EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

CMPE 324 - Computer Architecture and Organization

# Lab 5: Introduction to Circuit Synthesis Using ALTERA QUARTUS

## 1. Objective:

To get familiar with the VHDL development and simulation tools by generating a full adder circuit entry using the graphical capture. Within the framework of EDA (Electronic Design Automation), in this Lab we will learn the use of "Quartus" software to design and test logic circuits. The Quartus program is very popular and it is used in the construction of many modern devices.

## 1. Introduction

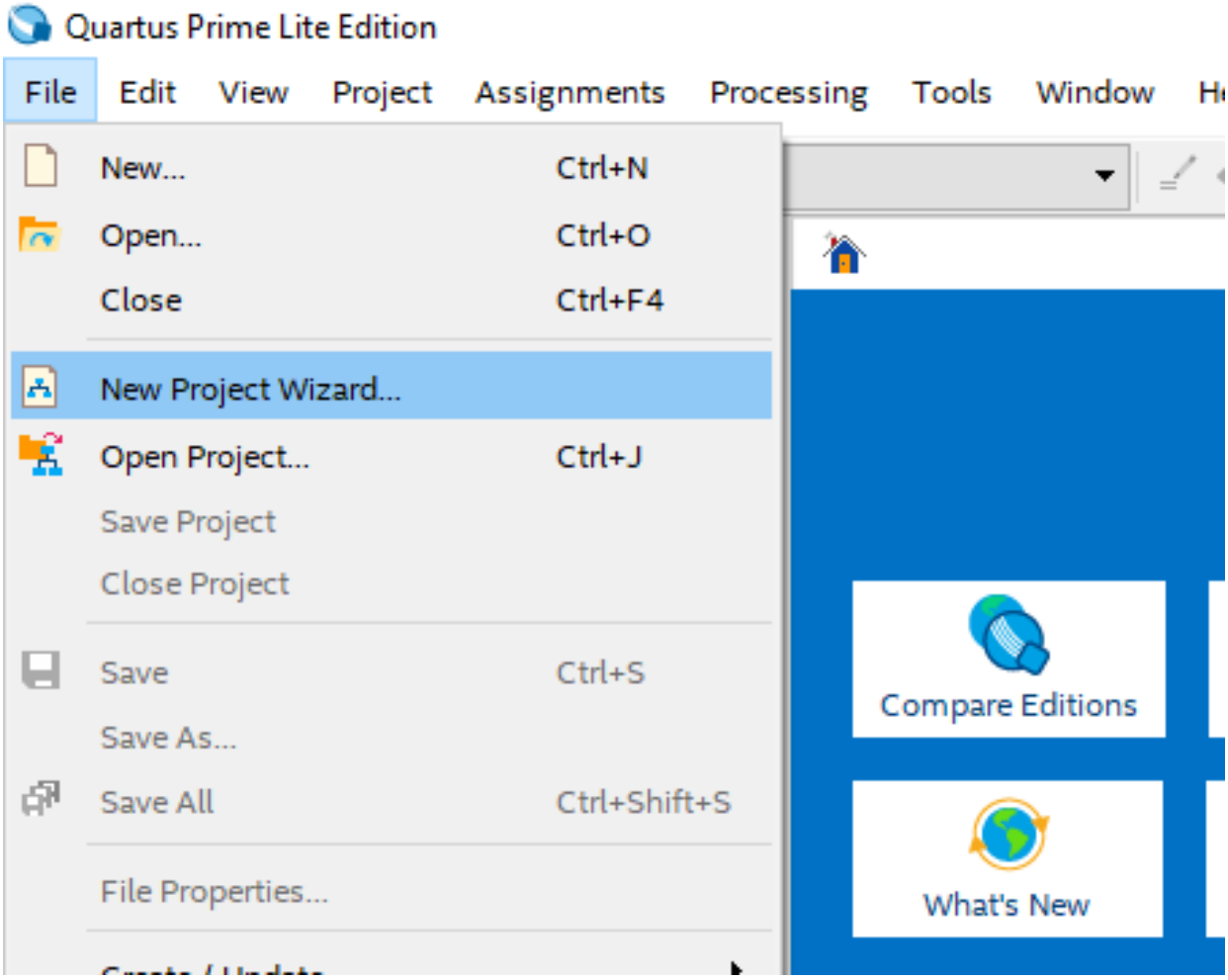
VHDL is an abbreviation for the VHSIC Hardware Description Language, where VHSIC is acronym for the Very High-Speed Integrated Circuits. It is originated from the standardization efforts of US. Department of Defence to provide a convenient and precise form of communication for the defence contracts. Today, the digital circuits are mostly implemented with digital programmable logic devices namely SPLDs CPLDs and FPGAs.

