CSC 343 Spring 2021

Self-Check Laboratory Exercise

MUX Design, Simulation and waveform verification in Model-Sim
Design Entry Specified: Using BDF, Components, VHDL, LPM

Instructor: Professor Izidor Gertner

What to Submit:

Please post on direct private channel to Instructor just Figures with **SCREENSHOTS** of waveforms. Figure Captions should state **IN ONE SENTENCE** why the design is correct. The file name has to have your last name and title, signals have to have your last name as a prefix.

No report, nor video is required to submit.

No grade will be given for this Self-Check lab.

A Check Mark **will be assigned. The criteria used for Check Mark** (✔) A **check mark**, **checkmark** or **tick** (✓) is a **mark** used to indicate the concept "yes" (e.g. "yes; this has been verified", "yes; that **is the** correct answer", "yes; this has been completed".

1. Design MUX 4:1

1.1. Specification of 4-to-1 Multiplexer

A 4-to-1 multiplexer is, most basically, a device that "selects" from 4 input pins. The selected signal is forwarded to the output pin, and is chosen by the values of the selector pins. Note that in a previous lab we only worked with a 2-to-1 multiplexer, which requires just one selector input. Since the 4-to-1 multiplexer selects from 4 inputs, 2 bits of information are required for the selection, meaning there are two selector inputs, for a total of 6 input ports and one output port.

1.1.1. Boolean Function for MUX 4:1

Assuming our four inputs are I_0, I_1, I_2, and I_3 and our two selectors are S_0 and S_1, a 4-to-1 multiplexer can be expressed with the following Boolean function:

$$f(I_0, I_1, I_{2,} I_3, S_{0,} S_1) = (I_0, \bar{S}_{0,} \bar{S}_1) + (I_1 S_0 \overline{S_1}) + (I_2 \bar{S}_0 S_1) + (I_3 S_0 S_1)$$

What we see here is that in each of the terms of the functions, the selectors (or their inversions) are "ANDed" in such a way that the result is the input's value only if the input is "selected," otherwise it is always "0". When the terms are added (a four way OR is applied, it is only possible that the selected input's value is "1", meaning that value shows up in the output. The truth table is useful for understanding the Boolean function. Due to the nature of what a multiplexer does, it is possible to create a much simpler table (this will be useful to us when simulating and verifying):

S_0	S_1	f
0	0	$I_!$
0	1	$I_!$
1	0	$I_!$
1	1	$I_!$

Figure 1. Simplified but comprehensive truth table for 4-to-1 multiplexer.

CSC 343   Spring 2021

Self-Check Laboratory Exercise

MUX Design, Simulation and waveform verification in Model-Sim

Design Entry Specified: Using BDF, Components, VHDL, LPM

Instructor: Professor Izidor Gertner

1.1.2. Block Diagram (BDF)

Having thoroughly specified the 4-to-1 multiplexer in theory using a Boolean function and a truth table, it's time to design the circuit using the Quartus II Block Diagram Editor. As an exercise, we designed two versions of the 4-to-1 multiplexer: one using only 2-input NAND gates, and the other using the 2-to-1 multiplexer symbol we created in the previous lab.

In the previous lab tutorial, we used Quartus II to generate symbols for MUX21that we can later import into the block diagram for the multiplexer MUX 4:1.

In order to create the second version, we must first create a Symbol for the 2-to-1 multiplexer Block Diagram created in the previous lab. Once this is complete, we can use the 2-to-1 multiplexer (which we already simulated and verified) in the new 4-to-1 multiplexer block diagram.

