LAB 1 Introduction to Quartus II Software Design

Objectives

This laboratory experiment is intended:

- To initiate the students who are not familiar with the Altera **Quartus II** Software Design.
- To act as a review for the more advance students.

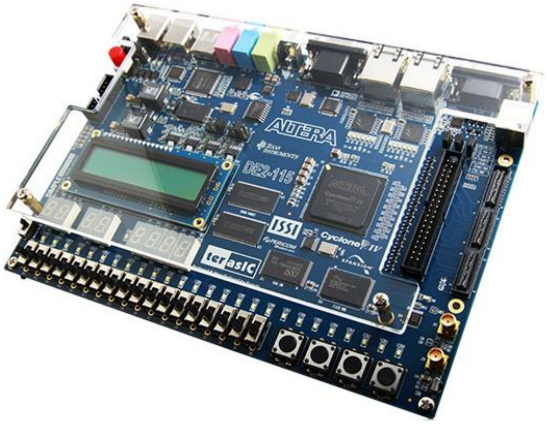


Figure 1: Altera DE2-115 board

On completion of this tutorial, the student will be able to:

- Understand the basic of the Altera environment.
- Design a simple logic circuit using the Graphic editor.
- Compile, simulate, debug, and test their design.
- Download and run their design on the Altera DE2-115 board.

PreLab

1. Read the "*Lab General Instructions*" and "*Introduction to the Altera DE2-115 Based HW/SW Platform with Environment Setup*" posted in the *Lab Instructions* section of your CEG2136 *Virtual Campus* and for more details on the Altera DE2-115 board consult the corresponding *DE2-115 User Manual* posted in the *Documentation* section under the *Laboratories* tab of CEG2136 *Virtual Campus*.
2. Derive the equation of the logic function implemented by the circuit of Fig. 2.
3. Draw the truth table for NAND, NOR functions.
4. Derive the truth table for the circuit in Figure 2.

   <u>You should complete PART I and PART II of the following LAB instructions</u>.

Laboratory

In this tutorial, we will implement a simple circuit as shown below with AND, NAND and NOR functions to provide an introduction to the Altera Quartus II tools.

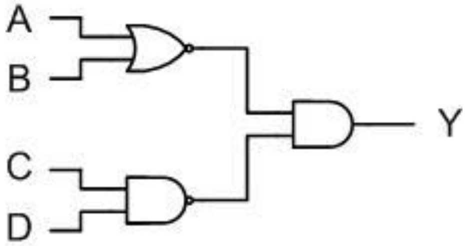


Figure 2 : A simple circuit with AND, NAND and NOR gates

PART I

Each logic circuit being designed with Quartus II software is called a *project*. The software works on one project at a time and keeps all information for that project in a single directory (folder) in the file system.

A. Starting a New Project

1. Start **QUARTUS II 13.0 sp1 (64-bit)** software. Program can be found under Start → All Programs → Software Development.
2. To create a new project, go to the F**ile** menu and select '**New Project Wizard**'. This opens 'New Project Wizard' dialog box which asks for the name and directory of the project. Set the working directory to be … \intro and choose **lab_1** as the name for both the project and the top-level entity (confirm creating the new directory)
3. When the "**Add Files**" window pops up just click **Next**, since there are no previous files to be included in the project.
4. In the "**Family & Device Settings**" select **Cyclone IV E** as the **device family** and check "**Specific device selected in 'Available devices' list**." From the list of **available devices**, choose the device called **EP4CE115F29C7** which is used on Altera's DE2-115 board and press **Next.**
5. In the next window (**EDA Tool Settings**), since in this lab we will rely solely on Quartus II tools, we will not choose any other tools and we will press **Next,** stepping to the final window (**Summary**). Press **Finish** to conclude and return to the main Quartus window.

B. Design Using the Graphic Editor

The Quartus II **Graphic Editor** can be used to specify a circuit in the form of a logic diagram.

