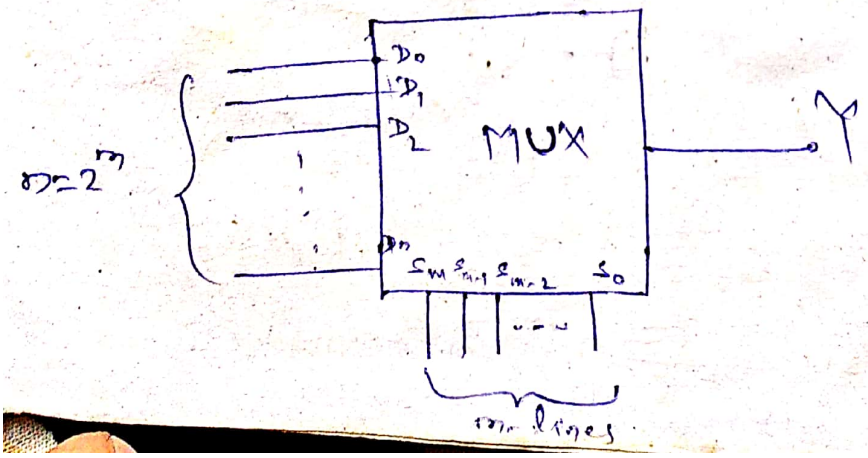
# MULTIPLEXERS & DE-MUX.



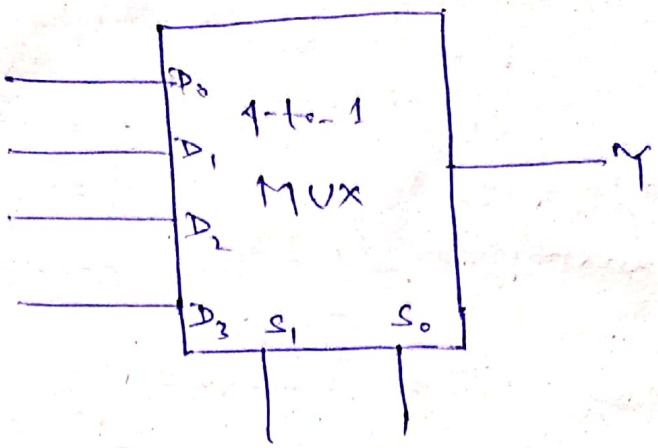
The word "Multiplex" stands for "many into one". Multiplexing is a process of transmitting a large number of information over a single transmission line or channel.

MUX:- A multiplexer is a combinational ckt which combine a number of data input or information and transmit it over a single transmission line. In other word, a multiplexer is a combinational ckt or data selector that can select any one of input from a large number of inputs and transmit it through a single transmission line. Thus, a multiplexer has several number of input lines and only one output line. The selection of particular input line is decided by select input line. The no. of input lines of a MUX is also decided by the number of select input lines. If 'm' is the number of select input lines then number of data-input lines $n = 2^m$. A basic block diagram is shown in fig



# Basic four input MUX or 4-to-1 MUX






Above fig. shows a block diagram of a basic 4-to-1 MUX. As the name, it has 4-input lines ($D_0 - D_3$) and $4 = 2^2$, 2-select input lines.

