Whitepaper

NVIDIA's Next Generation CUDA™ Compute Architecture: Fermi™

A Brief History of GPU Computing

The graphics processing unit (GPU), first invented by NVIDIA in 1999, is the most pervasive parallel processor to date. Fueled by the insatiable desire for life-like real-time graphics, the GPU has evolved into a processor with unprecedented floating-point performance and programmability; today's GPUs greatly outpace CPUs in arithmetic throughput and memory bandwidth, making them the ideal processor to accelerate a variety of data parallel applications.

Efforts to exploit the GPU for non-graphical applications have been underway since 2003. By using high-level shading languages such as DirectX, OpenGL and Cg, various data parallel algorithms have been ported to the GPU. Problems such as protein folding, stock options pricing, SQL queries and MRI reconstruction achieved remarkable performance speedups on the GPU. These early efforts that used graphics APIs for general purpose computing were known as GPGPU programs.

While the GPGPU model demonstrated great speedups, it faced several drawbacks. First, it required the programmer to possess intimate knowledge of graphics APIs and GPU architecture. Second, problems had to be expressed in terms of vertex coordinates, textures and shader programs, greatly increasing program complexity. Third, basic programming features such as addressable read and write were not supported, greatly restricting the programming model. Lastly, the lack of double precision support (until recently) meant some scientific applications could not be run on the GPU.

To address these problems, NVIDIA introduced two key technologies—the G80 unified graphics and compute architecture (first introduced in GeForce 8800®, Quadro FX 5600®, and Tesla C870® GPUs), and CUDA, a software and hardware architecture that enabled the GPU to be programmed with a variety of high level programming languages. Together, these two technologies represented a new way of using the GPU. Instead of programming dedicated graphics units with graphics APIs, the programmer could now write C programs with CUDA extensions and target a general purpose, massively parallel processor. We called this new way of GPU programming "GPU Computing"—it signified broader application support, wider programming language support, and a clear separation from the early "GPGPU" model of programming.

The G80 Architecture

NVIDIA's GeForce 8800 was the product that gave birth to the new GPU Computing model. Introduced in November 2006, the G80 based GeForce 8800 brought several key innovations to GPU Computing:

- G80 was the first GPU to support C, allowing programmers to use the power of the GPU without having to learn a new programming language.
- G80 was the first GPU to replace the separate vertex and pixel pipelines with a single, unified processor that executed vertex, geometry, pixel, and computing programs.
- G80 was the first GPU to utilize a scalar thread processor, eliminating the need for programmers to manually manage vector registers.
- G80 introduced the single-instruction multiple-thread (SIMT) execution model where independent multiple threads execute concurrently using a single instruction.
- G80 introduced shared memory and barrier synchronization for inter-thread communication.

In June 2008, NVIDIA introduced a major revision to the G80 architecture. The second generation unified architecture—GT200 (first introduced in the GeForce GTX 280, Quadro FX 5800, and Tesla T10 GPUs)—increased the number of streaming processor cores (subsequently referred to as CUDA cores) from 128 to 240. Each processor register file was doubled in size, allowing a greater number of threads to execute on-chip at any given time. Hardware memory access coalescing was added to improve memory access efficiency. Double precision floating point support was also added to address the needs of scientific and high-performance computing (HPC) applications.

When designing each new generation GPU, it has always been the philosophy at NVIDIA to improve both existing application performance and GPU programmability; while faster application performance brings immediate benefits, it is the GPU's relentless advancement in programmability that has allowed it to evolve into the most versatile parallel processor of our time. It was with this mindset that we set out to develop the successor to the GT200 architecture.

NVIDIA's Next Generation CUDA Compute and Graphics Architecture, Code-Named "Fermi"

The Fermi architecture is the most significant leap forward in GPU architecture since the original G80. G80 was our initial vision of what a unified graphics and computing parallel processor should look like. GT200 extended the performance and functionality of G80. With Fermi, we have taken all we have learned from the two prior processors and all the applications that were written for them, and employed a completely new approach to design to create the world's first computational GPU. When we started laying the groundwork for Fermi, we gathered extensive user feedback on GPU computing since the introduction of G80 and GT200, and focused on the following key areas for improvement:

- While single precision floating point performance was on the order of ten times the performance of desktop CPUs, some GPU computing applications desired more double precision performance as well.
- ECC memory so some GPU computing users could deploy large numbers of GPUs in datacenter installations.
- Some parallel algorithms were unable to use the GPU shared memory, and users requested a true cache architecture to aid them.
- Many CUDA programmers requested more than 16 KB of SM shared memory to speed up their applications.
- Users requested faster context switches between application programs and faster graphics and computing interoperation.
- Users requested faster read-modify-write atomic operations for their parallel algorithms.

With these requests in mind, the Fermi team designed a processor that greatly increases raw compute horsepower, and through architectural innovations, also offers dramatically increased programmability and compute efficiency. The key architectural highlights of Fermi are:

- **Third Generation Streaming Multiprocessor (SM)**
  - 32 CUDA cores per SM, 4x over GT200
  - 8x the peak double precision floating point performance over GT200
  - Dual Warp Scheduler that schedules and dispatches two warps of 32 threads per clock
  - 64 KB of RAM with a configurable partitioning of shared memory and L1 cache
- **Second Generation Parallel Thread Execution ISA**
  - Unified Address Space with Full C++ Support
  - Optimized for OpenCL and DirectCompute
  - Full IEEE 754-2008 32-bit and 64-bit precision
  - Full 32-bit integer path with 64-bit extensions
  - Memory access instructions to support transition to 64-bit addressing
  - Improved Performance through Predication
- **Improved Memory Subsystem**
  - NVIDIA Parallel DataCache™ hierarchy with Configurable L1 and Unified L2 Caches
  - First GPU with ECC memory support
  - Greatly improved atomic memory operation performance
- **NVIDIA GigaThread™ Engine**
  - 10x faster application context switching
  - Concurrent kernel execution
  - Out of Order thread block execution
  - Dual overlapped memory transfer engines