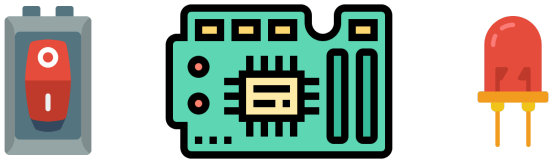
New York City College of Technology
Computer Engineering Technology



Exploring Switches & Lights Using the Altera DE-10 Lite

Lab 003

CET 4805 Component and Subsystem Design II
Section OL40
Professor Hakan Pekcan
Spring 2021
Report Written by Maria Pedroza & Galib F. Rahman

Table of Contents

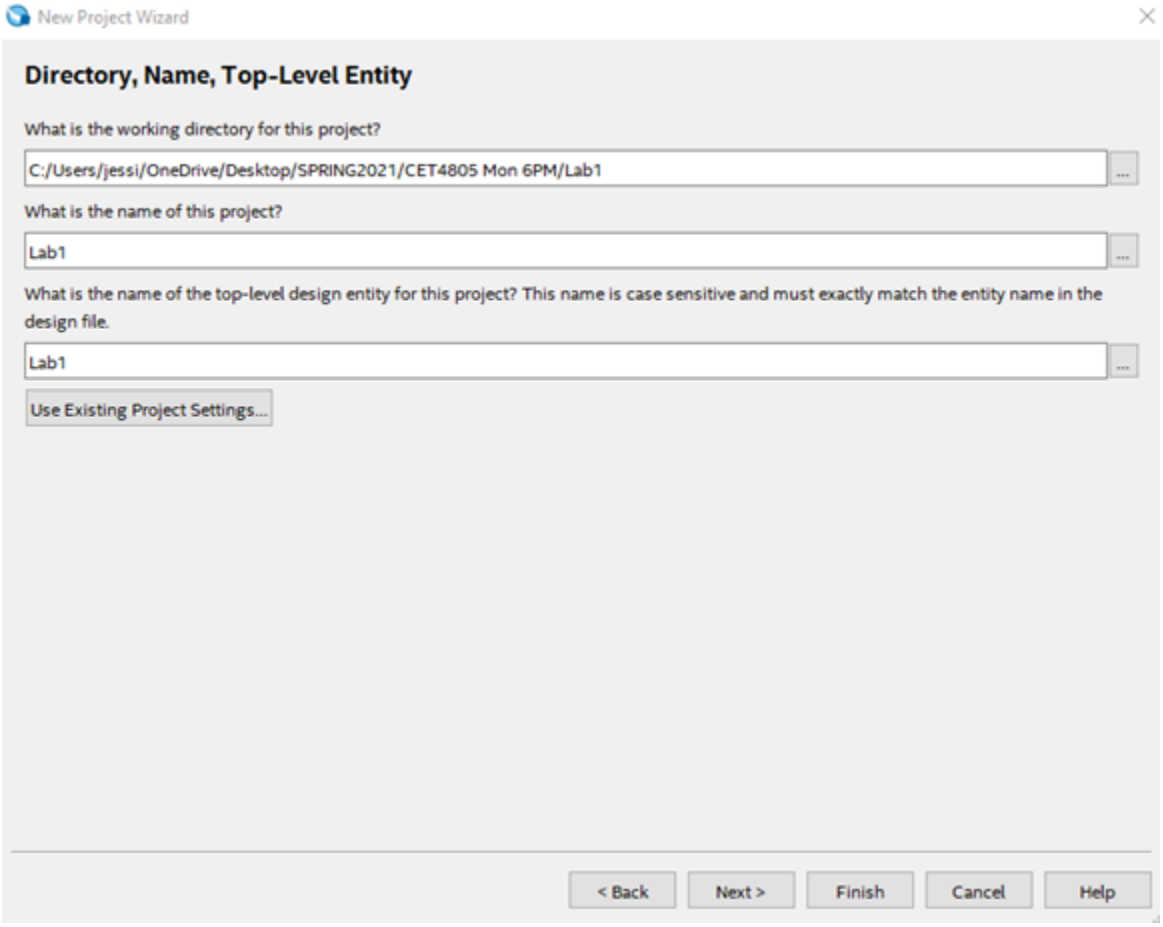
Objective

The objective of this laboratory exercise is to learn to interact with input and output devices using the FPGA chip and implement a circuit using said devices. The primary focus of this lab is to utilize the onboard switches (SW9-0) as input devices and control the onboard LEDs as output devices using VHDL.

