King Abdullah University of Science and Technology

KAUST

CS 380 - GPU and GPGPU Programming
Lecture 2: Introduction, Pt. 2

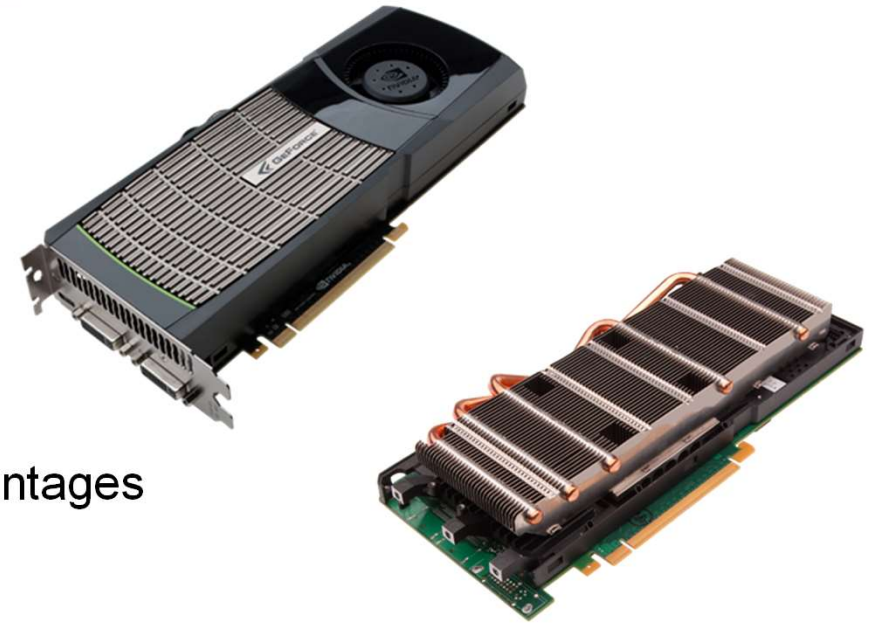
Markus Hadwiger, KAUST

Reading Assignment #1 (until Sep 6)

Read (required):

- Orange book, chapter 1 (*Review of OpenGL Basics*)
- Orange book, chapter 2 (*Basics*)

What are GPUs?



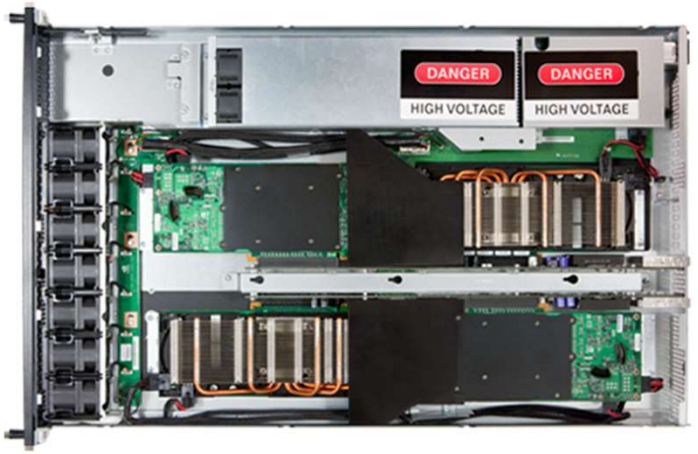
Graphics Processing Units

But evolved toward

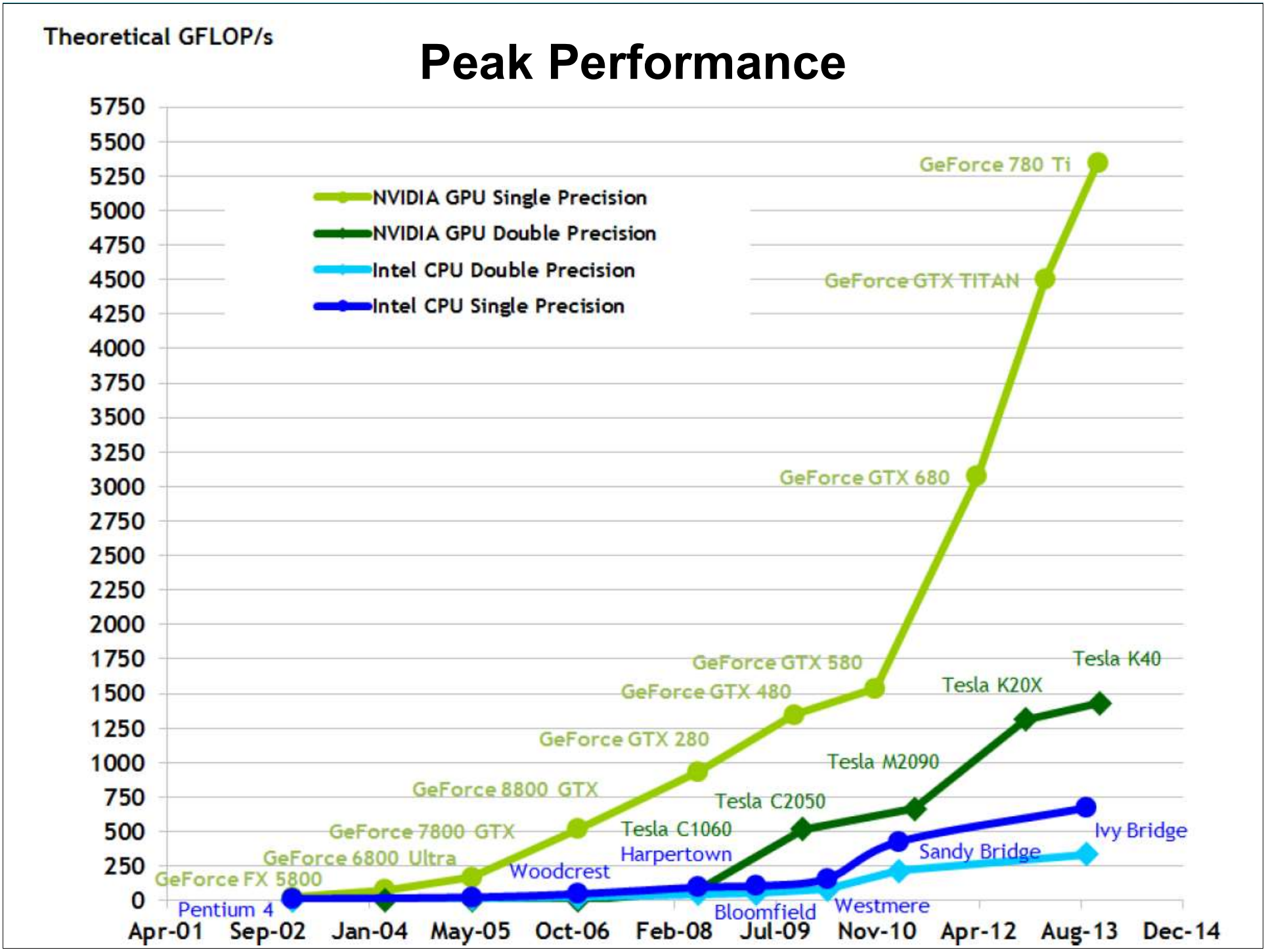
- Very flexible, massively parallel floating point co-processors
- But not entirely programmable!
- Fixed-function parts have definite advantages (e.g., texture filtering, z-buffering)