Select **File > New,** and in the newly opened window choose **Block Diagram / Schematic File**, and click **OK** to create a blank schematic worksheet in the **Graphic Editor** window.

The first step is to specify a name for the file that will be created. Select **File > Save As** and in the box labelled **Save as type** choose **Block Diagram/Schematic File (*.bdf)**. In the box labelled **File name** type **lab_1**, to match the name given when the project was created. Put a checkmark in the box **Add file to current project**. Click **Save**, which puts the file into the directory **intro** and leads to the **Graphic Editor** window.

C. Creating the Schematic

Logic Gate Symbols

1. To open the **Symbol** window
   - Double click in the center of the worksheet of the **Graphic Editor, or**
   - Right click in the center of the worksheet and then choose in the pop-up window **Insert→Symbol**.
2. In the **Symbol** window's ***library*** *box* expand **…/quartus/libraries** and scroll down in the **primitives → logic** directory to select **nor2;** the corresponding gate-symbol will appear in the worksheet of the **Symbol** window. Click **OK** to close the **Symbol** window and transfer the selected symbol to the center of the ***lab_1.bdf*** worksheet of the **Graphic Editor**.

   To the same effect just double click the gate's name in the library directory.

   Now drag the gate-symbol to the wanted position and fix it there with a click.

NOTE: To avoid repetitive inserts, uncheck the Repeat-insert mode option in the **Symbol** window.

3. Repeat step 1 and 2 and select a **nand2** symbol.
4. Repeat step 1 and 2 again and select an **and2** symbol, or simply type the components' names (**nor2, nand2** or **and2**) in the **Symbol**'s ***Name*** box.

NOTE: Multiple copies of a symbol can be obtained by "copy & paste" or it can be generated successively in one sequence if in step # 1 the **Symbol** window is opened by clicking on the **Symbol Tool** (represented by an **AND** gate symbol in the toolbar next to the worksheet). As the selected symbol appears in the center of the **Graphic Editor** it can be moved to the wanted position. Click once to leave there a copy of the symbol; then move the cursor to drag the symbol to the next position and click again to leave it there. Repeat this procedure until you draw all the symbols of the same type.

D. **Input and Output Symbols**

1. Open the **Symbol** window as shown above.
2. In the **Symbol** windows' **Libraries** box, double click on the **Primitives → Pin** and select and place an **Output** symbol on the worksheet of the **Graphic Editor**.
3. Repeat step 1 and 2 to place four **Input** symbols on the worksheet.

E. **Connecting the Symbols**

1. Go to the end of a symbol with the mouse and when the arrow cursor changes to a cross-symbol press the left button of the mouse and drag the wire to the point to connect to; as the cursor reaches the destination, the cursor changes again to a small square, and at that point you can release the mouse; see diagram below for the connection.
2. Repeat the previous step for all connections.
3. If a wire is not properly run, just select it (wire turns red) and hit delete to remove it.
4. If you have problem running the wire from one point completely to another, try running half way from both devices.
5. The mouse can also be used to move a wire to the desired position.
6. Now, your diagram should look like the one below.

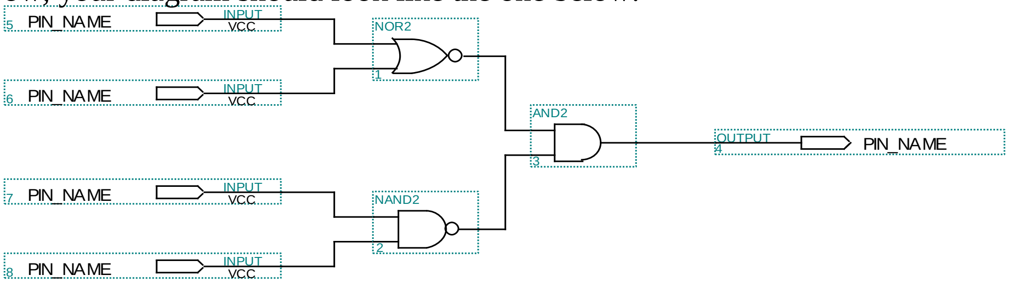


Figure 3**:** Schematic with input and output wired

F. **Editing Pin Names**

1. **Right click** (or double click) on an **INPUT** symbol and select **Properties → General/Pin Name.**

2. Name the pin as shown below (i.e., replace “pin_name” with “A”) and click OK.
3. Repeat for all pins, and save eventually your files.

