Harrison DeWitt and Theo Watkins
Final Project Report

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? You can make a different decision for branches that go forward or backward.

   For a backward branch, it is better to assume that the branch is taken, because a backward branch is indicative of a loop and is likely to be taken multiple times, and only not taken once. For a forward branch, it is better to assume the branch is not taken, because it will more likely only be evaluated once, and because the pc is already incrementing to the next line it is more efficient to assume the branch is not taken.

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

   According to the fib data, a 32 byte block size is best. It allows for the most blocks while still having the least amount of data misses. This means fewer clock cycles are spent searching for the data, without sacrificing additional data misses. While other programs may see even fewer misses by using a 64 or 128 byte block size, the difference would likely be negligible, and come at the cost of taking more clock cycles spent searching for data. The 32 byte block size appears to have the best balance.

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

   Compiling with the -O2 flag significantly reduces the execution time of the program. The compiler performs many optimizations on the code, at the cost of a longer compilation time and slightly increased memory usage.

Notes:

When we ran our program with fib.sim, our R7 and the stack pointer registers ended with data still in them, as opposed to the example output, which had cleared both registers to 0. Additionally, our cache data was slightly different. While we ended up with the same number of memory access calls, our hits and misses were off by one or two for multiple block sizes.