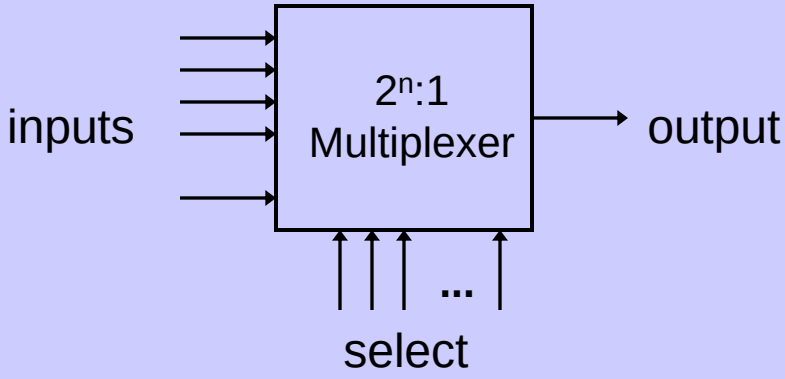
Digital Systems
18B11EC213

Module 1: Boolean Function Minimization Techniques and Combinational Circuits-13

Dr. Saurabh Chaturvedi

Multiplexer

- A multiplexer is a device which has
  (i) a number of *input* lines
  (ii) a number of *selection* lines
  (iii) one *output* line
- It steers one of 2^n inputs to a single output line, using n selection lines. Also known as a *data selector*.

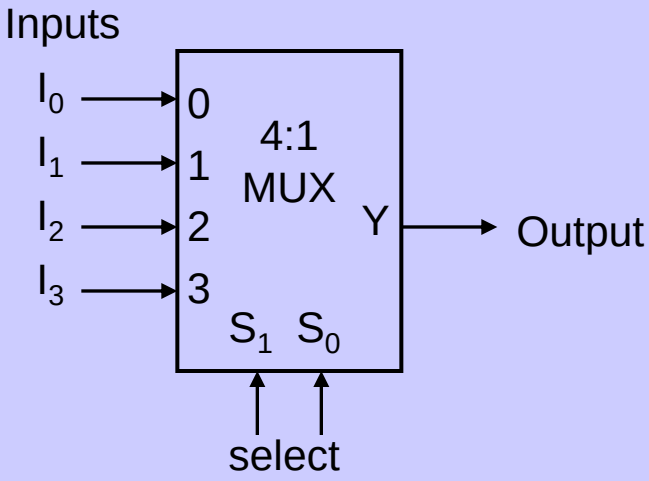


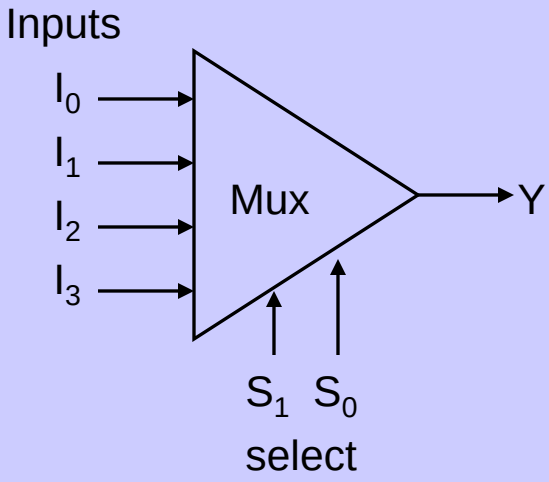
Multiplexer

- Truth table for a 4-to-1 (4:1) multiplexer:

I_0	I_1	I_2	I_3	S_1	S_0	Y
d_0	d_1	d_2	d_3	0	0	d_0
d_0	d_1	d_2	d_3	0	1	d_1
d_0	d_1	d_2	d_3	1	0	d_2
d_0	d_1	d_2	d_3	1	1	d_3

