SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with 'A' Grade by NAAC

QUALITY SYSTEM CERTIFICATION

DNV·GL

ISO 9001:2008

# Lecture session 5_ UNIT-3

# Unit-3-COMBINATIONAL LOGIC MULTIPLEXER AND DEMULTIPLEXER

**By**

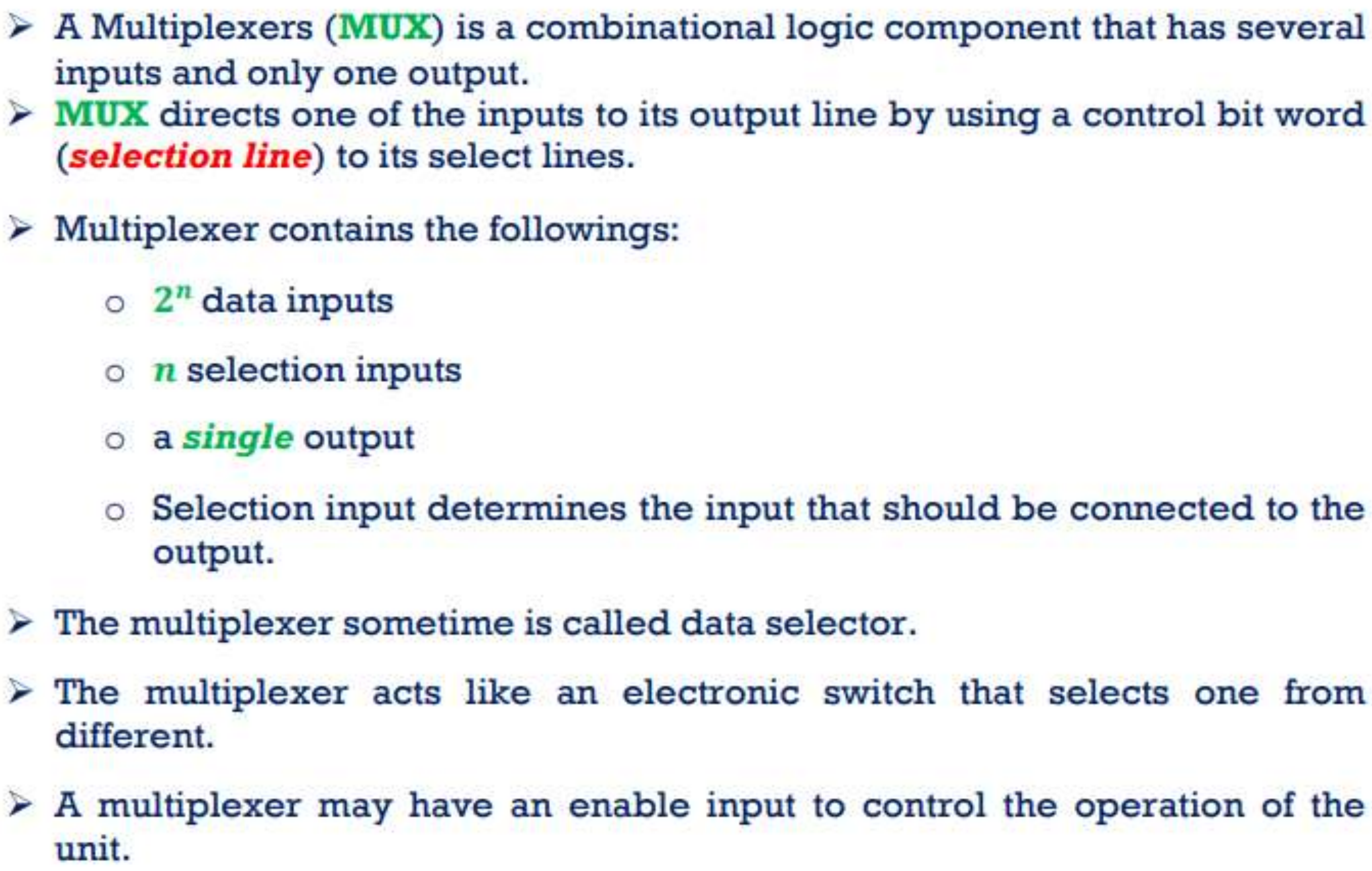
**V.GEETHA**

**ASSISTANT PROFESSOR/EEE**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

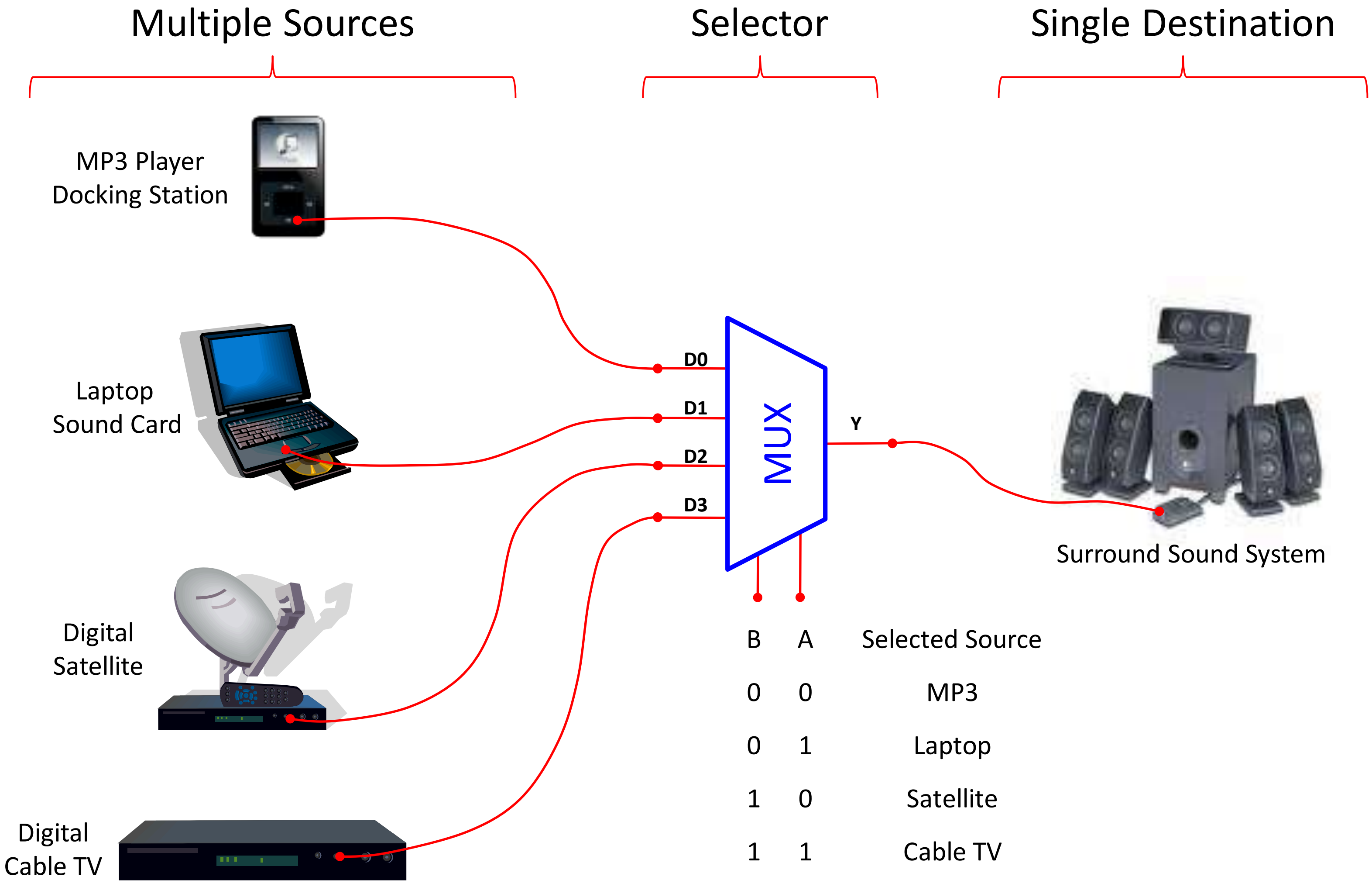
**CHENNAI-119**

# Multiplexer



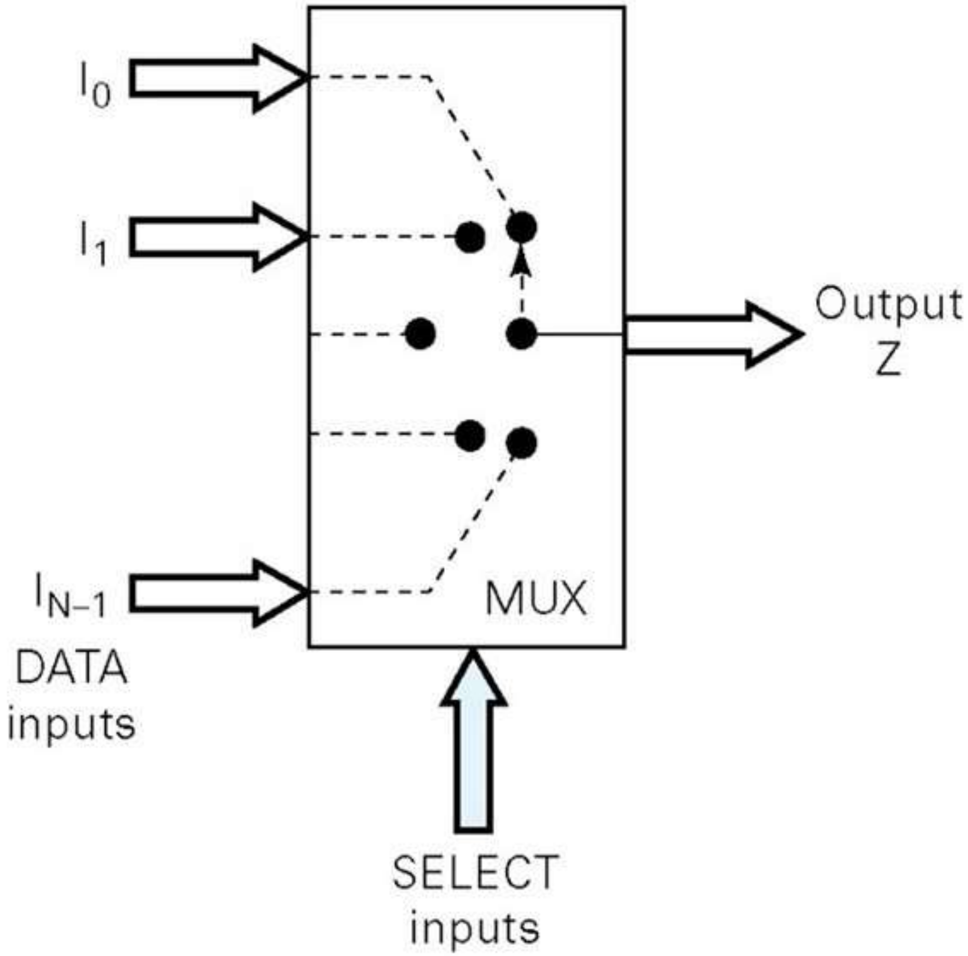
- A Multiplexers (**MUX**) is a combinational logic component that has several inputs and only one output.
- **MUX** directs one of the inputs to its output line by using a control bit word (***selection line***) to its select lines.
- Multiplexer contains the followings:
    - $2^n$ data inputs
    - $n$ selection inputs
    - a ***single*** output
    - Selection input determines the input that should be connected to the output.
- The multiplexer sometime is called data selector.
- The multiplexer acts like an electronic switch that selects one from different.
- A multiplexer may have an enable input to control the operation of the unit.

