Contents

[1 GPU property description 2](#_Toc13494323)

[1) name: GeForce GTX 480 2](#_Toc13494324)

[2) Compute capability 2.0 2](#_Toc13494325)

[3) Total global memory(KB): 1509504 2](#_Toc13494326)

[4) Shared mem per block: 49152 B 2](#_Toc13494327)

[5) Registers per block: 32768 2](#_Toc13494328)

[6) warp size: 32 2](#_Toc13494329)

[7) Max threads per block: 1024 3](#_Toc13494330)

[8) Max thread dimension x:1024 y:1024 z:64 3](#_Toc13494331)

[9) Max grid size z:65535 y:65535 x:65535 3](#_Toc13494332)

[10) Clock rate: 3](#_Toc13494333)

[11) Total constant memory (bytes): 65536 3](#_Toc13494334)

[12) multiprocessor count 15 4](#_Toc13494335)

[13) memory bus width: 384 4](#_Toc13494336)

[14) Memory clock rate (KHz): 1848000 4](#_Toc13494337)

[15) L2 cache size (bytes): 786432 4](#_Toc13494338)

[16) max threads per SM: 1536 5](#_Toc13494339)

[2 Vector addition time comparison 6](#_Toc13494340)

[3 Mean Filter implementation 6](#_Toc13494341)

[a) GPU vs CPU time for execution 6](#_Toc13494342)

[b) Speedup Conclusion 6](#_Toc13494343)

# GPU property description

Device 0:

* ID of the device that we are connected to.

## name: GeForce GTX 480

* Name of the GPU version that used in the particular machine
* Contains 480 CUDA cores

## Compute capability 2.0

* Hardware version is referred as compute capability
* X.Y (X- major changes , Y- Minor changes)
* Hardware design, Number of cores, cache size, support arithmetic instruction are different for different version of compute capability
* Compute capability 2.x mean it has higher hardware resource compare to cc 1.x (e.g. in memory per thread )

## Total global memory(KB): 1509504

* SMs can read or write memory using GLS/GST instructions.
* A memory which is shared among the blocks and visible to all threads inside of a block. In this case entire memory allocated for all threads are 1.5GB. Apart from that each thread has its own memory called private local memory.

## Shared mem per block: 49152 B

* All Blocks are assigned to a SM which means it has a memory to share with all blocks. 49152B is the memory capacity is shared among the blocks which is assign to same SMs (If there are N number of block maximumly each block can have 49152/N number of Bytes of allocated memory)

## Registers per block: 32768

* Since blocks are assigned to Streaming Multiprocessors (SMs) it has some amount of registers on it. When the blocks are assigned to SMs, registers are shared with blocks. The maximum number of registers can be allocated to a block when there is only one block present.

## warp size: 32

* Number of threads in a warp(executed by the multi-processor)
* Blocks are divided into sub-blocks called warp

## Max threads per block: 1024

* Total number of threads in a block.
* Maximum number of threads that can be allocated from the SM(Streaming Multiprocessor).

## Max thread dimension x:1024 y:1024 z:64

* This refers to the maximum number of threads that can be existed on each dimension. For compute capability 2.0 machine, some Thread dimension combinations are
  + 1024 X 1 X 1
  + 512 X 2 X 1
  + Total numbers of thread in thread dimension should not exceed the Maximum number of threads in a block.

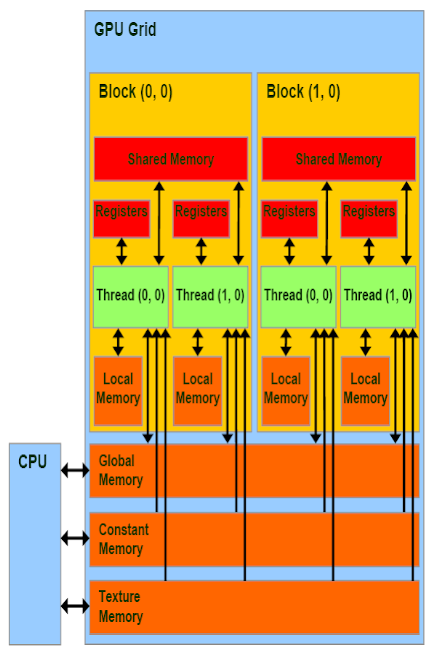
## Max grid size z:65535 y:65535 x:65535

* An upper bound for the grid in x, y, z dimensions. At the execution time maximumly that number of grids can be created with limited constrain of SMs.

## Clock rate:

* The speed of the instruction per second related to the core.

## Total constant memory (bytes): 65536



* Data that will not change over the course of kernel execution.

## multiprocessor count 15

* Number of multiprocessors that are presented on the GPU. Since this GPU has a 480 Core that mean each SM have 32 processor itself (480/15).

## memory bus width: 384

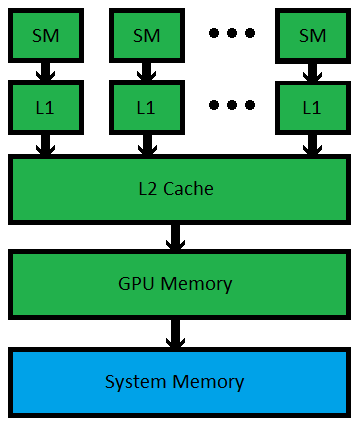
* The rate at which data can be read from or stored into memory

## Memory clock rate (KHz): 1848000

* The frequency of the VRAM linked to the GPU.

## L2 cache size (bytes): 786432

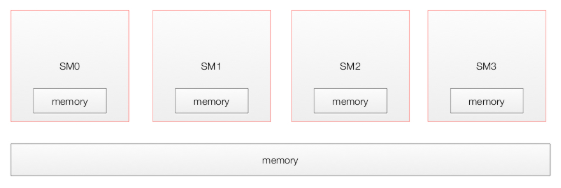
* Total memory shared among the SMs.
* Secondary level memory for the faster performance.



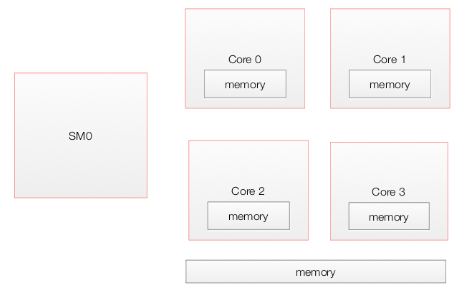
## max threads per SM: 1536

All blocks are assigned to the SMs. Max thread per SM means, maximum number of threads that can handle by the SMs’ multiple core. In this particular machine one SM can only handle 1536 thread that mean it can be assigned less than 2 blocks at once(Maximum number of threads per block is 1024).

* GPU consist of many streaming multiprocessors (SMs) with a global memory accessible by all SMs and a local memory



* Each SM contain multiple cores which share a shared memory as well as a local memory.



# Vector addition time comparison

Below table will elaborate the time comparison between GPU and CPU for 1-dimension matrix.

|  |  |  |
| --- | --- | --- |
| No. elements | GPU Time | CPU Time |
| 100 | 0.000016 | 0.000006 |
| 10000 | 0.000022 | 0.000086 |
| 1000000 | 0.000181 | 0.004772 |
| 10000000 | 0.000169 | 0.041456 |

# Mean Filter implementation

## GPU vs CPU time for execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image size | Window size | GPU Time | CPU Time | Speedup  (CPU time/GPU time) |
| 640x640 | 3 | 0.000014 | 0.01185 | 846 |
| 640x640 | 5 | 0.000014 | 0.02662 | 1929 |
| 1280x1280 | 3 | 0.000019 | 0.04727 | 2449 |
| 1280x1280 | 5 | 0.000019 | 0.10709 | 5492 |

## Speedup Conclusion

1. Speed up at its lowest : 640x640 window size 3
   1. 640x640 image GPU time are exact same but CPU time are totally different because, window 5 will consume more computation power (Host) when comparing to window 3.
   2. Therefore, 640x640 windows 3 will have lower speed.
   3. In the case of 1280x1280 image, number of cells are comparatively high than 640x640. Therefore, CPU time execution will increase. But, GPU time for execution not increased like CPU time because it was executed parallelly.
   4. Therefore, speedup for 1280x1280 comparatively high.
2. Speed up at its highest : 1280x1280 window size 5
   1. In the case of 1280x1280 image, number of cells are comparatively high than 640x640. Therefore, CPU time execution will increase. But, GPU time for execution not increased like CPU time because it was executed parallelly.
   2. Therefore, speedup for 1280x1280 comparatively high.
   3. In the GPU, irrespective of the image it will get almost same amount of time to execute because of the parallel running.