Report - A0

Q1)

```
PS P:\heterogeneous-parallel-computing\A0> nvcc .\q1.cu
q1.cu
   Creating library a.lib and object a.exp
PS P:\heterogeneous-parallel-computing\A0> ./a


Device Name: GeForce 940MX
Compute capability: major: 5    minor: 0

Maximum block dimension in x: 1024
Maximum block dimension in y: 1024
Maximum block dimension in z: 64

Maximum grid dimension in x: 2147483647
Maximum grid dimension in y: 65535
Maximum grid dimension in z: 65535

Shared memory per block: 49152
Total global memory: 4294967296
Total constant memory: 65536

Warp size: 32
```

1. What is the architecture and compute capability of your GPU?
A1) GeForce 940MX has 28nm Maxwell architecture. Compute capability: Major-5 and Minor-0.

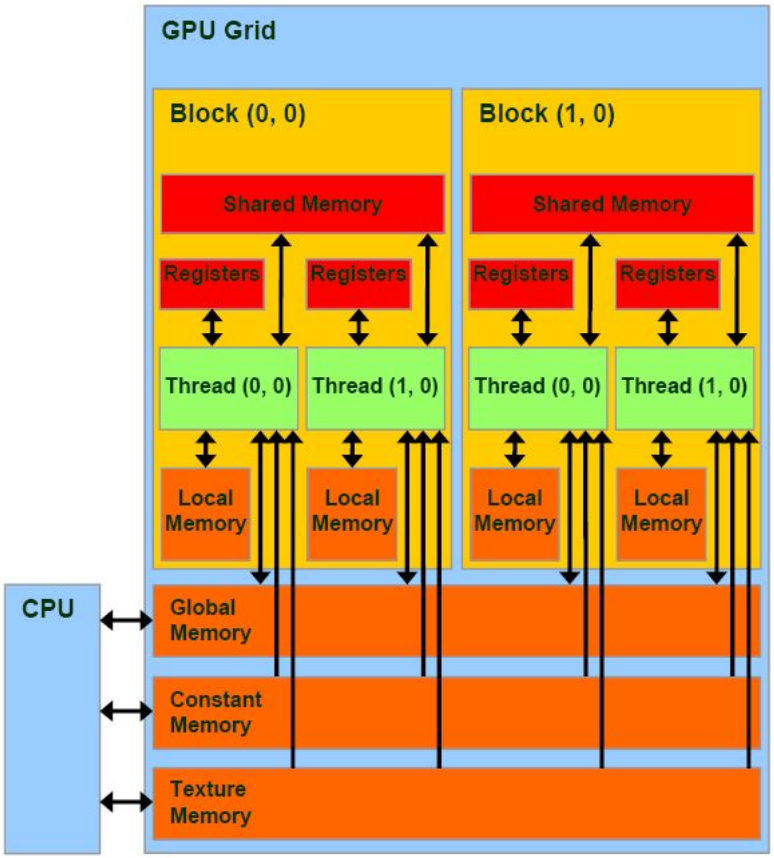
2. What are the maximum block dimensions for your GPU?
A2) Maximum block dimensions are X=1024, Y=1024 Z=64.

3. Suppose you are launching a one dimensional grid and block. If the hardware's maximum grid dimension is 65535 and the maximum block dimension is 512, what is the maximum number threads can be launched on the GPU?
A3) 512 * 65535 = 33,533,920 is the maximum number of threads that can be launched in GPU.

4. Under what conditions might a programmer choose not want to launch the maximum number of threads?
A4) In case the number of threads required by the program input is less, launching maximum number of threads is unnecessary. For example if we need to add two arrays each with N elements, launching MAX threads (where MAX>N) is not required.

5. What can limit a program from launching the maximum number of threads on a GPU?
A5) Register pressure limits launching of threads. Resources allocated to the threads limits the number of threads launched. If the number of resources used per thread is high, launching maximum number of threads is not possible.

6. What is shared memory? How much shared memory is on your GPU?
A6) Shared memory is the memory which is allocated per thread block, so all threads in the block have access to the same shared memory. Threads can access data in shared memory loaded from global memory by other threads within the same thread block.
Shared memory per block = 49,152 bytes.



