Lab 1: Introduction to EEL4712 Digital Design Lab

EEL 4712 – Fall 2023

Objective:

The objective of this lab is to introduce the software and hardware development tools to be used in EEL 4712 to design, construct, and test digital circuits. In particular, you will review the use of the Quartus/ModelSim software package for the synthesis of a digital design. You will be provided an 8-bit counter using the schematic-capture features of Quartus (.bdf file), and will create a 4-bit ripple-carry adder using VHDL. Both designs will be synthesized in Quartus and simulated in ModelSim (Altera edition). In the lab, you will be introduced to the Terasic DE10-lite board.

Required tools and parts:

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

Pre-requisite:

- You should be "up-to-speed" with Quartus and ModelSim before coming to lab (i.e., completed Lab 0).
- You should understand and have performed pin assignments for the DE-10-lite board using the following resources: Pin assignment instructions and Pin assignment video tutorial
- Also helpful: download and read the DE10-lite documents before coming to lab.

Other helpful resources:

- ModelsimQuickStart2023.pdf
- Tutorials for ModelSim V6.6, V10.4
- DE10-lite Board documentation

Pre-lab Requirements:

Part 1: Simulation of an 8-bit counter design (using the provided counter.bdf file)

(a) Create a Quartus project and compile the counter design

- Download the provided "counter.bdf" file and put it in a project directory of your choice.
- Create a new Quartus project (see details in **Section 1** of the **Quartus and ModelSim Quickstart guide of Lab 0**).
  - Select any one of the **MAX II** FPGAs. Note that MAX II is not the FPGA on your DE10-Lite board. But Quartus does not support timing simulations for the MAX 10. It does support the MAX II.
  - On the EDA Tools Settings screen, under "Simulation", select "ModelSim-Altera" as the simulator and choose VHDL as the format.
- Add the counter.bdf file to your project. Use **Method 2 in Section 2** of the Quickstart guide of Lab 0.
- Compile the design (**Step 4** of Quickstart guide).

(b) Perform functional simulation using ModelSim-Altera

- Follow the instructions in **Section 5** of the Quickstart guide, except you add the counter.vho and counter_tb.vhd files to the ModelSim project.

- Compile the project ModelSim-Altera (**Section 7** of Quickstart guide).
  - Perform a **functional** simulation for the circuit (**Section 8** of Quickstart guide). A functional simulation ignores the timing of the device and assumes that all signals update simultaneously (i.e., all propagation delays are 0).

(c) Perform timing simulation using ModelSim-Altera

- Perform a **timing** simulation for the circuit (**Section 9** in Quickstart guide). Note that in the resulting waveforms, the outputs are delayed by a small amount, instead of changing immediately on every rising clock edge.

Part 1 Pre-lab submission (on Canvas) before your scheduled lab

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

- Screenshot of functional and timing simulations
  - Should show "Simulation Finished" in transcript window.
  - Should show 0 assertion errors.

Part 2: Design and simulation of a 4-bit ripple-carry adder: VHDL specification

(a) Full adder

- Create a full adder entity in VHDL, using the template provided on the website (fa.vhd). Do not change the names of any port signals.
- Perform a **functional** simulation of the full adder in ModelSim using the provided testbench (fa_tb.vhd).

(b) 4-bit ripple-carry adder

- Create a 4-bit ripple-carry adder in VHDL using a structural architecture (i.e., PORT MAP statements) that combines four full adders into a 4-bit ripple-carry adder. Make sure to use the template provided on the website (adder.vhd) and do not change any of the port signals.
- Perform a **functional** simulation of the ripple-carry adder in ModelSim using the provided testbench (adder_tb.vhd).
- Perform a **timing** simulation of the ripple-carry adder.
  - Note that in order to perform the timing simulation, you should have already created the .sdo file (**Section 4** in the Quickstart guide, using the instructions in the blue box).
  - Add the .sdo file to the simulation (**Section 9** of Quickstart guide)
  - Use the same testbench as the previous step.

Part 2 Pre-lab submission (on Canvas)

Include the following content in **`your-UFID/P2`** directory:

- All VHDL files for Part 2
  - adder.vhd
  - fa.vhd
- Screenshots from each simulation

Part 3: Downloading Part 1 design to the DE10-Lite

To download and run the Part 1 counter on your board:

1. Create a new Quartus project with the following options:
   a. On the "Family & Device Settings" page, select the "Board" tab at the top. Select the MAX 10 DE10-Lite board.
   b. Unselect the "Create top-level design file" option.
2. Add your counter files of Part 1 (.bdf) and compile them.
3. Assign pins via the Quartus Pin Planner. (See how in Pin assignment instructions and Pin assignment video tutorial)
   a. Make sure all your unused pins are reserved "As input tri-stated" when assigning pins.
      i. This option can be found under "Assignments > Device > Device and Pin Options > Unused Pins".
      ii. **IMPORTANT: Remember to do this for every project you create**.
   b. Assign all inputs (except for the clock) to the switches and/or buttons.
   c. Assign the clock to one of the pushbuttons in the DE10-Lite user manual. Again see Pin assignment instructions and Pin assignment video tutorial.
      i. The pushbuttons are debounced, but the switches are not, so be sure to use a pushbutton for your clock.
   d. Assign combinational outputs to the LEDs on the board.
4. You need to re-compile again after assigning pins.
5. Download the 8-bit counter from the Part 1 pre-lab to your board using the Quartus Programmer tool. If you want to do this step before coming to lab, you can go to the office hour of a TA. Otherwise, your lab TA will demonstrate and help you with any issues during your lab.

Part 3 Pre-lab submission

There is nothing to turn in for Part 3 for pre-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

1. Present the following simulations of the counter in ModelSim:
    a. Functional Simulation
    b. Timing Simulation
2. Present the following simulations of the adder in ModelSim:
    a. Functional simulation
    b. Timing Simulation
3. Step 5 from above to download to board, debug and then do in-lab demonstration on the DE10-Lite board. You may be asked to make minor changes or extensions to your design during your demonstration.

Grade Breakdown

Criteria	Pts
---------- Pre-lab ----------	
Counter .bdf	7.5 pts
Counter Func Sim	5 pts
Counter Timing Sim	5 pts
FA .vhd	7.5 pts
FA sim	5 pts
Adder .vhd	10 pts
Adder Func Sim	5 pts
Adder Timing Sim	5 pts
---------- In-lab ----------	
Counter Func Sim Live demonstration of the Counter functional simulation	10 pts
Counter Timing Sim Live demonstration of the Counter timing simulation	10 pts
Adder Func Sim Live demonstration of the Adder functional simulation	10 pts
Adder Timing Sim Live demonstration of the Adder timing simulation	10 pts
In-lab demonstration on the DE10-Lite board	10 pts