Jordan University of Science and Technology
Computer Engineering Department
CPE 552-Computer Design - Project 1
Dr. Lo'ai Tawalbeh-Spring 2007

Attention: Individual project.

Design an 8-bit ALU that has two data inputs A and B, and performs the following operations:

- addition/subtraction ($A + B$ and $A - B$);
- unsigned multiplication ($A * B$);
- transference of the value at input B to the output;
- logic operations: A AND B, A OR B, A XOR B, and A';
- shift A left, shift A right;
- rotate A left, rotate A right.

Each one of these operations is selected by a 4-bit function code F as shown in Table 1. You **must** implement the ALU according to the given function codes. The use of the ALU in the CPU design (Project 3) and **the test/grading of your circuit by the TA will depend on this requirement**. The block diagram for the ALU is shown in Figure 1.

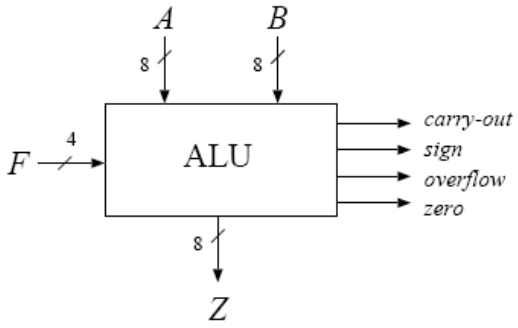

Figure 1: 8-bit ALU block diagram

The ALU must be described using a structural model that represents the main ALU blocks, as presented in chapter 1 of the class notes. Besides the multiplication block, you may describe each ALU block using behavior or structural model. The multiplier must be defined based on its structure. In this case you may define a basic cell using behavioral model and use this cell to build the array structure for multiplication. The multiplication of two 8-bit vectors has a result represented in a 16-bit vector. The most-significant byte

F	z
0000	A'
0001	shiftl A
0010	shiftr A
0011	rotatel A
0100	rotater A
0101	$A + B$
0110	$A - B$
0111	A and B
1000	A or B
1001	A $\oplus$ B
1010	–
1011	–
1100	B
1101	$A * B$ (MSB)
1110	$A * B$ (LSB)
1111	–

Table 1: Function codes for 8-bit ALU

(MSB) or the least-significant byte (LSB) of the multiplication result is provided as an ALU's output based on the function code F (1101 or 1110 respectively).

The ALU must generate status information on the operation or the output value. This information is coded in the following bit signals: *zero, carry-out, overflow*, and *sign*. The definition of these signals was presented in class and it is also available in the class notes.

The interface of the ALU is defined as follows:

```
entity ALU is
port (A, B: in std_logic_vector (7 downto 0);
      F: in std_logic_vector (3 downto 0);
      zero,sign,overflow,carry_out: out std_logic;
      Z: out std_logic_vector (7 downto 0));
end ALU;
```

What should you turn in?

For this project you should turn in all the VHDL files used in the ALU design. The TA for the class will post instructions on the procedure to submit these files electronically. The testbench for the ALU will also be posted, so you have a chance to test your design using the same testbench that will be used by the TA to grade your project.

Each student must also turn in a short report that explains how the VHDL description was organized and shows the derivation of any switching expressions or functions that were used in the VHDL code. Printouts of the VHDL code or simulation results are not required, please, do not attach them to your report.