



Master Informatics Eng.

2018/19

A.J.Proença

Beyond traditional PUs (GPU/CUDA, Tensor Cores, ...) (most slides are borrowed)

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

1

Beyond Vector/SIMD architectures



- Vector/SIMD-extended architectures are hybrid approaches
 - mix (super)scalar + vector op capabilities on a single device
 - **highly pipelined** approach to reduce memory access penalty
 - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
 - **CPU cores with wider vector units**
 - x86 many-core: **Intel** MIC / Xeon KNL
 - IBM Power cores with SIMD extensions: BlueGene/Q Compute
 - other many-core: **ShenWay** 260
 - **coprocessors (require a host scalar processor): accelerator devices**
 - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
 - ISA-free architectures, code compiled to silica: **FPGA**
 - focus on SIMT/SIMD to hide memory latency: **GPU-type** approach
 - **heterogeneous processors (multicore with GPU-cores, SoC)**
 - ...

- Question to GPU architects:
 - Given the hardware invested to do graphics well, how can we supplement it to improve the performance of a wider range of applications?
- Key ideas:
 - Heterogeneous execution model
 - CPU is the *host*, GPU is the *device*
 - Develop a C-like programming language for GPU
 - Unify all forms of GPU parallelism as *CUDA_threads*
 - Programming model follows SIMT: “*Single Instruction Multiple Thread*”

Copyright © 2012, Elsevier Inc. All rights reserved.

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

3

#cores/processing element in several devices

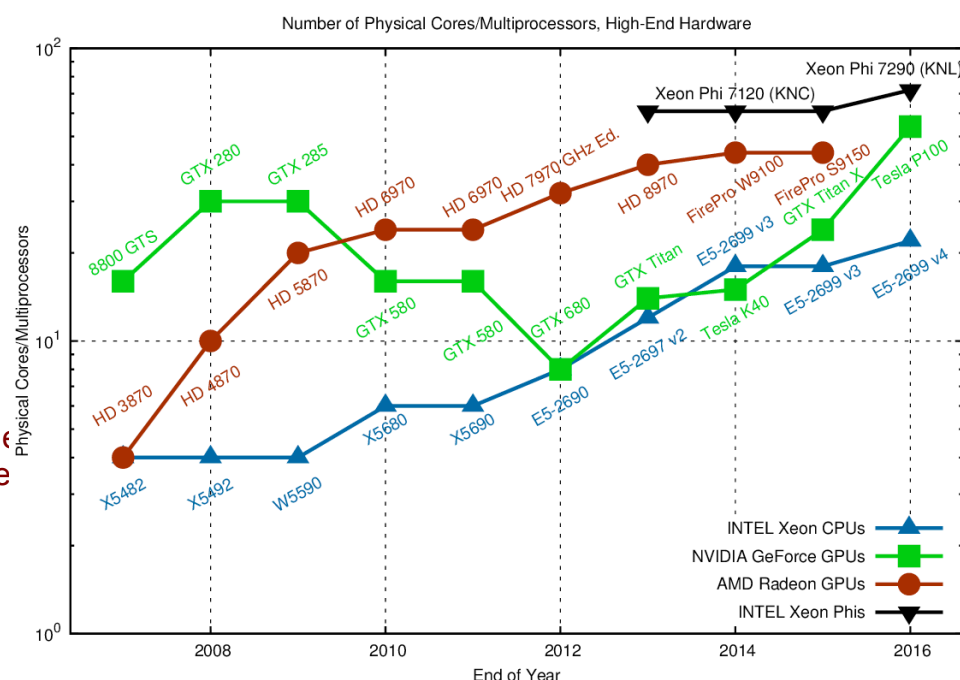
Key question: what is a **core**?

a) IU+FPU?
GPU-type...

b) A SIMD processor?
CPU-type..

This updated slide and in this course - b)

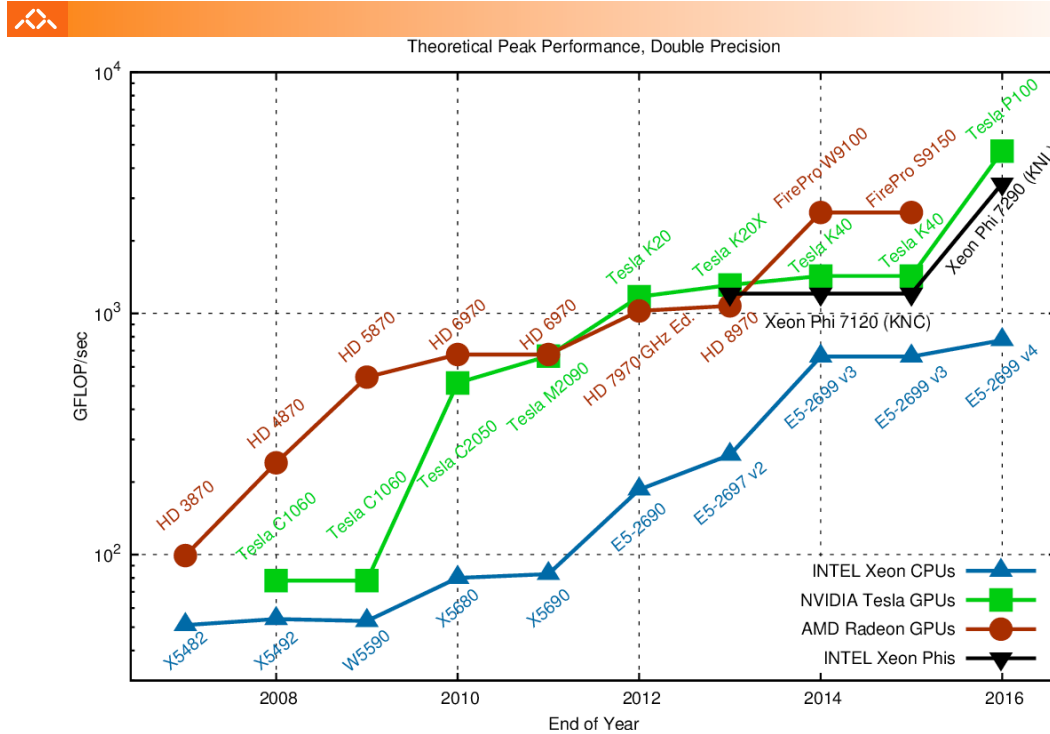
Note: the web link with these plots was updated in Aug'16



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

4

Theoretical peak performance in several computing devices (DP)

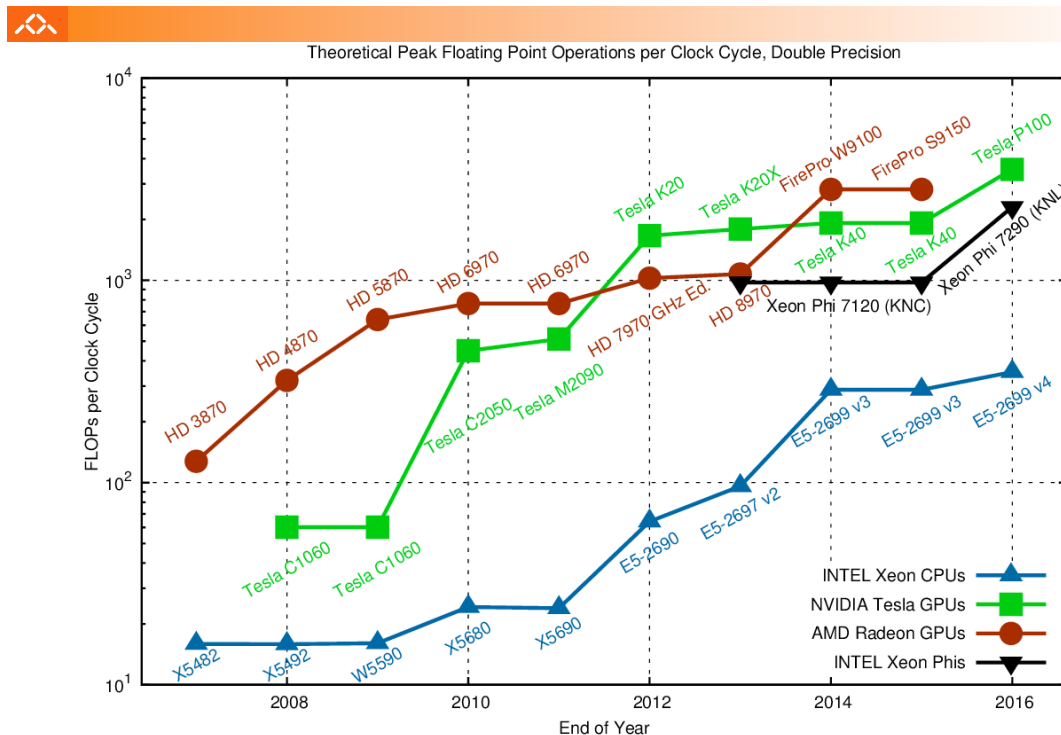


AJProença, Parallel Computing, MiEI, UMinho, 2018/19

5

<http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/>

Theoretical peak FP Op's per clock cycle in several computing devices (DP)



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

6

<http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/>

NVIDIA GPU Architecture

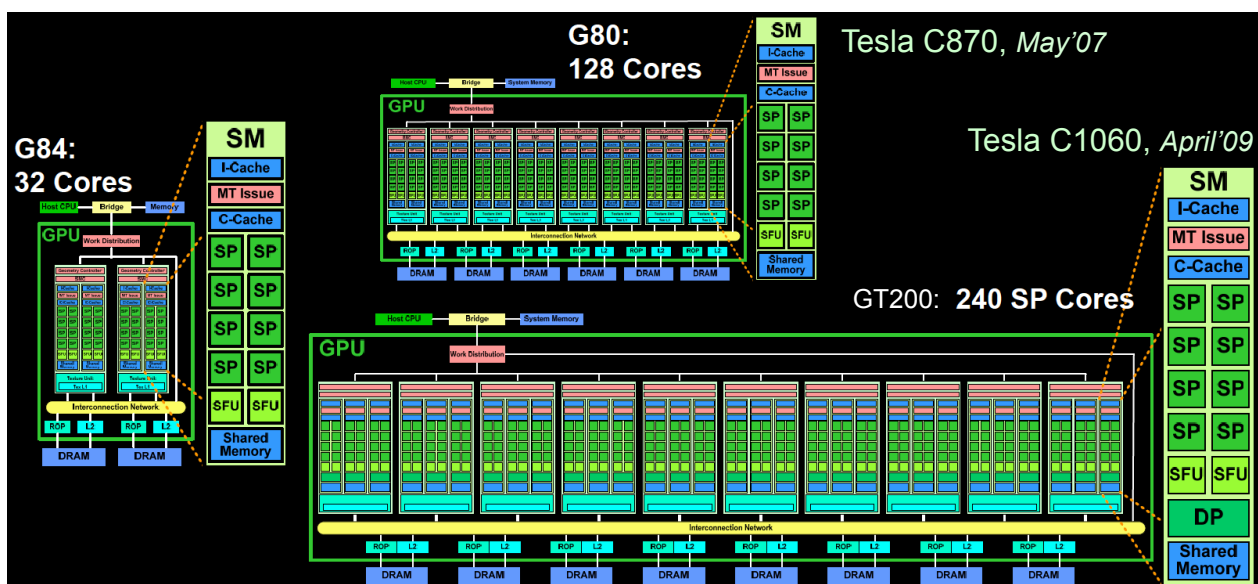
- Similarities to vector machines:
 - Works well with data-level parallel problems
 - Scatter-gather transfers
 - Mask registers
 - Large register files
- Differences:
 - No scalar processor
 - Uses multithreading to hide memory latency
 - Has many functional units, as opposed to a few deeply pipelined units like a vector processor

Copyright © 2012, Elsevier Inc. All rights reserved.

