Whitepaper

NVIDIA Tesla P100

The Most Advanced Datacenter Accelerator Ever Built

Featuring Pascal GPU Architecture

Table of Contents

Introduction **4**

Tesla P100: Revolutionary Performance and Features for GPU Computing **5**

- Extreme Performance for High Performance Computing and Deep Learning 6
- NVLink: Extraordinary Bandwidth for Multi-GPU and GPU-to-CPU Connectivity 7
- HBM2 High-Speed GPU Memory Architecture 8
- Simplified Programming for Developers with Unified Memory and Compute Preemption 9

GP100 GPU Hardware Architecture In-Depth **10**

- Exceptional Performance and Power Efficiency 11
- Pascal Streaming Multiprocessor 12
- Designed for High-Performance Double Precision 13
- Support for FP16 Arithmetic Speeds Up Deep Learning 14
- Better Atomics 14
- L1/L2 Cache Changes in GP100 15
- GPUDirect Enhancements 15
- Compute Capability 16
- ¢ Ŝ ă f l· t м л л Y ² 2 NJ f Ř Ω ă C λ NJ ă ú D t ¦ ó λ ú K l a н 16
  - Memory Resilience 18
- Tesla P100 Design 18

NVLink High Speed Interconnect **20**

- NVLink Configurations 21
  - GPU-to-GPU NVLink Connectivity 21
  - CPU-to-GPU NVLink Connectivity 22
- NVLink Interface to the Tesla P100 24

Unified Memory **25**

- Unified Memory History 25
- Pascal GP100 Unified Memory 27
- Benefits of Unified Memory 28

Compute Preemption **30**

NVIDIA DGX-1 Deep Learning Supercomputer **31**

- 250 Servers in a Box 31
- 12X DNN Speedup in One Year 32
- DGX-1 Software Features 32
- NVIDIA DGX-1 System Specifications 33

Conclusion **34**

Appendix A: NVLink Signaling and Protocol Technology **35**

- NVLink Controller Layers 35
  - Physical Layer (PL) 35
  - Data Link Layer (DL) 36
  - Transaction Layer 36

Appendix B: Accelerating Deep Learning and Artificial Intelligence with GPUs 37
Deep Learning in a Nutshell 37
NVIDIA GPUs: The Engine of Deep Learning 40
Tesla P100: The Fastest Accelerator for Training Deep Neural Networks 41
Comprehensive Deep Learning Software Development Kit 41
Big Data Problem Solving with NVIDIA GPUs and DNNs 42
Self-driving Cars 43
Robots 44
Healthcare and Life Sciences 44

Introduction

Nearly a decade ago, NVIDIA® pioneered the use of GPUs to accelerate computationally-intensive workloads with the introduction of the G80 GPU and the NVIDIA® CUDA® parallel computing platform. Today, NVIDIA® Tesla® GPUs accelerate thousands of High Performance Computing (HPC) applications across many areas including computational fluid dynamics, medical research, machine vision, financial modeling, quantum chemistry, energy discovery, and several others.

NVIDIA Tesla GPUs are installed [illegible] accelerating discovery and innovation, enabling increasingly complex simulations across multiple domains. Data centers are using NVIDIA Tesla GPUs to speed up numerous HPC and Big Data applications, while also enabling leading-edge Artificial Intelligence (AI) and Deep Learning systems.

NVIDIA [illegible] **NVIDIA Tesla P100** accelerator (see Figure 1) using the groundbreaking new **NVIDIA® Pascal™ GP100 GPU** takes GPU computing to the next level. This paper details both the Tesla P100 accelerator and the Pascal GP100 GPU architectures.

Also discussed is [illegible] that utilizes eight Tesla P100 accelerators effectively an AI supercomputer in a box. The DGX-1 is purpose-built to assist researchers advancing AI, and data scientists requiring an integrated system for Deep Learning.