# Typical Application of a MUX



| B | A | Selected Source |
|---|---|---|
| 0 | 0 | MP3 |
| 0 | 1 | Laptop |
| 1 | 0 | Satellite |
| 1 | 1 | Cable TV |

# Functional Diagram Of a Multiplexer



# 2 : 1 Multiplexer

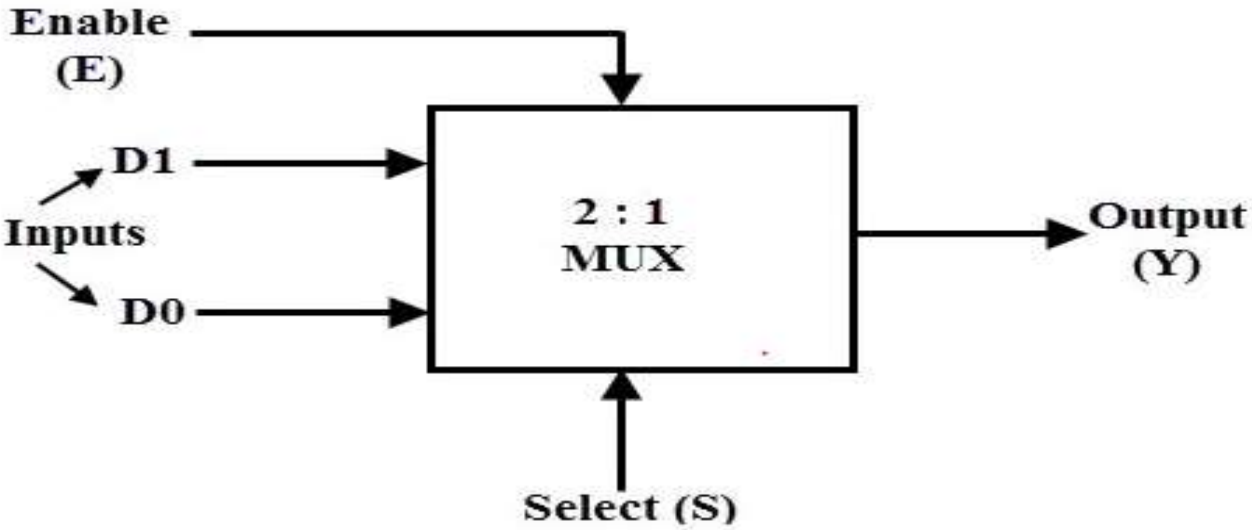
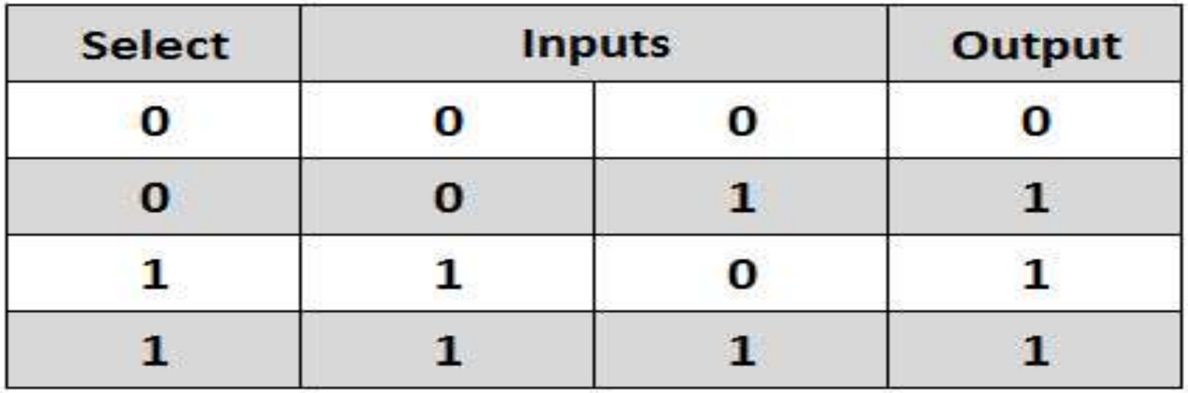
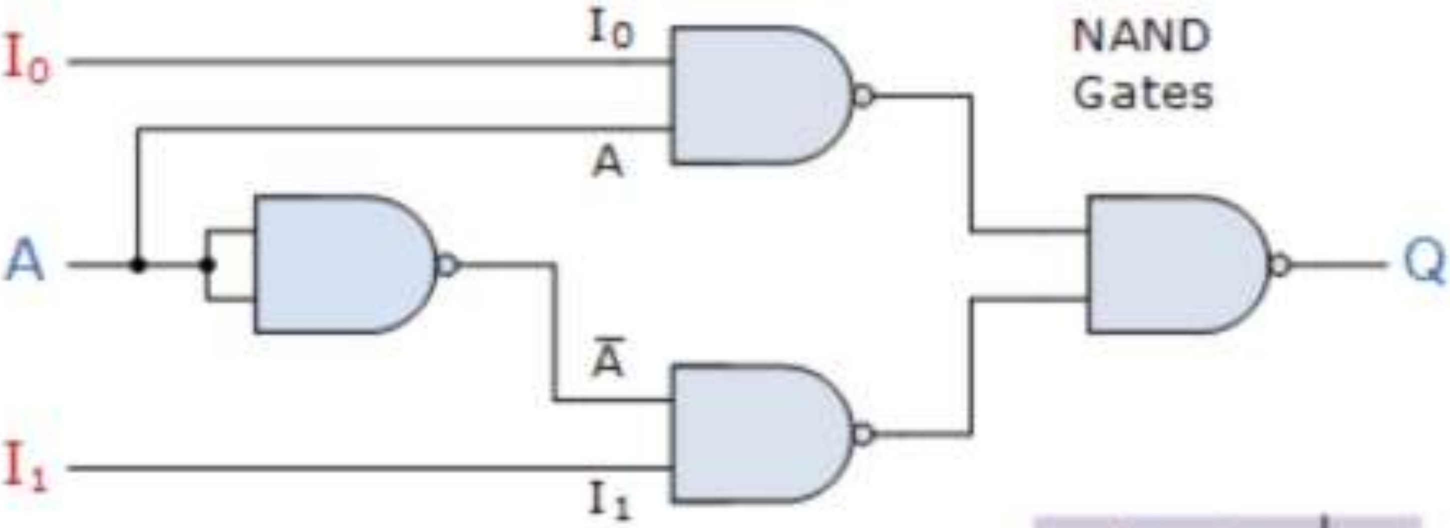
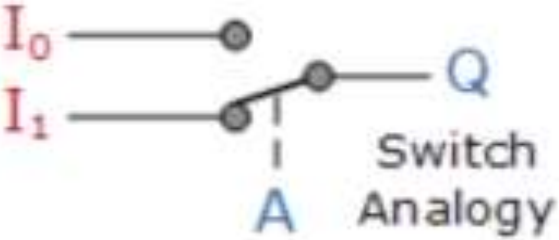


Fig: Block Diagram of 2-to-1-MUX



| Select | Inputs | | Output |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |





Switch Analogy

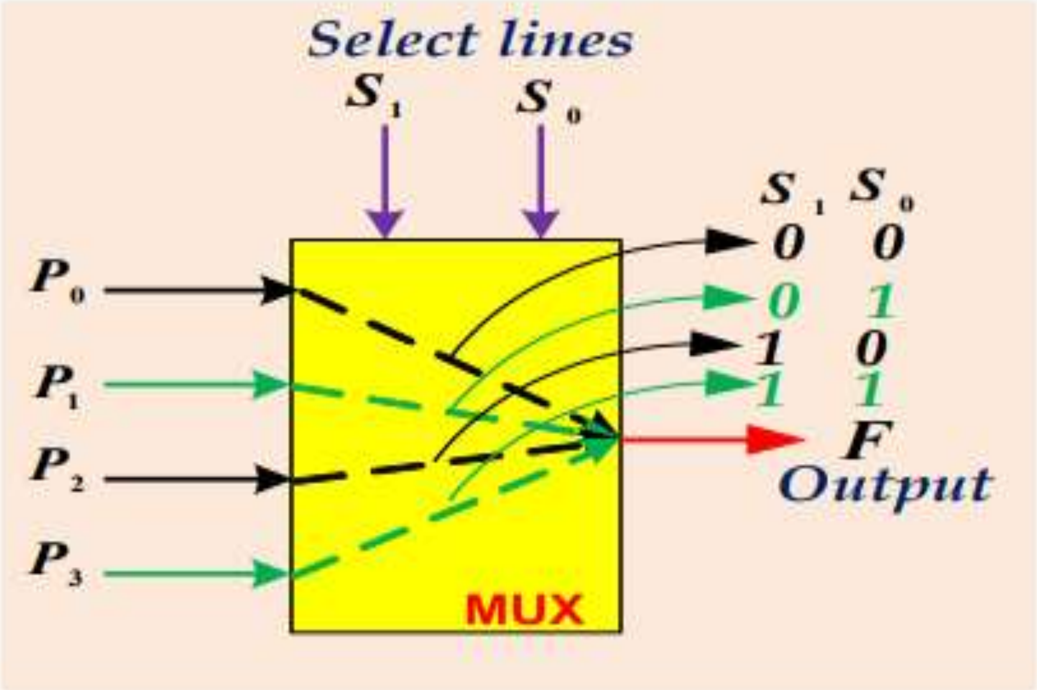
Truth Table

| Inputs | | | |
|---|---|---|---|
| A | $I_1$ | $I_0$ | Q |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## a) 4-to-1 Multiplexers

