Objectives:

The objective of this lab is to use VHDL to specify the designs and testbenches for various ***MSI combinatorial*** components. You will also explore the new features in Quartus, ModelSim, and the Altera DE10-lite board.

Required tools and parts:

Intel Quartus Prime, ModelSim-Altera, Terasic DE10-Lite board

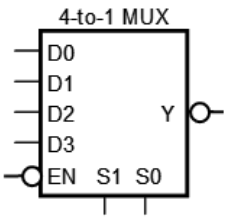
Important: For this and the subsequent labs, put your name and section number as a comment at the beginning all VHDL files and other materials that you submit.

Pre-lab Requirements:

Part 1: 4-to-1 MUX in VHDL

Specified the following 4-to-1 MUX in VHDL:

- Note that the MUX has an **active low** input (EN) and an **active low** output Y. So, you must specify your VHDL code accordingly.
- Use a **SELECT signal assign statement** or a **conditional signal assignment** statement.
- Modify the provided testbench (mux4to1WithAssert_tb) and perform a **functional simulation**.



Part 1 Pre-lab submission (on Canvas)

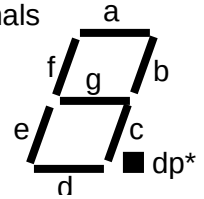
Include the following content in `your-UFID/P1` directory (see end of document for more information about submission formatting):

- VHDL code for Part 1 (mux4to1.vhd and modified mux4to1WithAssert_tb)
- Screenshot of functional simulation: showing "Simulation Finished" and 0 assertion errors.

Part 2: Display "decoder" for using 7-segment LED displays

Design and simulate the following display "decoder" for using the 7-segment LED displays. The ***logic*** table for the display "decoder" and the corresponding display are shown on the right.

- You can use any VHDL statements to implement the display "decoder", including **simple signal assignment statements**.
- Note that unlike the discrete LEDs, the 7-segment LEDs on the Altera DE10-LITE board are **active low**. In other words, signals must be asserted "low" to illuminate the desired segments. So, you must specify your VHDL code accordingly.
- Write a testbench and perform a functional simulation. You don't have to use ASSERT statements for the display-decoder outputs.



* dp: decimal point

Logic Table
(1 = true, 0 = false)

i3	i2	i1	i0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

Part 2 Pre-lab submission (on Canvas)

Include in `your-UFID/P2` directory: VHDL code for Part 2 and screenshot of simulation.

Part 3: 8-bit ripple-carry adder

Design and simulate an 8-bit ripple-carry adder using the 4-bit adders from Lab 1.

- The 4-bit ripple-carry adder from Lab 1 (VHDL version) will be used as a component in this design.
- Use PORT MAP statements to specify the 8-bit adder.
- Modify the provided testbench (adder_tb from Lab 1) and perform a functional simulation.

Part 3 Pre-lab submission (Submit on Canvas)

Include in the **`your-UFID/P3`** directory: VHDL for Part 3 and screenshot of simulation.

Part 4: Integrate 8-bit adder with 7-segment LED displays

Create an entity called Top-Level, in which you use PORT MAP statements to integrate the 8-bit adder (from Pre-lab 3) with **two** instances of the 7-segment LED display "decoder" (from Part 2).

- The sum[7..0] outputs from the 8-bit adder should be connected to two instances of the display "decoder". The most significant 4 bits of the adder connect to one display "decoder" and the least significant 4 bits connect to the second display "decoder".
- Perform a functional simulation using the same testbench as in Part 3, but modify it to show both the sum[7..0] outputs from the adder and the outputs from the two display "decoders". However, you don't have to use ASSERT statements for the new display-decoder outputs.
- Note that during the in-lab Step 2 (below), the outputs of the two display decoders will be connected to two 7-segment LED displays on the Altera DE10-LITE board.

Part 4 Pre-lab submission

There is nothing to turn in for Part 4, but to save time in the lab, you should perform all you can to prepare for Step 2 of the in-lab tasks (including pin assignments to switches, LEDs, and 7-segment displays. Your lab TA can help you with any debugging any issues during your lab. However, if you don't do anything beforehand for In-lab Step 2, you may run out of time in the lab.

***** Important: For full credit,** all prelab materials must be submitted to Canvas by the beginning of your scheduled lab time.*

Pre-Lab submission guidelines

Create the following folder structure:

```
your-UFID/
├── P1/
│   └── part1-files-here
└── P2/
    └── part2-files-here
```

Replace "your-UFID" with your actual UFID. Compress the folder into your-UFID.zip and submit it on Canvas.

In-lab Procedure:

Step 1: Stand-alone testing of the 7-segment LED display "decoder" from Part 2 of the pre-lab.

- Select one of the four 7-segment LED displays on the Altera DE10-LITE board:
  - Determine the FPGA pins that are connected to the selected 7-segment LED from the DE10-LITE User Manual.
  - Use Pin Planner to assign the appropriate pins to the "decoder" outputs.
- Select 4 switches on the Altera DE10-LITE board:
  - Determine the FPGA pins that are connected to the selected slide switches from the DE10-LITE User Manual.
  - Use Pin Planner to assign the appropriate pins to the "decoder" inputs.
- Test the display "decoder" using a variety of test vectors. Your TA will ask you to display the results for one or more random test vectors.

Step 2: Test the integrated design from Part 4 of pre-lab.

- Similar to the steps outlined above in Part 1 of in-lab, determine the FPGA pins assigned to the outputs of **both** display "decoders".
- Assign the appropriate FPGA pins on the Altera DE10-LITE board to the 7-segment LED displays.
- Assign the carry-out Cout from the 8-bit adder to an LED.
- You will need switches for 17 inputs; but there are only 10 switches on the DE-10 board. As directed by your TA, "hardcode" 7 of the inputs to '0' or '1' and use the 10 switches for the remaining 10 inputs. Test the 8-bit adder. As before, your TA will ask you to display the results for one or more random test vectors.

1. Lab quiz: your TA will assign a task for you to design a MSI combinatorial component using VHDL, simulate it, and download it to the Altera DE10-LITE board.

Grade Breakdown

Criteria	Pts
---------- Pre-lab ----------	**60 pts total**
Part 1: 4-to-1 MUX VHDL (7.5), testbench (7.5)	15 pts
Functional Simulation	5 pts
Part 2: Display decoder VHDL (7.5), testbench (7.5)	15 pts
Functional Simulation	5 pts
Part 3: 8-bit adder VHDL (7.5), testbench (7.5)	15 pts
Functional Simulation	5 pts
---------- In-lab ----------	**40 pts total**
1. Demo of 7-segment LED decoder **(from Part 2 of pre-lab)**	15 pts
2. Demo of integrated design **(from Part 4 pre-lab)**	15 pts
3. Lab quiz	10 pts