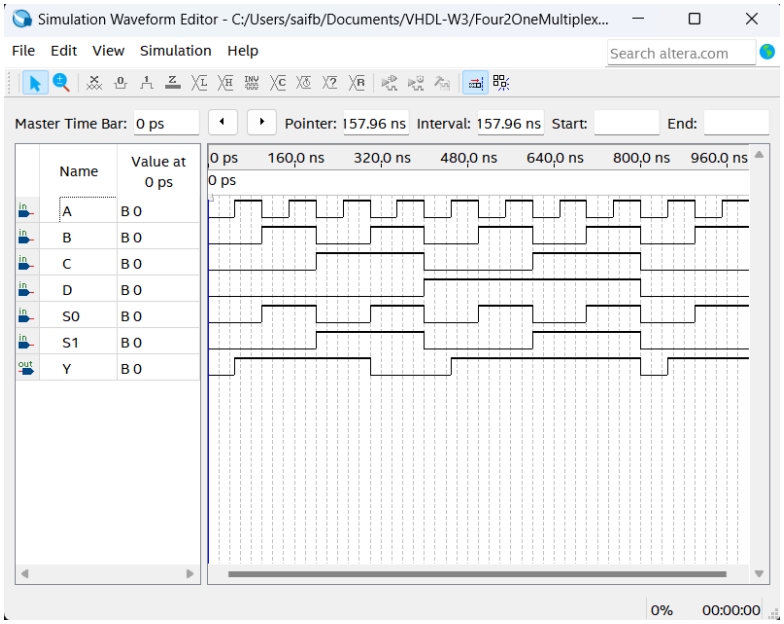
Saif Ba Madhaf - 527028

VHDL Week 3

Exercise chapter 6:

1. Design and simulate a behavioral description of a 4 to 1 multiplexer.



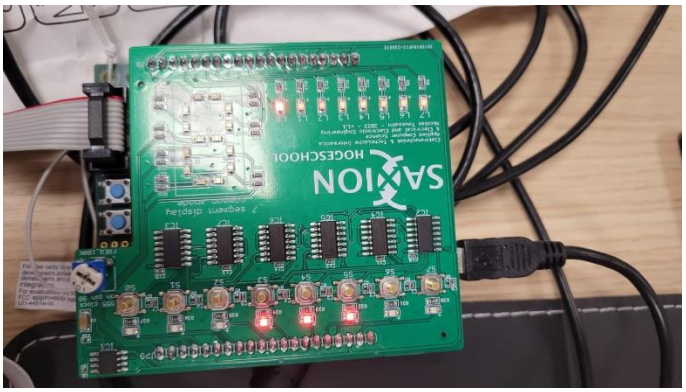
A 4 to 1 multiplexer is a circuit that selects one of four inputs signals to an output Y. Below is the truth table.

Table

S1	S0	Input	Output (Y)
0	0	A	Y = A
0	1	B	Y = B
1	0	C	Y = C
1	1	D	Y = D

How it works is if for example our S0 and S1 are both 1 and our D is also one then Y will equal D, however if D was a zero then the Y will be 0.

2. **Implement the 4 to 1 multiplexer on the max 10 board (see the digital course for info) (submit photos from the result)**

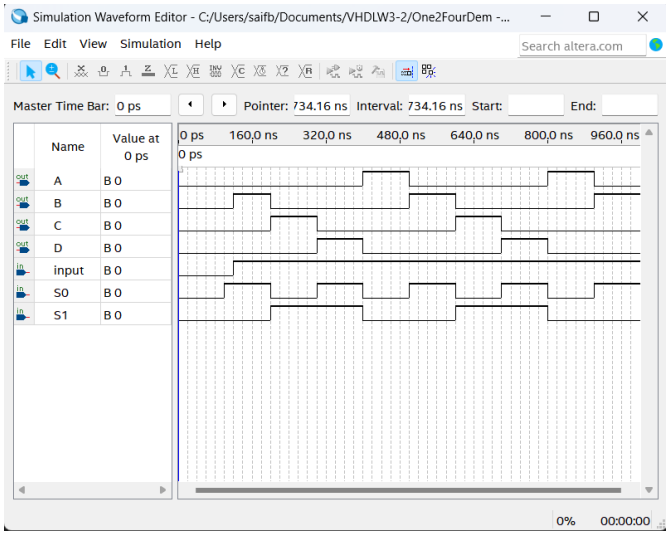


INPUTS
S1 = S4
S0 = S5
A = S0
B = S1
C = S2
D = S3

OUTPUTS
Y = LED0

Here is an example for the example given above.