AJProença, *Parallel Computing*, MiEI, UMinho, 2018/19

7

Early NVidia GPU Computing Modules



AJProença, *Parallel Computing*, MiEI, UMinho, 2018/19

8

NVIDIA GPU Memory Structures

- Each SIMD Lane has private section of **off-chip DRAM**
 - “Private memory” (*Local Memory*)
 - Contains stack frame, spilling registers, and private variables
- Each multithreaded SIMD processor (*SM*) also has local memory (*Shared Memory*)
 - Shared by SIMD lanes / threads within a block
- Memory shared by SIMD processors (*SM*) is GPU Memory, *off-chip DRAM* (*Global Memory*)
 - Host can read and write GPU memory

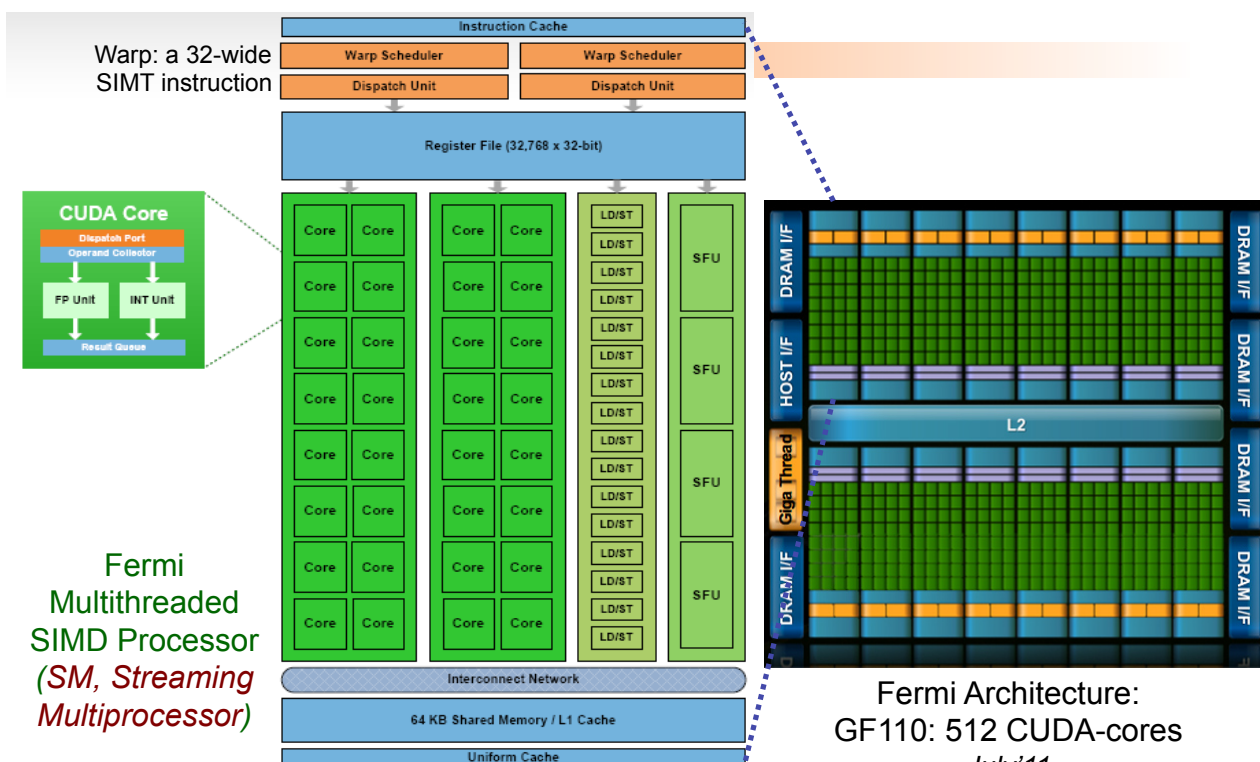


Copyright © 2012, Elsevier Inc. All rights reserved.

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

9

The NVidia Fermi architecture



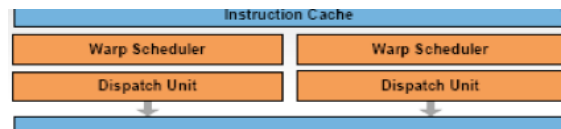
AJProença, Parallel Computing, MiEI, UMinho, 2018/19

10

Fermi Architecture Innovations



- Each SIMD processor has
 - Two SIMD thread schedulers, two instruction dispatch units
 - 16 SIMD lanes (SIMD width=32, chime=2 cycles), 16 load-store units, 4 special function units
 - Thus, two threads of SIMD instructions are scheduled every two clock cycles
- Fast double precision
- Caches for GPU memory (16/64KB_L1/SM and global 768KB_L2)
- 64-bit addressing and unified address space
- Error correcting codes
- Faster context switching
- Faster atomic instructions

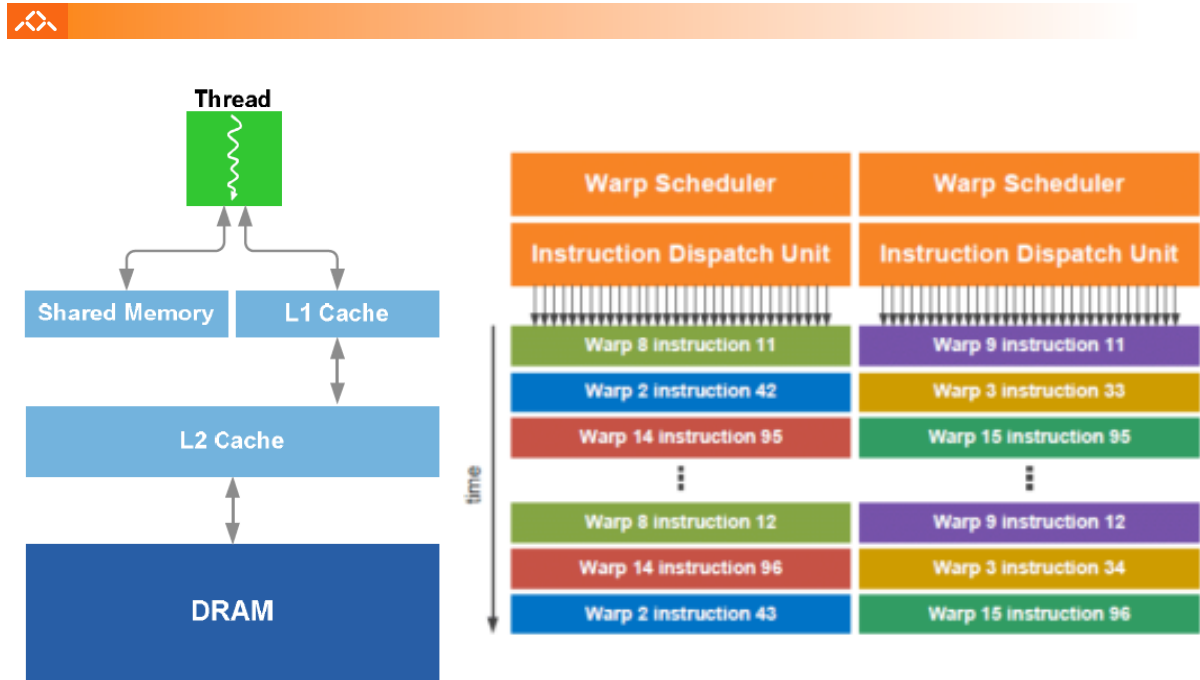


Beyond Vector/SIMD architectures



- Vector/SIMD-extended architectures are hybrid approaches
 - mix (super)scalar + vector op capabilities on a single device
 - highly pipelined approach to reduce memory access penalty
 - tightly-closed access to shared memory: lower latency
- Evolution of Vector/SIMD-extended architectures
 - **PU (Processing Unit) cores with wider vector units**
 - x86 many-core: Intel MIC / Xeon KNL
 - other many-core: IBM BlueGene/Q Compute, ShenWay 260
 - **coprocessors (require a host scalar processor): accelerator devices**
 - on disjoint physical memories (e.g., Xeon KNC with PCI-Expr, PEZY-SC)
 - ISA-free architectures, code compiled to silica: FPGA
 - focus on SIMT/SIMD to hide memory latency: GPU-type approach
 - ...
 - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
 - ...

Fermi: Multithreading and Memory Hierarchy



AJProença, *Parallel Computing*, MiEI, UMinho, 2018/19

13

TOP500 list in November 2010: 3 systems in the top4 use Fermi GPUs



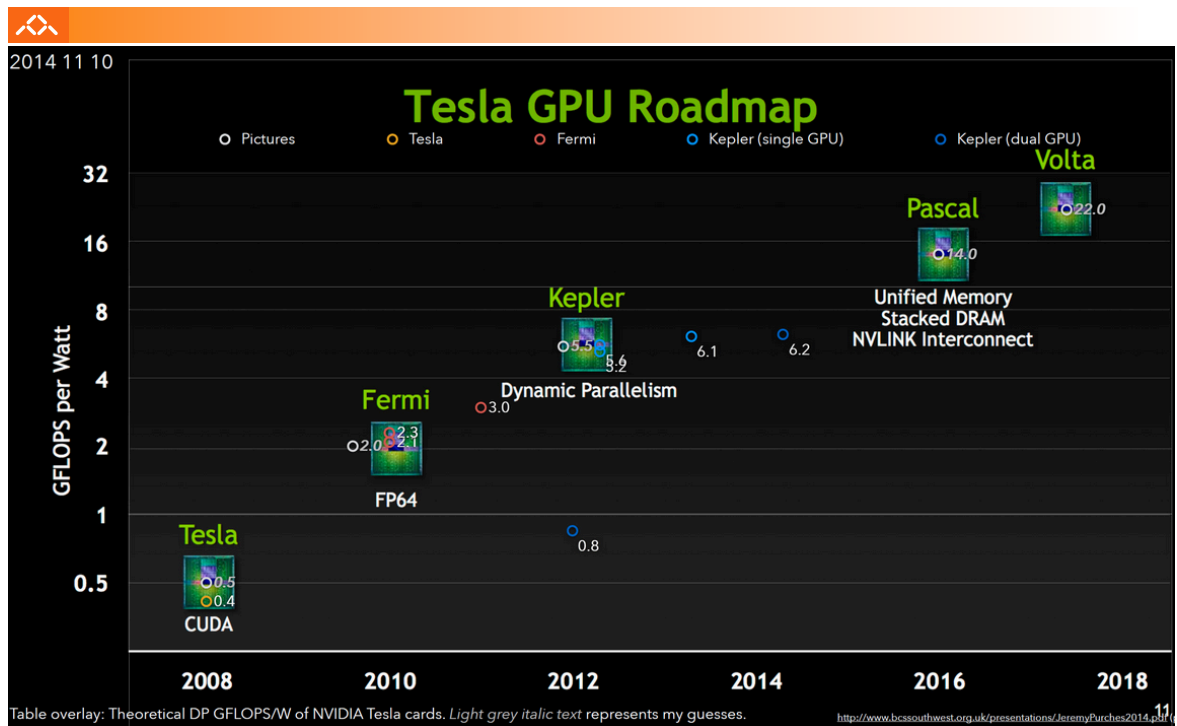
HIGHLIGHTS: NOVEMBER 2010

- The Chinese Tianhe-1A system is the new No. 1 on the TOP500 and clearly in the lead with 2.57 petaflop/s performance.
- No. 3 is also a Chinese system called Nebulae, built from a Dawning TC3600 Blade system with Intel X5650 processors and NVIDIA Tesla C2050 GPUs
- There are seven petaflop/s systems in the TOP10
- The U.S. is tops in petaflop/s with three systems performing at the petaflop/s level
- The two Chinese systems and the new Japanese Tsubame 2.0 system at No. 4 are all using NVIDIA GPUs to accelerate computation and a total of 28 systems on the list are using GPU technology.

