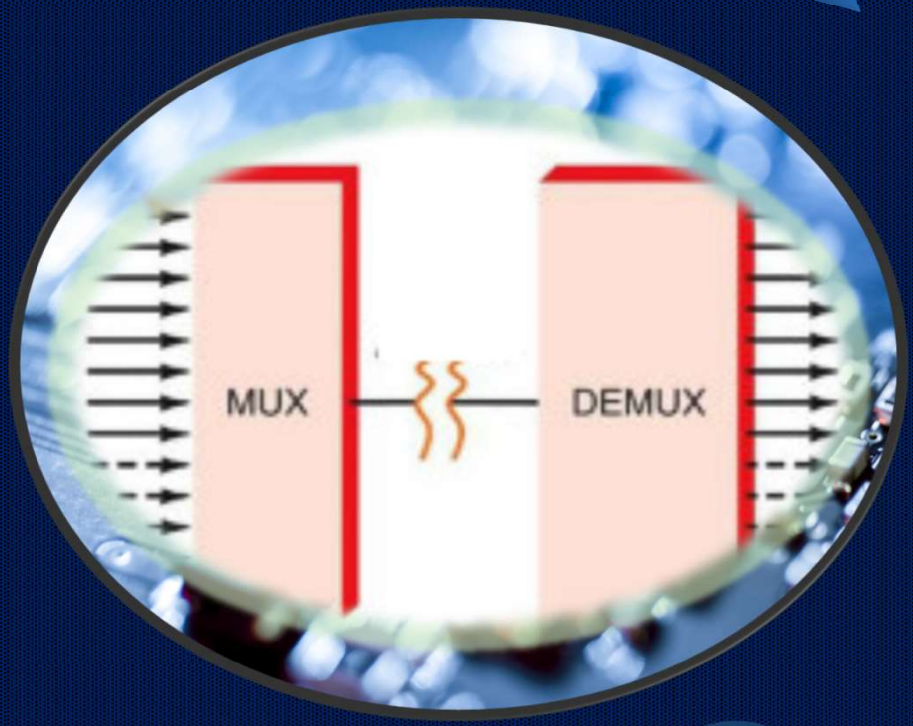
Multiplexers & Demultiplexers

Deepa Mathews

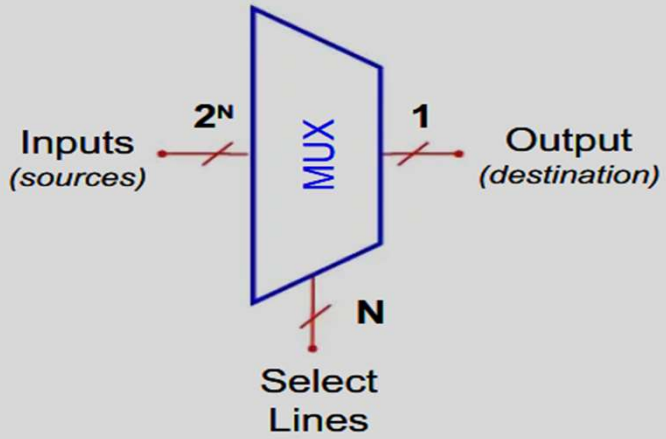


MULTIPLEXER

- **Multiplex means Many-to-One**
- Multiplexing is the process of transmitting a large number of information over a single line.
- A **digital multiplexer (MUX)** is a combinational circuit that **selects one** digital information from several sources and transmits the selected information on a **single output line**.
- A multiplexer (MUX) acts like a digitally controlled multi-position switch that **has multiple inputs and a single line output**. The select inputs (address inputs) determine which input is connected to the output.

