

Part1: Write Up

In this part of the assignment, I generated C++ code using Python to do loop unrolling. Loop unrolling is used to exploit instruction level parallelism (ILP). In part1 each iteration contains dependent instructions to impede the processor's ability to use ILP. I thought of this as two different flavors of ice cream. One was with each iteration performed sequentially and one with interleaved instructions.

The sequential loop requires the dependency chain to not be re-ordered because the floating-point operations must be done in order. This results in the compiler producing an ISA instruction for each of the addition operations. The interleaved loop requires a breaking of dependencies that expose ILP. These interleaved instructions can be pipelined or executed on a superscalar processor. This makes the interleaved loop technique faster than the sequential loop.

After running both loops, the results were very clear. With an unroll factor of 16, the interleaved code had a speed up of about 6.7x where the sequential code only had a speed up of 1.3x when compared to the reference loop. This is consistent with the lectures and slides taught in class.