NOTE: The pin name can be edited if you click once on the pin (to select it) and then click a bit later on the “pin_name.”

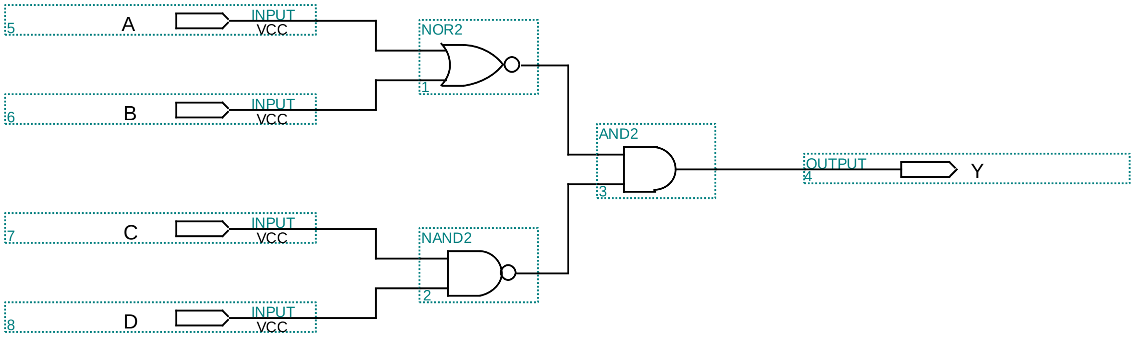
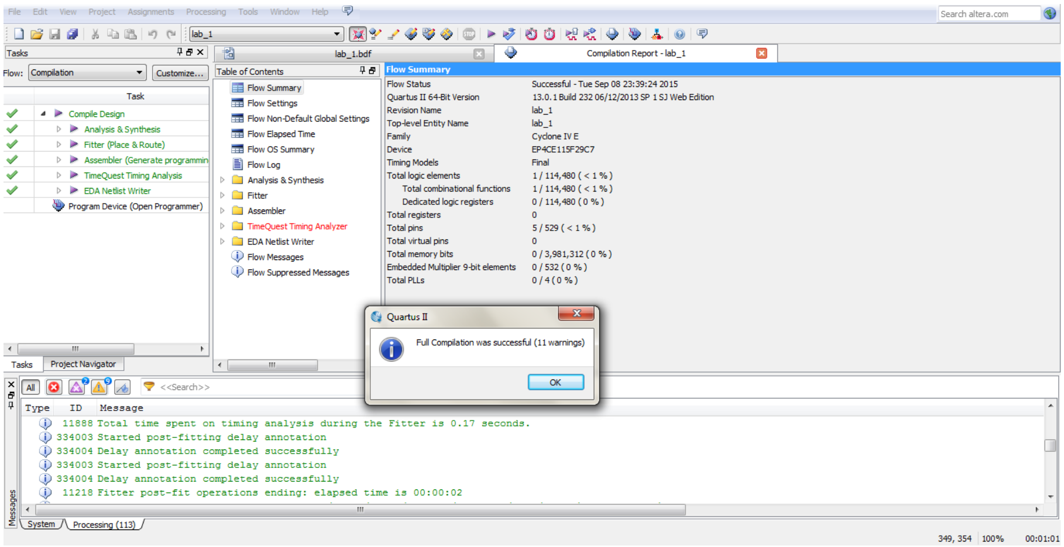


Figure 4: Schematic Ports Name

G. Compiling your Project

1. Select **Processing → Start compilation** or click on the toolbar icon ▶ or press Ctrl+L.
2. A similar window should appear.