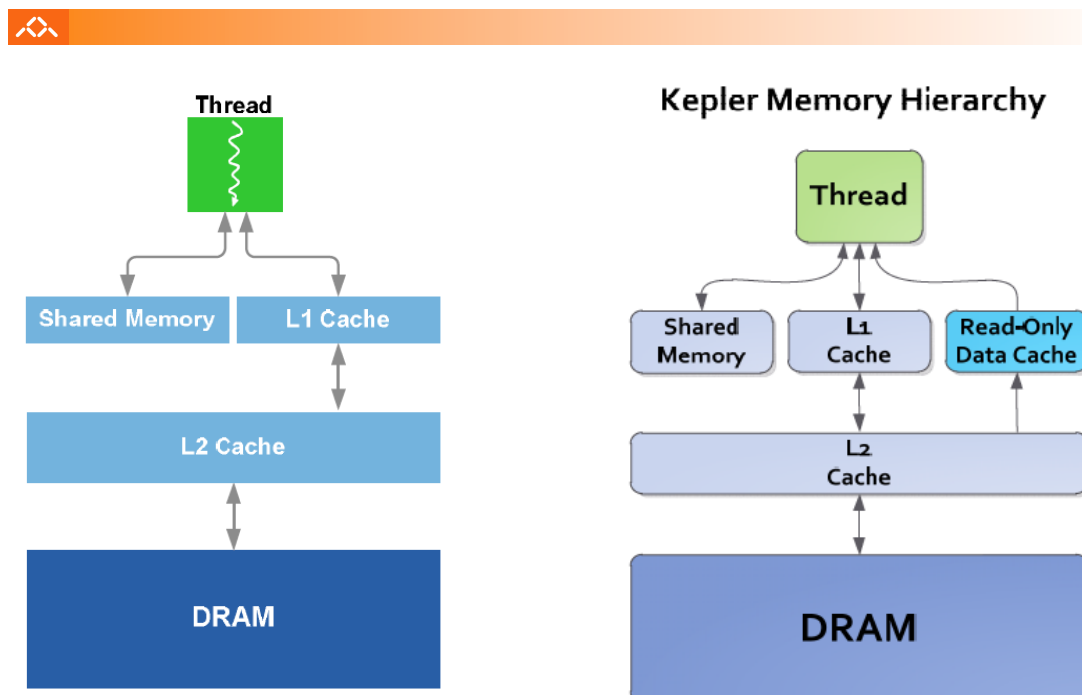
AJProença, *Parallel Computing*, MiEI, UMinho, 2018/19

14

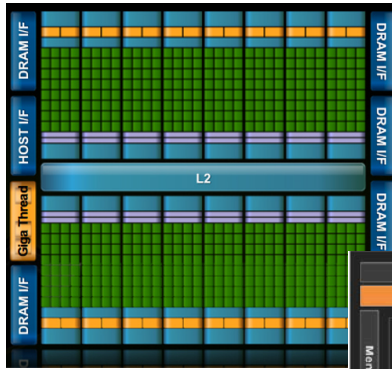
Families in NVidia Tesla GPUs



From Fermi into Kepler: The Memory Hierarchy



From the GF110 to the GK110 Kepler Architecture



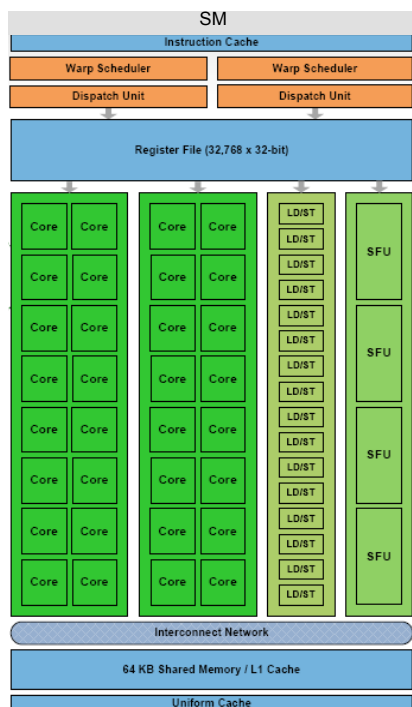
Fermi:
512 CUDA-cores
July'11



Kepler:
2880 CUDA-cores
October'13

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

17

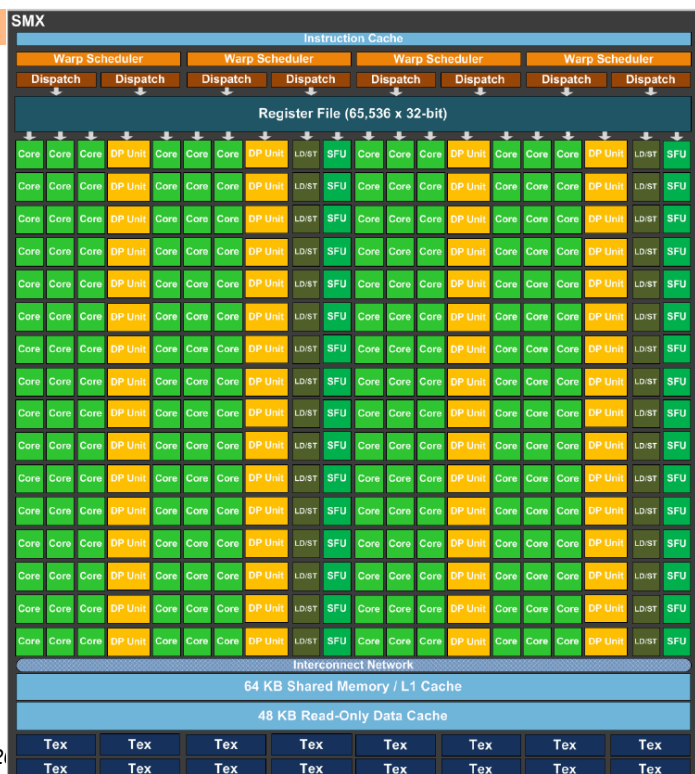


SMX:
192 CUDA-cores

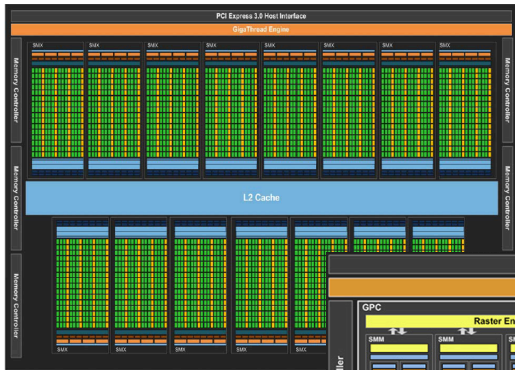
Ratio DPunit : SPunit → 1 : 3

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

From Fermi to Kepler core: SM and the SMX Architecture

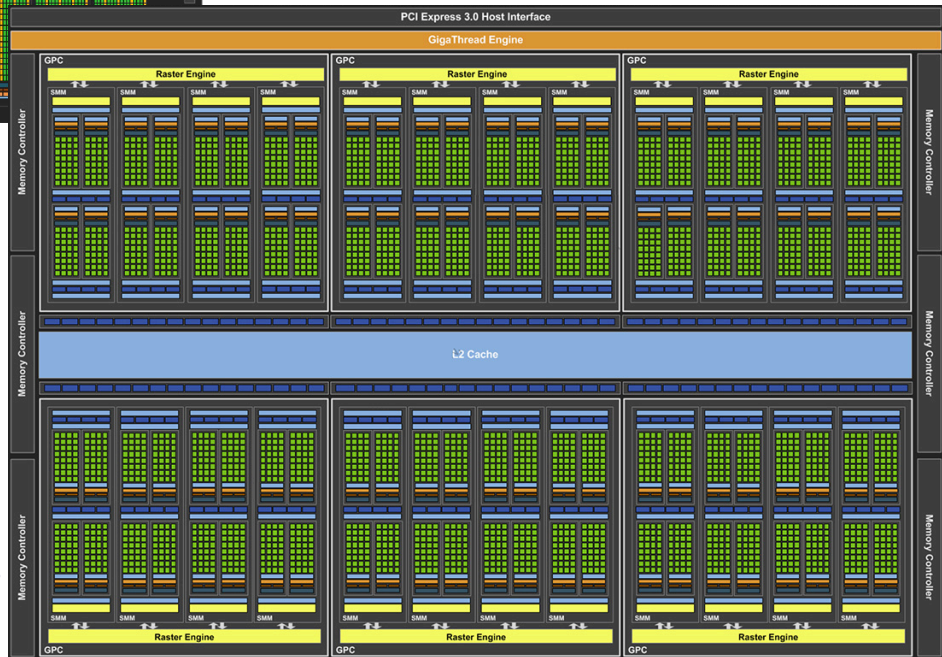


From the GK110 to the GM200 Maxwell Architecture



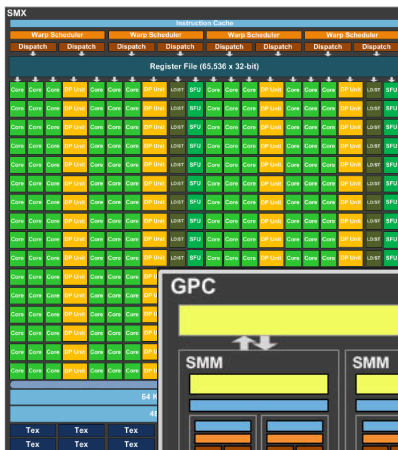
Kepler:
2880 CUDA-cores
October'13

Maxwell:
3072 CUDA-cores
November'15



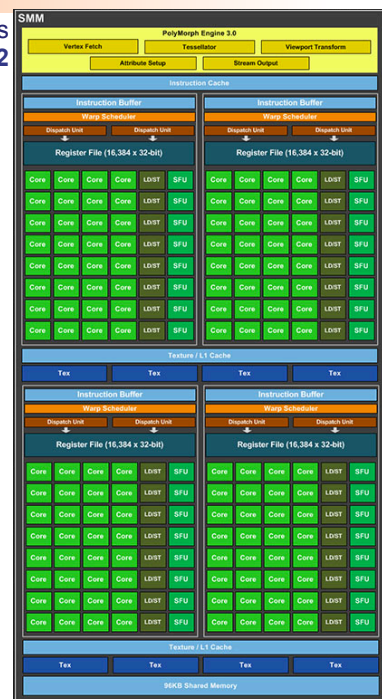
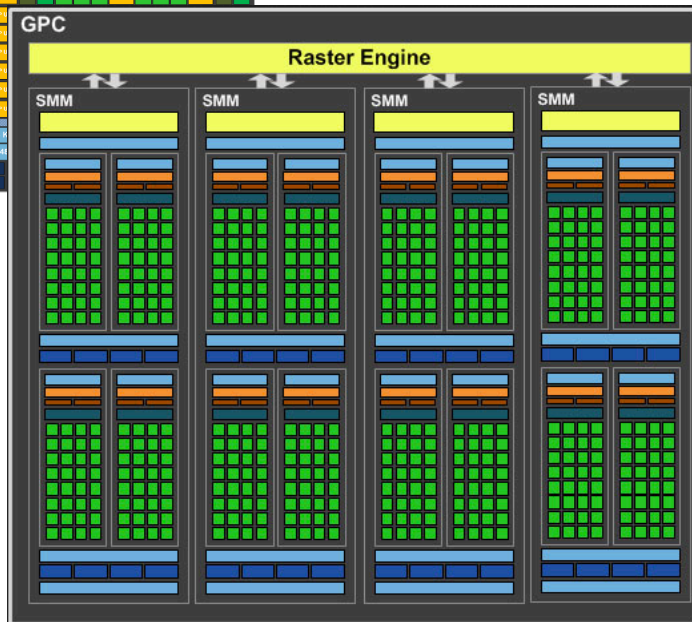
AJProença, Parallel Computing, MiEI, UMinho, 2018/19

19



The move from Kepler to Maxwell :
from 15 SMXs to 48 SMMs in 6 GPCs

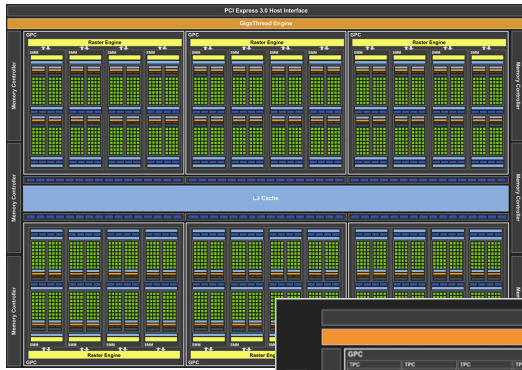
SMM: 128 CUDA-cores
Ratio DPunit : SPunit → 1 : 32



