NVIDIA AMPERE GA102 GPU ARCHITECTURE

THE ULTIMATE PLAY

V1.0

Table of Contents

List of Figures

List of Tables

Introduction

Since inventing the world's first GPU (Graphics Processing Unit) in 1999, NVIDIA GPUs have been at the forefront of 3D graphics and GPU-accelerated computing. Each NVIDIA GPU Architecture is carefully designed to provide breakthrough levels of performance and efficiency.

The family of new NVIDIA® Ampere architecture GPUs is designed to accelerate many different types of computationally intensive applications and workloads. The first NVIDIA Ampere architecture GPU, the A100, was released in May 2020 and provides tremendous speedups for AI training and inference, HPC workloads, and data analytics applications. The A100 GPU is described in detail in the NVIDIA A100 GPU Tensor Core Architecture Whitepaper.

The newest members of the NVIDIA Ampere architecture GPU family, GA102 and GA104, are described in this whitepaper. GA102 and GA104 are part of the new NVIDIA "GA10x" class of Ampere architecture GPUs. GA10x GPUs build on the revolutionary NVIDIA Turing™ GPU architecture. Turing was the world's first GPU architecture to offer high performance real-time ray tracing, AI-accelerated graphics, energy-efficient inference acceleration for the datacenter, and professional graphics rendering all in one product.

GA10x GPUs add many new features and deliver significantly faster performance than Turing GPUs. In addition, GA10x GPUs are carefully crafted to provide the best performance per area and energy efficiency for traditional graphics workloads, and even more so for real-time ray tracing workloads. Compared to the Turing GPU Architecture, the NVIDIA Ampere Architecture is up to 1.7x faster in traditional raster graphics workloads and up to 2x faster in ray tracing.

GA102 is the most powerful Ampere architecture GPU in the GA10x lineup and is used in the GeForce RTX 3090 and GeForce RTX 3080 GPUs. The GeForce RTX 3090 is the highest performing GPU in the GeForce RTX lineup and has been built for 8K HDR gaming. With 10496 CUDA Cores, 24GB of GDDR6X memory, and the new DLSS 8K mode enabled, it can run many games at 8K@60 fps.

New HDMI 2.1 and AV1 decode features in GA10x GPUs allow users to stream content at 8K with HDR. Additionally, at up to 2x the performance of the GeForce RTX 2080, the GeForce RTX 3080 delivers the greatest generational leap of any GPU that has ever been made. Finally, the GeForce RTX 3070 GPU uses the new GA104 GPU and offers performance that rivals NVIDIA's previous generation flagship GPU, the GeForce RTX 2080 Ti.

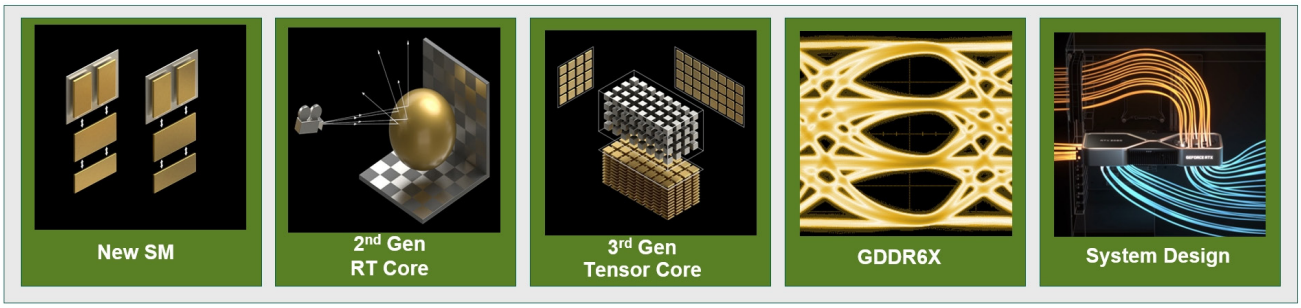


Figure 1. Ampere GA10x Architecture - A Giant Leap

This document focuses on NVIDIA GA102 GPU-specific architecture, and also general NVIDIA GA10x Ampere GPU architecture and features common to all GA10x GPUs. Additional GA10x GPU specifications are included in *Appendix A* on page 33. Other GA10x GPUs will be released in the future for different markets and price-points.

GA102 Key Features

Fabricated on Samsung's 8nm 8N NVIDIA Custom Process, the NVIDIA Ampere architecture-based GA102 GPU includes 28.3 billion transistors with a die size of 628.4 mm^2. Like all GeForce RTX GPUs, at the heart of GA102 lies a processor that contains three different types of compute resources:

- **Programmable Shading Cores**, which consist of **NVIDIA CUDA Cores**
- **RT Cores,** which accelerate Bounding Volume Hierarchy (BVH) traversal and intersection of scene geometry during ray tracing
- **Tensor Cores**, which provide enormous speedups for AI neural network training and inferencing

A full GA102 GPU incorporates 10752 CUDA Cores, 84 second-generation RT Cores, and 336 third-generation Tensor Cores, and is the most powerful consumer GPU NVIDIA has ever built for graphics processing. A GA102 SM doubles the number of FP32 shader operations that can be executed per clock compared to a Turing SM, resulting in 30 TFLOPS for shader processing in GeForce RTX 3080 (11 TFLOPS in the equivalent Turing GPU). Similarly, RT Cores offer double the throughput for ray/triangle intersection testing, resulting in 58 RT TFLOPS (compared to 34 in Turing). Finally, GA102's new Tensor Cores can process sparse neural networks at twice the rate of Turing Tensor Cores which do not support sparsity, yielding 238 sparse Tensor TFLOPS in RTX 3080 compared to 89 non-sparse Tensor TFLOPS in RTX 2080.

2x FP32 Processing

Most graphics workloads are composed of 32-bit floating point (FP32) operations. The Streaming Multiprocessor (SM) in the Ampere GA10x GPU Architecture has been designed to support double-speed processing for FP32 operations. In the Turing generation, each of the four SM processing blocks (also called partitions) had two primary datapaths, but only one of the two could process FP32 operations. The other datapath was limited to integer operations. GA10x includes FP32 processing on both datapaths, doubling the peak processing rate for FP32 operations. As a result, GeForce RTX 3090 delivers over 35 FP32 TFLOPS, an improvement of over 2x compared to Turing GPUs.

Second-Generation RT Core

The new RT Core includes a number of enhancements, combined with improvements to caching subsystems, that effectively deliver up to 2x performance improvement over the RT Core in Turing GPUs. In addition, the GA10x SM allows RT Core and graphics, or RT Core and compute workloads to run concurrently, significantly accelerating many ray tracing operations. These new features will be described in more detail later in this document.

Third-Generation Tensor Cores

The GA10x SM incorporates NVIDIA's new third-generation Tensor Cores, which support many new data types for improved performance, efficiency, and programming flexibility. A new

Sparsity feature can take advantage of fine-grained structured sparsity in deep learning networks to double the throughput of Tensor Core operations over the prior generation Turing Tensor Cores. The third-generation Tensor Cores accelerate AI features such as NVIDIA DLSS for AI super resolution now with support for up to 8K, the NVIDIA Broadcast app for AI-enhanced video and voice communications, and the NVIDIA Canvas app for AI-powered painting.

GDDR6X Memory

GDDR6X is the newest high-speed graphics memory. It currently supports speeds of 19.5 Gbps on the GeForce RTX 3090, and 19 Gbps for the GeForce RTX 3080. With its 320-bit memory interface and GDDR6X memory, the GeForce RTX 3080 delivers 1.5x more memory bandwidth than its predecessor, the RTX 2080 Super.

Third-Generation NVLink®

GA102 GPUs utilize NVIDIA's third-generation NVLink interface, which includes four x4 links, with each link providing 14.0625 GB/sec bandwidth in each direction between two GPUs. Four links provide 56.25 GB/sec bandwidth in each direction, and 112.5 GB/sec total bandwidth between two GPUs. Two RTX 3090 GPUs can be connected together for SLI using NVLink. (Note that 3-Way and 4-Way SLI configurations are not supported.)

PCIe Gen 4

GA10x GPUs feature a PCI Express 4.0 host interface. PCIe Gen 4 provides double the bandwidth of PCIe 3.0, up to 16 Gigatransfers/second bit rate, with a x16 PCIe 4.0 slot providing up to 64 GB/sec of peak bandwidth.

The first graphics card based on the Ampere GA10x GPU Architecture is the GeForce RTX 3080. Table 2 below provides a high-level comparison of the GeForce RTX 3080 versus its predecessor, the RTX 2080 Super GPU. (Specifications for other GeForce RTX graphics cards using GA102 and GA104 GPUs can be found in Appendix A.)

Ampere GPU Architecture In-Depth

GPC, TPC, and SM High-Level Architecture

Like prior NVIDIA GPUs, GA102 is composed of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), Raster Operators (ROPS), and memory controllers. The full GA102 GPU contains seven GPCs, 42 TPCs, and 84 SMs.

The GPC is the dominant high-level hardware block with all of the key graphics processing units residing inside the GPC. Each GPC includes a dedicated Raster Engine, and now also includes two ROP partitions (each partition containing eight ROP units), which is a new feature for NVIDIA Ampere Architecture GA10x GPUs and described in more detail below. The GPC includes six TPCs that each include two SMs and one PolyMorph Engine.

Note: The GA102 GPU also features 168 FP64 units (two per SM), which are not depicted in this diagram. The FP64 TFLOP rate is 1/64th the TFLOP rate of FP32 operations. The small number of FP64 hardware units are included to ensure any programs with FP64 code operate correctly, including FP64 Tensor Core code.

Figure 2. GA102 Full GPU with 84 SMs

Each SM in GA10x GPUs contain 128 CUDA Cores, four third-generation Tensor Cores, a 256 KB Register File, four Texture Units, one second-generation Ray Tracing Core, and 128 KB of L1/Shared Memory, which can be configured for differing capacities depending on the needs of the compute or graphics workloads.

The memory subsystem of GA102 consists of twelve 32-bit memory controllers (384-bit total). 512 KB of L2 cache is paired with each 32-bit memory controller, for a total of 6144 KB on the full GA102 GPU.

ROP Optimizations

In previous NVIDIA GPUs, the ROPs were tied to the memory controller and L2 cache. Beginning with GA10x GPUs, the ROPs are now part of the GPC, which boosts performance of raster operations by increasing the total number of ROPs, and eliminating throughput mismatches between the scan conversion frontend and raster operations backend.

With seven GPCs and 16 ROP units per GPC, the full GA102 GPU consists of 112 ROPs instead of the 96 ROPS that were previously available in a 384-bit memory interface GPU like the prior generation TU102. This improves multisample anti-aliasing, pixel fillrate, and blending performance.

GA10x SM Architecture

The Turing SM was NVIDIA's first SM architecture to include dedicated cores for Ray Tracing operations. Volta GPUs introduced Tensor Cores, and Turing included enhanced second-generation Tensor Cores. Another innovation supported by the Turing and Volta SMs was concurrent execution of FP32 and INT32 operations. The GA10x SM improves upon all the above capabilities, while also adding many powerful new features.

Like prior GPUs, the GA10x SM is partitioned into four processing blocks (or partitions), each with a 64 KB register file, an L0 instruction cache, one warp scheduler, one dispatch unit, and sets of math and other units. The four partitions share a combined 128 KB L1 data cache/shared memory subsystem.

Unlike the TU102 SM which includes two second-generation Tensor Cores per partition and eight Tensor Cores total, the new GA10x SM includes one third-generation Tensor Core per partition and four Tensor Cores total, with each GA10x Tensor Core being twice as powerful as a Turing Tensor Core.

Compared to Turing, the GA10x SM's combined L1 data cache and shared memory capacity is 33% larger. For graphics workloads, the cache partition capacity is doubled compared to Turing, from 32KB to 64KB.

Figure 3. GA10x Streaming Multiprocessor (SM)

2x FP32 Throughput

In the Turing generation, each of the four SM processing blocks (also called partitions) had two primary datapaths, but only one of the two could process FP32 operations. The other datapath was limited to integer operations. GA10X includes FP32 processing on both datapaths, doubling the peak processing rate for FP32 operations. One datapath in each partition consists of 16

FP32 CUDA Cores capable of executing 16 FP32 operations per clock. Another datapath consists of both 16 FP32 CUDA Cores and 16 INT32 Cores, and is capable of executing either 16 FP32 operations OR 16 INT32 operations per clock. As a result of this new design, each GA10x SM partition is capable of executing either 32 FP32 operations per clock, or 16 FP32 and 16 INT32 operations per clock. All four SM partitions combined can execute 128 FP32 operations per clock, which is double the FP32 rate of the Turing SM, or 64 FP32 and 64 INT32 operations per clock.

Modern gaming workloads have a wide range of processing needs. Many workloads have a mix of FP32 arithmetic instructions (such as FFMA, floating point additions (FADD), or floating-point multiplications (FMUL)), along with many simpler integer instructions such as adds for addressing and fetching data, floating point compare, or min/max for processing results, etc. Turing introduced a second math datapath to the SM, which provided significant performance benefits for these types of workloads. However, other workloads can be dominated by floating point instructions. Adding floating point capability to the second datapath will significantly help these workloads. Performance gains will vary at the shader and application level depending on the mix of instructions. Ray tracing denoising shaders are a good example of a workload that can benefit greatly from doubling FP32 throughput.

The GA10x SM continues to support double-speed FP16 (HFMA) operations which are supported in Turing. And similar to TU102, TU104, and TU106 Turing GPUs, standard FP16 operations are handled by the Tensor Cores in GA10x GPUs.

Table 1. Comparative X-Factors for FP32 Throughput

(Relative to FP32 operations in the Pascal GP102 GPU used in GeForce GTX 1080 Ti)

	Turing	GA10x
FP32	1X	2X
FP16	2X	2X

Larger and Faster Unified Shared Memory and L1 Data Cache

As we mentioned previously, like the prior generation Turing architecture, GA10x features a unified architecture for shared memory, L1 data cache, and texture caching. This unified design can be reconfigured depending on workload to allocate more memory for the L1 or shared memory depending on need. The L1 data cache capacity has increased to 128 KB per SM.

In compute mode, the GA10x SM will support the following configurations:

- 128 KB L1 + 0 KB Shared Memory
- 120 KB L1 + 8 KB Shared Memory
- 112 KB L1 + 16 KB Shared Memory
- 96 KB L1 + 32 KB Shared Memory
- 64 KB L1 + 64 KB Shared Memory
- 28 KB L1 + 100 KB Shared Memory

For graphics workloads and async compute, GA10x will allocate 64 KB L1 data / texture cache (increasing from 32 KB cache allocation on Turing), 48 KB Shared Memory, and 16 KB reserved for various graphics pipeline operations.

The full GA102 GPU contains 10752 KB of L1 cache (compared to 6912 KB in TU102). In addition to increasing the size of the L1, GA10x also features double the shared memory bandwidth compared to Turing (128 bytes/clock per SM versus 64 bytes/clock in Turing). Total L1 bandwidth for GeForce RTX 3080 is 219 GB/sec versus 116 GB/sec for GeForce RTX 2080 Super.

Table 2. Comparison of GeForce RTX 3080 to GeForce RTX 2080 Super

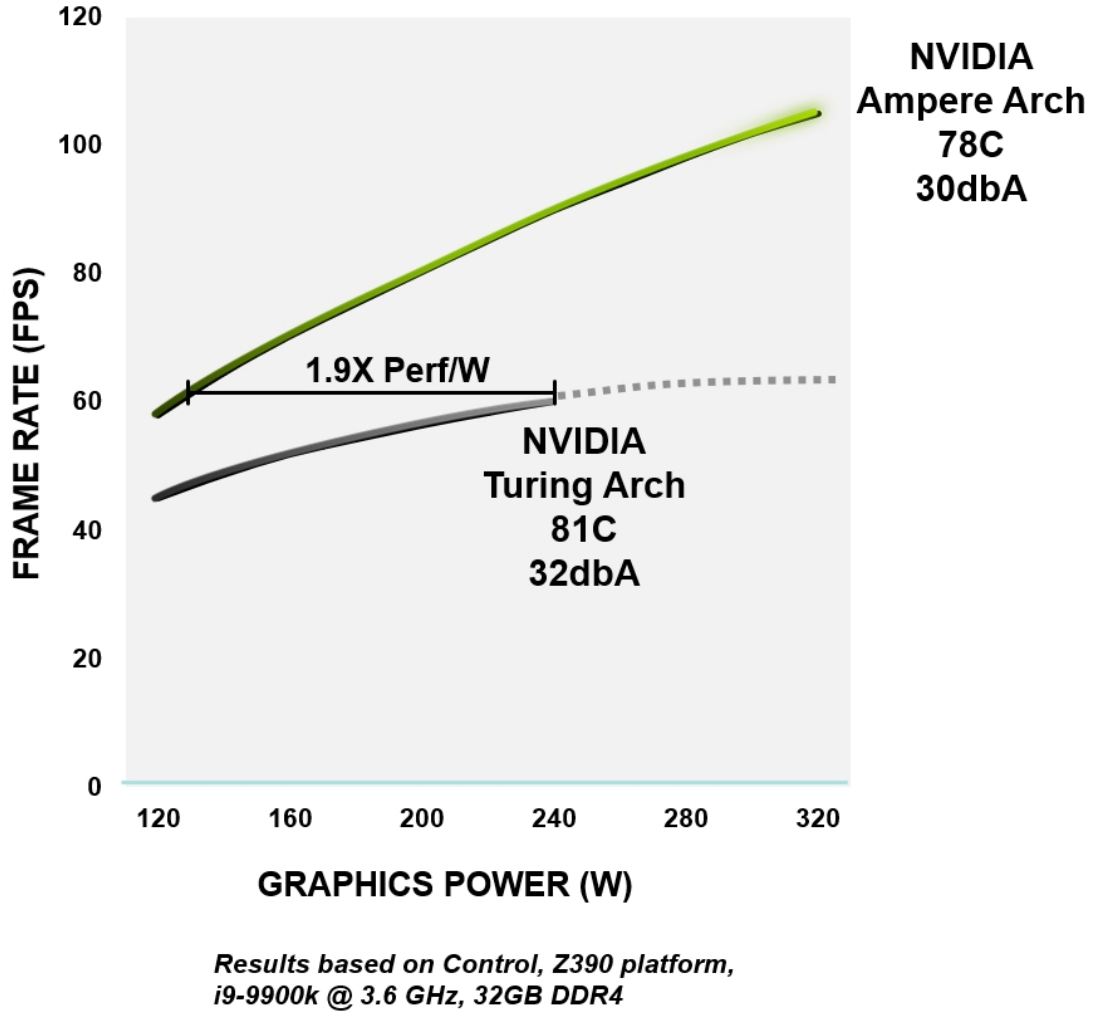
Graphics Card	GeForce RTX 2080 Founders Edition	GeForce RTX 2080 Super Founders Edition	GeForce RTX 3080 10 GB Founders Edition
GPU Codename	TU104	TU104	GA102
GPU Architecture	NVIDIA Turing	NVIDIA Turing	NVIDIA Ampere
GPCs	6	6	6
TPCs	23	24	34
SMs	46	48	68
CUDA Cores / SM	64	64	128
CUDA Cores / GPU	2944	3072	8704
Tensor Cores / SM	8 (2nd Gen)	8 (2nd Gen)	4 (3rd Gen)
Tensor Cores / GPU	368	384 (2nd Gen)	272 (3rd Gen)
RT Cores	46 (1st Gen)	48 (1st Gen)	68 (2nd Gen)
GPU Boost Clock (MHz)	1800	1815	1710
Peak FP32 TFLOPS (non-Tensor)[1]	10.6	11.2	29.8
Peak FP16 TFLOPS (non-Tensor)[1]	21.2	22.3	29.8
Peak BF16 TFLOPS (non-Tensor)[1]	NA	NA	29.8
Peak INT32 TOPS (non-Tensor)[1,3]	10.6	11.2	14.9
Peak FP16 Tensor TFLOPS with FP16 Accumulate[1]	84.8	89.2	119/238[2]
Peak FP16 Tensor TFLOPS with FP32 Accumulate[1]	42.4	44.6	59.5/119[2]
Peak BF16 Tensor TFLOPS with FP32 Accumulate[1]	NA	NA	59.5/119[2]
Peak TF32 Tensor TFLOPS[1]	NA	NA	29.8/59.5[2]
Peak INT8 Tensor TOPS[1]	169.6	178.4	238/476[2]
Peak INT4 Tensor TOPS[1]	339.1	356.8	476/952[2]
Frame Buffer Memory Size and Type	8192 MB GDDR6	8192 MB GDDR6	10240 MB GDDR6X
Memory Interface	256-bit	256-bit	320-bit
Memory Clock (Data Rate)	14 Gbps	15.5 Gbps	19 Gbps
Memory Bandwidth	448 GB/sec	496 GB/sec	760 GB/sec
ROPs	64	64	96
Pixel Fill-rate (Gigapixels/sec)	115.2	116.2	164.2
Texture Units	184	192	272
Texel Fill-rate (Gigatexels/sec)	331.2	348.5	465
L1 Data Cache/Shared Memory	4416 KB	4608 KB	8704 KB

L2 Cache Size	4096 KB	4096 KB	5120 KB
Register File Size	11776 KB	12288 KB	17408 KB
TGP (Total Graphics Power)	225 W	250 W	320W
Transistor Count	13.6 Billion	13.6 Billion	28.3 Billion
Die Size	545 mm^2	545 mm^2	628.4 mm^2
Manufacturing Process	TSMC 12 nm FFN (FinFET NVIDIA)	TSMC 12 nm FFN (FinFET NVIDIA)	Samsung 8 nm 8N NVIDIA Custom Process

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

Performance Per Watt

The entire NVIDIA Ampere GPU architecture is crafted for efficiency - from custom process design, to circuit design, logic design, packaging, memory, power, and thermal design, down to the PCB design, the software, and algorithms. At the same performance level, Ampere architecture GPUs are up to 1.9x more power efficient than Turing GPUs.



RTX 3080 Power Efficiency Compared to Turing Architecture GeForce RTX 2080 Super

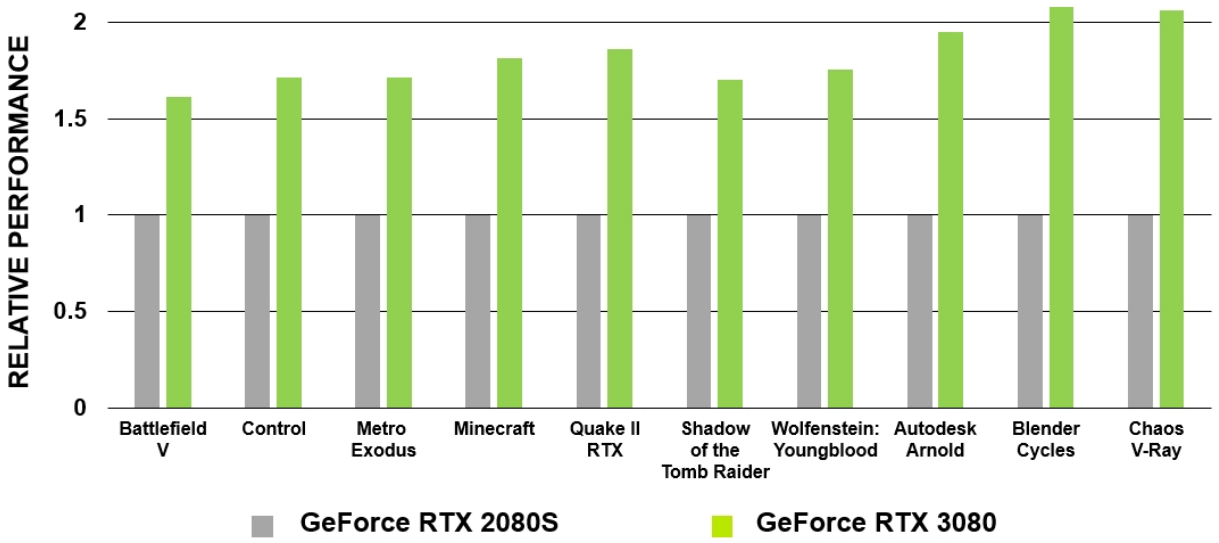
Figure 4. NVIDIA Ampere GA10x Architecture Power Efficiency

Second-Generation Ray Tracing Engine in GA10x GPUs

Turing-based GeForce RTX GPUs were the first GPUs to make real-time, cinema-quality ray traced graphics a reality in PC games. Prior to the arrival of Turing, rendering high-quality ray traced scenes in real time with fluid frame rates was thought to be years away. Thanks to many Turing architectural advancements (such as dedicated RT Cores, Tensor Cores, and software advances in denoising and ray tracing algorithms), along with NVIDIA engineers working closely with Microsoft on their DirectX Ray Tracing (DXR) API, and the Khronos Group on ray tracing extensions to the Vulkan API, this dream became possible. (Please refer to the NVIDIA Turing GPU whitepaper for more background information on ray tracing.)

GA10x GPUs feature NVIDIA's second-generation ray tracing technology, which builds on the learnings from Turing. Like Turing, the second-generation RT Core in GA10x includes dedicated hardware units for BVH traversal and ray-triangle intersection testing. Once the SM has cast the ray, the RT Core will perform all of the calculations needed for BVH traversal and triangle intersection tests, and will return a hit or no hit to the SM.

Ampere architecture RT Cores double the ray/triangle intersection testing rate over Turing architecture RT Cores.



NVIDIA Ampere architecture-based GeForce RTX 3080 is up to 2x faster in ray tracing workloads than Turing Architecture-based GeForce RTX 2080 Super. Games and apps run at 4K using a Core i9-10900K CPU.

Figure 5. GeForce RTX 3080 vs GeForce RTX 2080 Super RT Performance

Simultaneous Compute and Graphics (SCG), more commonly known as Async Compute, is a feature that allows the GPU to perform compute and graphics workloads simultaneously. Typical scenes in modern games are increasingly mixing graphics functions with effects that rely on asynchronous compute operations, improving GPU utilization and enhancing visual quality.

With the introduction of real-time ray tracing, usage of compute workloads is expanding even further.

The GA10x GPU enhances the Async Compute functionality of prior NVIDIA GPUs with a new capability that allows RT Core and graphics, or RT Core and compute workloads to be processed concurrently in each GA10x GPU SM. The GA10x SM can process two compute workloads simultaneously, and is not limited to just compute and graphics simultaneously as in prior GPU generations, allowing scenarios such as a compute-based denoising algorithm to run concurrently with RT Core-based ray tracing work.

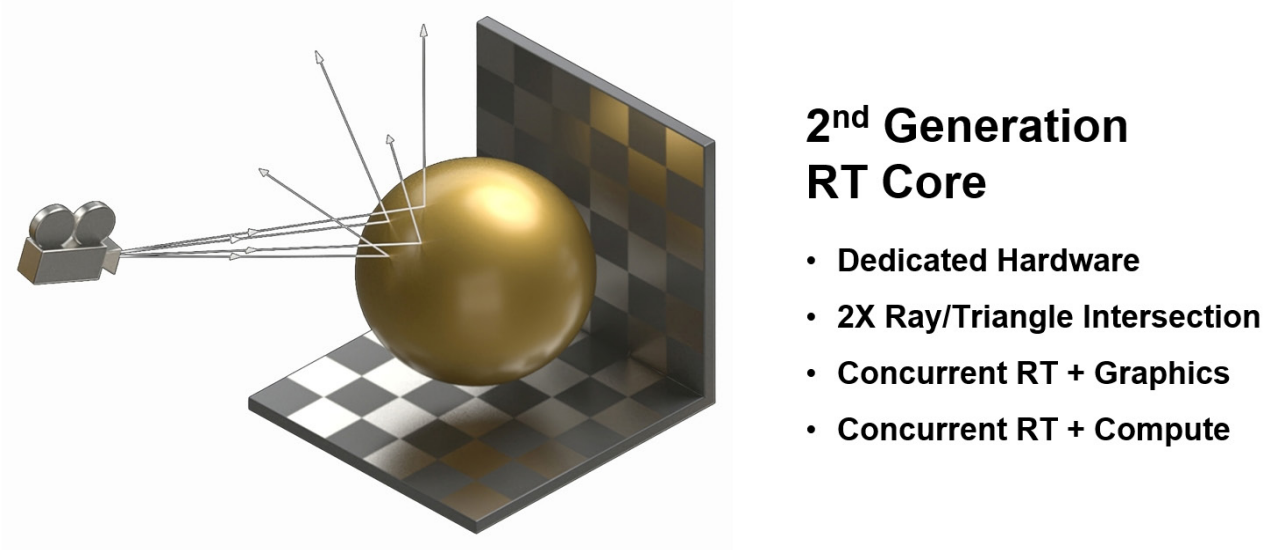


Figure 6. Second-Generation RT Core in GA10x GPUs

Note that RT Core-heavy workloads do not substantially stress SM cores, leaving much SM processing power available for other workloads. This is a big advantage over competitive architectures that do not have dedicated RT Cores and must use their standard processing cores, texture units, etc. to perform both graphics and ray tracing operations.

In addition to Concurrent RT operations, the GA10x GPUs can also run various other types of simultaneous compute-compute workloads. New software controls permit different priorities to be assigned to workloads, further improving effectiveness.

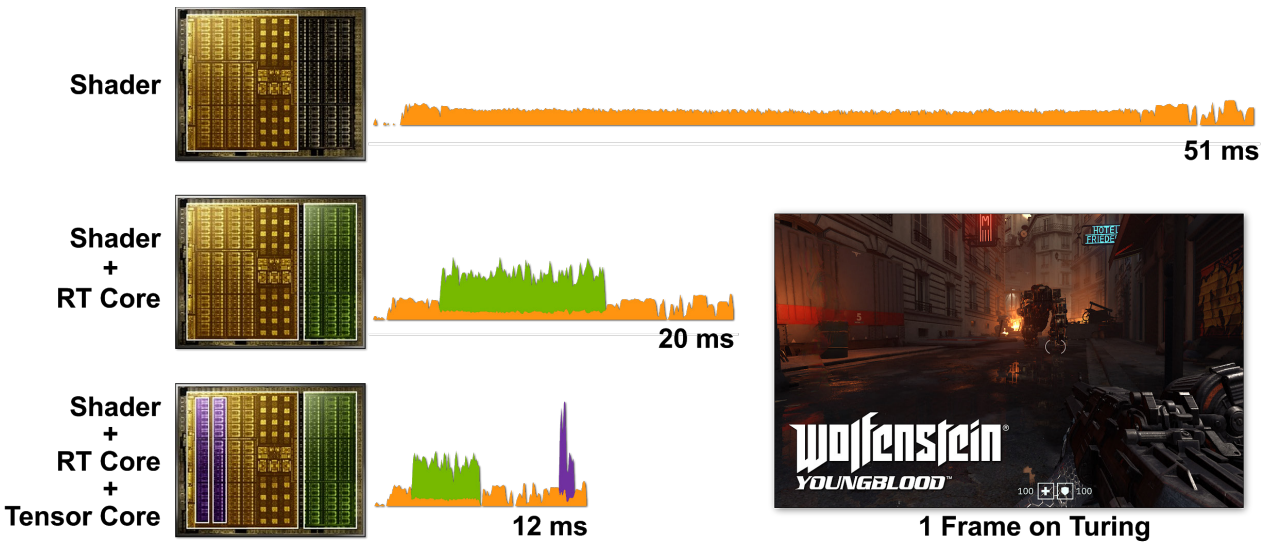
Table 3. Ray Tracing Feature Comparison

	NVIDIA Turing Architecture *(TU102 Full-Chip)*	NVIDIA Ampere Architecture *(GA102 Full-Chip)*
Dedicated RT Cores	Yes (72 RT Cores)	Yes (84 RT Cores)
Ray / Bounding Box Acceleration	Yes	Yes
Ray / Triangle Acceleration	Yes	Yes
Tree Traversal Acceleration	Yes	Yes
Instance Transform Acceleration	Yes	Yes
Concurrent RT and Shading	No	Yes

Dedicated L1 Interface	Yes	Yes
Ray / Triangle Intersection Test Culling Rate Speedup	1.0x	2.0x
Overall GPU RT Speedup	1.0x	2.0x

Ampere Architecture RTX Processors in Action

Ray tracing and shader work is computationally demanding. But it would be much more expensive to run everything with shaders (CUDA Cores) alone. Offloading work to the RT Cores and Tensor Cores speeds up processing significantly. Figure 7 shows one frame trace of the ray-traced Wolfenstein: Youngblood game to see RTX technology in action running on a Turing architecture-based RTX 2080 Super.

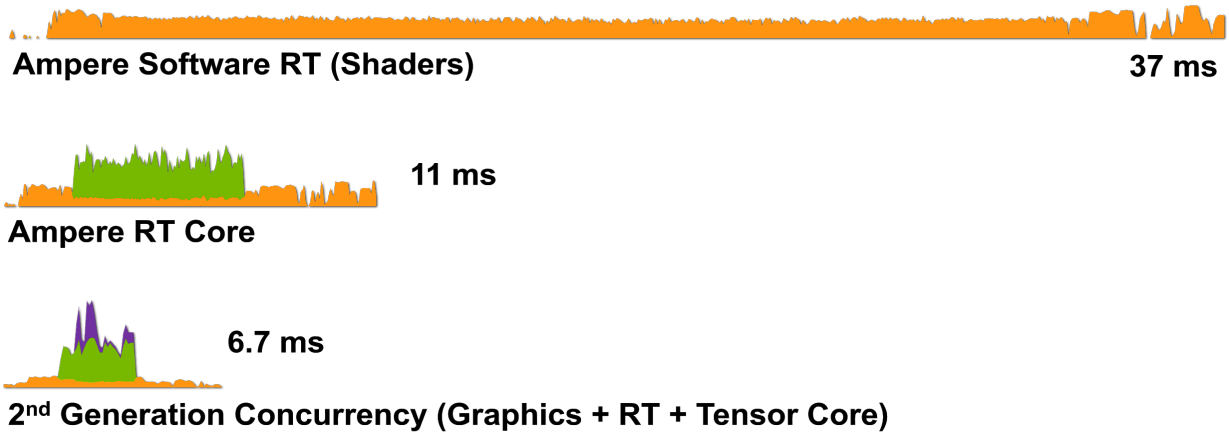


Turing-based RTX 2080 Super GPU rendering one frame of Wolfenstein: Youngblood using Shader Cores (CUDA Cores) only, Shader Cores + RT Cores, and Shader Core + RT Cores + Tensor Cores. Note the progressively reduced frame times when adding the different RTX processing cores.

Figure 7. Turing RTX Technology Improves Performance

Using shaders alone, it takes 51 ms to run this single frame (~20FPS). When ray tracing work is moved to the RT Cores and run concurrently, the frame renders at a much faster 20 ms (50 FPS). Utilizing Tensor Cores to enable DLSS reduces the frame time to just 12 ms (~83 FPS).

NVIDIA Ampere architecture improves performance far greater when rendering the same frame from the same game.



The Ampere architecture-based RTX 3080 GPU rendering one frame of Wolfenstein: Youngblood using Shader Cores (CUDA Cores) only, Shader Cores + RT Cores, and Shader Core + RT Cores + Tensor Cores

Figure 8. Ampere Architecture RTX Technology Further Improves Performance

With its 2x FP32, L1 cache advancements, second-generation RT Cores, third-generation Tensor Cores, GDDR6x memory, new concurrency features, and other new technologies, the RTX 3080 renders the frame in 6.7 ms (150 FPS), a huge improvement.

GA10x GPU Hardware Acceleration for Ray-Traced Motion Blur

Motion blur is a very popular and important computer graphics effect used in movies, games, and many different types of professional rendering applications to better simulate reality, or just to create cool or artistic effects. To understand motion blur, think of how a camera generates the effect. A photographic image is not created instantaneously, it's created by exposing film to light for a finite period of time. Objects moving quickly enough with respect to the camera's shutter duration will appear as streaks or smears in the photograph. For a GPU to create realistic-looking motion blur when objects in the scene (geometries) are moving quickly in front of a static camera, or the camera is scanning across static or moving objects, it must simulate how a camera and film reproduce such scenes. Motion blur is especially important for film production, because films are played at 24 frames per second and a scene rendered without motion blur will appear choppy.

GPUs have been able to approximate motion blur using a variety of techniques for years, both for offline high-quality rendering such as in movies, and for real-time applications like games. High-quality blur effects can be very computationally-intensive, often requiring rendering and blending multiple frames over some time interval. Post-processing may also be required to further improve results. A number of tricks and short-cut methods are often used for real-time motion blur, such as in games, but the blurring can lack realism. For example, the images might be unnaturally smeary, noisy, or have ghosting/strobing artifacts, or the motion blur effect might be altogether missing in reflections and translucent materials. In contrast, ray-traced motion blur can be more accurate and realistic looking without unwanted artifacts, but can also take a long time to render on a GPU, especially without hardware acceleration of ray tracing operations integral to performing blurring effects.

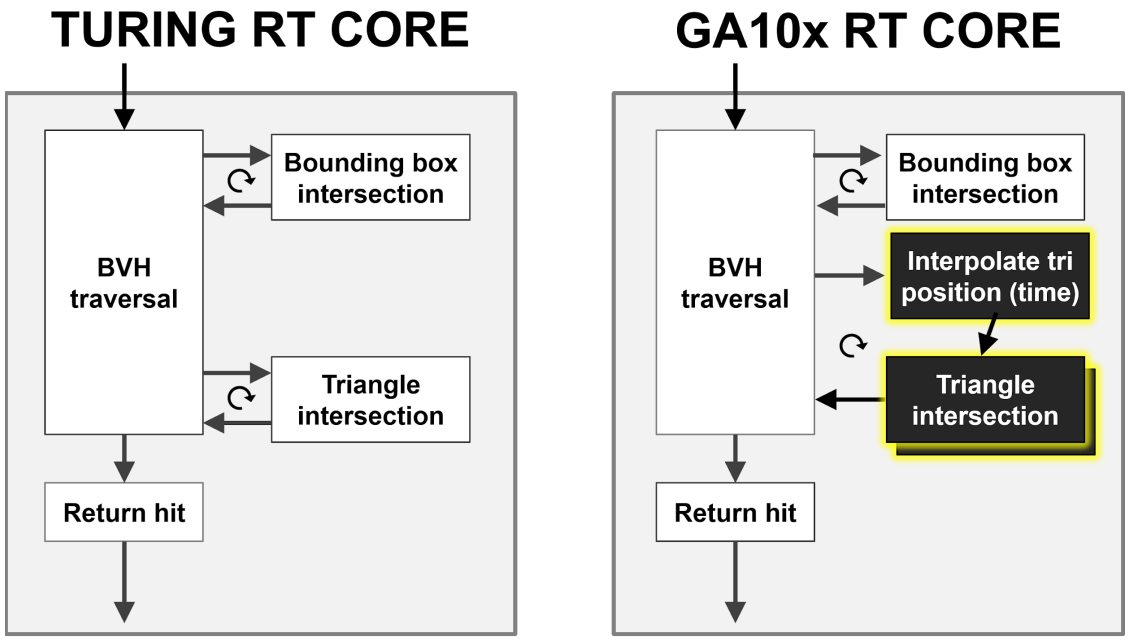
Different algorithms, or combinations of algorithms can be employed to perform ray-traced motion blur. One popular method randomly shoots a number of timestamped rays into the scene. A motion blur-capable BVH returns the ray's hit information against geometry moving over time, sampled at the timestamp associated with each ray. Those samples are then shaded and combined to create the final blur effect. Many variations on the above method exist. NVIDIA's OptiX API has supported such techniques since OptiX 5.0, introduced in 2017.

Motion blur can occur when the camera is moving and looking across various static objects in the scene, or where geometry is moving in front of a static (or moving) camera. Our Turing GPUs can accelerate the moving camera type of motion blur quite well. Multiple rays can be shot into a scene over a time interval hitting static geometry, and the RT Cores accelerate BVH traversals, perform ray/triangle intersection tests, and return results to create blurring effects. However, performing motion blur on moving geometry can be more challenging within a given time interval as BVH information changes as objects move.

The GA10x RT Core includes new acceleration features that work in concert with small modifications to the BVH to significantly accelerate both moving geometry and deforming geometry types of motion blur. NVIDIA OptiX 7 enables developers to specify motion paths for geometry and associate a time with each ray to enable all of these motion blur effects.

As seen in Figure 9, the Turing RT Core includes hardware-based BVH traversal, ray/bounding box intersection testing, and ray/triangle intersection testing. The GA10x RT Core improves on the performance and capabilities of the Turing RT Core architecture, including the addition of a new Interpolate Triangle Position unit (explained below) that accelerates ray-traced motion blur.

Both the Turing and GA10x RT Cores implement a MIMD (Multiple Instruction Multiple Data) architecture that can process many rays at once.



The GA10x RT Core doubles the ray/triangle intersection testing rate over Turing architecture RT Cores, and also adds a new Interpolate Triangle Position acceleration unit to assist in ray-traced motion blur operations.

Figure 9. Ampere Architecture Motion Blur Hardware Acceleration

Diving deeper into the problem, the challenge with motion blur is that the triangles in the scene are no longer fixed in time. In basic ray tracing, every triangle is defined inside the BVH acceleration structure. When a ray is traced, ray/bounding box and ray/triangle intersection tests are performed, and if a triangle is hit, the hit information is sampled and returned. With motion blur, as shown in Figure 10, no triangle has a fixed position. Instead, what's inside the BVH is a formula that says, "if you tell me what the time is, I can tell you where this triangle is in space." Every ray is assigned a timestamp indicating its time to be traced which feeds into the BVH equations to solve for the position of the triangle and the ray/triangle intersection.

If this process is not GPU hardware-accelerated, it becomes a bottleneck for a motion blur workload. It's a challenging problem to solve, basically looking across an arc of times. And the process might not be linear. It could be tracking the curve of a propeller blade or a wheel to find triangle(s).

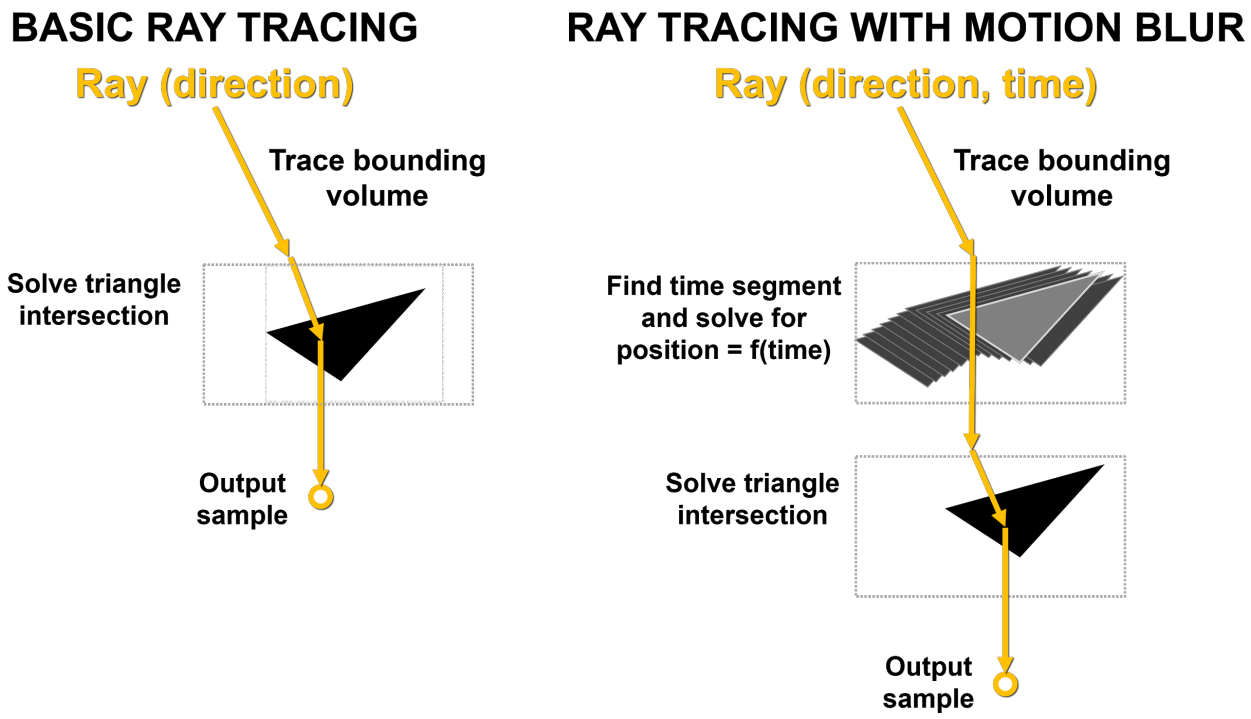


Figure 10. Basic Ray Tracing vs Ray Tracing with Motion Blur

On the left side of Figure 11 below, shooting rays into the scene without motion could have many rays as shown, but they're all hitting the same triangle at the same point in time. The white dots show where they actually hit the triangle, and output is produced and returned. With motion blur, every ray exists at a different point in time. Three ray colors are shown below for illustration to differentiate them, but every ray is randomly assigned a different timestamp. For example, the orange rays try to intersect orange triangles at different points in time, then green and blue rays try to intersect green and blue triangles, respectively. The result is a more smeared out, mathematically correct filtered output as shown at the bottom, derived from a mix of samples generated by the rays hitting the triangles at different positions across different points in time.

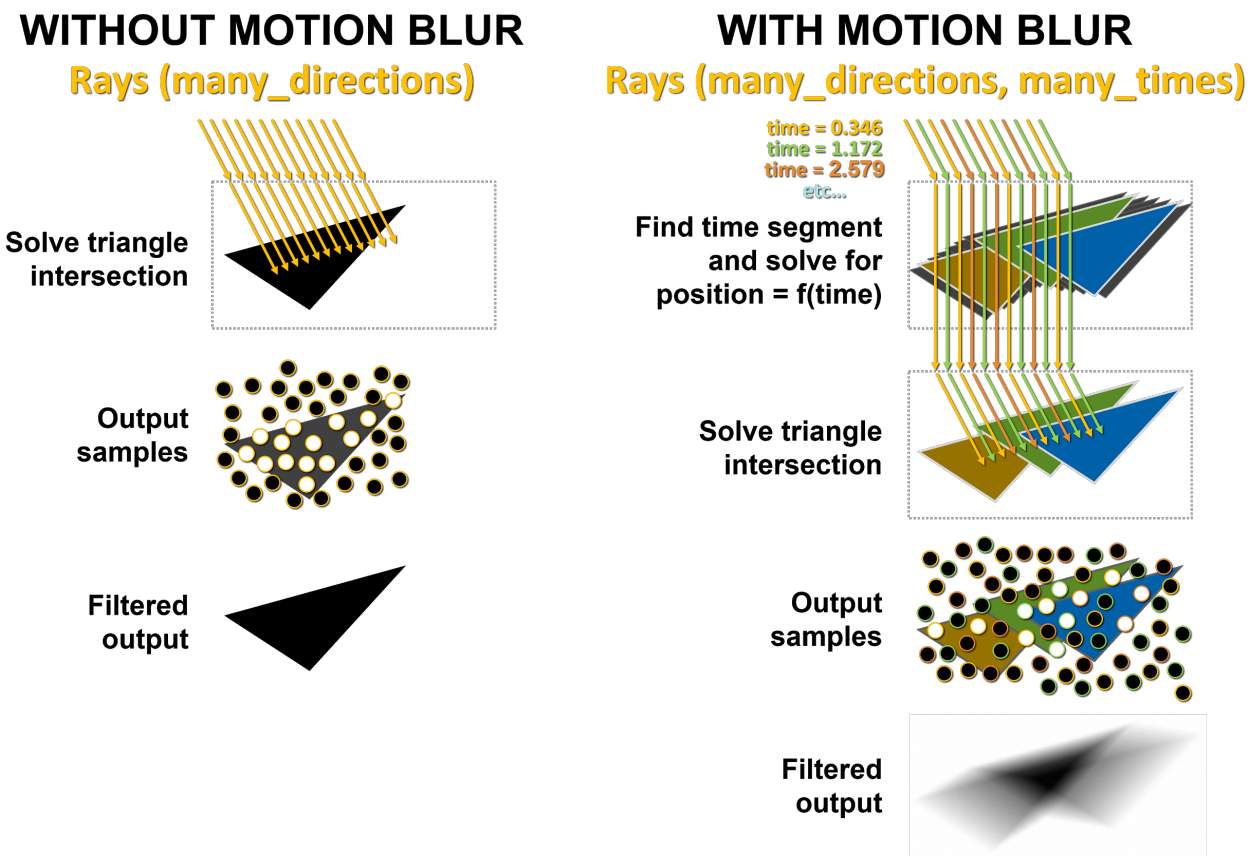


Figure 11. Rendering Without vs With Motion Blur on GA10x

The new Interpolate Triangle Position unit is able to generate triangles in the BVH in between existing triangle representations based on object motion, so that rays can intersect triangles at their expected positions in object space at the times specified by the ray timestamps. This new unit allows accurate ray-traced motion blur rendering to occur up to eight times faster than the Turing GPU architecture.

GA10x hardware-accelerated motion blur is supported by Blender 2.90, Chaos V-Ray 5.0, Autodesk Arnold, and Redshift Renderer 3.0.X using the NVIDIA OptiX 7.0 API.

We've measured up to 5x faster motion blur rendering with a GA102-based RTX 3080 compared to an RTX 2080 Super running our own Blender Cycles 2.90 ray tracing demo that uses OptiX 7.0.

Third-Generation Tensor Cores in GA10x GPUs

Tensor Cores are specialized execution units designed specifically for performing the tensor / matrix operations that are the core compute function used in Deep Learning. First introduced in Volta and further improved on Turing, Tensor Cores provide tremendous speed-ups for matrix computations at the heart of deep learning neural network training and inferencing operations. (Refer to the NVIDIA Tesla V100 GPU Architecture for background information on basic Tensor Core operation.)

Tensor Cores accelerate the matrix-matrix multiplication at the heart of neural network training and inferencing functions. Inference computations are at the core of most AI-based graphics applications, in which useful and relevant information can be inferred and delivered by a trained deep neural network (DNN) based on a given input. Examples of inference include enhancing graphics qualities through DLSS (Deep Learning Super Sampling), AI-based denoising, removing background noise of in-game voice chats through RTX Voice, AI-based green-screen effects in NVIDIA RTX Broadcast engine, and many more.

The Turing Tensor Core design (second-generation Tensor Core architecture) added INT8 and INT4 precision modes for inferencing workloads that can tolerate quantization. The introduction of Tensor Cores into Turing-based GeForce gaming GPUs made it possible to bring real-time deep learning to gaming applications for the first time.

The new third-generation Tensor Core architecture in GA10x GPUs accelerates more data types, and includes a new Sparsity feature (described below) that delivers up to a 2x speedup for matrix multiplication compared to the Tensor Cores in the Turing architecture.

The third-generation Tensor Core design in GA10x GPUs further increases raw performance and brings new precision modes such as TF32 and BFloat16 for AI-based features of NVIDIA NGX Neural Services that enhance graphics, rendering, and other types of client-side applications. Examples of NGX AI features include DLSS, AI Super Rez, and AI Slow-Mo.

Comparison of Turing vs GA10x GPU Tensor Cores

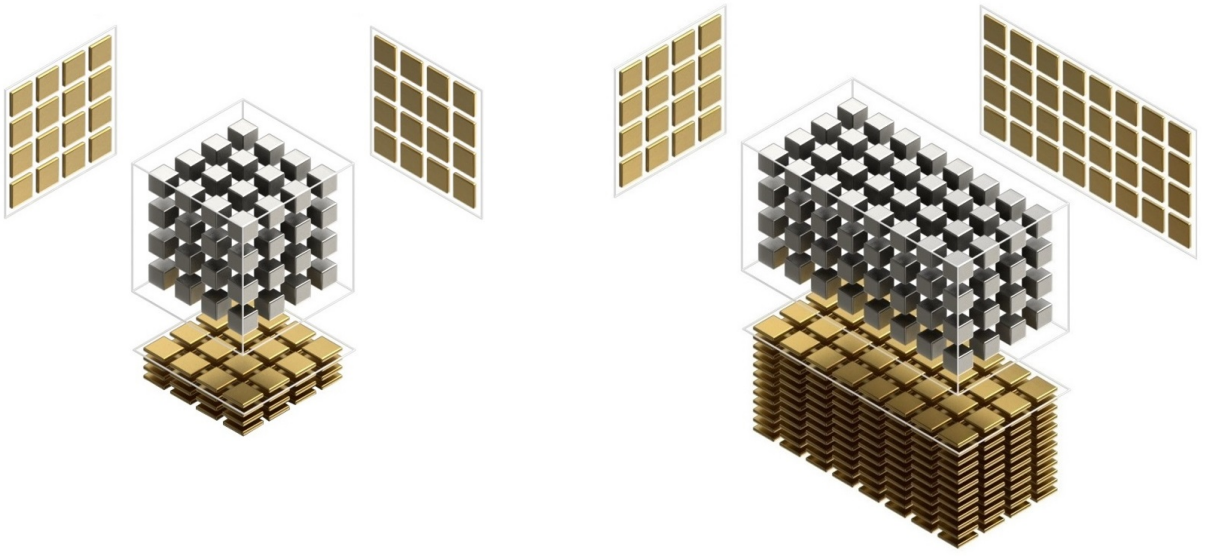
NVIDIA GA10x GPUs includes an area-optimized version of the new Ampere architecture Tensor Core that was first introduced in A100 and our DGX A100 deep learning system. Compared to Turing, Ampere architecture Tensor Cores are reorganized to improve efficiency and reduce energy consumption. Each Ampere architecture SM has a smaller number of Tensor Cores, but each Tensor Core is more powerful.

In addition, Ampere architecture GPUs introduce hardware support for processing matrices with specific sparsity patterns at up to 2x throughput, by skipping the zero-valued elements. In the GA10x configuration, each SM has double the throughput of a Turing SM when processing sparse matrices, while retaining the same total throughput of a Turing SM for dense operations. Sparsity enables the RTX Ampere architecture products to efficiently deliver a huge boost in throughput compared to Turing GPUs, with RTX 3080 offering 2.7x more tensor operation throughput compared to RTX 2080 Super.

Table 4. Comparison of NVIDIA Turing vs Ampere Architecture Tensor Core

	TU102 SM (RTX 2080 Super)	GA100 SM (A100)	GA10x SM (RTX 3080)
GPU Architecture	NVIDIA Turing	NVIDIA Ampere	NVIDIA Ampere
Tensor Cores per SM	8	4	4
FP16 FMA operations per Tensor Core	64	Dense: 256 Sparse: 512	Dense: 128 Sparse: 256
Total FP16 FMA operations per SM	512	Dense: 1024 Sparse: 2048	Dense: 512 Sparse: 1024

Below is a visual depiction of a single Ampere architecture Tensor Core and a single Turing architecture Tensor Core performing matrix math calculations and showing comparative throughputs of RTX 3080 vs RTX 2080 Super as represented by the stacks of completed operations performed over the same amount of time.



TURING ARCHITECTURE TENSOR CORE
(GeForce RTX 2080 Super)

AMPERE ARCHITECTURE TENSOR CORE with Sparsity
(GeForce RTX 3080)

With Sparsity enabled, the GeForce RTX 3080 delivers 2.7X higher peak FP16 Tensor Core operation throughput compared to a GeForce RTX 2080 Super with dense Tensor Core operations.

Figure 12. Ampere Architecture Tensor Core vs Turing Tensor Core

With a total of 68 SMs per GPU and a boost clock of 1710 MHz, the GeForce RTX 3080 GPU delivers 119 peak FP16 Tensor TFLOPS with FP16 accumulate, and with Sparsity enabled, 238 peak FP16 Tensor TFLOPS with FP16 accumulate. RTX 3080 delivers 238 peak INT8 Tensor TOPS and 476 peak INT4 Tensor TOPS, and double those rates with Sparsity enabled.

NVIDIA Ampere Architecture Tensor Cores Support New DL Data Types

In addition to FP16 precision introduced on the Volta Tensor Core, and the INT8, INT4 and binary 1-bit precisions added in the Turing Tensor Core, the GA10x Tensor Core adds support for TF32 and BF16 data types, similar to the recently introduced NVIDIA A100 GPU. (Note that GA10x GPUs do not include Tensor Core acceleration for double-precision (FP64) operations, as provided in A100.)

BF16 is an alternative to IEEE FP16, and includes an 8-bit exponent, 7-bit mantissa, and 1 sign-bit. Both FP16 and BF16 have been shown to successfully train neural networks in mixed-precision mode, matching FP32 training results without hyper-parameter adjustment. Both FP16 and BF16 modes of Tensor Cores provide 4x more math throughput than standard FP32 in GA10x GPUs.

Today, the default math for AI training is FP32, without Tensor Core acceleration. The NVIDIA Ampere architecture introduces new support for TF32, enabling AI training to use Tensor Cores by default with no effort on the user's part. Non-tensor operations continue to use the FP32 datapath, while TF32 Tensor Cores read FP32 data and use the same range as FP32 with reduced internal precision, before producing a standard IEEE FP32 output. TF32 includes an 8-bit exponent (same as FP32), 10-bit mantissa (same precision as FP16) and 1 sign-bit. TF32 mode of an Ampere architecture GPU Tensor Core provides 2x more throughput than standard FP32.

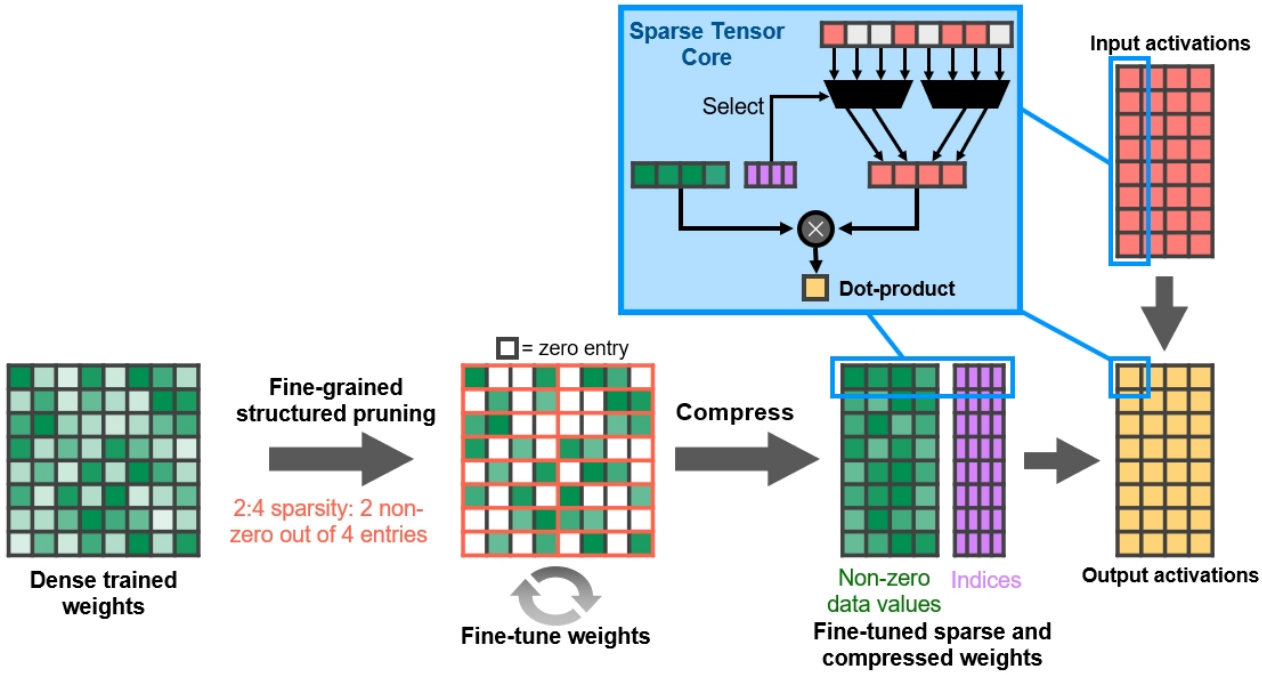
Fine-Grained Structured Sparsity

With the A100 GPU, NVIDIA introduced Fine-Grained Structured Sparsity, a novel approach which doubles compute throughput for deep neural networks. This feature is also supported on GA10x GPUs and helps accelerate certain AI-based graphics inference workloads.

An in-depth description of the implementation of Fine-Grained Structured Sparsity in the Ampere GPU architecture and a primer on Sparsity is available in the NVIDIA A100 Tensor Core GPU whitepaper.

Sparsity is possible in deep learning because the importance of individual weights evolves during the learning process, and by the end of network training, only a subset of weights have acquired a meaningful purpose in determining the learned output. The remaining weights are no longer needed.

Fine grained structured sparsity imposes a constraint on the allowed sparsity pattern, making it more efficient for hardware to do the necessary alignment of input operands. NVIDIA engineers have found that because deep learning networks are able to adapt weights during the training process based on training feedback, in general the structure constraint does not impact the accuracy of the trained network for inferencing. This enables inferencing acceleration with sparsity.



Fine-Grained Structured Sparsity prunes trained weights with a 2-out-of-4 non-zero pattern, followed by a simple and universal recipe for fine-tuning the non-zero weights. The weights are compressed for a 2x reduction in data footprint and bandwidth, and the Sparse Tensor Core operations double math throughput by skipping the zeros.

Figure 13. Fine-Grained Structured Sparsity

NVIDIA has developed a simple and universal recipe for sparsifying deep neural networks for inference using a 2:4 structured sparsity pattern. The network is first trained using dense weights, then fine-grained structured pruning is applied, and finally the remaining non-zero weights are fine-tuned with additional training steps. This method results in virtually no loss in inferencing accuracy based on evaluation across dozens of networks spanning vision, object detection, segmentation, natural language modeling, and translation.

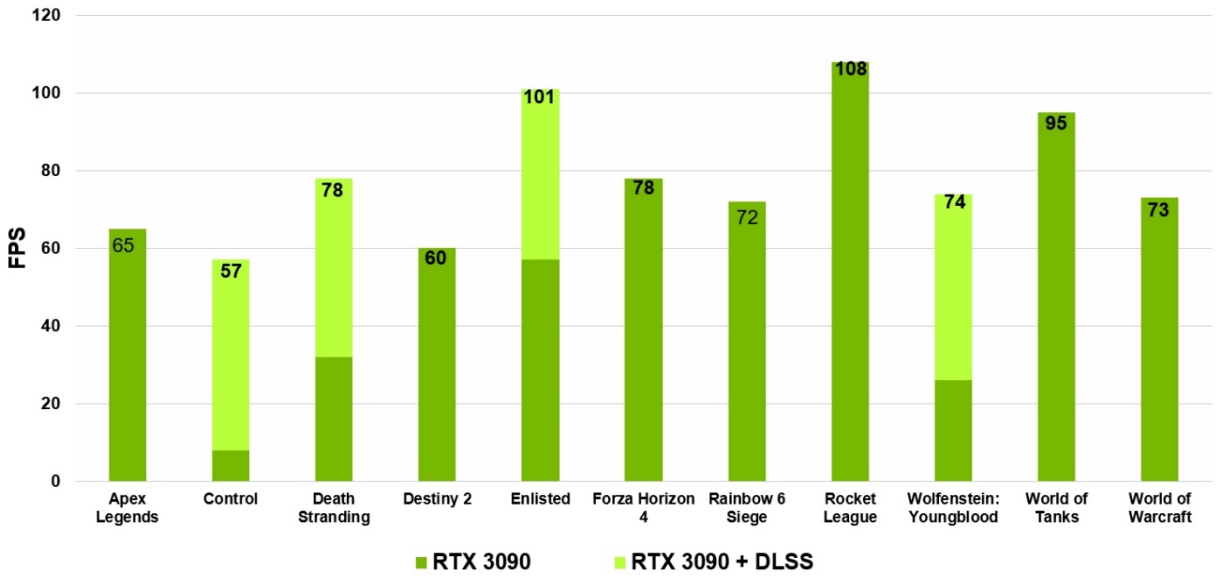
NVIDIA DLSS 8K

Rendering a ray-traced image in real-time with high frame rates is an extremely computationally expensive process. Before the introduction of NVIDIA's Turing GPU Architecture, it was considered to be years away from becoming a reality. To help solve this problem, NVIDIA created Deep Learning Super Sampling (DLSS). Powered by the Tensor Cores in GeForce RTX GPUs, DLSS leverages a deep neural network to extract multidimensional features of the rendered scene, and intelligently combine details from multiple frames to construct a high-quality final image that looks comparable to native resolution, while delivering higher performance.

Note the sharper text and detail provided by 8K DLSS.

Figure 14. Watch Dogs: Legion with 8K DLSS compared to 4K and 1080p resolution.

DLSS is further enhanced on NVIDIA Ampere architecture GPUs by leveraging the performance of the third-generation Tensor Cores and a new 9x Super Resolution scaling factor that makes 8K gaming with 60 fps frame rates feasible for the first time on ray-traced games.



The GeForce RTX 3090 can deliver 60 fps gaming in many titles at 8K / 8K with DLSS. Games used high graphics settings and ray tracing enabled where applicable. Tested with Core i9-10900K CPU.

Figure 15. Built for 8K Gaming

For more details on DLSS, please read this article on the NVIDIA website.

GDDR6X Memory

The GA10x memory subsystem utilizes new high-speed GDDR6X memory. NVIDIA worked closely with the DRAM industry and Micron Technology to develop the world's first GPUs that use GDDR6X. Today's PC games and creative applications require significantly more memory bandwidth to handle increasingly complex scene geometries, multiple graphics buffers, larger and more detailed textures, intensive ray tracing and AI inference operations, and of course shading and displaying more pixels at higher resolutions and frame rates.

GDDR6X is the next big advance in high-bandwidth GDDR DRAM memory design. GDDR6X preserves the same data access granularity and memory module size as the widely accepted and high-performance GDDR6 memory standard introduced in 2018, but improves data rate and transfer efficiency in many ways.

GDDR6X is the first consumer GPU graphics memory to reach memory bandwidth over 900 GB/sec. To achieve this breakthrough, innovative signal transmission technology and four-level pulse amplitude modulation (PAM4) completely redefines how the memory subsystem moves data. By using PAM4 multilevel signaling techniques, GDDR6X transfers more data and at a much faster rate, moving two bits of information at a time, doubling the I/O data rate of the previous PAM2/NRZ signaling scheme. Data-hungry workloads, such as AI inference, game ray tracing, and 8K video rendering, can now be fed data at high rates, opening new opportunities for computing and new end-user experiences.

Figure 16 shows "data eye" comparisons between GDDR6 (left) and GDDR6X (right). The same amount of data can be transferred across the GDDR6X interface at half the frequency compared to GDDR6. Alternatively, GDDR6X can double the effective bandwidth compared to GDDR6 at a given operating frequency.

PAM4 signaling is a big upgrade from the two-level NRZ signaling on GDDR6 memory. Instead of transmitting two binary bits of data each clock cycle (one bit on the rising edge and one bit on the falling edge of the clock), PAM4 sends two bits each clock edge, encoded using four different voltage levels. The voltage levels are divided into 250 mV steps with each level representing two bits of data - 00, 01, 10, or 11 sent on each clock edge (still DDR technology).

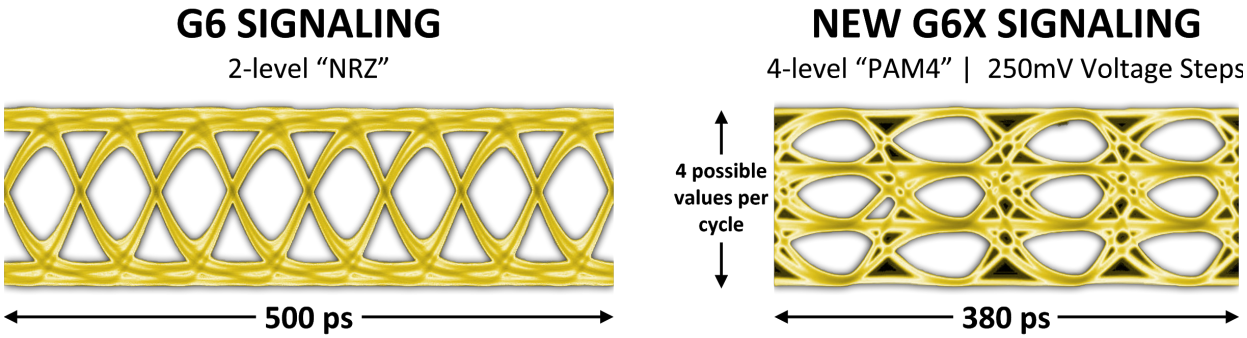


Figure 16. GDDR6X Improved Performance and Efficiency using PAM4 Signaling

To address the SNR challenges introduced with PAM4 signaling, a new encoding scheme entitled MTA (Maximum Transition Avoidance – see Figure 17) was developed to limit transitions on the high-speed signals. MTA prevents signals transitioning from the highest to lowest level and vice versa, which improves the interface SNR. This is achieved by having a part of the data burst for each pin in a byte transmitted on the encoding pin (time interleaved), and then having the remaining portion of the data burst mapped to a sequence devoid of maximum transitions using judiciously chosen codewords. In addition, new interface trainings, adaptations, and equalization schemes were introduced. Finally, the package and PCB designs required careful planning and comprehensive signal and power integrity analyses to achieve the higher data rates.

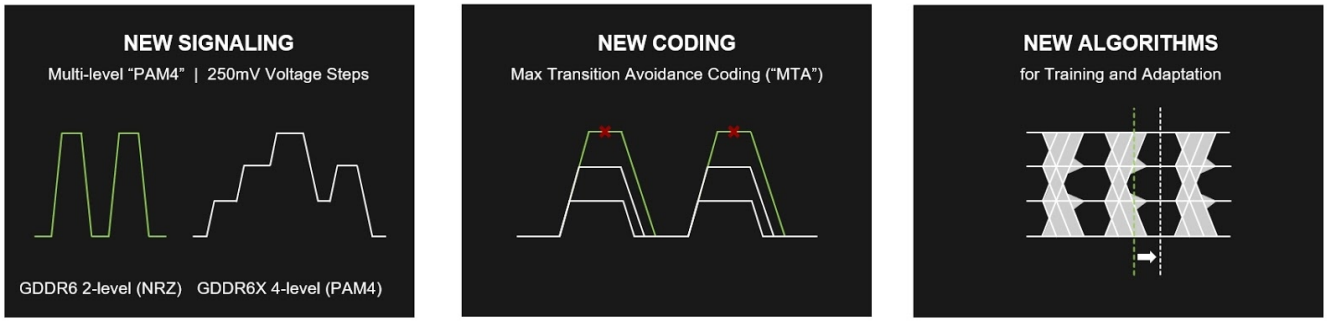


Figure 17. GDDR6X New Signaling, New Coding, New Algorithms

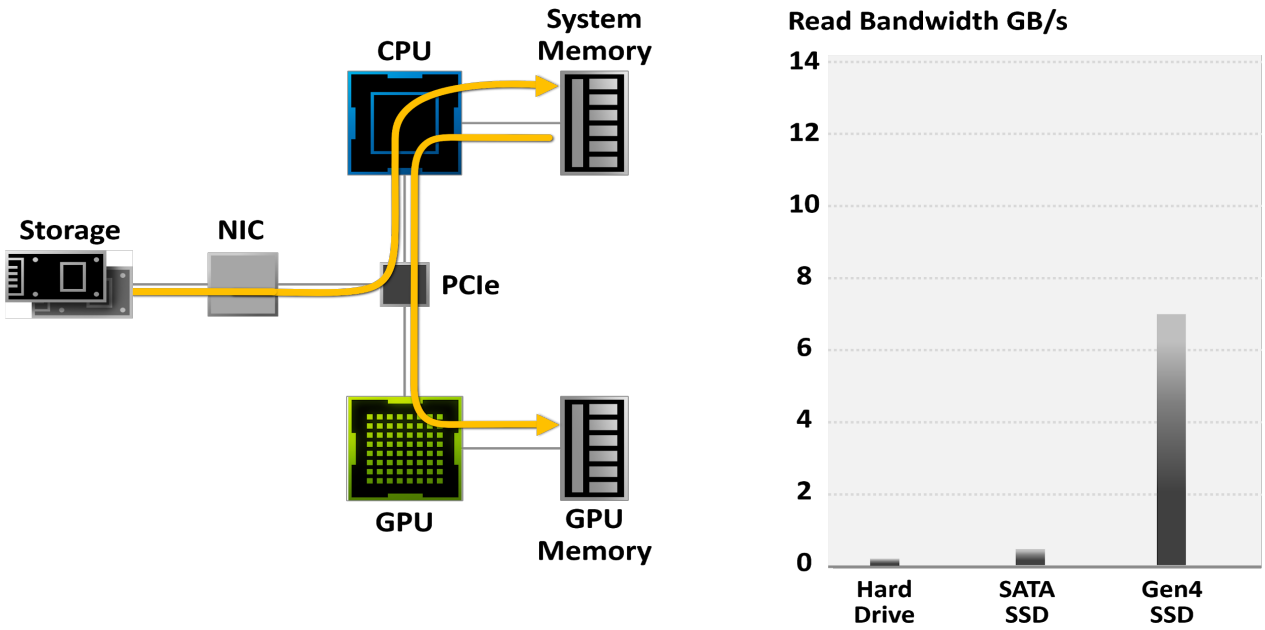
Supporting data rates up to 19.5 Gbps on GA10x GPUs, GDDR6X delivers up to 936 GB/sec of peak memory bandwidth, an improvement of over 52% compared to the TU102 GPU used in the GeForce RTX 2080 Ti. The entire memory subsystem has been optimized and carefully crafted to meet the demanding standards that are required for such high frequency operation.

Now featuring pseudo-independent memory channels, GDDR6X helps with BVH ray traversal in RT Cores. With up to 64% more bandwidth than previous GPU generations, GDDR6X is the biggest generational leap in bandwidth in 10 years, since the GeForce 200 series GPUs.

RTX IO

Today's games are giant worlds. With technology advances like photogrammetry, games are increasingly mimicking real life, and as a result they are continuing to grow in size. The largest games are over 200GB, which is 3x larger than four years ago, and their storage footprints will only continue to expand with the next generation.

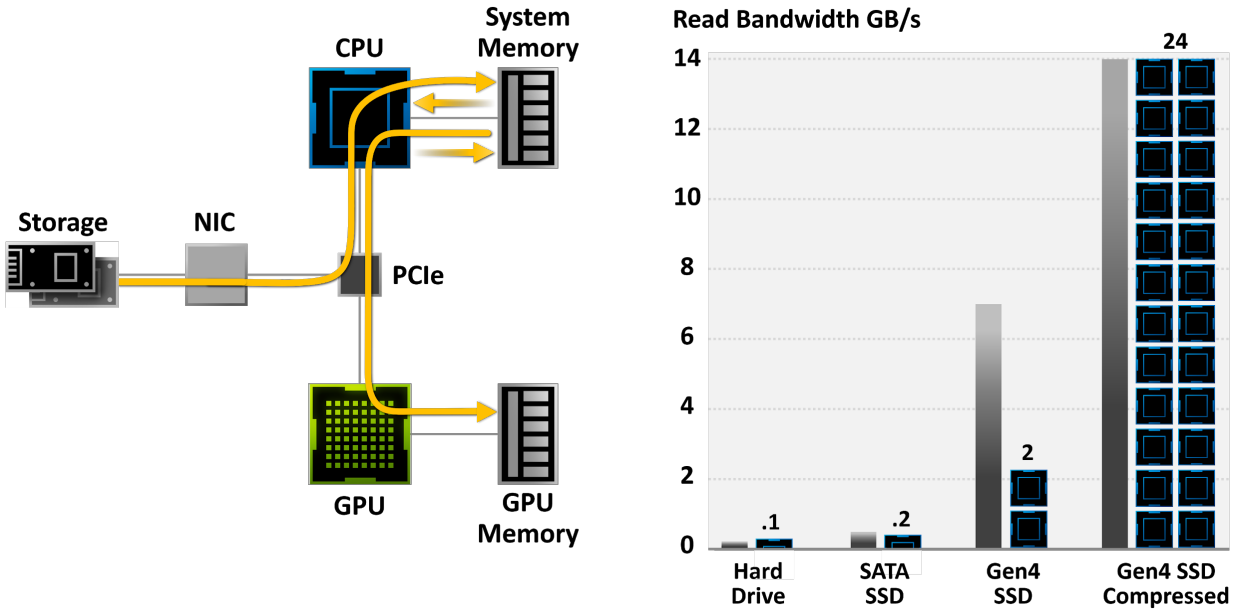
As storage sizes have grown, so has storage performance. Gamers are increasingly turning to SSDs to reduce game load times: while hard drives are limited to 50-100 MB/sec throughput, the latest M.2 PCIe Gen4 SSDs deliver up to 7 GB/sec.



With the traditional storage model, game data is read from the hard disk, then passed from the system memory and CPU before being passed to the GPU.

Figure 18. Games Bottlenecked by Traditional I/O

Historically games have read files from the hard disk, using the CPU to decompress the game image. Developers have used lossless compression to reduce install sizes and to improve I/O performance. However, as storage performance has increased, traditional file systems and storage APIs have become a bottleneck. For example, decompressing game data from a 100 MB/sec hard drive takes only a few CPU cores, but decompressing data from a 7 GB/sec PCIe Gen4 SSD can consume more than twenty AMD Ryzen™ Threadripper™ 3960X CPU cores!



Using the traditional storage model, game decompression can consume all 24 cores on a Threadripper CPU. Modern game engines have exceeded the capability of traditional storage APIs. A new generation of I/O architecture is needed. Data transfer rates are the gray bars, CPU cores required are the black/blue blocks.

Figure 19. Compressed Data Needed, but CPU Cannot Keep Up

Introducing NVIDIA RTX IO

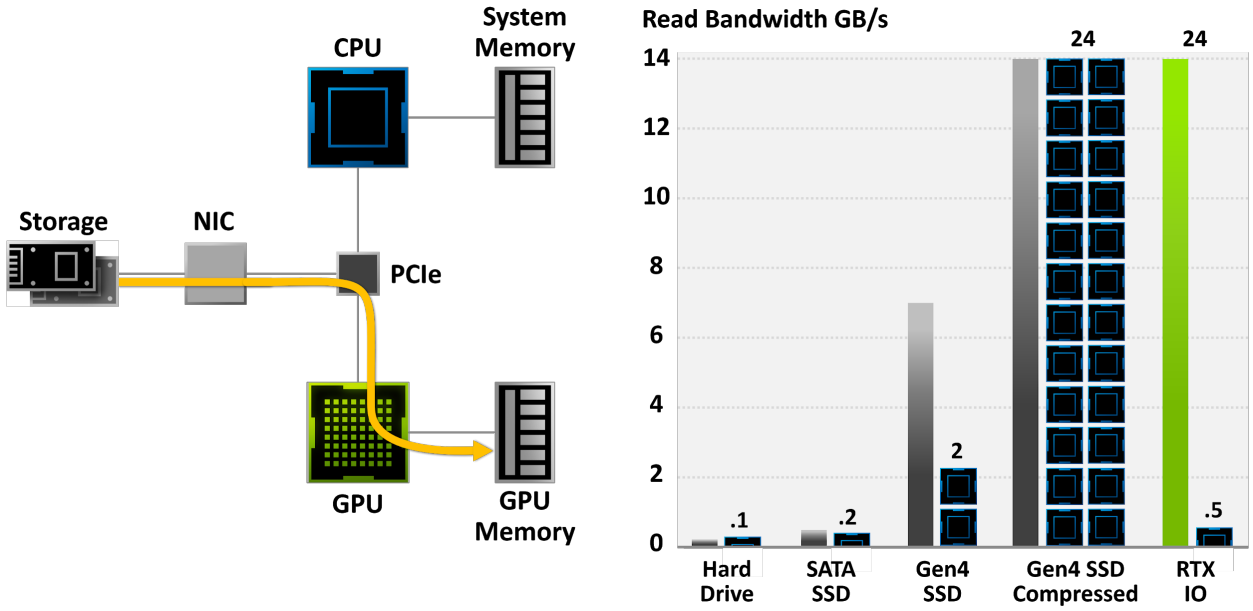
NVIDIA RTX IO is a suite of technologies that enable rapid GPU-based loading and decompression of game assets, accelerating I/O performance by up to 100x compared to hard drives and traditional storage APIs. When used with Microsoft's new DirectStorage for Windows API, RTX IO offloads dozens of CPU cores' worth of work to your RTX GPU, improving frame rates, enabling near-instantaneous game loading, and opening the door to a new era of large, incredibly detailed open world games.

Object pop-in and stutter can be reduced, and high-quality textures can be streamed at incredible rates, so even if you're speeding through a world, everything runs and looks great. In addition, with lossless compression, game download and install sizes can be reduced, allowing gamers to store more games on their SSD while also improving their performance.

How NVIDIA RTX IO Works

NVIDIA RTX IO plugs into Microsoft's upcoming DirectStorage API which is a next-generation storage architecture designed specifically for state-of-the-art NVMe SSD-equipped gaming PCs and the complex workloads that modern games require. Together, streamlined and parallelized APIs specifically tailored for games allow dramatically reduced IO overhead, and maximize performance / bandwidth from NVMe SSDs to your RTX IO-enabled GPU.

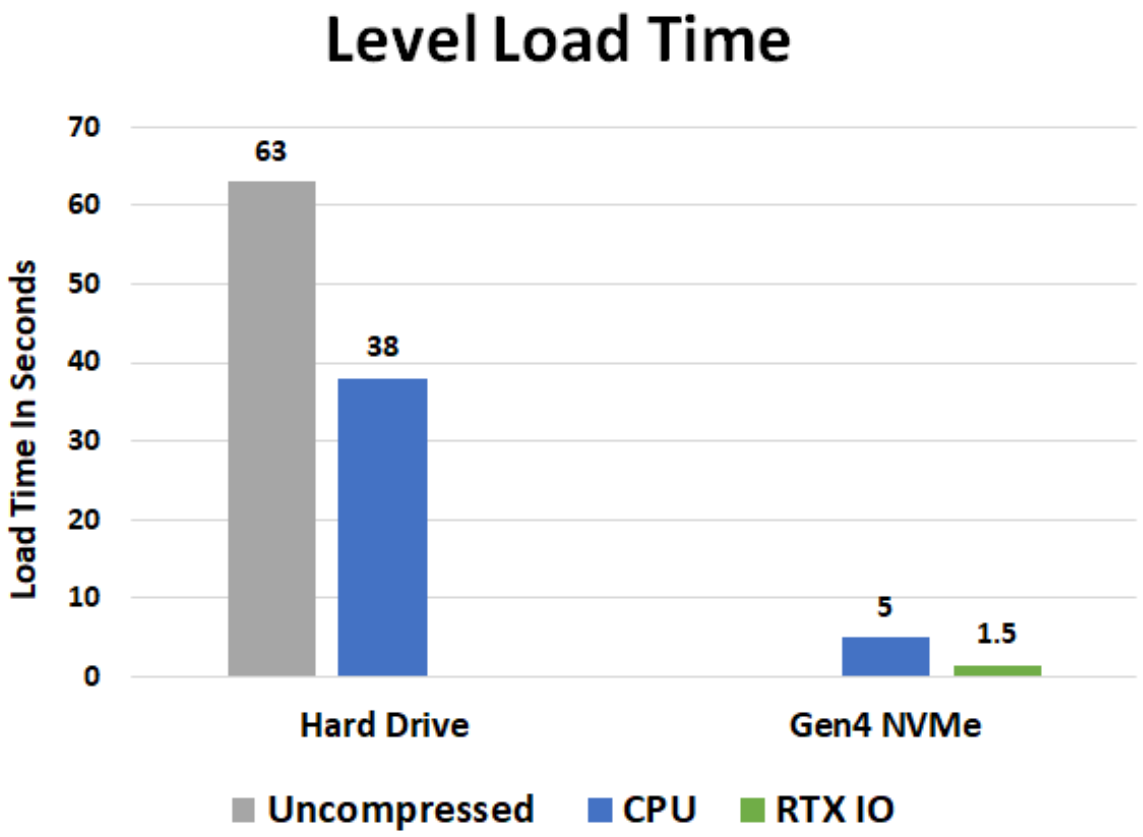
Specifically, NVIDIA RTX IO brings GPU-based lossless decompression, allowing reads through DirectStorage to remain compressed and delivered to the GPU for decompression. This removes the load from the CPU, moving the data from storage to the GPU in a more efficient, compressed form, and improving I/O performance by a factor of two.



Data transfer rates are the gray and green bars, CPU cores required are the black/blue blocks.

Figure 20. RTX IO Delivers 100X Throughput, 20X Lower CPU Utilization

GeForce RTX GPUs will deliver decompression performance beyond the limits of even Gen4 SSDs, offloading potentially dozens of CPU cores' worth of work to ensure maximum overall system performance for next-generation games. Lossless decompression is implemented with high performance compute kernels, asynchronously scheduled. This functionality leverages the DMA and copy engines of Turing and Ampere, as well as the advanced instruction set, and architecture of these GPU's SM's. The advantage of this is that the enormous compute power of the GPU can be leveraged for burst or bulk loading (at level load for example) when GPU resources can be leveraged as a high performance I/O processor, delivering decompression performance well beyond the limits of Gen4 NVMe. During streaming scenarios, bandwidths are a tiny fraction of the GPU capability, further leveraging the advanced asynchronous compute capabilities of Turing and Ampere.



Load test running on 24-core Threadripper 3960x platform, prototype Gen4 NVMe m.2 SSD, alpha software.

Figure 21. Level Load Time Comparison

Microsoft is targeting a developer preview of DirectStorage for Windows for game developers next year, and NVIDIA Turing & Ampere gamers will be able to take advantage of RTX IO-enhanced games as soon as they become available.

More information on NVIDIA RTX IO can be found here.

Display and Video Engine

DisplayPort 1.4a with DSC 1.2a

The march towards ever higher resolutions and faster refresh-rate displays continues, and NVIDIA Ampere architecture GPUs include support for both. Gamers and hardware enthusiasts who crave the smoothest possible experience can now game on 4K (3820 x 2160) displays at 120Hz and 8K (7680 x 4320) displays at 60Hz—four times more pixels than 4K.

The display engine that powers Ampere architecture GPUs is designed to support many of the new technologies included with today's fastest display interfaces. This includes DisplayPort 1.4a allowing 8K resolution at 60 Hz with VESA's Display Stream Compression (DSC) 1.2a technology, providing higher compression that is visually lossless. Ampere architecture GPUs can drive two 8K displays at 60Hz with one cable for each display.

Table 5. DisplayPort Versions - Spec Comparison

	Max Bandwidth	Bandwidth/ Lane	Max Resolution Supported
DisplayPort 1.2	21.6 Gbps	5.4 Gbps	4K @ 60Hz
DisplayPort 1.3	32.4 Gbps	8.1 Gbps	8K @ 60Hz[1] 4K @ 120Hz
DisplayPort 1.4a	32.4 Gbps	8.1 Gbps	8K @ 60Hz + HDR[2] 4K @ 240Hz + HDR[2]

1. Using 4:2:0 pixel format, software support also required
2. Requires Display Stream Compression 1.2a (DSC) enabled

HDMI 2.1 with DSC 1.2a

NVIDIA Ampere architecture GPUs are the first discrete GPUs to provide support for HDMI 2.1, the most recent update of the HDMI specification. HDMI 2.1 adds support for a number of higher video resolutions and refresh rates including 8K60 and 4K120. Maximum bandwidth is increased to 48 Gbps, which allows for dynamic HDR formats as well. DSC 1.2a or 4:2:0 pixel format enabled is required to support 8K @ 60Hz with HDR.

Table 6 . HDMI Versions - Spec Comparison

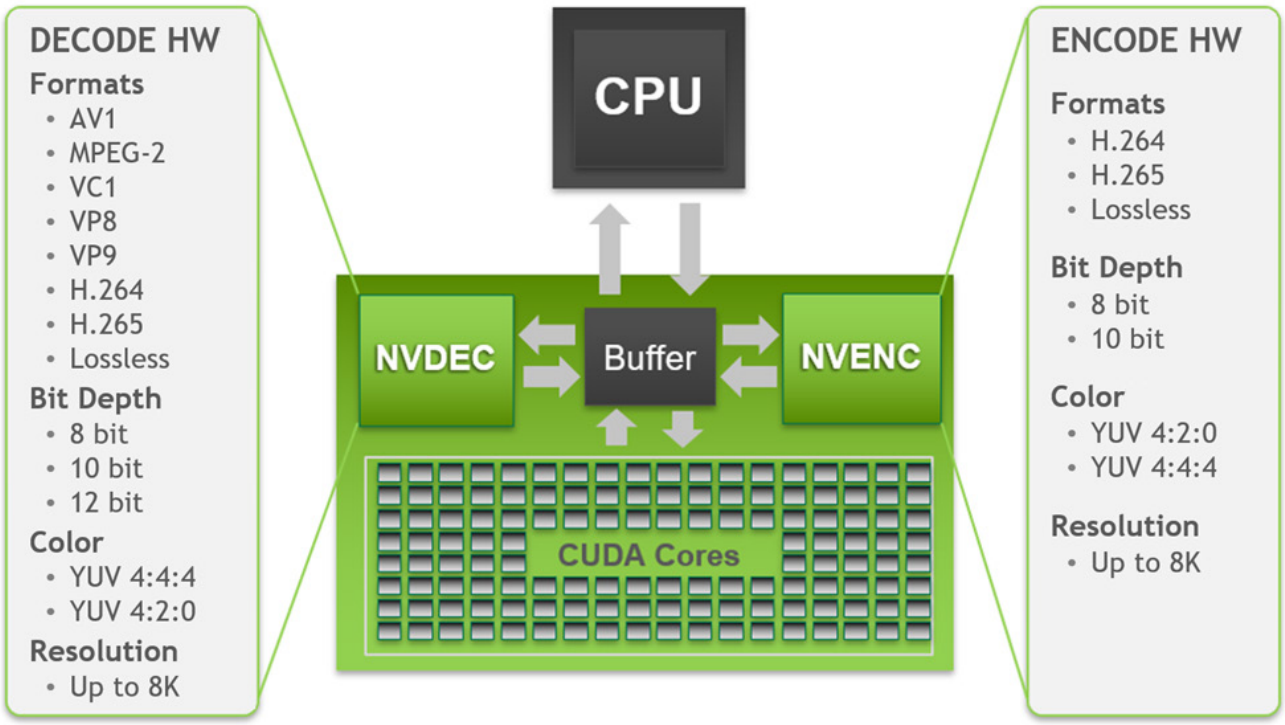
	Total Bandwidth	Bandwidth/ Channe	Max Resolution Supported
HDMI 1.4	10.2 Gbps	3.4 Gbps	4K @ 30Hz
HDMI 2.0b	18 Gbps	6 Gbps	4K @ 60Hz 8K @ 30Hz[1]
HDMI 2.1	48 Gbps	12 Gbps	4K @ 240Hz + HDR[2] 8K @ 60Hz + HDR[2]

1. Using 4:2:0 pixel format
2. Requires Display Stream Compression 1.2a (DSC) or 4:2:0 pixel format enabled

The Ampere GPU architecture supports both HDCP (High-bandwidth Digital Content Protection) 2.3 and HDCP 1.x. HDCP 2.3 is designed to work with both DisplayPort and HDMI.

Fifth Generation NVDEC - Hardware-Accelerated Video Decoding

NVIDIA GPUs contain a fifth-generation hardware-based decoder (referred to as NVDEC) that provides fully-accelerated hardware-based video decoding for several popular codecs. With complete decoding offloaded to NVDEC, the graphics engine and the CPU are free for other operations. NVDEC supports much faster than real-time decoding which makes it suitable to be used for transcoding applications, in addition to video playback applications.



Note: 4:2:2 is not natively supported on HW. 8K resolution support is codec dependent

Figure 22. Video Decode and Encode Formats Supported on GA10x GPUs

The NVDECODE API enables software developers to configure the NVDEC decoder.

The Fifth-Generation NVIDIA decoder in GA10x supports hardware-accelerated decoding of the following video codecs on Windows and Linux platforms: MPEG-2, VC-1, H.264 (AVCHD), H.265 (HEVC), VP8, VP9, and AV1.

NVIDIA is the first GPU vendor to support AV1 decode shipping in hardware (see *AV1 Decode* below).

The following chroma subsampling / color modes and bit depths are supported for these popular codecs for decoding:

- HEVC 4:2:0 and 4:4:4 color modes with 8/10/12 bit depths are supported.
- VP9 4:2:0 color mode with 8/10/12 bit depths are supported.
- H.264 4:2:0 color mode with 8 bit depth is supported.

AV1 Hardware Decode

AV1 (AOMedia Video 1) is an open, royalty-free video coding format developed by AOM (Alliance for Open Media) that is designed primarily for video transmissions over the Internet. GA10x GPUs are the first GPUs to provide AV1 hardware decode support.

AV1 provides better compression and quality compared to existing codecs like H.264, HEVC, and VP9, and is being adopted by many of the top video platforms and browsers. AV1 will generally provide 50-55% bitrate savings over H.264.

While AV1 is very efficient at compressing video, decoding it is very computationally intensive. Current software decoders cause high CPU utilization, and struggle to play ultra resolutions. In our testing, an Intel i9 9900K CPU averaged 28 FPS playback with an 8K60 HDR video on YouTube, and CPU utilization was above 85%. GA10x GPUs can play AV1 by offloading the decoding to NVDEC, which is capable of playing up to 8K60 HDR content with very low CPU usage (~4% on the same CPU as the previous test).

AV1 decode in GA10x GPUs include the following features:

- AV1 Profile 0 - monochrome / 4:2:0, 8/10 bit decode support
- Up to level 6.0 (exclude large scale tile).
- Maximum supported resolution of 8192x8192 and minimum supported resolution is 128x128.
- Support for histogram collection, film grain synthesis, and sub sample map (SSM).
- 8K @ 60 HW decode
- Supported path includes DX9, DX11, and DX12.

Seventh Generation NVENC - Hardware-Accelerated Video Encoding

NVIDIA GPUs contain a hardware-based encoder (referred to as NVENC) that provides fully accelerated hardware-based video encoding, and is independent of graphics performance. Video encoding can be a computationally complex task, and with encoding offloaded to NVENC, the graphics engine and the CPU are free for other operations. For example, in a game recording and streaming scenario like streaming to Twitch.tv using Open Broadcaster Software (OBS), offloading video encoding to NVENC allows the GPU's graphics engine to be dedicated to game rendering and the CPU free for other tasks.

NVENC makes it possible to:

- Encode and stream games and applications at high quality and ultra-low latency without utilizing CPU
- Encode at very high quality for archiving, OTT streaming, Web videos
- Encode with ultra-low power consumption per stream (Watts/stream)

GA10x GPUs include the seventh generation NVENC encoder unit that was introduced with the Turing architecture. With common Twitch and YouTube streaming settings, NVENC-based hardware encoding in GA10x GPUs exceeds the encoding quality of software-based x264 encoders using the Fast preset and is on par with x264 Medium, a preset that typically requires a dual PC setup. This dramatically lowers CPU utilization. 4K encoding is too heavy a workload for a typical CPU setup, but the GA10x NVENC encoder makes high resolution encoding seamless up to 4K on H.264, and even 8K on HEVC.

.

Conclusion

With every new GPU architecture, NVIDIA strives to provide groundbreaking performance for the next generation, while introducing new features that enhance image quality. Turing was the first GPU to introduce hardware-accelerated ray tracing, a feature that was once considered to be the holy grail of computer graphics, and years away from becoming a reality. Today, incredibly realistic and physically accurate ray tracing effects are being added to many new AAA PC games, and GPU-accelerated ray tracing is considered a must-have feature by many PC gamers. The new NVIDIA GA10x Ampere architecture GPUs provide the necessary features and performance to enjoy these new ray-traced games at up to 2x higher frame rates. Another Turing innovation - advanced GPU-accelerated AI processing that enhances games, rendering, and other graphics applications - is taken to the next level with NVIDIA GA10x Ampere architecture GPUs.

The NVIDIA GA10x Ampere architecture provides up to 2x performance improvements in programmable shading, ray tracing, and artificial intelligence. In the GA10x SM, 2x FP32 processing significantly improves performance for the most common graphics operations. The benefits also extend to compute and ray tracing applications.

GeForce GA10x GPUs and RTX 30-series graphics boards incorporate NVIDIA's second-generation RT Core. The second-generation RT Core doubles ray/triangle intersection performance compared to the first generation. In addition, the new RT Core also supports acceleration for ray-traced motion blur. Finally, support for concurrent ray tracing and shading, or ray tracing and compute, provides a competitive advantage that no other GPU can match.

The GA10x Ampere architecture also features NVIDIA's third-generation Tensor Core which implements structural sparsity, capable of doubling the Tensor Core's effective throughput. With enhancements made to NVIDIA DLSS, the GeForce RTX 3090 will introduce gamers to 8K HDR rendering for the first time.

Working closely with Micron Technology, the GA10x Ampere architecture is the first GPU to support GDDR6X memory. GDDR6X is the next big advance in graphics memory. Utilizing PAM4 signaling we are able to achieve memory data rates as high as 19.5 Gbps, yielding 936 GB/sec of peak memory bandwidth in the GeForce RTX 3090.

NVIDIA RTX IO addresses the bottleneck caused by today's outdated file systems and storage standards. RTX IO provides new APIs for fast loading and streaming directly from the SSD to the GPU's memory, as well as support for GPU-based loading and decompression of game assets. Working in conjunction with Microsoft's new DirectStorage API for Windows, RTX IO offloads work that was previously handled by potentially dozens of CPU cores to the GeForce RTX GPU, improving frame rates, enabling near-instantaneous game loading, and opening the door to a new era of large, incredibly detailed open world games.

The NVIDIA GA10x Ampere architecture is our greatest generational leap ever. By providing up to 2x performance advancements in programmable shading, ray tracing, and AI, GeForce RTX 30-Series GPUs are able to deliver photorealistic ray-traced graphics at high frame rates, enabling amazing PC gaming experiences.

Appendix A - Additional GeForce GA10x GPU Specifications

GeForce RTX 3090

The GeForce RTX 3090 incorporates the GA102 GPU and is built for creators, data scientists, AI developers, and extreme gamers who want the fastest performing graphics card in the world.

The GeForce RTX 3090 is a beast. Like the previous generation TITAN RTX, it ships with 24 GB of memory, allowing data scientists to process large data sets, while gamers can experience next-generation gaming with ray tracing and DLSS 8K. The RTX 3090 is 1.5x faster than TITAN RTX and ships with 82 SMs, 10496 CUDA Cores, and 24 GB of GDDR6X memory running at 19.5 Gbps.

Table 7. Comparison of GeForce RTX 3090 to NVIDIA Titan RTX

Graphics Card	NVIDIA TITAN RTX	RTX 3090 FE Founders Edition
GPU Codename	TU102	GA102
GPU Architecture	NVIDIA Turing	NVIDIA Ampere
GPCs	6	7
TPCs	36	41
SMs	72	82
CUDA Cores / SM	64	128
CUDA Cores / GPU	4608	10496
Tensor Cores / SM	8 (2nd Gen)	4 (3rd Gen)
Tensor Cores / GPU	576 (2nd Gen)	328 (3rd Gen)
RT Cores	72 (1st Gen)	82 (2nd Gen)
GPU Boost Clock (MHz)	1770	1695
Peak FP32 TFLOPS (non-Tensor)[1]	16.3	35.6
Peak FP16 TFLOPS (non-Tensor)[1]	32.6	35.6
Peak BF16 TFLOPS (non-Tensor)[1]	NA	35.6
Peak INT32 TOPS (non-Tensor)[1,3]	16.3	17.8
Peak FP16 Tensor TFLOPS with FP16 Accumulate[1]	130.5	142/284[2]
Peak FP16 Tensor TFLOPS with FP32 Accumulate[1]	65.2	71/142[2]
Peak BF16 Tensor TFLOPS with FP32 Accumulate[1]	NA	71/142[2]
Peak TF32 Tensor TFLOPS[1]	NA	35.6/71[2]

Peak INT8 Tensor TOPS[1]	261	284/568[2]
Peak INT4 Tensor TOPS[1]	522	568/1136[2]
Frame Buffer Memory Size and Type	24576MB GDDR6	24576 MB GDDR6X
Memory Interface	384-bit	384-bit
Memory Clock (Data Rate)	14 Gbps	19.5 Gbps
Memory Bandwidth (GB/sec)	672 GB/sec	936 GB/sec
ROPs	96	112
Pixel Fill-rate (Gigapixels/sec)	169.9	193
Texture Units	288	328
Texel Fill-rate (Gigatexels/sec)	509.8	566
L1 Data Cache/Shared Memory	6912 KB	10496 KB
L2 Cache Size	6144 KB	6144 KB
Register File Size	18432 KB	20992 KB
Total Graphics Power (TGP)	280 Watts	350 Watts
Transistor Count	18.6 Billion	28.3 Billion
Die Size	754 mm^2	628.4 mm^2
Manufacturing Process	TSMC 12 nm FFN (FinFET NVIDIA)	Samsung 8 nm 8N NVIDIA Custom Process

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

GeForce RTX 3070

The GeForce RTX 3070 incorporates NVIDIA's GA104 GPU. This GPU is designed to deliver the best performance and energy efficiency in its class. In fact, the RTX 3070 delivers performance that is faster than the former flagship GeForce RTX 2080 Ti GPU from the previous generation! GA104 retains most of the key new features that were added to NVIDIA's GA10x Ampere GPU Architecture and ships with GDDR6 memory.

Figure 23. GA104 Full GPU with 48 SMs

Table 8. Comparison of GeForce RTX 3070 to GeForce RTX 2070 Super

Graphics Card	RTX 2070 Super Founders Edition	RTX 3070 Founders Edition
GPU Codename	TU104	GA104
GPU Architecture	NVIDIA Turing	NVIDIA Ampere
GPCs	5 or 6	6
TPCs	20	23
SMs	40	46
CUDA Cores / SM	64	128
CUDA Cores / GPU	2560	5888
Tensor Cores / SM	8 (2nd Gen)	4 (3rd Gen)
Tensor Cores / GPU	320 (2nd Gen)	184 (3rd Gen)
RT Cores	40 (1st Gen)	46 (2nd Gen)
GPU Boost Clock (MHz)	1770	1725
Peak FP32 TFLOPS (non-Tensor)[1]	9.1	20.3
Peak FP16 TFLOPS (non-Tensor)[1]	18.1	20.3
Peak BF16 TFLOPS (non-Tensor)[1]	NA	20.3
Peak INT32 TOPS (non-Tensor)[1,3]	9.1	10.2
Peak FP16 Tensor TFLOPS with FP16 Accumulate[1]	72.5	81.3/162.6[2]
Peak FP16 Tensor TFLOPS with FP32 Accumulate[1]	36.3	40.6/81.3[2]
Peak BF16 Tensor TFLOPS with FP32 Accumulate[1]	NA	40.6/81.3[2]
Peak TF32 Tensor TFLOPS[1]	NA	20.3/40.6[2]
Peak INT8 Tensor TOPS[1]	145	162.6/325.2[2]
Peak INT4 Tensor TOPS[1]	290	325.2/650.4[2]
Frame Buffer Memory Size and Type	8192MB GDDR6	8192 MB GDDR6
Memory Interface	256-bit	256-bit
Memory Clock (Data Rate)	14 Gbps	14 Gbps
Memory Bandwidth	448 GB/sec	448 GB/sec
ROPs	64	96
Pixel Fill-rate (Gigapixels/sec)	113.3	165.6
Texture Units	160	184
Texel Fill-rate (Gigatexels/sec)	283.2	317.4
L1 Data Cache/Shared Memory	3840	5888

L2 Cache Size	4096 KB	4096 KB
Register File Size	10240	11776
TGP (Total Graphics Power)	215 Watts	220 Watts
Transistor Count	13.6 Billion	17.4 Billion
Die Size	545 mm^2	392.5 mm^2
Manufacturing Process	TSMC 12 nm FFN (FinFET NVIDIA)	Samsung 8 nm 8N NVIDIA Custom Process

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

Appendix B - New Memory Error Detection and Replay (EDR) Technology

Many GeForce users are hardware enthusiasts who like to overclock their graphics card beyond the reference GPU specifications. The resulting higher clock speeds improve performance, but the downside is it often requires users to push their card until it crashes or hangs. This procedure could also cause various other problems with the PC.

To address this issue, a new feature called Error Detection and Replay (EDR) has been added to the GDDR6X memory interface on GA10x GPUs. With EDR, the GPU's memory subsystem can detect when a data transmission error occurs. When the GDDR6X data link protection CRC check operation shows that there has been a data transmission error, the transmission in error is retried or "replayed" until a successful memory transaction occurs. As the user continues to push their GDDR6X memory overclock and cause more errors, the retried transactions reduce useful memory bandwidth, resulting in plateauing performance. This indicates that the user has hit the overclocking limits:

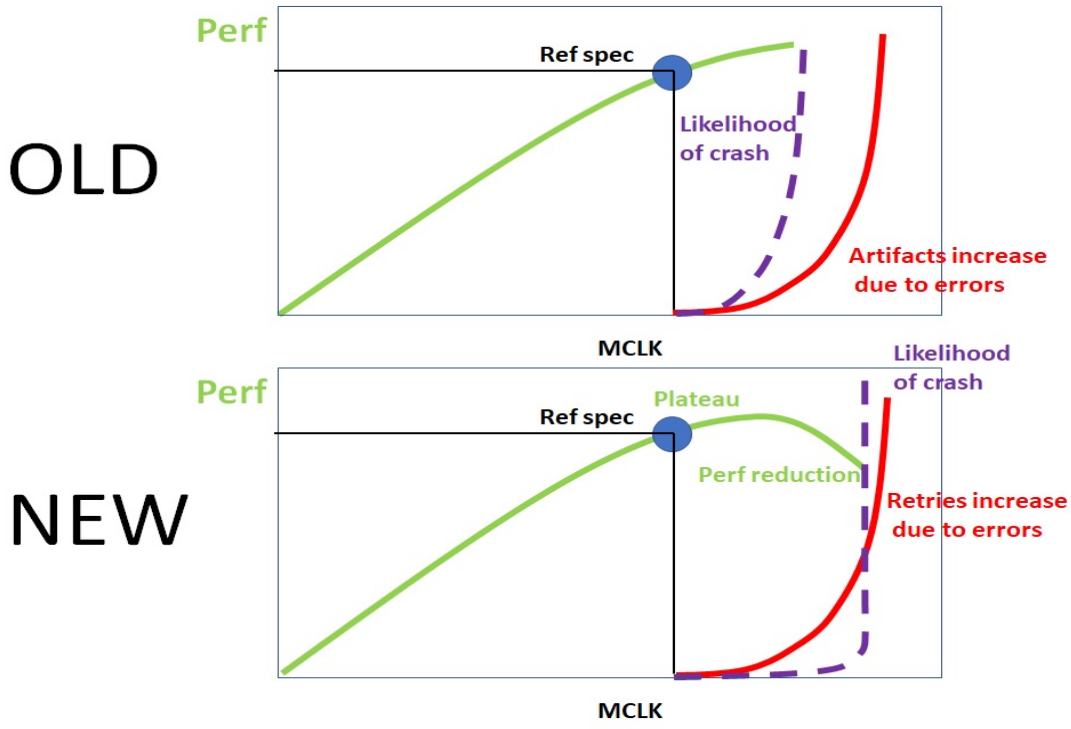


Figure 24. Old Overclocking Method vs Overclocking with EDR

As seen in Figure 24, with EDR, users no longer have to overclock their GPU's memory until it crashes - as soon as plateauing performance is observed, the user has reached the limits of the memory and should stop increasing clock frequency. Note that if the user continues to push the system, crashes will still occur. Additionally, EDR may not prevent all crashes before performance plateaus.

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 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.

Copyright