EECS 3201 :: Lab 1
Introduction to the DE10-Lite

Overview

In this lab you will:

- Download and install the Quartus Prime Lite software;
- Use Quartus Lite to create a simple Verilog module;
- Configure the module for the DE10-Lite;
- Program the DE10-Lite with your module; and
- Demonstrate that the program works correctly.

Deliverable:

- A short video of yourself correctly demonstrating the program on the DE10-lite.

Scoring:

- You will receive full marks for submitting the deliverable. Grades might be deducted for minor errors.

Value:

- This lab is worth 2/20 of your total lab score.

Quartus Prime Lite – Download and install

Go to https://fpgasoftware.intel.com/ and on the “Select edition:” download, pick “Lite”. On “Select Release:”, it should show “20.1.1” after you pick the “Lite” edition. (Don’t pick the other editions as they require a paid license, while Lite is free.)

Quartus Prime Lite is available for Windows and Linux. Select your OS and click the download button. It may ask you to register first, in which case you will need to create an account and then try again. The Lite software package is approximately 6 GB in size.

> *If you only have a Mac, you can run Quartus in a virtual machine: install VirtualBox (free), install Ubuntu in the VirtualBox, and then install Quartus on the virtual Ubuntu. See here for details: https://siytek.com/quartus-mac-virtualbox-ubuntu/ (note that the instructions at the link are for older versions of Quartus and Ubuntu, and the current versions have higher memory and disk space requirements)*
>
> ***If you have a new Mac with an M1 processor, VirtualBox will not work.*** *You can try other virtual machine software such as UTM (free) or Parallels (paid). I don’t have access to an M1 Mac so I have not tried these.*

First Quartus project – A simple logic gate

Open Quartus and select the New Project Wizard.

- "What is the name of the project?" – type "Lab1" (no space). To keep things organized, you might want to specify a directory for the project; this is optional. Click "Next".
- Select "Empty project" and click "Next"
- Add files – You don't want to add any files yet – just click "Next"
- Family, Device & Board settings:
    - At the top of the page, select the "Board" tab
    - From the "Family:" dropdown, select MAX 10
    - Under "Available boards:" select "MAX 10 DE10 – Lite"
    - Click "Next"
- EDA Tool Settings – we won't use these settings in this lab – just click "Next"
- Click "Finish" on the summary page

At top left you will see the project navigator. Click the dropdown (it might say "Hierarchy") and select "Files". For some reason a file named "DE10_Lite_Golden_Top.v" has been added to the project by default. We don't need this file, so right-click on it and select "Remove file from project".

Now we are ready to add a Verilog module to the project. From the menu select "File" -> "New", select "Verilog HDL File", and click OK. From here you can add whatever you want in Verilog; you can also organize your project into different Verilog files. We are simply going to create a single NAND gate. Add the following Verilog code:

```
module Lab1(a,b,c);

      input a,b;
      output c;

      assign c = ~(a & b);

endmodule
```

(What happens in this file? The tilde ~ means logical NOT, and the ampersand & means logical AND. The value of a & b is True if both a and b are True, and False otherwise. Thus, ~(a & b) inverts the result, i.e., False if both a and b are True, and True otherwise. This operation is Not-AND, or NAND.)

(Also notice that every line gets a semicolon except endmodule.)

Save the file. Module names should match the file name, so ensure this file is saved as "Lab1.v".

Now we need to designate the top-level entity in this project, kind of like the main() method in a Java program. In the project navigator, under “Files” right-click on Lab1.v and select “Set as top-level entity”.

Finally, we can compile the design to ensure that our Verilog is correct. Below the project navigator you will see the “Tasks” window, which should have “Compilation” selected from the dropdown. (If not, select it.) Double-click on the blue triangle next to “Compile Design” to do a full compilation. If you get a green checkmark, hooray! If not, correct any errors.

Configuring the project for the DE10-Lite

The Verilog module you wrote has two inputs (a, b) and one output (c). Now you must assign them to input and output elements on the DE10-Lite board.

After compiling, Quartus will have found the input and output pins in your project, though they are not assigned to anything. We want to assign the inputs to two of the DE10-Lite’s switches, and the output to one of the DE10-Lite’s LEDs.

From the menu, select Assignments -> Pin Planner, which gives you a map of the MAX 10’s pins. At the bottom you should see a list. The pins a, b, and c should be at the top of this list, and they should not have any assignments. (If you can’t see them, go to the “Filter:” dropdown at the right, and select “Pins: unassigned”.)

The two rightmost switches on the DE10-Lite are labelled PIN_C10 and PIN_C11 (note, the pattern doesn’t continue, the next switch isn’t PIN_C12!). The rightmost LED on the DE10-Lite is labelled PIN_A8. For a complete list of pins for all accessories on the board, see the DE10-Lite user manual.

For pin a, double-click on the Location box. You can now type PIN_C10 as the location, or you can select it from the dropdown. (You might get an error that the pin is already assigned, or it might be missing from the dropdown list. In this case, go to Assignments -> Remove Assignments, check “Pin, Location & Routing Assignments”, click OK, and try again.) Do the same to assign PIN_C11 to b, and PIN_A8 to c.

Re-compile the design. Again, you should get green checkmarks.

Program the DE10-Lite with your design

At the bottom of the Tasks window you should see “Program Device (Open Programmer)”. Right-click on that and select “Open” to open the programmer.

Plug your board into any USB port, and click “Hardware Setup …”. Under the “Hardware Settings” tab, you should see a list of “available hardware items”, which should include “USB-

Blaster”. On the “Currently selected hardware” dropdown, select “USB-Blaster”, then close the setup window.

> *If you are running Ubuntu, and you see “USB-Blaster Variant” instead of “USB-Blaster”, then you don’t have the correct permissions for the USB port. This can be fixed as follows:*
>
> - *Unplug your device from the USB port*
> - *Log in as root (sudo -s)*
> - *Create a new file with the name /etc/udev/rules.d/90-usb-blaster.rules*
> - *In the new file, write the following line:*
>
>   ```
>   SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6001", MODE="0666"
>   ```
> - *Save and close, then plug your device back into the USB port. It should now work.*

You should now see a single line representing your output file, along with checkboxes for “Program/Configure”, “Verify”, etc. Right-click on this line, select “Change file”, navigate to the “output_files” directory, and select “Lab1.sof”.

Ensure the “Program/Configure” checkbox is checked, and then click “Start”.

If the “Progress” bar reaches “100% (Successful)”, the device should now be programmed.

Demonstrate your program

Take a video of yourself with your correctly programmed board. Your video should include yourself stating your name and student number, and then (with no cuts/breaks) a demonstration of the program. Demonstrate the program by going through all four possible configurations of the rightmost two switches, showing that the LED only goes out when both switches are “on”.

Submit your video via eClass.

Grading Rubric

You will receive full marks if your demonstration video exactly matches the description above.