

CS 380 - GPU and GPGPU Programming

Lecture 2: Introduction, Pt. 2

Markus Hadwiger, KAUST

Reading Assignment #1 (until Sep 7)



Read (required):

- Programming Mass. Parallel Proc. book, 4th ed., Chapter 1 (*Introduction*)
- Programming Mass. Parallel Proc. book, 2nd ed., Chapter 2 (*History of GPU Computing*)
- OpenGL Shading Language (orange) book, Chapter 1 (*Review of OpenGL Basics*)

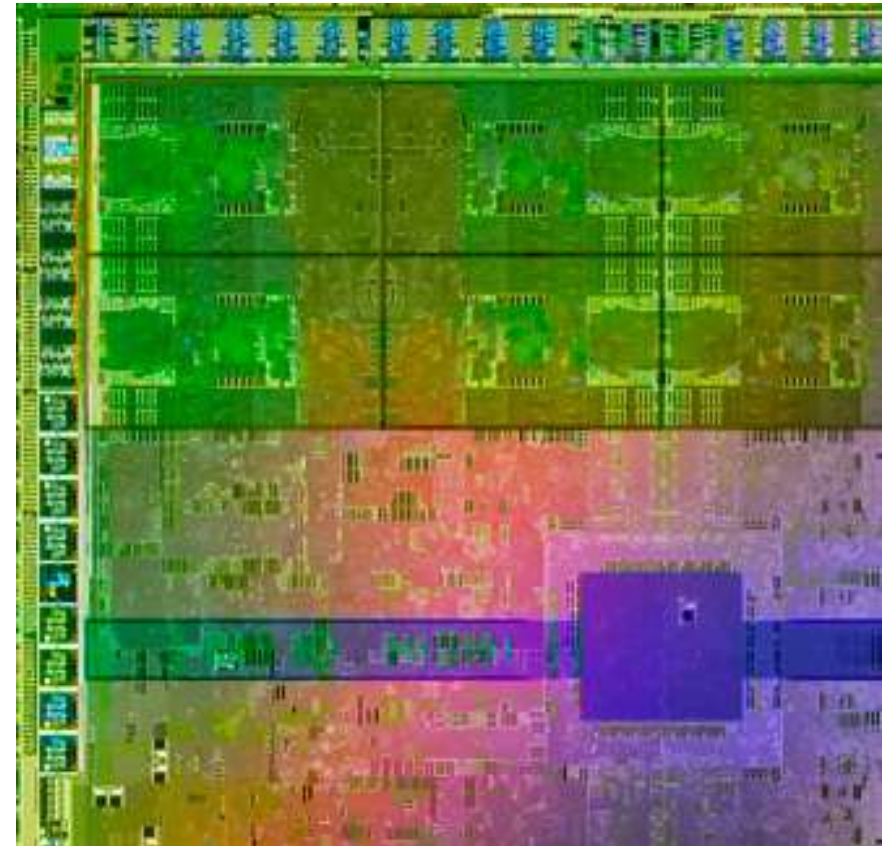
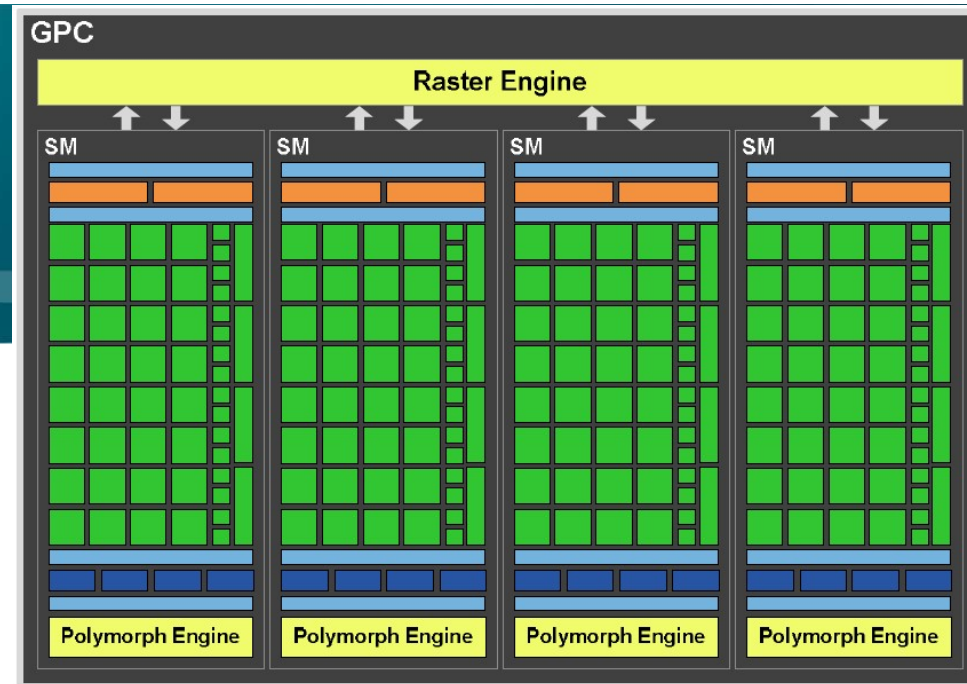
Read (optional):

- OpenGL Shading Language 4.6 (current: Aug 14, 2023) specification: Chapter 2
<https://www.khronos.org/registry/OpenGL/specs/gl/GLSLangSpec.4.60.pdf>
- Download OpenGL 4.6 (current: May 5, 2022) specification
<https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf>

Syllabus (1)

GPU Basics and Architecture (~September, early October)

- Introduction
- **GPU architecture**
- How compute/shader cores work
- GPU shading and GPU compute APIs
 - General concepts and overview
 - Learn syntax details on your own !
 - CUDA book
 - GLSL book
 - Vulkan tutorial
 - online resources, ...

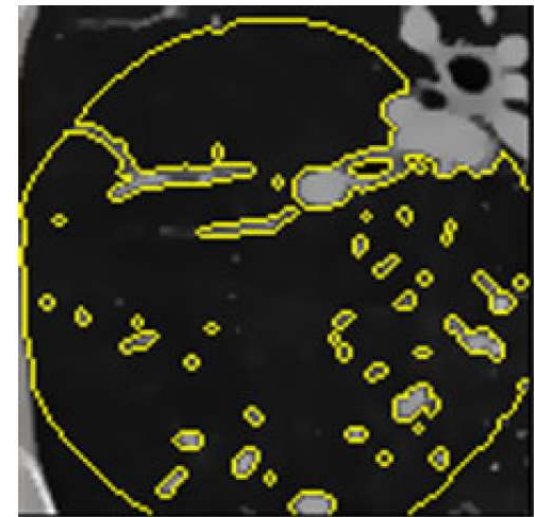


Syllabus (2)



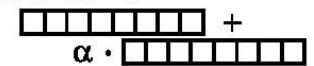
More GPU Computing (~October)

- GPGPU, important parallel programming concepts
- CUDA memory access
- Reduction, scan
- Linear algebra on GPUs
- Deep learning on GPUs
- Combining graphics and compute
 - Display the results of computations
 - Interactive systems (fluid flow, ...)

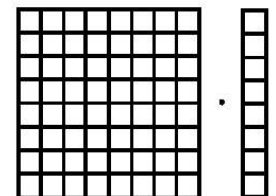


segmentation

SAXPY



SGEMV

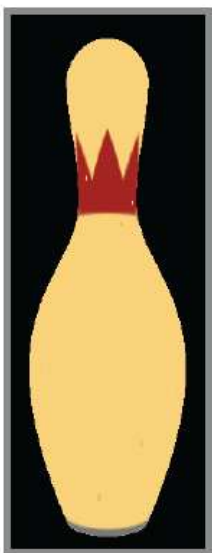


linear algebra

Syllabus (3)

GPU Graphics (~November)

- GPU (virtual) texturing, filtering
- GPU (texture) memory management
- Neural rendering, neural shading
- Modern game engine technologies



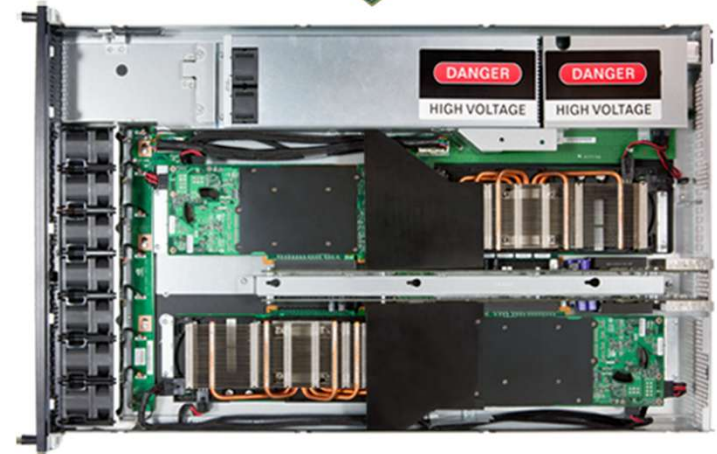
What are GPUs?



Graphics Processing Units

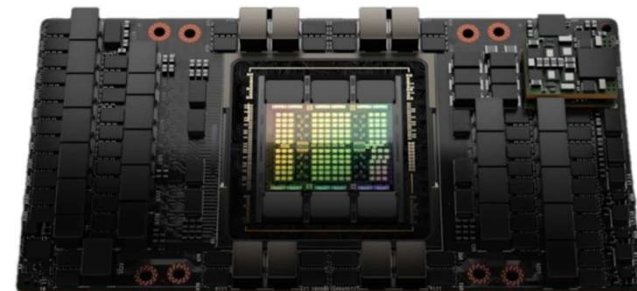
But evolved toward

- Very flexible, massively parallel floating point co-processors
- But not entirely programmable!
- Fixed-function parts have definite advantages (e.g., texture filtering, z-buffering)



