University of Windsor
Department of Electrical and Computer Engineering
06-88-443 Embedded System Design Lab Project 1
Winter 2017
Instructor: Dr. Aaron
Due: Monday February 27, 2017

You will design, model and simulate an **8-bit ALU** with Verilog and the Altera Quartus II CAD tool. You will implement it on the **ALTERA DE2-115 Development Board** and its LCD character display module. The ALU handles ten different operations on two 8-bit operands. Both operands (**Operand A** and **Operand B**) can have any unsigned 8-bit value while the result can have any unsigned 8-bit value. The following steps describe the functional behavior of the ALU:

1. At the beginning the LCD module shows: "`OPERATION =`".
2. The user specifies the type of operation (increment, decrement, addition, subtraction, bitwise NOT, AND, OR, XNOR, Left Shift, Right Shift) by setting the operation code (op-code) using the on-board switches **SW7-SW0** of the DE2-115 dev-board. The codes are given in the table below:

A = Arithmetic L = Logical = 1	**R = Rotate S = Shift**	**RR = Right L = Left**	**RSE = Rotate/Shift Enable**	**OP1 = Operator 1 OP2 = Operator 2**

Opcode								**Result**
RR = 0 L = 1	R = 0 S = 1	RSE = 1	A = 0 L = 1					
0	0	0	0	0	0	0	0	Result = OP1 + OP2
0	0	0	0	0	0	0	1	Result = OP1 **-** OP2
0	0	0	0	0	0	1	0	Result = OP2 **-** OP1
0	0	0	0	0	0	1	1	Result = OP1 * OP2
0	0	0	0	0	1	0	0	Result = OP1 / OP2
0	0	0	0	0	1	0	1	Result = OP2 / OP1
0	0	0	0	0	1	1	0	Result = OP1 **%** OP2
0	0	0	0	0	1	1	1	Result = OP2 **%** OP1
0	1	1	0	1	0	1	0	Result = OP1 Arithmetic Shift Right by LSB 3 bit of OP2
1	1	1	0	1	0	1	1	Result = OP1 Arithmetic Shift Left by LSB 3 bit of OP2
0	0	0	1	0	0	0	0	Result = OP1 **and** OP2
0	0	0	1	0	0	0	1	Result = OP1 **or** OP2
0	0	0	1	0	0	1	0	Result = **not** OP1
0	0	0	1	0	0	1	1	Result = **not** OP2
0	0	0	1	0	1	0	0	Result = OP1 **nand** OP2
0	0	0	1	0	1	0	1	Result = OP2 **nor** OP1

0	0	0	1	0	1	1	0	Result = OP1 **xor** OP2
0	0	0	1	0	1	1	1	Result = OP2 **xnor** OP1
0	0	1	1	1	0	0	0	Result = OP1 Logical Rotate Right by LSB 3 bit of OP2
1	0	1	1	1	0	0	1	Result = OP1 Logical Rotate Left by LSB 3 bit of OP2
0	1	1	1	1	0	1	0	Result = OP1 Logical Shift Right by LSB 3 bit of OP2
1	1	1	1	1	0	1	1	Result = OP1 Logical Shift Left by LSB 3 bit of OP2

As the user is adjusting the op-code value on the switches, the op-code sign (ADD, SUB, etc.) should be echoed on the LCD display following "`OPERATION =`".

3. The user presses push button 0 (KEY0) on the DE2-115 dev-board to indicate the insertion of the op-code on the DIP switch.
4. Next, the LCD module shows: "`OPERAND1 =`".
5. The user specifies the value for the first operand using the onboard 8-bit switches **SW7-SW0**. The unsigned hexadecimal equivalent of the value specified on the DIP switch should be echoed on the LCD following the "`OPERAND1 =`" phrase. When the desired value is set on switches **SW7-SW0**, the user presses push button 0 (KEY0) to indicate the insertion of the first operand.
6. Next, The LCD display is cleared and shows "`OPERAND2 =`".
7. The user specifies the value for the second operand using the onboard switches **SW15-SW8**. The unsigned hexadecimal equivalent of the value specified on the switches **SW15-SW8** should be echoed on the LCD following the "`OPERAND2 =`" phrase.
8. When the desired value is set on the DIP switch, the user presses push button 0 (KEY0) to indicate the insertion of the second operand.
9. The LCD screen is cleared and it shows "`ANSWER =`".
10. The LCD should show the result of the function directly following the phrase "`ANSWER =`". Show "`OVERFLOW`" if the number should be less than 0 or above "FF". Assume that for the case of shift, rotate operand2 value given will be between 0 and 7 (3 LSB of the 8 bits. Example: XXXXX**110**).
11. If the user presses KEY0 again the ALU goes back to step 1.
12. In any step if the user presses push button 1 (KEY1) on the DE2-115 dev-board, the ALU should go back to step 1. (Reset)

Report:

You will need to write a 3 to 6 page report for this project. Describe how you implemented the ALU, LCD controller, etc. and what problems or difficulties you have encountered, and how you solved them. Make sure to include the Simulation Results showing an example of each operation.

Useful Hints:

- Simulate the operation of your Verilog code by testbench written in verilog using Modelsim Tool. This tool is an excellent way to verify that your code is operating as you planned and according to the specs. You only need to show the TA simulation of calculation part of the ALU, not the display.
- For simplicity assume that all operands and results are unsigned.
- The DE2-115 dev-board schematic is available on the course website. Please read it carefully. As senior engineering students, you are expected to be able to study and understand engineering data sheets effectively.
- Description of the **HD44780 LCD module** and its operation is provided in the datasheet. Read carefully Tables 23/24 and page 52/53 for timing, Tables 11/12/13 for example operation and Table 4 for character encoding. You can choose to use 4 or 8 bit operation mode. Also see Section 4.5 of the DE2-115 dev-board user manual for pin assignments. Notice that you may need clock divider and state machines to meet the timing requirement and interface with LCD.
- It is highly recommended that each group writes the Verilog code BEFORE coming to the lab. This will save significant amount of their time.
- TA will check the results of simulation and functionality of the design on the DE2-115 dev-boards. After that TA will ask oral questions related to the project to each team member. Please ensure that each member of the group have thorough understanding of the project. All group members should participate in developing the code and simulation.
- You will be marked on the functionality, simulation, oral questions and report.
- Each group MUST email their Verilog code to the instructor and GAs every week (every Monday). This will help us track your progress.
- Late submissions will not be accepted.
- Copied code will result in a grade of 0 and other penalties.