Procedure & Results

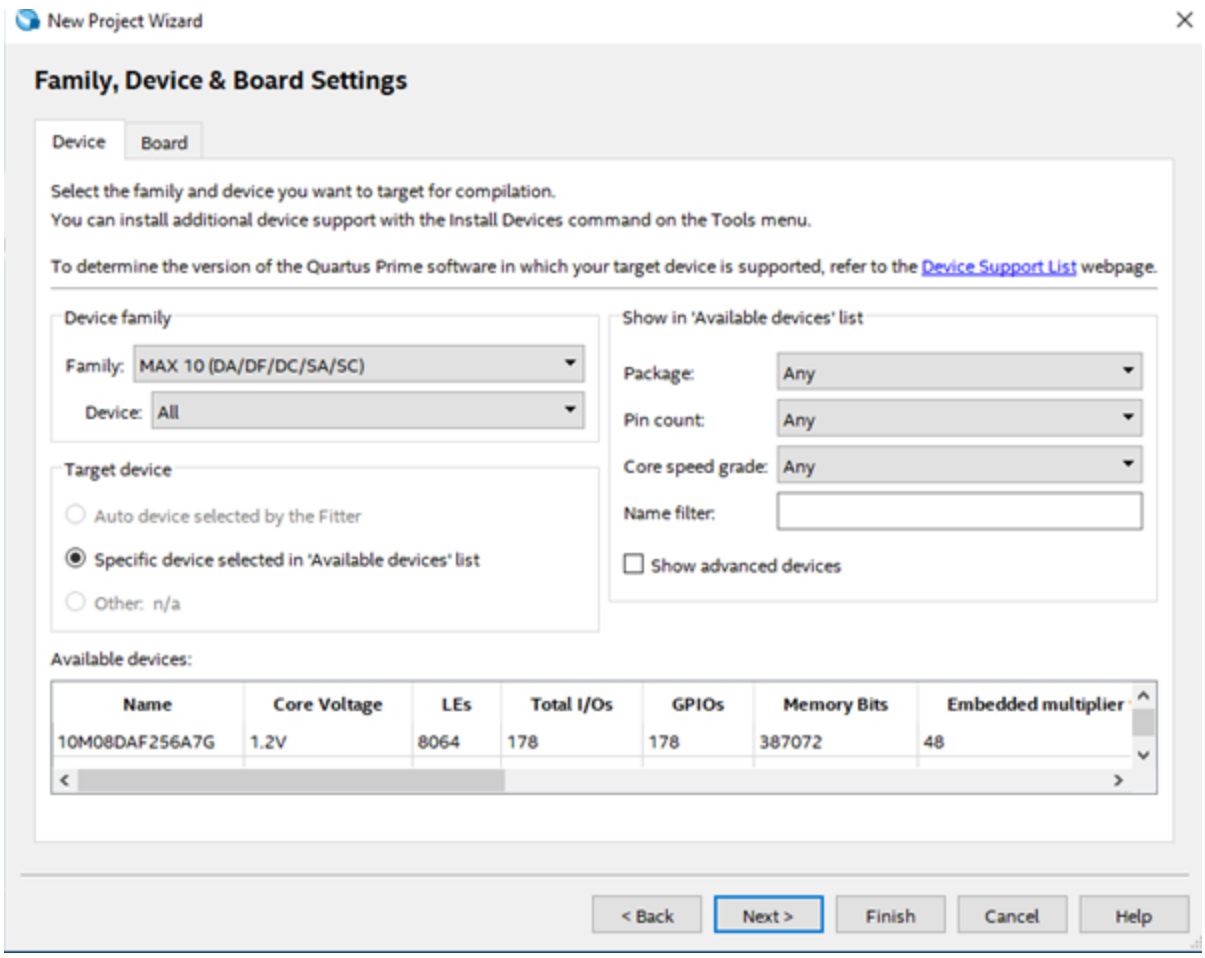
Creating A Project

1. To begin we will use the **New Project Wizard** to create a new project for our circuit.



2. The Project Type can be: **Empty Project**
   Add Files: Click on Next. You do not need to add any file to your project at this point.

