King Abdullah University of Science and Technology

KAUST

CS 380 - GPU and GPGPU Programming
Lecture 9: GPU Architecture 6

Markus Hadwiger, KAUST

Reading Assignment #5 (until Oct 5)

Read (required):

- Programming Massively Parallel Processors book, 2nd edition (!),
  Chapter 4 (*Data-Parallel Execution Model*) [beginning is very similar to Chap. 4, 1st ed.]
- NVIDIA CUDA C++ Programming Guide (Sep 2020):
  Read Chapter 2.5 (Compute Capability); go through Appendix I (Compute Capabilities)

  `https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf`

Read (optional):

- NVIDIA Pascal (GP100) white paper:

`https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf`

- NVIDIA Volta (V100) white paper:

`http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf`

- NVIDIA Turing (TU102, TU104, TU106) white paper:

`https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf`

NVIDIA Architectures (since first CUDA GPU)

Tesla: 2007-2009

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

Fermi: 2010

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

Kepler: 2012

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

Maxwell: 2015

- GM107, ... (GTX 750Ti, ...)
  GM204, ... (GTX 980, Titan X, ...)

Pascal: 2016

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

Volta: 2017/2018

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

Turing: 2018/2019

- TU102, TU104, TU106, TU116, ...
  (Titan RTX, RTX 2070, 2080, 2080Ti, ...)

Ampere: 2020

- GA100, GA102, GA104, ...
  (A100, RTX 3070, 3080, 3090, ...)

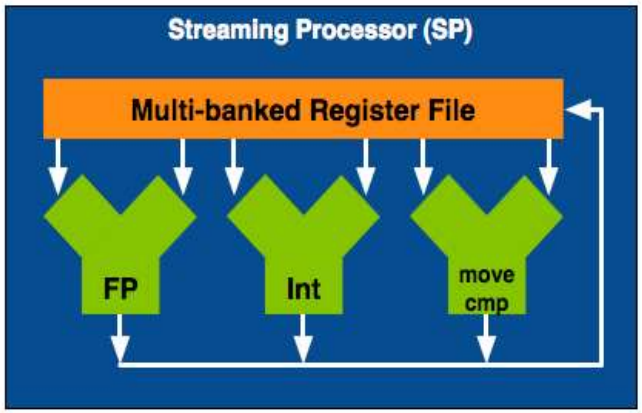
NVIDIA Tesla Architecture

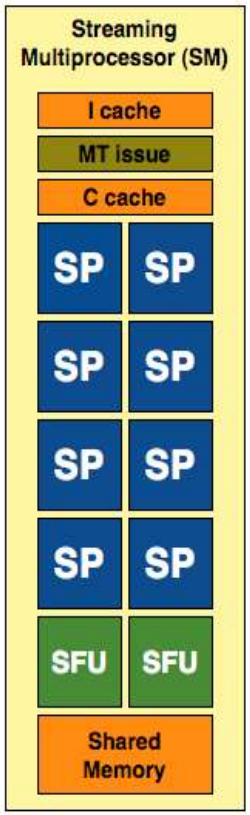
2007-2009

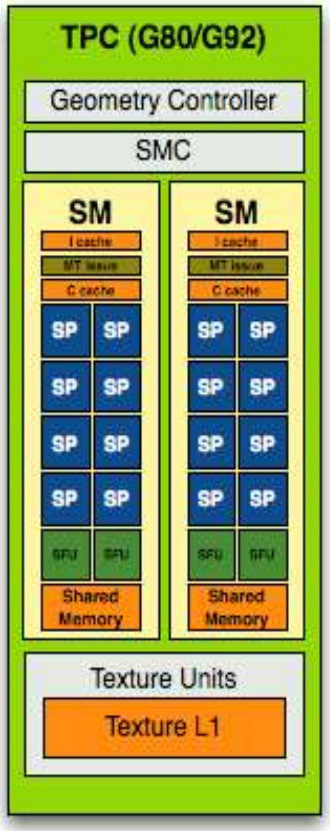
G80, G9x: 2007 (Geforce 8800, ...)
GT200: 2008/2009 (GTX 280, ...)

