Master Informatics Eng.

2020/21

A.J.Proença

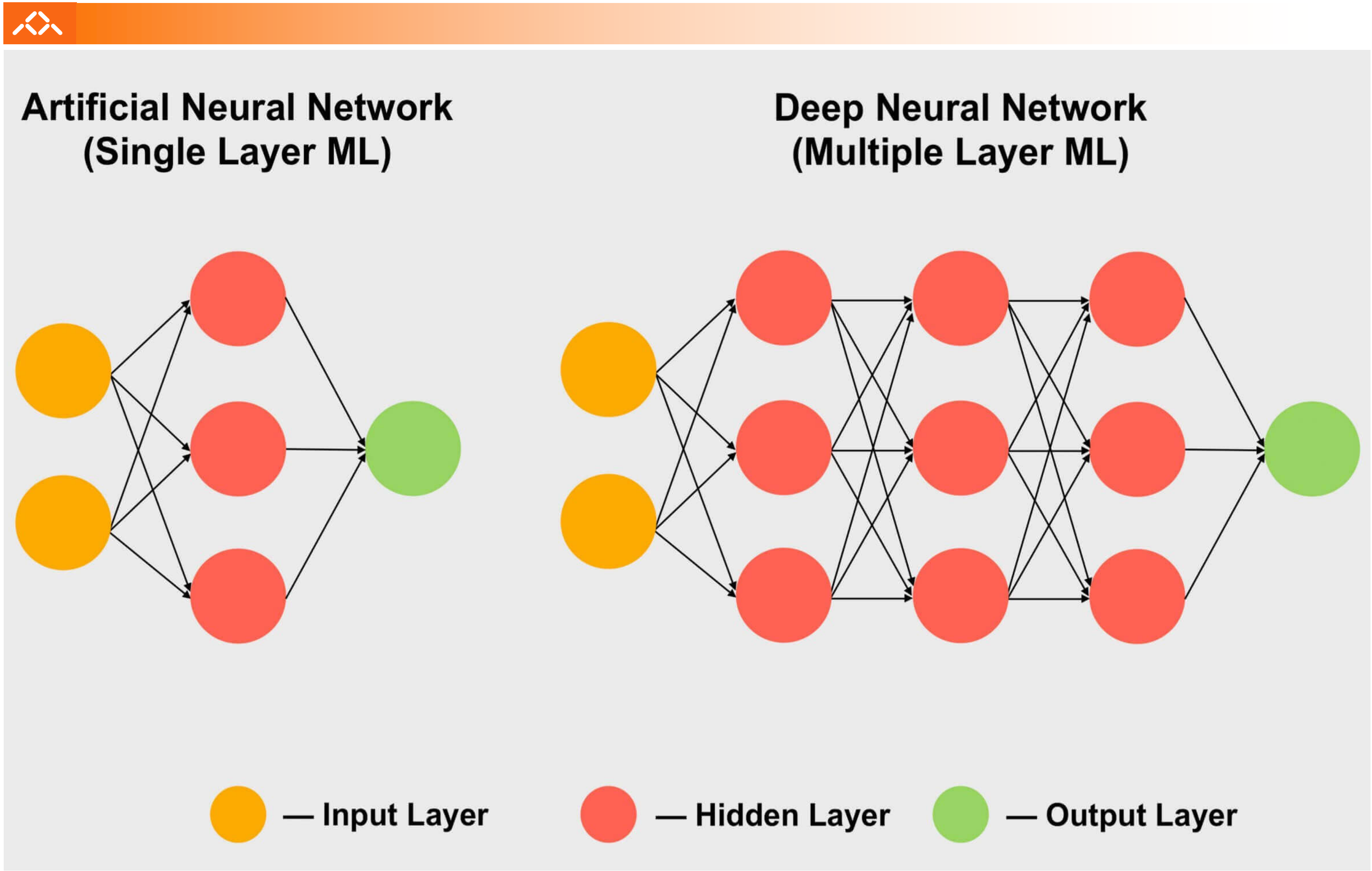
Beyond Vector Extensions *(online)*

(most slides are borrowed)

Beyond vector extensions

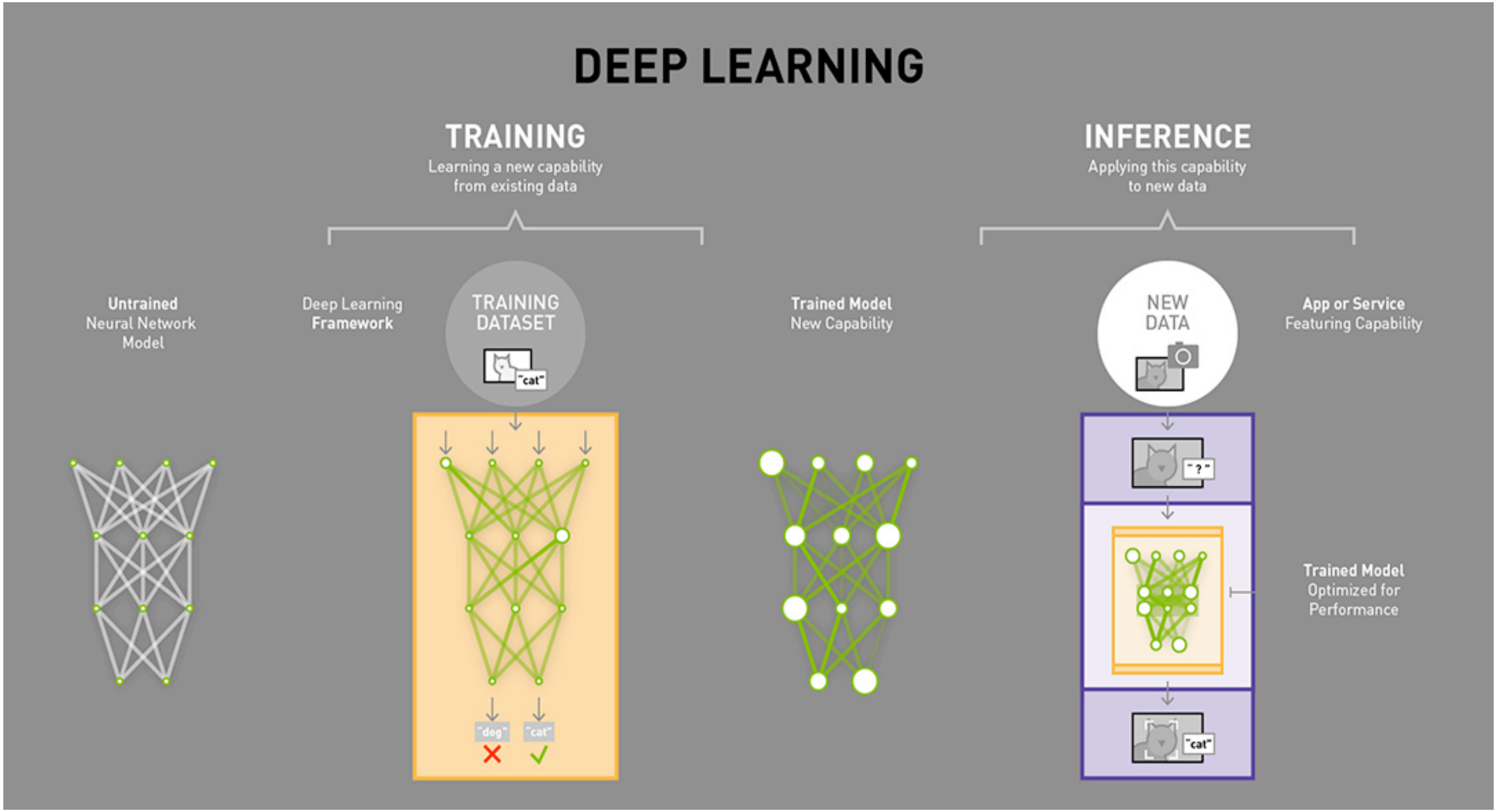
- Evolution of vector/SIMD-extended architectures
  - **accelerators optimized for number crunching (GPU)**
  - **support for matrix multiply + accumulate operations**
    - most scientific, engineering, AI & finance applications use matrix computations, namely the dot product: multiply and accumulate the elements in a row of a matrix by the elements in a column from another matrix
    - typically these extensions are **Tensor Processing Unit** (TPU)
  - **support for half-precision FP & 8-bit integer**
    - **machine learning using neural nets** is becoming very popular; to compute the model parameters during training phase, intensive matrix products are used and with very low precision (is adequate!)

Machine learning w/ neural nets & deep learning

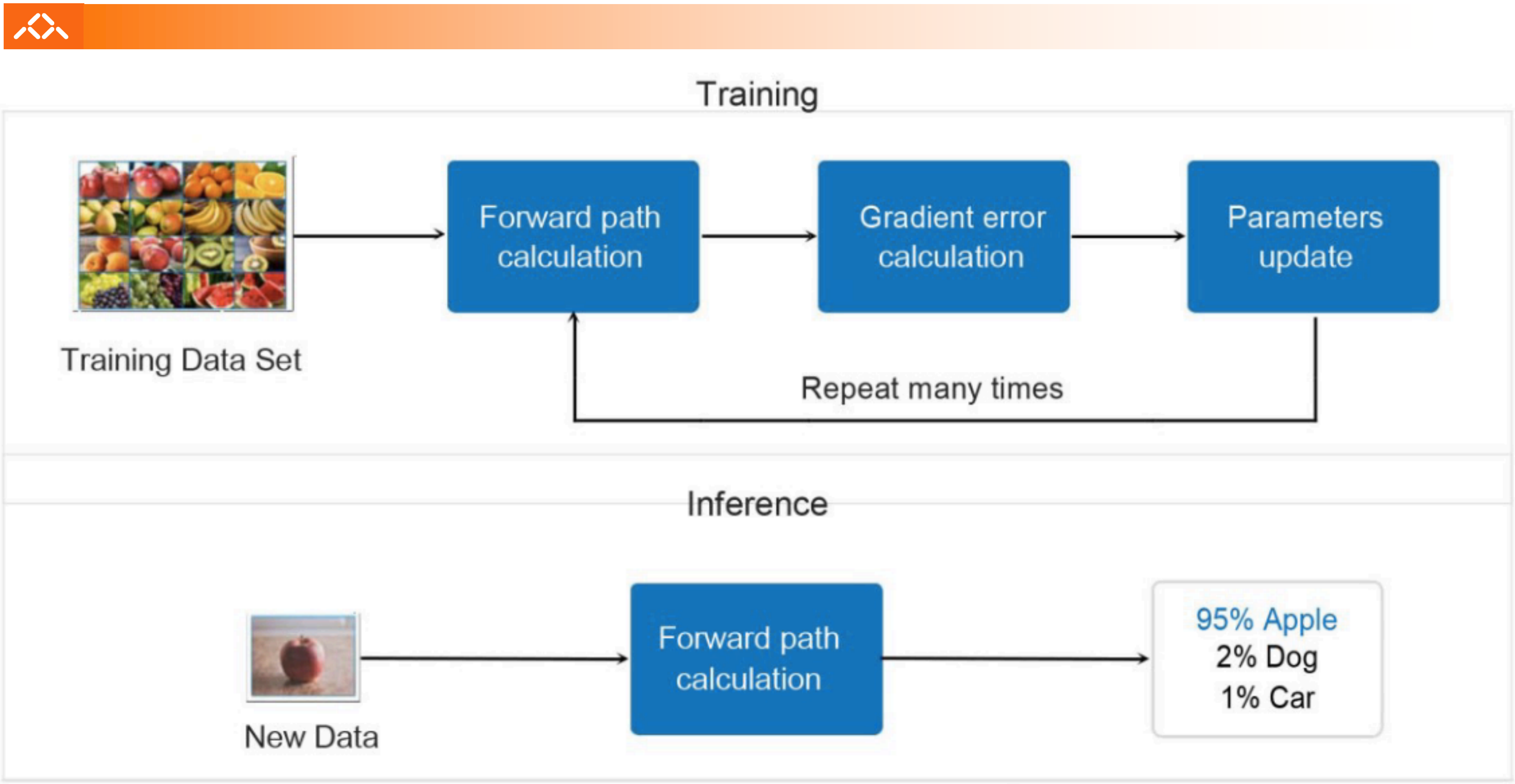


https://www.techspot.com/article/2049-what-are-tensor-cores/

Deep Learning phases


DEEP LEARNING
TRAINING
Learning a new capability from existing data
INFERENCE
Applying this capability to new data
Untrained Neural Network Model
Deep Learning Framework
TRAINING DATASET
"cat"
"dog"
"cat"
✗
✓
Trained Model New Capability
NEW DATA
"?"
"cat"
App or Service Featuring Capability
Trained Model Optimized for Performance

Deep Learning workflow

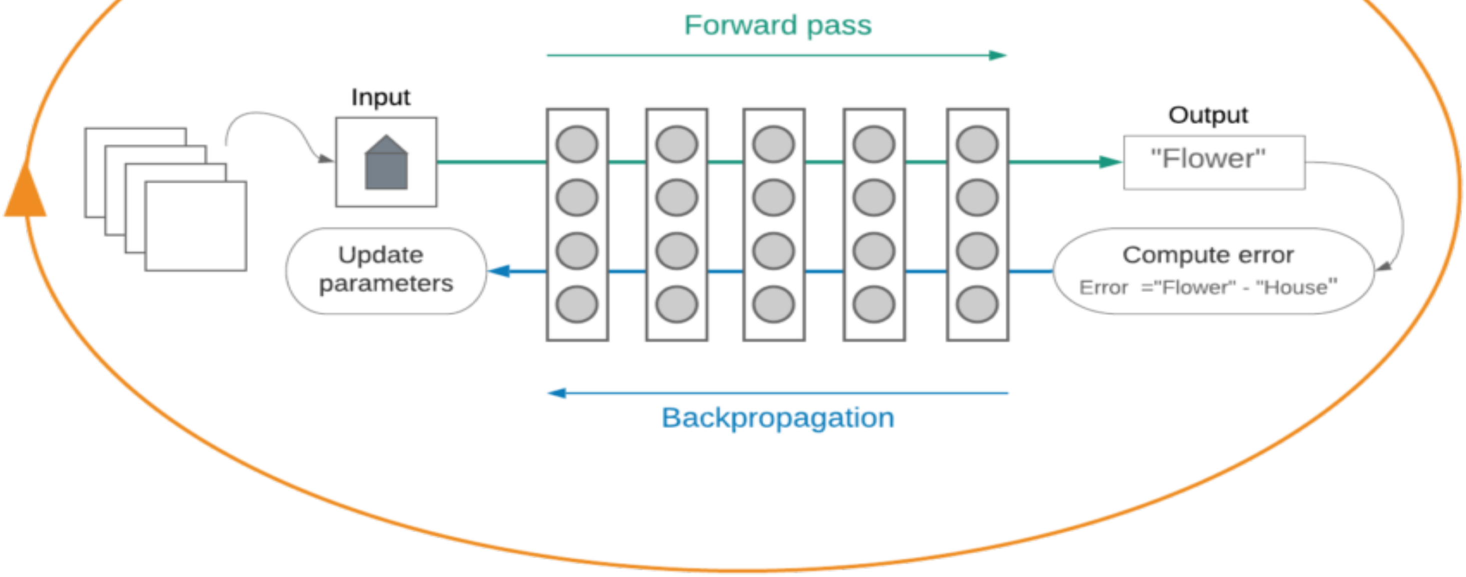


Key algorithms to train & classify use matrix dot products, but do not require high precision numbers!

