

CS 210 Laboratory Exercise Two

Design of a Ripple-Carry Adder Circuit Simulation with QuartusII

In this lab, you will begin the actual design of a processor by designing the ripple-carry adder that we studied in class. Then you will learn about how to observe the behavior of this circuit by using the waveform simulator built into QuartusII.

Starting QuartusII

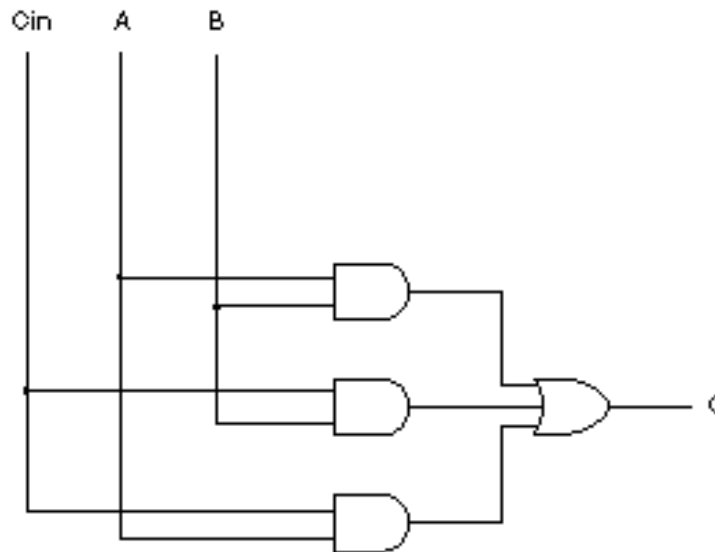
Recall that in order to start up QuartusII , you need to do the following:

1. After logging into the PC, access your Unix account by clicking on **Start → Unix Servers (X-Windows) → csa3.bu.edu**. Then login to your account by typing your username and password.
2. Start the program by typing **quartus**.

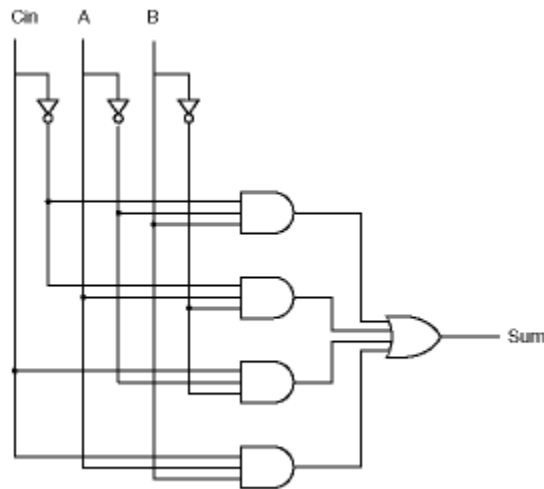
Ripple-Carry Adder Design

Now that you have started QuartusII, and are familiar with the Graphical Editor from a previous lab, you are ready to design a real circuit: a ripple-carry adder. As you know from lecture, this circuit has two parts: Carry-Out (aka the Majority function) and Sum (aka the Even-Parity function). Here are the circuit diagrams for each of these.

Carry-Out (Majority Function):



Sum (Even-Parity Function):



Designing a Hierarchical Circuit

Now you must draw each of these circuits as a separate BDF file.

1. Select **New** from the **File** menu and draw the CarryOut circuit, then choose **Save As** from the **File** menu and save the file as **carryout.bdf** in the **quartus** directory you created in a previous lab. Then, from the **File** menu choose **Project → Set Project to Current File**. Quartus II pop up window will ask if you want to open, create, or convert a project. Choose **Create Project** option. Name the project and top level entity file as **adder1** (it will be implemented in the later steps). Click on **Next**. Another window will ask which files you want to add to the project. Add **carryout.bdf**. Then click on **Finish**.
2. Select **Create Default Symbol** from the **Max+Plus II->File** menu to create an icon that will represent this circuit in future designs. Name the symbol file as **carryout.bsf**. Close this design file.