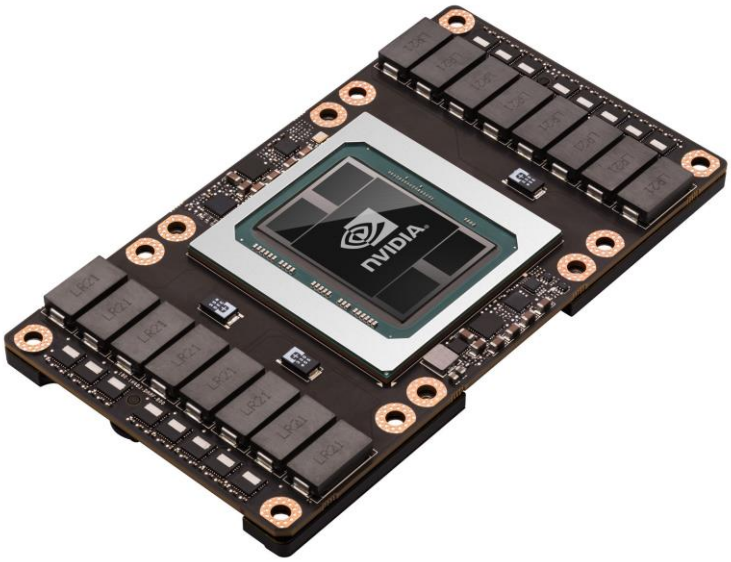

Figure 1. NVIDIA Tesla P100 with Pascal GP100 GPU

Tesla P100: Revolutionary Performance and Features for GPU Computing

With a 15.3 billion transistor GPU, a new high performance interconnect that greatly accelerates GPU peer-to-peer and GPU-to-CPU communications, new technologies to simplify GPU programming and exceptional power efficiency, Tesla P100 is not only the most powerful, but also the most architecturally complex GPU accelerator architecture ever built.

Key features of Tesla P100 include:

- **Extreme performance**
  Powering HPC, Deep Learning, and many more GPU Computing areas
- **b ± [λ y ˜ ⌊ ϰ**
  b ± L 5 L !high speed, high bandwidth interconnect for maximum application scalability
- **HBM2**
  Fast, high capacity, extremely efficient CoWoS (Chip-on-Wafer-on-Substrate) stacked memory architecture
- **Unified Memory, Compute Preemption, and New AI Algorithms**
  Significantly improved programming model and advanced AI software optimized for the Pascal architecture
- **16nm FinFET**
  Enables more features, higher performance, and improved power efficiency

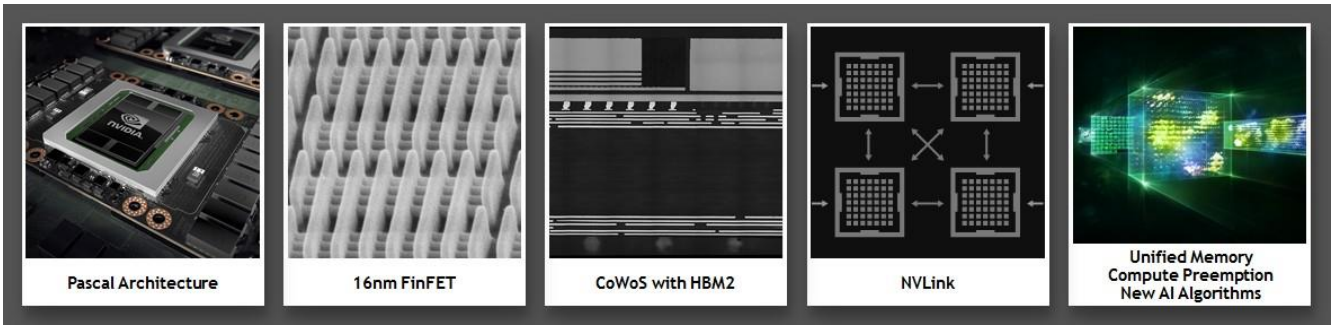

Figure 2. New Technologies in Tesla P100

Extreme Performance for High Performance Computing and Deep Learning

Tesla P100 was built to deliver exceptional performance for the most demanding compute applications, delivering:

- 5.3 TFLOPS of double-precision floating point (FP64) performance
- 10.6 TFLOPS of single-precision (FP32) performance
- 21.2 TFLOPS of half-precision (FP16) performance

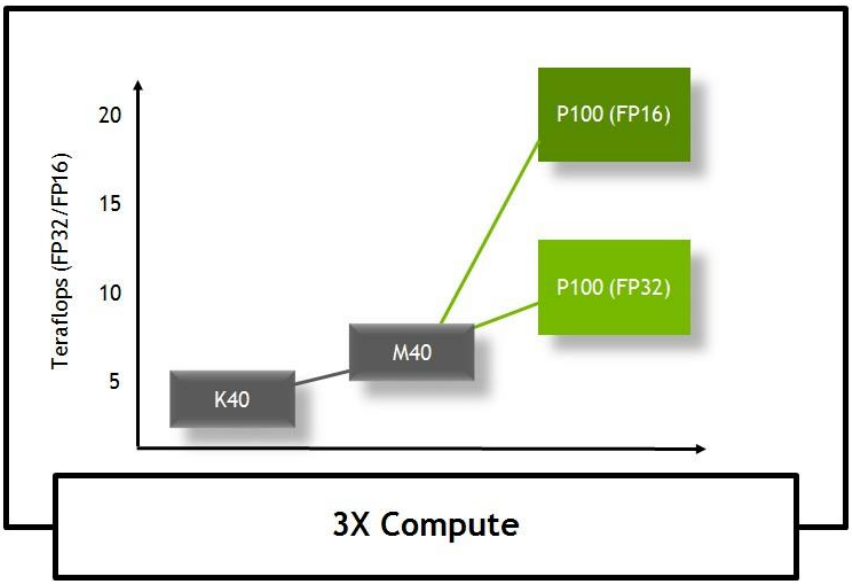

Figure 3. Tesla P100 Significantly Exceeds Compute Performance of Past GPU Generations

In addition to the numerous areas of high performance computing that NVIDIA GPUs have accelerated for a number of years, most recently Deep Learning has become a very important area of focus for GPU acceleration. NVIDIA GPUs are now at the forefront of deep neural networks (DNNs) and artificial intelligence (AI). They are accelerating DNNs in various applications by a factor of 10x to 20x compared to CPUs, and reducing training times from weeks to days. In the past three years, NVIDIA GPU-based computing platforms have helped speed up Deep Learning network training times by a factor of fifty. In the past two years, the number of companies NVIDIA collaborates with on Deep Learning has jumped nearly 35x to over 3,400 companies.