https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf

Training a Neural Net Model

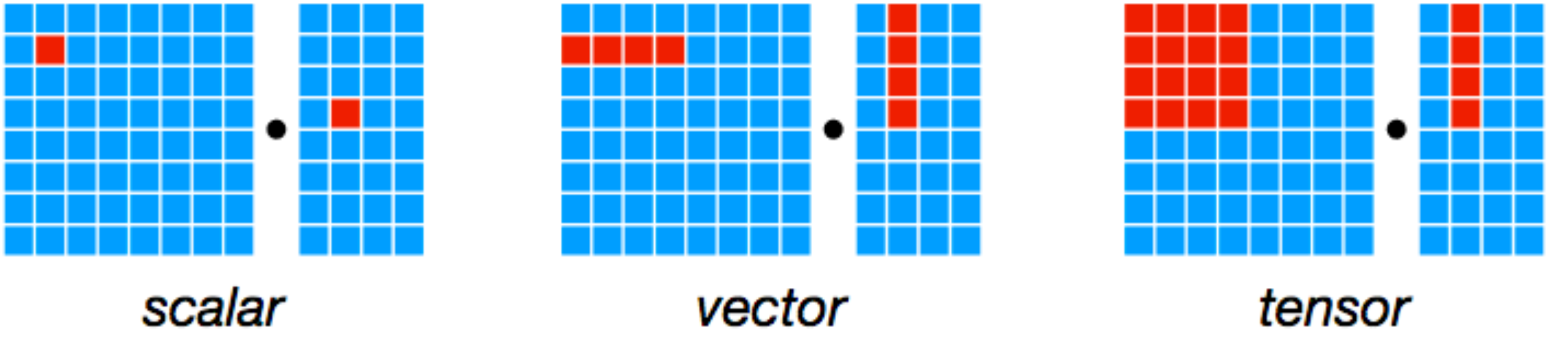
*During model **training**, labeled data samples flow from input to output (I to O) through all layers of parametrized transformations — a forward pass. At the output end, the output, or prediction, is compared to the correct answer for that particular input. Prediction error is computed; error being the difference between the predicted output and the correct one. Then, the error begins to work its way backwards in the O-to-I direction, via the backpropagation algorithm. As the error flows through each layer, it interacts with the I-to-O data that produced it (that data has been parked there, waiting for the error to come back) and, together, they determine how to change the layer's parameters to most effectively reduce the error. The parameters are then adjusted, and this process of forward-backpropagation steps continues for numerous passes over the set of training examples, until the error becomes insignificant or doesn't decrease anymore.*



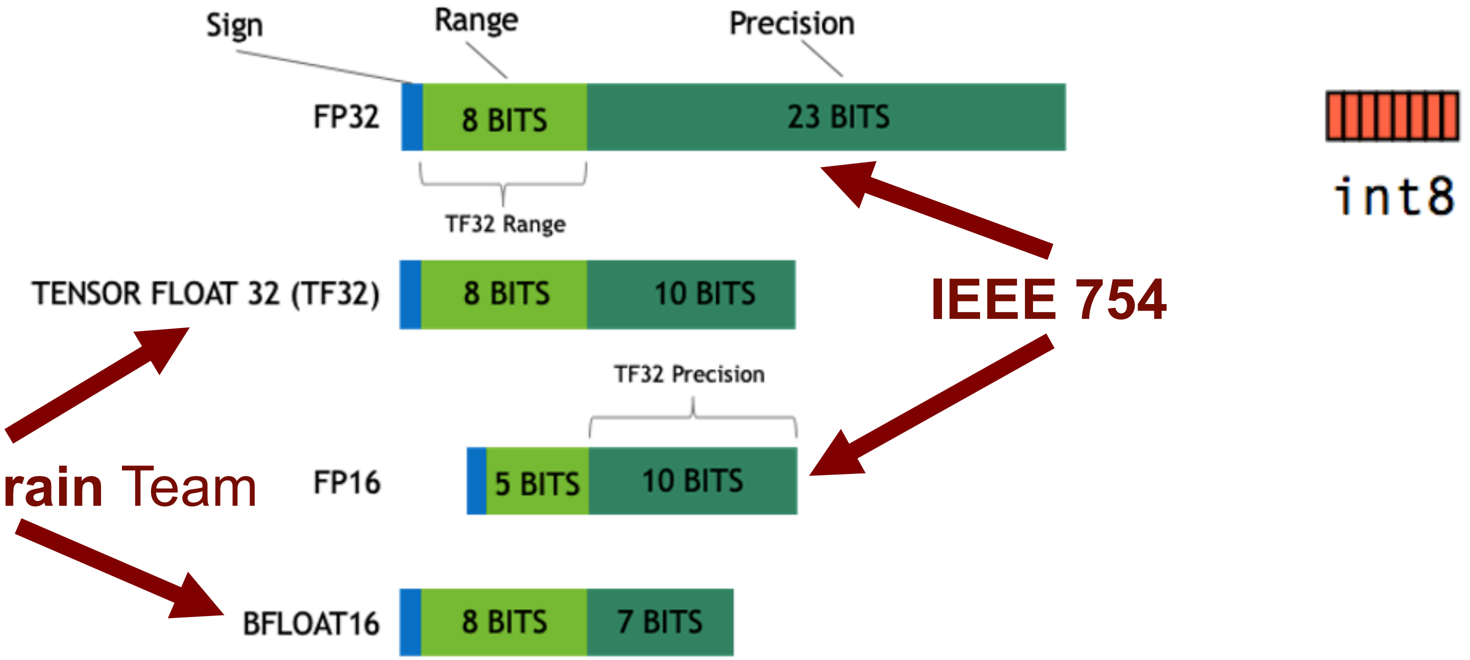
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

Required hardware operations & data types to train & classify neural nets

Compute primitives



Data type



IEEE 754

Google Brain Team

Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

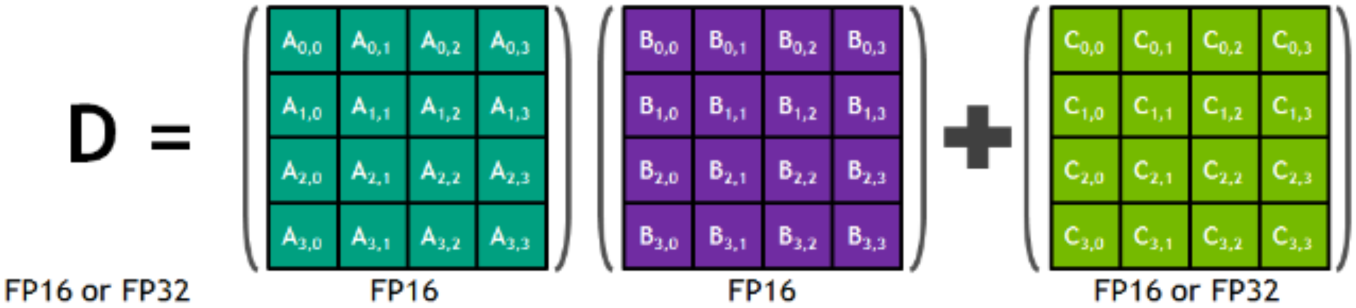
- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

NVidia Volta Architecture: the new Tensor Cores



$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

Figure 8. Tensor Core 4x4 Matrix Multiply and Accumulate

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

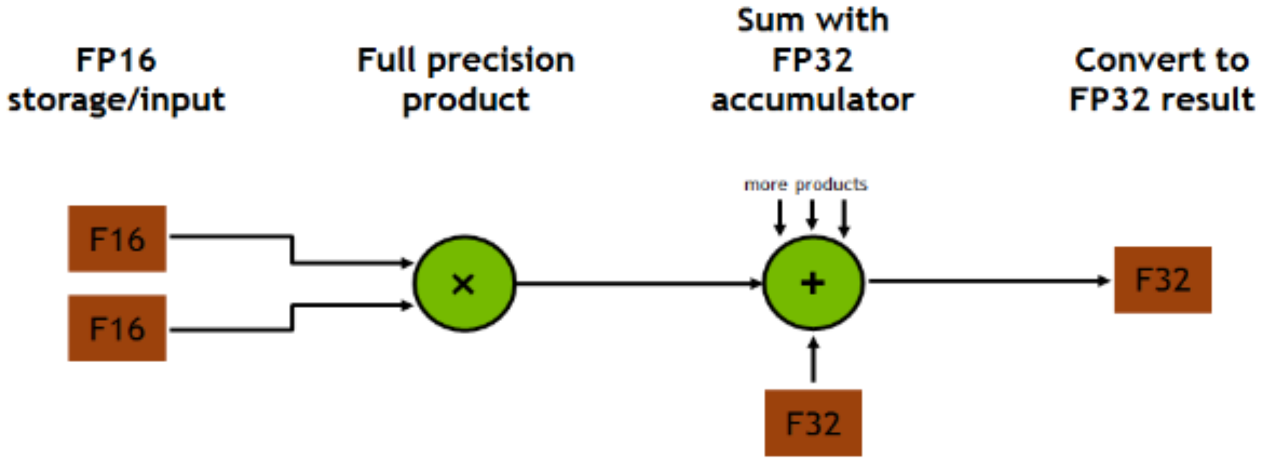
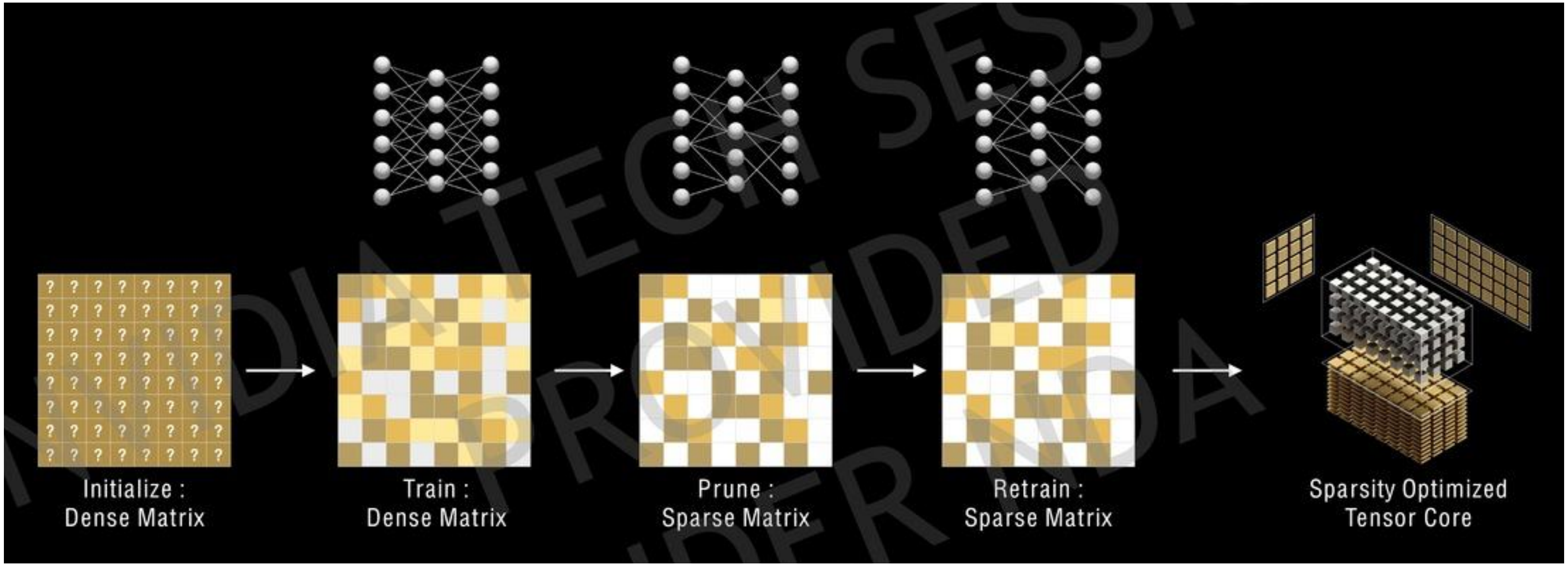


Figure 9. Mixed Precision Multiply and Accumulate in Tensor Core

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

NVidia Ampere Architecture: the new 3rd generation Tensor Cores

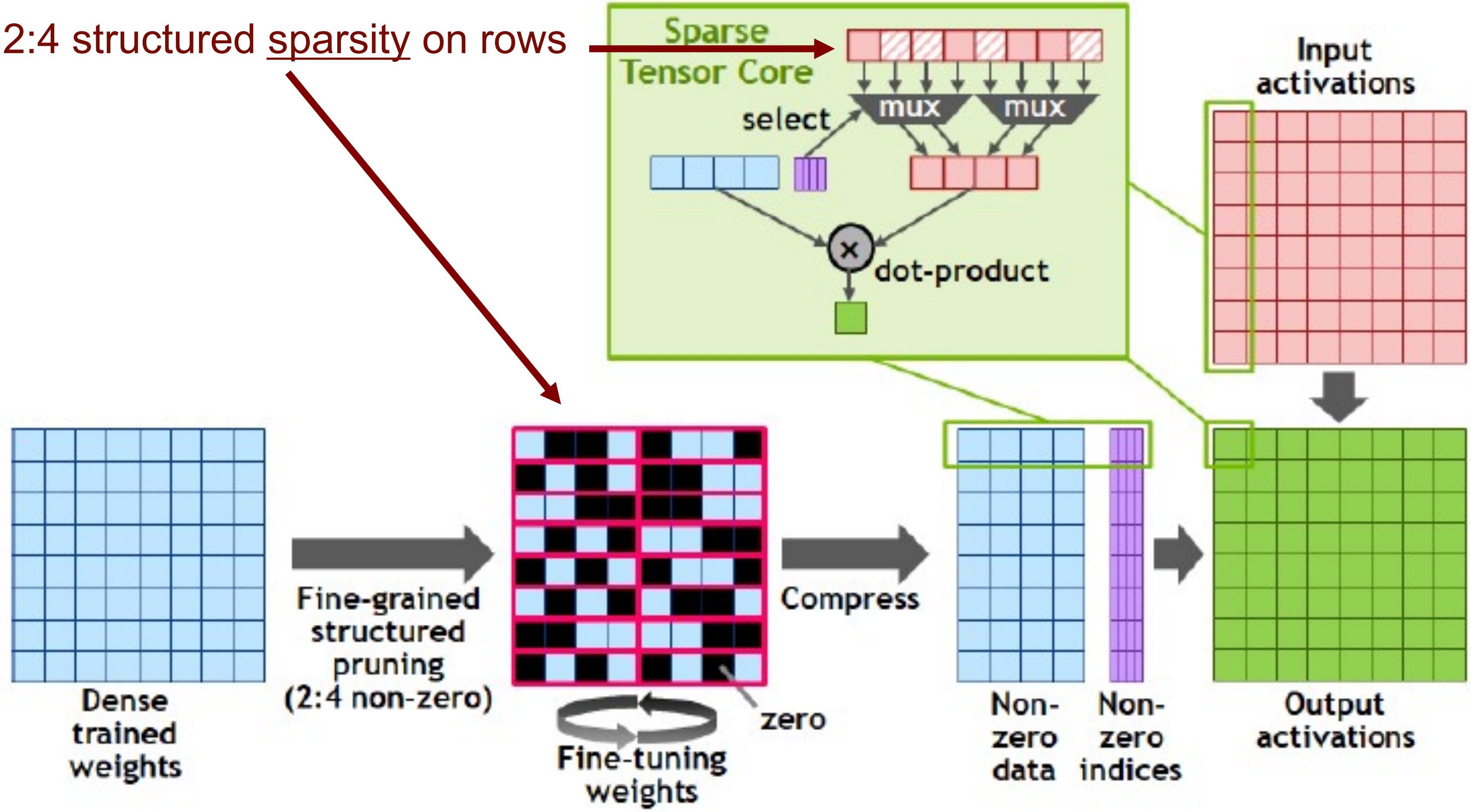
- GEMM *(Generic Matrix Multiplication)* computes ***D*** = ***A*** * ***B*** + ***C***
  - ***A*** can be up to 8x8 matrix (mixed-precision ith ***B***)
  - ***B, C, D*** can be up to 8x4 matrix
- Each SM in A100 has 4 Tensor Cores
  - 4x 256 FMA ops/cycle (FP16), **2 KFLOP/cycle**
- A100 with Fine-Grained Structured Sparsity
  - 2:4 structured sparsity on rows (2 non-zero values in every 4-entry vector)



https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

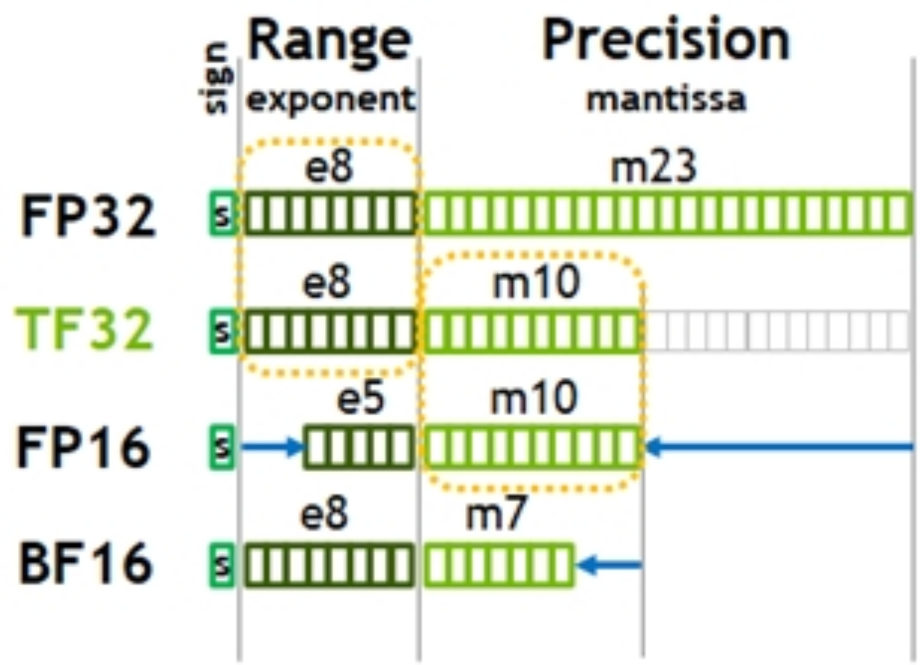
Fine-Grained Structured Sparsity in A100