## 2. Experimental Practice

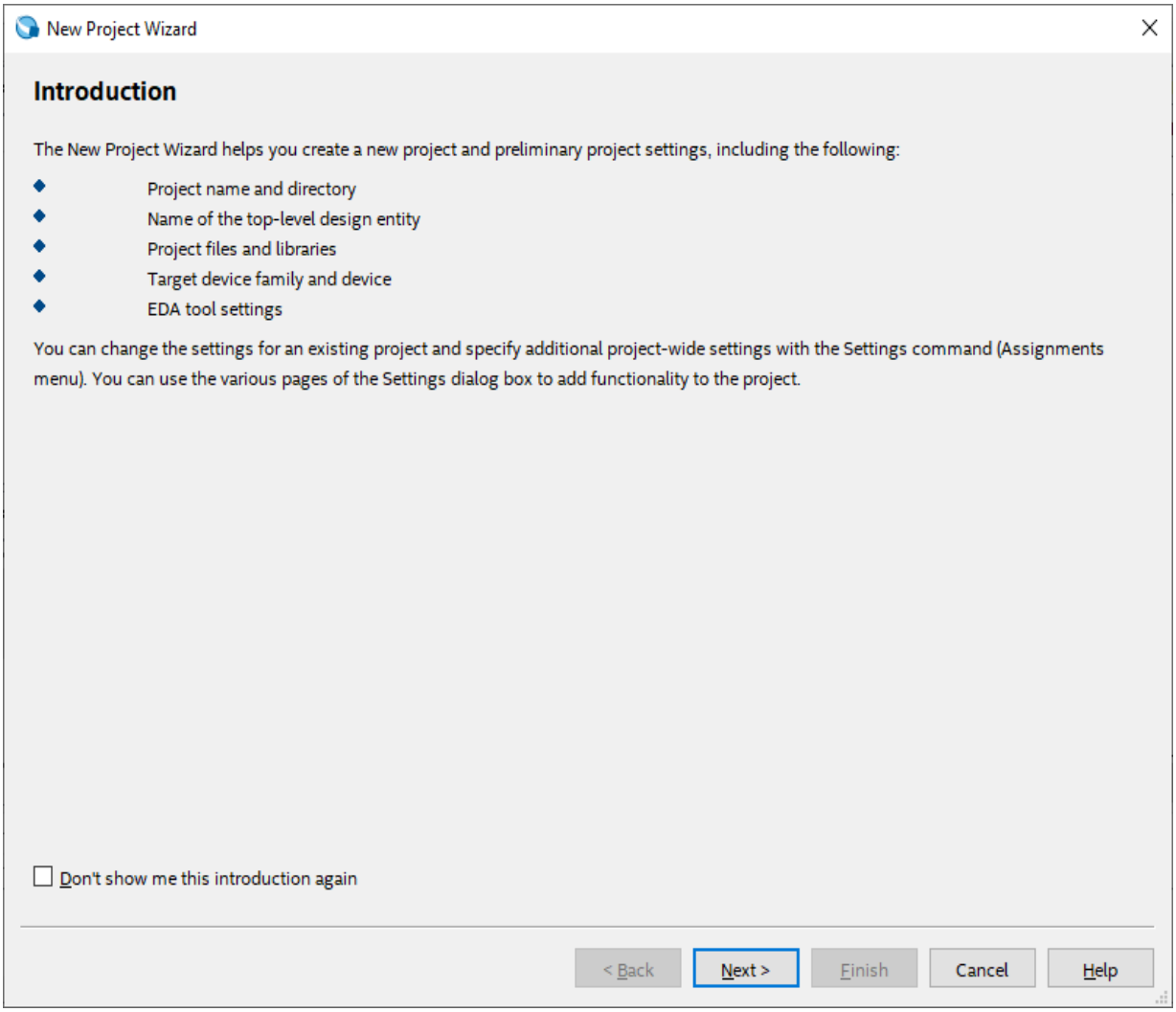
**Part 1: about Quartus and first simple experiment**

Step 1. Create a Project: Every new design you make in the Quartus program must be under a project name. After opening the program, to create a project, click on:

[File] -> [New Project Wizard]



