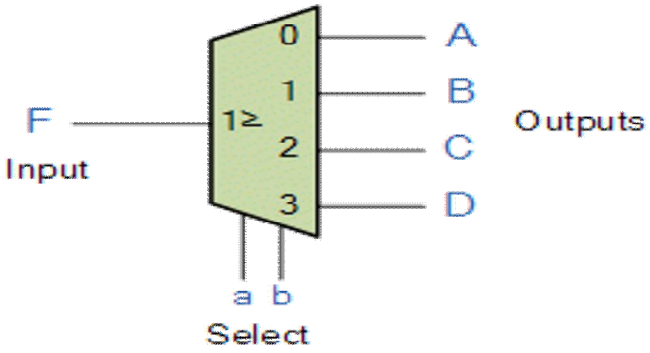
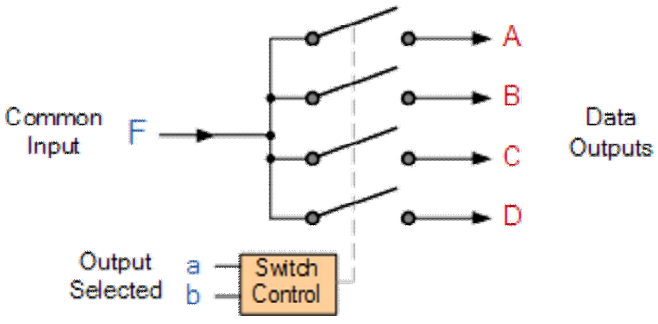
Demultiplexer

De-Multiplexer is a combinational circuit that performs the reverse operation of Multiplexer. It has single input, ‘n’ selection lines and maximum of 2n outputs. The input will be connected to one of these outputs based on the values of selection lines.

Since there are ‘n’ selection lines, there will be 2n possible combinations of zeros and ones. So, each combination can select only one output. De-Multiplexer is also called as De-Mux.



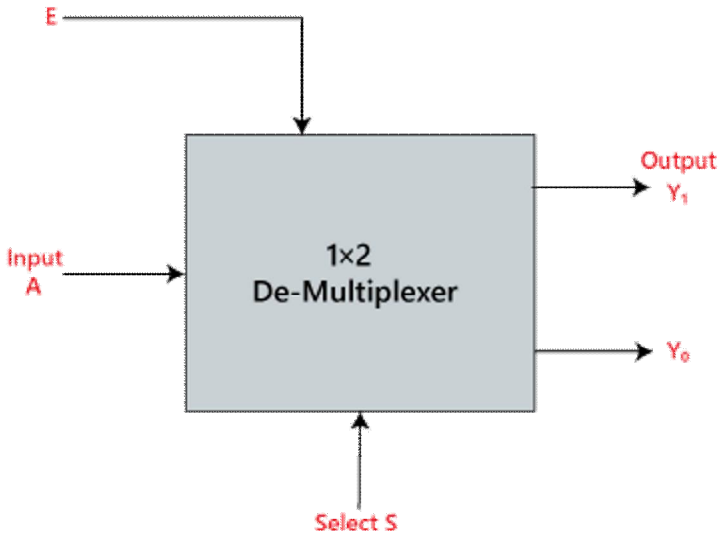


Output Select		Data Output Selected
a	b	
0	0	A
0	1	B
1	0	C
1	1	D

1×2 De-multiplexer:

In the 1 to 2 De-multiplexer, there are only two outputs, i.e., Y_0, and Y_1, 1 selection lines, i.e., S_0, and single input, i.e., A. On the basis of the selection value, the input will be connected to one of the outputs. The block diagram and the truth table of the 1×2 multiplexer are given below.

Block Diagram:



Truth Table:

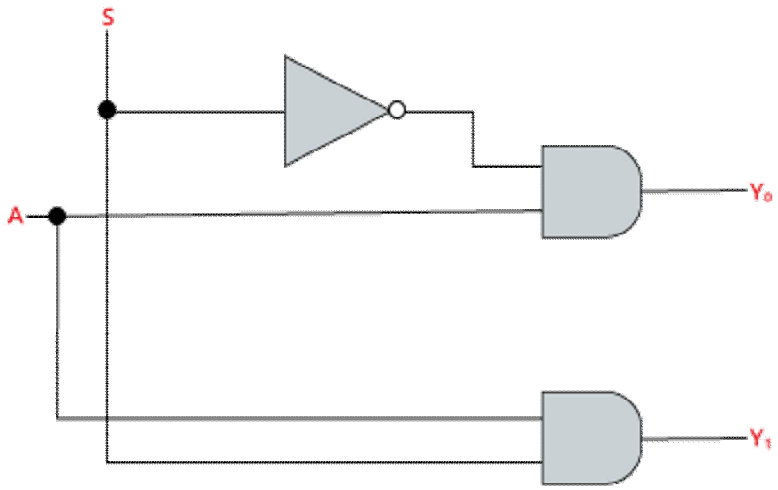
INPUTS	Output	
S_0	Y_1	Y_0
0	0	A
1	A	0

The logical expression of the term Y is as follows:

$Y_0 = S_0'.A$

$Y_1 = S_0.A$

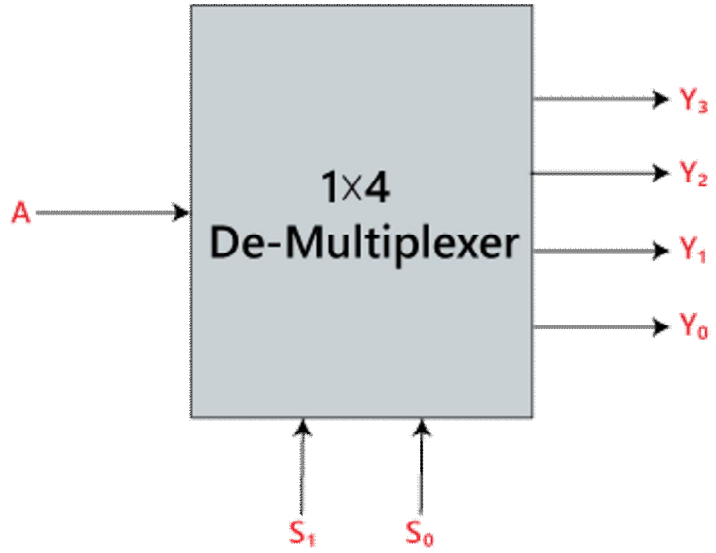
Logical circuit of the above expressions is given below:



1×4 De-multiplexer:

In 1 to 4 De-multiplexer, there are total of four outputs, i.e., Y_0, Y_1, Y_2, and Y_3, 2 selection lines, i.e., S_0 and S_1 and single input, i.e., A. On the basis of the combination of inputs which are present at the selection lines S_0 and S_1, the input be connected to one of the outputs. The block diagram and the truth table of the 1× 4 multiplexer are given below.

Block Diagram:



Truth Table:

INPUTS		Output			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	A
0	1	0	0	A	0
1	0	0	A	0	0
1	1	A	0	0	0

The logical expression of the term Y is as follows:

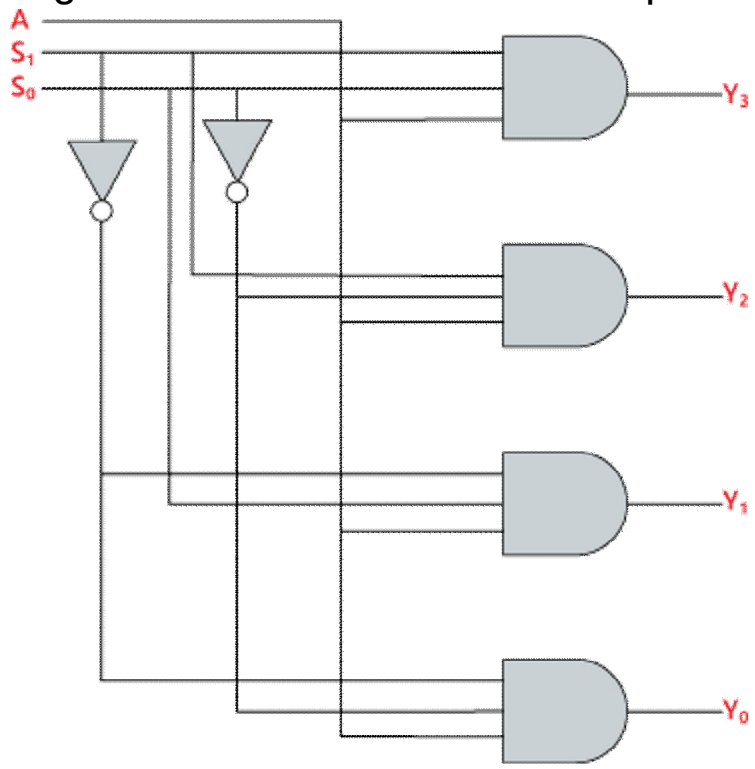
$Y_0 = S_1' S_0' A$

$y_1 = S_1' S_0 A$

$y_2 = S_1 S_0' A$

$y_3 = S_1 S_0 A$

Logical circuit of the above expressions is given below:

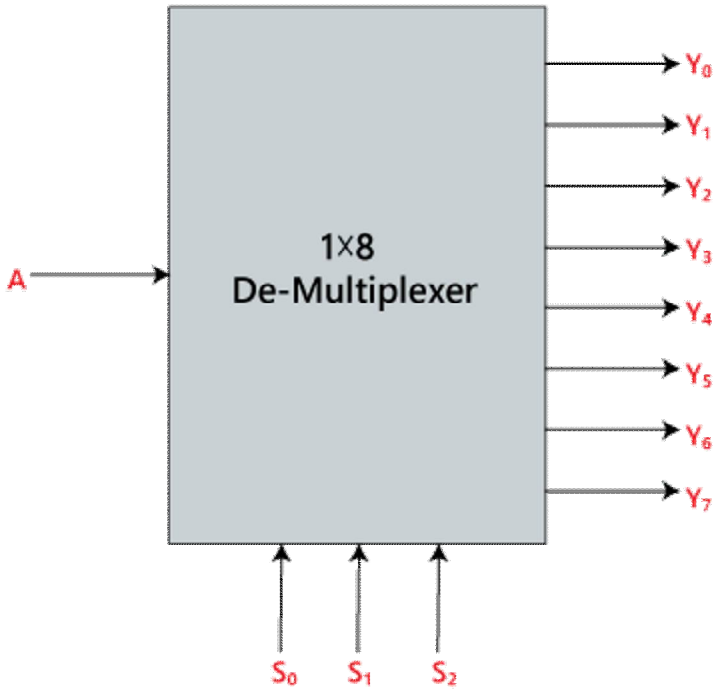


1×8 De-multiplexer

In 1 to 8 De-multiplexer, there are total of eight outputs, i.e., Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, and Y_7, 3 selection lines, i.e., S_0, $S_1$and S_2 and single input, i.e., A. On the basis of the combination of inputs which are present at the selection lines S^0, S^1 and S_2, the input will be connected to one of these outputs. The block diagram and the truth

table of the 1×8 de-multiplexer are given below.

Block Diagram:



Truth Table:

INPUTS			Output							
S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	A
0	0	1	0	0	0	0	0	0	A	0
0	1	0	0	0	0	0	0	A	0	0
0	1	1	0	0	0	0	A	0	0	0
1	0	0	0	0	0	A	0	0	0	0
1	0	1	0	0	A	0	0	0	0	0
1	1	0	0	A	0	0	0	0	0	0
1	1	1	A	0	0	0	0	0	0	0

The logical expression of the term Y is as follows:

$Y_0 = S_0'.S_1'.S_2'.A$

$Y_1 = S_0.S_1'.S_2'.A$

$Y_2 = S_0'.S_1.S_2'.A$

$Y_3 = S_0.S_1.S_2'.A$

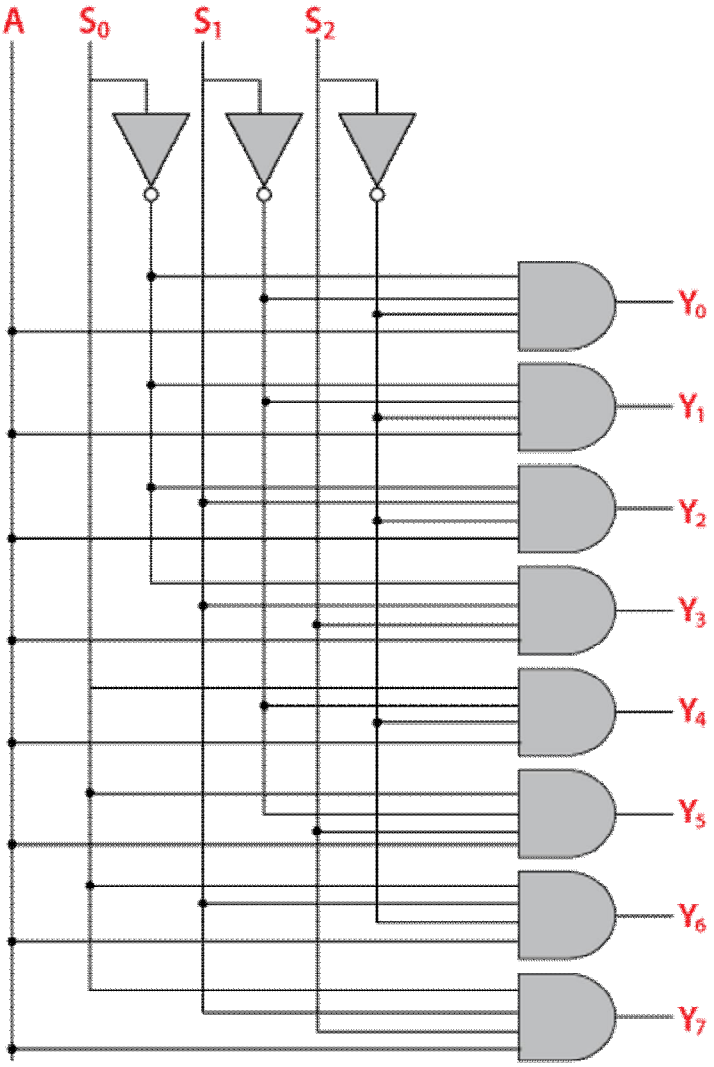
$Y_4 = S_0'.S_1'.S_2\ A$

$Y_5 = S_0.S_1'.S_2\ A$

$Y_6 = S_0'.S_1.S_2\ A$

$Y_7 = S_0.S_1.S_3.A$

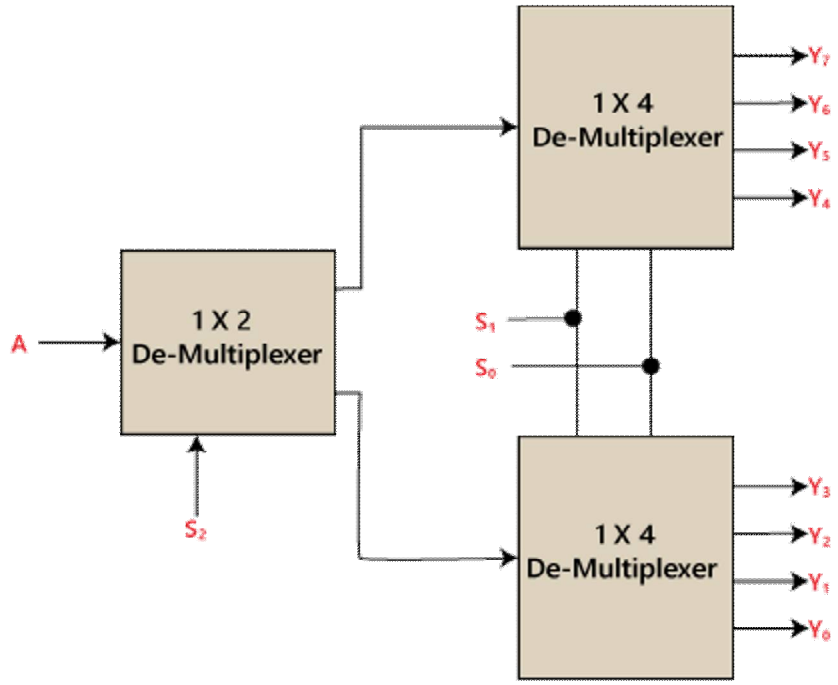
Logical circuit of the above expressions is given below:



1×8 De-multiplexer using 1×4 and 1×2 de-multiplexer

We can implement the 1×8 de-multiplexer using a lower order de-multiplexer. To implement the 1×8 de-multiplexer, we need two 1×4 de-multiplexer and one 1×2 de-multiplexer. The 1×4 multiplexer has 2 selection lines, 4 outputs, and 1 input. The 1×2 de-multiplexer has only 1 selection line.

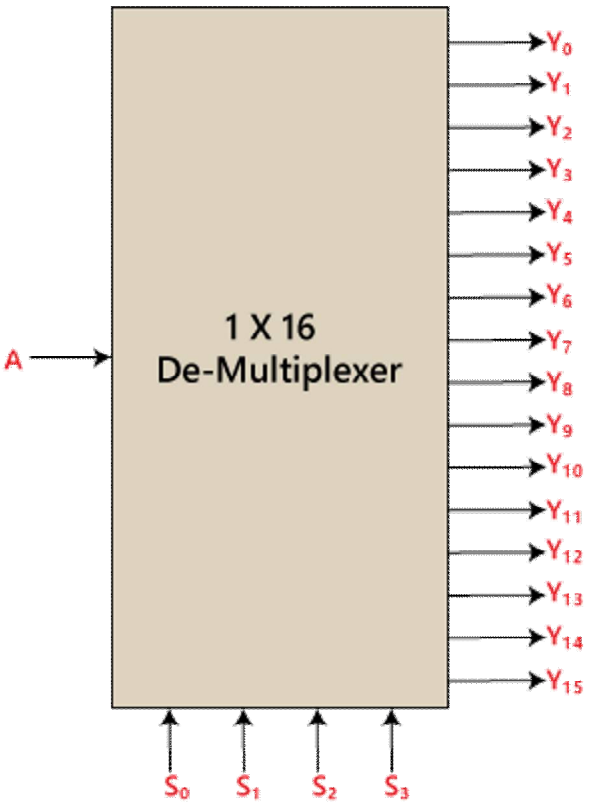
For getting 8 data outputs, we need two 1×4 de-multiplexer. The 1×2 de-multiplexer produces two outputs. So, in order to get the final output, we have to pass the outputs of 1×2 de-multiplexer as an input of both the 1×4 de-multiplexer. The block diagram of 1×8 de-multiplexer using 1×4 and 1×2 de-multiplexer is given below.



1 x 16 De-multiplexer

In 1×16 de-multiplexer, there are total of 16 outputs, i.e., Y_0, Y_1, ..., Y_{16}, 4 selection lines, i.e., S_0, S_1, S_2, and S_3 and single input, i.e., A. On the basis of the combination of inputs which are present at the selection lines S^0, S^1, and S_2, the input will be connected to one of these outputs. The block diagram and the truth table of the 1×16 de-multiplexer are given below.

Block Diagram:



Truth Table:

INPUTS				OUTPUTS															
S_3	S_2	S_1	S_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