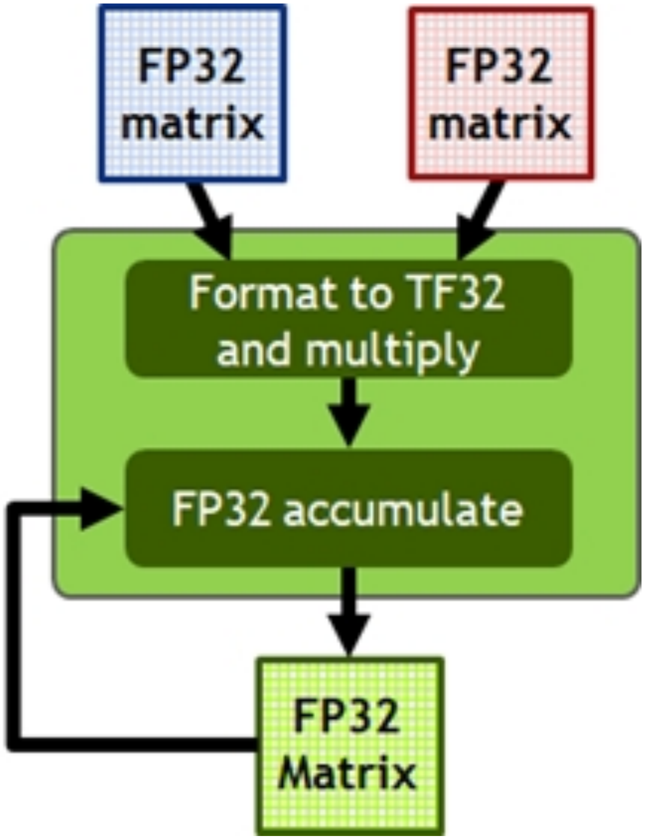
2:4 structured sparsity on rows



https://www.nextplatform.com/2020/05/28/diving-deep-into-the-nvidia-ampere-gpu-architecture/

Data Types in Ampere Tensor Cores



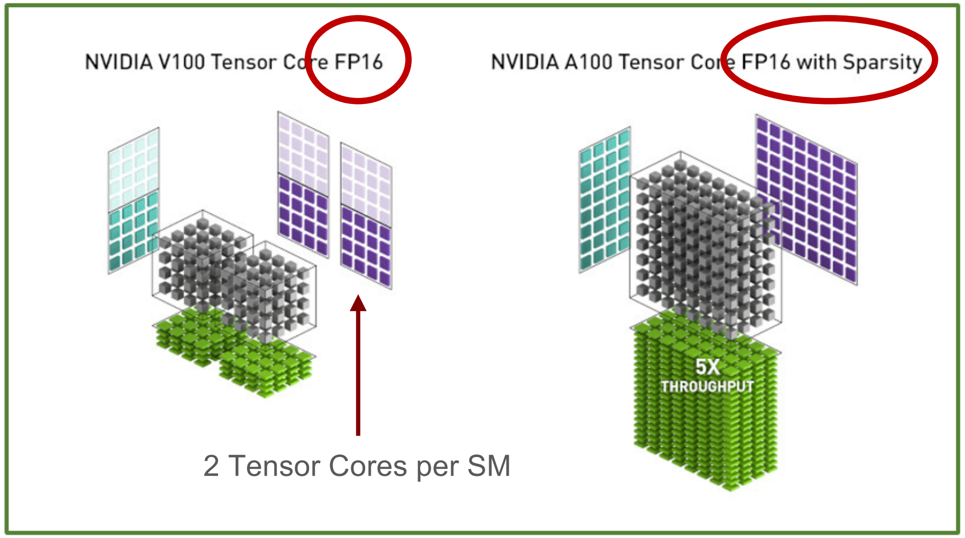


https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

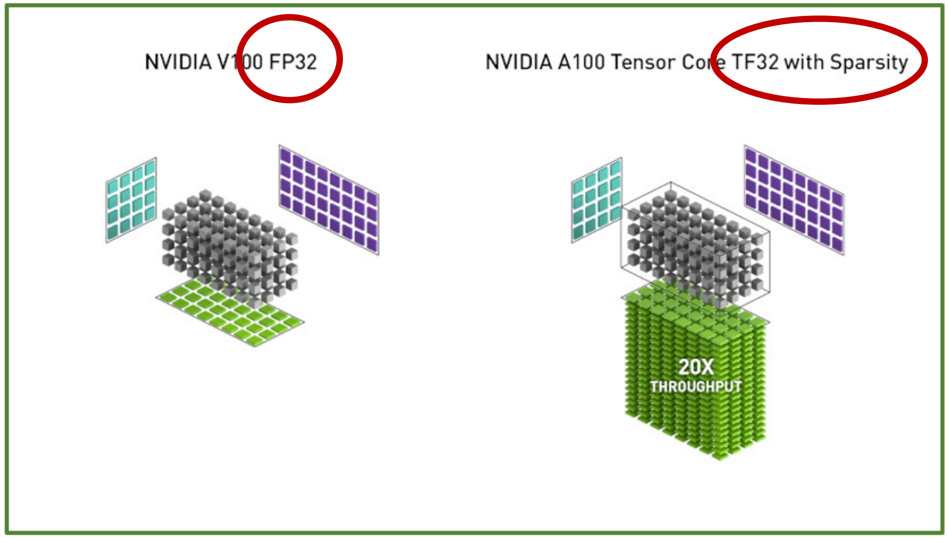
TF32: same range as FP32 and same precision as FP16

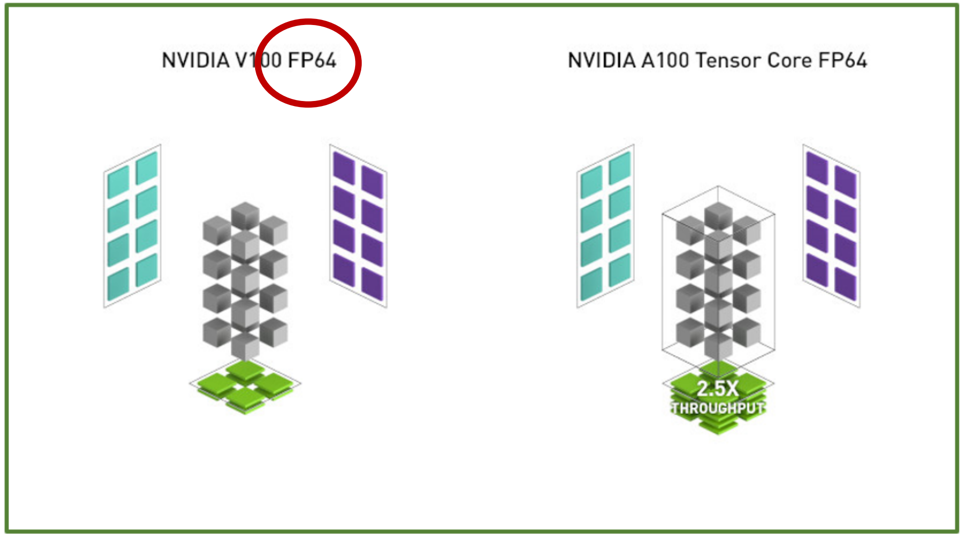
The FP multiplier scales with the square of the mantissa width ($8^2/11^2 \approx 0.5$)

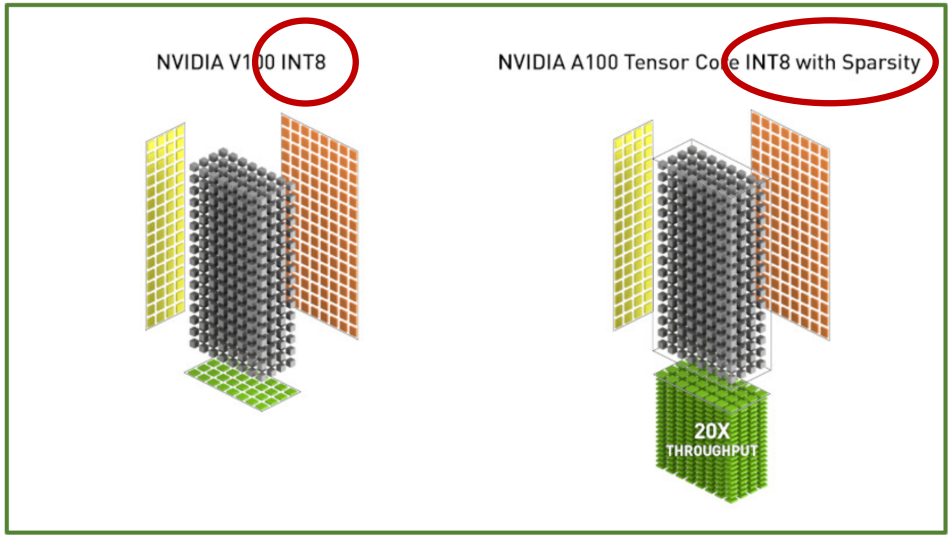
Tensor Cores per SM: Volta vs. Ampere


NVIDIA V100 Tensor Core FP16
NVIDIA A100 Tensor Core FP16 with Sparsity
5X
THROUGHPUT

2 Tensor Cores per SM

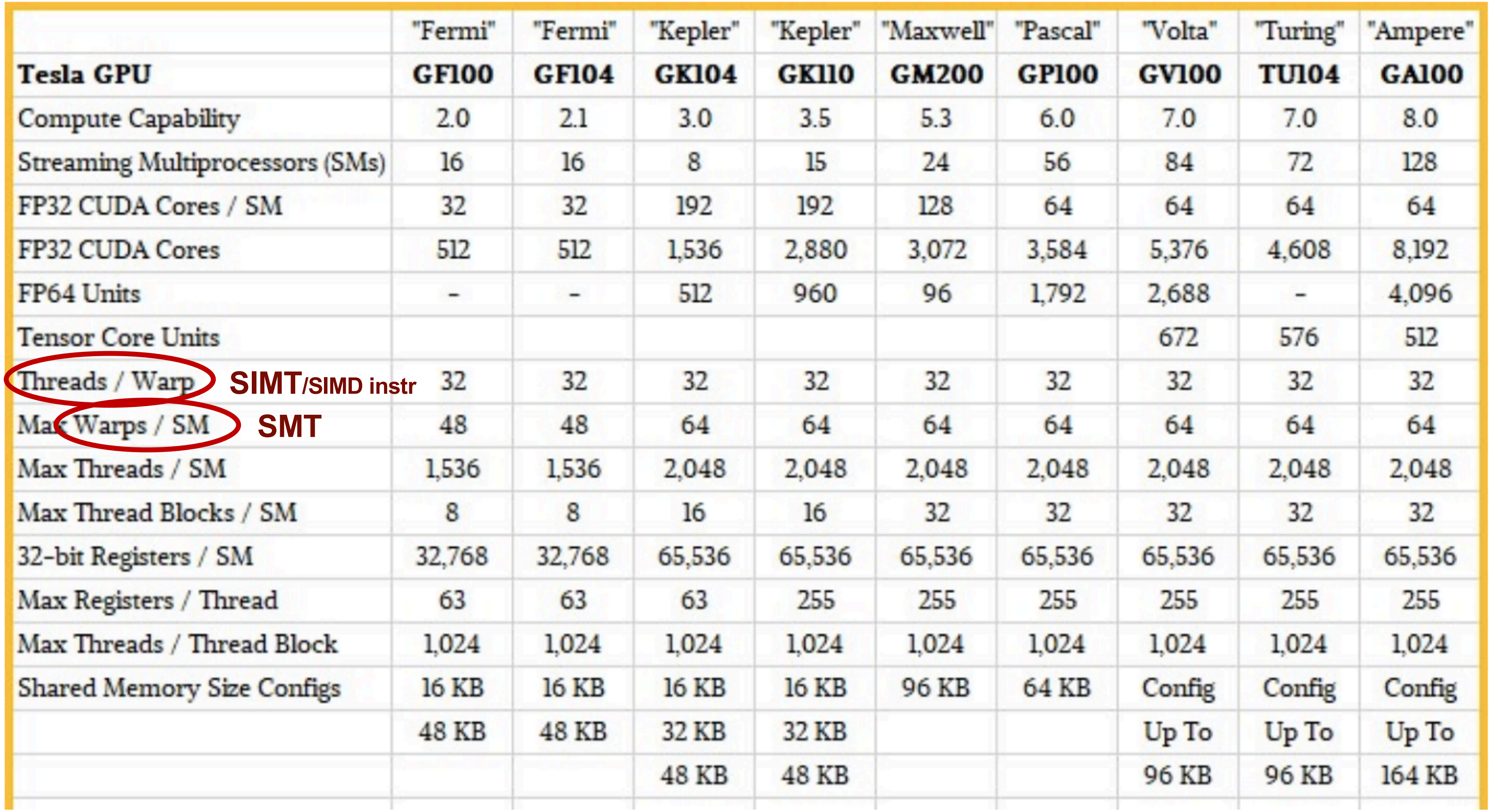






https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

NVidia GPUs



	"Fermi"	"Fermi"	"Kepler"	"Kepler"	"Maxwell"	"Pascal"	"Volta"	"Turing"	"Ampere"
Tesla GPU	**GF100**	**GF104**	**GK104**	**GK110**	**GM200**	**GP100**	**GV100**	**TU104**	**GA100**
Compute Capability	2.0	2.1	3.0	3.5	5.3	6.0	7.0	7.0	8.0
Streaming Multiprocessors (SMs)	16	16	8	15	24	56	84	72	128
FP32 CUDA Cores / SM	32	32	192	192	128	64	64	64	64
FP32 CUDA Cores	512	512	1,536	2,880	3,072	3,584	5,376	4,608	8,192
FP64 Units	-	-	512	960	96	1,792	2,688	-	4,096
Tensor Core Units							672	576	512
Threads / Warp	32	32	32	32	32	32	32	32	32
Max Warps / SM	48	48	64	64	64	64	64	64	64
Max Threads / SM	1,536	1,536	2,048	2,048	2,048	2,048	2,048	2,048	2,048
Max Thread Blocks / SM	8	8	16	16	32	32	32	32	32
32-bit Registers / SM	32,768	32,768	65,536	65,536	65,536	65,536	65,536	65,536	65,536
Max Registers / Thread	63	63	63	255	255	255	255	255	255
Max Threads / Thread Block	1,024	1,024	1,024	1,024	1,024	1,024	1,024	1,024	1,024
Shared Memory Size Configs	16 KB	16 KB	16 KB	16 KB	96 KB	64 KB	Config	Config	Config
	48 KB	48 KB	32 KB	32 KB			Up To	Up To	Up To
			48 KB	48 KB			96 KB	96 KB	164 KB