Flow Summary

Flow Status	Successful - Tue Sep 08 23:39:24 2015
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	lab_1
Top-level Entity Name	lab_1
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1 / 114,480 (< 1 %)
Total combinational functions	1 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	5 / 529 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Quartus II

Full Compilation was successful (11 warnings)

OK

```
Type  ID     Message
      11888 Total time spent on timing analysis during the Fitter is 0.17 seconds.
     334003 Started post-fitting delay annotation
     334004 Delay annotation completed successfully
     334003 Started post-fitting delay annotation
     334004 Delay annotation completed successfully
      11218 Fitter post-fit operations ending: elapsed time is 00:00:02
```

Figure 5: Compiling Window

3. The project should compile with **0 errors**. If any errors appear verify if you have performed the entire steps correctly. (Ignore the warning about Timing)

4. Close the compiler window.

NOTE: Every time changes are made to a file, the project must be saved and compiled again.

H. **Assign PIN Numbers**

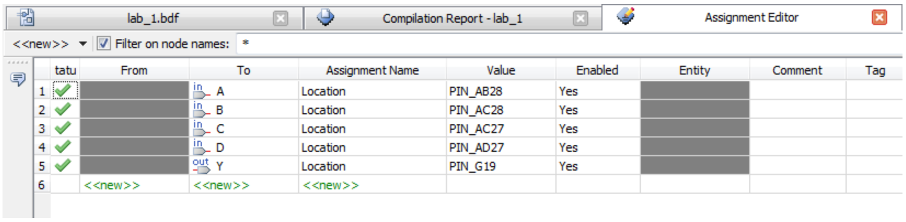
For static verification of the circuit we want to connect the slide switches SW0 – SW3 of the DE2-115 board to the circuit's inputs (A, B, C and D) and the red LED LEDR0 to the circuit's output Y. In the experimental part of this lab, the logic levels applied to the inputs can then be controlled by the slide switches, while the value of the output can be visualized with LEDR0. These switches and LED are directly connected on the PCB to the Cyclone FPGA pins, and they will be assigned in Quartus accordingly, as shown below.

1. Select **Assignments → Assignment Editor;** under **Category** select **ALL**.
2. Double-click on the entry **<<new>>** in the column labeled **To**. Press the binoculars to open the Node finder window, then select
   a. **Filter → Pins: all**, then click on **List**.
   b. Select (highlight) A, B, C, D and Y from the left column of **Nodes Found** and then click on "**>"** to have all A, B, C, D, and Y in the right **Selected Nodes** column. The content of the left column is entirely copied to the right (without having to select them) by simply hitting ">>"
   c. Click **OK** to close the **Node Finder** window.
3. On the box to the right of the new **A** entry, in the column labeled **Assignment Name,** double click the blank field and select Location (Accept wildcards/groups) from the drop-down the list. Note: once Location (Accept wildcards/groups), text defaults to Location.
4. In the Value column for the **A** entry type PIN_AB28.
5. Repeat steps 2 and 3 to assign the rest of the signals to the pins as described by Table 1, below.
6. Save the assignments you made by choosing **File > Save**. You can also simply close the **Assignment Editor** window, in which case a pop-up box will ask if you want to save the changes to assignments; click **Yes**.

Recompile the circuit, so that it will be compiled with the correct pin assignments.