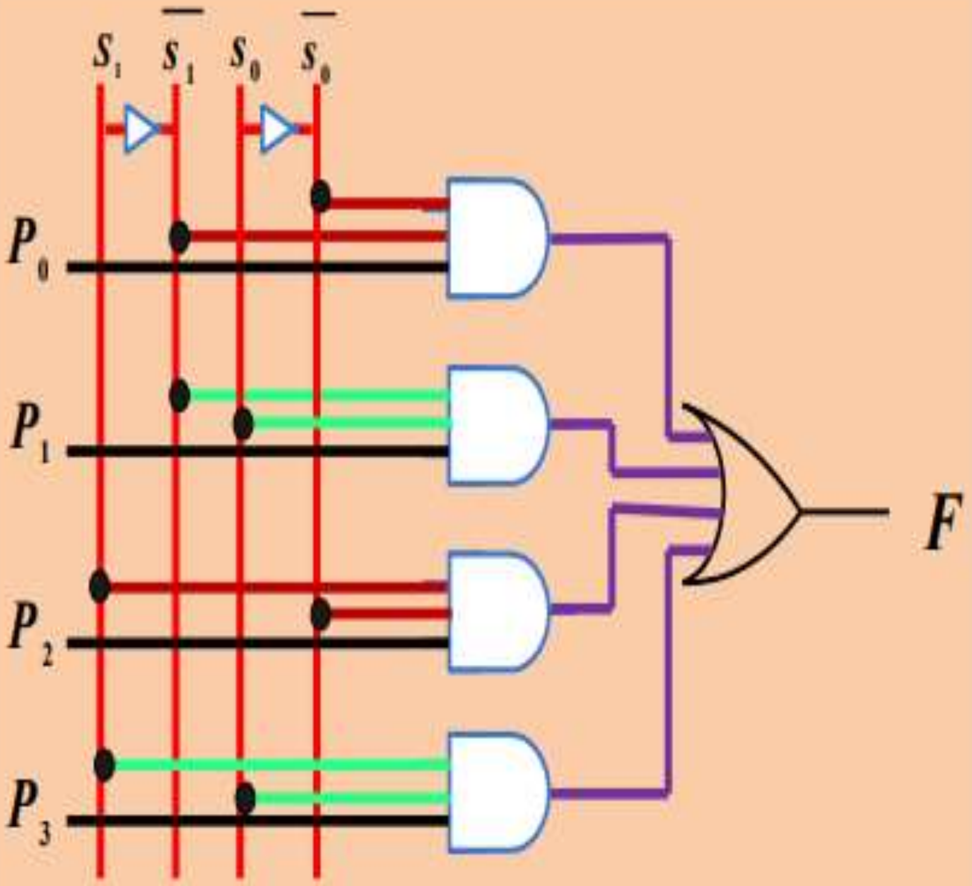
- ✓ 4-data input MUX
- ✓ $s_1, s_0$ - Select lines.
- ✓ $p_0, p_2, p_3, p_1$ - Input lines.
- ✓ $F$- Single output line.

| Select lines | | Output |
|---|---|---|
| $S_1$ | $S_0$ | $F$ |
| 0 | 0 | $P_0$ |
| 0 | 1 | $P_1$ |
| 1 | 0 | $P_2$ |
| 1 | 1 | $P_3$ |

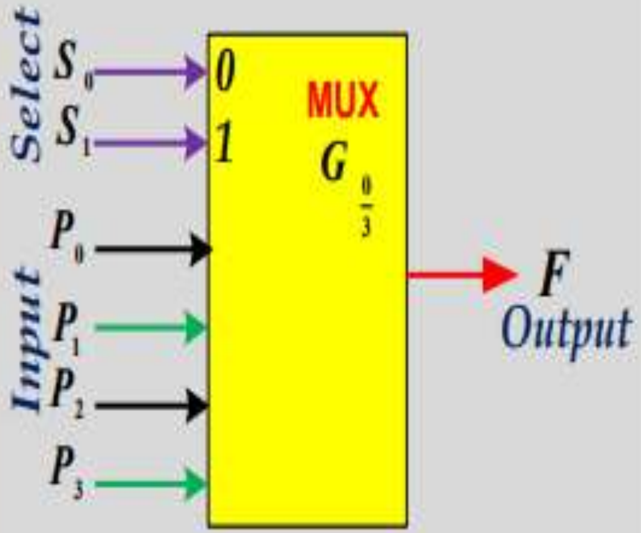


$$F = \overline{S_1}\,\overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$

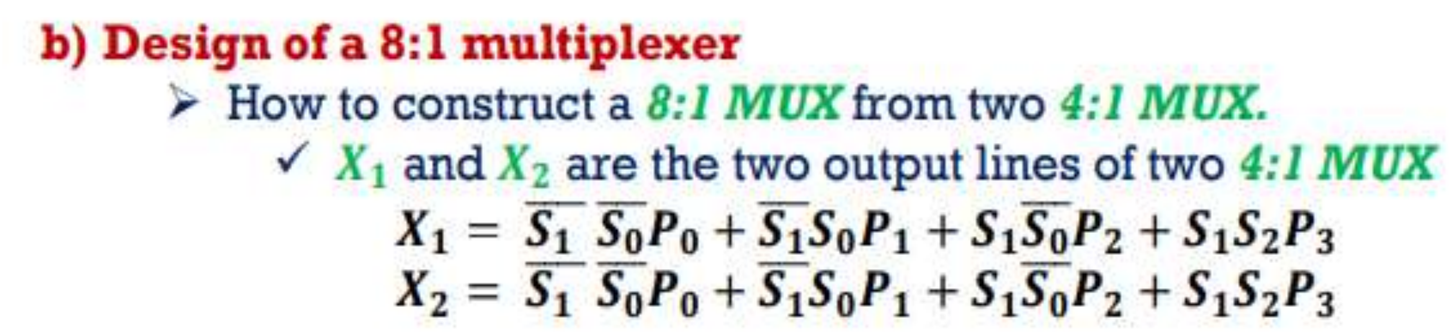
# MUX implementation



Internal logic circuit for 4-1 MUX



4:1 MUX module
"digital element"

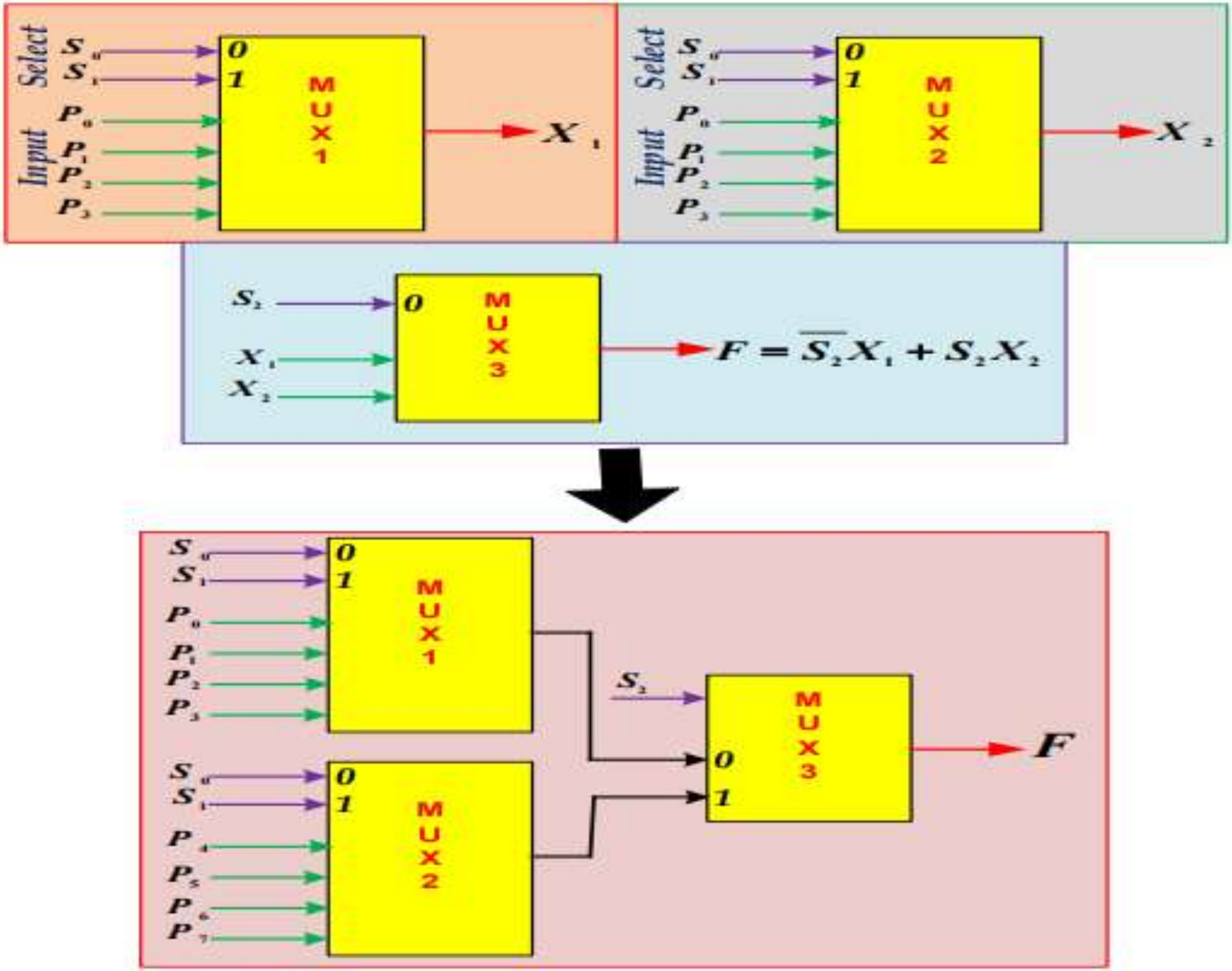


## b) Design of a 8:1 multiplexer

- How to construct a *8:1 MUX* from two *4:1 MUX.*
  - ✓ $X_1$ and $X_2$ are the two output lines of two *4:1 MUX*

$$X_1 = \overline{S_1}\ \overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$
$$X_2 = \overline{S_1}\ \overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$