SIMT/SIMD instr

SMT

https://www.nextplatform.com/2020/05/28/diving-deep-into-the-nvidia-ampere-gpu-architecture/

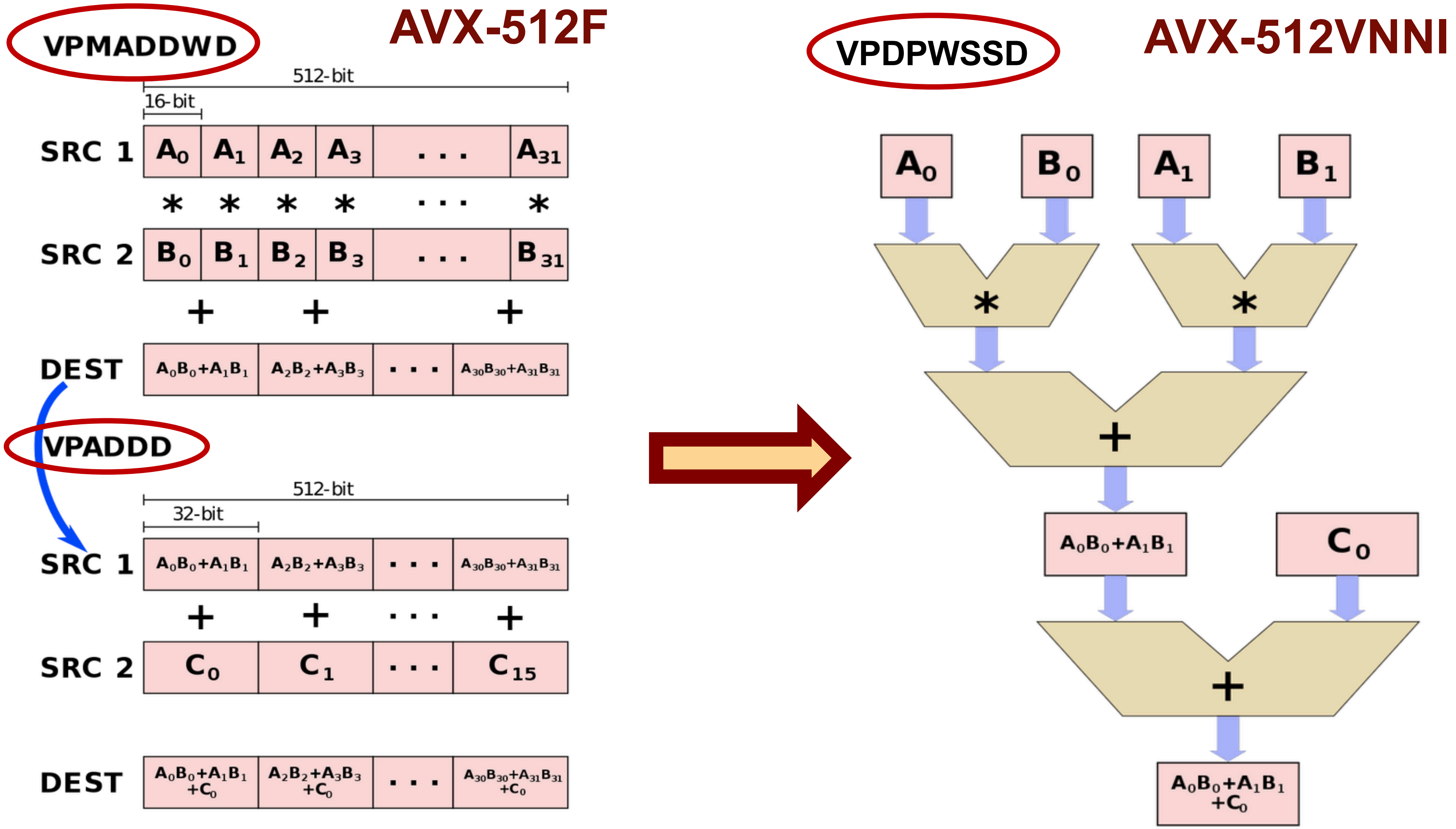
Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000,
      Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

Intel AVX-512:
Vector Neural Network Instructions (VNNI)

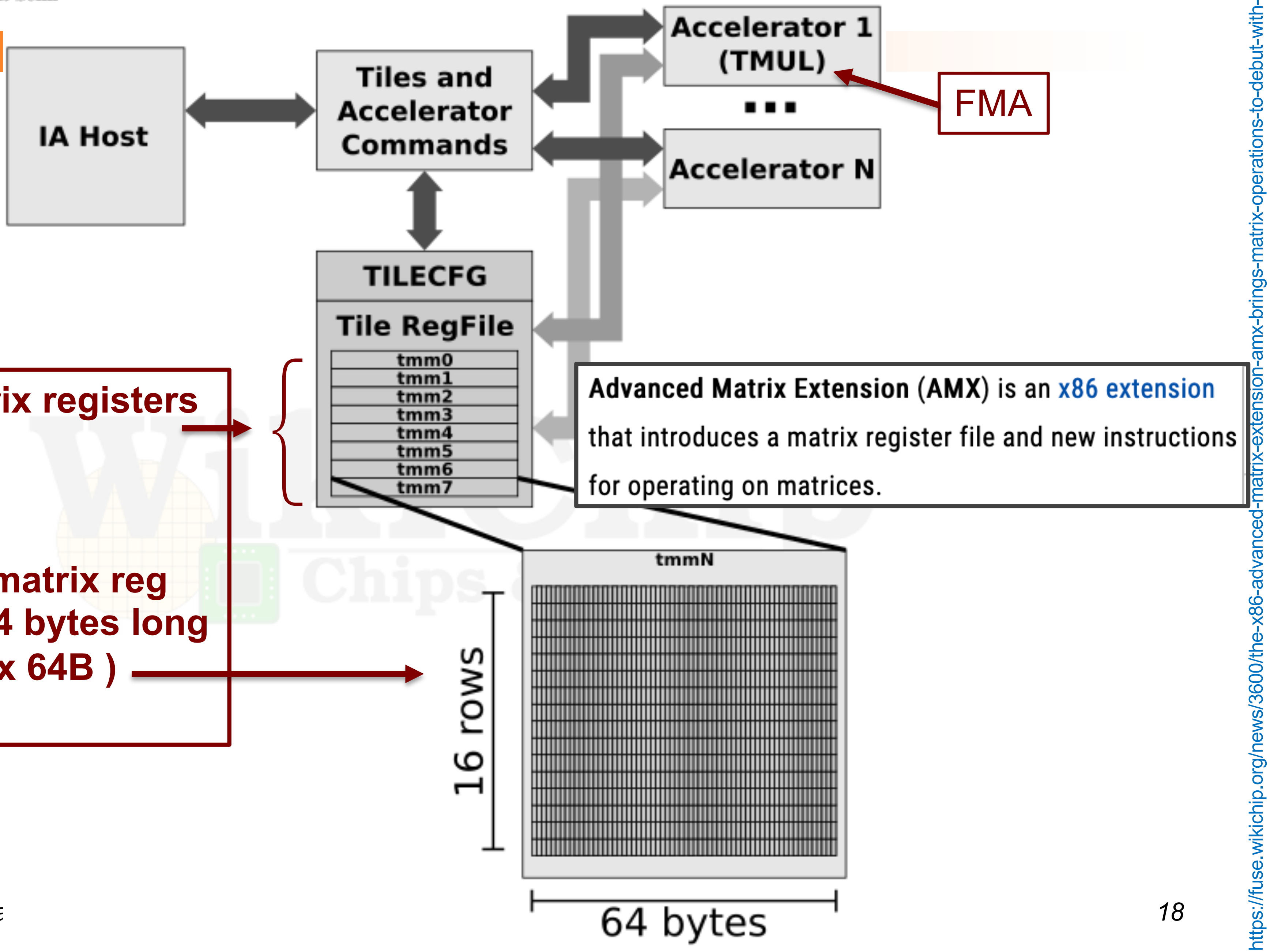
VNNI: 2 new instr + 2 new extensions to merge previous set of 2 & 3 instr; ex.:



https://en.wikichip.org/wiki/x86/avx512_vnni

The Intel Advanced Matrix Extension (AMX)

(expected in 2021)



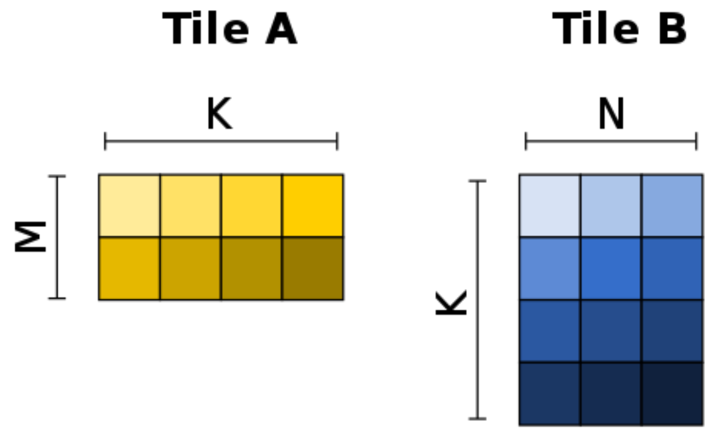
Advanced Matrix Extension (AMX) is an x86 extension that introduces a matrix register file and new instructions for operating on matrices.

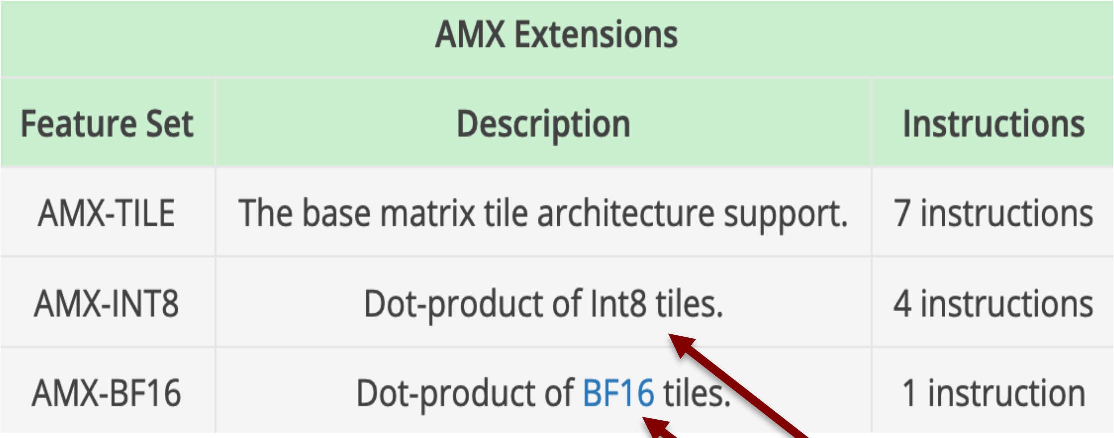
8 matrix registers

Each matrix reg is 1024 bytes long (= 16 x 64B)

https://fuse.wikichip.org/news/3600/the-x86-advanced-matrix-extension-amx-brings-matrix-operations-to-debut-with-sapphire-rapids/

AMX instructions in Accelerator 1

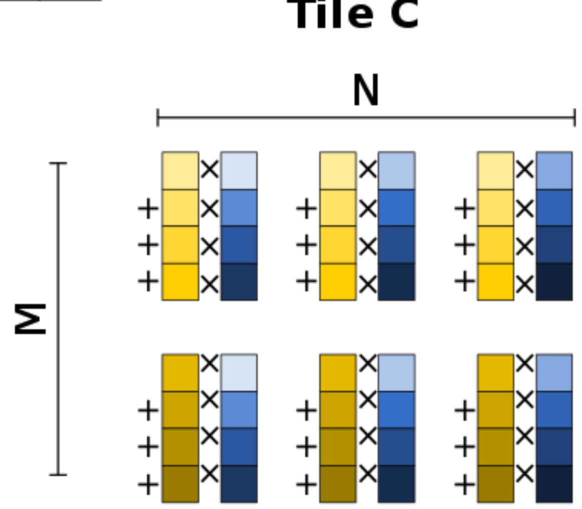


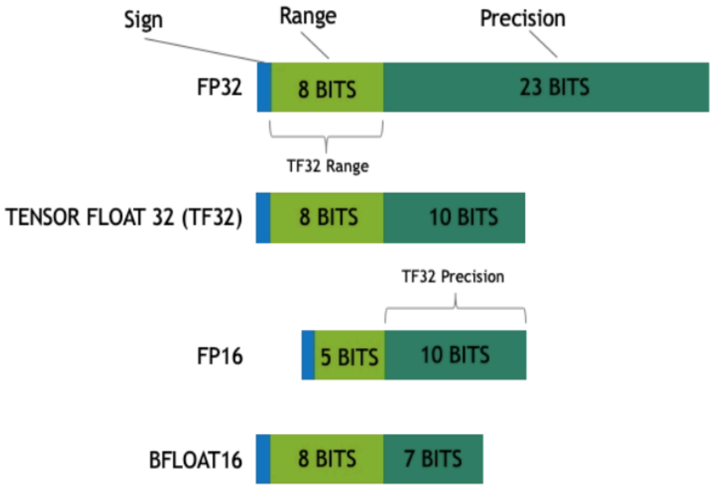


AMX Extensions		
Feature Set	**Description**	**Instructions**
AMX-TILE	The base matrix tile architecture support.	7 instructions
AMX-INT8	Dot-product of Int8 tiles.	4 instructions
AMX-BF16	Dot-product of BF16 tiles.	1 instruction

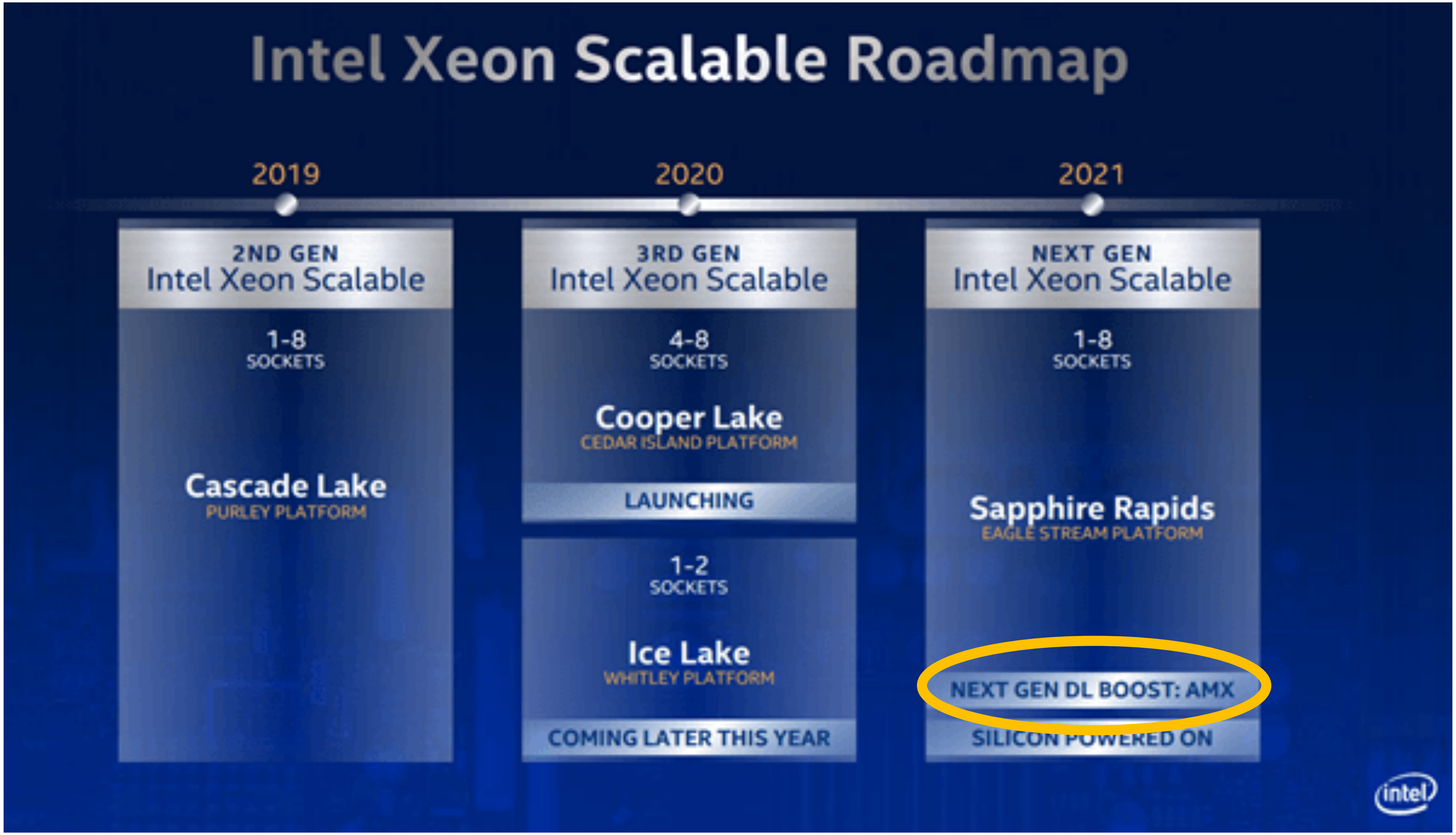
Data types

TMUL
C += A * B





Intel Scalable Xeon roadmap



Intel Xeon Scalable Roadmap

2019	2020	2021
2ND GEN Intel Xeon Scalable	3RD GEN Intel Xeon Scalable	NEXT GEN Intel Xeon Scalable
1-8 SOCKETS	4-8 SOCKETS Cooper Lake CEDAR ISLAND PLATFORM LAUNCHING	1-8 SOCKETS
Cascade Lake PURLEY PLATFORM	1-2 SOCKETS Ice Lake WHITLEY PLATFORM COMING LATER THIS YEAR	Sapphire Rapids EAGLE STREAM PLATFORM NEXT GEN DL BOOST: AMX SILICON POWERED ON

Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana,
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

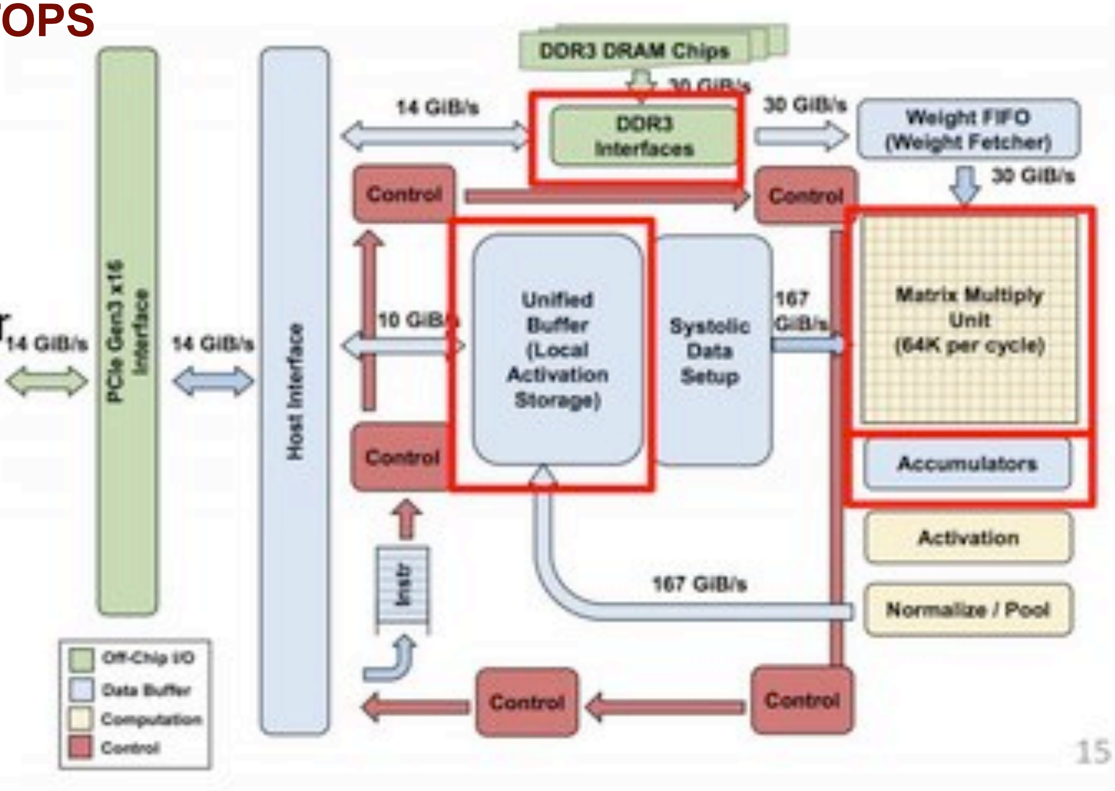
Google

Tensor Processing Unit

Google Tensor Processing Unit, TPU (May'16)

- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units ← **INT8**
- 700 MHz clock rate **FMA**
- Peak: 92T operations/second
  - ○ 65,536 * 2 * 700M → **92 TOPS**
- >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) **SRAM**
- 3.5X as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GiB of off-chip weight DRAM memory

TPU: High-level Chip Architecture



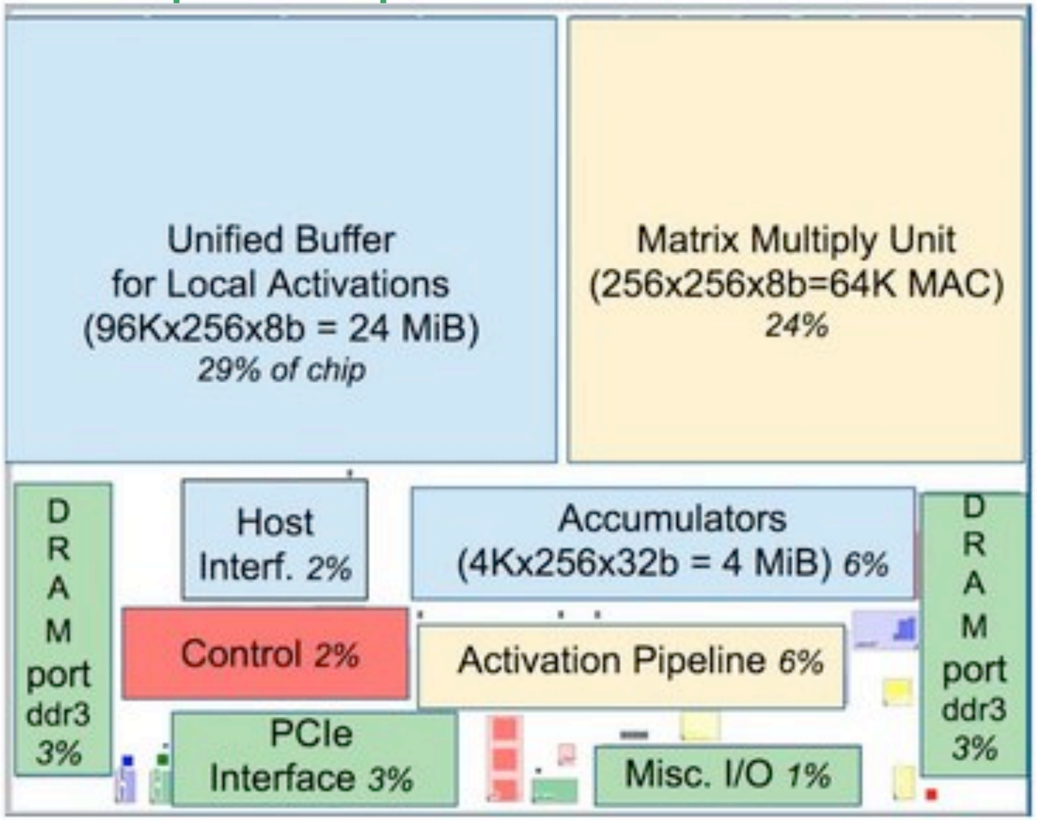
Google

Tensor Processing Unit

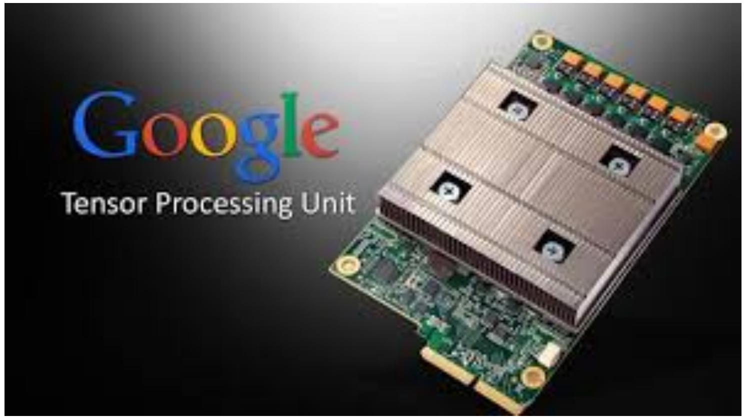
Google TPU

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

Chip floor plan



TPU: a Neural Network Accelerator Chip



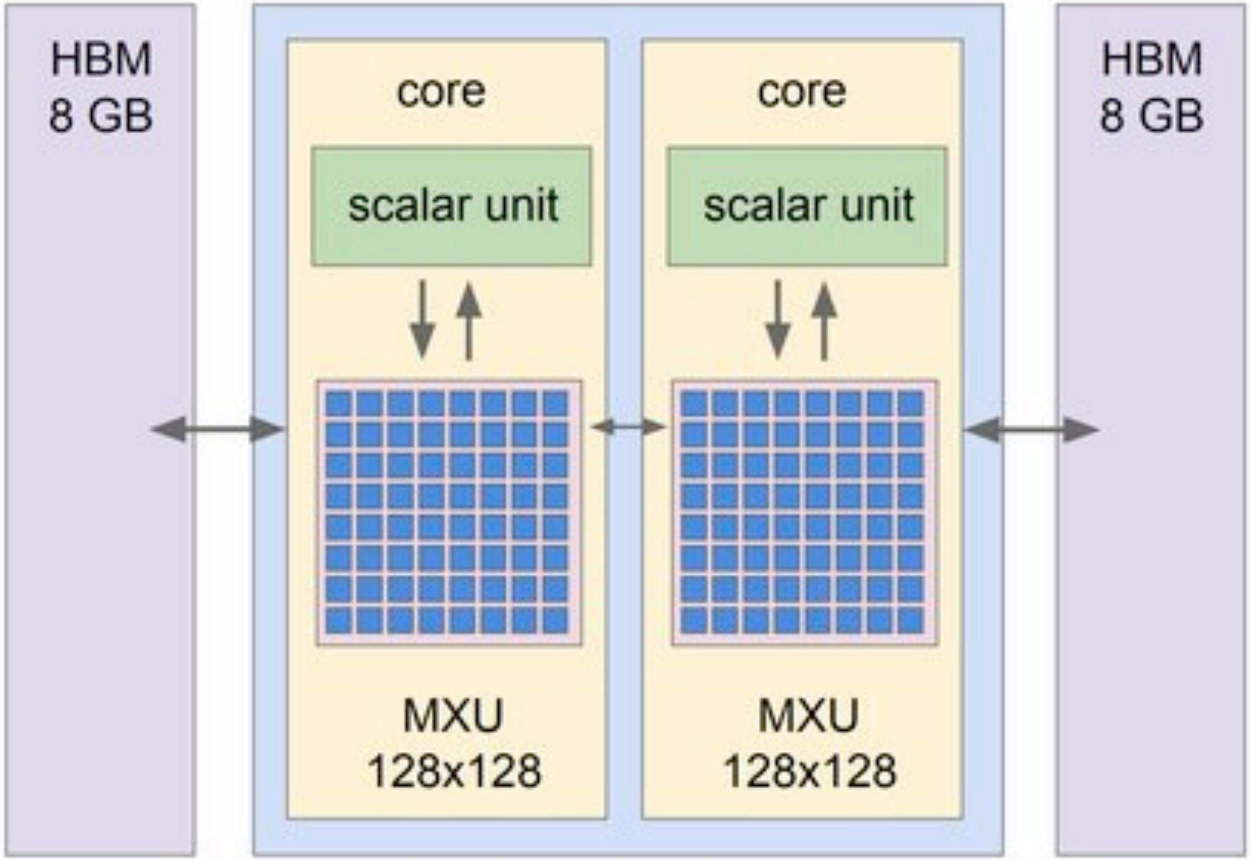
TPUs are intensively used by Google, namely in Google Photos, RankBrain, StreetView & Google Translate