If $S_1 = 0, S_0 = 0$, $Y = D_0$

$S_1 = 0, S_0 = 1$, $Y = D_1$

$S_1 = 1, S_0 = 0$, $Y = D_2$

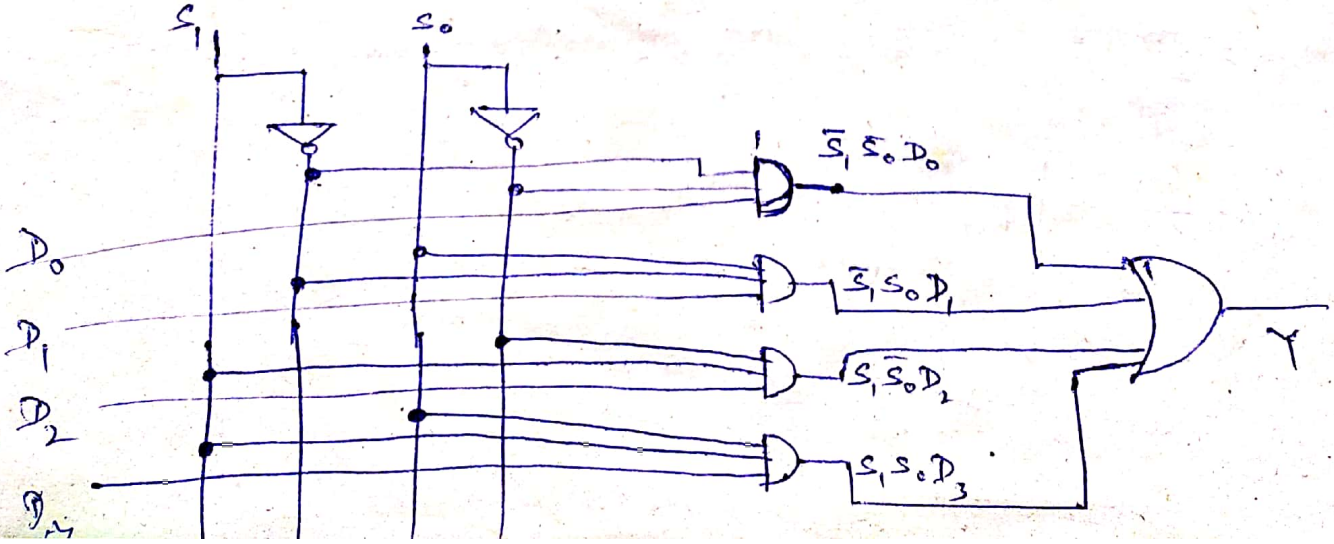
$S_1 = 1, S_0 = 1$, $Y = D_3$

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

By truth table, we can express output in boolean expression as below —

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

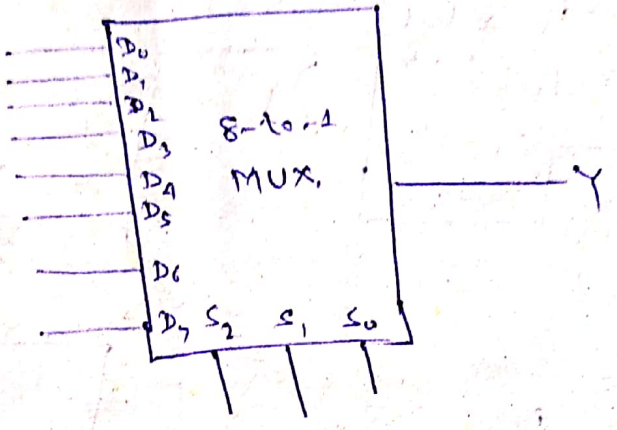
This boolean expression can be implemented by basic logic gates as below —



