
Many-core Architectures

Sathish Vadhiyar

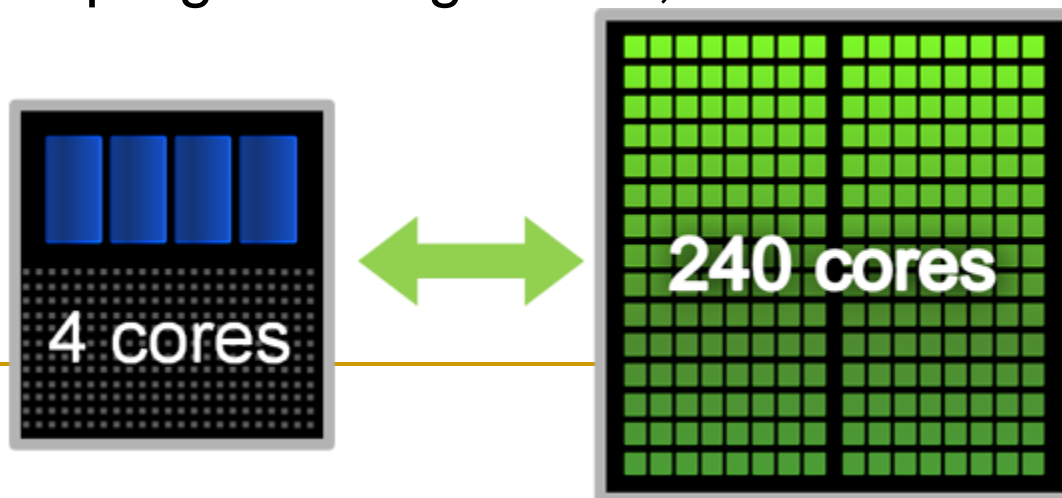
Introduction to Scalable Systems

Introduction

- Graphical Processing Unit
 - A single CPU can consist of a max of 128 cores
 - GPUs consist of a large number of light-weight cores
 - Primarily proposed for graphics applications
 - Consists of stages with each stage consisting of independent computations
-

GPU and CPU, CUDA

- Typically GPU and CPU coexist in a heterogeneous setting
- “Less” computationally intensive part runs on CPU (coarse-grained parallelism), and more intensive parts run on GPU (fine-grained parallelism)
- NVIDIA’s GPU architecture is called CUDA (Compute Unified Device Architecture) architecture, accompanied by CUDA programming model, and CUDA C language