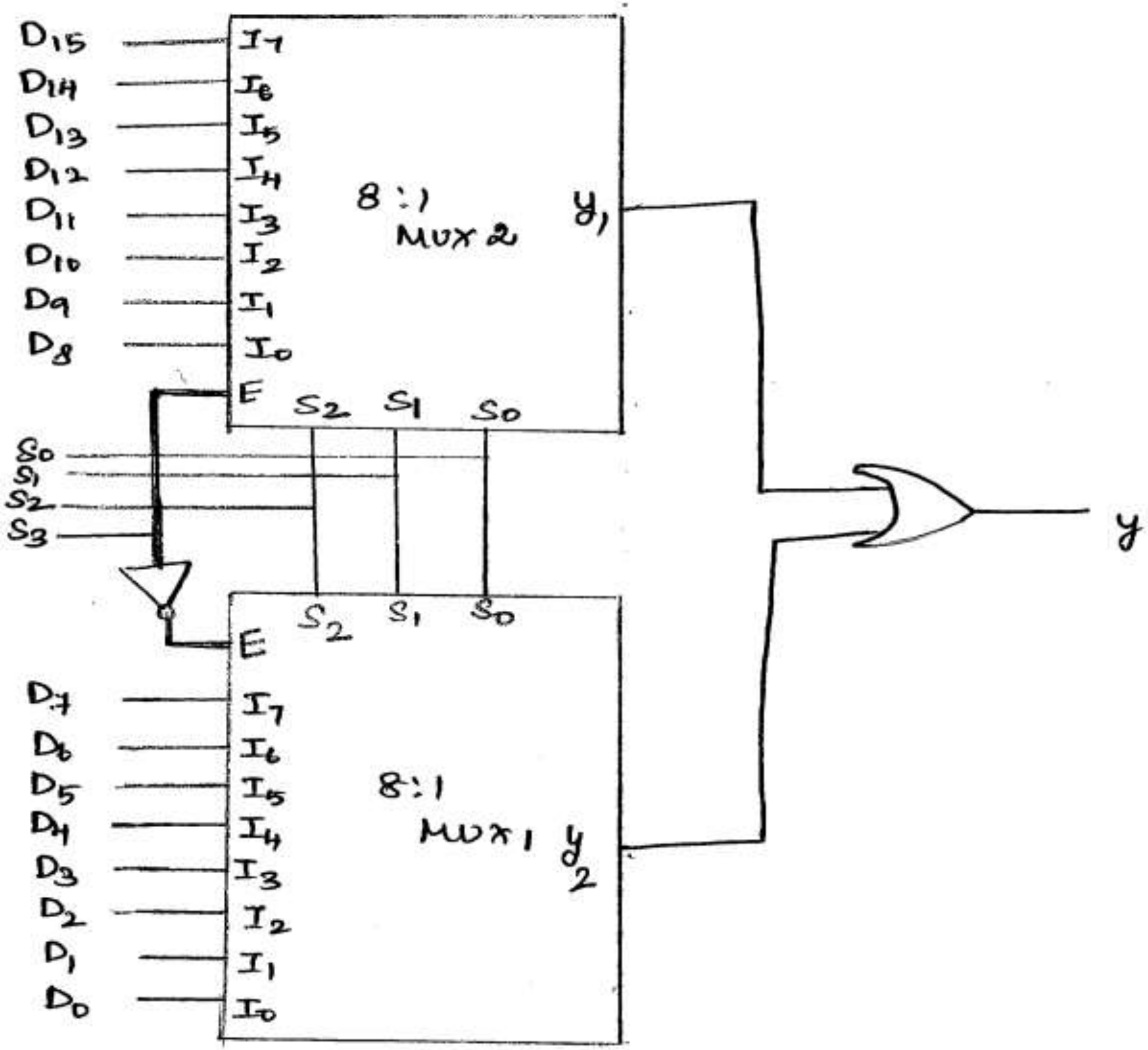
| S0 | S1 | S2 | INPUT |
|---|---|---|---|
| 0 | 0 | 0 | P0 |
| 0 | 0 | 1 | P1 |
| 0 | 1 | 0 | P2 |
| 0 | 1 | 1 | P3 |
| 1 | 0 | 0 | P4 |
| 1 | 0 | 1 | P5 |
| 1 | 1 | 0 | P6 |
| 1 | 1 | 1 | P7 |



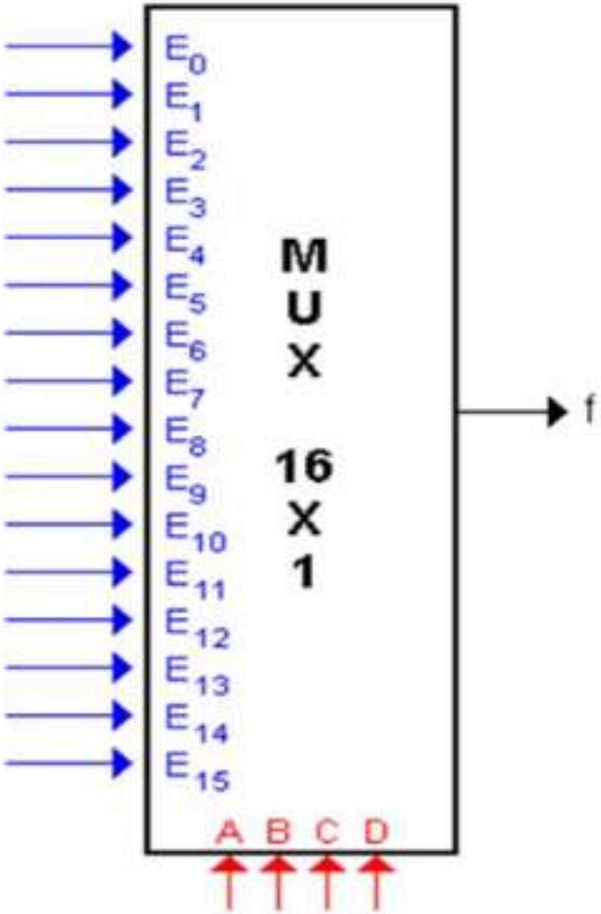
*Design for a 8:1 MUX network*

Example:

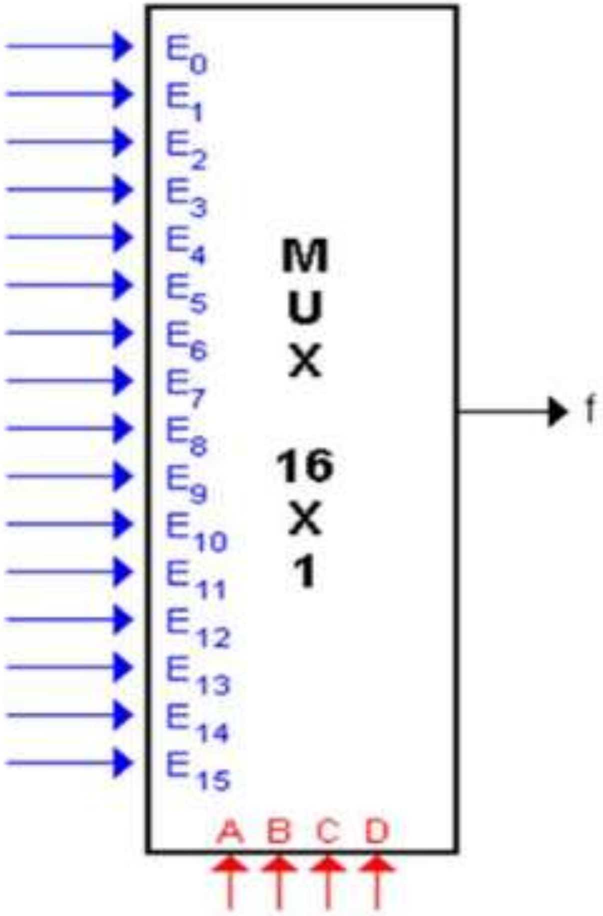
1. Design 16:1 MUX useng 8:1 MUX.



# DESIGN OF 16X1 USING 4X1 MULTIPLEXER



# DESIGN OF 16X1 USING 4X1 MULTIPLEXER



## 16X1 MULTIPLEXER

16=2^4

No.oF SELECT LINES = 4

A,B,C,D

No of 4x1 multiplexer required is

No of inputs =16 so 4x4 =16

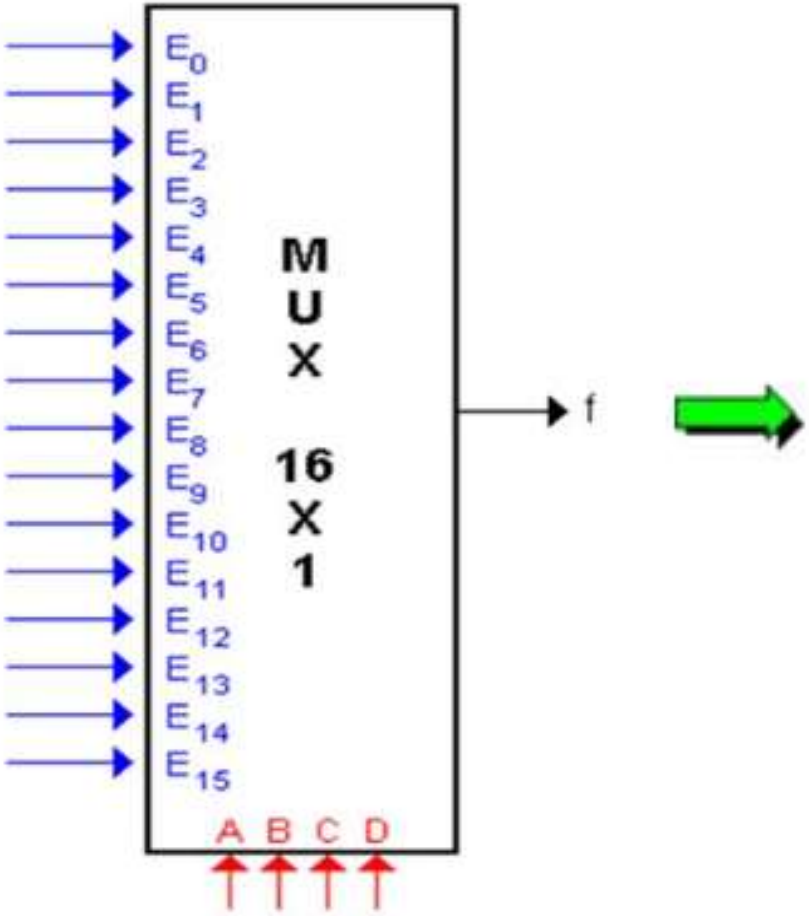
So 4 4x1 will be needed

4 4x1 multiplexer will give 4 outputs but no.of output required is 1

so we need to add one more 4x1 mux.

