Whitepaper

NVIDIA GeForce GTX 1080

Gaming Perfected

Table of Contents

Introduction

The continuous advancement of high performance and fully programmable NVIDIA® graphics processing units (GPUs) has led to tremendous improvements in both 3D graphics and GPU-accelerated computing. This constant evolution of the GPU makes possible the beautiful graphics that consumers enjoy in today's games and films, in addition to enabling groundbreaking advances in Artificial Intelligence (AI), Deep Learning, autonomous driving systems, and numerous other compute-intensive applications.

Based on the revolutionary NVIDIA® Pascal™ GPU architecture first introduced in the high-end, datacenter-class GP100 GPU, NVIDIA's next Pascal GPU—GP104—is poised to usher in the next generation of DirectX 12 and Vulkan graphics; power the latest Virtual Reality (VR) headsets, games, and applications; and drive 4K, 5K, and HDR displays with incredible fidelity. The first graphics card to ship with the new GP104 GPU is the GeForce GTX® 1080.

Figure 1: GeForce GTX 1080 Founders Edition Graphics Card

As a world leader in visual computing, NVIDIA has pioneered numerous innovations in GPU hardware and software technologies. Pascal delivers tremendous speedups and improved PC gaming, and NVIDIA GameWorks software libraries enable developers to readily implement more interactive and cinematic experiences. The GeForce GTX 1080's performance combined with GameWorks libraries enables stunning visual effects, physical simulations, and VR experiences for PC gaming. The combined benefits of the new Pascal architecture and implementation, the 16 nm FinFET manufacturing process, and the latest GDDR5X memory technology give GeForce GTX 1080 a 70% performance lead over the prior generation GeForce GTX 980.

With 2560 CUDA Cores running at speeds over 1600 MHz in the GeForce GTX 1080, GP104 is the fastest gaming GPU in the world. Pascal is also the most efficient GPU architecture in the world: the GeForce GTX 1080 is 1.5x more power efficient than the GTX 980, and 3x more power efficient than the GeForce GTX 780!

Figure 2: GeForce 10 Series GPUs Deliver Outstanding Performance and Support New Gaming Features

The GeForce GTX 1080 not only allows gamers to experience their favorite games with richer details, better simulations, and higher frame rates than ever before, it will also deliver more consistent frame rates and a smoother gaming experience. NVIDIA has developed a number of new technologies that are designed to reduce stuttering and other distracting elements that can hinder the gaming experience. We have also developed new rendering techniques that are designed to reduce latency and improve the performance for Virtual Reality gaming. In this whitepaper you will learn about the new technologies NVIDIA has integrated into the Pascal architecture to make all of this possible.

Pascal Innovations

The demands on the GPU for enthusiast PC gaming have never been greater. Display resolutions are continuing to increase, with 4K and 5K displays requiring extremely powerful GPUs (or multiple GPUs) to maintain playable frame rates with high image quality. Virtual Reality headsets now demand GPUs deliver a sustained 90 fps—rendering to both eyes—with very low latency, to ensure an immersive experience that tracks closely with the user's movement. The GeForce GTX 1080 was designed with these new technologies in mind to usher in the latest wave of gaming experiences. Key highlights of the GeForce GTX 1080 include:

- **Cutting-Edge Pascal Architecture**
  The GeForce GTX 1080's Pascal architecture is the most efficient GPU design ever built. Comprised of 7.2 billion transistors and including 2560 single-precision CUDA Cores, the GeForce GTX 1080 is the world's fastest GPU. With an intense focus on craftsmanship in chip and board design, NVIDIA's engineering team achieved unprecedented results in frequency of operation and energy efficiency.
- **16 nm FinFET**
  The GeForce GTX 1080's GP104 GPU is fabricated using a new 16 nm FinFET manufacturing process that allows the chip to be built with more transistors, ultimately enabling new GPU features, higher performance, and improved power efficiency.
- **GDDR5X Memory**
  GDDR5X provides a significant memory bandwidth improvement over the GDDR5 memory that was used previously in NVIDIA's flagship GeForce GTX GPUs. Running at a data rate of 10 Gbps, the GeForce GTX 1080's 256-bit memory interface provides 43% more memory bandwidth than NVIDIA's prior GeForce GTX 980 GPU. Combined with architectural improvements in memory compression, the total effective memory bandwidth increase compared to GTX 980 is 1.7x.
- **Simultaneous Multi-Projection (SMP)**
  The field of display technology is undergoing significant changes from the days of a single, flat display monitor. Recognizing this trend, NVIDIA engineers developed a new Simultaneous Multi-Projection technology that for the first time enables the GPU to simultaneously map a single primitive onto up to sixteen different projections from the same viewpoint. Each projection can be either mono or stereo. This capability enables GeForce GTX 1080 to accurately match the curved projection required for VR displays, the multiple projection angles required for surround display setups, and other emerging display use cases.

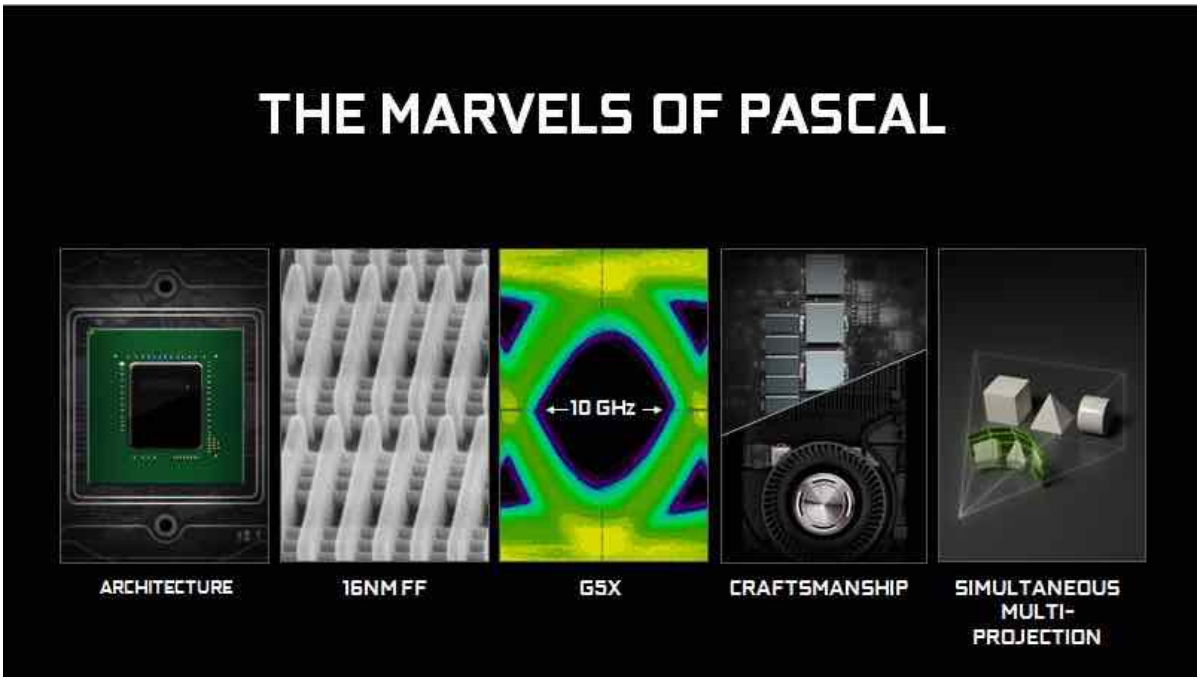


Figure 3: Key New Features of Pascal GPUs

GeForce GTX 1080 GPU Architecture In-Depth

Pascal GPUs are composed of different configurations of Graphics Processing Clusters (GPCs), Streaming Multiprocessors (SMs), and memory controllers. Each SM is paired with a PolyMorph Engine that handles vertex fetch, tessellation, viewport transformation, vertex attribute setup, and perspective correction. The GP104 PolyMorph Engine also includes a new Simultaneous Multi-Projection unit that will be described below. The combination of one SM plus one Polymorph Engine is referred to as a TPC. If you aren't familiar with the functions performed by the GPC and SM we suggest you first read the Fermi whitepaper.

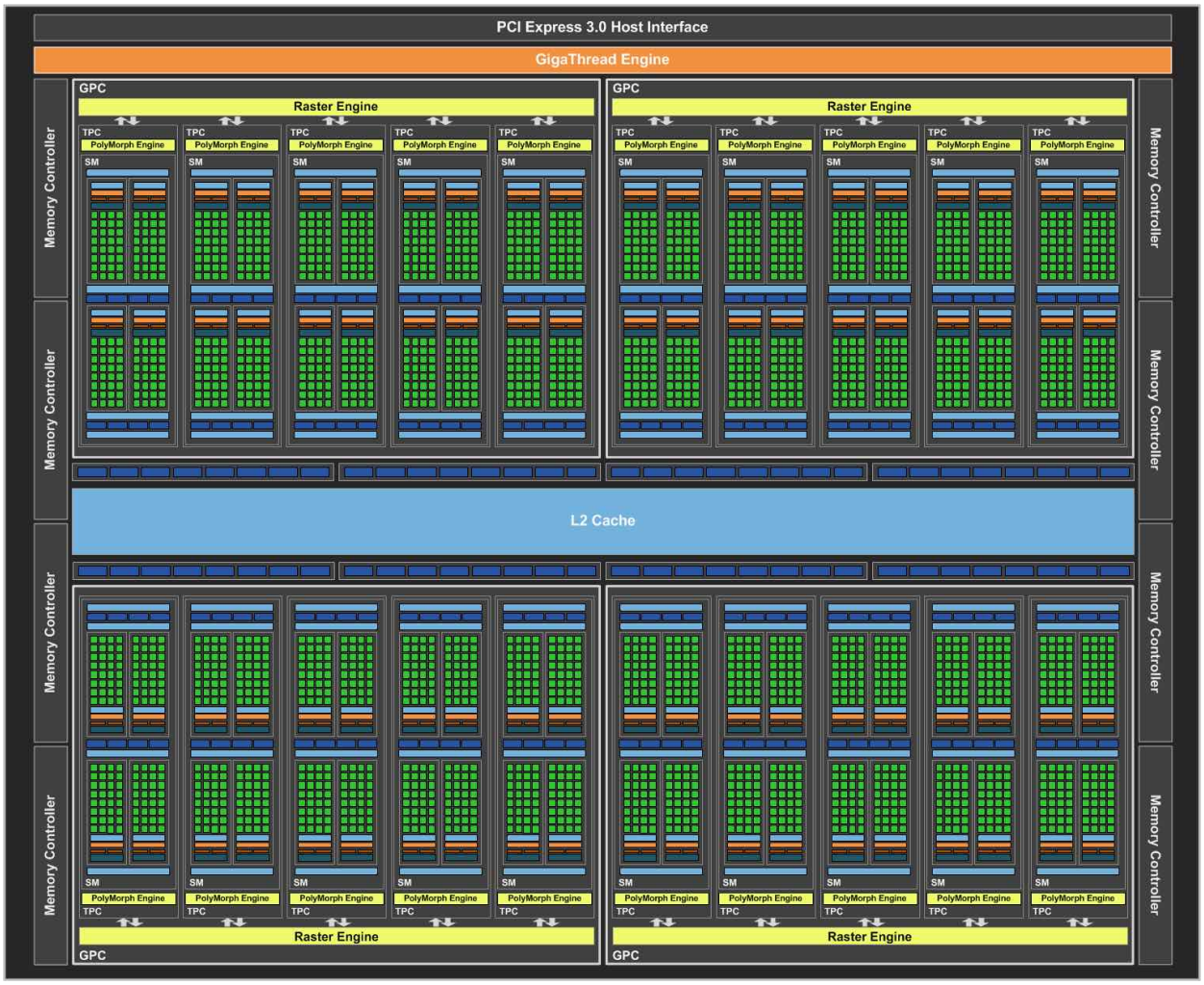


Figure 4: Block Diagram of the GP104 GPU

The GeForce GTX 1080 and its GP104 GPU consist of four GPCs, twenty Pascal Streaming Multiprocessors, and eight memory controllers. In the GeForce GTX 1080, each GPC ships with a dedicated raster engine and five SMs. Each SM contains 128 CUDA cores, 256 KB of register file capacity, a 96 KB shared memory unit, 48 KB of total L1 cache storage, and eight texture units.

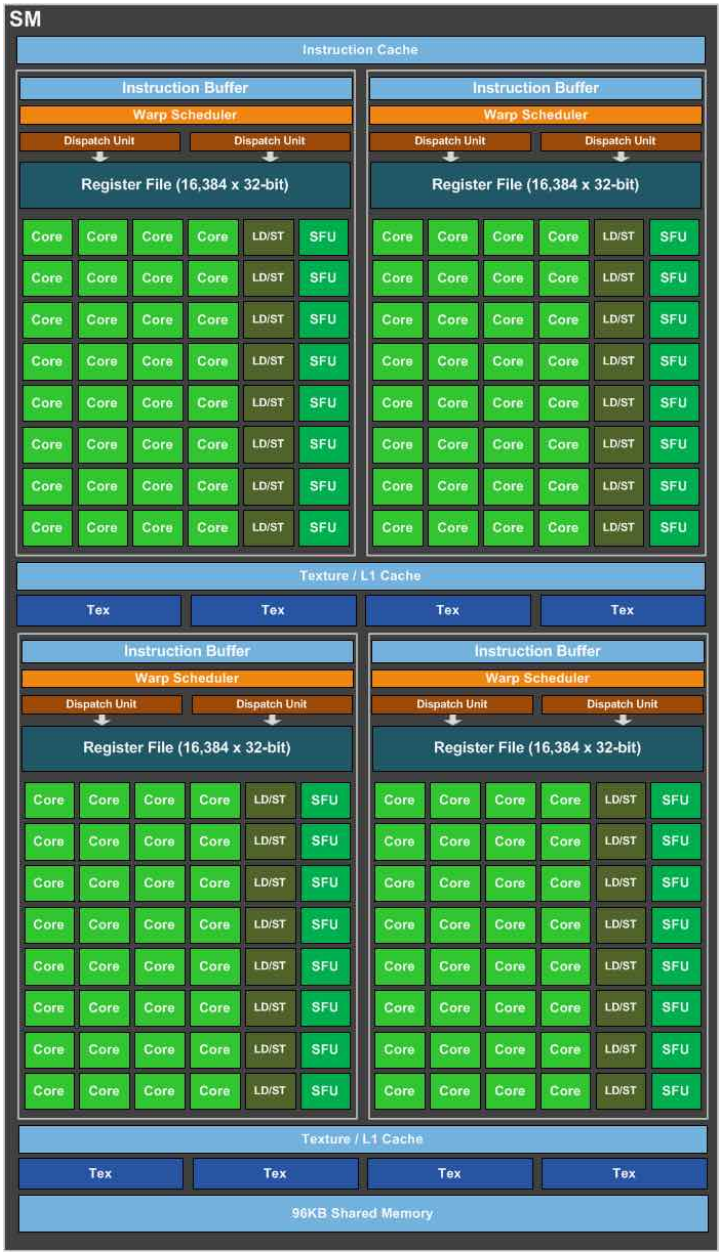


Figure 5: GP104 SM Diagram

The SM is a highly parallel multiprocessor that schedules warps (groups of 32 threads) to CUDA cores and other execution units within the SM. The SM is one of the most important hardware units within the GPU; almost all operations flow through the SM at some point in the rendering pipeline. With 20 SMs, the GeForce GTX 1080 ships with a total of 2560 CUDA cores and 160 texture units.

The GeForce GTX 1080 features eight 32-bit memory controllers (256-bit total). Tied to each 32-bit memory controller are eight ROP units and 256 KB of L2 cache. The full GP104 chip used in GTX 1080 ships with a total of 64 ROPs and 2048 KB of L2 cache.

The following table provides a high-level comparison of GeForce GTX 1080 versus the previous-generation GeForce GTX 980 GPU:

GPU	GeForce GTX 980 (Maxwell)	GeForce GTX 1080 (Pascal)
SMs	16	20
CUDA Cores	2048	2560
Base Clock	1126 MHz	1607 MHz
GPU Boost Clock	1216 MHz	1733 MHz
GFLOPs	4981[1]	8873[1]
Texture Units	128	160
Texel fill-rate	155.6 Gigatexels/sec	277.3 Gigatexels/sec
Memory Clock (Data Rate)	7,000 MHz	10,000 MHz
Memory Bandwidth	224 GB/sec	320 GB/sec
ROPs	64	64
L2 Cache Size	2048 KB	2048 KB
TDP	165 Watts	180 Watts
Transistors	5.2 billion	7.2 billion
Die Size	398 mm²	314 mm²
Manufacturing Process	28 nm	16 nm

[1] *The GFLOPS and texel fill rates in this chart are based on GPU Boost Clock*

Pascal Architecture: Crafted For Speed

Craftsmanship in every aspect of GPU design was a major focus for the Pascal development effort. Pascal is the most energy efficient GPU ever, due not only to the 16FF process but also due to continued improvements in energy efficiency of the GPU implementation. Clock frequency was another major investment area for the NVIDIA engineering team. Clock frequency is set not by the average circuit path timing in the design, but by the single slowest path among the millions of total timing paths. Careful design and optimization of critical paths is crucial to ensure that the capability of the overall design is not restricted. As a result of intense effort in this area, GeForce GTX 1080 delivers an over 40% increase in Boost Clock compared to GTX 980, well above what the 16FF process transition alone would enable.