New innovations in our Pascal architecture, including native 16-bit floating point (FP) precision, allow GP100 to deliver great speedups for many Deep Learning algorithms. These algorithms do not require high levels of floating-point precision, but they gain large benefits from the additional computational power FP16 affords, and the reduced storage requirements for 16-bit datatypes.

NVLink: Extraordinary Bandwidth for Multi-GPU and GPU-to-CPU Connectivity

As GPU-accelerated computing has risen in popularity, more multi-GPU systems are being deployed at all levels, from workstations to servers, to supercomputers. Many 4-GPU and 8-GPU system configurations are now used to solve bigger and more complex problems. Multiple groups of multi-GPU systems are being interconnected using InfiBand® and 100Gb Ethernet to form much larger and more powerful systems. The ratio of GPUs to CPUs has also increased. [illegible] Titan located at the Oak Ridge National Lab, deployed one GK110 GPU per CPU daughtercard. Today, two or more GPUs are more commonly being paired per CPU as developers increasingly expose and leverage the available parallelism provided by GPUs in their applications. As this trend continues, PCIe bandwidth at the multi-GPU system level becomes a bigger bottleneck.

To address this issue, Tesla P100 features a new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

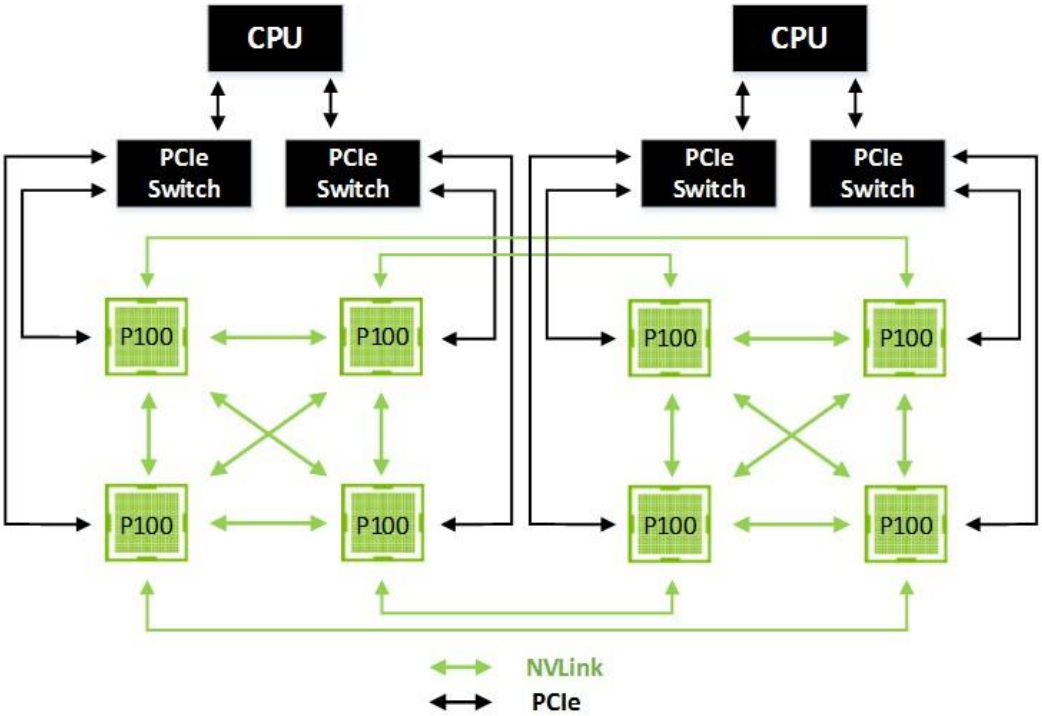

Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

Figure 5 shows the performance for various workloads, demonstrating the performance scalability a server can achieve with up to eight GP100 GPUs connected via NVLink. (Note: These numbers are measured on pre-production P100 GPUs.)

