[Methods]

## Part 5

A device was needed to interface the provided ALU with a user, so the switches, buttons, and seven-segment display were used in one project to do so. The provided button debounce, ALU, and seven-segment display driver components were used, along with a new component, called UXCntl to create a working device. The UXCntl was created to latch the eight switches to either of the three input registers on the ALU, depending on the input from the debounced buttons. The ALU’s outputs were wired directly to the provided seven-segment driver’s inputs for the two right-most digits of the physical display. A test bench was created in order to test the new UXCntl component. Once the component proved to work in the simulator, a programming file was generated, and the device was flashed with the bit file. After some testing, the device proved to work successfully.

[Results]

## Part 5

The final product overall was relatively simple, however there was quite a learning process to get the final product. All that was needed was one more component and proper wiring in the top level file, however doing this task in a programming language is completely different than in a hardware description language. Changing from the programming language mindset to the hardware description language mindset proved to be the hardest part. In a specific example, a variable can be set by any number of functions or equations because each one is performed one at a time. In a hardware description language however, a signal cannot be driven by multiple sources, therefore only one process can change the value of a wire or a multiplexer must be used to pick which source will set the value of the wire. Next time, the provided components would be placed under the new device top level instead of integrating all the component top levels into one. This would simplify the debugging and design process since the components are broken up nicer.