NVIDIA ADA LOVELACE PROFESSIONAL GPU ARCHITECTURE

Designed to deliver outstanding, professional graphics, AI, and compute performance.

V1.1

Table of Contents

List of Figures

List of Tables

Introduction

Launched in 2018, NVIDIA's® Turing™ GPU Architecture ushered in the future of 3D graphics and GPU-accelerated computing. Turing provided major advances in efficiency and performance for PC gaming, professional graphics applications, and deep learning inferencing. Using new hardware-based accelerators, Turing fused rasterization, real-time ray tracing, AI, and simulation to enable incredible realism in professional content creation software, cinematic-quality interactive experiences, and PC games. Two years later in 2020, the NVIDIA Ampere architecture incorporated more powerful RT Cores and Tensor Cores, along with a novel SM structure that offered 2x FP32 performance, clock-for-clock, compared to Turing GPUs. These innovations allowed the Ampere architecture to run up to 1.7x faster than Turing in traditional raster graphics, and up to 2x faster in ray tracing.

The new NVIDIA Ada Lovelace GPU architecture, named after mathematician Ada Lovelace, who is often regarded as the world's first computer programmer[1], raises the bar far above Turing and Ampere GPUs. While improvements in the silicon manufacturing process have slowed, modern computer graphics have seen an exponential rise in complexity. Increases in geometric complexity and innovations in lighting have resulted in graphics that look more lifelike than ever before.

Ada provides the largest generational performance upgrade in the history of NVIDIA. This is made possible by three key innovations:

- **Revolutionary New Architecture**: NVIDIA Ada architecture GPUs deliver outstanding performance for graphics, AI, and compute workloads with exceptional architectural and power efficiency. After the baseline design for the Ada SM was established, the chip was scaled up to shatter records. Manufacturing innovations and materials research enabled NVIDIA engineers to craft a GPU with 76.3 billion transistors and 18,432 CUDA Cores capable of running at clocks over 2.5 GHz, while maintaining the same 300W TGP as the prior generation professional graphics flagship NVIDIA RTX™ A6000 GPU. The result is the world's fastest professional GPU with the power, acoustics, and temperature characteristics expected of a high-end graphics card.
- **New Ada RT Core for Faster Ray Tracing:** For decades, rendering ray-traced scenes with physically correct lighting in real time has been considered the holy grail of graphics. At the same time, the geometric complexity of environments and objects continues to increase as professional graphics continually strive to provide the most accurate representations of the real world. The Ada RT Core has been enhanced to deliver 2x faster ray-triangle intersection testing and includes two important new hardware units. An Opacity Micromap Engine speeds up ray tracing of alpha-tested geometry by a factor of 2x, and a Displaced Micro-Mesh Engine generates Displaced Micro-Triangles on-the-fly to create additional geometry. The Micro-Mesh Engine provides the benefit of increased geometric complexity without the traditional performance and storage costs of complex geometries.
- **Shader Execution Reordering:** NVIDIA Ada GPUs support Shader Execution Reordering (SER) which dynamically organizes and reorders shading workloads to improve RT shading efficiency.
- **NVIDIA DLSS 3:** The Ada architecture features an all-new Optical Flow Accelerator and AI frame generation that boosts DLSS 3's frame rates up to 2x over the previous DLSS 2.0 while maintaining or exceeding native image quality. Compared to traditional brute-force graphics rendering, DLSS 3 is ultimately up to 4x faster while providing low system latency.

The NVIDIA RTX 6000 Ada Generation is the first NVIDIA professional graphics card based on the new Ada architecture. At the heart of the RTX 6000 is the AD102 GPU, which is the most powerful GPU based on the NVIDIA Ada architecture. AD102 has been designed to deliver revolutionary performance for professional and creative workloads.

1 - https://en.wikipedia.org/wiki/Ada_Lovelace

Ada GPU Architecture In-Depth

NVIDIA engineers set clear design goals for every new GPU architecture.

With its groundbreaking RT and Tensor Cores, the Turing architecture laid the foundation for a new era in graphics, which includes ray tracing and AI-based neural graphics. Ampere's revamped SM, enhanced RT and Tensor Cores, and innovative GDDR6 memory subsystem established the bridge between traditional raster-based and ray traced graphics, accelerating both, and providing tremendous performance gains at the highest screen resolutions.

The launch of the new Ada GPU architecture is a breakthrough moment for 3D graphics: the Ada GPU has been designed to provide revolutionary performance for ray tracing and AI-based neural graphics. Performance improvements of 2-4x (up to 4x with the use of DLSS 3) over prior generation Ampere GPUs are possible. The Ada architecture provides a higher level of baseline GPU performance, marking the tipping point where ray tracing and neural graphics become mainstream.

AD102 is the flagship of the Ada GPU lineup, the RTX 6000 Ada Generation is the first professional GPU to utilize the AD102 GPU.

This section will be focused on the AD102 GPU.

Ada AD102 GPU

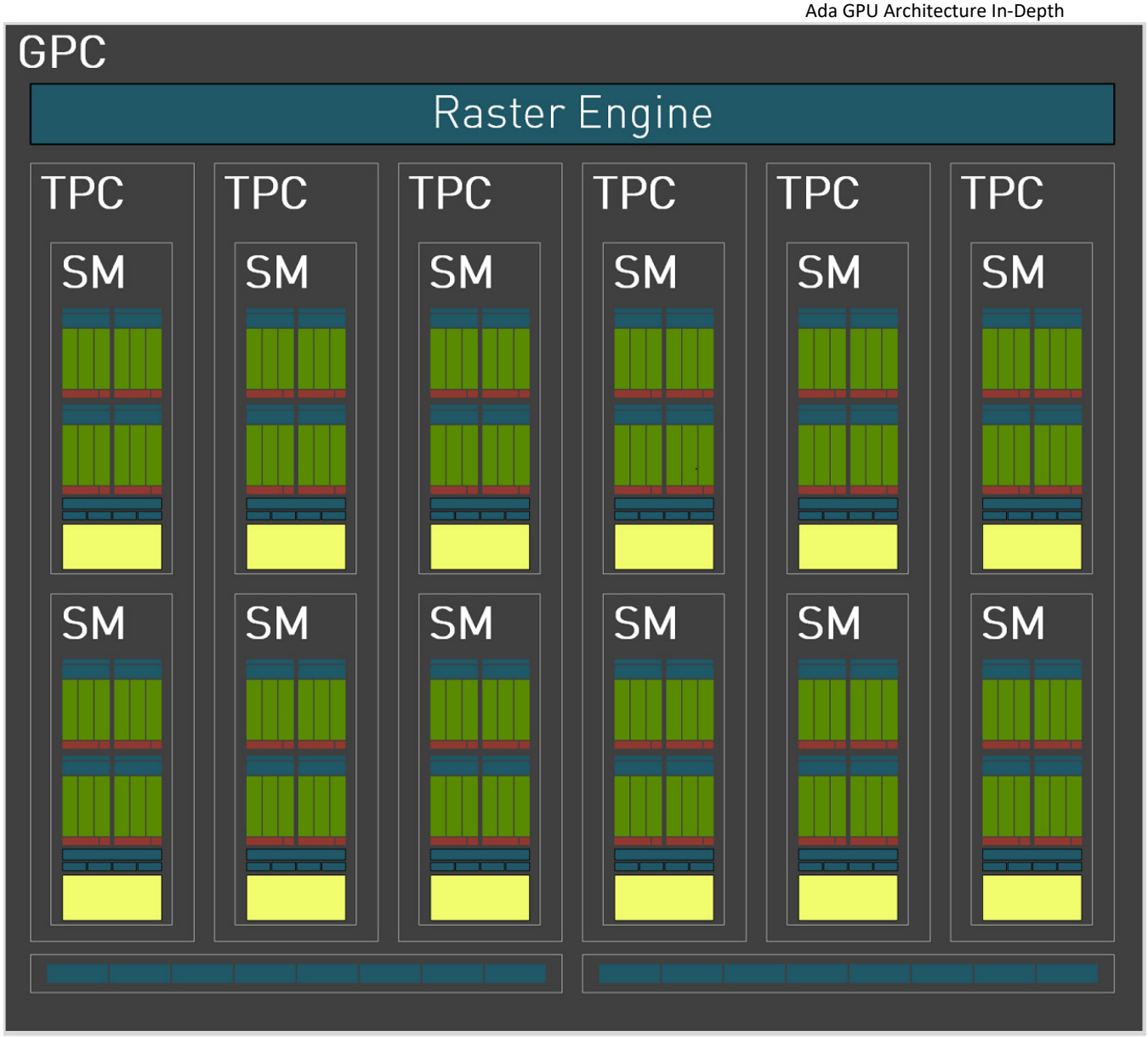
The full AD102 GPU includes 12 Graphics Processing Clusters (GPCs), 72 Texture Processing Clusters (TPCs), 144 Streaming Multiprocessors (SMs), and a 384-bit memory interface with 12 32-bit memory controllers.

