Temple University

College of Engineering

ECE 4612: Advanced Processor Systems

A red square with white letters

Description automatically generated

Adam Love

ALU Design

11/09/2023

**Objective**

This report is to detail the design and implementation of the ALU module that will be used within the final design of the single-cycle processor. The operations that the ALU must be able to do are addition, subtraction, multiplication, division, AND, and OR. In addition, a zero flag and an overflow flag are also included within the ALU.

**Tools/Equipment**

The ALU, all the ALU components, and any test benches were programmed using Verilog and were programmed within VS code. The ALU and each individual module was tested using Vivado to generate output waveforms.

**Procedure**

Within the ALU there are three discrete components. These components are an adder, a multiplier, and a divider. The adder is used for both addition and subtraction. In addition to these three components, a bitwise AND and a bitwise OR operation is also simulated within the ALU. Each of the three components (adder, multiplier, and divider) were created and tested individually before being tested within the ALU.

**Testing**

Each module was tested with a few values to verify that the zero and overflow flags work, as well as making sure that nothing broke in transferring the modules over into the ALU. The waveforms for each operation are shown below.

Adder test:

A screenshot of a computer

Description automatically generated

Subtractor test:

A screenshot of a video game

Description automatically generated

Multiplier test:

A screen shot of a computer

Description automatically generated

Divider test:

A screenshot of a computer

Description automatically generated

AND test:

A screenshot of a computer

Description automatically generated

OR test:

A screenshot of a computer

Description automatically generated

**Results/Observations**

When designing the multiplier and divider, I had to be careful when I was checking each bit and when each calculation was happening. Initially in the multiplier I had two checks on the same bit that were within if statements, and I forgot to use an else. After one of the if statements was complete, the other if statement could also be done, which would cause the incorrect value to be calculated. I ended up changing this to a case statement, which solved my issue. When designing the divider, I made sure to avoid this issue and was able to complete the design much easier.

**Conclusion**

Testing each module individually made putting everything together into the ALU much simpler. I designed the ALU module without knowing how my control and ALU control modules are going to be designed. Because of that, I made it so that the operation select value can be changed easily. There is also room to add two more operations within this ALU, if deemed necessary.