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

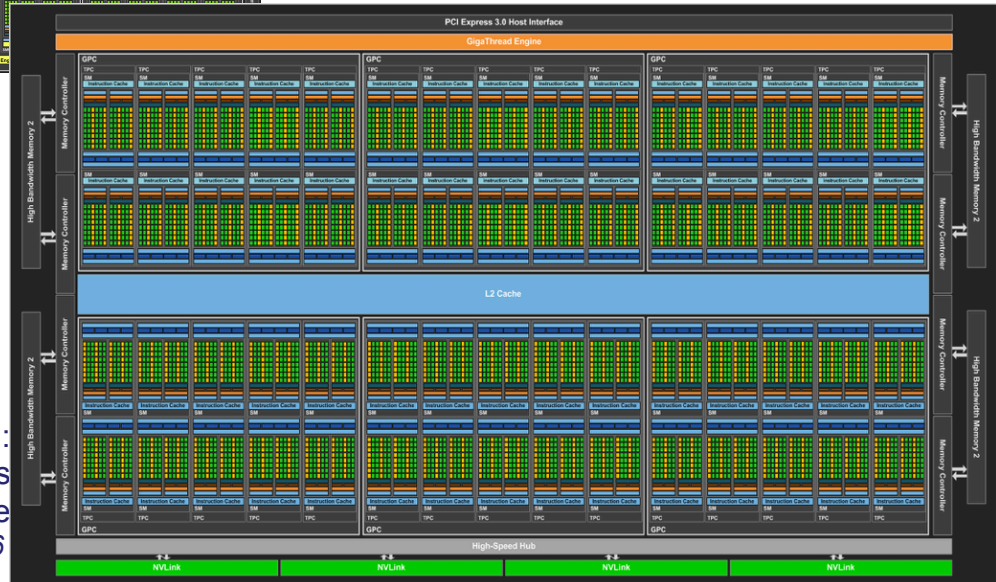
20

From the M200 to the GP100 Pascal Architecture



Maxwell:
3072 CUDA-cores
November'15

Pascal:
3584 CUDA-cores
HBM on-package
September'16



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

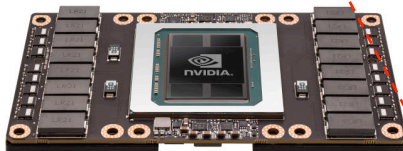
21



Pascal Architecture:
6x GPCs, 60 SMs

Pascal SM:
64 CUDA-cores
Ratio DPunit : SPunit → 1 : 2

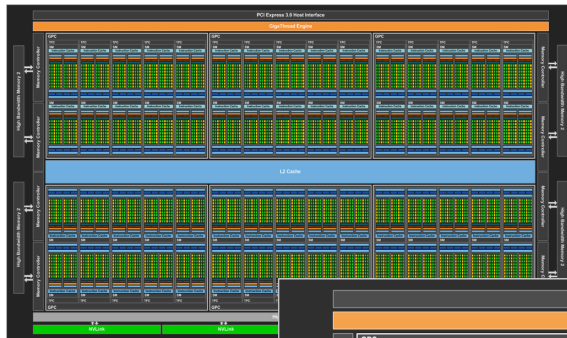
Pascal P100 w/ 16GB HBM2



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

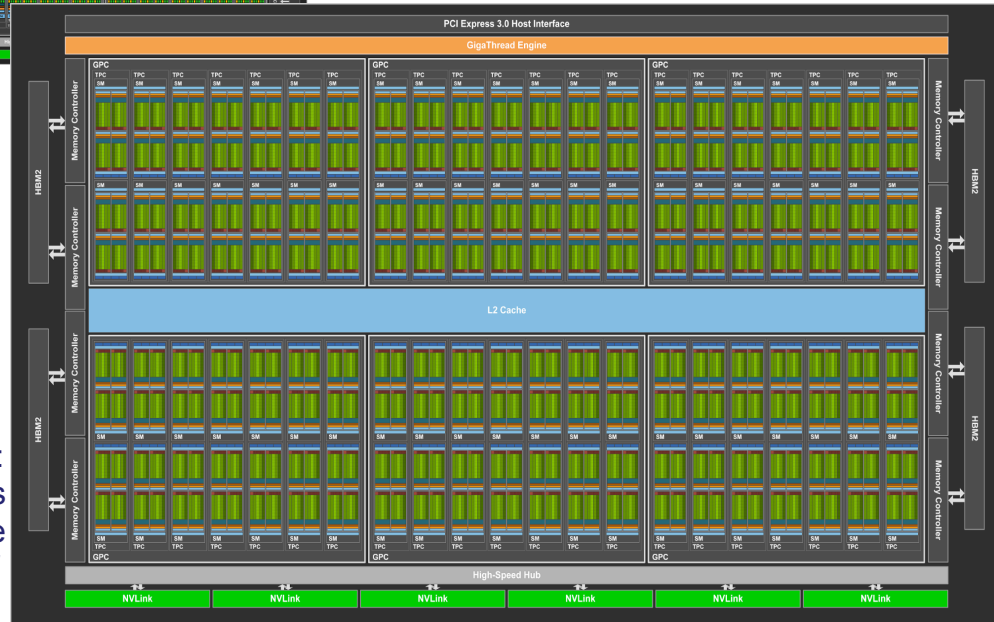
22

From the GP100 to the GV100 Volta Architecture



Pascal:
3584 CUDA-cores
November'15

Volta:
5120 CUDA-cores
HBM on-package
June'17



AJProença, Parallel Computing, MiEI, UMinho, 2018/19

23



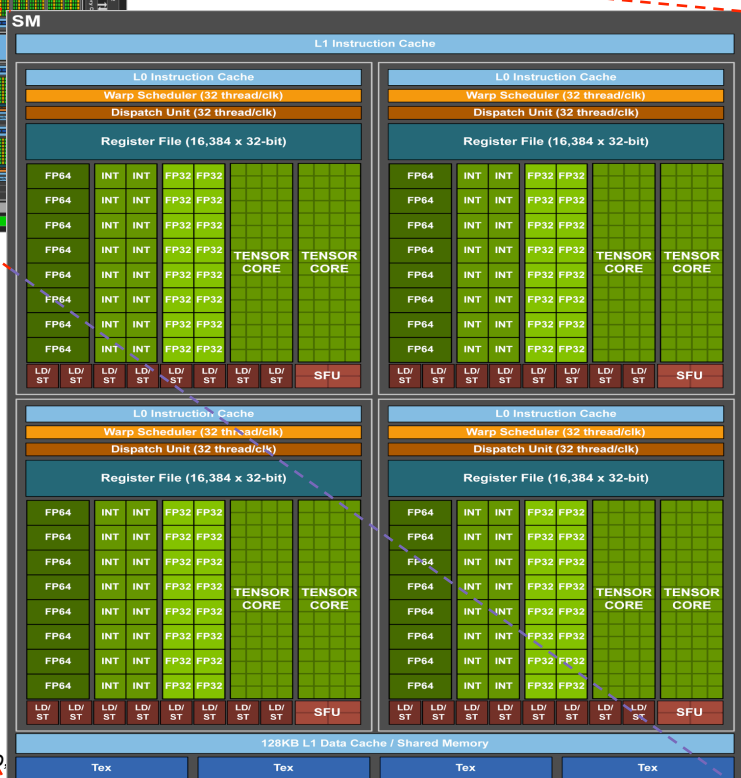
Volta Architecture:
6x GPCs, 80 SMs

Volta SM:
64 CUDA-cores
New: 8 Tensor-cores
Ratio DPunit : SPunit → 1 : 2

Volta V100 w/ 16GB HBM2

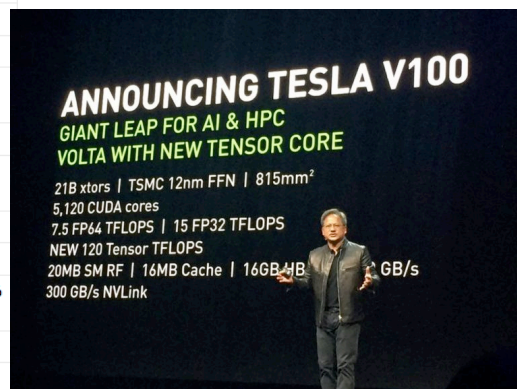


AJProença, Parallel Computing, MiEI, UMinho



Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOP/s*	5.04	6.8	10.6	15.7
Peak FP64 TFLOP/s*	1.68	.21	5.3	7.8
Peak Tensor Core TFLOP/s*	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

Tesla accelerators: recent evolution



<https://devblogs.nvidia.com/paralleforall/inside-volta/>

25

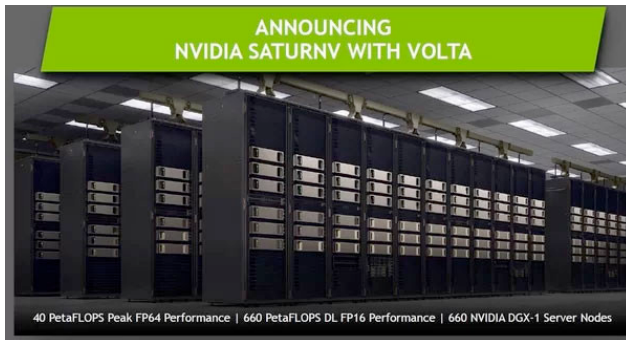


Current top 10 greener-HPC systems Nov'17 Green500

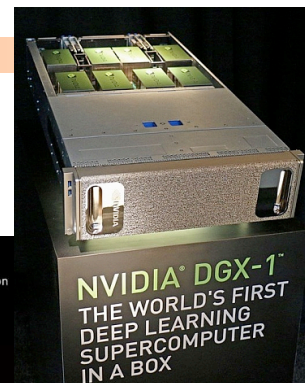
Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	259	Shoubu system B - ZettaScaler-2.2, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2, PEZY Computing / Exascaler Inc. Advanced Center for Computing and Communication, RIKEN Japan	794,400	842.0	50	17.009
2	307	Suiren2 - ZettaScaler-2.2, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2, PEZY Computing / Exascaler Inc. High Energy Accelerator Research Organization /KEK Japan	762,624	788.2	47	16.759
3	276	Sakura - ZettaScaler-2.2, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband EDR, PEZY-SC2, PEZY Computing / Exascaler Inc. PEZY Computing K.K. Japan	794,400	824.7	50	16.657
4	149	DGX SaturnV Volta - NVIDIA DGX-1 Volta36, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla V100, NVIDIA Corporation United States	22,440	1,070.0	97	15.113
5	4	Gyokou - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz, ExaScaler Japan Agency for Marine-Earth Science and Technology Japan	19,860,000	19,135.8	1,350	14.173
6	13	TSUBAME3.0 - SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni-Path, NVIDIA Tesla P100 SXM2, HPE GSIC Center, Tokyo Institute of Technology Japan	135,828	8,125.0	792	13.704
7	195	AIST AI Cloud - NEC 4U-8GPU Server, Xeon E5-2630Lv4 10C 1.8GHz, Infiniband EDR, NVIDIA Tesla P100 SXM2, NEC National Institute of Advanced Industrial Science and Technology Japan	23,400	961.0	76	12.681
8	419	RAIDEN GPU subsystem - NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100, Fujitsu Center for Advanced Intelligence Project, RIKEN Japan	11,712	635.1	60	10.603
9	115	Wilkes-2 - Dell C4130, Xeon E5-2650v4 12C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100, Dell EMC University of Cambridge United Kingdom	21,240	1,193.0	114	10.428
10	3	Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries Interconnect, NVIDIA Tesla P100, Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	2,272	10.398

AJP

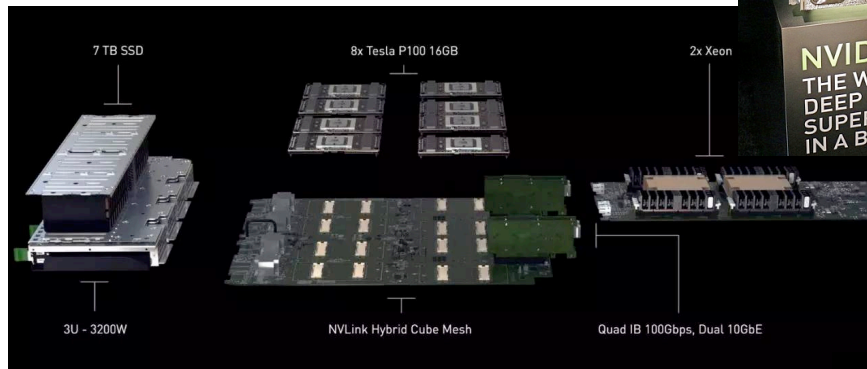
26



NVidia DGX-1 SaturnV

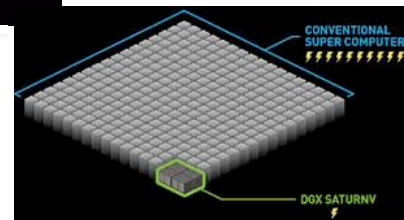


\$149,000



4	149	DGX SaturnV Volta	22,440	1,070.0	97	15.113
		NVIDIA DGX-1 Volta36,				
		Xeon E5-2698v4 20C				
		2.2GHz, Infiniband EDR,				
		NVIDIA Tesla V100 , Nvidia				
		NVIDIA Corporation				
		United States				

AJProença, Parallel Computing, MiEI, UMinho, 2018/19



The CUDA programming model

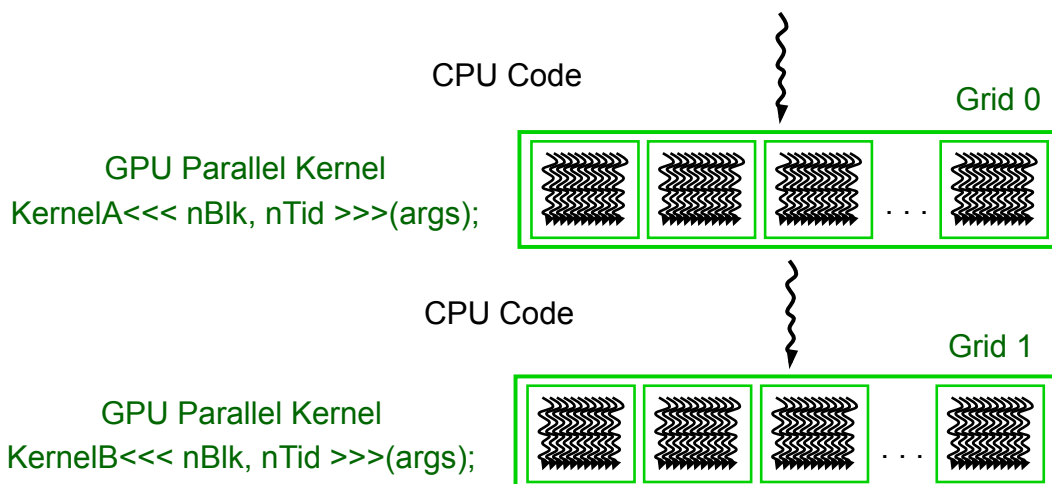


- **Compute Unified Device Architecture**
- CUDA is a recent programming model, designed for
 - a multicore CPU **host** coupled to a many-core **device**, where
 - **devices** have wide SIMD/SIMT parallelism, and
 - the **host** and the **device** do not share memory
 - CUDA provides:
 - a thread abstraction to deal with SIMD
 - synchr. & data sharing between small groups of threads
 - CUDA programs are written in C with extensions
- OpenCL inspired by CUDA, but hw & sw vendor neutral
 - programming model essentially identical

