OSU Oregon State University

Electrical Engineering & Computer Science

COLLEGE OF ENGINEERING

ECE/CS 472/572:

Special Topics Part III: GPU Architecture Introduction

Prof. Lizhong Chen

Spring 2019

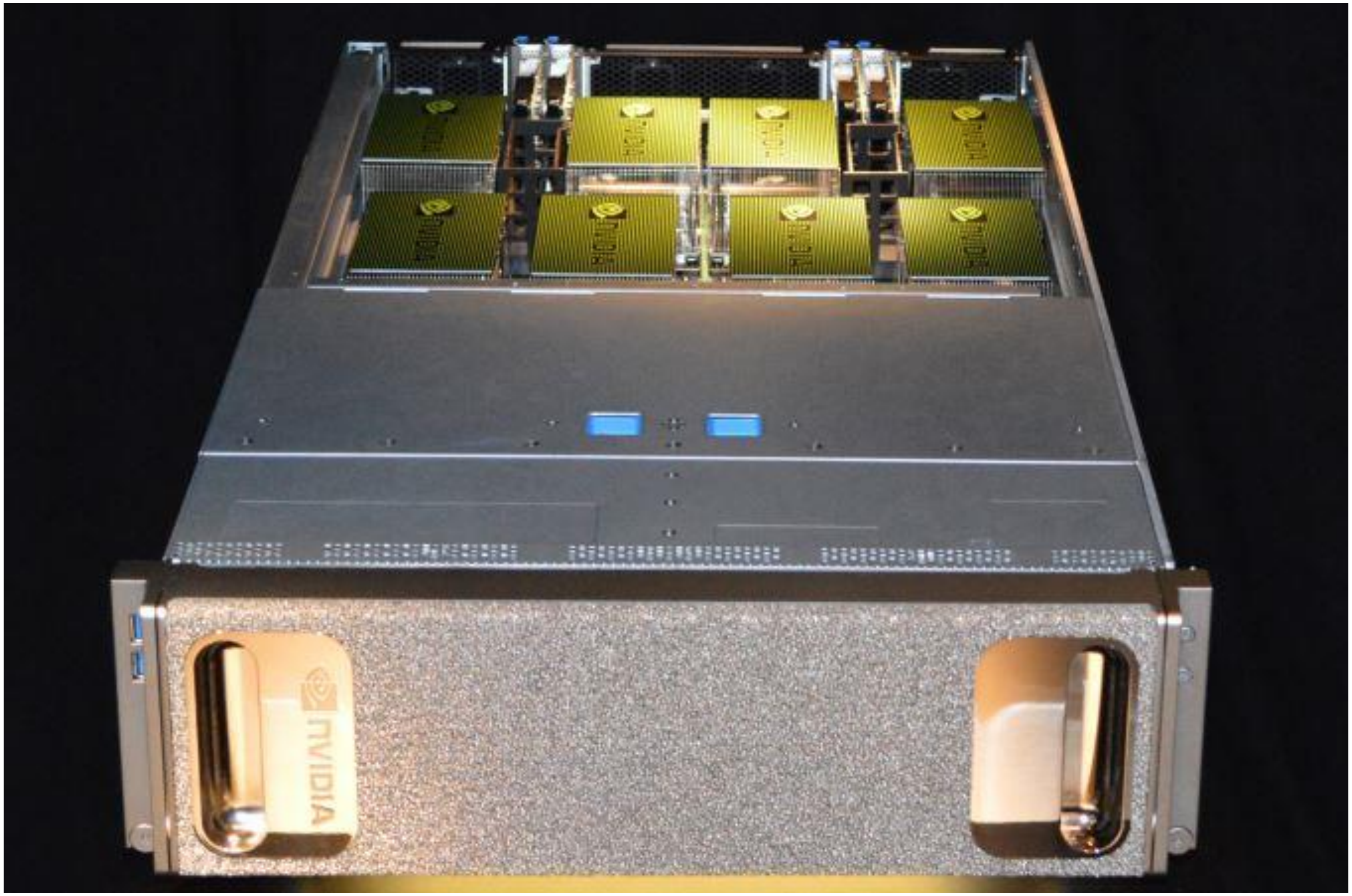
GPU: System Organization

- Nvidia DGX-1 High-performance computing (HPC) sever

http://www.anandtech.com/show/10229/nvidia-announces-dgx1-server

GPU: System Organization

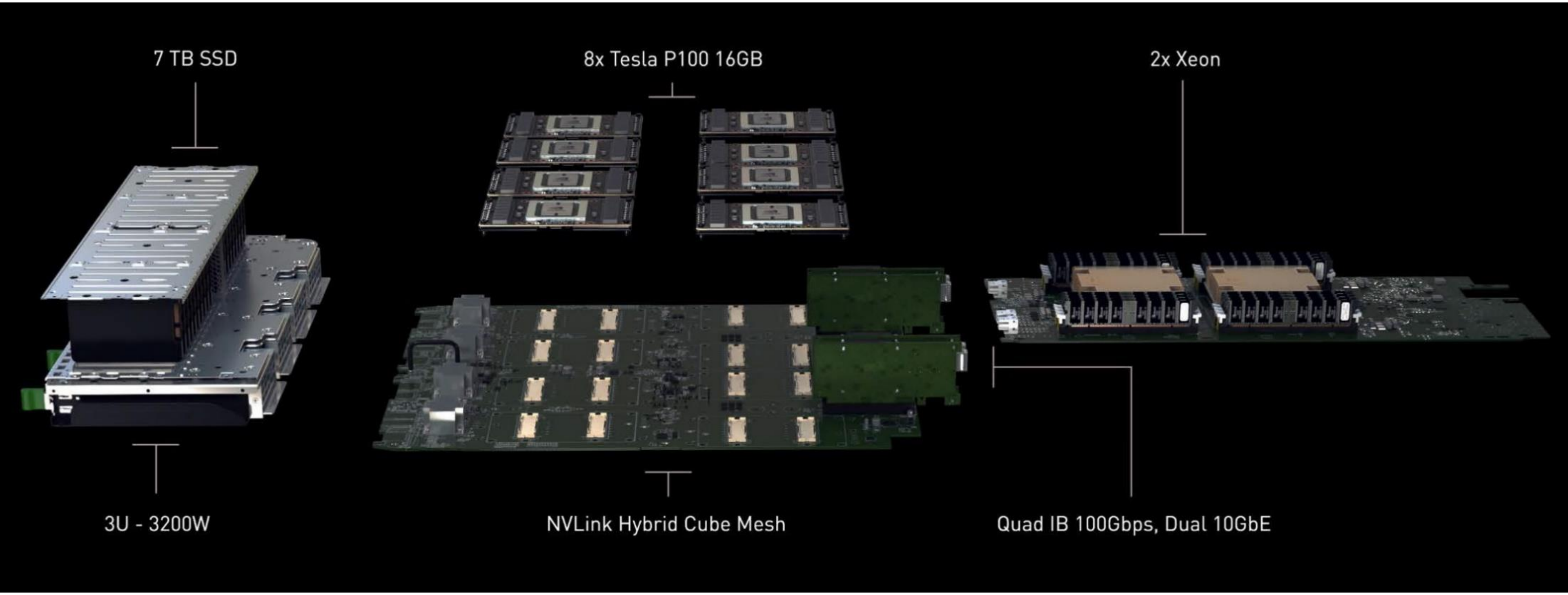
