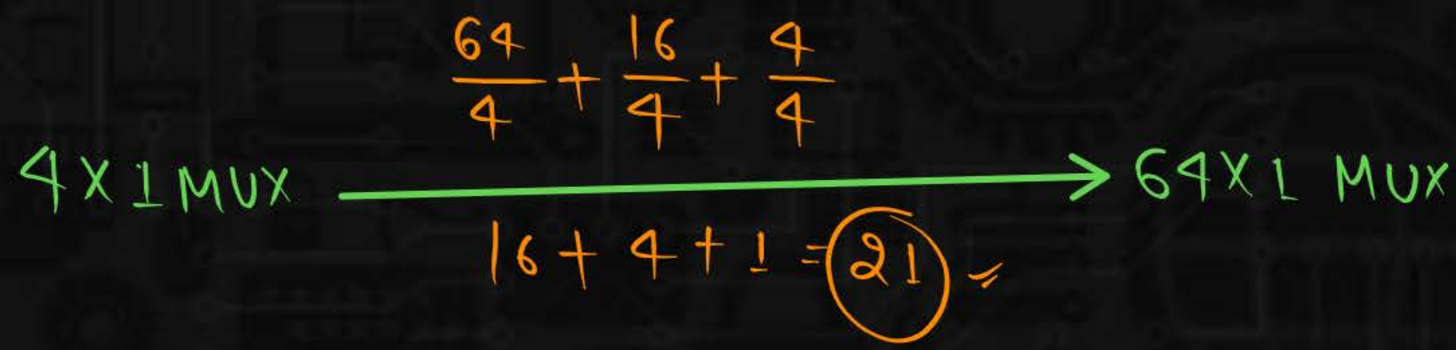
8X1 MUX

↳ 3 select Variable



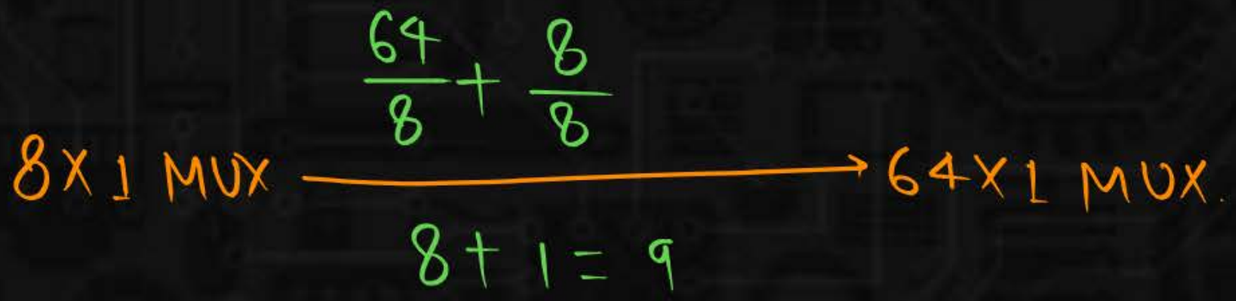
MULTIPLEXER

Q.10 Design 64 × 1 MUX using 4 × 1 MUX.



MULTIPLEXER

Q.11 Design 64 × 1 MUX using 8 × 1 MUX.



MULTIPLEXER

Q.12 Design 256 × 1 MUX using 16 × 1 MUX.

16X1 MUX $\xrightarrow{\frac{256}{16} + \frac{16}{16}}$ 256X1 MUX

16+1 = (17)

Ans

Q = 4x1 MUX $\xrightarrow{\frac{32}{4}+\frac{8}{4}+\frac{2}{4}}$ $\xrightarrow{8+2+1=}$ (!!) Ans → 32x1 MUX

PYQ
- Logic GATE
- Boolean Algebra
- K-MAP
- HA, FA, H.S, F.S,


THANK YOU GW
SOLDIERS !