GDDR5X Memory

Since the introduction of GDDR5 memory in 2009, NVIDIA's memory designers have been studying the possibilities for a next generation of memory signaling technology. GDDR5X is the culmination of that effort—the fastest and most advanced interface standard in history, achieving 10 Gbps transfer rates, or roughly 100 picoseconds (ps) between data bits. To put that speed of signaling in context, consider that light travels only about an inch in a 100 ps time interval. And the GDDR5X IO circuit has less than half that time available to sample a bit as it arrives, or the data will be lost as the bus transitions to a new set of values.

To achieve this high speed of operation, a new IO circuit architecture was needed, and it required a multi-year development project incorporating the latest advances in the field. Every aspect of the new design was carefully crafted to meet the exacting standards of high frequency operation. Significant energy efficiency improvement was also attained through a combination of these circuit advances, the lower 1.35 V GDDR5X standard, and new process technologies, resulting in the same power consumption at 43% higher frequency.

In addition, the "channel" between the GPU die and the memory die had to be designed with great attention to detail. The speed of operation of the interface is determined solely by the speed of the weakest signal on the bus. Every signal between the GPU and memory die was carefully studied along its entire path out of the GPU, through the package, onto the board and over to the memory die, with attention to channel loss, crosstalk, and discontinuities that can degrade the signal.

Overall, the circuit and channel improvements described above not only enable GDDR5X at 10 Gbps, but also provide benefits for future products that use GDDR5 memory.

The following pictures illustrate some elements of the design, including the IO circuit layout, and an extracted model of the path of one signal, highlighted in yellow, from the GPU (upper left) to DRAM pads (lower right) on the GeForce GTX 1080's new board channel design.

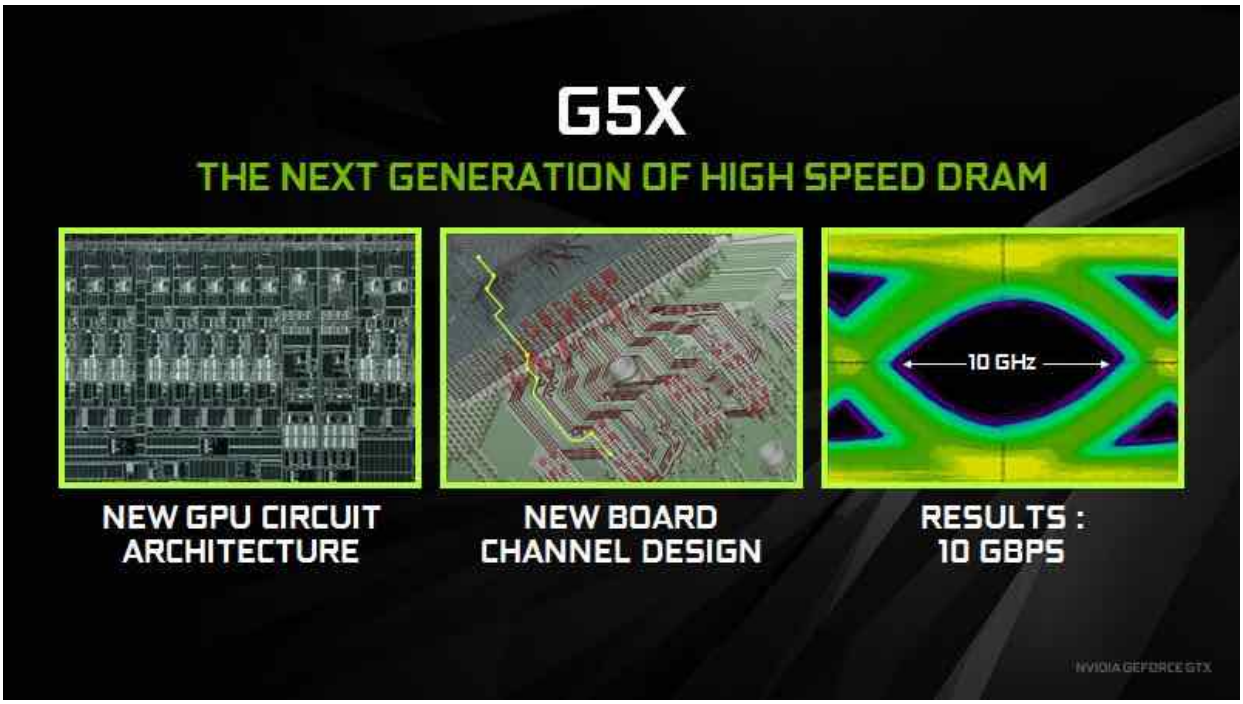


Figure 6: Engineering Advances Needed To Make GDDR5X Memory A Reality On the GeForce GTX 1080

The right image above is a long-term scope capture of one signal of the GDDR5X memory system in operation. Every bit sample (either 0 or 1 or a transition) is captured, and over time, a history is built up that can be studied to check the margins of the system. The large and symmetrical data “eye” in the middle of each transition demonstrates a successful design, with good margin for data capture.

Enhanced Memory Compression

Like previous GeForce GPUs, the memory subsystem of GeForce GTX 1080 uses lossless memory compression techniques to reduce DRAM bandwidth demands. The bandwidth reduction provided by memory compression provides a number of benefits:

- Reduces the amount of data written out to memory
- Reduces the amount of data transferred from memory to L2 cache; effectively providing a capacity increase for the L2 cache, as a compressed tile (block of frame buffer pixels or samples) has a smaller memory footprint than an uncompressed tile
- Reduces the amount of data transferred between clients such as the Texture Unit and the frame buffer

The GPU's compression pipeline has a number of different algorithms that intelligently determine the most efficient way to compress the data. One of the most important algorithms is delta color compression. With delta color compression, the GPU calculates the differences between pixels in a block and stores the block as a set of reference pixels plus the delta values from the reference. If the deltas are small then only a few bits per pixel are needed. If the packed together result of reference values plus delta values is less than half the uncompressed storage size, then delta color compression succeeds and the data is stored at half size (2:1 compression).

GeForce GTX 1080 includes a significantly enhanced delta color compression capability:

- 2:1 compression has been enhanced to be effective more often
- A new 4:1 delta color compression mode has been added to cover cases where the per pixel deltas are very small and are possible to pack into ¼ of the original storage
- A new 8:1 delta color compression mode combines 4:1 constant color compression of 2x2 pixel blocks with 2:1 compression of the deltas between those blocks

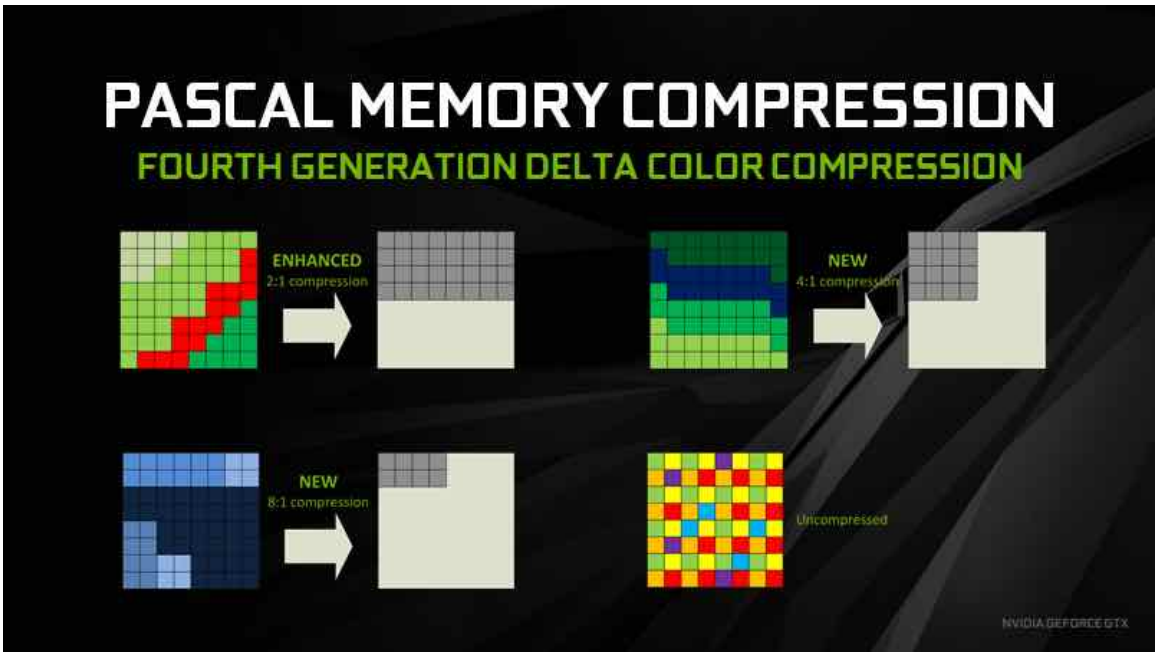


Figure 7: Compression Modes of Pascal's Memory Compression Engine

The following screenshots from *Project CARS* illustrate the benefit of Pascal's color compression. The parts of the scene that can be compressed have been replaced with magenta. While Maxwell was able to compress much of the scene, most of the vegetation and parts of the car could not be compressed. With Pascal, very little is left uncompressed.

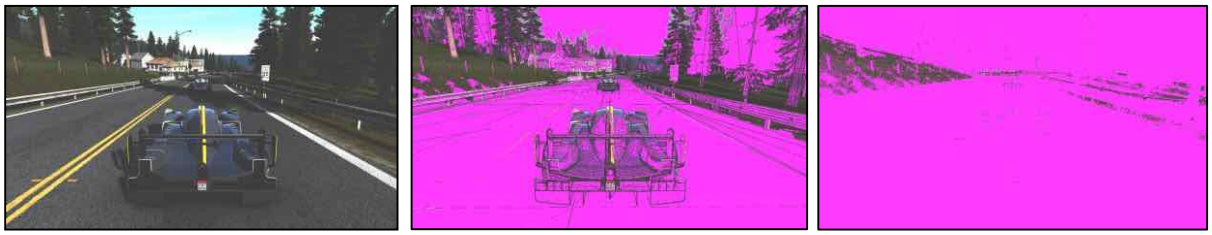


Figure 8: Original image, Maxwell Compression, Pascal Compression

As a result of the improvements in memory compression, the GeForce GTX 1080 is able to significantly reduce the number of bytes that have to be fetched from memory per frame. This reduction in bytes fetched translates to roughly 20% additional effective bandwidth, and when combined with GeForce GTX 1080's 10 Gbps GDDR5X memory, this ultimately provides the GTX 1080 a 1.7x effective memory bandwidth increase over GeForce GTX 980.

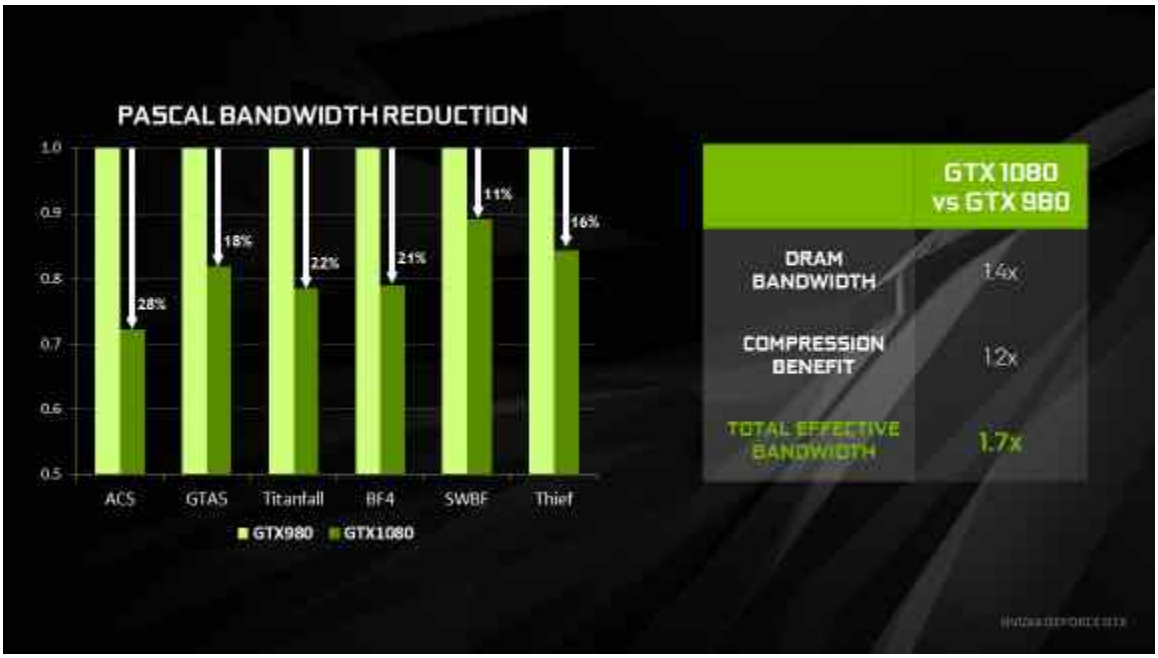


Figure 9: GeForce GTX 1080's Memory Compression Engine Compared to GTX 980

Asynchronous Compute

Modern gaming workloads are increasingly complex, with multiple independent, or "asynchronous," workloads that ultimately work together to contribute to the final rendered image. Some examples of asynchronous compute workloads include:

- GPU-based physics and audio processing
- Postprocessing of rendered frames
- Asynchronous timewarp, a technique used in VR to regenerate a final frame based on head position just before display scanout, interrupting the rendering of the next frame to do so

These asynchronous workloads create two new scenarios for the GPU architecture to consider.

The first scenario involves overlapping workloads. Certain types of workloads do not fill the GPU completely by themselves. In these cases there is a performance opportunity to run two workloads at the same time, sharing the GPU and running more efficiently—for example a PhysX workload running concurrently with graphics rendering.

For overlapping workloads, Pascal introduces support for "dynamic load balancing." In Maxwell generation GPUs, overlapping workloads were implemented with static partitioning of the GPU into a subset that runs graphics, and a subset that runs compute. This is efficient provided that the balance of work between the two loads roughly matches the partitioning ratio. However, if the compute workload takes longer than the graphics workload, and both need to complete before new work can be done, and the portion of the GPU configured to run graphics will go idle. This can cause reduced performance that may exceed any performance benefit that would have been provided from running the workloads

overlapped. Hardware dynamic load balancing addresses this issue by allowing either workload to fill the rest of the machine if idle resources are available.

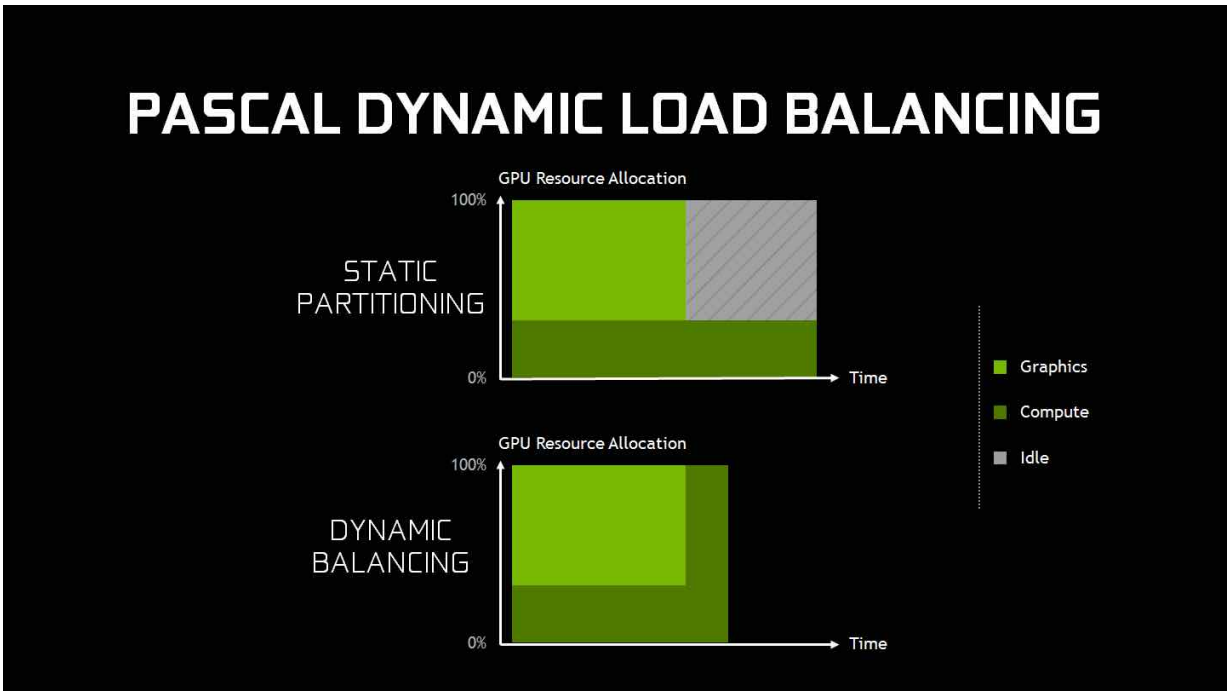


Figure 10: Pascal's Dynamic Load Balancing Reduces GPU Idle Time When Graphics Work Finishes Early, Allowing the GPU to Quickly Switch to Compute

Time critical workloads are the second important asynchronous compute scenario. For example, an asynchronous timewarp operation must complete before scanout starts or a frame will be dropped. In this scenario, the GPU needs to support very fast and low latency preemption to move the less critical workload off of the GPU so that the more critical workload can run as soon as possible.

