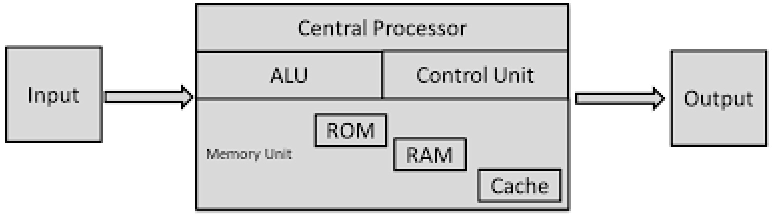
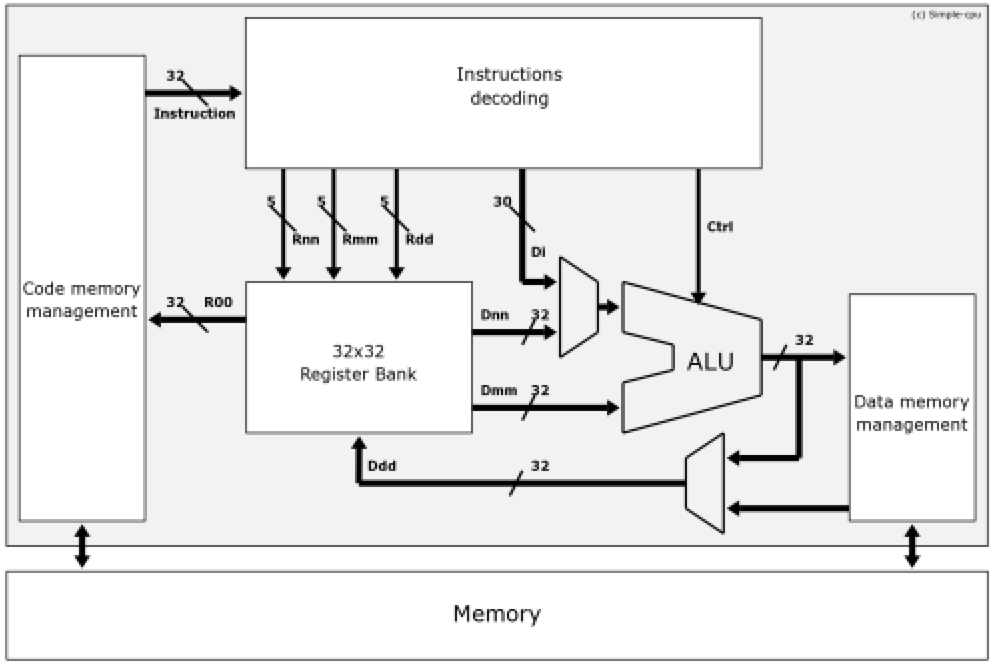
MSRA SH Triton Study Group
0. GPU Basic Knowledge

