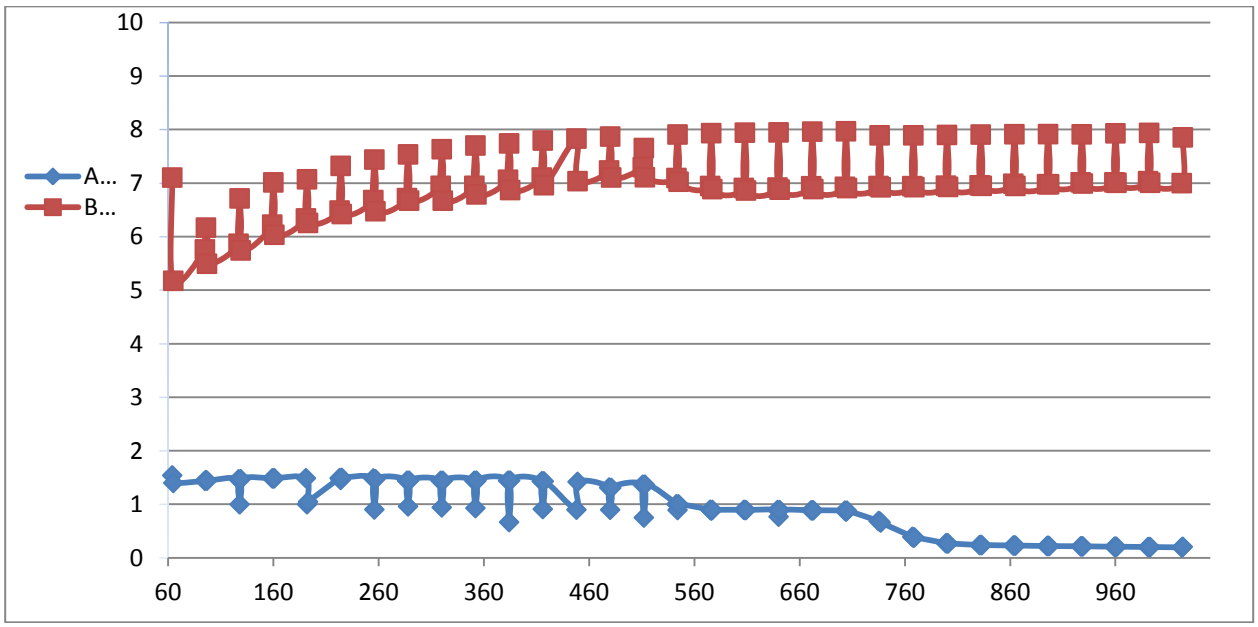
Name: Yijia Zang
Login: cs61c-fp

In the innermost loop of my code, I used SSE instructions to compute a single 4 * 4 matrix multiplication. It's kind of like a cache blocking, but not really. So I loaded 4 columns of A first by calling 4 _mm_loadu_ps intrinsic functions. Each one of them could read 4 single-precision floating points at once. Actually I am loading 4 columns of A transpose since I passed in A transpose as the parameter when I called this function Do the same to the other Matrix B. And then here comes the handy part, which is to calculate the values of the 4*4 block. So we need to multiply and add multiple times. I found the intrinsic function _mm_hadd_ps pretty useful, because it adds products horizontally. I used it nested with _mm_mul_ps and _mm_hadd_ps.

To handle fringes case, I simply fill the original matrices with 0s to make the size divisible by 4. In the function called edge_hadling I had one loop for copying whatever is in the original matrix, and since the copy is dynamically allocated, so it was zeroed out when it was created. Therefore, after copying the original matrices to a new matrix, the left space would automatically be zeros.



64 by 64 matrix	1.54187 Gflop/s
65 by 65 matrix	1.40081 Gflop/s
95 by 95 matrix	1.44066 Gflop/s
96 by 96 matrix	1.45405 Gflop/s
97 by 97 matrix	1.44253 Gflop/s
127 by 127 matrix	1.46789 Gflop/s
128 by 128 matrix	1.00062 Gflop/s
129 by 129 matrix	1.47011 Gflop/s