MULTIPLEXER

- It is also called a **data selector** as it selects one of many inputs and steers the binary information to the output line. Normally, there are 2^n input lines and n selection lines.
- **A m-to-1 Multiplexer has**
  - m (ie., 2^n) inputs, one Output,
  - n Control (Select) inputs,
  - One (or more) Enable input(s)

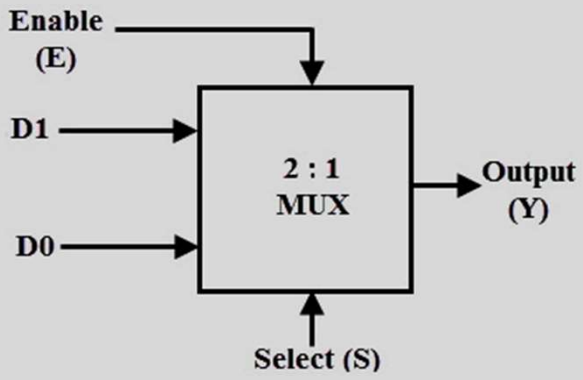


Example: 2-to-1 (1 select line) , 4-to-1 (2 select lines)

8-to-1 (3 select lines), 16-to-1 (4 select lines)

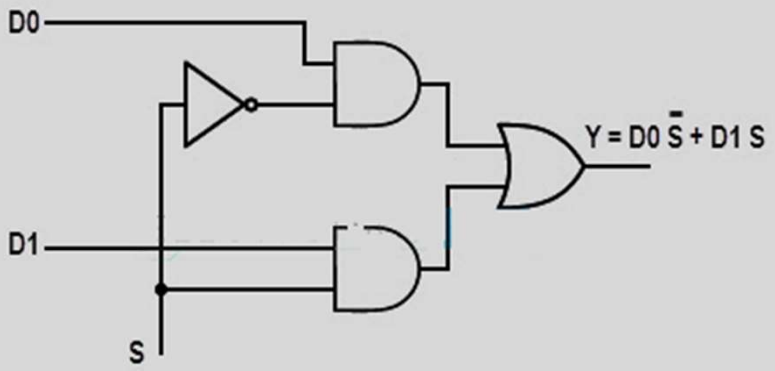
2-TO-1 MULTIPLEXER

- A **2-to-1 multiplexer** consists of two inputs D0 and D1, one select input S and one output Y.
- Depends on the select signal, the output is connected to either of the inputs.
- If the select line is low, then the output will be switched to D0 input, whereas if select line is high, then the output will be switched to D1 input.



