**Example Lab: Unsigned 8-Bit Binary Adder With Quartus II, Altera DE2 Board, and Verilog HDL**

**Introduction**

This lab demonstrates how to program a DE2 Board to become an 8-bit binary adder. By toggling two sets of switches on the board, red LEDs light up to represent the sum of the two numbers in binary.

Verilog is a hardware description language (HDL) that can describe electrical systems. For this lab, it is much simpler to describe this circuit by using Verilog, rather than using several logic gates. Verilog is fairly easy to use if you have prior experience with programming languages. It is not expected for students to know all there is to know about Verilog, but they should know how the code in this lab works.

This is how the final block diagram schematic should look:

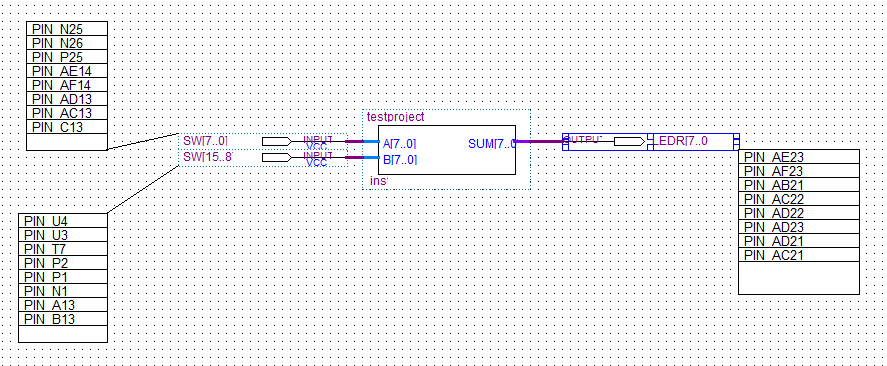


Figure – Block Diagram Schematic of 8-Bit Binary Adder in Quartus II software

And here is an example of the DE2 board with the 8-bit adder program loaded onto it, in action:

**What You Will Need**

* Quartus II Web Edition (license/subscription not required)
  + [**https://www.intel.com/content/www/us/en/software-kit/711791/intel-quartus-ii-web-edition-design-software-version-13-0sp1-for-windows.html**](https://www.intel.com/content/www/us/en/software-kit/711791/intel-quartus-ii-web-edition-design-software-version-13-0sp1-for-windows.html)
* Altera DE2 Cyclone II (For this lab, model number EP2C35F672C6 was used)
* Power Supply

**Instructions**

**Creating A New Project In Quartus II**

1. Launch Quartus II. Create a new project by launching the New Project Wizard from the pop-up window that appears.
2. If the Introduction appears, you can ignore it and click Next. Choose a directory for where you want to this project to be located. It is suggested that you create a folder in your Documents folder for this lab. You can name the folder ‘ExampleLab’, then set this as the working directory for the project in Quartus II. Give the project a name, like ‘Adder’. The top-level design entity will be automatically filled out as you typed. Once the project has been named, click Next.
3. You do not need to add any files. Click Next.
4. For Family & Device Settings, please be sure to enter the CORRECT device that you are working with. You can find your device name on its programmable chip:

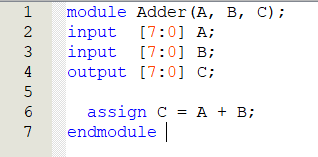
A close-up of a circuit board

Description automatically generated with medium confidence

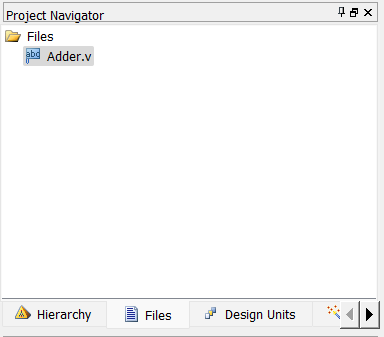
1. Since we are working with a Cyclone II Board, select Cyclone II as the Device Family from the dropdown. Then, in the Name filter on the right, type ‘EP2C35F672C6’. Available devices will be filtered as you type. Select your device under Available devices.
2. EDA Tool Settings page will appear. You can ignore this and click Next.
3. Click Finish. The New Project Wizard will disappear.

**Using Verilog To Describe A Circuit**

1. You now have a new project open in Quartus II. We will now create a Verilog file to describe an 8-bit adder. Go to *File>New…* which causes a window to appear. Under Design Files, select Verilog HDL File, then click OK.
2. A simple text editor will appear. Type the following to describe an unsigned, 8-bit binary adder:



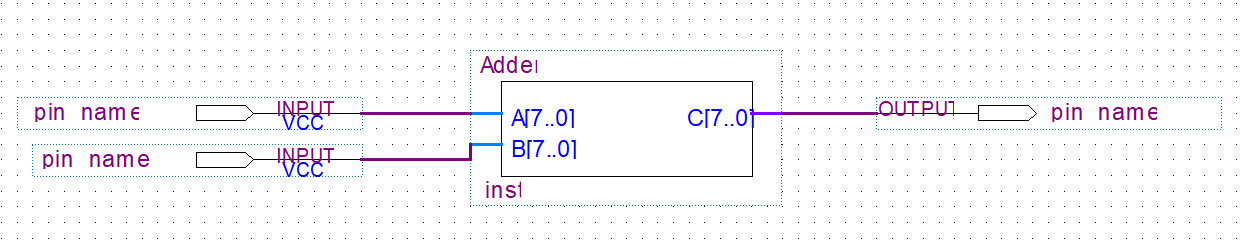
1. ***Description of Verilog Code:*** A module is a block of Verilog code, analogous to a function or method in other programming languages. This module is named ‘Adder’ and accepts 3 ports: A, B, and C. In the next three lines, A and B are described as 8-bit input wires while C is an 8-bit output wire. C is assigned to be the sum of A and B. All modules must end with the endmodule keyword.
2. Save the Verilog file with Ctrl + S. Give the file the same name as the module (‘Adder’).
3. On the left of your screen, locate the Project Navigator. Underneath it, click on Files. Find Adder.v and right-click it, then select Set as Top-Level-Entity from the menu that appears.



1. With the Verilog editor still open, now it is time to compile the code. Go to *Processing>Start>Start Analysis and Synthesis.* If compilation was unsuccessful, check your code for any typos, then save and re-compile.
2. Now create a symbol file so that it can be used a block diagram. Go to *File>Create / Update>Create Symbol Files for Current File.* If you get a prompt saying symbol file creation was successful, move on to the next step.

**Creating A Block Diagram Schematic**

1. Create a Block Diagram file by going to *File>New…* then select Block Diagram/Schematic File under Design Files in the window that appears. Select OK.
2. Find the Symbol Tool in the menu above the blank diagram. It should look like this:  Click on it, which causes a window to appear. Find the symbol file you just created by expanding the Project folder under Libraries. Your Adder should appear. Select it, then click OK. Place one Adder onto your diagram.
3. Next, add inputs and output. Find the Pin Tool in the menu above the diagram, which should look like this:  From the dropdown, select Input. Place two input pins to the left of the Adder you placed. Then, go back to the Pin Tool, and select output from the dropdown. Place one output pin to the right of the Adder.
4. Now we will connect the pins to the Adder. Select the Orthogonal Bus Tool, which should look like this:  Make sure it is the Bus Tool and NOT the Node Tool. The Bus Tool has a thicker line than the Node Tool, and has the same thickness as the lines coming from the Adder. Connect these lines by drawing a line from the end of one input to the end of a line from the Adder. Make sure a square appears, showing they are properly connected. Repeat this for the second input. Then, connect the output pin with the output of Adder using the same tool. Your block diagram should look similar to this:



1. Give names to the pins. For the input pin connected to A[7..0] triple click on pin name and give it the name SW[7..0]. Click OK. For the input pin connected to B[7..0] triple click on pin name and give it the name SW[15..8] then click OK. For the output pin connected to C[7..0] give it the name LEDR[7..0]. Click OK.
2. Save the Block Diagram file. Give it a different name from the Verilog file, like ‘AdderDiagram’. Go back to the Project Navigator, and set AdderDiagram.bdf as the Top-Level-Entity.
3. Compile the block diagram file by going to *Processing>Start>Start Analysis and Synthesis.* If compilation was unsuccessful, go back over previous steps and look for any mistakes before proceeding to the next part.

**Pin Assignment**

1. Before you can start working with the board, you must make sure pins are assigned correctly. This can either be done manually or automatically by importing a pin assignment file.
2. To do so manually, go to *Assignments>Pin Planner* and locate the list of Pins at the bottom of the window. Double click on location, and type in the proper location for each pin based on pin assignments given in the DE2 User Manual. For the EP2C35F672C6, pin assignments should appear like this:

Graphical user interface

Description automatically generated with medium confidence

1. If you already have a pin assignment file, you can import it into your project. To import pin assignments, you can go to *Assignments>Import Assignments.* In the window that appears, locate the file you would like to import, then click OK.
2. Exit out of the Pin Planner if it is open. We can now compile our entire project by going to *Processing>Start Compilation.* Compilation may take a few minutes. If compilation is successful, you are now ready to connect to the board.

**Downloading Project Onto The Board**

1. Connect your board into a power supply. Turn the board on.
2. Connect your board to your computer or laptop with the USB cable. Make sure the board’s end of the cable is plugged into the port named BLASTER, which is printed on the board.
3. Go to *Tools>Programmer.* At the top left of the window, click on Hardware Setup, and select USB Blaster from the list of Available hardware items from the window that appears.
4. Close out of the Hardware Setup window. Click Auto Detect from the list of buttons on the left in the Programmer Window. Your device should appear. If no output file is specified, double click on it and find the .sof file associated with this project (should be Adder.sof, in this example.)
5. Make sure the Program/Configure box is checked, then click Start from the list of buttons on the left. If compilation is successful, test your program on the board. When you flick the switches on and off on the board, what lights are lit up?