Arithmetic for Accelerators

Stuart Oberman
April 2013

ARITH21

NVIDIA

CPUs, GPUs, Other Accelerators

- CPUs
  - Most well-known programmable processors: they run the OS
  - Typically optimized for low-latency, low-thread count application execution
    - Minimize time/computation, high ratio of mem/computation
  - Intel, AMD, ARM processors, many with FMA
- GPUs
  - Optimized to accelerate high-thread count, highly parallel applications while still holding a day job accelerating graphics applications
    - Maximize computation/time, high ratio of computation/mem
  - High memory bandwidth
  - NVIDIA, AMD, Intel, Qualcomm, Imagination, ARM GPUs, most with FMA
- Other Accelerators
  - Optimized to accelerate high-thread count, parallel applications: may run an OS
  - E.g. Intel Xeon Phi

Where are GPUs and Other Accelerators Used?

From Super Phones to Super Cars

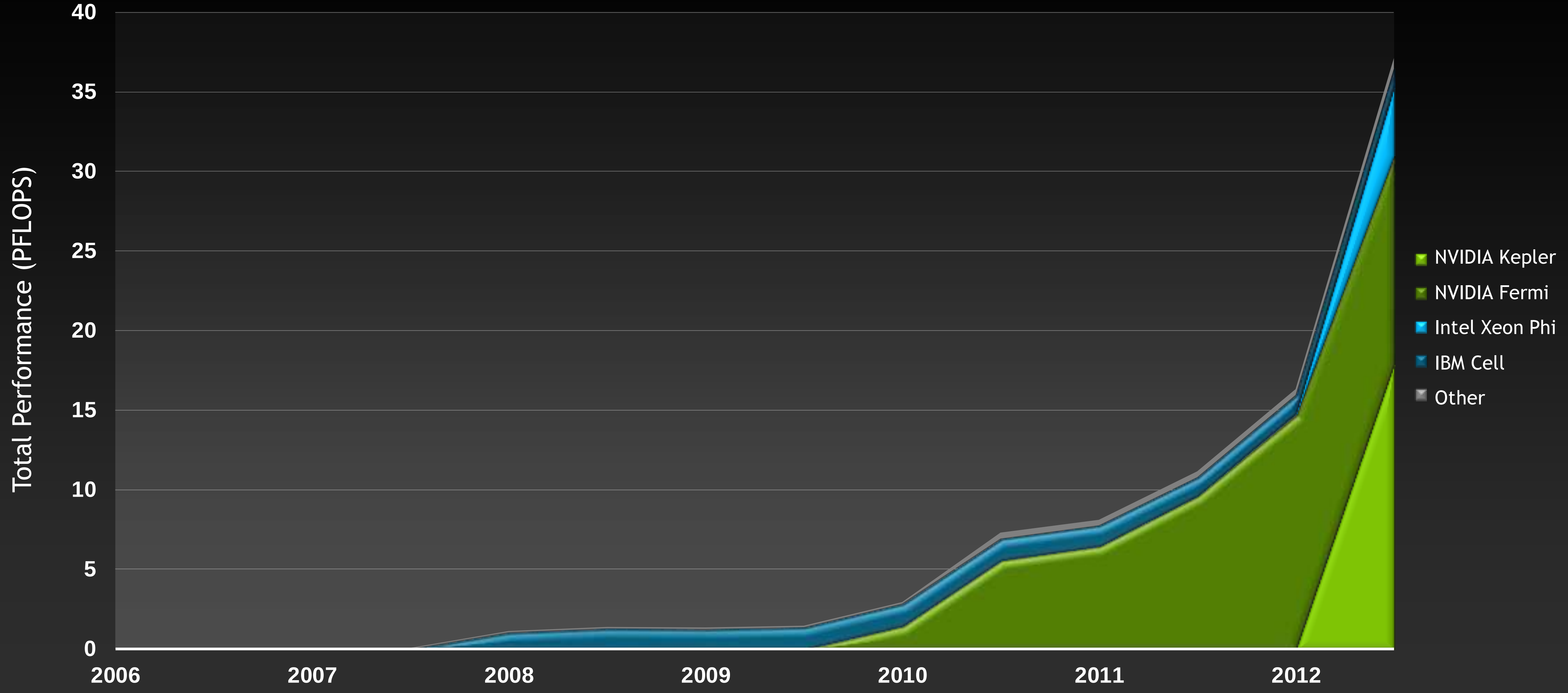
GPUs in Mobile Applications


htc
10
08
Barcelona
12°C
Play Store
Favorites
Start
THE HUNGER GAMES
SUZANNE COLLINS


