% xmm0 % xxm 1 % rax % rbx % rcx % rdx

mov
mul
Add
% xmm1

load
load
comp add
jl
rdz
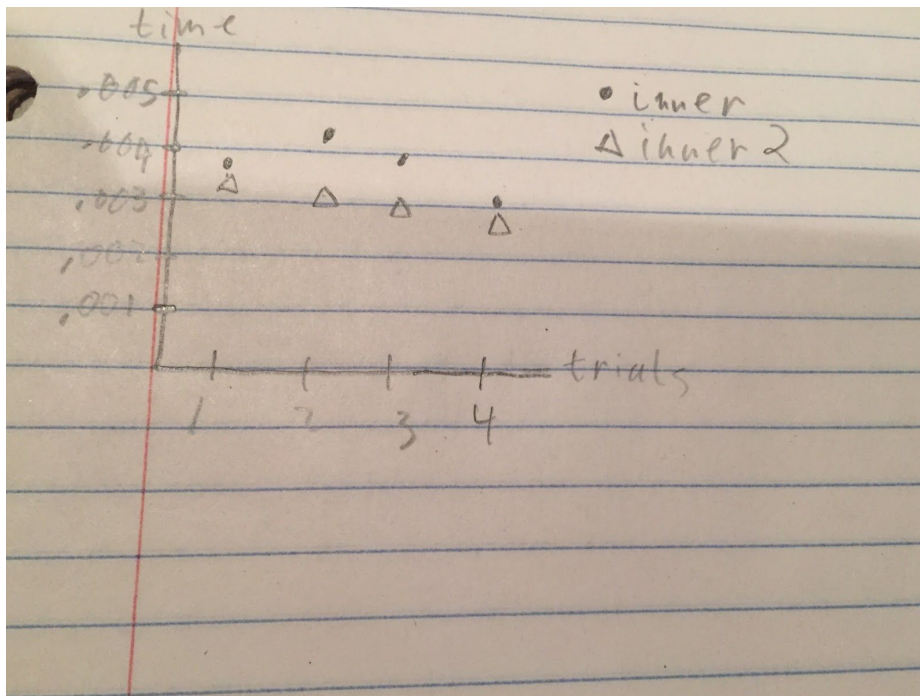
2b.   Operation mulss cannot be pipled. Due to this, adds cannot be piplined either since it has to retrieve the product that will be in %xmmm0. adds will have to wait for mulls to writeback in order to execute properly. the latency of movss is 4, iff the loading is 4. Addq has latency 1 while adds has latency 3. Cmpq has a latency 1. All these latencies put together gives a lower latency bound of 22.

2d. In testing my 7-2 code, I found that inner2 on average executes faster than inner. This occures due to loop unrolling, packing 4 elements in each loop iteration is much faster than doing one procedure ever loop.

time
.005
.004
.003
.002
.001

• inner
△ inner 2

trials
1 2 3 4

inner times: 0.003695, 0.004123, 0.003810, 0.003018

inner2 times: 0.003145, 0.003378, 0.003023, 0.002943