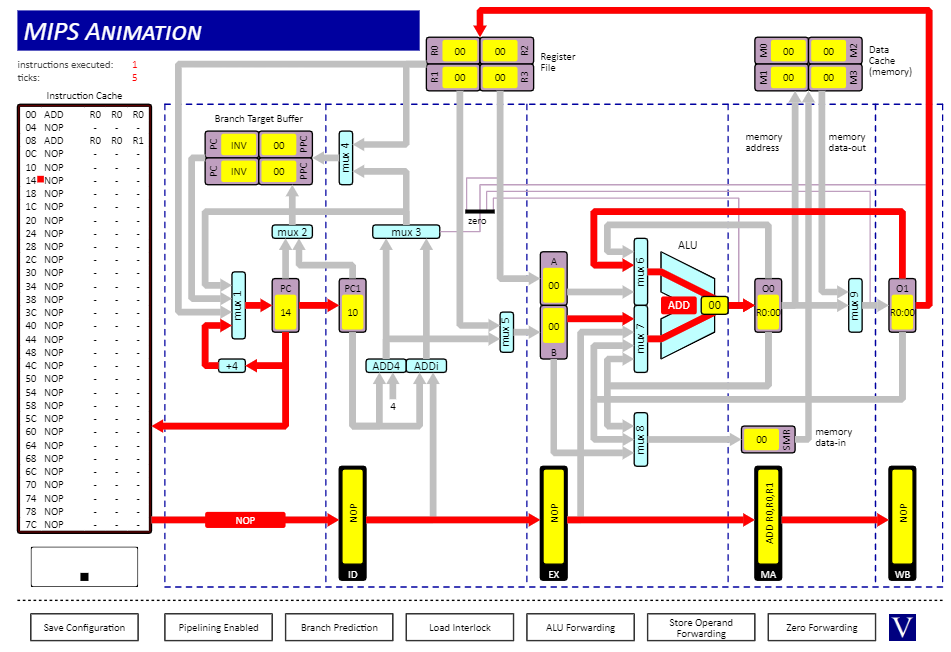
Tutorial 4

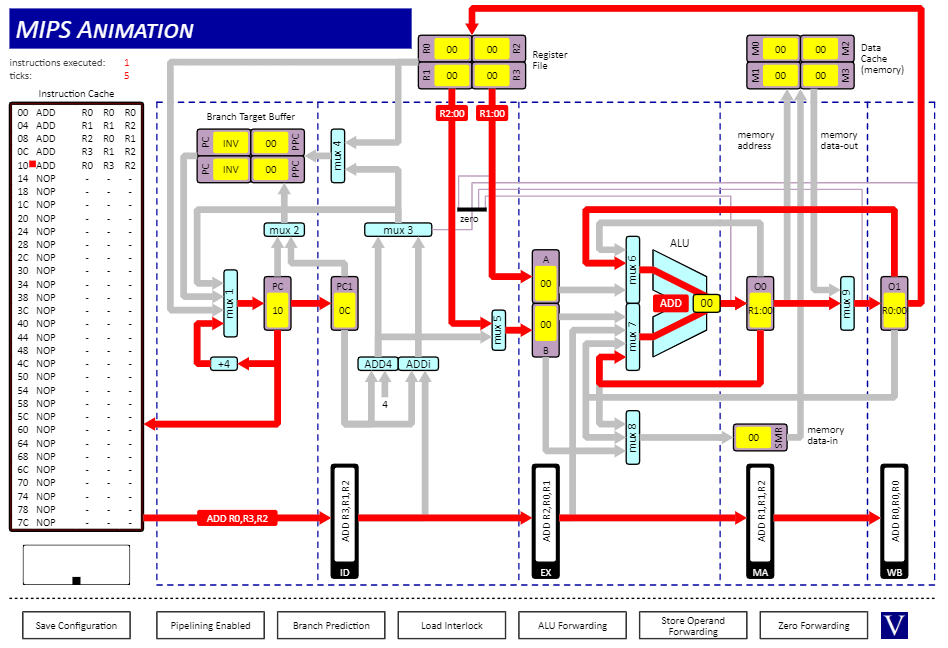
Donal Hogan - 15324573

# Q1

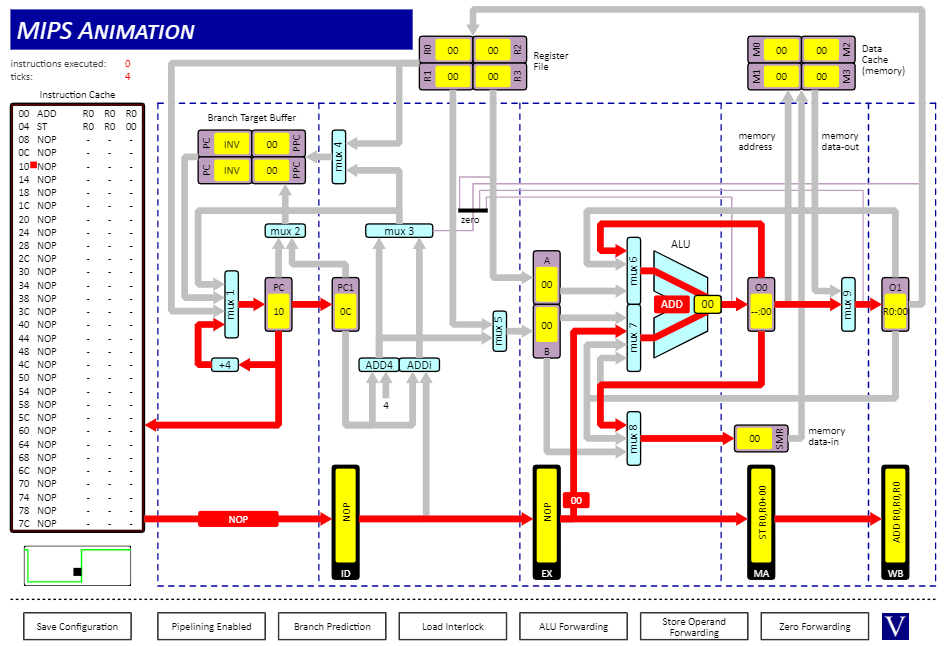
## Part 1



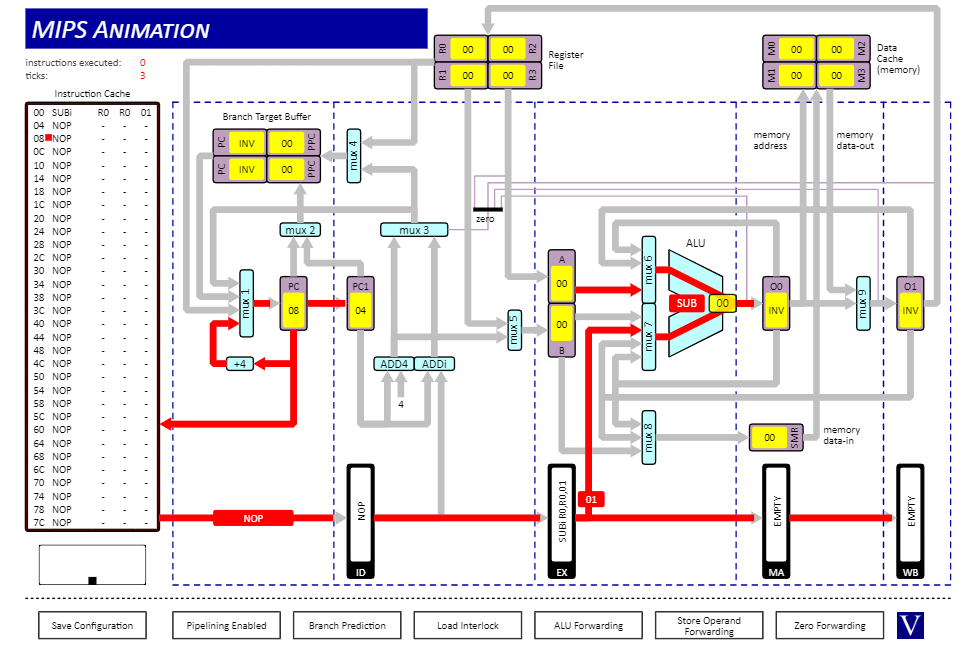
## Part 2



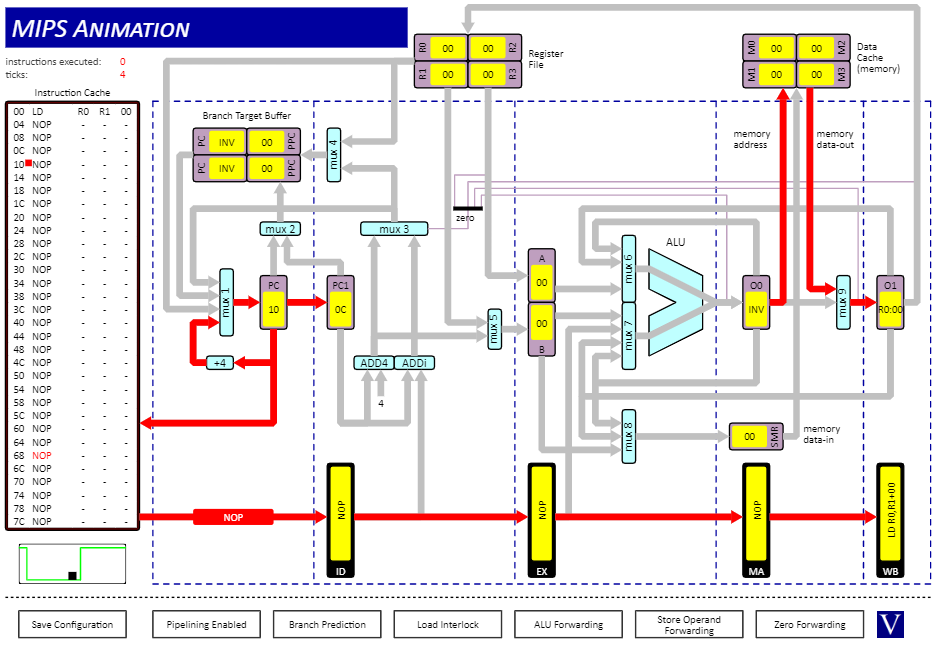
## Part 3



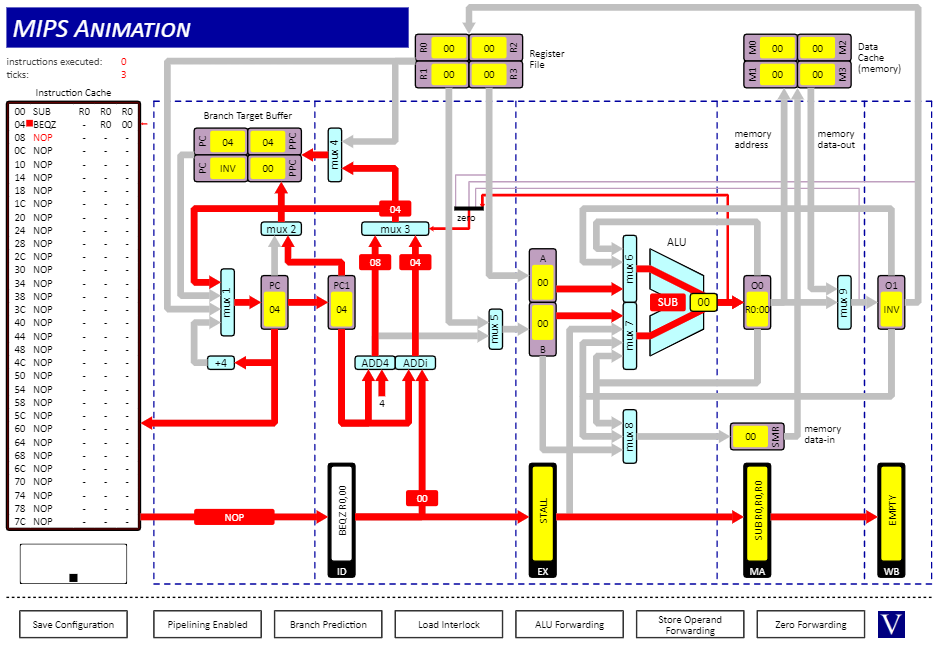
## Part 4



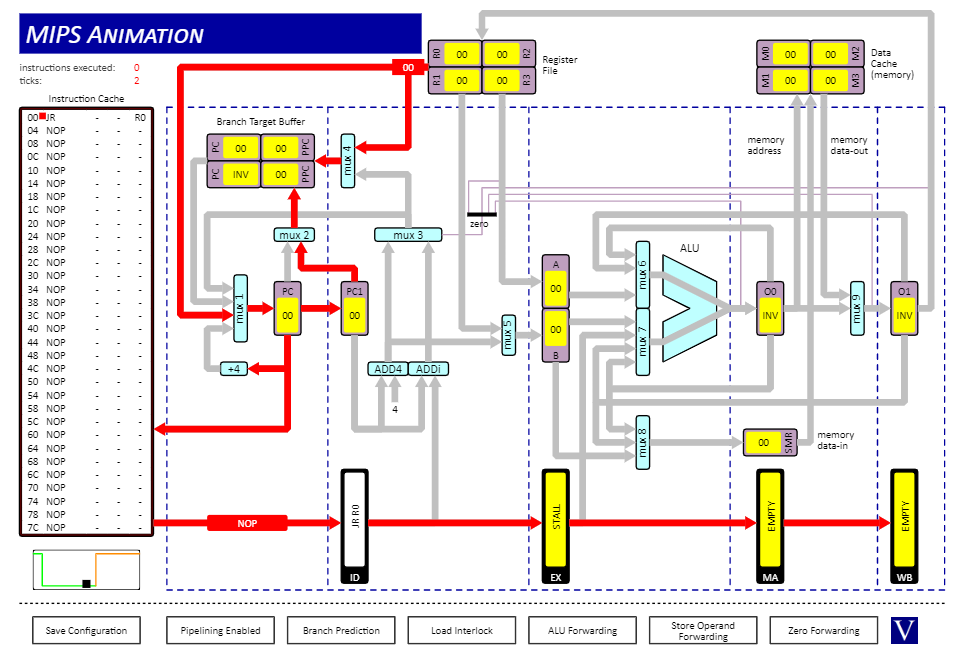
## Part 5



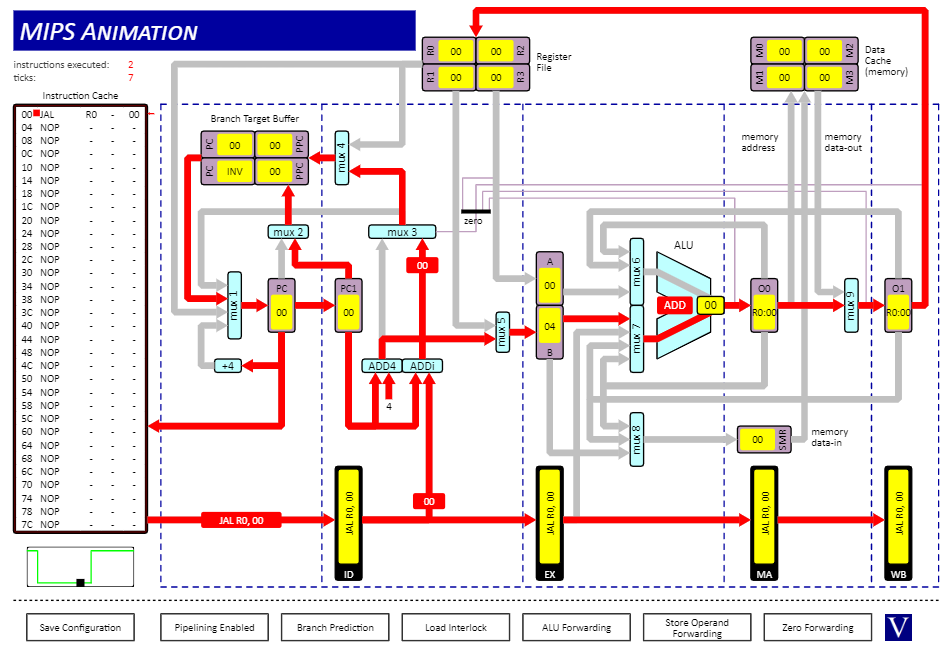
## Part 6



## Part 7

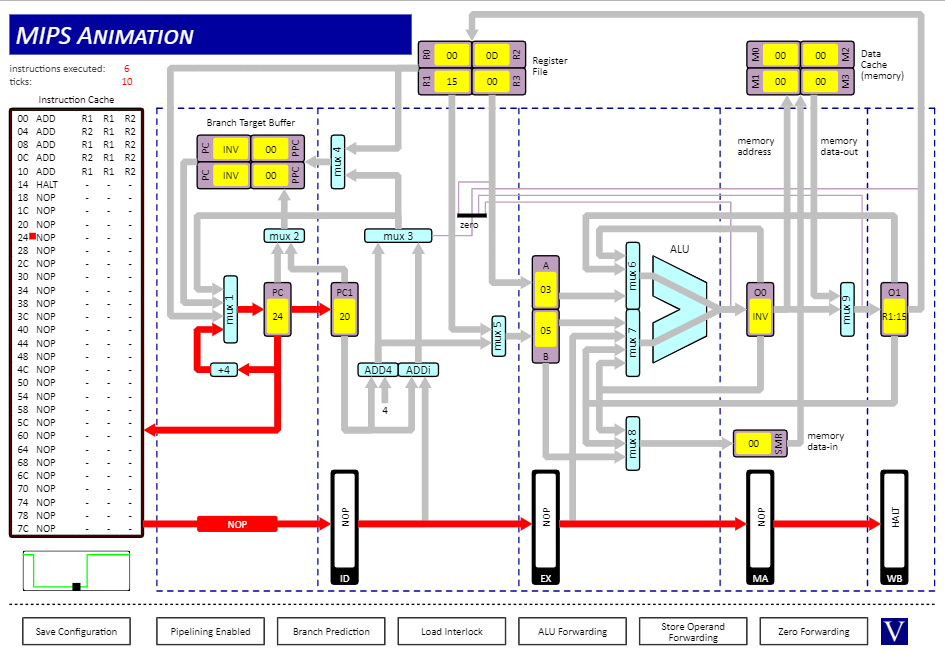