55
243
Today's Hit Radio
Do You Want It All?
Two Door Cinema Club
Now Playing
Today's Hit Radio
Do You Want It All?
TWO DOOR CINEMA CLUB
70°
73°

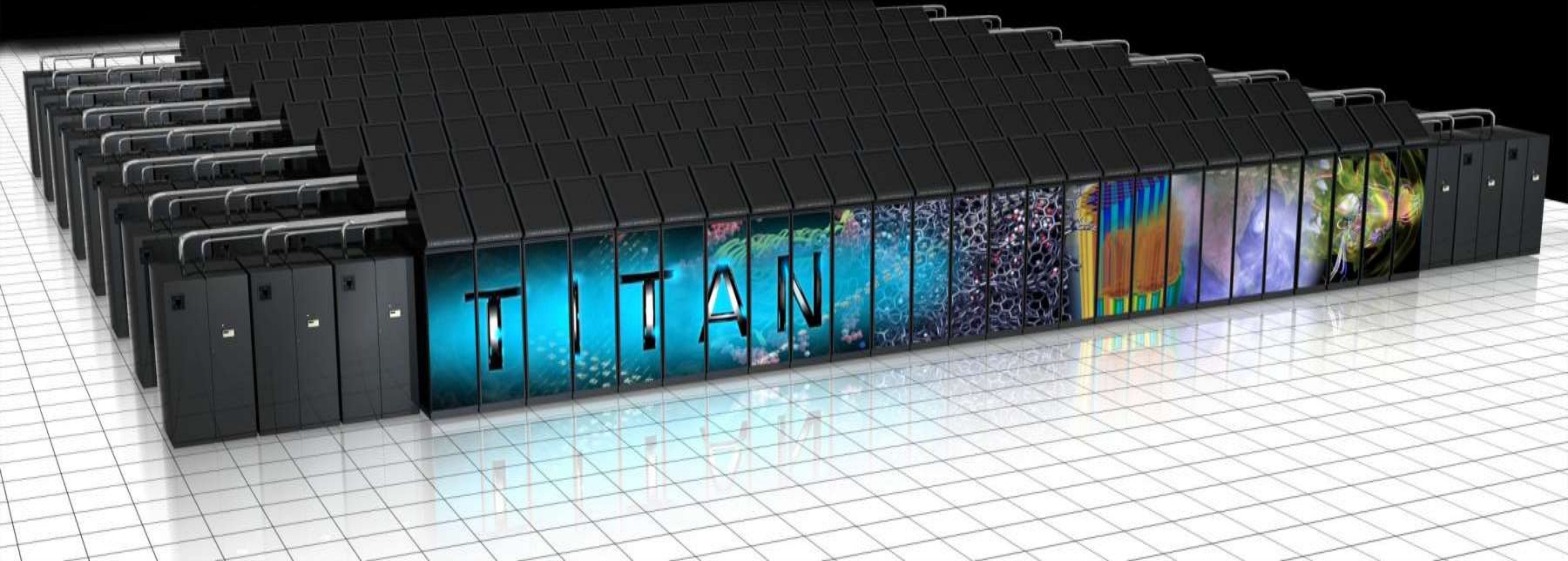
GPUs and Accelerators in High Performance Computing