2025/06/27

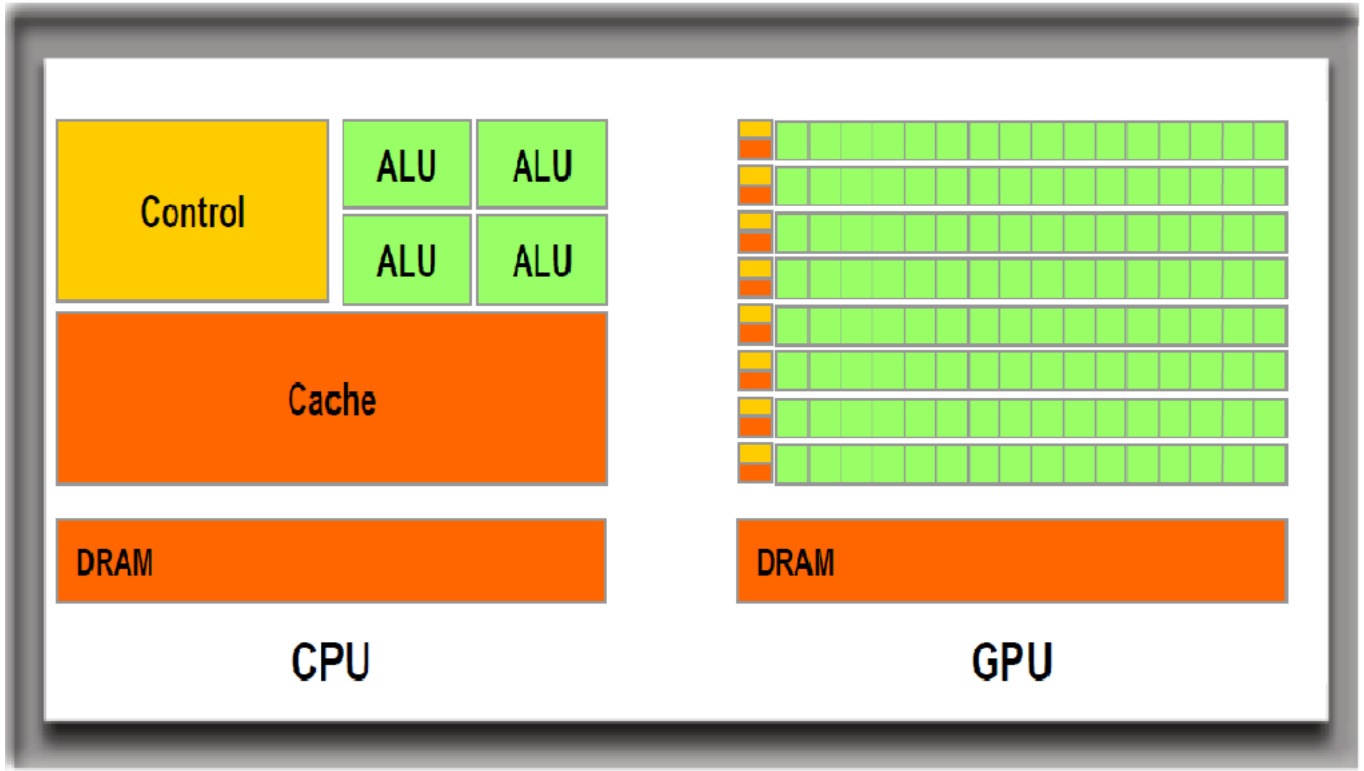
CPU Architecture



Central Processing Unit

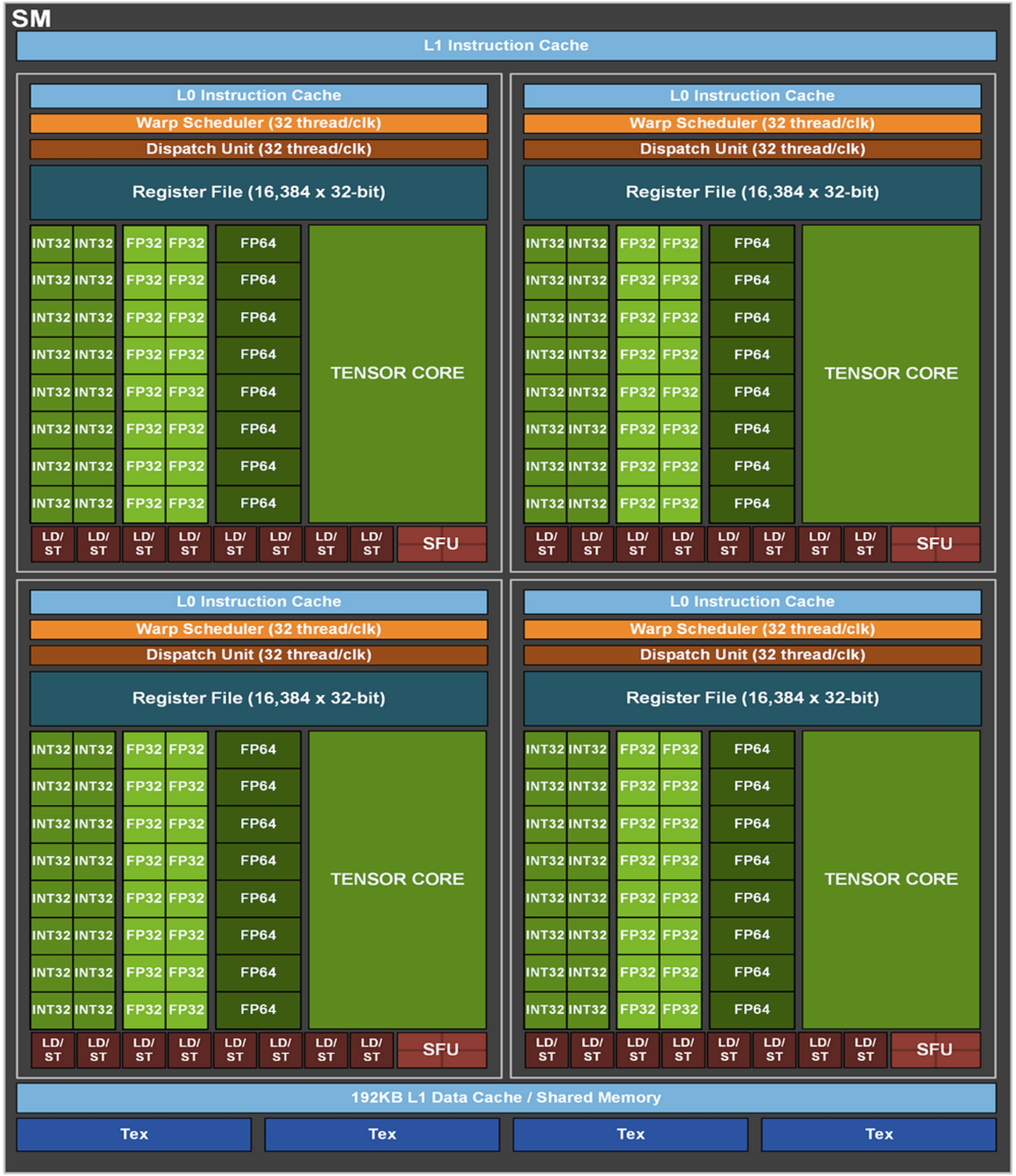


CPU vs. GPU



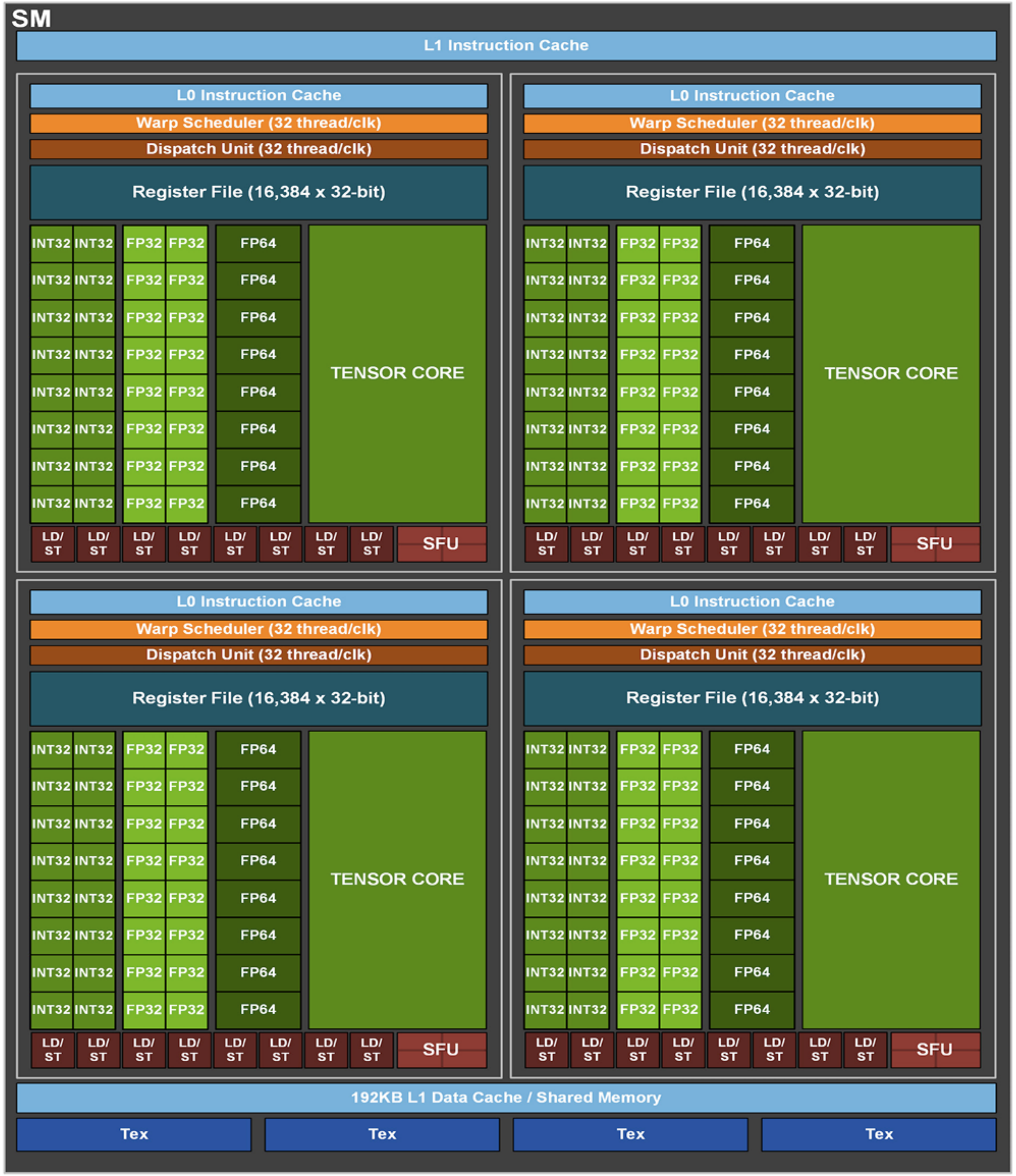
Modern GPU Example:

- Warp Level:
  - Computation:
    - 32 clock-synchronized threads (lanes)
    - 32/64-bit CUDA cores
    - 1 big 8/16-bit tensor core
    - SFU: special function units
  - Communication:
    - LD/ST: 8 load/store units
    - 4-clock shuffle-sync on registers
  - Memory:
    - L0 instruction cache
    - 255 32-bit registers / thread


SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST
SFU
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST
SFU
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST
SFU
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST
SFU
192KB L1 Data Cache / Shared Memory
Tex
Tex
Tex
Tex

Modern GPU Example:

- SM (Streaming Multiprocessor):
  - 4 physical warps
  - L1 Cache
  - Shared Memory
  - Logical: thread-block