As a single rendering command from a game engine can potentially contain hundreds of draw calls, with each draw call containing hundreds of triangles, and each triangle containing hundreds of pixels that have to be shaded and rendered. A traditional GPU implementation that implements preemption at a high level in the graphics pipeline would have to complete all of this work before switching tasks, resulting in a potentially very long delay.

To address this issue, Pascal is the first GPU architecture to implement Pixel Level Preemption. The graphics units of Pascal have been enhanced to keep track of their intermediate progress on rendering work, so that when preemption is requested, they can stop where they are, save off context information about where to start up again later, and preempt quickly. The illustration below shows a preemption request being executed.

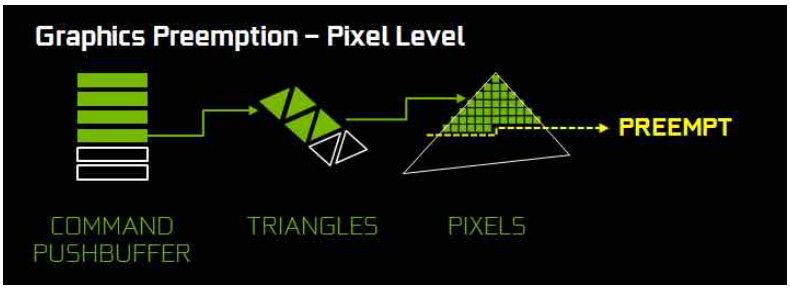


Figure 11: Pascal Supports Pixel- Level Graphics Preemption, Allowing the GPU to Switch Workloads Mid-Triangle

In the command pushbuffer, three draw calls have been executed, one is in process and two are waiting. The current draw call has six triangles, three have been processed, one is being rasterized and two are waiting. The triangle being rasterized is about halfway through. When a preemption request is received, the rasterizer, triangle shading and command pushbuffer processor will all stop and save off their current position. Pixels that have already been rasterized will finish pixel shading and then the GPU is ready to take on the new high priority workload. The entire process of switching to a new workload can complete in less than 100 microseconds (µs) after the pixel shading work is finished.

Pascal also has enhanced preemption support for compute workloads. The illustration below shows the execution of a compute workload.

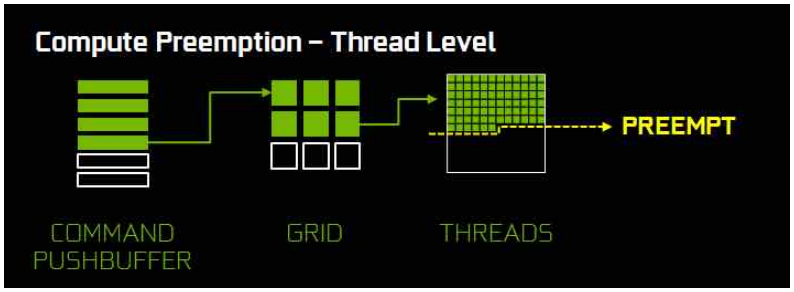


Figure 12: Pascal Supports Compute Preemption at the Thread Level for DX12 Graphics

Thread Level Preemption for compute operates similarly to Pixel Level Preemption for graphics. Compute workloads are composed of multiple grids of thread blocks, each grid containing many threads. When a preemption request is received, the threads that are currently running on the SMs are completed. Other units save their current position to be ready to pick up where they left off later, and then the GPU is ready to switch tasks. The entire process of switching tasks can complete in less than 100 µs after the currently running threads finish.

For gaming workloads, the combination of pixel level graphics preemption and thread level compute preemption gives Pascal the ability to switch workloads extremely quickly with minimal preemption overhead.

For CUDA compute tasks, Pascal is also capable of preempting at the finest granularity possible—instruction level.

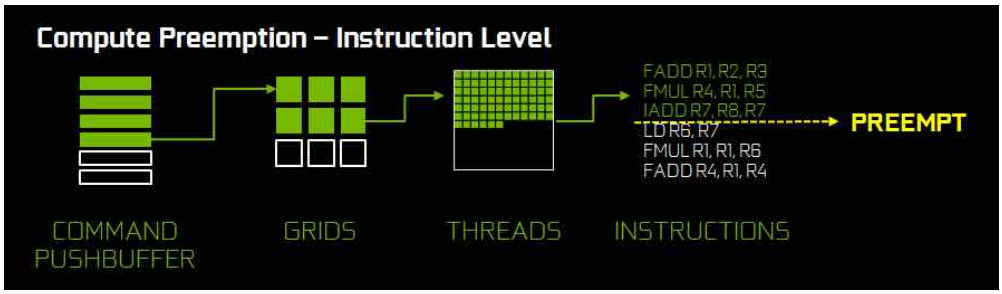


Figure 13: Pascal GPUs Support Instruction-Level Compute Preemption when Running CUDA Apps

In this mode of operation, when a preemption request is received, all thread processing stops at the current instruction and state is switched out immediately. This mode of operation involves substantially more state information, because all the registers of every running thread must be saved, but this is the most robust approach for general GPU compute workloads that may have substantial per-thread runtimes.

One example application of preemption in gaming is asynchronous timewarp. The left side of the illustration below shows an asynchronous timewarp operation with traditional GPU preemption. The ATW process runs as late as possible before the display refresh interval. However the ATW work has to be given to the GPU several milliseconds in advance, because without fine grained preemption, there is variability in the time it will take to preempt and start execution of the ATW process. On the right image, with fine-grained preemption (pixel level graphics plus thread level compute preemption), the preemption time is much faster and more deterministic, so the ATW work can be submitted much later, while still being assured of completion before the display refresh deadline.

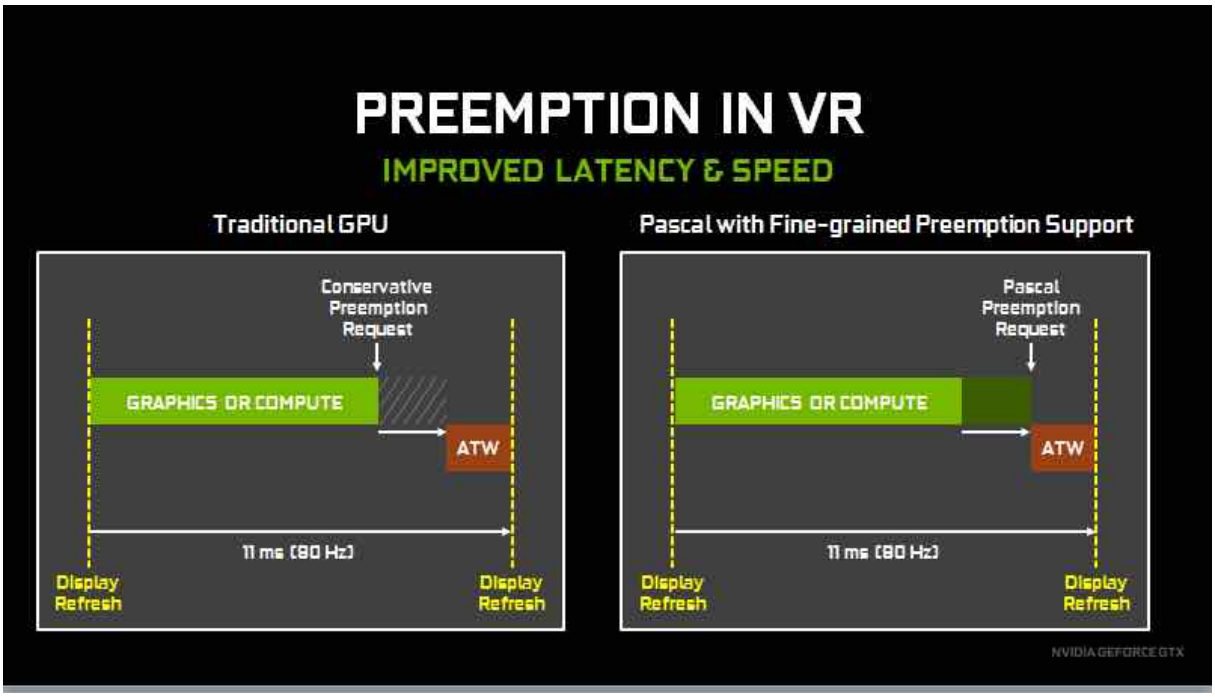


Figure 14: Pascal Preemption Support Prevents Idling In the Async Timewarp Scenario Above

Simultaneous Multi-Projection Engine

The Simultaneous Multi-Projection block is a new hardware unit, which is located inside the PolyMorph Engine at the end of the geometry pipeline and right in front of the Raster Unit. As its name implies, the Simultaneous Multi-Projection (SMP) unit is responsible for generating multiple projections of a single geometry stream, as it enters the SMP engine from upstream shader stages.

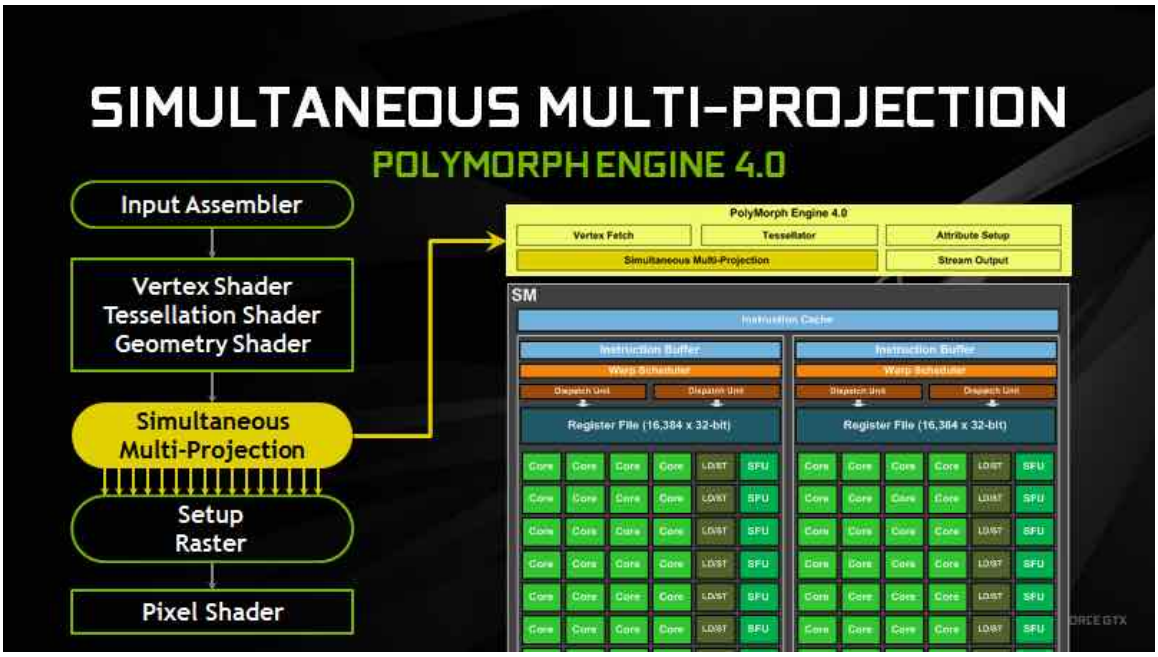


Figure 15: GeForce GTX 1080 Features a New PolyMorph Engine that Supports Simultaneous Multi-Projection

The Simultaneous Multi-Projection Engine is capable of processing geometry through up to 16 pre-configured projections, sharing the center of projection (the viewpoint), and with up to 2 different projection centers, offset along the X axis. Projections can be independently tilted or rotated around an axis. Since each primitive may show up in multiple projections simultaneously, the SMP engine provides multi-cast functionality, allowing the application to instruct the GPU to replicate geometry up to 32 times (16 projections x 2 projection centers) without additional application overhead as the geometry flows through the pipe.

In all scenarios, the processing is hardware-accelerated, and the stream of data never leaves the chip. Since the multi-projection expansion happens after the geometry pipeline, the application saves all the work that would otherwise need to be performed in upstream shader stages. The savings are particularly important in geometry-heavy scenarios, such as tessellation, where running the geometry processing pipeline multiple times (once for each projection) would be prohibitively expensive. In extreme cases, the SMP engine can reduce the amount of required geometry work by up to 32x!

One example application of SMP is optimal support for surround displays. The correct way to render to a surround display is with a different projection for each of the three displays, matching the display angle. This is supported directly in a single pass by Pascal SMP, by specifying three separate projections, each corresponding to the appropriately tilted monitor. Now, the user has the flexibility to choose the desired tilt for their side displays and will see their graphics rendered with geometrically correct perspectives, at a much wider field of view (FOV). Note that an application using SMP to generate surround display images must support wide FOV settings, and also use SMP API calls to enable the wider FOV.

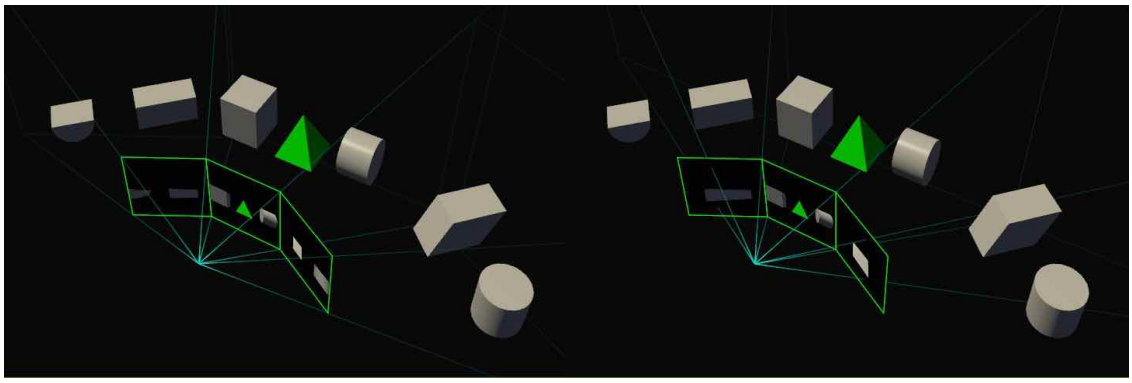


Figure 16: Surround setups with SMP perspective correction (left) and without SMP perspective correction (right). Note how in the right image the rendering frustum used by the application is inconsistent with display placement, resulting in geometric distortions on the side displays

In cases where the projection surface can't be exactly represented with finite number of planar projections, the SMP engine can still provide substantial efficiency gains by generating a much closer approximation to the desired projection surface.

SMP: Designed for the New Display Revolution

Since the early days of 3D rendering, the graphics pipeline has been designed with a simple assumption that the render target is a single, flat display screen. However, in recent years advances in display technology have led to many new types of display scenarios that do not fit the classical assumption. Surround multi-monitor setups are an excellent solution to give a sense of immersive realism in 3D games, and curved single-display monitors are also becoming popular. VR display systems put a lens between the viewer and the screen, requiring a new type of projection that is different from the standard flat planar projection that traditional GPUs support. The figure below illustrates a variety of these technologies that are both here today or still in development.