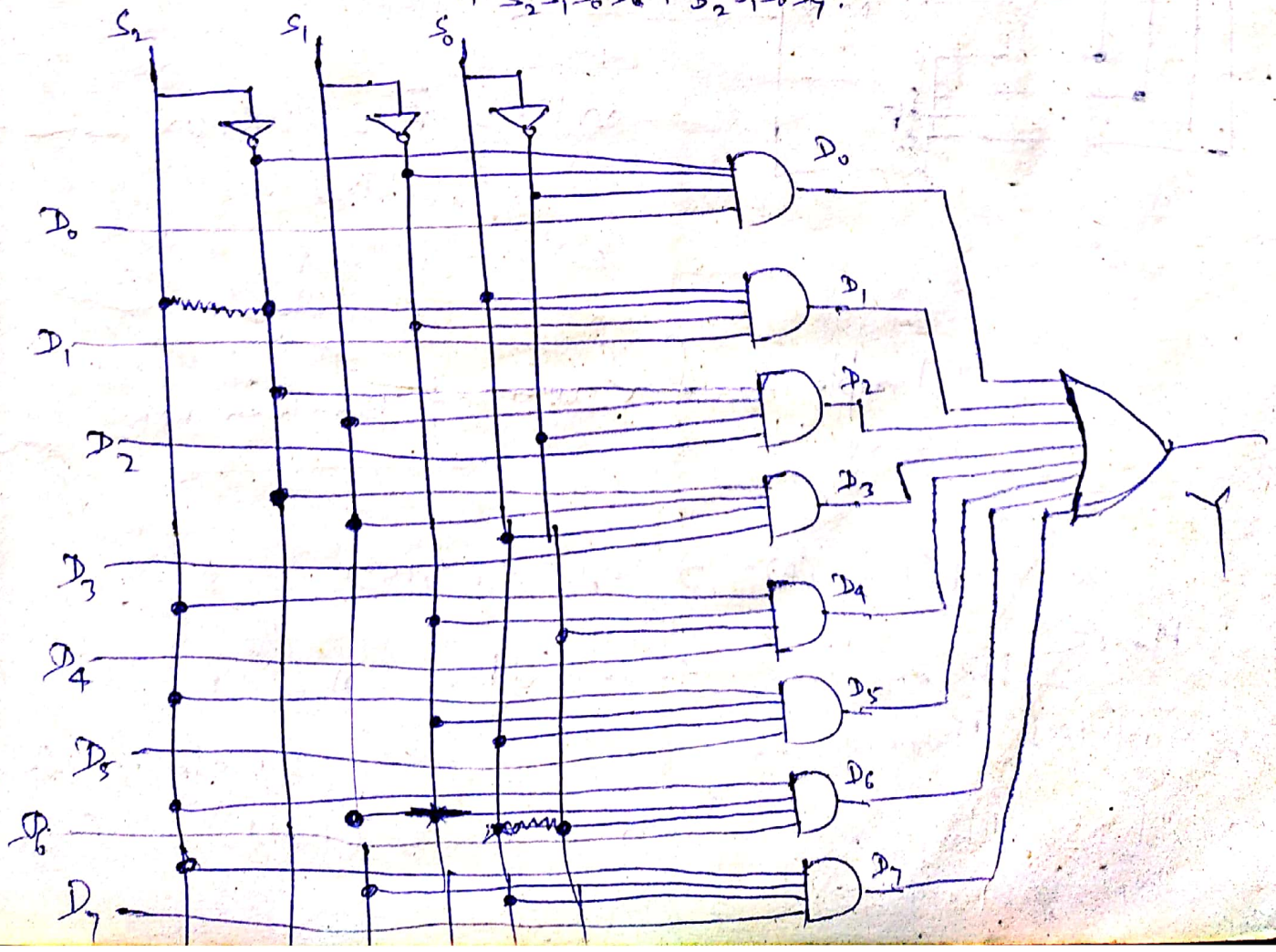
# 8-to-1 MUX

8-to-1 MUX has 8-input lines and $8=2^3$, 3-select ~~input~~ lines and a single output line. The fig. as below shows a block diagram of 8-to-1 MUX.

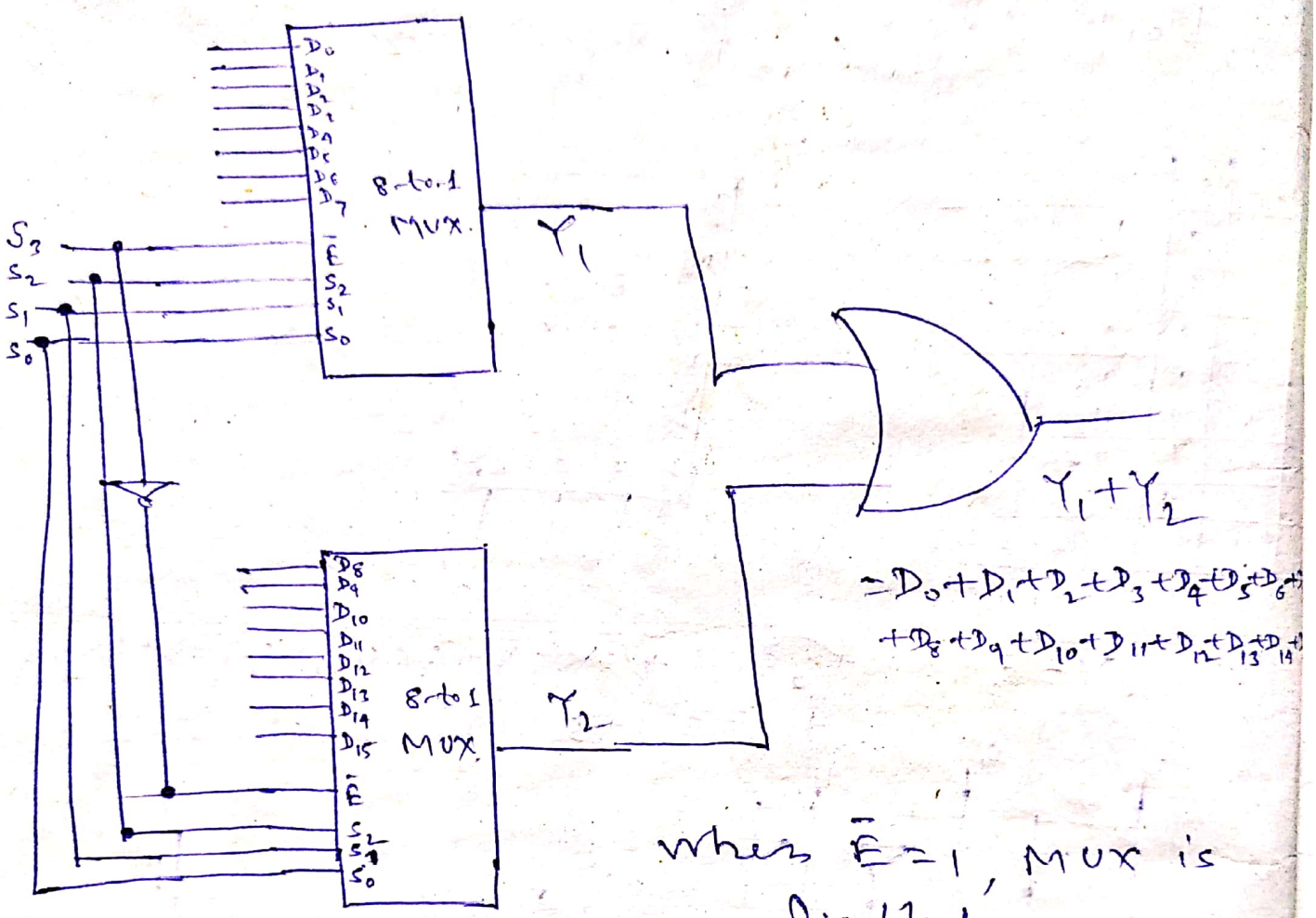
| $S_2$ | $S_1$ | $S_0$ | Y |
|---|---|---|---|
| 0 | 0 | 0 | $D_0$ |
| 0 | 0 | 1 | $D_1$ |
| 0 | 1 | 0 | $D_2$ |
| 0 | 1 | 1 | $D_3$ |
| 1 | 0 | 0 | $D_4$ |
| 1 | 0 | 1 | $D_5$ |
| 1 | 1 | 0 | $D_6$ |
| 1 | 1 | 1 | $D_7$ |




