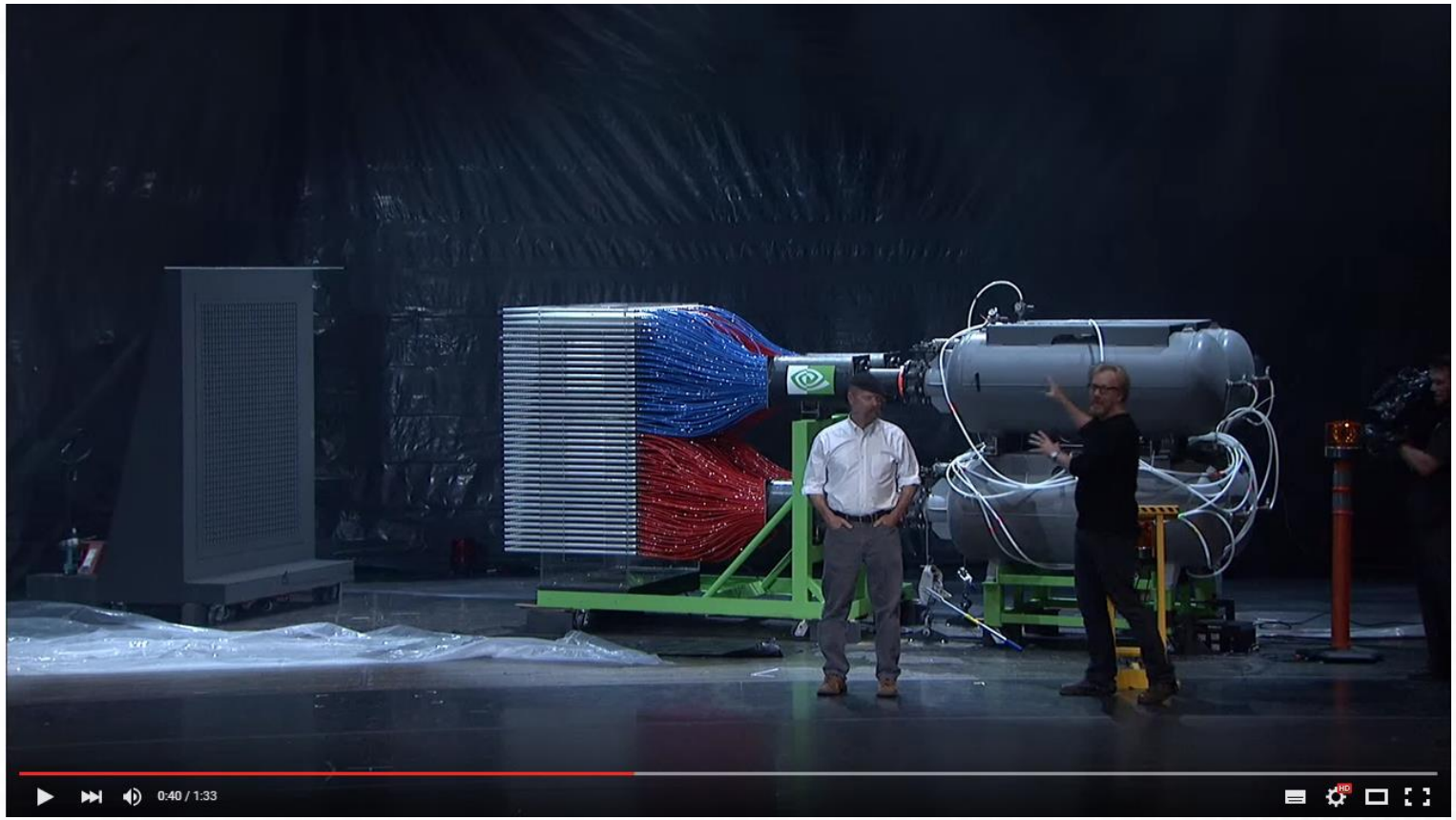
GPU Computing: GPU Architectures

Dr Paul Richmond
http://paulrichmond.shef.ac.uk

- Introduction and Context
- Accelerators (CPUs vs GPUs)
- NVIDIA Hardware Model

GPU Introduction



The University Of Sheffield.