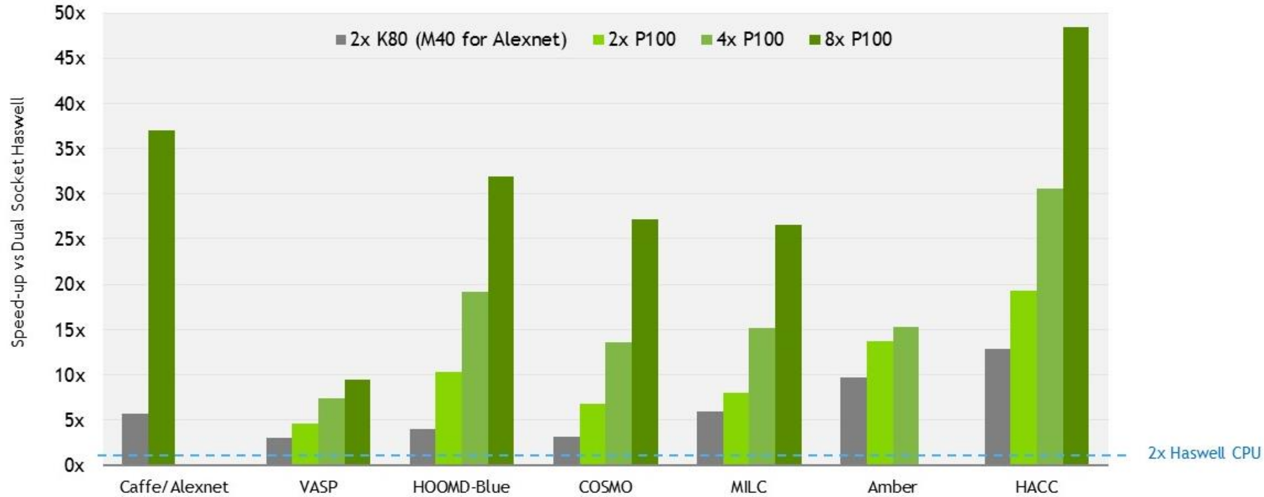

Figure 5. Largest Performance Increase with Eight P100s connected via NVLink

HBM2 High-Speed GPU Memory Architecture

Tesla P100 is the world's first GPU architecture to support HBM2 memory. HBM2 offers three times (3x) the memory bandwidth of the Maxwell GM200 GPU. This allows the P100 to tackle much larger working sets of data at higher bandwidth, improving efficiency and computational throughput, and reduce the frequency of transfers from system memory.

Because HBM2 memory is stacked memory and is located on the same physical package as the GPU, it provides considerable space savings compared to traditional GDDR5, which allows us to build denser GPU servers more easily than ever before.

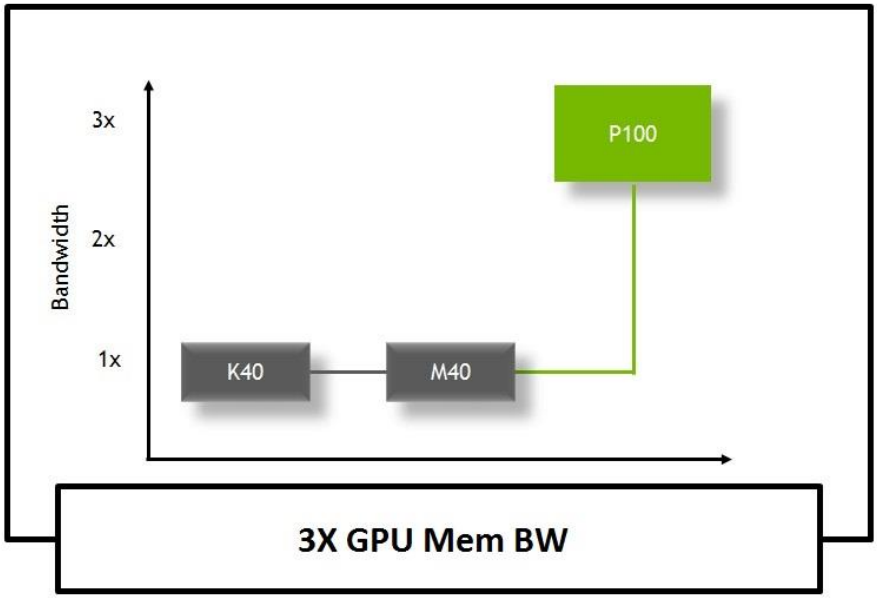

Figure 6. Tesla P100 with HBM2 Significantly Exceeds Memory Bandwidth of Past GPU Generations

Simplified Programming for Developers with Unified Memory and Compute Preemption

Unified Memory is a significant advancement for NVIDIA GPU computing and a major new hardware and software-based feature of the Pascal GP100 GPU architecture. It provides a single, seamless unified virtual address space for CPU and GPU memory. Unified Memory greatly simplifies GPU programming and porting of applications to GPUs, and also reduces the GPU computing learning curve. Programmers no longer need to worry about managing data sharing between two different virtual memory systems. GP100 is the first NVIDIA GPU to support hardware page faulting, and when combined with new 49-bit (512 TB) virtual addressing, allows transparent migration of data between the *full* virtual address spaces of both the GPU and CPU.