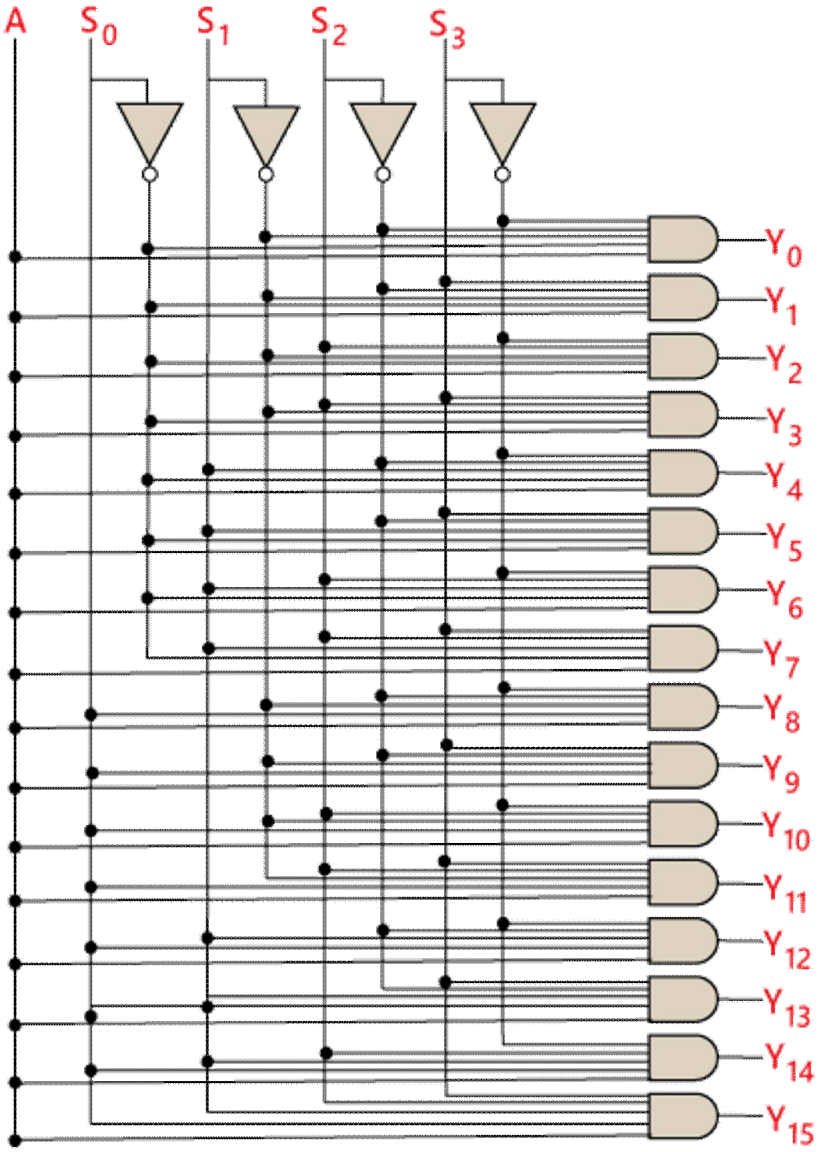
The logical expression of the term Y is as follows:

$Y_0 = A.S_0'.S_1'.S_2'.S_3'$

$Y_1 = A.S_0'.S_1'.S_2'.S_3$

$Y_2 = A.S_0'.S_1'.S_2.S_3'$
$Y_3 = A.S_0'.S_1'.S_2.S_3$
$Y_4 = A.S_0'.S_1.S_2'.S_3'$
$Y_5 = A.S_0'.S_1.S_2'.S_3$
$Y_6 = A.S_0'.S_1.S_2.S_3'$
$Y_7 = A.S_0'.S_1.S_2.S_3$
$Y_8 = A.S_0.S_1'.S_2'.S_3'$
$Y_9 = A.S_0.S_1'.S_2'.S_3$
$Y_{10} = A.S_0.S_1'.S_2.S_3'$
$Y_{11} = A.S_0.S_1'.S_2.S_3$
$Y_{12} = A.S_0.S_1.S_2'.S_3'$
$Y_{13} = A.S_0.S_1.S_2'.S_3$
$Y_{14} = A.S_0.S_1.S_2.S_3'$
$Y_{15} = A.S_0.S_1.S_2'.S_3$

Logical circuit of the above expressions is given below:

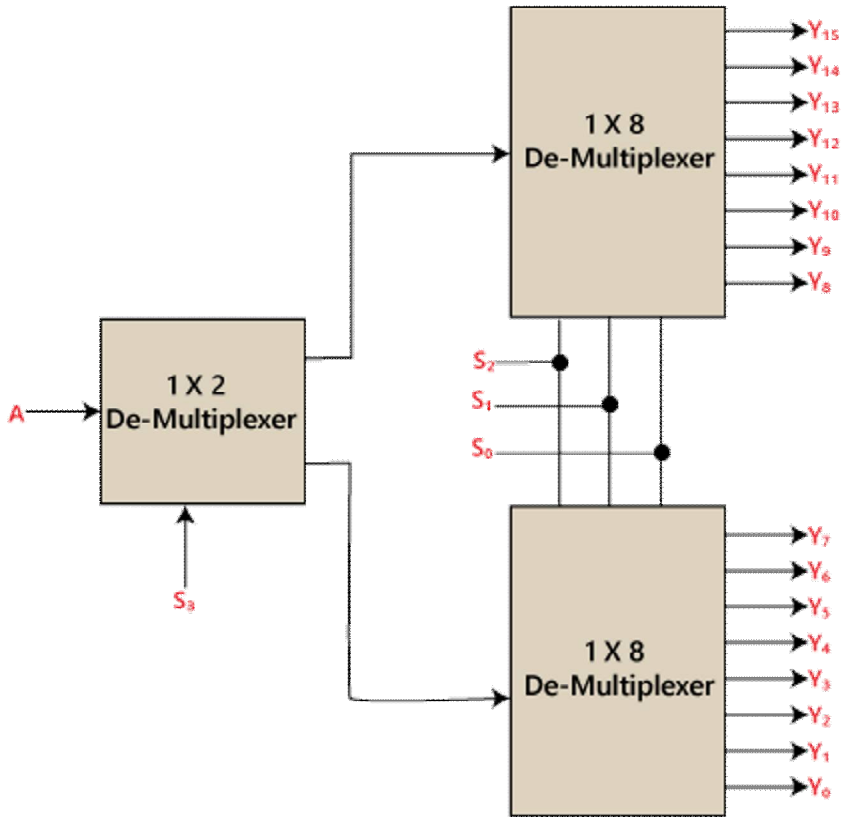


1×16 de-multiplexer using 1×8 and 1×2 de-multiplexer

We can implement the 1×16 de-multiplexer using a lower order de-multiplexer. To implement the 1×16 de-multiplexer, we need two 1×8 de-multiplexer and one 1×2 de-multiplexer. The 1×8 multiplexer has 3 selection lines, 1 input, and 8 outputs. The 1×2 de-multiplexer has only 1 selection line.

For getting 16 data outputs, we need two 1×8 de-multiplexer. The 1×8 de-multiplexer produces eight outputs. So, in order to get the final output, we need a 1×2 de-multiplexer to produce two outputs from a single input. Then we pass these outputs into both the de-multiplexer as an input. The block diagram of 1×16 de-

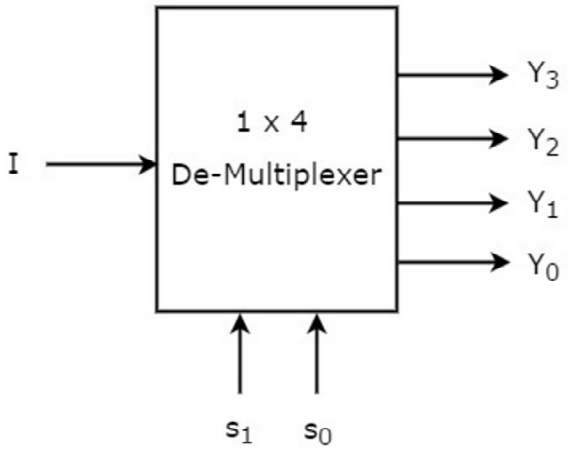
multiplexer using 1×8 and 1×2 de-multiplexer is given below.



From <https://www.javatpoint.com/de-multiplexer-digital-electronics>

1x4 De-Multiplexer

1x4 De-Multiplexer has one input I, two selection lines, s1 & s0 and four outputs Y3, Y2, Y1 &Y0. The block diagram of 1x4 De-Multiplexer is shown in the following figure.



The single input ‘I’ will be connected to one of the four outputs, Y3 to Y0 based on the values of selection lines s1 & s0. The Truth table of 1x4 De-Multiplexer is shown below.

Selection Inputs		**Outputs**			
S1	**S0**	**Y3**	**Y2**	**Y1**	**Y0**
0	0	0	0	0	**I**
0	1	0	0	**I**	0
1	0	0	**I**	0	0
1	1	**I**	0	0	0

From the above Truth table, we can directly write the Boolean functions for each output as

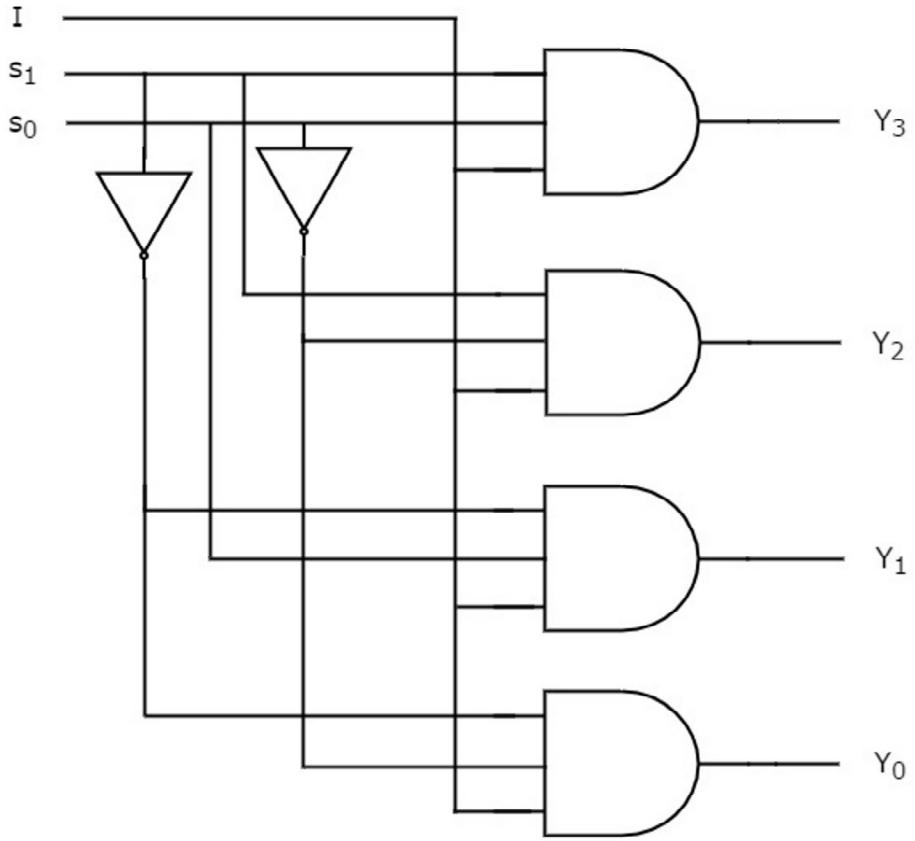
Y3=s1s0I

Y2=s1s0′I

Y1=s1′s0I

Y0=s1′s0′I

We can implement these Boolean functions using Inverters & 3-input AND gates. The circuit diagram of 1x4 De-Multiplexer is shown in the following figure.