Note: The AD102 GPU also includes 288 FP64 Cores (2 per SM) which are not depicted in the above diagram. The FP64 TFLOP rate is 1/64th the TFLOP rate of FP32 operations. The small number of FP64 Cores are included to ensure any programs with FP64 code operate correctly, including FP64 Tensor Core code.

Figure 1. AD102 GPU Full-Chip Block Diagram.

The full AD102 GPU includes:

- 18432 CUDA Cores
- 144 RT Cores
- 576 Tensor Cores
- 576 Texture Units



Ada GPC with Raster Engine, 6 TPCs, 12 SMs, and 16 ROPs (8 per ROP partition).

Figure 2. Ada GPC Block Diagram

The GPC is the dominant high-level hardware block within all AD10x Ada family GPUs, with all of the key graphics processing units residing within a GPC. Each GPC includes a dedicated Raster Engine, two Raster Operations (ROPs) partitions, with each partition containing eight individual ROP units, and six TPCs. Each TPC includes one PolyMorph Engine and two SMs.

Each SM in AD10x GPUs contain 128 CUDA Cores, one Ada Third-Generation RT Core, four Ada Fourth-Generation Tensor Cores, four Texture Units, a 256 KB Register File, and 128 KB of L1/Shared Memory, which can be configured for different memory sizes depending on the needs of the graphics or compute workload.

The RT Core in Turing and Ampere GPUs includes dedicated hardware units for accelerating Bounding Volume Hierarchy (BVH) data structure traversal, and performing the ray-triangle and ray-bounding box intersection testing calculations that are critical for ray tracing. In the Ampere RT Core diagram below, BVH traversal is accelerated by the **Box Intersection Engine** (represented by the group of bounding boxes on the left) and ray-triangle intersection testing is accelerated by the **Triangle Intersection Engine** (triangle on the right). By

providing dedicated resources for these highly important ray tracing functions, work is offloaded from the SM, freeing it up to perform other pixel, vertex, and compute shading tasks. In testing with synthetic benchmarks and applications, the RT Core found in Turing and Ampere GPUs has proven to be the highest performing engine for processing RT workloads to date.

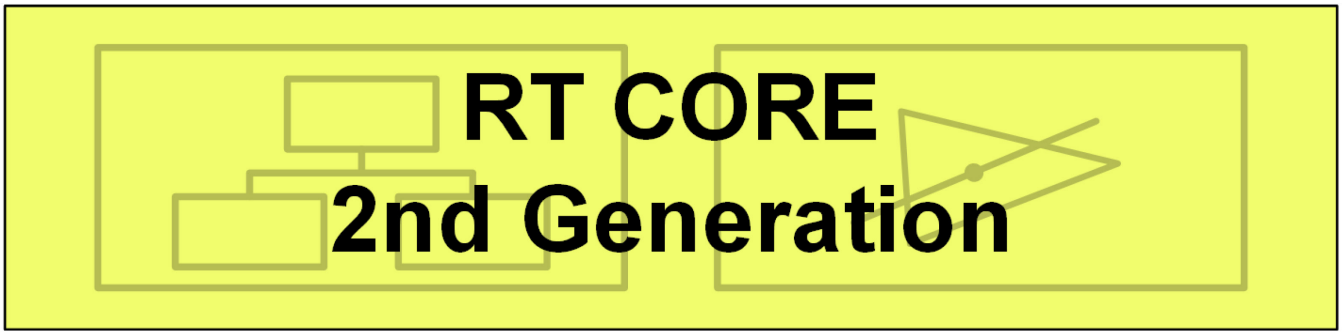


Figure 3. RT Core Second Generation Block Diagram (Ampere architecture)

In addition to these two functions, the Third-Generation RT Core found in Ada GPUs includes dedicated units known as the **Opacity Micromap Engine** and the Displaced Micro-Mesh Engine. The Opacity Micromap Engine evaluates Opacity Micromaps (represented by the triangle with foliage on the bottom left), which are used to accelerate alpha traversal. The **Displaced Micro-Mesh Engine** generates meshes of micro-triangles that are known as Displaced Micro-Meshes (represented by the triangle on the bottom right in the diagram below). Displaced Micro-Meshes allow the Ada RT Core to ray trace geometrically complex objects and environments with significantly less BVH build time and storage costs. Finally, ray-triangle intersection testing is 2x faster in Ada's Third-Generation RT Core compared to the Ampere GPU generation.

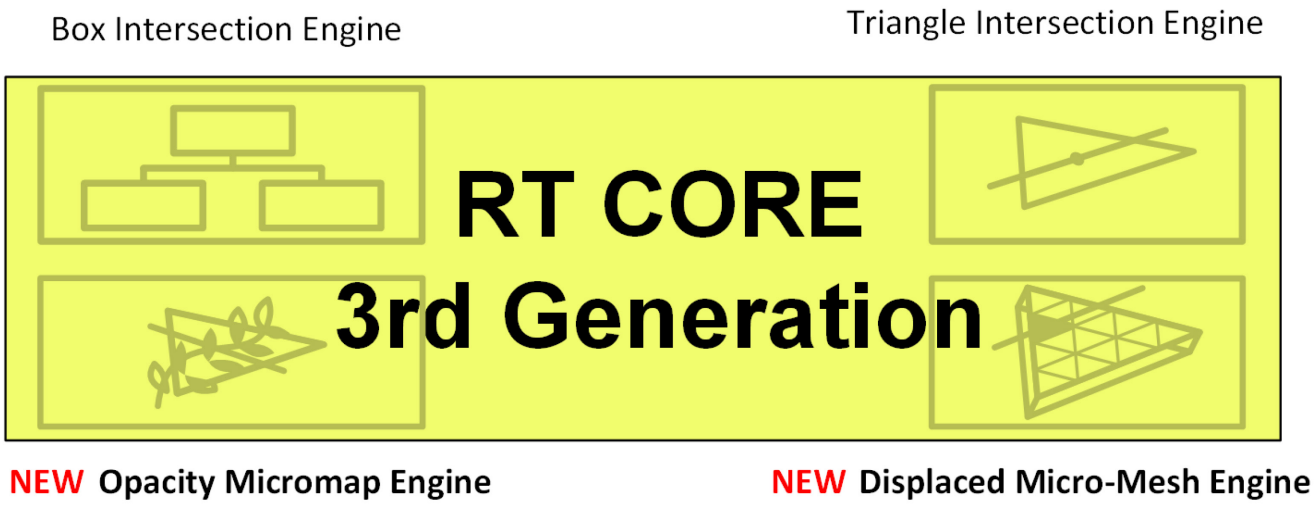


Figure 4. RT Core Third Generation Block Diagram (Ada architecture)

Altogether these enhancements make the Ada Third-Generation RT Core the most powerful RT Core NVIDIA has ever built.

Like prior GPUs, the AD10x SM is divided into four processing blocks (or partitions), with each partition containing a 64 KB register file, an L0 instruction cache, one warp scheduler, one dispatch unit, 16 CUDA Cores that are dedicated for processing FP32 operations (up to 16 FP32 operations per clock), 16 CUDA Cores that can process FP32 or INT32 operations (16 FP32 operations per clock or 16 INT32 operations per clock), one Ada

Fourth-Generation Tensor Core, four Load/Store units, and a Special Function Unit (SFU) which executes transcendental and graphics interpolation instructions.

