Master Informatics Eng.

2017/18

A.J.Proença

Data Parallelism 2 (*SIMD++, Intel MIC, NVidia GPU ...*)

(most slides are borrowed)

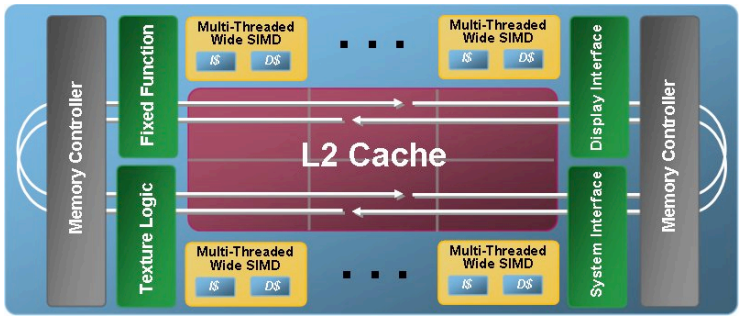
Beyond Vector/SIMD architectures

- 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
    - ...
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr**, PEZY-SC**)
    - ...
  - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
    - ...

Intel MIC: Many Integrated Core

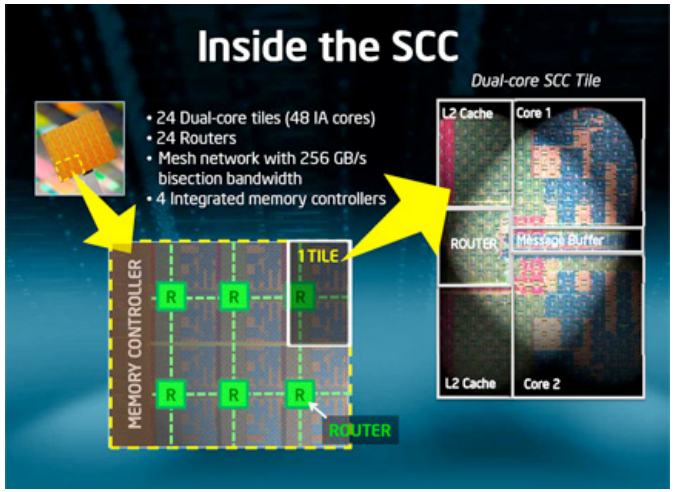
Intel evolution, from:

- Larrabee (80-core GPU)



& SCC

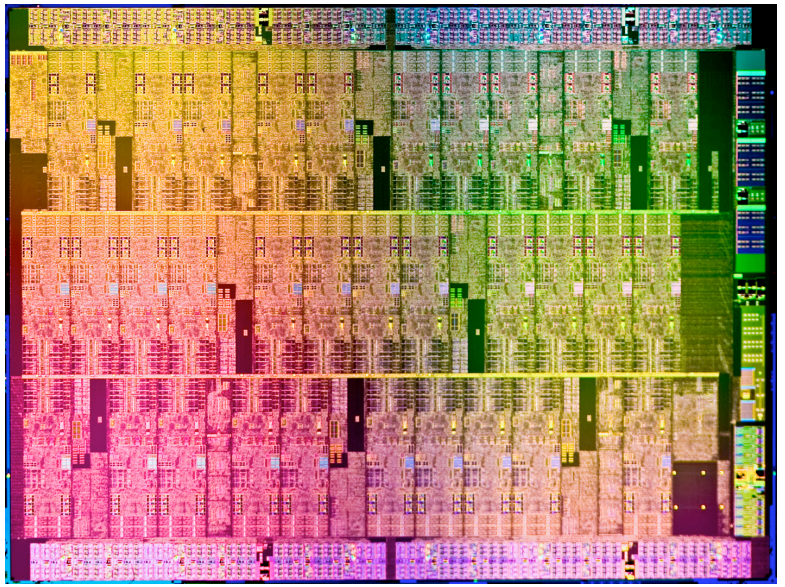
Single-chip
Cloud
Computer,
24x
dual-core tiles



to MIC:

- Knights Ferry (pre-production, Stampede)
- Knights Corner →

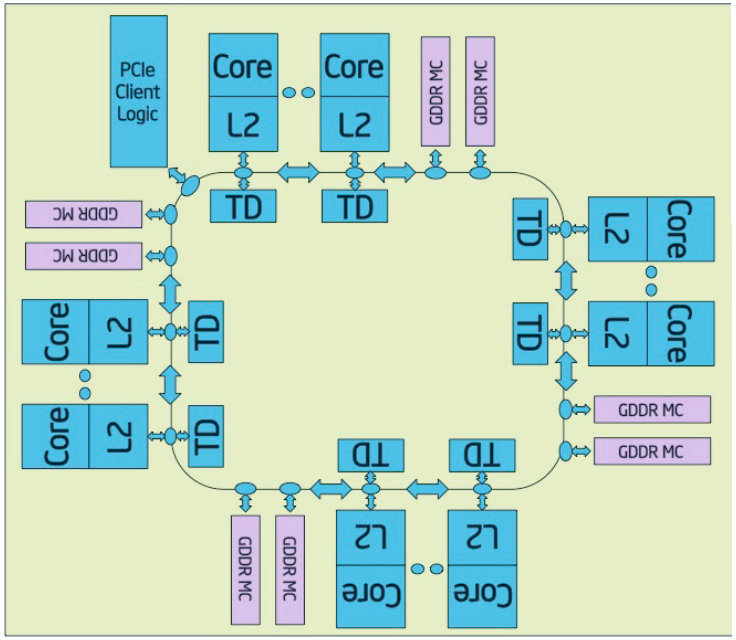
  Xeon Phi co-processor up to 61 Pentium cores

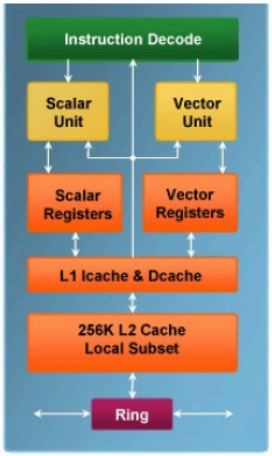


- Knights Landing & Knights Mill

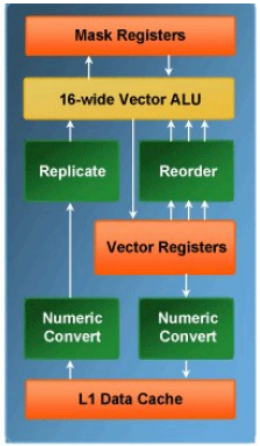
  Xeon Phi full processor up to 36x dual-core Atom tiles

Intel Knights Corner architecture

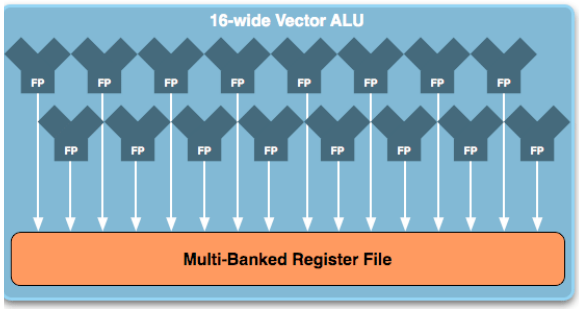


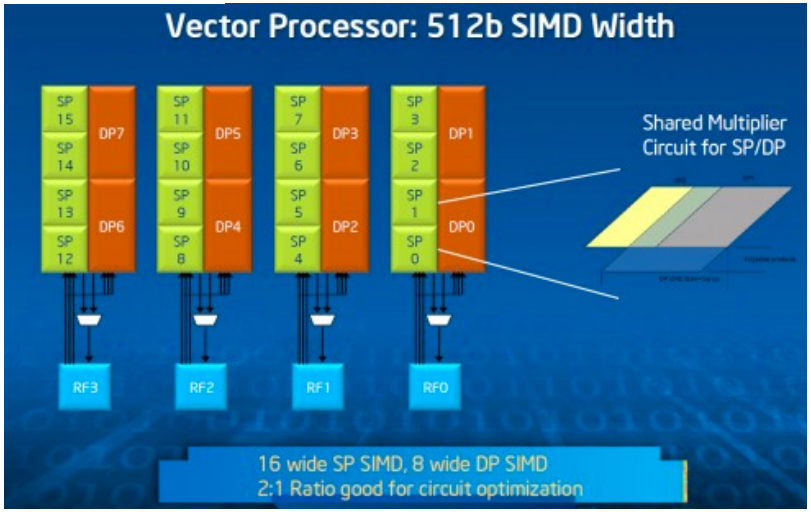


Processor Core

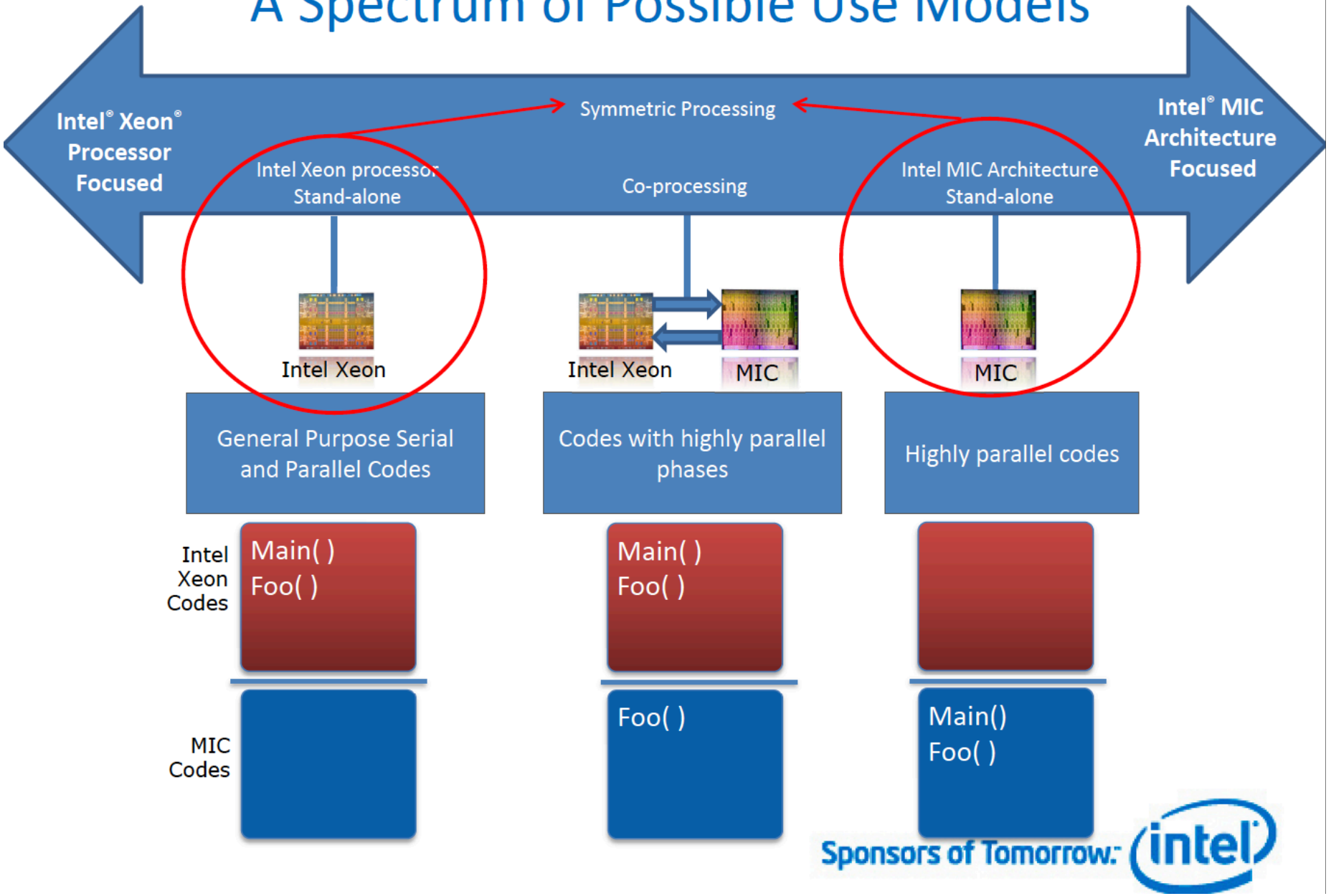


Vector Unit

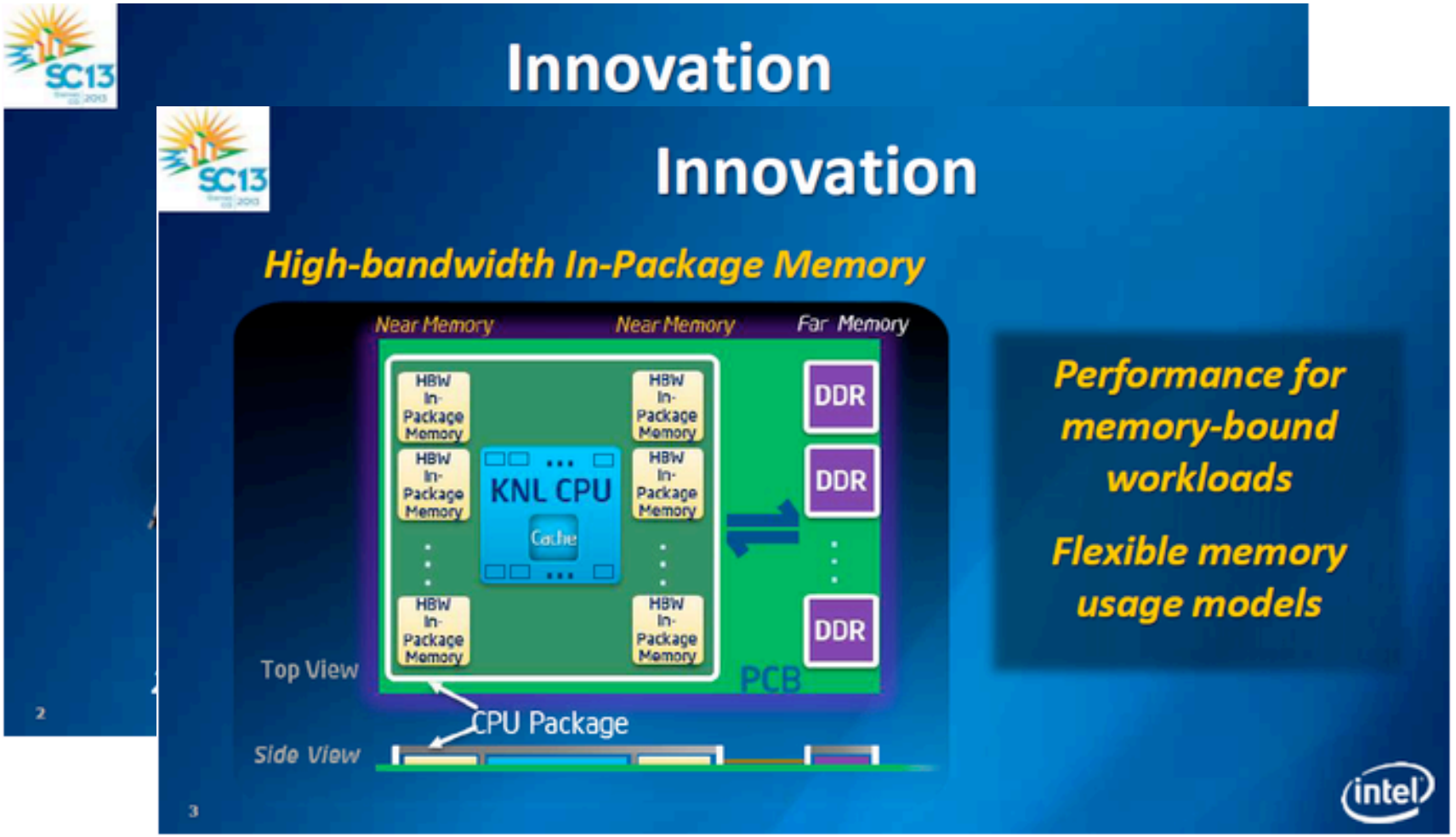




A Spectrum of Possible Use Models

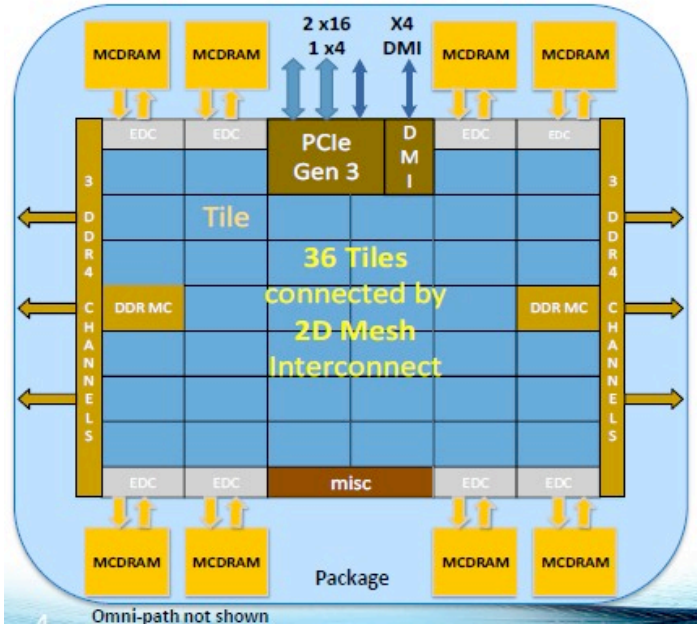


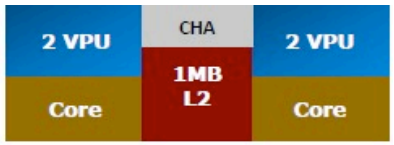
The new Knights Landing architecture



Intel Knights Landing in 2016: Xeon Phi com 72 cores

Knights Landing Overview



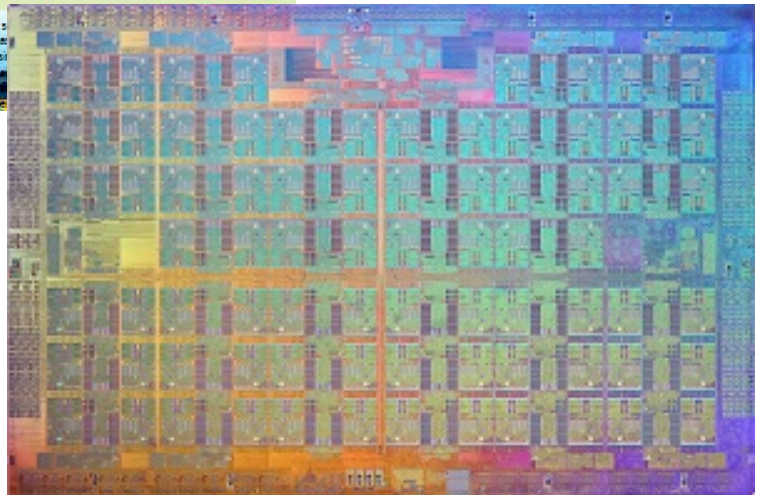


Chip: 36 Tiles interconnected by **2D Mesh**
Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW
DDR4: 6 channels @ 2400 up to 384GB
IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset
Node: 1-Socket only
Fabric: Omni-Path on-package (not shown)

Vector Peak Perf: 3+TF DP and 6+TF SP Flops
Scalar Perf: ~3x over Knights Corner
Streams Triad (GB/s): MCDRAM : 400+; DDR: 90+

Source Intel: All products, computer systems, dates and figures s
are subject to change without notice. KNL data are preliminary ba
without notice. 1Binary Compatible with Intel Xeon processors usi
numbers are based on STREAM-like memory access pattern whe
estimated based on internal Intel analysis and



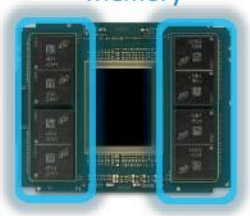
More details in a later set of slides...

Intel Knights Mill expected in 2018: similar to KNL, but...

Knights Mill SOC

Host CPU for Highly-Parallel Apps

Integrated Memory



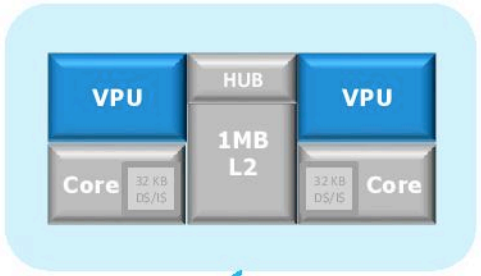
6 channels of up to DDR4 2400
(High capacity, up to 384GB)

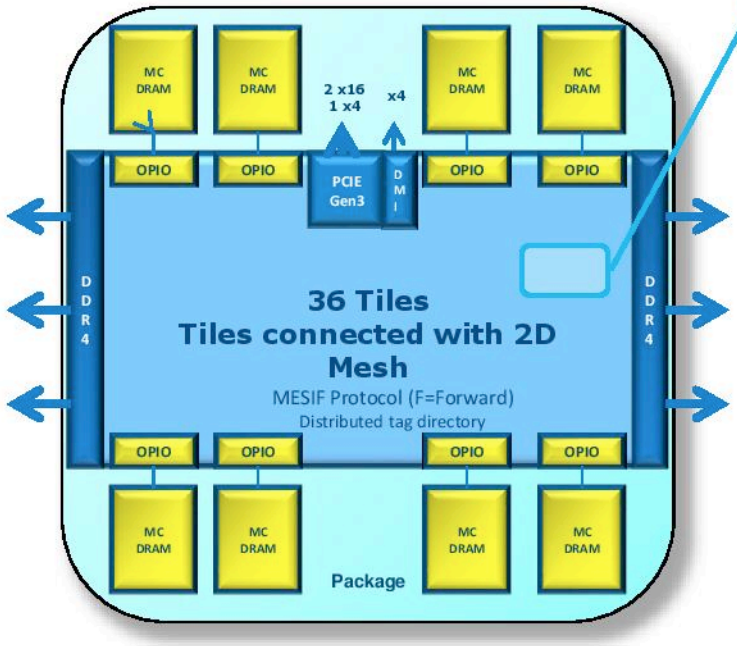
16GB of IPM (MCDRAM)
(High memory bandwidth)

36 lanes PCIE Gen3
(High IO performance)

1 MB L2 per tile
2 cores per tile

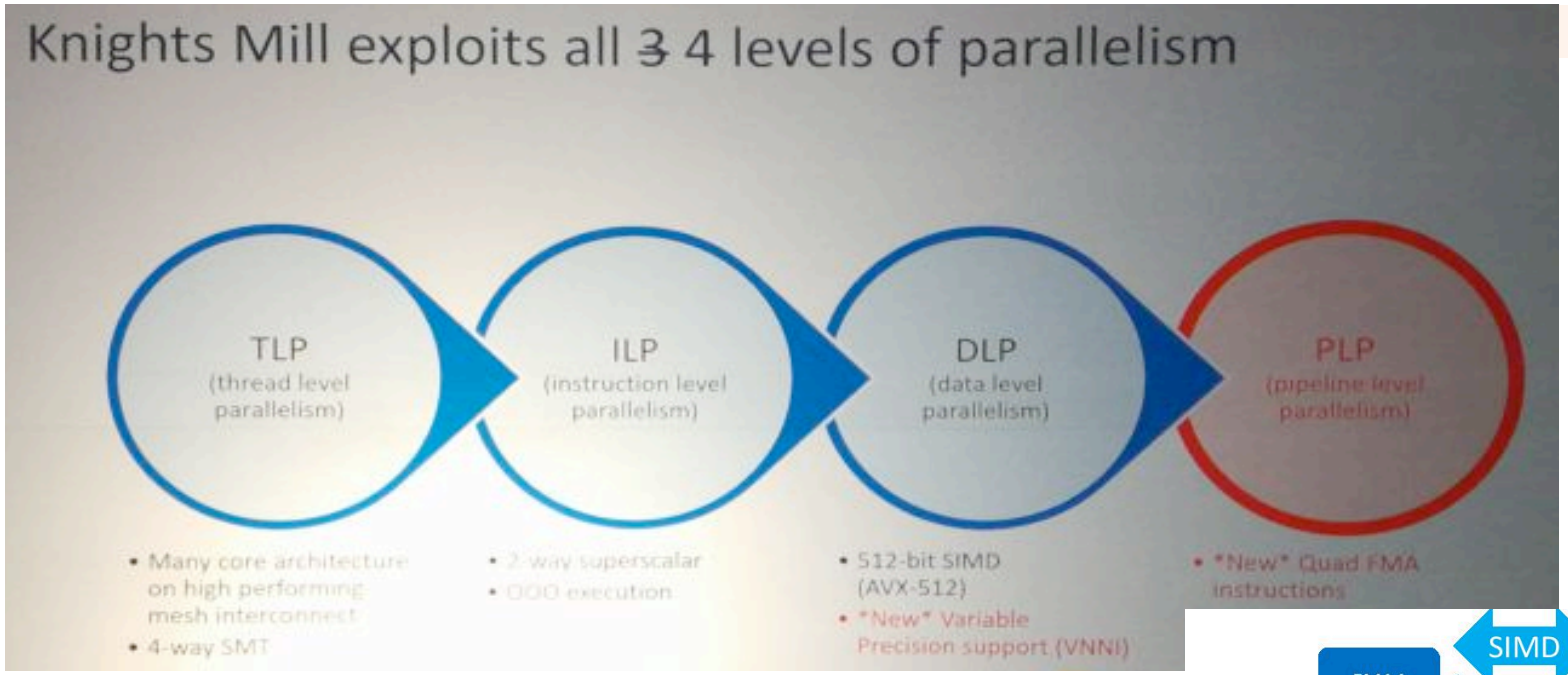
AVX512-F (512b SIMD)
16 DP flops/VPU
128 SP flops/VPU
256 VP(*) ops/VPU



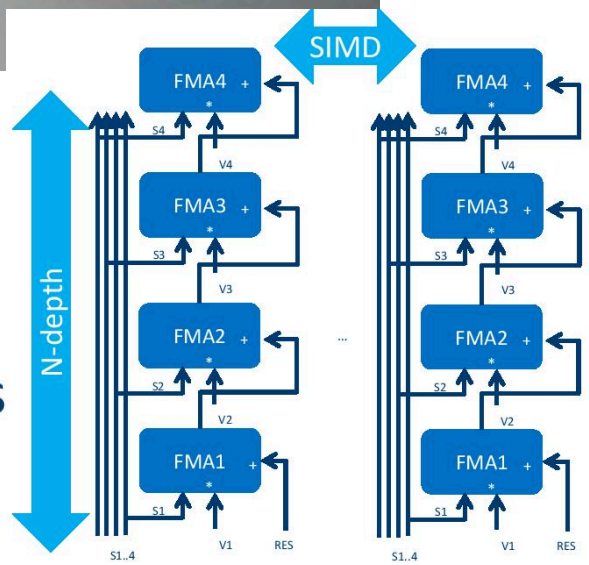


(*) Variable precision

Intel Knights Mill expected in 2018: similar to KNL, but...



Quad FMA FP32 Instructions



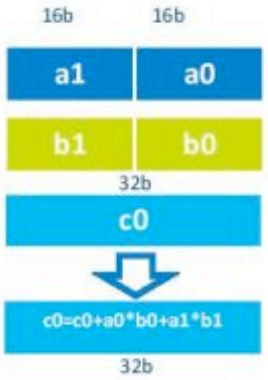
Intel Knights Mill expected in 2018: similar to KNL, but...

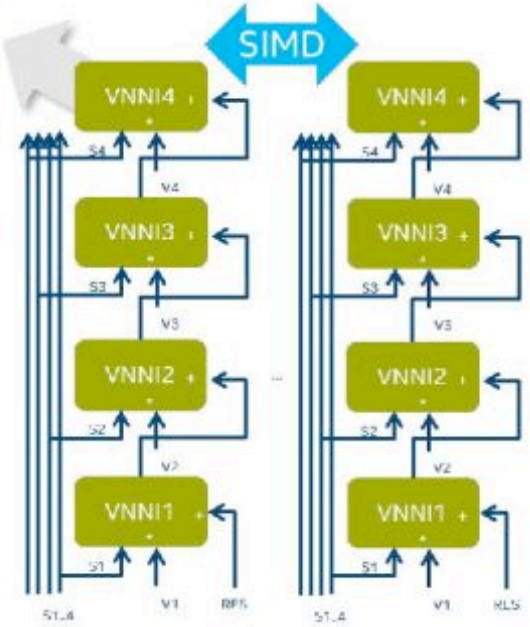
Variable Precision: What is VNNI-16?

- Vector Neural Network Instructions
- Variable precision
  - ***Inputs***: 16-bit INT
  - ***Outputs***: 32-bit INT
  - ***Semantics***: 2 x int16 multiplies horizontally accumulating into single 32-bit output
- Variable precision is best of both worlds
  - **Same operations/instruction as 'half precision'**
    - 2x OPS vs Single Precision
  - **Similar output precision for optimal training convergence**
    - 31 bits of INT32 vs 24 bits of mantissa in FP32
  - The obvious trade-off is the associated overhead on handling dynamic range in software (fixed precision)

Intel Knights Mill expected in 2018: similar to KNL, but...

QVNNI = QFMA + VNNI





Instruction	Format	Description
VP4DPWSSD	zmm1 {k1}, zmm2+3, mem128	Quadruple INT16 to INT32 horizontal MAC
VP4DPWSSDS	zmm1 {k1}, zmm2+3, mem128	Quadruple INT16 to INT32 horizontal MAC with signed saturation

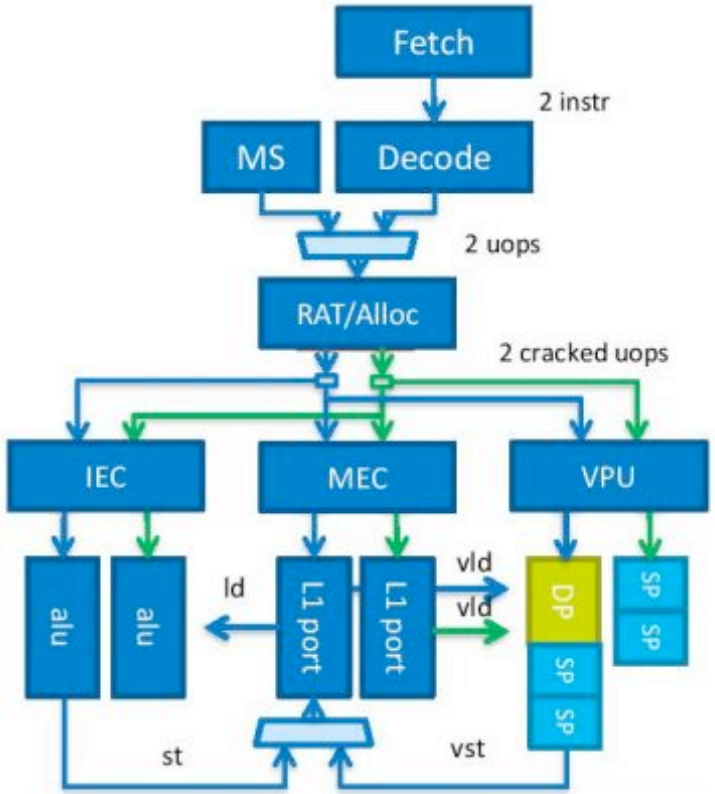
- Example
  - VP4DPWSSD zmm4 {k1}, zmm0+3, m128
    - for i=0..15
      - zmm4.int32[i] = zmm4.int32[i]
        + (zmm0.int16[2*i]*m128.int16[0] + zmm0.int16[2*i+1]*m128.int16[1])
        + (zmm1.int16[2*i]*m128.int16[2] + zmm1.int16[2*i+1]*m128.int16[3])
        + (zmm2.int16[2*i]*m128.int16[4] + zmm2.int16[2*i+1]*m128.int16[5])
        + (zmm3.int16[2*i]*m128.int16[6] + zmm3.int16[2*i+1]*m128.int16[7])

Intel Knights Mill expected in 2018: similar to KNL, but...

Knights Mill Core

Enhanced Knights Landing core

- 2-way, OOO execution
- 4-way SMT
- 1 MB L2 bandwidth (64 bytes/cyc)
- 46 PA bits, 48 VA bits
- 2x 512b loads, 1x 512b store
- 32 KB D-cache (8-way),
  32 KB I-cache (8-way)
- 72 inflight uops
- *RS sizes:* IEC (2 x 12), MEC (12), VPU (2 x 20)
- 1st level uTLB: 64 entries
  2nd level dTLB:
  256 4K, 128 2M, 16 1G pages



ISA: SSE, AVX, AVX512-F

DP stack

- 1 VPU port/core
- 1x 16 DP flops per cycle
- 6 cycles of latency

SP/VNNI stack

- 2 VPU ports/core
- 2 stacked FMAs per port
- 2x 64 SP flops per cycle
- 2x 128 VP ops/cycle
- 3+3 cycles of latency

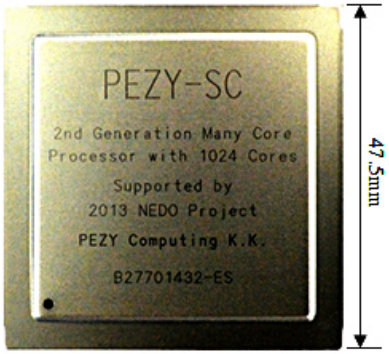
intel | 10

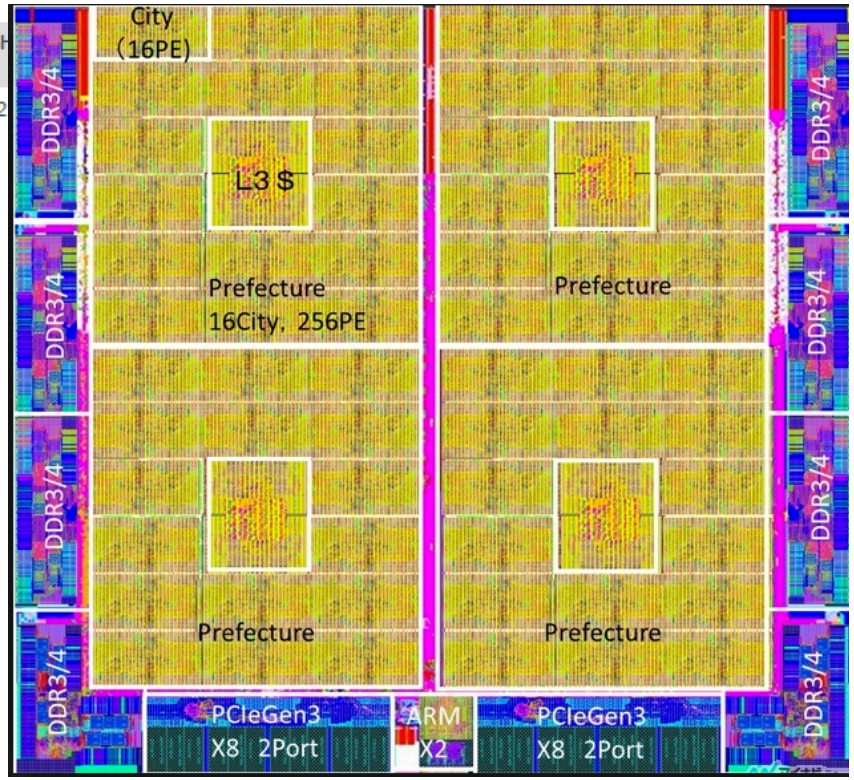
PEZY-SC: Peta_Exa_Zetta_Yotta-SuperComputer: a 1024-core many-core processor chip

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1 (94)	6,673.84	Advanced Center for Computing and Communication, RIKEN	Shoubu - ZettaScaler-1.6, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SCnp	149.99
2 (486)	6,195.22	Computational Astrophysics Laboratory, RIKEN	Satsuki - ZettaScaler-1.6, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SCnp	46.89
3 (1)	6,051.30	National Supercomputing Center in Wuxi	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	15,371.00
4 (440)	5,272.09	GSI Helmholtz Center	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
5 (446)	4,778.46	Institute of Modern Physics (IMP), Chinese Academy of Sciences	Sugon Cluster W780I, Xeon E5-2640v3 8C 2.6GH QDR, NVIDIA Tesla K80	
6 (122)	4,112.11	Stanford Research Computing Center	XStream - Cray CS-Storm, Intel Xeon E5-2680v2 Infiniband FDR, Nvidia K80	

Top500 Rank

Green500 list
June'2016





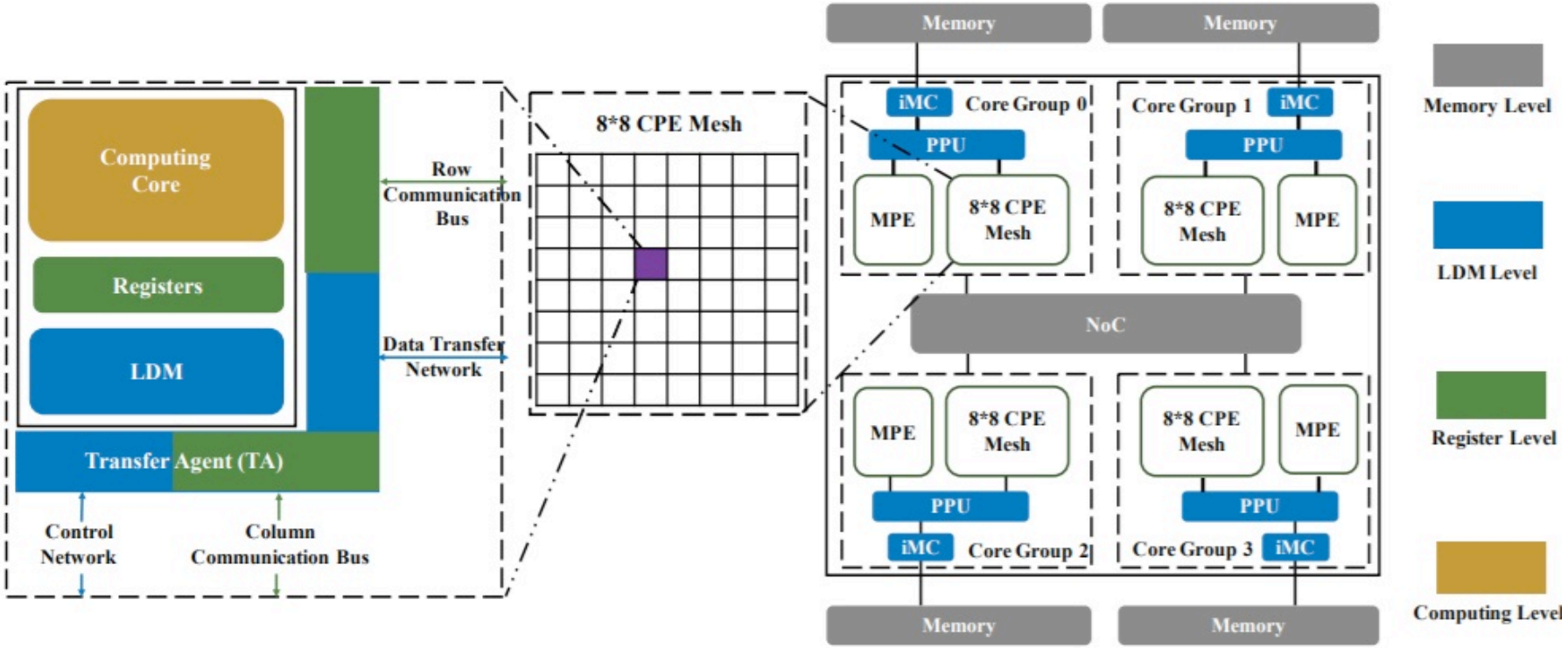
Beyond Vector/SIMD architectures

- 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: **ShenWei** 260
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
    - ...
  - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
    - ...

Sunway, or ShenWei, chip (Chinese:神威)

SW26010, 260 cores (in Sunway TaihuLight, #1 in TOP500 since June'16)

- 4x management cores (MPE) + 4x computer clusters (CPE)
- each CPE: 4x 64-core 64-bit RISC processors
- each core: only cache L1 & w/ **256-bit vector** instructions
- next generation: SW52010, with 8x MPE-cores and 8x 64-cores CPE meshes

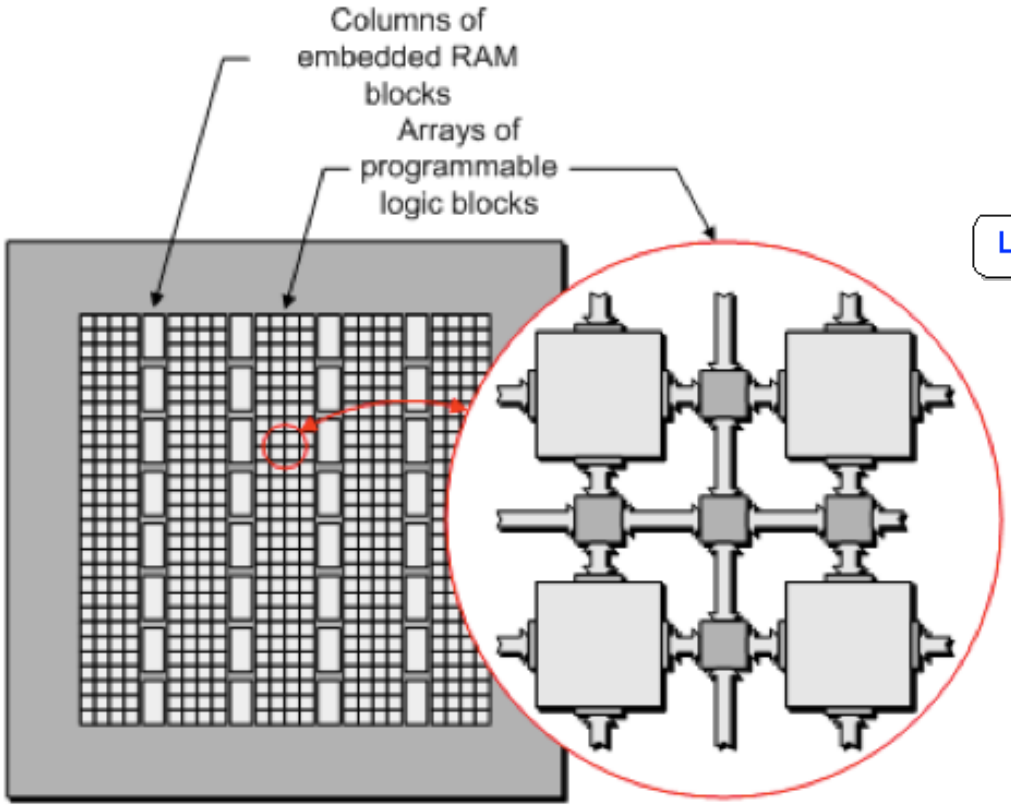


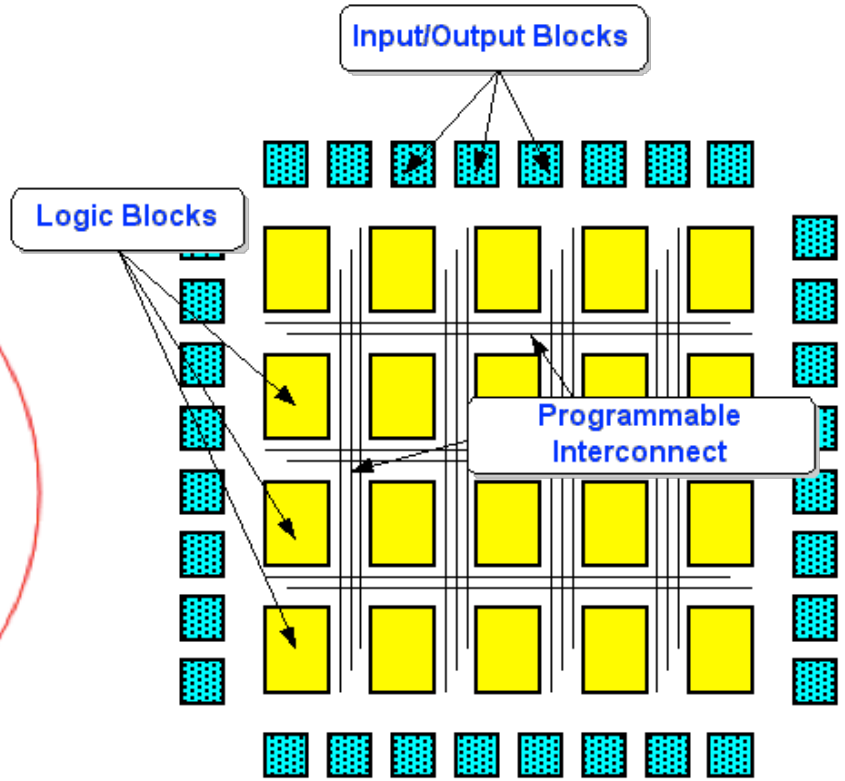
Beyond Vector/SIMD architectures

- 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: **ShenWei** 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**
    - ...
  - **heterogeneous PUs in a SoC: multicore PUs with GPU-cores**
    - ...

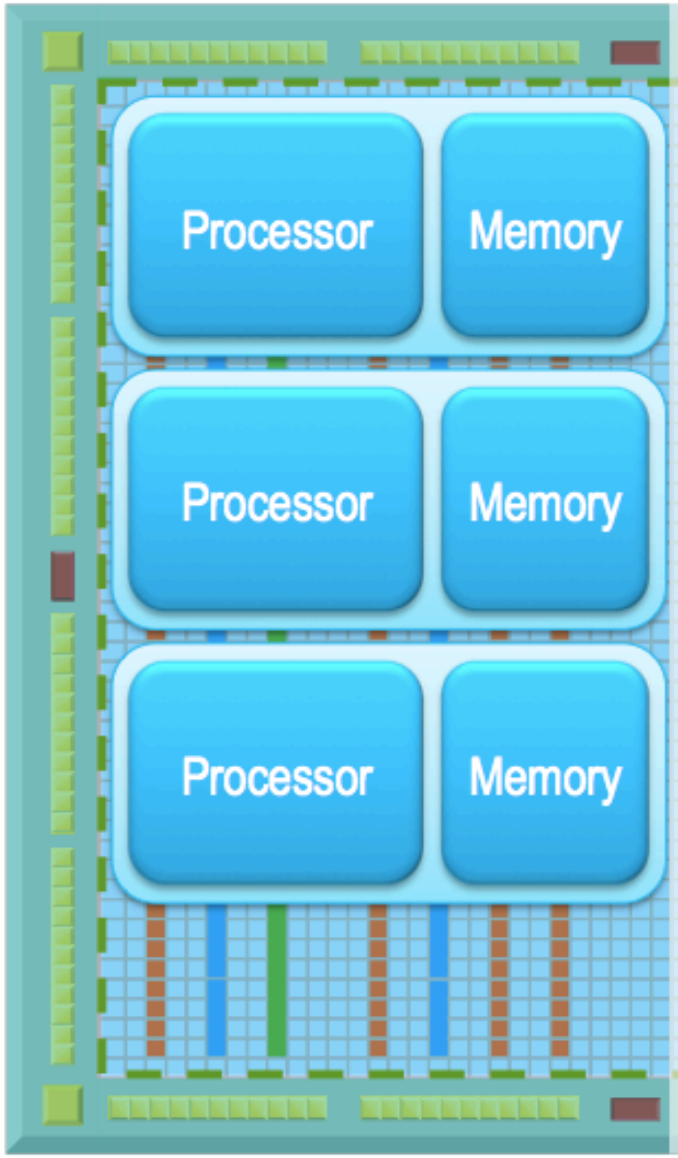
Field-Programmable Gate Arrays (FPGA)

A fabric with 1000s of simple configurable logic cells with LUTs, on-chip SRAM, configurable routing and I/O cells



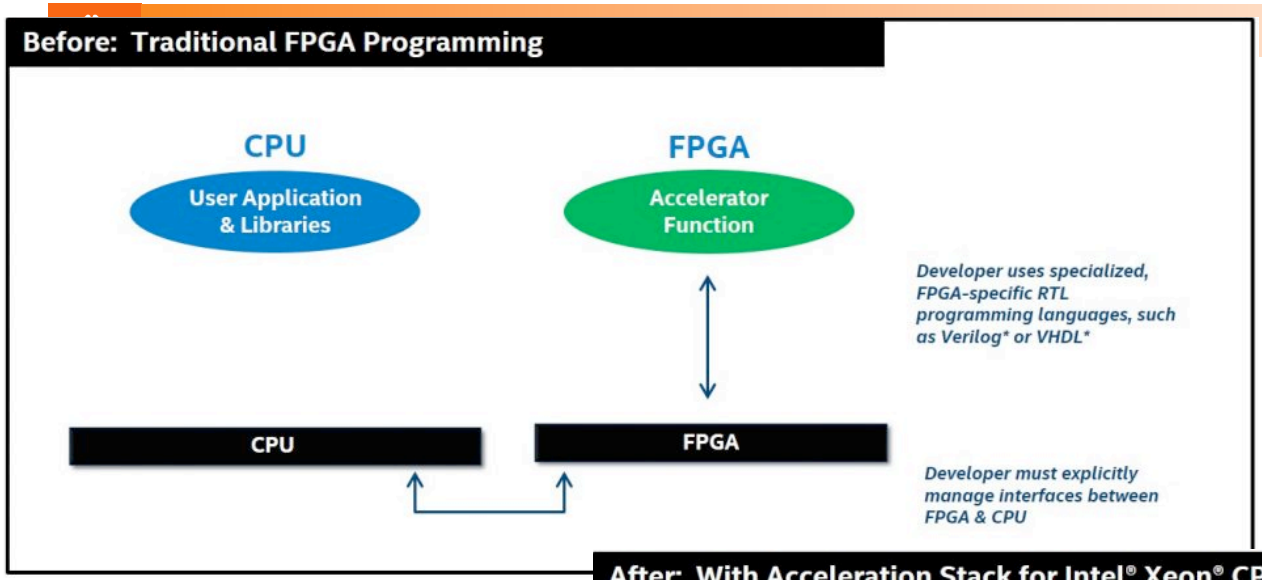


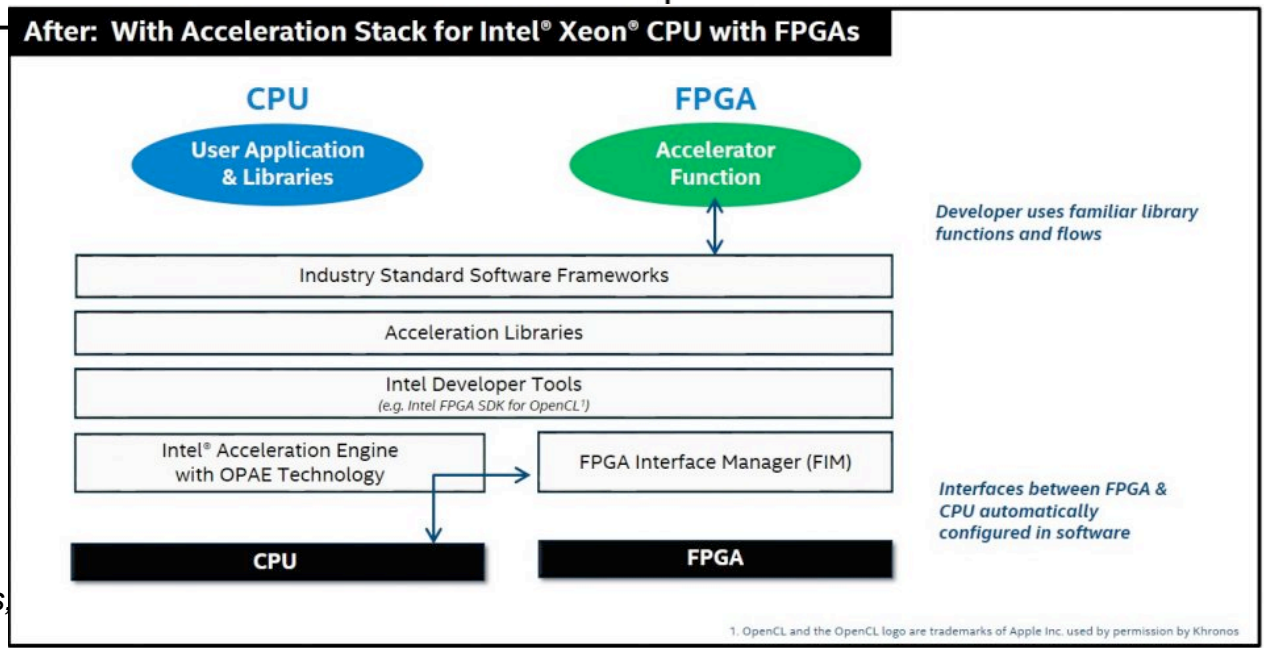
FPGA as a multiple configurable ISA



- Many coarse-grained processors
  - Different Implementation Options
    - Small soft scalar processor
    - or Larger vector processor
    - or Customized hardware pipeline
  - Each with local memory
- Each processor can exploit the fine grained parallelism of the FPGA to more efficiently implement it's "program"
- Possibly heterogeneous
  - Optimized for different tasks
- Customizable to suit the needs of a particular application

FPGA as a computing accelerator





https://builders.intel.com/blog/fpga-in-the-data-center-programming-for-all/

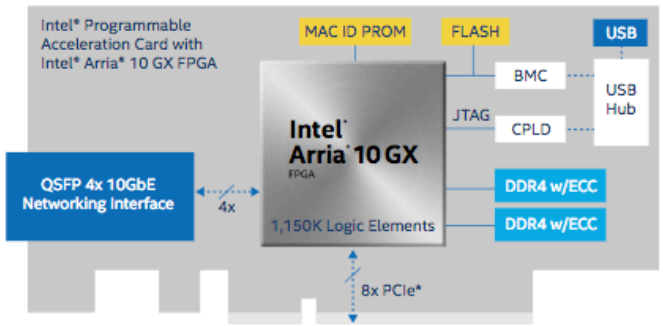
The Intel Programmable Acceleration Card

PRODUCT BRIEF

Intel® Programmable Acceleration Card (Intel® PAC) with Intel® Arria® 10 GX FPGAs

Introduction

This PCIe-based FPGA acceleration card for data centers offers both inline and lookaside acceleration. It provides the performance and versatility of FPGA acceleration and is one of several platforms supported by the Acceleration Stack for Intel® Xeon® CPUs with FPGAs. This acceleration stack provides a common developer interface for both application and accelerator function developers, and includes drivers, application programming interfaces (APIs), and an FPGA interface manager. Together with acceleration libraries and development tools, the acceleration stack saves developer's time and enables code re-use across multiple Intel FPGA platforms. The card can be deployed in a variety of servers with its low-profile form factor, low-power dissipation, and passive heat sink.

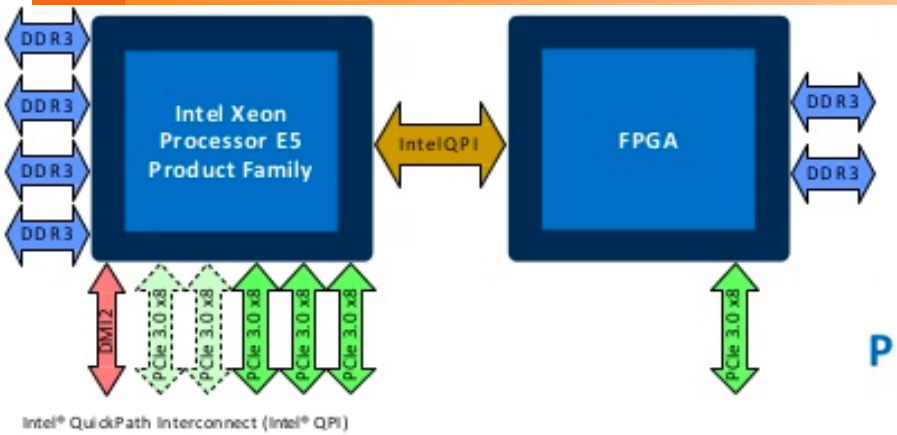


Targeted Workloads

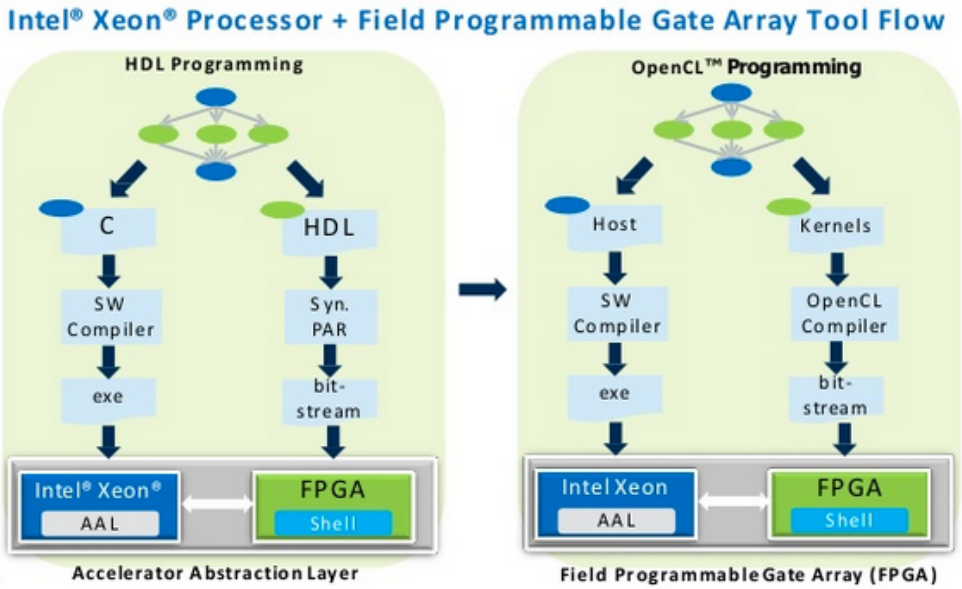
- Big data analytics
- Artificial intelligence
- Video transcoding
- Cyber security
- High-performance computing (HPC), such as genomics and oil and gas
- Financial technology, or FinTech

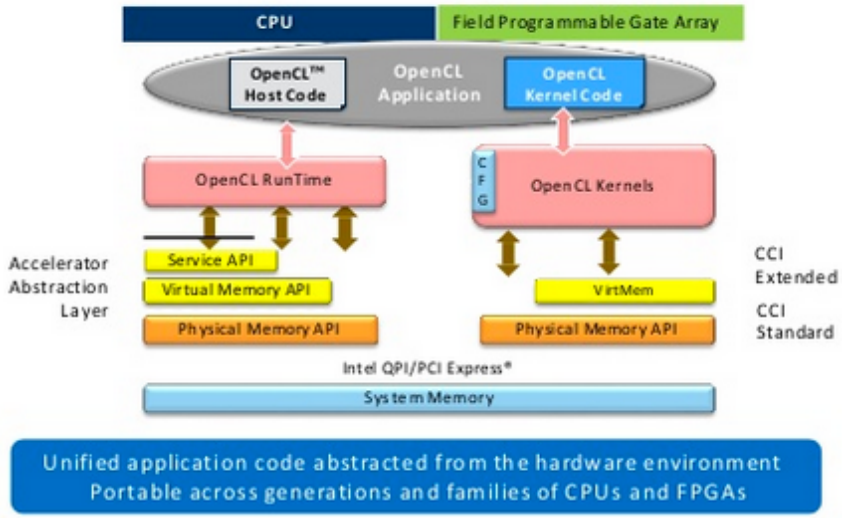
https://builders.intel.com/blog/fpga-in-the-data-center-programming-for-all/

Faster integration of programmable acceleration cards at Intel



Intel® QuickPath Interconnect (Intel® QPI)





https://www.slideshare.net/insideHPC/using-xeon-fpga-for-accelerating-hpc-workloads

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: **ShenWei** 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**
    - …

Graphical Processing Units

- 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*"

cores/processing elements in several devices

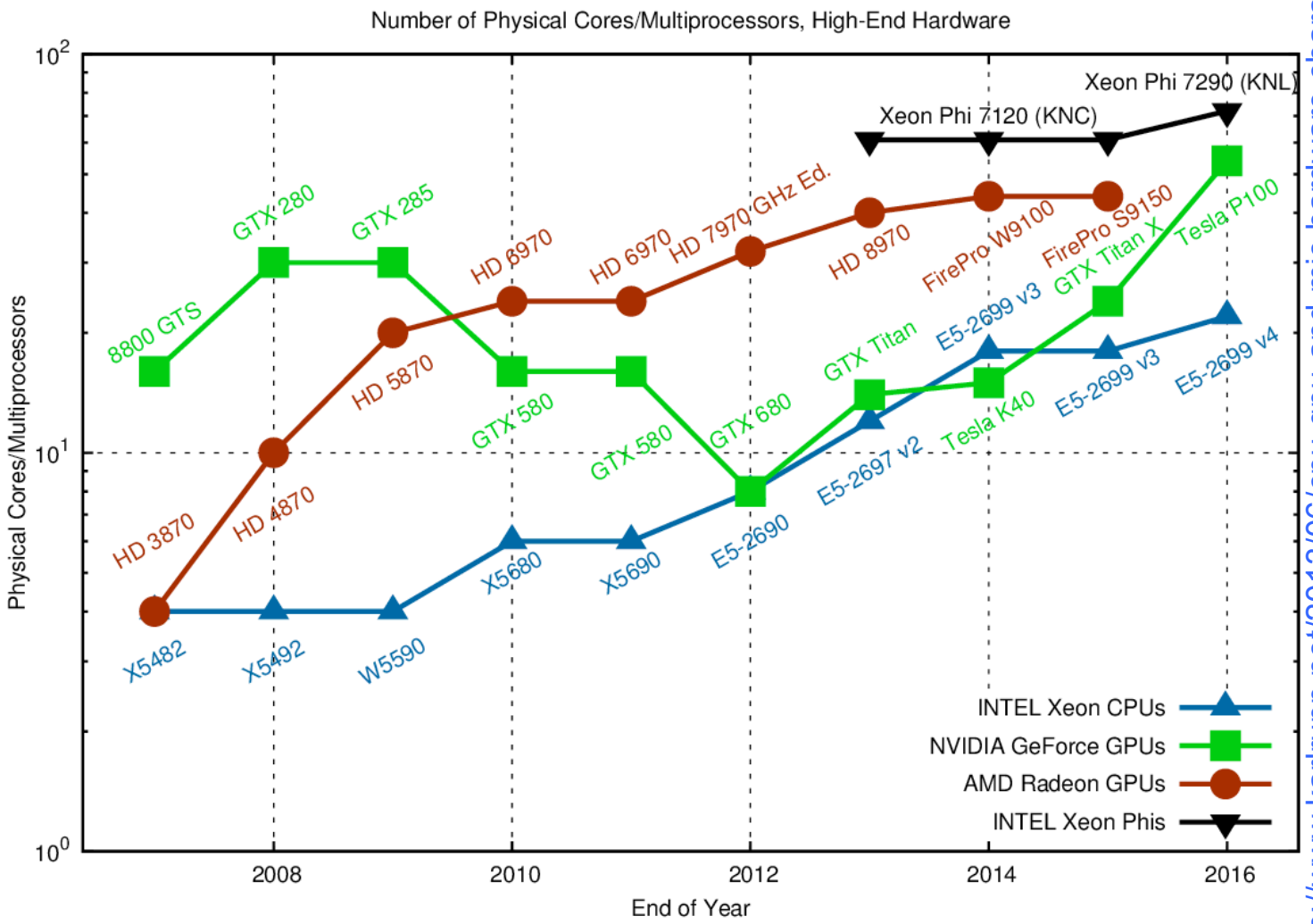
Key question: what is a **core**?

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

b) A SIMD processor?
CPU-type..

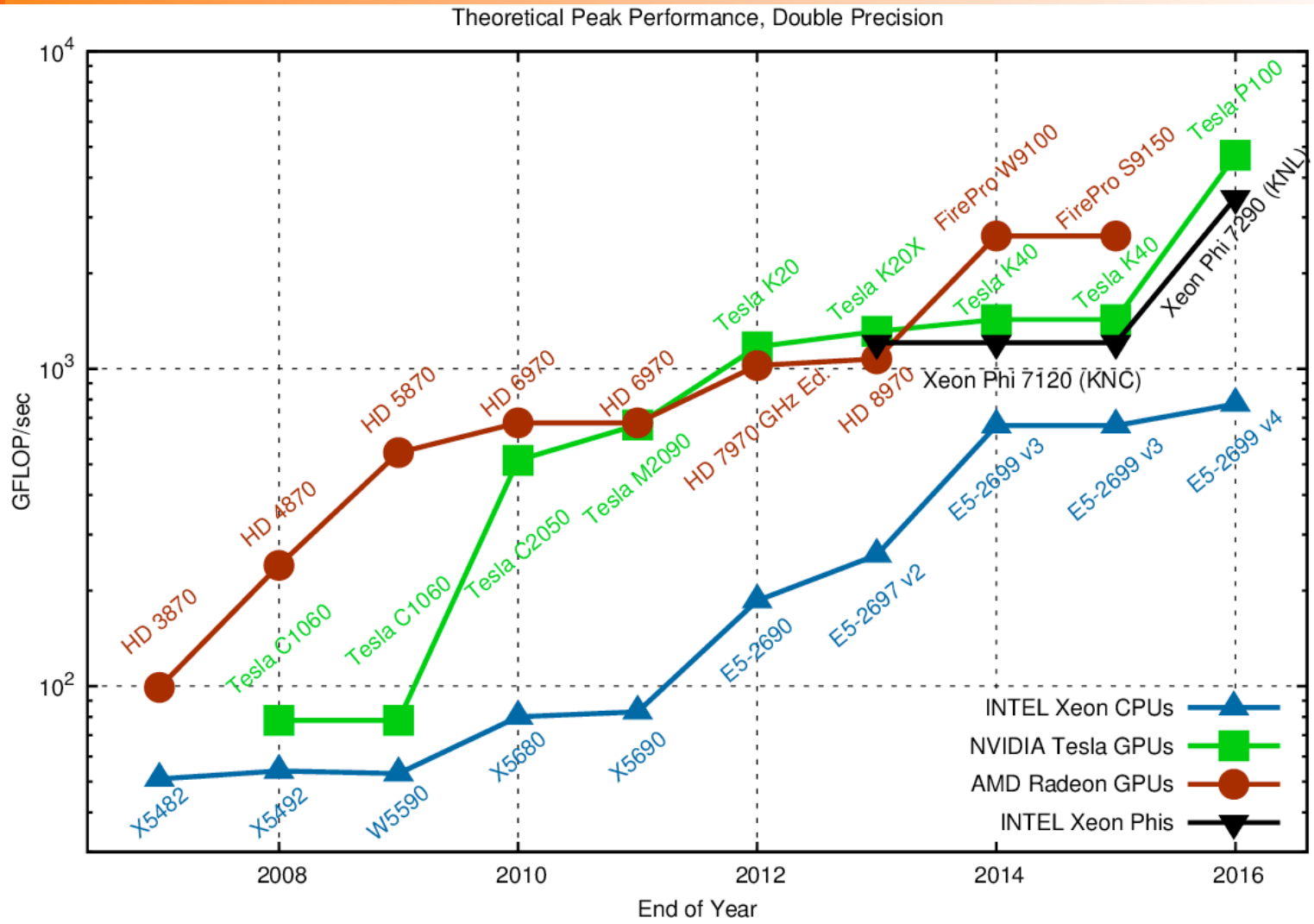
This updated slide and in this course:
- **b)**

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



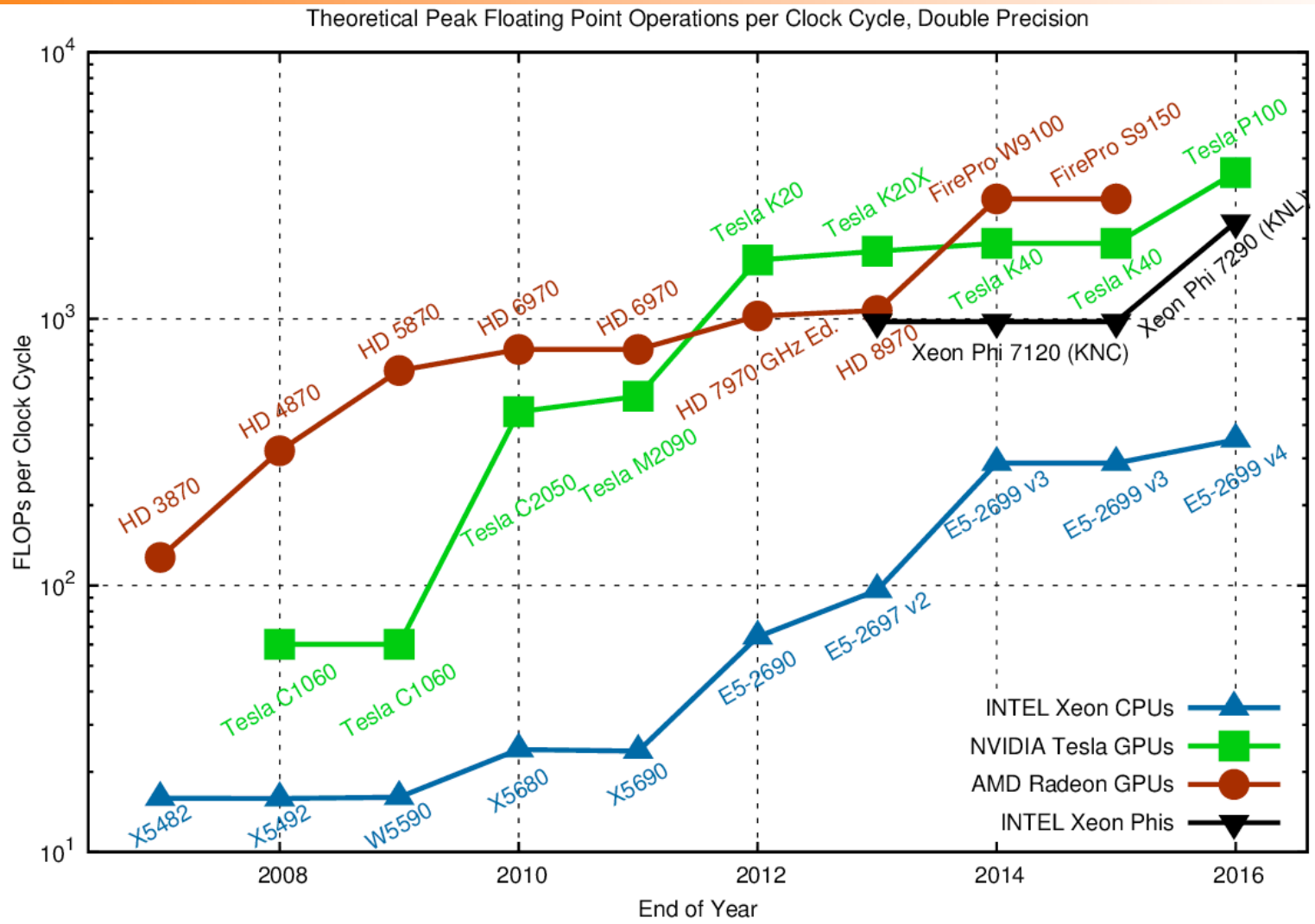
http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

Theoretical peak performance in several computing devices (DP)



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)



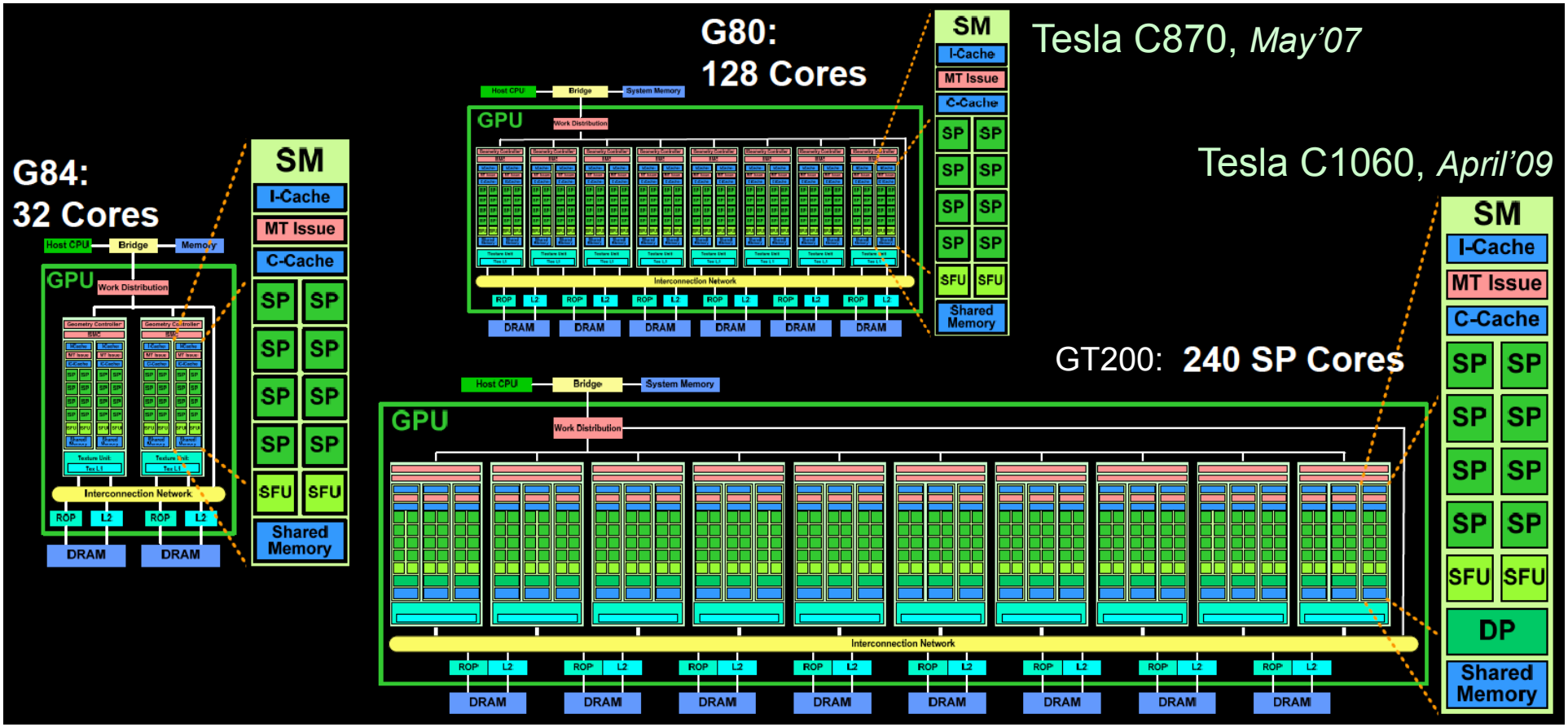
http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

NVIDIA GPU Architecture

- Similarities to vector machines:
  - Works well with data-level parallel problems
  - Scatter-gather transfers
  - Mask registers
  - Large register files

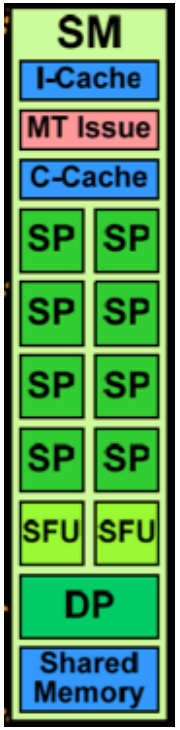
- Differences:
  - No scalar processor
  - Uses multithreading to hide memory latency
  - Has many functional units, as opposed to a few deeply pipelined units like a vector processor

Early NVidia GPU Computing Modules



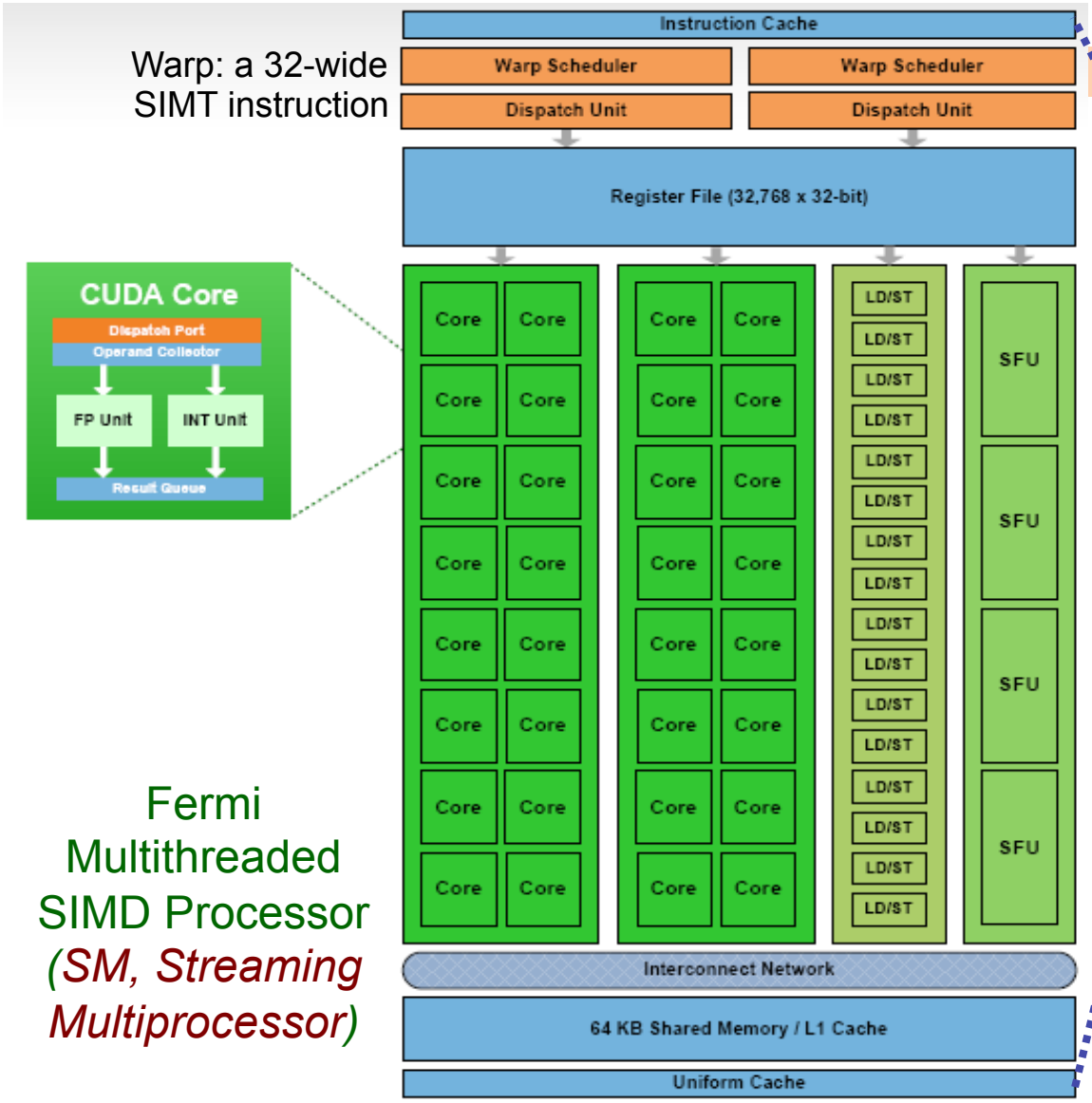
NVIDIA GPU Memory Structures

- 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

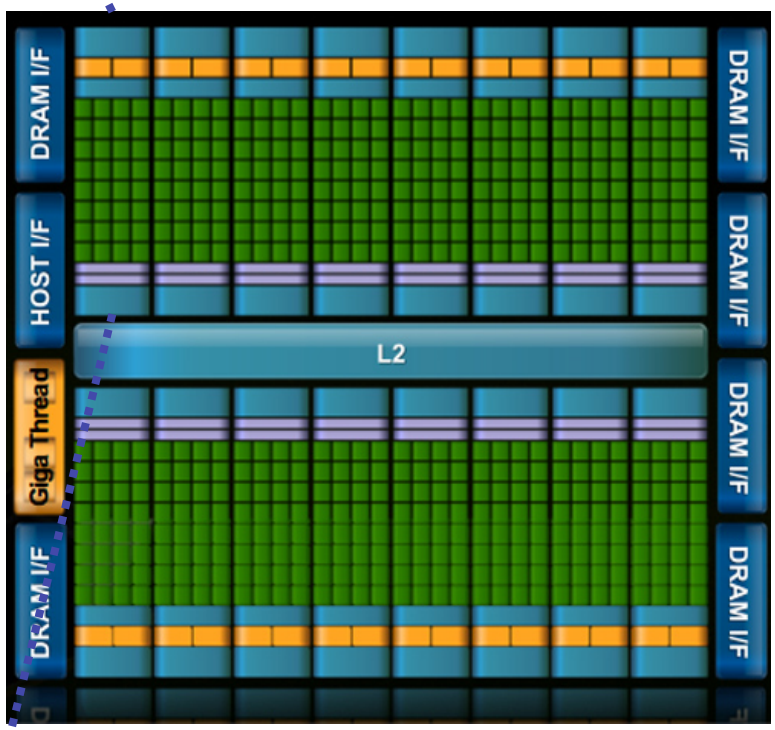


The NVidia Fermi architecture

Warp: a 32-wide SIMT instruction



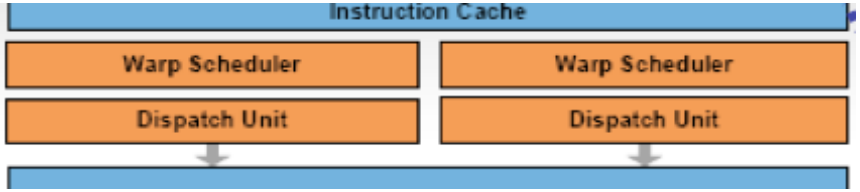
Fermi Multithreaded SIMD Processor *(SM, Streaming Multiprocessor)*



Fermi Architecture:
GF110: 512 CUDA-cores
July'11

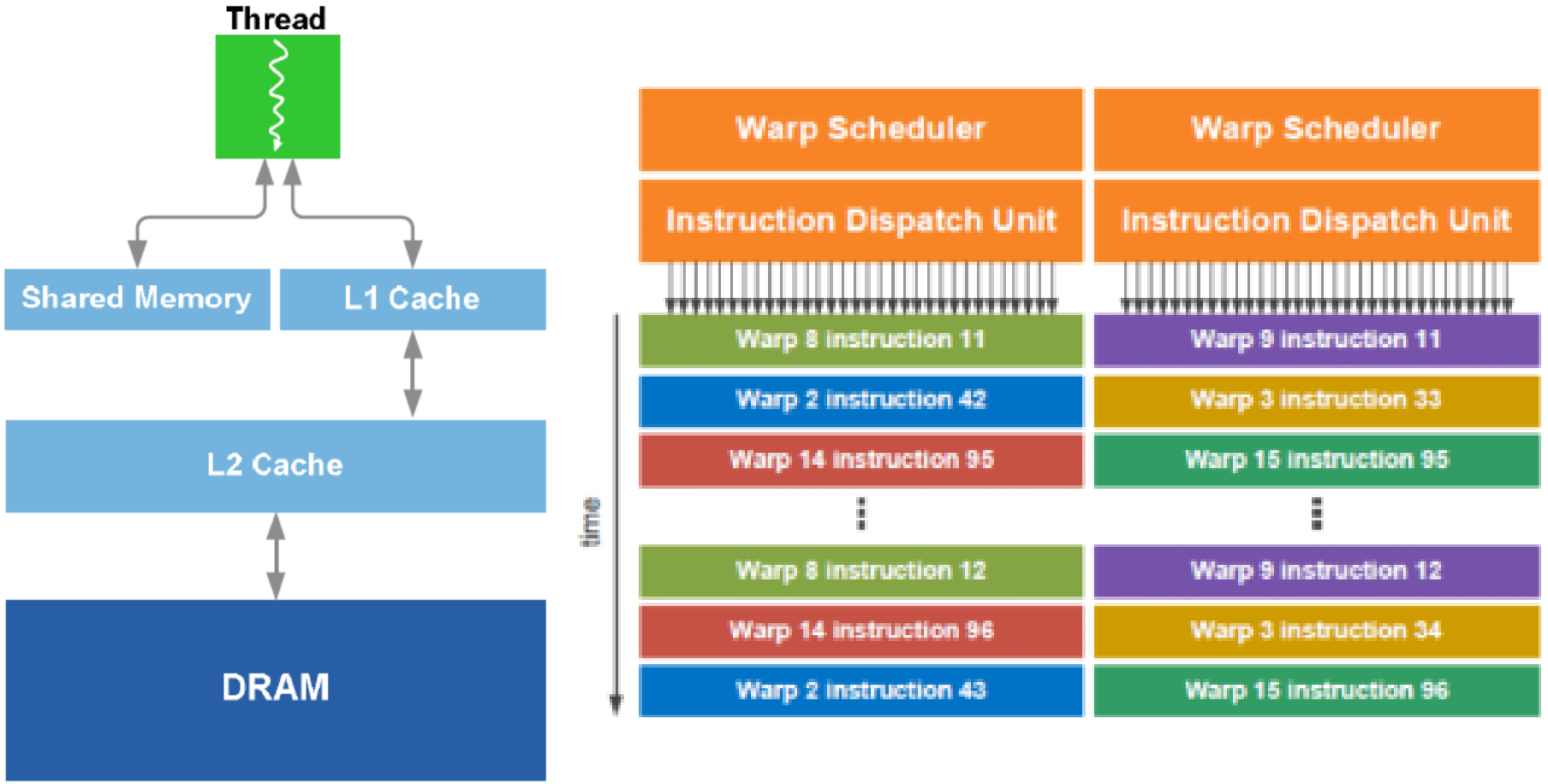
Fermi Architecture Innovations

- 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

Fermi: Multithreading and Memory Hierarchy

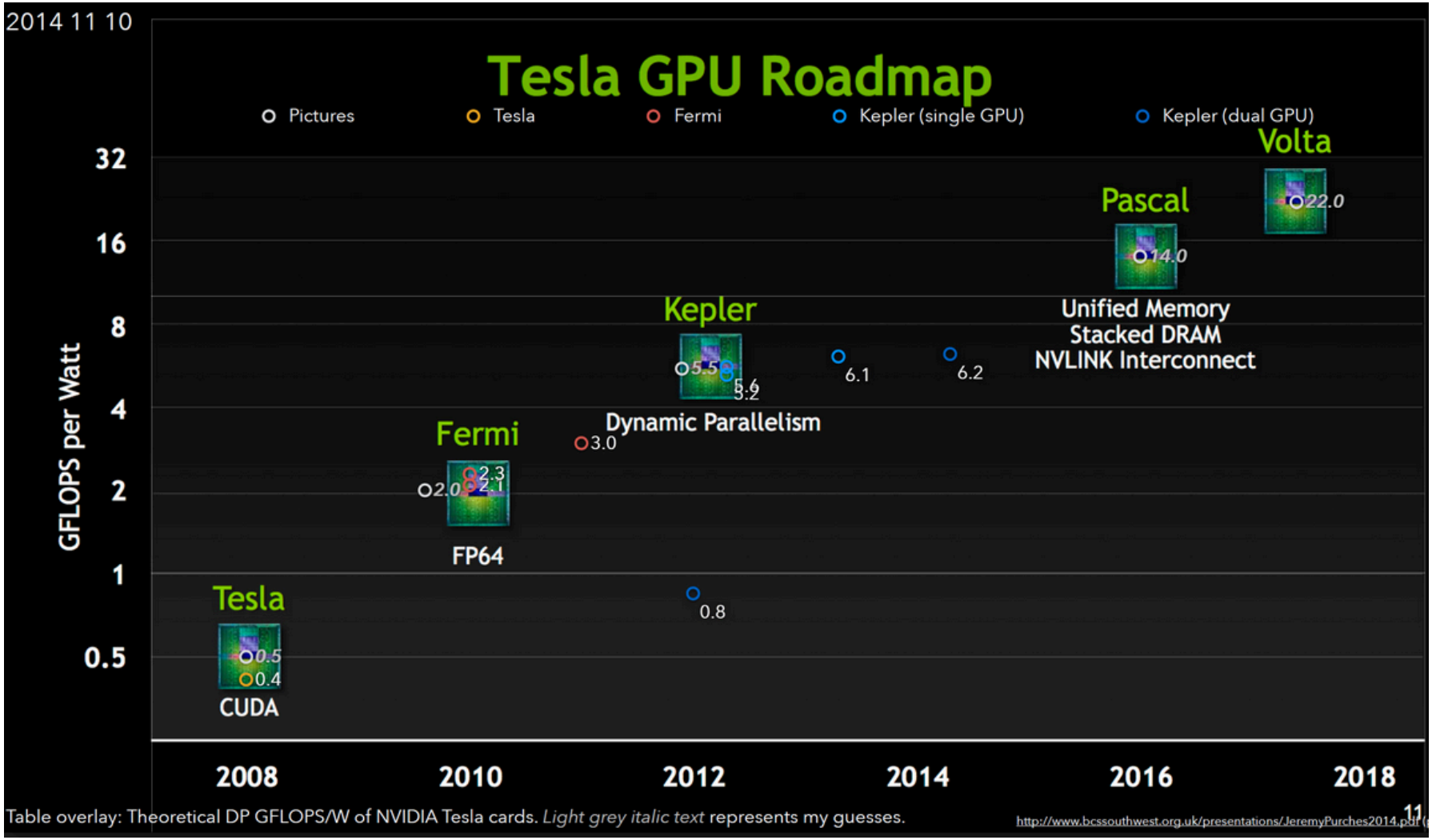


TOP500 list in November 2010: 3 systems in the top4 use Fermi GPUs

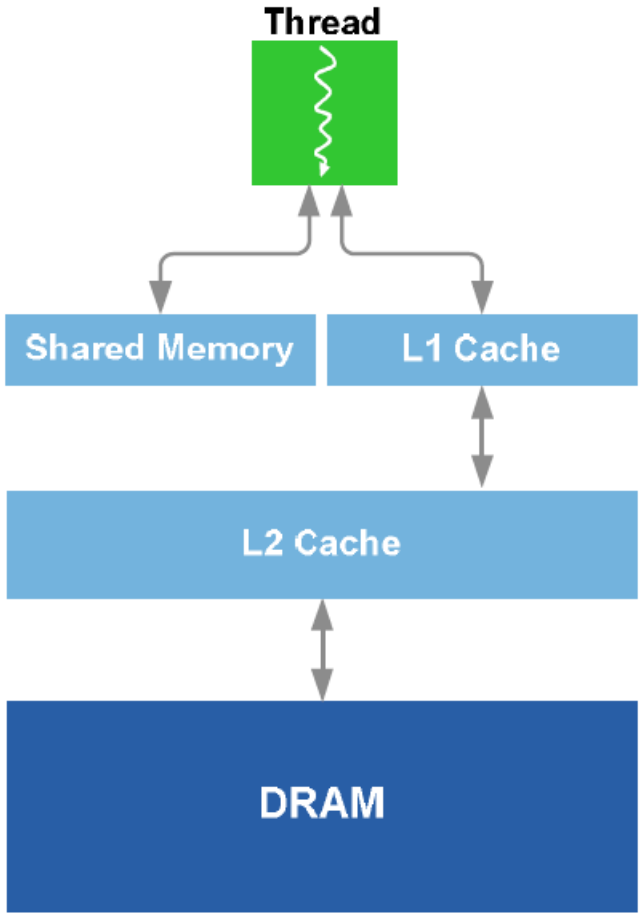
HIGHLIGHTS: NOVEMBER 2010

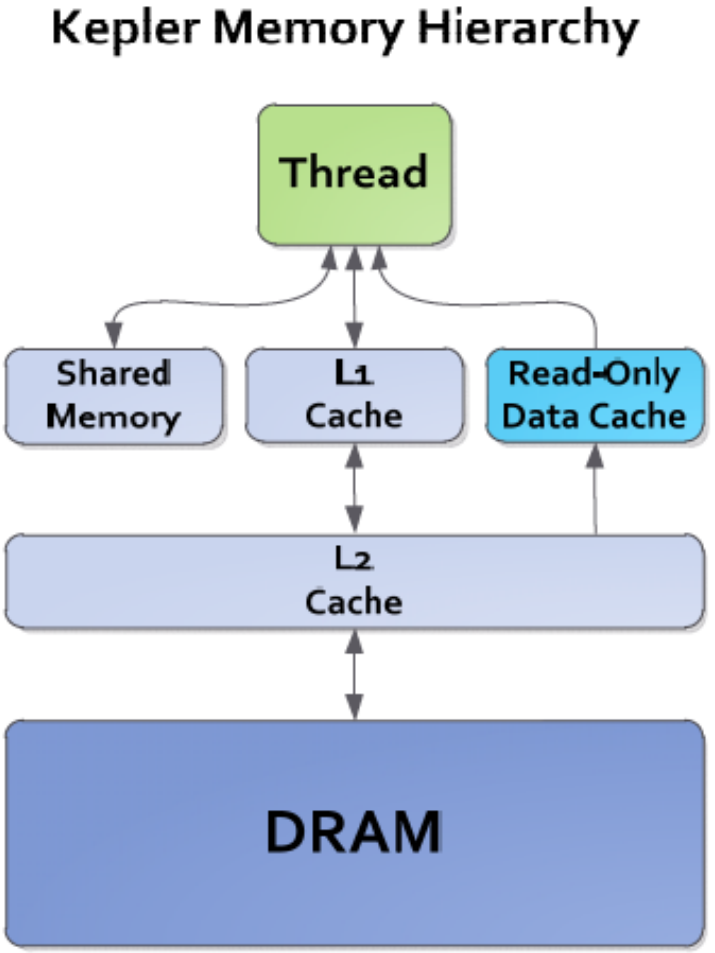
- The Chinese Tianhe-1A system is the new No. 1 on the TOP500 and clearly in the lead with 2.57 petaflop/s performance.
- No. 3 is also a Chinese system called Nebulae, built from a Dawning TC3600 Blade system with Intel X5650 processors and NVIDIA Tesla C2050 GPUs
- There are seven petaflop/s systems 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.

Families in NVidia Tesla GPUs

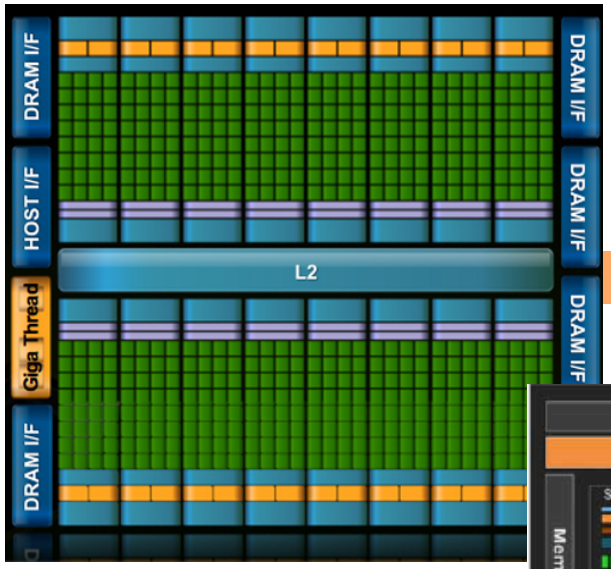


From Fermi into Kepler: The Memory Hierarchy





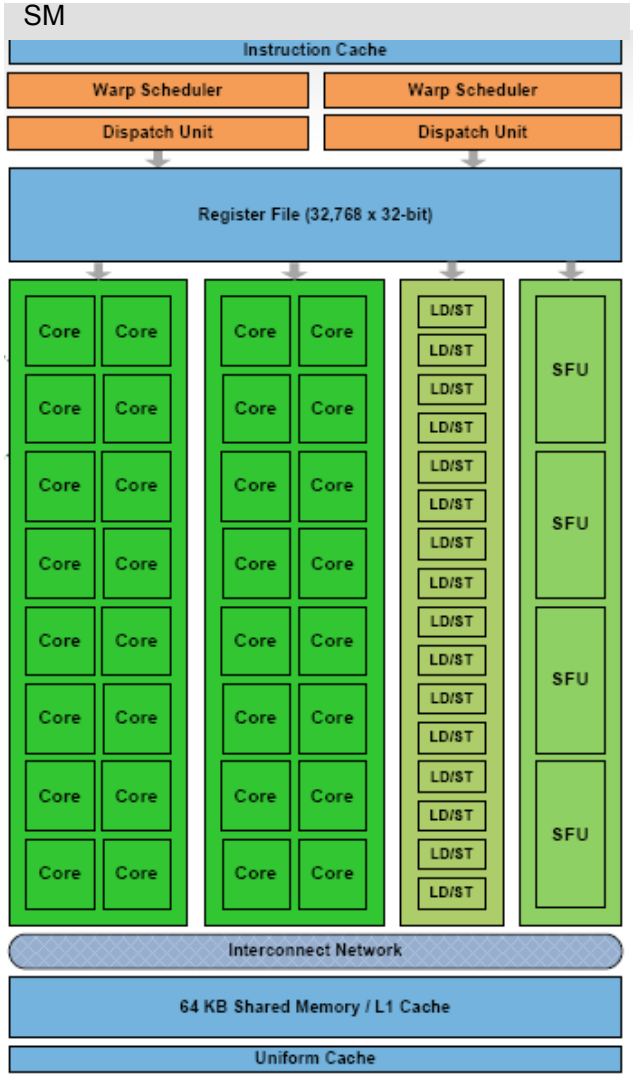
From the GF110 to the GK110 Kepler Architecture



Fermi:
512 CUDA-cores
July'11

Kepler:
2880 CUDA-cores
October'13

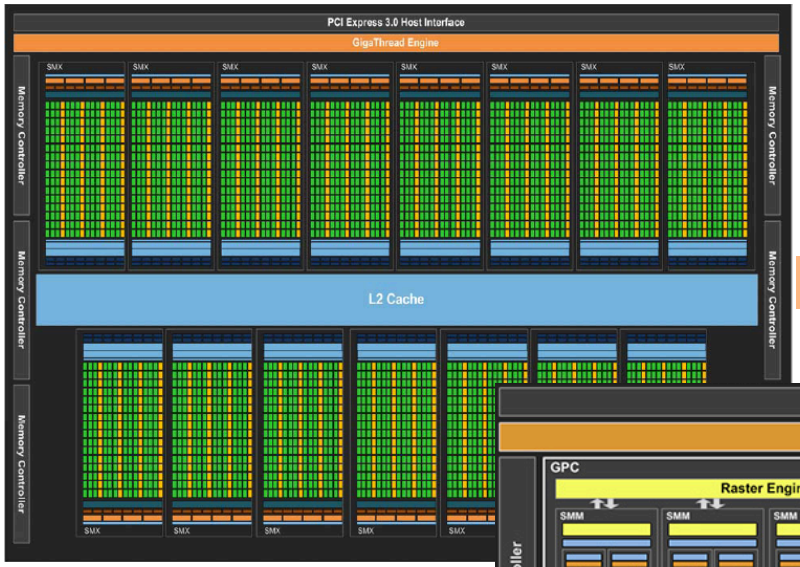
From Fermi to Kepler core: SM and the SMX Architecture



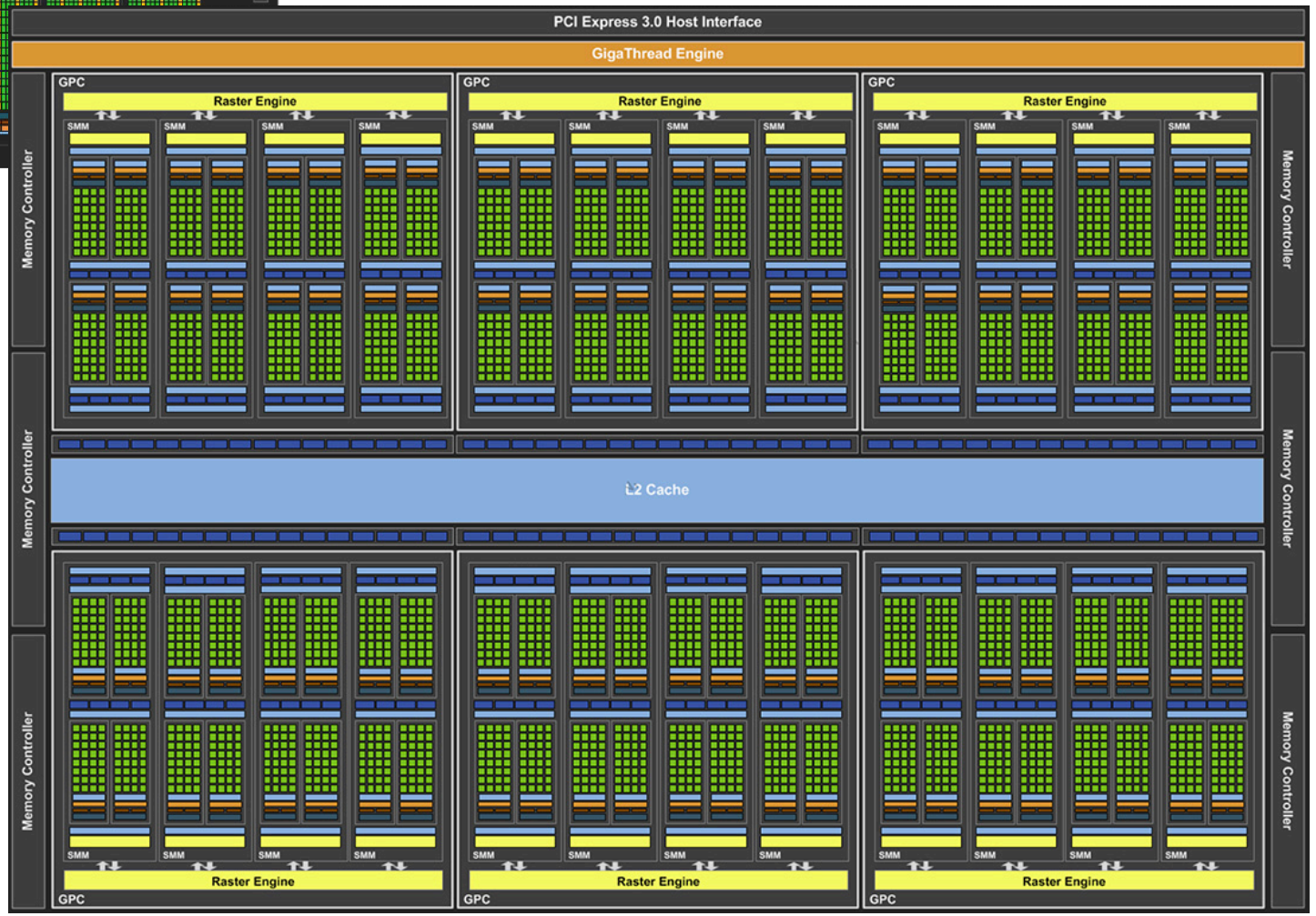
SMX:
192 CUDA-cores

Ratio **DP**unit : **SP**unit —> **1 : 3**

From the GK110 to the GM200 Maxwell Architecture



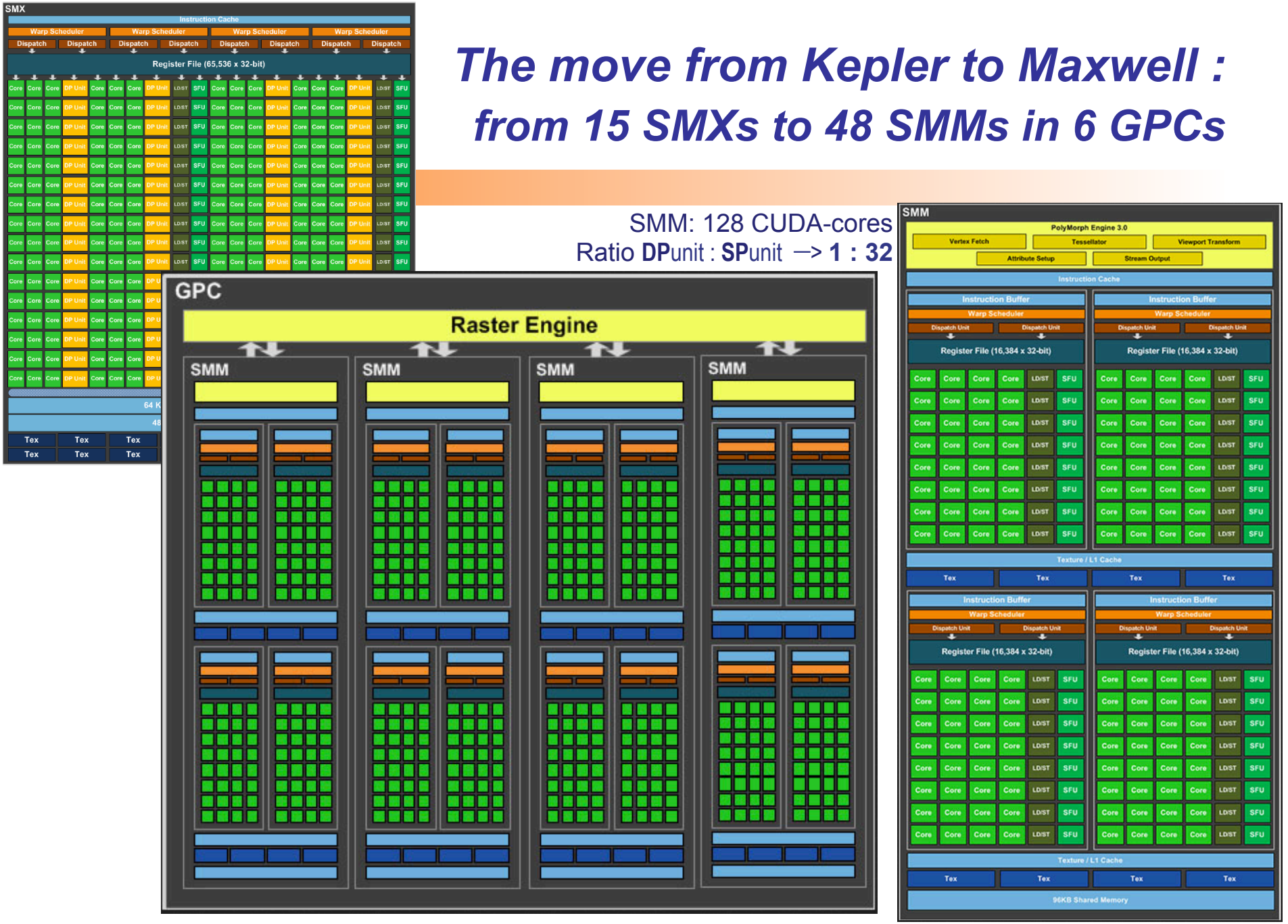
Kepler:
2880 CUDA-cores
October'13



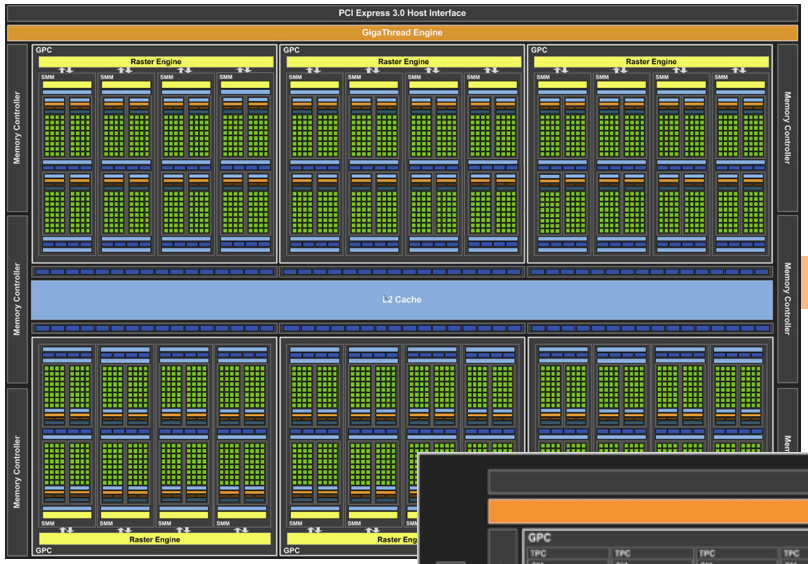
Maxwell:
3072 CUDA-cores
November'15

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

SMM: 128 CUDA-cores
Ratio **DP**unit : **SP**unit —> **1 : 32**



From the M200 to the GP100 Pascal Architecture



Maxwell:
3072 CUDA-cores
November'15

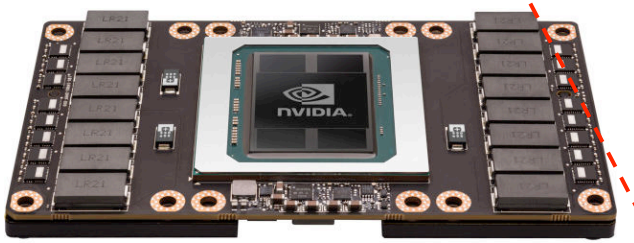
Pascal:
3584 CUDA-cores
HBM on-package
September'16

Pascal Architecture: 6x GPCs, 60 SMs

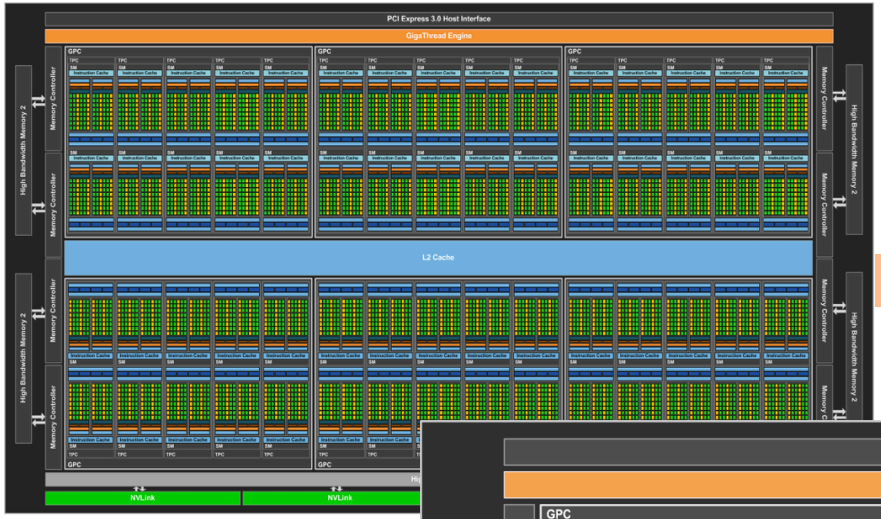
Pascal SM:
64 CUDA-cores

Ratio **DP**unit : **SP**unit —> **1 : 2**

Pascal P100 w/ 16GB HBM2



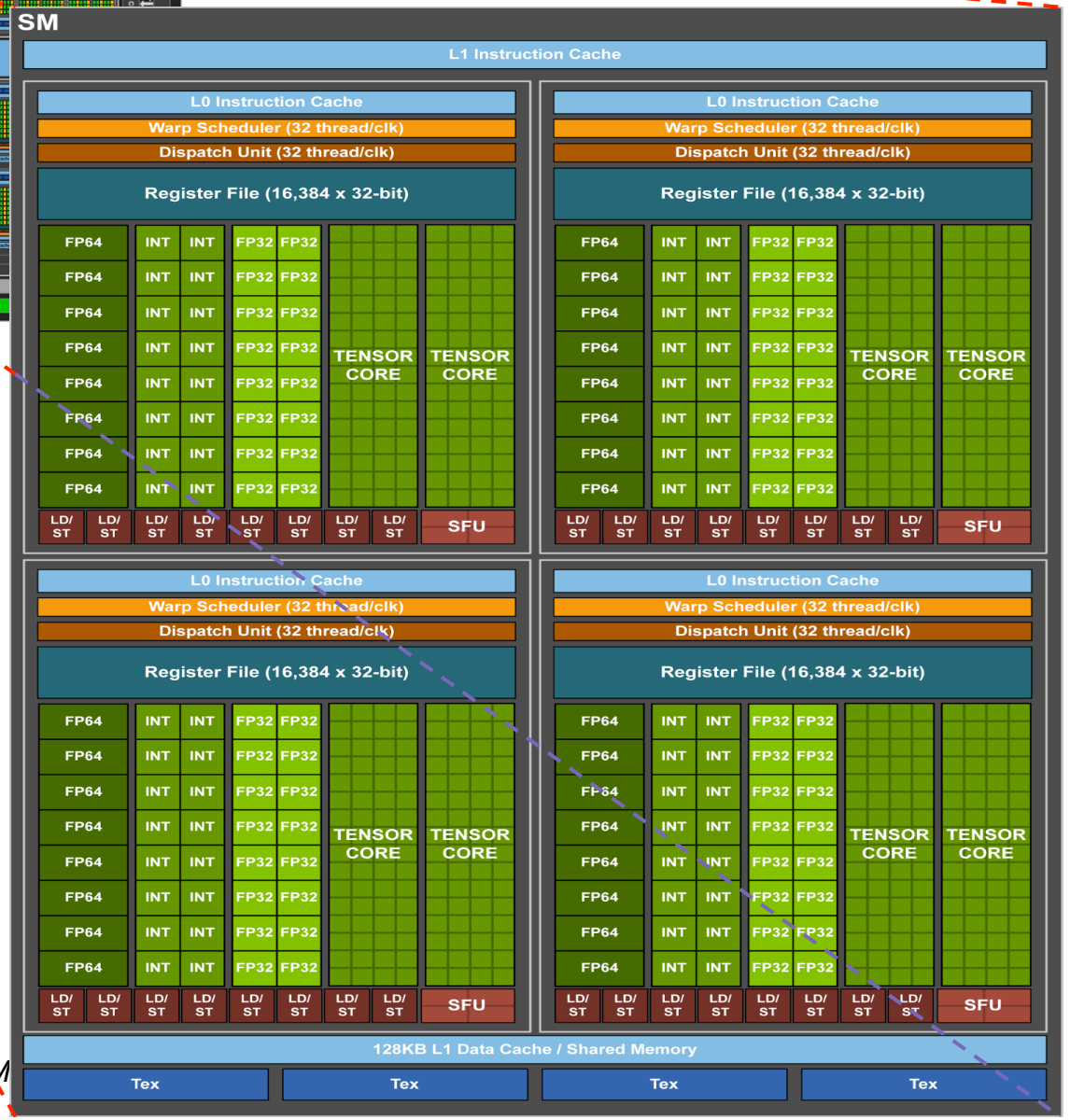
From the GP100 to the GV100 Volta Architecture



Pascal:
3584 CUDA-cores
November'15

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

Volta Architecture: 6x GPCs, 80 SMs



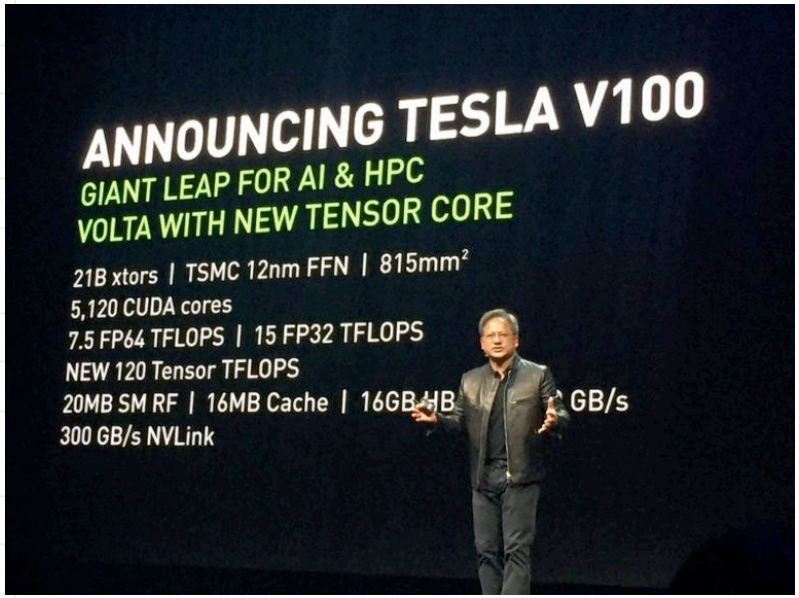
Volta SM:
64 CUDA-cores
New: 8 Tensor-cores

Ratio **DP**unit : **SP**unit —> **1 : 2**

Volta V100 w/ 16GB HBM2

Tesla accelerators: recent evolution

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

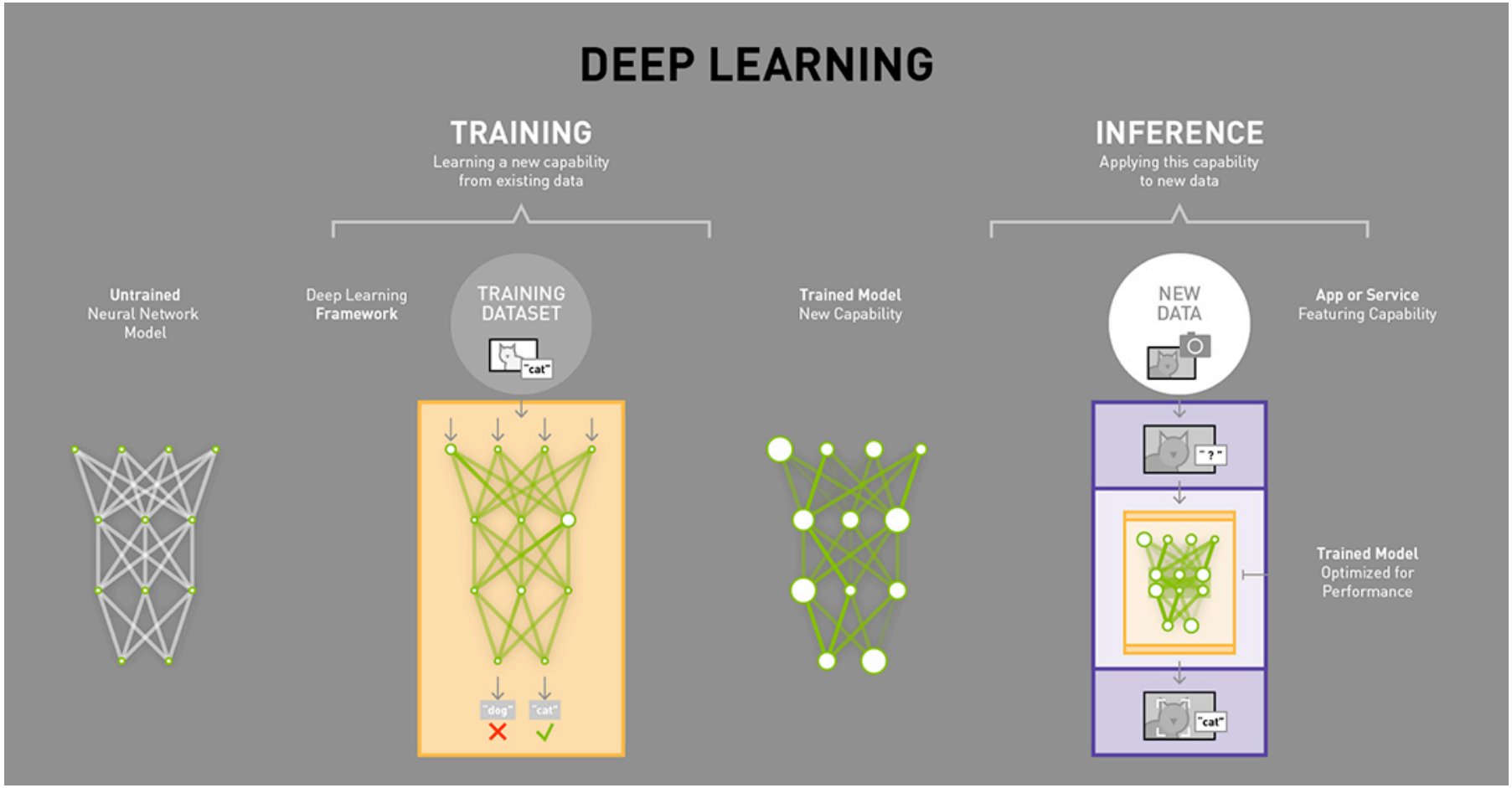


https://devblogs.nvidia.com/parallelforall/inside-volta/

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: **ShenWei** 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**
    - …

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



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

NVidia Volta Architecture: the new Tensor Cores

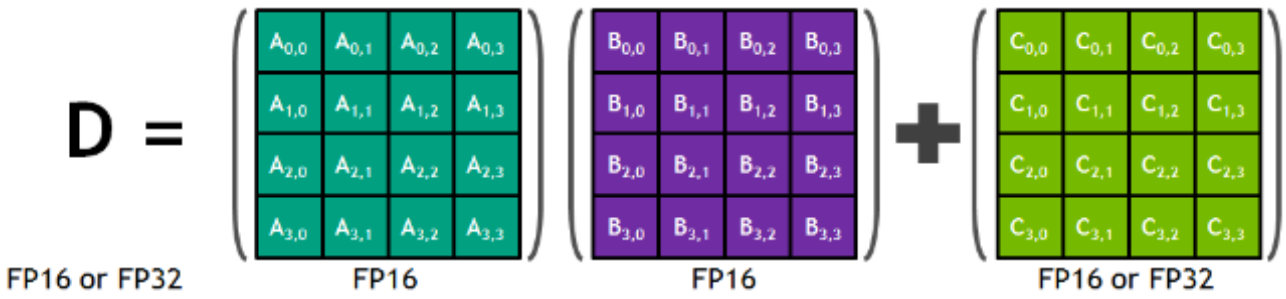


Figure 8. Tensor Core 4x4 Matrix Multiply and Accumulate

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

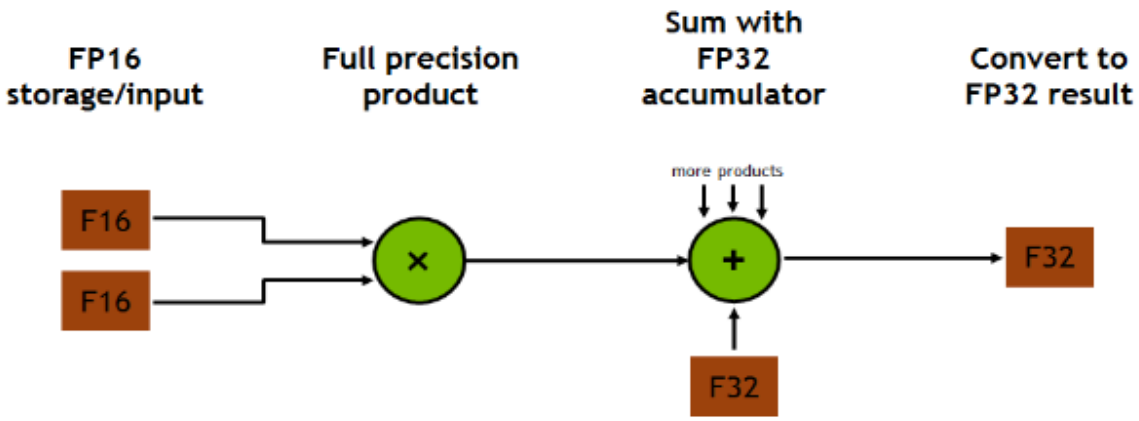


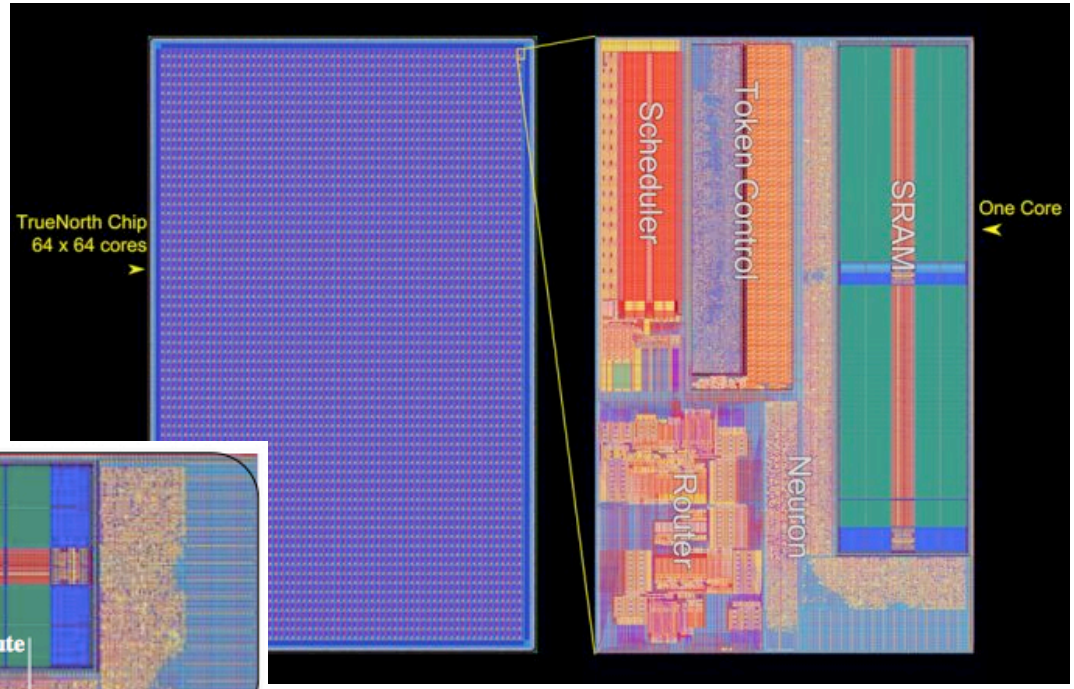
Figure 9. Mixed Precision Multiply and Accumulate in Tensor Core

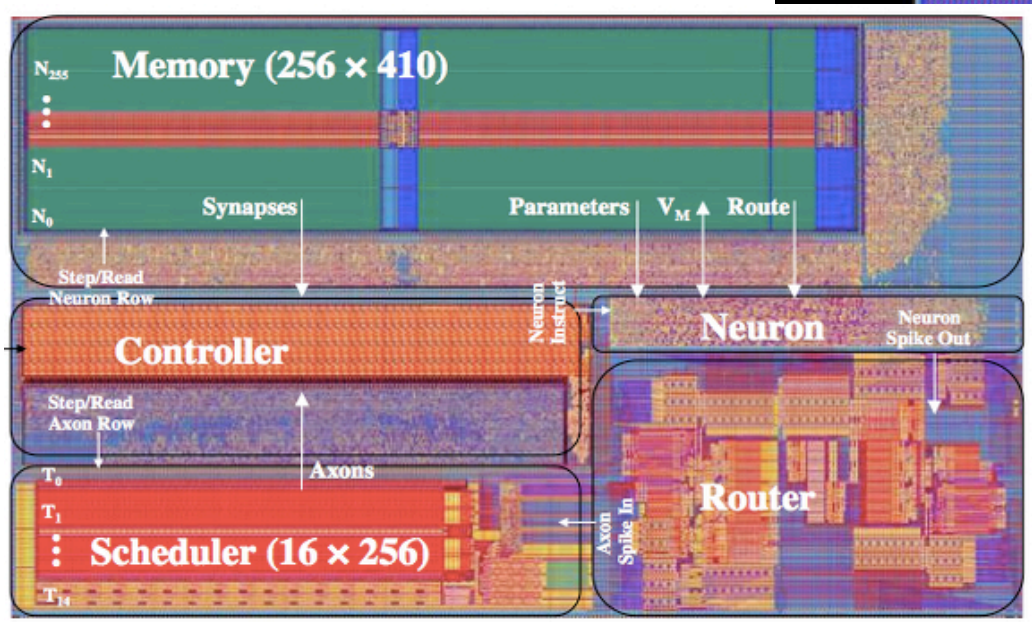
http://www.nvidia.com/content/gated-pdfs/Volta-Architecture-Whitepaper-v1.1.pdf

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

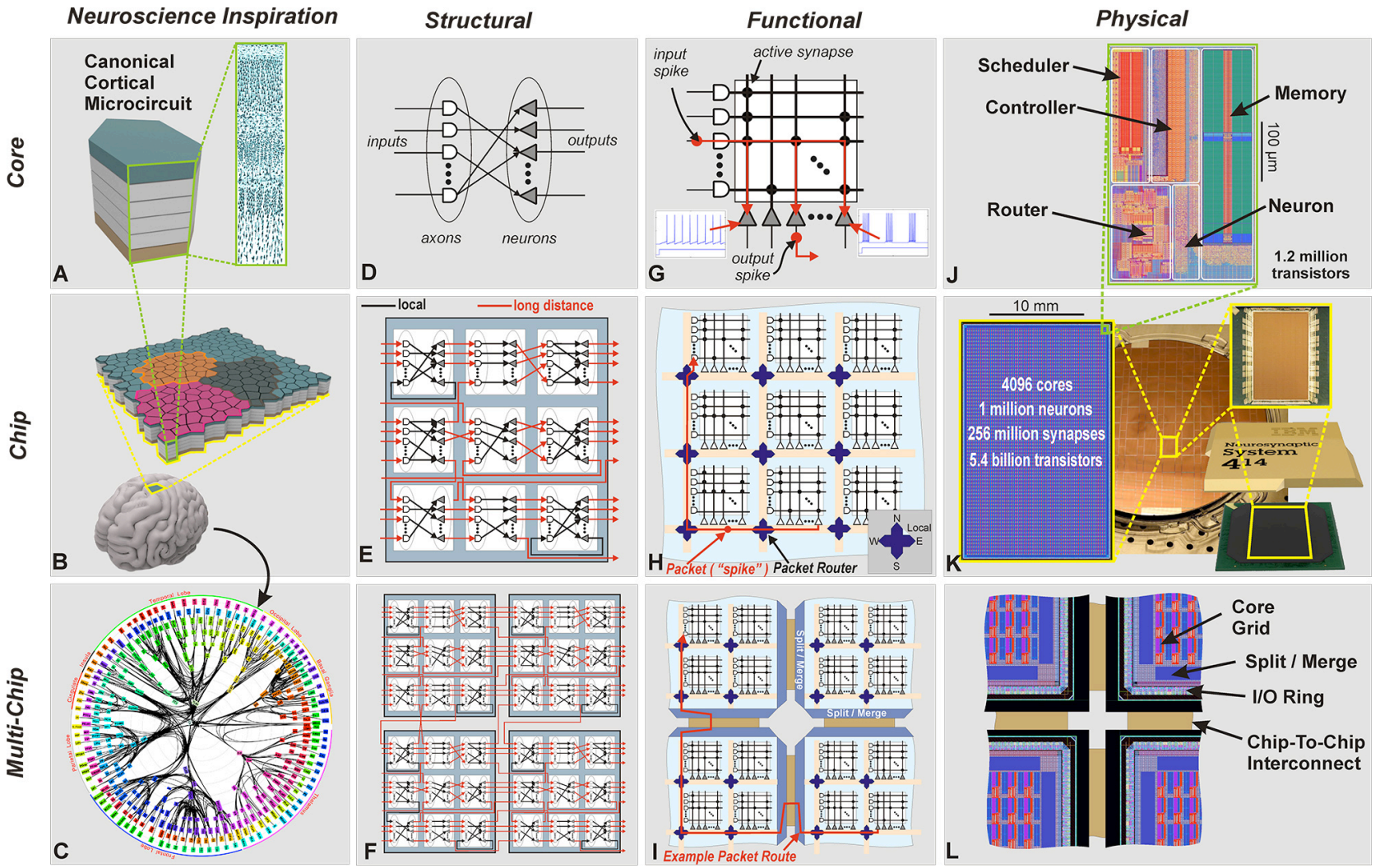
TrueNorth Chip:
- 4096 neurosynaptic cores

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





NVidia competitors with neural net features: the IBM TrueNorth architecture



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

History

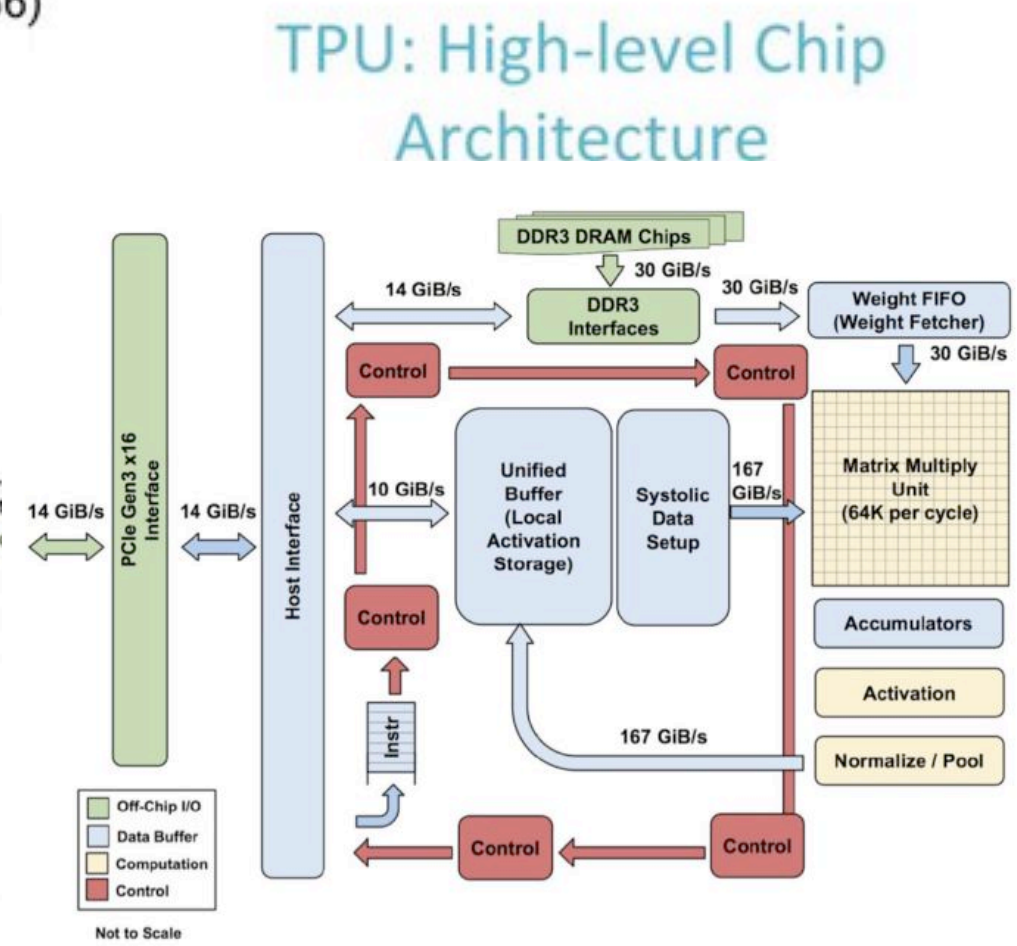
- Nervana Engine announced in May'16
- Key features:
  - ASIC chip, focused on matrix multiplication,convolutions,... (for neural nets)
  - HBM2: 4x 8GB in-package storage & 1TB/sec memory access b/w
  - no h/w managed cache hierarchy (saves die area, higher compute density)
  - built-in networking (6 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'17
  - 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/

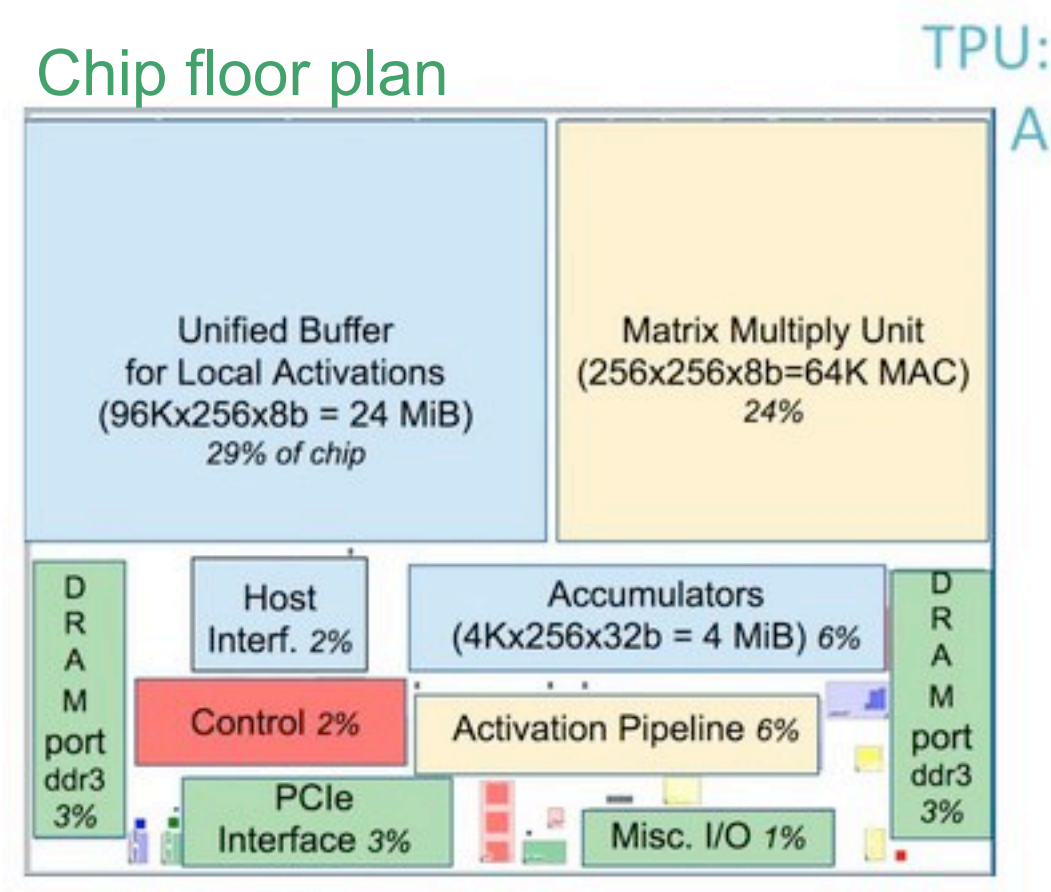
NVidia competitors with neural net features: Google Tensor Processing Unit, TPU (April'17)

- 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



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

Chip floor plan



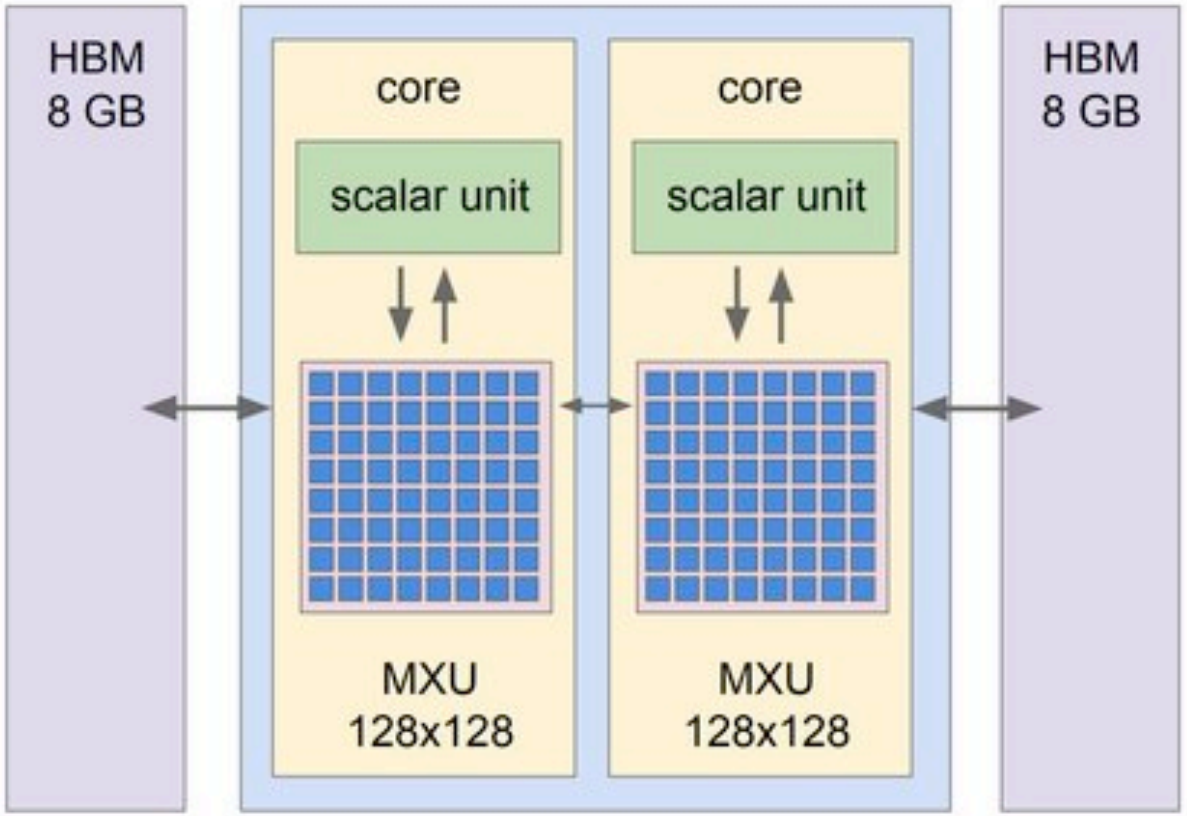
TPUs are intensively used by Google, namely in RankBrain, StreetView & Google Translate

https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

NVidia competitors with neural net features: Google TPUv2 (September'17)

TPUv2 Chip

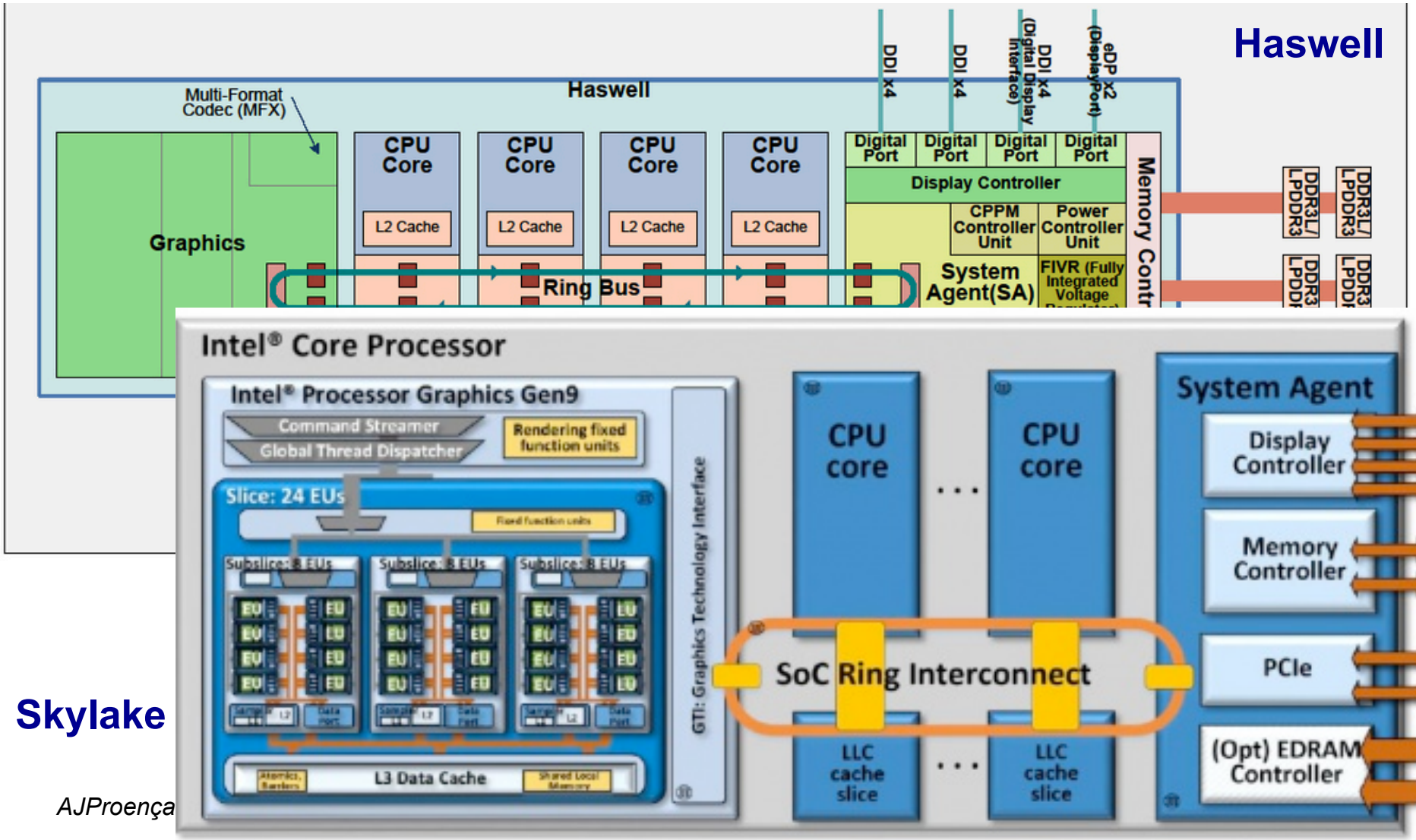
- 16 GB of HBM
- 600 GB/s mem BW
- Scalar unit: 32b float
- MXU: 32b float accumulation but reduced precision for multipliers
- 45 TFLOPS



Beyond Vector/SIMD architectures

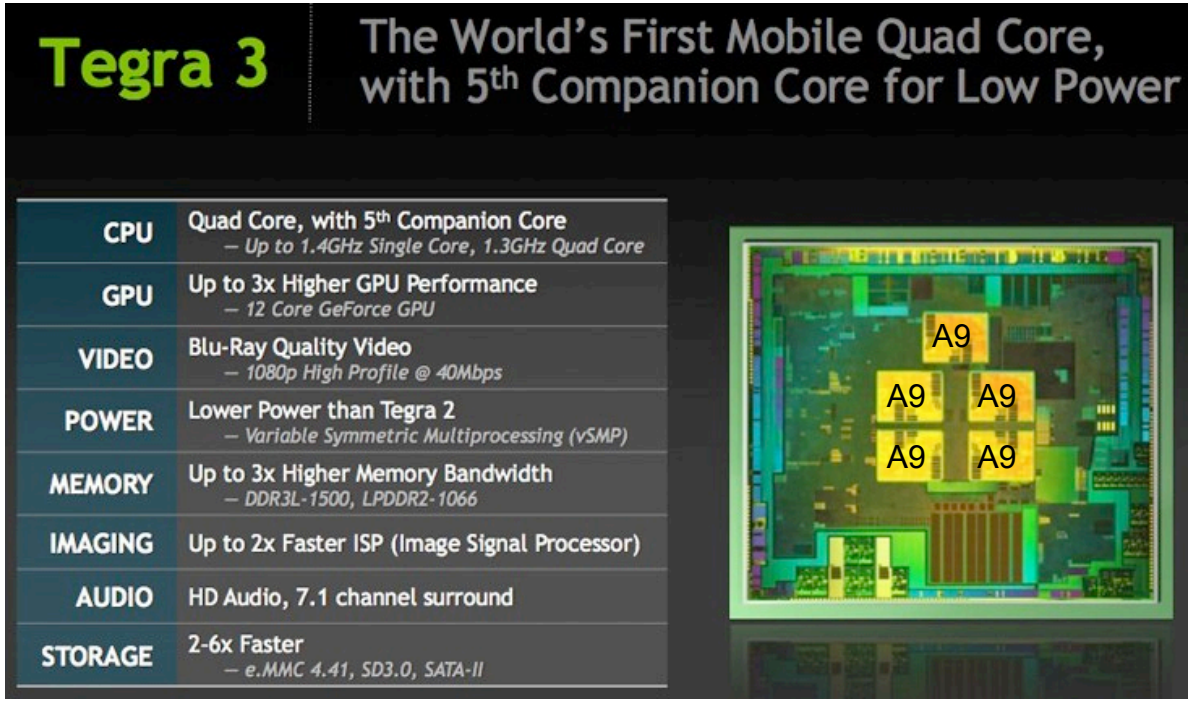
- Vector/SIMD-extended architectures are hybrid approaches
  - mix (super)**scalar + vector** op capabilities on a single device
  - **highly pipelined** approach to reduce memory access penalty
  - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
  - **PU (Processing Unit) cores with wider vector units**
    - x86 many-core: **Intel** MIC / Xeon KNL
    - other many-core: **ShenWei** 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



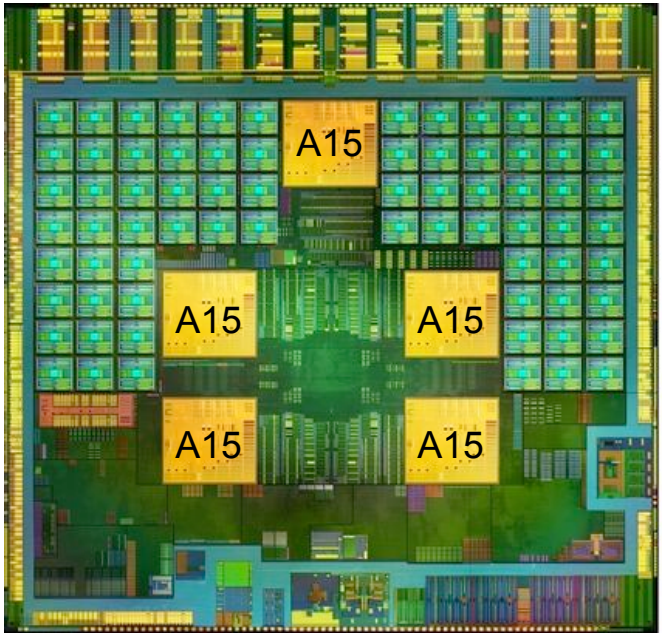
NVidia Tegra: SoC partnership with ARM (1)

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



Tegra 3 **Nov'2011**

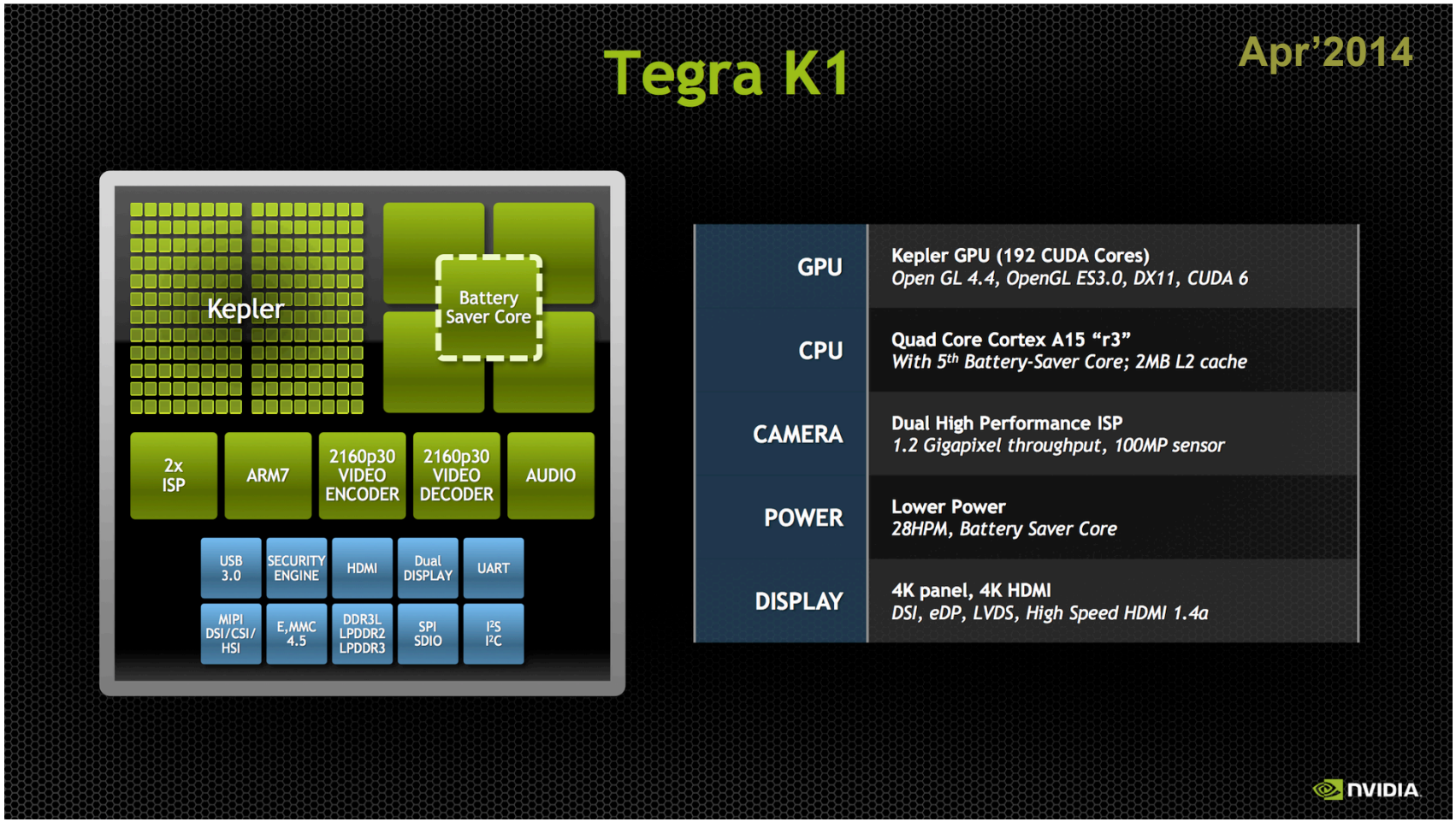
Tegra 4:
replace the 32-bit ARM Cortex A9 by Cortex A15, and add 72 CUDA-cores



Tegra 4 **May'2013**

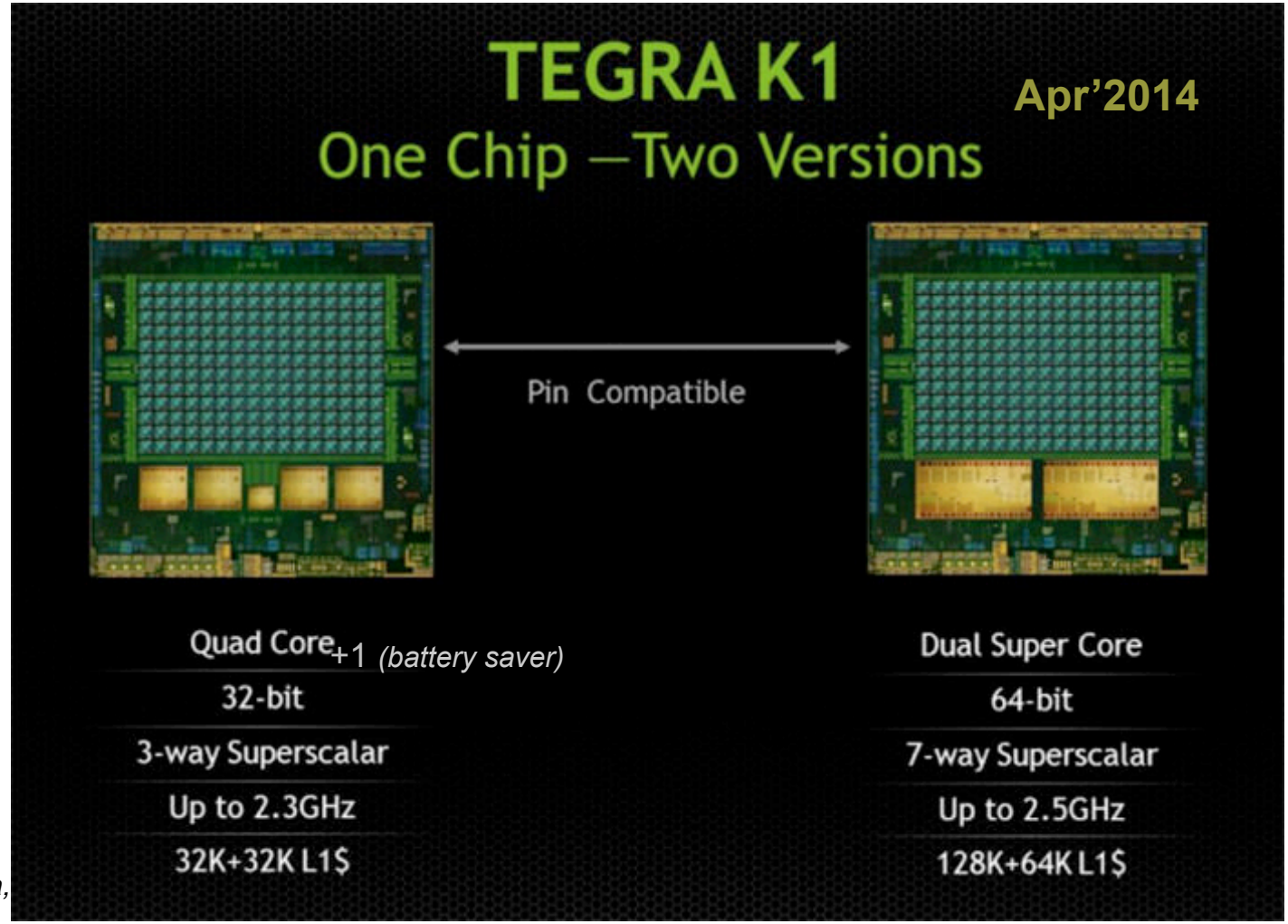
NVidia Tegra: SoC partnership with ARM (2)

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



NVidia Tegra: SoC partnership with ARM (3)

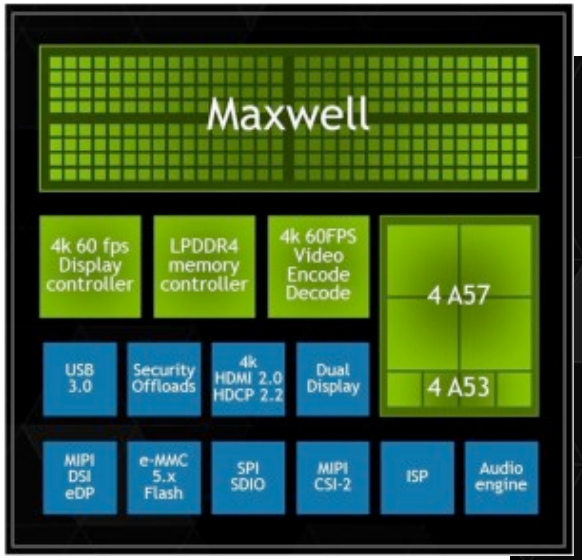
Replace ... => Tegra K1

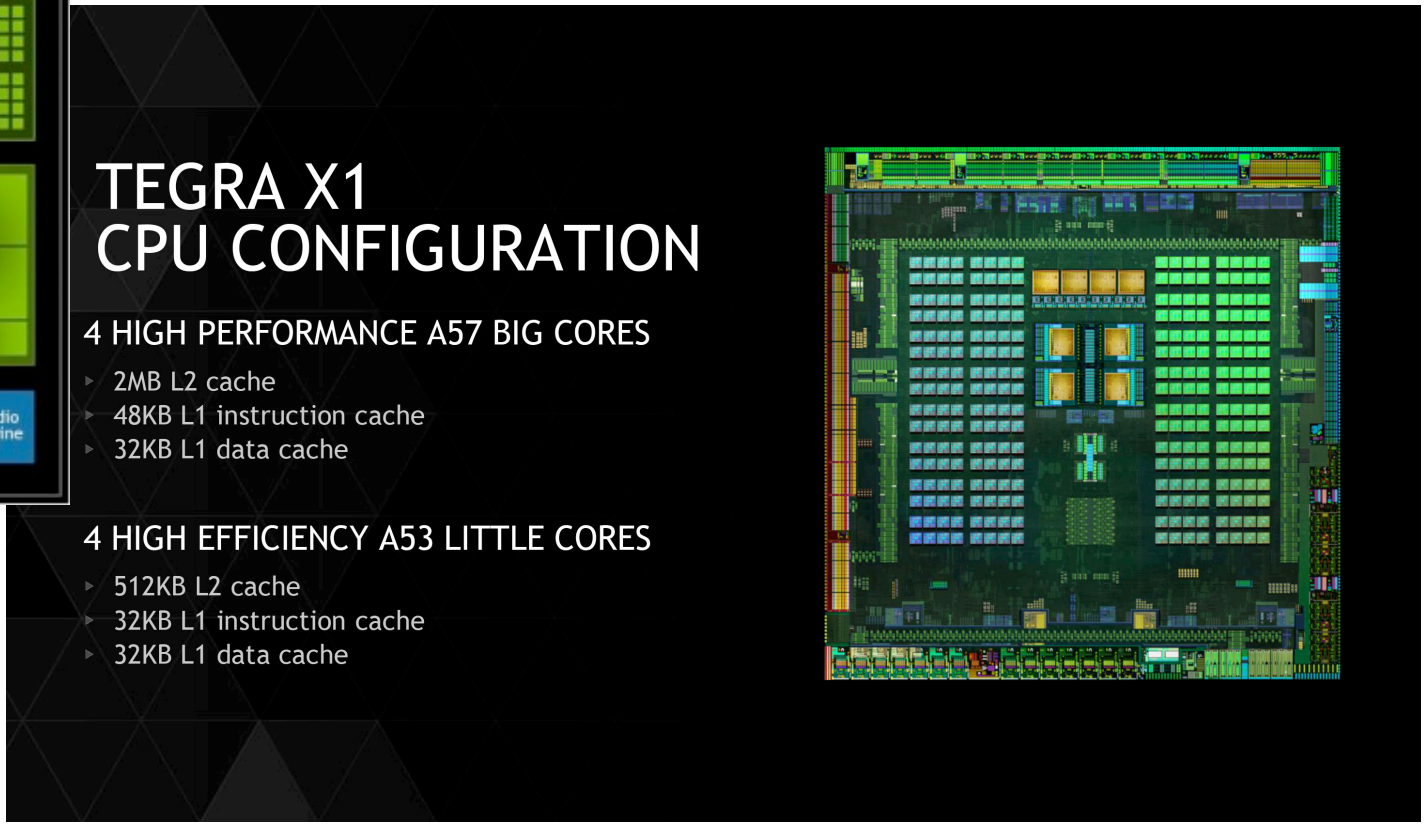


NVidia Tegra: SoC partnership with ARM *(4)*

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

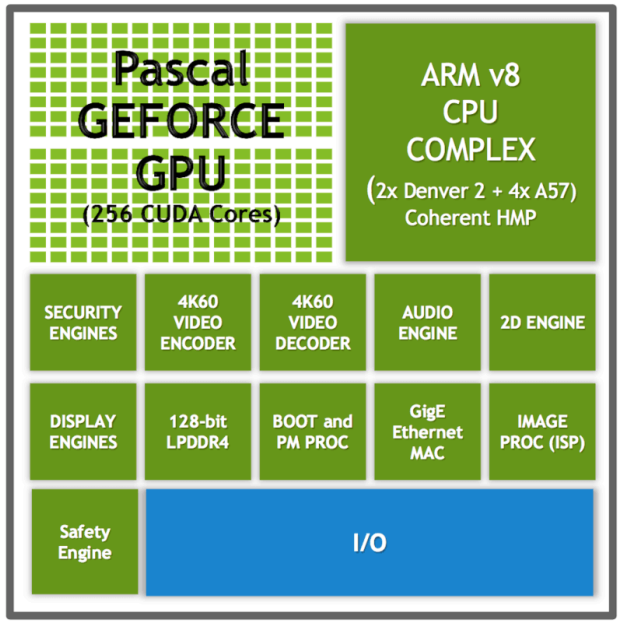
May'2015





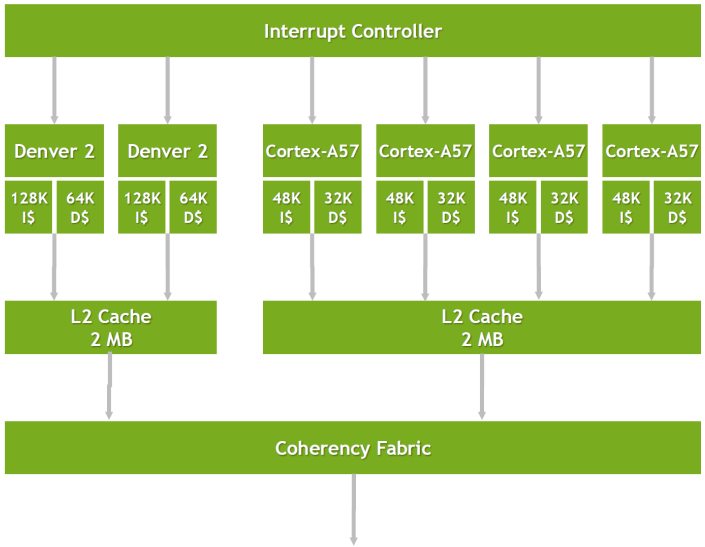
NVidia Tegra: pathway towards ARM-64 (1)

- Upgrade 32-bit ARM to 64-bit ARMv8 (*Denver 2 & A57*) and replace Maxwell cores by Pascal ones => **Parker** **Aug'2016**



"PARKER" CPU COMPLEX

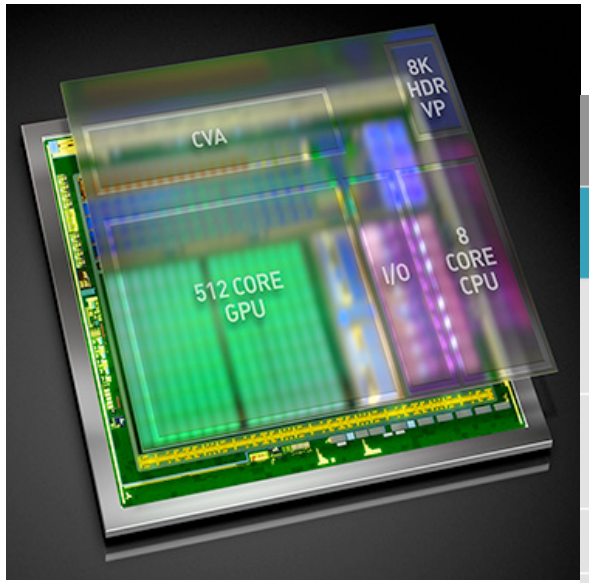
- 2x Denver2 + 4x Cortex-A57
- Fully Coherent HMP system
  - Proprietary Coherent Interconnect
- ARM V8 64-bit
- Highest performance ARM CPU
  - 2nd generation Denver core
  - Significant Perf/W improvements
- Dynamic Code Optimization
  - OoO execution without the power
  - Optimize once, use many times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



7 NVIDIA

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 X1)
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

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: **ShenWei** 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