Compute Preemption is another important new hardware and software feature added to GP100 that allows compute tasks to be preempted at instruction-level granularity, rather than thread block granularity as in prior Maxwell and Kepler GPU architectures. Compute Preemption prevents long-running applications from either monopolizing the system (preventing other applications from running) or timing out. Programmers no longer need to modify their long-running applications to play nicely with other GPU applications. With Compute Preemption in GP100, applications can run as long as needed to process large datasets or wait for various conditions to occur, while scheduled alongside other tasks. For example, both interactive graphics tasks and interactive debuggers can run in concert with long-running compute tasks.

GP100 GPU Hardware Architecture In-Depth

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 GPU accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

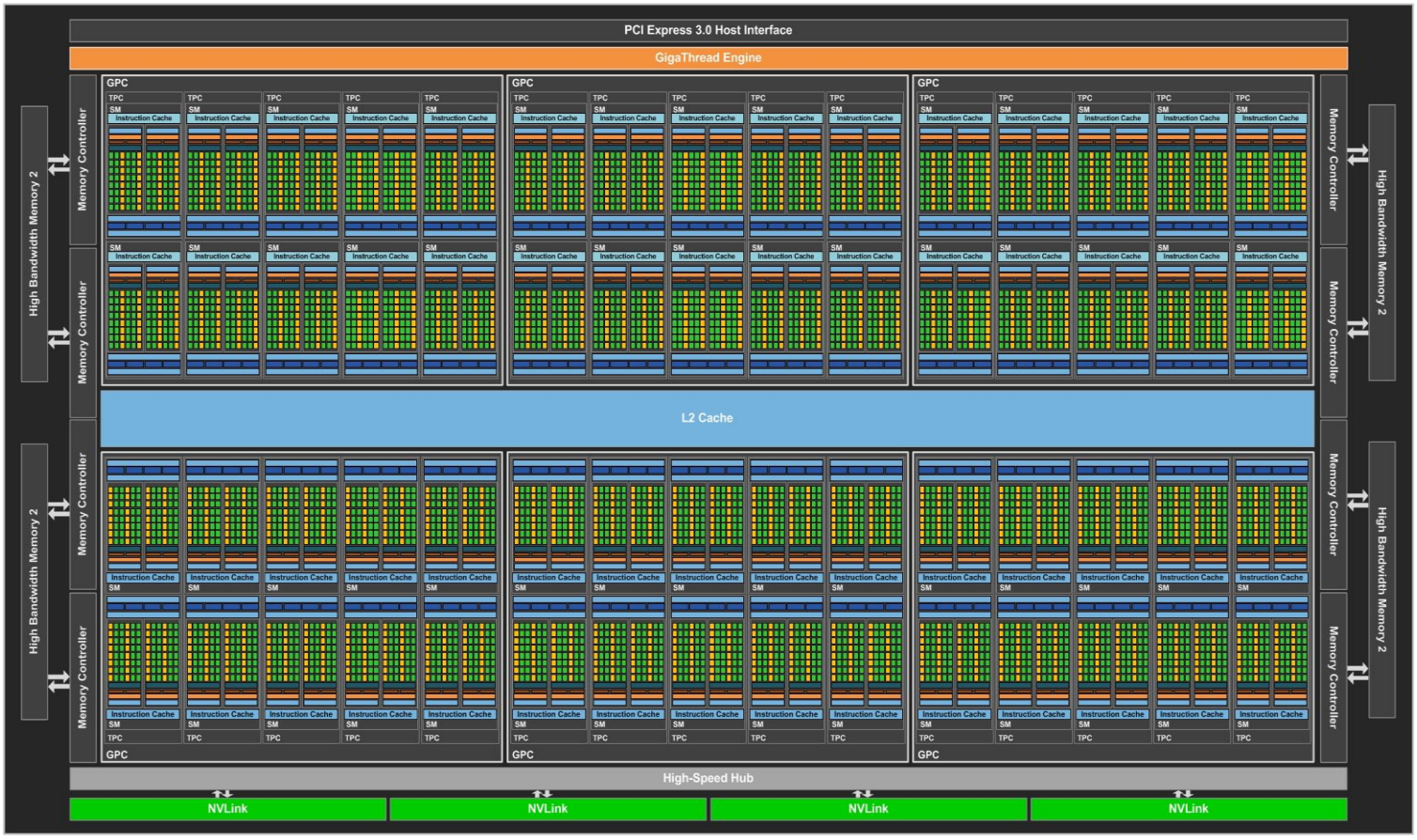

Figure 7. Pascal GP100 Full GPU with 60 SM Units

Exceptional Performance and Power Efficiency

Delivering higher performance and improving energy efficiency are two key goals for new GPU architectures. A number of changes to the SM in the Maxwell architecture improved its efficiency compared to Kepler. Pascal has built on this and incorporates additional improvements that allow us to increase performance per watt even further over Maxwell. While ¢ { a / 2nm FinFET manufacturing process plays an important role, many GPU architectural modifications were also implemented to further reduce power consumption while maintaining high performance.