- A compute **device**
 - is a coprocessor to the CPU or **host**
 - has its own DRAM (**device memory**)
 - runs many **threads in parallel**
 - is typically a **GPU** but can also be another type of parallel processing device
- Data-parallel portions of an application are expressed as device **kernels** which run on many threads - **SIMT**
- Differences between GPU and CPU threads
 - GPU threads are extremely lightweight
 - very little creation overhead, **requires LARGE register bank**
 - GPU needs 1000s of threads for full efficiency
 - multi-core CPU needs only a few

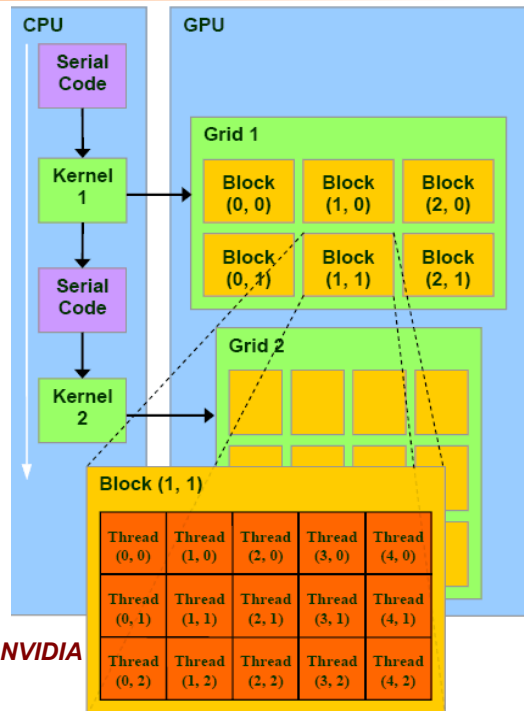
CUDA basic model: Single-Program Multiple-Data (SPMD)

- CUDA integrated CPU + GPU application C program
 - Serial C code executes on CPU
 - Parallel **Kernel** C code executes on GPU **thread blocks**



Programming Model: SPMD + SIMT/SIMD

- Hierarchy
 - Device => Grids
 - Grid => Blocks
 - Block => Warps
 - Warp => Threads
- Single kernel runs on multiple blocks (SPMD)
- Threads within a warp are executed in a lock-step way called single-instruction multiple-thread (SIMT)
- Single instruction are executed on multiple threads (SIMD)
 - Warp size defines SIMD granularity (32 threads)
- Synchronization within a block uses shared memory

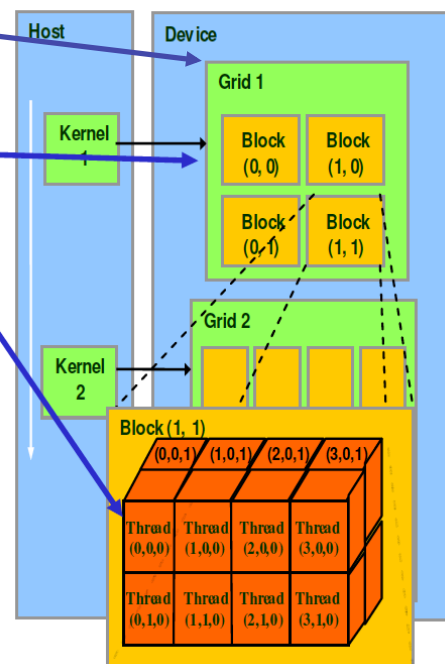


Courtesy NVIDIA

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

The Computational Grid: Block IDs and Thread IDs

- A **kernel** runs on a **computational grid of thread blocks**
 - Threads share global memory
- Each thread uses IDs to decide what data to work on
 - Block ID: 1D or 2D
 - Thread ID: 1D, 2D, or 3D
- A thread block is a batch of threads that can cooperate by:
 - Sync their execution w/ barrier
 - Efficiently sharing data through a low latency shared memory
 - Two threads from two different blocks cannot cooperate



© David Kirk / NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

Example

- Multiply two vectors of length 8192
 - Code that works over all elements is the grid
 - Thread blocks break this down into manageable sizes
 - 512 threads per block
 - SIMD instruction executes 32 elements at a time
 - Thus grid size = 16 blocks
 - Block is analogous to a strip-mined vector loop with vector length of 32
 - Block is assigned to a *multithreaded SIMD processor* by the *thread block scheduler*
 - Current-generation GPUs (Fermi) have 7-16 multithreaded SIMD processors

Copyright © 2012, Elsevier Inc. All rights reserved.

AJProença, *Parallel Computing*, MiEI, UMinho, 2018/19

33

C with CUDA Extensions: C with a few keywords

```
void saxpy_serial(int n, float a, float *x, float *y)
{
    for (int i = 0; i < n; ++i)
        y[i] = a*x[i] + y[i];
}
// Invoke serial SAXPY kernel
saxpy_serial(n, 2.0, x, y);
```

Standard C Code

```
__global__ void saxpy_parallel(int n, float a, float *x, float *y)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if (i < n) y[i] = a*x[i] + y[i];
}
// Invoke parallel SAXPY kernel with 256 threads/block
int nblocks = (n + 255) / 256;
saxpy_parallel<<<nblocks, 256>>>>(n, 2.0, x, y);
```

Parallel C Code

NVIDIA Confidential

Terminology (and in NVidia)

- Threads of SIMD instructions (**warps**)
 - Each has its own IP (up to 48/64 per SIMD processor, Fermi/Kepler)
 - Thread scheduler uses scoreboard to dispatch
 - No data dependencies between threads!
 - Threads are organized into blocks & executed in groups of 32 threads (**thread block**)
 - Blocks are organized into a grid
- The thread block scheduler schedules blocks to SIMD processors (**Streaming Multiprocessors**)
- Within each SIMD processor:
 - 32 SIMD lanes (**thread processors**)
 - Wide and shallow compared to vector processors

Copyright © 2012, Elsevier Inc. All rights reserved.

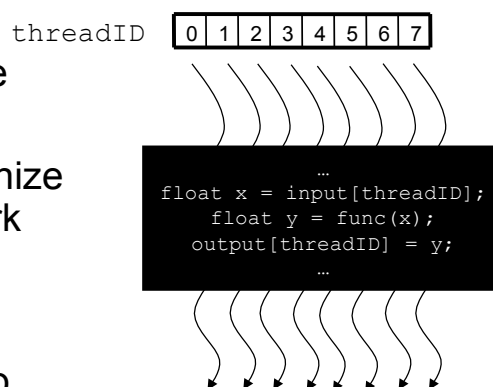
AJProença, Parallel Computing, MiEI, UMinho, 2018/19

35

CUDA Thread Block

- Programmer declares (Thread) Block:
 - Block size 1 to **512** concurrent threads
 - Block shape 1D, 2D, or 3D
 - Block dimensions in threads
- All threads in a Block execute the same thread program
- Threads share data and synchronize while doing their share of the work
- Threads have **thread id** numbers within Block
- Thread program uses **thread id** to select work and address shared data

CUDA Thread Block

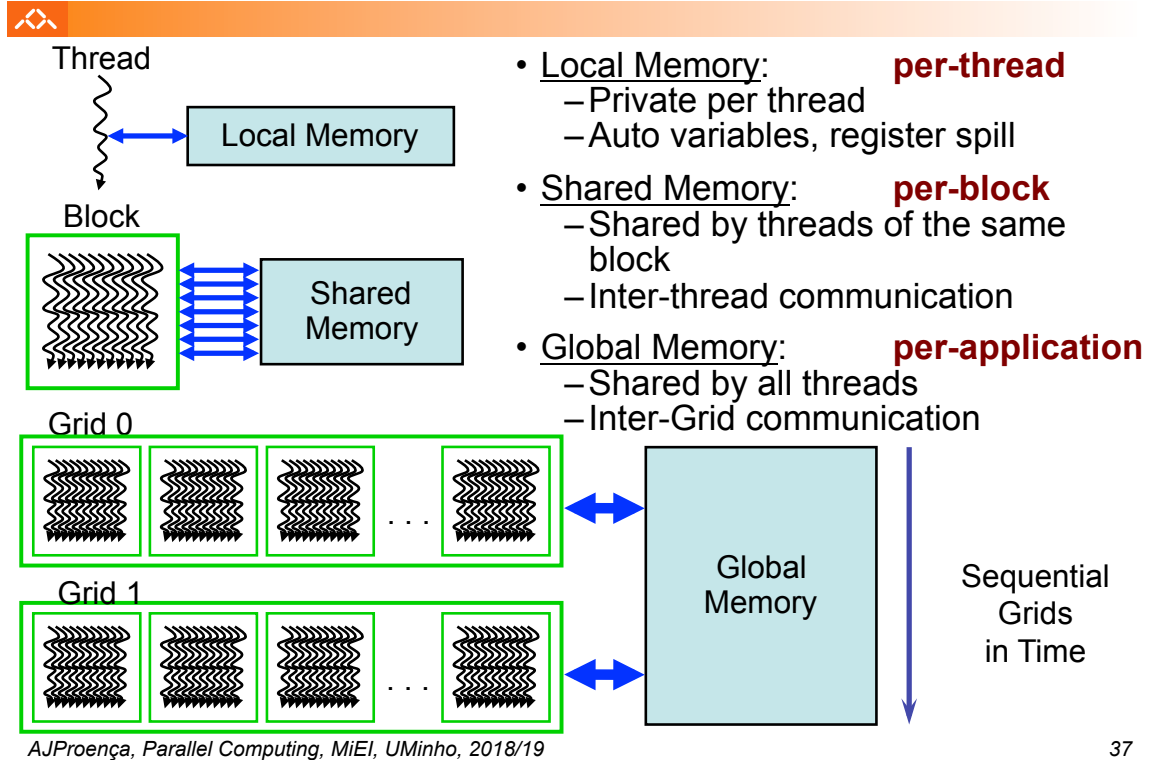


© David Kirk / NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

36

Parallel Memory Sharing

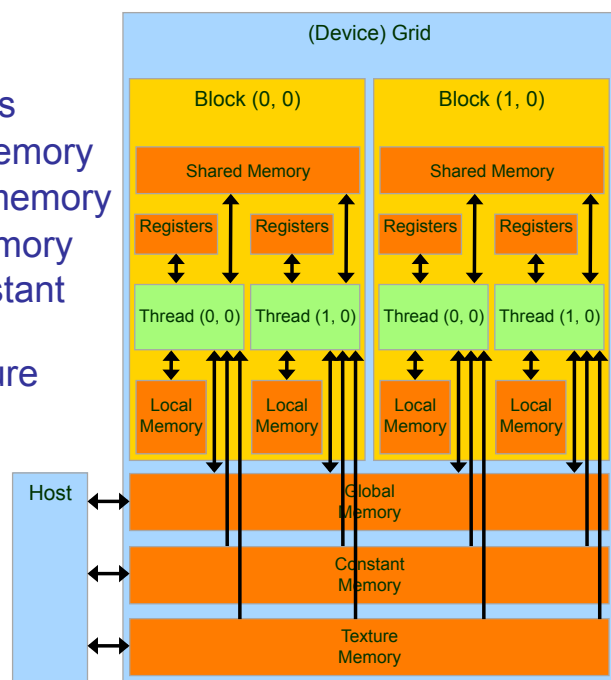


© David Kirk / NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

37

CUDA Memory Model Overview

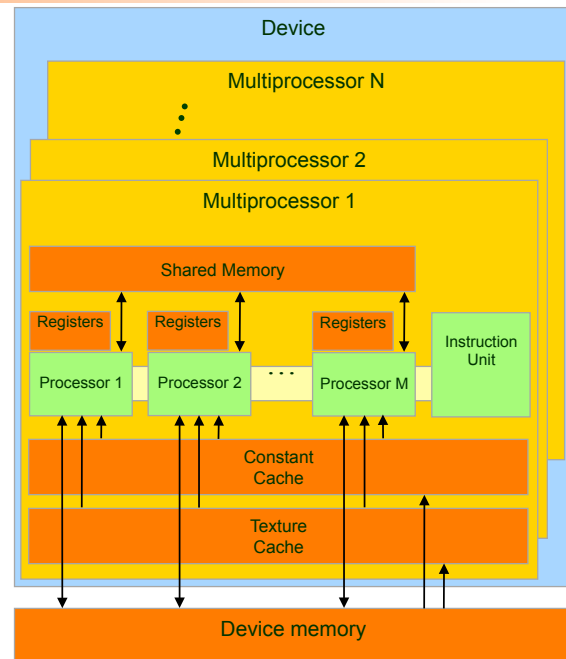
- Each thread can:
 - R/W per-thread **registers**
 - R/W per-thread **local memory**
 - R/W per-block **shared memory**
 - R/W per-grid **global memory**
 - Read only per-grid **constant memory**
 - Read only per-grid **texture memory**
- The host can R/W global, constant, and texture memories