NVIDIA GPU RESEARCH CENTER

CPU Limitations

- Power = Frequency x Voltage²
- Performance Improvements traditionally realised by increasing frequency
  - Voltage decreased to maintain steady power
- Voltage cannot be decreased any further
  - 1's and 0's represented by different voltages
  - Need to be able to distinguish between the two

Moores Law

- Moores Law: A doubling of transistors every couple of years
  - BUT Clock speeds are not increasing
  - Longer more complex pipelines?
- Increase performance by adding parallelism
  - Perform many operations per clock cycle
    - More cores
    - More operations per core
  - Keep power per core low

Accelerators

- Much of the functionality of CPUs is unused for HPC
  - Branch prediction, out of order execution, etc.

- Ideally for HPC we want: **Simple**, **Low Power** and **Highly Parallel** cores

- Problem: Still need operating systems, I/O, scheduling
- Solution: “Hybrid Systems” – CPUs provide management, “Accelerators” (or co-processors) provide compute power.

- Introduction and Context
- Accelerators (CPUs vs GPUs)
- NVIDIA Hardware Model

Designing an Accelerator

- Chip fabrication prohibitively expensive
  - HPC market relatively small
- Graphics Processing Units (GPUs) have evolved from the desire from improved graphical realism in games
  - Significantly different architecture
  - Lots of number crunching cores
  - Highly parallel
- Initially GPUs started to be used for general purpose use (GPGPU)
- NVIDIA and AMD now tailor their architectures for HPC

Latency vs. Throughput

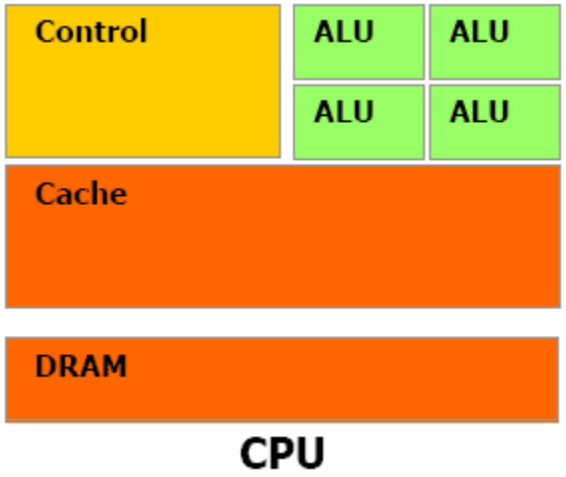
- Latency: The time required to perform some action
  - Measure in units of time
- Throughput: The number of actions executed per unit of time
  - Measured in units of what is produced

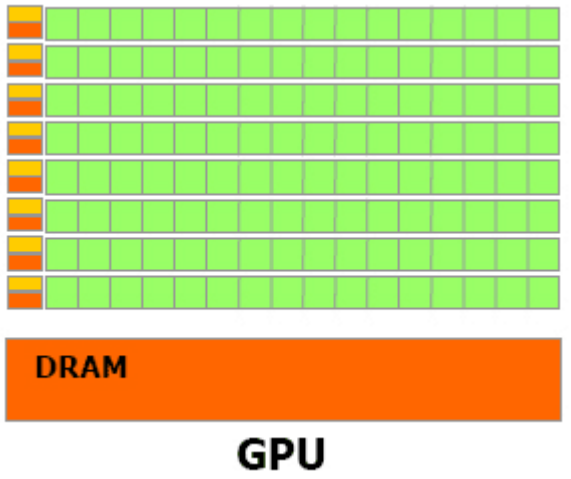
- E.g. An assembly line manufactures GPUs. It takes 6 hours to manufacture a GPU but the assembly line can manufacture 100 GPUs per day.