Table 1. Tesla P100 Compared to Prior Generation Tesla products

Tesla Products	Tesla K40	Tesla M40	Tesla P100
GPU	GK110 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)
SMs	15	24	56
TPCs	15	24	28
FP32 CUDA Cores / SM	192	128	64
FP32 CUDA Cores / GPU	2880	3072	3584
FP64 CUDA Cores / SM	64	4	32
FP64 CUDA Cores / GPU	960	96	1792
Base Clock	745 MHz	948 MHz	1328 MHz
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz
Peak FP32 GFLOPs[1]	5040	6840	10600
Peak FP64 GFLOPs[1]	1680	210	5300
Texture Units	240	192	224
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB
Register File Size / SM	256 KB	256 KB	256 KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB
TDP	235 Watts	250 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion
GPU Die Size	551 mm²	601 mm²	610 mm²
Manufacturing Process	28-nm	28-nm	16-nm FinFET

[1] The GFLOPS in this chart are based on GPU Boost Clocks.

Pascal Streaming Multiprocessor

D t м л л Ω ă { a ƛ y Ŏ generation ESM architecture improves CUDA Core utilization and power efficiency, resulting in significant overall GPU performance improvements, and allowing higher core clock speeds compared to previous GPUs.

D t м л л Ω ă { a ƛ y Ŏ-precision (FP32) CUDA Cores compared to the Maxwell and Kepler SMs had 128 and 192 FP32 CUDA Cores, respectively. The GP100 SM is partitioned into two processing blocks, each having 32 single-precision CUDA Cores, an instruction buffer, a warp scheduler, and two dispatch units. While a GP100 SM has half the total number of CUDA Cores of a Maxwell SM, it maintains the same register file size and supports similar occupancy of warps and thread blocks. D. t м л л Ω ă { a Ƙ l ă ű Ƙ Ŝ number of registers as Maxwell GM200 and Kepler GK110. But the entire GP100 GPU has far more SMs, and thus many more registers overall. This means threads across the GPU have access to more registers, and GP100 supports more total warps, and thread blocks in flight compared to prior GPU generations.

Overall shared memory across the GP100 GPU is also increased due to the increased SM count, and aggregate shared memory bandwidth is effectively more than doubled. A higher ratio of shared memory, registers, and warps per SM in GP100 allows the SM to more efficiently execute code. There are more warps for the instruction scheduler to choose from, more loads to initiate, and more per-thread bandwidth to shared memory.

Figure 8 shows the resulting block diagram of the GP100 SM.

/ 2 Y LJ l NJ Ŝ Ř ű 2 Y Ŝ LJ f Ŝ NJ Σ t l ă Ŏ l f Ω ă { a Ŧ Ŝ l ű dz NJ Ŝ ă l ă ƛ Y LJ f Ŝ less power to manage data transfers within the SM. Pascal also provides superior scheduling and overlapped load/store instructions to increase floating point utilization. The new SM scheduler architecture in GP100 improves upon the advances of the Maxwell scheduler and is even more intelligent, providing increased performance and reduced power consumption. Each warp scheduler (one per processing block) is capable of dispatching two warp instructions per clock.

O y Ŝ y Ŝ ø Ŏ l LJ l ō ƛ f ƛ ű è ű Ƙ l ű Ƙ l ă ō Ŝ Ŝ y l Ř Ř Ŝ Ř ű 216-bit D t м л л Ω ă and 32-bit precision instructions and data, as described later in this paper. FP16 operation throughput is up to twice FP32 operation throughput.

Figure 8. Pascal GP100 SM Unit

Designed for High-Performance Double Precision

Double precision arithmetic is at the heart of many HPC applications such as linear algebra, numerical simulation, and quantum chemistry. Therefore, one of the key design goals for GP100 was to significantly improve the delivered performance for these use cases.

Each SM in GP100 features 32 double precision (FP64) CUDA Cores, which is half the number of FP32 single precision CUDA Cores. A full GP100 GPU has 1920 FP64 CUDA Cores. This 2:1 ratio of single precision (SP) units to double precision (DP) dz ƴ ƛ ű ă ŀ ƒ ƛ Ǝ ƴ ă ō Ŝ ű ű ŜNJ ǿ ƛ ű K D t м л л Ω allowing the GPU to process DP workloads more efficiently. Like previous GPU architectures, GP100 supports full IEEE 754-2008 compliant single precision and double precision arithmetic, including support for the fused multiply-add (FMA) operation and full speed support for denormalized values.

> Note: Kepler GK110 had a 3:1 ratio of SP units to DP units.

Support for FP16 Arithmetic Speeds Up Deep Learning

Deep learning is one of the fastest growing fields of computing. It is a critical ingredient in many important applications, including real-time language translation, highly accurate image recognition, automatic image captioning, autonomous driving object recognition, optimal path calculations, collision avoidance, and others. Deep learning is a two-step process.