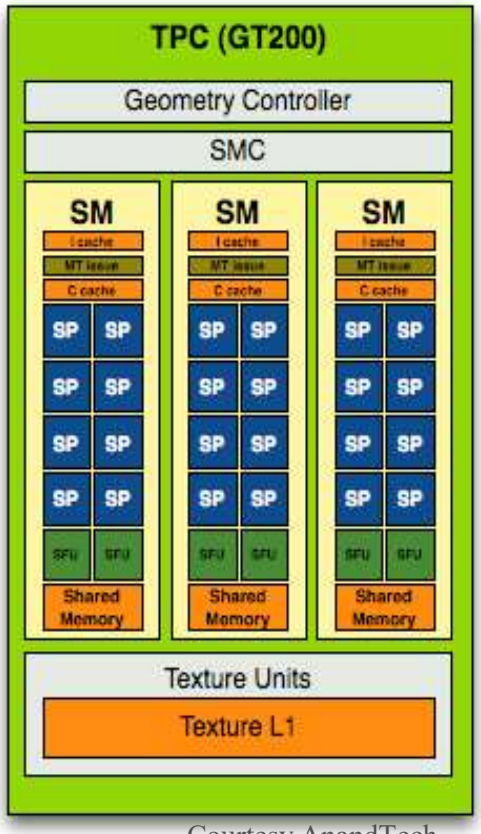
(this is not the Tesla product line!)

NVIDIA Tesla Architecture (not the Tesla product line!), G80: 2007, GT200: 2008/2009









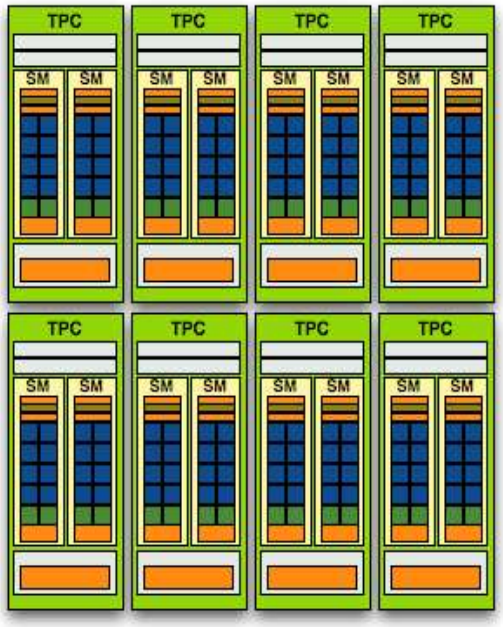
Courtesy AnandTech

G80: first CUDA GPU!

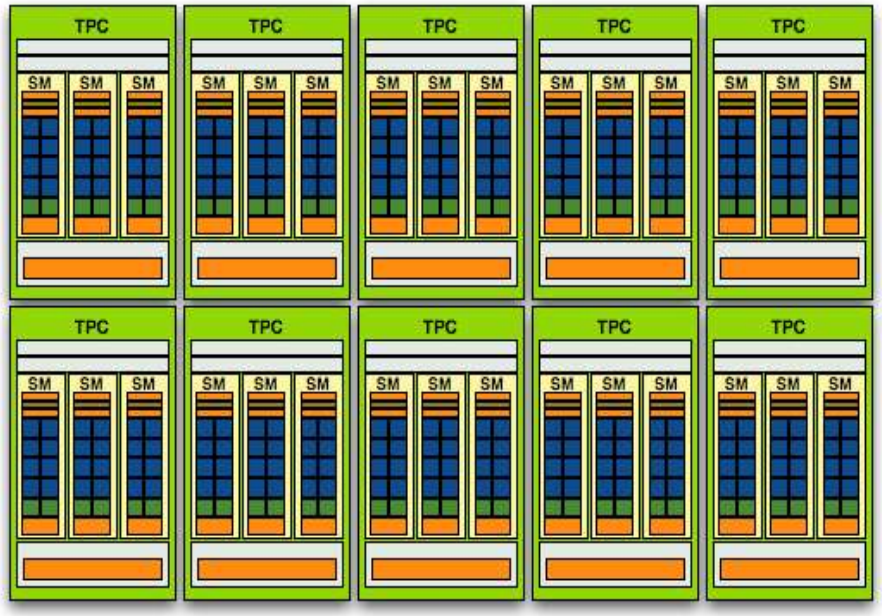
- Streaming Processor (SP) [or: CUDA core; or: FP32 / FP64 / INT32 core, ...]
- Streaming Multiprocessor (SM)
- Texture/Processing Cluster (TPC)

