Program 4

# Problem Statement

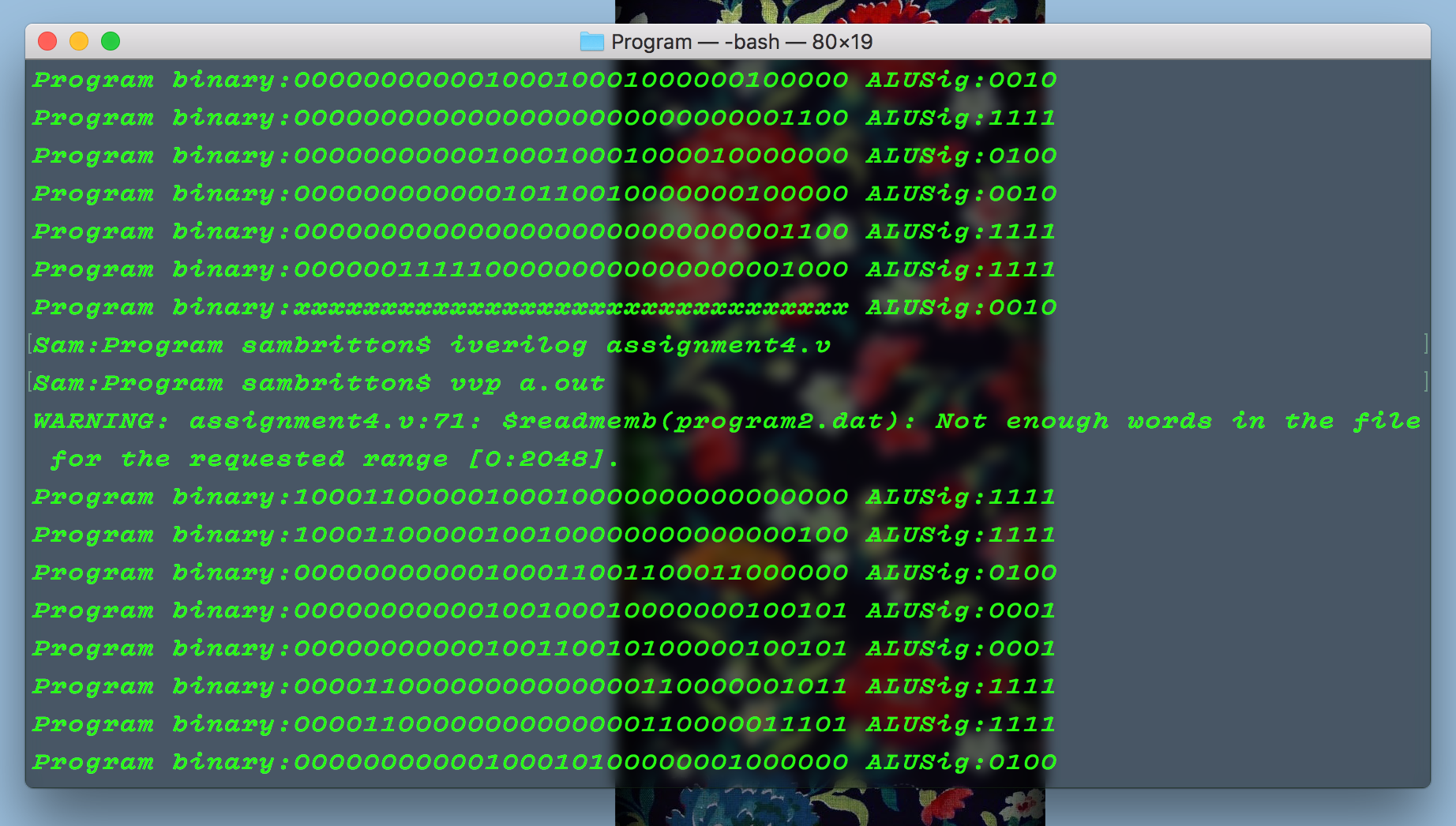
In this section of the overall project, we were tasked with design and implementation of a Verilog program that would emulate the ALU control unit of a MIPS processor, and to connect said control unit to the pre-existing program counter.

# Approach to solution

Approach to this segment ended up being similar to the previous segment. We were to take each of the required instructions and make sure they would all be supported in this ALU controller. For this segment however, there was less lookup and research for each instruction as the operations that the ALU does in regard to each instruction was usually very obvious per each instruction. More attention was also payed to the previous assignment of the MIPSALU, as the required instructions needed more functionality than what the previous iteration of the module supported initially.

# Solution Description

The design of program 4 was similar to that of program 3 as each of the programs ended up being simple lookup tables for each of the functions. However, also included in this final version of program 4, more functionality was added to the MIPS ALU, as the original design only supported a handful of operations, and more had to be added in order for the full implementation of each of the instructions.



Above is a sample from the compiling and printing of an example MIPS program. Per the IM module from an earlier project, each of the instructions are read and their respective ALUSig (ALU Signals) are displayed. Each of the ALUSig corresponds with a value that could be found in the MIPSALU section of the program. It is notable that a good number of the ALUSig is a value of “1111”, meaning this is an operation that does not utilize the ALU.