Cpr E 281 LAB06
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Programming the DE2-115 Board to Perform Binary Addition

PRELAB!

Read the entire lab, and **complete** the prelab questions (Q1-Q4) on the report template. Ensure that you submit your completed prelab to Canvas **before** joining the laboratory.

1.0 Objective

In this lab we will perform addition of two 4-bit binary inputs and display the answer in decimal. *(NOTE: All binary numbers in this lab are in unsigned representation.)*

2.0 Setup

Refer to **Figure 1**. We will use the inputs in ModelSim to provide two 4-bit inputs. Each set of four inputs represents one binary number. The inputs are wired to a block that adds the two numbers together. The result is processed to convert the results into two decimal digits (binary coded decimal or BCD representation is used for this purpose). For example, if we add number 7 (binary 0111) and number 5 (binary 0101), the result is 12, and it is displayed as two decimal digits, 1 and 2, after converting the result into two decimal digits each in binary, using the module **bcd_converter** block. The two decimal digits are fed to their respective 4-bit buses. **Use the same name for the pins as it is in Figure 1.**

***IMPORTANT**: Read through the rest of the document before wiring a circuit like **Figure 1.** There are instructions below that will save you lots of time.*

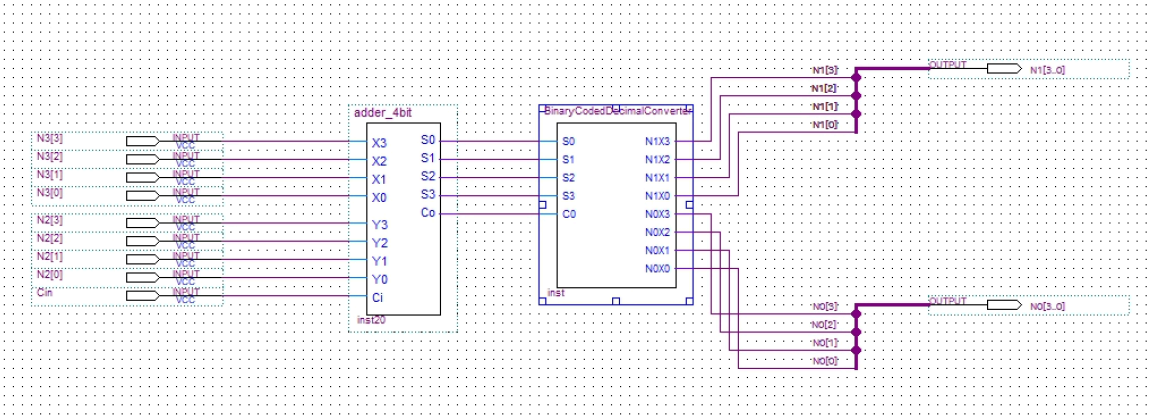


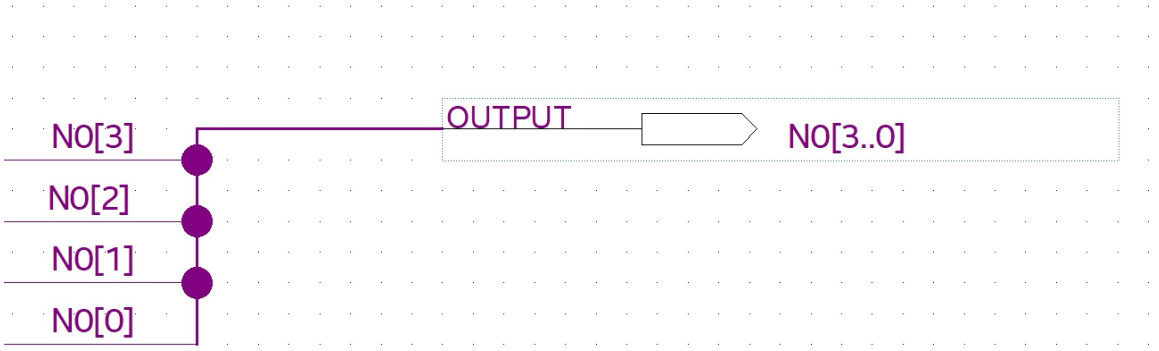
Figure 1: Adder with BCD Display

3.0 The Design

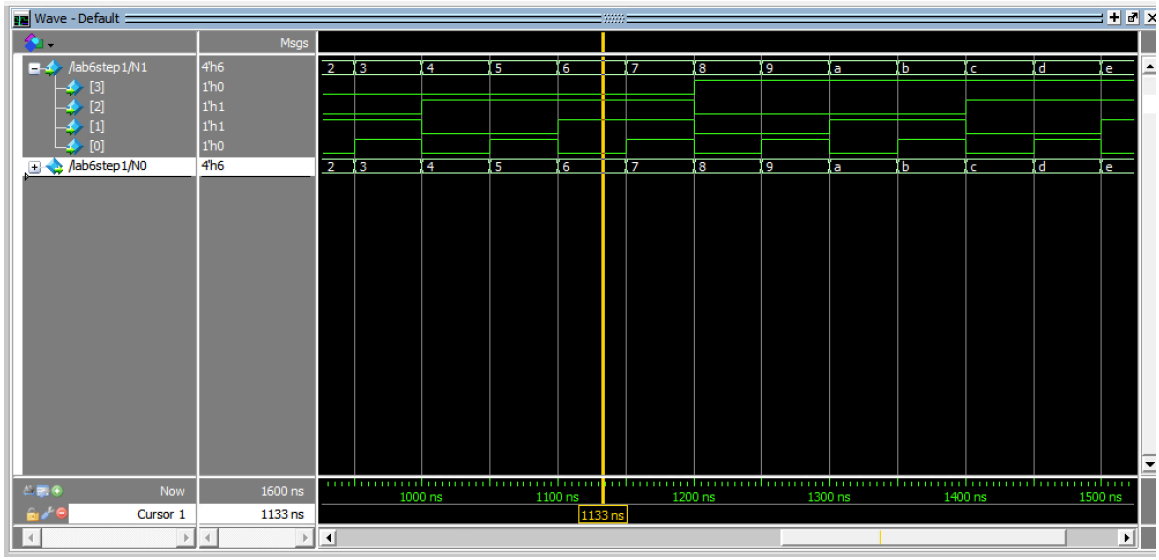
3.1 Displaying numbers in ModelSim

Since we don't have access to actual 7-segment displays we are going use ModelSim's built-in waveform to show the output in hexadecimal. Although the output of the BCD convertor is in decimal, we can represent that using hexadecimal digits in ModelSim. For displaying numbers you will only use one output pin and name it N#[3..0]. Next create a small node line (orthogonal node tool), one on the output of each of the 4 input lines.

Then run a bus line (**orthogonal bus tool**) to the output pin and connect each node line to the bus line. Next you must label each node line to establish its position in the bus (to name the node line use the same name as the output like N#[i] where i is the position on the bus). To do this simply right click the node line and give it a name. An example is shown in figure below.