- Nvidia DGX-1 with panel open



http://www.anandtech.com/show/10229/nvidia-announces-dgx1-server

GPU: System Organization

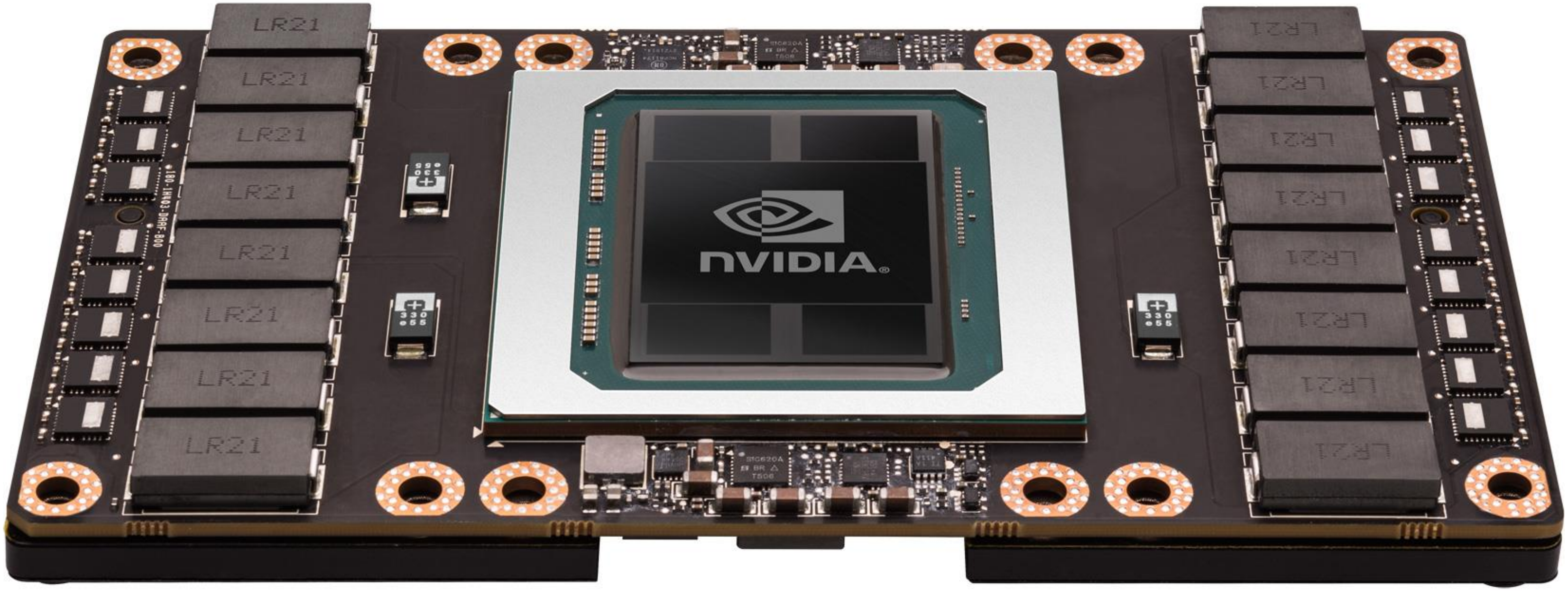
- Inside Nvidia DGX-1



http://www.anandtech.com/show/10229/nvidia-announces-dgx1-server

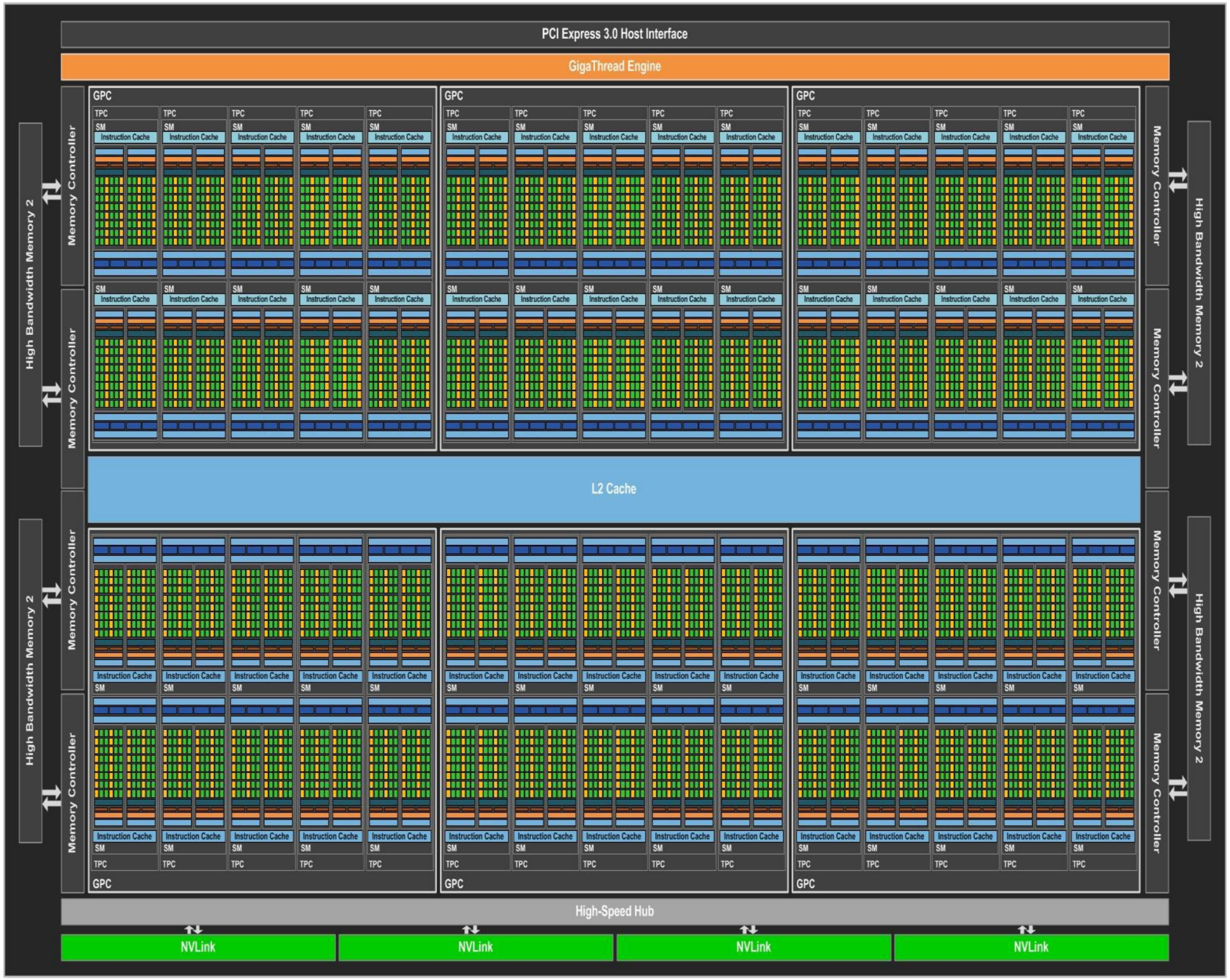
GPU: System Organization

- Tesla P100 in Nvidia DGX-1



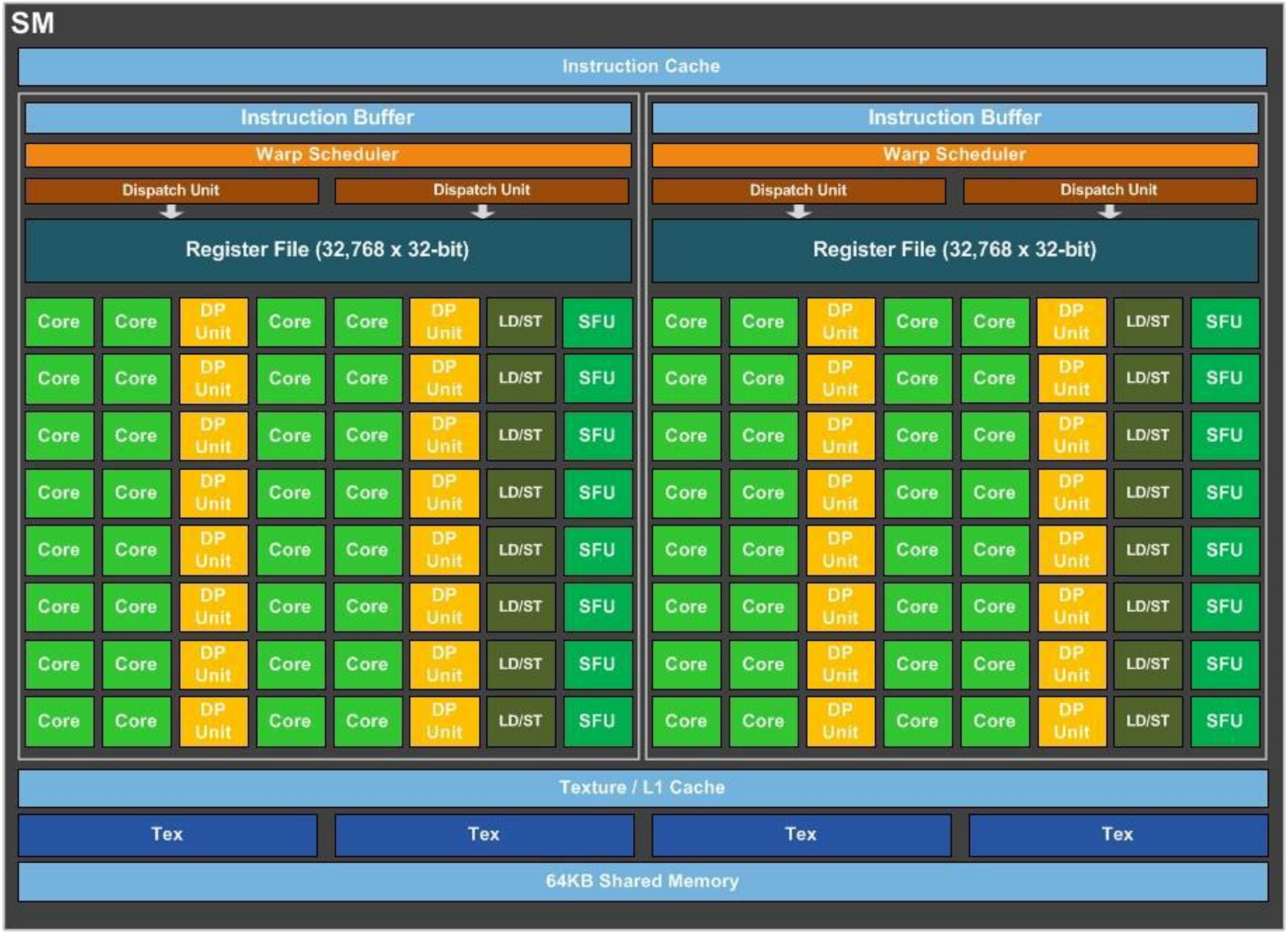
http://www.anandtech.com/show/10229/nvidia-announces-dgx1-server

NVIDIA Tesla GP100



- 6 GPCs
- 30 TPCs
- 60 SMs
- 3840 Cores
- 240 Tex
- 8 MCs
  - 4096-bit

NVIDIA Telsa GP100 SM



SM

Instruction Cache

Instruction Buffer	Instruction Buffer
Warp Scheduler	Warp Scheduler
Dispatch Unit / Dispatch Unit	Dispatch Unit / Dispatch Unit
Register File (32,768 x 32-bit)	Register File (32,768 x 32-bit)

Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU
Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU	Core	Core	DP Unit	Core	Core	DP Unit	LD/ST	SFU