- First, a neural network must be trained.
- Second, the network is deployed in the field to run inference computations, where it uses the results of previous training to classify, recognize, and generally process unknown inputs.

Compared to CPUs, GPUs can provide tremendous performance speedups for deep learning training and inference.

Unlike other technical computing applications that require high-precision floating-point computation, deep neural network architectures have a natural resilience to error due to the backpropagation algorithm used in their training. In fact, to avoid overfitting a network to a training dataset, approaches such as dropout aim at ensuring a trained network generalizes well and is not overly reliant on the l· Ŏ Ŏ dz NJ l· Ŏ è 2 Ŧ ó 2 NJ Ŝ NJ NJ 2 NJ ă. ƛ ƴ ú l· ƴ è ∃ ƛ Ø Ŝ ƴ dz ƴ ƛ ú Ω ă Ŏ 2 Y LJ dz

Storing FP16 data compared to higher precision FP32 or FP64 reduces memory usage of the neural network and thus allows training and deploying of larger networks. Using FP16 computation improves performance up to 2x compared to FP32 arithmetic, and similarly FP16 data transfers take less time than FP32 or FP64 transfers.

> **Note:** In GP100, two FP16 operations can be performed using a single-operation instruction.

Architectural improvements in GP100, combined with support for FP16 datatypes, allow significantly reduced Deep Learning processing times compared to what was achievable just last year.

Better Atomics

Atomic memory operations are important in parallel programming, allowing concurrent threads to correctly perform read-modify-write operations on shared data structures.

Kepler featured shared memory atomic operations of the same form as Fermi. Both architectures implemented shared memory atomics using a lock/update/unlock pattern that could be expensive in the case of high contention for updates to particular locations in shared memory.

Maxwell improved atomic operations by implementing native hardware support for shared memory atomic operations for 32-bit integers, and native shared memory 32-bit and 64-bit compare-and-swap (CAS), which can be used to implement other atomic functions with reduced overhead (compared to the Fermi and Kepler methods which were implemented in software).

GP100 builds upon Maxwell by also improving atomic operations using new Unified Memory and NVLink features (described in the following paragraph). The atomic addition operation in global memory has been extended to include FP64 data. The atomicAdd() function in CUDA now applies to 32-bit and 64

integer and floating-point data. The rounding mode for floating-point is *round-to-nearest-even* for all floating-point atomic add operations (formerly, FP32 atomic addition used *round-to-zero*).

L1/L2 Cache Changes in GP100

While Fermi and Kepler GPUs featured 64 KB configurable shared memory and L1 cache that could split the allocation of memory between L1 and shared memory functions depending on workload, beginning with Maxwell, the cache hierarchy was changed. The GP100 SM has its own dedicated pool of shared memory (64 KB/SM) and an L1 cache that can also serve as a texture cache depending on workload. The unified L1/texture cache acts as a coalescing buffer for memory accesses, gathering up the data requested by the threads of a warp prior to delivery of that data to the warp.

> **Note:** One CUDA Thread Block cannot allocate 64 KB of shared memory itself, but two Thread Blocks could use 32 KB each, etc..

A dedicated shared memory per SM means applications no longer need to select a preference of the L1/shared split for optimal performance, and the full 64 KB per SM is always available for shared memory.

GP100 features a unified 4096 KB L2 cache that provides efficient, high speed data sharing across the D t ¦ Φ L y ˜ Ŏ 2 Y LJ l · NJ λ ă 2 y ˜ Σ KB, while GM200 shipped with 3072 KB of L2 cache. With more cache located on-chip, fewer requests to the D t ¦ DRAM are needed, which reduces overall board power, reduces memory bandwidth demand, and improves performance.

GPUDirect Enhancements

Whether you are working through mountains of geological data, or researching solutions to complex scientific problems, you need a computing platform that delivers the highest data throughput and lowest latency possible. GPUDirect is a capability that enables GPUs within a single computer, or GPUs in different servers located across a network, to directly exchange data without needing to go to CPU/system memory.

The RDMA feature in GPUDirect introduced in Kepler GK110 allows third party devices such as InfiniBand (IB) adapters, network interface cards (NICs), and SSDs to directly access memory on multiple GPUs within the same system, eliminating unnecessary memory copies, dramatically lowering CPU overhead, and significantly decreasing the latency of MPI send and receive messages to/from GPU memory. It also reduces the demands on system memory bandwidth and frees GPU DMA engines for use by other CUDA tasks.

GP100 doubles the delivered RDMA bandwidth reading data from the source GPU memory and writing to the target NIC memory over PCIe. Doubling the bandwidth of GPUDirect is very important for many use cases, especially Deep Learning. In fact, Deep Learning machines have a high ratio of GPUs to CPUs (in some cases 8 GPUs per CPU), so it is very important for the GPUs to interact quickly with IO without falling back to the CPU for data transfers.