NVIDIA Tesla Architecture (not the Tesla product line!), G80: 2007, GT200: 2008/2009

- G80/G92: 8 TPCs * (2 * 8 SPs) = 128 SPs [= CUDA cores]
- GT200: 10 TPCs * (3 * 8 SPs) = 240 SPs [= CUDA cores]
- **Arithmetic intensity** has increased (num. of ALUs vs. texture units)



G80 / G92



GT200

Courtesy AnandTech

NVIDIA Fermi Architecture

2010

GF100, ... (GTX 480, ...)

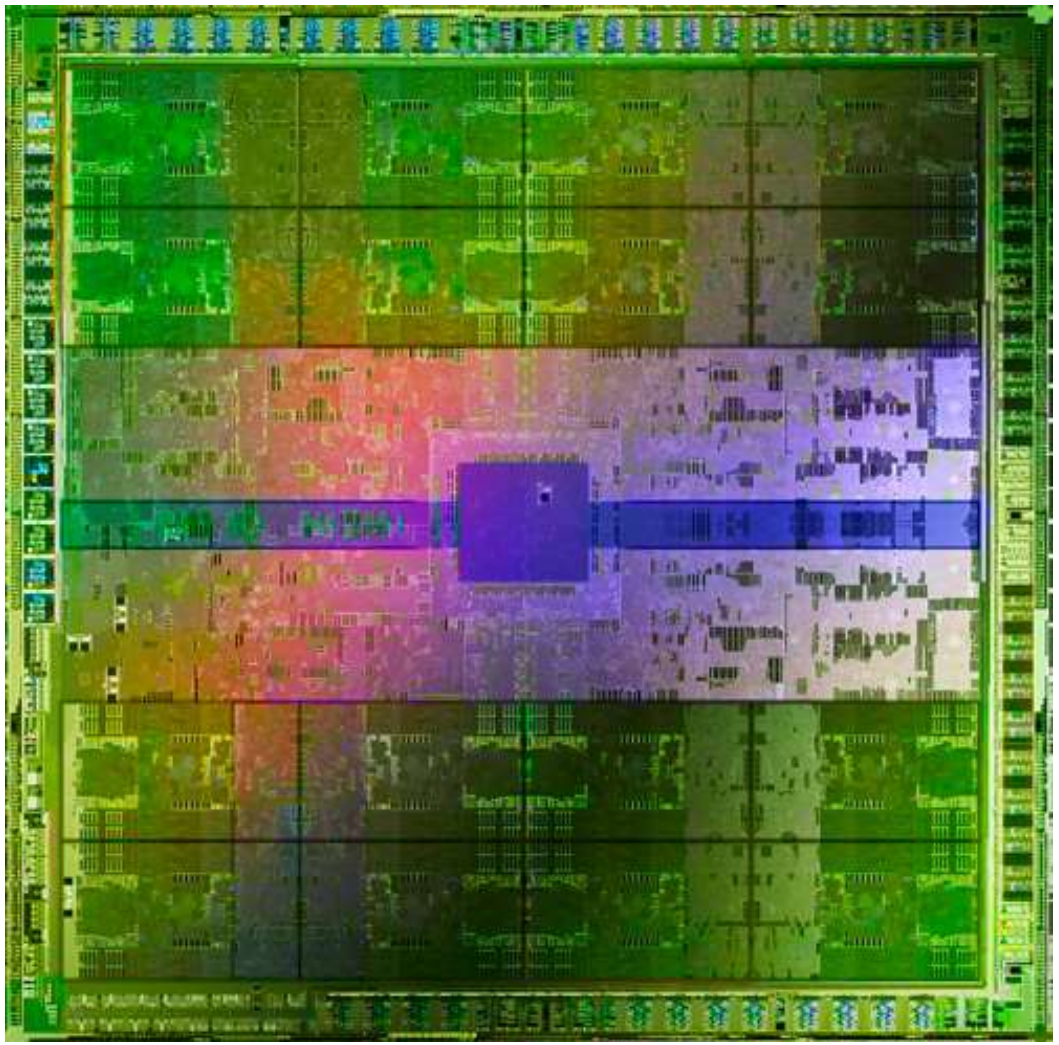
GF110, ... (GTX 580, ...)