S	Y
0	D0
1	D1

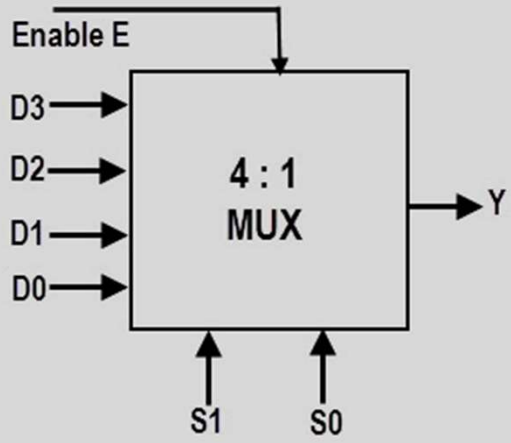
$$Y = D_0 \bar{S} + D_1 S$$



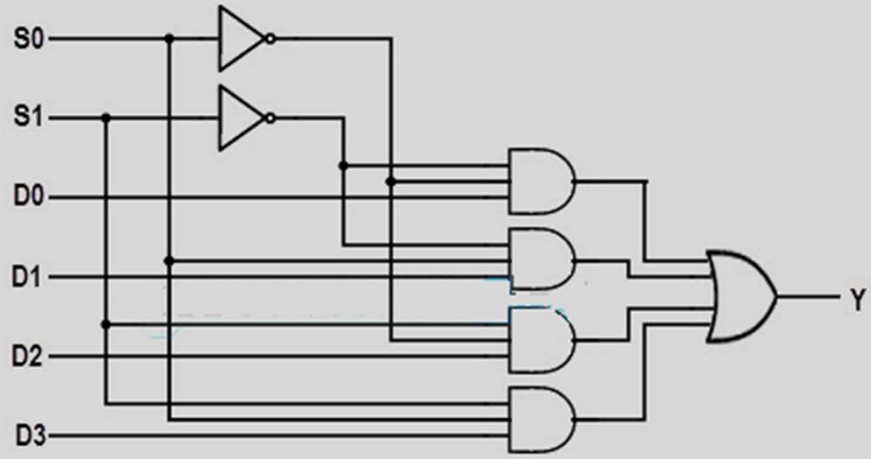
4-TO-1 MULTIPLEXER

4:1 Multiplexer has **four data inputs** D_3, D_2, D_1 & D_0, **two selection lines** S_1 & S_0 and **one output** Y.

One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines



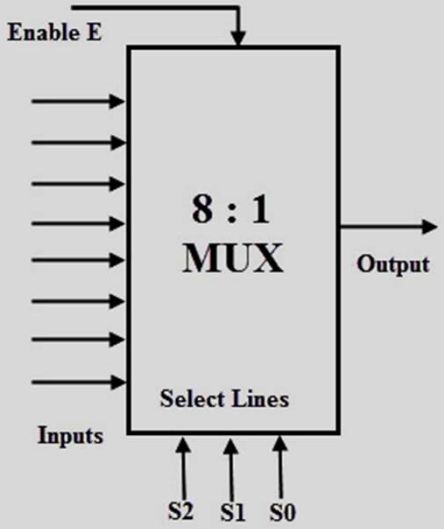
Select Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



$$Y = S_1' S_0' D_0 + S_1' S_0 D_1 + S_1 S_0' D_2 + S_1 S_0 D_3$$

8-TO-1 MULTIPLEXER

It consists of **eight data inputs** D_0 through D_7**, three selection lines** S_2, S_1 & S_0 and **one output** Y. Depending on the select lines combinations, multiplexer decodes the inputs.

