

Ryan Blair, Austin McInnis

Lupo

CPE 315-01

5 June 2016

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

I would most likely almost always predict taken because the majority of instructions on all the programs run by the compiler were taken. This would cause a restart in the data pipeline for some of the instructions, but in terms of static prediction, this may be the simplest way. Another criteria I would add to the static branch predictor is whether it is forward or backward. If it goes backward, I would predict taken because most likely the programmer wanted the loop to continue multiple times. That would also mean I would predict not taken for forward loops.

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

I should choose a block line size of either 16 or 32 bytes, as they were consistently high percentages in hit-miss ratio then compared to the other block sizes.

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

Including optimization dramatically decreases the amount of instructions accessing data from both registers and memory. This would drop the run time because there are less instructions looking for something in memory. In shang, O2 optimization decreases the total number of instructions by 1,065,045. One interesting point to make is that the cache hit rate decreased with increased optimization.