Quick Start Guide

By: Jongse Park
Date: September 26, 2016

To help you set things up so you can use the board, here's a list of things you'll need to do. Most of this comes from the manuals on the vendor's website, so this is more of a "quickstart" description than a step-by-step guide. For the detailed guide, find the board's manuals in https://www.altera.com/support/training/university/boards.highResolutionDisplay.html#de0-cv

1. Get a MS-Windows environment. We verified that Windows 10 Pro and Education are compatible with the hardware and software that we will use in this course. Older versions of Windows should not have any problems but we do not recommend you to use them since we did not verify their compatibility. You may need to download an installation image file of Windows from Microsoft Imagine (https://imagine.microsoft.com/). The TSO webpage, which is located in https://support.cc.gatech.edu/resources/downloads, shows how you can create a Dreamspark (Imagine) account and download Microsoft products. If you use Mac, you can use Boot Camp (https://support.apple.com/en-us/HT201468) and if you use Linux, you can install Windows alongside with Linux with Dual Boot. You can also install virtualization software such as VirtualBox to use virtualized system environment and then install Windows system on it.
2. (Optional) Install useful tools. There are lots of editors that support Verilog but we recommend one of them, TextPad. By default, TextPad does not support syntax highlighting for Verilog but you can enable TextPad to support Verilog with a few steps of configuration.
    - Go to Configure → New Document Class
    - Type "Verilog" at the textbox of "Document class name:".
    - Type "*.v" at the textbox of "Class members:"
    - Check the checkbox of "Enable syntax highlighting" and select "verilog.syn" in the list of "Syntax definition file".

    You may also need to install Git to manage your codes. You can find it in http://gitscm.com/download/win.
3. Get Quartus II Web Edition 13.0sp1 from Altera, which is located in http://dl.altera.com/13.0sp1/?edition=web&platform=windows&download_manager=direct. Note that we will not be using the latest version which is 16.0; thus, you should select and download the correct version at "Select release," which is the 13.0sp1. Then the easiest thing to do is to follow instructions for the "Combined Files" tab. If you are short on disk space you can also use the "Individual Files" tab – at the minimum, you'll want to install the Quartus II Software, Modelsim, and the Cyclone V device support.
4. Install Quartus
    - Extract the downloaded *Quartus-web-13.0.1.232-windows.tar* file and then double-click *setup*.
    - The installation will take quite a long time.
5. If using an existing QAR file (e.g. Lectures.qar), you can skip steps 6-8.
6. Create a new project
    - When you start Quartus, select File → New Project Wizard in the menu.
    - Select a working directory (or just use the default Quartus directory).
    - Give the project and its top-level entity a name (e.g. SwitchToLedR), then click Next.
    - Do not add any files (Next).
    - Select Cyclone V Family (Device family -> Family -> Cyclone V (E/GX/GT/SX/SE/ST))
    - Type 5CEBA4F23C7 on "Name Filter" to select our FPGA device (Cyclone V family, 5CEBA4F23C7N device)
    - Click Finish.
7. Take care of the pin assignments

- Download the pin assignment file at the Altera supported board materials webpage, which is located in ftp://ftp.altera.com/up/pub/Altera_Material/Boards/DE0-CV/DE0_CV.qsf. You can find and download a Quartus Setting File with Pin Assignments for DE0-CV. Save this *DE0_CV.qsf* somewhere.
- In Quartus, go to Assignments → Import Assignments and import the *DE0_CV.qsf* file.
- Go to Assignments → Device, then click on the "Device and Pin Options" button. In the dialog that pops up, go to the "Unused Pins" category, change "Reserve all unused pins" to "As input tri-stated".

8. Enable the design assistant
    - Go to Assignments → Settings, then look at the "Category" list on the left side of the dialog.
    - Select the "Design Assistant" category, check the "Run Design Assistant during compilation" box.
9. Create a Verilog file with your code. Go to File → New, select "Verilog HDL file" (from "Design Files") in the dialog box. Write something in there and when you try to save there will be a dialog box, choose the file name (e.g. SwitchToLedR.v), make sure the "Add file to current project" is checked, and find a directory where you want the file saved (e.g. the project directory) but in any case remember where the file is saved (you will need to find it so you can upload it on T-Square).

