Lab #1: Getting Started with Digital Logic

1. Introduction

The aim of these labs is to provide a hands-on introduction to digital logic design using the **Quartus Prime** suite of design tools and the Intel **DE10-Standard** FPGA board (shown below in Fig. 1) on which your designs will be implemented. The general digital design procedure is as follows:

1. Within Quartus, create and configure a '**Project**', or copy and modify another project.
2. A **Project** can contain several design files which you **create** either as **schematic** circuit diagram(s) (Labs 1,2) or in **VHDL** code (Labs 3 onwards). You can also **add** previously created design or provided files to your project.
3. **Compile** your design and correct any errors
4. In many cases it may be necessary to specify how the inputs/outputs of your design are connected to the physical input/output **pin assignments** of the DE10 board, after which you may need to **re-compile**.
5. Optionally **Simulate** your design (e.g. in Lab 3) using the **ModelSim** simulator.
6. '**Program**'/download your design to the DE10 board.
7. **Test** to check that your design works as expected (see Fig. 1):
   - The **switches** act as 1-bit logic 0-1 inputs (up=1, down=0)
   - The **led's** or segments of the **7-segment displays** act as logic outputs (1=on, 0=off)
   - The **pushbuttons** (used in later labs) generate 0-1 rising edge, or 1-0 falling edge signal generators.