We will cover both perspectives

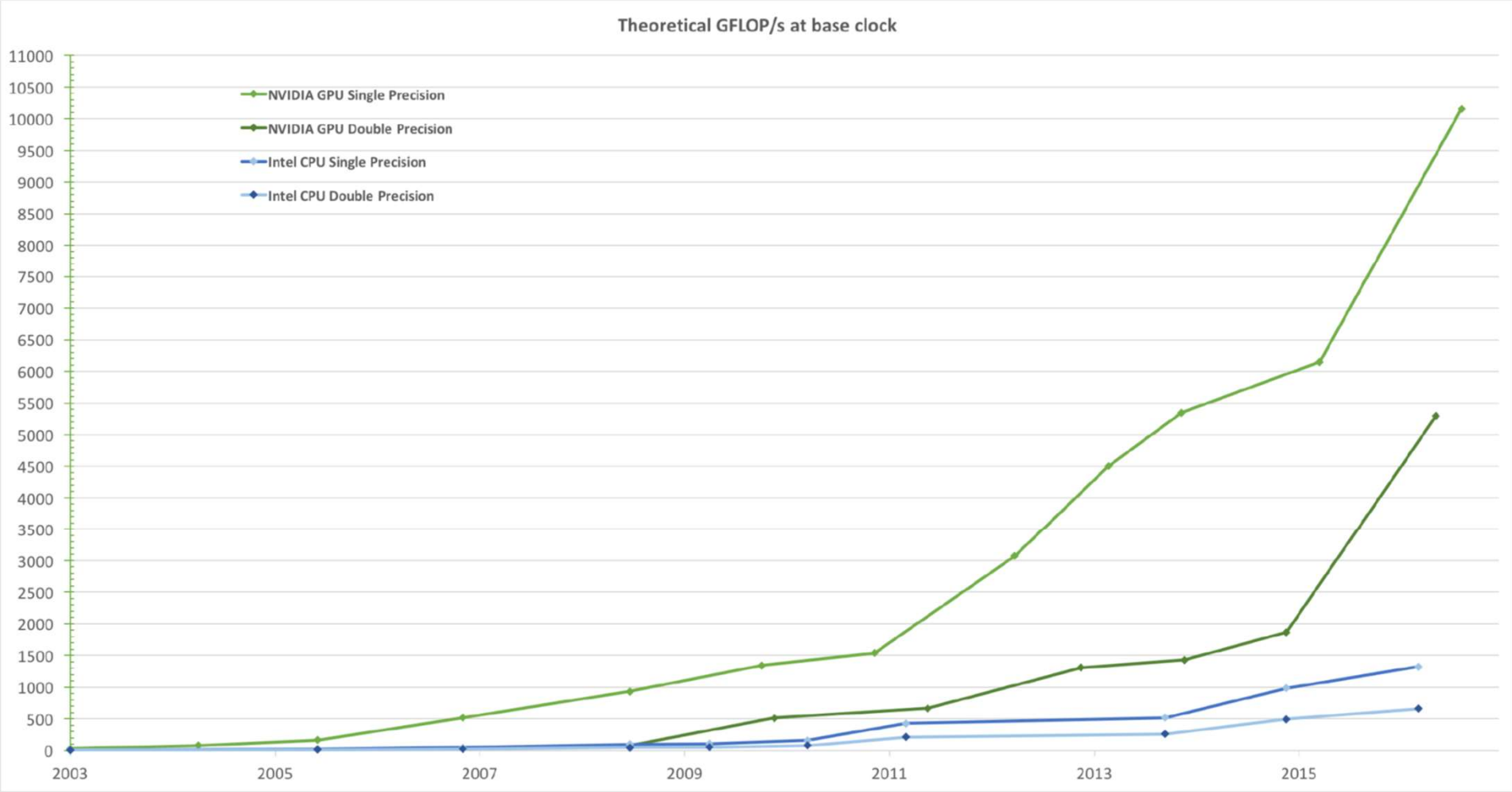
- GPUs for graphics
- GPU computing (GPGPU – general purpose computation on GPU)