NVIDIA Fermi Architecture (2010)

Full size

- 4 GPCs
- 4 SMs each
- 6 64-bit memory controllers (= 384 bit)

NVIDIA Fermi (GF100) Die Photo



Full size

- 4 GPCs
- 4 SMs each

NVIDIA Fermi SM (2010)

Streaming processors now called *CUDA cores*

32 CUDA cores per Fermi streaming multiprocessor (SM)

16 SMs = 512 CUDA cores

CPU-like cache hierarchy

- L1 cache / shared memory
- L2 cache

Texture units and caches now in SM

(instead of with TPC=multiple SMs in G80/GT200)

Graphics Processor Clusters (GPC)

(instead of TPC on GT200)

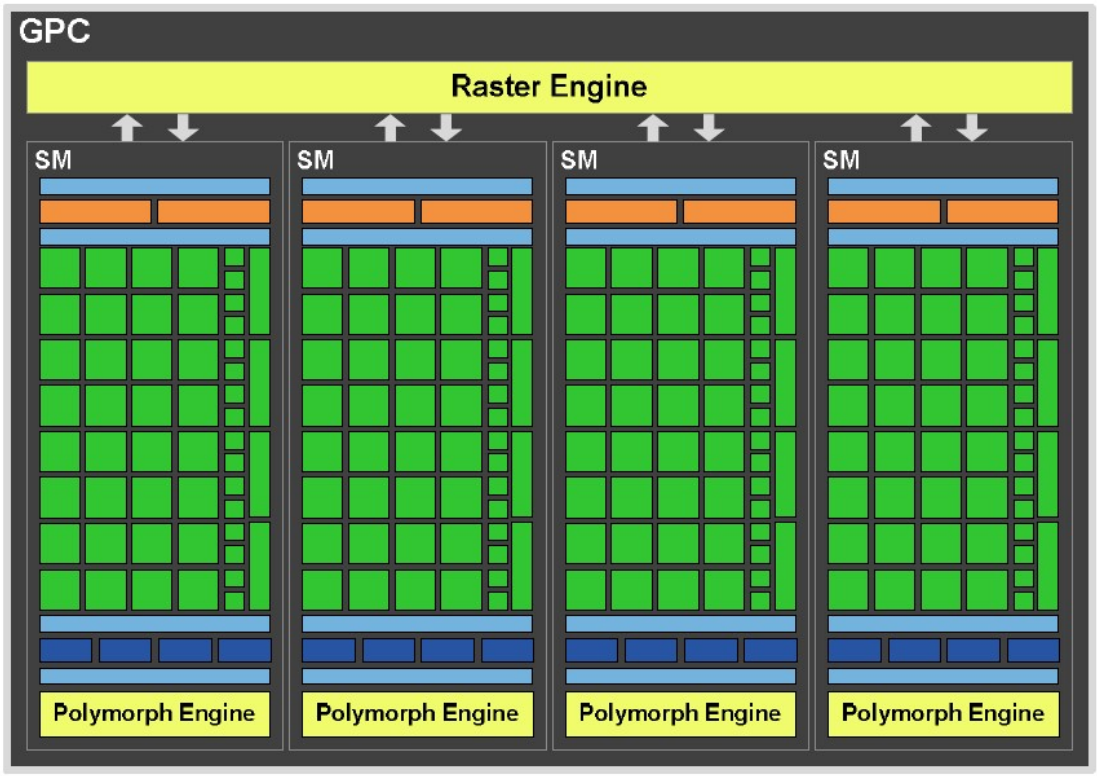
4 SMs

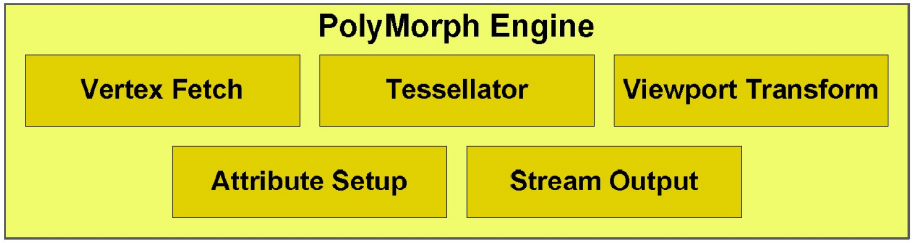
32 CUDA cores / SM

4 SMs / GPC =
128 cores / GPC

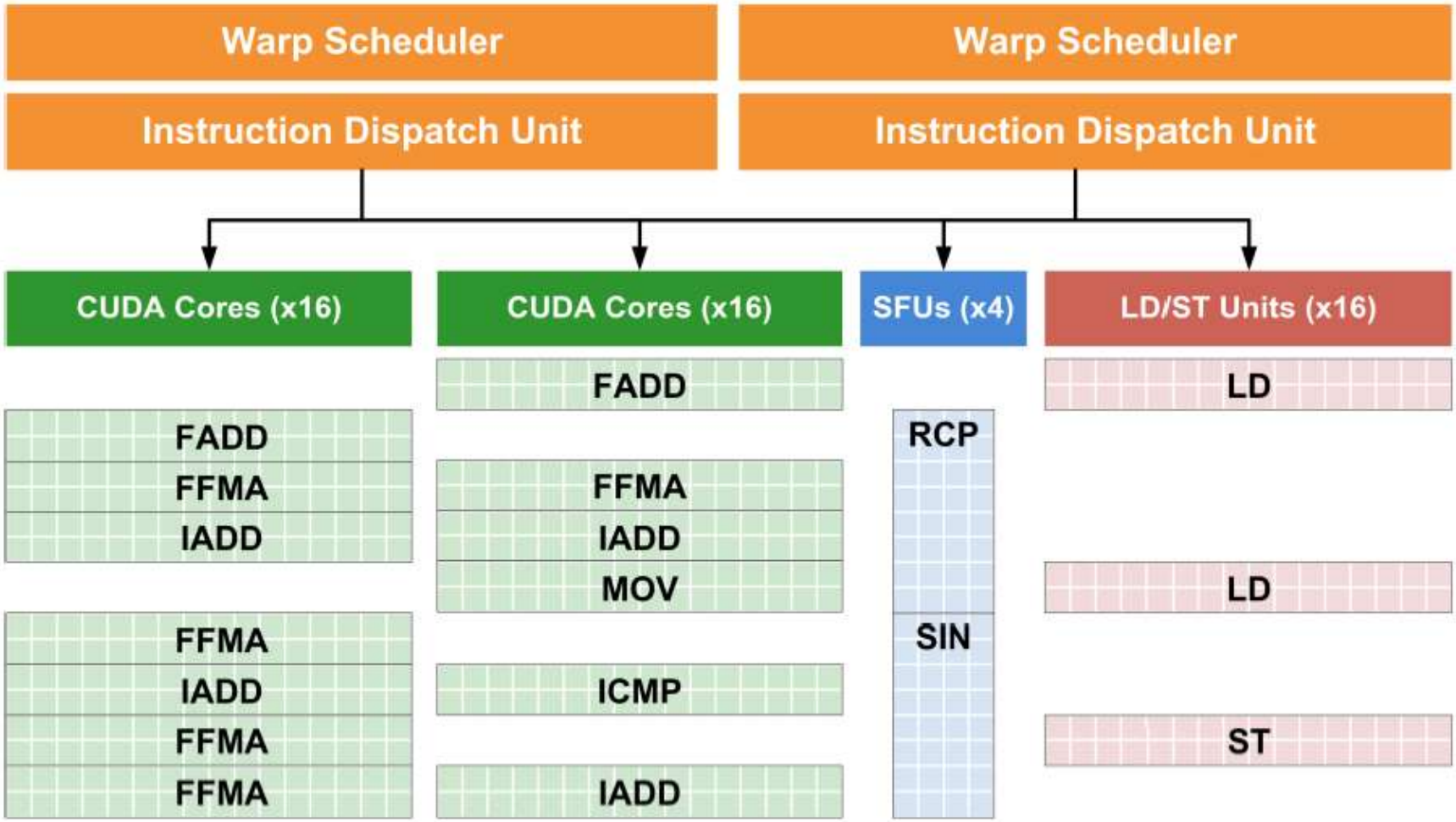
Decentralized rasterization and geometry