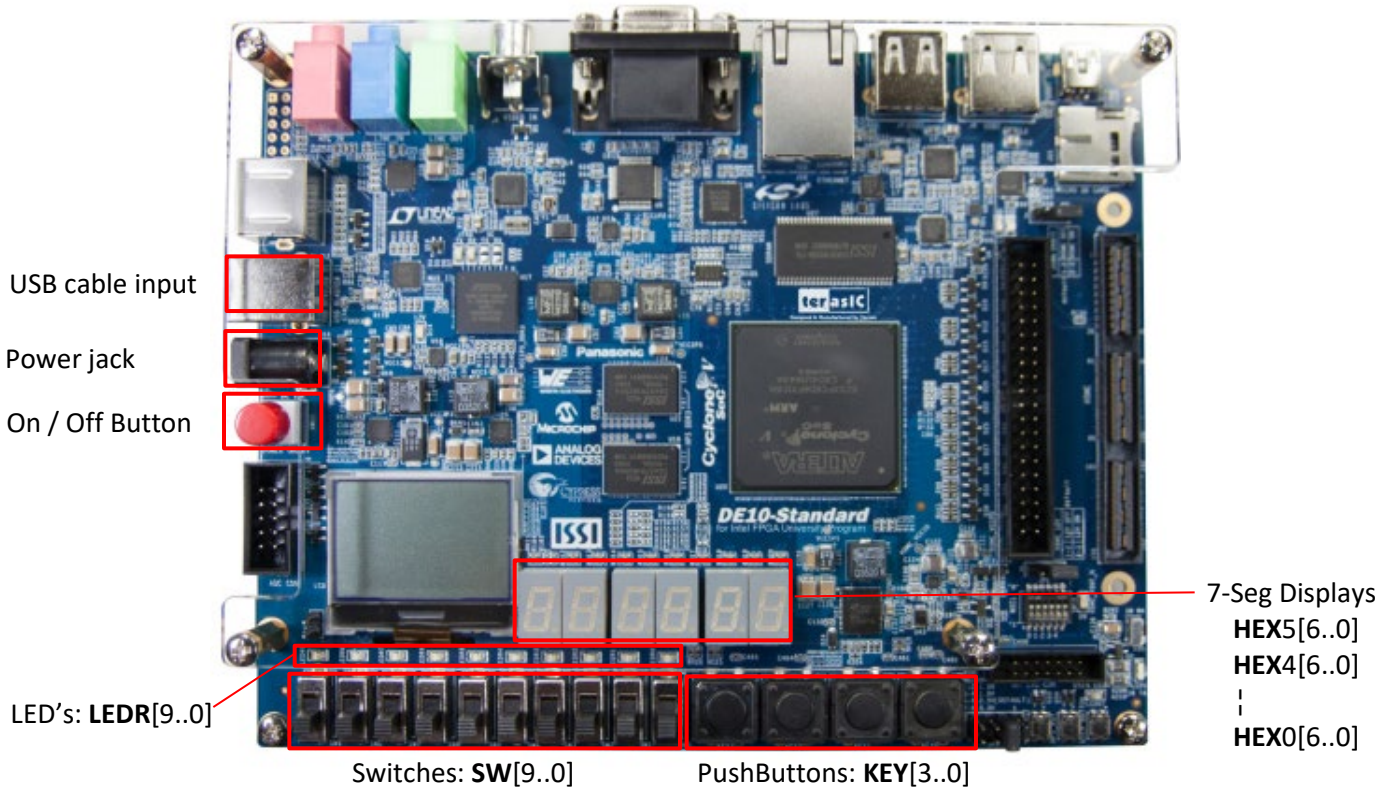

Fig. 1: The DE10-Standard Board

1.1 Getting Started (myFirstDesign)

The aim of this section is to walk you through the general design procedure by getting a design successfully compiled and running on your DE10 board ☺!

1. Make a 'working' directory, e.g. **ECE 2043**, on the H drive of the lab computer. Store all your lab work in this folder this semester.

   If this is the first time you are using the current computer, click on the shortcut "**ece-students-share"** on the desktop to map the H drive. DO NOT save your work on the C drive of lab computers. You will lose your work when they are reimaged. **DO NOT use white space in any name you use for folders, files, projects, designs, etc.**

2. Download the **Lab1.zip** file from **Blackboard** to this directory and extract it, thereby creating a **Lab1** subfolder.

3. From the start menu, start **Quartus** Prime 20.1 Lite Edition which brings up the Integrated Development Environment (IDE). You may also find a shortcut for Quartus on the desktop.

4. If you want to learn how to **configure** a new project from scratch, check out the instructions in Lab 2.4. For now, simply click on the **Open Project** icon in the center of the screen, or go **File->Open Project**. Navigate to the **Lab1** directory you just unzipped, and select the **Lab1.qpf** file (*.qpf = Quartus Project File).

5. A Project may contain several design files that you have created and added to the Project to solve a given problem (e.g. circuit diagrams *.bdf files, or VHDL code *.vhd files). Select **Files** in the Top-Left **Project Navigator** pane pulldown menu to see a list of these files.

6. In this case there is only one file called **myFirstDesign.bdf**. Double-click on the file name to open it if it is not already open, or select **File->Open**. This is a very simple example where the input from switch SW[0] is piped directly to output led LEDR[0], and also inverted and piped to LEDR[1]. Check you understand what it does.

7. Now we need to define **Pin Assignments** which map the input/output names we have used in our design such as SW[0], LEDR[0], LEDR[1] etc. to the physical pin connections on the DE10 board. First clear any old assignments by selecting **Assignments->Remove Assignments**, check the **Pin, Location** etc. **checkbox** and hit **ok**. Then select **Assignments->Import Assignments** and browse to your downloaded Lab1 **DE10-pins.qsf** file and hit **ok**.
   - Note that Pin numbers now appear adjacent to the names (like SW[0], and LEDR[0], LEDR[1]) that we used in our design. This indicates that **Pin assignments** have been successfully made.