## Part 8



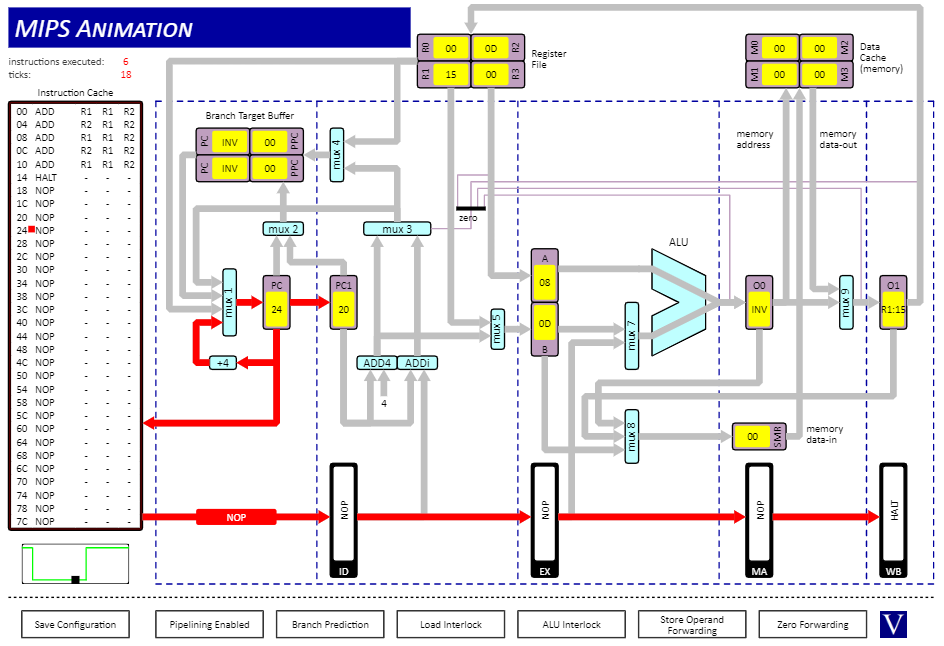
# Q2

## Part 1



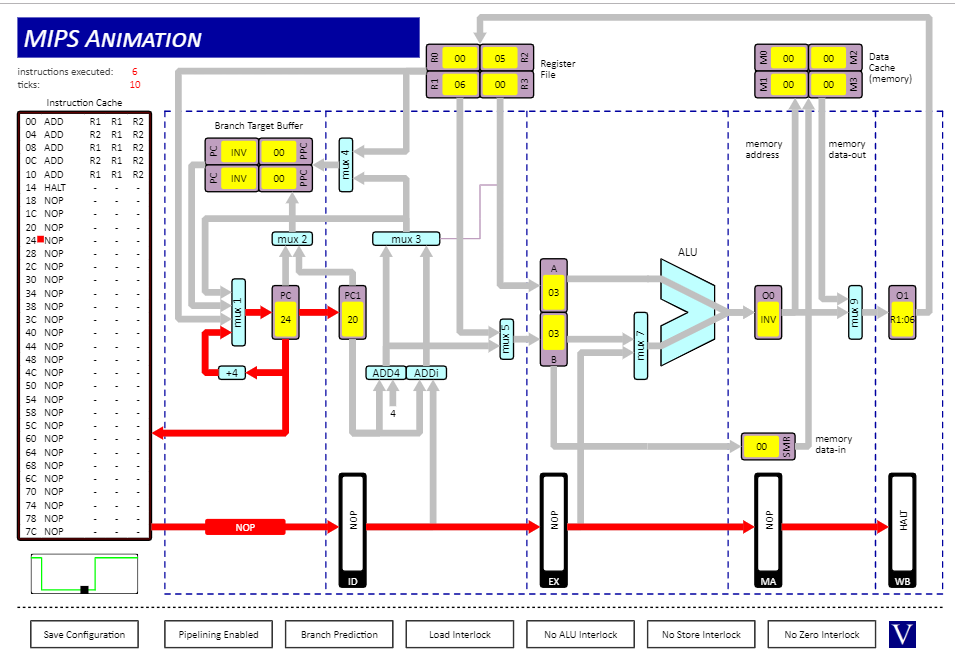
This is the fastest of the 3 while still being correct. ALU forwarding allows for use of the results in O0 and O1 to be used as inputs to the ALU directly rather than waiting for them to be stored in the registers and then taking the values from there.

## Part 2



Slowest of the 3 as it couldn’t use the results in O0 and O1 but instead had to wait for