3. Design and simulate a structural description of a 1 to 4 demultiplexer.

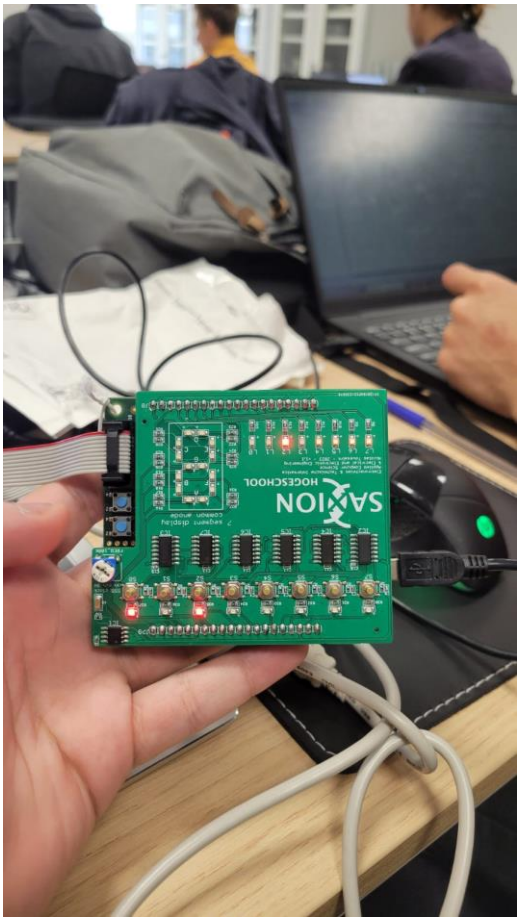


A 1 to 4 demultiplexer is a circuit that takes one input and outputs 4 lines. It has two selects S0 and S1.

S1	S0	Input	A	B	C	D
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

How it works is if for example our S0 is 0 and S1 our is 1 the our C becomes 1. Input is always 1.

4. **Implement the 1 to 4 demultiplexer on the max 10 board (see the digital course for info) (submit photos from the result)**



INPUTS
S1 = S0
S0 = S1
INPUT = S3

OUTPUTS
A = LED0
B = LED1
C = LED2
D = LED3

Here is an example for the example given above.

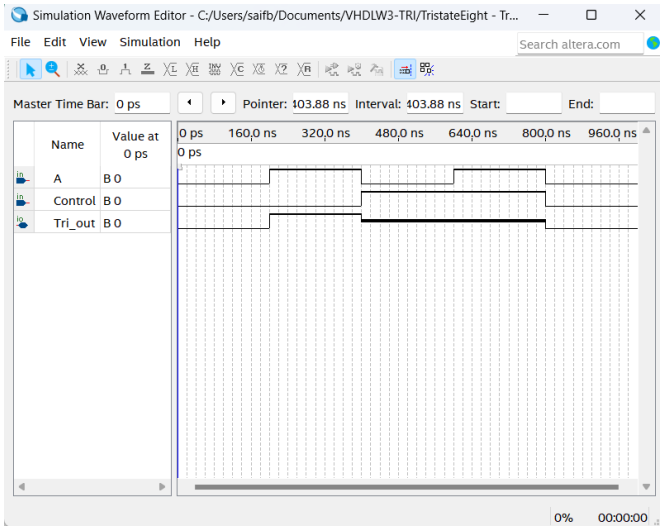
self-test chapter 7:

When is a tristate needed?