CPU vs GPU

- CPU
  - Latency oriented
  - Optimised for serial code performance
  - Good for single complex tasks
- GPU
  - Throughput oriented
  - Massively parallel architecture
  - Optimised for performing many similar tasks simultaneously (data parallel)

CPU vs GPU

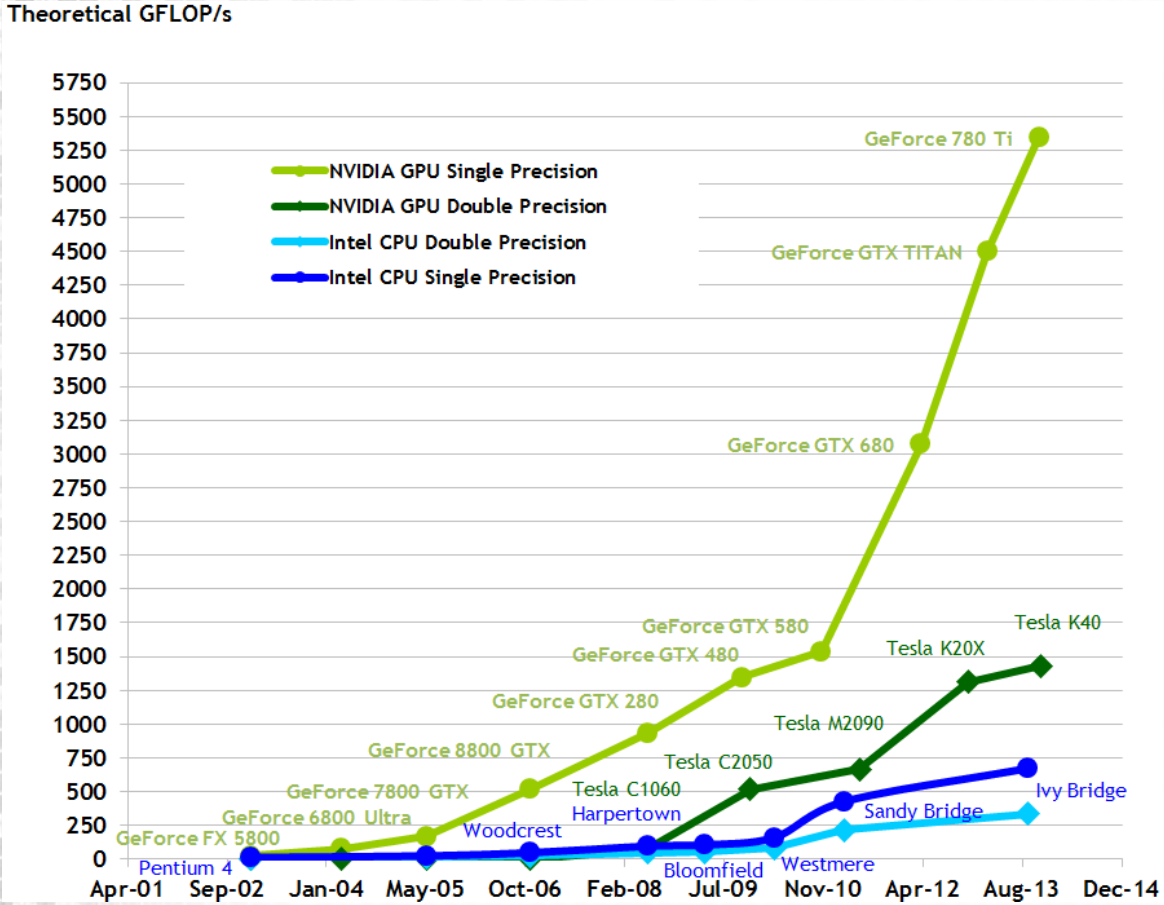


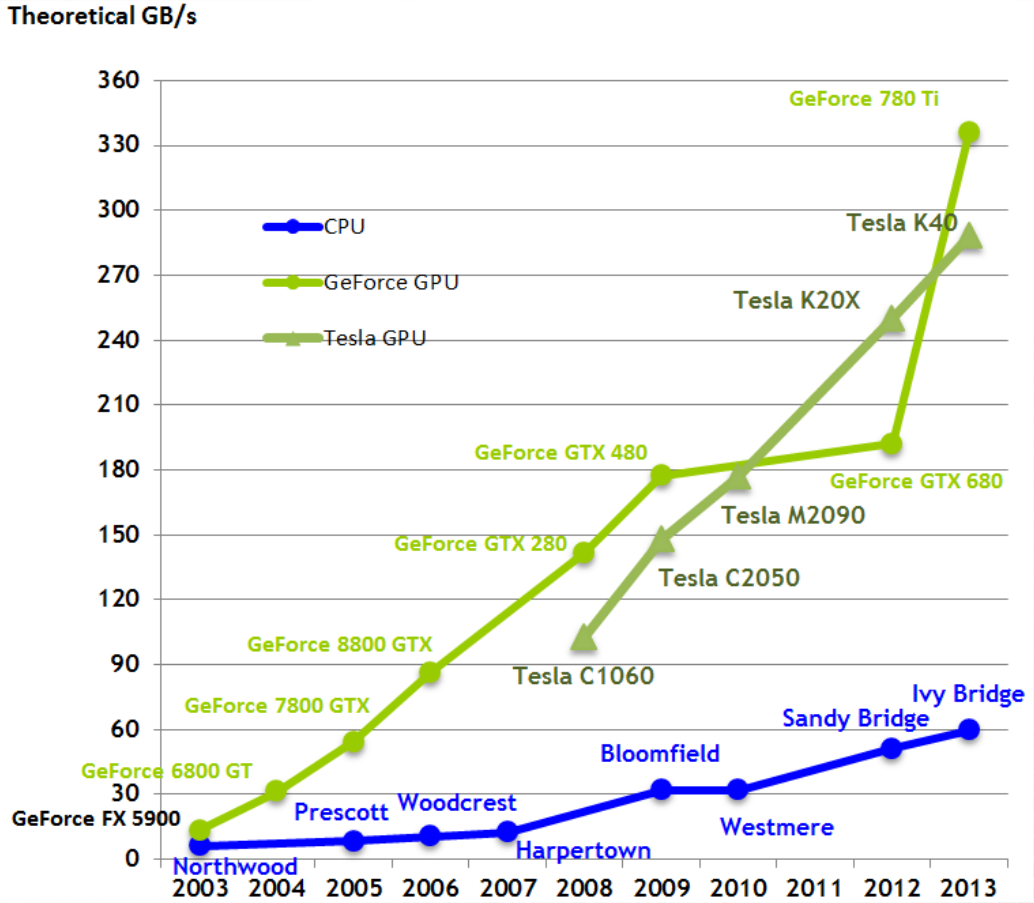


- Large Cache
  - Hide long latency memory access
- Powerful Arithmetic Logical Unit (ALU)
  - Low Operation Latency
- Complex Control mechanisms
  - Branch prediction etc.

