# Lab 9: ALU Design
# Last Lab

*TA: Andrew White*
*Nov 19, Nov 24, Nov 26, Dec 1*

## 1 Overview

In this lab, you will use the structural and behavioral design techniques from previous labs to build an essential datapath component: the ALU.

## 2 Lab

### 2.1 Description

For this lab, you are to design a basic generic width ALU using structural VHDL. The ALU needs to have four selectable operations, as seen in the table below:

| $S_1S_0$ | Operation | Description |
|---|---|---|
| 00 | Add | Z = A + B |
| 01 | Subtract | Z = A - B |
| 10 | OR | Z = A OR B |
| 11 | AND | Z = A AND B |

Table 1: ALU operations where output based on 2-bit select line.

### 2.2 Procedure

You are to create the generic width ALU using the following guidelines:

- MUST BE GENERIC WIDTH! (excuse the redundancy)
- The ALU should have three inputs. A and B are generic width operand inputs. S is a 2-bit select line.
- The ALU should have two outputs. Z is the generic width output for the ALU result. Cout will be the 1-bit carry out of the Adder/Subtractor. Function of Cout does not matter when performing non-arithmetic operations.

- The ALU will not detect overflow nor perform any other special functions.
- Only one Adder/Subtractor component should be used within the ALU. This single component will need to perform both addition and subtraction.
- A multiplexer type component must be used for selection of appropriate output of ALU. Any multiplexer component can be described in behavioral VHDL.
- The ALU itself should be purely structural VHDL design.
- The Adder/Subtractor component should be purely structural VHDL design. Will be built using the full adder slice from a previous lab and any basic gates (NOT, AND, OR, NAND, NOR, XOR) necessary.
- Any basic gates required can be described in VHDL as single bit width or generic bit width.

To test your ALU, instantiate an 8-bit version of the ALU in the testbench and test all operations to verify function.

### 2.3 Report

The lab report will consist of a cover sheet, a block diagram either neatly hand-drawn or created using a drawing program (visio, paint, xfig, etc.) showing the structure for a 4-bit instance of your ALU, the VHDL code for the entire design except the full adder slice, the test bench used, and the simulation waveform. The waveform should be clearly labeled what operation is being performed and your own hand written verification that the result given is what is expected.