**OLYMPIA:**

**\*-pci:1**

**description: PCI bridge**

**product: Xeon E7 v2/Xeon E5 v2/Core i7 PCI Express Root Port 2a**

**vendor: Intel Corporation**

**physical id: 2**

**bus info: pci@0000:00:02.0**

**version: 04**

**width: 32 bits**

**clock: 33MHz**

**capabilities: pci normal\_decode bus\_master cap\_list**

**configuration: driver=pcieport**

**resources: irq:25 ioport:c000(size=4096) memory:ee000000-ef0fffff ioport:d0000000(size=301989888)**

**\*-display**

**description: VGA compatible controller**

**product: GK107GL [Quadro K600]**

**vendor: NVIDIA Corporation**

**physical id: 0**

**bus info: pci@0000:05:00.0**

**version: a1**

**width: 64 bits**

**clock: 33MHz**

**capabilities: vga\_controller bus\_master cap\_list rom**

**configuration: driver=nvidia latency=0**

**resources: irq:42 memory:ee000000-eeffffff memory:d0000000-dfffffff memory:e0000000-e1ffffff ioport:c000(size=128) memory:c0000-dffff**

**PORSCHE:**

**\*-pci:2**

**description: PCI bridge**

**product: Xeon E7 v3/Xeon E5 v3/Core i7 PCI Express Root Port 3**

**vendor: Intel Corporation**

**physical id: 3**

**bus info: pci@0000:00:03.0**

**version: 02**

**width: 32 bits**

**clock: 33MHz**

**capabilities: pci normal\_decode bus\_master cap\_list**

**configuration: driver=pcieport**

**resources: irq:27 ioport:1000(size=4096) memory:f2000000-f30fffff ioport:d0000000(size=301989888)**

**\*-display**

**description: VGA compatible controller**

**product: GM200 [GeForce GTX TITAN X]**

**vendor: NVIDIA Corporation**

**physical id: 0**

**bus info: pci@0000:03:00.0**

**version: a1**

**width: 64 bits**

**clock: 33MHz**

**capabilities: vga\_controller bus\_master cap\_list rom**

**configuration: driver=nvidia latency=0**

**resources: irq:38 memory:f2000000-f2ffffff memory:d0000000-dfffffff memory:e0000000-e1ffffff ioport:1000(size=128) memory:f3080000-f30fffff**

**FERRARI:**

**\*-pci:2**

**description: PCI bridge**

**product: Xeon E7 v3/Xeon E5 v3/Core i7 PCI Express Root Port 3**

**vendor: Intel Corporation**

**physical id: 3**

**bus info: pci@0000:00:03.0**

**version: 02**

**width: 32 bits**

**clock: 33MHz**

**capabilities: pci normal\_decode bus\_master cap\_list**

**configuration: driver=pcieport**

**resources: irq:27 ioport:1000(size=4096) memory:f2000000-f30fffff ioport:d0000000(size=301989888)**

**\*-display**

**description: VGA compatible controller**

**product: GM204 [GeForce GTX 980]**

**vendor: NVIDIA Corporation**

**physical id: 0**

**bus info: pci@0000:03:00.0**

**version: a1**

**width: 64 bits**

**clock: 33MHz**

**capabilities: vga\_controller bus\_master cap\_list rom**

**configuration: driver=nvidia latency=0**

**resources: irq:38 memory:f2000000-f2ffffff memory:d0000000-dfffffff memory:e0000000-e1ffffff ioport:1000(size=128) memory:f3080000-f30fffff**

**MACHINE INFO:**

**OLYMPIA:**

**Name: Quadro K600**

**Cuda Cores: 192**

**Clock Speed: 876 MHz**

**Memory Clock Speed: 891 Mhz**

**Total amount of shared memory per block: 49152 bytes**

**L2 Cache Size: 262144 bytes**

**CUDA Capability level: 8.0**

**FERRARI:**

**name:GeForce GTX 980**

**Cuda Cores: 2048**

**Clock Speed: 1216 MHz**

**Memory Clock Speed: 3505 Mhz**

**Total amount of shared memory per block: 49152 bytes**

**L2 Cache Size: 2097152 bytes**

**CUDA Capability level: 5.2**

**name:NVS 315**

**Cuda Cores: 48**

**Clock Speed: 1046 MHz**

**Memory Clock Speed: 875 Mhz**

**Total amount of shared memory per block: 49152 bytes**

**L2 Cache Size:65536 bytes**

**CUDA Capability level:2.1**

**PORSCHE:**

**name: GeForce GTX TITAN X**

**Cuda Cores: 3072**

**Clock Speed: 1076 MHz**

**Memory Clock Speed: 3505Mhz**

**Total amount of shared memory per block: 49152 bytes**

**L2 Cache Size: 65536 bytes**

**CUDA Capability level: 8.0**

**name: NVS 315**

**Cuda Cores: 48**

**Clock Speed: 1046 MHz**

**Memory Clock Speed: 875 Mhz**

**Total amount of shared memory per block: 49152 bytes**

**L2 Cache Size: 65536 bytes**

**CUDA Capability level: 8.0**

**SUMMARY:**

**Basic questions:**

1. What is CUDA and what is the compiler that we are using?

CUDA is a platform for parallel computing, NVCC

1. Who is the host? Who is the device?

The host is the cpu, the device is the gpu

1. What is the difference between normal C function and CUDA kernel function?

CUDA functions require declarations like \_\_global\_\_ or \_\_device\_\_

1. What is \_\_global\_\_ and what does it indicate?

It indicates a kernel.