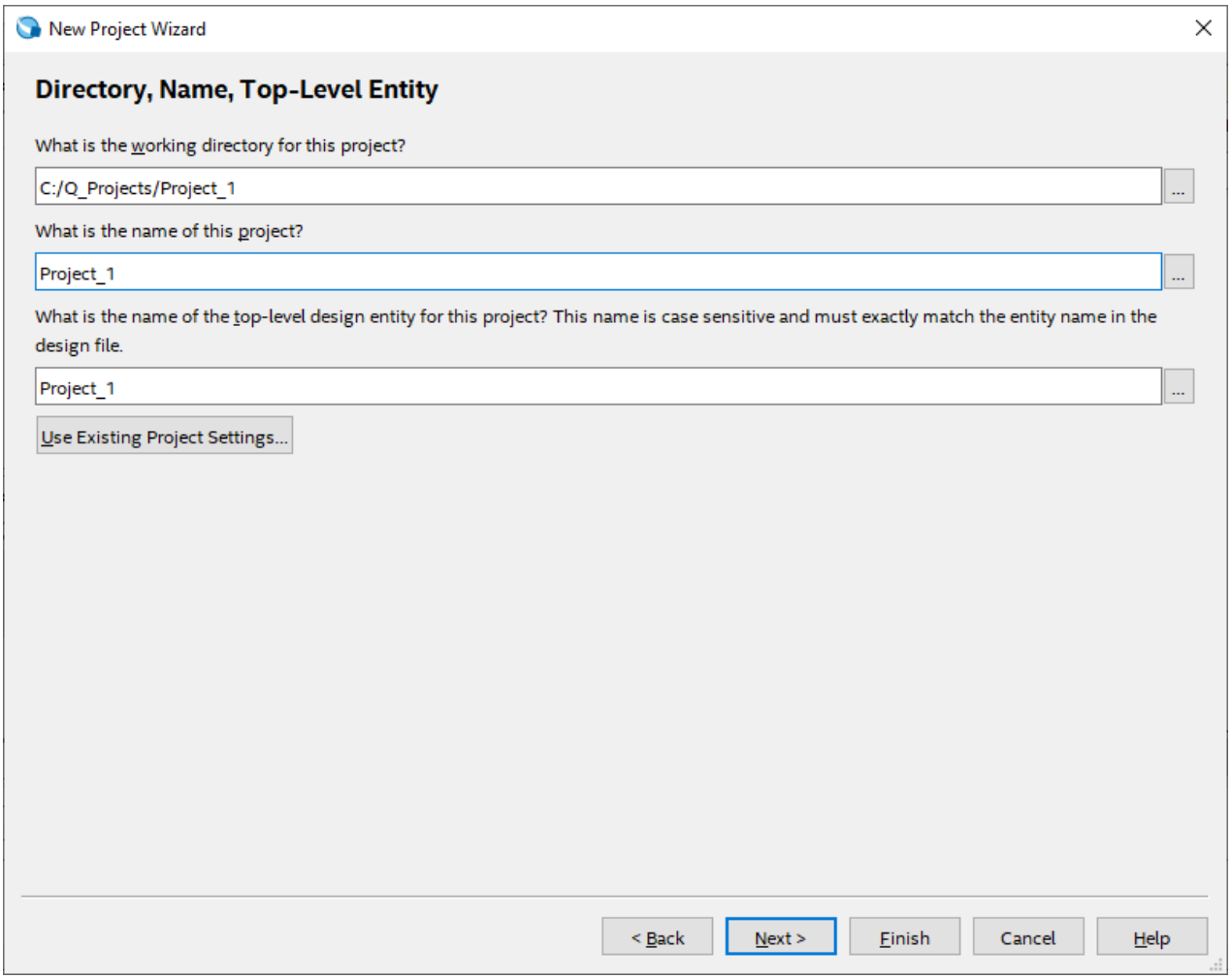

After this stage you will see the following project builder screen.