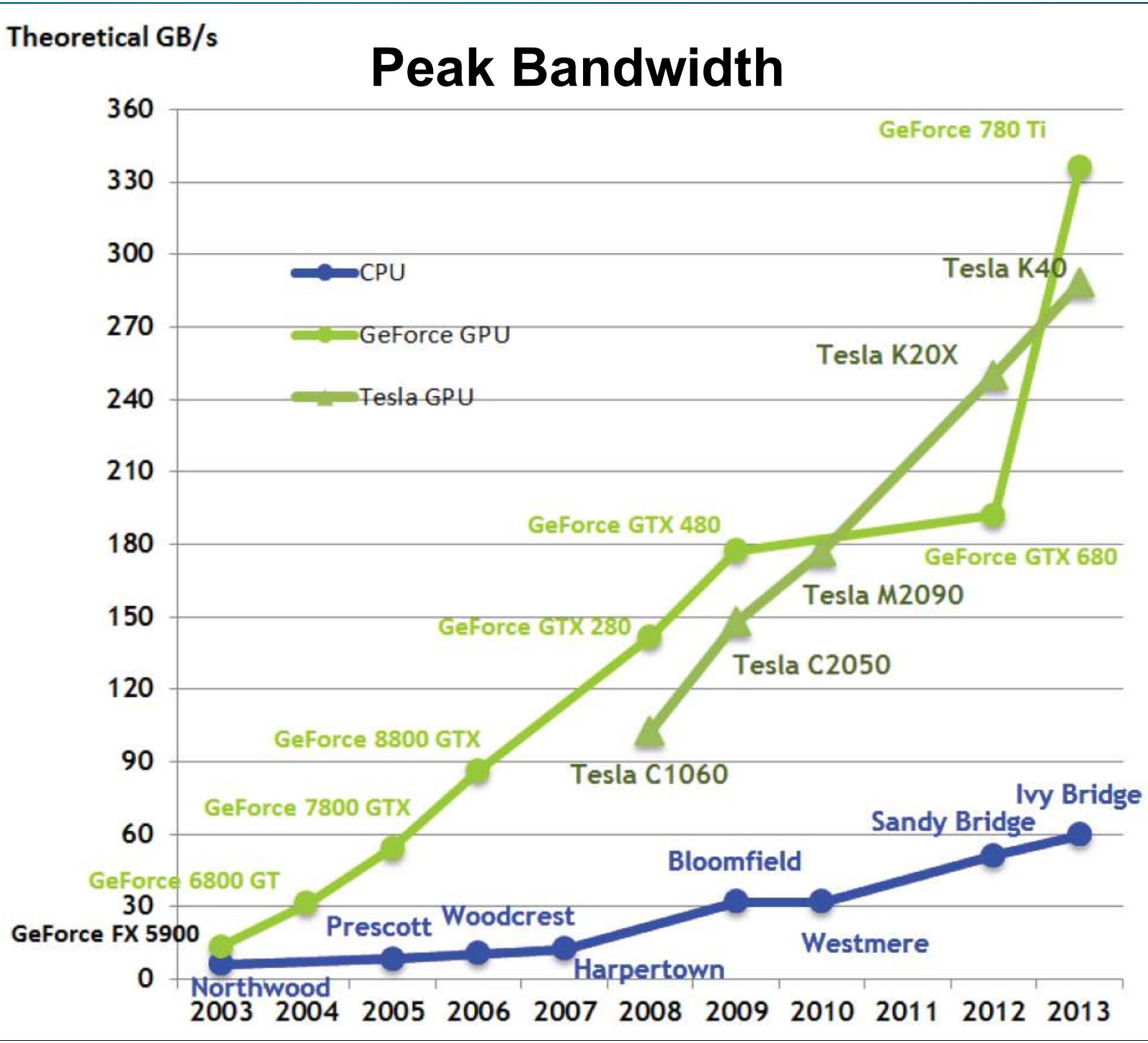


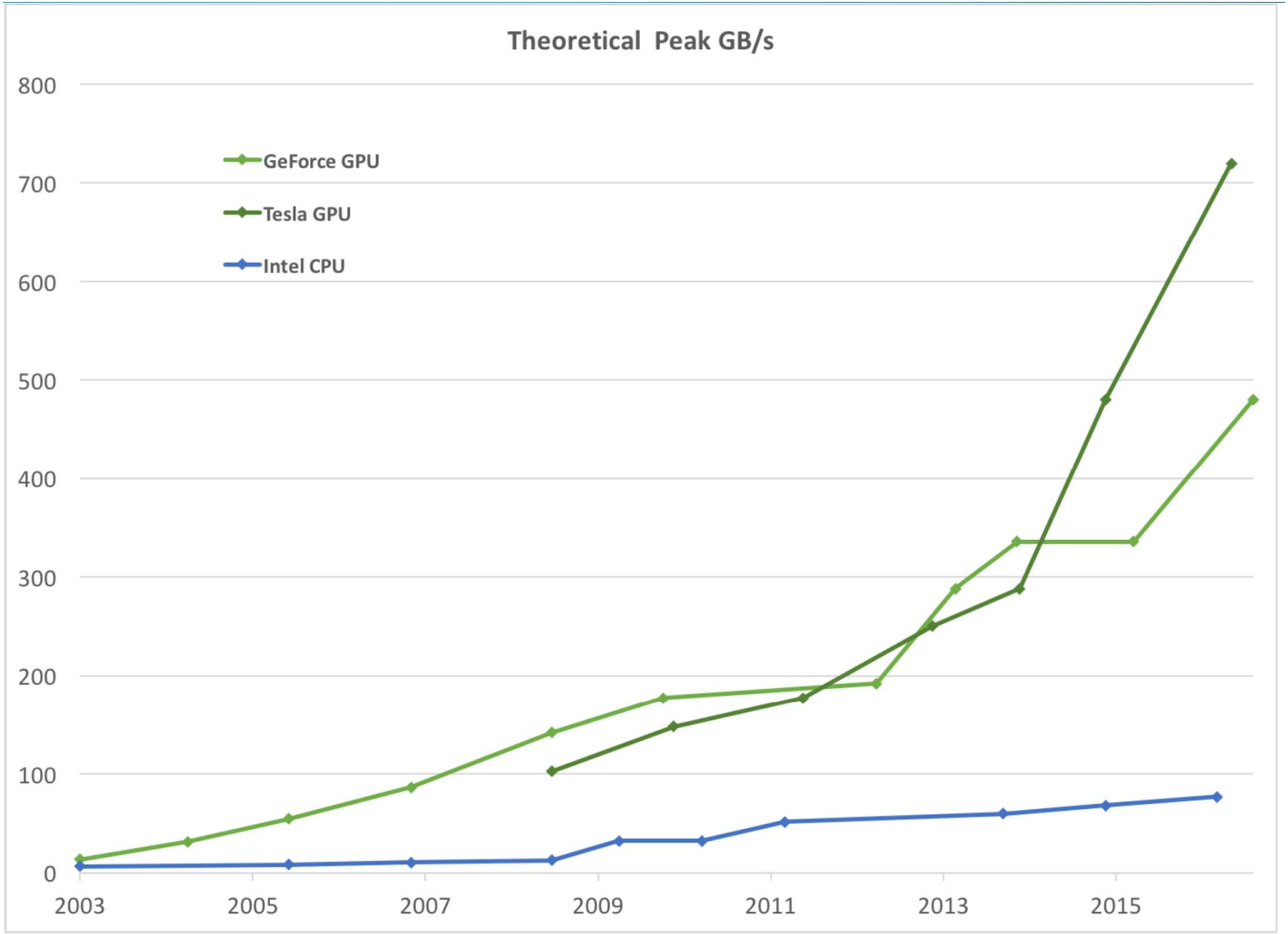
Peak Performance

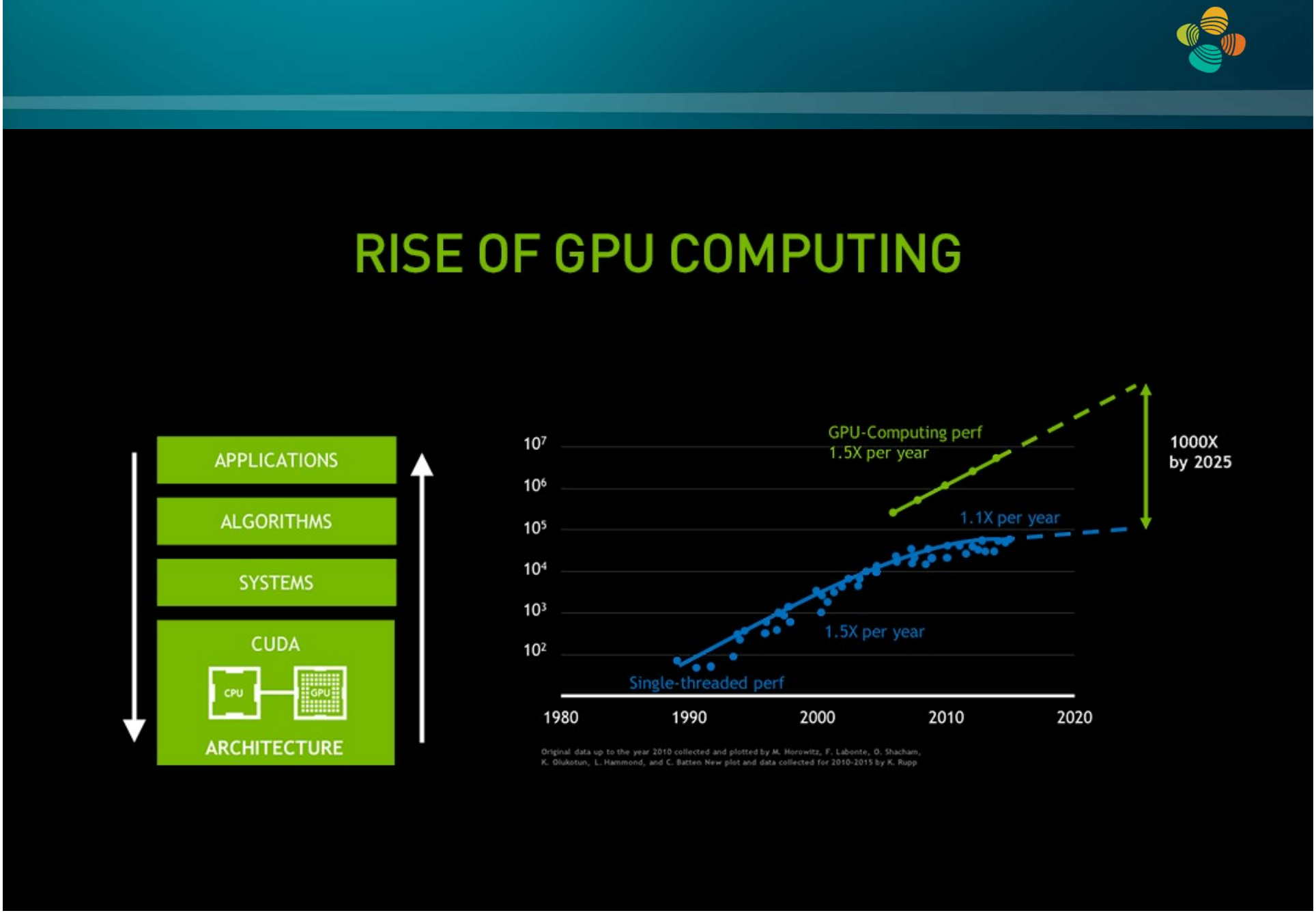


Peak Performance

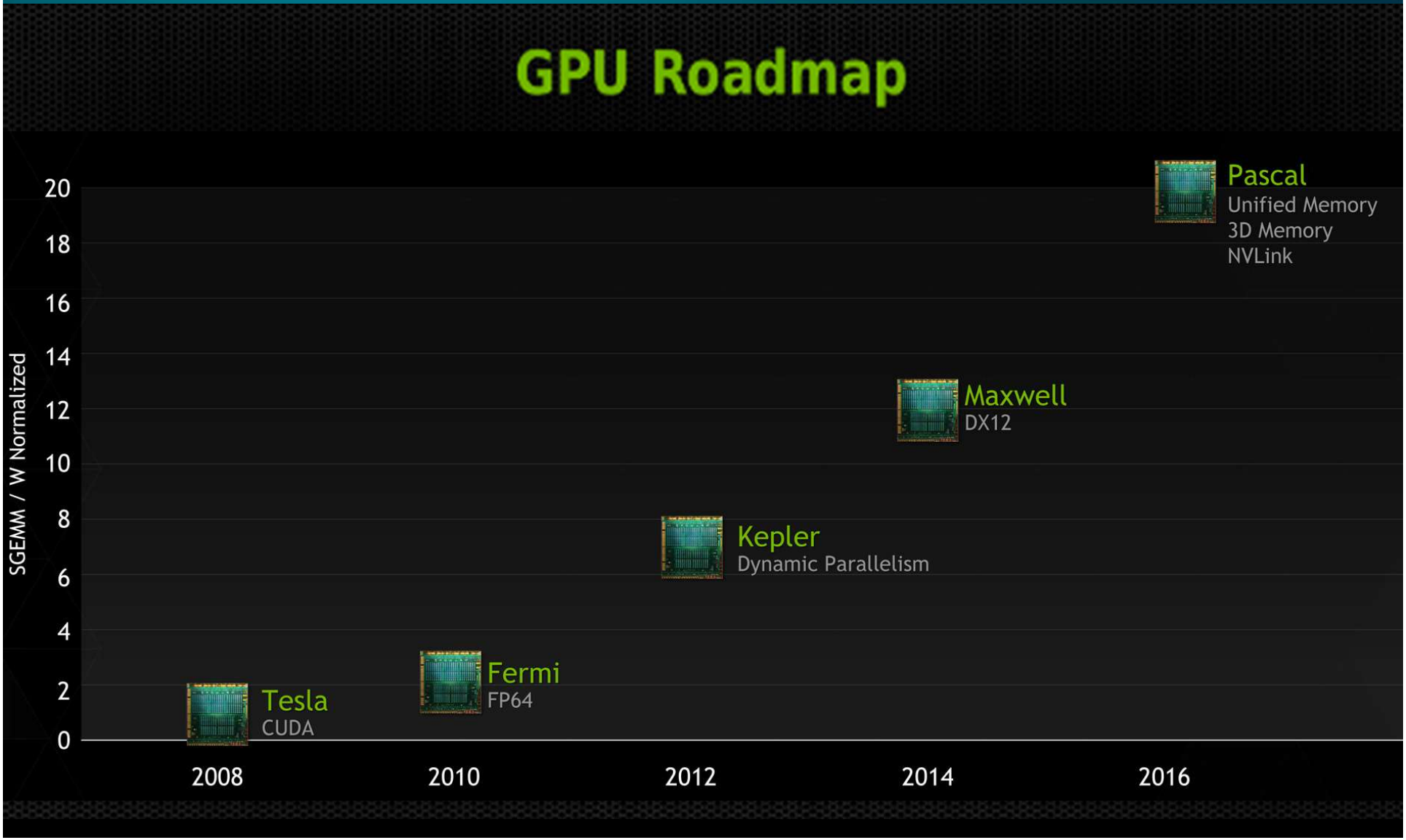



Peak Bandwidth
Theoretical GB/s
360
330
300
270
240
210
180
150
120
90
60
30
0
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
CPU
GeForce GPU
Tesla GPU
GeForce 780 Ti
Tesla K40
Tesla K20X
GeForce GTX 480
GeForce GTX 680
Tesla M2090
GeForce GTX 280
Tesla C2050
GeForce 8800 GTX
Tesla C1060
GeForce 7800 GTX
GeForce 6800 GT
GeForce FX 5900
Northwood
Prescott
Woodcrest
Harpertown
Bloomfield
Westmere
Sandy Bridge
Ivy Bridge