- Small cache
  - But faster memory throughput
- Energy efficient ALUs
  - Long latency but high throughput
- Simple control
  - No branch prediction

Performance Characteristics

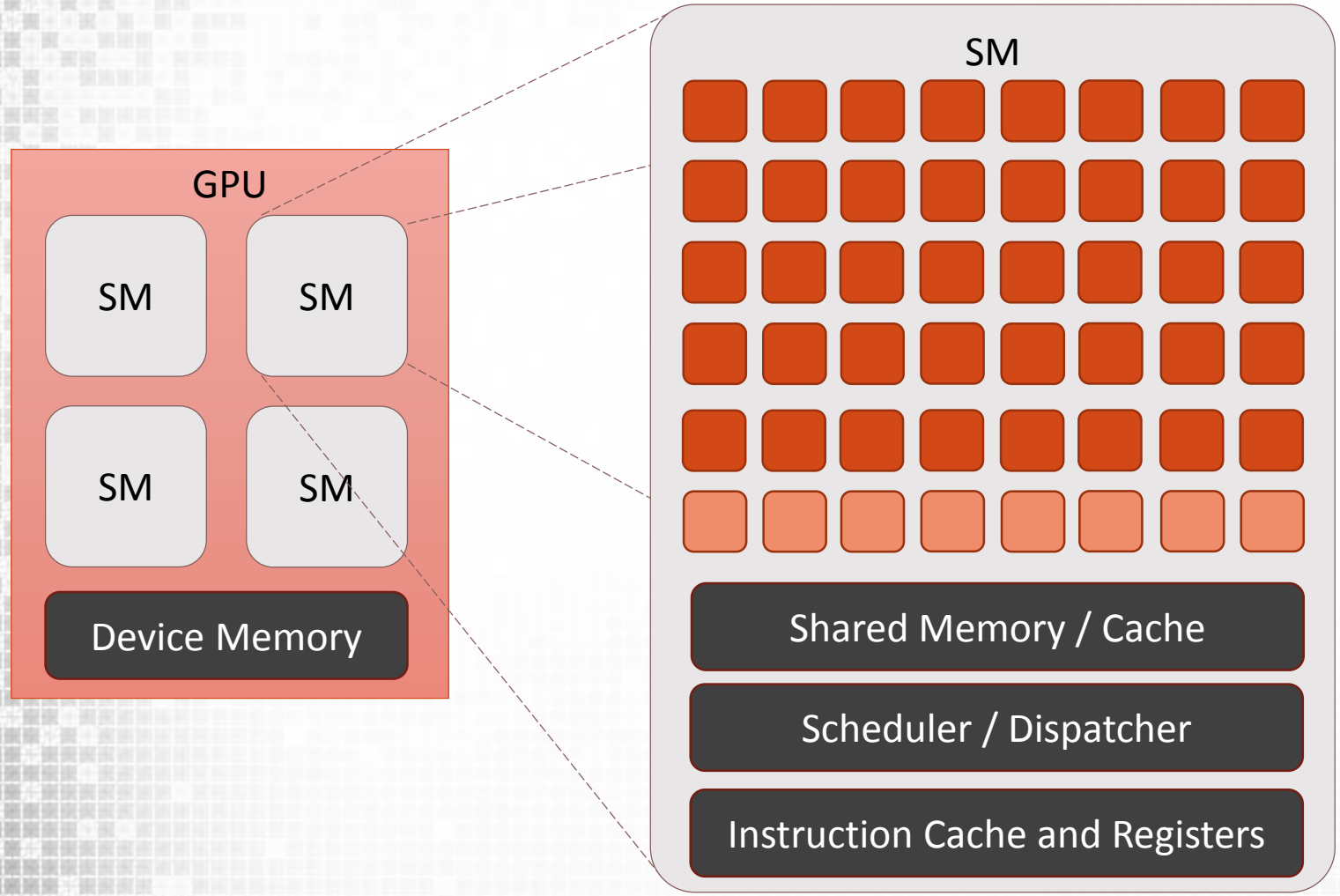




Source: NVIDIA Programming Guide (http://docs.nvidia.com/cuda/cuda-c-programming-guide)

- ❑ Introduction and Context
- ❑ Accelerators (CPUs vs GPUs)
- ❑ NVIDIA Hardware Model