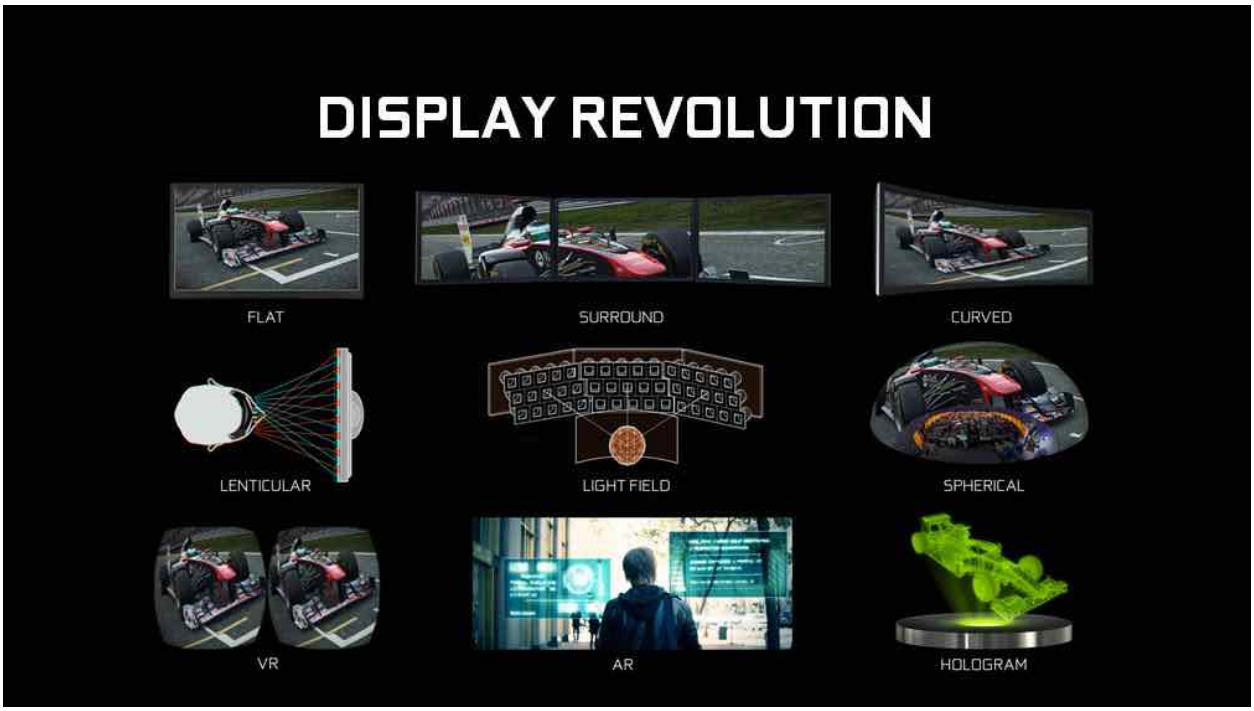


Figure 17: Displays are rapidly evolving beyond the traditional single flat display.

Traditional GPUs can support these types of displays, but only with significant inefficiencies—either requiring multiple rendering passes, or rendering with overdraw and then warping the image to match the display, or both.

Maxwell had a limited "Multi-Resolution" capability that was a precursor to Pascal SMP. Maxwell was able to flip a projection by exactly 90 degrees (ie for cube mapping), or take a single projection direction and proportionally scale the resolution in subregions of the screen. While useful for applications such as VXGI, this capability was limited and also didn't match up efficiently with the needs of these new display scenarios.

With Simultaneous Multi-Projection and the ability to handle multiple tilted or rotated projections at once, Pascal GPUs can now support these new display use cases directly, with dramatically improved efficiency.

Projections in 3D Graphics

The notion of "projection" has been fundamental since the dawn of 3D computer graphics. Geometric objects in the scene are modeled in three dimensions. However, in order to display a view of the scene on a flat display, the scene needs to be projected onto the screen, a process referred to as perspective projection. Projection is the computer graphics equivalent of drawing a picture on a window that exactly matches the view of the real world that you saw when looking through the window. One of Albrecht Durer's prints, reproduced below, depicts the process of constructing a perspective projection of a scene and illustrates the analogy to viewing the world through a window.

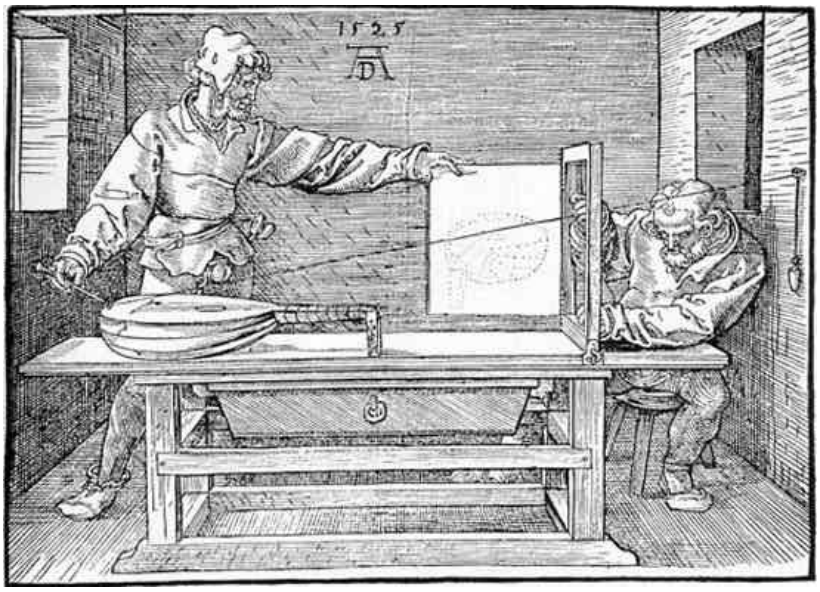


Figure 18: Albrecht Durer's Pictures for Geometry

Just as in the illustration above, perspective projection is performed by taking a line from each point in the scene to the location of the eye of the viewer, and finding the spot at which the line intersects the projection plane. The projection is defined by two pieces of information, (a) the location of the eye of the viewer, and (b) the direction that the viewer is looking.

The image below shows a basic projection performed on the GPU. The green box represents a projection that could be displayed on a computer screen, which would create a proper view of a cube, pyramid, and cylinder.

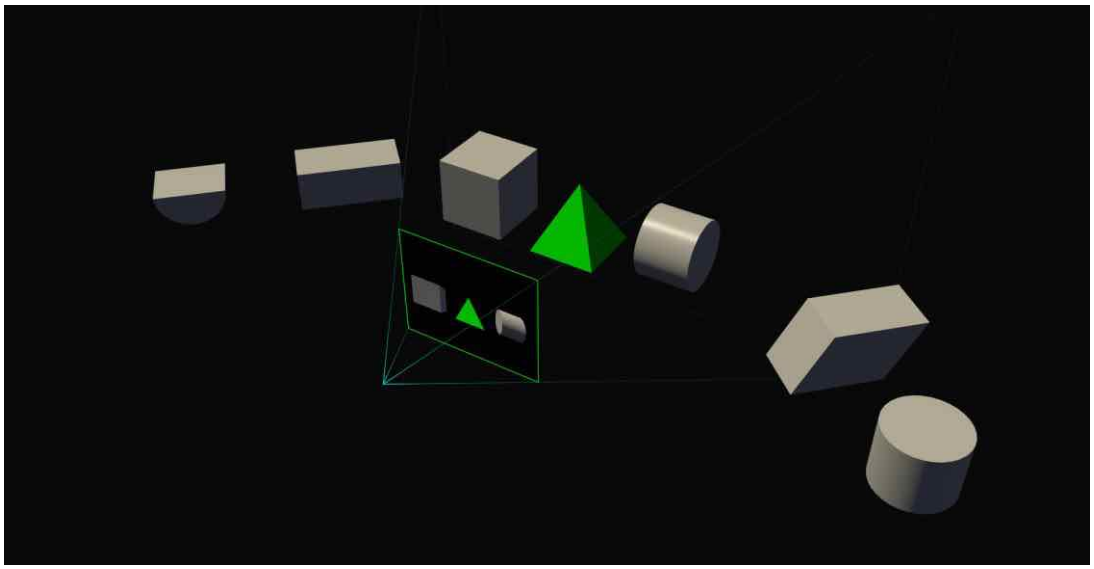


Figure 19: A 3-dimensional scene projected onto a flat plane

However, as discussed above, there are multiple display scenarios that do not match this simple model.

Perspective Surround

Let's take an example of a typical surround display setup, comprising three separate monitors, placed side-by-side directly next to each other. In this setup, a game would assume a wider horizontal field of view, but it would still render assuming a single, flat plane projection. See the figure below:

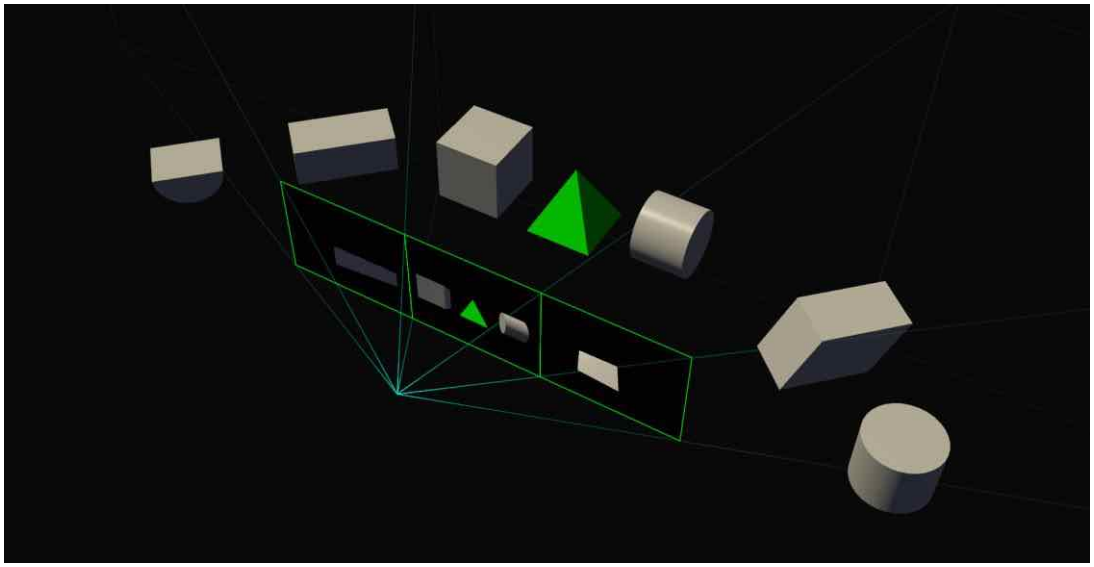


Figure 20: Single Plane Projection in Surround Gaming

If the user arranges their monitors to form a flat plane, the final result will be geometrically correct. However, this setup requires a large amount of desk space and offers a limited field of view. It is preferable to rotate the left and right side monitor inwards, which should dramatically increase the field of view. However, if the game is rendered assuming a single planar projection, the apparent perspective of the image will no longer be correct—it will appear excessively stretched and distorted on the sides.

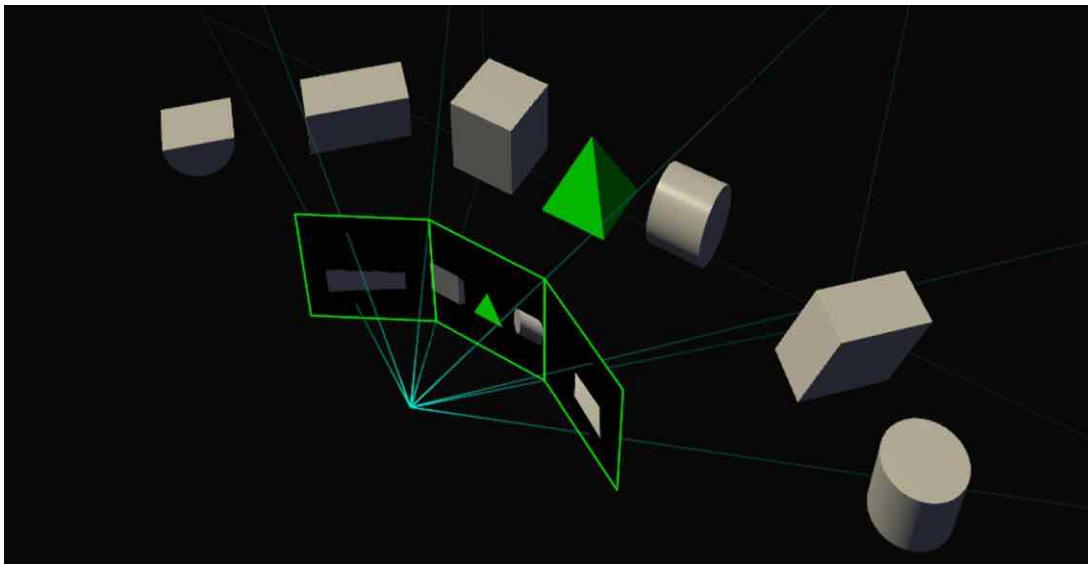


Figure 21: With Tilted Monitors, a Single Flat Projection Will No Longer Appear Correct To The Viewer

This occurs because although the displays are tilted inwards, the rendering assumes they are in a flat plane. Note that the lines of projection are unchanged although the displays have moved—so the blue lines on the edges no longer match up with the side displays. The projection no longer matches the display setup and is therefore incorrect.

In order to address this issue, one option is to render each monitor separately, appropriately adjusting projection parameters to match the tilt of each display. However, this approach results in a significant increase in the rendering workload, since the scene effectively has to be rendered three times, resulting in the game engine performing 3x the scene management work, 3x the OS runtime and driver work, and 3x the GPU front end and geometry work.

Instead, the Pascal SMP feature allows a single rendering pass. Perspective Surround is configured to know that there are three active projections - one for each monitor and each primitive, and will apply each of the active projections for each display on the fly. The result is a correctly rendered surround view, with no loss of performance.

Single Pass Stereo

An important aspect of VR rendering is the requirement that at least two projections need to be generated, one for each eye. As in the surround case, this is normally supported by apps today by rendering to each eye separately, which results in twice the amount of work for the entire pipeline, starting from the driver and the OS, and all the way down to the GPU's raster backend.

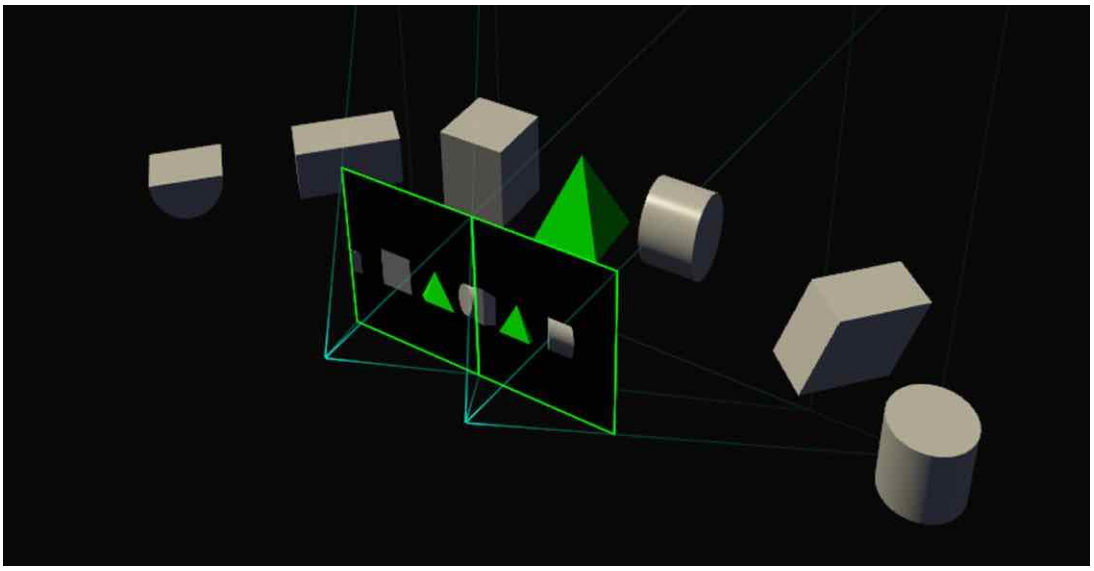


Figure 22: To generate a stereo pair, two projections of the sceen need to be rendered, one for each eye

With Pascal's SMP engine, which supports two separate projection centers, the GPU can render the two stereo projections directly in a single rendering pass. This SMP capability is known as **Single Pass Stereo**. With Single Pass Stereo, all the pipeline work, including scene submission, driver and OS scheduling, and geometry processing on the GPU, can be performed only once, which resulting in substantial performance gains—see NVIDIA's "Barbarian" demo as an example.

In Single Pass Stereo mode, the application runs vertex processing only once, but outputting two, (rather than one), positions for each vertex being processed. The two positions represent locations of the vertex as viewed from the left and from the right eye. The SMP hardware takes care of picking the right version of the vertex and routing it to the appropriate eye. From that point on, the SMP hardware can further apply a number of projections to the current primitive, as explained in the SMP architecture section. This functionality is important for combining the Single Pass Stereo with Lens Matched Shading, which we will review in the next section.

Lens Matched Shading

The explosive growth of interest in VR applications has increased the importance of supporting displays which require rendering to non-planar projections. VR displays have a lens in between the viewer and the display, which bends and distorts the image. For the image to look correct to the user, it would have to be rendered with a special projection that inverts the distortion of the lens. Then when the image is viewed through the lens, it will look undistorted, because the two distortions cancel out.