TOTAL NO.OF MULTIPLEXER REQUIRED IS 5

# DESIGN OF 16X1 USING 4X1 MULTIPLEXER



## 16X1 MULTIPLEXER

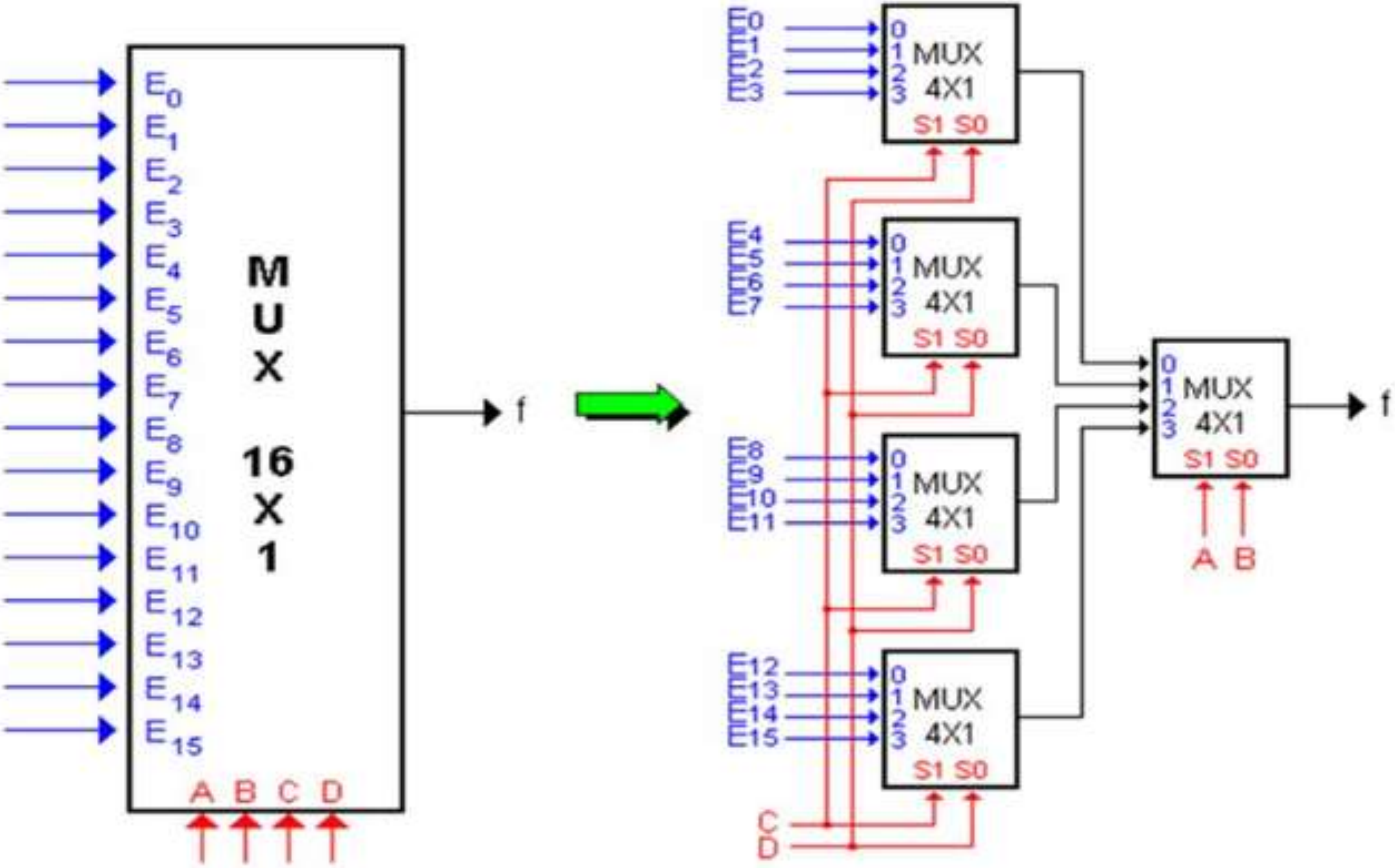
Now no.of select lines available is 4
We need 10 select lines for 5 4x1 mux

How are we going to utilize the select lines?

C,D WILL BE A COMMON SELECT LINE FOR THE FIRST FOUR MUX

A,B WILL BE THE SELECT INPUT FOR THE FIFTH 4X1 mux.

# DESIGN OF 16X1 USING 4X1 MULTIPLEXER



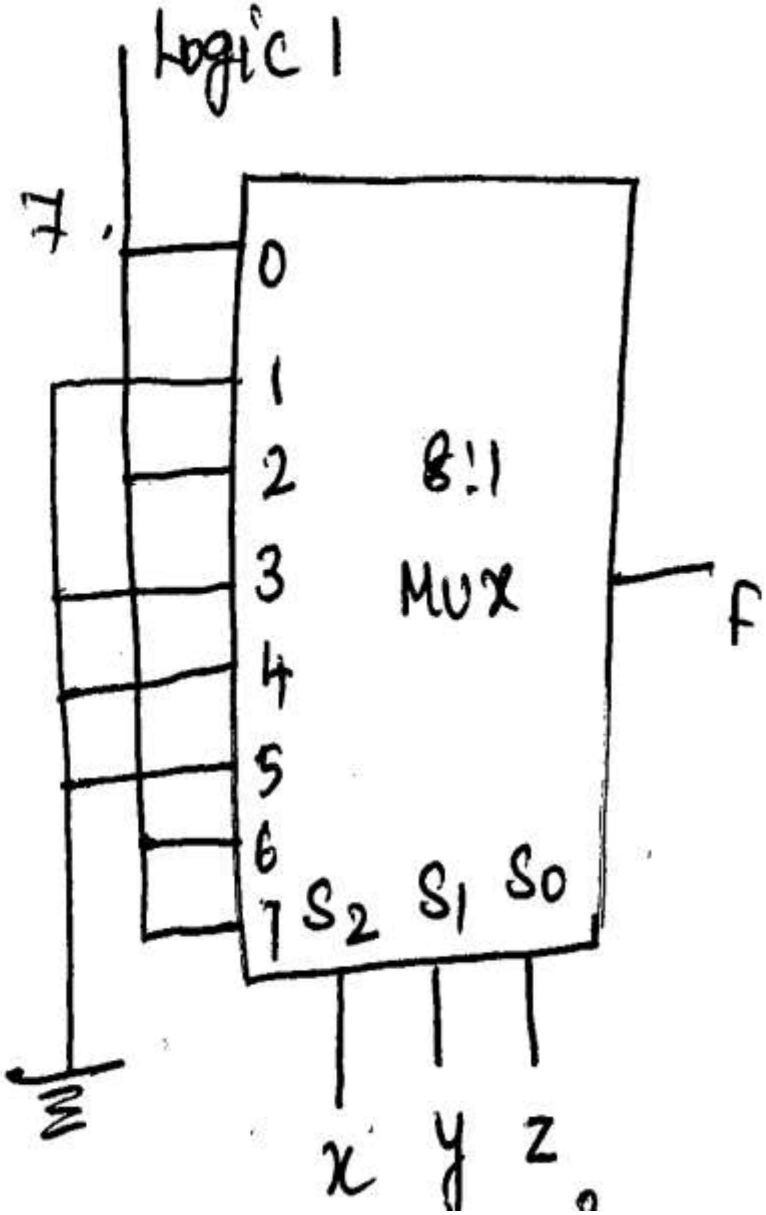
Implement the given function using multiplexer

$f(x,y,z) = \Sigma(0,2,6,7)$

Logic 1 for 0, 2, 6, 7.

$2^3 = 8:1$ MUX

Logic 0 for 1, 3, 5, 4



6. Implementation of following Boolean function using 4:1 mux.

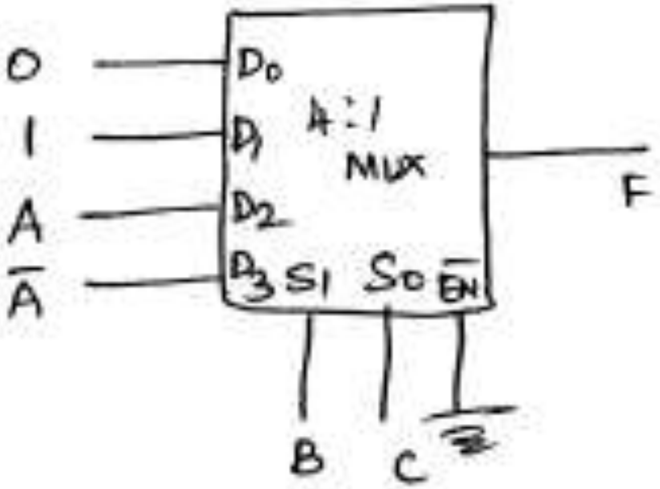
$$f(A,B,C) = \Sigma m(1,3,5\ 6)$$

Logic 1 for 1, 3, 5 6

Logic 0 for 0, 2, 4, 7

|   | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|
| $\bar{A}$ | 0 | ① | 2 | ③ |
| A | 4 | ⑤ | ⑥ | 7 |
|   | 0 | 1 | A | $\bar{A}$ |

Implementation table



* If minterms in a column are not circled 0 is applied.

* If both are circled 1 is applied

* If row 2 is encircled A is applied.

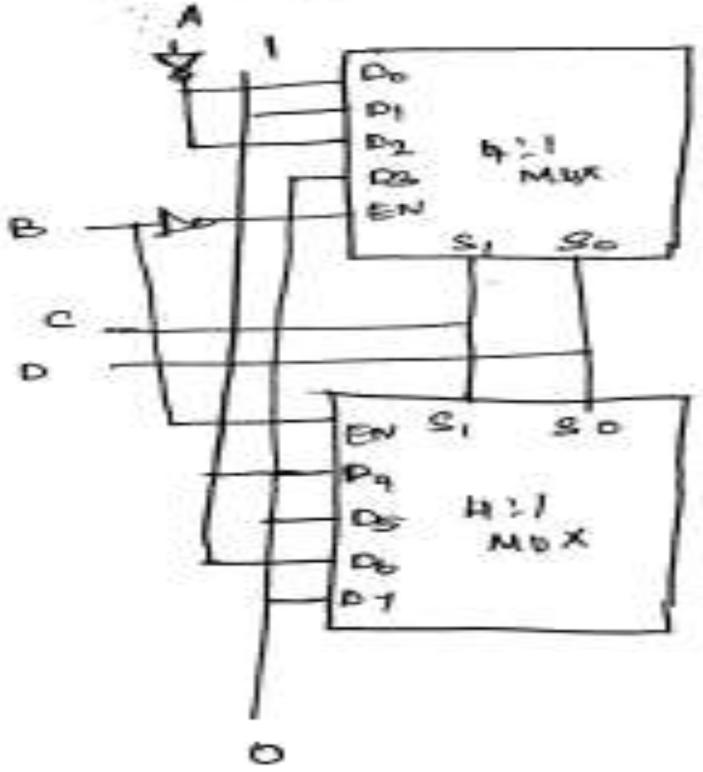
* If row 1 is encircled $\bar{A}$ is applied.