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



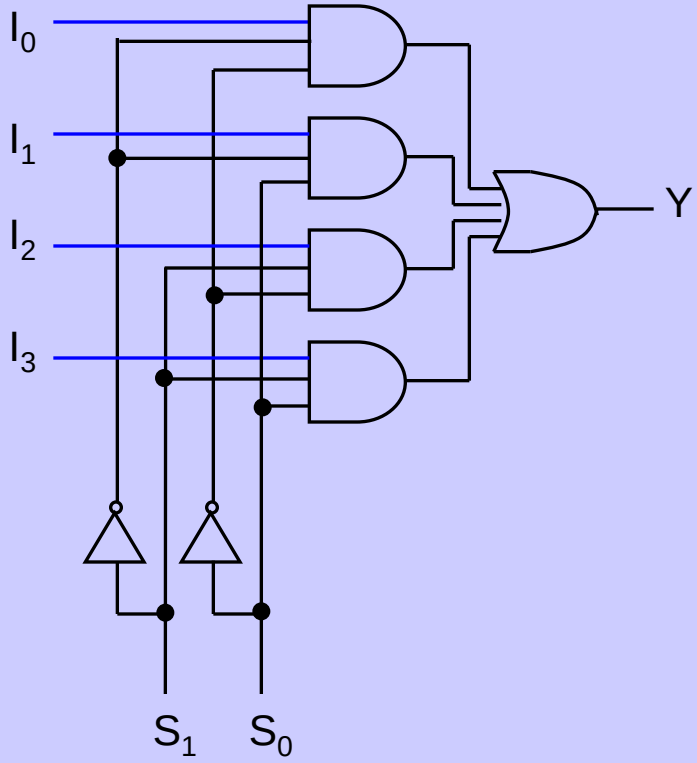


$$Y = I_0.(S_1'.S_0') + I_1.(S_1'.S_0) + I_2.(S_1.S_0') + I_3.(S_1.S_0)$$

Multiplexer

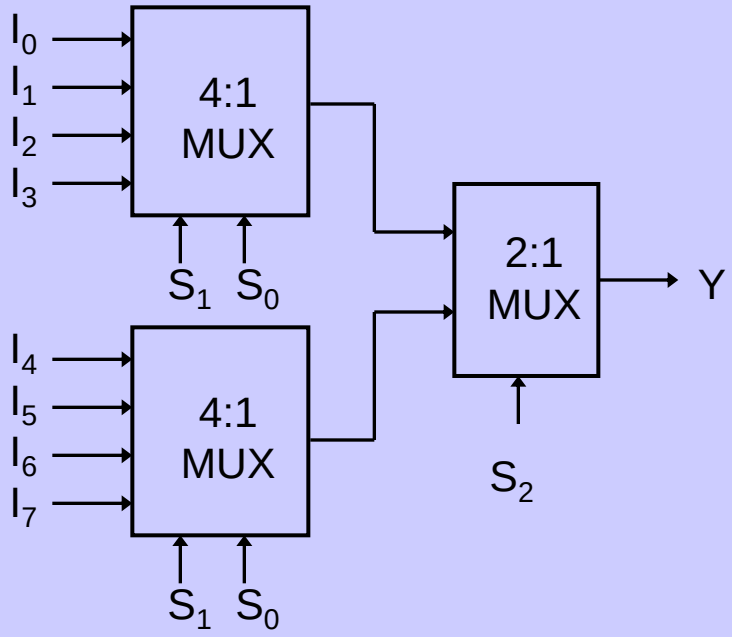
$$Y = I_0.(S_1'.S_0') + I_1.(S_1'.S_0) + I_2.(S_1.S_0') + I_3.(S_1.S_0)$$

Circuit diagram of a 4:1 mux:



Larger Multiplexers

- Larger multiplexers can be constructed from smaller ones.
- For example, as shown below, an 8:1 multiplexer can be constructed using two 4:1 multiplexers and one 2:1 multiplexer:

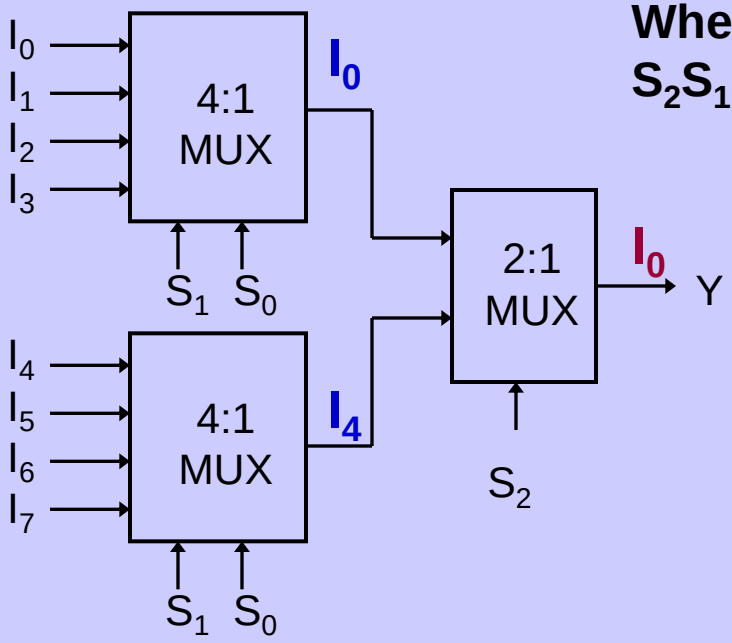


Truth table of 8:1 mux:

S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Note the placement of selection lines.