Compute Capability

The GP100 GPU supports the new Compute Capability 6.0. Table 2 compares the parameters of different Compute Capabilities for NVIDIA GPU architectures.

Table 2. Compute Capabilities: GK110 vs GM200 vs GP100

GPU	Kepler GK110	Maxwell GM200	Pascal GP100
Compute Capability	3.5	5.2	6.0
Threads / Warp	32	32	32
Max Warps / Multiprocessor	64	64	64
Max Threads / Multiprocessor	2048	2048	2048
Max Thread Blocks / Multiprocessor	16	32	32
Max 32-bit Registers / SM	65536	65536	65536
Max Registers / Block	65536	32768	65536
Max Registers / Thread	255	255	255
Max Thread Block Size	1024	1024	1024
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB

Tesla P100 Y ² 2 NJ f Ř Ω ă CBM2 J ă ű D t ¦ ó λ ű K

As the use of GPUs to accelerate compute applications has grown greatly in recent years, so has the appetite for data in many of those applications. Much larger problems are being solved by GPUs, requiring much larger datasets and higher demand for DRAM bandwidth. To address this demand for higher raw bandwidth, Tesla P100 is the first GPU accelerator to use High Bandwidth Memory 2 (HBM2).

HBM2 enables a significant boost in DRAM bandwidth by fundamentally changing the way the DRAMs are packaged and connected to the GPU.

Rather than requiring numerous discrete memory chips surrounding the GPU as in traditional GDDR5 GPU board designs, HBM2 includes one or more vertical stacks of multiple memory dies. The memory dies are linked using microscopic wires that are created with through-silicon vias and microbumps. One GB HBM2 die contains over 5,000 through-silicon via holes. A passive silicon interposer is then used to connect the memory stacks and the GPU die. The combination of HBM2 stack, GPU, and Silicon interposer are packaged in a single 55 mm × 55 mm BGA package. See Figure 9 for an illustration of the GP100 and two HBM2 stacks, and Figure 10 for a photomicrograph of an actual P100 with GPU and memory.

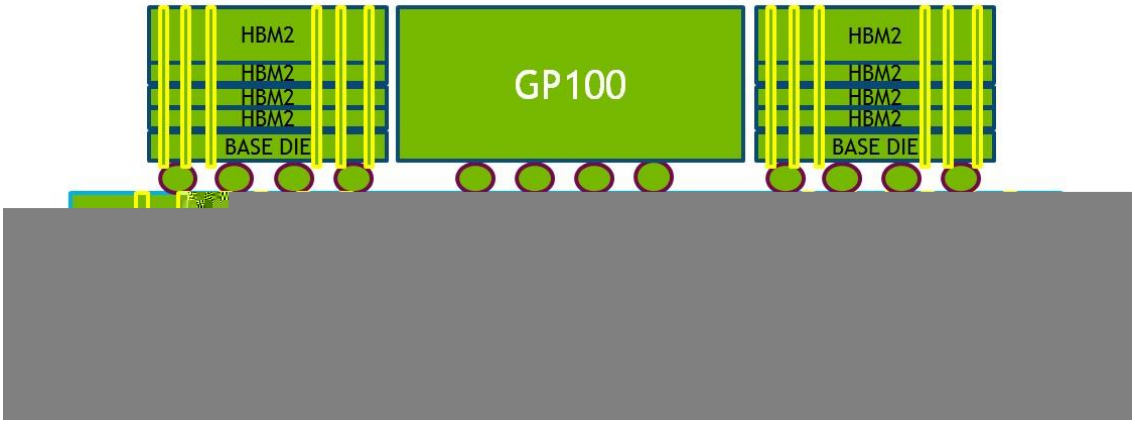

Figure 9. Cross-section Illustrating GP100 and Adjacent HBM2 stacks

Figure 10. Cross-section Photomicrograph of a P100 HBM2 stack and GP100 GPU

The photomicrograph in Figure 10 shows a cross-section of a Tesla P100 HBM2 stack and the GP100 GPU. The HBM2 stack in the upper left is built out of five dies -- a base die and 4 memory dies above it. The top memory die layer is very thick. When assembled, the top die and GPU are ground to the same height to present a coplanar surface for heat sink.

Compared to the prior HBM1 generation, HBM2 offers higher memory capacity and memory bandwidth. HBM2 supports four or eight DRAM dies per stack, while HBM1 only supports four DRAM dies per stack. HBM2 supports up to 8Gb per DRAM die, while HBM1 supports only 2Gb per die. Where HBM1 was limited to 125GB/sec of bandwidth per stack, P100 supports 180GB/sec per stack with HBM2.

As shown in the GP100 full-chip block diagram (Figure 7), the GP100 GPU connects to four HBM2 DRAM stacks. Two 512-bit memory controllers connect to each HBM2 stack for an effective 4096-bit-wide HBM2 memory interface. Initially, Tesla P100 accelerators will ship with four 4-die HBM2 stacks, for a total of 16 GB of HBM2 memory.