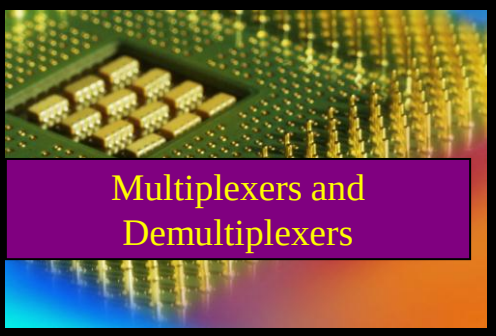
EEE251

Logic Design and Circuits

Assist.Prof. Yavuz BAYAM



Multiplexers and Demultiplexers

December 16, 2013

Multiplexers

The term multiplex means **"many into one."** Multiplexers transmit large numbers of information channels to a smaller number of channels

The selection of the particular input channel is controlled by a set of select inputs.
A digital multiplexer of 2^n input channels can be controlled by n numbers of select lines and an input line is selected according to the bit combinations of select lines.

Multiplexer

- A multiplexer switches (or routes) data from 2^N inputs to one output, where N is the number of select (or control) inputs.

- A multiplexer (mux) is a digital switch.

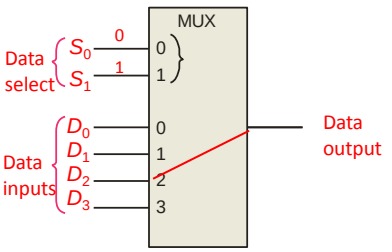
Multiplexers

A multiplexer (MUX) selects one data line from two or more input lines and routes data from the selected line to the output. The particular data line that is selected is determined by the select inputs.

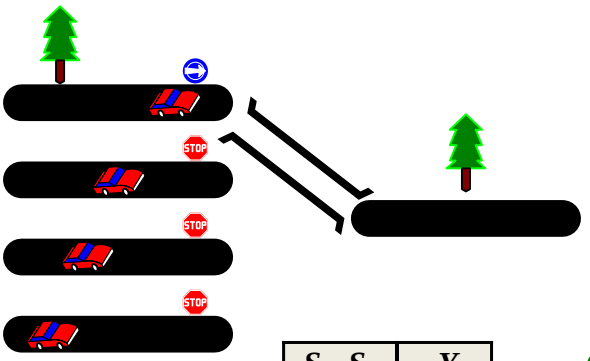
Two select lines are shown here to choose any of the four data inputs.

Question

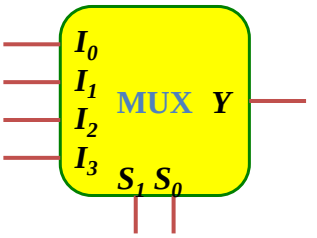
Which data line is selected if $S_1S_0 = 10$? D_2



Multiplexers

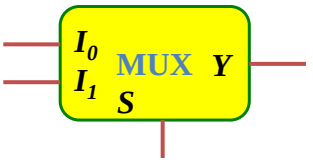


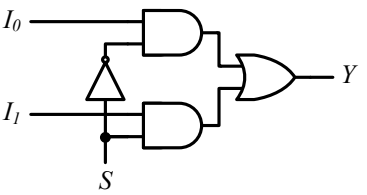
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



