Problem 1.4

Use nvcc -Xptxas -v matrixmul_kernel.cu to see the resource usage of your kernel (although compilation will fail, it will only do so after compiling the kernel and displaying the relevant information. “smem” stands for shared memory and “cmem” stands for constant memory.) Then, answer the following questions for your implementation:

ptxas info : Used 25 registers, 2048 bytes smem, 368 bytes cmem[0]

(a) How much shared memory is used per block? What is the maximum number of blocks per SM, if the shared memory was the bottleneck?

2048 bytes,

sharedMemPerBlock 49152 bytes.

49152 / 2048 = 24.

(b) How many registers are used per block? What is the maximum number of blocks per SM, if the number of registers was the bottleneck?

regsPerBlock 65536

65536 / 25 * 16 *16 = 10.

(c) What is the maximum number of blocks per SM, if the number of thread contexts was the bottleneck?

maxThreadsPerBlock 1024

1024 / 256 = 4.

(d) For your kernel implementation and GPU, how many threads can be simultaneously executing?

multiProcessorCount : 16

Each block is executed as 32-threads Wraps.

BlockNumber = P.width * P.height / 256

Simultaneously threads = P.width * P.height / 32