

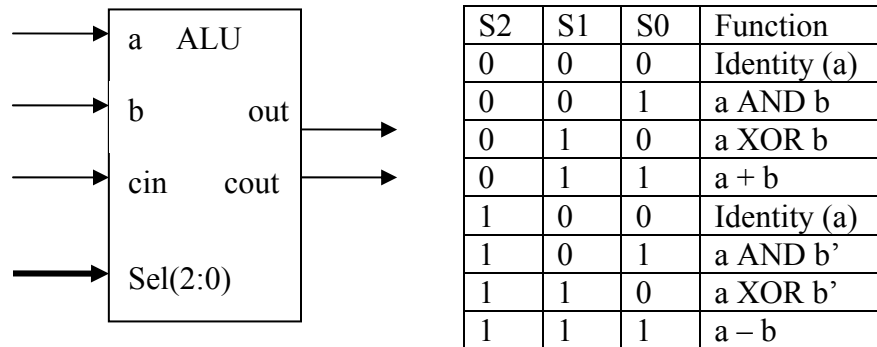
Assignment # 2
February 7th, 2007
Due: February 14th, 2007

The purpose of this assignment is to exercise structure, component hierarchy, and generic statements, and to introduce generate statements.

1 Warm-Up

This part of the assignment is intended to help you prepare for the main part of the assignment as well as create a few useful code pieces that you can use.

Create a 5-function, 1-bit ALU that has single-bit data inputs **a** and **b**, and a 3-bit select bus. S1 and S0 basically choose the function, and could be the select inputs to a 4-1 MUX for choosing which function is seen on the ALU output: **a**, AND, XOR, or Full Adder.



When S2 is high, the input **b** is inverted. This allows the FA to perform two's complement subtraction. Thus, when $S = b'111$, the **cin** input to the FA should be set to 1 (thereby performing $b' + 1$), i.e., subtraction.

Finally, associate two generic parameters with the ALU component: one to control **cout** delay, and one to control **out** delay.

2 Main Assignment

Now your task is to design and implement a generic ALU with ripple carry using generate statements to create an **n**-bit ALU. This component will have three generic parameters: **cout_delay**, **out_delay**, and **bit_width**. Note that the **cout** from each slice will have to be connected to the **cin** of the next most significant slice (ripple carry). Make the default width 16 bits.

When testing the generic ALU, use a bit width of 8. Design your testbench to instantiate an ALU of this width.

3 Submission Instructions:

Your submission should include three files: your single-bit ALU component, the generic bit-width ALU component, and the testbench used to verify the 8-bit ALU. Use the following naming convention:

- lastname_firstname_alu1.vhd
- lastname_firstname_gen_alu.vhd
- lastname_firstname_gen_alu_tb.vhd

Zip all three files into a compressed folder, and email to mbales3@gatech.edu by 9 pm on the due date. Place **4170 Assignment 2** in the subject line.