Texture / L1 Cache

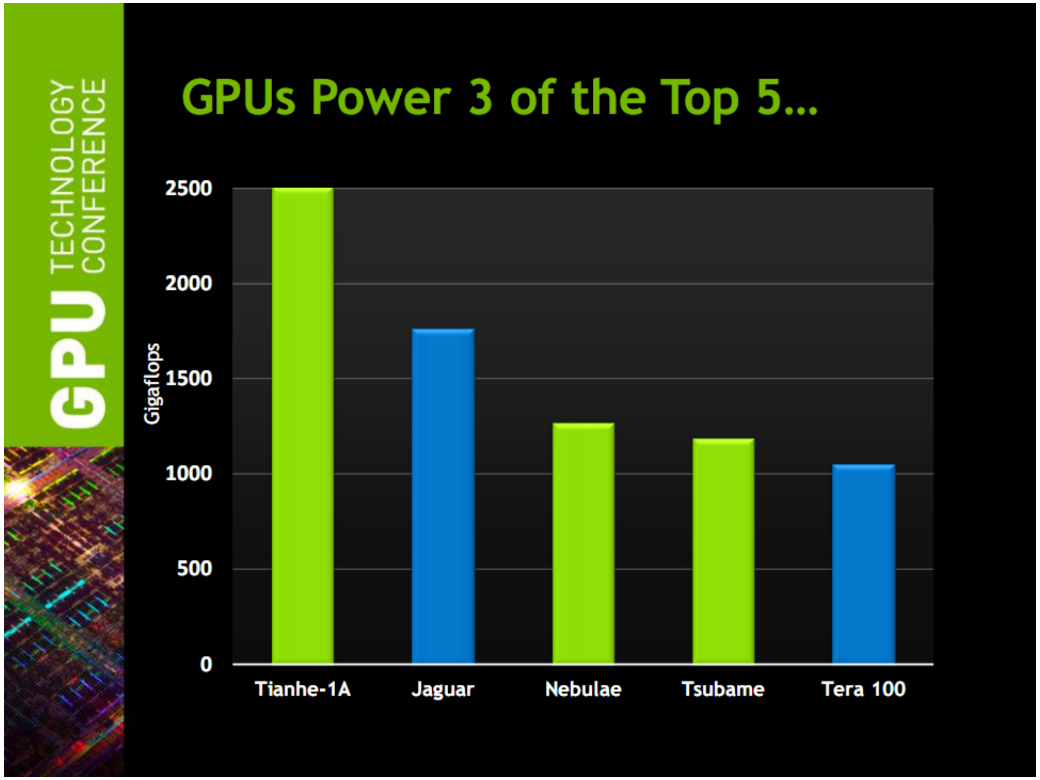
Tex	Tex	Tex	Tex

64KB Shared Memory

- 64 FP32 cores
  - Supports 16-bit FP
- 32 FP64 cores
- 2 32K 32-bit Registers
- Separate shared mem.
- Unified Texture/L1 $

GPUs Everywhere

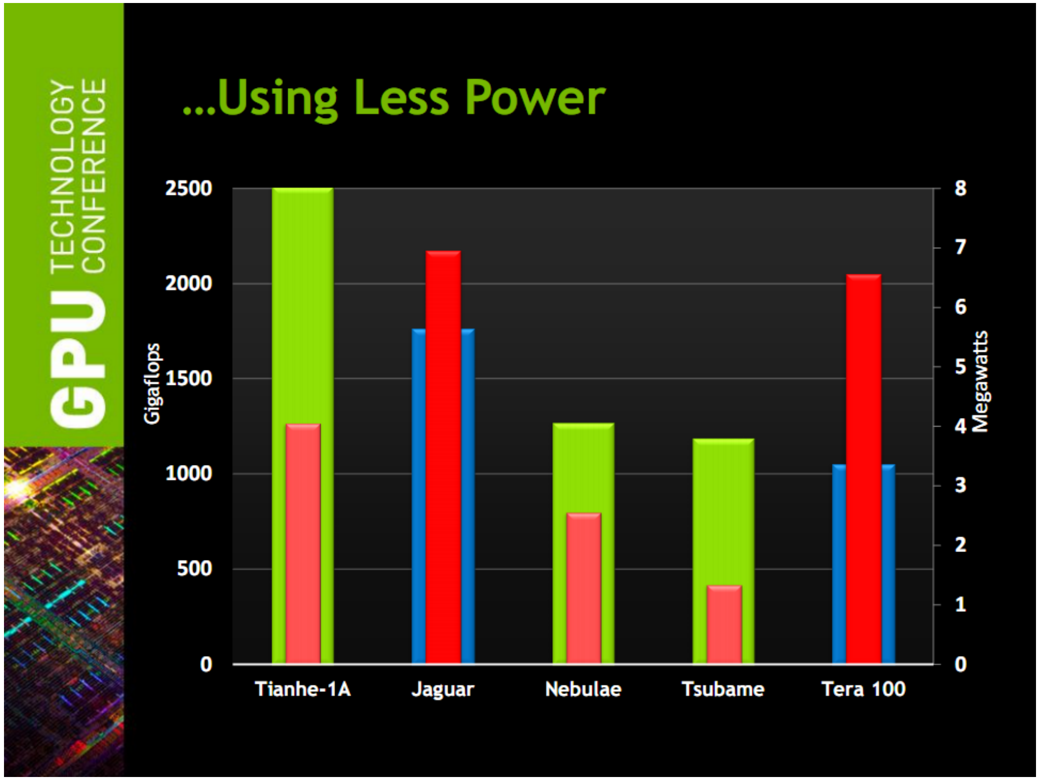
- Many of the top-ranked supercomputers are based on GPU
  - The World #1 supercomputer is powered by NVIDIA GPU



GPU Computing: Past, Present and Future, David Luebke, NVIDIA

But Why are GPUs Everywhere?

- It is the performance/Watt metric
  - Mobile computers, Supercomputers, Laptops: Share power agony



GPU Computing: Past, Present and Future, David Luebke, NVIDIA

GPU: Initially for Graphics

- Take a scene and project on a screen
  - How does each polygon translate into a screen pixel

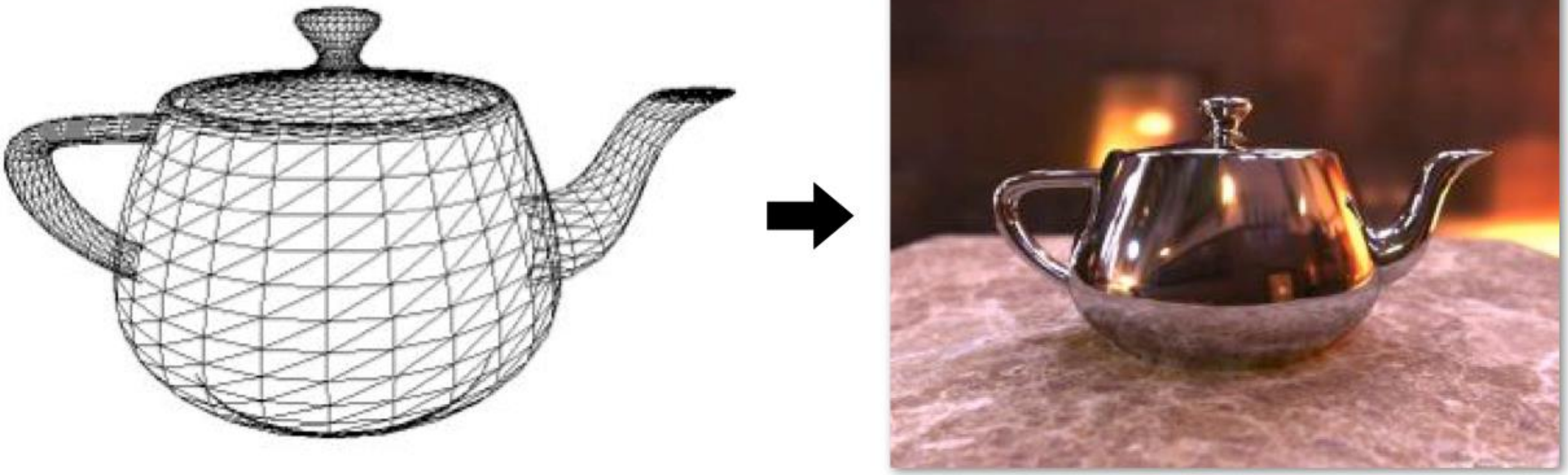
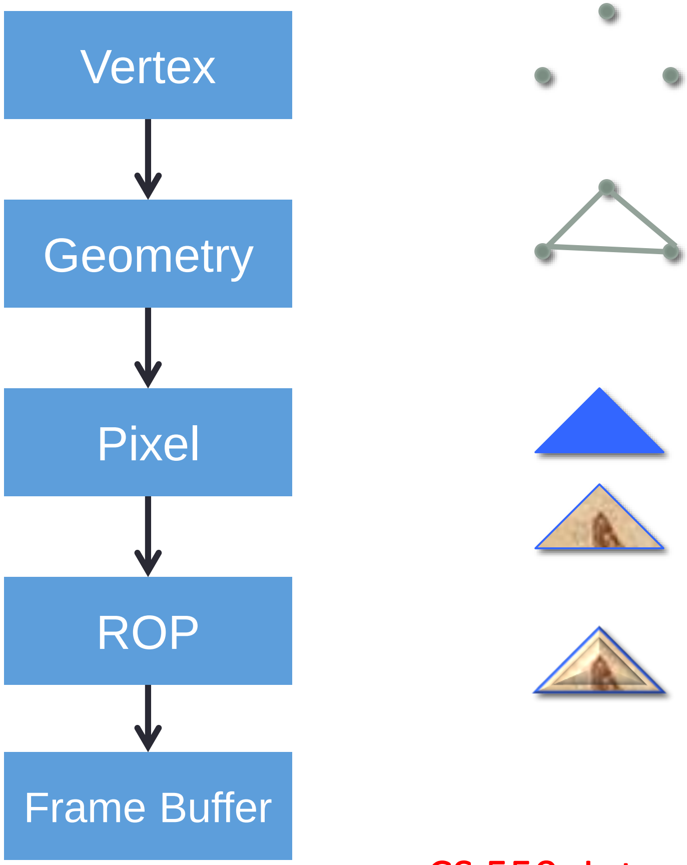