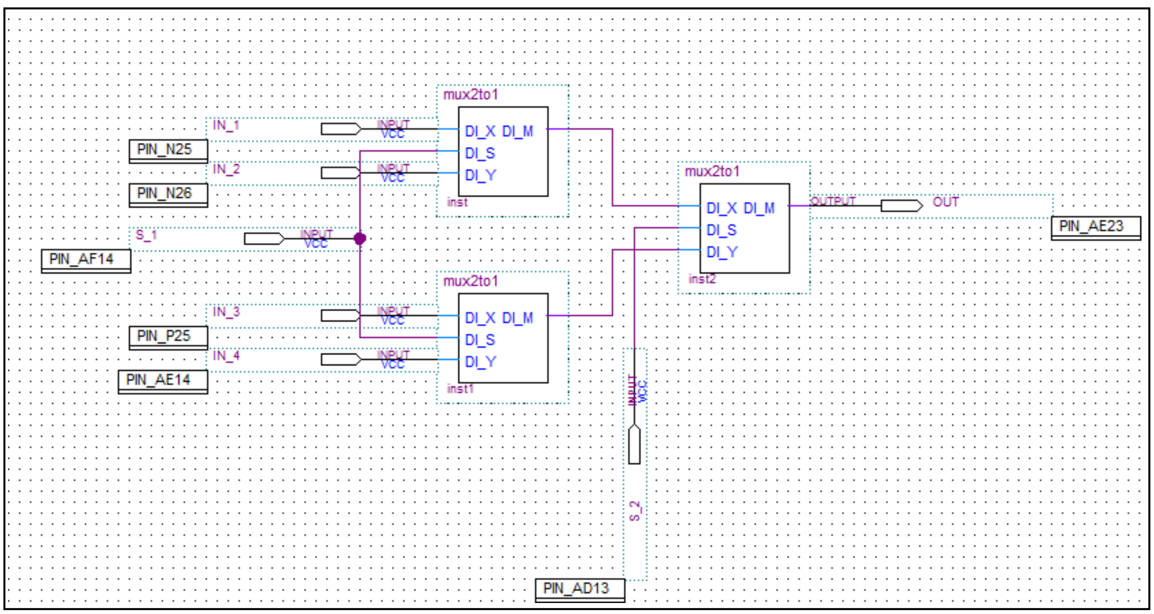

Figure 2. 4-to-1 multiplexer block diagram using the symbol for the 2-to-1 multiplexer. In your design ignore PIN_ symbols.

1.2. TASKS TO DO

TASK 1.2.1.

Design MUX 4:1 using Symbol MUX2:1 you have created in your previous lab. Your schematic design of MUX 4:1 should look similar to design shown in Figure 2.

TASK 1.2.2.

Create VHDL code for MUX4:1 using MUX2:1 as a component. For reference, please use example on BUZZER architecture, which uses gates as components, as we have discussed in class, and described in the VHDL tutorial.

```
….
Fill ME in Entity declarations and architectures
of components used as needed…
….
architecture Fill in MY_MUX41 of Fill in  is
      -- Declarations
      component MUX21
           port (in1, in2, S: in
                 std_logic; out1: out
                 std_logic);
      end component;
      -- declaration of signals used to
      interconnect

      signal  Fill in list all signals and their
      types here : std_logic;

 begin

-- Component instantiations statements
 U0: MUX21 port map (Fill_in,Fillin,Fill in );
 U1: MUX21 port map (Fill_in,Fillin,Fill in );
 U2: MUX21 port map (Fill_in,Fillin,Fill in );

 end Fill in MY_MUX41;
```

TASK 1.2.3.

Please write VHDL code using “ `std_logic_vector` ”for Four-to-One multiplexer of which each input is 4-bit word and another one with each input 8-bit word.

Entity Declaration example is shown here:

```
entity mux4-to-1 is
      port (I0,I1,I2,I3: in std_logic_vector(7 downto 0);

 SEL: in std_logic_vector (1 downto 0);
       OUT1: out std_logic-_vector(7 downto 0));
 end mux4_to_1;
```

Please write in VHDL ARCHITECTURE BODY for both cases.

```
ARCHITECTURE NAME_OF_ARCHITECTURE OF NAME_OF_ENTITY IS
……
Fill me in
….

END NAME_OF_ARCHTECTURE
```

TASK 1.2.4.

Please write VHDL code using LPM modules for Four-to-One multiplexer of which each input is 4-bit word and another one with each input 8-bit word.

TASK 1.2.5.

Please verify the correctness of all 4 designs using waveforms in Model_SIM. The values of 8 bit , and 4-bit words have to be displayed in hexadecimal. Each signal and input/output symbol has to have your last name as a prefix.