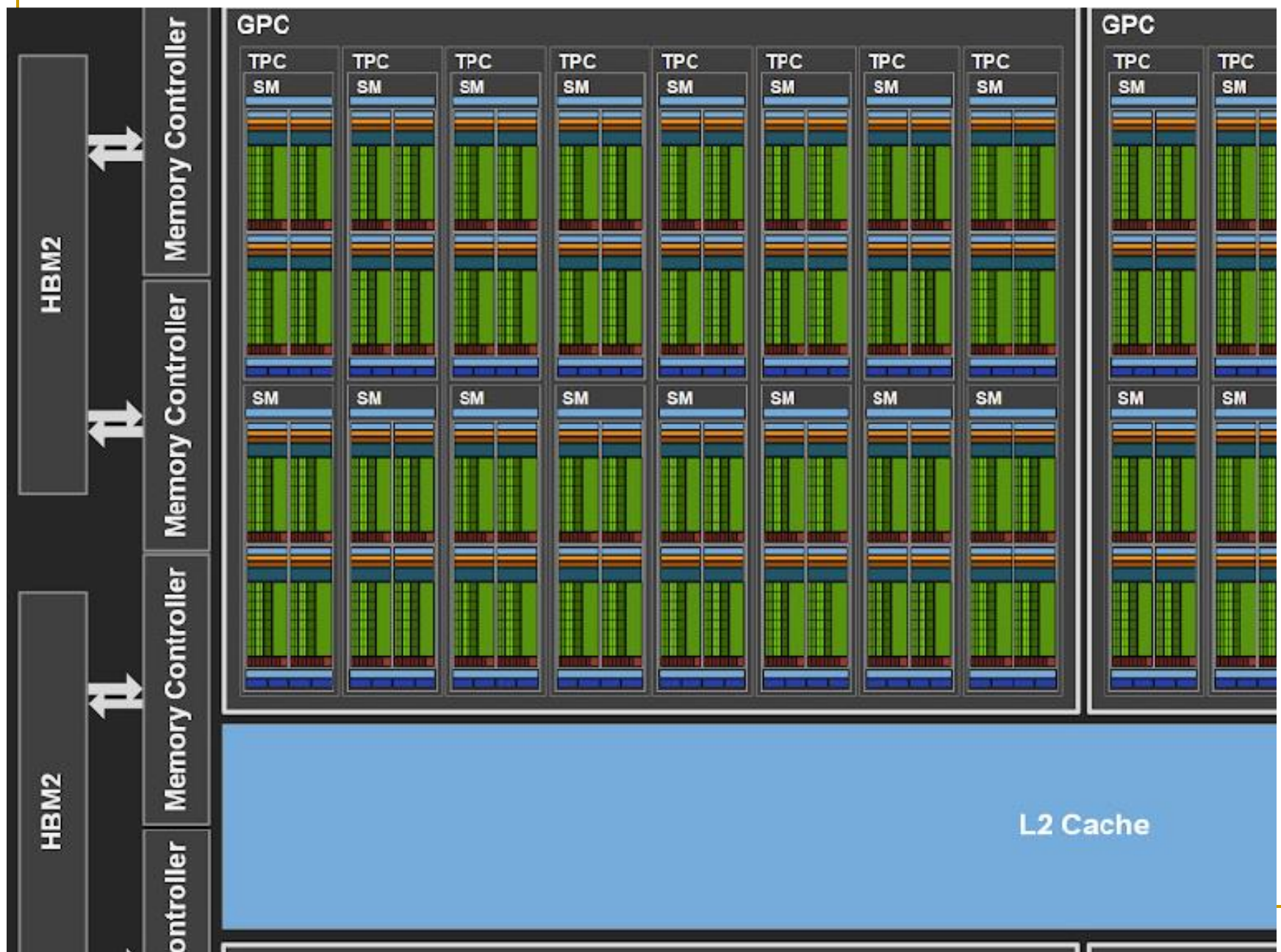


NVIDIA A100

- Graphic Processing Clusters (GPCs) – 7 Nos.
- Each GPC has:
 - 7 or 8 Nos. Texture Processing Clusters (TPCs).
Hence a total of 64 TPCs
 - Up to 16 Nos. Streaming Multiprocessors (SMs).
A total of 108 SMs



Figure 6. GA100 Full GPU with 128 SMs (A100 Tensor Core GPU has 108 SMs)



NVIDIA A100 SM

- Each SM has:
 - ❑ 64 Nos. FP32 GPU cores. A total of 6912 GPU cores.
 - ❑ 4 tensor cores. Hence a total of 432 tensor cores. Together they deliver 1024 FP16/FP32 operations per clock
 - ❑ 192 KB of combined shared memory and L1 data cache