$$Y = \bar{S}_2\bar{S}_1\bar{S}_0 D_0 + \bar{S}_2\bar{S}_1 S_0 D_1 + \bar{S}_2 S_1\bar{S}_0 D_2 + \bar{S}_2 S_1 S_0 D_3 + S_2\bar{S}_1\bar{S}_0 D_4 + S_2\bar{S}_1 S_0 D_5 + S_2 S_1\bar{S}_0 D_6 + S_2 S_1 S_0 D_7.$$



# Implementation of 16-to-1 MUX by using two 8-to-1 MUX.




When $\bar{E}=1$, MUX is disabled

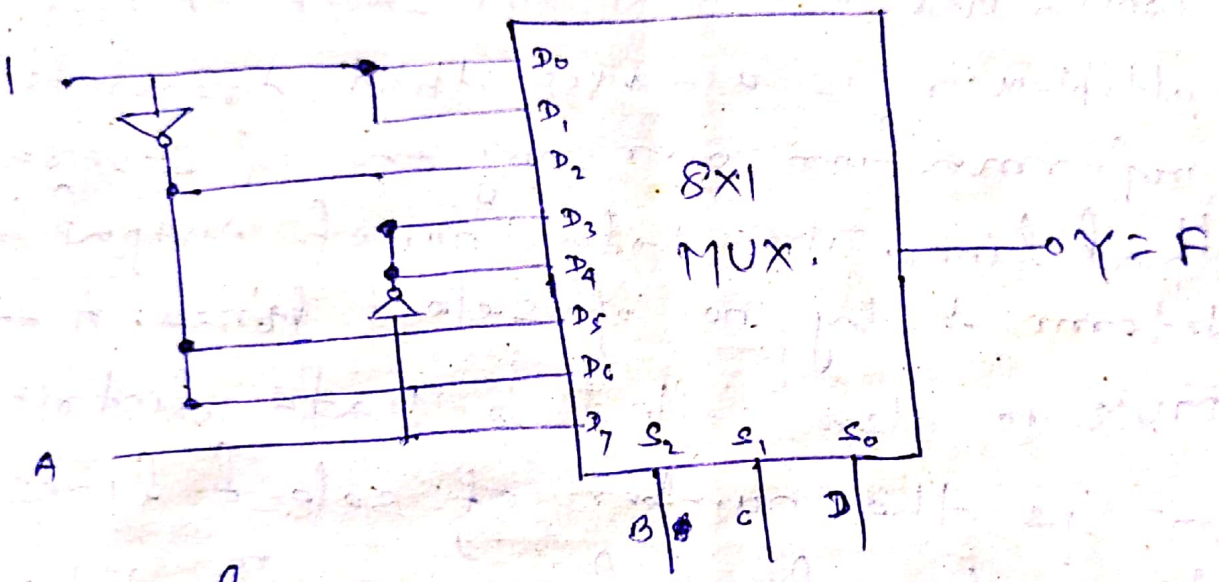
When $\bar{E}=0$, MUX is Enabled.