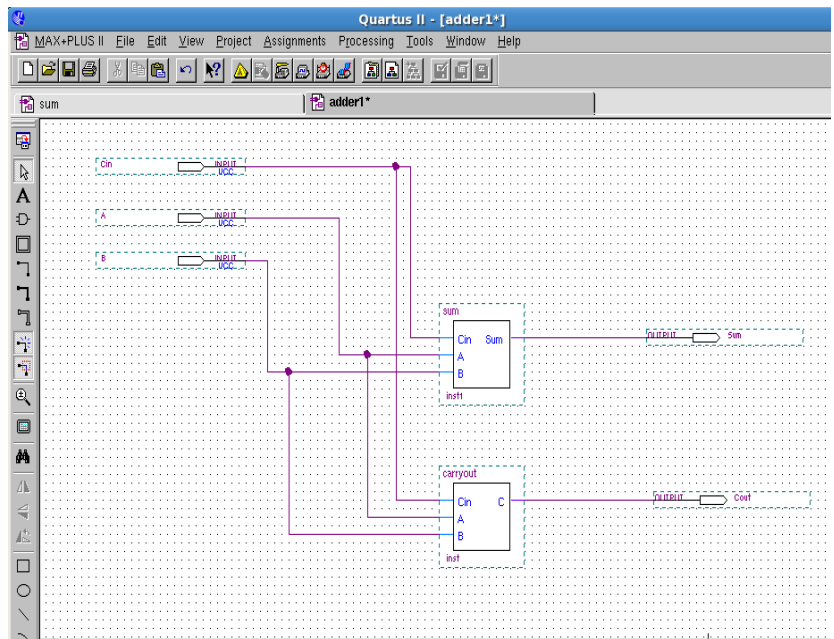


Figure 1: One-bit Adder (Full Adder)

3. Create a new file from **Max+Plus II -> File -> New**. Choose **Block Diagram/Schematic File**. Save this file as sum.bdf and do the same steps for the sum circuit (draw the circuit and create a default symbol, name the symbol file as **sum.bsf**)
4. Now create a new design file to represent the Full Adder. Name this file as adder1.bdf. You will need to build this from your two previous circuits; simply double-click on the graphic editor, Symbol window will pop-up, and from the Name dialog box in the pop-up, browse your quartus directory. This directory is where your default symbols are. Select your circuits as symbols. They will be inserted with a default icon that has the name of the file attached, and you can connect wires to them as if they were primitive gates. Build the full adder circuit which combines the carryout and sum circuits, as described in class (i.e., in the inputs are A, B, and Cin, and the outputs are C and S). Figure 1 shows the basic design in QuartusII.
5. Save the design as adder1.bdf, and compile the full adder. **Project → Save and Compile**. This will pop a compiler window to compile the design. As the Compiler processes the project, any information, error or warning messages appear in a Processing window that opens automatically. When the compilation is finished, icons representing the output files generated by the Compiler appear below the module boxes. QuartusII will give you advice on errors if you select an error message using the Message button. The most common error is that you didn't connect a device's inputs or outputs properly. Ask your TF about any errors that you can't figure out. Create the default symbol as usual.
6. Notes: If you double-click on the symbols you created, it will open the original .bdf file that implemented it; this is useful to remind yourself how it works. If you select Hierarchy Display from the Max+PlusII menu, you can see the relationship between the various files that make up a circuit.

Running a Circuit Simulation

Now you need to define the inputs to the circuit during the simulation, and actually running the simulation and observing the outputs. For this lab you will perform the simulation on the **adder1.bdf** file, observing what it does when you add $A=1$, $B=0$, $C_{in}=1$ to obtain $C=1$, $S=0$.