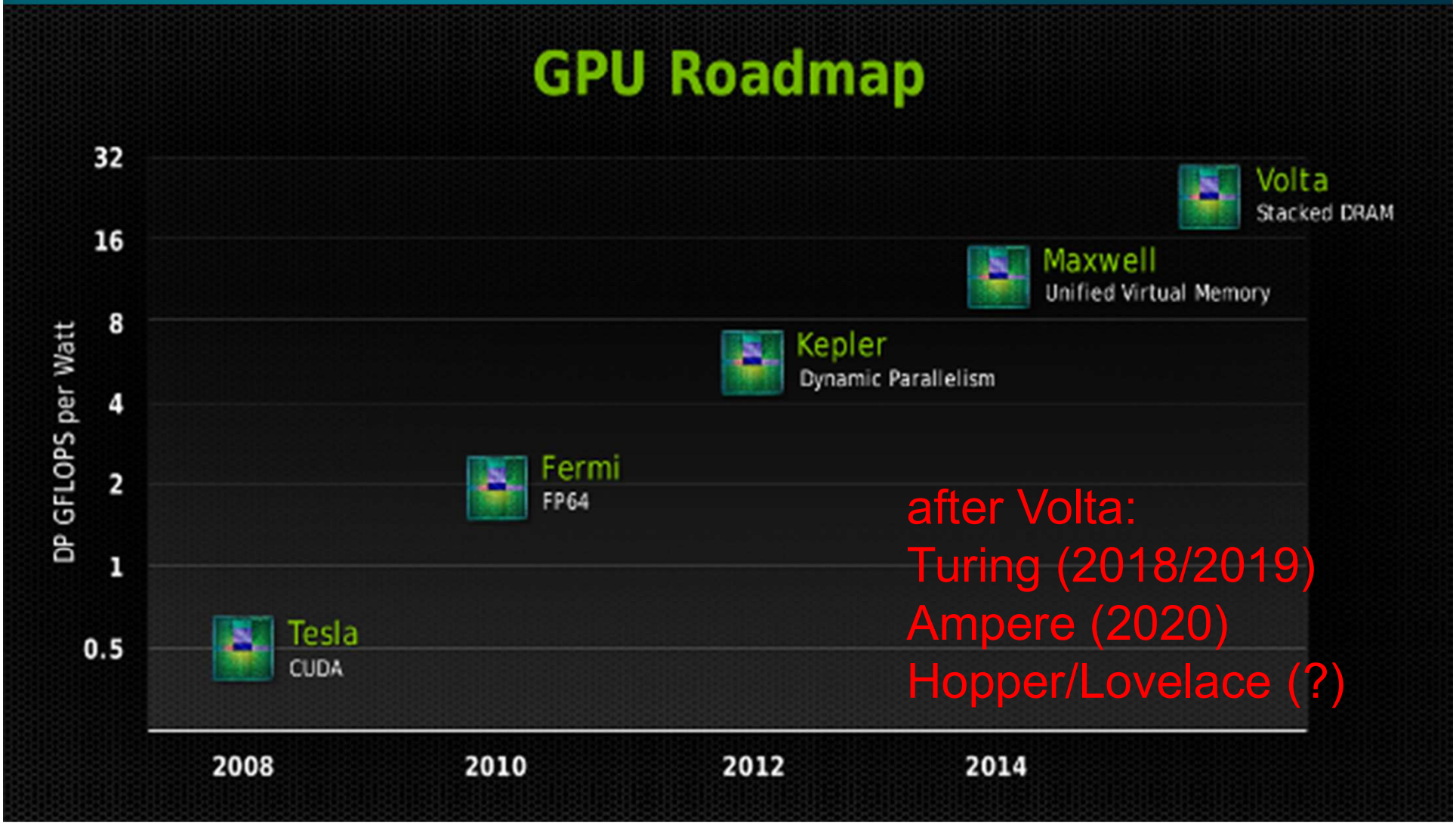

Theoretical Peak GB/s
800
700
600
500
400
300
200
100
0
GeForce GPU
Tesla GPU
Intel CPU
2003
2005
2007
2009
2011
2013
2015


RISE OF GPU COMPUTING
APPLICATIONS
ALGORITHMS
SYSTEMS
CUDA
CPU
GPU
ARCHITECTURE
GPU-Computing perf
1.5X per year
1000X
by 2025
1.1X per year
1.5X per year
Single-threaded perf
10^7
10^6
10^5
10^4
10^3
10^2
1980
1990
2000
2010
2020
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 Architectures Over the Years



Recent Updates

NVIDIA Ampere architecture (2020)

https://en.wikipedia.org/wiki/Ampere_(microarchitecture)

Promo presentation from Sep 1, 2020:

https://www.nvidia.com/en-us/geforce/special-event/

Geforce 30-series (Ampere):

https://nvidia.com/en-us/geforce/graphics-cards/30-series/

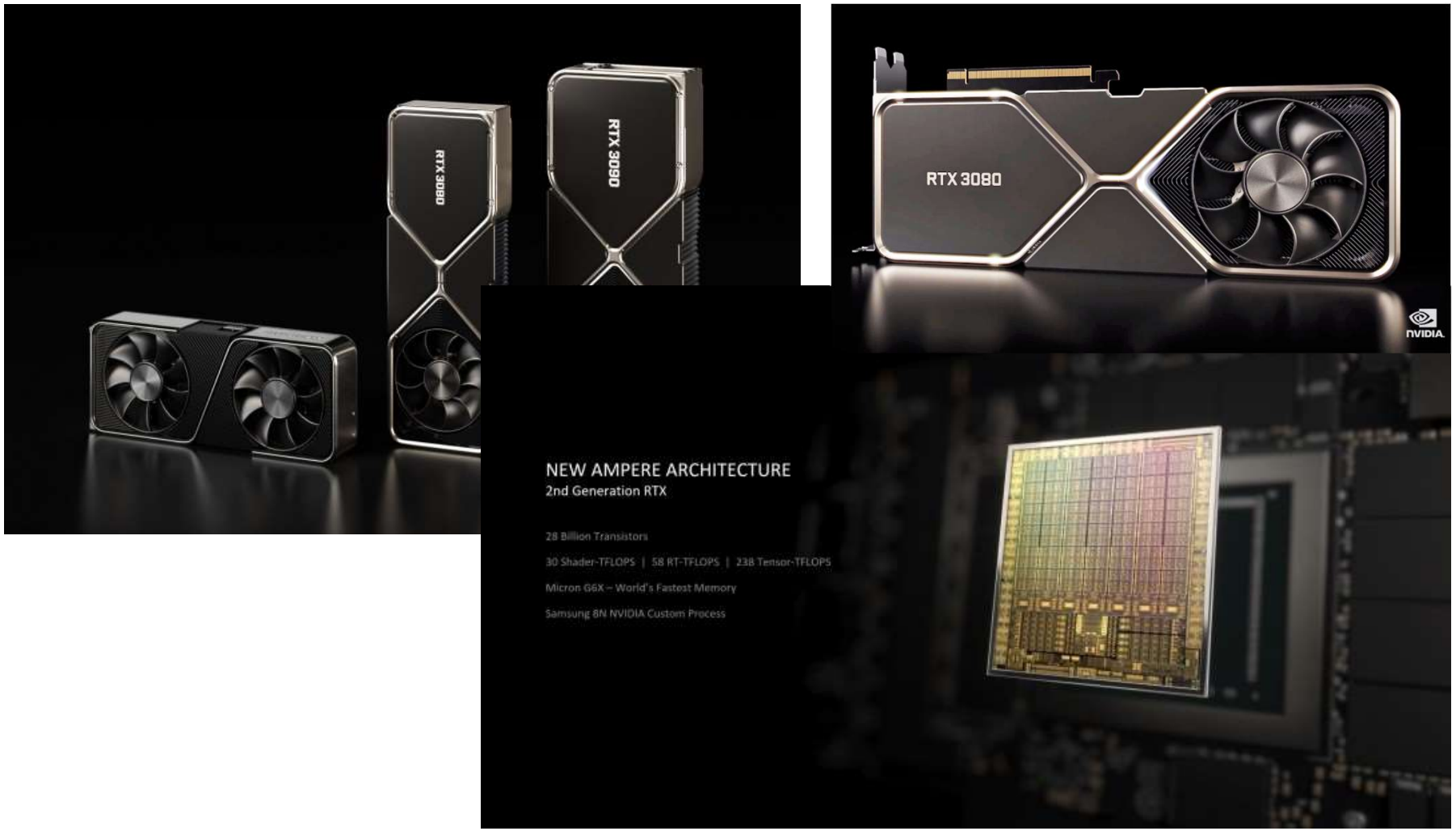
RTX 3090 has 10,496 CUDA cores

A100 (Ampere):

https://www.nvidia.com/en-us/data-center/a100/

A100 has 6,912 CUDA cores

Recent Updates



Overviews and Specs

Wikipedia has many comprehensive lists of architectures and specs

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

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

What is in a GPU?

Lots of floating point processing power

- Stream processing cores

  different names:
  stream processors,
  CUDA cores, ...

