**LAB EXPERIMENT 3**

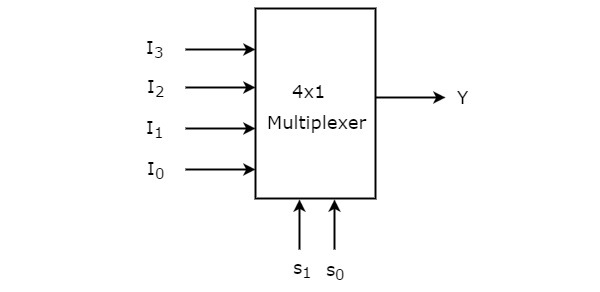
**Aim:** To design a 4:1 mux and 1:8 De mux in Xilinx Software using Verilog coding language and then view check your outputs with the truth table.

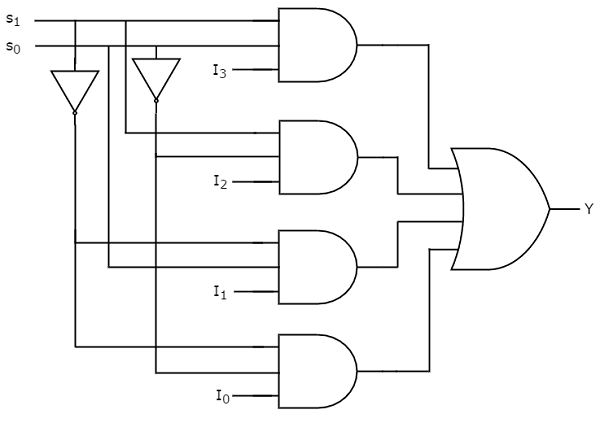
**Theory:**

1. 4:1 Mux: **Multiplexer** is a combinational circuit that has maximum of 2n data inputs, ‘n’ selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

4x1 Multiplexer has four data inputs I3, I2, I1 & I0, two selection lines s1 & s0 and one output Y. One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.

Circuit Diagram:



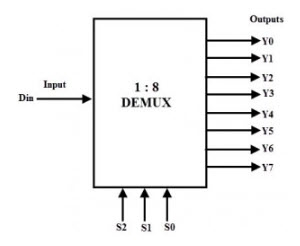


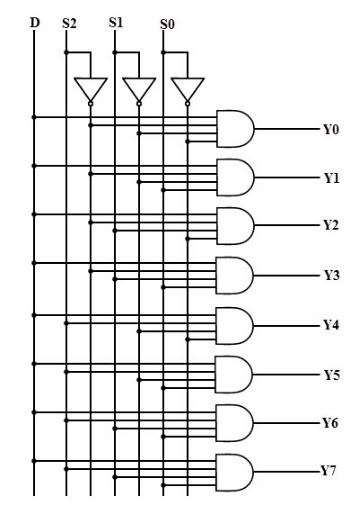
Truth Table:

|  |  |  |
| --- | --- | --- |
| **Selection Lines** | | **Output** |
| **S1** | **S0** | **Y** |
| 0 | 0 | I0 |
| 0 | 1 | I1 |
| 1 | 0 | I2 |
| 1 | 1 | I3 |

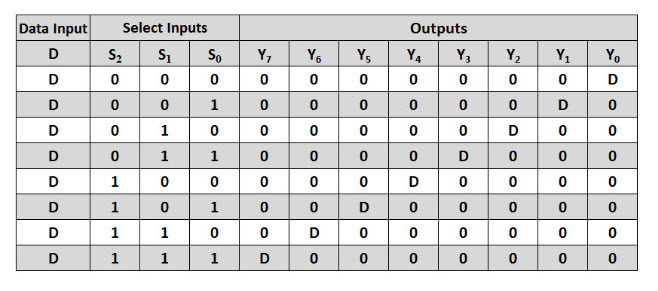
1. 1:8 De mux: A[1 to 8 demultiplexer](https://www.elprocus.com/designing-3-line-to-8-line-decoder-demultiplexer/) consists of one input line, 8 output lines and 3 select lines. Let the input be D, S1 and S2 are two select lines and eight outputs from Y0 to Y7. It is also called as 3 to 8 de mux because of the 3 selection lines. Below is the block diagram of 1 to 8 de mux.

Circuit Diagram:





Truth Table:



**Verilog Code of the Program and Outputs**

1. **4:1 Mux**
2. **Verilog code of the program:**

module Mux\_Rahil\_18070123062(

    input s1,

    input s0,

    input i0,

    input i1,

    input i2,

    input i3,

    output out

    );

assign out = (~s1&~s0&i0)|(~s1&s0&i1)|(s1&~s0&i2)|(s1&s0&i3);

endmodule

1. **Screenshots of the Program**

Graphical user interface, text, application

Description automatically generated

A picture containing text, screenshot, indoor

Description automatically generated

1. **RTL Schematics**

A picture containing text, indoor, screenshot, electronics

Description automatically generated

A picture containing text, indoor, screenshot

Description automatically generated

1. **1:8 De Mux:**
2. **Verilog Code of the Program:**

module Demux8to1\_18070123062(input I,

input S0,

input S1,

input S2,

output Out0,

output Out1,

output Out2,

output Out3,

output Out4,

output Out5,

output Out6,

output Out7);

assign Out0=I &(~S2) &(~S1) &(~S0);

assign Out1=I &(~S2) &(~S1) &(S0);

assign Out2=I &(~S2) &(S1) &(~S0);

assign Out3=I &(~S2) &(S1) &(S0);

assign Out4=I &(S2) &(~S1) &(~S0);

assign Out5=I &(S2) &(~S1) &(S0);

assign Out6=I &(S2) &(S1) &(~S0);

assign Out7=I &(S2) &(S1) &(S0);

endmodule

1. **Screenshot of the program and Outputs**

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

1. **RTL Schematics:**

A picture containing text, indoor, screenshot, electronics

Description automatically generated

**Conclusion:** From this experiment we have learnt how to implement the concept of multiplexer and demultiplexer in Xilinx with Verilog language of programming. We have also learnt how to check outputs of the circuit in the Simulator.