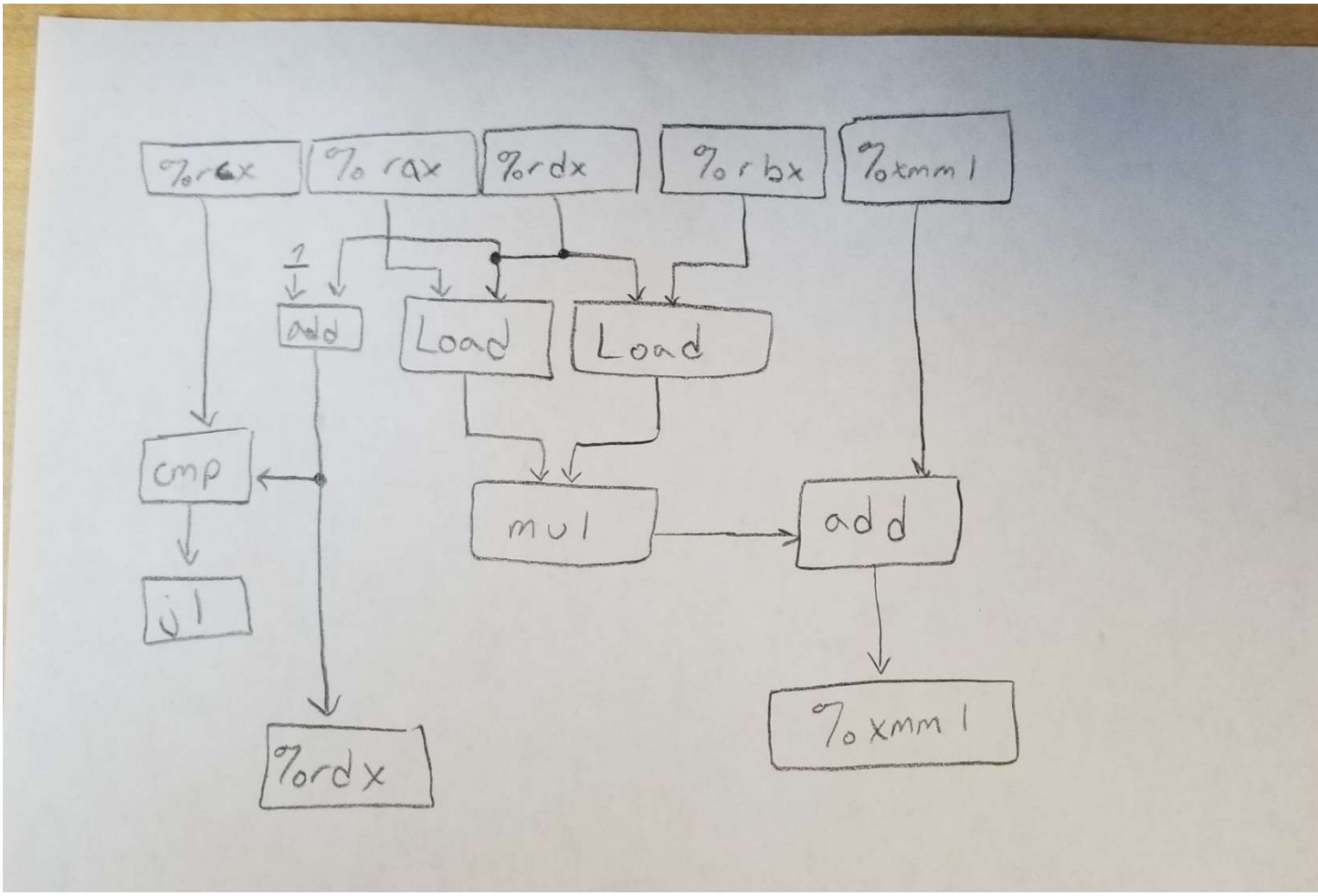
Part 6-2 [a]



Part 6-2 [b]

The operations that cannot be pipeline are the adds instruction and the cmpq instruction.

For adds: because floating point arithmetic is dependent on the order of operations, adds must be executed sequentially.

cmpq cannot be pipelined because it sets the processor flags for the next jump instruction. Thus, it must be executed right before the jump instruction, with no other instruction

being executed. Otherwise, unpredictable jumping can and will occur.
Because of this, a lower bound for each loop iteration is 13 CPE.

Part 6-2 [d]

Based on the data collected, the optimizations are only slightly faster. Though, this would probably more noticeable on data sets greater than what I tested (> 1 billion). Also, since each processor will handle pipelining and optimizations differently, I'm sure that the optimizations would be more prominent on a different system.

Data:

N	inner	inner2
10	0.00001 sec	0.00009 sec
100	0.000011 sec	0.000011 sec
1,000	0.000027 sec	0.000026 sec
10,000	0.000207 sec	0.000191 sec
100,000	0.002167 sec	0.001864 sec
1,000,000	0.016881 sec	0.015806 sec
10,000,000	0.143373 sec	0.141107 sec
100,000,000	1.045 sec	0.923461 sec
1,000,000,000	10.639 sec	9.383 sec