8. **Important**: Before compiling your project, return to the **Project Navigator->Hierarchy** tab, and ensure that:
   - The hierarchy correctly specifies a **CycloneV: 5CSXFC6D6F31C6** device (the hardware for our board). [To change the Device, **R-click -> Device** on the currently listed Device, and in the window that appears select **5CSXFC6D6F31C6** (6th from the end!) in the list of **Available Devices** (lower pane of window)].
   - The hierarchy also specifies the correct '**top level entity**' design file from among the (possibly many) that you have created and added to the Project (otherwise you will compile the wrong thing!).
     [To change the top level entity, return to the **Project Navigator->Files** list, select the desired file and either **R-click->Set As Top Level Entity** or (shortcut) **Ctrl+Shift+V**]

9. Now **Compile** your project either by clicking the '**Play**' icon, or **Ctrl+L**, or select **Processing->Compile** from the main menu. This process can take quite a while so be **patient**! Progress is reported in the left-hand **Task** pane, and errors or warnings (which are generally ok) appear in the lowermost **Message** / Cmd pane.
   - Sometimes Timing errors are reported, but they generally go away if you try compiling a second time.
   - You can close the Compilation **Report tab** once compilation is complete.
   - The compiled file will appear in a new **output_files** subdirectory as a *.sof file, e.g. **Lab1.sof**
10. We now want to download / **Program** the ***.sof** file to your DE10 board to implement it in hardware.
   - Connect the **DE10** board and **power on** as indicated in Fig. 1.
   - Click the Programmer icon or select **Tools->Programmer** and follow the steps below (see also Fig. 2)

   a) Click the **Hardware Setup** button at the top left and double click **DE-SoC** then **Close**. [See Troubleshooting section at the end of this document if **No Hardware** is detected].

   b) Click the **Auto Detect** button and select **5CSXFC6D6** (second from last) then **Ok**. If it says 'Device chain does not match' and asks if you want to '**Update the device list**, overwriting existing settings', click **Yes**. [A second device should now have appeared in the lower pane, and in the central device list pane. This is the device we want to program].

   c) Click on the second device in the central device list pane to **highlight** it, then click the **Change File** button. Navigate to your compiled **Lab1.sof** file (in the output_files subfolder) and click **Open**.

   d) Check the **Program/Configure** check box in the fifth column

   e) Finally click **Start**. The **Progress** box, at the top to the right, will turn green when the DE10 has been successfully programmed.

