Benchohra, Sereen
Kaanan Kharwa
CPE 315
8/26/2020

Thumbsim Project

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.

   In order to do branch prediction, we should implement a 2-bit branch predictor which has four states (states are Strongly Taken, Taken, Not Taken, Strongly Not Taken). According to our data, the prediction we should take for our backward branches are taken since statistically, the majority of the branches the three programs have used taken in the backward branch. The starting state for our Forward Branch should be not taken because according to our stats, it was more likely that the forward branches in the three programs were not taken. As a result, there is less likely going to be a misprediction and therefore cost less cycles when using these starting points.

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

Block Size (bytes)	Fib (%)	ChkSum(%)	Matmul(%)	Average Hit Rate (%)
4	94.544	95.8172	90.5940	93.6517
8	97.2727	96.9815	90.7341	94.9961
16	98.1818	96.4209	89.6078	94.7368
32	99.0909	94.0923	88.2433	93.8088
64	99.0909	94.2648	83.5851	92.3136
128	99.0909	88.8314	83.6688	90.5304
256	99.0909	77.8784	75.6014	84.1902

   From our calculations, the block size that would give the highest average from the three programs hit rate would be the block size of 8 bytes.

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

   N/A