

Digital Electronics

Combinational Logic Functions

Multiplexers & Demultiplexers

Multiplexers:

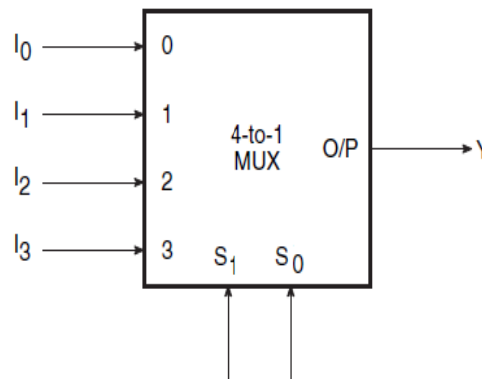
Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from any of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Generally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.



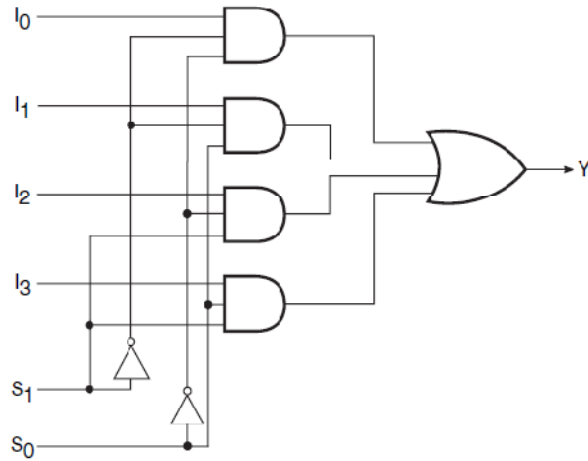
4 × 1 Multiplexer:

A 4 × 1 Multiplexer shown, consists of four input lines I_0 to I_3 , two select lines S_1 and S_0 .

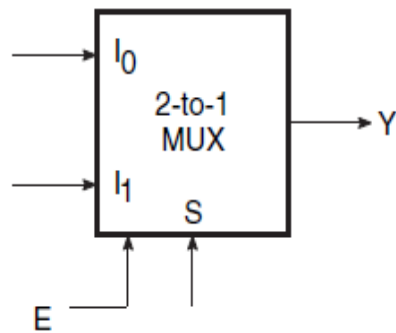
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



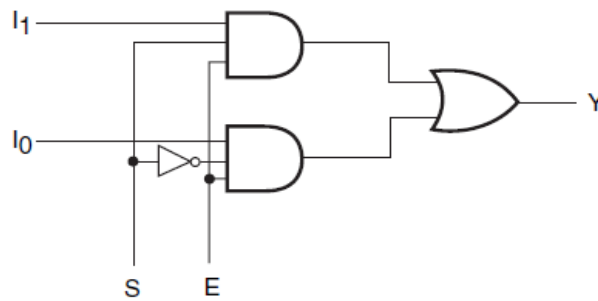
$$Y = I_0S_1'S_0 + I_1S_1S_0 + I_2S_1S_0' + I_3S_1S_0$$



2 × 1 Multiplexer with Enable:



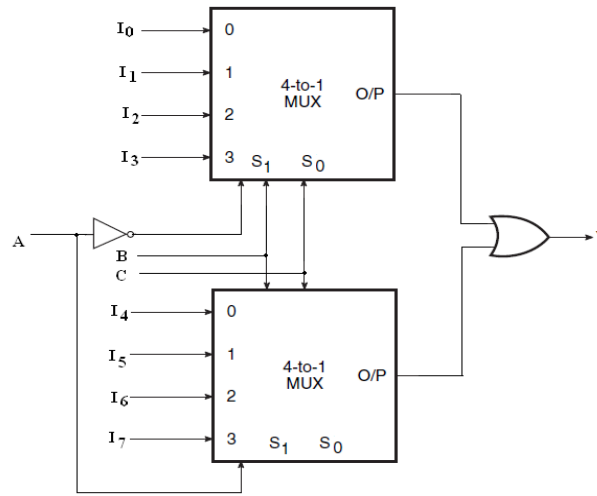
S	E	Y
X	0	0
0	1	I ₀
1	1	I ₁



Multiplexers usually have an ENABLE input that can be used to control the multiplexing function. When this input is enabled, that is, when it is in logic '1' or logic '0' state, depending upon whether the ENABLE input is active HIGH or active LOW respectively, the output is enabled. The multiplexer functions normally. When the ENABLE input is inactive, the output is disabled and permanently goes to either logic '0' or logic '1' state, depending upon whether the output is un-complemented or complemented.

Cascading of Multiplexers:

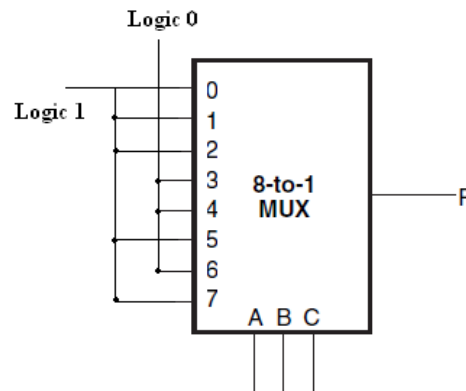
Implement a 8×1 Multiplexer using two 4×1 multiplexers.



Combinational Logic Implementation:

One of the most common applications of a multiplexer is its use for implementation of combinational logic Boolean functions. The simplest technique for doing so is to employ a 2^n -to-1 MUX to implement an n -variable Boolean function. The input lines corresponding to each of the minterms present in the Boolean function are made equal to logic '1' state. The remaining minterms that are absent in the Boolean function are disabled by making their corresponding input lines equal to logic '0'.

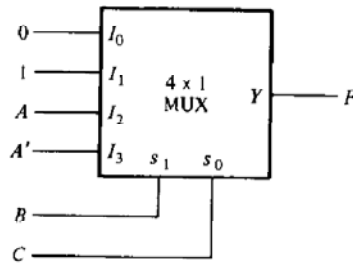
1. Implement $F(A, B, C) = \sum(0, 1, 2, 5, 7)$ using a 8×1 Multiplexer.



2. Implement $F(A, B, C) = \sum(1, 3, 5, 6)$ using a 4×1 Multiplexer.

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

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



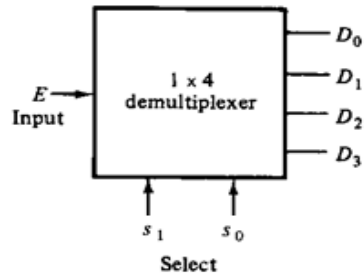
List the inputs of the multiplexer and under them list all the minterms in two rows. The first row lists all those minterms where A is complemented, and the second row all the minterms with A uncomplemented. Circle all the minterms of the function and inspect each column separately.

If the two minterms in a column are not circled, apply 0 to the corresponding multiplexer input. If the two minterms are circled, apply 1 to the corresponding multiplexer input. If the bottom minterm is circled and the top is not circled, apply A to the corresponding multiplexer input. If the top minterm is circled and the bottom is not circled, apply A' to the corresponding multiplexer input.

Demultiplexer:

A Demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output line is controlled by the bit values of n selection lines.

1 × 4 Demultiplexer:



S_1	S_0	E	D_0	D_1	D_2	D_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

$$D_0 = S_1' S_0' E, \quad D_1 = S_1' S_0 E, \quad D_2 = S_1 S_0' E, \quad D_3 = S_1 S_0 E$$