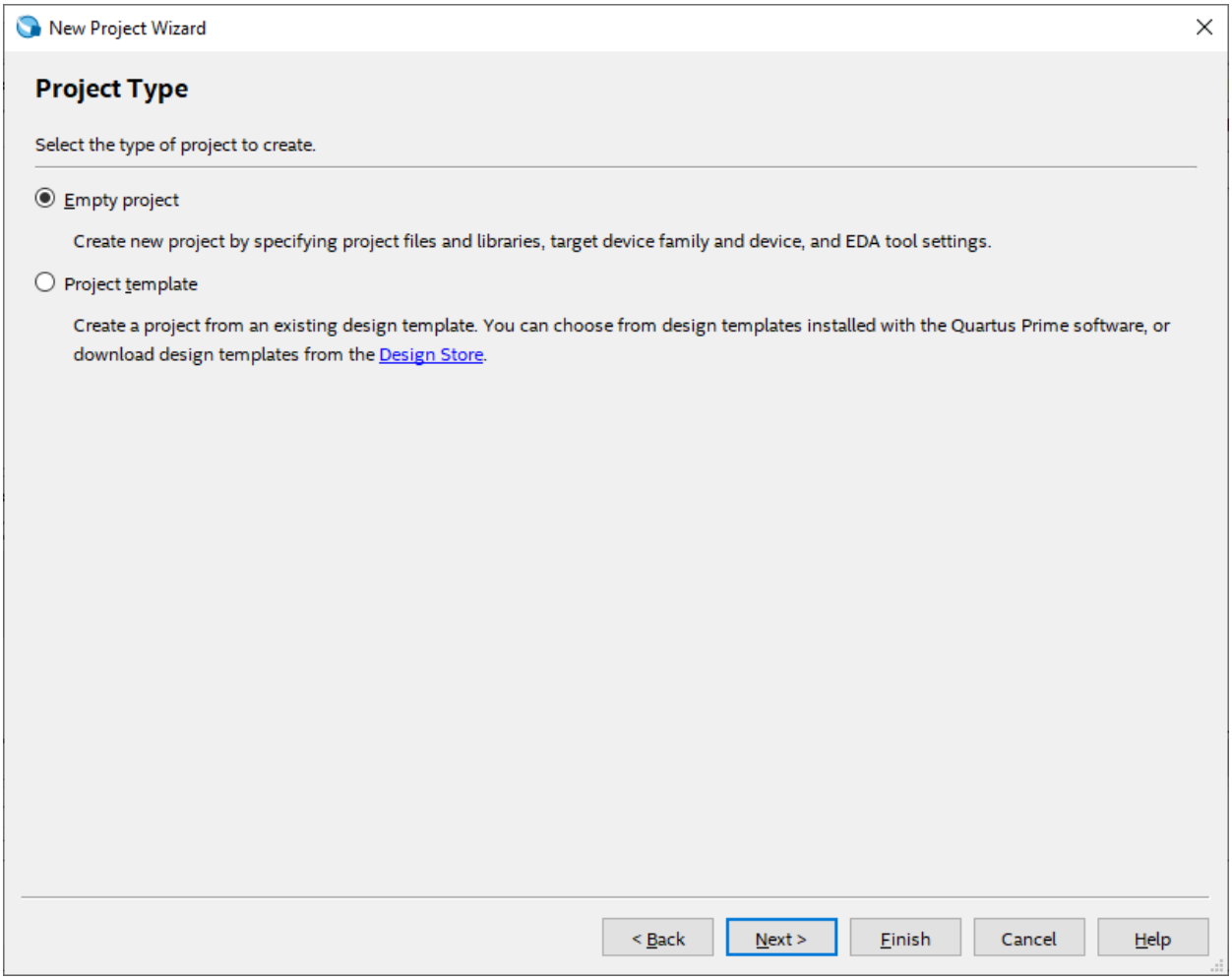

This screen is the home screen of the project builder. Here you will see which stages of your project are formed and what you can organize.

Here, press [Next >] button to continue.