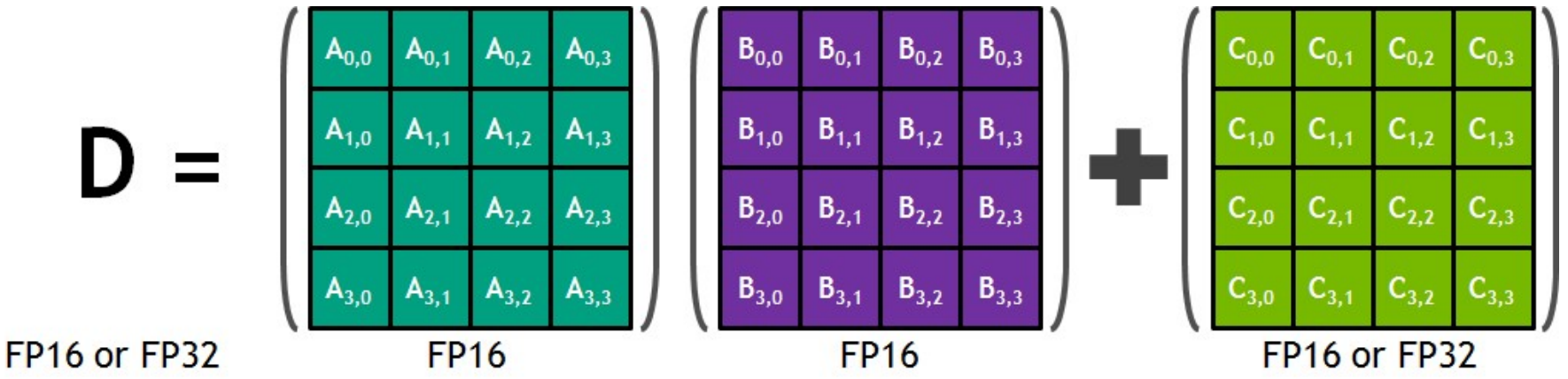
- Was vector processing, now scalar cores!

Still lots of fixed graphics functionality

- Attribute interpolation (per-vertex -> per-fragment)
- Rasterization (turning triangles into fragments/pixels)
- Texture sampling and filtering
- Depth buffering (per-pixel visibility)
- Blending/compositing (semi-transparent geometry, ...)
- Frame buffers

Example for “Special Cores”: Tensor Cores

Mixed-precision, fast matrix-matrix multiply and accumulate



From this, build larger sizes, higher dimensionalities, ...

NVIDIA Volta SM

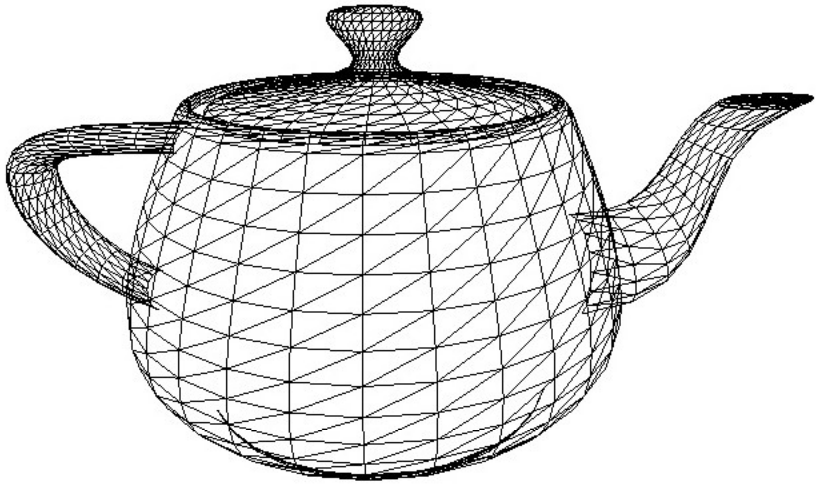
Multiprocessor: SM

- 64 FP32 + INT32 cores
- 32 FP64 cores
- 8 tensor cores (FP16/FP32 mixed-precision)

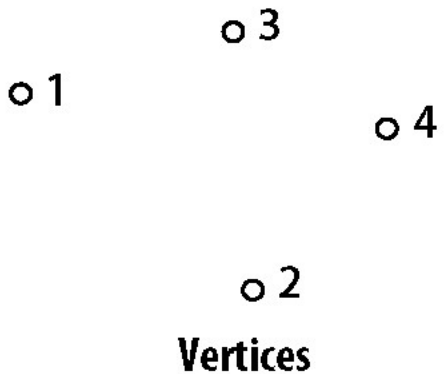
4 partitions inside SM

- 16 FP32 + INT32 cores each
- 8 FP64 cores each
- 8 LD/ST units each
- 2 tensor cores each
- Each has: warp scheduler, dispatch unit, register file

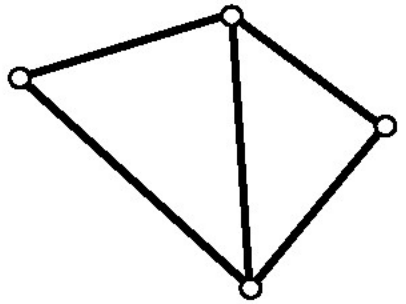
Real-time graphics primitives (entities)



Represent surface as a 3D triangle mesh



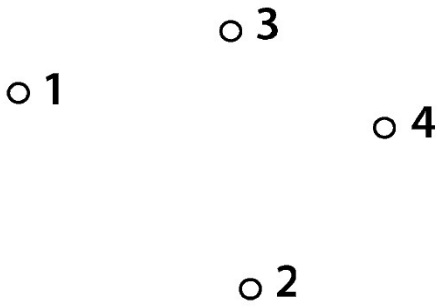
Vertices



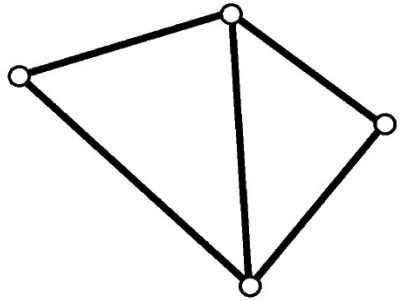
Primitives
(e.g., triangles, points, lines)

