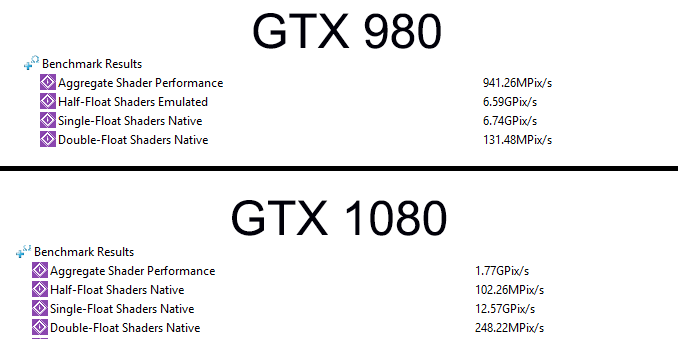
https://www.anandtech.com/show/10325/the-nvidia-geforce-gtx-1080-and-1070-founders-edition-review/5

Low precision operations are in turn seen by NVIDIA as one of the keys into further growing their increasingly important datacenter market, as deep learning and certain other tasks are themselves rapidly growing fields. Pascal isn’t just faster than Maxwell overall, but when it comes to FP16 operations on the FP16x2 core, Pascal is a **lot** faster, with theoretical throughput over similar Maxwell GPUs increasing by over three-fold thanks to the combination of overall speed improvements and double speed FP16 execution.

GeForce GTX 1080, on the other hand, is **not** faster at FP16. In fact it’s downright slow. For their consumer cards, NVIDIA has severely limited FP16 CUDA performance. GTX 1080’s FP16 instruction rate is 1/128th its FP32 instruction rate, or after you factor in vec2 packing, the resulting theoretical performance (in FLOPs) is 1/64th the FP32 rate, or about 138 GFLOPs.

[](https://images.anandtech.com/doci/10325/SandaThroughput.png)

After initially testing FP16 performance with SiSoft Sandra – one of a handful of programs with an FP16 benchmark built against CUDA 7.5 – I reached out to NVIDIA to confirm whether my results were correct, and if they had any further explanation for what I was seeing. NVIDIA was able to confirm my findings, and furthermore that the FP16 instruction rate and throughput rates were different, confirming in a roundabout manner that GTX 1080 was using vec2 packing for FP16.

As it turns out, when it comes to FP16 NVIDIA has made another significant divergence between the HPC-focused GP100, and the consumer-focused GP104. On GP100, these FP16x2 cores are used throughout the GPU as both the GPU’s primarily FP32 core and primary FP16 core. However on GP104, NVIDIA has retained the old FP32 cores. The FP32 core count as we know it is for these pure FP32 cores. What isn’t seen in NVIDIA’s published core counts is that the company has built in the FP16x2 cores separately.

To get right to the point then, each SM on GP104 only contains a single FP16x2 core. This core is in turn only used for executing native FP16 code (i.e. CUDA code). It’s not used for FP32, and it’s not used for FP16 on APIs that can’t access the FP16x2 cores (and as such promote FP16 ops to FP32). The lack of a significant number of FP16x2 cores is why GP104’s FP16 CUDA performance is so low as listed above. There is only 1 FP16x2 core for every 128 FP32 cores.

Limiting the performance of compute-centric features in consumer parts is nothing new for NVIDIA. FP64 has been treated as a Tesla feature since the beginning, and consumer parts have either shipped with a very small number of FP64 CUDA cores for binary compatibility purposes, or when a GeForce card uses an HPC-class GPU, FP64 performance is artificially restricted. This allows NVIDIA to include a feature for software development purposes while enforcing strict market segmentation between the GeForce and Tesla products. However in the case of FP64, performance has never been slower than 1/32, whereas with FP16 we’re looking at a much slower 1/128 instruction rate. Either way, the end result is that like GP104’s FP64 support, GP104’s FP16 support is almost exclusively for CUDA development compatibility and debugging purposes, not for performant consumer use.