Traditional GPUs do not support this type of projection; instead they only support a standard "planar" projection with a uniform sampling rate. Producing a correct final image with traditional GPUs requires two steps—first, the GPU must render with a standard projection, generating more pixels than needed. Second, for each pixel location in the output display surface, look up a pixel value from the rendered result from the first step to apply to the display surface.

The images below provide an example of this traditional GPU two-step process. On the left is the first step rendering. On the right is an example of the final image as it would be shown on the VR display. The center of the image looks about the same as it would with a standard projection, but on the sides the image is squeezed. The final image in this example (based on Oculus Rift parameters) is 1.1 Mpixels per eye. If the source rendering was perfectly matched to the final projection, it should also be 1.1 Mpixels per eye. However, due to the mismatch in projections, the source image is 2.1 Mpixels per eye—86% more pixels than necessary.

Figure 23: First Pass Image and Final Image Required For Correct Viewing Through the HMD Optics

Leveraging SMP's ability to use multiple projection planes for a single viewpoint, we can attempt to approximate the shape of the lens-distorted projection. This feature is known as **Lens Matched Shading**.

With Lens Matched Shading, the SMP engine subdivides the display region into four quadrants, with each quadrant applying its own projection plane. The parameters can be adjusted to approximate the shape of the lens distortion as closely as possible. The left image below is the new rendered image with

Lens Matched Shading, compared to the final image on the right. The source image on the left is now 1.4 Mpixels per eye instead of 2.1 Mpixels, a significant reduction in shading rate that translates to a 50% increase in throughput available for pixel shading.

Figure 24: First Pass Image with Lens Matched Shading, and Final Image

One step in determining the Lens Matched Shading parameters is to check the sampling rate compared to the sampling rate required for the final image. The objective for the default / "conservative" setting of Lens Matched Shading is to always match or exceed the sampling rate of the final image. The image below shows an example comparison for the lens matched shading image above.

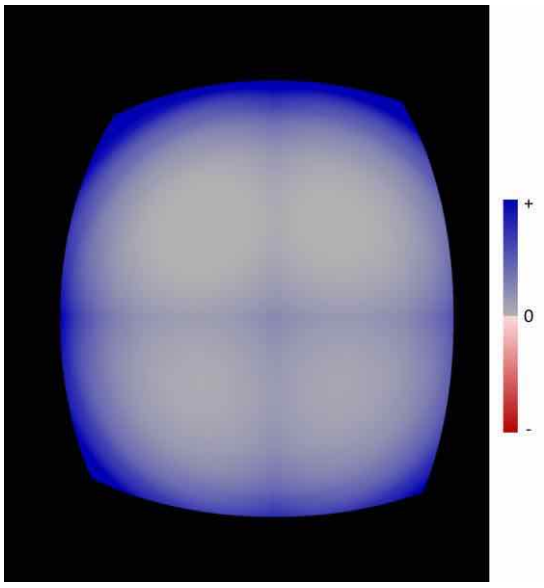
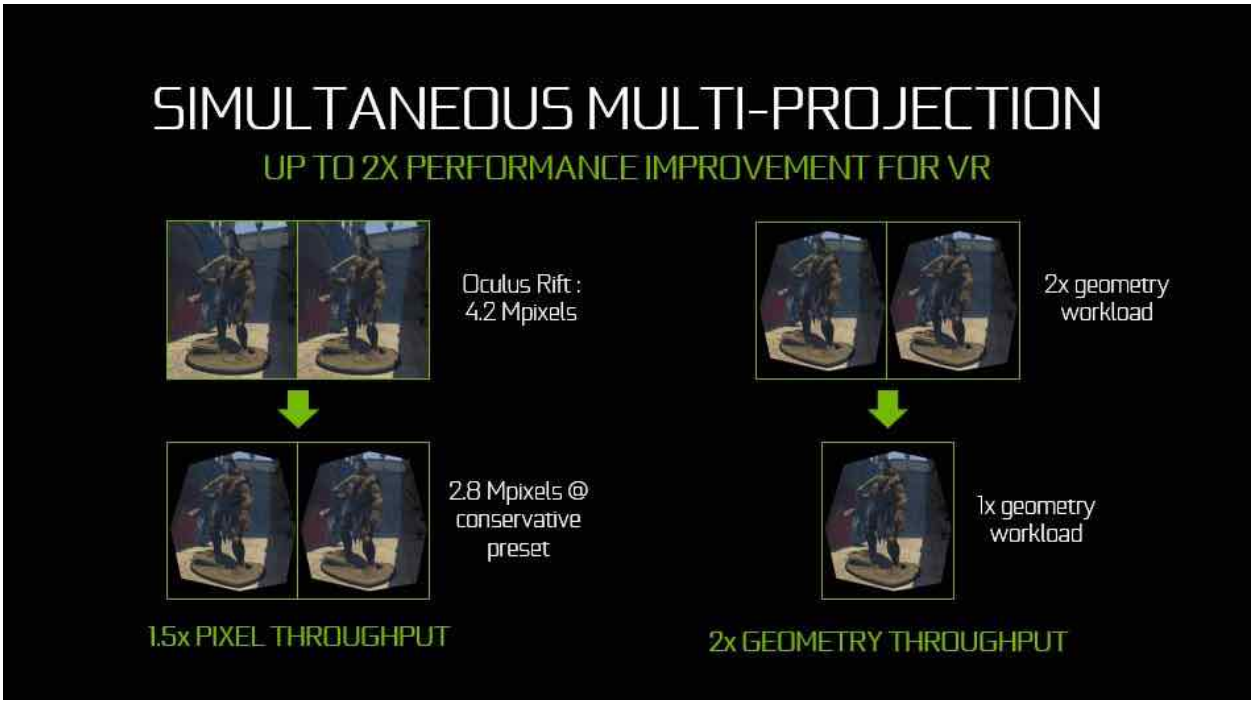


Figure 25: First pass image sampling rate compared to final image

Blue indicates pixels that were sampled at a higher rate than required, gray indicates a matched rate, and any red pixels would indicate initial sampling that was below the rate in the final image. The absence of red pixels confirms that the setting matches the objective.

In addition, developers will have the option to use different settings; for example one could use a setting that is higher resolution in the center and undersampled in the periphery, to maximize frame rate without significant visual quality degradation.

In summary, with the combination of Single Pass Stereo and Lens Matched Shading, Pascal can deliver up to 2x performance improvement for VR, compared to a GPU without support for Simultaneous Multi-Projection.



Enhanced SLI Interface

Gaming enthusiasts rely on NVIDIA SLI technology to deliver the very best gaming experience at the highest screen resolutions and graphics settings. One critical ingredient to NVIDIA's SLI technology is the SLI Bridge, which is a digital interface that transfers display data between GeForce graphics cards in a system.

Two of these interfaces have historically been used to enable communications between three or more GPUs (i.e., 3-Way and 4-Way SLI configurations). The second SLI interface is required for these scenarios because all other GPUs need to transfer their rendered frames to the display connected to the master GPU, and up to this point each interface has been independent.

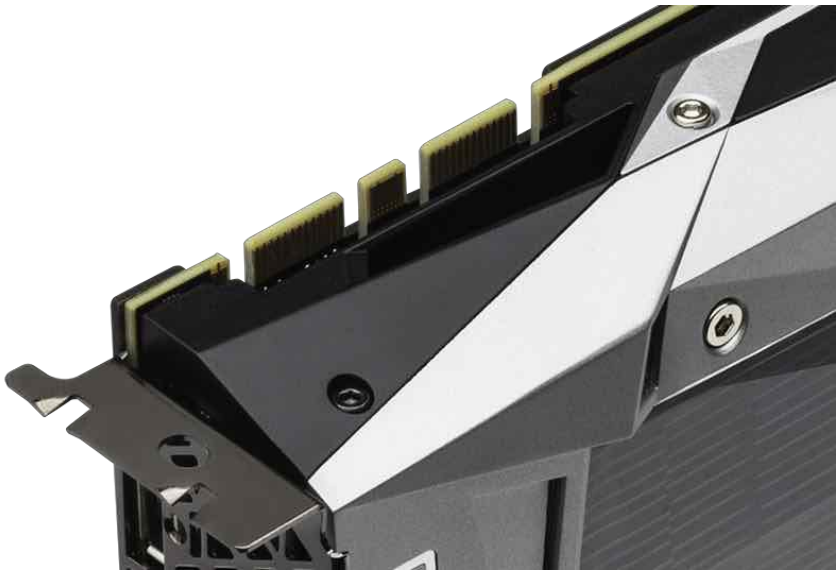


Figure 26: Dual SLI Interface Connectors on GeForce GTX 1080

Beginning with NVIDIA Pascal GPUs, the two interfaces are now linked together to improve bandwidth between GPUs. This new dual-link SLI mode allows both SLI interfaces to be used in tandem to feed one Hi-res display or multiple displays for NVIDIA Surround.

Dual-link SLI mode is supported with a new SLI Bridge called SLI HB. The bridge facilitates high-speed data transfer between GPUs, connecting both SLI interfaces, and is the best way to achieve full SLI clock speeds with GeForce GTX 1080 GPUs running in SLI ***(NOTE: The GeForce GTX 1080 is also compatible with legacy SLI bridges; however, the GPU will be limited to the maximum speed of the bridge being used).***

Using this new SLI HB Bridge, GeForce GTX 1080's new SLI interface runs at 650 MHz, compared to 400 MHz in previous GeForce GPUs using legacy SLI bridges. Where possible though, older SLI Bridges will also get a speed boost when used with Pascal. Specifically, custom bridges that include LED lighting will now operate at up to 650MHz when used with GTX 1080, taking advantage of Pascal's higher speed IO.

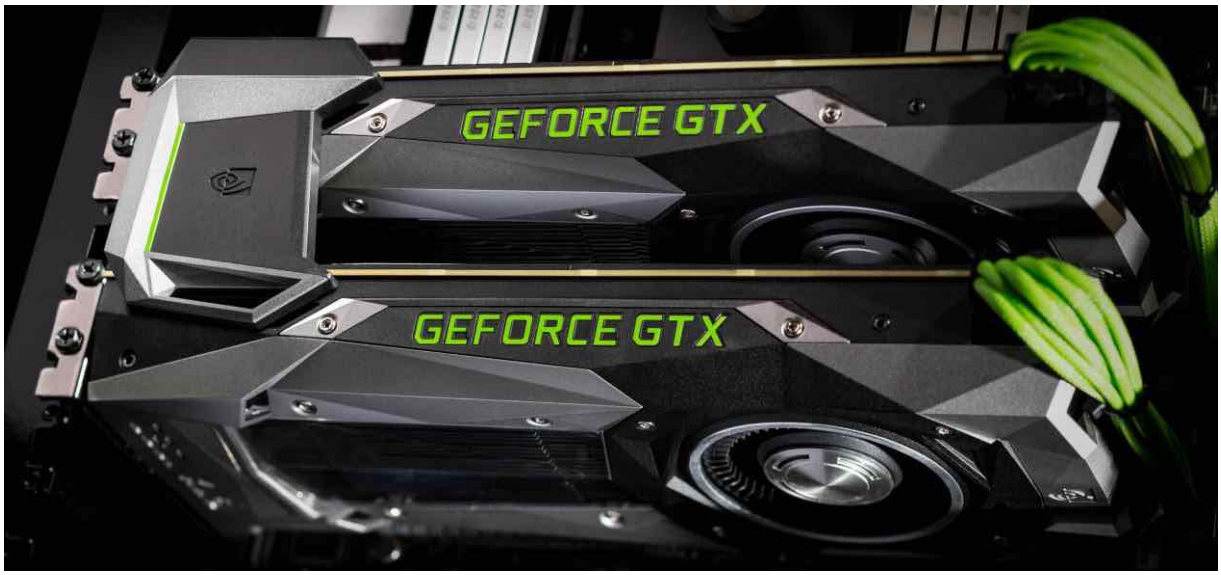
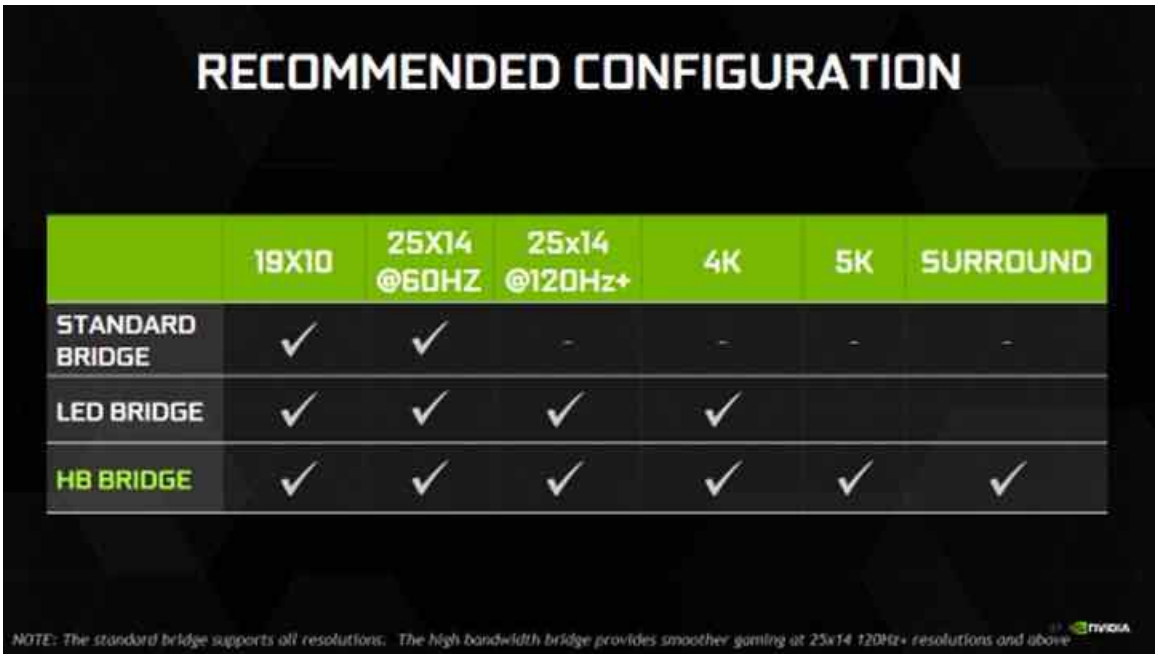


Figure 27: Two GeForce GTX 1080 Graphics Cards Connected with SLI-HB Bridge

The GeForce GTX 1080's new SLI subsystem provides more than double the bandwidth between GPUs compared to the SLI interface used on prior generation GeForce GTX GPUs. This is particularly important for high resolutions like 4K and 5K and surround.