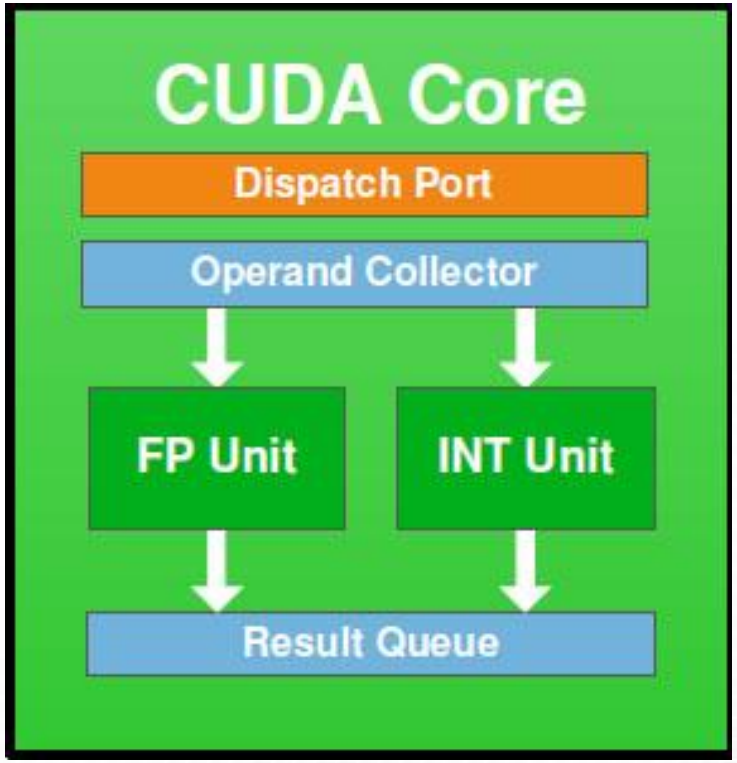
Hardware Model



- NVIDIA GPUs have a 2-level hierarchy
  - Each Streaming Multiprocessor (SMP) has multiple vector cores
  - The number of SMs varies across different hardware implementations
  - The design of SMPs varies between GPU families
  - The number of cores per SMP varies between GPU families

NVIDIA CUDA Core

- CUDA Core
  - Vector processing unit
  - Stream processor
  - Works on a single operation



Tesla Range Specifications

	"Fermi" 2070	"Fermi" 2090	"Kepler" K20	"Kepler" K20X	"Kepler" K40	"Maxwell" M40
CUDA cores	448	512	2496	2688	2880	3072
Chip Variant	GF110	GF110	GK110	GK110	GK110B	GM200
Cores per SM	32	32	192	192	192	128
Single Precision Performance	1.03 Tflops	1.33 Tflops	3.52 Tflops	3.93 Tflops	4.29 Tflops	**7.0 Tflops**
DP Performance	0.51 TFlops	0.66 TFlops	1.17 TFlops	1.31 TFlops	**1.43 Tflops**	0.21 Tflops
Memory Bandwidth	144 GB/s	178 GB/s	208 GB/s	250 GB/s	288 GB/s	288GB/s
Memory	6 GB	6 GB	5 GB	6 GB	12 GB	12GB

Kepler Family of Tesla GPUs

- Streaming Multiprocessor Extreme (SMX)
- Huge increase in the number of cores per SMX
  - Smaller 28nm processes
- Increased L2 Cache

