# CSC-591 (005)

# homework 0

02/15/2018

MODE SPEEDS

|  |  |  |
| --- | --- | --- |
| mode | Time (ms) | Speed up (m0 base line) |
| m0 | 219 | 0 |
| m1 | 182 | 37 |
| m2 | 163 | 56 |

Mode 1 faster than mode 0?

Mode 1 is faster than mode 0, this is primarily due to a large fact that mode 0 is loading and storing values of the array one at a time. Due to the single action nature, it will need to iterate through the for-loop just as many times as there are elements in the array. Using Neon, this can be alleviated by loading and operating on four different elements at the same time by loading and performing addition on them all at once, this way you will be iterating through the array four times less.

Mode 2 faster than mode 0?

Mode 2 is much faster than mode 0. Using OpenCL and its hardware acceleration, mode 2 is easily faster than mode 0 at computing the sums of the arrays by loading each into a queue to be read later by the FGPA.

Mode 2 faster than mode 1?

Mode 2 is still faster than the mode 1 I wrote. The issue with my implementation was that I was running a variable for-loop that cannot be unrolled by the compiler. Along with the fact that I am basically doing the same operations as mode 0 but just four at a time. My algorithm could not compete with the efficiency of mode 2’s hardware accelerated actions, but I could possibly decrease the separation between cache calls by loading all values of input\_a into output first and then in a separate loop adding in input\_b.

Students Sharing Board:

Daniel Karamitrov