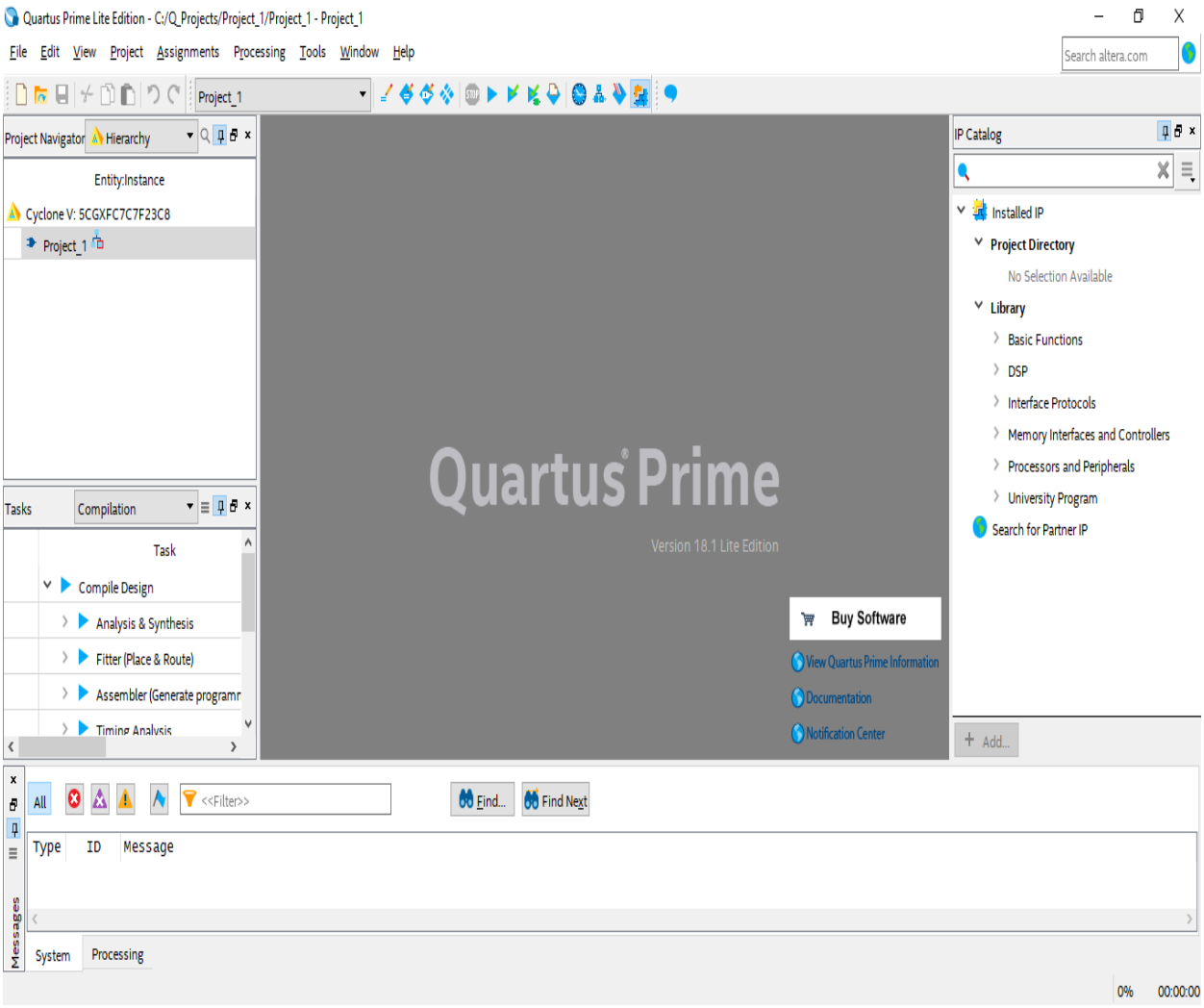
On the new screen, you will see the input sections that ask us to enter the directory and the name of the related project. Please fill in the appropriate information here. (Project names for the Quartus program is very important.)




The project type display gives you additional setting options for what hardware the project will be compiled and implemented. Since, we'll simulate our projects on the program and we won't run on any real hardware, we'll choose the [Empty Project] option here, and then click the [Finish] button to create our project.

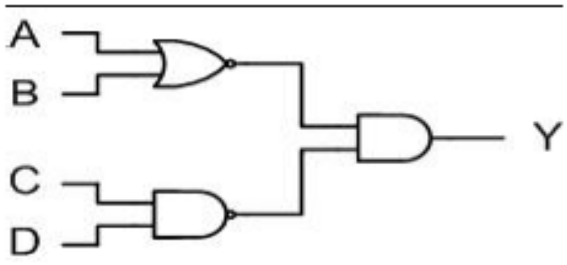



Once your project is created, you will see a screen like this:




Step 2. Simple schematic design:

Once you have successfully created your project, we can make a simple design and simulation process. At this stage, we will use the design below as an example.



The next step is to create an empty Block Diagram. To do so, click [File] -> [New] to select Block Diagram / Schematic File on the pop-up menu and click [OK]. At this stage, an empty design window will be opened.

When you double-click on the empty worksheet that opens, you will see a screen showing the libraries that are built in the Quartus program. On this screen you will see Logic gates libraries under the Primitives libraries. Once you have chosen this, you will come across logical gates that you can use in your designs.

Here, we will include 1 nand2, 1 nor2, 1 and2, 4 inputs and finally 1 output components as shown in our sample circuit.