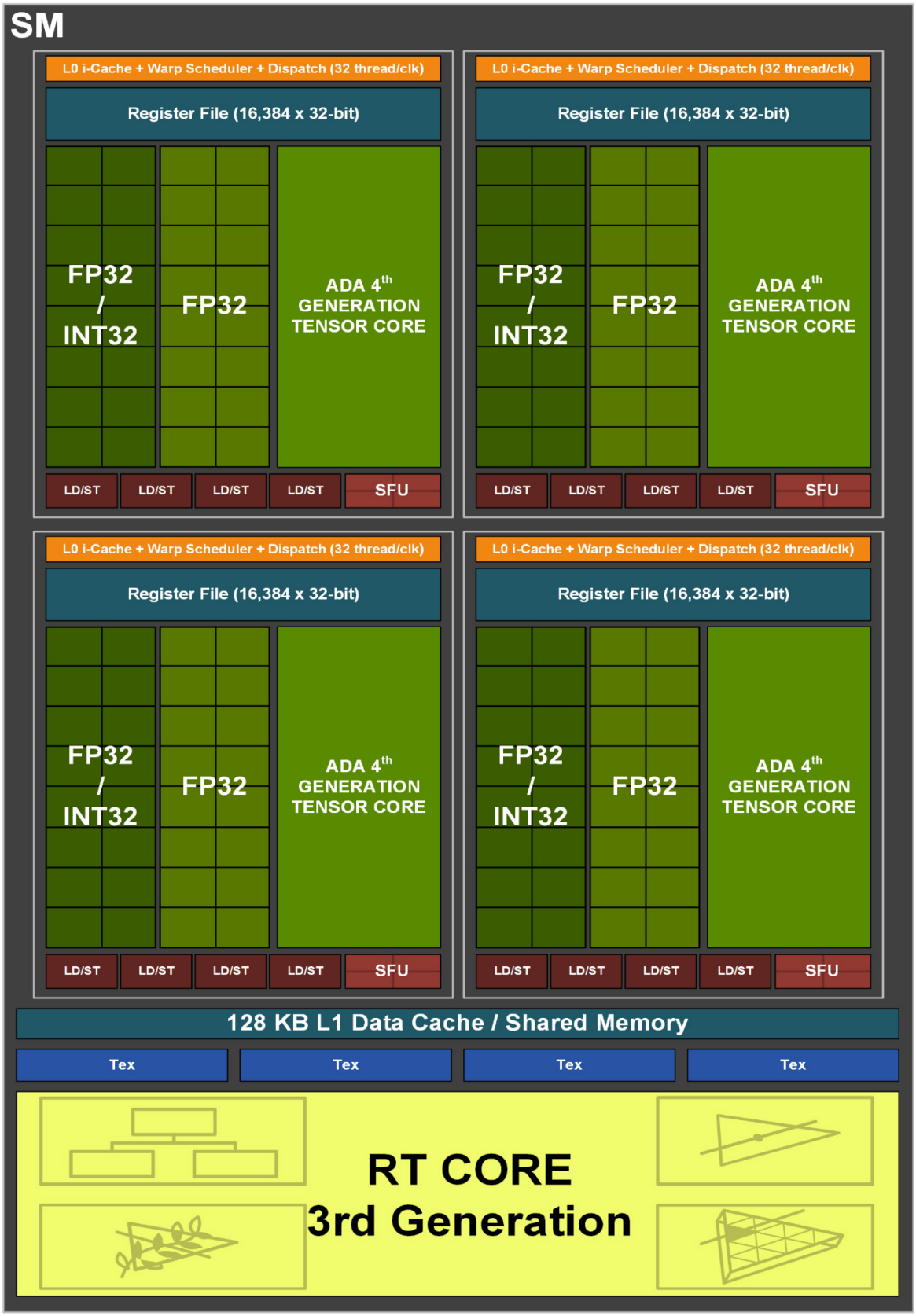


Figure 5. Ada Streaming Multiprocessor (SM)

Memory Subsystem

The Ada SM contains 128 KB of Level 1 cache. This cache features a unified architecture that can be configured to function as an L1 data cache or shared memory depending on the workload. The full AD102 GPU contains 18432 KB of L1 cache (compared to 10752 KB in GA102).

Compared to Ampere, Ada's Level 2 cache has been completely revamped. AD102 has been outfitted with 98304 KB of L2 cache, an improvement of 16x over the 6144 KB that shipped in GA102. All applications will benefit from having such a large pool of fast cache memory available, and complex operations such as ray tracing (particularly path tracing) will yield the greatest benefit.

4N Manufacturing Process and Power Efficiency

Ada GPUs are fabricated on TSMC's 4N manufacturing process. NVIDIA engineers worked closely with TSMC to optimize the process for GPU production. Using the 4N process enabled NVIDIA to integrate dramatically more cores: AD102 contains 70% more CUDA Cores than the prior generation GA102 GPU. In total, the AD102 GPU contains 76.3 billion transistors, making it one of the most complex chips ever made.

Ada also operates at high clock frequencies. NVIDIA optimized the design of the GPU, using high speed transistors in critical paths that could otherwise restrict the rest of the chip. Running at a GPU Boost clock of 2.52 GHz, the RTX 6000 Ada Generation ships with the highest clock frequency of any professional NVIDIA GPU.

At the same time however, RTX 6000 high clock speeds and core count deliver the highest performance per watt. When running at the same power as the RTX A6000, the RTX 6000 delivers over 2x more performance.

Table 1. RTX 6000 Ada Generation vs RTX A6000 / RTX 6000 Specifications

Graphics Card	RTX 6000	RTX A6000	RTX 6000 Ada Generation
CUDA Cores	4608	10752	18176
GPCs	6	7	12
TPCs	36	42	71
SMs	72	84	142
GPU Boost Clock (MHz)	1770	1800	2505
FP32 TFLOPS	16.3	38.7	91.1
Tensor Cores	576 (2nd Gen)	336 (3rd Gen)	568(4th Gen)
Tensor TFLOPS (FP8)	N/A	N/A	728.5/1457[1]
RT Cores	72 (1st Gen)	84 (2nd Gen)	142(3rd Gen)
RT TFLOPS	49.2	75.6	210.6
Texture Units	288	336	568
Texture Fill Rate	444.7	625	1290.2
ROPS	88	112	176
Pixel Fill Rate	510	604.8	1422.8
Memory Size and Type	24576 MB GDDR6	49152 MB GDDR6	49152 MB GDDR6
Memory Clock (Data Rate)	14 Gbps	16 Gbps	20.0 Gbps
Memory Bandwidth	672 GB/sec	768 GB/sec	960 GB/sec

L1 Cache/Shared Memory	96 KB	10752 KB	18176 KB
L2 Cache	6144 KB	6144 KB	98304 KB
TGP	260 W	300W	300W
Transistor Count	18.6 Billion	28.3 Billion	76.3 Billion
Die Size	754 mm^2	628.4 mm^2	608.4 mm^2
Manufacturing Process	TSMC 12 nm FFN (FinFET NVIDIA)	Samsung 8 nm 8N NVIDIA Custom Process	TSMC 4N NVIDIA Custom Process

1- Using Sparsity feature

For the full list of RTX 6000 Ada Generation specifications, please see Appendix A at the back of this document.

Ray Tracing

The advent of real-time ray tracing has elevated the visual quality computer graphics by delivering realistic lighting effects, physically accurate shadows, and better reflections, creating a final rendered image that approaches photorealism; all while running in real-time. Working in with real time ray-tracing in applications allows professional users to better visualize how designs will compare to their real world counterparts, and allow content creators to more closely match elements in the creation phase to the final output.

In the pursuit of producing more realistic graphics, there is an exploding demand for deeply detailed environments. Vast libraries of objects are available as ingredients. Developers draw on the talents of their artists to craft compellingly intricate custom models.

Generally, developers mine geometric content from two major veins: scans of physical objects and artistically and/or algorithmically synthesized models. The former technique, photogrammetry, captures every minute geometric detail as well as material properties critical for accurate shading. Professional visualization artists also create fantastically detailed models. The result is objects often composed of millions of triangles, and environments composed of billions.

When working with environments like these, developers face two major challenges: storage and rendering performance. In a given frame, level of detail (LOD) techniques can mitigate some of the performance impact of scene complexity, but it's limited, since there is little control over where the camera/player may wander, and what scattering rays may hit (e.g., behind the camera).

NVIDIA engineers have developed three new features in the Ada RT Core to enable high-performance ray tracing of highly complex geometry:

- First, Ada's Third-Generation RT Core features **2x Faster Ray-Triangle Intersection Throughput** relative to Ampere; this enables developers to add more detail into their virtual worlds.
- Second, Ada's RT Core has **2x Faster Alpha Traversal;** the RT Core features a new **Opacity Micromap Engine** to directly alpha-test geometry and significantly reduce shader-based alpha computations. With this new functionality, developers can very compactly describe irregularly shaped or translucent objects, like ferns or fences, and directly and more efficiently ray trace them with the Ada RT Core.
- Third, the new Ada RT Core supports **10x Faster BVH Build** in **20X Less BVH Space** when using its new **Displaced Micro-Mesh Engine** to generate micro-triangles from micro-meshes on-demand. The micro-mesh is a new primitive that represents a structured mesh of micro-triangles that the Ada RT Core processes natively, saving the storage and processing compared to what is normally required when describing complex geometries using only basic triangles.

Taken together these three advances incorporated into the Ada RT Core enable order-of-magnitude increases in richness without commensurate increases in processing time or memory consumption.