Image credit: Henrik Wann Jensen

http://www.cs.cmu.edu/afs/cs/academic/class/15869-f11/www/lectures/01_intro.pdf

Graphics-based GPU Pipeline (before 2006)



Vertex

Geometry

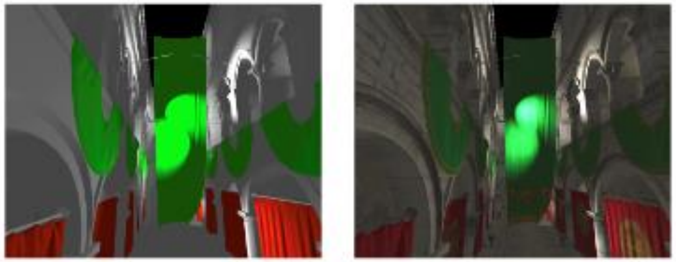
Pixel

ROP

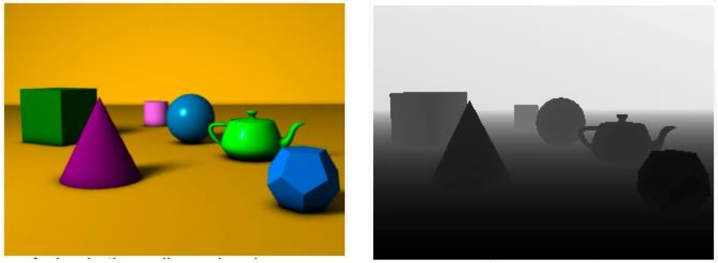
Frame Buffer

GPU Pipeline Functions

- Vector Shader : Takes as input vectors given in 3D format (X,Y,Z) and projects to a 2D space
- Pixel Shader: Takes each projected point in 2D space and puts texture, color and depth for each point
  - Texture mapping: Adding an image on top of a scene
  - Depth mapping: Shows which objects will obscure other objects



Example Image: Before and After Texture Mapping

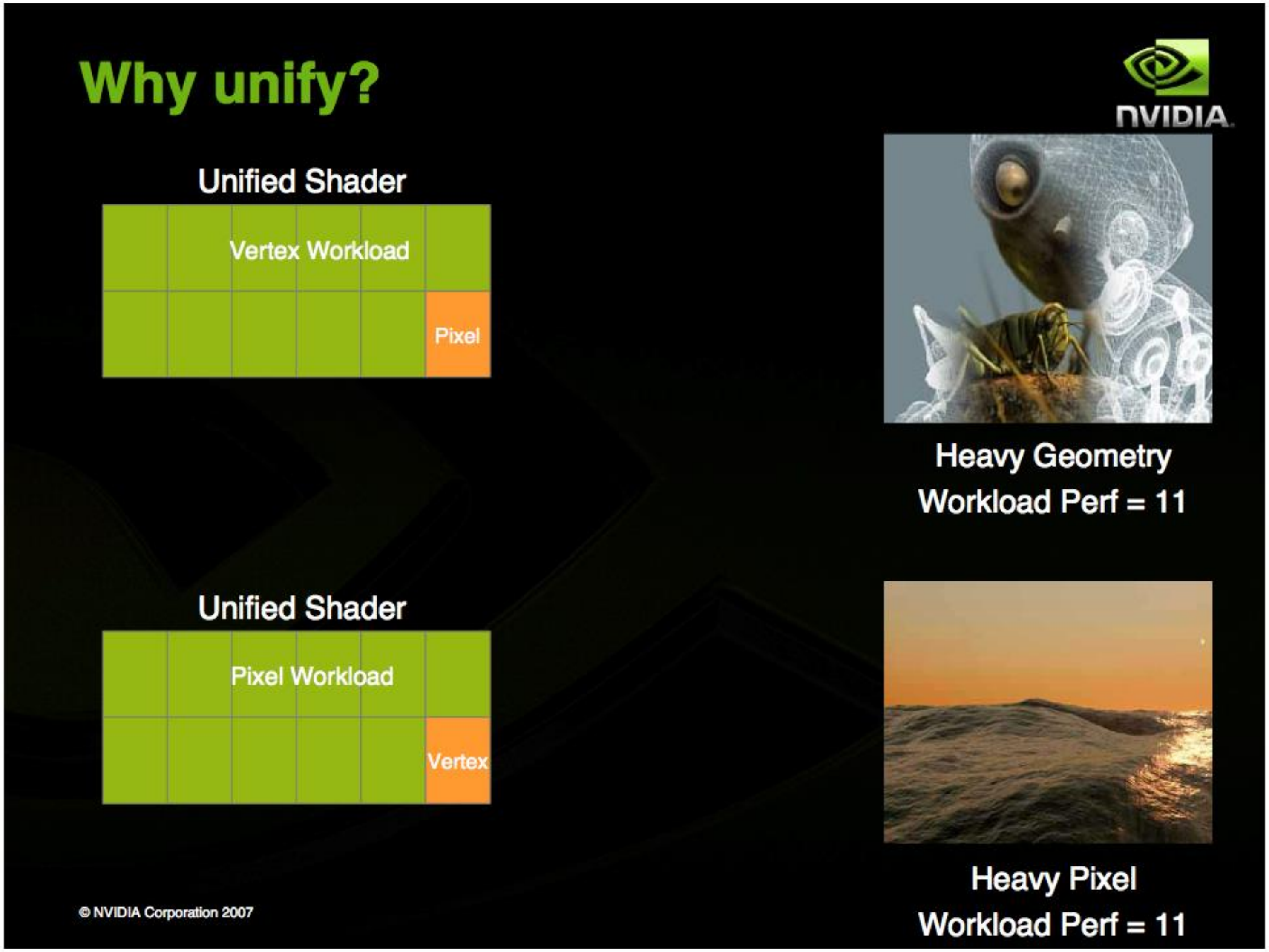


Example Image: Z-Buffer Representation of Image

- Geometry Shaders: Operates directly on graphics primitives, such as lines, triangles (as opposed to pixels)

