

PRELAB!

Read the entire lab, and **complete** the prelab questions (Q1-Q4) on the answer sheet **before** coming to the laboratory.

1.0 Objective

In Lab 05, we created a circuit to display a 4-bit digital input (set on a set of toggle switches) onto a 7-segment display. In this lab we will perform addition of two 4-bit binary inputs and display the answer in decimal onto two 7-segment displays. (*NOTE: All binary numbers in this lab are in unsigned representation.*)

2.0 Setup

Refer to **Figure 1**. We will use the toggle switches available on the DE2-115 board to provide two 4-bit inputs. Each set of four switches 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 7-segment display. The two 4-bit inputs are also displayed using two 7-segment displays.

*NOTE: 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.*

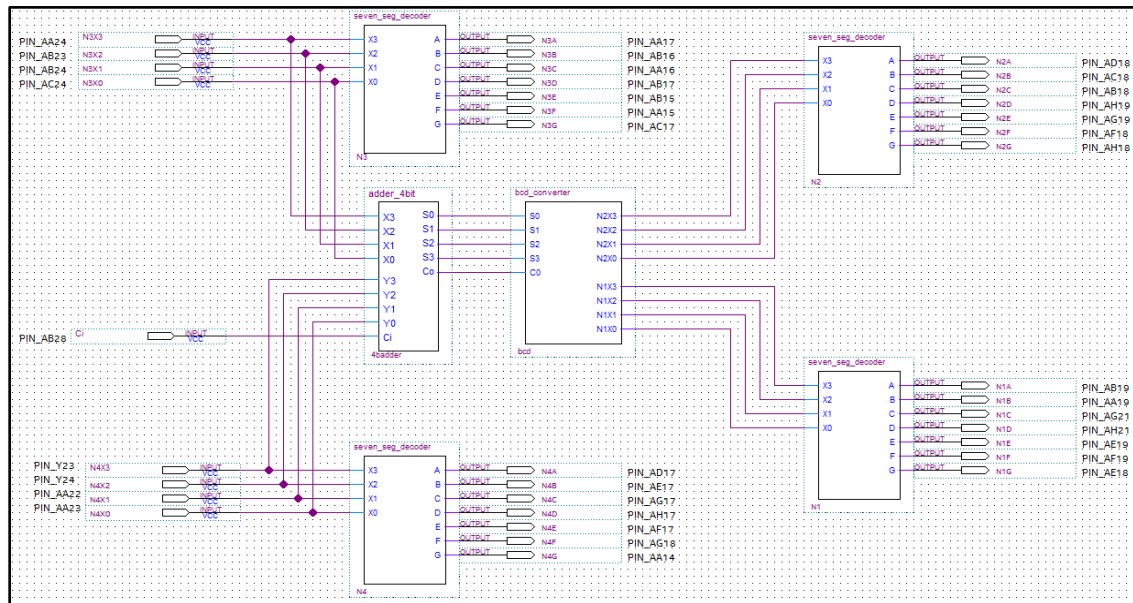


Figure 1: Adder with BCD Display

3.0 The Design

To start your project, create a New Project and name this project **Lab06**. Create a new Block Design File and save this file as **lab06.bdf**. This block design file will be the top-level design file; that is, this file will contain references to all other sub-files that will be compiled and simulated for this lab.

3.1 The 7-Segment Code Converter (*seven_seg_decoder*)

Use the 4-bit input 7-bit output seven segment decoder from Lab05 to display the 4-bit number onto a 7-segment display. Copy the *.bsf* and *.v* files for the symbol you want to use into your project folder.

NOTE: Recall from the last lab that it was tedious to create and assign all inputs and outputs. Therefore, rather than adding new symbols in the same way you add any new component, it is easier to copy the work you already have done and reuse it. To do this, just open your *lab5step1.bdf* file. Select all of the components and wires, then simply copy (Ctrl+C) them. Then in your current design, simply paste (Ctrl+V) them. It is important that you copy the *.bsf* and corresponding *.v* file to your project directory so that your design will compile. Compile your design once and fix any errors you might have before copying over the pin assignments.