Self-Check Laboratory Exercise

MUX Design, Simulation and waveform verification in Model-Sim

Design Entry Specified: Using BDF, Components, VHDL, LPM

Instructor: Professor Izidor Gertner

4. Examples of screenshots with waveforms.

MUX 4-to-1 Multiplexer

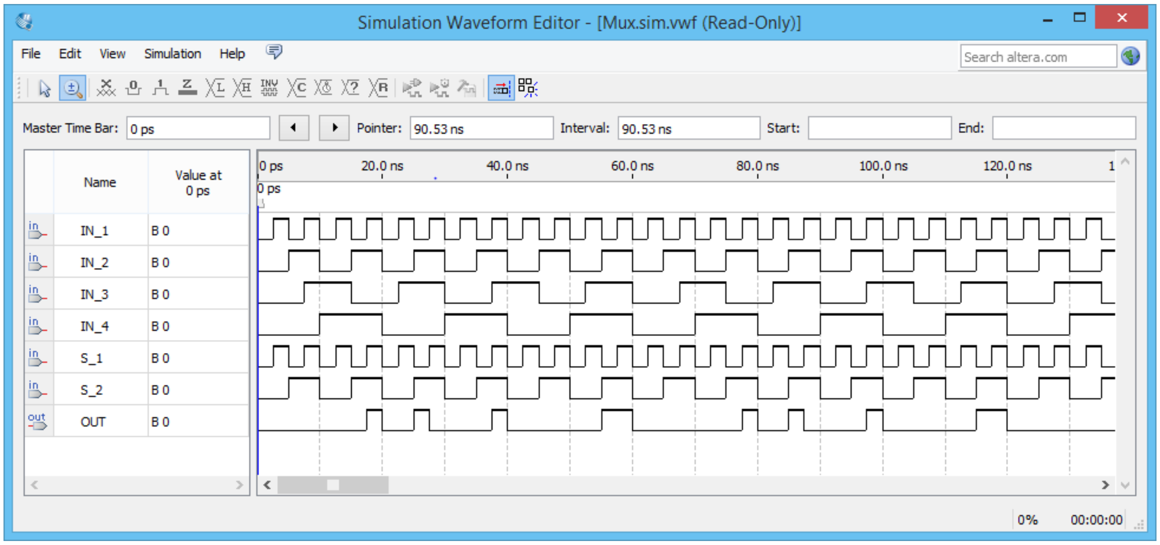

Figure 8. Functional simulation of the 4-to-1 multiplexer waveforms.

The simulation shown above provides different values every 5ns on the grid. The result matches our truth table—the input signal chosen by the selectors is forwarded to the output port.

MUX 2-to-1 8-bit Multiplexer

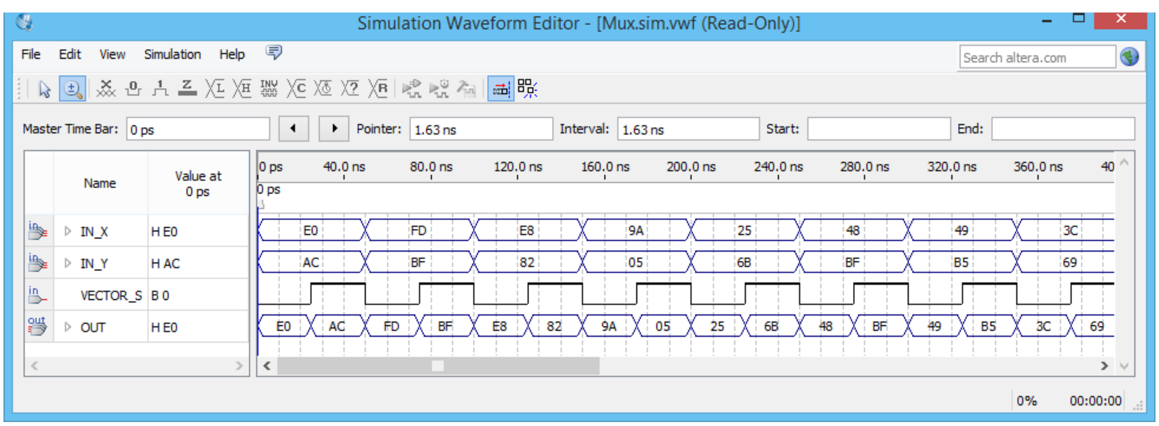

Figure 9. MUX 2-to-1 8-bit vector multiplexer simulation waveforms. Note that the radix was set to HEX.