ECEN 4856: Laboratory Exercise 2

This is an exercise in designing combinational circuits that can perform binary-coded-decimal (BCD) addition and binary-to-decimal number conversion. In addition, this exercise is to design a combinational circuit that can multiply two unsigned numbers. First, you will design an array multiplier that does not use the multiplier LPM module from the Quartus II system. Consult your textbook for a description of the array multiplier. Then, you will use the LPM module to accomplish the same task, and compare the results achieved. To make the design task more manageable, start with a simple case of 4-bit numbers.

Part I

We wish to display on the 7-segment displays, HEX_3 to HEX_0, the hexadecimal values set by the switches $SW_{15–0}$. Let the values denoted by $SW_{15–12}$, $SW_{11–8}$, $SW_{7–4}$ and $SW_{3–0}$ be displayed on HEX_3, HEX_2, HEX_1 and HEX_0, respectively.

1. Create a new project which will be used to implement the desired circuit on the Altera DE2 board.
2. Write a VHDL file that provides the necessary functionality.
3. Include the VHDL file in your project and compile the project.
4. Assign the pins on the FPGA to connect to the switches and 7-segment displays, as indicated in the User Manual for the DE2 board.
5. Recompile the project and download the compiled circuit into the FPGA chip.
6. Test the functionality of your design by toggling the switches and observing the output display.

Part II

Design a circuit that can add two 2-digit BCD numbers as follows:

1. Use switches $SW_{15–8}$ and $SW_{7–0}$ to represent 2-digit BCD numbers A and B, respectively. The value of A should be displayed on the 7-segment displays HEX_7 and HEX_6, while B should be on HEX_5 and HEX_4. Write VHDL code to specify a circuit that generates C = A + B and displays the sum, C, in the BCD form on the 7-segment displays HEX_2, HEX_1 and HEX_0.
2. Compile the designed circuit and simulate its functional behavior.
3. Make the necessary pin assignments and recompile the circuit.
4. Download the circuit into the FPGA chip.
5. Test your circuit by trying different values for numbers A and B.

Part III

Design a combinational circuit that converts a 6-bit binary number into a 2-digit decimal number represented in the BCD form. Use switches $SW_{5–0}$ to input the binary number and

7-segment displays HEX_1 and HEX_0 to display the decimal number. Implement your circuit on the DE2 board and demonstrate its functionality.

Part IV

Design an array multiplier that multiplies 4-bit numbers, as follows:

1. Create a new project which will be used to implement the desired circuit on the Altera DE2 board.
2. Use switches SW_{11-8} to represent the number A and switches SW_{3-0} to represent B. The hexadecimal values of A and B are to be displayed on the 7-segment displays HEX_6 and HEX_4, respectively. The result C = A ∗ B is to be displayed on HEX_1 and HEX_0.
3. Generate the required VHDL file, include it in your project, and compile the circuit.
4. Use functional simulation to verify that your code is correct.
5. Assign the pins on the FPGA to connect to the switches and 7-segment displays, as indicated in the User Manual for the DE2 board.
6. Recompile the circuit and download it into the FPGA chip.
7. Test the functionality of your design by toggling the switches and observing the 7-segment displays.

Part V

Extend your multiplier to multiply 8-bit numbers. Use switches SW_{15-8} to represent the number A and switches SW_{7-0} to represent B. The hexadecimal values of A and B are to be displayed on the 7-segment displays HEX_{7-6} and HEX_{5-4}, respectively. The result C=A∗B is to be displayed on HEX_{3-0}.

Part VI

Change your VHDL code to implement the 8 x 8 multiplier by using the *“*” symbol*. Complete the design steps above. Compare the results in terms of the number of logic elements (LEs) needed.

Each group is to provide a lab report with well-documented VHDL code that clearly identifies each part. The lab report should include a coversheet, brief explanation of the how the solutions were developed and a summary of results.

Note: Useful VHDL reference: http://www.csee.umbc.edu/portal/help/VHDL/index.shtml