© David Kirk / NVIDIA and Wen-mei W. Hwu, 2007-2009
ECE 498AL, University of Illinois, Urbana-Champaign

Hardware Implementation: Memory Architecture

- Device memory (DRAM)
 - Slow (2~300 cycles)
 - Local, global, constant, and texture memory
- On-chip memory
 - Fast (1 cycle)
 - Registers, shared memory, constant/texture cache



Courtesy NVIDIA

39

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

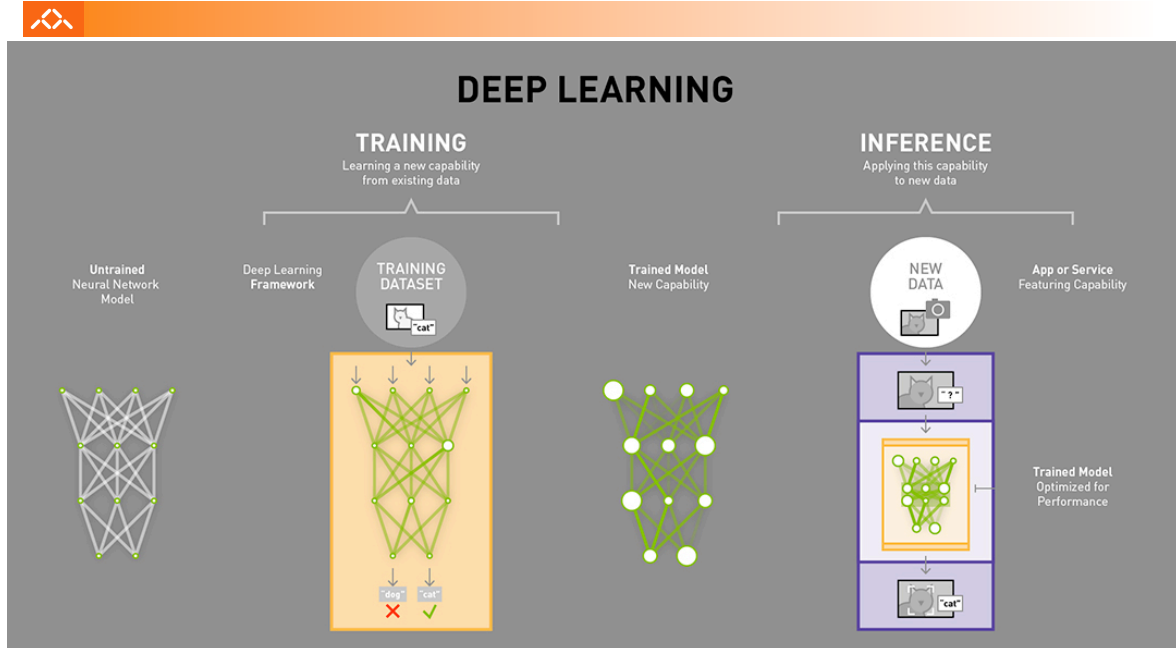
Beyond Vector/SIMD architectures

- Vector/SIMD-extended architectures are hybrid approaches
 - mix (super)scalar + vector op capabilities on a single device
 - highly pipelined approach to reduce memory access penalty
 - tightly-closed access to shared memory: lower latency
- Evolution of Vector/SIMD-extended architectures
 - **PU (Processing Unit) cores with wider vector units**
 - x86 many-core: Intel MIC / Xeon KNL
 - other many-core: IBM Power BlueGene/Q Compute, ShenWay 260
 - **coprocessors (require a host scalar processor): accelerator devices**
 - on disjoint physical memories (e.g., Xeon KNC with PCI-Expr, PEZY-SC)
 - ISA-free architectures, code compiled to silica: FPGA
 - focus on SIMT/SIMD to hide memory latency: GPU-type approach
 - focus on tensor/neural nets cores: Nvidia, IBM, Intel NNP, Google TPU
 - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
 - ...

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

40

Machine learning w/ neural nets & deep learning...



Key algorithms to train & classify use matrix products, but require lower precision numbers!

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

41

NVidia Volta Architecture: the new Tensor Cores

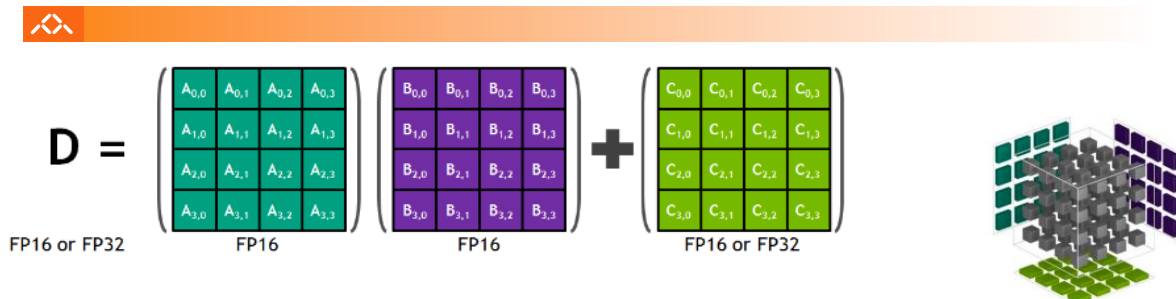


Figure 8. Tensor Core 4x4 Matrix Multiply and Accumulate



For each SM:
8x 64 FMA ops/cycle
1k FLOPS/cycle!

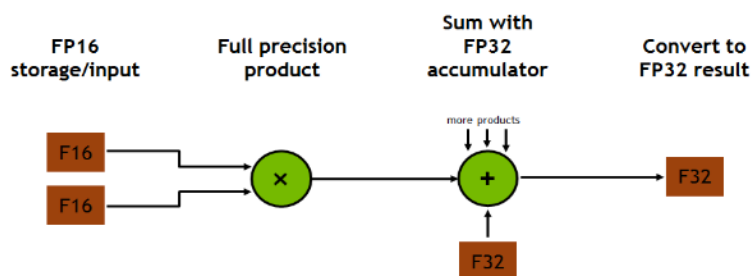


Figure 9. Mixed Precision Multiply and Accumulate in Tensor Core

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

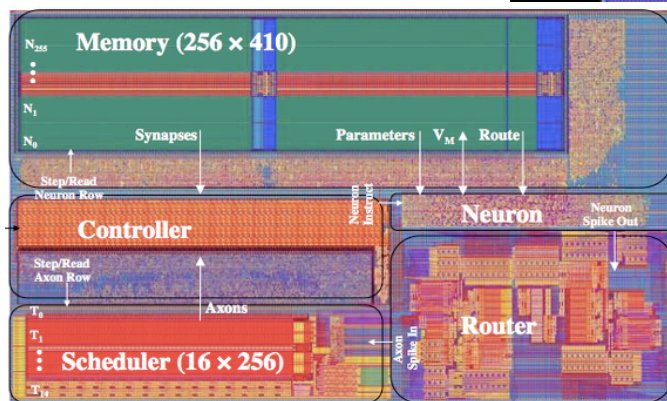
42

NVidia competitors with neural net features: IBM TrueNorth chip array (August'2014)

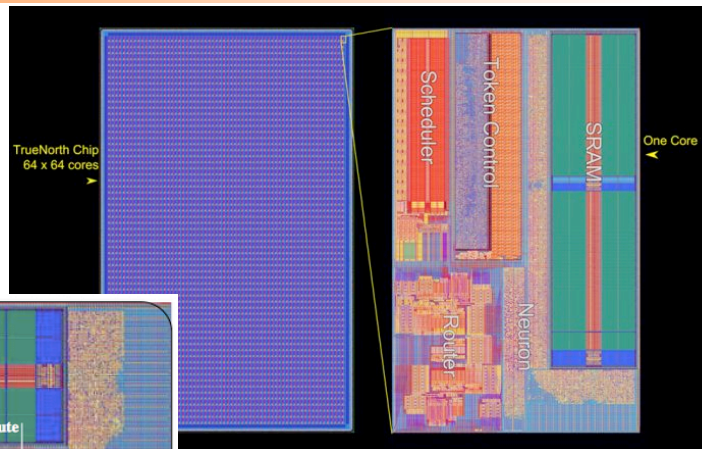


TrueNorth Chip:

- 4096 neurosynaptic cores
- Each core:
 - 256 inputs (axons)
 - 256 outputs (neurons)
 - RAM w/ data for each neuron
 - router (any neuron to any axon)

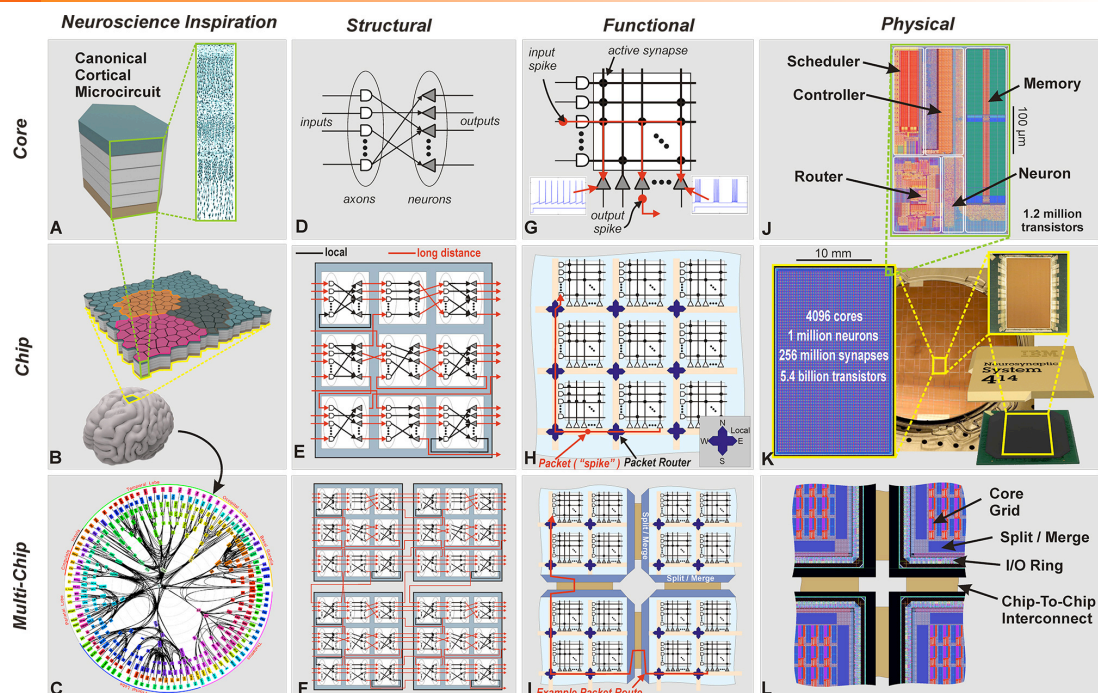


AJProença, Parallel Computing, MiEI, UMinho, 2018/19



43

NVidia competitors with neural net features: the IBM TrueNorth architecture



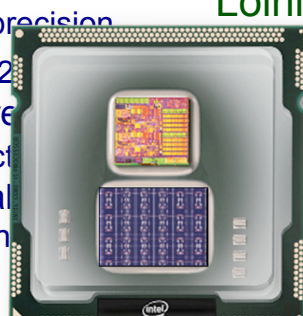
AJProença, Parallel Computing, MiEI, UMinho, 2018/19

44

NVidia competitors with neural net features: Intel Nervana Neural Network Processor, NNP

History

- Nervana Engine announced in May'16
 - Key features:
 - ASIC chip, focused on matrix multiplication, convolutions, ... (for neural nets)
 - HBM2: 4x 8GB in-package storage & 1TB/sec memory access b/w
 - no h/w managed cache hierarchy (saves die area, higher compute density)
 - built-in networking (6 bi-directional high-b/w links)
 - separate pipelines for computation and data management
 - proprietary numeric format Flexpoint in-between floating point and fixed point precision
 - Nervana acquired by Intel in August 2016
 - renamed the project to "Lake Crest"
 - later to Nervana NNP, launched in October 2016
 - Loihi test chip w/ self-learning capabilities announced in Sept'17, to be launched in 2018



Loihi

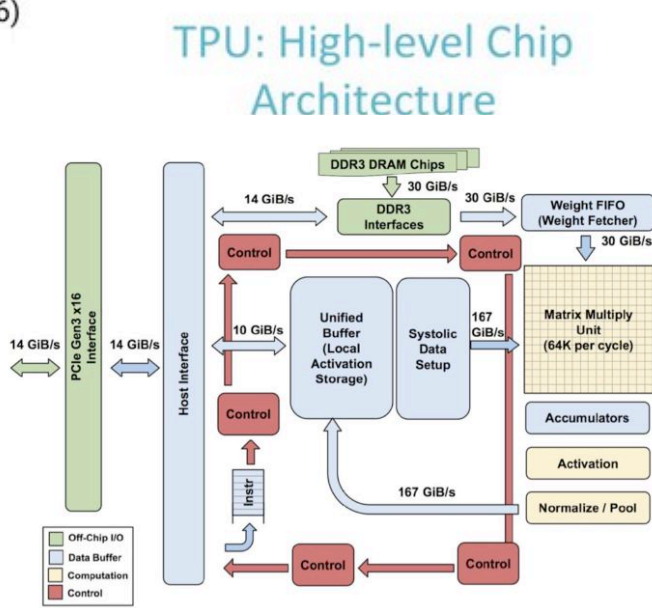
<https://www.top500.org/news/intel-will-ship-first-neural-network-chip-this-year/>

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

NVidia competitors with neural net features: Google Tensor Processing Unit, TPU (April'17)

TPU: High-level Chip Architecture

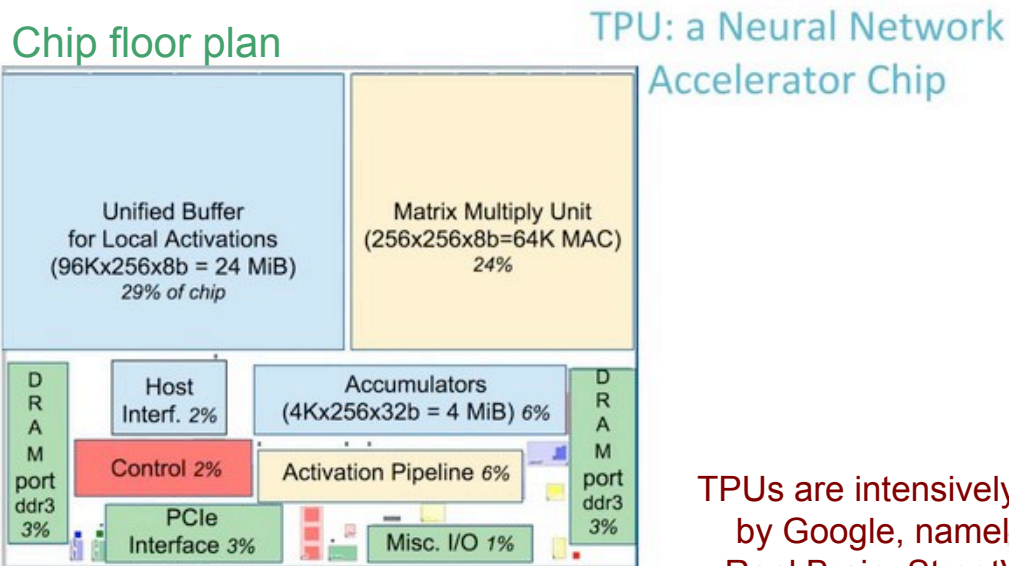
- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units
- 700 MHz clock rate
- Peak: 92T operations/second
 - $65,536 * 2 * 700M$
- >25X as many MACs vs GPU
- >100X as many MACs vs CPU
- 4 MiB of on-chip Accumulator memory
- 24 MiB of on-chip Unified Buffer (activation memory)
- 3.5X as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GiB of off-chip weight DRAM memory



Not to Scale

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

NVidia competitors with neural net features: Google Tensor Processing Unit, TPU (April'17)



TPUs are intensively used by Google, namely in RankBrain, StreetView & Google Translate

<https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processor>

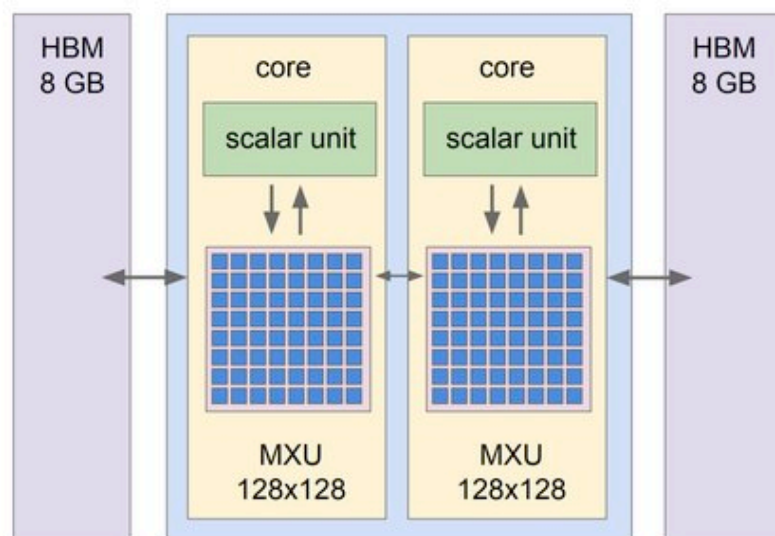
NVidia competitors with neural net features: Google TPuv2 (September'17)



TPUv2 Chip



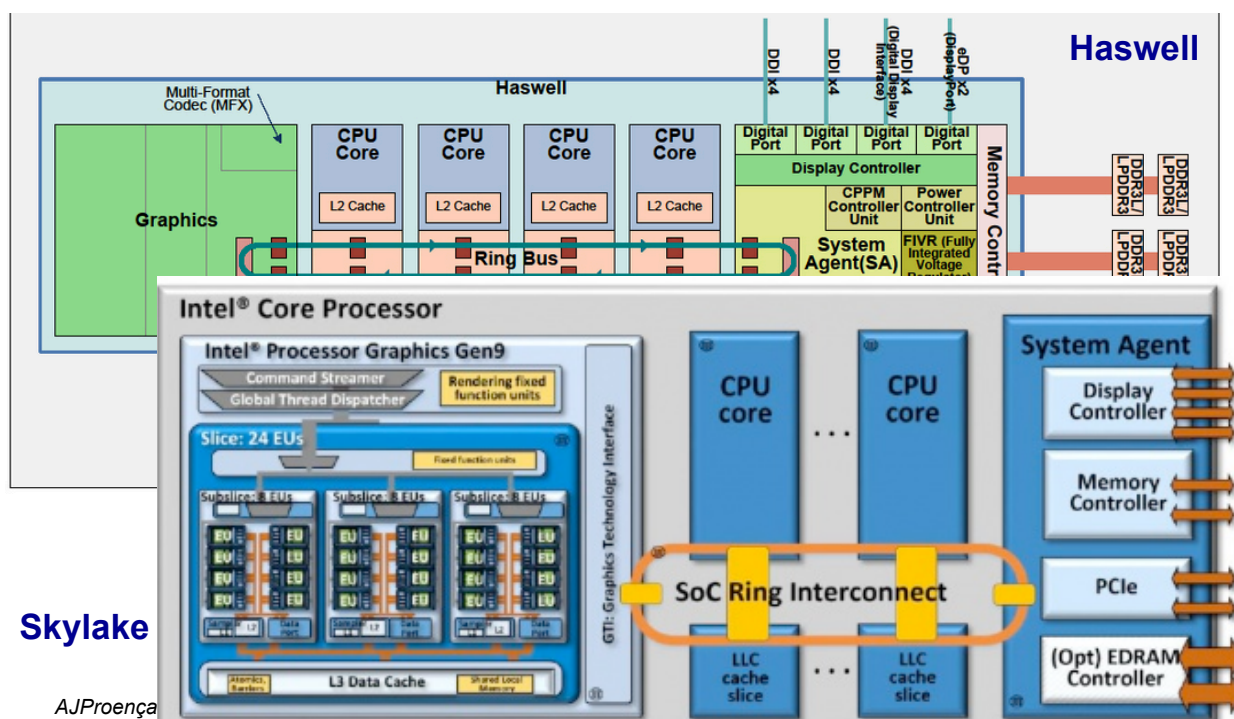
- 16 GB of HBM
- 600 GB/s mem BW
- Scalar unit: 32b float
- MXU: 32b float accumulation but reduced precision for multipliers
- 45 TFLOPS



Beyond Vector/SIMD architectures

- Vector/SIMD-extended architectures are hybrid approaches
 - mix (super)scalar + vector op capabilities on a single device
 - highly pipelined approach to reduce memory access penalty
 - tightly-closed access to shared memory: lower latency
- Evolution of Vector/SIMD-extended architectures
 - **PU (Processing Unit) cores with wider vector units**
 - x86 many-core: **Intel MIC / Xeon KNL**
 - other many-core: **IBM Power BlueGene/Q Compute, ShenWay 260**
 - **coprocessors (require a host scalar processor): accelerator devices**
 - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
 - ISA-free architectures, code compiled to silica: **FPGA**
 - focus on SIMT/SIMD to hide memory latency: **GPU-type approach**
 - focus on tensor/neural nets cores: **Nvidia, IBM, Intel NNP, Google TPU**
 - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
 - x86 multicore coupled with SIMT/SIMD cores: **Intel i5/i7**
 - **ARMv8** cores coupled with SIMT/SIMD cores: **Nvidia Tegra**

Intel multicore coupled with GPU-cores

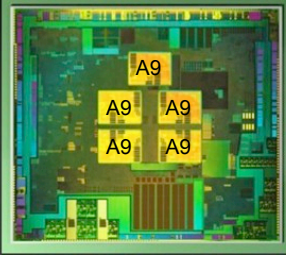


NVidia Tegra: SoC partnership with ARM (1)

- Tegra 2 in Android (2010) ...
- **Tegra 3** in Audi infotainment (2012) ...

Tegra 3 The World's First Mobile Quad Core, with 5th Companion Core for Low Power

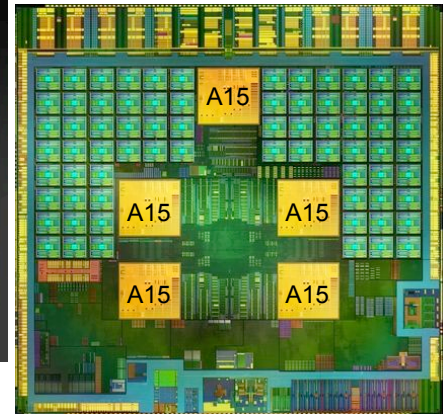
CPU	Quad Core, with 5 th Companion Core — Up to 1.4GHz Single Core, 1.3GHz Quad Core
GPU	Up to 3x Higher GPU Performance — 12 Core GeForce GPU
VIDEO	Blu-Ray Quality Video — 1080p High Profile @ 40Mbps
POWER	Lower Power than Tegra 2 — Variable Symmetric Multiprocessing (vSMP)
MEMORY	Up to 3x Higher Memory Bandwidth — DDR3L-1500, LPDDR2-1066
IMAGING	Up to 2x Faster ISP (Image Signal Processor)
AUDIO	HD Audio, 7.1 channel surround
STORAGE	2-6x Faster — eMMC 4.41, SD3.0, SATA-II



Tegra 3 Nov'2011

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

Tegra 4:
replace the 32-bit ARM Cortex A9 by Cortex A15, and add 72 CUDA-cores



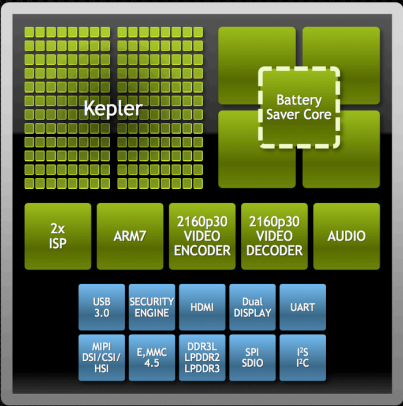
Tegra 4 May'2013

51


NVidia Tegra: SoC partnership with ARM (2)

Replace the GPU block by 192 GPU-cores (from Kepler) and offer either 32/64-bit CPU cores => **Tegra K1**

Tegra K1 Apr'2014



GPU	Kepler GPU (192 CUDA Cores) Open GL 4.4, OpenGL ES3.0, DX11, CUDA 6
CPU	Quad Core Cortex A15 "r3" With 5 th Battery-Saver Core; 2MB L2 cache
CAMERA	Dual High Performance ISP 1.2 Gigapixel throughput, 100MP sensor
POWER	Lower Power 28HPM, Battery Saver Core
DISPLAY	4K panel, 4K HDMI DSI, eDP, LVDS, High Speed HDMI 1.4a



NVidia Tegra: SoC partnership with ARM (2)



Replace the GPU block by 192 GPU-cores (from Kepler) and offer either 32/64-bit CPU cores => **Tegra K1**

TEGRA K1 Apr'2014
One Chip – Two Versions

↔

Pin Compatible

<p>Quad Core +1 (battery saver)</p> <p>32-bit</p> <p>3-way Superscalar</p> <p>Up to 2.3GHz</p> <p>32K+32K L1\$</p>	<p>Dual Super Core</p> <p>64-bit</p> <p>7-way Superscalar</p> <p>Up to 2.5GHz</p> <p>128K+64K L1\$</p>
--	--

AJProença,

53

NVidia Tegra: SoC partnership with ARM (3)



- Replace the 5x 32-bit ARM by 2x4 32-bit Cortex (A57 & A53) and the 192 Kepler CUDA cores by 256 Maxwell => **Tegra X1**
May'2015



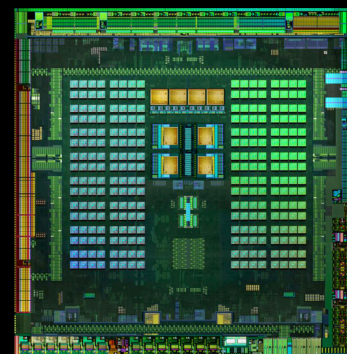
TEGRA X1 CPU CONFIGURATION

4 HIGH PERFORMANCE A57 BIG CORES

- 2MB L2 cache
- 48KB L1 instruction cache
- 32KB L1 data cache

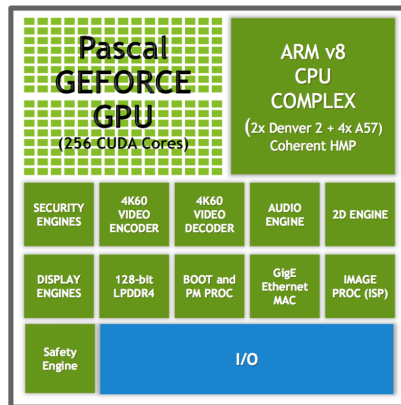
4 HIGH EFFICIENCY A53 LITTLE CORES

- 512KB L2 cache
- 32KB L1 instruction cache
- 32KB L1 data cache



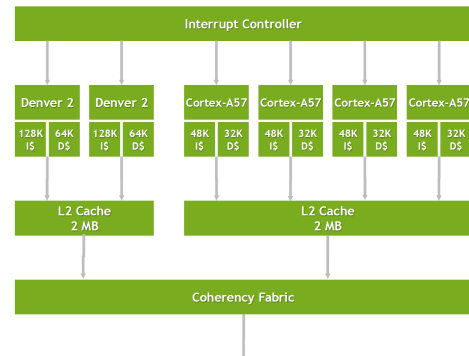
NVidia Tegra: pathway towards ARM-64 (1)

- Upgrade 32-bit ARM to 64-bit ARM (*Denver 2 & A57*) and replace Maxwell cores by Pascal ones => **Parker** Aug'2016



“PARKER” CPU COMPLEX

- 2x Denver2 + 4x Cortex-A57
- Fully Coherent HMP system
 - Proprietary Coherent Interconnect
- ARM V8 64-bit
- Highest performance ARM CPU
 - 2nd generation Denver core
 - Significant Perf/W improvements
- Dynamic Code Optimization
 - OoO execution without the power
 - Optimize once, use many times
- 7-wide superscalar
- Low power retention states



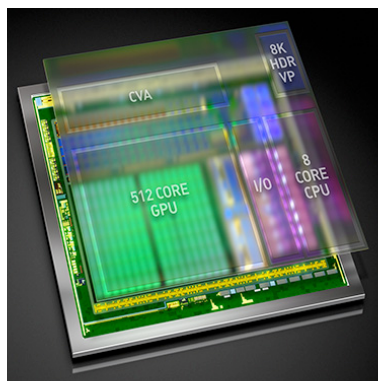
7 NVIDIA

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

55

NVidia Tegra: pathway towards ARM-64 (2)

- Increment #ARMv8-cores (*custom architecture*) and replace Pascal cores by Volta (w/ tensor cores) => **Xavier** Jan'2018?



NVIDIA ARM SoCs			
	Xavier	Parker	Erista (Tegra XI)
CPU	8x NVIDIA Custom ARM	2x NVIDIA Denver + 4x ARM Cortex-A57	4x ARM Cortex-A57 + 4x ARM Cortex-A53
GPU	Volta, 512 CUDA Cores	Pascal, 256 CUDA Cores	Maxwell, 256 CUDA Cores
Memory	?	LPDDR4, 128-bit Bus	LPDDR3, 64-bit Bus
Video Processing	7680x4320 Encode & Decode	3840x2160p60 Decode 3840x2160p60 Encode	3840x2160p60 Decode 3840x2160p30 Encode
Transistors	7B	?	?
Manufacturing Process	TSMC 16nm FinFET+	TSMC 16nm FinFET+	TSMC 20nm Planar

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

56

Beyond Vector/SIMD architectures

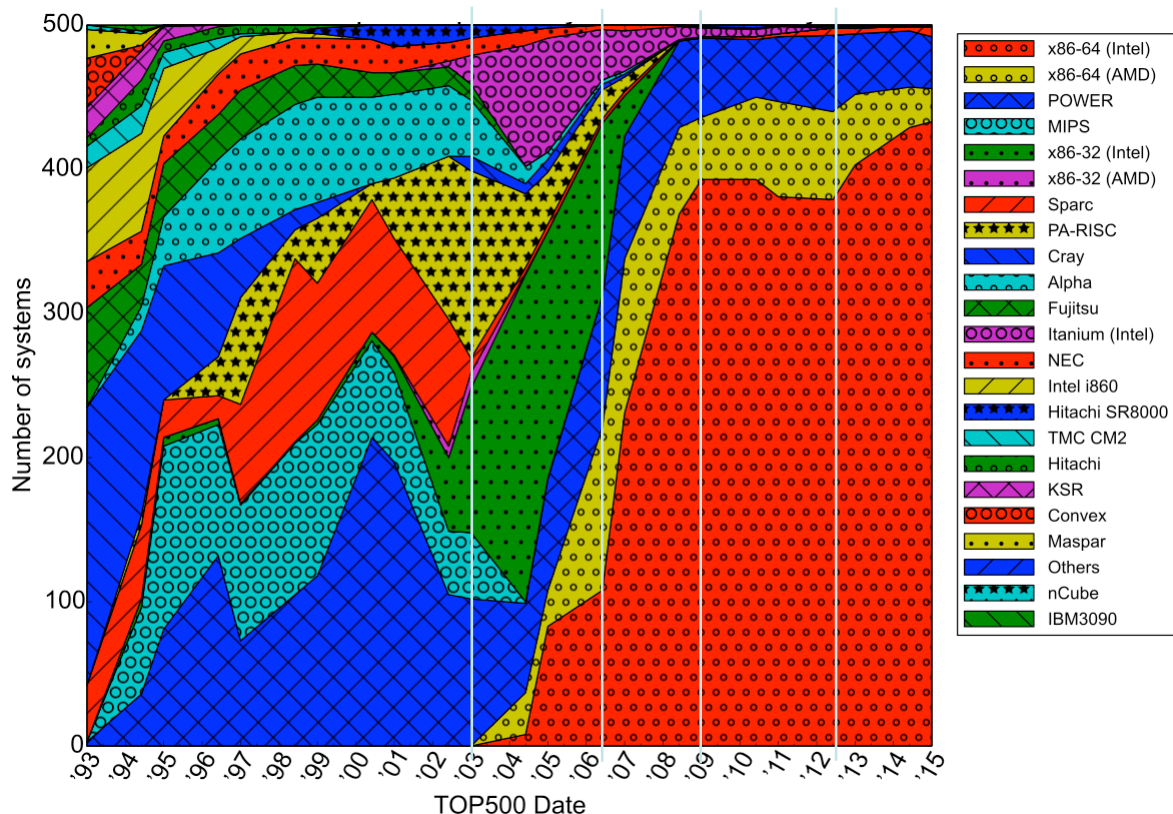
- Vector/SIMD-extended architectures are hybrid approaches
 - mix (super)scalar + vector op capabilities on a single device
 - **highly pipelined** approach to reduce memory access penalty
 - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
 - **PU (Processing Unit) cores with wider vector units**
 - x86 many-core: **Intel MIC / Xeon KNL**
 - other many-core: **IBM Power BlueGene/Q Compute, ShenWay 260**
 - **coprocessors (require a host scalar processor): accelerator devices**
 - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
 - ISA-free architectures, code compiled to silica: **FPGA**
 - focus on SIMT/SIMD to hide memory latency: **GPU-type** approach
 - focus on tensor/neural nets cores: **Nvidia, IBM, Intel NNP, Google TPU**
 - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
 - x86 multicore coupled with SIMT/SIMD cores: **Intel i5/i7**
 - **ARMv8** cores coupled with SIMT/SIMD cores: **Nvidia Tegra**

AJProença, Parallel Computing, MiEI, UMinho, 2018/19

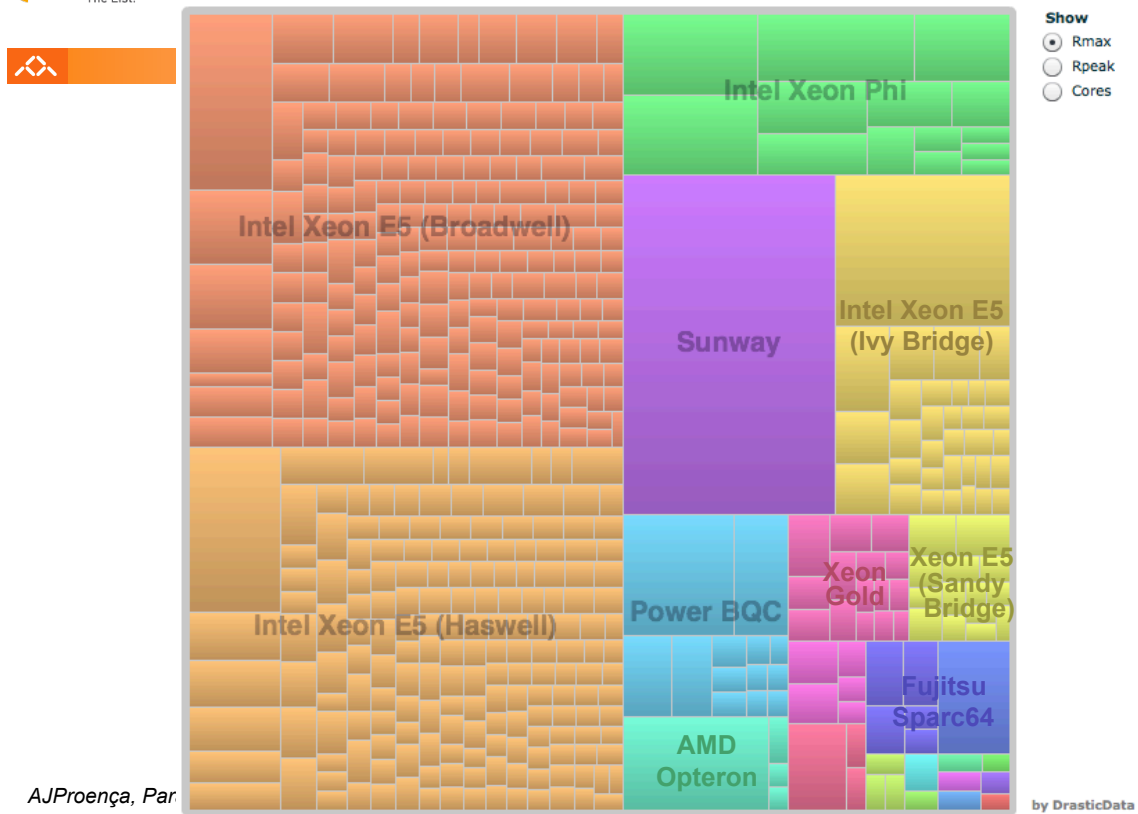
57



Past: processor family distribution of all systems



Processor generations in November'17



Accelerator family distribution over all systems Nov'17