20% of Flops in Top500 are Powered by GPUs and Other Accelerators

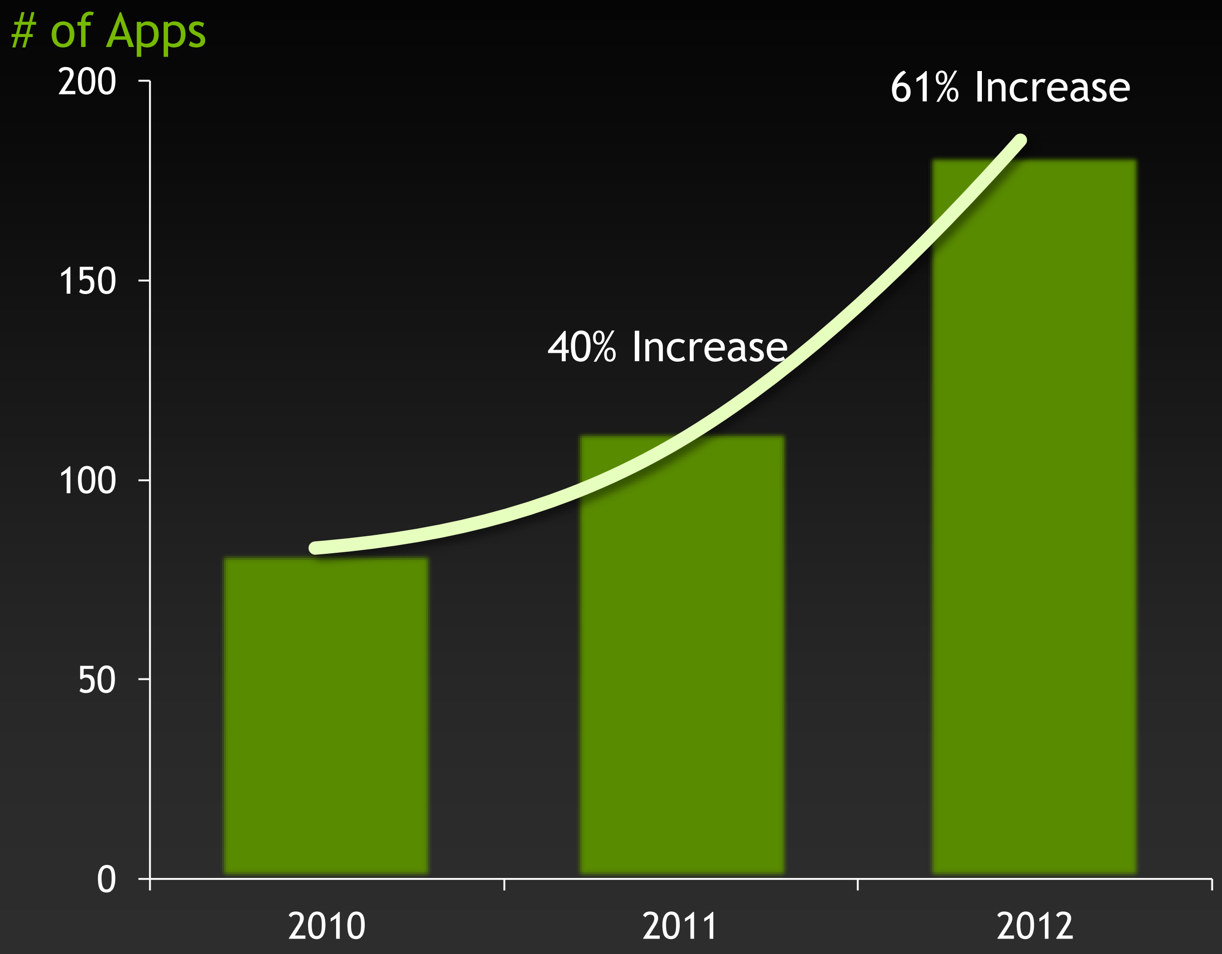


WORLD’S #1 SUPERCOMPUTER

With a peak performance of 27 petaflops, the Titan supercomputer at Oak Ridge National Labs is the world’s fastest. 18,688 NVIDIA Tesla GPUs provide 90% of the machine’s computing power.