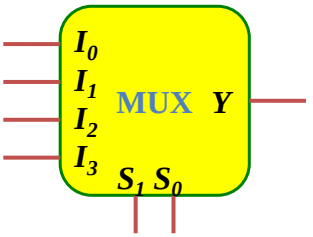
Multiplexers

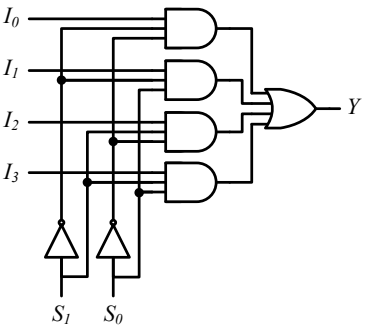
- 2-to-1 MUX



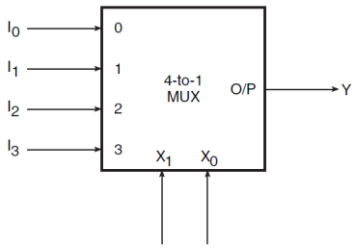


- 4-to-1 MUX





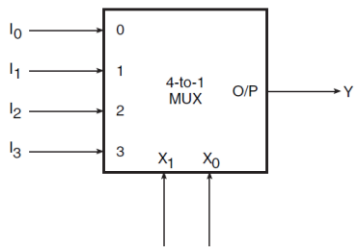
Multiplexers



X_1	X_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

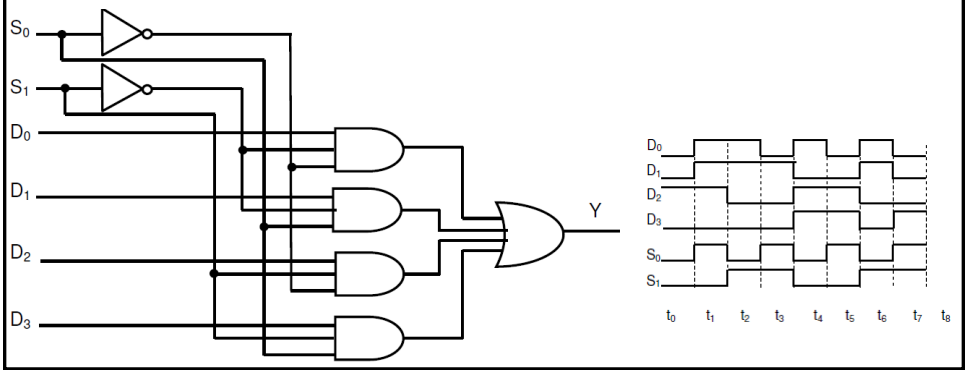
$$Y = I_0.\overline{S_1}.\overline{S_0} + I_1.\overline{S_1}.S_0 + I_2.S_1.\overline{S_0} + I_3.S_1.S_0$$

Multiplexers



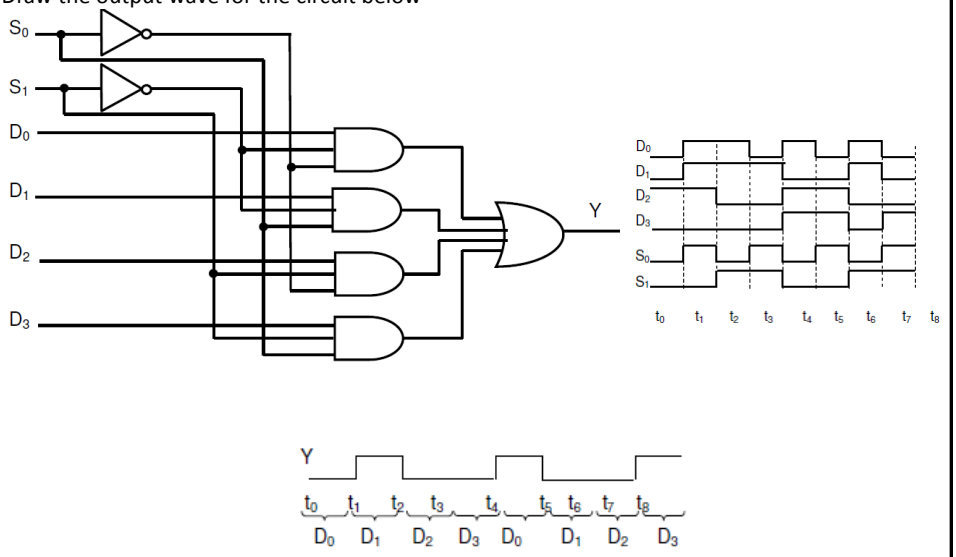
X_1	X_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = I_0.\overline{S_1}.\overline{S_0} + I_1.\overline{S_1}.S_0 + I_2.S_1.\overline{S_0} + I_3.S_1.S_0$$