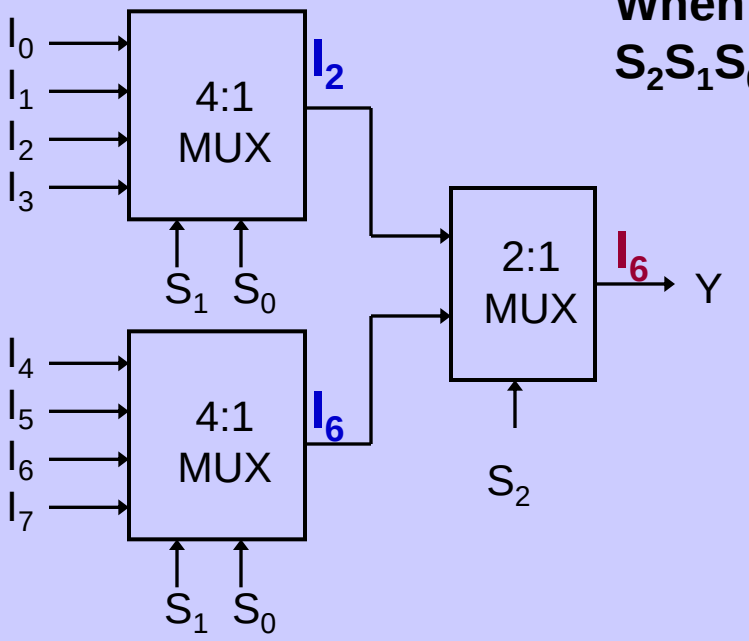
Larger Multiplexers



When
$S_2S_1S_0$ = 000

S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Larger Multiplexers

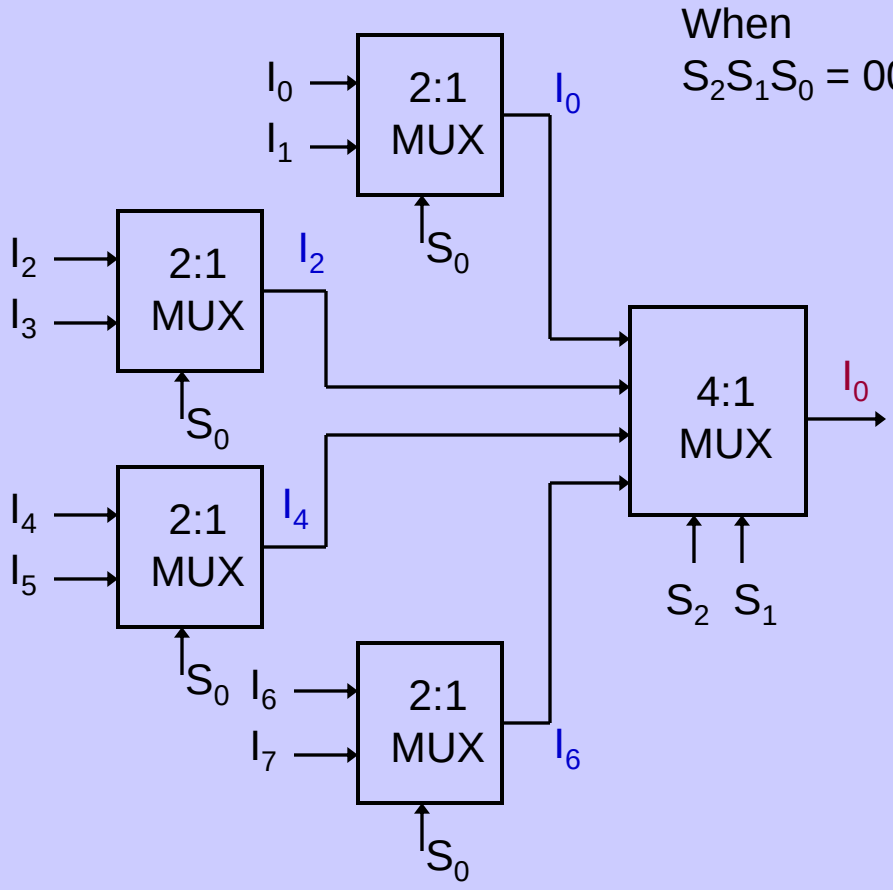


When
$S_2S_1S_0 = 110$

S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Larger Multiplexers

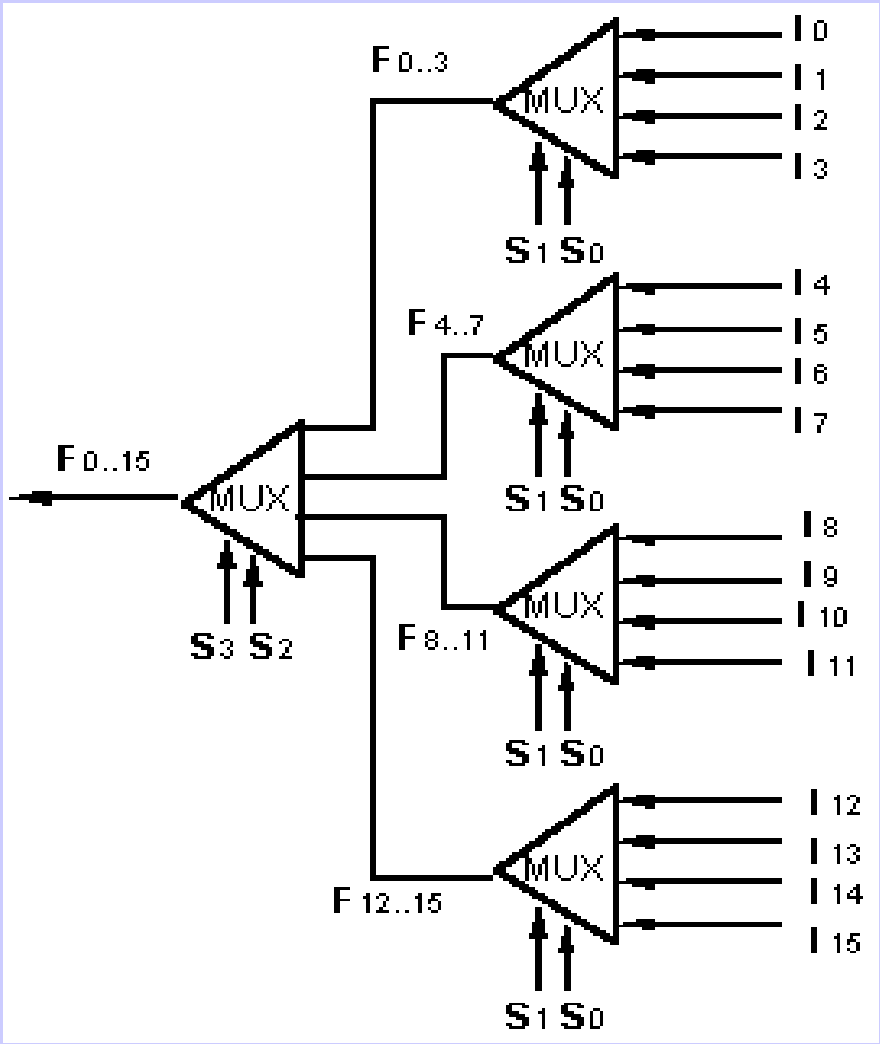
- Another implementation of an 8:1 multiplexer using smaller multiplexers – Using four 2:1 muxs and one 4:1 mux:



S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Larger Multiplexers

- A 16:1 multiplexer can be constructed using five 4:1 multiplexers, as shown in the figure:



Multiplexers: Implementing Functions

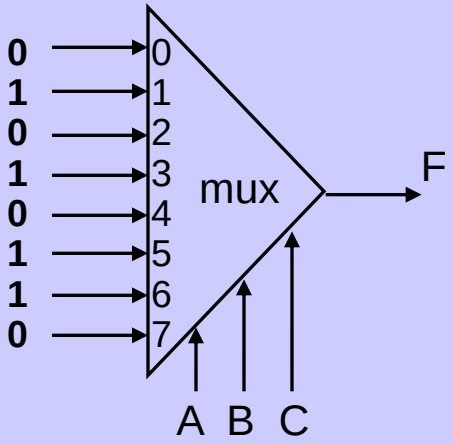
- A Boolean function can be implemented using multiplexers.
- A 2^n-to-1 multiplexer can implement a Boolean function of *n* input variables as follows:
  - (i) Express in sum-of-minterms form.

    Example: F(A,B,C) = A'B'C + A'BC + AB'C + ABC'

    = Σ m(1,3,5,6)
  - (ii) Connect *n* variables to the *n* selection lines.
  - (iii) Put a '1' on a data line if it is a minterm of the function, '0' otherwise.

Multiplexers: Implementing Functions

Implementation of F(A,B,C) = Σ m(1,3,5,6) using an 8:1 multiplexer.



This method works because:

Output = $m_0.I_0 + m_1.I_1 + m_2.I_2 + m_3.I_3 + m_4.I_4 + m_5.I_5 + m_6.I_6 + m_7.I_7$

Supplying '1' to I_1, I_3, I_5, I_6 , and '0' to the rest:

Output = $m_1 + m_3 + m_5 + m_6$

Using Smaller Multiplexers

- Earlier, we saw how a 2^n-to-1 multiplexer can be used to implement any Boolean function of *n* (input) variables.
- However, we can use a single smaller $2^{(n-1)}$-to-1 multiplexer to implement any Boolean function of *n* (input) variables.
- In particular, the earlier function

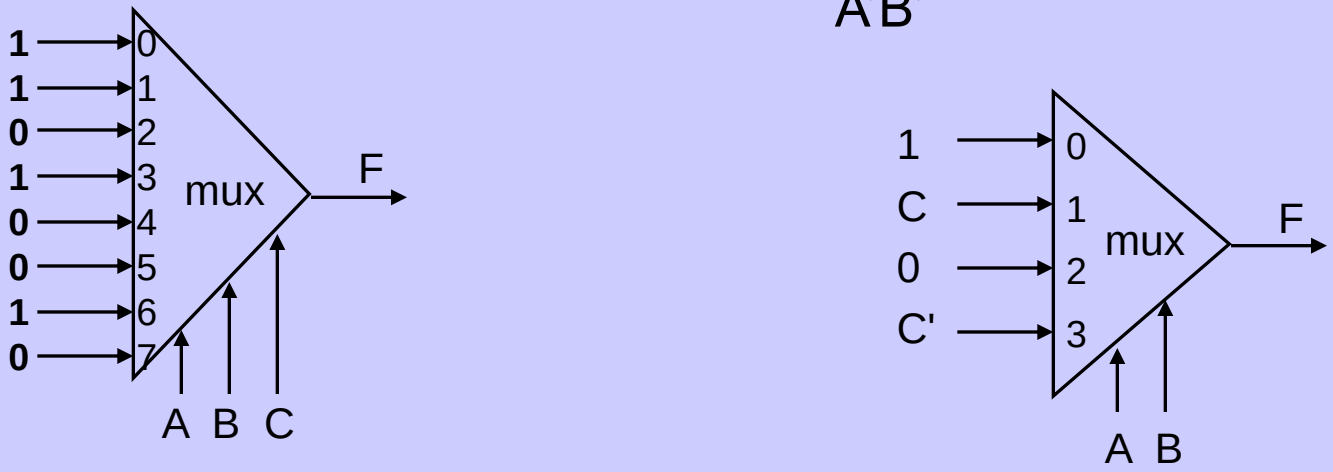
$$F(A,B,C) = \sum m(1,3,5,6)$$

can be implemented using a 4-to-1 multiplexer (rather than an 8-to-1 multiplexer).

