# EE/CSCI 451
# Fall 2020
# Homework 1 solution
**Total Points: 100**

**Different results under different assumptions are also accepted.**

# 1 [20 points]

Please refer to slides and textbook.

# 2 [20 points]

Solution - option 1 (assumption: DDR access is non-streaming):

- (1) 400 cycles to read one pair of two vector elemets from memory, 1 cycle to conduct multiply-add (2 FLOPs), hence the peak performance is 2 FLOPs/401 cycles $\approx$ 5 MFLOPS. This is a memory bound process.
- (2) 400 cycles to read sixteen pairs of elements from memory, 2 cycles to conduct 16 multiply-adds (32 FLOPs), hence the peak performance is 32 FLOPs/402 cycles $\approx$ 80 MFLOPS. This is a memory bound process.

Solution - option 2 (assumption: DDR access is streaming):

- (1) 400 cycles (initial latency) to read one pair of two vector elements from memory, 1 cycle to conduct multiply-add (2 FLOPs), each data fetch after the initial access takes 1 cycle and computation can be hidden with memory access. Hence, the peak performance is 2*dim FLOPs/(401+2*dim) cycles $\approx$ 1 GFLOPS when dim is very large. This is a memory bound process.
- (2) 400 cycles to (initial) read sixteen pairs of elements from memory, 2 cycles to conduct 16 multiply-adds (32 FLOPs), each 16-words data fetch after the initial access takes 1 cycle and each 16 multiply-adds computation take 2 cycles. hence the peak performance is 2*dim FLOPs/(400+2*dim*16/2) cycles $\approx$ 8 GFLOPS. This is a compute bound process (computation cannot be hidden by communication).

Both options are ok as long as the calculation matches the assumption.

# 3 [20 points]

The vector is cached, hence 1 cycle is used to fetch 4 words of the vector from cache. 100 cycles are used to fetch 4 words of the matrix from memory. 2 cycles are used for the processor to conduct 8 FLOPs on the 4 pairs of words (4 FLOPs in each cycle). Hence the peak performance is 8 FLOPs/(1+100+2)cycles = 8 FLOPs/103 ns $\approx$ 77.7 MFLOPS.

# 4 [10 points]

The matrices are too big to be cached. Thus, processor accesses data from external memory. 200 cycles are used to fetch 8 words of the same row of A. 200 $\times$ 8 cycles are used to get the 8 words of the same column of B, since they are in 8 different rows and B is stored in row-major order. 2 cycles are used for the processor to conduct 16 FLOPs on the 8 pairs of words. Hence the peak performance is 16 FLOPs/ 902 cycles = 16 FLOPs/ 1802 ns $\approx$ 8.88 MFLOPS.

# 5 [15 points]

Three assumptions: 1.If the cache access and computation can be perfectly overlapped (the best case), the total execution time is $100 \times w + n$. 2.If the cache access and computation can not be overlapped at all (the worst case), the total execution time is $(100 + k) \times w + n$. 3.If not only the cache access but also DRAM access can be hidden (streaming DDR access), the total execution time is $100 + k \times w + n$.

# 6 [15 points]

Based on the given pseudo code, in each iteration (the loop in Line 2), the read accesses are performed as follows.

- Read a[0], a[1] $\rightarrow$ miss, miss (bring a[0], a[1] into cache)
- Read a[1], a[2] $\rightarrow$ hit, miss (bring a[2], a[3] into cache)
- Read a[2], a[3] $\rightarrow$ hit, hit (Note that a[3] has been brought into cache together with a[2] because the cache line size is 2 words)
- Read a[3], a[4] $\rightarrow$ hit, miss (bring a[4], a[5] into cache)
- Read a[4], a[5] $\rightarrow$ hit, hit
- Read a[5], a[6] $\rightarrow$ hit, miss (bring a[6], a[7] into cache)
- Read a[6], a[7] $\rightarrow$ hit, hit
- ...
- Read a[61], a[62] $\rightarrow$ hit, miss (bring a[62], a[63] into cache)
- Read a[62], a[63] $\rightarrow$ hit, hit

Thus, the cache read hit ratio = $\frac{31+62}{63\times2}$ = 73.8%