Laboratory 1: Circuit Designs and Testing

Shein Htike   Brandon Vasquez

CSC 343 Spring 2025

Contents

1 Exercise A: 8x1 Multiplexer Using Logic Gates

1.1 Objective

The goal of this exercise is to create an 8x1 multiplexer in two ways: using logic gates and using VHDL code.

1.2 Functionality and Specifications

1.2.1 Logic

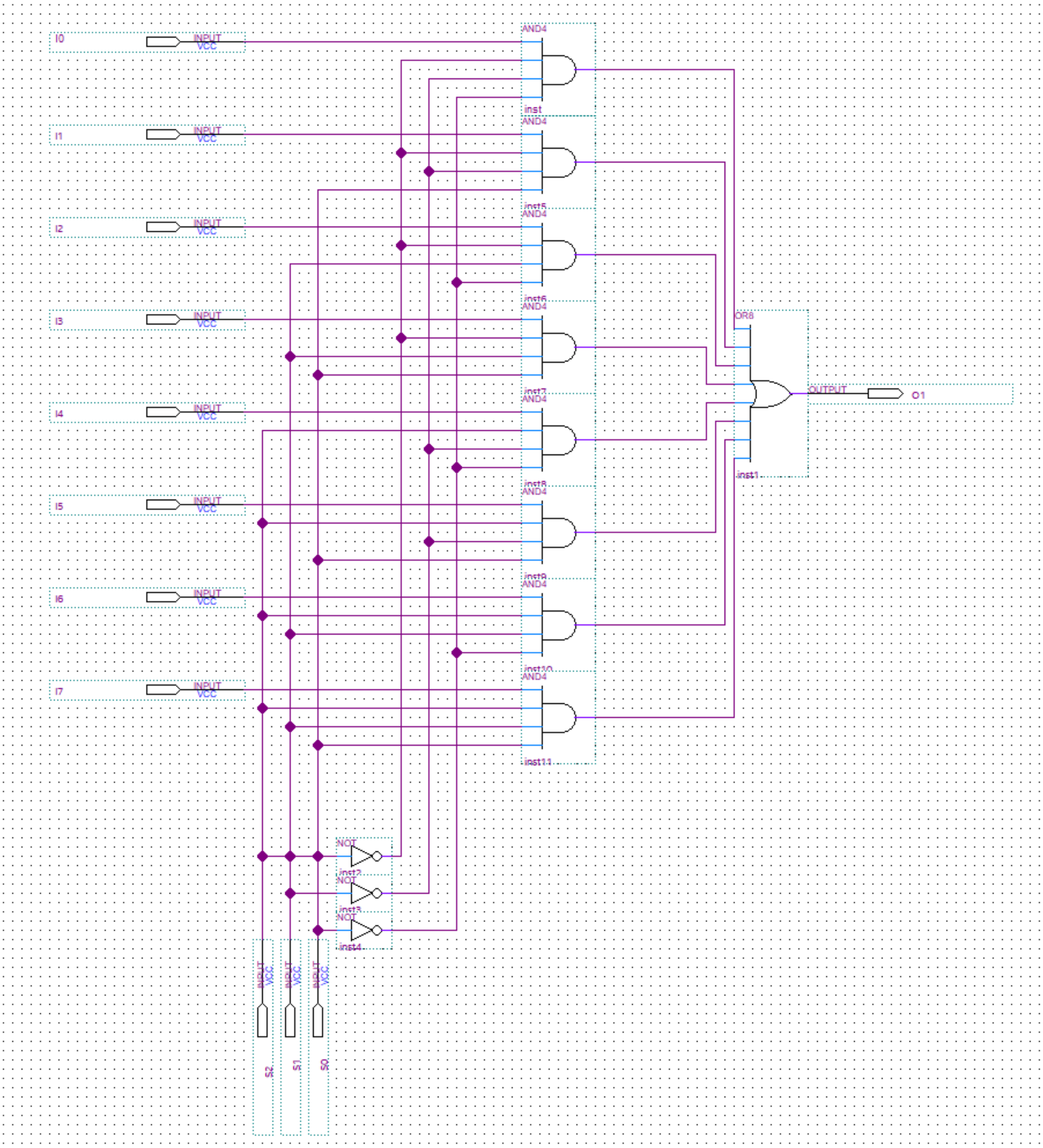
The output of the 8×1 multiplexer is given by the following logic equation:

$$\begin{aligned} \text{output} = {} & \overline{S_2}\,\overline{S_1}\,\overline{S_0}\,I_0 \;\lor\; \overline{S_2}\,\overline{S_1}\,S_0\,I_1 \;\lor \\ & \overline{S_2}\,S_1\,\overline{S_0}\,I_2 \;\lor\; \overline{S_2}\,S_1\,S_0\,I_3 \;\lor \\ & S_2\,\overline{S_1}\,\overline{S_0}\,I_4 \;\lor\; S_2\,\overline{S_1}\,S_0\,I_5 \;\lor \\ & S_2\,S_1\,\overline{S_0}\,I_6 \;\lor\; S_2\,S_1\,S_0\,I_7 \end{aligned} \tag{1}$$

In this multiplexer design, we select one of the eight inputs, I_0 through I_7 and connect it to a single output based on the binary value of the three select signals, S_2, S_1, and S_0 (with S_2 being the most significant bit).

1.2.2 Circut Design

In order to implement the 8x1 multiplexer, we created this circuit design in Intel Quartus Prime. This circuit was then compiled into VHDL and imported into ModelSim in order to simulate and test our design.



1.2.3 VHDL Code

```
library IEEE;
use IEEE.std_logic_1164.all;

entity multiplexer8x1v2 is
    port (
        I : in  std_logic_vector(7 downto 0);
        S : in  std_logic_vector(2 downto 0);
        O : out std_logic
    );
end multiplexer8x1v2;

architecture Behavioral of multiplexer8x1v2 is
begin
    with S select
        O <= I(0) when "000",
             I(1) when "001",
             I(2) when "010",
             I(3) when "011",
             I(4) when "100",
             I(5) when "101",
             I(6) when "110",
             I(7) when "111",
             '0'    when others;  -- Default case if needed
end Behavioral;
```

1.3 Simulation

2 Exercise B: 1x8 De-Multiplexer Using 1x4 and 1x2 De-Multiplexers

2.1 Objective

2.2 Functionality and Specifications

2.3 Simulation

3 Exercise C: 3-to-8 Decoder

3.1 Objective

3.2 Functionality and Specifications

3.3 Simulation

4 Exercise D: 8-to-3 Priority Encoder

4.1 Objective

4.2 Functionality and Specifications

4.3 Simulation

5 Exercise E: Set-Reset Flip-Flop & D Flip-Flop (Positive Edge Trigger)

5.1 Objective

5.2 Functionality and Specifications

5.3 Simulation

6 Conclusions