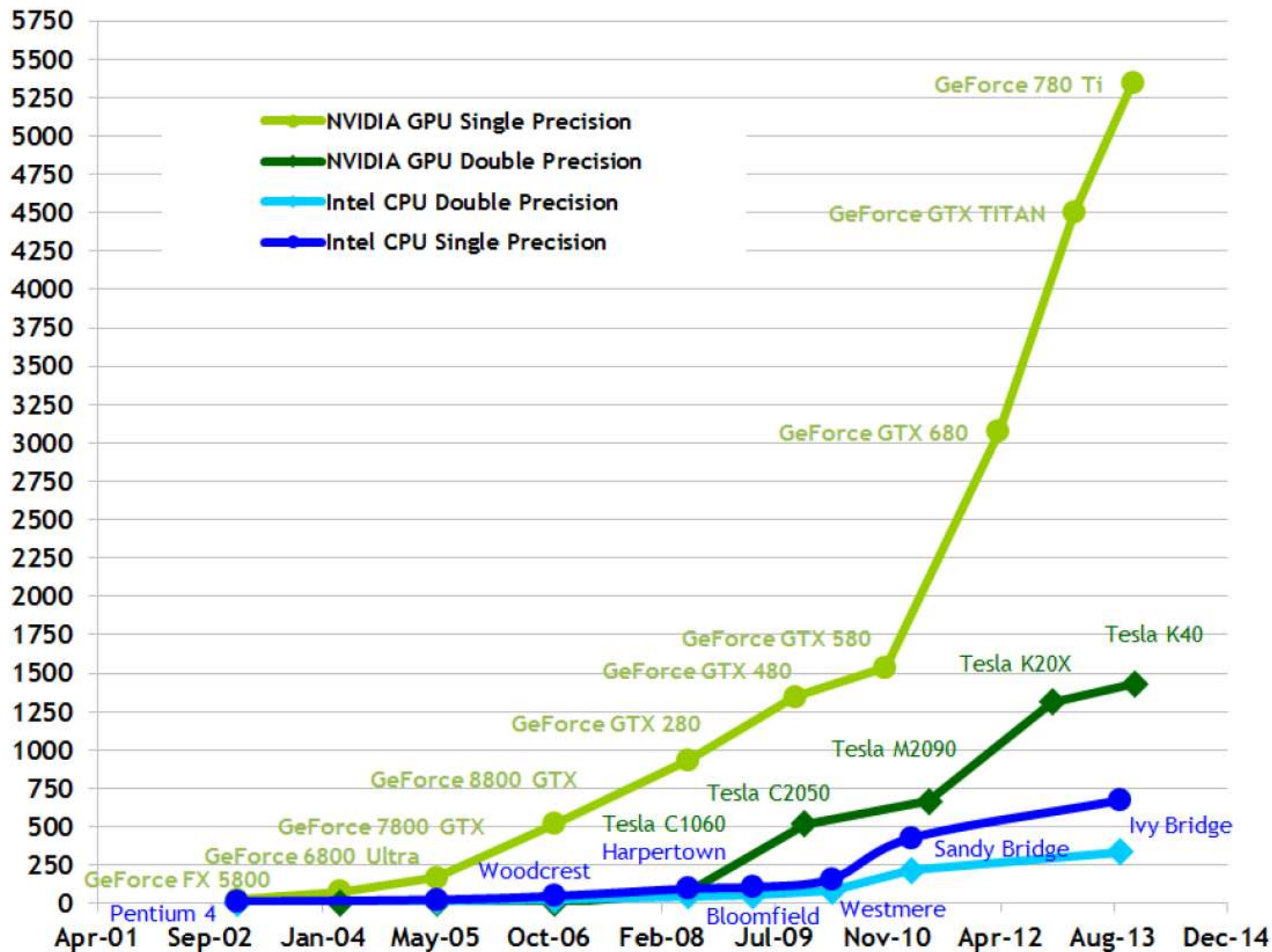
We will cover both perspectives

- GPUs for graphics
- GPU for compute (GPGPU – general purpose computation on GPU)