7. Implement the following Boolean function using 4:1 MUX

$F(A,B,C,D) = \Sigma m(0,1,2,4,6,9,12,14)$.

Implementation table.

|  | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{A}$ | ⓪ | ① | ② | 3 | ④ | 5 | ⑥ | 7 |
| $A$ | 8 | ⑨ | 10 | 11 | ⑫ | 13 | ⑭ | 15 |
|  | $\bar{A}$ | 1 | $\bar{A}$ | 0 | 1 | 0 | 1 | 0 |



11. Implement full adder using two 4:1 mux.

Variables A, B, C

select inputs = 2 $\Rightarrow n-1 = 3-1$

data inputs $2^{n-1} = 2^{(3-1)} = 4$

Sum = $\Sigma m(1, 2, 4\ 7)$

Carry = $\Sigma m(3, 5, 6, 7)$.

| 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 |

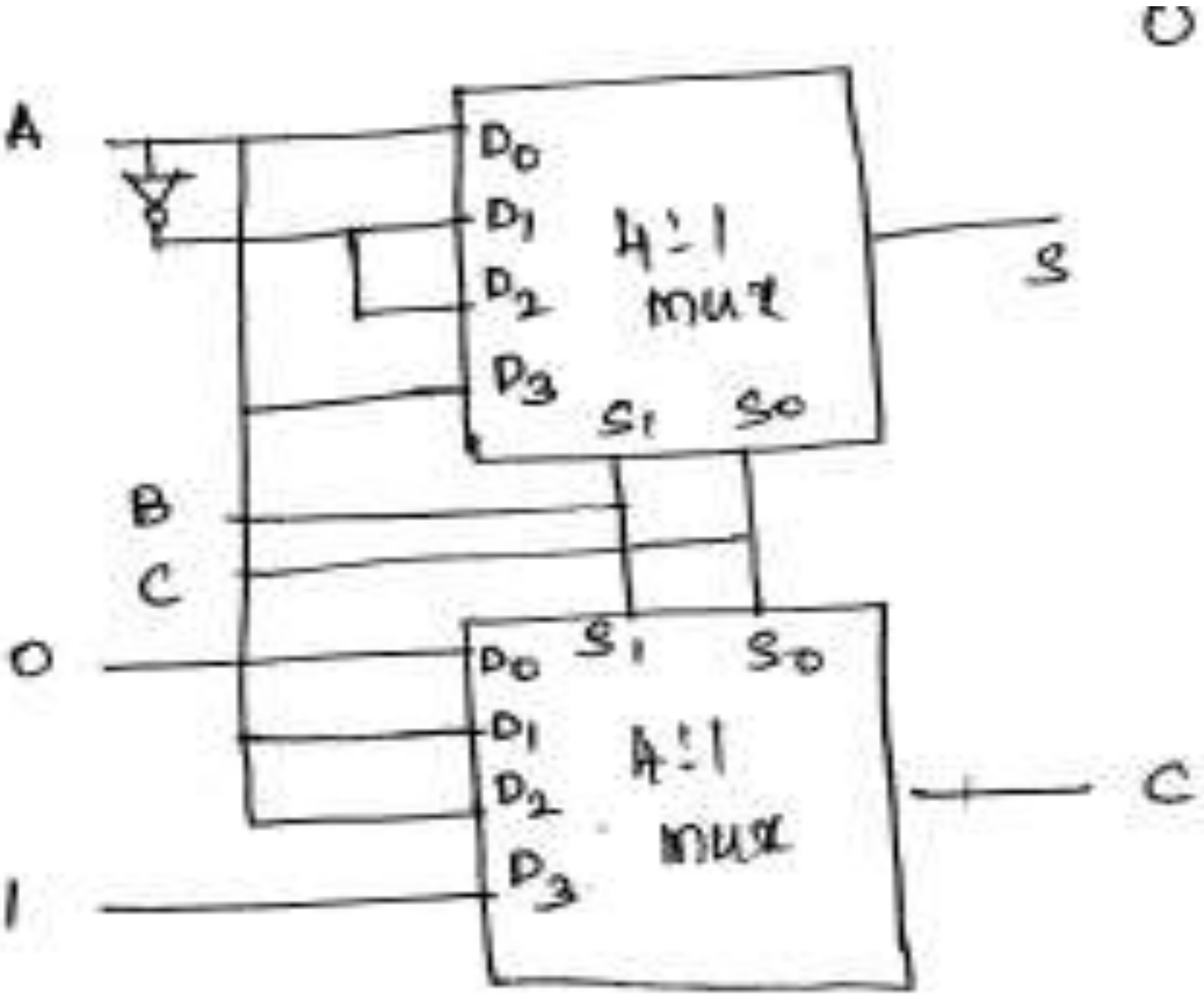
Implementation table

Sum

|  | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|
| $\bar{A}$ | 0 | ① | ② | 3 |
| A | ④ | 5 | 6 | ⑦ |
|  | A | $\bar{A}$ | $\bar{A}$ | A. |

Carry

|  | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|
| $\bar{A}$ | 0 | 1 | 2 | ③ |
| A | 4 | ⑤ | ⑥ | ⑦ |
|  | 0 | A | A | 1 |


A
D0
D1
D2
D3
4:1
mux
S
S1
S0
B
C
0
S1
S0
D0
D1
D2
D3
4:1
mux
C
1

# Demultiplexer (Data Distributor)

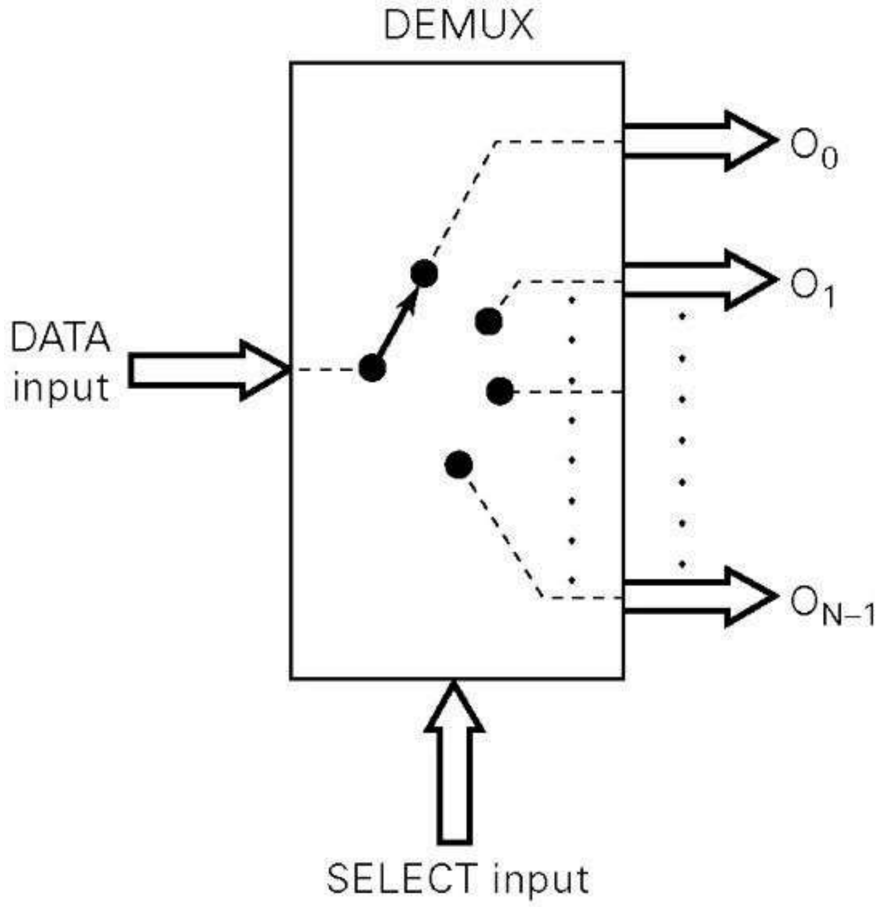
- Definition : A DEMULTIPLEXER (DEMUX) basically reverses the multiplexing function. It takes data from one line and distributes them to a given number of output lines. For this reason, the demultiplexers is also known as a data distributor.

- Single data input lines
- Some select line (less than the no. of output lines)
- Several output line

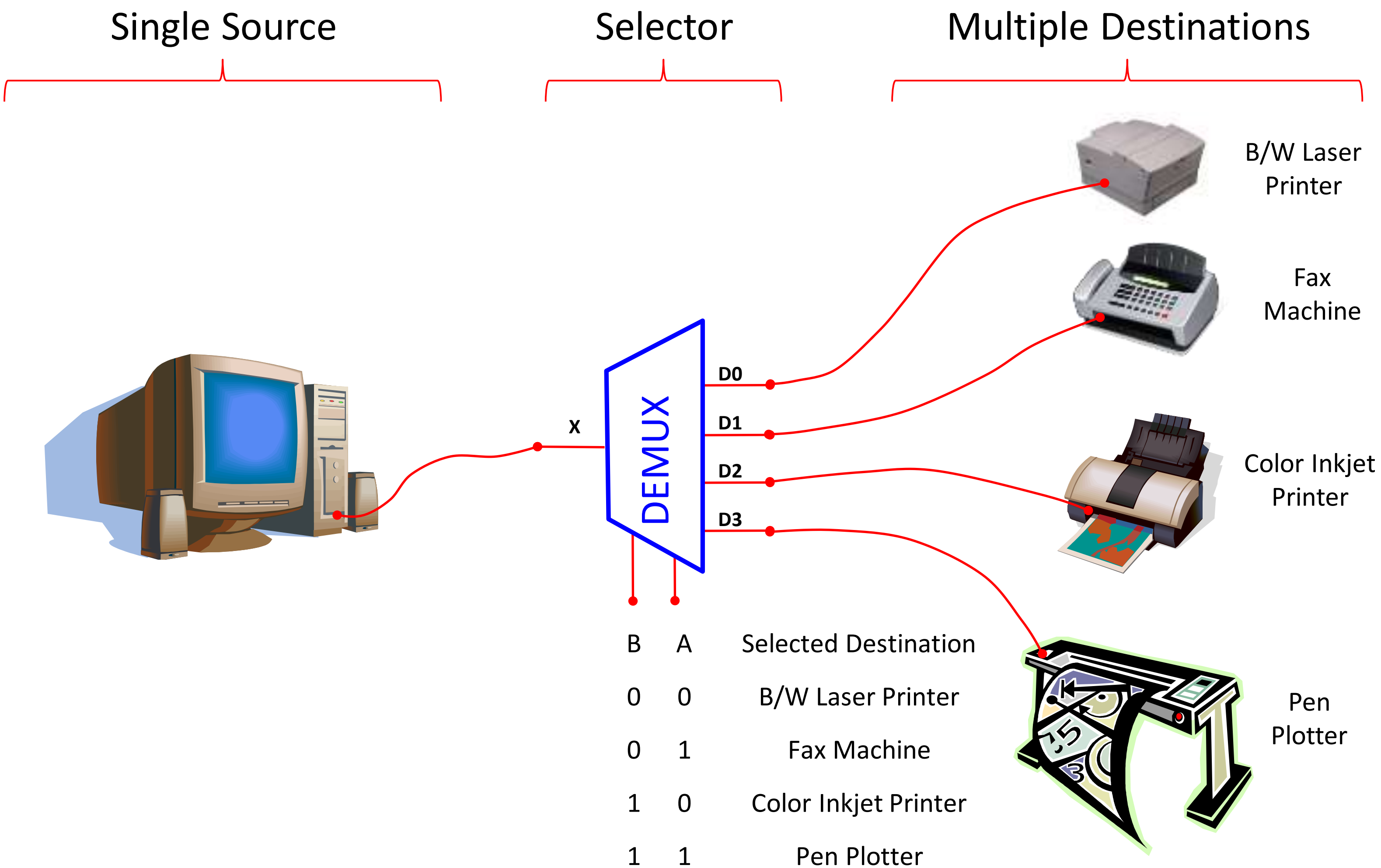
- If there are n data output lines and m select lines, then