Also, for the input pins, name them like N#[i] where i is the position of that bit. That will show the input in hexadecimal in ModelSim.



What you expect to see in the ModelSim should look like the figure below. If you click on the plus button by the name of the pin you can see each wire of the bus.



3.2 The 4-bit Adder

In this lab, you will use a 4-bit adder that you have not implemented yet, but it is provided to you. Download *full_adder.bdf* and *adder_4bit.bdf*, as well as their respective *.bsf* files, to your Lab06 directory from the files provided in your lab assignment folder. The full adder is used by the 4-bit adder to do the addition. The function of this block is to add two 4-bit numbers. Its output is a 4-bit number and a "carry-out bit" *C0*, or, in other

words, a five-bit number. **You have to make sure that you generate a Verilog HDL file for both *full_adder.bdf* and the *adder_4bit.bdf* and compile them in the ModelSim.**
Note:
You should have 4 files to compile in the ModelSim.

3.3 Binary Coded Decimal Converter

This block converts the adder output into two decimal digits, each of which is represented using 4-bits, and thus produces eight outputs for the two 4-bit buses. Here, for example, the output N1X0 is the least significant bit (X0) of the tens-place digit (N1) of the output, and {N1X3 N1X2 N1X1 N1X0} correspond to the four bits that represent this tens-place digit.

In prelab Q4, you will create Boolean expressions to represent each of these digits. Use these expressions to create a Verilog file named **BinaryCodedDecimalConverter.** Make sure you create a symbol file for this and use it in the main project. **Note**: you can use five variable k-maps to fully minimize each of these expressions, but you are only required to find a correct expression (not necessarily the most minimal one).

3.4 Test the BCD Adder

Verify that your BCD adder works as expected. To verify the additions performed in the prelab run **additiontest.do**. To verify the sums 0 to 31 run **lab06.do**.

In your report, include:

- your code for the BCD adder
- screenshot of your block design file
- screenshot(s) of the lab06.do file waveform.

IMPORTANT: Ensure that your waveform is zoomed in enough to see the value of each sum. You may have to zoom in within ModelSim and take multiple screenshots.

5.0 Complete

You are done with this lab. Ensure that all files are closed and then exit Quartus Prime, log off the computer, and submit your report. **Don't forget to include your name and your lab section number**.