Since the GeForce GTX 1080 now supports different types of bridges, it is important to understand which bridges work best for the intended use case. Below is a simple table highlighting recommend configurations:



With the additional bandwidth provided by the new SLI interface and SLI HB bridge, GeForce GTX 1080 SLI will provide gamers with a smoother gaming experience compared to prior SLI solutions as seen in the figure below:

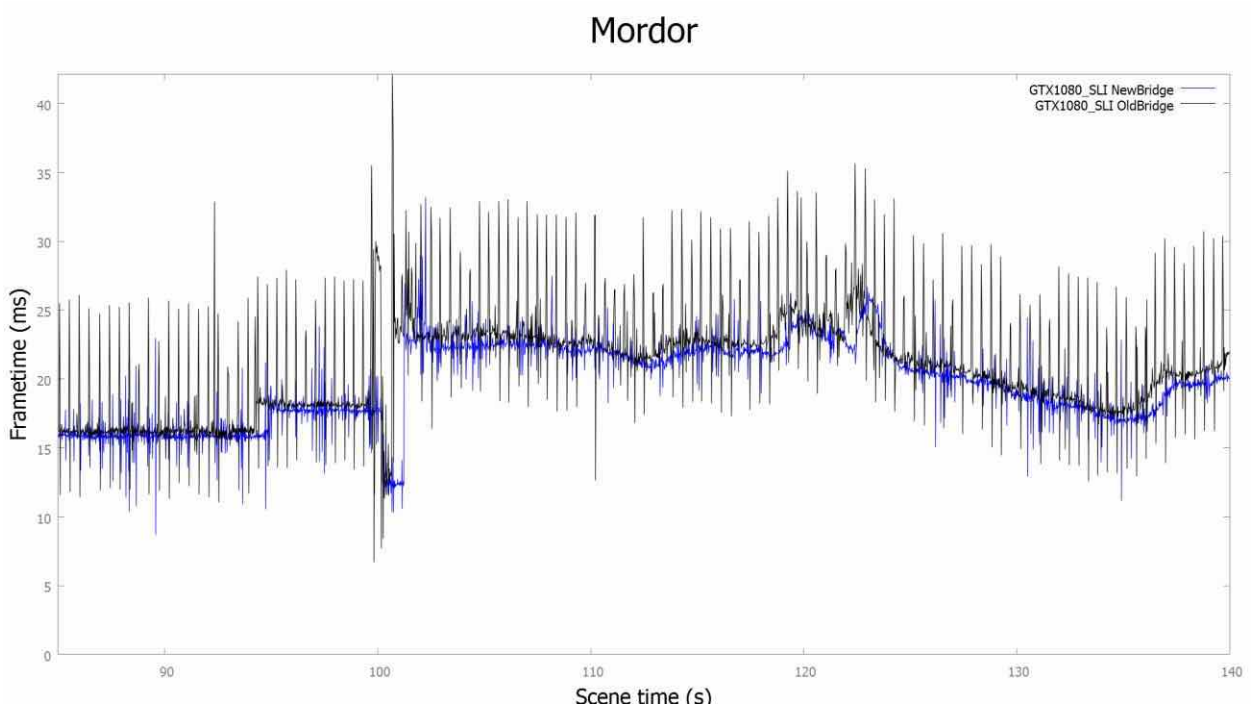


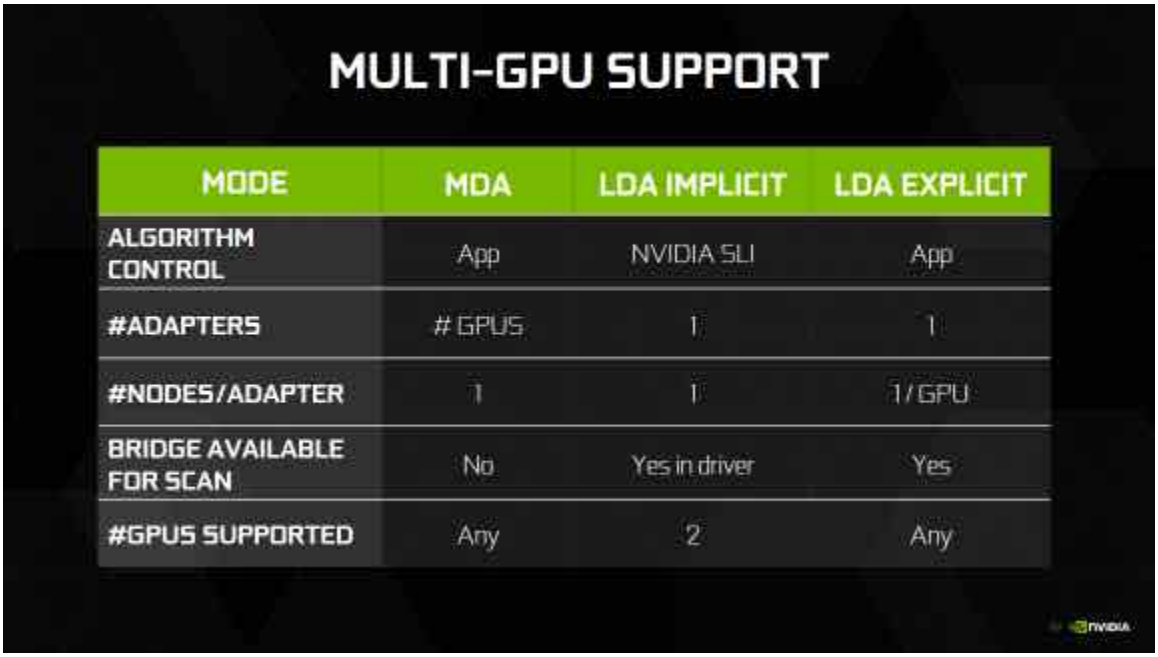
Figure 28: Frame Times with GeForce GTX 1080 running Shadow of Mordor at 11520x2160 with the new SLI bridge (blue) vs the old SLI bridge (black). Notice the large spikes with the old bridge, which indicates more stuttering.

New Multi-GPU Modes

Compared to prior DirectX APIs, Microsoft has made a number of changes that impact multi-GPU functionality in DirectX 12. At the highest level, there are two basic options for developers to use multi-GPU on NVIDIA hardware in DX12: Multi Display Adapter (MDA) Mode, and Linked Display Adapter (LDA) mode.

LDA Mode has two forms : **Implicit LDA Mode** which NVIDIA uses for SLI, and **Explicit LDA Mode** where game developers handle much of the responsibility needed for multi-GPU operation to work successfully. MDA and LDA Explicit Mode were developed to give game developers more control.

The following table summarize the three modes supported on NVIDIA GPUs:



In LDA Mode, each GPU's memory can be linked together to appear as one large pool of memory to the developer (although there are certain corner case exceptions regarding peer-to-peer memory); however, there is a performance penalty if the data needed resides in the other GPU's memory, since the memory is accessed through inter-GPU peer-to-peer communication (like PCIe). In MDA Mode, each GPU's memory is allocated independent of the other GPU: each GPU cannot directly access the other's memory.

LDA is intended for multi-GPU systems that have GPUs that are similar to each other, while MDA Mode has fewer restrictions—discrete GPUs can be paired with integrated GPUs, or with discrete GPUs from another manufacturer—but MDA Mode requires the developer to more carefully manage all of the operations that are needed for the GPUs to communicate with each other.

By default, GeForce GTX 1080 SLI supports **up to two GPUs**. 3-Way and 4-Way SLI modes are no longer recommended. As games have evolved, it is becoming increasingly difficult for these SLI modes to provide beneficial performance scaling for end users. For instance, many games become bottlenecked by the CPU when running 3-Way and 4-Way SLI, and games are increasingly using techniques that make

it very difficult to extract frame-to-frame parallelism. Of course systems will still be built targeting other Multi-GPU software models including:

1. MDA or LDA Explicit targeted
2. 2-Way SLI + dedicated PhysX GPU

Enthusiast Key

While NVIDIA no longer recommends 3 or 4 way systems for SLI, we know that true enthusiasts will not be swayed…and in fact some games will continue to deliver great scaling beyond two GPUs. For this class of user we have developed an Enthusiast Key that can be downloaded off of NVIDIA's website and loaded into an individual's GPU. This process involves:

1. Run an app locally to generate a signature for your GPU
2. Request an Enthusiast Key from an upcoming NVIDIA Enthusiast Key website
3. Download your key
4. Install your key to unlock the 3 and 4-way function

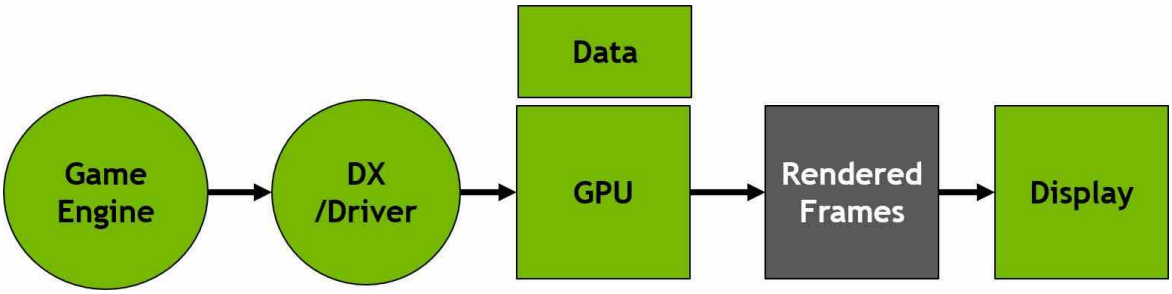
Full details on the process are available on the NVIDIA Enthusiast Key website, which will be available at the time GeForce GTX 1080 GPUs are available in users' hands.

Fast Sync

Fast Sync is a latency-conscious alternative to traditional Vertical Sync (V-SYNC) that eliminates tearing, while allowing the GPU to render unrestrained by the refresh rate to reduce input latency.

Rendered Frames – Traditional Method

This is a rough outline of how frame rendering works through the NVIDIA graphics pipeline:



The game engine is responsible for generating the frames that are sent to DirectX. The game engine also calculates animation time; the encoding inside the frame that eventually gets rendered. The draw calls and information are communicated forward, the NVIDIA driver and GPU converts them into actual rendering, and then spits out a rendered frame to the GPU frame buffer. The last step is to scan the frame to the display.

We are doing something different now with Pascal.

HIGH FPS Games

High FPS games like *Counter-Strike: Global Offensive* are running at many hundreds of frames per second today on Pascal. The question is: what good is that? Today, there are two choices on how to display the game; with V-SYNC ON or with V-SYNC OFF.

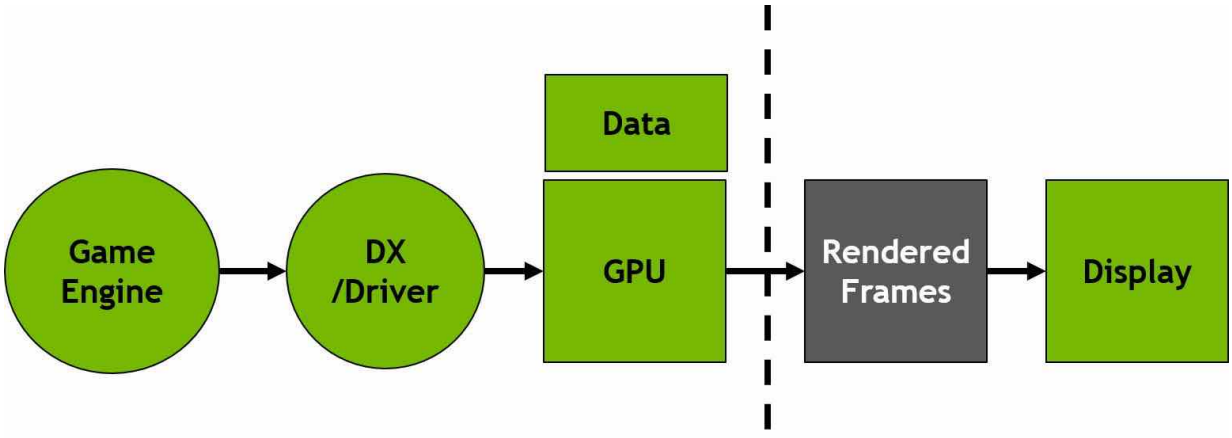
	V-SYNC ON	V-SYNC OFF
Flow Control	Backpressure	None
Input Latency	High	Low
Frame Tearing	None	Tearing

If you use V-SYNC ON, the pipeline gets back-pressured all the way to the game engine, and the entire pipeline slows down to the refresh rate of the display. With V-SYNC ON, the display is essentially telling the game engine to slow down, because only one frame can be effectively generated for every display refresh interval. The upside of V-SYNC ON is the elimination of frame tearing, but the downside is high input latency.

When using V-SYNC OFF, the pipeline is told to ignore the display refresh rate, and to deliver game frames as fast as possible. The upside of V-SYNC OFF is low input latency (as there is no backpressure), but the downside is frame tearing.

These are the choices that gamers face today, and the vast majority of eSports gamers are playing with V-SYNC OFF to leverage its lower input latencies, lending them a competitive edge. Unfortunately, tearing at high FPS causes a vast amount of jittering, which can hamper their gameplay.

Decoupled Render and Display



NVIDIA has taken another look at how the traditional process works, and for the first time, rendering and display are being decoupled from the pipeline. This allows the rendering stage to continually generate new frames from data sent by the game engine and driver at full speed, and those frames can can be temporarily stored in the GPU frame buffer.

Rendered Frames - FAST SYNC

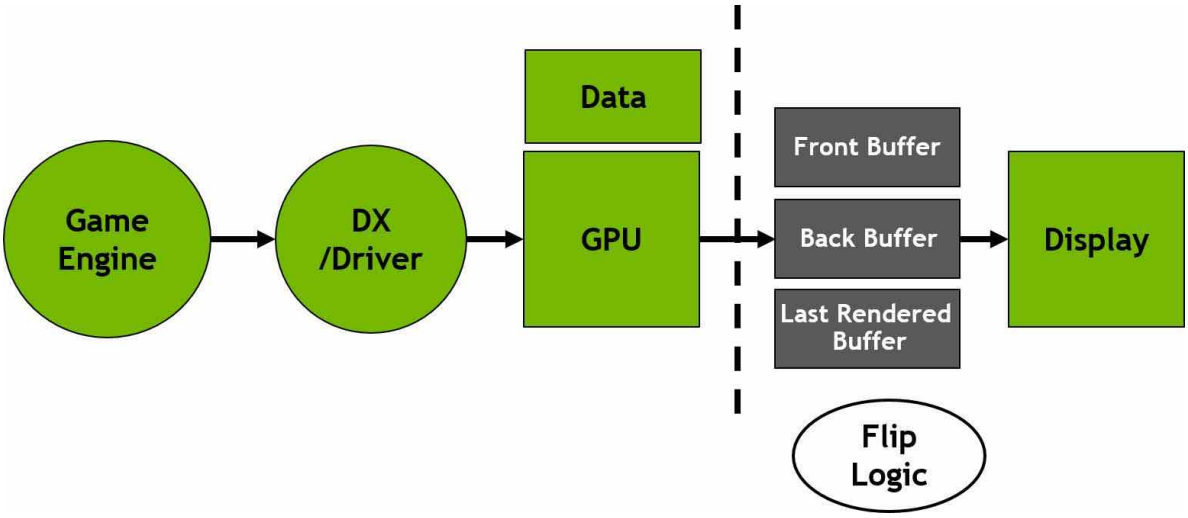
NVIDIA has decoupled the front end of the render pipeline from the backend display hardware. This allows different ways to manipulate the display that can deliver new benefits to gamers. Fast Sync is one of the first applications of this new approach.

With Fast Sync, there is no flow control. The game engine works as if V-SYNC is OFF. And because there is no backpressure, input latency is almost as low as with V-SYNC OFF. Best of all, there is no tearing because FAST SYNC chooses which of the rendered frames to scan to the display. FAST SYNC allows the front of the pipeline to run as fast as it can, and it determines which frames to scan out to the display, while simultaneously preserving entire frames so they are displayed without tearing.

	V-SYNC ON	**V-SYNC OFF**	**FAST SYNC**
Flow Control	Backpressure	None	None
Input Latency	High	Low	**Low**
Frame Tearing	None	Tearing	**None**

The experience that FAST SYNC delivers, depending on frame rate, is roughly equal to the clarity of V-SYNC ON combined with the low latency of V-SYNC OFF.

Decoupled Buffers



One way to think about Fast Sync is to imagine that three areas in the frame buffer have been allocated in three different ways. The first two buffers are very similar to double-buffered VSYNC in classic GPU pipelines. The Front Buffer (FB) is the buffer scanned out to the display. It is a fully rendered surface. The Back Buffer (BB) is the buffer that is currently being rendered to and it can't be scanned out until it is completed. Using traditional VSYNC In high render-rate games is not good for latency, since the game must wait for the display refresh interval to flip the back buffer to become the front buffer before another frame can be rendered into the back buffer. This slows down the entire process…and adding additional back buffers just adds latency, since they could all fill up at high rendering rates, causing similar stalls of the game engine.

Fast Sync introduces a third buffer called the Last Rendered Buffer(LRB) which is used to hold all newly rendered frames just completed in the back buffer – in effect having a copy of the most recently rendered back buffer - until the front buffer has finished scanning, at which point the Last Rendered Buffer is copied to the Front buffer and the process continues. Actual buffer copies would be inefficient, so instead the buffers are just renamed. The buffer being scanned to the display is the FB, the buffer being actively rendered to is the BB and the buffer holding the most recently rendered frame is the LRB. New flip logic in the Pascal architecture controls the entire process.