$$2^m = n$$

# Functional Diagram Of a  Demultiplexer

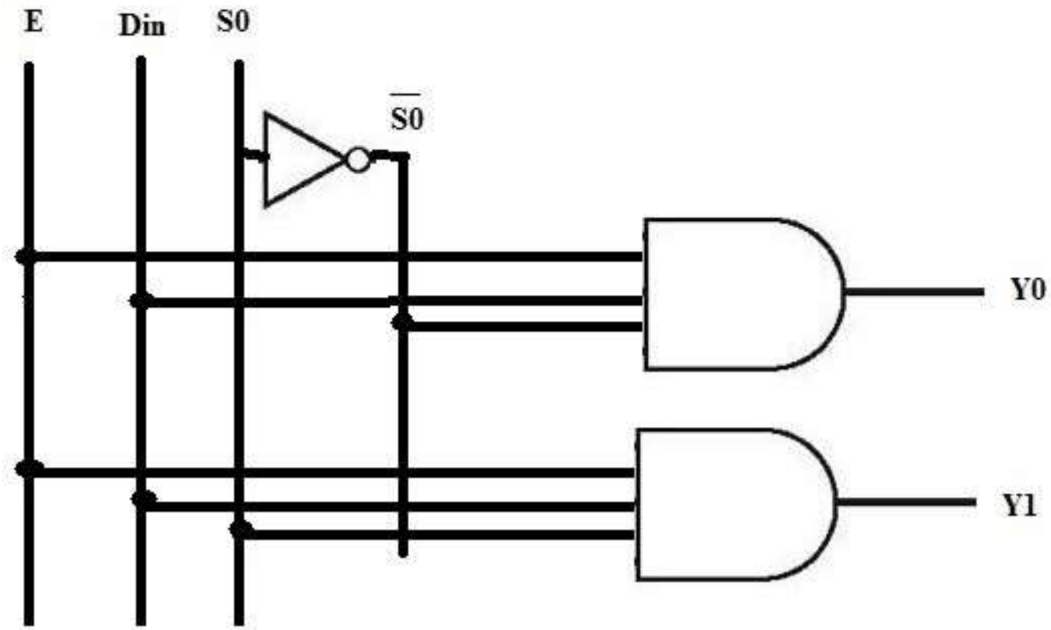


# Typical Application of a DEMUX



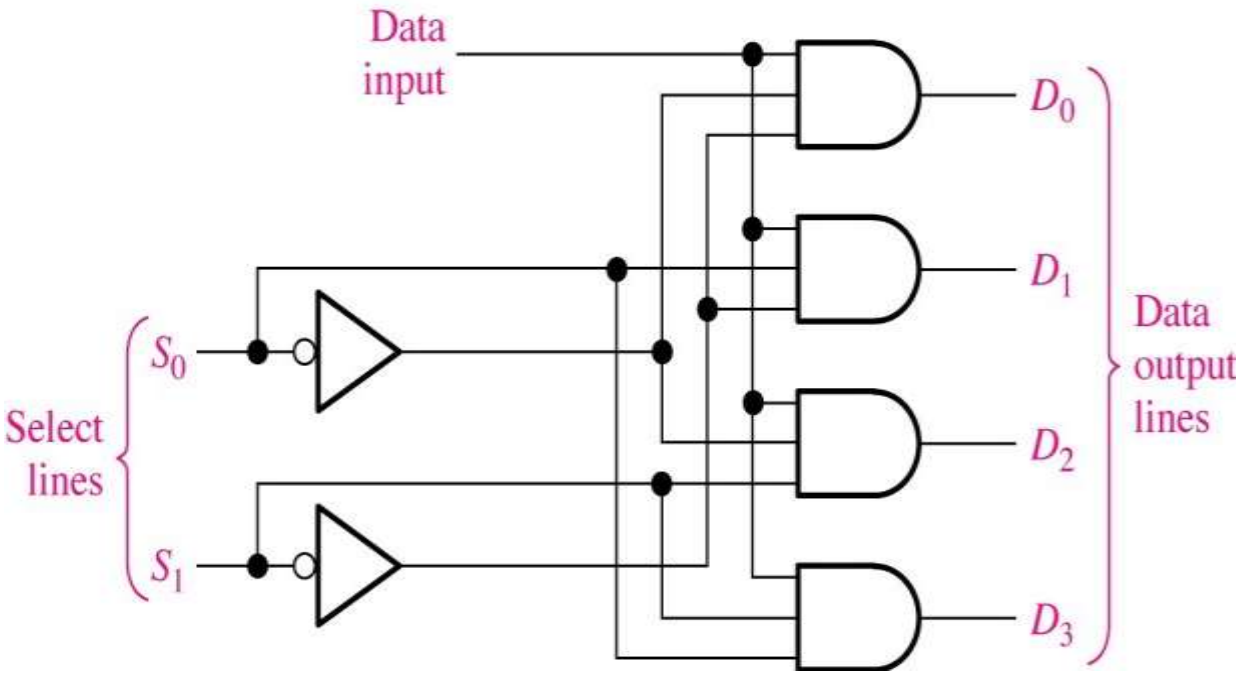
| B | A | Selected Destination |
|---|---|---|
| 0 | 0 | B/W Laser Printer |
| 0 | 1 | Fax Machine |
| 1 | 0 | Color Inkjet Printer |
| 1 | 1 | Pen Plotter |

# 1 : 2 Demultiplexer



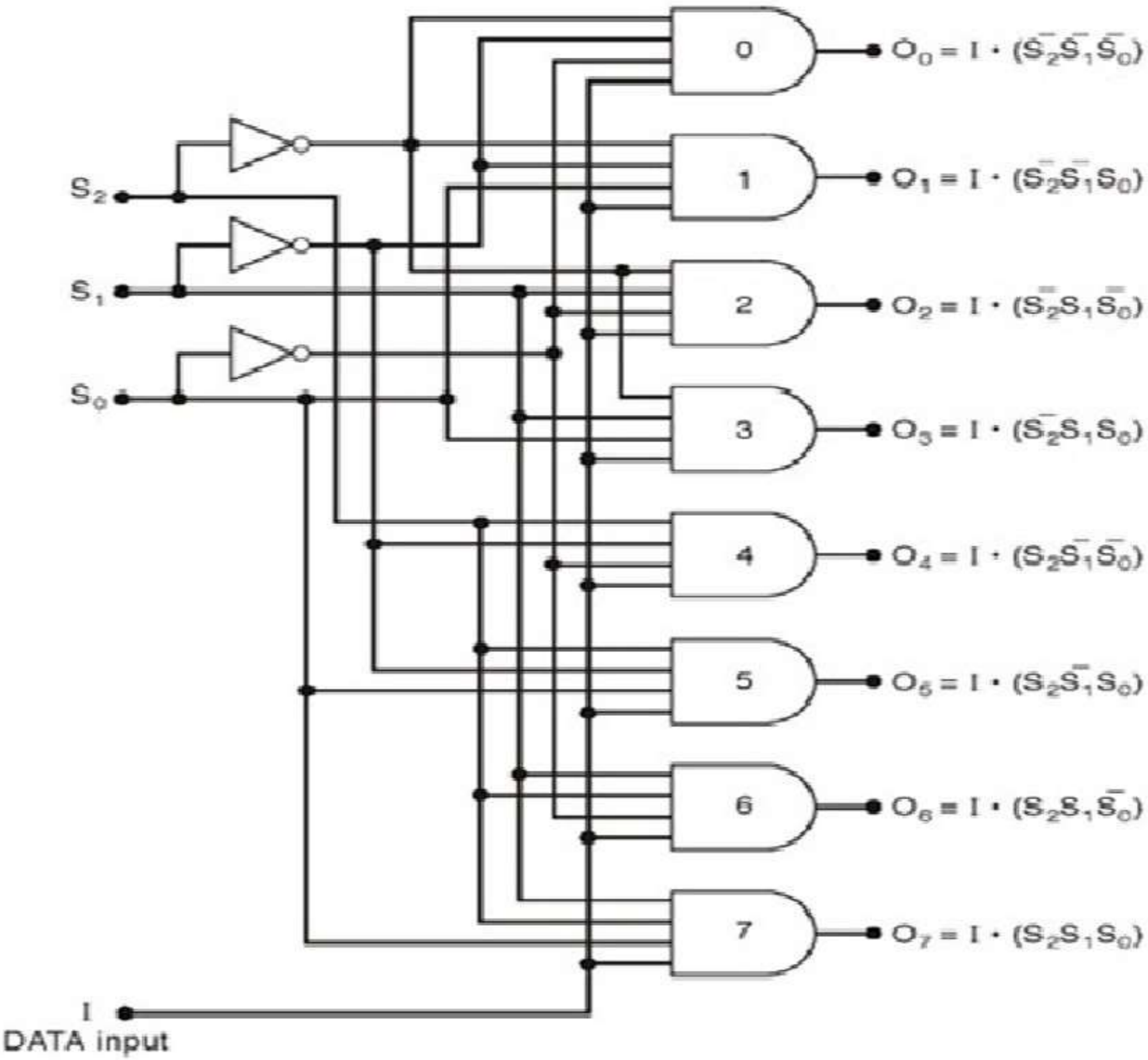
| $S_0$ | $Y_0$ | $Y_1$ |
|---|---|---|
| 0 | D | 0 |
| 1 | 0 | D |