SM

L1 Instruction Cache

L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST SFU

L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST SFU

L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST SFU

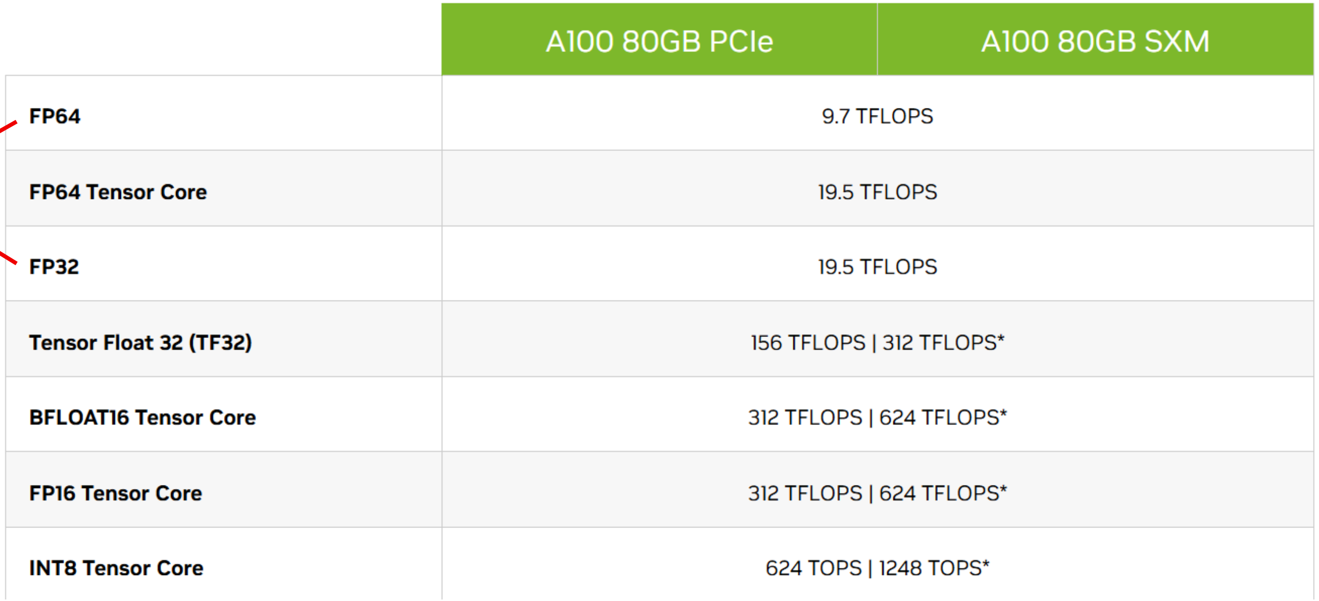
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
INT32 INT32 FP32 FP32 FP64
TENSOR CORE
LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST LD/ST SFU

192KB L1 Data Cache / Shared Memory

Tex Tex Tex Tex

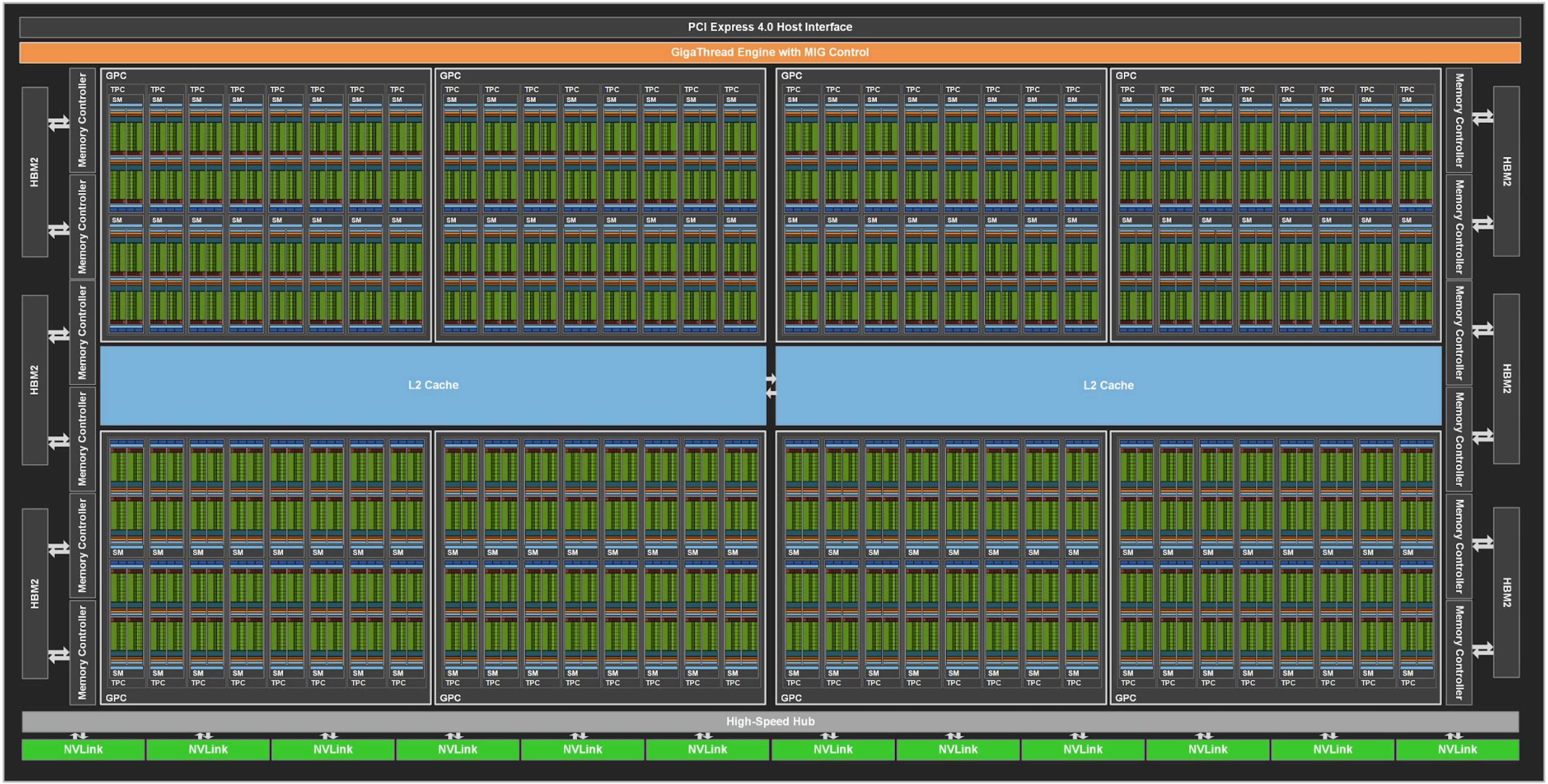
CUDA Core, Tensor Core and Matrix Multiplication