Table 1: **Device Pin connections and name**

Signal	Pin Location Value	Component
A	PIN_AB28	SW0
B	PIN_AC28	SW1
C	PIN_AC27	SW2
D	PIN_AD27	SW3
Y	PIN_G19	LEDR0



	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		in A	Location	PIN_AB28	Yes			
2	✓		in B	Location	PIN_AC28	Yes			
3	✓		in C	Location	PIN_AC27	Yes			
4	✓		in D	Location	PIN_AD27	Yes			
5	✓		out Y	Location	PIN_G19	Yes			
6		<<new>>	<<new>>	<<new>>					

Figure 6: Assignment Editor completed

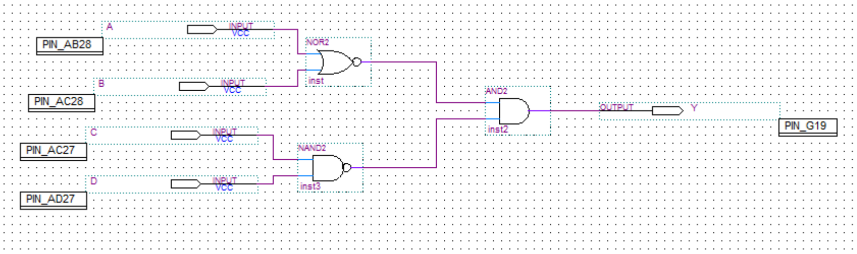


Figure 7: Block diagram with pin assignments

PART II

Simulating your project

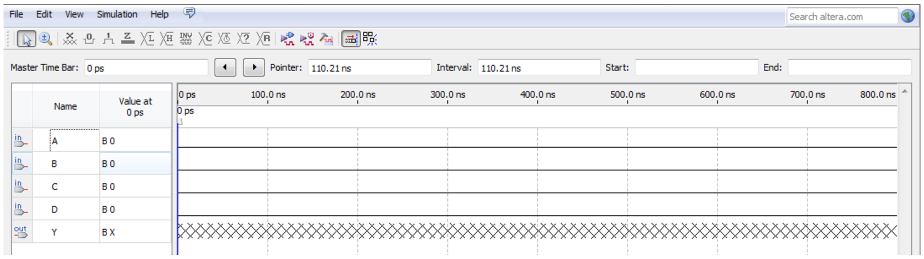
Quartus is provided with a **Waveform Editor** which allows generating the testing stimuli (test vectors) that are to be applied at the inputs of the simulation of the designed circuit.

2. Select **File → New→ University Program VWF** and click **OK** and the window of the **Waveform Editor** pops-up. This window has two sub-windows: time diagrams are displayed on the right hand, while on the left column the terminal signals are enlisted. **Save** the file under the name *lab*_1.*vwf.*
3. To populate the list of terminals for the simulation, right click on left side of the waveform editor (in the column under the **Name** header) and select **Insert → Insert Node or Bus;** in the new window click on **Node Finder,** then select
   a. **Filter → Pins: all**, then click on **List**.
   b. Select (highlight) A, B, C, D and Y from the left column of **Nodes Found** and

then click on “**>**” to have all A, B, C, D, and Y in the right **Selected Nodes** column. The content of the left column is entirely copied to the right (without having to select them) by simply hitting “>>”

c. Click **OK** to close the **Node Finder** window.

4. Click **OK** to close the **Insert Node or Bus** window.
5. Click on a pin to select a row and drag it to a different position if you want to modify their initial order.
6. Set the desired simulation to run from 0 to 800 ns by selecting **Edit** > **End Time** and entering 800 ns in the dialog box that pops up. To apply a new *test vector* every 100 ns choose **Edit** > **Grid Size** and enter a **Time Period** of 100 ns in the dialog box that pops up. Selecting **View** > **Fit in Window** displays the entire simulation range of 0 to 800 ns in the window.



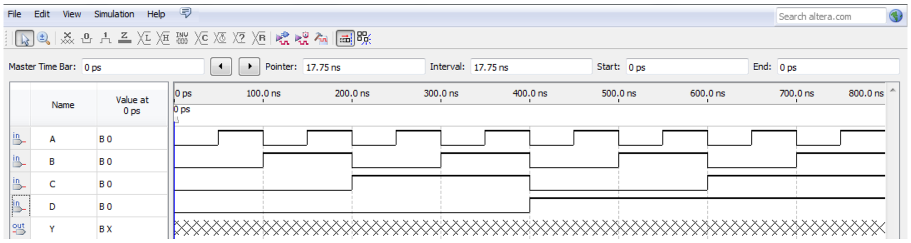
Master Time Bar: 0 ps
Pointer: 110.21 ns
Interval: 110.21 ns
Start:
End:

Name	Value at 0 ps
A	B 0
B	B 0
C	B 0
D	B 0
Y	B X

Figure 8: Waveform diagram incomplete

You may wish to resize the window using a couple of short-cuts: *control+space* for zoom-in and *control+shift+space* zoom-out.