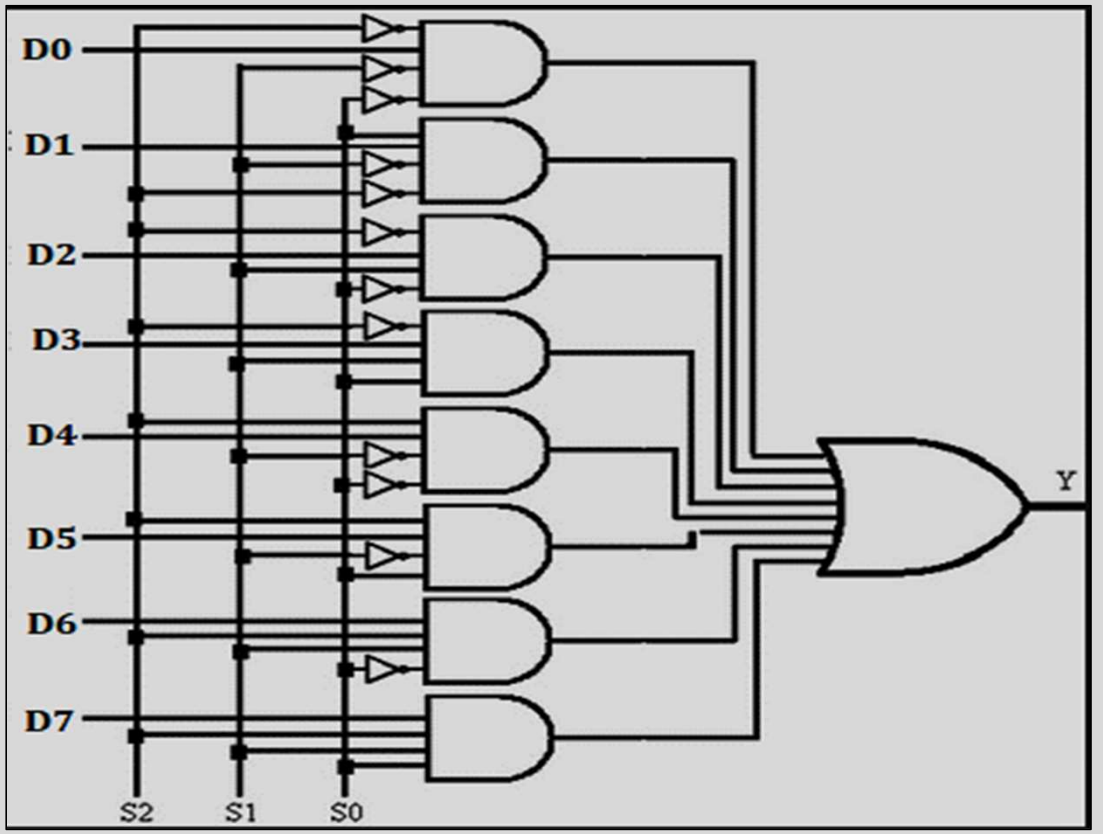


Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

$$Y=S_0'S_1'S_2'D_0+S_0S_1'S_2'D_1+S_0'S_1S_2'D_2+ S_0S_1S_2'D_3+S_0'S_1'S_2D_4+S_0S_1'S_2D_5+ S_0'S_1S_2\,D_6+S_0S_1S_3D_7$$

8-TO-1 MULTIPLEXER

$$Y=S_0'S_1'S_2'D_0+S_0S_1'S_2'D_1+S_0'S_1S_2'D_2+$$
$$S_0S_1S_2'D_3+S_0'S_1'S_2D_4+S_0S_1'S_2D_5+$$
$$S_0'S_1S_2\,D_6+S_0S_1S_3D_7$$



APPLICATION OF MULTIPLEXER

- **Data Selection**
- **Data Routing**
- **Operation Sequencing**
- **Parallel – to – Serial Conversion**
- **Waveform Generation**
- **Logic Function Generation**

LOGIC FUNCTION GENERATOR

A multiplexer can be used in place of logic gates to implement a logic expression.

Design of a function generator Using a Multiplexer

Type 1 : The Type of mux can be decided by the given number of variables.

1. The first step is to select the multiplexer. If the given expression has n variables, then determine the multiplexer using the formula 2^n:1
2. Connect the inputs, that correspond to the given minterms to logic 1.
3. Connect all the other inputs to the ground(logic 0).
4. Connect the input variables as the selection lines.

IMPLEMENT THE SOP EXPRESSION USING MUX EXERCISES

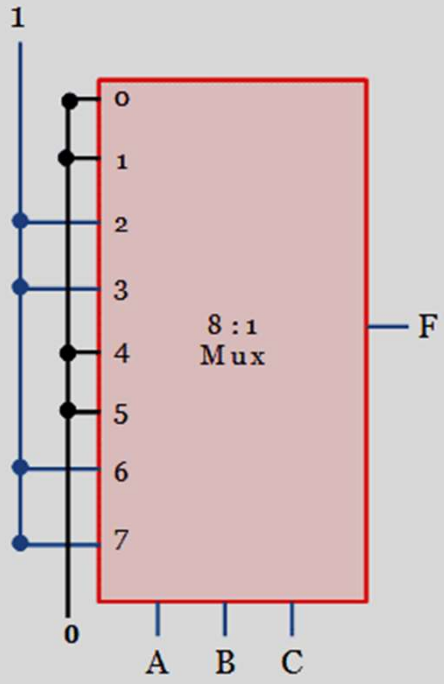
Implement the boolean expression F(A, B, C) = ∑ m(2, 3, 6, 7) using a multiplexer.

There are 3 variables in given expression, hence $2^n = 2^3 =$ **8 : 1 multiplexer**.

So, the mux has **8 input lines, 3 selection lines, and one output**.

The inputs, corresponding to the **minterms (2, 3, 6, 7)** are connected to the **logic 1** and remaining terms to the logic 0(grounded).