- 4 raster engines
- 16 ”PolyMorph” engines



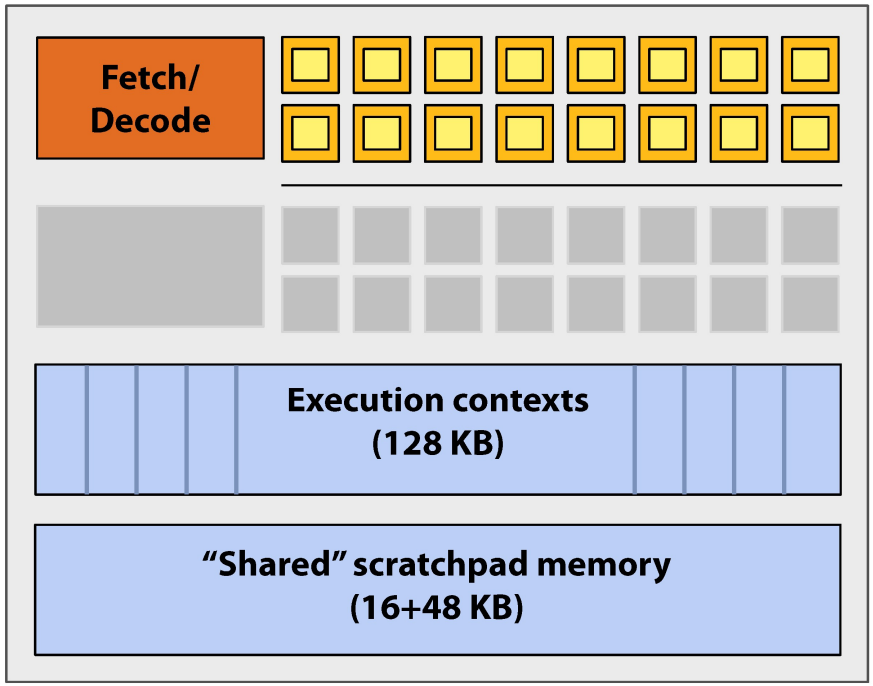


Dual Warp Schedulers

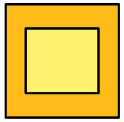


NVIDIA Fermi Architecture (2010)

NVIDIA GeForce GTX 480 "core"



Source: Fermi Compute Architecture Whitepaper
CUDA Programming Guide 3.1, Appendix G