Using Smaller Multiplexers

Example:

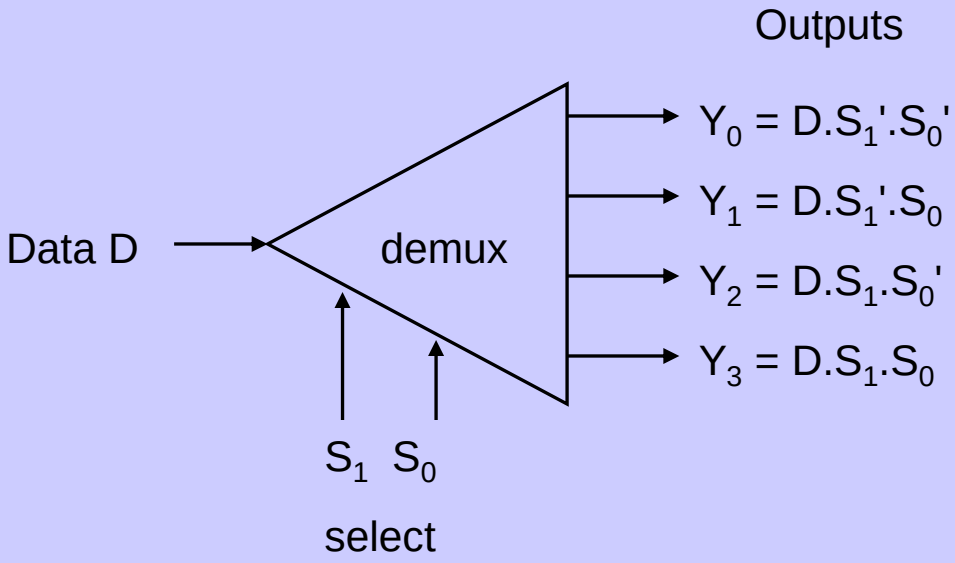
$$F(A,B,C) = \Sigma\, m(0,1,3,6) = \underbrace{A'B'C' + A'B'C}_{A'B'} + A'BC + ABC'$$



- Two of the variables (A and B) are applied as the selection inputs of the 4:1 multiplexer, while the data inputs of the multiplexer contain 1, C, 0 and C'.

Demultiplexer

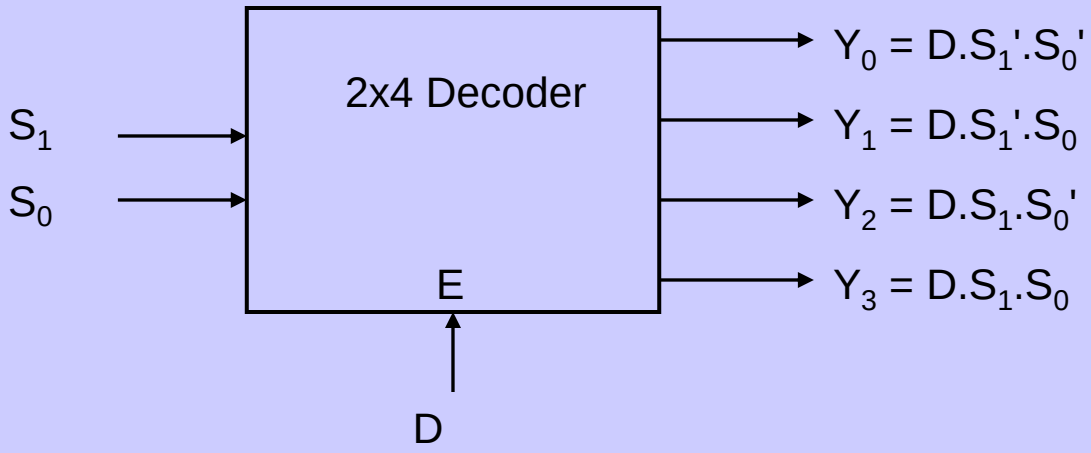
- Given an input line and a set of selection lines, a demultiplexer directs data from input to a selected output line.
- An example of a 1-to-4 (1:4) demultiplexer:



S_1	S_0	Y_0	Y_1	Y_2	Y_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

Demultiplexer

- A demultiplexer is actually identical to a decoder with an enable signal, as illustrated below:



References

- M. M. Mano, *Digital Logic and Computer Design*, 5th ed., Pearson Prentice Hall, 2013.
- R. P. Jain, *Modern Digital Electronics*, 4th ed., Tata McGraw-Hill Education, 2009.