Theoretical GFLOP/s

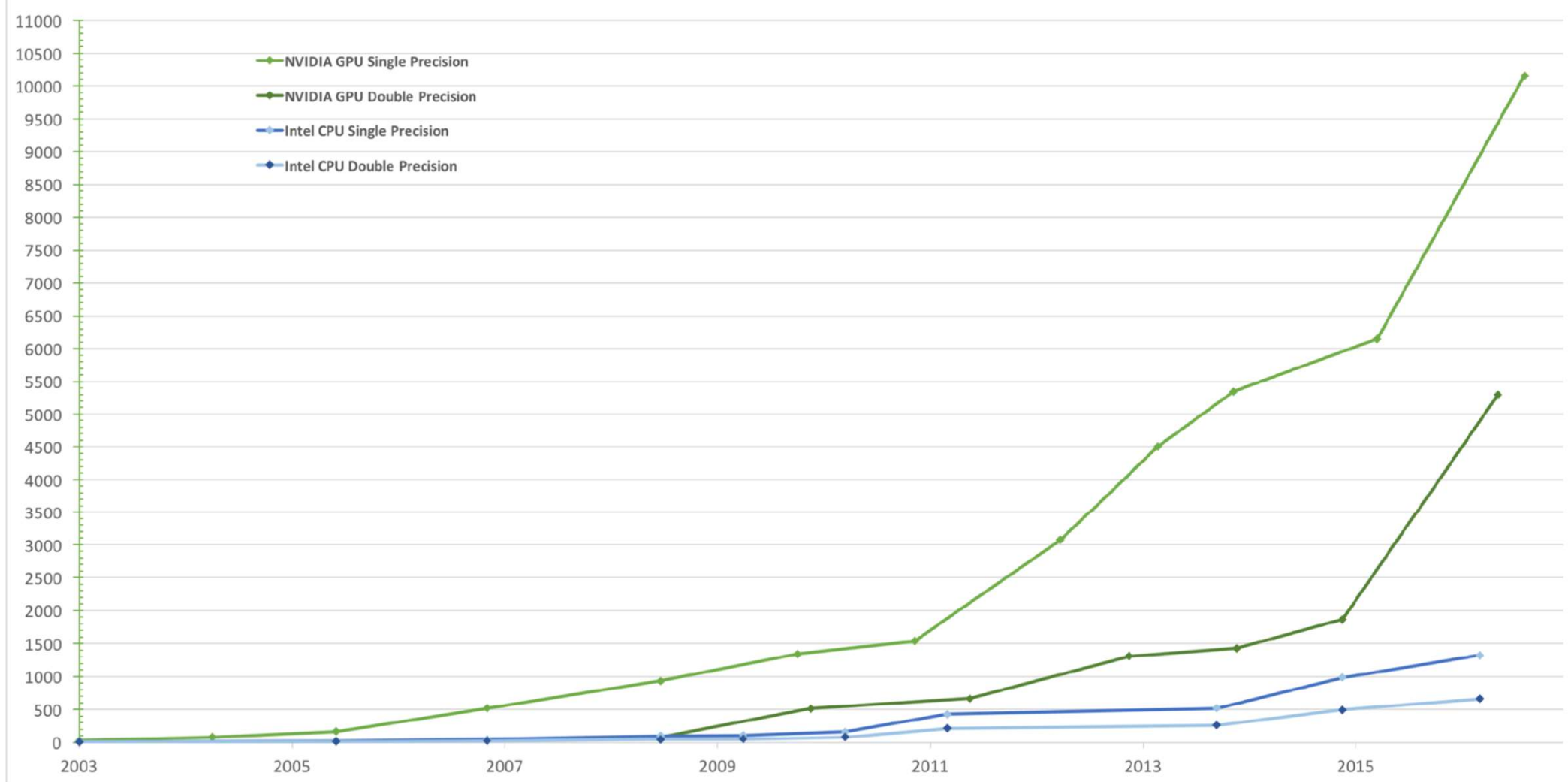
Peak Performance



Peak Performance

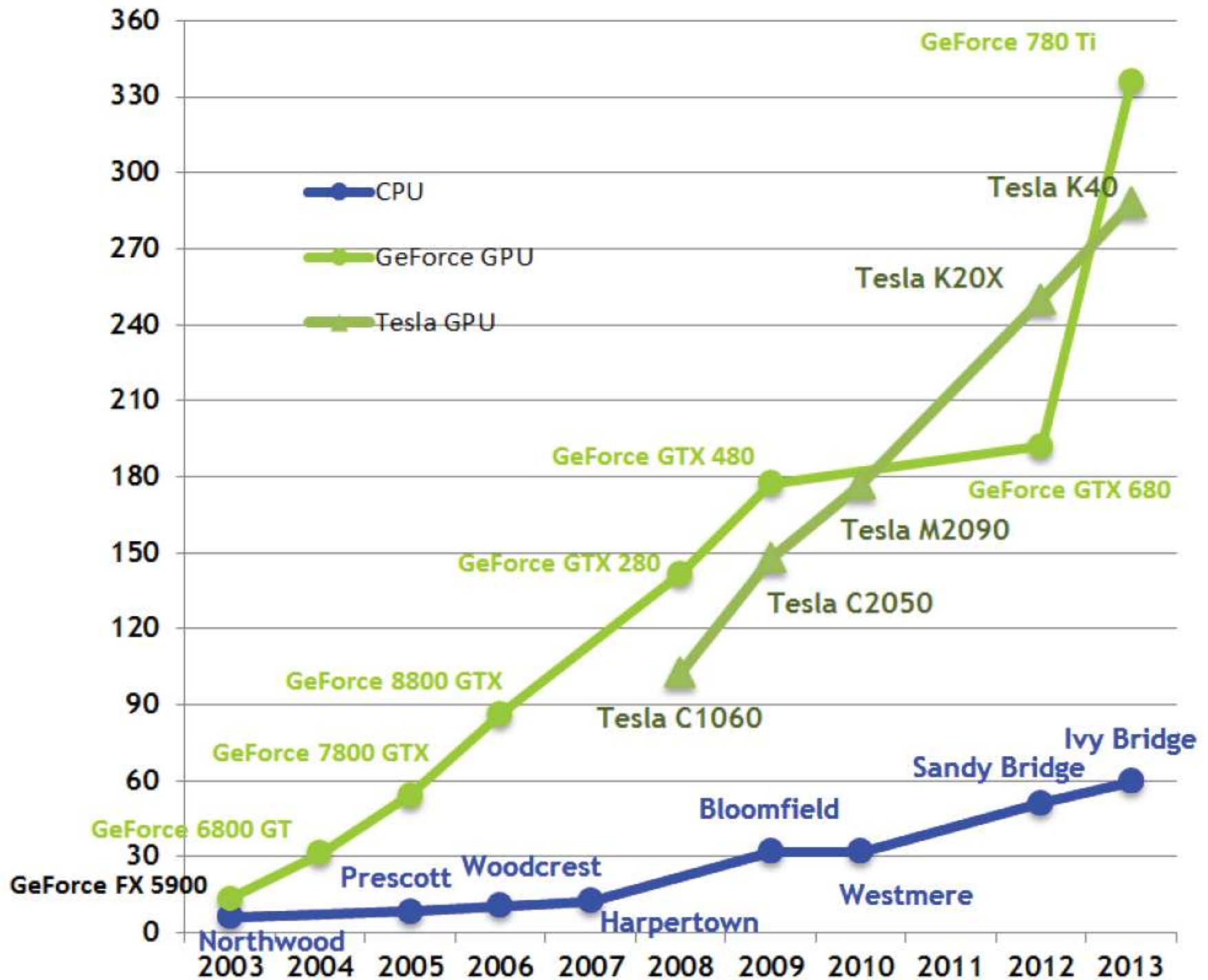


Theoretical GFLOP/s at base clock

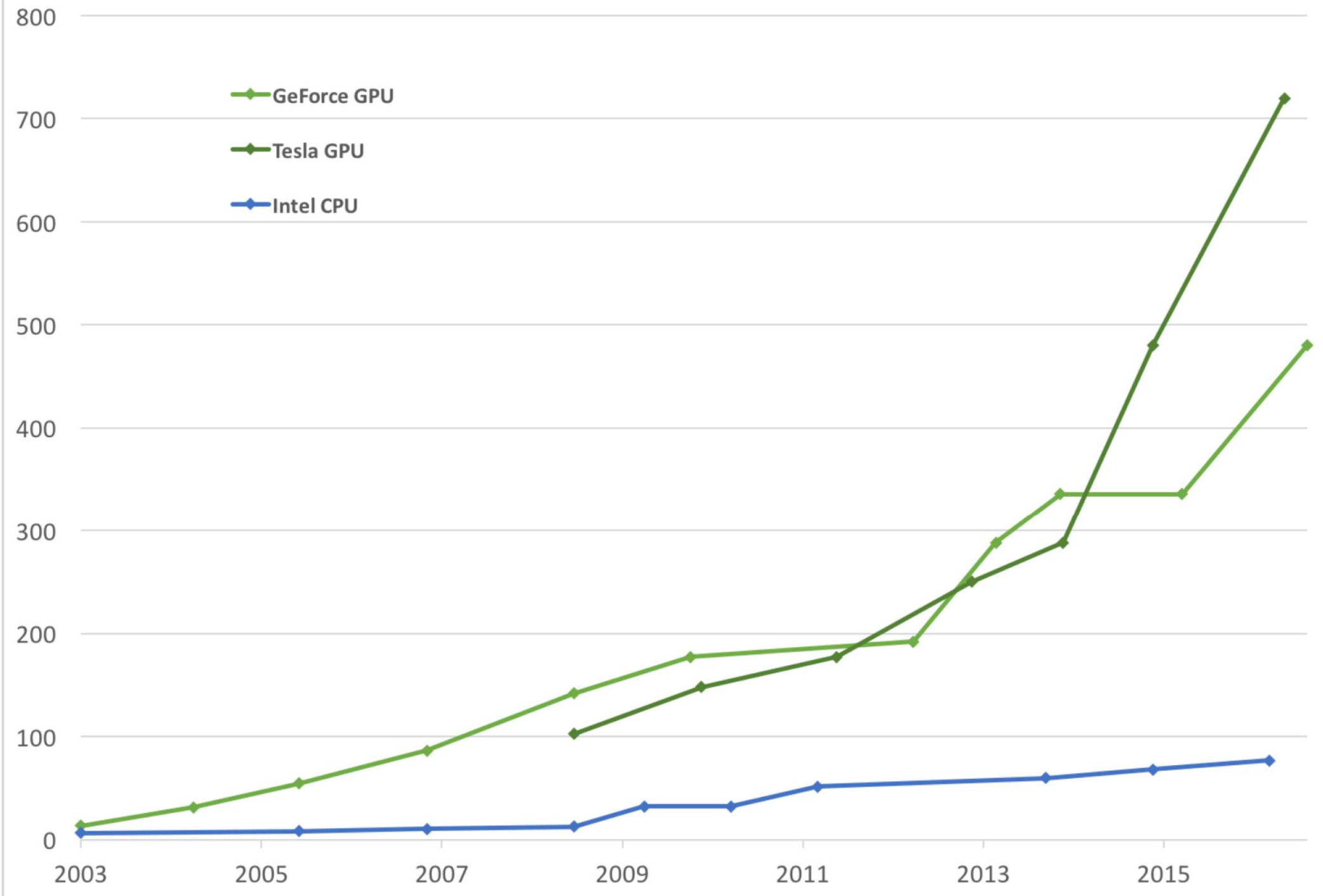


Theoretical GB/s

Peak Bandwidth

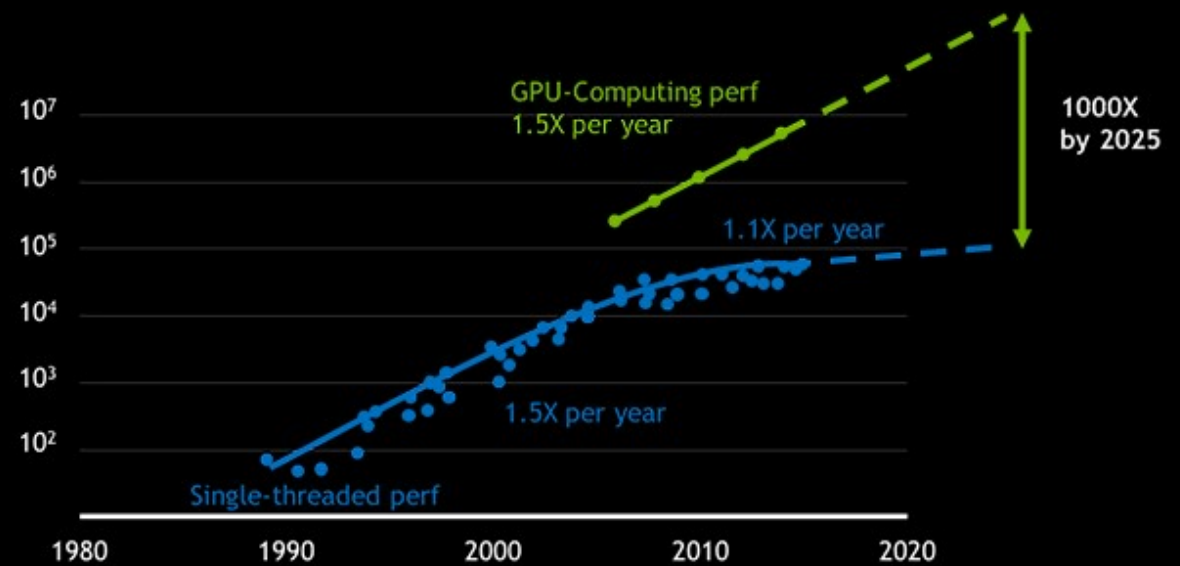
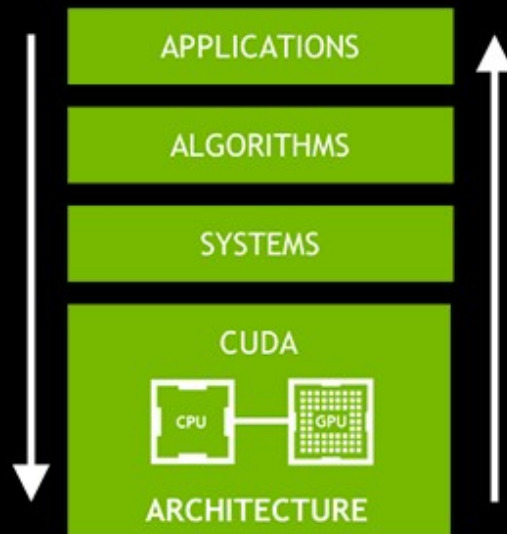


Theoretical Peak GB/s





RISE OF GPU COMPUTING



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. New plot and data collected for 2010-2015 by K. Rupp.