Quartus Prime stores the pin assignments in a *.qsf* file. Go to your *lab5step1* directory and open the *lab5step1.qsf* file with a text editor such as Notepad++. Select the pin information and copy it as shown in **Figure 2**.

```
53 set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT
54 set_global_assignment -name PARTITION_LOCATION 16764057 -section_id Top
55 set_location_assignment PIN_AC27 -to N1X0
56 set_location_assignment PIN_AD27 -to N1X1
57 set_location_assignment PIN_AB27 -to N1X2
58 set_location_assignment PIN_AC26 -to N1X3
59 set_location_assignment PIN_AD26 -to N2X0
60 set_location_assignment PIN_AB26 -to N2X1
61 set_location_assignment PIN_AC25 -to N2X2
62 set_location_assignment PIN_AB25 -to N2X3
63 set_location_assignment PIN_AC24 -to N3X0
64 set_location_assignment PIN_AB24 -to N3X1
65 set_location_assignment PIN_AB23 -to N3X2
66 set_location_assignment PIN_AA24 -to N3X3
67 set_location_assignment PIN_AA23 -to N4X0
68 set_location_assignment PIN_AA22 -to N4X1
69 set_location_assignment PIN_Y24 -to N4X2
70 set_location_assignment PIN_Y23 -to N4X3
71 set_location_assignment PIN_AB19 -to N1A
72 set_location_assignment PIN_AA19 -to N1B
73 set_location_assignment PIN_AG21 -to N1C
74 set_location_assignment PIN_AH21 -to N1D
75 set_location_assignment PIN_AE19 -to N1E
76 set_location_assignment PIN_AF19 -to N1F
77 set_location_assignment PIN_AE18 -to N1G
78 set_location_assignment PIN_AD18 -to N2A
79 set_location_assignment PIN_AC18 -to N2B
80 set_location_assignment PIN_AB18 -to N2C
81 set_location_assignment PIN_AH19 -to N2D
82 set_location_assignment PIN_AG19 -to N2E
83 set_location_assignment PIN_AF18 -to N2F
84 set_location_assignment PIN_AH18 -to N2G
85 set_location_assignment PIN_AA17 -to N3A
86 set_location_assignment PIN_AB16 -to N3B
87 set_location_assignment PIN_AA16 -to N3C
88 set_location_assignment PIN_AB17 -to N3D
89 set_location_assignment PIN_AB15 -to N3E
90 set_location_assignment PIN_AA15 -to N3F
91 set_location_assignment PIN_AC17 -to N3G
92 set_location_assignment PIN_AD17 -to N4A
93 set_location_assignment PIN_AE17 -to N4B
94 set_location_assignment PIN_AG17 -to N4C
95 set_location_assignment PIN_AH17 -to N4D
96 set_location_assignment PIN_AF17 -to N4E
97 set_location_assignment PIN_AG18 -to N4F
98 set_location_assignment PIN_AA14 -to N4G
99 set_location_assignment -name PARTITION_METADATA root_partition -to |
```

Figure 2: lab5step1 Pin Information

Next, go to your Lab06 project directory and open the *.qsf* file there in a text editor. Paste the pin assignment information into the end of the file and save it. Then compile your design again. Now you are ready to modify your design by adding the last two components.

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” *CO*, or, in other words, a five-bit number. The “carry-in bit” *Ci* is assigned to SW0.

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 7-segment displays. Here, for example, the output N2X0 is the least significant bit (X0) of the tens-place digit (N2) of the output, and {N2X3 N2X2 N2X1 N2X0} correspond to the four bits that represent this tens-place digit. This block is created using Verilog as part of *Prelab Q4*.

3.4 Test the BCD Adder

Perform the additions you carried out in the prelab to test your circuit. Demonstrate your design to your TA.

4.0 Complete

You are done with this lab. Ensure that all files are closed and then exit Quartus Prime. Log off the computer, power down the DE2-115 board, and hand in your answer sheet to your TA. **Don't forget to write down your name and your lab section number.**