7. There are several ways to create the desired *test vectors* that represent the input signals of the logic circuit you want to simulate. Let’s consider the following example:



Master Time Bar: 0 ps
Pointer: 17.75 ns
Interval: 17.75 ns
Start: 0 ps
End: 0 ps

Name	Value at 0 ps
A	B 0
B	B 0
C	B 0
D	B 0
Y	B X

Figure 9: Waveform diagram complete

a. Exhaustive testing (counter input).

   To cover all the logic combinations that can be generated by 4 input variables (say A, B, C, D in the above time diagram), you may want to apply test vectors whose binary equivalents would count from 0-15 (logic variable **D** being mapped to the msb of the binary number – weight 2^3, while **A** being mapped to lsb – 2^0). To realize such pattern,

   - click on **A** to highlight (select) the first row of the time diagram and then select the **clock** from the toolbar; choose a **Period** of 100 ns and a **duty cycle** of 50%.
   - click on **B** and in **Period** of 200 ns and a **duty cycle** of 50%.
   - click on **C** and in **Period** of 400 ns and a **duty cycle** of 50%.

b. Direct time editing

   - With the mouse left button, click and drag the mouse from 0ns to 400.0ns for Node D. This interval would then be highlighted.
   - Go to the waveform manipulation buttons and select 0 for this interval (click 0 on the tool bar).
   - Click and drag the mouse from 400ns to 800ns for Node D and select 1 for this interval (click 1 on the tool bar).

8. Save the file.
9. Make sure that under the simulator is the Quartus II simulator. Under Simulation menu go to Options and select Quartus II Simulator as the simulator. Simulation can be triggered by clicking or by selecting in the **Waveform diagram** menu **Simulation → Run Functional Simulation**
10. Once simulation is done, a new window would appear with the simulation results.
11. Simulate different scenarios and explain the simulation result.

PART III

A. Downloading your Project to the DE2-115 board and test

1. Make sure the **USB-Blaster** cable is attached to the board and to the USB port on the PC.
2. Make sure the RUN/PROG switch (SW19; leftmost toggle switch) is set to RUN.

3. Select **Tools → Programmer** in the **Quartus II** window.
4. From **Hardware Settings**, in the **Currently Selected Hardware** box, select **USB-Blaster** and click **Close.**

   Note: if the USB-Blaster doesn't show-up in the list of **Currently Selected Hardware**, close the window and open it again. You might have to repeat this process a few times.
5. In the **Programmer** window, check that the …/*output_files/lab_1.sof* file is listed. If it is not then click the **Add File** button on the left panel and look for the lab_1.sof file under the …/*output_files* directory in the current working directory.
6. Make sure **Program/Configure** is checked-in**.**

   Click **Start** and verify your circuit according to the simulation using the DIP switch as input (SW0 is input **A**, **B** is SW1, **C** is SW2 and **D** is SW3) and the LEDR0 as output (**Y**). Remember that a LED illuminates when its control input is 1.
7. NOTE: Once done you do not need to save the *.cdf file.
8. To verify your circuit, apply all 16 possible logic combinations to the circuit's inputs A, B, C, D by modifying the slide switches (SW0 is input **A**, **B** is SW1, **C** is SW2 and **D** is SW3) and observe the output (**Y**) on LEDR0. Remember that on this board a LED illuminates when its control input is 1.

References

1. "DE2-115 User Manual," Terasic Technologies Inc., http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502&PartNo=4 , 2013