GPU Architectures Over the Years



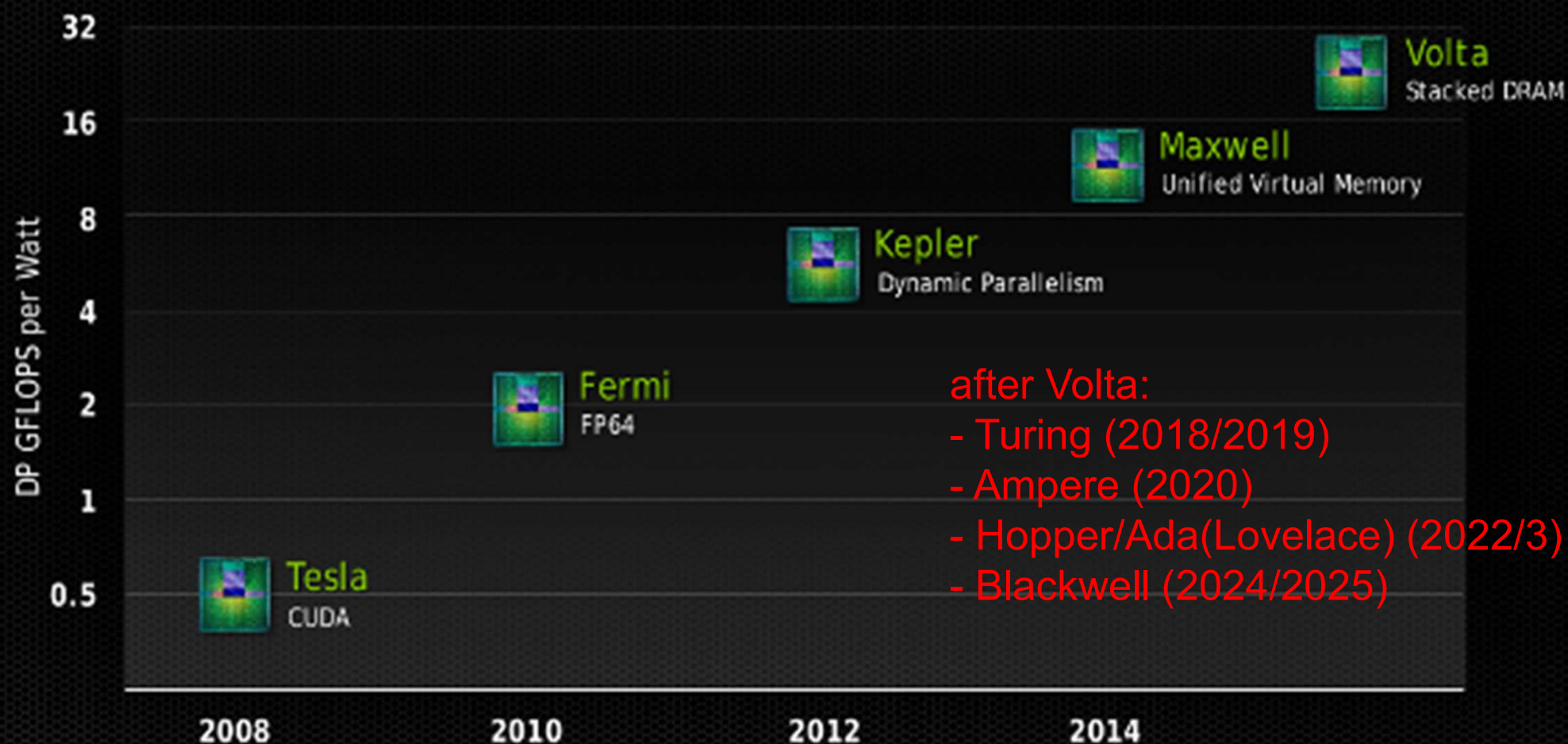
GPU Roadmap



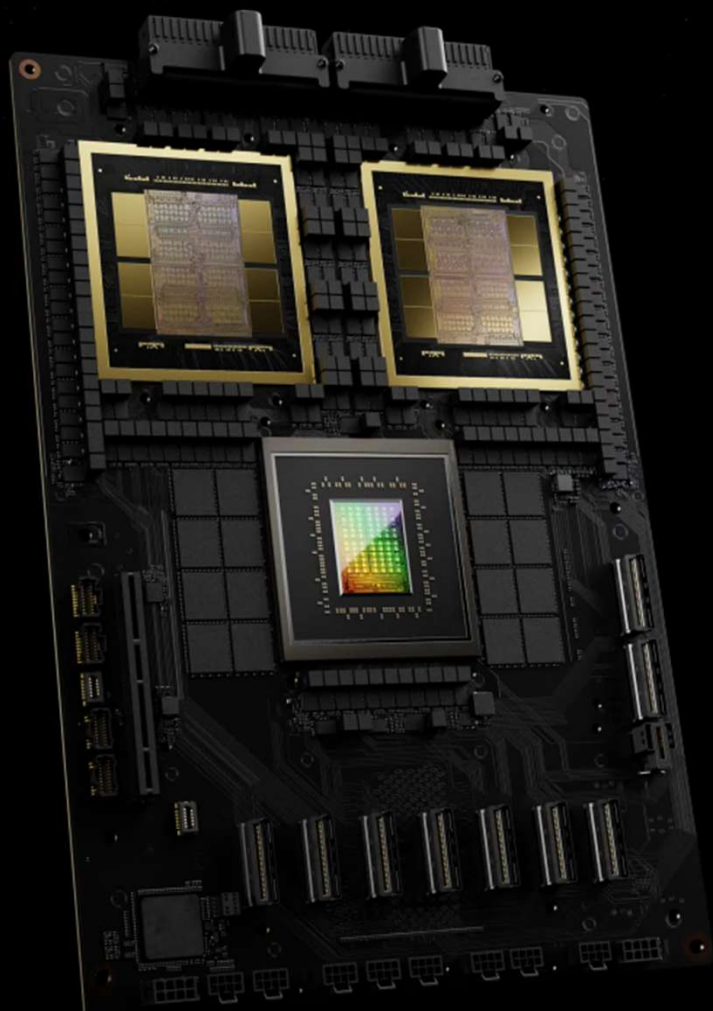
GPU Architectures Over the Years



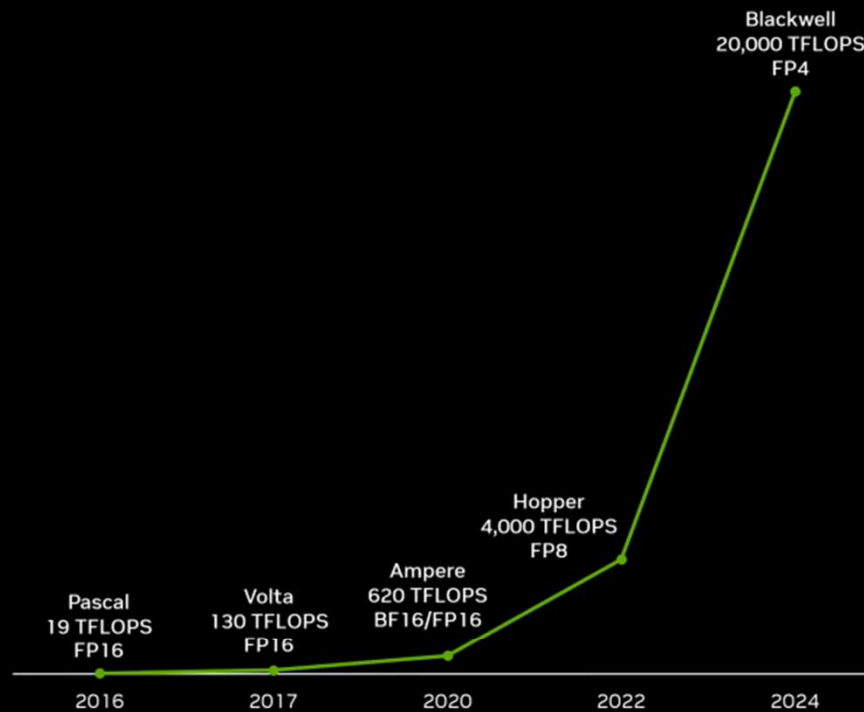
GPU Roadmap



GPU Architectures Over the Years



1000X AI Compute in 8 Years

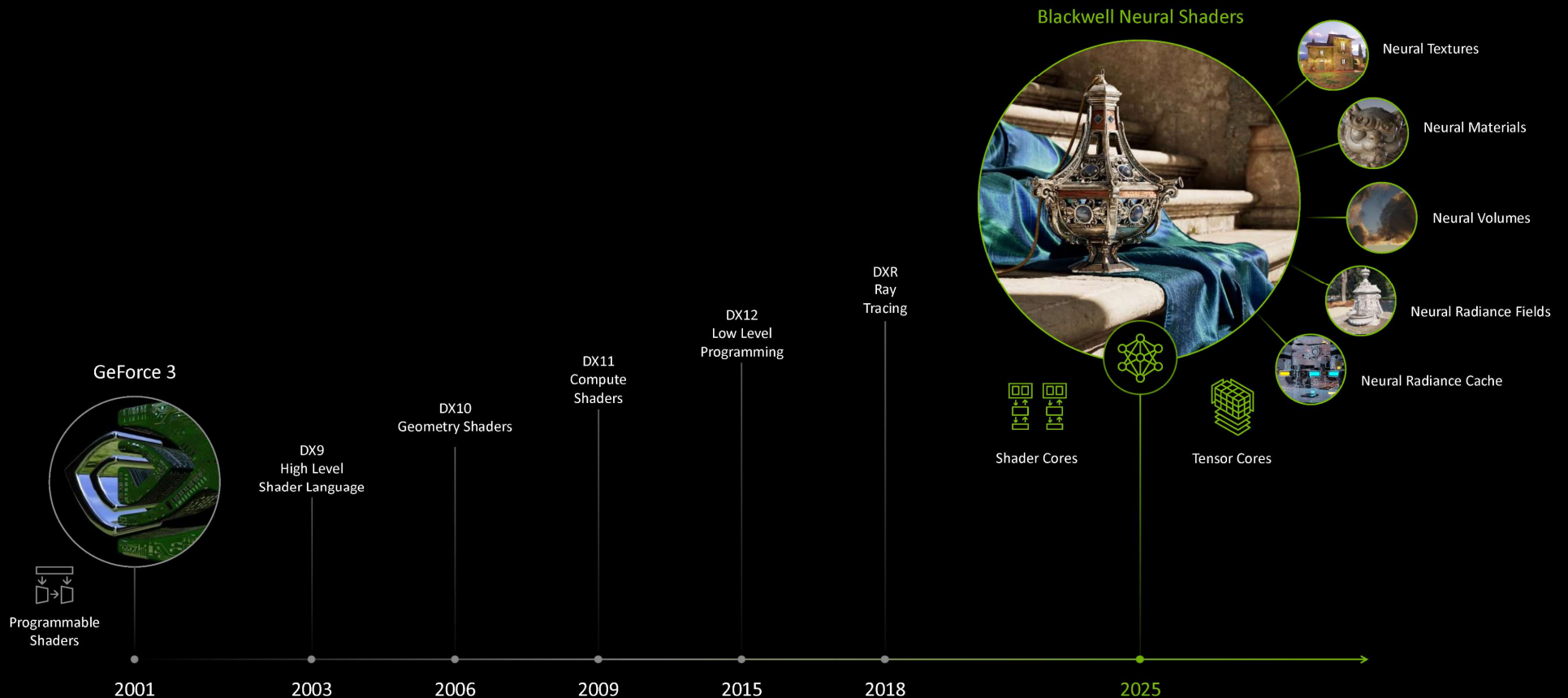


Neural Rendering / Neural Shading



Blackwell Brings AI to Shaders

Unlocking the next two decades of graphics innovation



Neural Rendering / Neural Shading



30 Years of Increasing Geometry

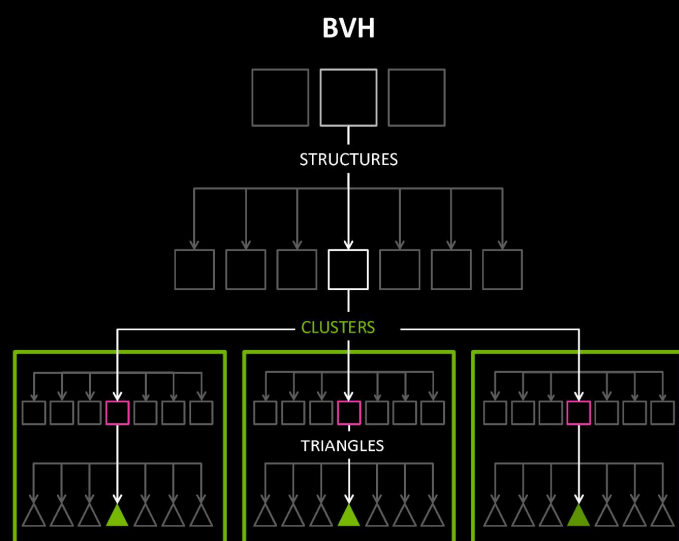


Neural Rendering / Neural Shading



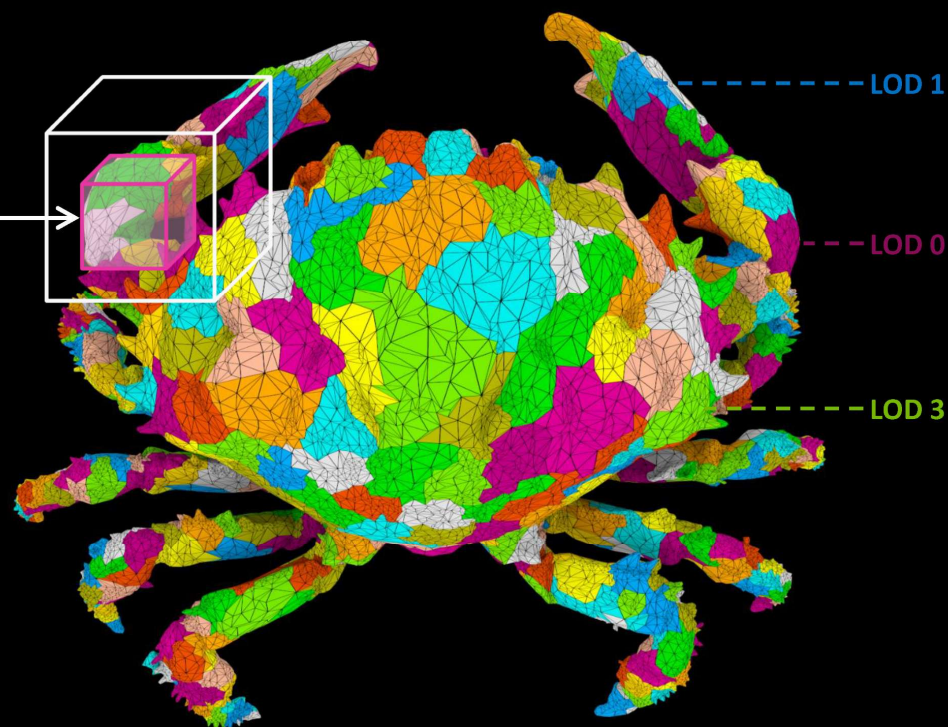
RTX Mega Geometry

Accelerate BVH updates for cluster-based systems like Nanite



Compress and Cache Clusters Over Many Frames

Trace Rays



Mesh Composed of Clusters at Different LOD Levels

Neural Rendering / Neural Shading



NVIDIA RTX Kit

Delivering neural rendering to developers



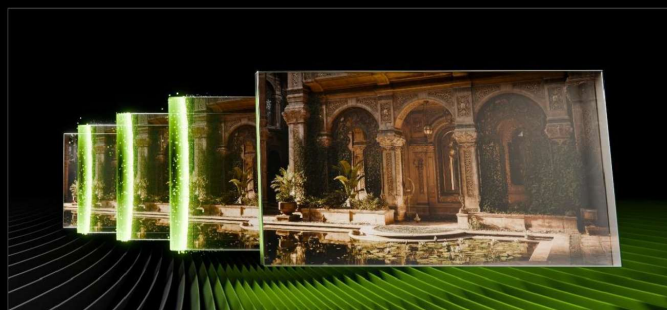
RTX Neural Shaders



RTX Mega Geometry



RTX Hair & Skin



DLSS 4



Reflex 2



RTX Remix

NVIDIA Architectures (since first CUDA GPU)



Tesla [CC 1.x]: 2007-2009

- G80, G9x: 2007 (Geforce 8800, ...)
GT200: 2008/2009 (GTX 280, ...)

Fermi [CC 2.x]: 2010 (2011, 2012, 2013, ...)

- GF100, ... (GTX 480, ...)
GF104, ... (GTX 460, ...)
GF110, ... (GTX 580, ...)

Kepler [CC 3.x]: 2012 (2013, 2014, 2016, ...)

- GK104, ... (GTX 680, ...)
GK110, ... (GTX 780, GTX Titan, ...)

Maxwell [CC 5.x]: 2015

- GM107, ... (GTX 750Ti, ...); [Nintendo Switch]