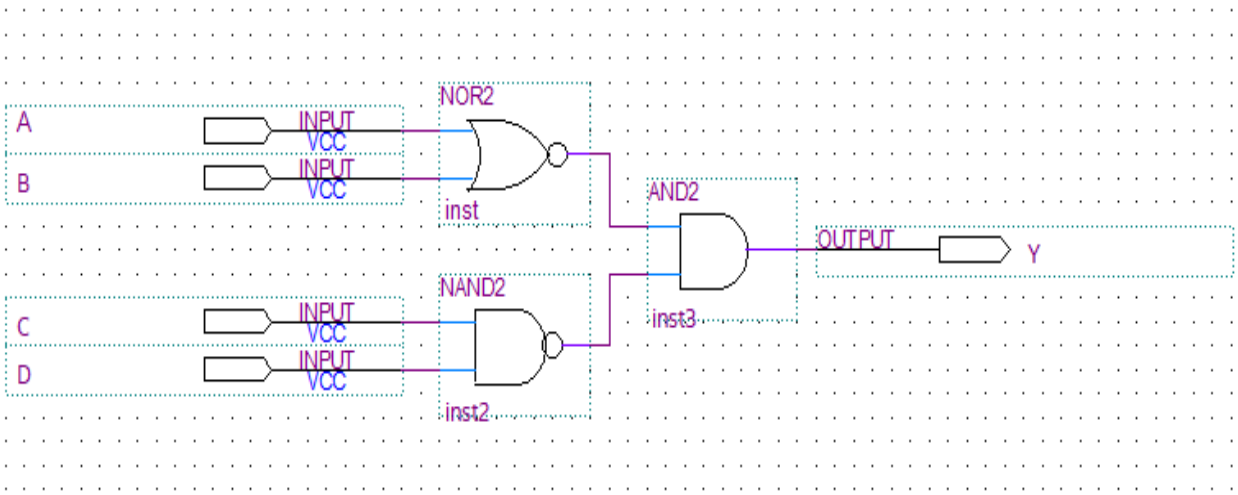
Connecting the elements together:
We will place the elements added to the Quartus schematic page by dragging them with the mouse or selecting the element and moving them with the arrow keys on our keyboard. After this step, to connect the circuit elements together as required, place the mouse cursor on the input or output pins of the circuit elements so that it will inform us that it is ready to make a connection by turning into a symbol related to the connection. Press the left mouse button and pull the connecting cable / wire to the other part to be connected. To indicate that the connection to the input is made correctly, a small square between the wire and the port will appear where the mouse cursor is located. You can leave the mouse button after this square appears.

Note: If your connection is faulty, there will be a sign similar to ‘’x’’ at the end of the cable that you connect. This sign shows that the connection has not been made correctly. In this case, you must right-click on the cable you connect and click ”delete“ to delete and reconnect the cable again.

After connecting our circuit without problems, double-click on the “PIN_NAME” pin posts on the input and output pins to name them as in the example.

After all the steps are finished, we can save our project with the combination of [ctrl + s] keys or [File] -> [Save] option.

Schematic Diagram:




Step 3. Compiling the Project:

We do the compilation of the project with the [Processing -> [Start Compilation] menu.

If there are no errors in your project, a pop-up menu with “Full compilation was successful” will appear.

Close the Compiler display.

Step 4. Simulating the Project:

A circuit can be simulated in two ways. The simplest way is to assume that the logical elements and the interconnecting wires are perfect, so it does not cause delays in the propagation of the signals throughout the circulation. This is called functional simulation. A more complex alternative leads to timing simulation, taking into account all propagation delays. Typically, the functional simulation is used to confirm the functional accuracy of a circuit as designed. This takes much less time

because the simulation can be performed using logical expressions that define a simple circuit. In this course we will use only functional simulation.

Select [File] -> [New] -> [Verification / Debugging Files] -> [University Program VWF] and press the "OK" button.

Select [Edit] -> [Insert Node or Bus] from the pop-up window.