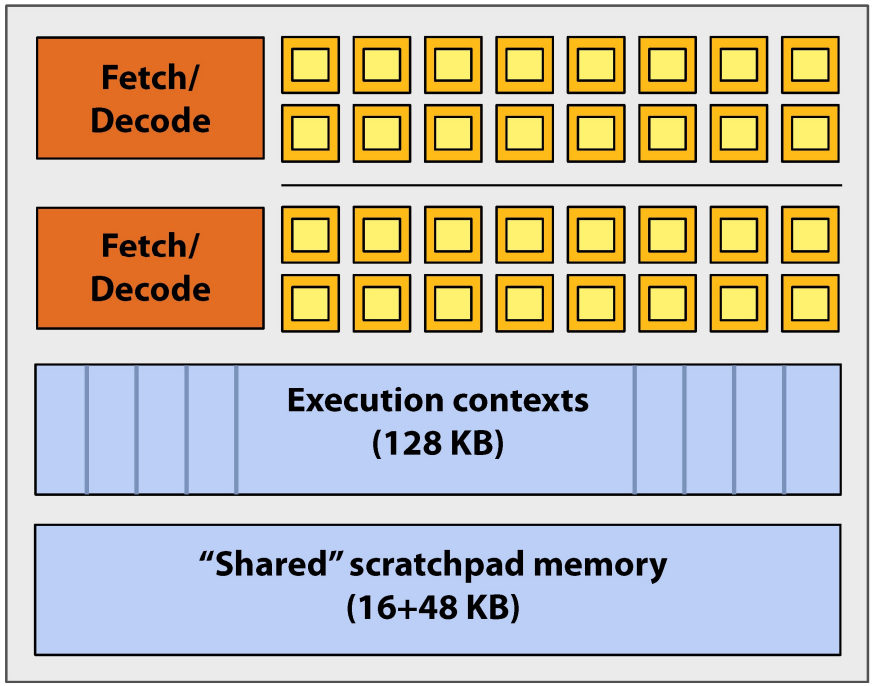


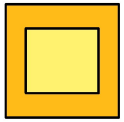
= SIMD function unit,
control shared across 16 units
(1 MUL-ADD per clock)

- Groups of 32 fragments share an instruction stream
- Up to 48 groups are simultaneously interleaved
- Up to 1536 individual contexts can be stored

NVIDIA Fermi Architecture (2010)

NVIDIA GeForce GTX 480 "core"





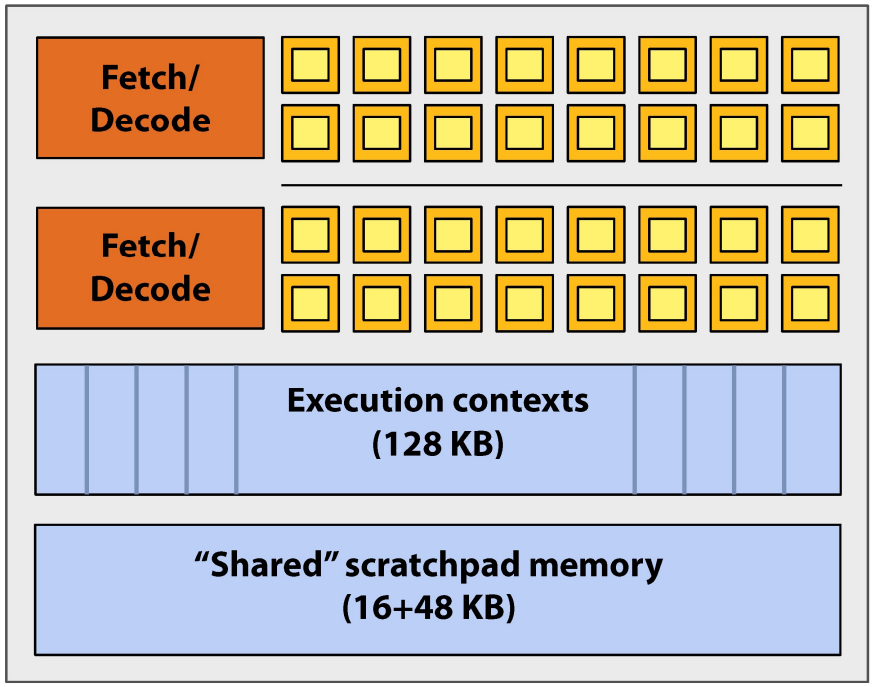
= SIMD function unit, control shared across 16 units (1 MUL-ADD per clock)