GM204, ... (GTX 980, Titan X, ...)

Pascal [CC 6.x]: 2016 (2017, 2018, 2021, 2022, ...)

- GP100 (Tesla P100, ...)
- GP10x: x=2,4,6,7,8, ...
(GTX 1060, 1070, 1080, Titan X *Pascal*,
Titan Xp, ...)

Volta [CC 7.0, 7.2]: 2017/2018

- GV100, ...
(Tesla V100, Titan V, Quadro GV100, ...)

Turing [CC 7.5]: 2018/2019

- TU102, TU104, TU106, TU116, TU117, ...
(Titan RTX, RTX 2070, 2080 (Ti), GTX 1650, 1660, ...)

Ampere [CC 8.0, 8.6, 8.7, 8.8]: 2020

- GA100, GA102, GA104, GA106, ...; [Nintendo Switch 2]
(A100, RTX 3070, 3080, 3090 (Ti), RTX A6000, ...)

Hopper [CC 9.0], Ada Lovelace [CC 8.9]: 2022/23

- GH100, AD102/103/104/106/107, ...
(H100, H200, GH200, L20, L40, L40S, L2, L4,
RTX 4080 (12/16 GB), RTX 4090, RTX 6000 (Ada), ...)

Blackwell [CC 10.0, 10.1(→11.0), 10.3, 12.0, 12.1]: 2024/2025

- GB100, GB200, GB202/203/205/206/207, G10, ...
(RTX 5080/5090, HGX B200/B300, GB200/GB300 NVL72,
RTX 4000/5000/6000 PRO Blackwell, B40, ...)

Rubin: 2026 ? Feynman: 2028 ?

Recent Updates (1): Hopper



NVIDIA Hopper architecture (2022)

[`https://en.wikipedia.org/wiki/Hopper\_\(microarchitecture\)`](https://en.wikipedia.org/wiki/Hopper_(microarchitecture))

Hopper Architecture Whitepaper:

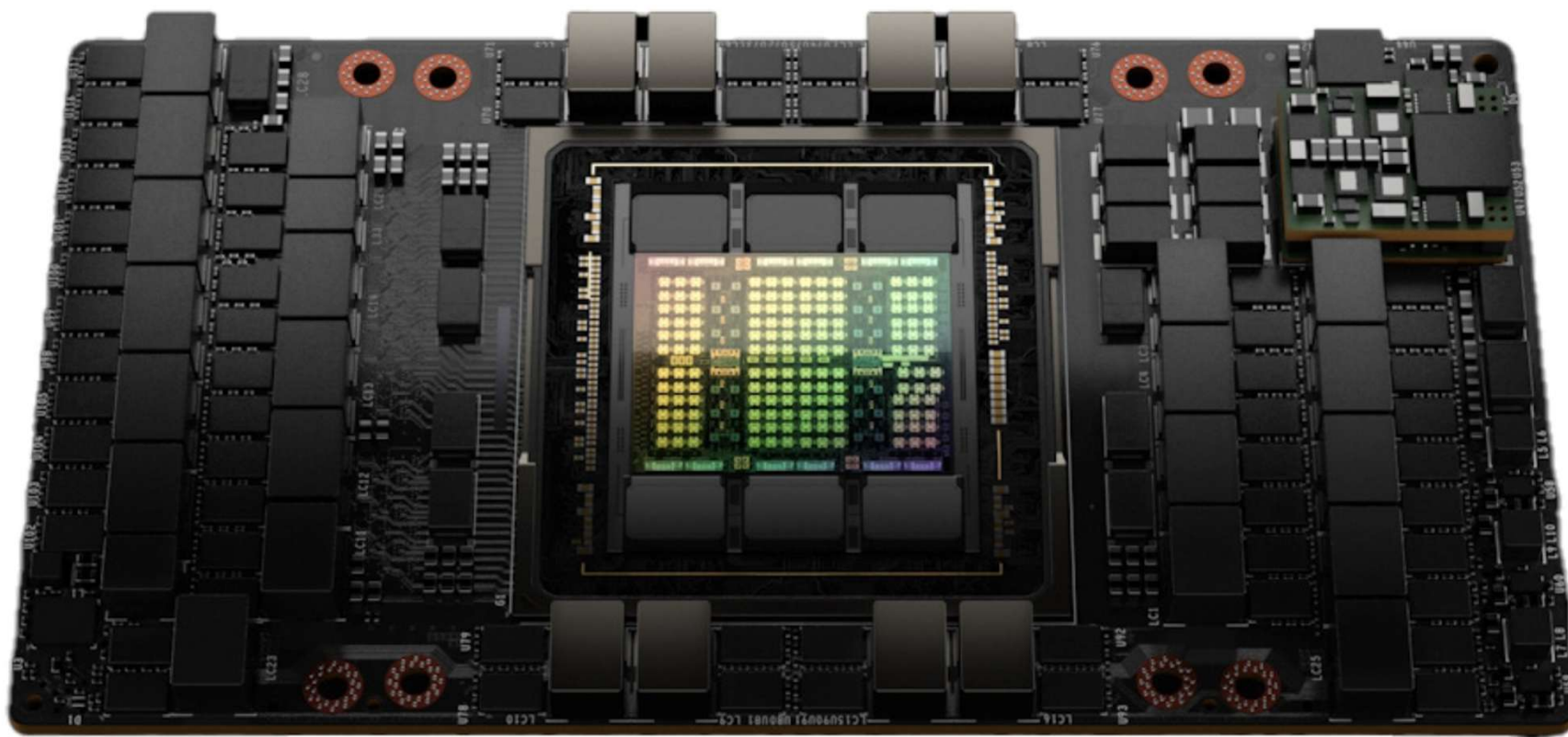
[`https://resources.nvidia.com/en-us-tensor-core/gtc22-whitepaper-hopper/`](https://resources.nvidia.com/en-us-tensor-core/gtc22-whitepaper-hopper/)

H100 (Hopper):

[`https://www.nvidia.com/en-us/data-center/h100/`](https://www.nvidia.com/en-us/data-center/h100/)