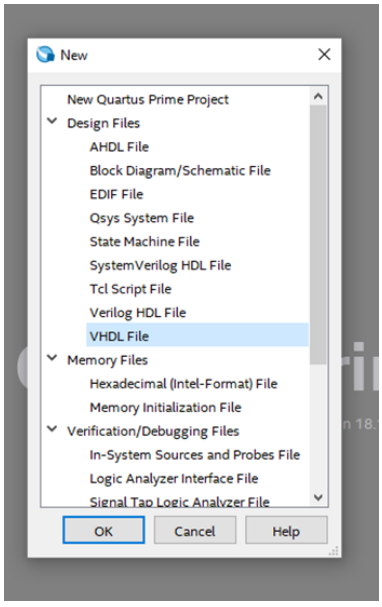
3. You should then see “Famly, Device, & Board Setting” screen



4. Click **Next** and **Finish**

5. Create a VHDL entity/architecture for the code in Figure 1 and include it in your project.

File > New > VHDL file > OK



6. Include in your project the required assignments for the DE10-Lite board, as discussed above. Compile the project. (You can skip pin assignments if you do not have the bord with you at this moment.)

7. Note that you must save your file as **Lab1.vdh** where the TOP Level Entity name **MUST** match your VHDL File name.

8. Download the compiled circuit into the FPGA chip. Test the functionality of the circuit by toggling the switches and observing the LEDs. (If you do not have the board with you, you should pore the design by a Waveform Simulation.
9. Select **New > File > University Program VWF**
10. Click on Left pane, right-click on **Insert Node or Bus > Node Finder > List.**
11. Bring any signal you would like to observe to the Right ane by highlighting the name and pressing > arrow
12. On the Waveform Editor Window assign input **Values** to input signals and run Functional Simulation.

Task 1: Compile and Simulate

Modify the VHDL code to do the following:

Any switch that is going to be switched ON will turn on the next LEDR instead.
For example, if you switch on SW6, it should turn on LEDR7. (Switch, 9 might turn on LEDR0)

The VHDL code to this would look like

```
LEDR(7) <= SW(6);
```

Alternatively, you may write a couple of statement for all such as:

```
LEDR(9 downto 2) <= SW (8 downto 1);
LEDR(1) <= SW(9);
```

Task 2: Compile and Simulate

Modify the VHDL code to do the following:

Turning on any switch numbered between 0 and 4 itself would turn on upper 5 LEDRs: LED(9:5)

Otherwise:

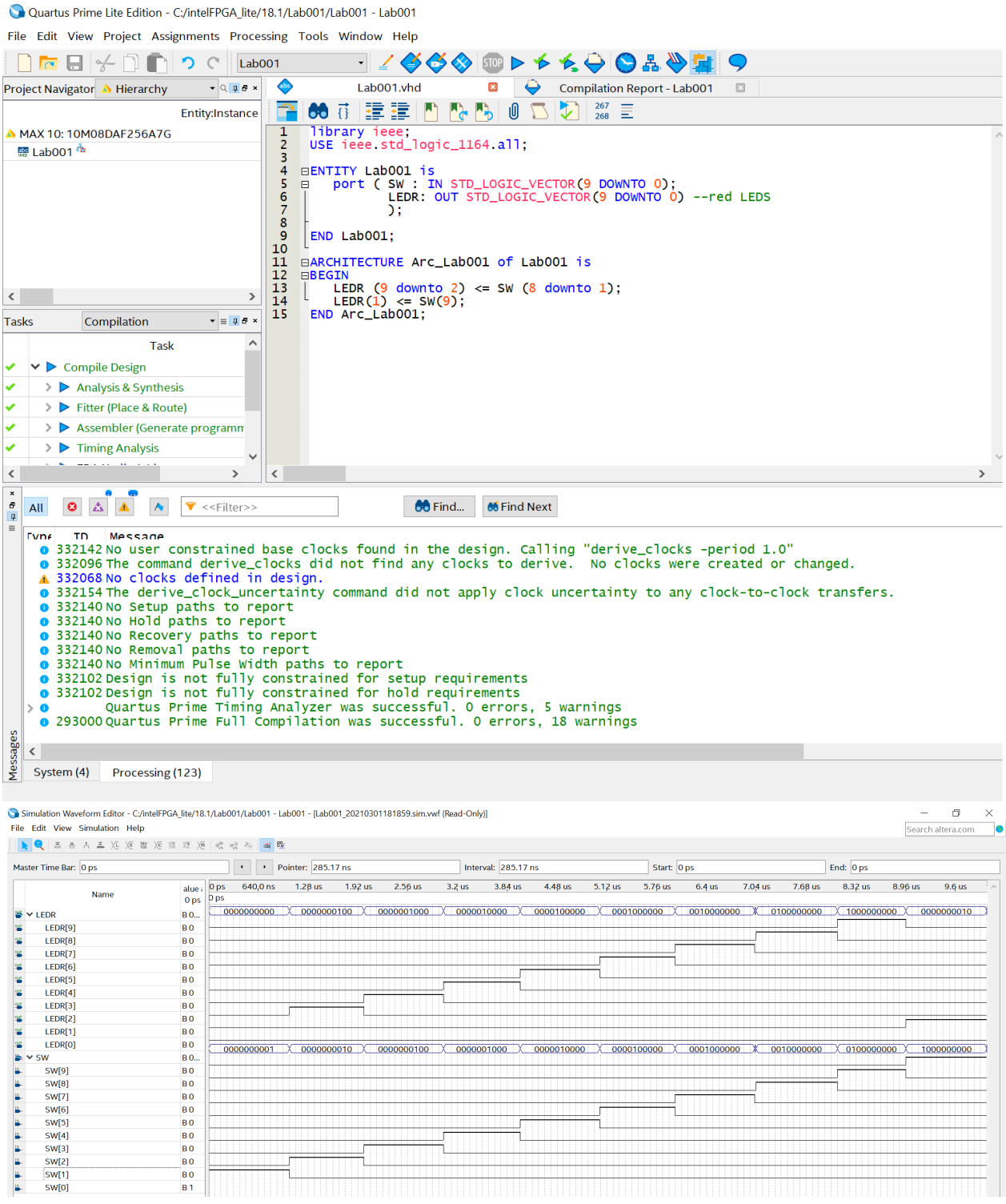
Turning on any switch numbered between 6 and 9 by itself would turn on first 5 LEDRs: LEDR(4:0)

Any other combination of switches will turn on ALL LEDRs.

Example code for the architecture section of VHDL file is: Complete the file below.