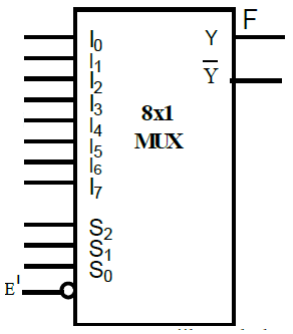


Multiplexers

Draw the output wave for the circuit below

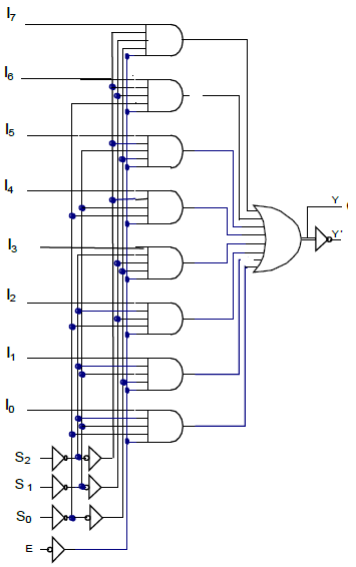


Multiplexers

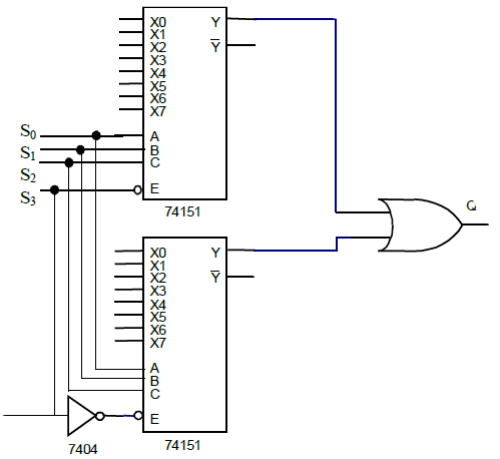


INPUTS				OUTPUTS	
E'	S_2	S_1	S_0	Y'	Y
1	X	X	X	1	0
0	0	0	0	I_0'	I_0
0	0	0	1	I_1'	I_1
0	0	1	0	I_2'	I_2
0	0	1	1	I_3'	I_3
0	1	0	0	I_4'	I_4
0	1	0	1	I_5'	I_5
0	1	1	0	I_6'	I_6
0	1	1	1	I_7'	I_7

Multiplexers

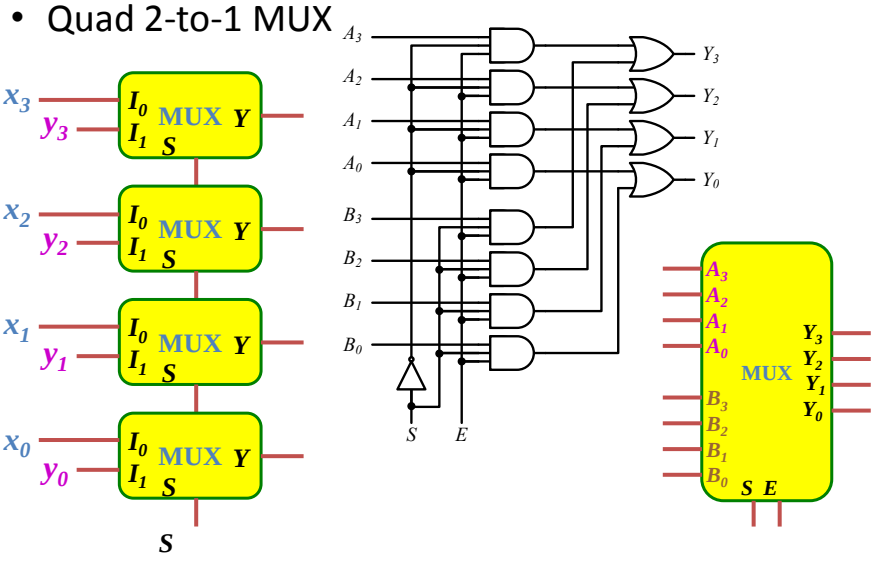


Designing 16x1 MUX with two 8x1 MUX



Multiplexers

- Quad 2-to-1 MUX



Boolean Function Implementation

- If a Boolean function consists of n+1 number of variables, n of these variables may be used as the select inputs of the multiplexer.
- The remaining single variable of the function is used as the input lines of the multiplexer.
- If X is the left-out variable, the input lines of the multiplexer may be chosen from four possible values, - X, X', logic 1, or logic 0.
- It is possible to implement any Boolean function with a multiplexer by intelligent assignment of the above values to input lines and other variables to selection lines.
- By this method a Boolean function of n+1 variables can be implemented by a 2n-to-1 line multiplexer.
- Assignment of values to the input lines can be made through a typical procedure, which will be demonstrated by the following examples.

