

## LAB 8 – ARM PART 3

### Report Questions

1. If you are building a processor and have to do static branch prediction (meaning you have to assume at compile time whether a branch is taken or not), how should you do it?

After calling decode within execute, I can call decode again on PC + 1. This ensures that the next instruction is decoded. If the prediction is wrong and thus not taken, it won't affect the result of the program because the instruction hasn't been executed yet, thus, it will void the previously decoded instruction, for the new one and execute the correct one instead.

2. If you are building a 256-byte direct-mapped cache, what should you choose as your block (line) size?

32 bytes, because it has the highest hit rate from the dynamic run time statistics for all the cache.

3. What conclusions can you draw about the differences between compiling with no optimization and -O2 optimization?

-O2 optimizations runs way quicker than no optimization.

### Notes about assignment

Professor Pantoja said that it was ok that my cache doesn't match exactly since it was close enough to be correct. I was only off by less than 0.2%