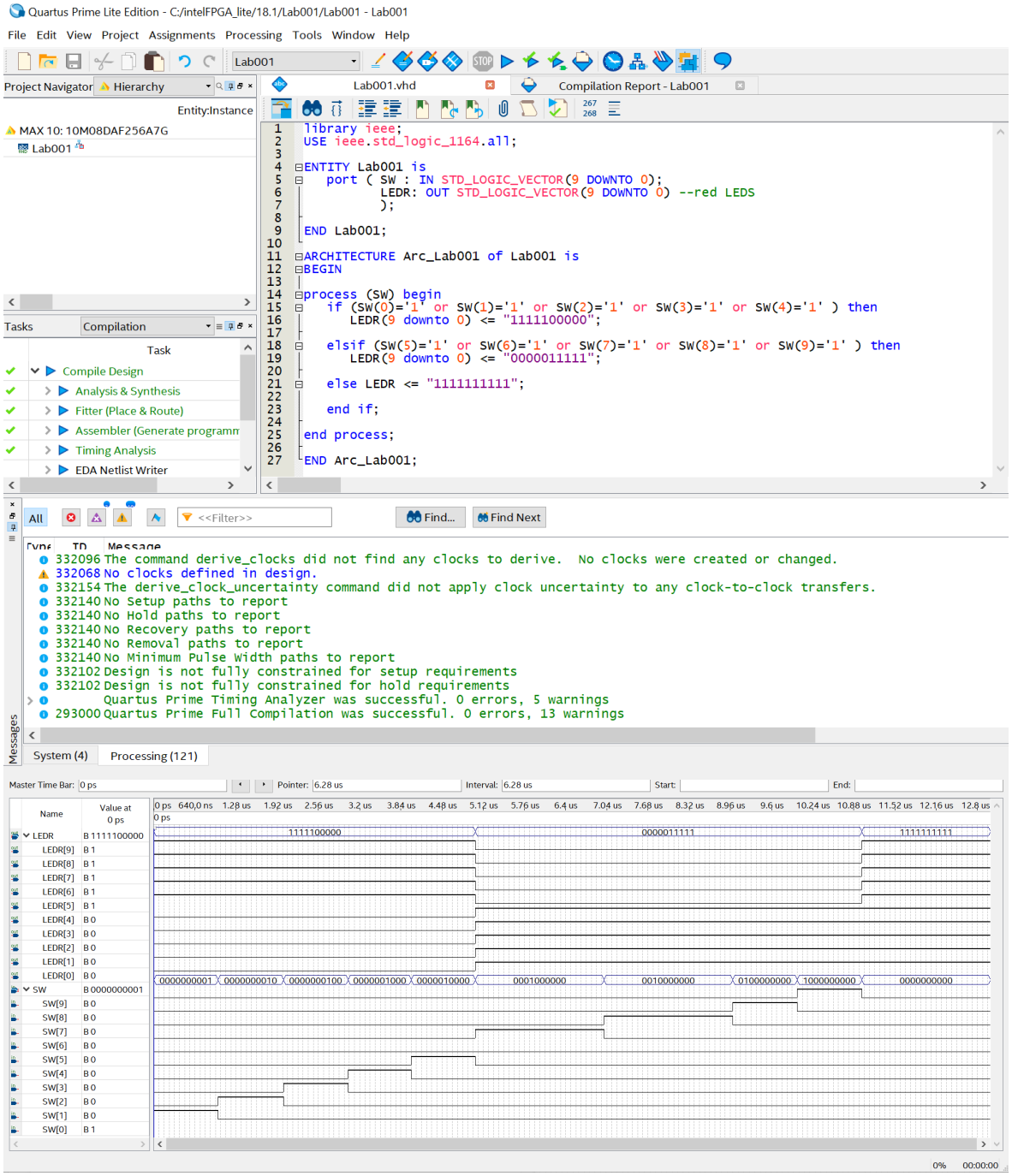
```
Process (SW) begin

    if (SW(0) = '1' or SW(1) = '1' or SW(2) = '1' or SW(3) ='1' or SW(4)= '1') then
        LEDR (9 downto 0) <= "1111100000"; -- turn on upper LEDRs

    else if (SW(5) = '1' or SW(6) = '1' or SW(7) = '1' or SW(8) = '1' or SW(9)= '1')
then
        LEDR(9 downto ) <= <fill here> --turn on lower LEDRs

else
    LEDR <= <fill here> ; --turn on all

end process;
```

Task 1: Compile and Simulate



This VHDL source code is designed such that the switch activates the succeeding LED with respect to the denoted number of said switch. In the simulated waveform we observe this output (e.g. when SW[4] is HIGH LED[5] is HIGH and all else is LOW)

Task 2: Compile and Simulate



This VHDL source code is designed such that if any one switch 0-4 is activated the upper 5 LEDRs 9:5 are HIGH and if any one switch 6-9 is activated LEDRs 4:0 are HIGH and all other combinations result in the activation of all LEDRs. This behavior is demonstrated in our simulated waveform. The waveform activates each switch individually and then all switches are deactivated (or LOW). The observed output correctly matches the objective of the source code.

Conclusion

Upon completion of this laboratory exercise, my partner (Galib Rahman) and I (Maria Pedroza) have successfully utilized the Quartus® Prime software to control the onboard LEDs as output devices by modifying the VHDL code. We utilized the onboard switches (SW9-0) and modified assigned input values, shown as input signals ,using the Waveform Editor Window and observed the output waveforms to verify the proper functionality of the VHDL source code.

Intel® Quartus® Prime Program Files