Explosive Growth of GPU Accelerated Apps



Top Scientific Apps

Computational Chemistry	AMBER CHARMM GROMACS	LAMMPS NAMD DL_POLY
Material Science	QMCPACK Quantum Espresso GAMESS-US	Gaussian NWChem VASP
Climate & Weather	COSMO GEOS-5	CAM-SE NIM WRF
Physics	Chroma Denovo GTC	GTS ENZO MILC
CAE	ANSYS Mechanical MSC Nastran SIMULIA Abaqus	ANSYS Fluent OpenFOAM LS-DYNA

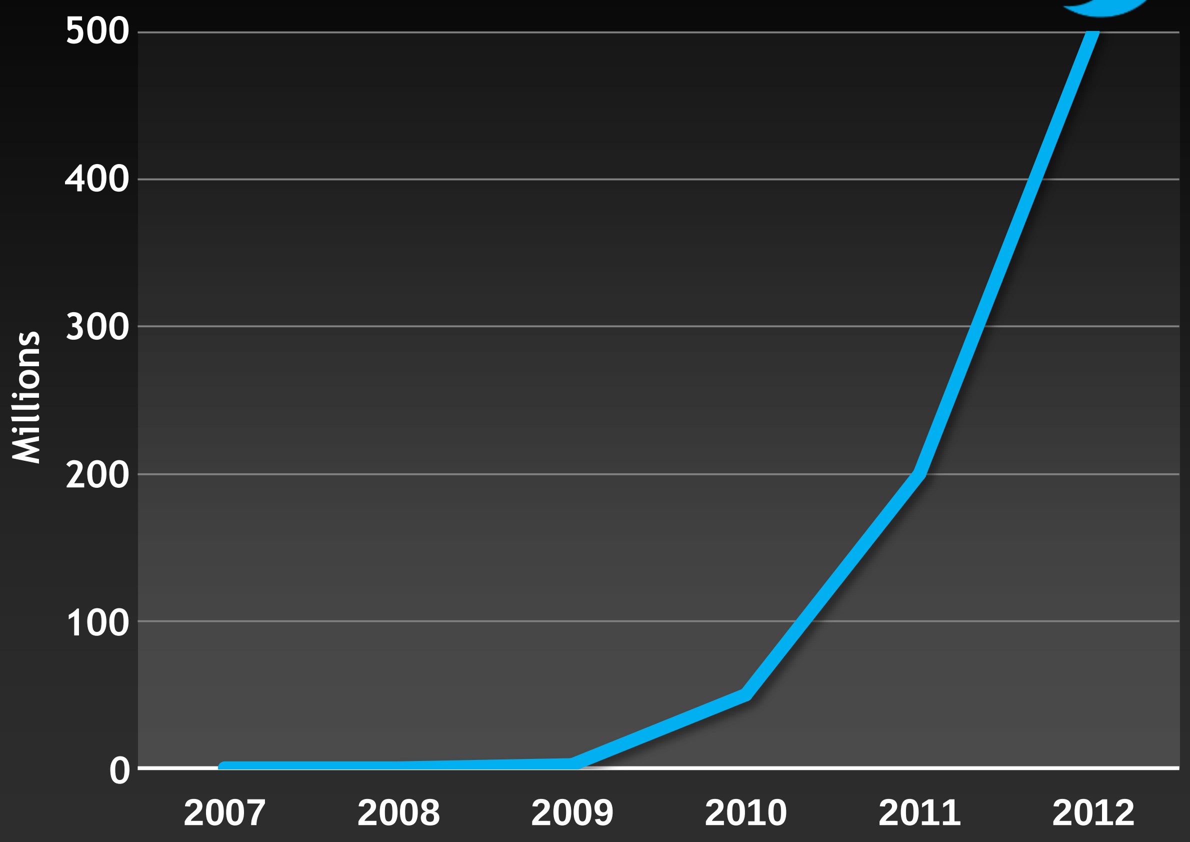
Accelerated, In Development

GPU Accelerators For Big Data Analytics

Analyzing Twitter	Searching Audio	Image-based Search	Real-time Video Delivery
	Shazam		