We can easily understand the operation of the above circuit. Similarly, you can implement 1x8 De-Multiplexer and 1x16 De-Multiplexer by following the same procedure.

Implementation of Higher-order De-Multiplexers

Now, let us implement the following two higher-order De-Multiplexers using lower-order De-Multiplexers.

- 1x8 De-Multiplexer
- 1x16 De-Multiplexer

1x8 De-Multiplexer

In this section, let us implement 1x8 De-Multiplexer using 1x4 De-Multiplexers and 1x2 De-Multiplexer. We know that 1x4 De-Multiplexer has single input, two selection lines and four outputs. Whereas, 1x8 De-Multiplexer has single input, three selection lines and eight outputs.

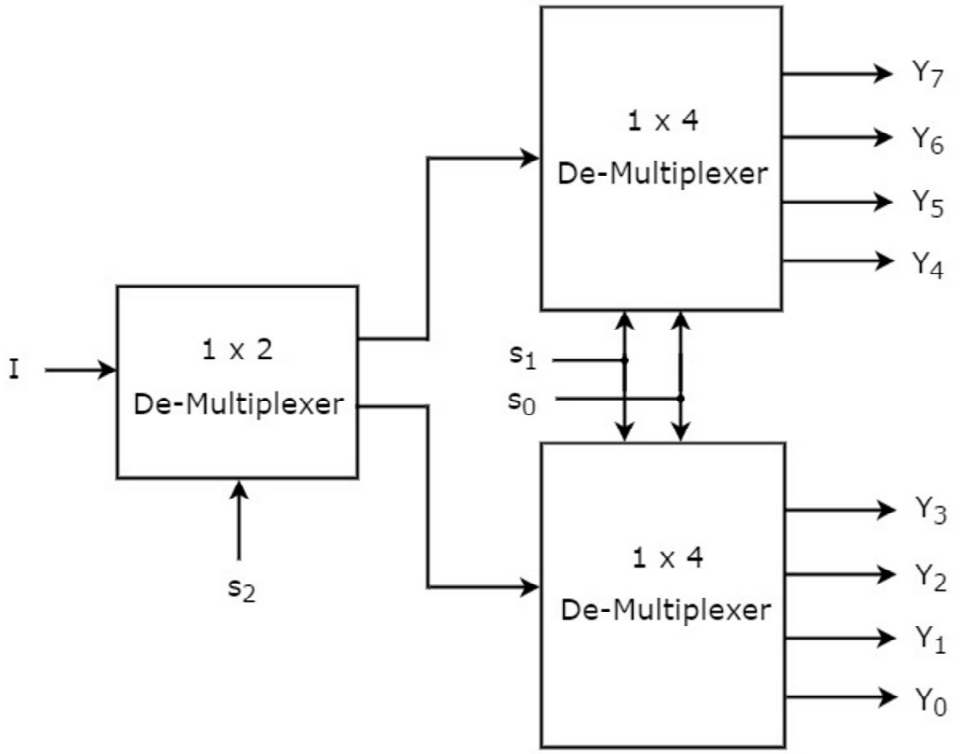
So, we require two 1x4 De-Multiplexers in second stage in order to get the final eight outputs. Since, the number of inputs in second stage is two, we require 1x2 DeMultiplexer in first stage so that the outputs of first stage will be the inputs of second

stage. Input of this 1x2 De-Multiplexer will be the overall input of 1x8 De-Multiplexer.

Let the 1x8 De-Multiplexer has one input I, three selection lines s2, s1 & s0 and outputs Y7 to Y0. The Truth table of 1x8 De-Multiplexer is shown below.

Selection Inputs			**Outputs**							
s2	**s1**	**s0**	**Y7**	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**
0	0	0	0	0	0	0	0	0	0	**I**
0	0	1	0	0	0	0	0	0	**I**	0
0	1	0	0	0	0	0	0	**I**	0	0
0	1	1	0	0	0	0	**I**	0	0	0
1	0	0	0	0	0	**I**	0	0	0	0
1	0	1	0	0	**I**	0	0	0	0	0
1	1	0	0	**I**	0	0	0	0	0	0
1	1	1	**I**	0	0	0	0	0	0	0

We can implement 1x8 De-Multiplexer using lower order Multiplexers easily by considering the above Truth table. The block diagram of 1x8 De-Multiplexer is shown in the following figure.



The common selection lines, s1 & s0 are applied to both 1x4 De-Multiplexers. The outputs of upper 1x4 De-Multiplexer are Y7 to Y4 and the outputs of lower 1x4 De-Multiplexer are Y3 to Y0.