Boolean Function Implementation

Example : Implement the 3-variable function F(A,B,C) = (0,2,4,7) with a multiplexer

- First, the function is expressed in its sum of the minterms form. Assume that the most significant variables will be used at input lines and the other n–1 variables will be connected to selection lines of the multiplexer in ordered sequence.
- This means the lowest significant variable is connected to S_0 input, the next higher significant variable is connected to S_1, the next higher variable to S_2, and so on.
- Now consider the single variable A.
- Since this variable represents the highest order position in the sequence of variables, it will be at complemented form in the minterms 0 to 2n-1, which comprises the first half of the list of minterms.
- The variable A is at uncomplemented form in the second half of the list of the minterms.
- For a three-variable function like the example, among the possible eight minterms, A is complemented for the minterms 0 to 3 and at uncomplemented form for the minterms 4 to 7.

Boolean Function Implementation

Example : Implement the 3-variable function F(A,B,C) = (0,2,4,7) with a multiplexer

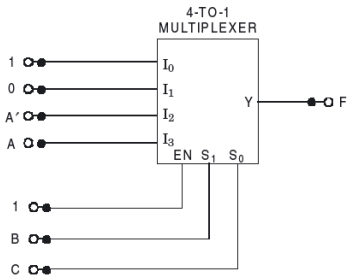
- An implementation table is now formed, where the input designations of the multiplexer are listed in the first row. Under them the minterms where A is at complemented form are listed row-wise.
- At the next row other minterms of A at uncomplemented form are listed.
- Circle those minterms that produce output to logic 1.
- If the two elements or minterms of a column are not circled, write 0 under that column.
- If both the two elements or minterms of a column are circled, write 1 under that column.
- If the upper element or minterm of a column is circled but not the bottom, write A' under that column.
- If the lower element or minterm of a column is circled but not the upper one, write A under that column.
- The lower most row now indicates input behavior of the corresponding input lines of the multiplexer as marked at the top of the column.

Boolean Function Implementation

Example : Implement the 3-variable function F(A,B,C) = (0,2,4,7) with a multiplexer

Minterms	*A*	*B*	*C*	*F*
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

	I_0	I_1	I_2	I_3
A′	(0)	1	(2)	3
A	(4)	5	6	(7)
	1	0	A′	A

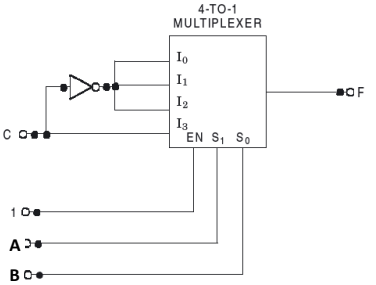


Boolean Function Implementation

Example : Implement the 3-variable function F(A,B,C) = (0,2,4,7) with a multiplexer

It may be noted that it is not necessary to reserve the most signifi cant variable foruse at multiplexer inputs.
Example may also be implemented if variable C is used at multiplexer inputs and, A and B are applied to selection inputs S_1 and S_0 respectively.
In this case the function table is modified as in Figure below and circuit implementation is shown in Figure.
Note that the places of minterms are changed in the implementation table in Figure due to the change in assignment of selection inputs.

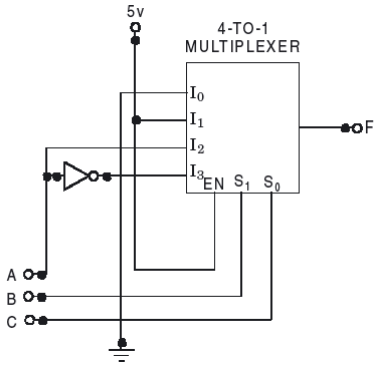
	I_0	I_1	I_2	I_3
C′	(0)	(2)	(4)	6
C	1	3	5	(7)
	C′	C′	C′	C



Boolean Function Implementation

Example : *Implement the following function using a multiplexer.*
F(A, B, C) = (1, 3, 5, 6)

	I_0	I_1	I_2	I_3
A'	0	(1)	2	(3)
A	4	(5)	(6)	7
	0	1	A	A'