SM



SM



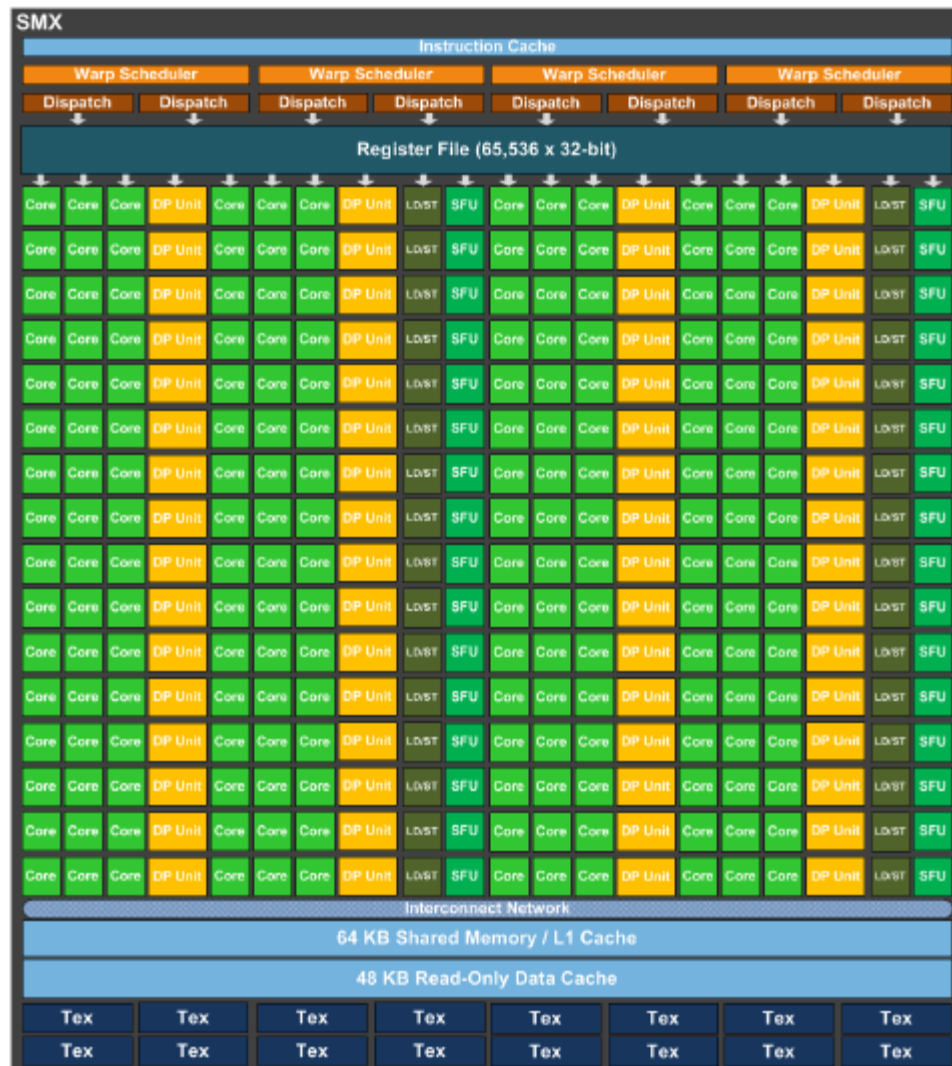
Tensor cores

- Specialized HPC compute cores for matrix operations for AI and HPC applications
- Perform matrix multiply and accumulate (MMA) operations

Streaming Multiprocessor (SMX)