The other selection line, s2 is applied to 1x2 De-Multiplexer. If s2 is zero, then one of the four outputs of lower 1x4 De-Multiplexer will be equal to input, I based on the values of selection lines s1 & s0. Similarly, if s2 is one, then one of the four outputs of upper 1x4 DeMultiplexer will be equal to input, I based on the values of selection lines s1 & s0.

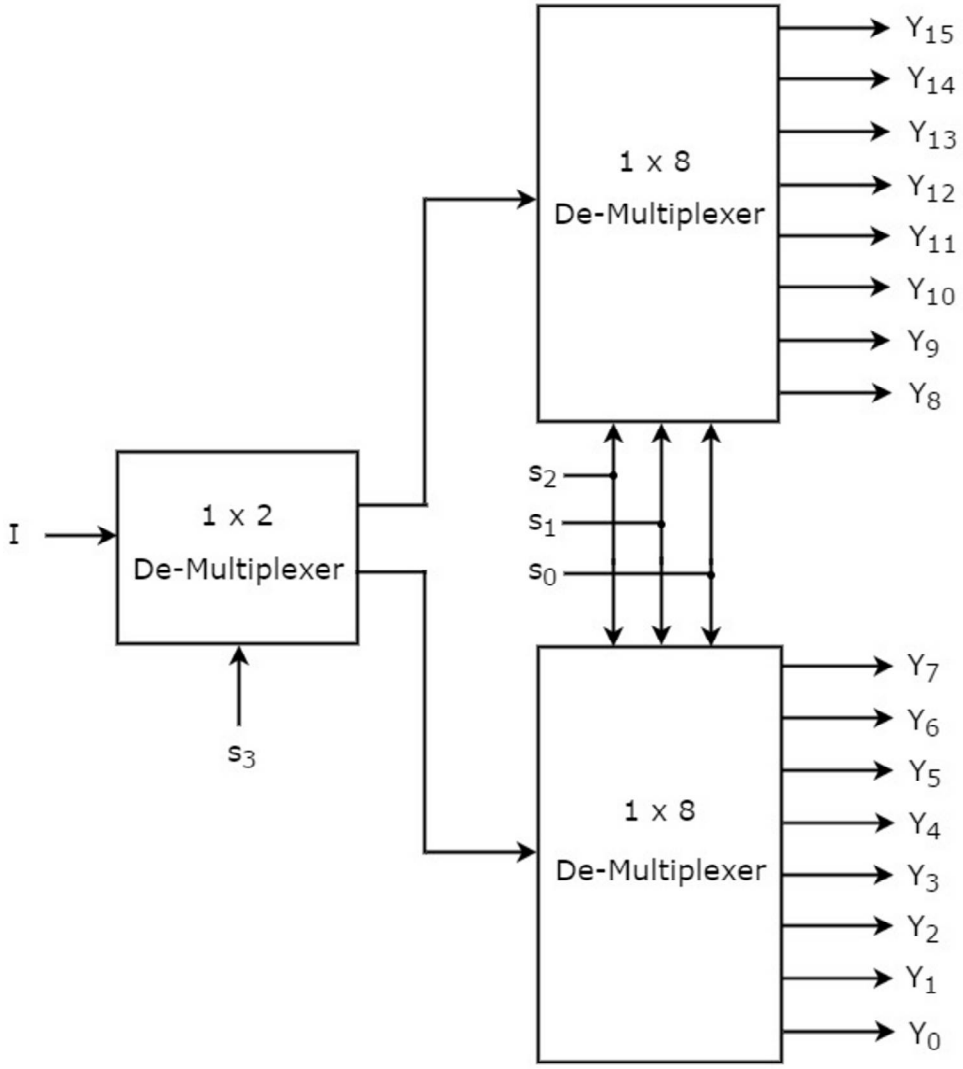
1x16 De-Multiplexer

In this section, let us implement 1x16 De-Multiplexer using 1x8 De-Multiplexers and 1x2 De-Multiplexer. We know that 1x8 De-Multiplexer has single input, three selection lines and eight outputs. Whereas, 1x16 De-Multiplexer has single input, four selection lines and sixteen outputs.

So, we require two 1x8 De-Multiplexers in second stage in order to get the final sixteen outputs. Since, the number of inputs in second stage is two, we require 1x2 DeMultiplexer in first stage

so that the outputs of first stage will be the inputs of second stage. Input of this 1x2 De-Multiplexer will be the overall input of 1x16 De-Multiplexer.

Let the 1x16 De-Multiplexer has one input I, four selection lines s3, s2, s1 & s0 and outputs Y15 to Y0. The block diagram of 1x16 De-Multiplexer using lower order Multiplexers is shown in the following figure.



The common selection lines s2, s1 & s0 are applied to both 1x8 De-Multiplexers. The outputs of upper 1x8 De-Multiplexer are Y15 to Y8 and the outputs of lower 1x8 DeMultiplexer are Y7 to Y0.

The other selection line, s3 is applied to 1x2 De-Multiplexer. If s3 is zero, then one of the eight outputs of lower 1x8 De-Multiplexer will be equal to input, I based on the values of

selection lines s2, s1 & s0. Similarly, if s3 is one, then one of the 8 outputs of upper 1x8 De-Multiplexer will be equal to input, I based on the values of selection lines s2, s1 & s0.