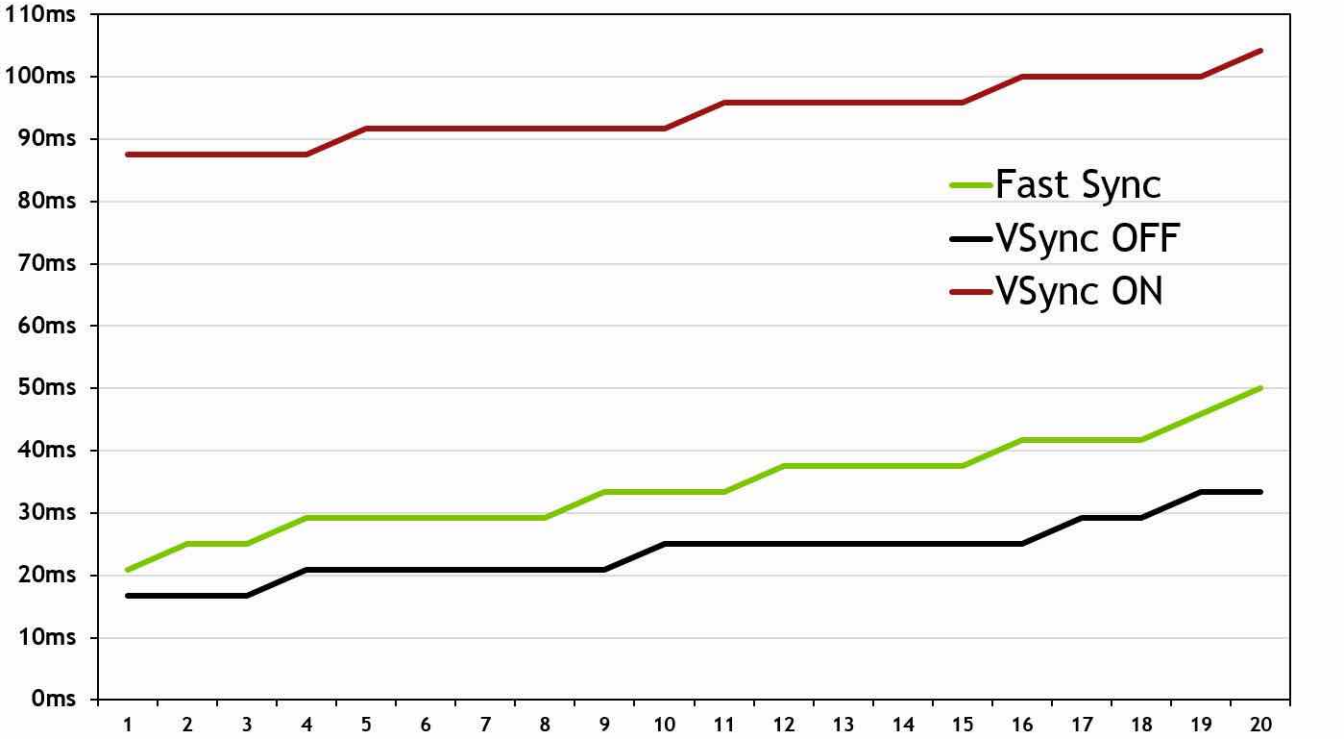
A typical process would look like this:

- Scan from FB
- Render to BB
- When Render completes
    - BB becomes LRB
    - LRB becomes BB and render continues
- When Render completes
    - BB becomes LRB
    - LRB becomes BB and render continues

- When Render completes
  - BB becomes LRB
  - LRB becomes BB and render continues
- When scan completes
  - LRB becomes FB
- Start scanning from the new FB

FAST SYNC Latency Results

The data is compelling. Latency is high with V-SYNC ON. Gamers of high FPS games today use V-SYNC OFF to get the best input response in their games, while dealing with the jitter caused by tearing at high frame rates. Turning FAST SYNC on delivers ~8ms more latency than V-SYNC OFF, while delivering entire frames without issues like tearing and jitter.



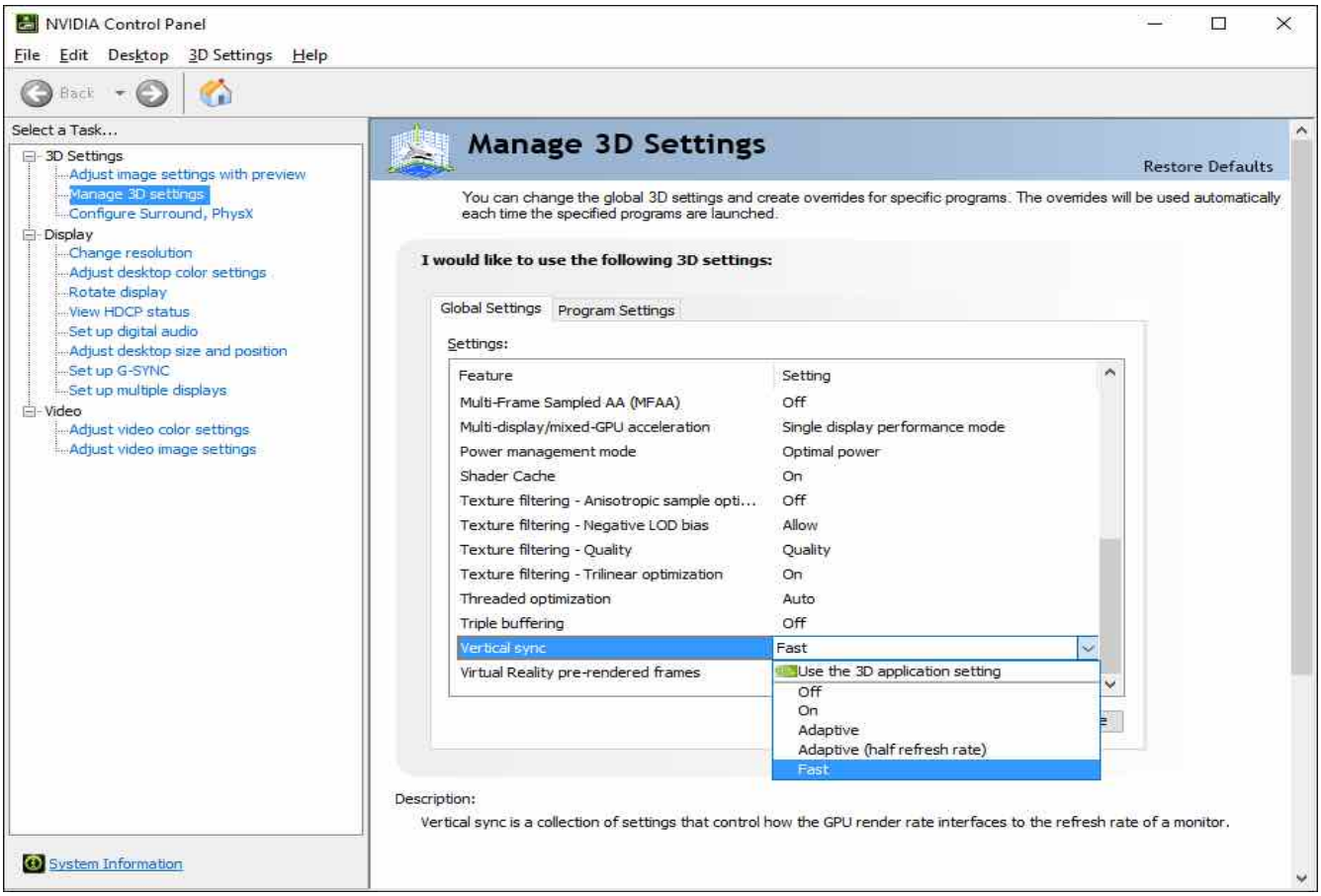
These samples were taken with a high-speed camera using *Counter-Strike: Global Offensive*.

NOTE Fast Sync is best tested with high FPS DX9 games.

NVIDIA Control Panel

FAST SYNC settings can be set in the NVIDIA Control Panel in the **Manage 3D settings** section under the **Vertical sync** setting.

NOTE: The default setting for Vertical Sync is to use the 3D application setting.



HDR

New High Dynamic Range (HDR) displays provide one of the biggest advance in display pixel quality in the last 20 years. The BT.2020 color gamut covers up to 75% of visible colors (up from 33% for sRGB)—an increase of over 2x in color range. In addition, HDR displays are estimated to deliver higher peak brightness (with LCD displays above 1000 nits) and a greater contrast ratio (>10:000 to 1).

With a greater range of brightness and more saturated colors, HDR content more closely mimics the real world: blacks are deeper and whites are brighter. Additional variations in color produces more vivid images that really pop: end users can finally see the reds and oranges in a fire or an explosion, instead of them looking washed out. And because HDR displays support higher contrast ratios, users will see more details in the brightest and darkest areas of scenes compared to today's standard dynamic range (SDR) displays.

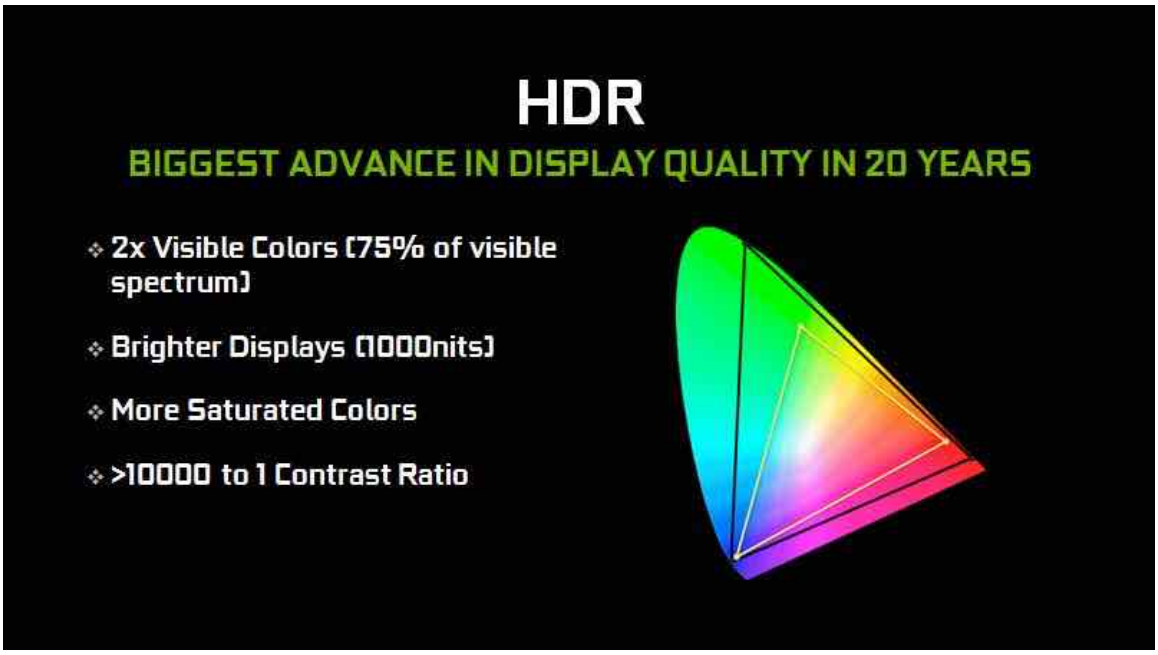


Figure 29: HDR Displays Offer More Colors, Brighter Picture, and More Contrast

The GeForce GTX 1080 supports all of the HDR display capabilities of the Maxwell GPU found previously in the GeForce GTX 980 – with the display controller capable of 12b color, BT.2020 wide color gamut, SMPTE 2084 (Perceptual Quantization), and HDMI 2.0b 10/12b for 4K HDR. In addition, Pascal introduces new features such as:

- 4K@60 10/12b HEVC Decode (for HDR Video)
- 4K@60 10b HEVC Encode (for HDR recording or streaming)
- DP1.4-Ready HDR Metadata Transport (to connect to HDR displays using DisplayPort)

Figure 30: Pascal GPU Features for HDR

HDR Televisions are available today, and the above features will enable users to enjoy HDR games on their HDR televisions even if their PC is not directly connected to the TV—by using HDR Gamestream, available in the near future.

Figure 31: GameStream in HDR from your GeForce GTX 1080 to SHIELD (coming soon!)

NVIDIA is working with game developers to bring HDR to PC games. NVIDIA is providing developers with the API and driver support as well as guidance needed to properly render an HDR image that is compatible with new HDR displays. As a result of these efforts, HDR games including *Obduction*, *The Witness, Lawbreakers, Rise of the Tomb Raider, Paragon, The Talos Principle* and *Shadow Warrior 2* are coming soon.

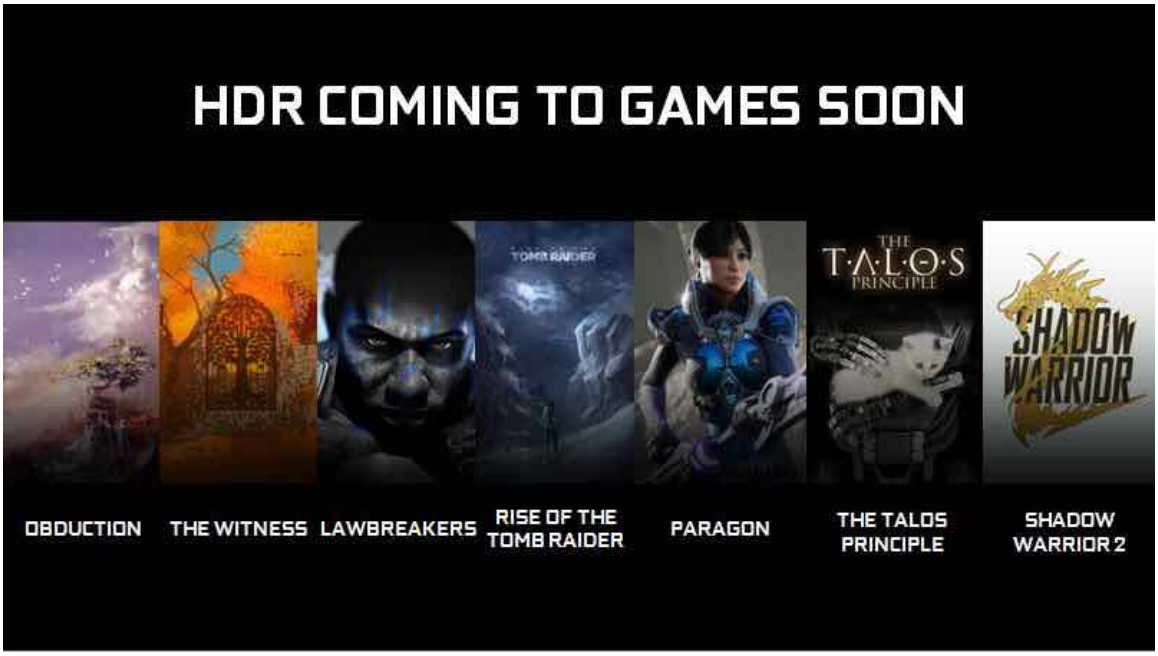


Figure 32: Upcoming HDR Games

Video and Display

Pascal GPUs meet the highest standards for PlayReady 3.0 (SL3000) and support HEVC decode in hardware, bringing the capability to watch 4K premium video on the PC for the first time. In coming months, consumers will be able to stream 4K Netflix content and 4K content from other premium content providers on Pascal-enabled PCs.

Figure 33: GeForce GTX 1080 Is 1st Device Ready to Stream 4K Premium Content

The GeForce GTX 1080 is DisplayPort 1.2 certified, DP 1.3/1.4 Ready, enabling support for 4K displays at 120Hz, 5K displays at 60Hz, and 8K displays at 60Hz (using two cables).

The table below summarizes the display features of the GeForce GTX 1080 relative to GTX 980:

	GeForce GTX 980	GeForce GTX 1080
Number of Active Heads	4	4
Number of Connectors	6	6
Max Resolution	5120 x 3200 @ 60 Hz (requires 2 DP 1.2 connectors)	7680 x 4320 @ 60 Hz (requires 2 DP 1.3 connectors)
Digital Protocols	LVDS, TMDS/HDMI 2.0, DP 1.2	HDMI 2.0b with HDCP 2.2, DP (DP 1.2 certified, DP 1.3 Ready, DP 1.4 Ready)

The GeForce GTX 1080 Founders Edition includes three DisplayPort connectors, one HDMI 2.0b connector, and one dual-link DVI connector. Up to four display heads can be driven simultaneously from one card.

Figure 34: GeForce GTX 1080 Founders Edition Bracket

The following table summarized the video features (Encode and Decode) of GTX 1080 (compared to GTX 980):

	GeForce GTX 980	GeForce GTX 1080
H.264 Encode	Yes	Yes (2x 4K@60 Hz)
HEVC Encode	Yes	Yes (2x 4K@60 Hz)
10-bit HEVC Encode	No	Yes
H.264 Decode	Yes	Yes (4K@120 Hz up to 240 Mbps)
HEVC Decode	No	Yes (4K@120 Hz/8K@30 Hz up to 320 Mbps)
VP9 Decode	No	Yes (4K@120 Hz up to 320 Mbps)
MPEG2 Decode	Yes	Yes
10-bit HEVC Decode	No	Yes
12-bit HEVC Decode	No	Yes

VRWorks

VRWorks™ is a comprehensive suite of APIs, libraries, and engines that enable application and headset developers to create amazing virtual reality experiences. VRWorks enables a new level of presence by bringing physically realistic visuals, sound, touch interactions, and simulated environments to virtual reality. For more details than described below, please refer to the VRWorks website.

VRWorks Graphics

Virtual Reality gaming requires low latency response times and high frame rates in order to provide an immersive experience for the user. While the graphics quality of the latest VR games generally looks good, they have yet to match the level of graphics seen in modern non-VR titles, largely because the high framerate requirements of VR leave little GPU horsepower for additional graphics effects.

Figure 35: VR Gaming is 7x More Demanding than gaming at 1080p

In order to bridge the graphics gap between VR and non-VR games, NVIDIA has developed a number of new technologies in the Pascal architecture to improve the performance of graphics rendering for VR applications. This ultimately allows the GeForce GTX 1080 to perform up to 2x faster than GeForce TITAN X in VR. With this performance improvement for Pascal GPUs, game developers can then crank up the graphics in their VR titles so that VR users can enjoy the same quality of graphics as traditional PC gamers.

