Comp1036 – ALU Assignment

Introduction

Previous exercises have used combinatorial logic and sequential logic to construct many of the various logic circuits that form the basis of a computer. In this exercise, we will put some of these circuits together to build a custom 8-bit or 16-bit ALU chip. **16-bit is recommended due some limitations in the Hardware simulator**

Tasks

1. To implement a chip that should provide the following functionality: **(maximum of 2 marks each, one for successful implementation, one for documentation/optimality/etc, 14 marks total):**

1) *Negate the value of x (eg. 3 becomes -3)*
2) *Increment the value of x (eg. 3 becomes 4)*
3) *Decrement the value of x (eg. 3 becomes 2)*
4) *Add x to y (eg. When x = 3 and y = 4 then out = 7)*
5) *Subtract y from x (eg. When x = 3 and y = 4 then out = -1)*
6) *Subtract x from y (eg. When x = 3 and y = 4 then out = 1)*
7) *You chip should automatically remember the output value and use it for the next iteration's x value (eg. x at time t+1 = out at time t) – as such you will need to be able to reset both x and y values to 0*

To achieve this you will use 6 binary control bits which each perform the following function

nx	if nx then x = Not(x)
zy	if zy then y =0
ny	if ny then y = Not(y)
no	if no then out = Not(out)
cx	if cx then the value used for x at t+1 = out
r	if r then x =0 and y =0

A test script is provided to test this. If your code does not pass all the tests you should reorder the test script to demonstrate as many passes as possible.

2. As well as the 6 designated control bits you have the option of a 7th control bit to perform an additional task of your own choosing. This will be worth up to 5 marks depending on complexity. **(Maximum of 5 marks in total.)**

Choose 1 of the following 2 options:

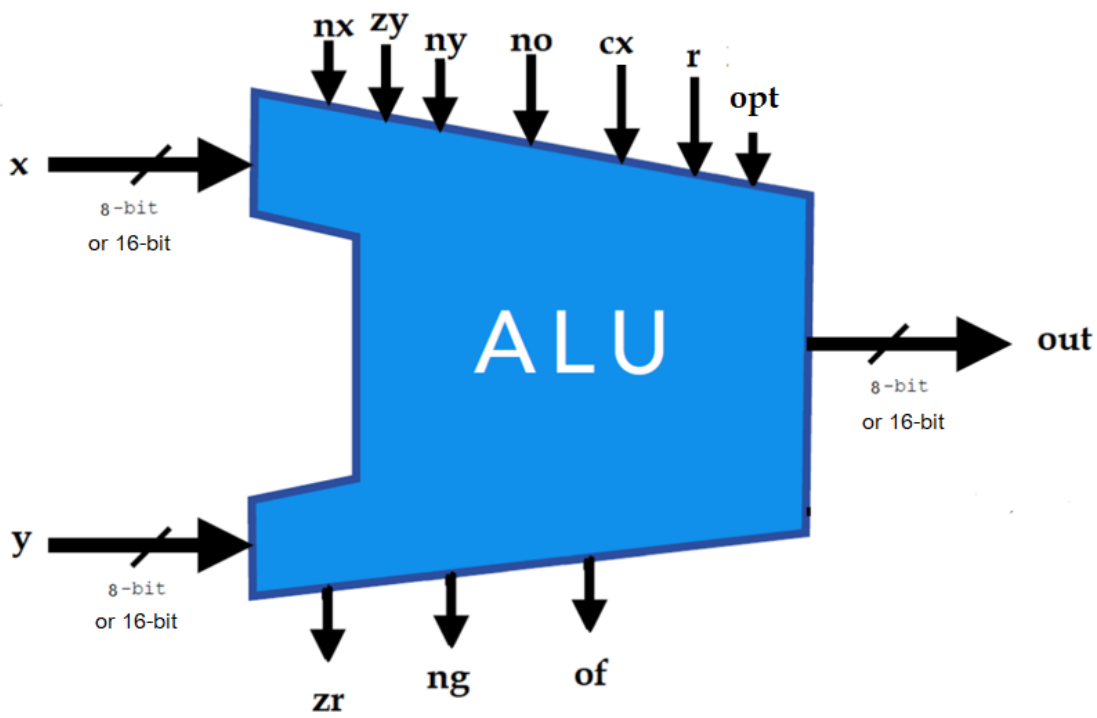
1) Multiply x by y (eg. When x = 3 and y = 4 then out = 12) – ***worth up to 5 marks***

2) Decide if x is a bigger number than y

out = -1 for false, 1 for true, 0 for same - ***worth up to 3 marks***

You should edit and comment the test script to demonstrate your 7th control bit. We also reserve the right to manually test your chip by entering our own values.

The structure of your chip should be as follows:



As well as the data output there should be 3 functional output bits

zr = this will be true if the output is 0

ng = this will be true if the output is negative

of = this will be true if the output exceeds the range of an 8 or 16 bit 2s complement number

3. Report (6 marks in total)

One A4 page of documentation that describes aspects of your implementation (including a gate diagram for your optional 7th bit) and explanation of what your optional control bit does

Submission requirements

You are required to submit via moodle the following files in one .zip file before **16:00pm 28th November**

The HDL file for your ALU chip, this will be named calu.hdl

The HDL files for any custom chips required by your HDL chip

One A4 page of documentation that describes your implementation (including a gate diagram for the optional chip and explanation of what your optional control bit does)

A test and compare file to demonstrate JUST your 7th control bit. The other 6 bits will be tested using the supplied test script.

Marking Schema

Up to 6 marks will be awarded for documentation

Up to 14 marks will be awarded depending on optimality, functionality and transparency of your software and if the chip passes the test script comparison.

Up to 5 marks will be awarded for the 7th control bit function

Note: Some students who submit code that is either extremely complex or extremely commonly occurring will be asked to verbally discuss and demonstrate their submission, their marks will be scaled depending on the outcome of this. You are reminded of the School's Policy on Plagiarism.

Late Submission

University standard late submission policy applies, i.e., 5% mark deduction for every 24 hours.