As we continue to approach photorealistic rendering with real-time ray tracing, increasing the accuracy with which we model the movement of light through extremely detailed, diverse environments means the raw processing workload becomes less and less coherent. Secondary rays used for reflections, indirect lighting, and translucency effects, for example, tend to shoot in different directions and hit different materials, resulting in secondary hit shaders being less ordered and less efficient.

Left unaddressed, a loss in execution regularity can lead to inefficient use of the GPU's processing units, the SMs.

To address this issue, the Ada architecture introduces **Shader Execution Reordering.** This feature intelligently schedules shading work on-the-fly, so that complex materials like brushed metal can be processed more effectively.

*For more information on the fundamentals of ray tracing and the GPU RT Core, please refer to the **NVIDIA Turing GPU Architecture Whitepaper** and the **NVIDIA Ampere GA102 GPU Whitepaper**.*

2x Faster Ray-Triangle Intersection Testing

Ray-triangle intersection testing is a computationally expensive operation that is commonly performed when rendering a ray-traced scene. Recognizing the importance of this function, with each new RTX GPU NVIDIA engineers have strived to improve intersection testing performance and efficiency. The Third-Generation RT Core in the Ada architecture provides double the throughput for ray-triangle intersection testing over Ampere (and 4x faster than the first-generation RT Core used in Turing GPUs).

2x Faster Alpha Traversal Performance with Opacity Micromap Engine

Developers frequently use a texture's alpha channel to economically cut out complex shapes or more generally to represent translucency. A leaf might be described using a couple of triangles, employing a texture's alpha channel to economically capture the complex shape. A flame's complex shape and translucency can also be approximated by alpha.

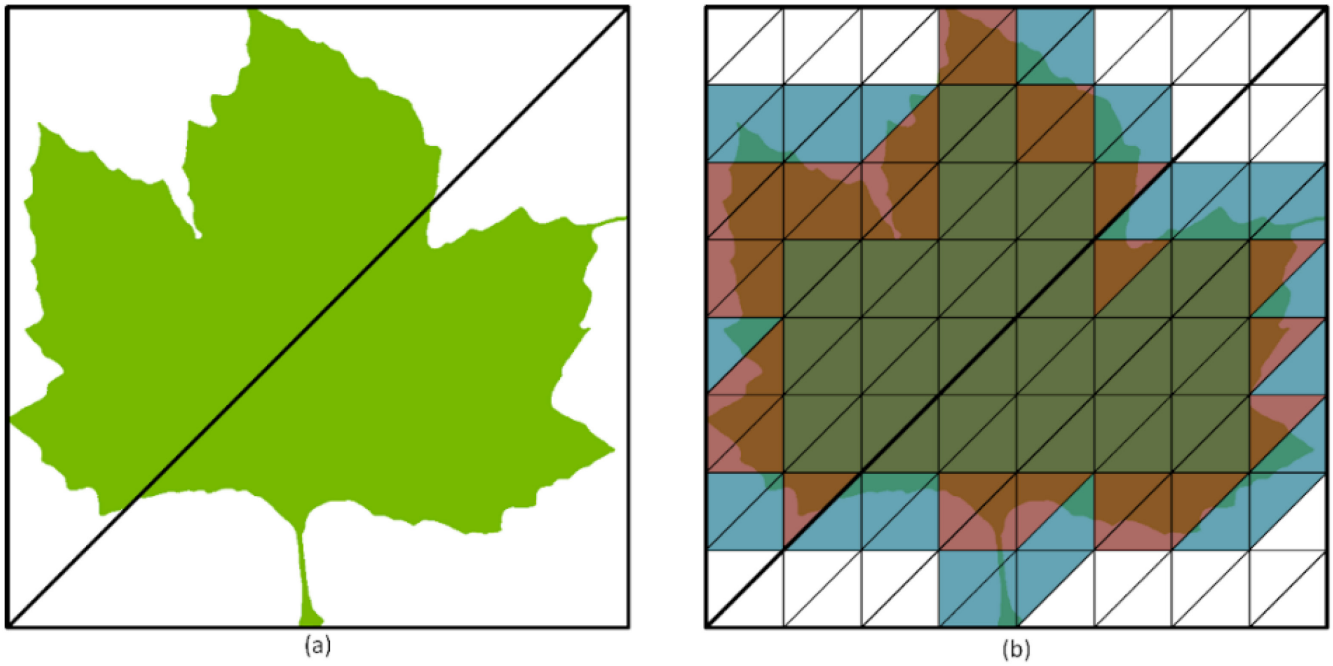
Complex shapes such as a leaf or a flame often use a texel's alpha channel to represent levels of transparency and opaqueness.

Figure 6. Complex Shapes Use Texel's Alpha Channel

Prior to Ada's RT Core, a developer could incorporate these kinds of content into a ray traced scene by tagging them as not opaque. When a leaf is hit by a ray, a shader is invoked to determine how to treat the intersection, even if the ray is simply characterized as a hit or a miss. This incurs noticeable cost. Specifically, when a warp of rays is cast towards non-opaque objects, individual ray queries may require multiple shader invocations to resolve, while other rays terminate immediately. The result is lingering live threads and commensurate inefficiency.

To efficiently handle these kinds of content, NVIDIA engineers have added an Opacity Micromap Engine to Ada's RT Core. An opacity micromap is a *virtual* mesh of micro-triangles, each with an opacity state that the RT Core uses to directly resolve ray intersections with non-opaque triangles. Specifically, the barycentric coordinates of an intersection are used to address the corresponding micro-triangle's opacity state. The opacity state may be opaque, transparent, or unknown. If opaque, then a hit is recorded and returned. If transparent, the intersection is ignored and the search for an intersection continues. If unknown, then control is returned to the SM, invoking a shader ("anyhit") to programmatically resolve the intersection.

The new Opacity Micromap Engine evaluates the opacity mask, which is a regular triangular mesh defined using the barycentric coordinate system used for reporting ray/triangle intersections. These meshes may be sized from one to *sixteen million* micro-triangles, with one or two bits associated with each micro-triangle. As a simple illustrative example, consider a detailed maple leaf described using two triangles and an alpha texture (see sub-Figure (a) in Figure 7).

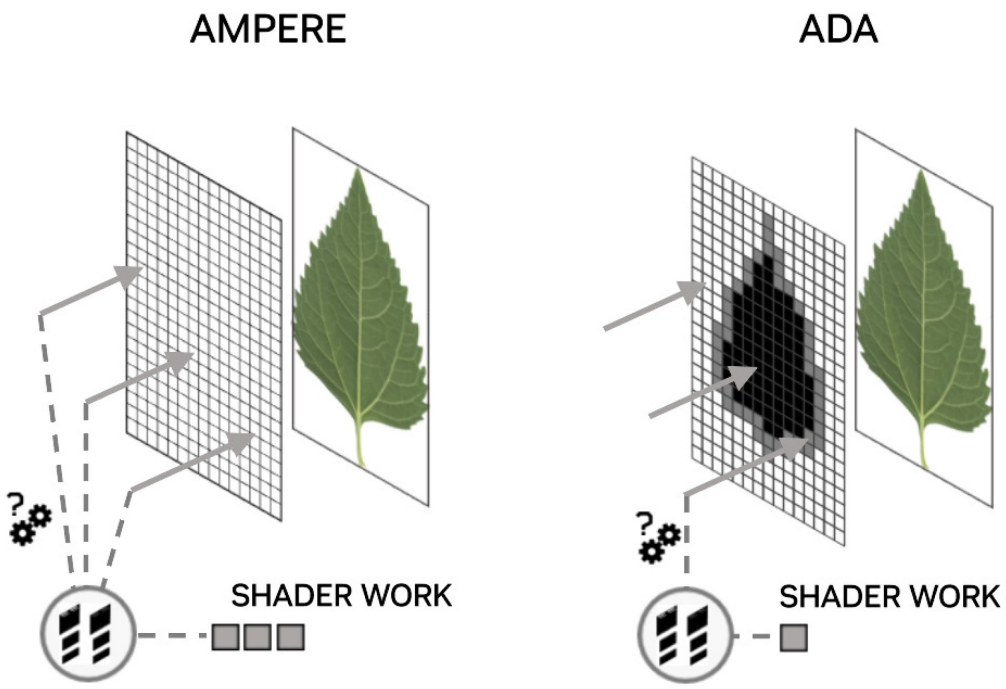


Opacity mask is applied to maple leaf which is composed of 2 triangles. The opacity engine evaluates the leaf and determines which sections are opaque, transparent, or unknown (in which case it must be sent back to the SM).