Google TPUv2 *(May'17)*

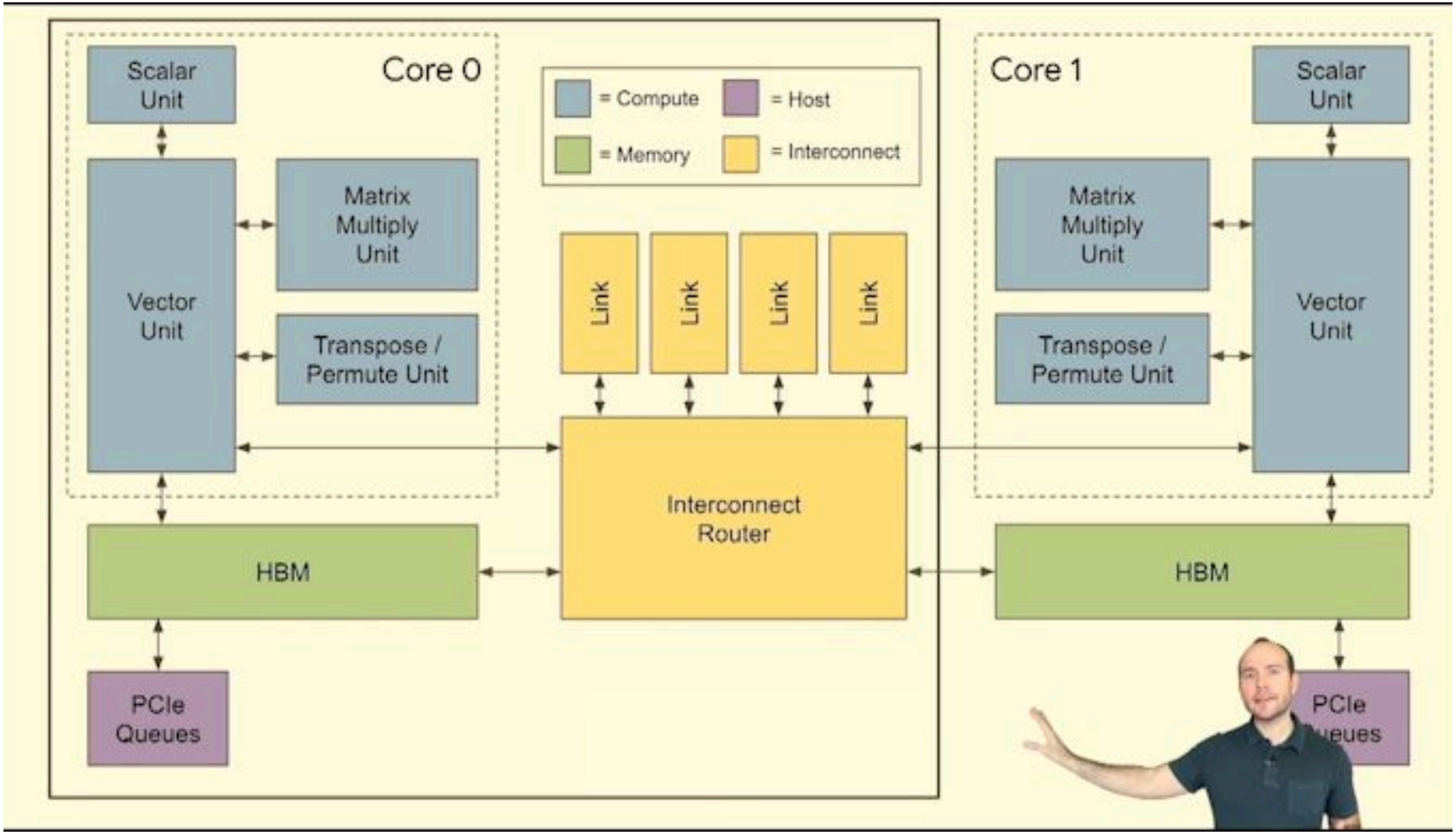
TPUv2 Chip

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

bfloat

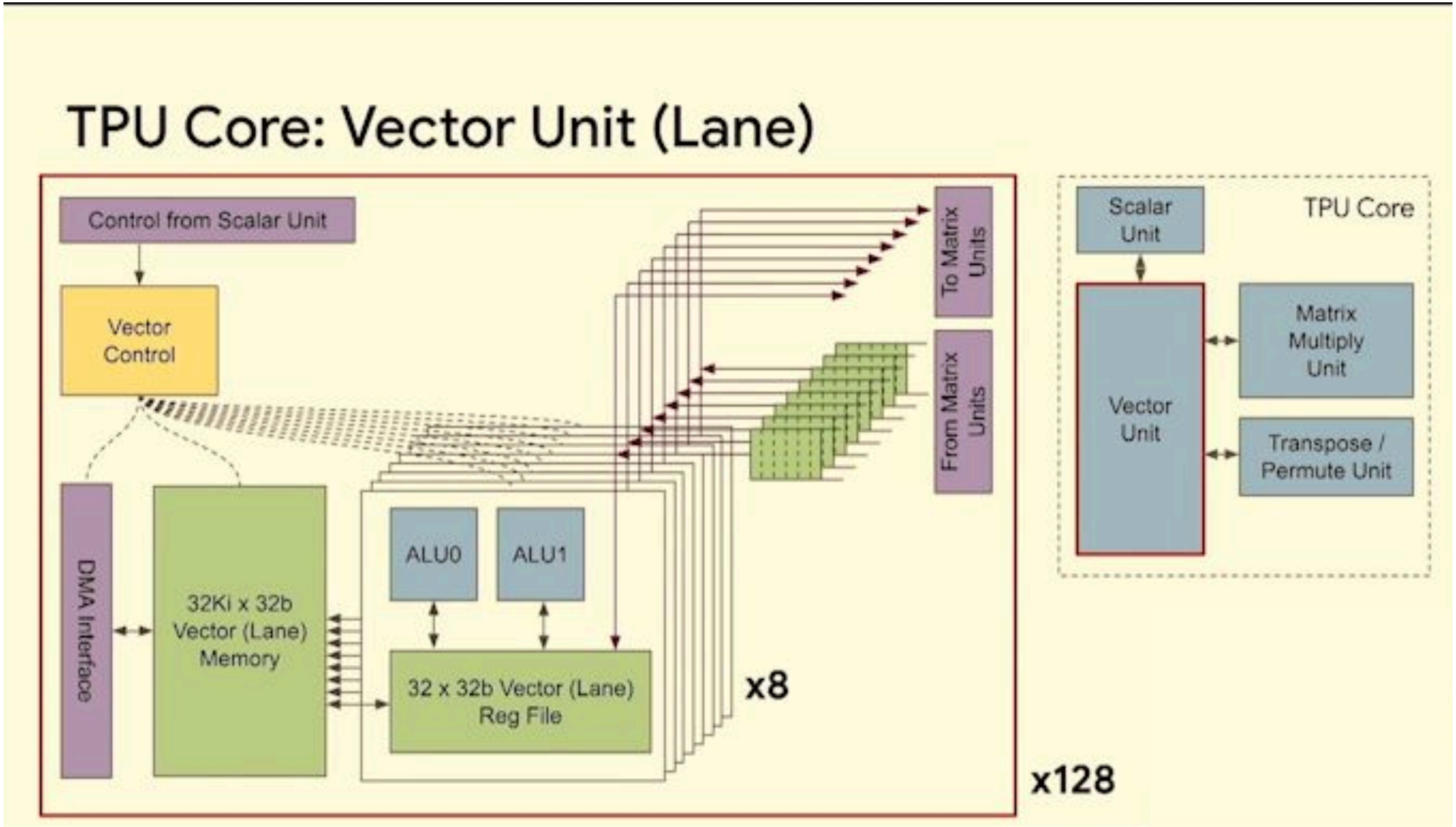


TPUv2 architecture



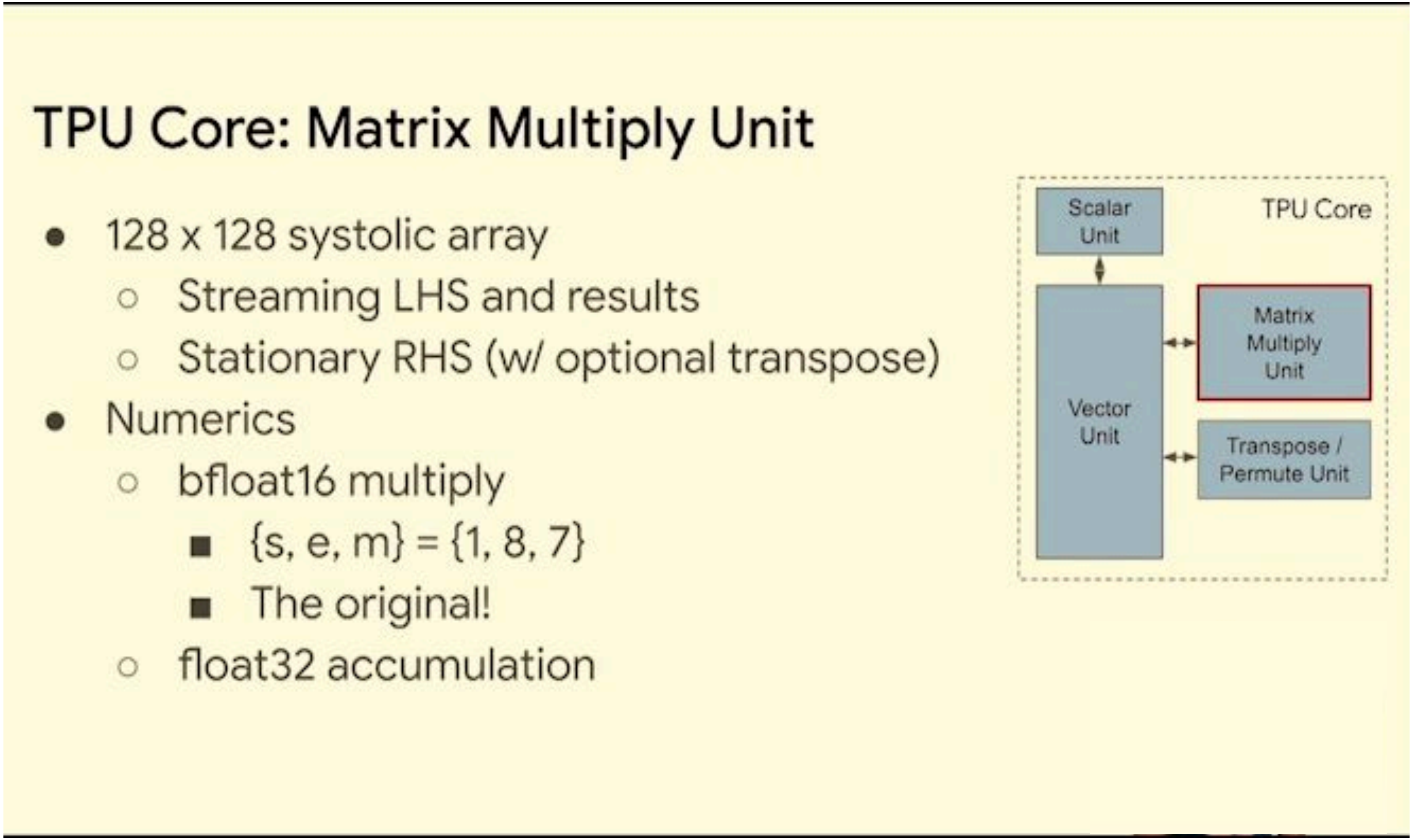
https://www.anandtech.com/show/16005/hot-chips-2020-live-blog-google-tpuv2-and-tpuv3-230pm-pt

TPUv2 core:
1 Lane of the Vector Unit



https://www.anandtech.com/show/16005/hot-chips-2020-live-blog-google-tpuv2-and-tpuv3-230pm-pt

TPUv2 core: the Matrix Multiply Unit

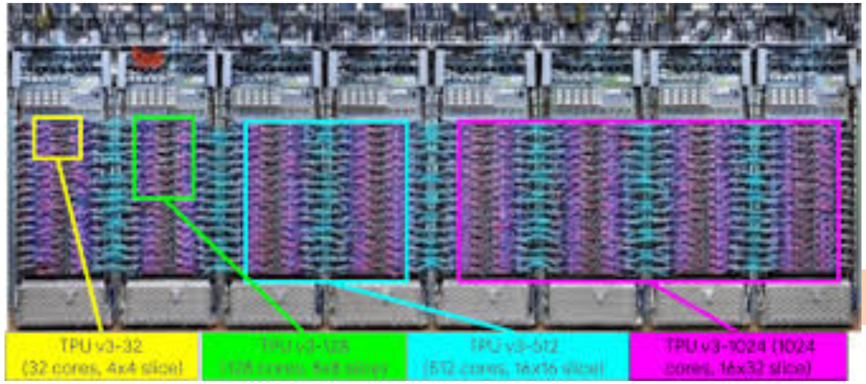


TPU Core: Matrix Multiply Unit

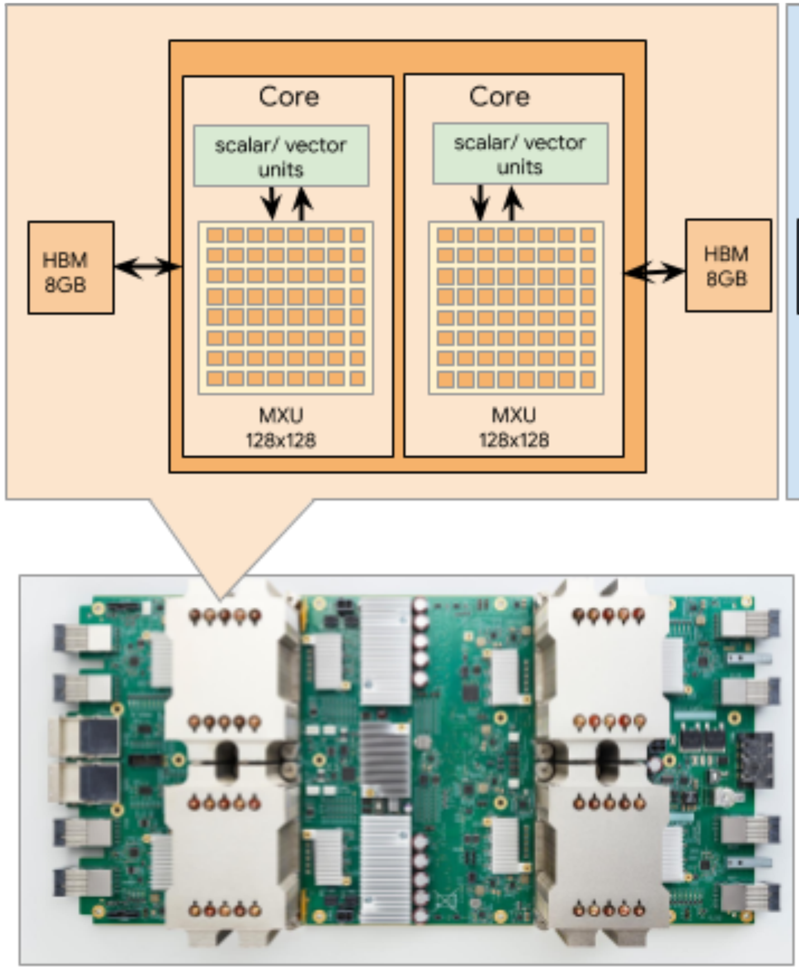
- 128 x 128 systolic array
  - Streaming LHS and results
  - Stationary RHS (w/ optional transpose)
- Numerics
  - bfloat16 multiply
    - {s, e, m} = {1, 8, 7}
    - The original!
  - float32 accumulation

https://www.anandtech.com/show/16005/hot-chips-2020-live-blog-google-tpuv2-and-tpuv3-230pm-pt

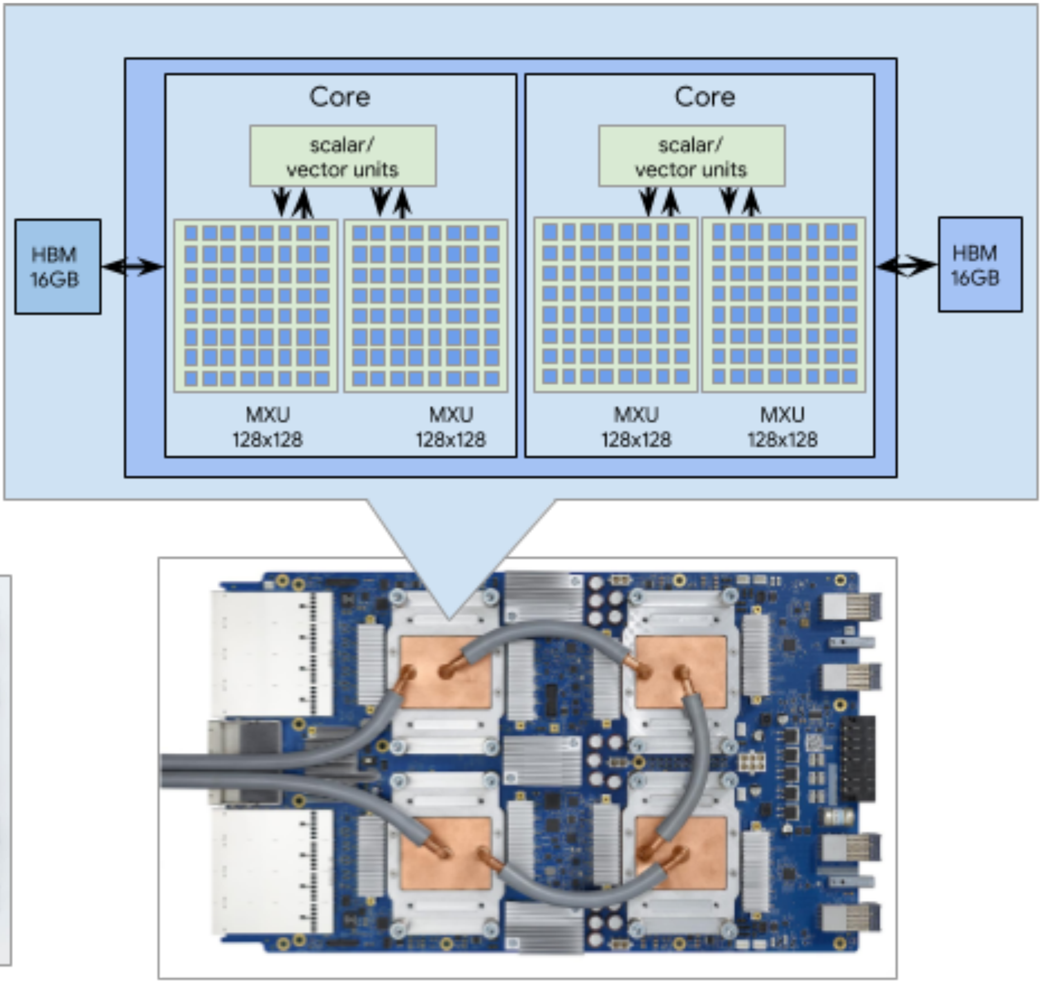
Google TPUv3 (May’18)