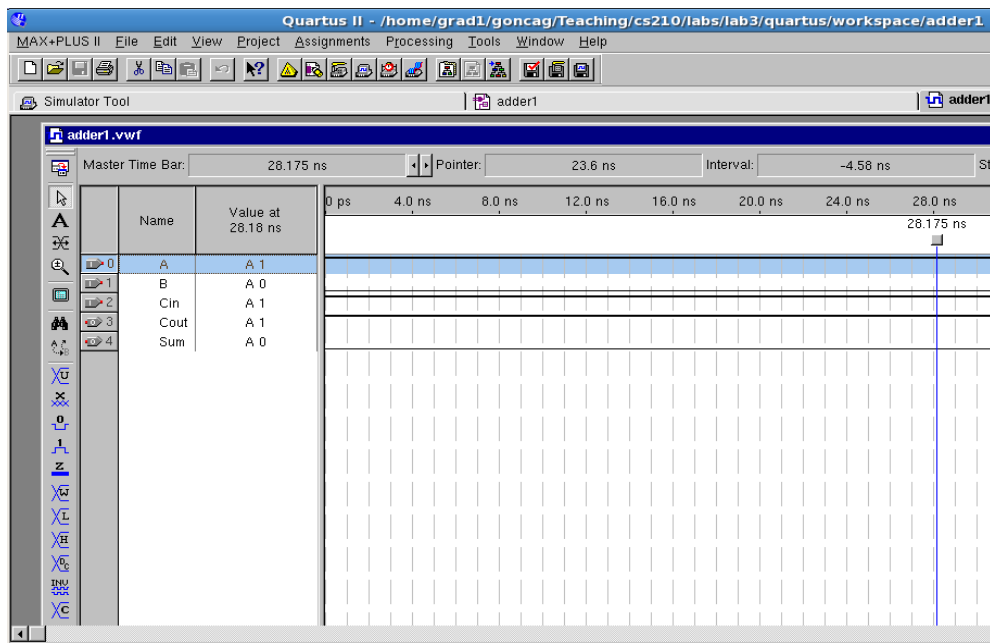
Creating a Waveform File

1. Choose **New** from the **File** menu, select **Waveform Editor File**. From File->Save As save the file adder1.vwf. Make sure adder1.vwf is in the same directory (your quartus work directory).
2. Choose **End Time** from the **Max+Plus II -> File** menu and type an end time of 1.0us (microseconds, "u" looks a lot like a Greek letter mu) if it not already typed in by default.
3. Choose **Grid Size** from the **Max+Plus II ->Options** menu, type 1.0ns and choose OK.
4. Choose **Enter Nodes from SNF** from the **Node** menu. The **Enter Nodes from SNF** dialog box is displayed. Click on **Node Finder**.

- From the Node Finder menu, select **Pins:all** for **Filter** and click **List** button to list the available input(I) and output(O) nodes.
- Choose the right direction button (-->) to copy the selected nodes in the Selected Nodes group. Then click OK.
- To enter input values in the Waveform editor, left click on the symbol (e.g., Cin) to edit its value. This will highlight the line with blue.
- Right click anywhere in the blue. Choose **Value** and make it High(1) if you want the input to be 1 or Low(0) if you want it to be 0. Alternately, use the buttons on the left toolbar (to assign values to the other inputs). Set the inputs to A=1, B=0, Cin=1.

Simulate the waveform and observe the output.