The **given input variables are connected as three selection lines.**



IMPLEMENT THE SOP EXPRESSION USING MUX EXERCISES

Type 2 : Boolean expression having n variables can be implemented with 2^{n-1} to 1 multiplexer.

From the given set of n+1 input variables, the n least significant variables are used as selection line inputs.

Implement a 4-variable (A,B,C&D) logic function using a multiplexer with 3 data select inputs.

- Construct the truth table for the function. In the truth table, ABC has the same value twice once with D=0 and with D=1.
  - When the same combination ABC occurs and if F =0 both times, connect logic 0 to the data input selected by that combination; If F=1, connect logic 1 to the data input.
  - If F is different for the 2 occurrences of a combination of ABC, and if F=D in each case, connect D to the data input selected by that combination; if F=D' in each case, connect D' to the data input selected by that combination.

IMPLEMENT THE SOP EXPRESSION USING MUX EXERCISES

Implement the SOP function F(A, B, C) = A XOR B XOR C using 4 : 1 multiplexer.

- In the given boolean expression, there are 3 variables. We should use $2^3 : 1 = 8 : 1$ multiplexer. But as per the question, it is to be implemented with 4 : 1 mux.

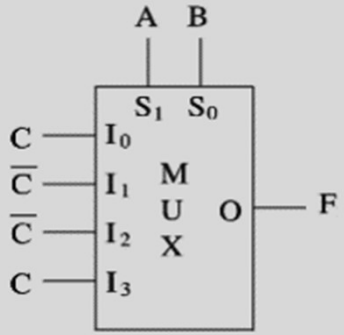
- If F is different for the 2 occurrences of a combination of AB, and
  - if F=C in each case, connect C to the data input selected by that combination.
  - if F=C' in each case, connect C' to the data input selected by that combination.

Original truth table

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

New truth table

A	B	F
0	0	C
0	1	$\overline{C}$
1	0	$\overline{C}$
1	1	C

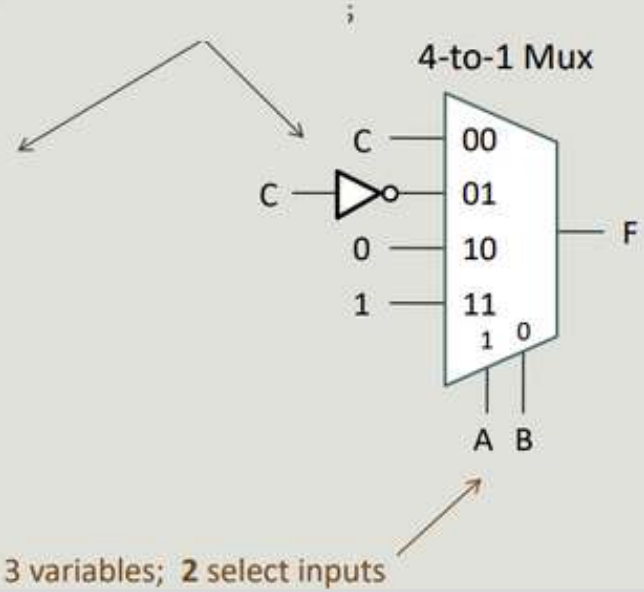


IMPLEMENT THE SOP EXPRESSION USING MUX EXERCISES

Implement F(A, B, C) = ∑ m(1, 2, 6, 7) using 4 : 1 multiplexer.

F(A, B, C) = Σ m(1, 2, 6, 7)

