Chase Johnson

Ryan Price

Adam Sheagren

Part 2 Description

Part 2 of the project involved updating the SISC processor, with adding branches to the instruction set. The four types of branches we had to build were a Branch Not Equal (BNE), Branch Not Equal Relative (BNR), Branch Arithmetic Absolute (BRA), and Branch Arithmetic Relative (BRR). We had to be able to branch based on if a condition was true which we would checked using the STAT register. If we were in a positive branch instruction (BRA or BRR), we would take the branch only if the zero bit of the STAT register is set to 1. On the other hand, for negative branch instructions (BNE or BNR), we take the branch only if the zero bit of the STAT register is set to 0. Taking a branch involved changing the program counter to a new value relatively (BRR or BNR) or absolutely (BNE or BNR) depending on the branch type. If a branch is not taken, the program counter does not change until the beginning of the next instruction.

The logic to change the program counter for branching was given in a file called Br.v that was used to calculate the location of the branch and would then send it to the program counter. Br.v took as input br\_sel from our ctrl.v file which determined if the program counter would be modified in a relative or absolute calculation. The program counter (PC) was held in PC.v, another file we were given. PC.v controls if a branch is taken or the PC was only incremented based on the input pc\_sel whose value is set in our ctrl.v file. When we started this section of the project we added all the new files we were given to the SISC.v file as new module instantiations and then connected the inputs and outputs via wires. Among the new files was the imem.data file which provided instructions to test our implementation. These instructions were run, and the outputs matched those expected as indicated in the comments from the imem.data file.