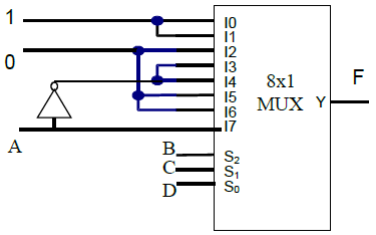


Boolean Function Implementation

Example : *Implement the following function with a multiplexer.*
F (A, B, C, D) = (0, 1, 3, 4, 8, 9, 15)

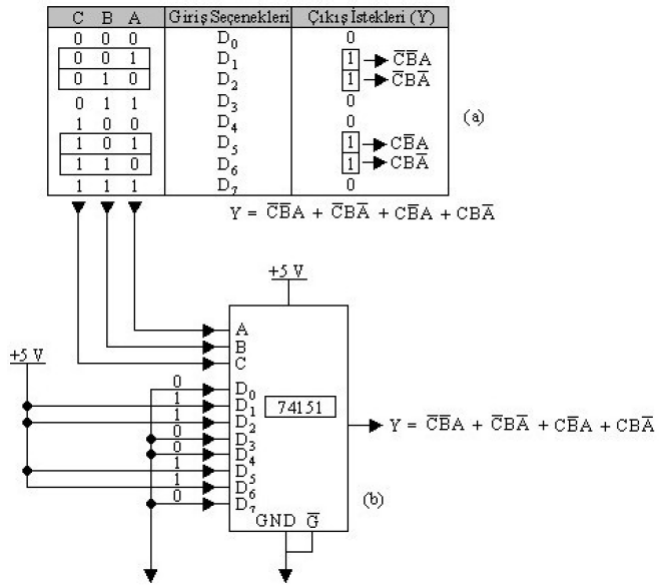
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	(0)	(1)	2	(3)	(4)	5	6	7
A	(8)	(9)	10	11	12	13	14	(15)
	1	1	0	A'	A'	0	0	A



Boolean Function Implementation

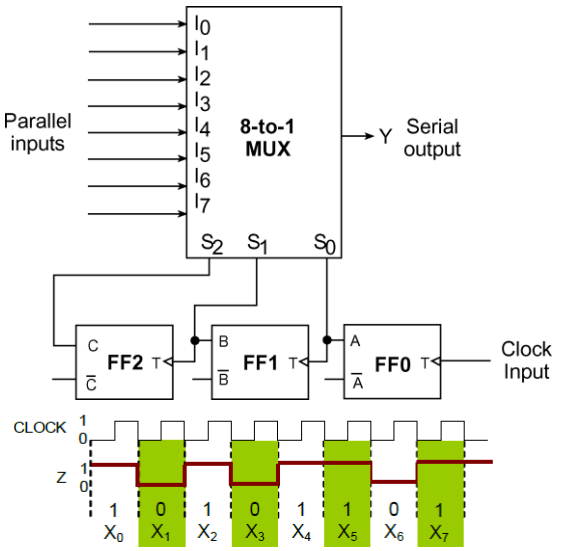
If input veriables are equal to selectors. There is no need to creat a table



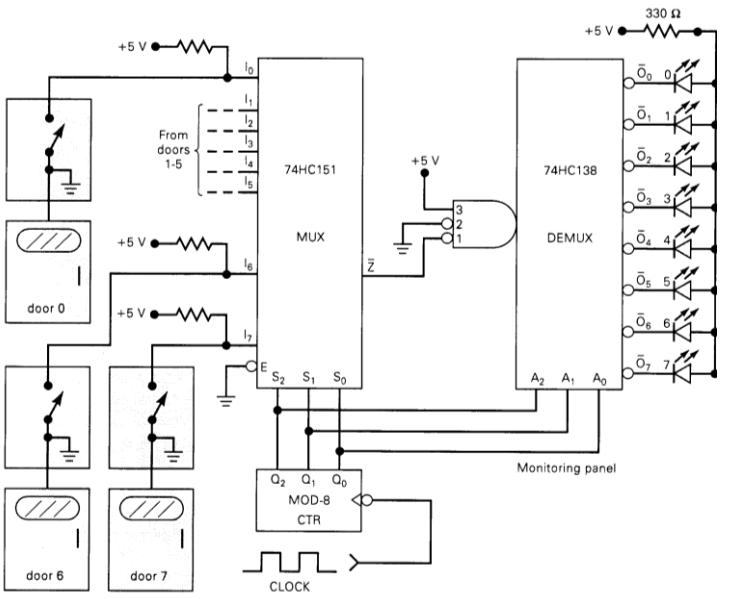
Multiplexers for Parallel-to-Serial Data Conversion