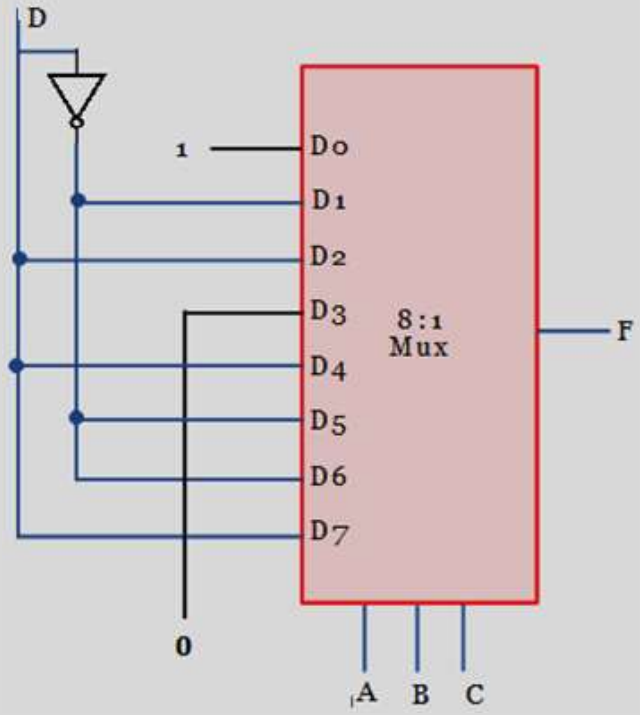
row	A	B	C	F(A,B,C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1



3 variables; **2** select inputs

IMPLEMENT THE SOP EXPRESSION USING MUX EXERCISES

Implement F(A, B, C, D) = ∑ m(0, 1, 2, 5, 9, 10, 12, 15) using 8 : 1 multiplexer.



EXERCISES:

1. **Implement the following Boolean function with a 8:1 multiplexer**

   $F = \sum(1,3,4,11,12,13,14,15)$

2. **Implement the following Boolean function with a 8:1 multiplexer**

   (a) F(A,B,C,D) = Sum(0,2,5,7,11,14)

   (b) F(A,B,C,D) =product(3,8,12,15)

DEMULTIPLEXER (DATA DISTRIBUTOR)

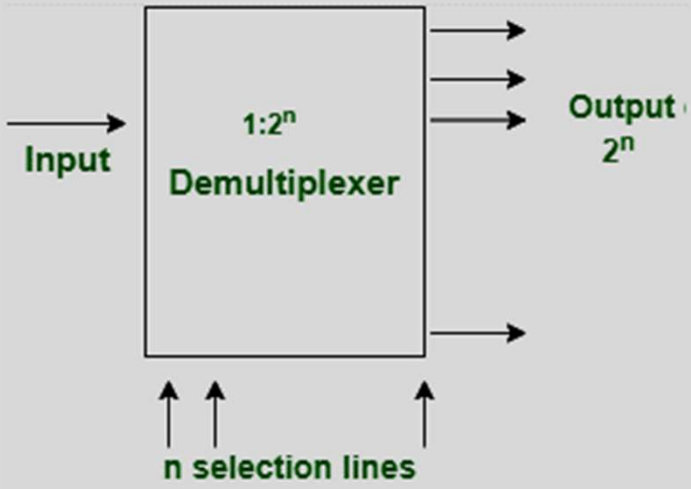
Reverse of multiplexer

Demultiplexer is a data distributor which takes a single input and gives several outputs as it transmits the same data to different destinations.

In demultiplexer we have 1 input and 2^n output lines where n is the selection line.

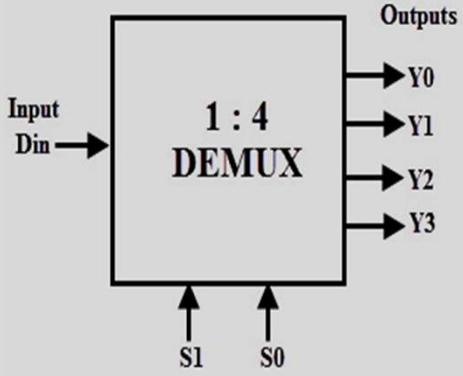
Types

- **1-to-2 (1 Select Lines)**
- **1-to-4 (2 Select Lines)**
- **1-to-8 (3 Select Lines)**
- **1-to-16 (4 Select Lines)**



