Nvidia compute capabilities and device architectures

6.7 Nvidia compute capabilities and device architectures[7]

There are limits on the number of threads and the number of blocks. The limits depend on what Nvidia calls the **compute capability** of the GPU. The compute capability is a number having the form *a.b*. Currently the *a*-value or major revision number can be 1, 2, 3, 5, 6, 7, 8. (There is no major revision number 4.) The possible *b*-values or minor revision numbers depend on the major revision value, but currently

[6] With the introduction of CUDA 9 and the Pascal processor, it became possible to synchronize threads in multiple blocks. See Subsection 7.1.13 and Exercise 7.6.

[7] The values in this section are current as of spring 2021, but some of them may change when Nvidia releases new GPUs and new versions of CUDA.

Table 6.3 GPU architectures and compute capabilities.

Name	Ampere	Tesla	Fermi	Kepler	Maxwell	Pascal	Volta	Turing
Compute capability	8.0	1.*b*	2.*b*	3.*b*	5.*b*	6.*b*	7.0	7.5

they fall in the range 0–7. CUDA no longer supports devices with compute capability < 3.

For devices with compute capability > 1, the maximum number of threads per block is 1024. For devices with compute capability 2.*b*, the maximum number of threads that can be assigned to a single SM is 1536, and for devices with compute capability > 2, the maximum is currently 2048. There are also limits on the sizes of the dimensions in both blocks and grids. For example, for compute capability > 1, the maximum *x*- or *y*-dimension is 1024, and the maximum *z*-dimension is 64. For further information, see the appendix on compute capabilities in the CUDA C++ Programming Guide [11].

Nvidia also has names for the microarchitectures of its GPUs. Table 6.3 shows the current list of architectures and some of their corresponding compute capabilities. Somewhat confusingly, Nvidia also uses Tesla as the name for their products targeting GPGPU.

We should note that Nvidia has a number of "product families" that can consist of anything from an Nvidia-based graphics card to a "system on a chip," which has the main hardware components of a system, such as a mobile phone in a single integrated circuit.

Finally, note that there are a number of versions of the CUDA API, and they do *not* correspond to the compute capabilities of the different GPUs.

Notes:

- Read this to gain a better understanding of the capabilities of your NVIDIA GPU
- In NVIDIA's CUDA architecture, Compute Capability is a version number that indicates the features and capabilities supported by a particular GPU architecture. It is represented as a two-part number: X.Y, where X is the major version and Y is the minor version.
- Key Aspects of Compute Capability:
    - GPU Architecture Generation:
        - The major version (X) indicates the GPU architecture generation (e.g., Kepler, Maxwell, Pascal, Volta, Turing, Ampere).
        - The minor version (Y) provides information about specific features within that generation.
    - Features and Functionality:
        - Compute Capability defines which features are supported by the GPU hardware, such as:
        - Maximum number of threads per block.
        - Maximum grid dimensions.
        - Warp size (number of threads per warp).
        - Support for specific instructions (e.g., double-precision, tensor cores).
        - Hardware acceleration features.
- I have on my laptop a 3050ti which has a 7.5 compute capability
    - This means that
        - Warp size: 32 threads
        - Maximum number of threads per block: 1024
        - Maximum x-dimension of a block: 1024
        - Maximum y-dimension of a block: 1024
        - Maximum z-dimension of a block: 64
        - Maximum shared memory per block: 64 KB
        - Maximum registers per block: 65,536
        - Maximum grid dimensions: (2^31 - 1, 65535, 65535)
        - Cooperative groups
        - Independent thread scheduling
        - AMONG OTHER FEATURES