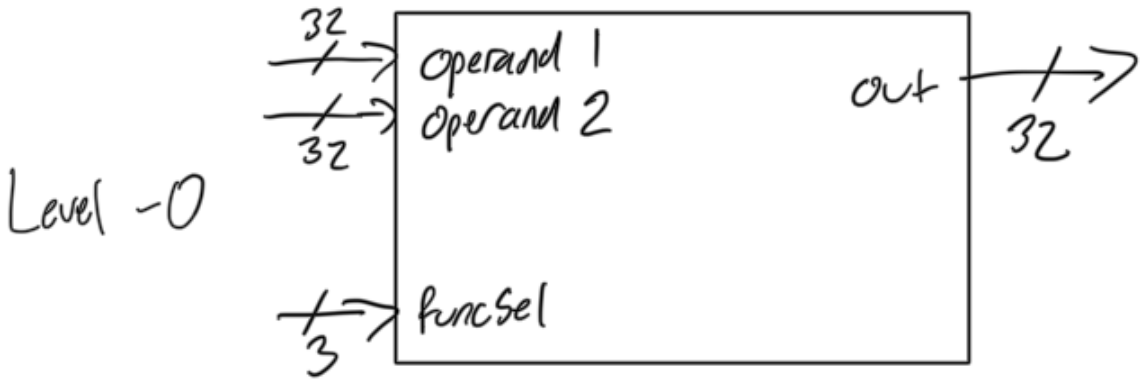
Lab 4 — Arithmetic Logic Unit

Objectives
For this lab, I am creating an arithmetic logic unit (ALU) as a part of a CPU which can implement several instructions. These include arithmetic addition, logic bitwise xor, logic bitwise and, logic bitwise or, logic bitwise nor, register right logic shift, and register left logic shift.

Introduction
An arithmetic logic unit is a critical part of a CPU. This is what performs arithmetic operations on data stored in memory. Because the purpose of a CPU is to manipulate data in a useful way, having these basic functions is crucial for functionality. My ALU has 7 basic functions which should be enough to demonstrate the concept of an ALU.

Methodology
The following is a level -0 diagram of my ALU. It has a 32 bit input for operand1, and a second 32 bit input for operand 2. The final input is 3 bits and is used to select from the 7 functions. Finally, there is a 32 bit output which outputs the operand values after they have been operated on.



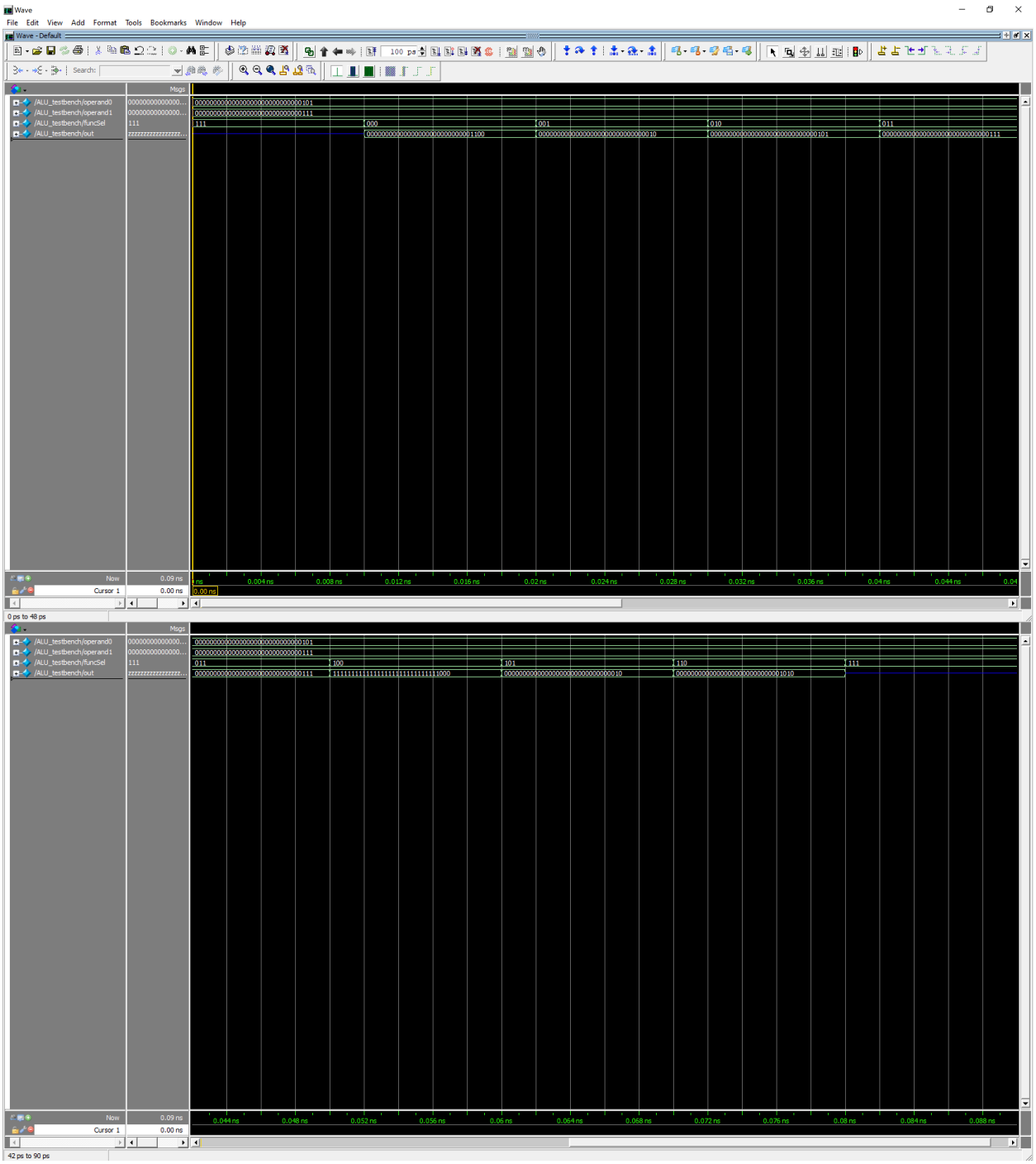
The functionality is implemented through a case statement. For each opcode input, out is assigned to its respective operation. They are as follows:
000: Arithmetic addition
001: Logic bitwise xor
010: Logic bitwise and
011: Logic bitwise or
100: logic bitwise nor
101: Register right logic shift
110: Register left logic shift
111: disable (hi-z output)

Result and analysis

When tested in ModelSim, each ALU function works as expected.

All code, diagrams, and simulation results can be found as files on GitHub: https://github.com/Eth7an/Lab-4.git

A video explanation of the top level design and ModelSim simulation can be found here: https://youtu.be/bVPm4lAHex4



Discussion and Conclusion

To conclude, I have built a fully working and tested 32 bit arithmetic logic unit with 7 basic functions. The functionality of all inputs and outputs has been tested and verified, and this module is complete and ready to be instantiated in higher level projects.