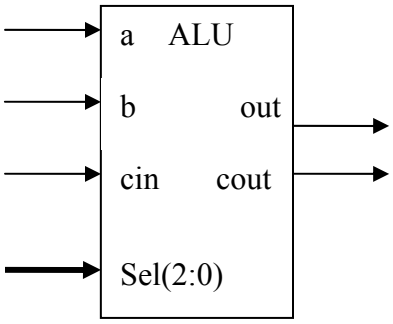
Assignment # 3
February 15th, 2007
Due: February 20th, 2007, 9pm.

The purpose of this assignment is to understand resolved type & associated resolution functions, and the use of configuration declarations.

1 Warm-Up

Your will use the 1-bit ALU that you created in Assignment 2 modified slightly as shown below.



S2	S1	S0	Function
0	0	0	Identity (a)
0	0	1	a AND b
0	1	0	a XOR b
0	1	1	0
1	0	0	Identity (a)
1	0	1	a AND b’
1	1	0	a XOR b’
1	1	1	a – b

Create two implementations of this ALU as follows.

1. one using concurrent signal assignment statements
2. one using a process construct
3. Place the entity description in a separate file.
4. Place each of the architecture descriptions in separate files.

Create an 8-bit ALU model that instantiates 8 bits as follow using one of the single bit ALUs

1. a component instantiation statement for bit 0. Note the use of the input carry for creating subtraction operations.
2. a generate statements as in the previous assignment for bits 1-(width-2) note that this ALU must have parametric width.
3. a component instantiation statement for the most significant bit.

Note that this muti-bit ALU is parametric. You will instantiate an 8-bit version as above. Test this ALU and attach the trace to demonstrate that it works. This part is virtually identical to Assignment 2.

2 Main Assignment

The main part of the assignment has two parts.

2.1 Using a Configuration Declaration

You have created two functionally equivalent descriptions of the single bit ALU. Call the entities ALU-dataflow and ALU-process. Now write a configuration declaration for the 16 bit ALU such that

1. bit 0 and bit (width-1) uses ALU-dataflow. The remaining use ALU-process. Compile, & execute with your testbench. Use the same test bench and test vectors for the warmup. You should get the same trace.
2. Modify the configuration declaration to use the ALU-process for bits 0 and MSB and ALU-dataflow for bits 1 through (width-2).
3. What files do you actually have to recompile when you go from 1-2? Make sure you check.

2.2 Using A Resolution Function

In the multi-bit ALU design create a new signal that is a resolved type and is a subtype of std_ulogic. Call this signal error. Write a resolution function for signals of this new resolved type such that the resolved signal has a value of 1 if any of the drivers has a value of 1.
Now add two concurrent processes to the architecture of the multi-bit ALU. The first process will check if the result is 0. If so drive the error signal to 1. Add a second concurrent process that will check if there is overflow – the carry into the MSB is not equal to the carry out from the MSB. If there is overflow then drive error to 1.

1. What happens to your model if error is not a resolved type?

3 Submission Instructions:

Name the files using the convention from the last assignment and zip all files into a compressed folder, and email to **mbales3@gatech.edu** by 9 pm on the due date. Place 4170 Assignment 3 in the subject line.

Note I will post solutions at 9 am on February 21st. There will no late assignments accepted. Consider this equivalent to studying for the test.