# 1 : 4 Demultiplexer



| $S_0$ | $S_1$ | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|---|
| 0 | 0 | D | 0 | 0 | 0 |
| 0 | 1 | 0 | D | 0 | 0 |
| 1 | 0 | 0 | 0 | D | 0 |
| 1 | 1 | 0 | 0 | 0 | D |

# 1 : 8 Demultiplexer


0
$O_0 = I \cdot (\bar{S}_2\bar{S}_1\bar{S}_0)$
1
$O_1 = I \cdot (\bar{S}_2\bar{S}_1S_0)$
$S_2$
$S_1$
2
$O_2 = I \cdot (\bar{S}_2S_1\bar{S}_0)$
$S_0$
3
$O_3 = I \cdot (\bar{S}_2S_1S_0)$
4
$O_4 = I \cdot (S_2\bar{S}_1\bar{S}_0)$
5
$O_5 = I \cdot (S_2\bar{S}_1S_0)$
6
$O_6 = I \cdot (S_2S_1\bar{S}_0)$
7
$O_7 = I \cdot (S_2S_1S_0)$
I
DATA input

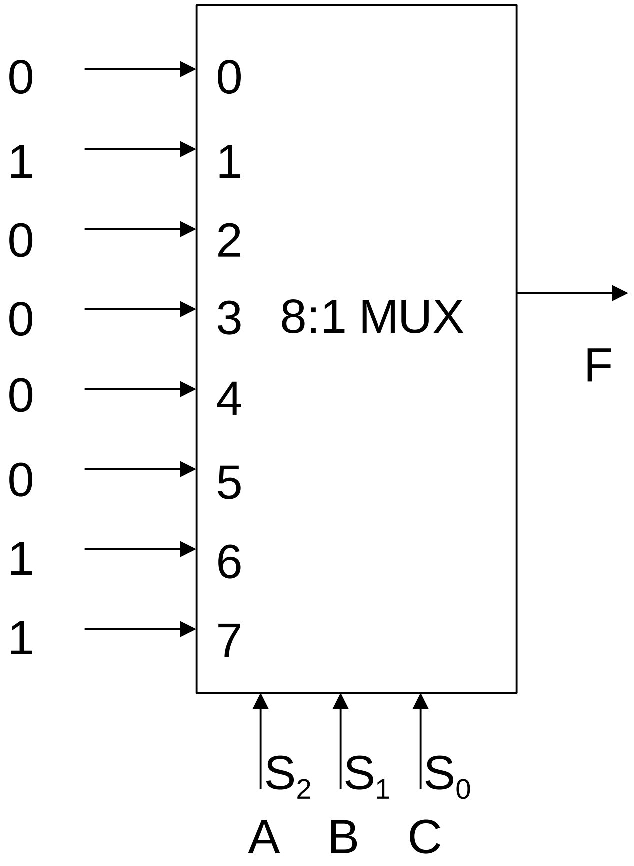
# 1 : 8 Demultiplexer (Truth Table)

| $S_0$ | $S_1$ | $S_3$ | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | D | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | D | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | D | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | D | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | D | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | D | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D |

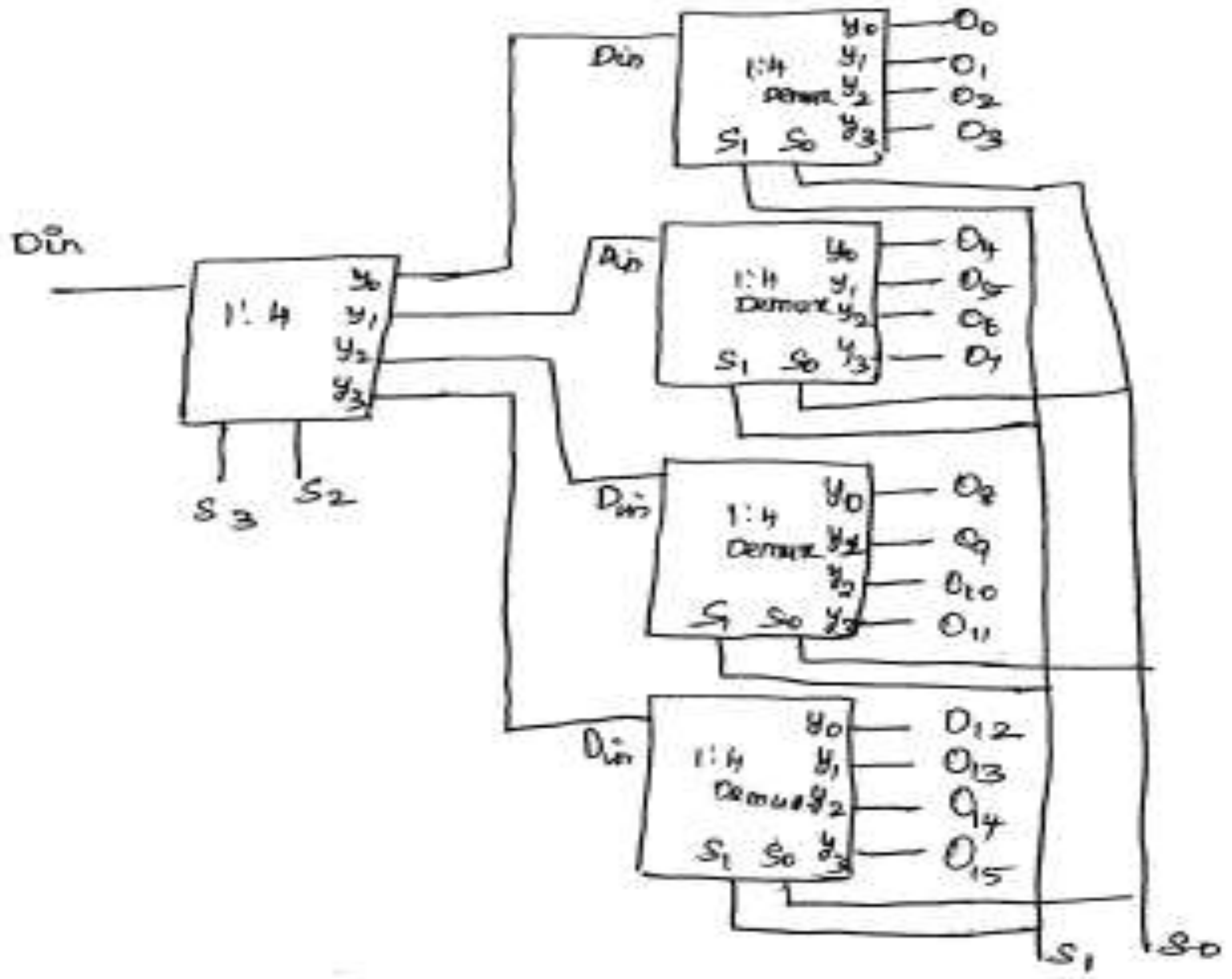
# Implementation Of Logic Functions using Multiplexer

f(a, b, c) = a’b’c + ab

| **A** | **B** | **C** | **F** |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



2. Implement 1:16 demux using 1:4 Demux.



3. Implementation of full subtractor using Demux.

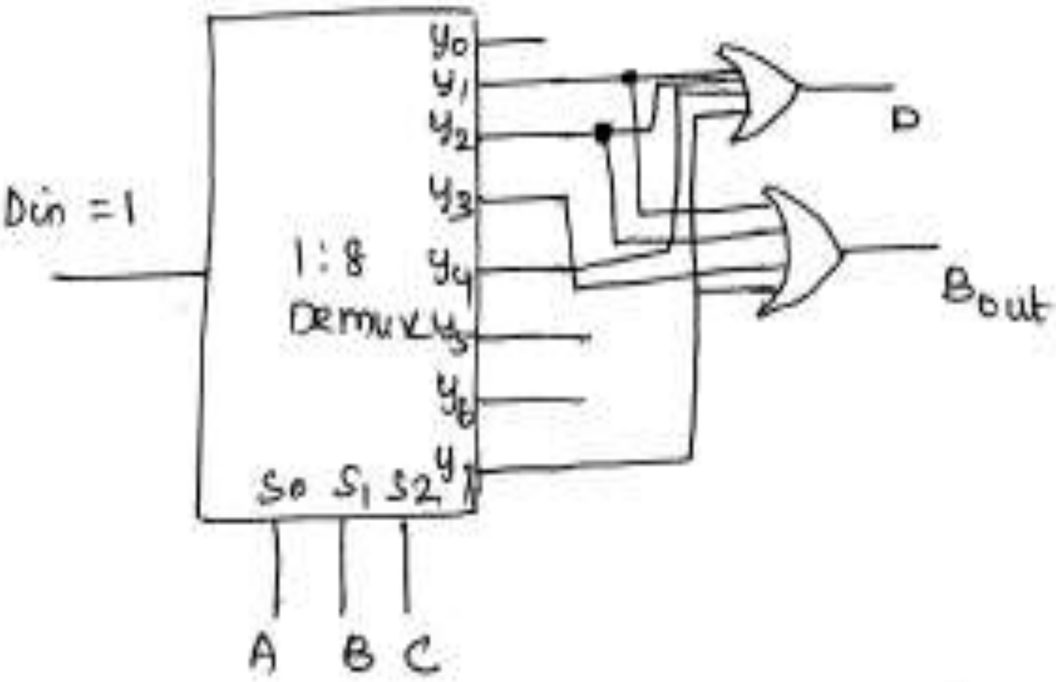
Truth table

| A | B | C | Sub | Bout |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Diff = Sub = $f(A,B,C)$ = $\Sigma m(1,2,4,7)$

$B_{out}$ = $f(A,B,C)$ = $\Sigma m(1,2,3,7)$.

* with $D_{in} = 1$, demux gives the minterms at the o/p by ORing required minterms


Din =1
1 : 8
Demux
y0
y1
y2
y3
y4
y5
y6
y7
S0
S1
S2
A
B
C
D
Bout

THANK YOU