Assignment 1 – ALU Design

# Introduction

The assignment required us to implement a 6-bit ALU using the 6-bit Carry Ripple Adder (CRA) from lab C, the 8-bit greater than or equal to (GTHE) module from lab B and the XNOR module from lab A. The top level ALU module could not have any logical gates in it, but only instantiations of the modules mentioned above. The ALU needed to perform different tasks based on the value of then “fxn” input, these are summarized in the table below.

|  |  |
| --- | --- |
| Fxn | X[5:0] |
| 000 | A |
| 001 | B |
| 010 | -A |
| 011 | -B |
| 100 | A<B |
| 101 | A XNOR B (Bitwise) |
| 110 | A+B |
| 111 | A-B |

# Modules

As stated, before only modules designed in previous labs could be used in this assignment. However, some minimal modifications needed to be made to the files in order for them to work. For example, the GTHE module was designed without 2s compliment numbers in mind. So, in order for it to work as a 2s compliment less than module, I needed to implement some logic to check if the numbers were negative and then in response either return the result of the greater than module or return the logical not of the result. A wrapper module also needed to be created for the XNOR module from the first lab, since that was a 1-bit XNOR module and the ALU required it to be a 6-bit bitwise XNOR. Aside from these changes, nothing else about the modules were changed. The 6-bit CRA module was used as is, without any changes.