- $C = AB, A \in \mathbf{R}^{[M,K]}, B \in \mathbf{R}^{[K,N]}, C \in \mathbf{R}^{[M,N]}$
- Parallelized: $C_{ij} = \sum_k A_{ik} B_{kj}$
- FLOPS: float-point operations per second, $2MNK/Latency$
  - 2 means multiplication and addition



	A100 80GB PCIe	A100 80GB SXM
FP64	9.7 TFLOPS	
FP64 Tensor Core	19.5 TFLOPS	
FP32	19.5 TFLOPS	
Tensor Float 32 (TF32)	156 TFLOPS | 312 TFLOPS*	
BFLOAT16 Tensor Core	312 TFLOPS | 624 TFLOPS*	
FP16 Tensor Core	312 TFLOPS | 624 TFLOPS*	
INT8 Tensor Core	624 TOPS | 1248 TOPS*	

CUDA Core



PCI Express 4.0 Host Interface
GigaThread Engine with MIG Control
GPC
TPC
SM
L2 Cache
L2 Cache
Memory Controller
HBM2
High-Speed Hub
NVLink

Modern GPU Example: A100

- TPC (texture processing cluster): Group of SMs
- GPC: Group of TPCs

- L2 Cache
- HBM: Global Memory

How Logical Kernel Code Maps to Physical Hardware

• Kernel	• GPU
• Thread Block	• Streaming Multiprocessor
• (Warp)	• Warp
• Thread	• Lane

Modern GPU Example: A100

- Computation:
  - 1 thread (lane) = 1 independent slow CPU core
  - 1 warp = 32 threads + 1 tensor core
  - 1 thread block (SM) = several warps
  - 1 kernel = several thread blocks on different SMs

- Memory:
  - Registers
  - Shared Memory (programmable in thread-block level) & L1 Cache
  - L2 Cache
  - Global Memory

Parallelization & Overlap

- Single instruction, multiple data
    - Warp-level parallelism requires data independence
    - Warp-level data sharing: shuffle-sync
    - A warp can cooperate with other warps by SM-level data sharing
    - Information exchange between thread-blocks can only occur through global memory

- Computation is fast, memory access is slow
    - Hierarchical memory
    - Buffer, double buffer and multi-stage

Reading Materials

- https://www.nvidia.com/en-us/data-center/a100/
- nvidia-ampere-architecture-whitepaper.pdf
- https://hao-ai-lab.github.io/cse234-w25/index.html
- https://people.maths.ox.ac.uk/~gilesm/cuda/