1. What is SIMT ? Were you able to observe it in HelloWorld.cu?

Single instruction, multiple threads, yes.

1. What happened when you uncommented the line "//if(threadIdx.x==0)" in HelloWorld.cu?

It caused the print statement to only happen once.

1. What happened when you commented the line "cudaDeviceSynchronize();" in HelloWorld.cu? What does that tell you?

It caused the program to instantly end without printing anything, this tells me that it causes the processes to communicate with each other.

1. What is the syntax to invoke a CUDA kernel?

<<<gridsize,blocksize>>>

**Moderate questions:**

1. What is a Streaming Multiprocessor (SM)? How many do we have in the 3 GPUs?

The part of the GPU that runs CUDA kernels.   
prosche: [TITAN:24| NVS 315:1]  
Ferrari:[GTX 980:16|NVS315: 1]  
Olympia:[Quadro: 1]

1. What is a grid, a block and a thread?

thread: Execution of a kernel of a given index.  
block: Group of threads  
grid: a group of blocks

1. Explain the memory model that CUDA exposes to the programmer.
2. Look into matmultKernel00.cu in PA5.tar. What does the \_shared\_ keyword tell you?

It tells me that shared memory is used

1. How is the CUDA memory model different from the standard memory model on CPU?

CUDA has much more distribution of memory to specific places.

1. With respect to the previous 3 questions, what is the advantage that comes with CPU programming? (Hint: Shared memory in GPU is equivalent to \_\_\_\_ of CPU)

Even though GPUs have more cores, the CPU cores run much faster and have more features

1. What are blockIdx.x, threadIdx.y, blockDim.z? What do they tell you?

They tell me the id of the block in the x dimension, the y ID of the thread, and the size of the z dimension of the blocks.

1. Assume a 1D grid and 1D block architecture: 4 blocks, each with 8 threads. How many threads do we have in total ? How do you calculate the global thread ID? Show the calculation for global threadID=26.

To find globalID, you multiply block number(indexed at 0) by threads per block, then add the position in thread  
(block id) 3\* (threads per block)\*8+(position) 2=26. We have 32 threads in total

1. Assume a 1D grid and 2D block architecture: 4 blocks, each with 2 threads in x direction and 4 threads in y direction. How many threads do we have in total ? How do you calculate the global thread ID? Show the calculation for global threadID=26.

number of blocks times block dimensions

4\*(2\*4)=32

for ID:

(blockID\*blocksize)+(threadIdx.y\*blockDim.x) + threadIdx.x

26=(3\*8)+(0\*4)+2

1. Assume a 2D grid and 3D block architecture: 2 blocks in x direction and 2 blocks in y direction, each block with 2 threads in x direction and 2 threads in y direction and 2 threads in z direction. How many threads do we have in total ? How do you calculate the global thread ID? Show the calculation for global threadID=26.

(2\*2)\*(2\*2\*2)=32 threads

for global ID:

(blockIdx.x + (blockIdx.y\*gridDim.x))  
\*(blockDim.x\*blockDim.y\*blockDim.z)  
+(threadIdx.z\*(blockDim.x \*blockDim.y))  
+(ThreadIdx.y\*blockDim.x)  
+threadIdx.x

26=(1+(1\*2))  
\*(2\*2\*2)  
+(0\*(2\*2))  
+(0\*2)  
+2

1. What were your observations on varying block and grid sizes in HelloWorld.cu? Show how you changed the print statements to reflect this.

**Advanced questions:** (You can refer PA5 tarball and/or Internet)

1. Is the memory shared between CPU and GPU? Describe how data is transferred between CPU and GPU.
2. Assume a 1D grid and 1D block architecture: 64 blocks, each with 64 threads. How many threads do we have in total? Do all the threads execute in parallel?

we have 4096 threads, not all threads execute in parallel

1. What is a Warp? What is the Warp size and the scheduler found in the 3 GPUs?

A warp is what the sm splits blocks into, the warp size is 32.

1. From above 2 questions, how many threads can run in parallel at a given time step for the 3 GPUs?

6144

1. What is the theoretical peak performance that the 3 GPUs can achieve in terms of GFLOPsS? How is this derived?

cores\*SIMDs\*()\*clock speed

1. How is the given vecaddKernel(in PA5) different from HelloWorld Kernel?