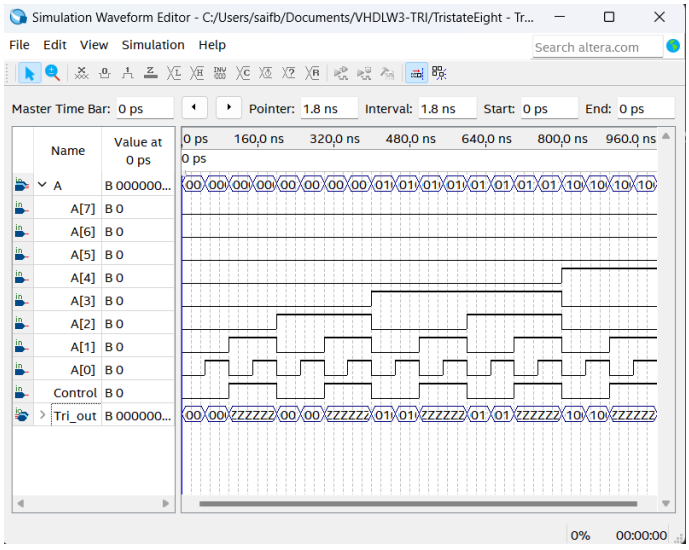
It is needed when you want to connect multiple devices to one bus.

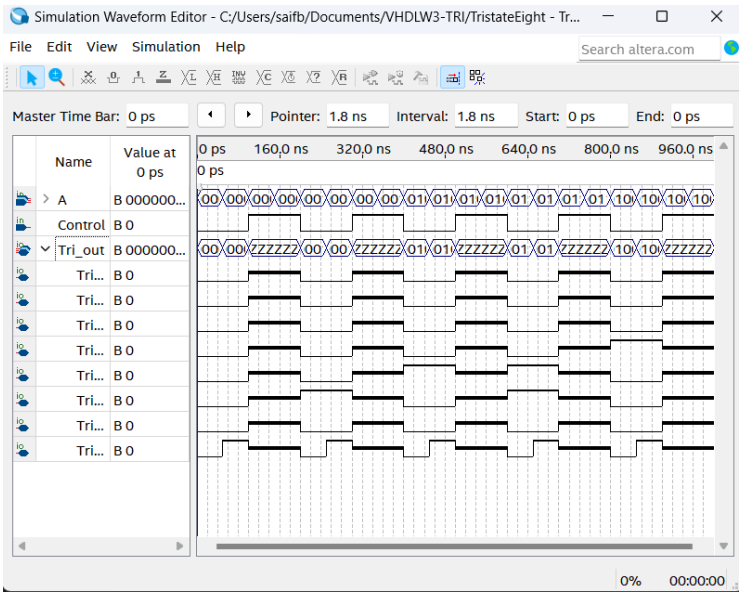
Exercise chapter 7:

1. **Simulate in Quartus the tristate model given in this chapter.**



2. **Design and simulate a tristate 8- bits buffer.**





self-test chapter 8:

Self-test chapter 8:

1. **Why is it not advisable to gate a signal with the clock?**

Because gating a signal only using a clock can lead to timing issues and glitches with the system.

2. **Describe the difference working of the two next descriptions of a flip flop.**

PROCESS (reset,Clock)

BEGIN

IF reset = '1' THEN

Q3 <= '0';

ELSIF (clock'EVENT AND clock='1') THEN

Q3 <= D;

END IF;

END PROCESS;

PROCESS (Clock)

BEGIN

IF reset = '1' THEN

Q3 <= '0';

ELSIF (clock'EVENT AND clock='1') THEN

Q3 <= D;

END IF;

END PROCESS;