TPUv4 released Jun 2020
but no data available yet…

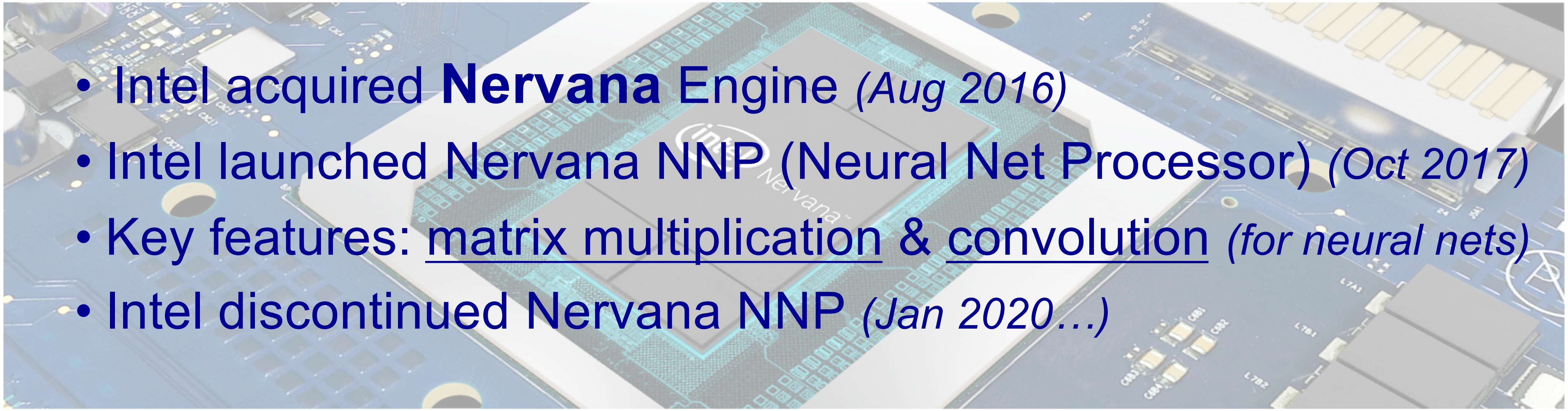


TPU v2 - 4 chips, 2 cores per chip

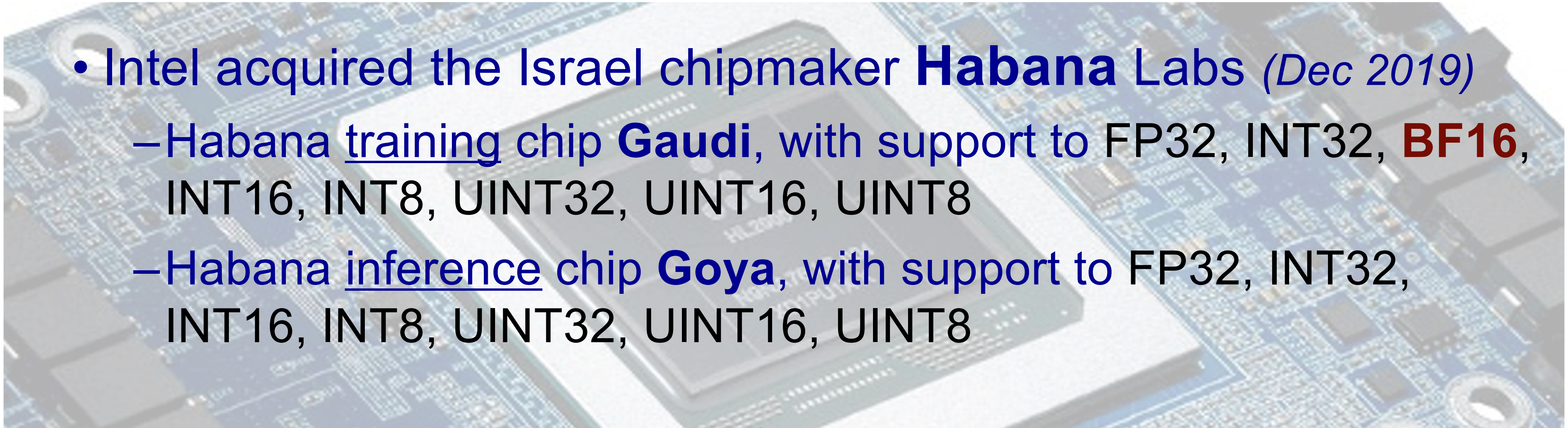


TPU v3 - 4 chips, 2 cores per chip

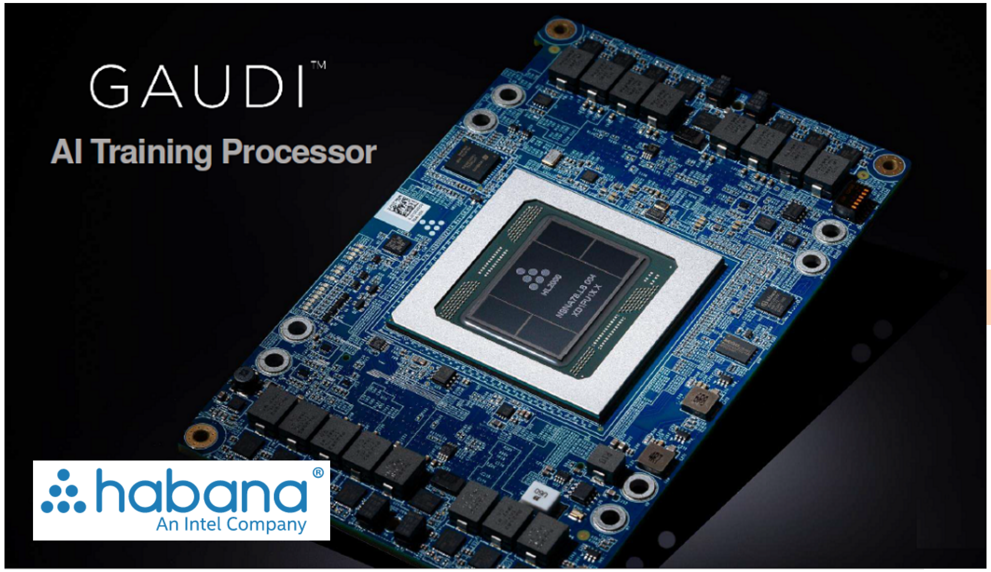
Neural net devices at Intel: Nervana & later Habana chips



- Intel acquired **Nervana** Engine *(Aug 2016)*
- Intel launched Nervana NNP (Neural Net Processor) *(Oct 2017)*
- Key features: matrix multiplication & convolution *(for neural nets)*
- Intel discontinued Nervana NNP *(Jan 2020…)*



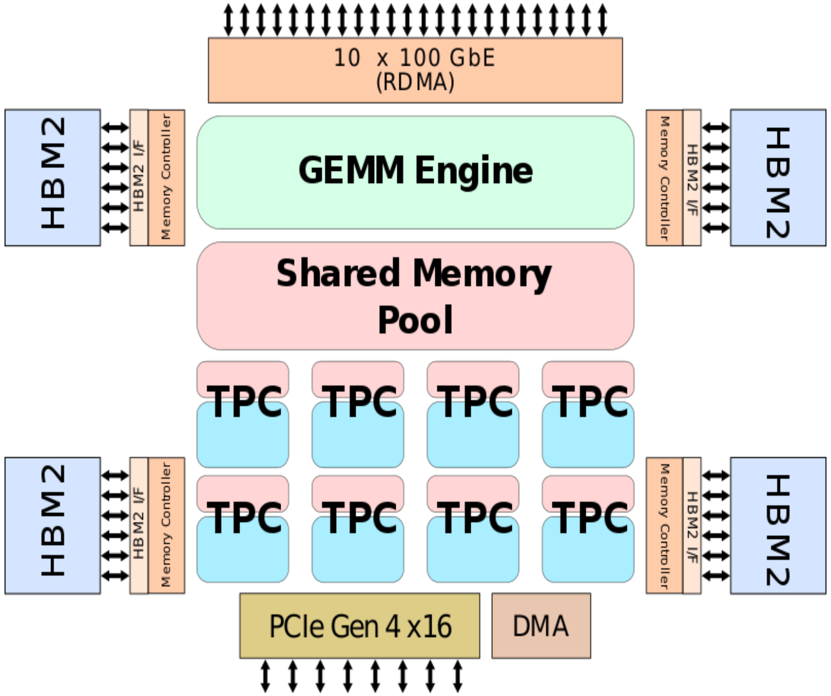
- Intel acquired the Israel chipmaker **Habana** Labs *(Dec 2019)*
  - Habana training chip **Gaudi**, with support to FP32, INT32, **BF16**, INT16, INT8, UINT32, UINT16, UINT8
  - Habana inference chip **Goya**, with support to FP32, INT32, INT16, INT8, UINT32, UINT16, UINT8

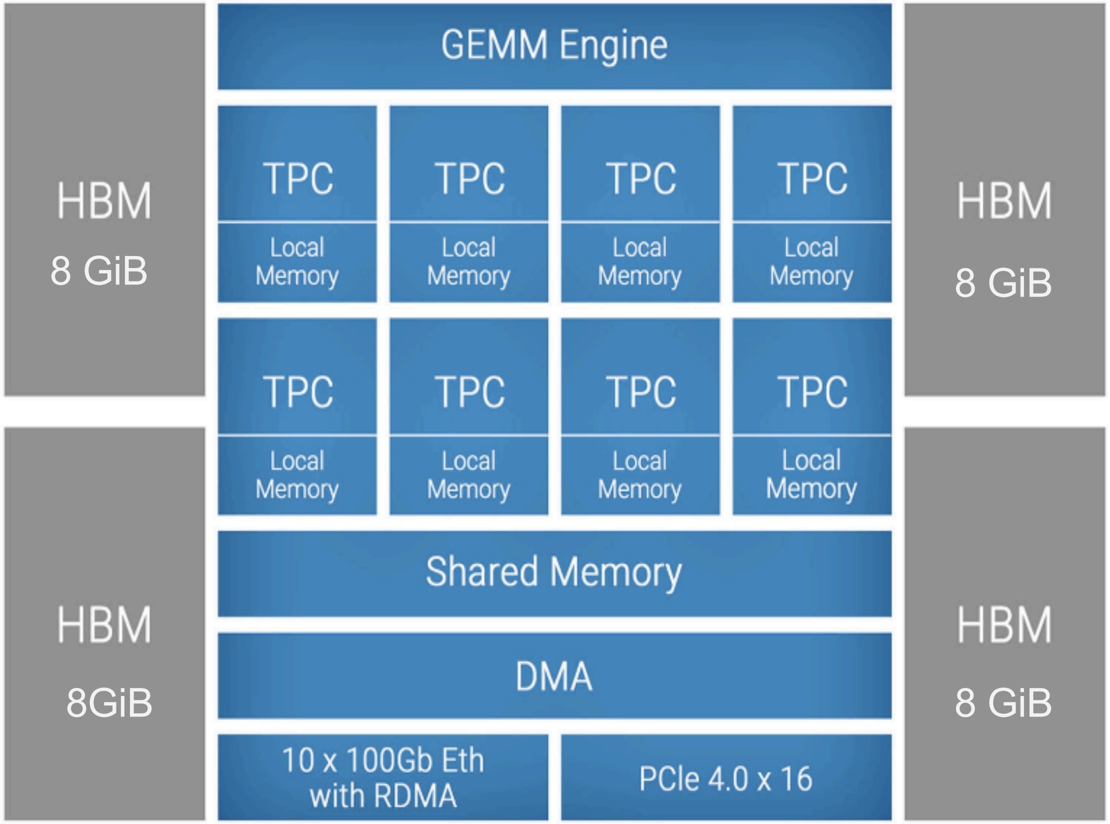


Training Processor: GAUDI

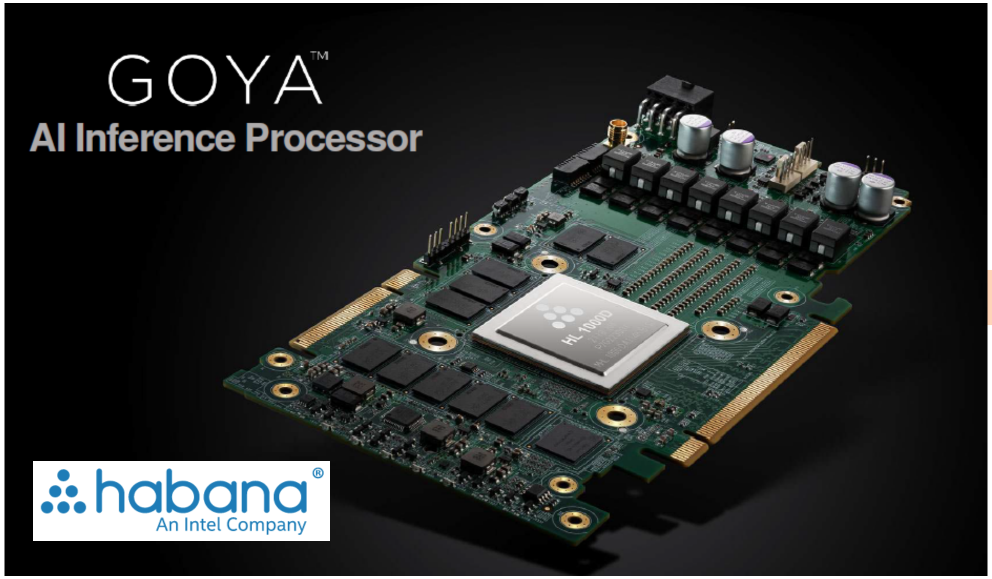
habana GAUDI™

https://en.wikichip.org/wiki/habana/microarchitectures/gaudi





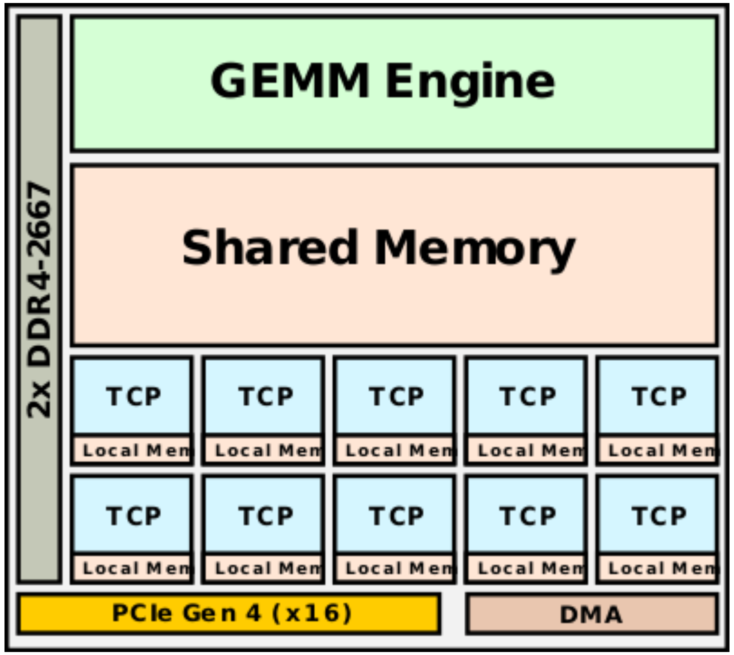
TPC: Tensor Processing Core
GEMM: General Matrix Multiply