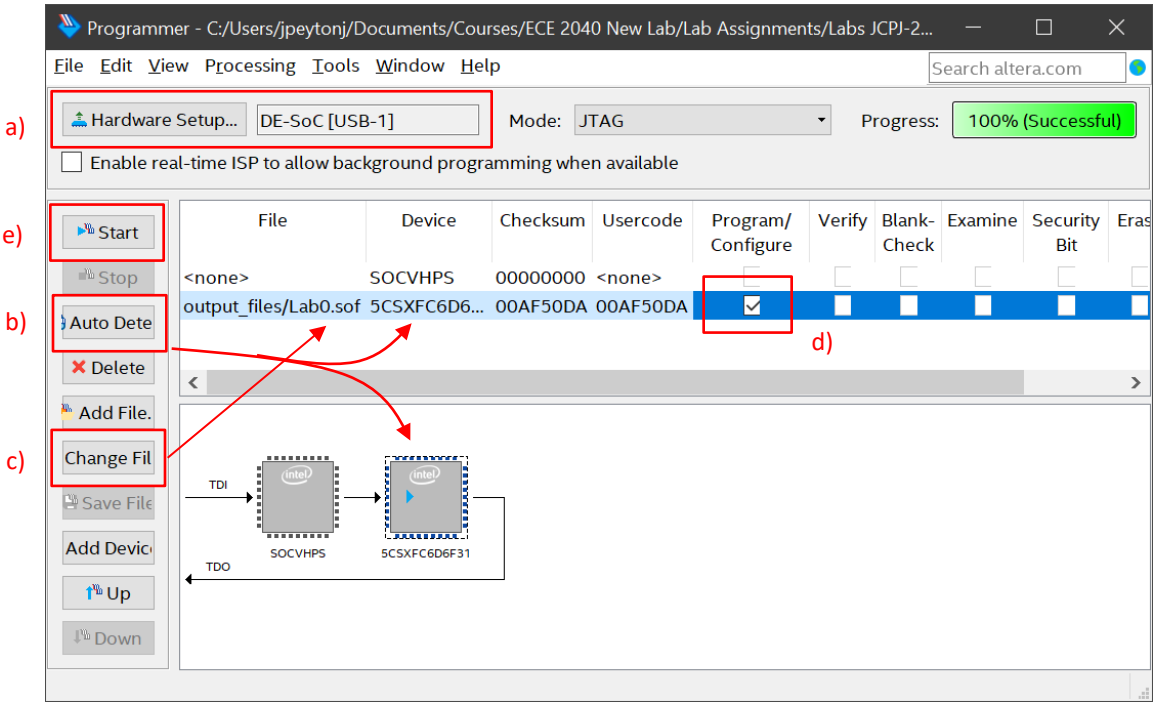

Fig. 2: How to Program the DE10-Standard board

11. Test to see if it works as expected! In this case, **LEDR[0]** should reflect the state of your **SW[0]** switch input (eg. only **on** when the switch is **up**/high), and **LEDR[1]** should be the opposite (eg. only **on** when the switch is **down** / low).

Demonstrate to the TA/Instructor.

1.2 Design and test a 1-bit full adder (Add1Bit)

Digital logic circuits can be designed to perform binary arithmetic/addition ☺. How can we do this? Consider the problem of adding two 4-bit binary numbers A, and B as illustrated below. The problem can be decomposed into how to **add two 1-bit numbers** A_i, B_i and a CarryIn (from the previous digit) in order to compute the correct Sum and CarryOut (for the next digit).

1. Using your knowledge of binary addition, **complete the 1-bit adder truth table** shown below **on your notebook** to show what values Sum and CarryOut should have for each possible combination of values A_i, B_i and CarryIn.

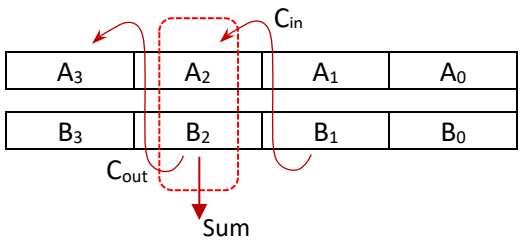

Fig. 3: Binary Addition

Input			Output	
C_{in}	A_i	B_i	Sum	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

2. A circuit which implements this truth table using primitive AND, OR and eXclusive OR (XOR) gates is shown below. The aim of this section is to create this design in Quartus and implement it on the DE10 board.

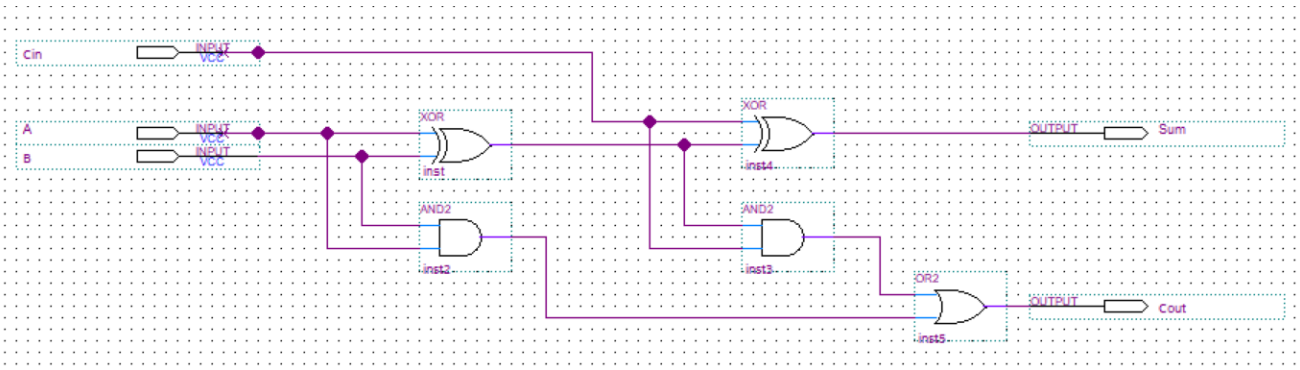

Fig. 4: A 1-bit 'full-adder' circuit.