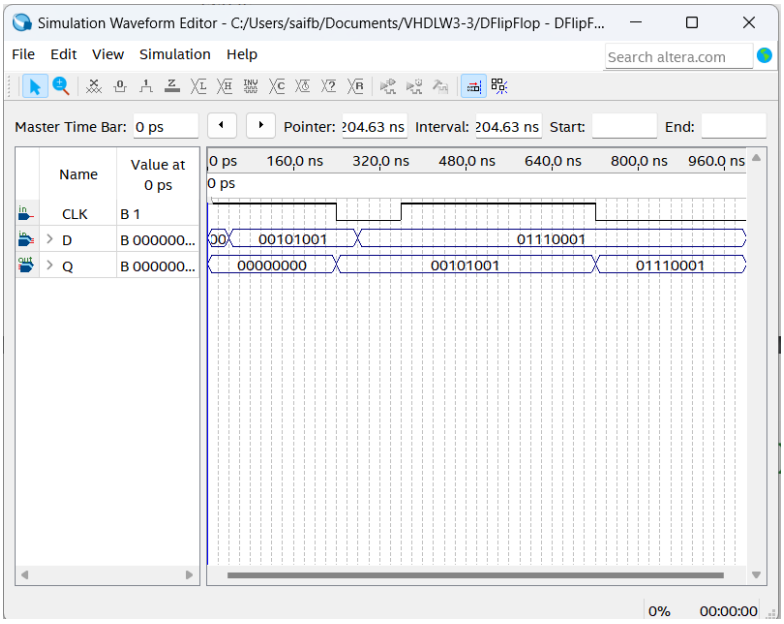
The difference between them is that in first description the process takes both the clock and the reset in the process which means that it looks at the changes of the clock and the reset. However, in the second description it only takes the clock in the process and does not look at the changes that are happening in the reset, which will cause issues in the circuit.

Exercise chapter 8:

1. Write VHDL for a D-Flipflop register (8 bits with a negative edge).

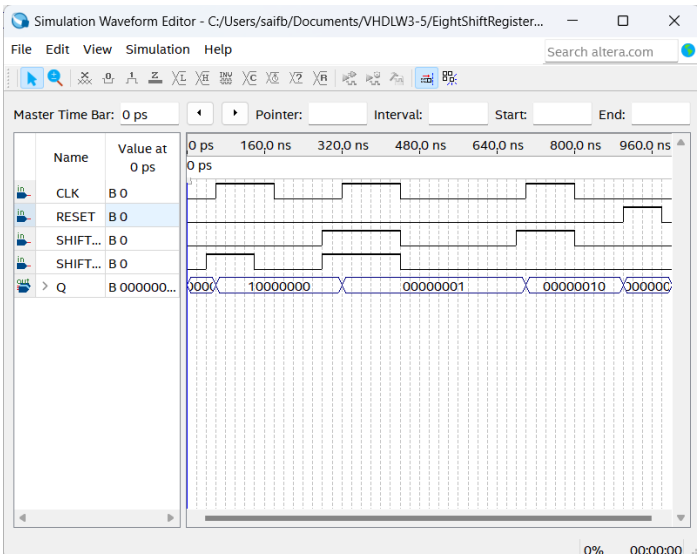
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DFlipFlop is
    Port ( CLK : in STD_LOGIC;
           D   : in STD_LOGIC_VECTOR (7 downto 0);
           Q   : out STD_LOGIC_VECTOR (7 downto 0));
end DFlipFlop;

architecture Behavioral of DFlipFlop is
    signal reg_data : STD_LOGIC_VECTOR (7 downto 0);

begin

    process(CLK)
    begin
        if falling_edge(CLK) then
            reg_data <= D;
        end if;
    end process;

    Q <= reg_data;

end Behavioral;
```

2. Simulate the D-flipflop register.



As we can see we have a clock and an input D and output Q. When there is a falling edge in the clock it starts by storing the data in D to Q. For example, we see that at the first falling edge the D is 00101001 and when the falling edge happens in the clock it immediately stored it in Q.

3. Design and simulate a 8 bits right/left shift register in VHDL



As we can see we have four inputs Clock, Reset (reseting Q to 0), Shift_in (whether to shift a 0 or 1), and Shift direction (shift left or right). There is only one output Q which is where shifting outputs. When there is a rising edge in the clock a shift begins.

4. Implement it on the on the MAX10 board.

INPUTS
CLOCK = S0
SHIFT_DIR = S1
SHIFT_IN = S2

OUTPUTS
Q (0 to 7)

As we can see in this example, we had 1 on LED 7 MSB and we shift it to right after the clock was high (rising edge) it shifted one to the right to LED6. How? Because our shift direction (S1) is 0 meaning it shifts to the right and there was a rising edge the system shifted one to the right.