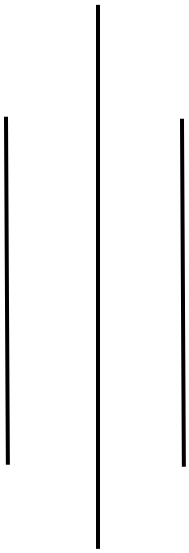
SHAID SMARAK COLLEGE

Kirtipur, Kathmandu



Lab no: 6 of Digital logics

Submitted by :-

1st semester

Amir Maharjan

Submitted to :-

Himal Raj Gental

LAB 6: Multiplexer

Objective:

- To about multiplexer and how to implement it.

Discussion:

A multiplexer makes it possible for several input signals to share one device or resource for example, one analog-to-digital converter or one communications transmission medium, instead of having one device per input signal. Multiplexers can also be used to implement Boolean functions of multiple variables

Multiplexer:

A digital multiplexer is a combinational circuit that selects binary information from one of many inputs lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. There are 2^n input lines and n selection lines whose bit combination determine which input is selected.

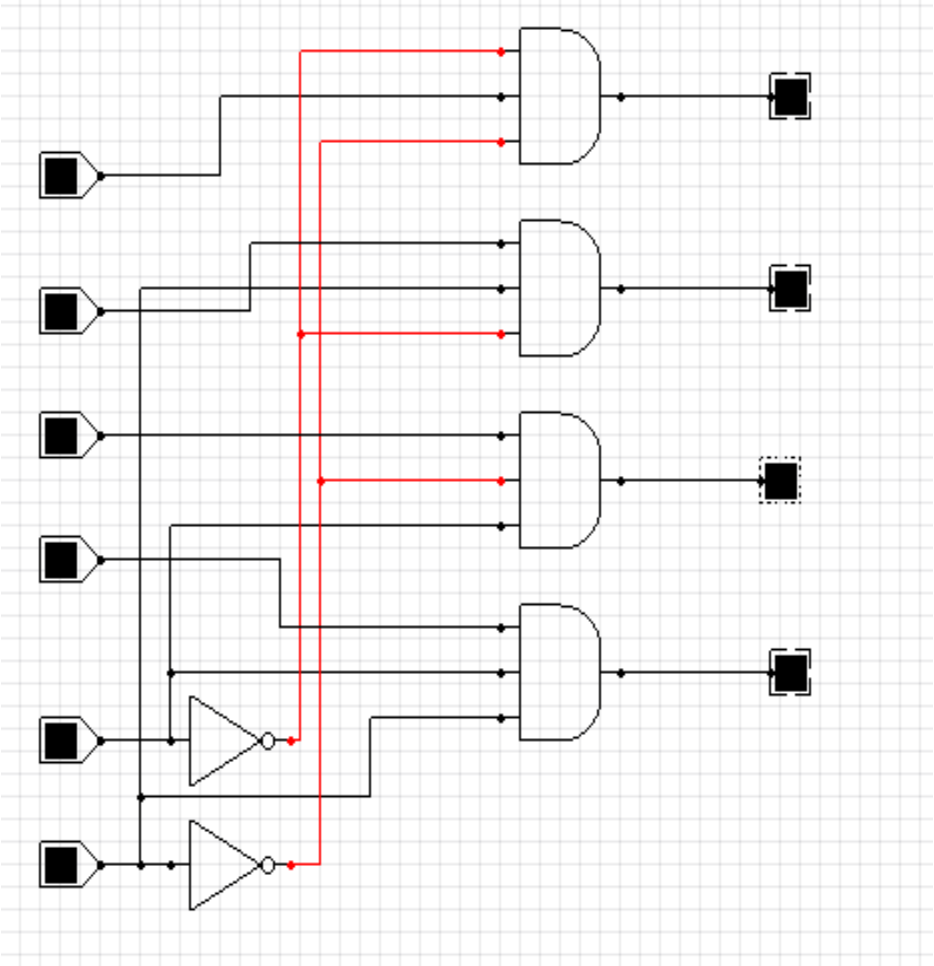


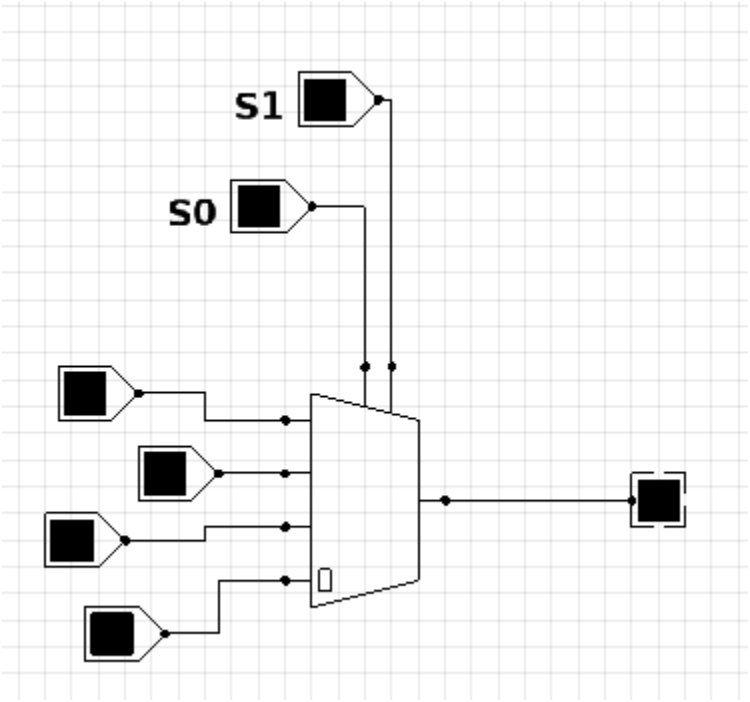
Fig: Logic diagram of 4-to-1 multiplexer.

- **Boolean Function Implementation**

As decoder can be used to implement a Boolean function by employing an external OR gate, we can implement any Boolean function with multiplexer (MUX) since MUX is essentially a decoder with the OR gate already available. If we have a Boolean function of n + 1 variables we take n of these variables and connect them to the selection lines of MUX. The remaining single variable is used for inputs in MUX. If A is this single variable, the inputs of the MUX are chosen to be either A or A' or 1 or 0. By judicious use of these four values for the inputs and by connecting the other variables to the selection lines, one can implement any Boolean function with MUX. So, it is possible to generate any function of n + 1 variable with 2^n-to-1 MUX.