Doggett, M., "Texture Caches," *Micro, IEEE* , vol.32, no.3, pp.136,141, May-June 2012
http://www.felixgers.de/teaching/jogl/depthbufferALgo.html

Need for Unified Shader



Why unify?

Unified Shader

Vertex Workload

Pixel

Heavy Geometry
Workload Perf = 11

Unified Shader

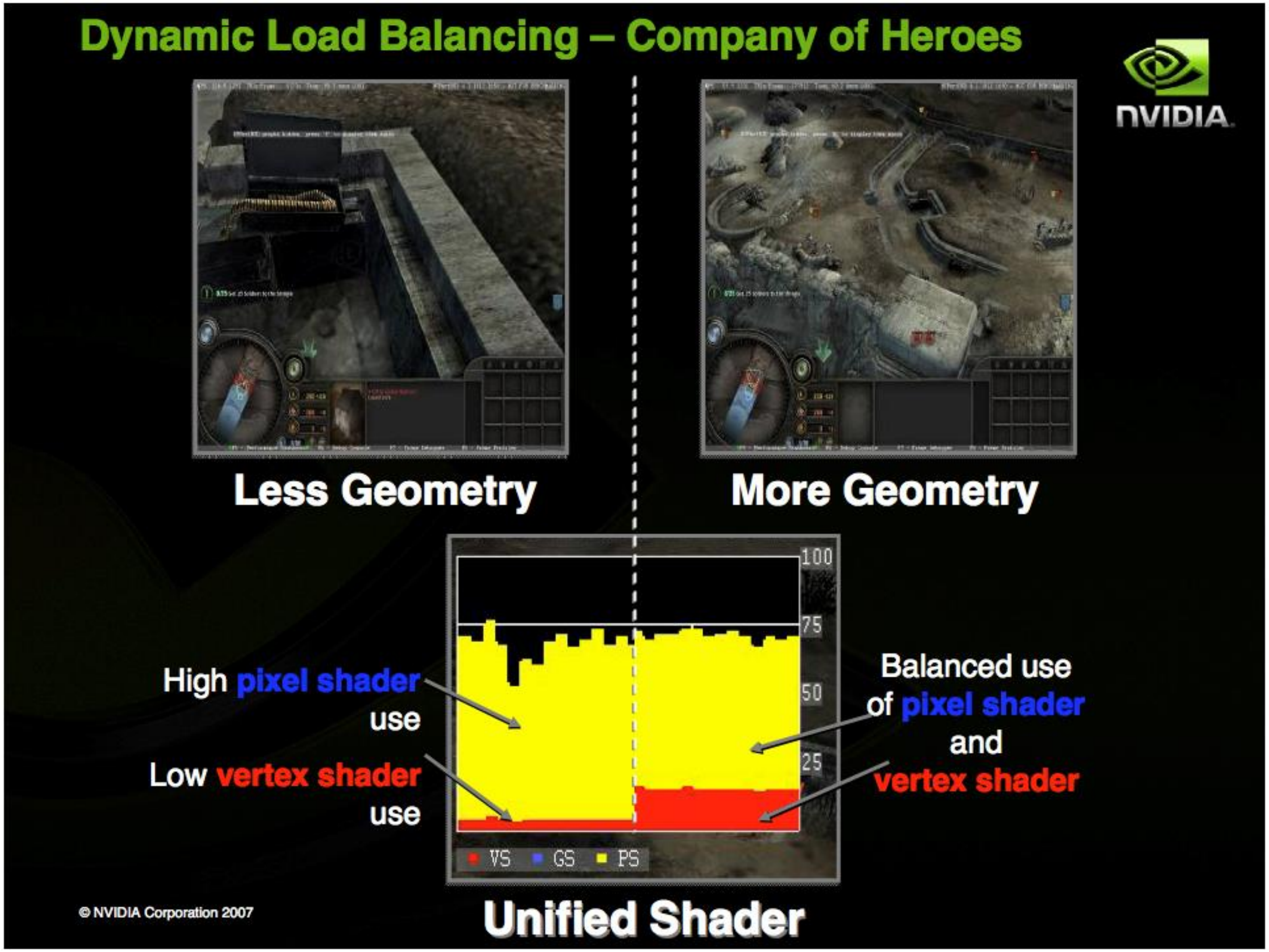
Pixel Workload

Vertex

Heavy Pixel
Workload Perf = 11

GPU Architecture: Implications & Trends D. Luebke, Nvidia

Resource Balancing Unified Shader



Dynamic Load Balancing – Company of Heroes

Less Geometry

More Geometry

High **pixel shader** use

Low **vertex shader** use

Balanced use of **pixel shader** and **vertex shader**

Unified Shader

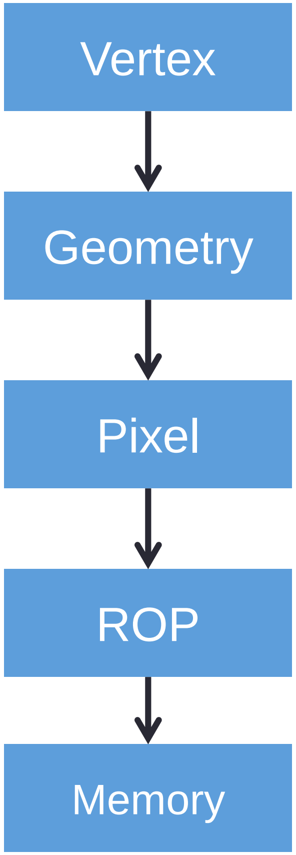
© NVIDIA Corporation 2007

GPU Architecture: Implications & Trends D. Luebke, Nvidia

GPU Pipeline evolution

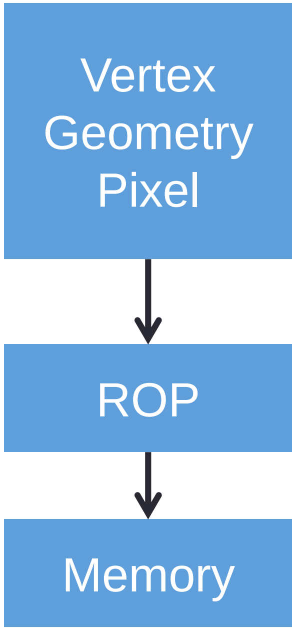
Before 2006

- Processor per function

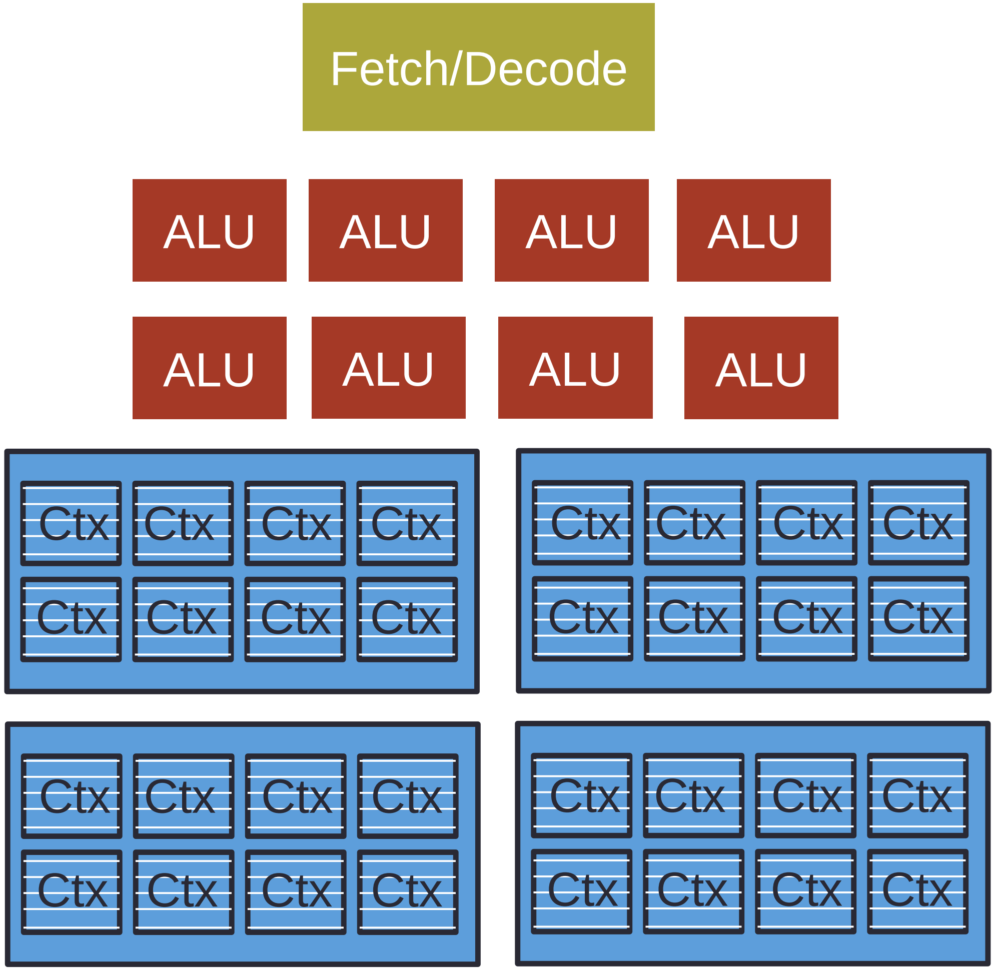


After 2006

- Unified Shader Model(GeForce 8 Series)
- Any work can be performed on any shader core → High performance computing