## Implementation of Boolean Exp. by using MUX

$$F(A,B,C,D) = \Sigma(0,1,3,4,8,9,15)$$

Here, there are 4-variables — A, B, C, D.
Out of three are connected to 3-select input
Thus, this function can be implemented by 8-to-1 MUX.

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




## Application of MUX

i> Data Routing :- Used to route data from several source to one destination.

ii> ~~ii~~

iii> Logic function generator :- It can be used to implement logic function eqn.

iii> Parallel to serial converter.

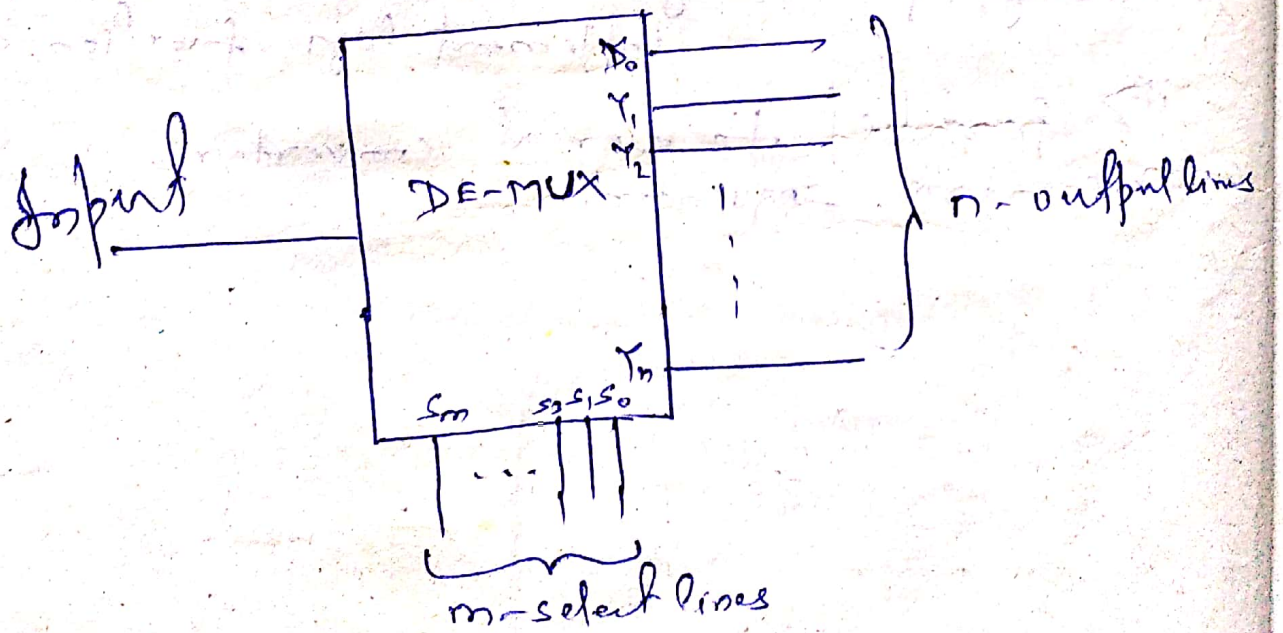
iv> Control sequencer.

# DE-MULTIPLEXERS (DATA DISTRIBUTORS)



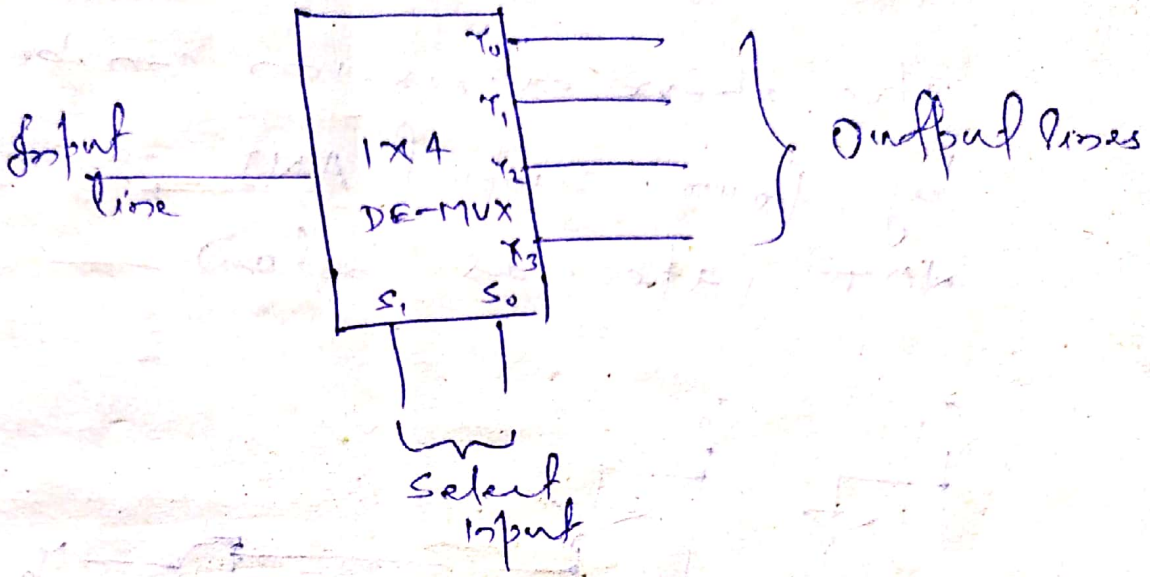
Demultiplex means "one to many". Demultiplexing is a process by which one inform is transmitted over one of the several transmitting lines.

