Laboratory 5, Total Points: 30

Due Date: March 21, 2019 (Thursday), 11:59 PM CST

Textbook Sections: Digital Design and Computer Architecture: 5.2.4

Arithmetic and Logic Unit is an essential component for a processor. In this lab, you will design a 32-bit Arithmetic Logic Unit (ALU) in VHDL. The ALU will provide four functions: ADD, SUB, AND, and OR. You will also write a VHDL testbench file to test the ALU.

Acknowledgement: The lab assignment is adopted from authors' website.

VHDL Code

Create a 32-bit ALU module in VHDL. Name the module alu.vhd. It should have the following entity declaration:

```
entity alu is
     port(A, B: in std_logic_vector (31 downto 0);
          ALUControl: in  std_logic_vector (1 downto 0);
          Result: out std_logic_vector (31 downto 0);
          ALUFlags: out std_logic_vector (3 downto 0));
end entity alu;
```

The four bits of **ALUFlags** should be TRUE if a condition is met. The four flags are as follows:

ALUFlag bit	Meaning
3	Result is negative
2	Result is 0
1	The adder produces a carry out
0	The adder results in overflow

Note: An adder is a relatively expensive piece of hardware. Be sure that your design uses no more than one adder.

Table 2: ALU Operations

Test	ALUControl[1:0]	A	B	Result	ALUFlags
ADD 0+0	0	00000000	00000000	00000000	4
ADD 0+(-1)	0	00000000	FFFFFFFF	FFFFFFFF	8
ADD 1+(-1)	0	00000001	FFFFFFFF	00000000	6
ADD FF+1	0	000000FF	00000001		
SUB 0-0	1	00000000	00000000	00000000	6
SUB 0-(-1)		00000000	FFFFFFFF	00000001	0
SUB 1-1		00000001			
SUB 100-1		00000100			
AND FFFFFFFF, FFFFFFFF		FFFFFFFF			
AND FFFFFFFF, 12345678		FFFFFFFF	12345678	12345678	0
AND 12345678, 87654321		12345678			
AND 00000000, FFFFFFFF		00000000			
OR FFFFFFFF, FFFFFFFF		FFFFFFFF			
OR 12345678, 87654321		12345678			
OR 00000000, FFFFFFFF		00000000			
OR 00000000, 00000000		00000000			

VHDL Testbench

Before writing the testbench, fill the missing cells in column 2, 4, 5, and 6 in Table 2. Now write a VHDL testbench that takes ALU inputs from Column 2, 3, and 4 and generates outputs. The produced outputs should be similar to Column 2 and 4.

Note: Remember that each hexadecimal digit in the table represents 4 bits. You may use hexadecimal values for signals by prefixing the value with X, e.g. A <= X"FFFFFFFF".

Submission

Submit the ALU block diagram (5 pts), completed Table 2 (5 pts), VHDL code and testbench (15 pts), and simulation results (5 pts).