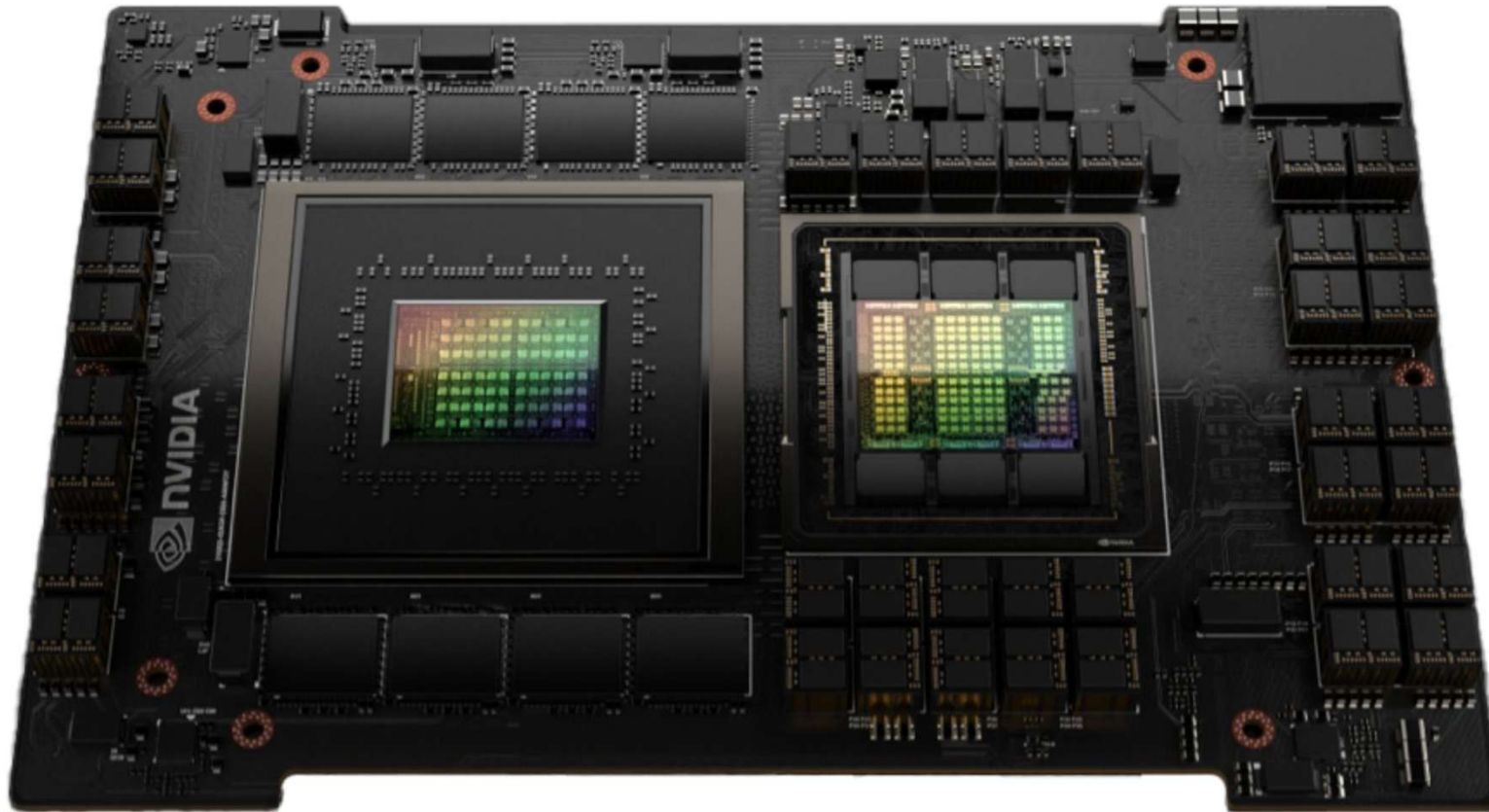
*H100 has up to 18,432 FP32 CUDA cores (max arch)
(H100 SXM5: 16,896; H100 PCIe: 14,592)*

Recent Updates (1): Hopper



NVIDIA H100 SXM5

Recent Updates (1): Hopper



NVIDIA Grace Hopper Superchip (Grace CPU + Hopper GPU)

Full GPU: 144 SMs (in 8 GPCs/72 TPCs)



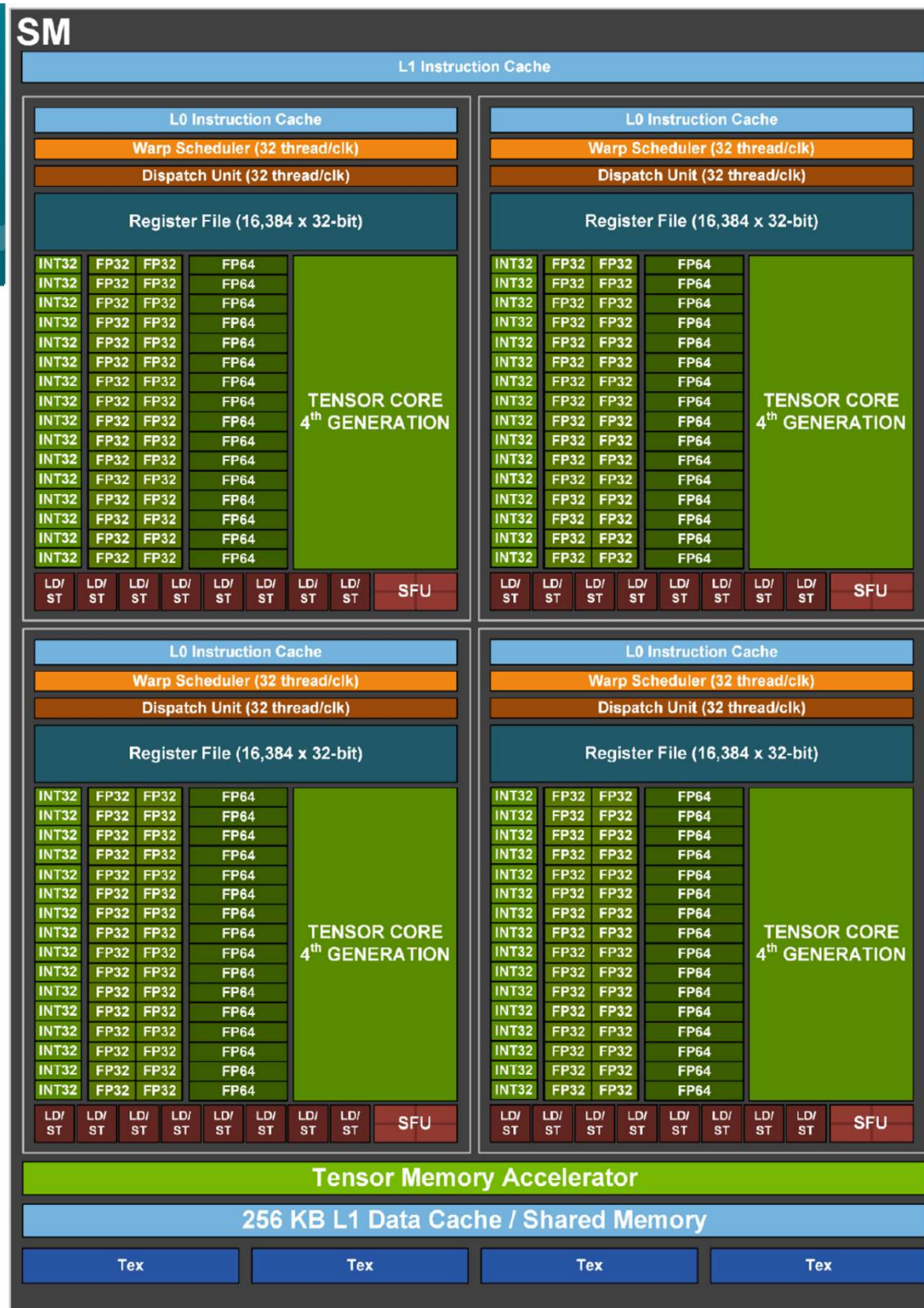
NVIDIA GH100 SM

CC 9.0 Multiprocessor

- 128 FP32 + 64 INT32 cores
- 64 FP64 cores
- 4x 4th gen tensor cores
- ++ thread block clusters, DPX insts., FP8, TMA

4 partitions inside SM

- 32 FP32 + 16 INT32 cores
- 16 FP64 cores
- 8x LD/ST units each
- 1x 4th gen tensor core each
- Each has: warp scheduler, dispatch unit, 16K register file



NVIDIA Hopper GH100 Architecture (2022)



GH 100 (H100)

Full GPU: 144 SMs (in 8 GPCs/72 TPCs)

- 64K 32-bit registers / SM = 256 KB register storage per SM
- 256 KB shared memory / L1 per SM

For 144 SMs on full GPU [SXM5: 132; PCIe: 114]

- 36 MB register storage, 36 MB shared mem / L1 storage = **72 MB context+”shared context” storage !**
- L2 cache size on H100: 50 MB
- 18,432 FP32 cores (128 FP32 cores per SM)
- 294,912 max threads in flight (max warps / SM = 64)

Recent Updates (2): Ada



NVIDIA Ada (Lovelace) (2022/2023)

[`https://en.wikipedia.org/wiki/Ada\_Lovelace\_\(microarchitecture\)`](https://en.wikipedia.org/wiki/Ada_Lovelace_(microarchitecture))

Ada Architecture Whitepaper:

[`https://images.nvidia.com/aem-dam/Solutions/geforce/ada/nvidia-ada-gpu-architecture.pdf`](https://images.nvidia.com/aem-dam/Solutions/geforce/ada/nvidia-ada-gpu-architecture.pdf)

Geforce 40-series:

[`https://en.wikipedia.org/wiki/GeForce\_40\_series/`](https://en.wikipedia.org/wiki/GeForce_40_series/)

RTX 4090 has 16,384 FP32 CUDA cores

NVIDIA Ada Lovelace AD102 Architecture (2022)



AD 102 (RTX 4090, ...)