them to be stored in the correct registers. Then it took the values from the register file and used them in the ALU. It knew to wait and use those values rather than take what was in the registers already since it could recognize that it would give the wrong answer otherwise.

## Part 3



Fast but also incorrect. It couldn’t use the results in O0 or O1 but also couldn’t tell that it had to wait for them to be stored in the registers to be used. This meant that it used outdated values from the register file for some of the instructions and so the result was incorrect.

# Q3

## Part 1

38 instructions executed

50 clock cycles

These are not equal due to the stall cycles where no instruction was executed so the clock cycles continued to go up while the no. of executed instructions remained the same. The stall cycles occurred when the branches did as branch prediction was on, so it knew that was coming and didn’t try to execute the instructions that came after the branch instruction. It did try to predict that the branch would do the same thing each time it came to the conditional branch instruction though so after the first time it would try and execute the instructions at the branch location which would be correct until the final loop through when the condition would be false and there would be no branch. This reduced the number of stalls somewhat but there were still stalls.

## Part 2

38 instructions executed

52 clock cycles

This was similar to the last time just without the prediction. At the branch it would begin to execute the next instructions but then detect the stall and stop before branching. Since there’s no prediction there’s a few more clock cycles than before though not many.

## Part 3

69 instructions executed

84 clock cycles

This made the program take a lot longer and execute many more instructions. This is because it made the program branch more often and so made more instructions execute.