- Although data are processed in parallel in many digital systems to achieve faster processing speeds, when it comes to transmitting these data relatively large distances, this is done serially.
- The parallel arrangement in this case is highly undesirable as it would require a large number of transmission lines.
- Multiplexers can possibly be used for parallel-to-serial conversion.
- An 8-to-1 multiplexer is used to convert eight-bit parallel binary data to serial form.
- A three-bit counter controls the selection inputs.
- As the counter goes through 000 to 111, the multiplexer output goes through I_0 to I_7.
- The conversion process takes a total of eight clock cycles

Multiplexers for Parallel-to-Serial Data Conversion

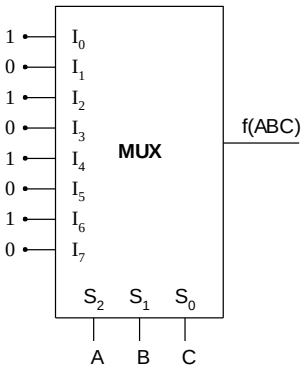


A SECURITY MONITORING SYSTEM with MUX And DEMUX



TEST

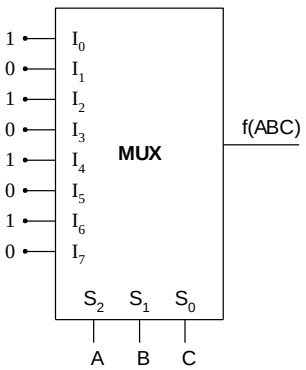
FILL IN THE TABLE



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

TEST

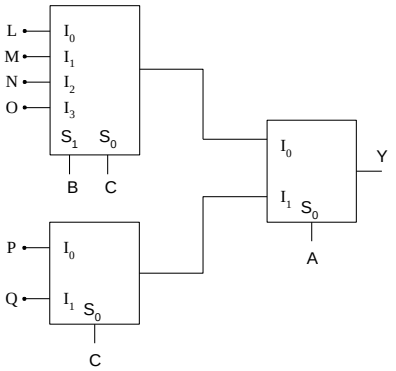
FILL IN THE TABLE



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

TEST

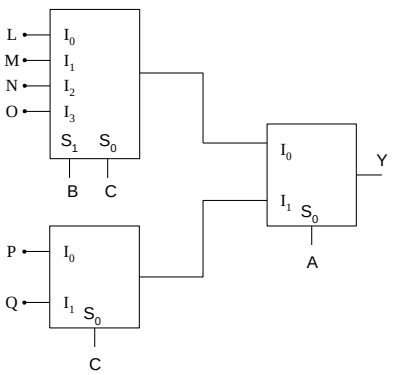
FILL IN THE TABLE



A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

TEST

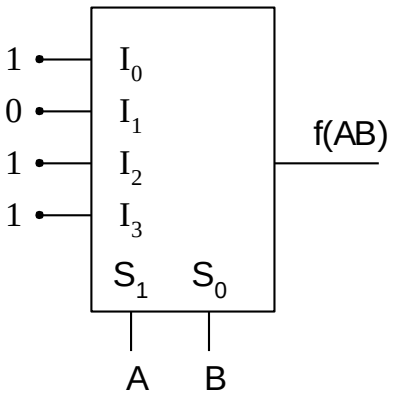
FILL IN THE TABLE



A	B	C	Y
0	0	0	L
0	0	1	M
0	1	0	N
0	1	1	O
1	0	0	P
1	0	1	Q
1	1	0	P
1	1	1	Q

TEST

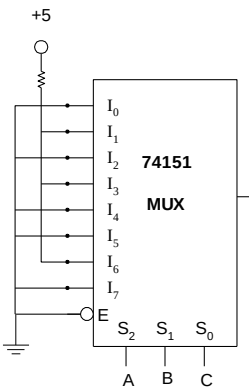
WRITE A BOOLEAN EXPRESSION FOR THE CIRCUIT



$$\overline{A}\,\overline{B} + A\overline{B} + AB$$

TEST

STATE THE BOOLEAN EXPRESSION



$$\overline{A}\,\overline{B}C + \overline{A}BC + AB\overline{C}$$

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

Demultiplexers

•A demultiplexer switches (or routes) data from one input to 2^N outputs, where N is the number of select inputs.