Full RTX 4090: 128 SMs (in 11 GPCs/64 TPCs)



NVIDIA AD102 SM

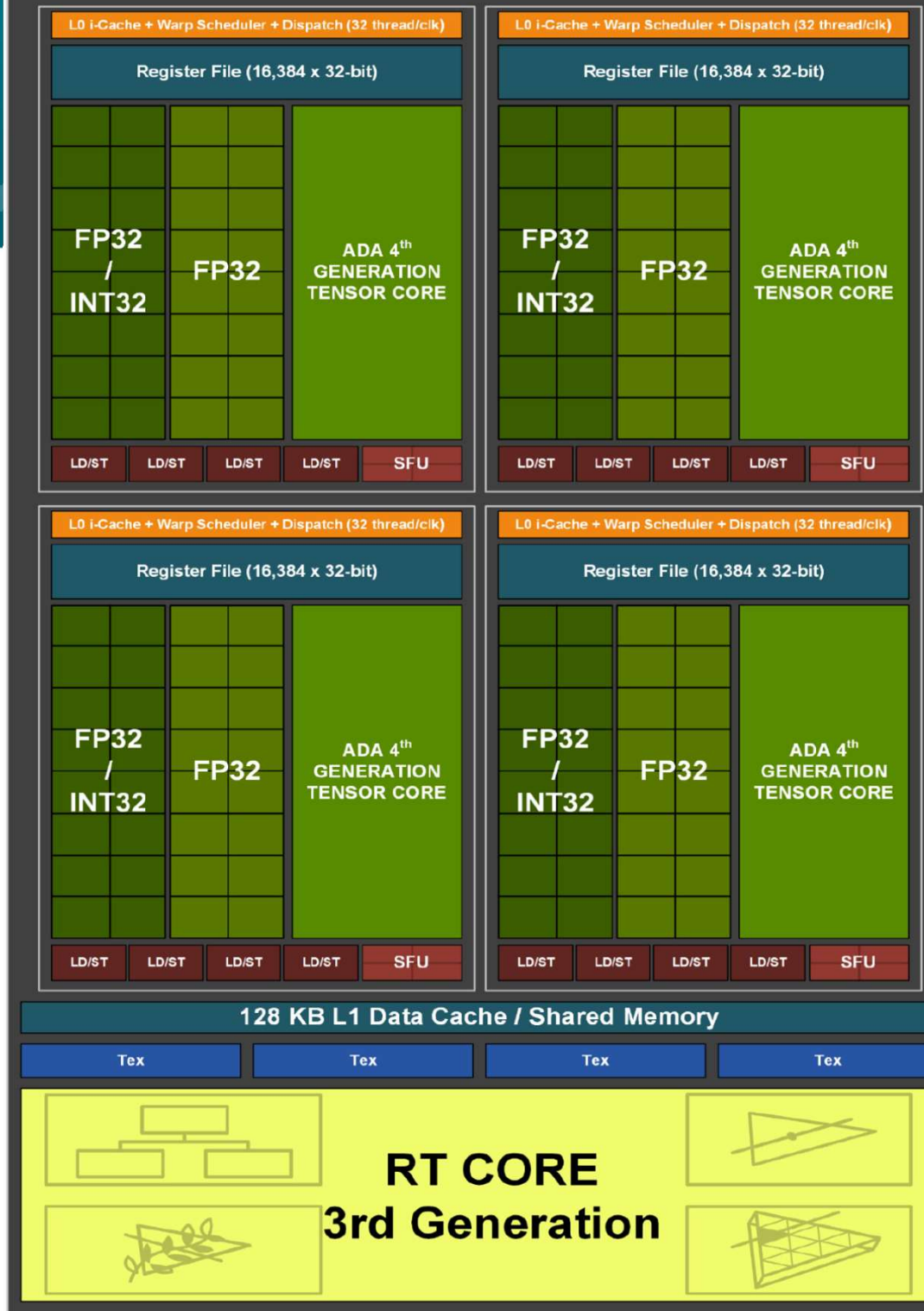
Multiprocessor: SM (CC 8.9)

- 128 (64+64) FP32 + 64 INT32 cores
- 2 (!) FP64 cores (not in diagram)
- 4x 4th gen tensor cores
- 1x 3rd gen RT (ray tracing) core
- ++ thread block clusters, FP8, ... (?)

4 partitions inside SM

- 32 (16+16) FP32 + 16 INT32 cores
- 4x LD/ST units; 4 SFUs each
- 1x 4th gen tensor core each
- Each has: warp scheduler, dispatch unit, 16K register file

SM



NVIDIA Ada Lovelace AD10x Architecture (2022)



AD 10x / AD 102 (RTX 4090)

Full GPU: 144 SMs (in 12 GPCs/72 TPCs)

- 64K 32-bit registers / SM = 256 KB register storage per SM
- 128 KB shared memory / L1 per SM

For 144 SMs on full GPU [*RTX 4090: 128; RTX 4080 16GB: 76; RTX 4080 12GB: 60*]

- 36 MB register storage, 18 MB shared mem / L1 storage =
54 MB context+”shared context” storage !
- L2 cache size on RTX 4090: 72 MB
- 18,432 FP32 cores (128 FP32 cores per SM) [*RTX 4090: 16,384*]
- 294,912 max threads in flight (max warps / SM = 64) [*RTX 4090: 262,144*]

Recent Updates (3): Blackwell



NVIDIA Blackwell architecture (2024/2025)

[https://en.wikipedia.org/wiki/Blackwell_\(microarchitecture\)](https://en.wikipedia.org/wiki/Blackwell_(microarchitecture))

Geforce 50-series:

https://en.wikipedia.org/wiki/GeForce_RTX_50_series/

RTX 5090 has 21,760 FP32 CUDA cores

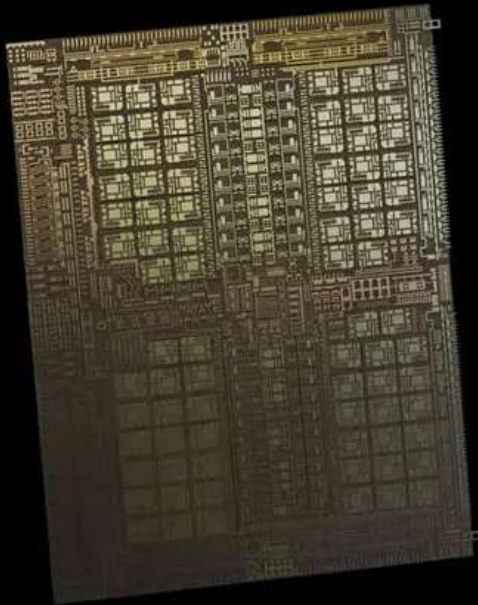
Blackwell Architecture Whitepaper:

<https://images.nvidia.com/aem-dam/Solutions/geforce/blackwell/nvidia-rtx-blackwell-gpu-architecture.pdf>

GB200 NVL72 / GB200 Grace Blackwell Superchip:

<https://www.nvidia.com/en-us/data-center/gb200-nv172/>

Recent Updates (3): Blackwell



BLACKWELL

THE LARGEST CHIP PHYSICALLY POSSIBLE

104 billion transistors

TSMC 4NP process

10TB/s NVIDIA High-Bandwidth Interface

Recent Updates (3): Blackwell



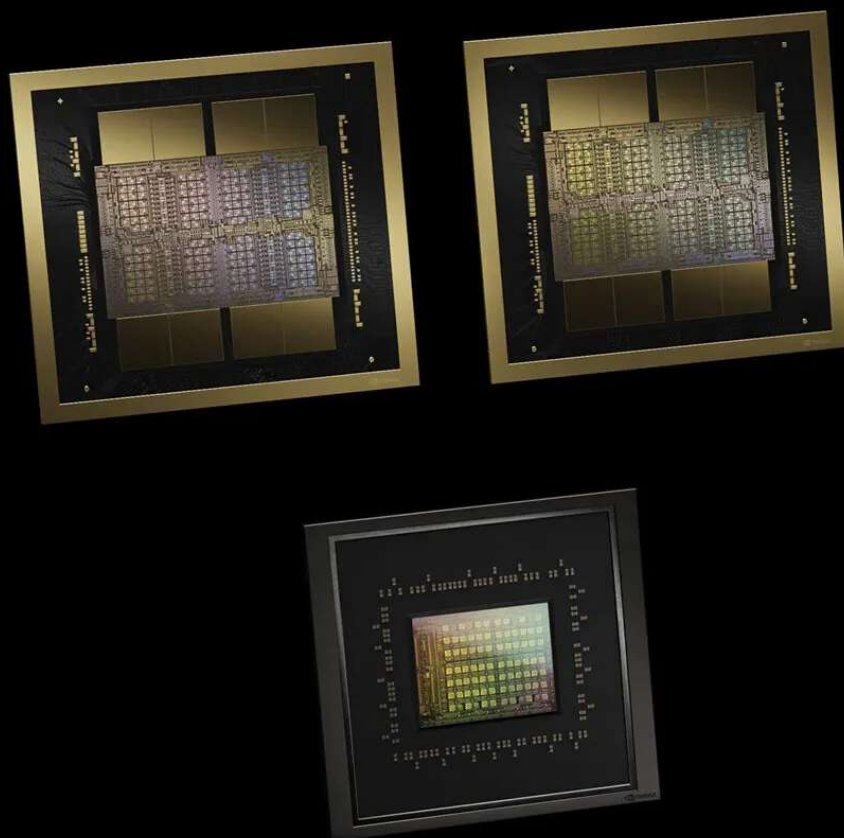
BLACKWELL GPU

THE TWO LARGEST DIES POSSIBLE—UNIFIED AS ONE GPU

208 billion transistors

Full-cache coherency

Recent Updates (3): Blackwell



TWO BLACKWELL GPUs AND ONE GRACE CPU

BUILDING BLOCKS OF THE GB200 SUPERCHIP

384GB of HBM3e

72 Arm Neoverse V2 CPU cores

900GB/s of NVLink-C2C bandwidth

NVIDIA Blackwell SM

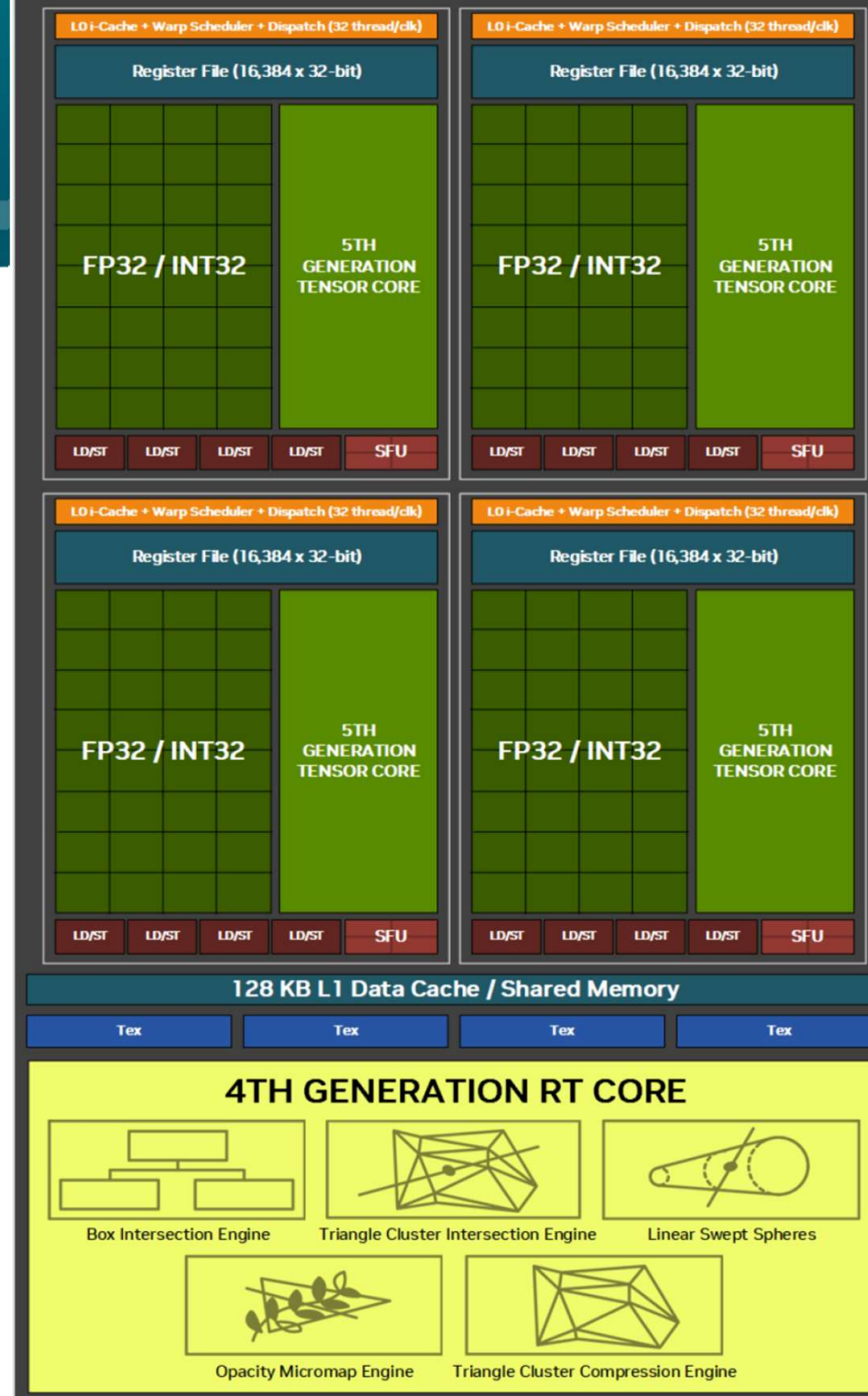
CC 12.0 SM (GB 202 Multiprocessor)