7. What is global memory? How much global memory is on your GPU?
A7) For all the threads there is a global memory which can be accessed no matter which block the thread comes from. The lifetime of global memory is the same with the kernel function.
Total Global Memory: 4294967296 bytes.

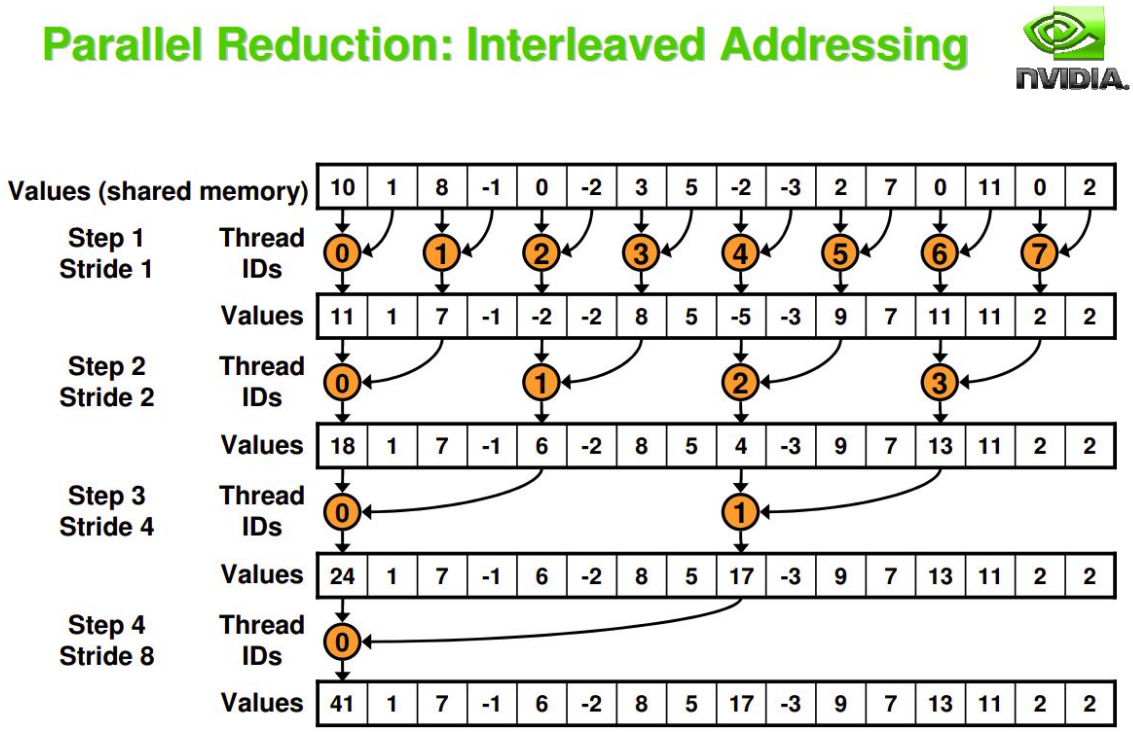
8. What is constant memory? How much constant memory is on your GPU?
A8) Constant memory is the memory used for data that will not change over the course of a kernel execution. Constant memory is common to all threads irrespective of the blocks as well (as shown in the diagram).
Total Constant Memory: 65536 bytes.

9. What does warp size signify on a GPU? What is your GPU's warp size?
A9) The multiprocessor creates, manages, schedules, and executes threads in groups of parallel threads called warps. Individual threads composing a warp start together at the same program address, but they have their own instruction address counter and register state and are therefore free to branch and execute independently.
Warp size: 32 threads.

10. Is double precision supported on your GPU?
A10) Yes in GeForce 940MX. All the current range of NVIDIA GPUs and since GT200 have double precision floating point

Q2) Write a CUDA program to calculate the sum of the elements in an array. The array contains single precision floating point numbers. Generate your input array.

All the 7 points have been documented in the source code attached.



Initialization of input array

```
70
71      // initialize host array elements
72      for (int i = 0; i < ARRAY_SIZE; ++i) {
73          host_arr[i] = (float)2;
74      }
75
```

Output

```
PS P:\heterogeneous-parallel-computing\A0> nvcc .\q2.cu
q2.cu
   Creating library a.lib and object a.exp
PS P:\heterogeneous-parallel-computing\A0> ./a
Sum = 2097152.000000
```