Courtesy Kayvon Fatahalian, CMU

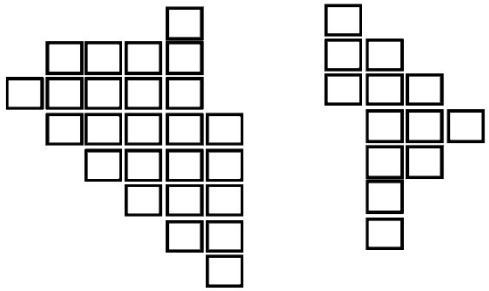
Real-time graphics primitives (entities)



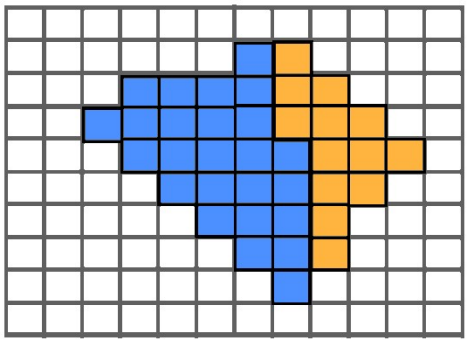
Vertices



Primitives
(e.g., triangles, points, lines)



Fragments

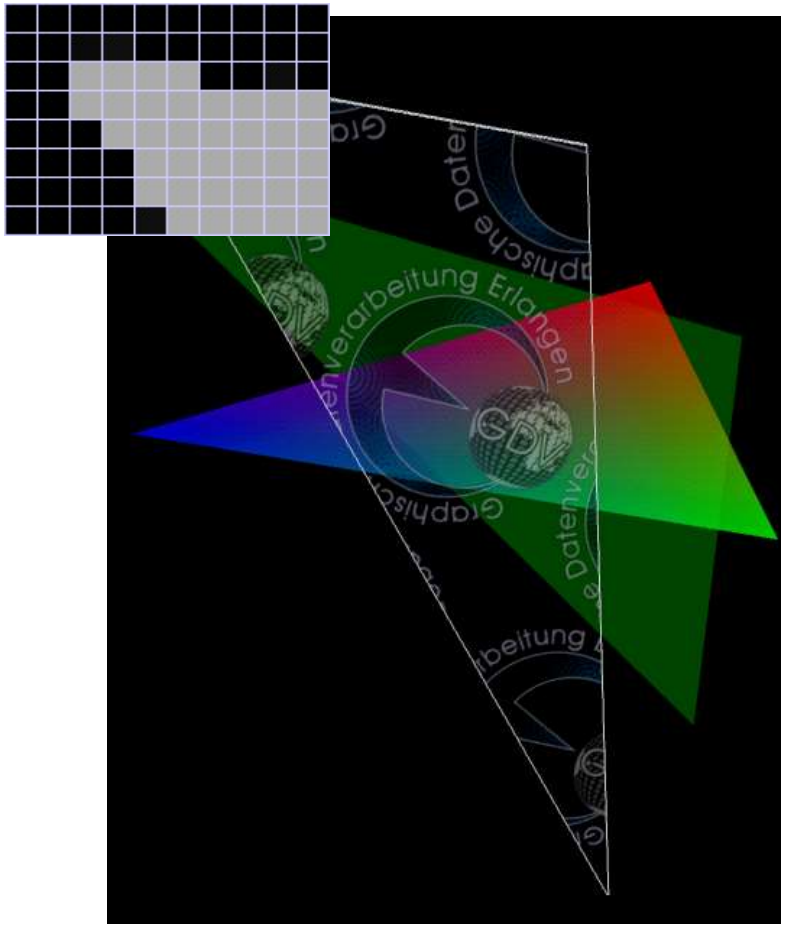


Pixels (in an image)

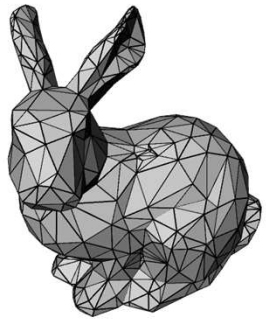
Courtesy Kayvon Fatahalian, CMU

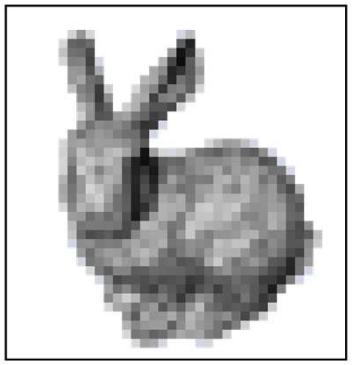
What can the hardware do?

- **Rasterization**
  - Decomposition into fragments
  - Interpolation of color
  - Texturing
    - Interpolation/Filtering
    - Fragment Shading
- **Fragment Operations**
  - Depth Test (Z-Test)
  - Alpha Blending (Compositing)



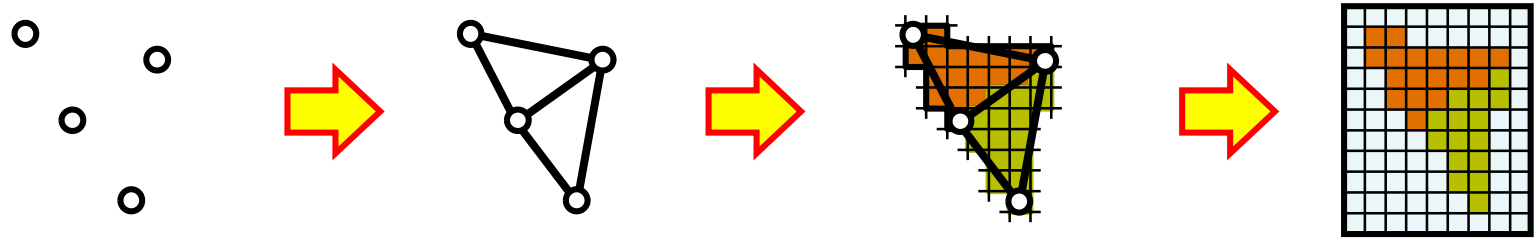
Graphics Pipeline





Scene Description

Raster Image



Vertices

Primitives

Fragments

Pixels

Thank you.