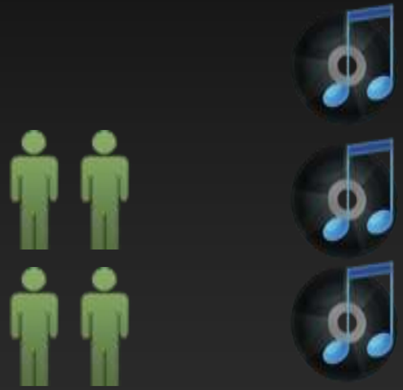
SalesForce.com: Analyzing Twitter Real-Time

500 Million Tweets per Day



CPU	GPU
10 Min. per Text Search	Real-Time Text Search

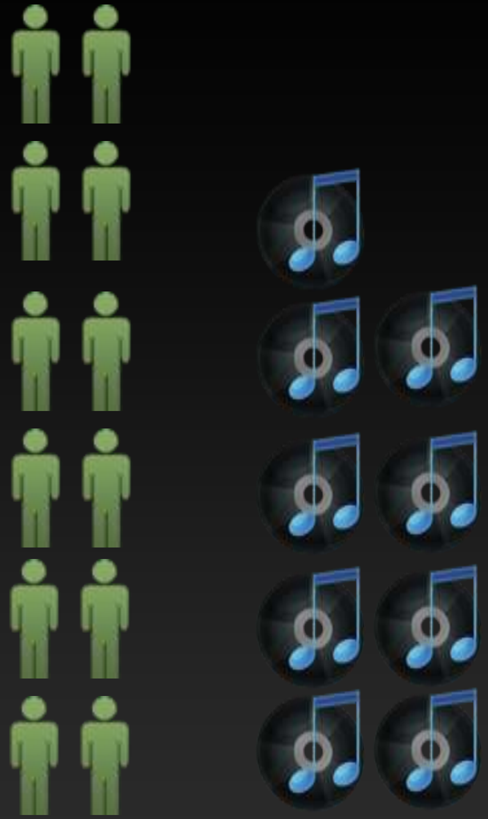
Shazam: 300M GPU Accelerated Searches



User Inquiries
100M

# of Records
10M

2011



User Inquiries
300M

# of Records
27M

2012

Hundreds
of GPUs in Datacenter

GPUs Enable Scalable Growth

User Inquiries averaged per Month

NVIDIA GPUs

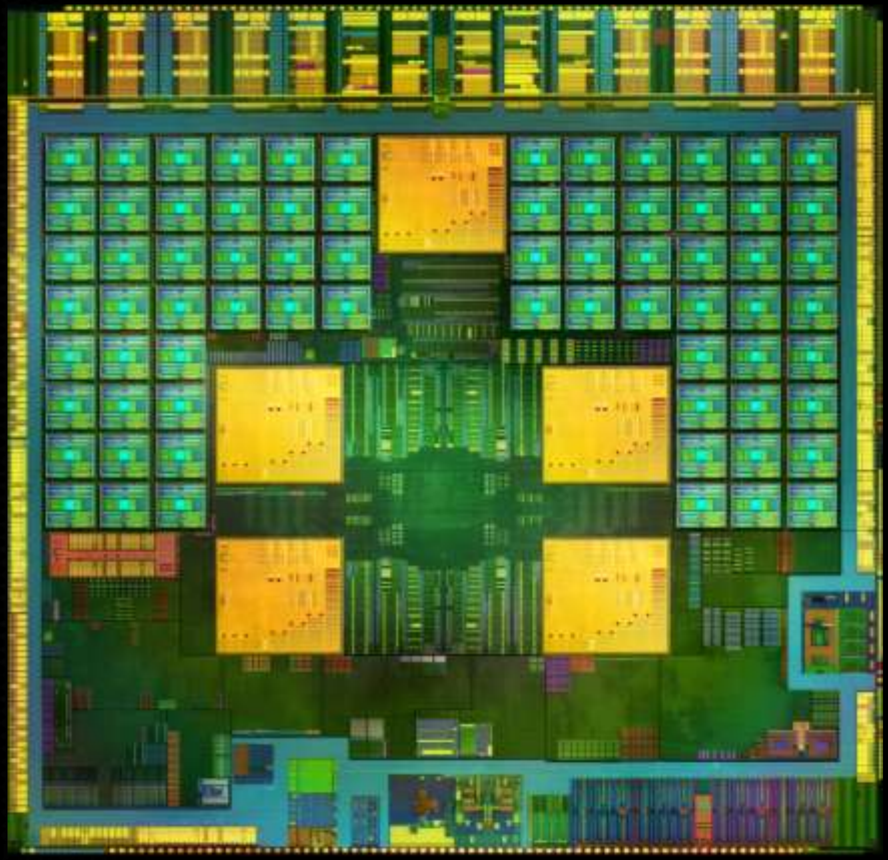
NVIDIA Tegra 4

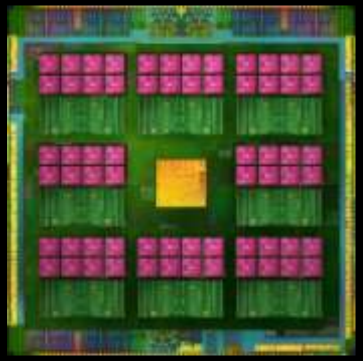
72
GPU Cores

4+1
A15
CPU Cores

4G LTE
Modem
Processor

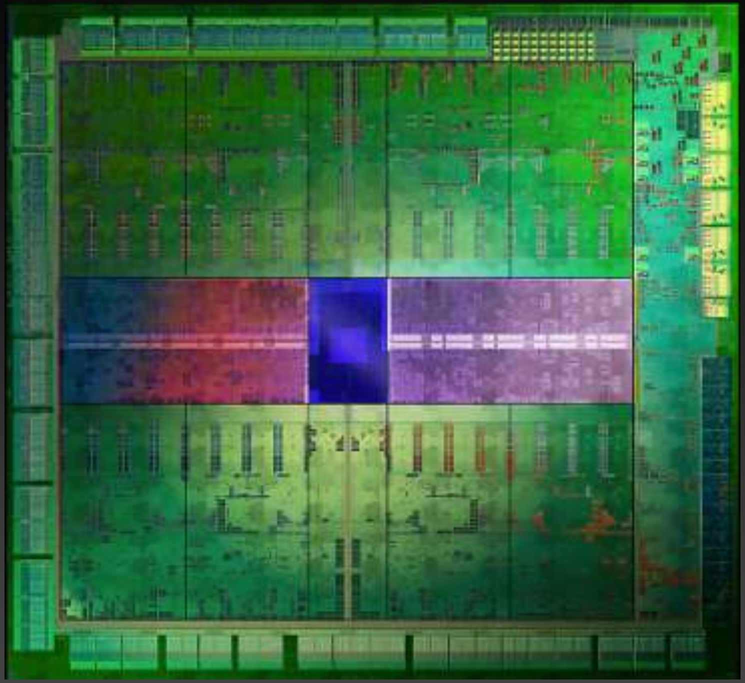
Mobile Processor





FP MAD throughput: 97 GFLOPS
fp20 and fp32

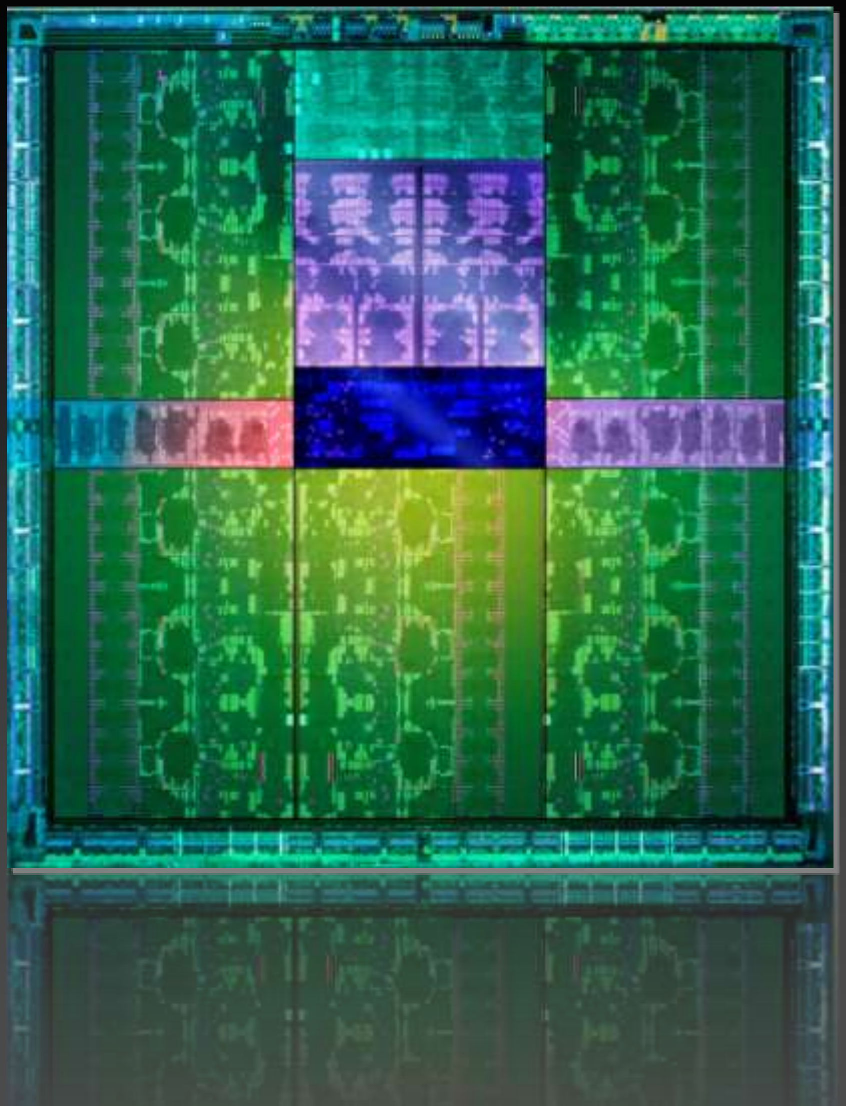
GPU area: 10.5mm2 in 28nm



NVIDIA GK104
Tesla K10
HPC GPU ACCELERATOR

