# Practical-7

**Name of Experiment: design a 16:1 multiplexer using 8:1 multiplexer**

**Tools required: Logisim**

**Theory:**

**Designing a 16:1 Multiplexer Using 8:1 Multiplexers**

**Understanding the Problem:**

A 16:1 multiplexer selects one out of 16 input lines based on a 4-bit select input. An 8:1 multiplexer selects one out of 8 input lines based on a 3-bit select input.

**Solution:**

To create a 16:1 multiplexer using 8:1 multiplexers, we can cascade two 8:1 multiplexers.

**Diagram:**

**Explanation:**

1. **Divide the 16 inputs:** Divide the 16 input lines into two groups of 8.
2. **Connect to 8:1 multiplexers:** Connect the first 8 inputs to the first 8:1 multiplexer and the remaining 8 inputs to the second 8:1 multiplexer.
3. **Use select inputs:** Use the first 3 bits of the 4-bit select input to control the first 8:1 multiplexer. Use the fourth bit of the select input to control the second 8:1 multiplexer and select the output of either the first or second multiplexer.

**Truth Table:**

|  |  |
| --- | --- |
| Select Input | Output |
| 0000 | Input 0 |
| 0001 | Input 1 |
| ... | ... |
| 1110 | Input 15 |
| 1111 | Input 15 |

**Code Implementation (using Verilog):**

Verilog

module mux16\_to\_1(

input [15:0] in,

input [3:0] sel,

output out

);

wire [7:0] out1, out2;

mux8\_to\_1 mux1(in[7:0], sel[2:0], out1);

mux8\_to\_1 mux2(in[15:8], sel[2:0], out2);

mux8\_to\_1 mux3(out1, out2, sel[3], out);

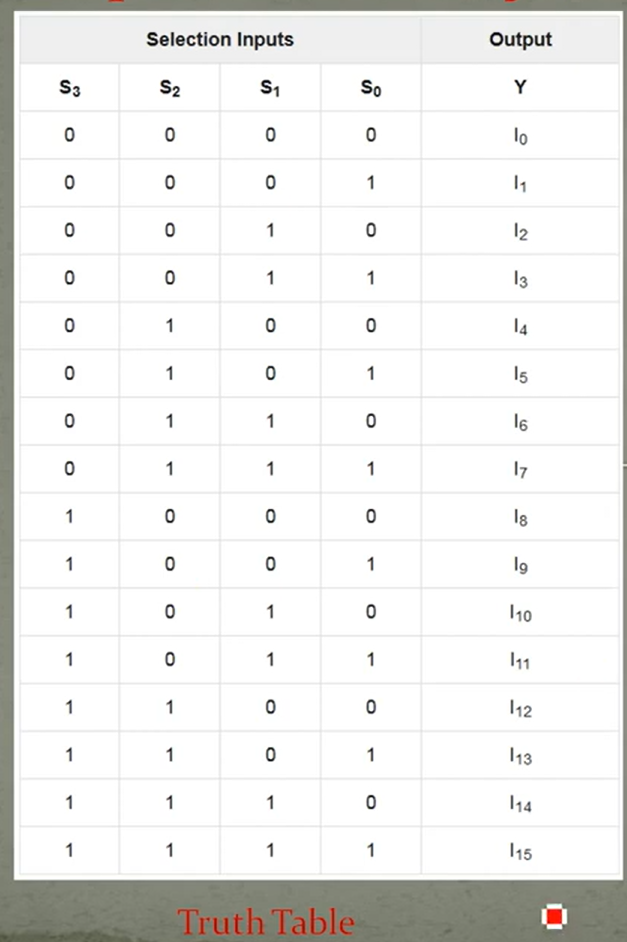
endmodule

**Advantages of this approach:**

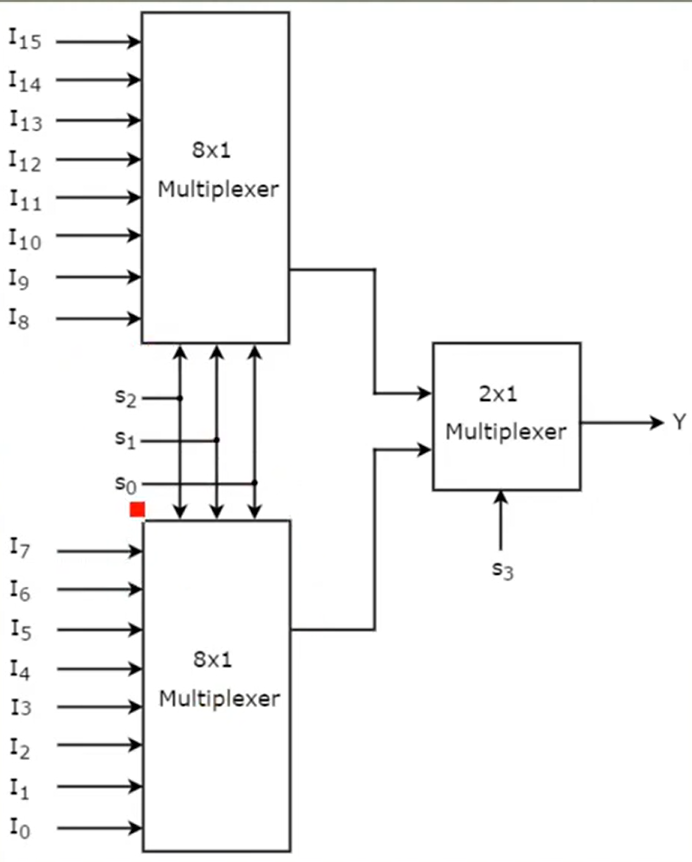
* **Modular design:** The 16:1 multiplexer is constructed using smaller, reusable 8:1 multiplexer modules.
* **Scalability:** This approach can be extended to create larger multiplexers by cascading more 8:1 multiplexers.
* **Efficiency:** The 16:1 multiplexer can be implemented efficiently using available hardware components.

By following these steps and using the provided code, you can design and implement a 16:1 multiplexer using 8:1 multiplexers.

**ANALYSIS TRUTH TABLE:**

****

**IMPLEMENTED BLOCK DIAGRAM :**

****