This survey collects your answers for the PIKA on Kirk & Hwu chapter 4 in CPSC 418 2017-2. In order to receive PIKA credit, you must enter your name and student number below.

Enter your last name (as it appears in Connect).

Enter your student number.

Select all types of memory through which two threads can share information. You need not consider synchronization issues.

	Global memory	Registers	Shared memory	Constant memory
Both threads in the same warp.	☐	☐	☐	☐
Both threads in the same block, but different warps.	☐	☐	☐	☐
Both threads in the same grid, but different blocks.	☐	☐	☐	☐

Rank the types of GPU memory from fastest (rank 1) to slowest (rank 4).

Global memory

Registers

Constant memory

Shared memory

Assume that you launch a kernel with 4096 blocks each with 256 threads (in 8 warps of 32 threads). If you declare a **local** variable in the kernel, how many versions of that variable are created?

- 1
- 256 / 32 = 8
- 32
- 256
- 4096
- 32 * 4096 = 131,072
- 256 * 4096 = 1,048,576

Assume that you launch a kernel with 4096 blocks each with 256 threads (in 8 warps of 32 threads). If you declare a **shared memory** variable in the kernel, how many versions of that variable are created?

- 1
- 256 / 32 = 8
- 32
- 256
- 4096
- 32 * 4096 = 131,072
- 256 * 4096 = 1,048,576

Consider multiplying two matrices of size 1024 x 1024. How many times is each element of the first matrix accessed from global memory if you do not use any tiling?

Consider multiplying two matrices of size 1024 x 1024. How many times is each element of the first matrix accessed from global memory if you use tiles of size 64 x 64?

Consider multiplying two matrices of size 1024 x 1024. How many tiles are needed if you use tiles of size 64 x 64?

Consider a kernel in which each thread requires 21 4-byte registers, each block allocates 12 KB of shared memory and each block contains 256 threads. Now consider that this kernel will run on a CC 6.1 GPU whose SMs have 64KB of registers, 96KB of shared memory and are limited to 2048 threads and 32 blocks. What is the maximum number of threads resident in an SM at any one time?

Consider a kernel in which each thread requires 21 4-byte registers, each block allocates 12 KB of shared memory and each block contains 256 threads. Now consider that this kernel will run on a CC 6.1 GPU whose SMs have 64KB of registers, 96KB of shared memory and are limited to 2048 threads and 32 blocks. What is the limiting factor that determines the number of threads resident in an SM at any one time? If multiple constraints yield the same bound, check them all.

- ☐ register limit of 64KB
- ☐ shared memory limit of 96KB
- ☐ thread limit of 2048
- ☐ block limit of 32

Consider a kernel in which each thread performs 1000 integer or single precision floating point operations but must load or store 33 4-byte data values to global memory while doing so. If the GPU device can perform 3470 GFLOPS (1 GFLOP is about 10^9 FLOPS) and has a memory bandwidth of 192 GB/s (1 GB is about 10^9 bytes), what is the limiting factor in the speed of execution of the kernel?

- ○ Compute resources
- ○ Memory bandwidth
- ○ Neither of the above

→