Figure 7. Opacity Mask Applied to Leaf

Sub-figure in Figure 7 above shows a pair of opacity masks, one per triangle. In the figure, transparent regions are white, they contain no leaf whatsoever. Dark green micro-triangles correspond to opaque areas of the leaf, lastly red and blue correspond to regions of mixed opacity (*unknown*). In the example above, the Opacity Micromap Engine tags 30 of the micro-triangles as transparent, 41 as opaque, and 57 as unknown. This means that over half of the leaf is fully characterized, and that more than half of the rays intersecting these triangles either miss the leaf, or unambiguously intersect the leaf's interior. The result is that the Ada RT Core can fully characterize these rays without invoking any shader code, while preserving the full resolution and fidelity of the original alpha texture. When an *unknown* state is encountered, control is returned to a shader for resolution.

Figure 8 below shows an example of how alpha-tested content would be handled in prior GPUs.



Ada's Opacity Micromap Engine with opacity mask reduces shader work compared to Ampere

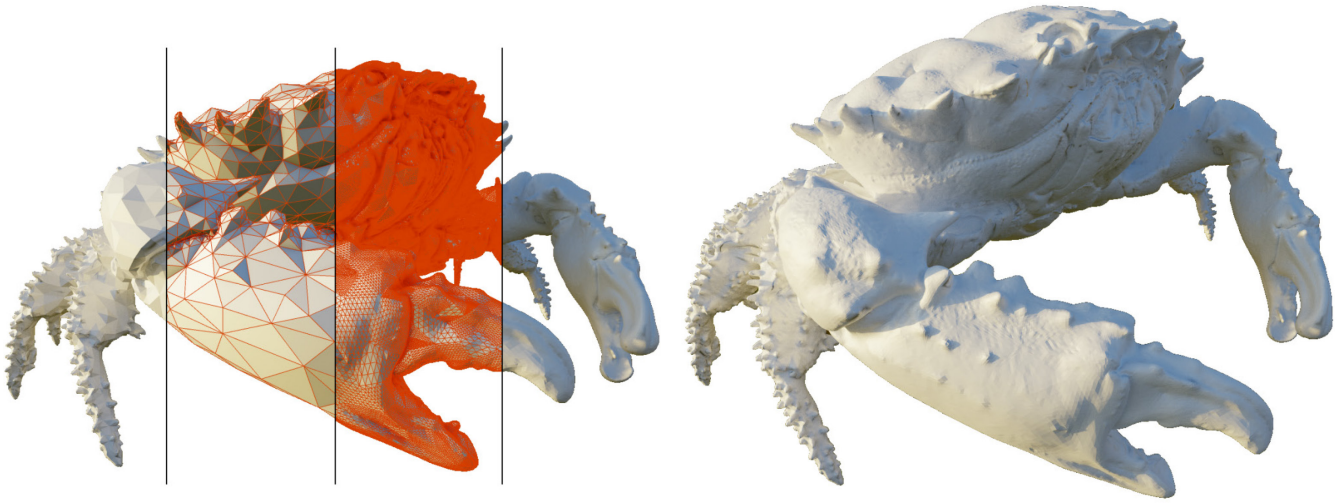
Figure 8. Ada Opacity Micromap Engine Compared to Ampere Arch

With the addition of the Opacity Micromap Engine to Ada's RT Core we have measured a doubling of scene traversal performance in applications with alpha-tested geometry. Performance gains heavily depend on usage, typically shadow rays cast against alpha-tested geometry see the largest gains. Ada's opacity mask support can significantly increase the amount and fidelity of detailed geometry within scenes, raising the realism bar.

10x Faster BVH Build in 20X Less BVH Space with Ada's Displaced Micro-Mesh Engine

Geometric complexity continues to rise with every new generation. Ray tracing performance scales attractively with increases in scene complexity. When we ray trace complex environments, tracing costs increase slowly, a one-hundred-fold increase in geometry might only double tracing time. However, creating the data structure (BVH) that makes that small increase in time possible requires roughly linear time and memory; 100x more geometry could mean 100x more BVH build time, and 100x more memory. Ada's Third-Generation RT Core with Displaced Micro-Meshes (DMM) helps significantly with both of the challenges of high geometric complexity - BVH build performance and memory/storage footprint. Asset storage and transmission costs are reduced as well.

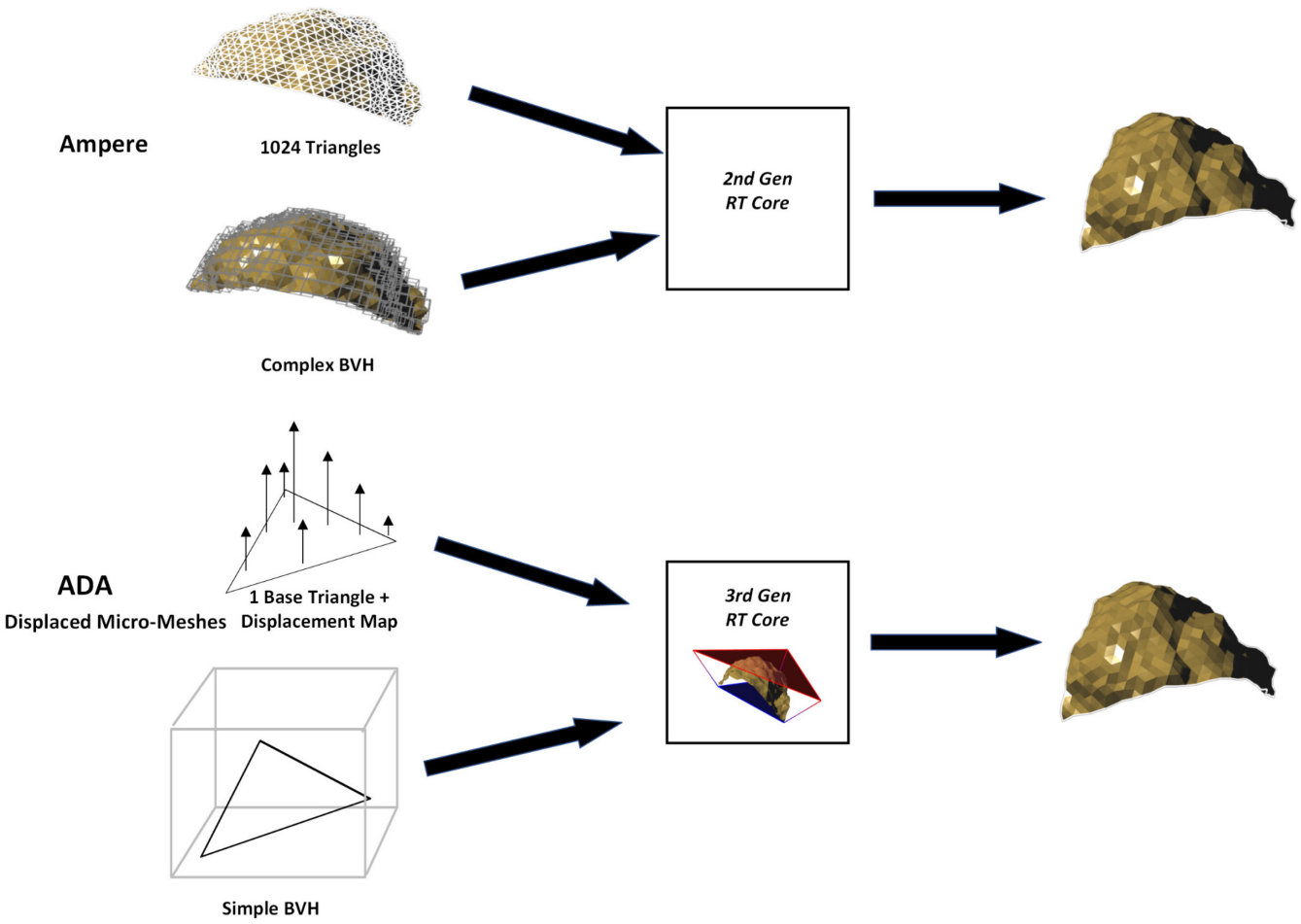
We developed Ada's displaced micro-mesh as a structured representation of geometry that exploits spatial coherence for compactness (compression) and exploits its structure for efficient rendering with an intrinsic level of detail (LOD) and light-weight animation/deformation. When ray tracing we use the displaced micro-mesh structure to avoid a large increase in BVH construction costs (time and space) while preserving efficient BVH traversal. When rasterizing we use the intrinsic micro-mesh LOD to rasterize right-sized primitives with Mesh Shaders or Compute Shaders.



Reef crab broken into base triangles represented in red (on the far left), with higher geometric detail (also in red) represented by the micro-meshes to the right. The final result is represented on the far right.