De-multiplexer is a combinational ckt which has one input and several output lines Demultiplexer is a device that transmits one information over any one of several output lines. The total no. of output lines is determined by no. of select lines. A DE-MUX is also called a Data distributo If 'm' is the number of select lines th number of output lines $n = 2^m$. A basic block diagram of DE-MUX is shown in fig as below —



# 1-to-4 (1×4) DE-MUX

1-to-4 DE-MUX has one input and 4-output lines and ($4=2^2$), 2-select lines.



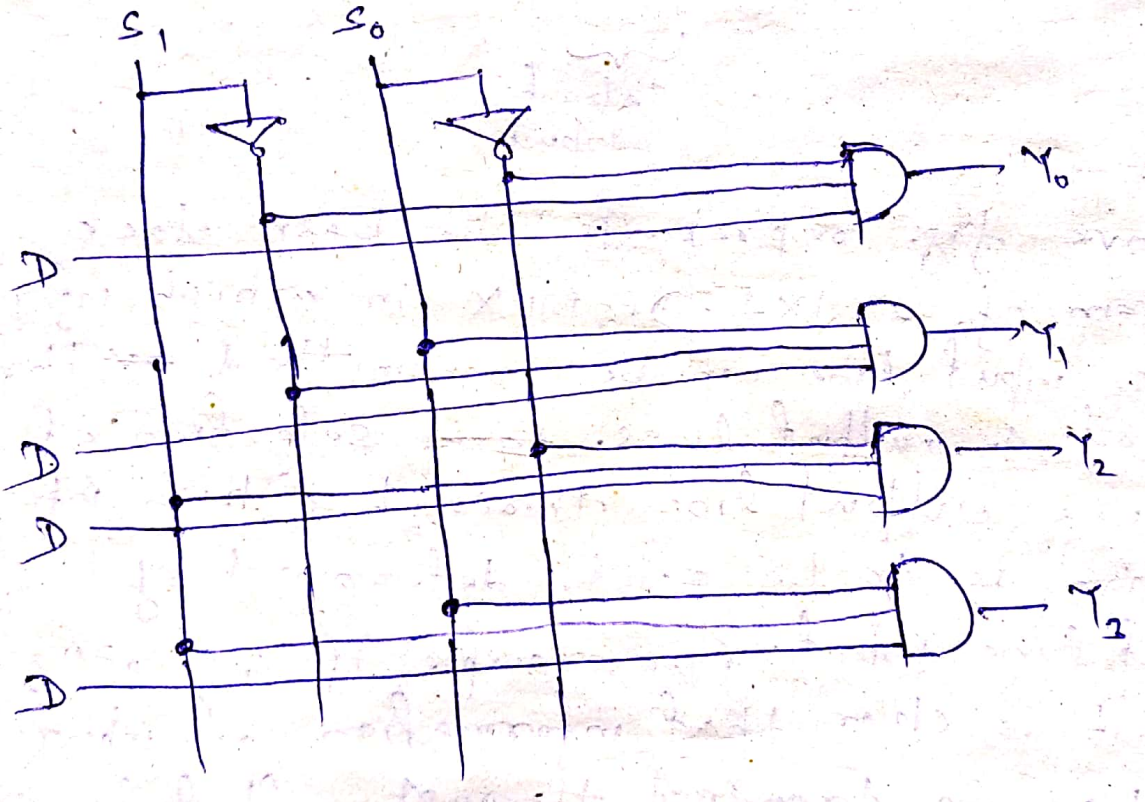

Above fig. represents the basic block diagram of a 1×4-DE-MUX in which information input line can be transmitted through one of 4-output lines. Final selection of any one output line through which information is to be send, determined by select line input. For example, if $S_1=0, S_0=0$, then it is clear that information at input line has to transmit through output line $Y_0$. This can be understand in better way by following truth table.