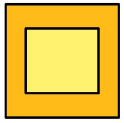
- The core contains 32 functional units
- Two groups are selected each clock (decode, fetch, and execute two instruction streams in parallel)

Source: Fermi Compute Architecture Whitepaper
CUDA Programming Guide 3.1, Appendix G

NVIDIA Fermi Architecture (2010)

NVIDIA GeForce GTX 480 "SM"





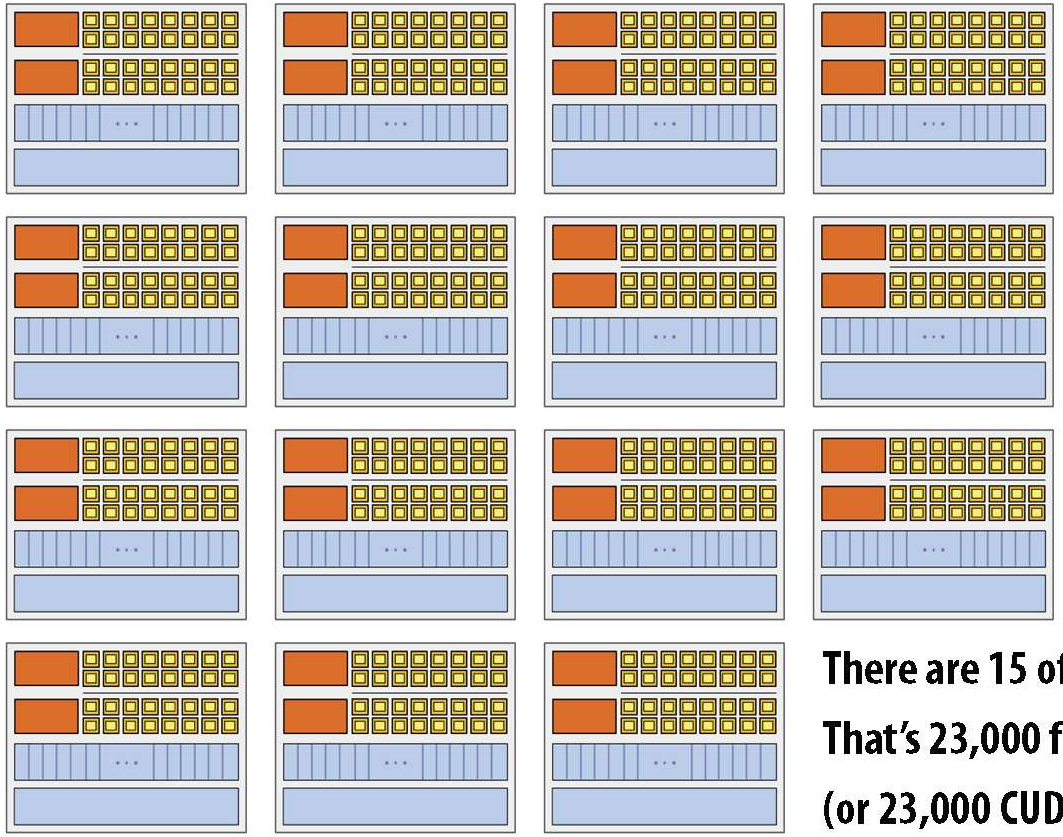
= CUDA core
(1 MUL-ADD per clock)

- The SM contains 32 CUDA cores
- Two warps are selected each clock (decode, fetch, and execute two warps in parallel)
- Up to 48 warps are interleaved, totaling 1536 CUDA threads

Source: Fermi Compute Architecture Whitepaper
CUDA Programming Guide 3.1, Appendix G

NVIDIA Fermi Architecture (2010)

NVIDIA GeForce GTX 480



There are 15 of these things on the GTX 480:

That's 23,000 fragments!

(or 23,000 CUDA threads!)

Thank you.