159 by 159 matrix	1.4861 Gflop/s
160 by 160 matrix	1.48436 Gflop/s
161 by 161 matrix	1.48787 Gflop/s
191 by 191 matrix	1.48916 Gflop/s
192 by 192 matrix	1.002 Gflop/s
193 by 193 matrix	1.48932 Gflop/s
223 by 223 matrix	1.49075 Gflop/s
224 by 224 matrix	1.45609 Gflop/s
225 by 225 matrix	1.49591 Gflop/s
255 by 255 matrix	1.48573 Gflop/s
256 by 256 matrix	0.90414 Gflop/s
257 by 257 matrix	1.4705 Gflop/s
287 by 287 matrix	1.44201 Gflop/s
288 by 288 matrix	0.96223 Gflop/s
289 by 289 matrix	1.44267 Gflop/s
319 by 319 matrix	1.4406 Gflop/s
320 by 320 matrix	0.94371 Gflop/s
321 by 321 matrix	1.44124 Gflop/s
351 by 351 matrix	1.44179 Gflop/s
352 by 352 matrix	0.930269 Gflop/s
353 by 353 matrix	1.44379 Gflop/s
383 by 383 matrix	1.44263 Gflop/s
384 by 384 matrix	0.668601 Gflop/s
385 by 385 matrix	1.4443 Gflop/s
415 by 415 matrix	1.43186 Gflop/s
416 by 416 matrix	0.912459 Gflop/s
417 by 417 matrix	1.43128 Gflop/s
447 by 447 matrix	1.42237 Gflop/s
448 by 448 matrix	0.900601 Gflop/s
449 by 449 matrix	1.42093 Gflop/s
479 by 479 matrix	1.31473 Gflop/s
480 by 480 matrix	0.899225 Gflop/s
481 by 481 matrix	1.30893 Gflop/s
511 by 511 matrix	1.36765 Gflop/s
512 by 512 matrix	0.753963 Gflop/s
513 by 513 matrix	1.36785 Gflop/s
543 by 543 matrix	1.00401 Gflop/s
544 by 544 matrix	0.893682 Gflop/s
545 by 545 matrix	1.00002 Gflop/s
575 by 575 matrix	0.900251 Gflop/s
576 by 576 matrix	0.883029 Gflop/s
577 by 577 matrix	0.897197 Gflop/s
607 by 607 matrix	0.897846 Gflop/s
608 by 608 matrix	0.894273 Gflop/s
609 by 609 matrix	0.894439 Gflop/s
639 by 639 matrix	0.895201 Gflop/s
640 by 640 matrix	0.769748 Gflop/s
641 by 641 matrix	0.893583 Gflop/s
671 by 671 matrix	0.888941 Gflop/s

672 by 672 matrix 0.889864 Gflop/s
673 by 673 matrix 0.888566 Gflop/s
703 by 703 matrix 0.883964 Gflop/s
704 by 704 matrix 0.863909 Gflop/s
705 by 705 matrix 0.883445 Gflop/s
735 by 735 matrix 0.689039 Gflop/s
736 by 736 matrix 0.682097 Gflop/s
737 by 737 matrix 0.660602 Gflop/s
767 by 767 matrix 0.394232 Gflop/s
768 by 768 matrix 0.377638 Gflop/s
769 by 769 matrix 0.38368 Gflop/s
799 by 799 matrix 0.273253 Gflop/s
800 by 800 matrix 0.275119 Gflop/s
801 by 801 matrix 0.269045 Gflop/s
831 by 831 matrix 0.239042 Gflop/s
832 by 832 matrix 0.2431 Gflop/s
833 by 833 matrix 0.23836 Gflop/s
863 by 863 matrix 0.230067 Gflop/s
864 by 864 matrix 0.23284 Gflop/s
865 by 865 matrix 0.229483 Gflop/s
895 by 895 matrix 0.220628 Gflop/s
896 by 896 matrix 0.229218 Gflop/s
897 by 897 matrix 0.220248 Gflop/s
927 by 927 matrix 0.215388 Gflop/s
928 by 928 matrix 0.217274 Gflop/s
929 by 929 matrix 0.215137 Gflop/s
959 by 959 matrix 0.207474 Gflop/s
960 by 960 matrix 0.211208 Gflop/s
961 by 961 matrix 0.207184 Gflop/s
991 by 991 matrix 0.202169 Gflop/s
992 by 992 matrix 0.203422 Gflop/s
993 by 993 matrix 0.201792 Gflop/s
1023 by 1023 matrix 0.195959 Gflop/s
1024 by 1024 matrix 0.207904 Gflop/s

Part1 Result:
64 by 64 matrix 7.10561 Gflop/s
65 by 65 matrix 5.17759 Gflop/s
95 by 95 matrix 5.7639 Gflop/s
96 by 96 matrix 6.71707 Gflop/s
97 by 97 matrix 5.49588 Gflop/s
127 by 127 matrix 5.86946 Gflop/s
128 by 128 matrix 6.71627 Gflop/s
129 by 129 matrix 5.74401 Gflop/s
159 by 159 matrix 6.22625 Gflop/s
160 by 160 matrix 7.01508 Gflop/s
161 by 161 matrix 6.03699 Gflop/s
191 by 191 matrix 6.34238 Gflop/s