| Data input | select line input | | Output Y | | | |
|---|---|---|---|---|---|---|
| | | | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| D | 0 | 0 | 0 | 0 | 0 | D |
| D | 0 | 1 | 0 | 0 | D | 0 |
| D | 1 | 0 | 0 | D | 0 | 0 |
| D | 1 | 1 | D | 0 | 0 | 0 |
| | | | $Y_3=S_1S_0D$ | $Y_2=S_1\bar{S}_0D$ | $Y_1=\bar{S}_1S_0D$ | $Y_0=\bar{S}_1\bar{S}_0D$ |

Thus, we see that, Depending select line input, there may be four outputs as

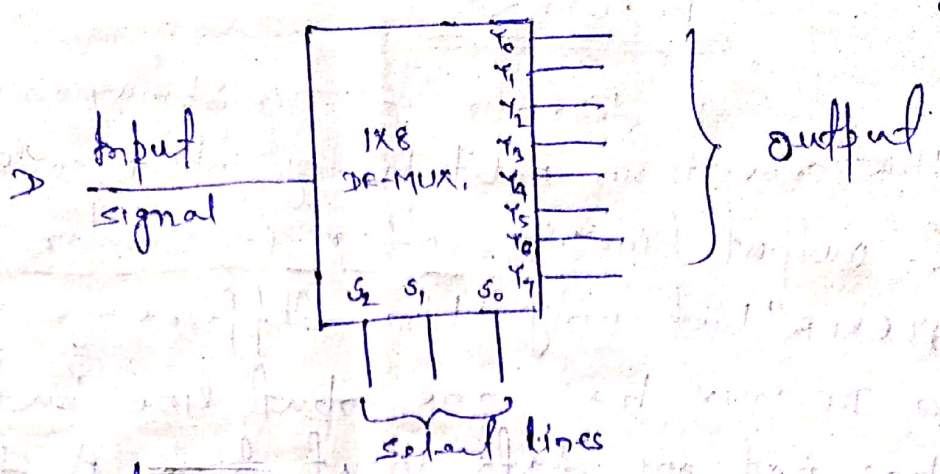
$$Y_0 = \bar{S}_1 \bar{S}_0 D \;,\; Y_1 = \bar{S}_1 S_0 D \;,\; Y_2 = S_1 \bar{S}_0 D \;,\; Y_3 = S_1 S_0 D$$

The above expressions can be implemented by four 3-input AND gates and two NOT gates as below —



