

**Project Assignment #6****DUE: Friday May 2<sup>nd</sup>**

In this assignment you will be writing *yet* another alternative architecture for the 8 bit ALU. The shell for this project step is in pr\_step6.vhdl. In this architecture you will be using a package and modifying the ENTITY of the 8 bit ALU to use a symbolic name for the operation rather than the P, K, and R binary values. Also, the procedures are moved from being declared in the process to the package. You will need to add a “USE” statement to the Entity to make the package procedures visible.

The ENTITY will have Pctl, Kctl, and Rctl replaced by alu\_op (This is why the declaration is moved to the package):

```
ENTITY alt_8bit_alu IS
  PORT (
    alu_op : IN operations;
    A, B   : IN BIT_VECTOR ( 7 downto 0);
    Cin    : IN BIT;
    Zout   : OUT BIT_VECTOR ( 7 downto 0);
    Cout   : OUT BIT );
END alt_8bit_alu;
```

Modify your ARCHITECTURE from step 5 such that the CASE statement uses alu\_op to determine the operation to be performed as illustrated below.

```
CASE alu_op IS
  WHEN op_A => Zout<= A; Cout <= '0';
  •
  •
  •
  WHEN op_negA => neg(A, Ztemp, CoutTemp);
                  Zout <= Ztemp; Cout <= CoutTemp;
  WHEN OTHERS => NULL;
END CASE;
```

1) First copy the file pr\_step6.vhdl. Note that you will need a package. Enter the package declaration, package body, ALU ENTITY and ALU ARCHITECTURE as noted.

2) Move the procedures that you used for binary addition, 2’s complement, and binary subtraction to the package. Remember that these are procedure bodies. You also need the procedure declaration in a package declarative design unit (Those items visible to other design units).

Note that this entity uses the type “operations” and must have access to the type declaration (you will need to make minor modifications to the testbench ARCHITECTURE in regards to the declaration for type operations ). Also remember the usage and function the of the USE clause (re. Use Navabi as needed).

3) Compile and simulate your design.

4) Turn in:

- a) a copy of all VHDL code
- b) a listing showing the results of simulation
- c) a waveform of the complete simulation using Zoom->Full Size

Once again there is a .do file which lists the desired signals.

NOTE: If you are getting strange results during simulation that don't seem to make sense, add signals to your architecture(s) so that you can see what is going on. Then use structure to descend the hierarchy and add these signals to the waveform.