S_1	S_0	Y
0	0	1_0
0	1	1_1
1	0	1_2
1	1	1_3

Function Table

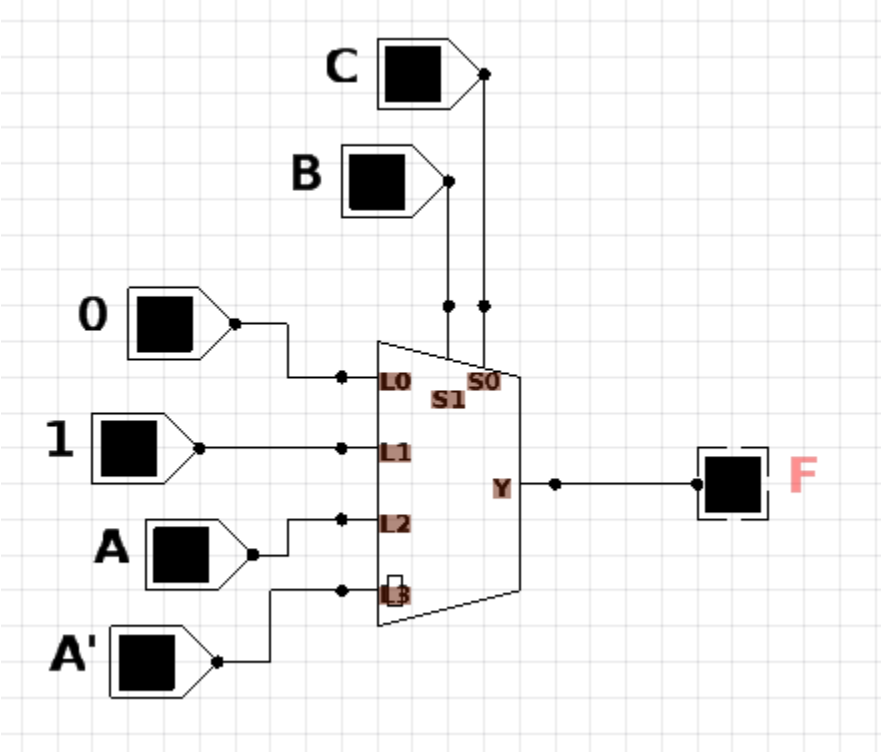


Block Diagram of MUX

 # Example: - Implement Boolean function

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

→The function can be implemented with 4-to-1 MUX. Two of the variables B and C are applied to the selection line in that order, i.e., B is connected to S_1 and C is connected to S_0. The inputs of the MUX are 0, 1, A and A'.



MUX Implementation

Minterms	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

Truth Table

	L_0	L_1	L_2	L_3
A’	0	1	2	3
A	4	5	6	7
	0	1	A	A’

Implementation Table

#Most important thing during this is the implementation table which is derived from following rules.

→List the inputs of the MUX and under them list all the minterms in 2 rows. The first-row list all those minterms where A is complemented, and the second row all the minters with A uncomplemented. Circle all the minterms and inspect each column separately, in this case the cell is shaded yellow.

1. If the two minterms in a column are not circled, apply 0 to the corresponding MUX input.
2. If the two minterms are circled, apply 1 to the corresponding MUX input.
3. If the bottom minterm is circled, apply A to the corresponding MUX input.
4. If the top minterm is circled and the bottom is not circled, apply A’ to the corresponding MUX input.