Figure 9. Displacement Micro-Mesh - Base Mesh and Micro-Meshes

The displaced micro-mesh is a new geometric primitive that was co-designed with the Micro-Mesh Engine in Ada's Third-Generation RT Core. Each micro-mesh is defined by a base triangle and a displacement map. The Micro-Mesh Engine on-demand generates micro-triangles from this definition in order to resolve ray micro-mesh intersections down to the individual micro-triangle. We use a watertight base-mesh of micro-meshes to represent highly detailed objects. We compress displacement magnitude into maps, one map per base triangle. Micro-triangle vertices are on a power-of-two, barycentric grid, and their barycentric coordinates (uv) are used to directly address micro-vertex displacements.



Ada's Third-Generation RT Core with Displaced Micro-Mesh Engine uses a simple BVH, 1 base triangle + displacement map to create a highly detailed geometric mesh with fewer required resources (both triangles and BVH structure) than Ampere's 2nd Generation RT Core.

Figure 10. DMM Simplified BVH, Base Triangle, and Displacement Map

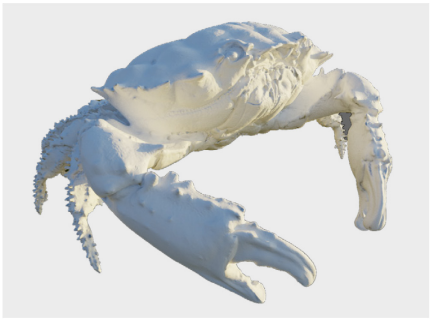
Displaced Micro-Mesh Measured Gains

Jewel Box – 11:1

153K micro-meshes,
11M micro-triangles,
13 bits/micro-triangle
BVH build:
8.5x faster, 6.5x smaller

Ewer – 28:1

175K micro-meshes,
57M micro-triangles,
5 bits/micro-triangle
BVH build:
>15x faster, 20x smaller



Reef Crab – 14:1

17K micro-meshes,
1.6M micro-triangles,
10 bits/micro-triangle
BVH build:
7.6x faster, 8.1x smaller

Displaced Micro-Meshes allow Ada's RT Core to generate complex geometry with faster build time and reduced storage requirements.

Figure 11. DMMs Reduce BVH Build Time and Storage Requirements

Targeted launch partners for displaced micro-meshes include Adobe and Simplygon (part of Xbox Game Studios).

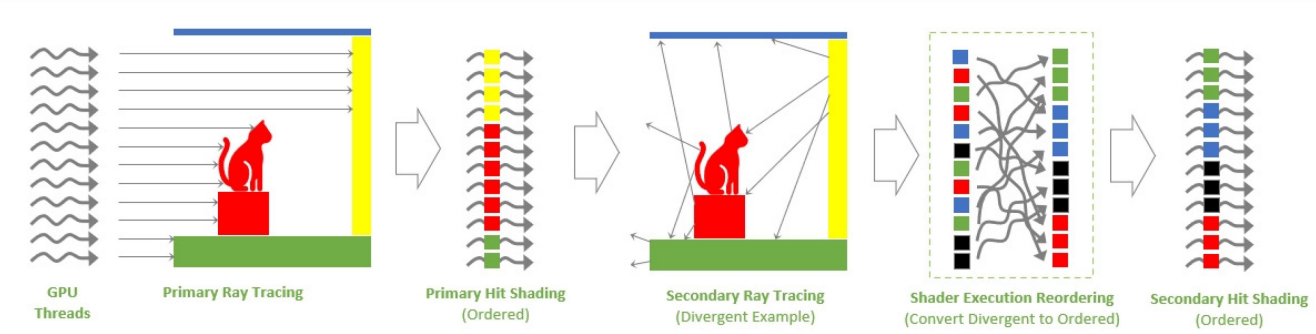
Shader Execution Reordering (SER)

Raw RT Core horsepower is not enough to ensure high frame rates with ray-traced content, as RT workloads can be bottlenecked by a number of factors. In particular, divergent RT shaders are increasingly becoming a limiter, for example when executing multi-bounce, stochastic path tracing algorithms, or when evaluating complex materials.

Divergence takes two forms: execution divergence, where different threads execute different shaders or code paths within a shader, and data divergence, where threads access memory resources that are hard to coalesce or cache. Both types of divergence occur naturally in many ray tracing scenarios. This leaves performance on the table, because GPUs operate most efficiently when the processed work is uniform.

Ada includes a new technology designed to enhance the efficiency of RT shader execution by tackling the divergence problem. Shader Execution Reordering (SER) is a new scheduling system that reorders shading work on-the-fly for better execution and data locality. Years of research and development have been invested in SER in order to minimize overheads and maximize its effectiveness. The Ada hardware architecture was designed with SER in mind and includes optimizations to the SM and memory system specifically targeted at efficient thread reordering.

SER is fully controlled by the application through a small API, allowing developers to easily apply reordering where their workload benefits most. The API additionally introduces new flexibility around the invocation of ray tracing shaders to the programming model, enabling more streamlined ways to structure renderer implementations while taking advantage of reordering. Furthermore, we are adding new features to the NSight Graphics shader profiler to help developers optimize their applications for SER. Developers can initially use NVIDIA-specific NVAPI extensions to implement SER in their code. We are also working with Microsoft and others to extend the standard graphics APIs with SER.



Shader Execution Reordering. Advanced lighting techniques such as path tracing cause shader divergence as secondary rays bounce off objects in the scene (denoted by various colors). In these scenarios, SER reorders shading work to improve efficiency.

Figure 12. Shader Execution Reordering Pipeline

The above diagram shows a simple ray tracing example. Starting at the top left, a number of GPU threads are shooting primary rays into a scene. Primary rays hitting the same objects can be assumed to be running the same shader program on each of the threads that hit those objects, and they are well-ordered, so the primary hit shading has high execution efficiency and data locality.

Secondary rays are generated at each primary ray hit point in the middle scene. Starting at the primary hit surfaces they shoot off in different directions, hitting different objects. Secondary hit shading tends to be less ordered and less efficient when executing on the GPU, because different shader programs are running on the different threads, and often must serialize execution. Examples of secondary rays that can benefit from SER include those used for path tracing, reflections, indirect lighting, and translucency effects.

Shader Execution Reordering adds a new stage in the ray tracing pipeline which reorders and groups the secondary hit shading to have better execution locality, thus much higher overall ray-traced shading efficiency.

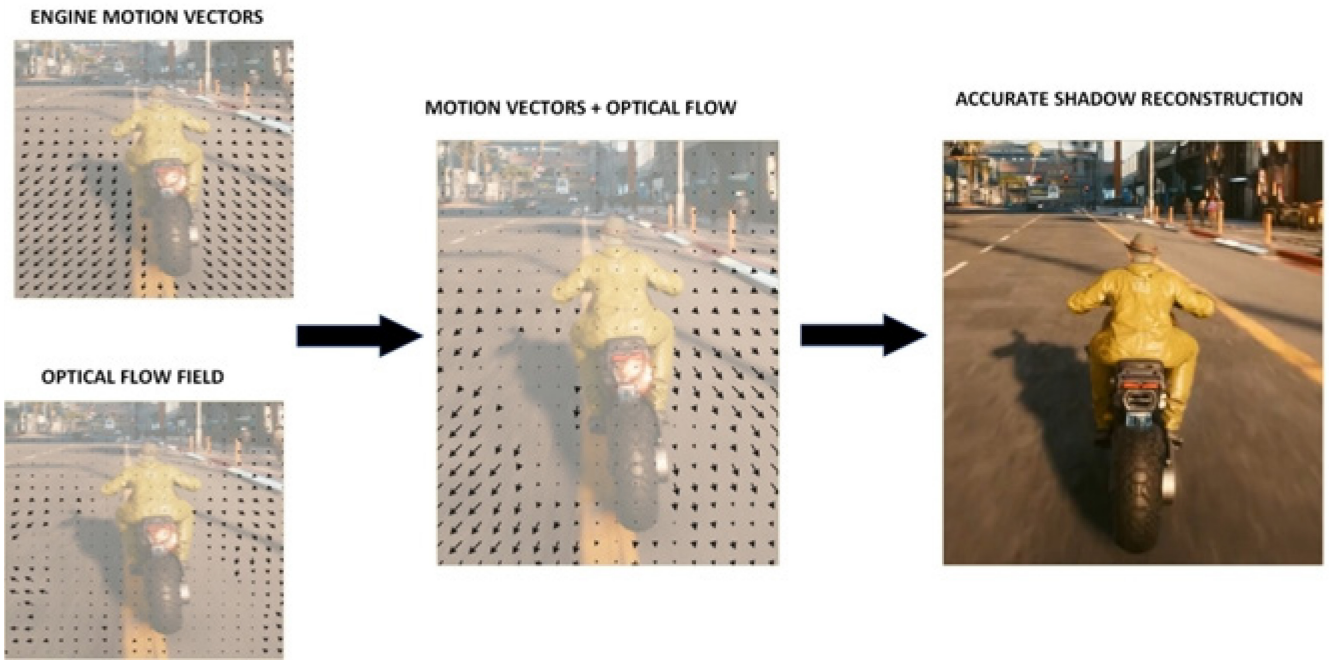
SER can often provide up to 2X performance improvement for RT shaders in cases with a high level of divergence (such as path tracing).

DLSS 3 and Optical Flow Acceleration

Over the past four years, the NVIDIA Applied Deep Learning Research team has been developing a frame generation technique that combines optical flow estimation with DLSS to improve the gaming experience. Inserting accurate synthesized frames between existing frames improves frame rate and provides a smoother gaming experience.

Optical flow estimation is commonly used in computer vision applications to measure the direction and magnitude of the apparent motion of pixels between consecutively rendered graphics frames or video frames. In the 3D graphics and video fields, typical use cases have included reducing latency in augmented and virtual reality, improving smoothness of video playback, enhancing video compression efficiency, and enabling video camera stabilization. Deep learning uses often include automotive and robotic navigation, and video analysis and understanding.

Optical flow is superficially similar to the motion estimation component of video encoding, but with much more challenging requirements for accuracy and consistency. As a result, different algorithms are used. Starting with the Ampere GPU architecture, NVIDIA's GPUs have had support for a standalone optical flow engine (OFA) that uses state of the art algorithms to ensure high quality results. Ada's OFA unit delivers 300 TeraOPS (TOPS) of optical flow work (over 2x faster than the Ampere generation OFA) and provides critical information to the DLSS 3 network.



To generate more accurate frames without distracting artifacts, DLSS 3 combines 3D engine motion vectors with the OFA Engine's optical flow field.

Figure 13. DLSS 3 Motion Vectors + Optical Flow = Accurate Motion Estimation

The Ada OFA unit and new motion vector analysis algorithms are fundamental components that enable accurate and performant frame generation capability within the new DLSS 3 technology framework. This new DL-based frame generation method improves frame rates by an additional 2x over DLSS 2. When DLSS 3 is combined with the new RT Core and other Ada architecture enhancements, Ada is up to 4x faster than prior GPUs.

DLSS 3 can also improve performance in cases where the GPU is bottlenecked by the CPU. This limits the performance benefits that traditional super resolution technologies can offer. In this case however, DLSS 3's ability to generate frames still provides up to a 2x performance improvement. For more information on OFA and DLSS 3, please read the ***NVIDIA Ada Science Whitepaper***.

Ada Fourth-Generation Tensor Core

Tensor Cores are specialized high performance compute cores that are tailored for the matrix multiply and accumulate math operations that are used in AI and HPC applications. Tensor Cores provide groundbreaking performance for the matrix computations that are critical for deep learning neural network training and inference functions that occur at the edge.

Compared to Ampere, Ada delivers more than double the FP16, BF16, TF32, INT8, and INT4 Tensor TFLOPS, and also includes the Hopper FP8 Transformer Engine, delivering over 1.45 PetaFLOPS of tensor processing in the RTX 6000 Ada Generation.

NVIDIA Broadcast/Video

NVIDIA innovations have democratized streaming so that more people can easily stream on their PC without the cost and complexity challenges that have traditionally been an issue for users in the past. The introduction of NVIDIA's NVENC encoder and optimizations for OBS (Open Broadcaster Software) eliminated the need for a dedicated PC for video capture, allowing users to play and stream from the same PC with good stream quality and high fps. Finally, NVIDIA's Broadcast suite, powered by AI, provides tools for noise and echo removal, virtual backgrounds, video noise removal, and automatic camera tracking so that anyone can start and work their way up to their dream streaming setup without the obligation to purchase professional microphones, cameras, and need a dedicated recording studio to stream.

Ada GPUs take streaming and video content to the next level, incorporating support for AV1 video encoding in the Ada **eighth generation dedicated hardware encoder** (known as NVENC). Prior generation Ampere GPUs supported AV1 decoding, but not encoding. Ada's AV1 encoder is 40% more efficient than the H.264 encoder used in previous generation Turing GPUs. AV1 will enable users who are streaming at 1080p today to increase their stream resolution to 1440p while running at the same bitrate and quality, or for users with 1080p displays, streams will look similar to 1440p, providing better quality.

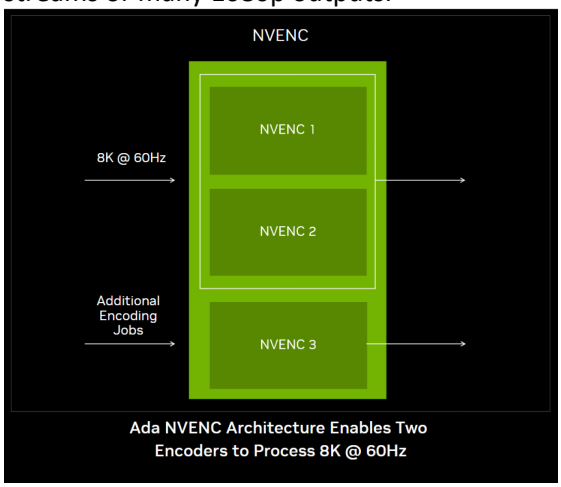
NVIDIA collaborated with OBS Studio to add AV1 — on top of the recently released HEVC and HDR support — within an upcoming software release, expected later this year. OBS is also optimizing encoding pipelines to reduce overhead by 35% for all NVIDIA GPUs. The new release will additionally feature updated NVIDIA Broadcast effects, including noise and room echo removal, as well as improvements to virtual background.

We've also worked with Discord to enable end-to-end livestreams with AV1. In an update releasing later this year, Discord will enable its users to use AV1 to dramatically improve screen sharing, to improve online collaborative experiences.

To further aid encoding performance, Ada professional GPUs with 12 GB of memory or more are equipped with **three NVENC encoders**. This enables video encoding at 8K/60 for professional video editing or four 4K/60. (Video streaming services can also take advantage of this to enable more simultaneous sessions, for instance.) Blackmagic Design's DaVinci Resolve, the popular Voukoder plugin for Adobe Premiere Pro, and Jianying — the top video editing app in China — are all enabling AV1 support, as well as a multi encoder support through encode presets. NVIDIA is also working with the popular video-effects app Notch to enable AV1, as well as Topaz to enable support for AV1 and the dual encoders.

For professional broadcast solutions, the multiple NVENC encode engines combined with AV1 encoding allows broadcasters more flexible streaming options. For example, a two NVENC encoders can work together to encode

and output a single 8K@60 stream while the third NVENC encoder can simultaneously be encoding multiple 4K streams or many 1080p outputs.



In addition to NVENC, Ada GPUs also include the **fifth-generation hardware decoder** that was first launched with Ampere (known as NVDEC). NVDEC supports hardware-accelerated video decoding of MPEG-2, VC-1, H.264 (AVCHD), H.265 (HEVC), VP8, VP9, and the AV1 video formats. 8K/60 decoding is also fully supported. In the future, NVIDIA is also working to enable high quality video production using AI. For more information on this topic, please read the ***NVIDIA Ada Science Whitepaper***.

Conclusion

With frame rates that are up to 4x faster than the previous generation, the Ada Lovelace GPU architecture provides performance that is beyond fast, delivering NVIDIA's greatest generational upgrade ever. Ada's record-breaking performance is made possible by a number of engineering innovations.

NVIDIA engineers worked closely with TSMC to create the 4N manufacturing process that is tailored for NVIDIA GPUs. The smaller process allows more processing units and memories to be integrated into the chip. NVIDIA's AD102 GPU contains 18,432 CUDA Cores (70% more CUDA Cores than Ampere), 18 MB of L1 cache, 96 MB of L2 cache (16x more than Ampere), and a large 36 MB register file. The entire GPU contains over 76 billion transistors, making it second only to NVIDIA's H100 in terms of GPU complexity. Even though the RTX 6000 Ada Generation runs at a Boost Clock frequency of 2.5 GHz – 705 MHz higher than the previous professional flagship RTX A6000 – it consumes the same TGP of 300W. Ultimately Ada delivers 2x higher power efficiency compared to prior generation Ampere. It is truly a marvel of engineering.

As massive as Ada's core counts, memory, and clocks are, the Ada GPU is about more than just those figures. The Ada SM has been significantly enhanced, especially for ray tracing workloads. Ada's Third-Generation RT Core offers 2x faster ray-triangle intersection throughput over prior generation Ampere GPUs (and 4x faster than Turing). Triangle intersection testing is a computationally expensive operation that is very commonly performed when rendering a ray-traced scene, so providing a 2x improvement is very significant.

The Ada RT Core also includes two new hardware units. The first, an Opacity Micromap Engine, speeds up alpha traversal by 2x. With this new capability, developers can very quickly assign opacity values to irregularly shaped objects (like ferns and fences) or translucent items (like flames or smoke) allowing the Ada RT Core to directly alpha test this geometry instead of relying on the GPU's SM.

The second new hardware unit that has been incorporated into the Ada RT Core is the Displaced Micro-Mesh Engine. The new Micro-Mesh Engine has been designed to reduce the BVH build time and storage requirements that are traditionally required when dealing with complex objects with high levels of geometric detail. With this new feature, a new displaced micro-mesh primitive has been developed for ray tracing. The Micro-Mesh Engine evaluates the micro-meshes, and when additional geometric detail is needed, the Micro-Mesh Engine can dynamically generate additional micro-triangles as needed. Compared to traditionally rendering these complex objects, the Micro-Mesh Engine reduces BVH build time by a factor of 10x, while reducing BVH storage requirements by a factor of 20x.

In addition, Ada introduces the Shader Execution Reordering (SER) scheduling system. Shader Execution Reordering organizes and reorders workloads on the fly so they can be processed by the SM and RT Core more efficiently. Shader Execution Reordering is as big of an innovation for GPUs as out-of-order execution was for CPUs back in the 1990s, offering 2-3x speedups for some RT workloads.

When combined, the improvements to SM throughput, higher clocks and core counts, Ada's Third-Generation RT Core, and new features such as Shader Execution Reordering all provide the Ada GPU with a performance uplift of up to 2x over Ampere. So how did we ultimately get to a generational performance uplift of up to 4x? The remainder comes from the Ada GPU's new Optical Flow Accelerator and DLSS 3.

NVIDIA's DLSS technology pioneered the concept of AI-based neural graphics. To date, 216 titles take advantage of DLSS, and the list continues to grow. Ada's new DLSS 3 technology builds on DLSS 2 Super Resolution, which internally renders using lower resolution pixels and uses AI algorithms to produce beautiful, sharp higher resolution images to dramatically improve performance compared to traditional raster-based graphics rendering.

DLSS 3 harnesses the new Optical Flow Accelerator found in Ada GPUs to calculate the motion flow of every pixel in a given frame. This data is combined with the traditional motion vectors and fed into the AI network, which then generates entire frames rather than just pixels.

DLSS 3 can also improve performance in CPU-bound cases that occur when the GPU is bottlenecked by the CPU and is therefore unable to generate higher frame rates. Because DLSS 3 is able to generate frames independent of the CPU, Ada GPUs with DLSS 3 are still capable of improving performance in these cases.

With the introduction of DLSS 3, neural graphics are taken to an entirely new level. DLSS 3 is easy for developers to integrate and is one of the most compelling features that NVIDIA has introduced for a new graphics architecture.

Ada GPUs include a new Fourth-Generation Tensor Core. The RTX 6000 offers double the throughput for existing FP16, BF16, TF32, and INT8 formats, and its Fourth-Generation Tensor Core introduces support for a new FP8 tensor format. Compared to FP16, FP8 halves the data storage requirements and doubles throughput. With the new FP8 format, the RTX 6000 delivers 1.4 PetaFLOPS of performance for AI inference workloads.

All Ada GPUs ship with NVIDIA's 8th Generation NVENC encoder, which adds support for AV1 encoding. AV1 is 40% more efficient than the prior H.264 encoder that was commonly used previously. NVIDIA is working closely with our partners in the professional video space to integrate AV1 support.

Additionally, the RTX 6000 Ada Generation supports three NVENC encoders, providing the ability to support simultaneous 8K/60 video encoding, and 4K or 1080p encoding on separate NVENC encoders. For video editing. NVIDIA has worked with all of the top software makers to integrate AV1 support, with DaVinci Resolve, Voukoder (a plugin for Adobe Premiere Pro), and Jianying (China's most popular video editing app).

The NVIDIA Ada Lovelace architecture delivers a quantum leap in GPU performance and capabilities, giving RTX 6000 users the power to accelerate advanced professional workflows.

Appendix A - RTX 6000 Ada Generation Full Specifications

Table 2. RTX 6000 Ada Generation vs RTX A6000 vs RTX 6000 Full Specifications

Graphics Card	RTX 6000	RTX A6000	RTX 6000 Ada Generation
GPU Codename	TU102	GA102	AD102
GPU Architecture	NVIDIA Turing	NVIDIA Ampere	NVIDIA Ada Lovelace
GPCs	6	7	12
TPCs	36	42	71
SMs	72	84	142
CUDA Cores / SM	64	128	128
CUDA Cores / GPU	4608	10752	18176
Tensor Cores / SM	8 (2nd Gen)	4 (3rd Gen)	(4th Gen)
Tensor Cores / GPU	576 (2nd Gen)	336 (3rd Gen)	568(4th Gen)
OFA TOPS[3]	--	131	319
RT Cores	72 (1st Gen)	84 (2nd Gen)	142(3rd Gen)
GPU Boost Clock (MHz)	1770	1800	2505
Peak FP32 TFLOPS (non-Tensor)[1]	16.3	38.7	91.1
Peak FP16 TFLOPS (non-Tensor)[1]	32.6	38.7	91.1

Peak BF16 TFLOPS (non-Tensor)[1]	--	38.7	91.1
Peak INT32 TOPS (non-Tensor)[1],[3]	16.3	19.4	44.5
RT TFLOPS	49.2	75.6	210.6
Peak FP8 Tensor TFLOPS with FP16 Accumulate[1]	N/A	N/A	728.5/1457[2]
Peak FP8 Tensor TFLOPS with FP32 Accumulate[1]	N/A	N/A	728.5/1457[2]
Peak FP16 Tensor TFLOPS with FP16 Accumulate[1]	130.5	154.8/309.6[2]	364.2/728.4[2]
Peak FP16 Tensor TFLOPS with FP32 Accumulate[1]	130.5	154.8/309.6[2]	364.2/728.4[2]
Peak BF16 Tensor TFLOPS with FP32 Accumulate[1]	N/A	154.8/309.6[2]	364.2/728.4[2]
Peak TF32 Tensor TFLOPS[1]	N/A	77.4/154.8[2]	182.1/364.2[2]
Peak INT8 Tensor TOPS[1]	261.00	309.7/619.4[2]	728.5/1457[2]
Peak INT4 Tensor TOPS[1]	522.0	619.3/1238.6[2]	1457/2914[2]
Frame Buffer Memory Size and Type	24576 MB GDDR6	49152 MB GDDR6	49152 MB GDDR6
Memory Interface	384-bit	384-bit	384-bit
Memory Clock (Data Rate)	14 Gbps	16 Gbps	20.0 Gbps
Memory Bandwidth	672 GB/sec	768 GB/sec	960 GB/sec

ROPs	96	112	192
Pixel Fill-rate (Gigapixels/sec)	170.0	201.6	481.0
Texture Units	288	336	568
Texel Fill-rate (Gigatexels/sec)	510	604.8	1422.8
L1 Data Cache/Shared Memory	96 KB	10752 KB	18176 KB
L2 Cache Size	6144 KB	6144 KB	98304 KB
Register File Size	18432 KB	21504 KB	36352 KB
Video Engines	1x Encode, 1x Decode	1x Encode, 2x Decode	3x Encode, 3x Decode
TGP (Total Graphics Power)	260 W	300W	300W
Transistor Count	18.6 Billion	28.3 Billion	76.3 Billion
Die Size	754 mm^2	628.4 mm^2	608.4 mm^2
Manufacturing Process	TSMC 12 nm FFN (FinFET NVIDIA)	Samsung 8 nm 8N NVIDIA Custom Process	TSMC 4N NVIDIA Custom Process
PCI Express Interface	Gen 3	Gen 4	Gen 4

1. Peak rates are based on GPU Boost Clock.
2. Effective TOPS / TFLOPS using the new Sparsity Feature
3. TOPS = IMAD-based integer math

Notice - The information provided in this specification is believed to be accurate and reliable as of the date provided. However, NVIDIA Corporation (“NVIDIA”) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This publication supersedes and replaces all other specifications for the product that may have been previously supplied.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regard to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

Trademarks - NVIDIA, the NVIDIA logo, GeForce, and GeForce RTX are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.