Xiaoru Chen
chenxia2@oregonstate.edu
Project #4
CS 475/575
2021-05-12

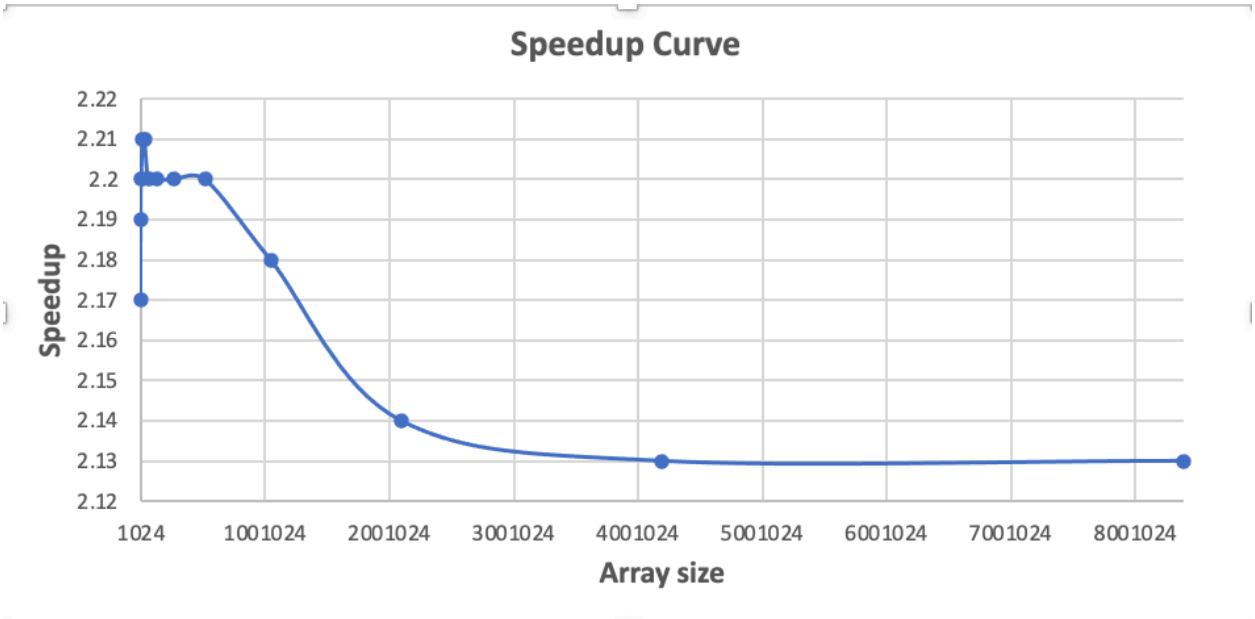
Machine I ran this on:
I used my MacBook pro to ssh to OSU ENGR server (flip).

Table of performances for each array size and the corresponding speedups:

	Non-SSE Peak Performance (MegaMults/Sec)	SSE Peak Performance (MegaMults/Sec)	Speedup
1024	289.96	628.29	2.17
2048	291.16	636.94	2.19
4096	291.73	641.68	2.2
8192	292.02	643.65	2.2
16384	292.17	645.02	2.21
32768	291.99	643.86	2.21
65536	291.57	642.84	2.2
131072	291.72	641.95	2.2
262144	290.55	640.36	2.2
524288	289.66	638.25	2.2
1048576	285.18	621.56	2.18
2097152	283.63	605.8	2.14
4194304	282.9	602.65	2.13
8388608	283.3	603.81	2.13

Table.1 Performances for each array size and the corresponding speedups

Graph of SIMD/non-SUMD speedup versus array size:



Graph.1 SIMD/non-SUMD speedup versus array size

What patterns am I seeing in the speedups?

When the length of the array gets larger and larger, the speed up shows a downward trend and finally becomes relatively stable.

Are they consistent across a variety of array size?

No, they are not.

Why or why not, do I think?

Because the registers are limited. Initially, when the size of the array is small, the registers are sufficient and the arithmetic wait is small. However, when the size of the array becomes very large, such as 4 million, 5 million, the registers becomes saturated, operations need to wait, but the operations still need to be performed, so at this point, the trend of decline is almost no longer reached.