1-TO-4 DEMULTIPLEXER

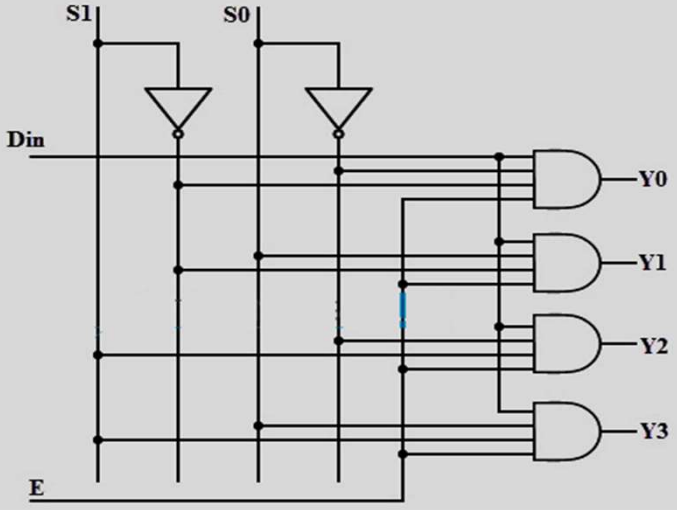
- A 1-to-4 demultiplexer has a single input (D), two selection lines (S1 and S0) and four outputs (Y0 to Y3). The input data goes to any one of the four outputs at a given time for a particular combination of select lines.
- This demultiplexer is also called as a **2-to-4 demultiplexer** which means that two select lines and 4 output lines.



Data Input	Select Inputs		Outputs			
D	S_1	S_0	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

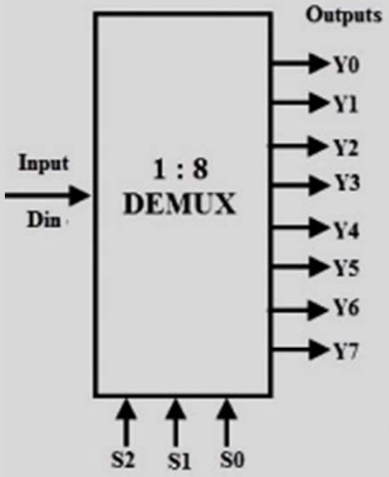
$Y0 = \overline{S1}\ \overline{S0}\ D$ $Y1 = \overline{S1}\ S0\ D$

$Y2 = S1\ \overline{S0}\ D$ $Y3 = S1\ S0\ D$



1-TO-8 DEMULTIPLEXER

- It consists of single input D, 3 select inputs S2, S1 and S0 and 8 outputs from Y0 to Y7.
- It is also called as 3-to-8 demultiplexer due to three select input lines. It distributes one input line to one of 8 output lines depending on the combination of select inputs.



Data Input	Select Inputs			Outputs							
D	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

$Y0 = D\,\overline{S2}\,\overline{S1}\,\overline{S0}$

$Y1 = D\,\overline{S2}\,\overline{S1}\,S0$

$Y2 = D\,\overline{S2}\,S1\,\overline{S0}$

$Y3 = D\,\overline{S2}\,S1\,S0$

$Y4 = D\,S2\,\overline{S1}\,\overline{S0}$

$Y5 = D\,S2\,\overline{S1}\,S0$

$Y6 = D\,S2\,S1\,\overline{S0}$

$Y7 = D\,S2\,S1\,S0$

1-TO-8 DEMULTIPLEXER

$Y0 = D\,\overline{S2}\,\overline{S1}\,\overline{S0}$

$Y1 = D\,\overline{S2}\,\overline{S1}\,S0$

$Y2 = D\,\overline{S2}\,S1\,\overline{S0}$

$Y3 = D\,\overline{S2}\,S1\,S0$

$Y4 = D\,S2\,\overline{S1}\,\overline{S0}$

$Y5 = D\,S2\,\overline{S1}\,S0$

$Y6 = D\,S2\,S1\,\overline{S0}$

$Y7 = D\,S2\,S1\,S0$