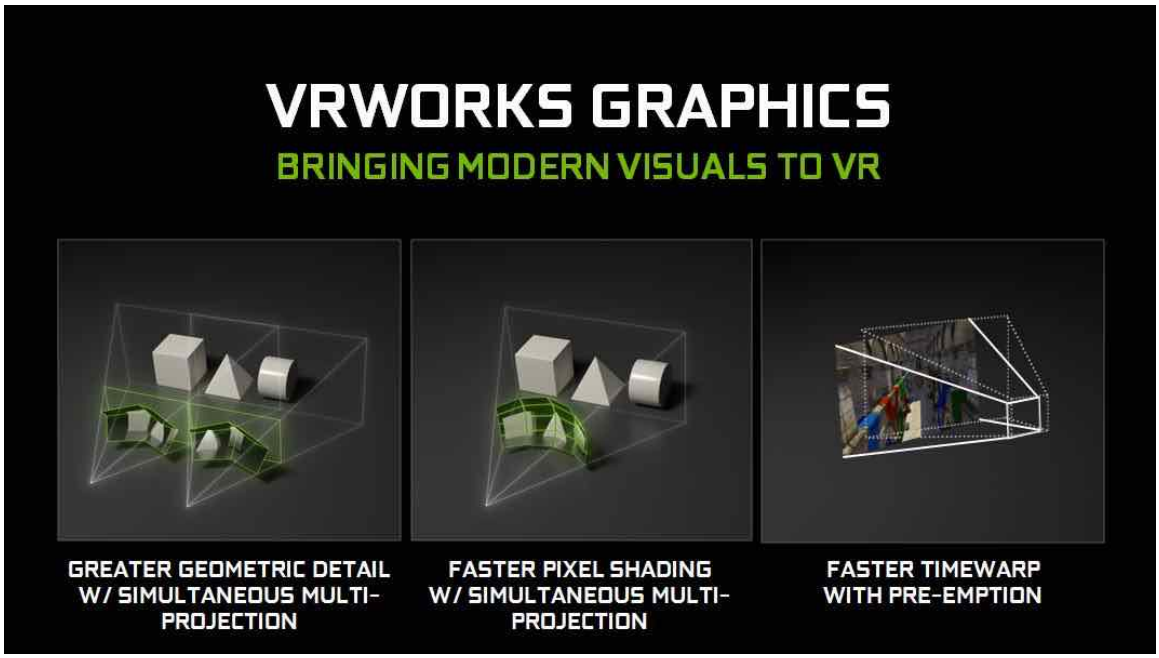


Figure 36: New VRWorks Graphics Features Enabled with Pascal Architecture

VRWorks Audio

Traditional VR and game audio provides an accurate 3D position of the audio source within a virtual environment. For example, if an enemy on your right fires at you, you will perceive the sound as coming from your right because it will be played louder in the right channel than the left channel, and will often be played slightly earlier. This simulates the difference in arrival time and arrival energy of the first waves of sound to arrive at the player, called the direct sound. The differences in energy and arrival time of the direct sound at each ear are called binaural effects.

Figure 37: Using HRTF's To Provide Direct Audio

In the real world, however, sound spreads in many directions, not just directly toward a listener. Some of that sound will bounce off surfaces and make its way to the user later than the direct sound. This is called indirect sound, reflected sound, or reverberation. The indirect sound depends on the size, shape, and material properties of the surrounding area. For instance, when walking into a small bathroom with tile walls and flooring, your footsteps will sound louder and produce more echoes compared to walking in the same bathroom with a carpeted floor and sheetrocked (drywall) walls. Sound interacts with different materials in very different ways. Some materials, like tile, reflect a large amount of sound energy, while some materials, like carpet, absorb a large amount of sound energy.

NVIDIA VRWorks Audio uses ray tracing, a technique used in generating images in computer graphics to trace the path of audio propagation through a virtual scene. VRWorks Audio simulates the propagation of acoustic energy through the surrounding environment. Rays are generated to trace the direct and indirect paths along which audio can travel from a source to a listener. When these rays encounter surfaces in the scene, called the geometry, these rays are absorbed, reflected, and scattered as a function of their angle of incidence and the material properties associated with the surface they are interacting with.

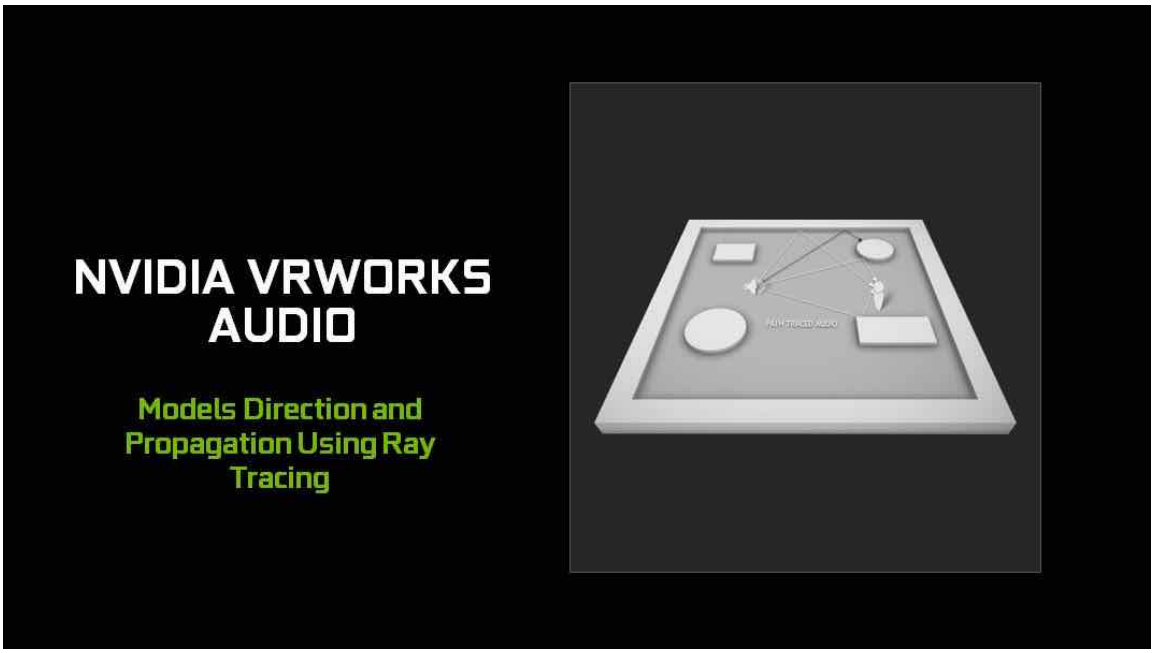


Figure 38: VRWorks Audi Uses Ray Tracing to Simulate Direct and Indirect Audio

VRWorks Audio creates the binaural effects on direct sound which gamers are accustomed to hearing today. In addition, VRWorks Audio creates indirect sound effects which give the player information about the size and structure of the space they are in.

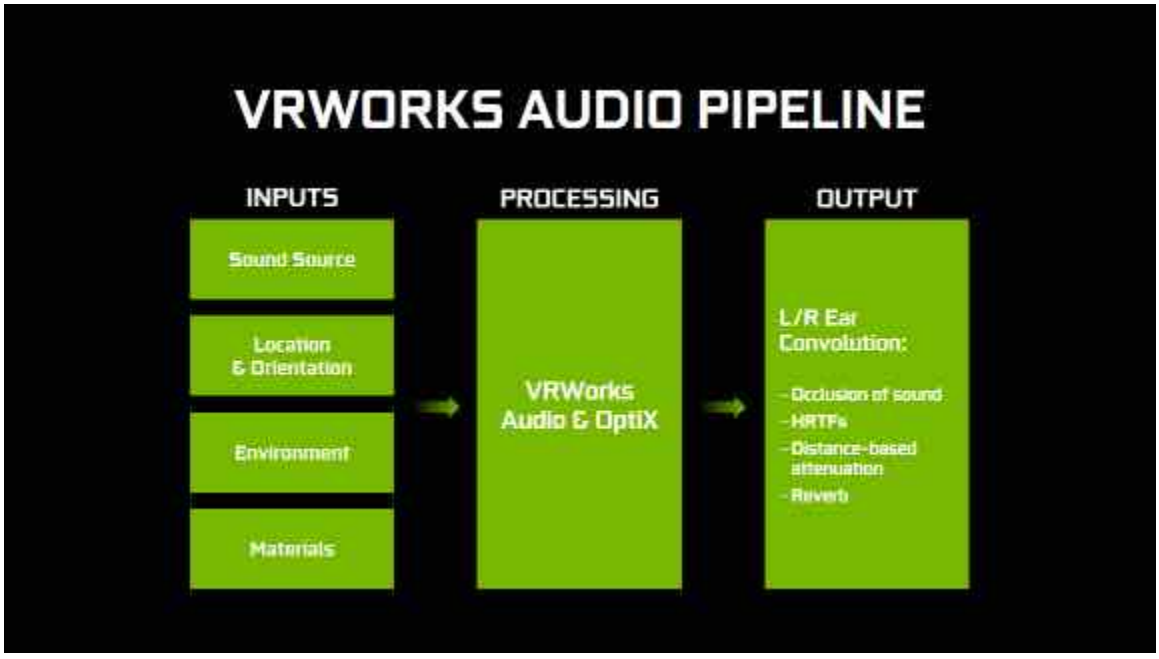


Figure 39: The VRWorks Audio Pipeline

VRWorks Audio uses the same high-performance NVIDIA OptiX ray tracing engine which is used to build ray tracing applications. OptiX can be used by games to accelerate a variety of tasks, such as accurate ambient occlusion and light baking. With VRWorks Audio, VR gamers will experience more convincing 3D audio to further enhance immersion in concert with the stunning 3D graphics produced by their GeForce GTX 1080 GPU.

PhysX for VR Touch & Environmental Simulation

Realistically modelling touch interactions and environment behavior is critical for delivering full presence in VR. Today's VR experiences deliver touch interactivity through a combination of positional tracking, hand controllers, and haptics. NVIDIA's PhysX Constraint Solver detects when a hand controller interacts with a virtual object and enables the game engine to provide a physically-accurate visual and haptic response.

Figure 40: VR Touch uses PhysX to Provide Realistic Haptic Feedback to Touch Controllers

PhysX also models the physical behavior of the virtual world around you so that all interactions, whether it be an explosion or hand splashing through water, are accurate and behave as in the real world.

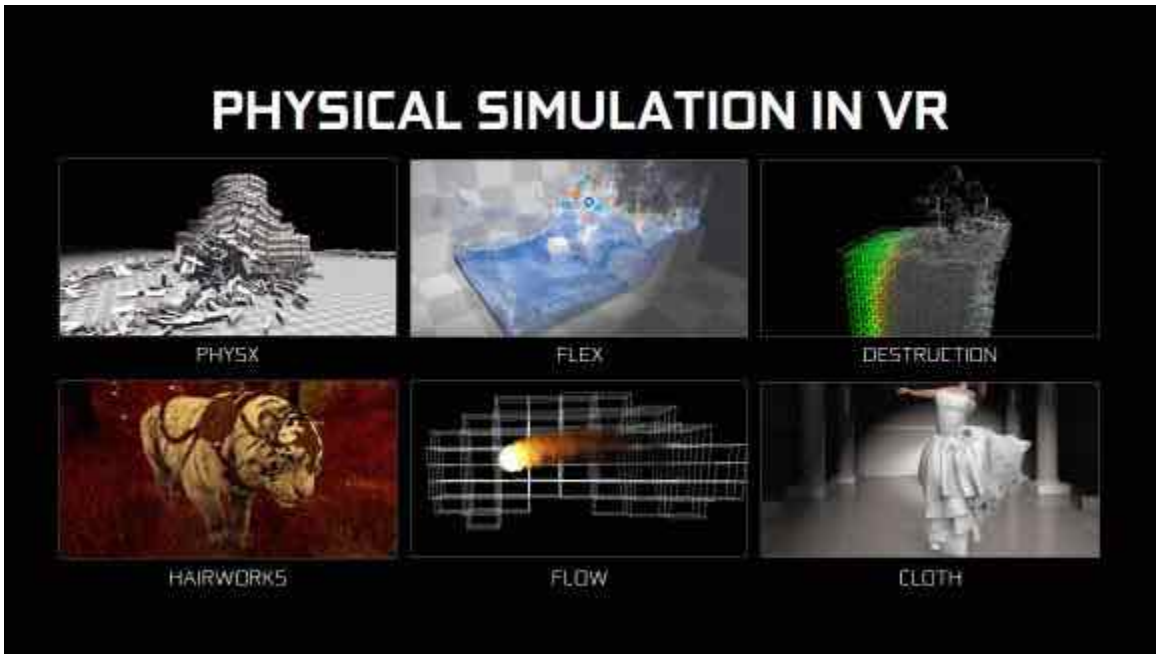


Figure 41: With PhysX, the New VRWorks Suite Supports More Realistic Physics

Conclusion

The GeForce GTX 1080 has been designed to deliver groundbreaking performance for the next generation of DirectX 12 and Vulkan games: it is the world's fastest GPU.

At the heart of the GeForce GTX 1080 lies NVIDIA's Pascal architecture. Pascal is the most advanced GPU architecture that has ever been built. The circuit paths of Pascal GPUs have been meticulously crafted to deliver blazing clock speeds. The GeForce GTX 1080 ships with a GPU Boost Clock of more than 1.7 GHz. The 16 nm FinFET manufacturing process and numerous architectural enhancements at both the GPU and board level make the GeForce GTX 1080 the most power-efficient graphics card ever built.

The GeForce GTX 1080 is also the first graphics card to ship with GDDR5X memory. GDDR5X is the next generation of high-speed DRAM, and enables extremely high data rates—the GeForce GTX 1080's GDDR5X memory operates at 10 Gbps. A new GPU circuit architecture and board channel design were needed to make these record-breaking speeds possible. But NVIDIA engineers did not stop there. The GeForce GTX 1080 also features an enhanced delta color compression capability that allows the GPU to more efficiently use its available memory bandwidth. As a result, the GeForce GTX 1080's memory subsystem effectively has up to 1.7x more bandwidth than its direct predecessor, the GeForce GTX 980.

Simultaneous Multi-Projection provides breakthrough performance for new display technologies. With this feature, the GPU can simultaneously render to unique viewports that are tailored for the needs of VR headsets and triple-display users. Lens Matched Shading leverages the GeForce GTX 1080's Simultantaneous Multi-Projection technology to improve pixel shading performance by rendering to up to 16 viewports that more closely match the unique shape of today's latest VR headsets. This avoids rendering many pixels that would otherwise be discarded before the image is sent to the VR headset. The new Single Pass Stereo feature uses Simultaneous Multi-Projection to render the geometry needed for both eyes in a VR headset in one rendering pass rather than a pass for each eye, effectively halving the geometry workload compared to traditional VR rendering.

Simultaneous Multi-Projection is beneficial for more use cases than just VR. Gamers with three displays will typically tilt the left and right displays inwards because it more closely represents peripheral vision and saves desk space. But doing this causes the images on the left and right displays to look out of scale in relation to the center display. With Perspective Surround, the GPU simultaneously renders distinct projections for each display with proper perspective views, so the user can see the world as it should look.

NVIDIA has also integrated new features into VRWorks to make VR more immersive than ever. Physics is incredibly important to delivering VR worlds that resemble real life. Therefore PhysX has been integrated into VR. PhysX models the physical behavior of objects so that any game interaction—whether it is an explosion that sends debris flying, or a flag waving in the wind—behaves as it does in the real world. NVIDIA VR Touch uses PhysX to detect when a hand controller interacts with a virtual object, and enables the game engine to provide a physically-accurate visual and haptic response. And finally, with VRWorks Audio, NVIDIA harnesses the power of the GPU to accurately simulate audio propagation— the paths sound takes passing through its surrounding environment. This allows the GPU

to precisely recreate the indirect sounds that occur when sound reflects off of objects, and is absorbed differently depending on the material that sound interacts with. Propagation is the missing ingredient that is lacking in today's existing audio solutions.

As the new flagship of the GeForce GTX lineup, gamers and hardware enthusiasts who crave the best visuals and performance should look no further than the GeForce GTX 1080. With revolutionary performance, extraordinary power efficiency, and support for new features that will enhance your VR experience, the GeForce GTX 1080 is gaming perfected.

Notice

ALL INFORMATION PROVIDED IN THIS WHITE PAPER, INCLUDING COMMENTARY, OPINION, 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 MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

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.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

DirectX

DirectX 12, DirectX, and DirectX Logo, are registered trademarks of Microsoft Corporation.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, FERMI, KEPLER, MAXWELL, PASCAL, TITAN, Tesla, GeForce, and SLI 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