Architecture in K40

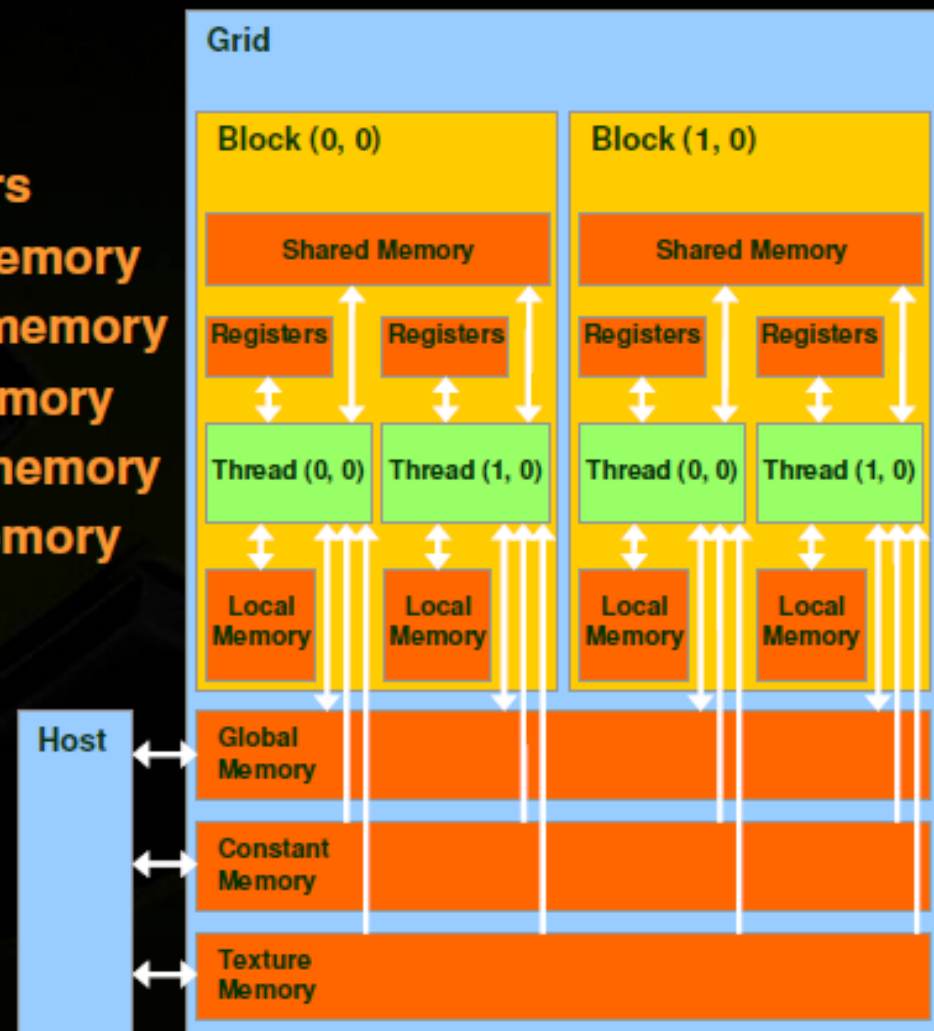
- 65536 registers
- 192 SP cores
- 64 DP cores
- 32 Special functional units (SFUs)
- 32 load/store units
- 16 texture filtering units
- Shared memory size = 16/32/48 K



CUDA Memory Spaces

- Each thread can:
 - Read/write per-thread **registers**
 - Read/write per-thread **local memory**
 - Read/write per-block **shared memory**
 - Read/write per-grid **global memory**
 - Read only per-grid **constant memory**
 - Read only per-grid **texture memory**

- The host can read/write **global, constant, and texture memory (stored in DRAM)**



Memory Hierarchy

■ Global or Device memory:

- ❑ Can be accessed by all the threads executing in all the SMXs
- ❑ Can be accessed by the CPU host
- ❑ Kepler K40: 12 GB.
- ❑ In A100s: 80 GB

■ Shared memory:

- ❑ In each SMX
- ❑ Shared by all the threads of a thread block executing in a SMX
- ❑ Kepler K40: 64 KB can be configured as 16/32/48KB for shared memory, with the rest for L1 cache
- ❑ In A100s: 192 KB

Memory Hierarchy

■ Registers

- In each SMX
- Used for storing the local data of the threads
- Kepler k40: 64K registers in each SMX
- A100: 64K registers in each SMX

■ Constant and texture memory

- Accessed by all the threads
- Texture memory: used to improve performance of reads that exhibit spatial locality among the threads

■ Latency of data access

- Device memory: 200-400 clock cycles (about 300 ns)
 - Shared memory: 20-30 clock cycles (about 5 ns)
-

Differences with the CPU threads

- Context switching fast: The state of a thread (thread block) stored in shared memory and registers stay till execution completion
 - Cache explicitly managed: User's program will have to explicitly bring the frequently accessed data from the device to the shared memory
-

- Backup

SPs and SMX

- GPU cores are called streaming processors (SPs)
- SP cores grouped into streaming multiprocessors (SMX)
- Promotes coarse and fine-level parallelism
- Kepler K40 – 15 SMXs consisting of 192 SP cores each for a total of 2880 SP cores
- Light-weight – 1.065 GHz
- SIMD execution (actually SIMT)