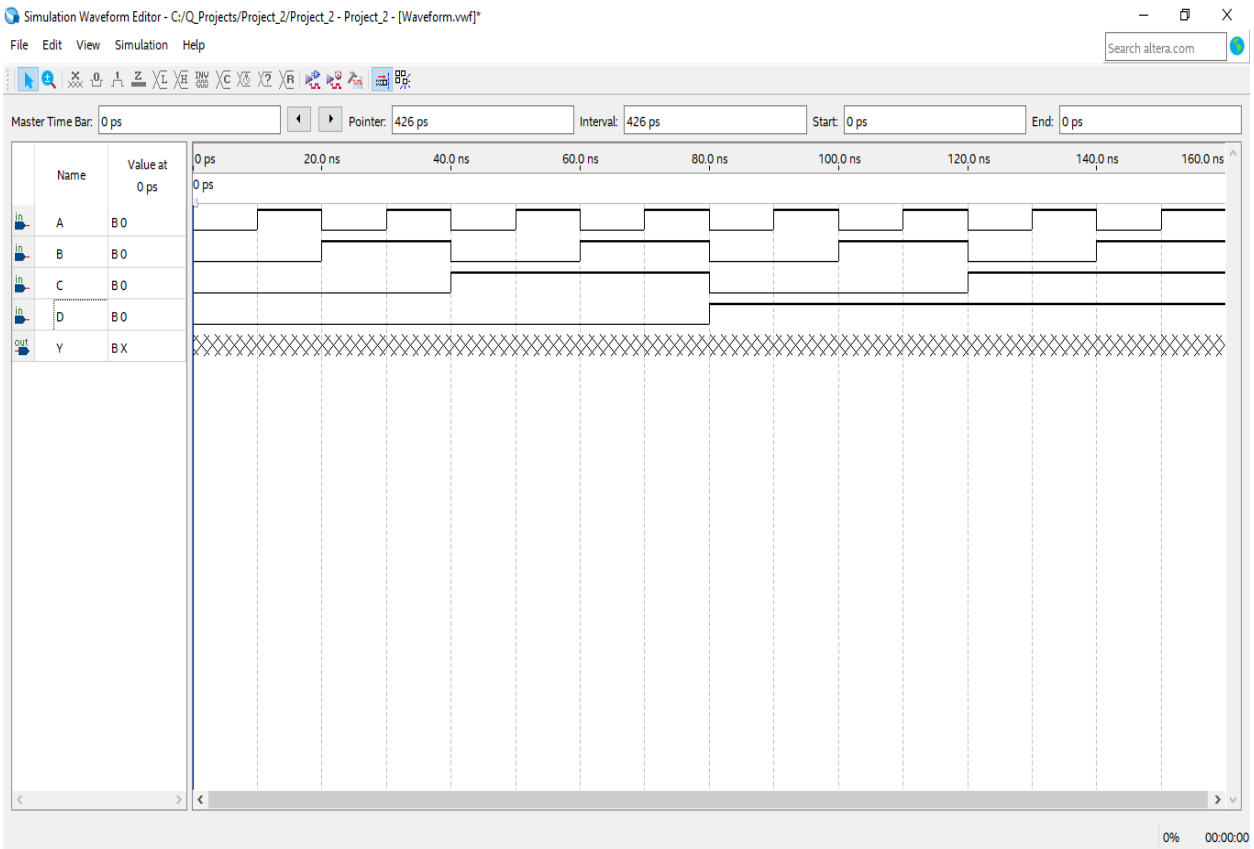
Click on the “Node Finder“ button.

In the list of the pins, press the Import all button “>>” and click “Start”.

You can change the order of the pins you see on your screen.

Enter [Edit] -> [End Time] menu to set “set end time” to 160 and change the time type section “us” -> “ns”.

Click the “Overwrite clock” button to set the time intervals of our pins to A: 20, B: 40, C: 80, D: 160.




Under [Simulation] -> “Run Simulation Functional Settings”

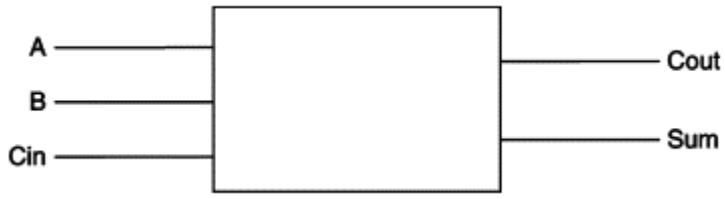
Save your simulation model by pressing [File] -> [Save].

Simulate your project by pressing [Processing] -> [Start Simulation].

**Part 2: single-bit full adder**

In this section, we will implement a full adder, an 8-bit adder, and compare the multiplication algorithms by simulation. First of all, build a single-bit full adder circuit entry Building a single-bit full adder circuit entry.

he full-adder extends the concept of the half-adder by providing an additional carry-in (Cin) input, as shown in Figure 5.21. This is a design with three inputs (A, B, and Cin) and two outputs (Sum and Cout). This cell adds the three binary input numbers to produce sum and carry-out terms




| A | B | Cin | Sum | Cout |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