<https://devblogs.nvidia.com/parallelforall/mixed-precision-training-deep-neural-networks/#more-8452>

**ACCUMULATION INTO FP32**

The NVIDIA Volta GPU architecture introduces Tensor Core instructions, which multiply half precision matrices, accumulating the result into either single- or half-precision output. We found that accumulation into single precision is critical to achieving good training results. Accumulated values are converted to half precision before writing to memory. The cuDNN and CUBLAS libraries provide a variety of functions that rely on Tensor Cores for arithmetic.

<http://www.nvidia.com/object/gpu-architecture.html>

New half-precision, 16-bit floating point instructions deliver over 21 TeraFLOPS for unprecedented training performance. With 47 TOPS (tera-operations per second) of performance, new 8-bit integer instructions in Pascal allow AI algorithms to deliver real-time responsiveness for deep learning inference.

## **https://en.wikipedia.org/wiki/Pascal\_(microarchitecture)**

### Chips[[edit](https://en.wikipedia.org/w/index.php?title=Pascal_(microarchitecture)&action=edit&section=6)]

* GP100: Nvidia Tesla P100 GPU accelerator is targeted at [GPGPU](https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units) applications such as FP64 double precision compute and deep learning training that uses FP16. It uses [HBM2 memory](https://en.wikipedia.org/wiki/HBM_2).[[18]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-InsidePascal-18) Quadro GP100 also uses the GP100 GPU.
* GP102: This GPU is used in the TITAN Xp,[[19]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-19) Titan X[[20]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-20) and the GeForce GTX 1080 Ti. It is also used in the Quadro P6000[[21]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-21) & Tesla P40.[[22]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-22)
* GP104: This GPU is used in the GeForce GTX 1070, GTX 1070 Ti and the GTX 1080. The GTX 1070 has 15/20 and the GTX 1070 Ti has 19/20 of its SMs enabled. Both are connected to GDDR5 memory, while the GTX 1080 is a full chip and is connected to GDDR5X memory. It is also used in the Quadro P5000, Quadro P4000 and Tesla P4.
* GP106: This GPU is used in the GeForce GTX 1060 with GDDR5 memory.[[23]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-23)[[24]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-24) It is also used in the Quadro P2000.
* GP107: This GPU is used in the GeForce GTX 1050 Ti and GeForce GTX 1050. It is also used in the Quadro P1000, Quadro P600 & Quadro P400.
* GP108: This GPU is used in the GeForce GT 1030.

On the GP104 chip an SM consists of 128 single-precision ALUs ("CUDA cores"), on the GP100 of 64 single-precision ALUs. Due to different organization of the chips, like number of double precision ALUs, the theoretical double precision performance of the GP100 is half of the theoretical one for single precision; the ratio is 1/32 for the GP104 chip.

## **Performance[**[**edit**](https://en.wikipedia.org/w/index.php?title=Pascal_(microarchitecture)&action=edit&section=7)**]**

The theoretical single-precision processing power of a Pascal GPU in [GFLOPS](https://en.wikipedia.org/wiki/GFLOPS) is computed as 2 (operations per FMA instruction per CUDA core per cycle) × number of CUDA cores × core clock speed (in GHz).

The theoretical double-precision processing power of a Pascal GPU is 1/2 of the single precision performance on GP100, and 1/32 on GP102, GP104, GP106, GP107 & GP108.

The theoretical half-precision processing power of a Pascal GPU is 2× of the single precision performance on GP100[[10]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-anandtech_pascal1-10) and 1/64 on GP102, GP104, GP106, GP107 & GP108.[[16]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-RyanSmithPrecision-16)

## **Successor[**[**edit**](https://en.wikipedia.org/w/index.php?title=Pascal_(microarchitecture)&action=edit&section=8)**]**

After Pascal, the next architecture will be preliminary codenamed [Volta](https://en.wikipedia.org/wiki/Volta_(microarchitecture)).[[29]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-blogs.nvidia.com-29) Nvidia announced that the Volta GPU would feature [High Bandwidth Memory](https://en.wikipedia.org/wiki/High_Bandwidth_Memory), [Unified Memory](https://en.wikipedia.org/wiki/Unified_Memory_Architecture), complete FP16 support (two times its FP32) and [NVLink](https://en.wikipedia.org/wiki/NVLink" \o "NVLink).[[29]](https://en.wikipedia.org/wiki/Pascal_(microarchitecture)#cite_note-blogs.nvidia.com-29)