Notes for Lab Assignment 5

1. For this assignment, you need not design at the gate level. You can use any form of description permitted by VHDL - structural (component instantiation), dataflow (concurrent assignments) or procedural (processes). You may use "+" operation with appropriate type conversions, rather than getting into gate level design of adder.

2. For conversion from std_logic_vector to integer use the following.

   to_integer(signed(expression_of_type_slv))

3. For conversion from integer to std_logic_vector use the following.

   std_logic_vector(to_signed(expression_of_type_integer, 32))

4. Your ALU design needs to produce all zero outputs for instructions that are not implemented or are undefined, in order to be compatible with the refernce design during automatic checking. Please refer to slide 20 of Lecture 11, dated 13.02.2017 for this purpose.

5. The input port "shifter_carry" is meant for passing the carry generated by the shifter to the ALU, where it may be used for logical instructions. However, I have missed out this in the assignment description as well as the reference design. Therefore, for the present assignment, do nothing with this port. Let it remain as a part of the entity declaration for the sake of compatibility.

6. Generate next_N, next_Z, next_C, next_V as you would do for add instruction, without bothering about which instruction it actually is as long as it is an instruction being implemented. Whether these signals are used to update the flags or not will be checked elsewhere.

7. Note that multiplier and shifter/rotator are not part of this exercise.