192 by 192 matrix	7.07316 Gflop/s
193 by 193 matrix	6.25776 Gflop/s
223 by 223 matrix	6.4895 Gflop/s
224 by 224 matrix	7.32403 Gflop/s
225 by 225 matrix	6.42439 Gflop/s
255 by 255 matrix	6.68524 Gflop/s
256 by 256 matrix	7.44186 Gflop/s
257 by 257 matrix	6.47728 Gflop/s
287 by 287 matrix	6.72212 Gflop/s
288 by 288 matrix	7.53863 Gflop/s
289 by 289 matrix	6.60938 Gflop/s
319 by 319 matrix	6.88224 Gflop/s
320 by 320 matrix	7.63537 Gflop/s
321 by 321 matrix	6.6741 Gflop/s
351 by 351 matrix	6.9512 Gflop/s
352 by 352 matrix	7.70431 Gflop/s
353 by 353 matrix	6.79099 Gflop/s
383 by 383 matrix	7.0648 Gflop/s
384 by 384 matrix	7.74389 Gflop/s
385 by 385 matrix	6.87204 Gflop/s
415 by 415 matrix	7.10663 Gflop/s
416 by 416 matrix	7.79726 Gflop/s
417 by 417 matrix	6.963 Gflop/s
447 by 447 matrix	7.20981 Gflop/s
448 by 448 matrix	7.83423 Gflop/s
449 by 449 matrix	7.03883 Gflop/s
479 by 479 matrix	7.23834 Gflop/s
480 by 480 matrix	7.87181 Gflop/s
481 by 481 matrix	7.10712 Gflop/s
511 by 511 matrix	7.29898 Gflop/s
512 by 512 matrix	7.65694 Gflop/s
513 by 513 matrix	7.11099 Gflop/s
543 by 543 matrix	7.09719 Gflop/s
544 by 544 matrix	7.91004 Gflop/s
545 by 545 matrix	7.02826 Gflop/s
575 by 575 matrix	6.94806 Gflop/s
576 by 576 matrix	7.93523 Gflop/s
577 by 577 matrix	6.89027 Gflop/s
607 by 607 matrix	6.9119 Gflop/s
608 by 608 matrix	7.94458 Gflop/s
609 by 609 matrix	6.86355 Gflop/s
639 by 639 matrix	6.91937 Gflop/s
640 by 640 matrix	7.94982 Gflop/s
641 by 641 matrix	6.8797 Gflop/s
671 by 671 matrix	6.94383 Gflop/s
672 by 672 matrix	7.96429 Gflop/s
673 by 673 matrix	6.89118 Gflop/s
703 by 703 matrix	6.93307 Gflop/s
704 by 704 matrix	7.97061 Gflop/s

705 by 705 matrix 6.90879 Gflop/s
735 by 735 matrix 6.94802 Gflop/s
736 by 736 matrix 7.89496 Gflop/s
737 by 737 matrix 6.92327 Gflop/s
767 by 767 matrix 6.94523 Gflop/s
768 by 768 matrix 7.89482 Gflop/s
769 by 769 matrix 6.92596 Gflop/s
799 by 799 matrix 6.95435 Gflop/s
800 by 800 matrix 7.90157 Gflop/s
801 by 801 matrix 6.93083 Gflop/s
831 by 831 matrix 6.96129 Gflop/s
832 by 832 matrix 7.90939 Gflop/s
833 by 833 matrix 6.95152 Gflop/s
863 by 863 matrix 6.99002 Gflop/s
864 by 864 matrix 7.91539 Gflop/s
865 by 865 matrix 6.94915 Gflop/s
895 by 895 matrix 6.98152 Gflop/s
896 by 896 matrix 7.91712 Gflop/s
897 by 897 matrix 6.97996 Gflop/s
927 by 927 matrix 7.01636 Gflop/s
928 by 928 matrix 7.91345 Gflop/s
929 by 929 matrix 6.99407 Gflop/s
959 by 959 matrix 7.01628 Gflop/s
960 by 960 matrix 7.9324 Gflop/s
961 by 961 matrix 7.00784 Gflop/s
991 by 991 matrix 7.0386 Gflop/s
992 by 992 matrix 7.94001 Gflop/s
993 by 993 matrix 7.00832 Gflop/s
1023 by 1023 matrix 7.00145 Gflop/s
1024 by 1024 matrix 7.85623 Gflop/s