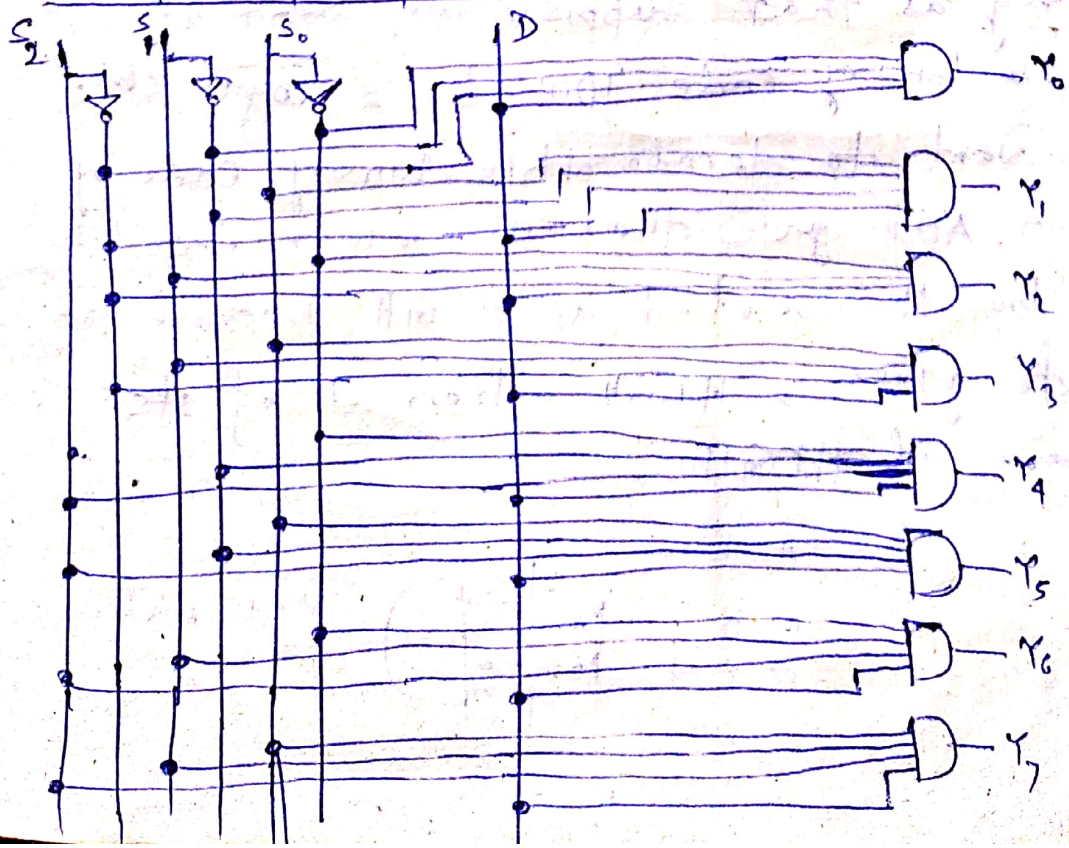

1 to-8 DE-MUX

'1X8' DEMUX has one input, 3-select lines and 8-output lines. In accordance to the input to select line $S_1S_0$, 1-to-8 DEMUX transmits input data through any one particular output line. A basic block diagram is shown in fig-(A)




The operation of 1X8 - DEMUX can be understood in better way by following truth table —

| Data input | Select line input | | | output | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_2$ | $S_1$ | $S_0$ | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| D | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D |
| D | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | D | 0 |
| D | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | D | 0 | 0 |
| D | 0 | 1 | 1 | 0 | 0 | 0 | 0 | D | 0 | 0 | 0 |
| D | 1 | 0 | 0 | 0 | 0 | 0 | D | 0 | 0 | 0 | 0 |
| D | 1 | 0 | 1 | 0 | 0 | D | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 0 | 0 | D | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | D | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



# DECODERS.

A decoder is a combinational ckt device that convert an n-bit binary input code (data) into $2^n$ - output lines. "Decoder is similar to DE-MULTIPLEXER" but only the difference is that a DE-MUX has one input line but a decoder has not any data input line. A decoder converts an n-bit binary code into $2^n$ - output lines so that each of $2^n$ output lines is activated only one of the possible combination of inputs. No. of output lines is greater than number of input lines. If the number of input and output lines are equal then it is called converter.

**Basic Binary Decoder** :- An AND gate can be used as decoder element. This can be understood in betterway as that Suppose we want to decode a binary code '1001' by a logic ckt. If we want to decode this binary code by using an AND gate, then the middle two bit should be inverted as HIGH because an AND gate gives a HIGH when all of its inputs are at HIGH.

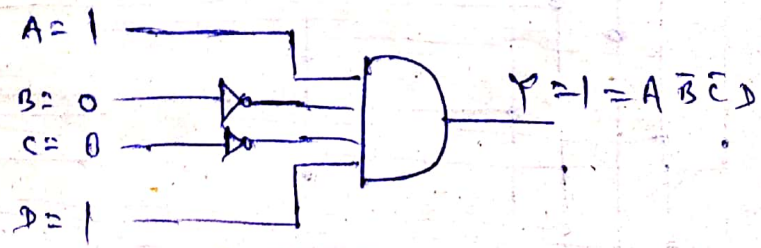



fig-(1)

Let us consider the fig. 1 where 4-input AND gate has input variables as ABCD = 1001. Input line 'B and C' are connected to two inverters. When we supply 1001, Y = 1 but other than this binary code 1001, ~~other~~ Output of this logic ckt is LOW or '0'. Hence we can say that this logic circuit decode only 1001.

## 3-to-8 Decoder

As the name is 3-to-8 decoder, it has three inputs (A, B, C) and eight outputs lines ($D_0$ to $D_7$ or $Y_0$ to $Y_7$). Based on the combination of three inputs A, B, C, one of 8 output line is activated. This can be understood in better way by following ~~the~~ truth table —

| Inputs | | | Outputs | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

from the above ~~truth~~ truth table we can write boolean expression for each activated output logic as below ——

$D_0 = \bar{A}\bar{B}\bar{C}$, $D_1 = \bar{A}\bar{B}C$, $D_2 = \bar{A}B\bar{C}$, $D_3 = \bar{A}BC$

$D_4 = A\bar{B}\bar{C}$, $D_5 = A\bar{B}C$, $D_6 = AB\bar{C}$, $D_7 = ABC$.

The above all expression can be implemented by a logic ckt using basic gates as below ——