3. To create a new / blank circuit diagram, select **File->New->Block Diagram/Schematic File** and then **File->Save As Add1Bit.bdf** (taking care to save this in your main Lab1 directory, not some subdirectory of Lab1). By default, the file will be added to your **Project Navigator->Files** list, or you can **Project->Add files** later.
4. **Double click** on the blank schematic to bring up the directory-like Component Library Browser of available components. To add the XOR gate that you need, expand **/…Libraries -> primitives -> logic -> XOR -> OK** and then **click** on the schematic to place your selected component.
5. Repeat this process to add the 2-input **AND** and 2-input **OR** gates that you need for your design. You can copy and paste, move them around, etc.
6. Also expand **/…Libraries -> primitives -> pin** to add the **input** and **output** pins that you need. Once placed on your design, **double-click** on the pin component, or **R-click->properties** on the pin component to edit the **pin name** to match the names in the circuit above.
7. Connect all these components using wires. There are a few ways to do this. Move your cursor to a pin of a gate then **click** and **drag** to your destination pin and release mouse button. Ask the instructor or TA if you need help. **File->Save** your work when done.
8. This part of the lab uses different pin names, so **Remove** the old **pin Assignments**, and **Import** the new assignments from the **Lab1P2-pins.qsf** file (as described in step #7 of Section 1.1).
9. In the **Project Navigator** pane pull-down menu, select **Files**, then **R-click** on the **Add1bit.bdf** file name and select **Set as top entity**.
10. **Compile** your project and **Program** it onto the DE10 board as described in steps #9, #10 of Section 1.1.
11. **Test** your design:
    - The inputs **C_{in}, A_i,** and **B_i** have been assigned to **SW[2..0],** respectively.
    - The **Sum** should appear on **LEDR[0].**
    - The **C_{out}** should appear on **LED[1]**.

    Test each combination of the input switches listed in the table, and **write down** the results you observe next to the truth table Output column on your notebook.

    Do these results match the original table that you worked out?

Input			Output	
C_{in}	A_i	B_i	Sum	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Demonstrate to the TA/Instructor.

1.3 Design and test a 4-bit full adder (Add4Bit)

The aim of this section is to use four copies of the 1-bit adder you just designed to make a 4-bit adder. Your design should look like this:

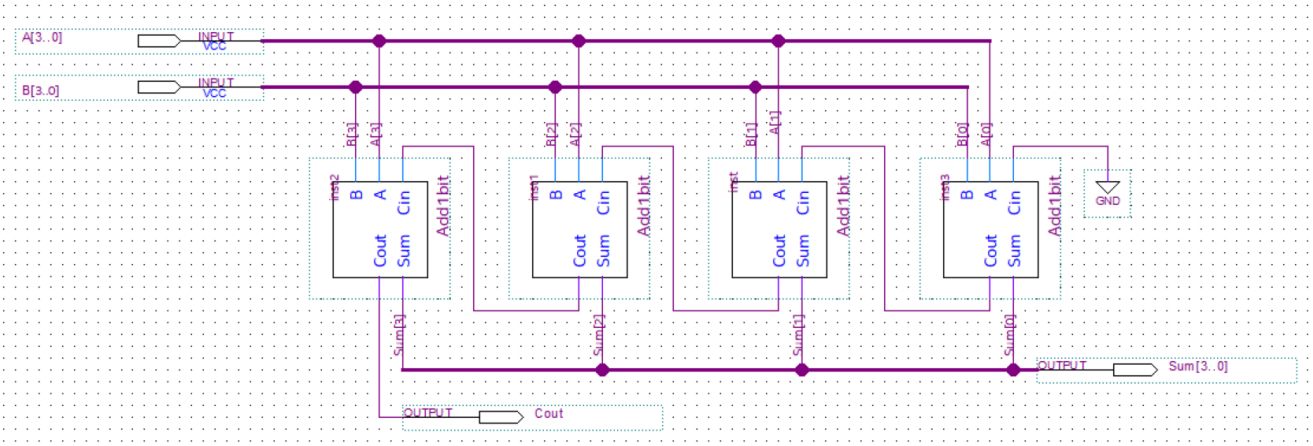

Fig. 5: A 4-bit ripple-adder circuit, made from four 1-bit full adders

1. Open / select your **Add1Bit.bdf** adder design, then click menu **File** -> **Create/Update** -> **Create Symbol File for Current File** and save the **Add1Bit.bsf** file to your **Lab1** working directory. Your 1-bit adder can now be re-used like the And / Or library primitives you used previously ☺.
2. Click menu **File** -> **New** -> **Block Diagram/Schematic File** to create a new schematic and immediately **File** -> **Save As Add4bit.bdf** in your Lab1 working directory.
3. **Double-click** on your empty Add4bit schematic and add four copies of your **1-bit adder** which you can find in the **Project** folder of the Component Library browser. You can **R-Click -> Rotate** these to match Fig 5.
4. Double-click to add an **Input pin** from the Primitives library and name it **A[3..0]** and draw a wire out from this input. This creates a thick 'bus wire' representing four wires in parallel. Connect the **A** input of individual 1-bit adders back to the bus using single wires. Select each individual wire and **R-click -> Properties** in order to name them **A[3]**, **A[2]**, etc., as shown in the above schematic to specify which of the four bus wires is being connected in each case.
5. Repeat this process for inputs **B[3..0]**, and outputs **Sum[3..0]** used to display the 4-bit Sum**.**
6. Hook the **Cout** of each 1-Bit adder to the **Cin** input of the next. Hook a **GND** input from the **Primitives/Other** folder of the Component Library browser into the **Cin** of the bit0 adder, and hook the final **Cout** to an **Output** pin, naming it **Cout** to display the Carry out. **File -> Save** when done.
7. We will want to re-use this design in Lab2, so **File** -> **Create/Update** -> **Create Symbol** and save the **Add4Bit.bsf** file to your **Lab1** working directory.
8. Now **File** -> **New** -> **Schematic File** and immediately **Save As->TestAdder.bdf** to create a very simple design for testing our 4-bit adder. Double-click to bring up the Component Library browser and add our **Add4Bit** component to this new design. Also add and hook up Input and output pins named **SW[3..0]** for the A input, **SW[7..4]** for the B input, **LEDR[3..0]** for the Sum output, and finally **LEDR[4]** for the Cout output. **File->Save**!
9. This part of the lab uses the original pin names, so **Remove** the old **pin Assignments**, and **Import** the new assignments from the **DE10-pins.qsf** file (as described in step #7 of Section 1.1).
10. In the **Project Navigator** pane pull-down menu, select **Files**, then **R-click** on the **TestAdder.bdf** file name and select **Set as top entity**.
11. **Compile** your project and **Program** it onto the DE10 board as described in steps #9, #10 of section 1.1.

12. **Test** your design using the switches to define the numbers you want added together in binary.
    To validate your results, **on your notebook**,
    - **Convert** the decimal number **11** to a 4-bit binary number: _____
    - **Convert** the decimal number **6** to a 4-bit binary number: _____
    - **Add** them manually, writing the result here: _____

    Now configure the switches on the board to compute the solution to this problem and check you get the correct result.

Demonstrate to the TA/Instructor.

Submission for Lab 1:

After you have been checked off all the three parts by the instructor/TA, zip the Lab1 project folder, and name the zip file *Lab1_<your last name>.zip, e.g. Lab1_Wang.zip*, and submit it to Lab1 assignment in Blackboard.
Finish prelab work on your notebook for Lab 2 before coming to the lab classroom for Lab 2.