Program 3

# Problem Statement

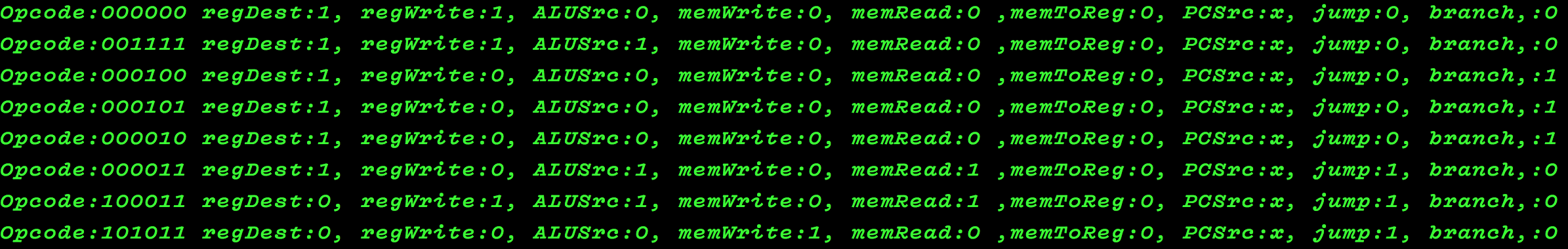
In this section of the overall project, we were tasked with design and implementation of a Verilog program that would emulate the control unit of a MIPS processor, and to connect said control unit to the pre-existing program counter.

# Approach to solution

This segment of the project was taken in a different approach than I did the last two, purely out of the inherent difference of the subject. For this section the Verilog code was very simple, as it was simply a lookup table, so it just needed to be implemented as a switch statement. The part that made it different was the research involved in finding the correct output for each case. The step of combining it with the previous assignment was trivial as the testbench I had used in the past would still contain the set-up program counter, so the two simply had to be attached.

# Solution Description

Much like the initial approach, the final solution was set up with a little amount of work, and the majority of the time was spent looking up each case and what the outputs should be based on the opcode provided. The testing of this module was implemented by simply changing the text in the data file to have lines corresponding to each of the required types.



Above is the output from testing the control module. The top line is regards to Opcode ‘000000’ and then are the following outputs are the respective outputs for the control unit. That particular opcode refers to each of the R-types of instructions as they are each handled the same way. From there each line responds to a different case we were required to implement with each line’s outputs.