Unified Shader Model



Fetch/Decode

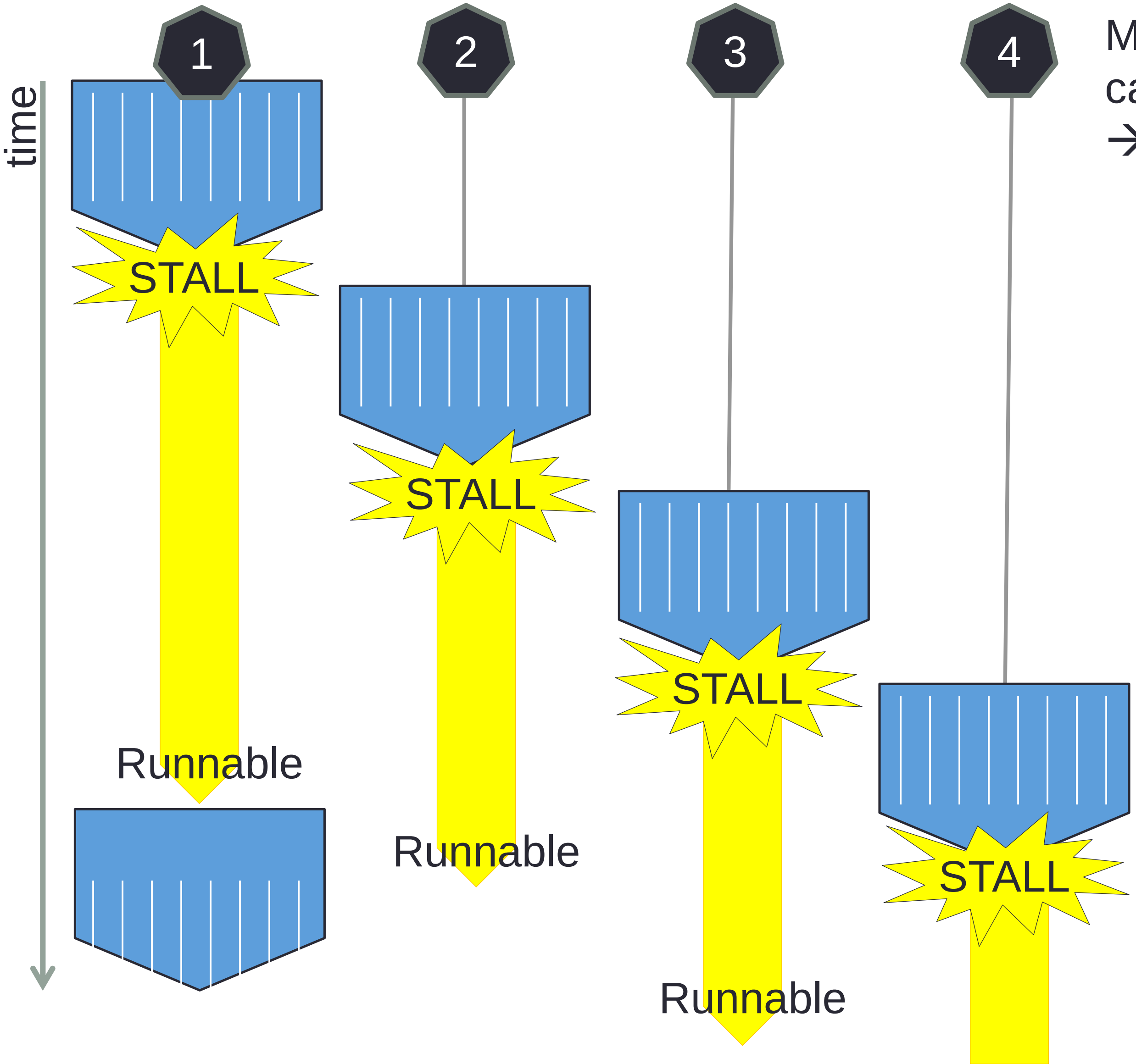
ALU ALU ALU ALU

ALU ALU ALU ALU

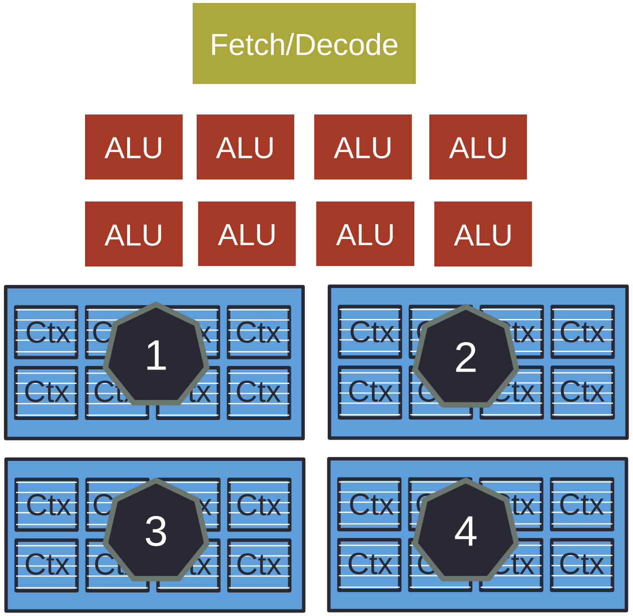
Ctx Ctx Ctx Ctx Ctx Ctx Ctx Ctx
Ctx Ctx Ctx Ctx Ctx Ctx Ctx Ctx

Ctx Ctx Ctx Ctx Ctx Ctx Ctx Ctx
Ctx Ctx Ctx Ctx Ctx Ctx Ctx Ctx

Unified Shader Model

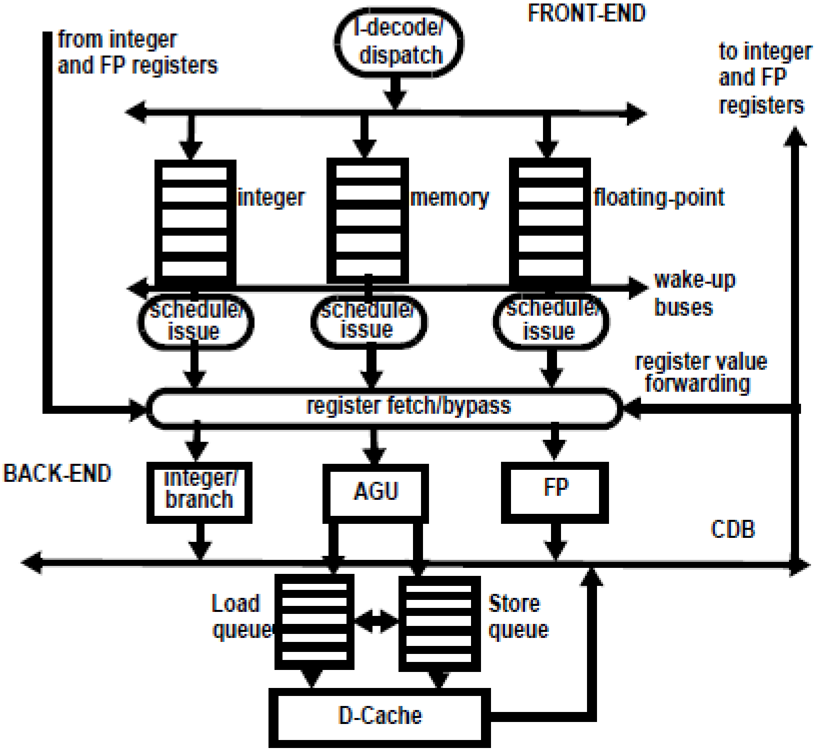


Multiple Contexts(Thread groups)
can be interleavingly scheduled
→ Can hide stall times



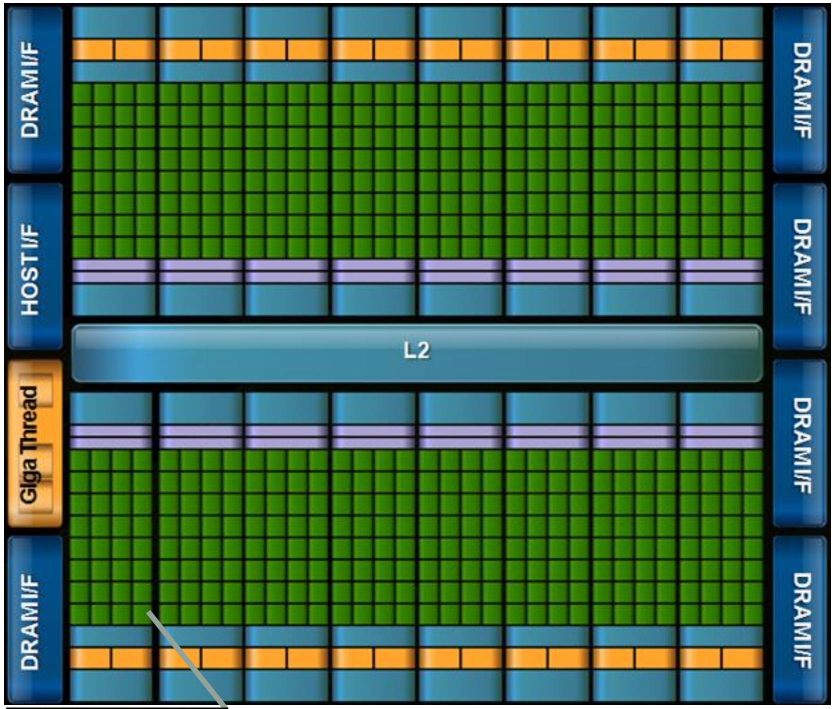
GPU vs CPU

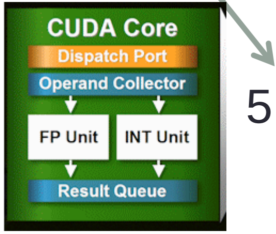
Complex OoO CPU



Parallel Computer Organization and Design, Cambridge University Press

GPU (Nvidia FERMI)





512 CUDA cores

http://techreport.com/review/17670/nvidia-fermi-gpu-architecture-revealed

Basic idea of high performance GPU

- Many simple cores
  - No fancy Branch prediction
  - No Complex O-o-O control logic
  - No memory prefetcher
  - No cache coherence
  - . . .
- Unified shader cores
  - Sharing instruction stream across groups of fragments
- Stall latency hiding
  - When a group stalls, work on another group