    [Test Code] SwitchToLedR.v

    ```
    module SwitchToLedR (SW, LEDR);
          input[9:0] SW;
          output[9:0] LEDR;
          assign LEDR = SW;
    endmodule
    ```
10. Write Verilog code for a circuit you want, save it, and compile it. To compile, click on the "Start Compilation" button on the toolbar (the button has what looks like a purple "Play" button from a music player), or select Processing → Start Compilation in the menu.
11. Connect the board (using the USB cable). When you connect the board, the system will ask for the driver if the driver was not successfully installed in Step 4. Tell the system to look in "C:\altera\13.0sp1\quartus\drivers\usb-blaster" for the driver (this path may be different if you installed Quartus in another directory).
12. Turn the board on and program it:
    - To turn the board on, connect the power cord to the power jack on the board and push the red button. The board should start doing its thing – all the FPGA user HEXs and LEDs will be flashing.
    - To program the board, click the "Programmer" button in Quartus (one of the last buttons on the toolbar, looks like some lines entering a blue chip), or select Tools → Programmer in the menu.
    - When the programmer dialog opens, make sure the "Hardware Setup" says "USB-Blaster [USB-0]" (if it doesn't, click the "Hardware Setup" button and select USB-Blaster [USB-0] there) and that the mode is set to "JTAG". The file should be SwitchToLedR.sof (or whatever the design name is, with a .sof extension). You can add this file by clicking "Add File…" and selecting a sof file (e.g., SwitchToLedR.sof) under the output_files directory.
    - Make sure that the checkbox for "Program/Configure" is marked.
    - Click the "Start" button in the programmer dialog. That should do it – the board's behavior should change from its default (just flashing) to whatever you programmed it to do. In our SwitchToLedR example, the LED will be lit when the corresponding switch is ON.
13. Run ModelSim from Quartus
    A. Implement a testbench file for your design.

    [Test Code] TestSwitchToLedR.v

    ```
    `timescale 1ns / 1ps
    module TestSwitchToLedR ();
          reg [9:0] sw;
          wire[9:0] ledr;
    ```

```
        SwithToLedR stlr (sw, ledr);
        initial begin
              sw = 10'b0000000000;
              #10
              sw[3] = 1;
              #10 $finish;
        end
endmodule
```

B. Find the path for ModelSim executable
   - Under "Tools" -> "Options...", select the "EDA Tools Options" category
   - Under Modelsim-Altera, enter the path to the modelsim altera directory
   - E.g., C:\altera\13.0sp1\modelsim_ase\win32aloem

C. Configure the simulation settings
   - Under "Assignments" -> "Settings", select the "EDA Tool Settings - Simulation" category
   - Under "Tool Name" select Modelsim-Altera
   - Under "Format for output netlist", select "Verilog HDL"
   - Under "Native Link Settings", enable "Compile Testbench" and add the testbench file that you implemented in the Step 1

D. Run simulation
   - Under "Tools" -> "Run Simulation Tool", click "RTL Simulation" to run the simulation
   - The ModelSim will be automatically started and you will see the results

E. Other materials
   - You can find other useful materials from Hadi's course webpage: http://www.cc.gatech.edu/~hadi/teaching/cs3220/03-2016fa/resources.html

14. Run ModelSim without Quartus
    A. Open ModelSim
    B. On menu bar, click "New" and select "Project"
    C. Find your project directory that contains your Verilog codes including the testbench code. Set the path to the "Project Location" through "Browse". Assign a project name and click "Ok".
    D. When the "Add items to the project" window pops up, click "Add Existing Files" and select your Verilog files including the testbench file.
    E. Under the menu bar, select "Compile" → "Compile All". You may be able to selectively compile the files you want with "Compile Selected". You will see green check marks on the status column of the files if the compilation was successful.
    F. Select "Library" tab right next to the "Project" tab. You will see a list of libraries. Under "work", find the testbench module and double-click it.
    G. Select your module under "sim - Default". Click "Add" in the menu bar", select "To Wave" → "All items in region".
    H. Run simulation. Click "Simulate" → "Run" → "Run 100". You can also change the simulation time. In the toolbox, there is a textbox that by default has "100 ps". You can change this simulation time to something else such as "200 ns" and rerun the simulation.