SP FMA throughput: 2.29 TFLOPS
DP FMA throughput: 95 GFLOPS

3.5 billion transistors
294mm2 in 28nm
TDP 225W (2x GK104)



NVIDIA GK110 Tesla K20X

HPC GPU ACCELERATOR

SP FMA throughput: 3.95 TFLOPS

DP FMA throughput: 1.31 TFLOPS

Key internal and external memories ECC protected

7.1 billion transistors

550mm2 in 28nm

TDP 235W

Challenges for Arithmetic in GPUs and Other Accelerators

- Always striving to deliver higher FP throughput
- Limitation to throughput: Power
  - Performance == Power
  - Mobile and HPC processors are power limited: increase power efficiency!
  - Chipwide solutions: wide and slow, run at Vmin
  - Arithmetic unit specific design techniques to optimize energy/op
    - Maximize GFLOPS/W
- Limitation to throughput: silicon die area
  - Performance == area == $
  - Mobile and HPC applications are often cost limited: increase area efficiency!
  - Arithmetic unit design techniques to optimize mm2/op
    - Maximize GFLOPS/mm2

Tradeoffs for Arithmetic Units in GPUs and Accelerators

- How to optimize arithmetic unit area and power efficiency?
- Latency
  - How sensitive are GPUs and accelerator applications to arithmetic unit latency?
  - What efficiency improvements can be made trading off latency?
  - Are there other costs?
- Frequency
  - If higher operating frequency is not always better, what is the right choice?
  - How to design efficient arithmetic units at good choices of operating frequency?
- Precision
  - Where and how to implement required precision within all of the arithmetic units?
    - FMA, MAD, fp32, fp64, fp16, or other?
    - IEEE 754-2008 Standard compliant? Denorms?