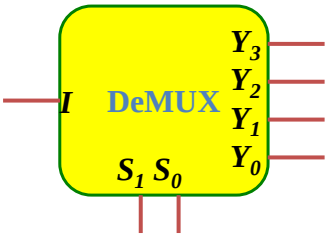
•A demultiplexer (demux) is also a digital switch.

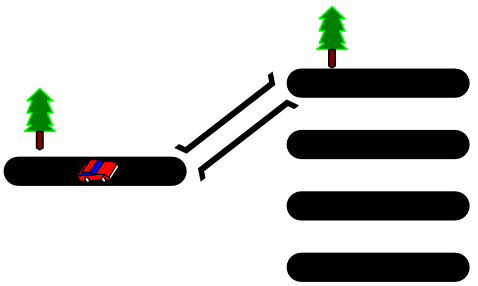
•A demultiplexer performs the opposite function of a multiplexer.

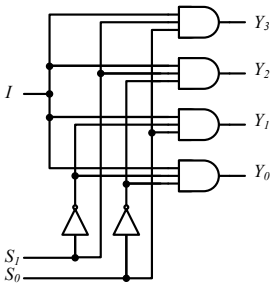
DeMultiplexers

- The term “demultiplex” means one into many.
- Demultiplexing is the process that receives information from one channel and distributes the data over several channels.
- It is the reverse operation of the multiplexer.
- A demultiplexer is a combinational logic circuit with an input line, 2^n output lines and n select lines.
- It routes the information present on the input line to any of the output lines.
- The output line that gets the information present on the input line is decided by the bit status of the selection lines.

DeMultiplexers







S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

DeMultiplexers

Seçme kodu			Çıkışlar							
S_2	S_1	S_0	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
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

$I_0 = I.(\overline{S_2}\ \overline{S_1}\ \overline{S_0})$

$I_1 = I.(\overline{S_2}\ \overline{S_1}\ S_0)$

$I_2 = I.(\overline{S_2}\ S_1\ \overline{S_0})$

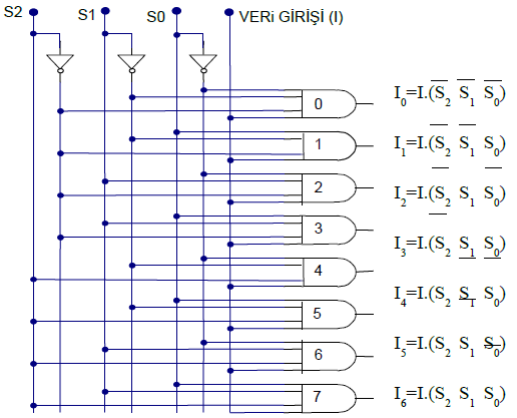
$I_3 = I.(\overline{S_2}\ S_1\ S_0)$

$I_4 = I.(S_2\ \overline{S_1}\ \overline{S_0})$

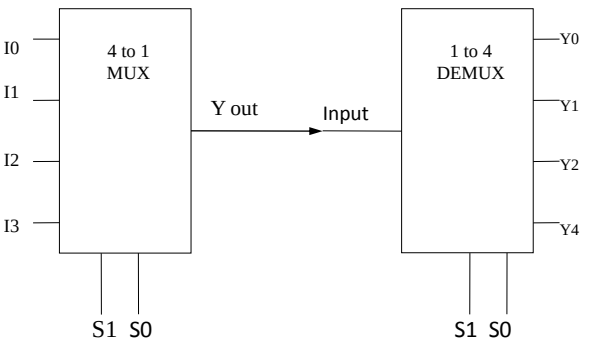
$I_5 = I.(S_2\ \overline{S_1}\ S_0)$

$I_6 = I.(S_2\ S_1\ \overline{S_0})$

DeMultiplexers



Relation between a multiplexer and a Demultiplexer.



41