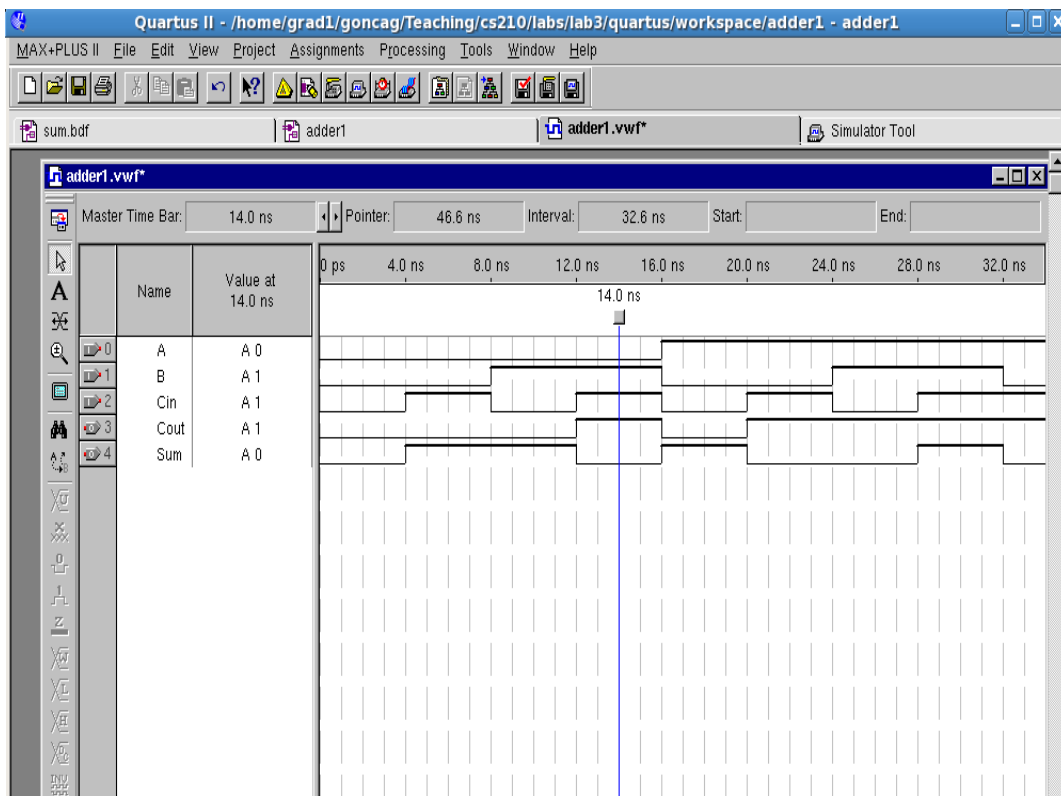
- Choose **Simulator** from the MAX+PLUS II menu, or click the appropriate button on the top toolbar (move the cursor over the buttons and read the bottom of the window).
- Choose an end time of 1.0 us in the pop up Simulator Tool dialog box.
- For **Simulation Mode**, select **Functional** and click on **Generate Functional Simulation List**.
- Click on **Start**. You can now look at the waveform window (behind the pop-up boxes) to see the values C=1 and S=0 on the output lines. It may ask you if you want to save the updated waveform file. Click yes. Then you will see the values in the “Value” column and also in the waveforms. The result is shown in the screen capture below.



- Now you should try setting different values at intervals of, say, 20 ns. The basic idea is that you may select not just the entire input line, but just a certain range. Say you want input signal A to be high during the period 150ns to 200ns. Click on the 150ns mark and drag the shaded section to 200ns and then click on the Value to 1 button on the left

toolbar. Now try to set the other inputs high or low at various times, and run the simulation again. You will observe that the outputs change to reflect the changing inputs (after some time delay for the signal to go through the circuit). By clicking at various points of time using the left mouse button, you can change the position of the “sample line” which gives the explicit values in the value column at those times; this is usually more readable than looking at the waveforms.

6. An excellent idea, for a circuit with a small number of inputs, is to systematically try all inputs by creating the sequence of inputs (for a three input circuit) 000, 001, ... , 111. This can be done as shown in the following screen capture. After the simulation step, you can either click on various locations, as mentioned above, or simply click on the arrows at the top of the window to move the “sample line” to the right and left through each distinct change in values.



Completing your design

Finally, create a 4-bit adder as described in class from four full-adder circuits. It should have inputs A3, A2, A1, A0, B3, B2, B1, B0, & Cin, and outputs S3, S2, S1, S0, & Cout. Save this as **adder4.bdf**, and do the process in step two to compile, test, and create a default symbol. You will submit the files created during this lab for the second homework.