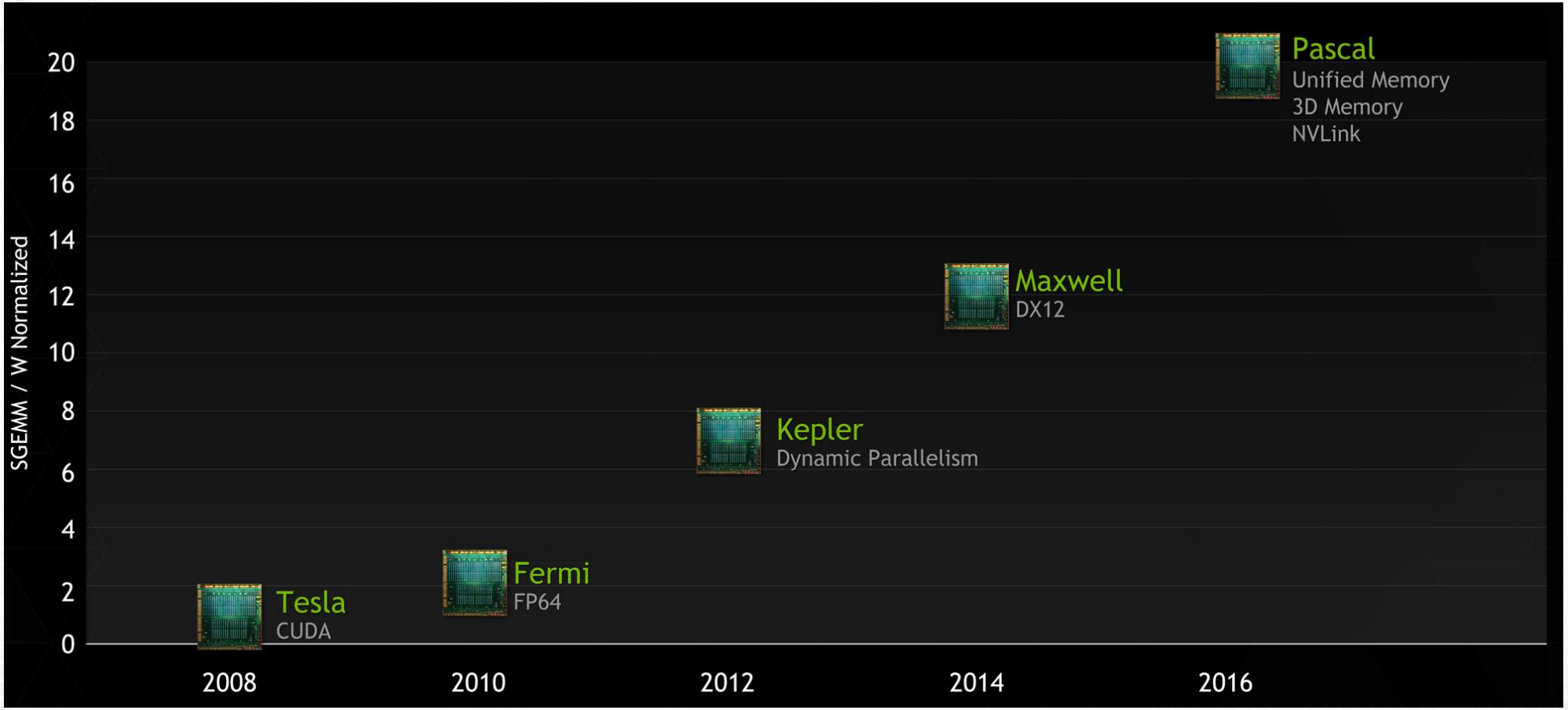
The University Of Sheffield.

NVIDIA GPU RESEARCH CENTER

Maxwell Family Tesla GPUs

- Streaming Multiprocessor Module (SMM)
- SMM Divided into 4 quadrants
  - Each has own instruction buffer, registers and scheduler for each of the 32 vector cores
- SMM has 90% performance of SMX (Kepler) at 2x energy efficiency
  - 128 cores vs. 192 in Kepler
  - BUT small die space = more SMMs
- 8x the L2 cache of Kepler (2MB)

NVIDIA Roadmap



Summary

- GPUs are better suited to parallel tasks than CPUs
- Accelerators are typically not used alone, but work in tandem with CPUs
- GPU hardware is constantly evolving
- GPU accelerated systems scale from simple workstations to large-scale supercomputers
- CUDA is a language for general purpose GPU (NVIDIA only) programming