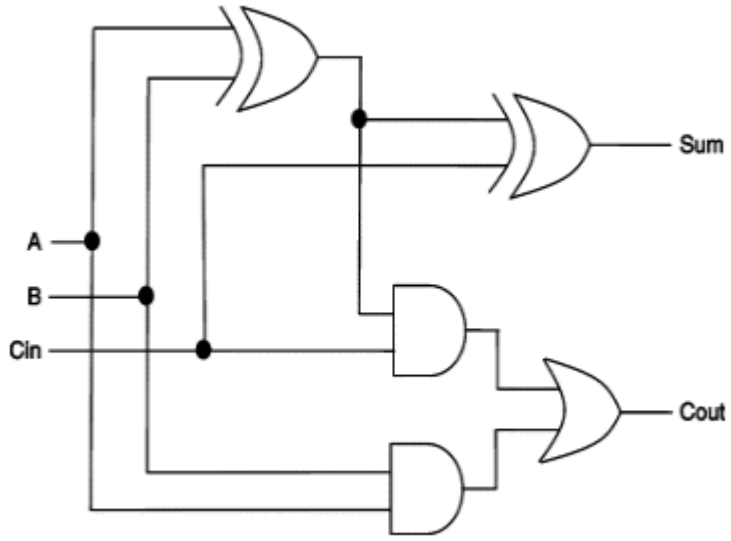
From viewing the truth table, the Sum output is only a logic 1 when *one or three* (but not two) of the inputs is logic 1. The Boolean expression for this is (in reduced form):

$$Sum=Cin \oplus (A \oplus B)$$

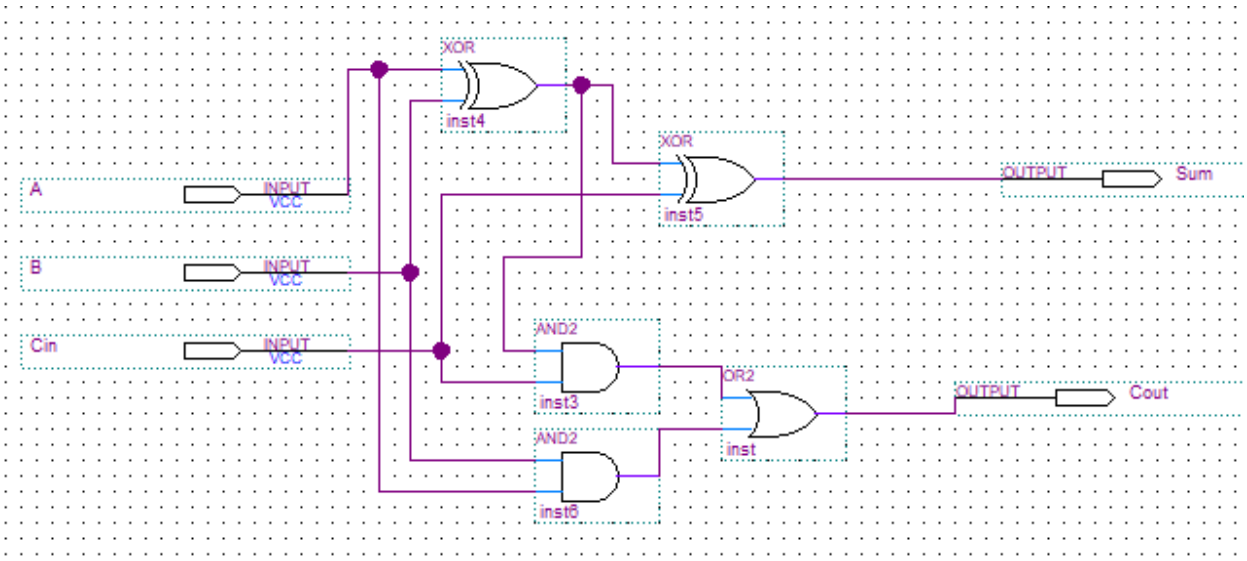
From viewing the truth table, the Cout output is only a logic 1 when *two or three* of the inputs is logic 1. The Boolean expression for this is (in reduced form):

$$Cout=(A.B)+(Cin.(A \oplus B))$$

This can be drawn as a circuit schematic as shown in Figure bellow:




Follow the on steps part 1 in order to design this circuit:




- Draw the "Schematic / Block Diagram" of the above functions using the Quartus software program.
- Simulate the project showing the WaveForm in the Quartus program "University program VWF" and show it to the lab assistant.
- Compare the truth table that you wrote in your notebook with the result of the simulation that Quartus gives you.

**Note:** be careful about connections.

## 3. Reporting

Before the Lab-time is over, show the simulation result to your lab assistant, in order to grading.

## 4. Homework: build a 4-bit full adder

Any number of half- and full-adder cells can be connected together to form an n-bit addition. The Figure shows the connections for a four-bit binary adder. In this design, there is no Cin input. Inputs A and B are four bits wide, and bit 0 (A(0) and B(0)) are the LSBs.

- Draw the "Schematic / Block Diagram" of the above functions using the Quartus software program.
- Simulate the project showing the WaveForm in the Quartus program "University program VWF" and show it to the lab assistant.
- Compare the truth table that you wrote in your notebook with the result of the simulation that Quartus gives you.

Grading:

Lab Performance:

Asst. Observations: