WOLF Advanced Technology

NVIDIA GPU ARCHITECTURE: FROM TURING TO BLACKWELL

– WHITEPAPER

Written by: Shari Beck
Sr. Product Manager

WOLF ADVANCED TECHNOLOGY

INTRODUCTION

This paper focuses on key improvements when upgrading an NVIDIA® GPU from Turing to Blackwell, looking at architecture improvements specifically for high-end embedded GPUs, Turing TU104, Ampere GA104, Ada AD103 and Blackwell GB203.

NVIDIA GPUs have always excelled at video graphics processing and in providing support for general purpose data processing that benefits from massive parallel processing algorithms. NVIDIA also became a leader in artificial intelligence (AI) processing with the inclusion of Tensor cores in GPUs. Tensor cores were introduced by NVIDIA in 2017 as a key feature of the Volta data center GPUs, followed by 2nd generation Tensor cores in 2018 in the Turing architecture for desktop and other use cases. The Turing architecture also introduced new Ray Tracing cores used to accelerate photo realistic rendering. With each new GPU generation NVIDIA made updates to CUDA® core processing data paths, updated Tensor cores with new data precision handling support, and updated Ray Tracing core capabilities. New manufacturing processes provided support for the design of denser GPUs with more cores running at higher clock speeds. Each generation GPUs have become more performant and more necessary for modern data processing.

HIGH-LEVEL COMPONENTS USED IN GPUS

At a high level the main building blocks of the NVIDIA GPU architecture have remained similar in function from generation to generation. The GPU communicates with the host system using PCIe, it has interfaces to communicate with dedicated external memory, it includes internal memory caches, and it has specialized scheduling and job handling components.

The processing cores (CUDA, Tensor, and Ray Tracing cores) are included in Graphics Processing Clusters (GPCs) which include blocks to efficiently route tasks to the appropriate core.

There are specialized blocks for hardware acceleration for video encoding (NVENC) and decoding (NVDEC), which are typically used to encode video to modern formats like AVC (H.264), HEVC (H.265) and AV1 (an Alliance for Open Media royalty free format).

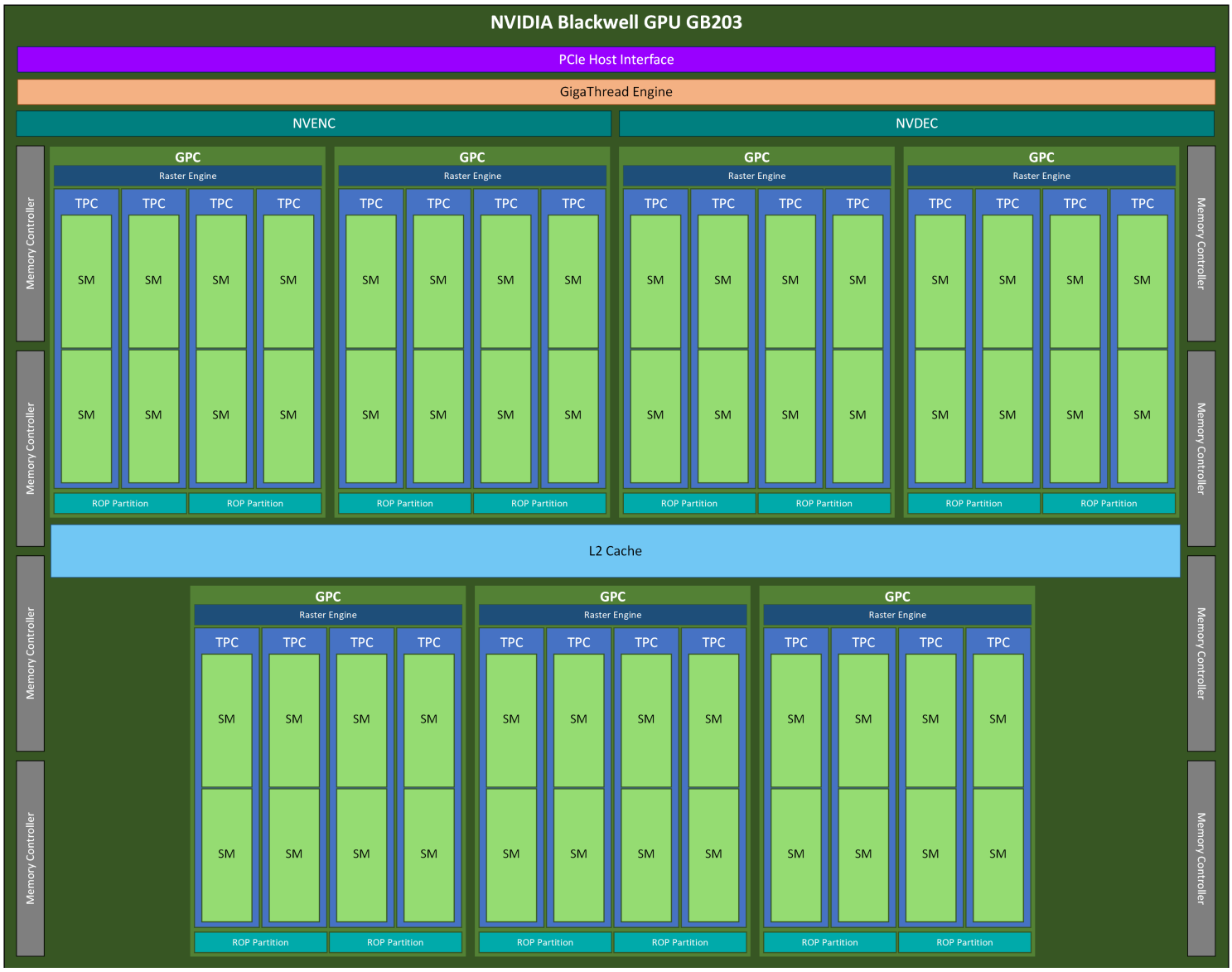

Figure 1: NVIDIA Ampere GA104 architecture. Details for each SM are shown in Figure 2.

Table 1: High-Level Component Blocks used in an NVIDIA GPUs

	Turing TU104	Ampere GA104	Ada AD103	Blackwell GB203
PCIe Host Interface	Gen 3	Gen 4	Gen 4	Gen 5
Memory type supported	GDDR6	GDDR6	GDDR6	GDDR7
Max. Memory Size (GB)	16 (8× 2GB)	16 (8× 2GB)	16 (8× 2GB)	24 (8× 3GB)
Max. Memory Bandwidth (GB/s)	448	512	576	896
L2 Cache Size (MB)	4	4	64	64
NVENC Gen	7th	7th	8th	9th
NVDEC Gen	4th	5th	5th	6th
Graphics Processing Clusters (GPCs)/GPU	6	6	7	7

PCIe Host Interface: The Ampere GPU updated the PCIe host interface to PCIe Gen 4, and then the Blackwell updated it again to Gen 5. Each PCIe generation can provide double the bandwidth compared to the previous generation but is still fully compatible with the previous PCIe generation interfaces.

Memory Type Supported: Turing introduced support for GDDR6 memory, then Blackwell introduced support for GDDR7 memory. Each new generation supports higher bandwidth and is more energy efficient than the previous generation. The GDDR7 memory can also use higher density 3GB memory chips, allowing more memory to be included using the same footprint.

L2 Cache Size: Accessing data from the internal L2 cache is faster than accessing it from the external memory. A large L2 cache is especially beneficial when performing complex operations that need quick access to frequently accessed data, such as some AI operations and Ray Tracing. The Ada architecture provided a large increase to the L2 cache size to support these more complex operations.

NVENC/NVDEC: Hardware accelerated encoding and decoding offload the most computationally intense tasks from the CPU to the GPU, providing real-time performance for high resolution encoding and decoding. NVIDIA GPUs support advanced video encoding and decoding options, with each generation supporting more formats, higher frame resolutions, higher frame rates, and more color format options.

HIGH-LEVEL CHANGES

NVIDIA has always pushed the boundaries of what is possible for GPU density. NVIDIA GPUs use advanced manufacturing processes that provide denser chips with more transistors, higher clock speeds, and therefore higher performance.

Table 2: High-Level Architecture Changes to NVIDIA GPUs

	Turing TU104	Ampere GA104	Ada AD103	Blackwell GB203
Manufacturing Process	12nm	8nm	5nm (TSMC 4N)	5nm (TSMC 4N)
Transistors per GPU	13.6 billion	17.4 billion	45.9 billion	45.6 billion
Max. TGP (Watts)	150	150	150	150
Max. GPU Boost Clock (MHz)	1770	1725	2505	2617
DisplayPort output	1.4a 4K @240 Hz 8K @ 60 Hz	1.4a 4K @240 Hz + HDR 8K @ 60 Hz + HDR	1.4a 4K @240 Hz + HDR 8K @ 60 Hz + HDR	2.1b 4K @480 Hz + HDR 8K @ 165 Hz + HDR
HDMI output	2.0b 4K @ 60Hz 8K @ 30Hz	2.1 4K @240 Hz + HDR 8K @ 60 Hz + HDR	2.1 4K @240 Hz + HDR 8K @ 60 Hz + HDR	2.1b 4K @240 Hz + HDR 8K @ 60 Hz + HDR

Manufacturing Process and Power Efficiency: Chips are manufactured using processes that determine the size of each transistor on the chip measured in nano meters (nm). The smaller the size is the faster the transistor will be and the less power it will use at the same performance level.

Display and Video Engine: With each generation support for higher resolution display output has increased, and when using an Ampere GPU with VESA Display Stream Compression (DSC) technology enabled High Dynamic Range (HDR) rendering is also supported.

COMPONENTS IN GRAPHICS PROCESSING CLUSTERS (GPCS)

Graphics processing clusters contain:

- A raster engine (with Raster Operation Partitions, or ROPs) used when processing graphical information to translate vector information into pixel data
- Texture Processing Clusters (TPCs) used to perform data processing

Table 3: Component blocks in an NVIDIA Graphics Processing Cluster (GPC)

	Turing TU104	Ampere GA104	Ada AD103	Blackwell GB203
Raster Engine ROPs	64 (tied to the memory controller and L2 cache)	96 (integrated into each GPC)	112 (integrated into each GPC)	112 (integrated into each GPC)
Texture Processing Clusters (TPCs) per GPC	4 per GPC	4 per GPC	6 per GPC	7 per GPC
TPC available/GPU	24	24	38	42

Raster Operator (ROP) Units: In the Turing architecture ROPs were tied to the memory controller and L2 cache. In later architectures ROPs are integrated into each Graphics Processing Cluster (GPC). Including ROP partitions in the GPC helps to eliminate bottlenecks.

Texture Processing Clusters (TPCs): Each TPC contains a PolyMorph Engine (for graphic information pre-processing when communicating with the Raster Engine) and two Streaming Multiprocessors (SM) which contain the CUDA cores, Tensor cores and Ray Tracing cores.

STREAMING MULTIPROCESSOR (SM) ARCHITECTURE

The Streaming Multiprocessors contain the cores that do the GPU processing. They include:

- CUDA cores for Floating Point (FP) and Integer (INT) calculations
- Tensor cores for performing matrix operations, which are typically used by AI algorithms
- Ray Tracing cores used for photo realistic graphics processing

Major improvements have been made to components found in the Streaming Multiprocessors in each subsequent generation.

Figure 2: NVIDIA Streaming Multiprocessor architecture for Turing, Ampere/Ada, Blackwell

Each Streaming Multiprocessor (SM) includes:

- Four SM Processing Blocks (Partitions), and each includes:
  - CUDA data paths which can handle Floating Point (FP) and/or Integer (INT) calculations. The way the CUDA cores are assigned to perform a FP or INT calculation has changed over the generations (see following section for more info).
  - Tensor Cores (introduced with Volta/Turing)
  - Instruction cache per SM Block
  - Warp scheduler and Dispatch Unit that assign tasks. The way tasks are assigned has been significantly improved in recent generations to optimize core use.
  - Register File
  - Load/store units (LD/ST units)
  - Special function units (SFU) for transcendental math functions (e.g., log x, sin x, cos x, ex)
- L1 Data Cache/Shared Memory
- Texture Units
- Ray Tracing Core (introduced with Turing)
- Two FP64 units

Table 4: Streaming Multiprocessor Changes

	Turing TU104	Ampere GA104	Ada AD103	Blackwell GB203
CUDA Cores/ SMUse (FP/INT or just FP	64 FP32 use only, 64 INT32 use only	64 FP32 use only, 64 FP32 or INT32	64 FP32 use only, 64 FP32 or INT32	All 128 cores can be used for FP32 or INT32
Tensor Cores	320 of Gen2	184 of Gen3	304 of Gen4	320 of Gen5
Ray Tracing Cores	Gen 1	Gen 2	Gen 3	Gen4
Shared Memory/ L1 Cache/SM	96 KB	128 KB	128 KB	128 KB
Total Shared Memory/L1 Cache	4608 KB	6144 KB	9728 KB	10752 KB

CUDA Datapath Changes

CUDA cores can be used for Floating Point FP32 or for Integer INT32 operations. With the Turning architecture SM partitions separated the CUDA cores into two data paths, one dedicated to FP32, and the other dedicated to INT32. With the Ampere and Ada architecture the two data paths were still present, and one of them was still dedicated to FP32, but the other data path could be used for either FP32 or INT32, depending on what was in demand. With the Blackwell architecture all CUDA cores can be used for either FP32 or INT32, providing maximum flexibility, ensuring that a task is never waiting for the appropriate data path to become available.

The change in how CUDA core data paths were assigned was primarily due to the change in how GPU use has continued to evolve. Graphic workloads typically required more FP32 calculations than INT32 calculations. As workloads shifted to doing more HPC and AI calculations there was a need for more INT32 calculation capability. With Blackwell the transition to maximum flexibility has been achieved, allowing any algorithm to fully utilize all cores without restrictions.

Tensor Cores Gen 1 to 5

Tensor cores were developed to support the matrix fused multiply and accumulate (FMA) math operations that are commonly used in AI and HPC applications. The initial implementation in the Volta GPU only supported FP16 data precision. With each subsequent generation Turing cores have supported more data precisions. This provides AI application developers with more options to develop more efficient algorithms that only use the required data precision.

With the Ampere Gen3 Tensor cores NVIDIA also supported a new Fine-Grained Structured Sparsity feature, which uses only the subset of weights that have acquired a meaningful purpose during the learning process, which leads to even more efficient inference acceleration when sparsity is applicable. This is typically seen when a large part of the dataset consists of zeros, for example when processing a video frame where many of the pixels are black, with zero RGB values. By ignoring the data that is not meaningful it is possible to double the amount of data that can be processed.

In theory, when the data precision is halved and therefore the data size is halved the number of operations can double, providing there are no bandwidth bottlenecks. For example, when going from FP16 to FP8 only half the data needs to be processed, so the performance could theoretically double.

Table 5: Tensor Core Data Precisions per GPU Generation

GPU Architecture	Tensor Core Gen	Supported Tensor Core Precisions	Number of Tensor Cores
Volta	Gen 1	FP16	(V100 only)
Turing	Gen 2	FP16 **Adds:** INT1, INT4, INT8	384 (TU104)
Ampere	Gen 3	FP16, INT4, INT8 **Adds:** TF32, BF16, Sparsity	160 (GA104)
Ada	Gen 4	FP16, INT8, TF32, BF16, Sparsity **Adds:** FP8	304 (AD103)
Blackwell	Gen 5	FP16, FP8, INT8, TF32, BF16, Sparsity **Adds:** FP4 and FP6, FP8 Gen2, microscaling precision support	320 (GB203)

Ray Tracing Cores Gen 1 to 4

The Gen 1 Ray Tracing Cores introduced in Turing provided content creators with a new and vastly improved way to use the GPU to render photo-realistic frames.

The Gen 2 Ray Tracing cores found in Ampere architecture GPUs can effectively deliver twice the performance of the first-generation Ray Tracing cores found in Turing architecture GPUs. Ampere SMs also allow RT core and CUDA core compute workloads to run concurrently, introducing even more efficiencies. For users who need to render complex models with accurate shadows, reflections and refractions, or to render ray-traced motion blur, the Ampere RT cores provide big performance improvements.

Ada RT Gen3 cores once again took a big leap up in performance providing new RT functionality that could provide up to double increased performance for developers who adopted the new RT features.

Blackwell RT Gen4 cores again double the throughput compared to the Ada RT Gen3 cores.

Warp Scheduling Changes

In an NVIDIA GPU the basic unit for executing an instruction is the warp. A warp is a collection of threads that all share the same code and are all executed simultaneously by a Streaming Multiprocessor (SM). Multiple warps can be executed on an SM at once.

Volta GPU SM processing blocks each had a single warp scheduler and a single dispatch unit. This meant that Volta could only issue one independent instruction per clock cycle. However, it gained independent thread scheduling, it included a program counter and call stack per thread, and it included a schedule optimizer. Taken together this allows threads to diverge at sub-warp granularity, which helps to ensure optimal usage of the cores. Subsequent GPUs inherited all the Volta improvements to warp scheduling, resulting in significant processing optimization compared to previous generations.

SOFTWARE TOOLS

NVIDIA provides numerous software tools to help developers to accelerate GPU-based application development. With each new GPU generation new tools and new features are added.

CUDA Toolkit and CUDA Compute

The CUDA Toolkit includes GPU-accelerated libraries, a compiler, development tools and the CUDA runtime. Each major new architecture release is accompanied by a new version of the CUDA Toolkit, which includes tips for using existing code on newer architecture GPUs, as well as instructions for using new features only available when using the newer GPU architecture.

CUDA Compute capability allows developers to determine the features supported by a GPU. For specific information the NVIDIA CUDA Toolkit Documentation provides tables that list the “Feature Support per Compute Capability” and the “Technical Specifications per Compute Capability”.

https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#compute-capabilities

NVIDIA AI and HPC SDKs and Libraries

NVIDIA provides a rich ecosystem of SDKs and libraries specifically to support AI and HPC development on NVIDIA GPUs. Some examples:

NVIDIA also provides integrated support in a number of open source partner libraries, providing built-in GPU acceleration for numerous types of applications.

For AI see: https://developer.nvidia.com/deep-learning-software

For HPC see: https://developer.nvidia.com/hpc-sdk

CONCLUSION

With the release of each new GPU generation NVIDIA has continued to deliver huge increases in performance and revolutionary new features. Whether an application requires enhanced image quality or powerful compute and AI acceleration, upgrading to the latest NVIDIA GPU will provide significant performance improvements.

REFERENCES: NVIDIA GPU WHITEPAPERS

Blackwell, v1.1: https://images.nvidia.com/aem-dam/Solutions/geforce/blackwell/nvidia-rtx-blackwell-gpu-architecture.pdf

Ada, v2.02: https://images.nvidia.com/aem-dam/Solutions/geforce/ada/nvidia-ada-gpu-architecture.pdf

Ampere, v.2.0: https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf

Turing, v.01: https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf

WOLF
ADVANCED TECHNOLOGY