- 128 FP32/INT32 cores
- 2 FP64 cores
- 4x 5th gen tensor cores
- ++ thread block clusters, DPX insts., FP8, NVFP4, TMA

4 partitions inside SM

- 32 FP32/INT32 cores
- 4x LD/ST units each
- 1x 5th gen tensor core
- Each has: warp scheduler, dispatch unit, 16K register file

SM

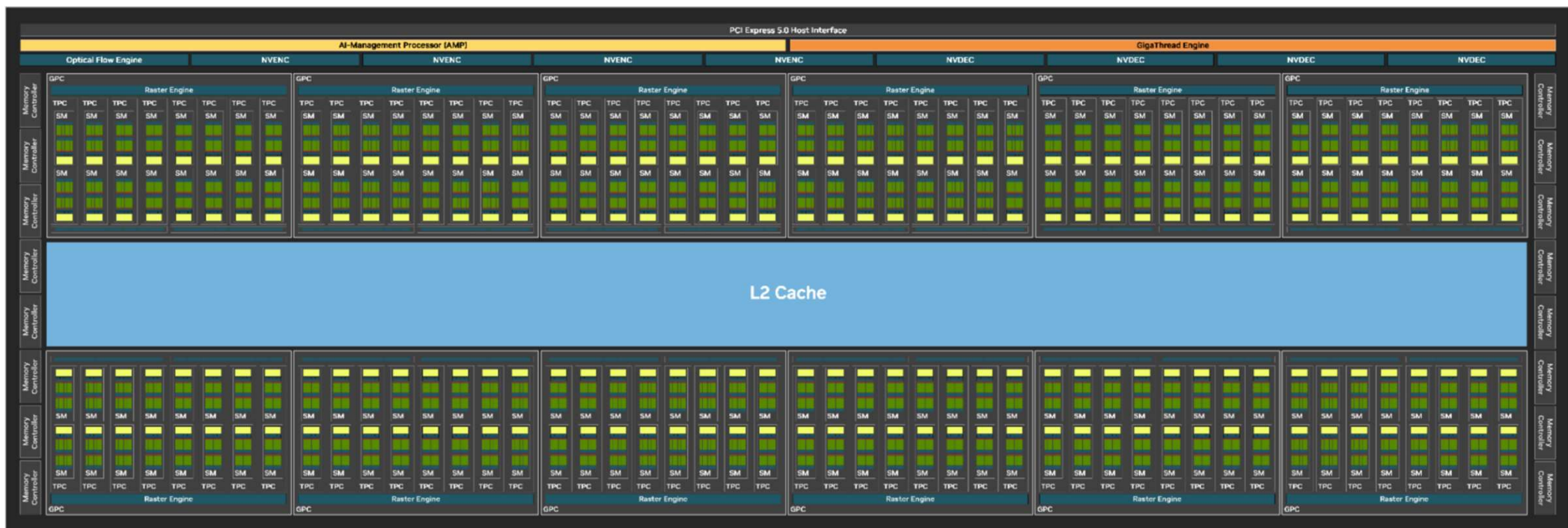


NVIDIA Blackwell GB202 Architecture (2025)



GB 202 (RTX GPU)

Full GPU: 192 SMs in 12 GPCs, 96 TPCs,
(24,576 FP32 cores)



NVIDIA Blackwell GB202 Architecture (2025)



GB 202 (RTX GPU)

Full GPU: 192 SMs in 12 GPCs, 96 TPCs
(RTX 5090: 170 SMs, 11 GPCs, 85 TPCs)

- 64K 32-bit registers / SM = 256 KB register storage per SM
- 128 KB shared memory / L1 per SM

For 192 SMs on full GPU (RTX PRO 6000: 12 GPCs, 94 TPCs, 188 SMs = 24,064 FP32 cores)

- 48 MB register storage, 24 MB shared mem / L1 storage =
72 MB context+”shared context” storage !
- L2 cache size 128 MB (e.g., RTX PRO 6000) (RTX 5090: 96 MB, RTX 5080: 64 MB)
- 24,576 FP32 cores (128 FP32 cores per SM), 768 tensor cores
(RTX PRO 6000: 752 tensor cores; RTX 5090: 170 SMs = 21,760 FP32 cores, 680 tensor cores)
- 294,912 max threads in flight (max warps / SM = 48)

NVIDIA Blackwell SM

CC 10.3 SM (GB300 Blackwell Ultra)

- 128 FP32/INT32 cores
- 64(?) FP64 cores
- 4x 5th gen tensor cores
- Tensor Memory Accelerator (TMA)
- ++ thread block clusters, DPX insts., FP8, NVFP4, 256 KB Tensor Memory (TMEM), needs 4 warps = warp group for full TMEM access (1 warp/partition)

4 partitions inside SM

- 32 FP32/INT32 cores
- 8x LD/ST units each
- 1x 5th gen tensor core
- 64 KB Tensor Memory (TMEM)
- Each has: warp scheduler, dispatch unit, 16K register file

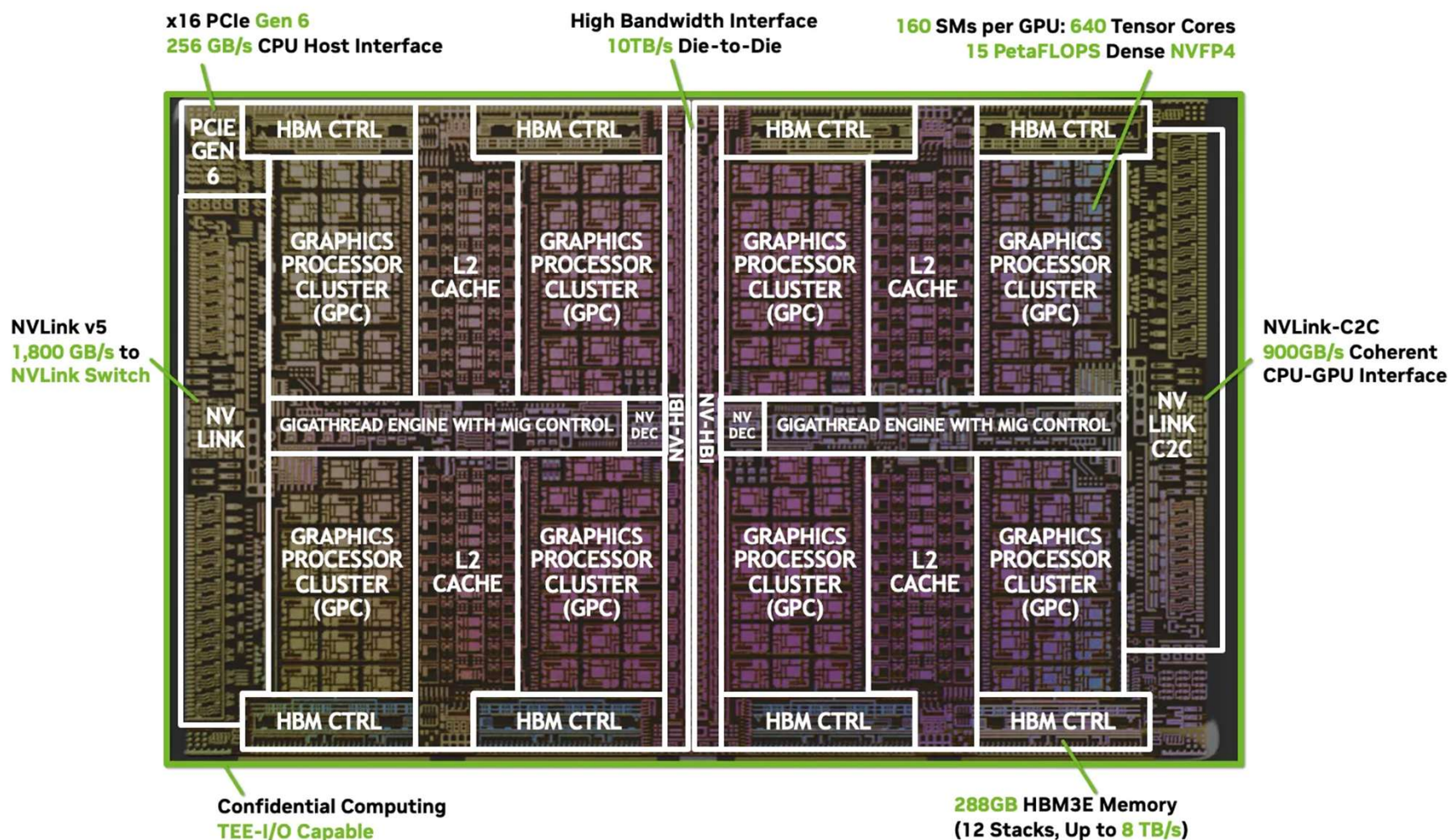
Streaming Multiprocessor (SM)



NVIDIA Blackwell Ultra (GB300) GPU



NVIDIA Blackwell Ultra GPU



Thank you.