Inference Processor: GOYA

habana GOYA™

https://en.wikichip.org/wiki/habana/microarchitectures/goya



2x DDR4-2667

GEMM Engine

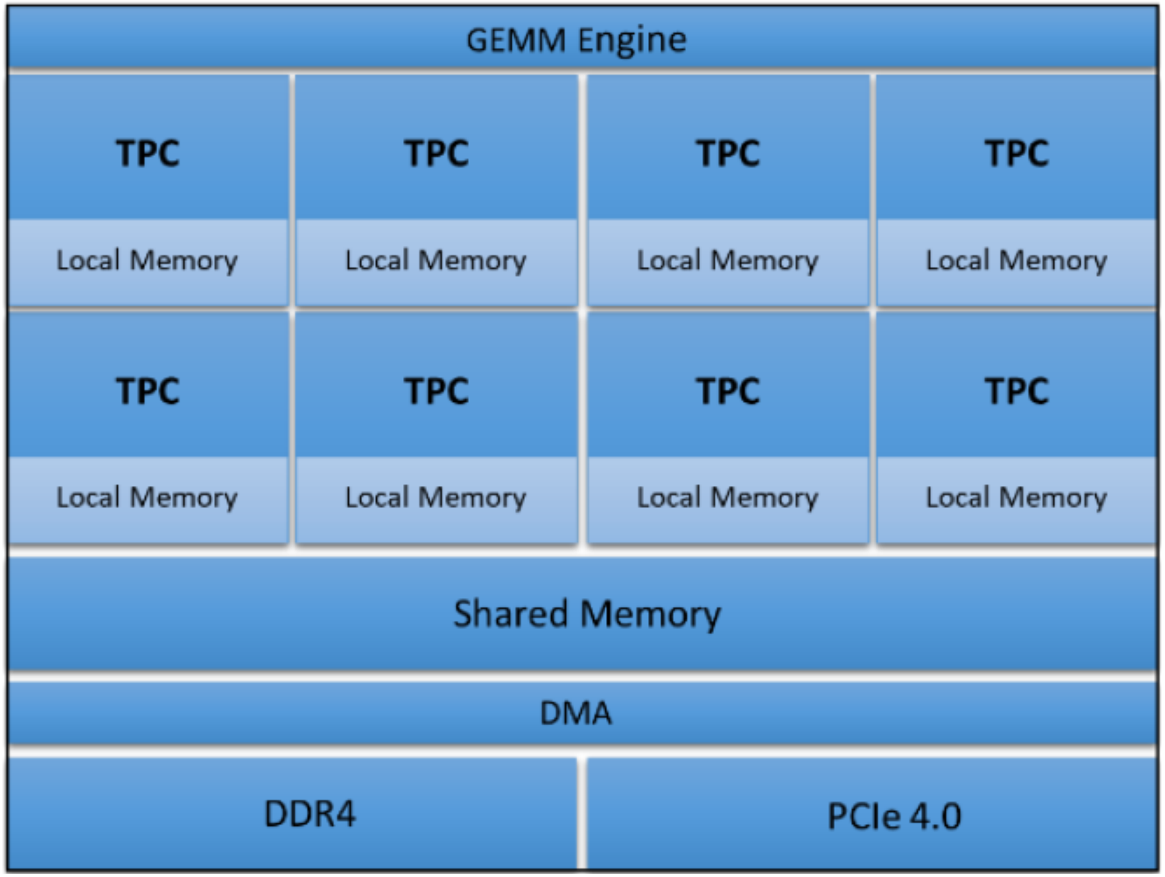
Shared Memory

TCP Local Men | TCP Local Men | TCP Local Men | TCP Local Men | TCP Local Men

TCP Local Men | TCP Local Men | TCP Local Men | TCP Local Men | TCP Local Men

PCIe Gen 4 (x16)

DMA



GEMM Engine

TPC Local Memory | TPC Local Memory | TPC Local Memory | TPC Local Memory

TPC Local Memory | TPC Local Memory | TPC Local Memory | TPC Local Memory

Shared Memory

DMA

DDR4

PCIe 4.0

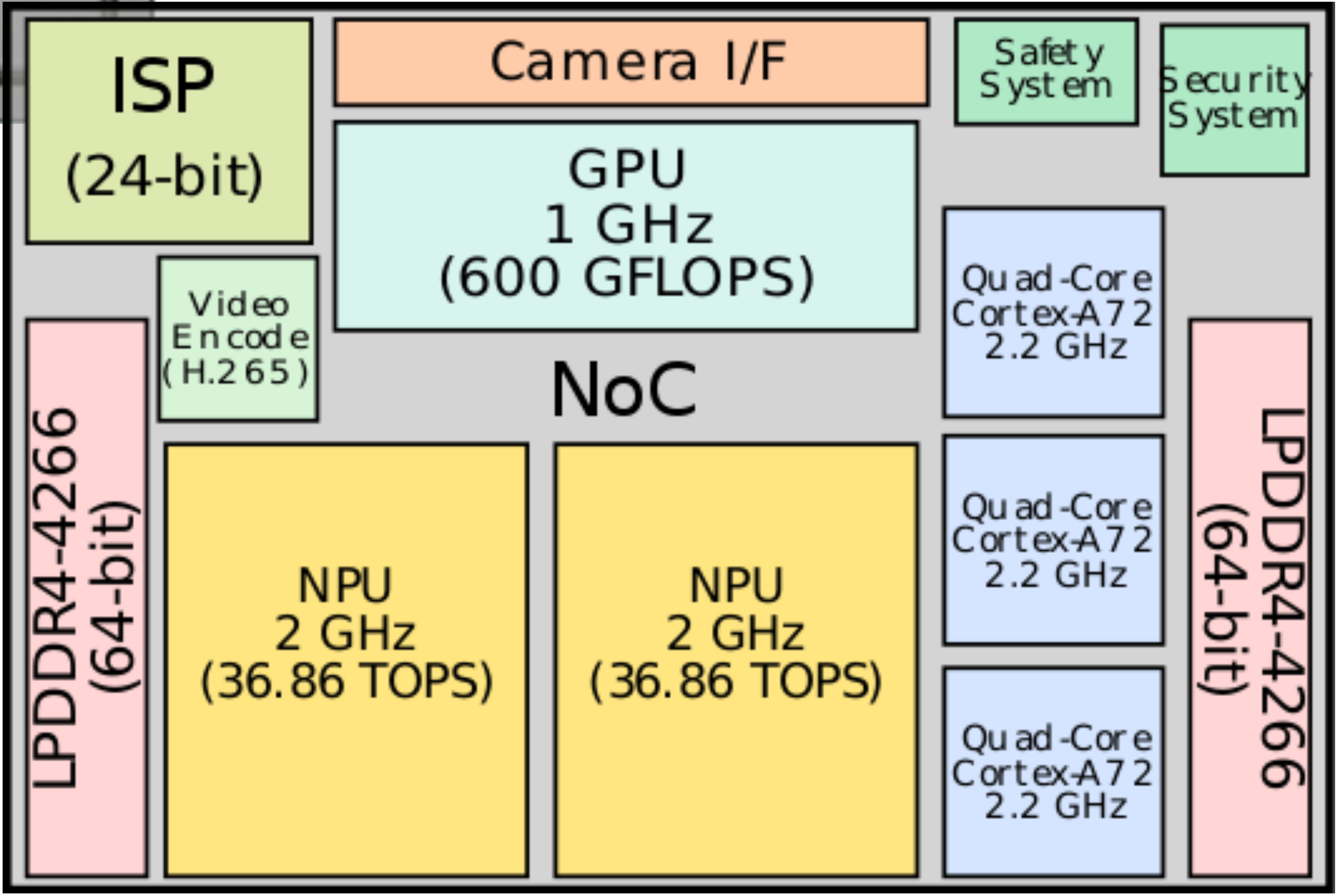
TPC: Tensor Processing Core
GEMM: General Matrix Multiply

Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

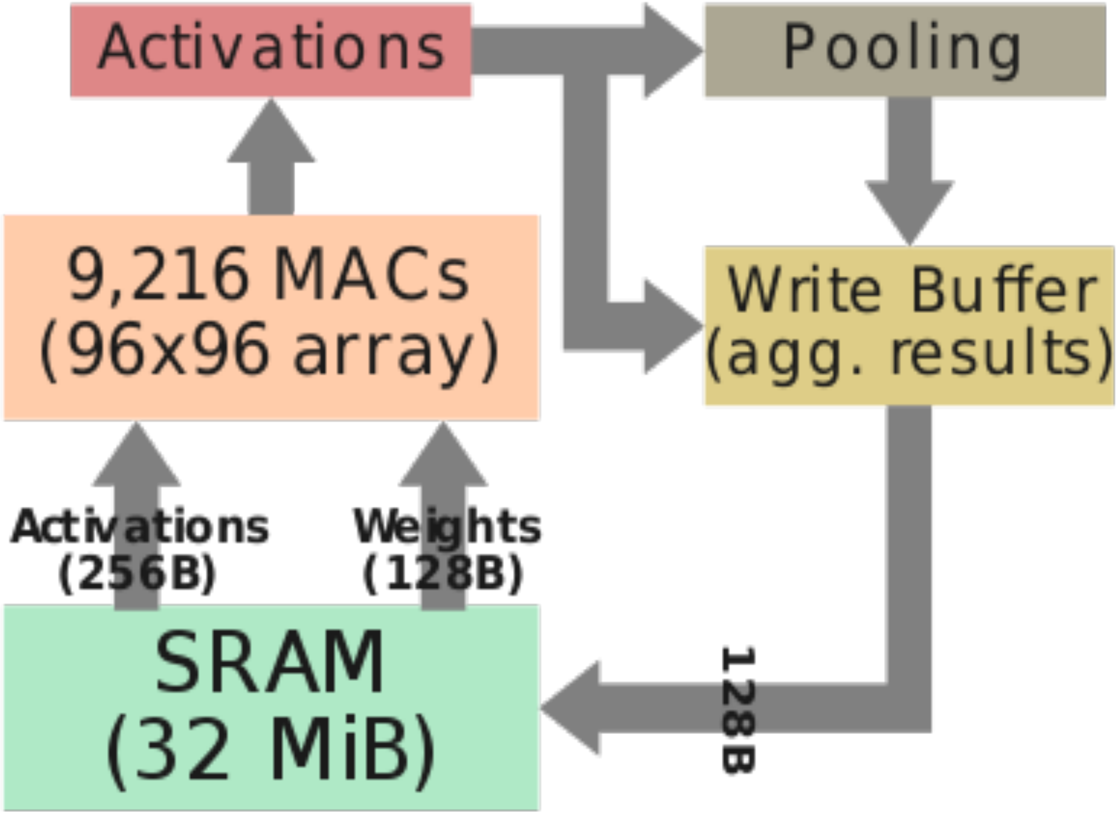
- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming

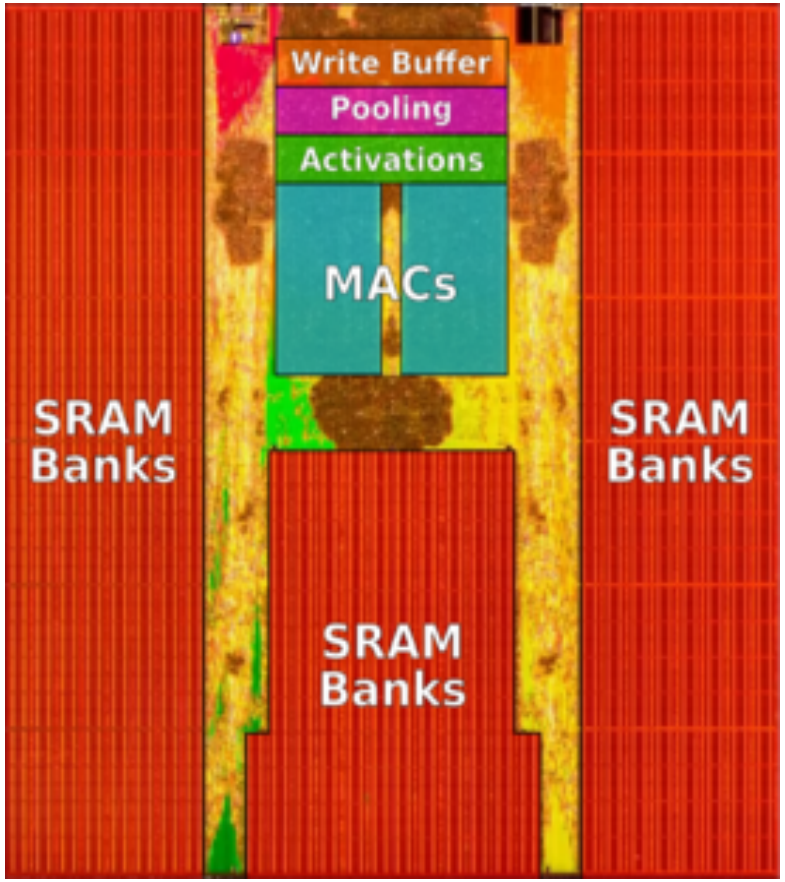
Tesla Full Self-Driving chip (FSD)



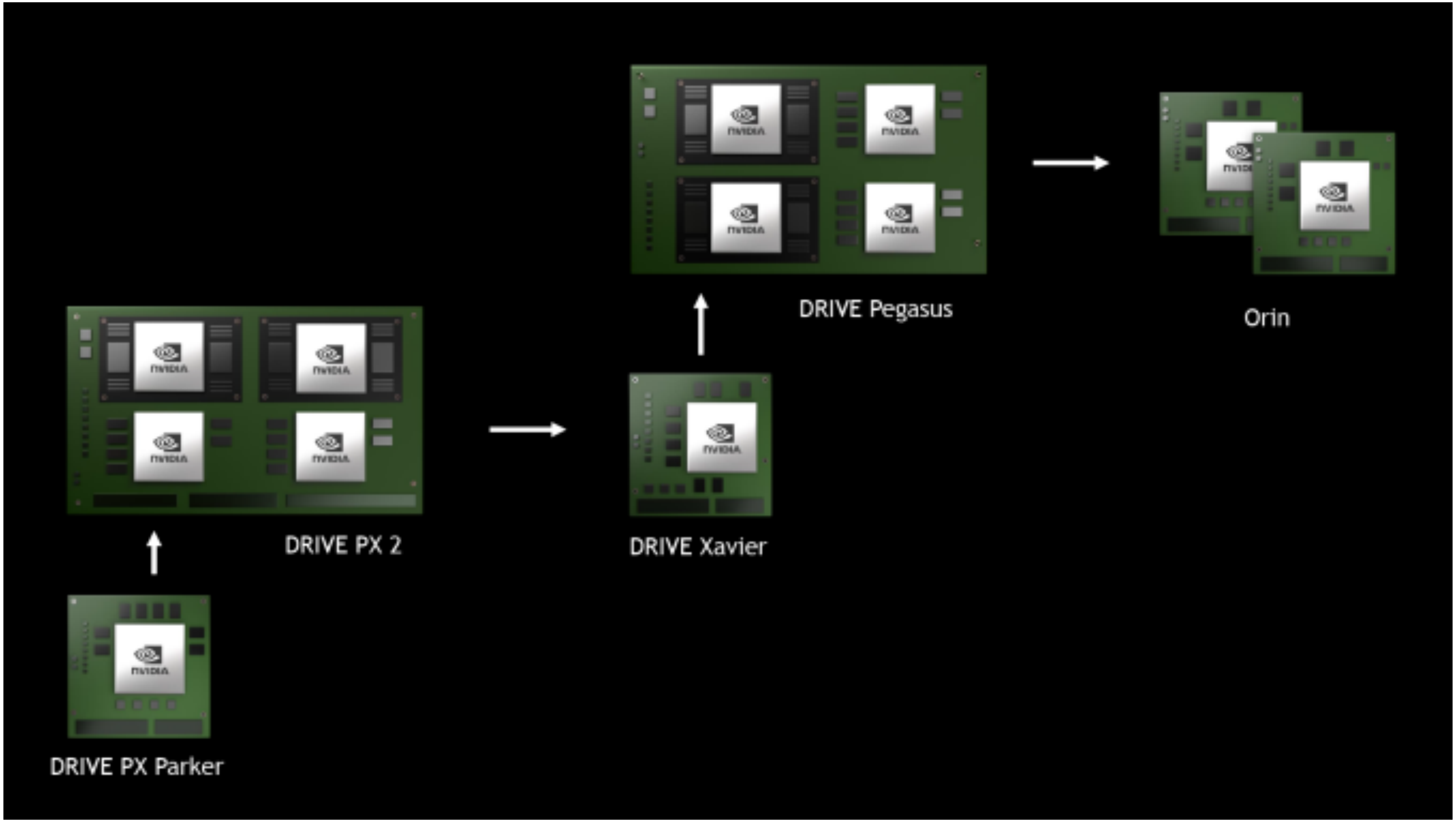
https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip

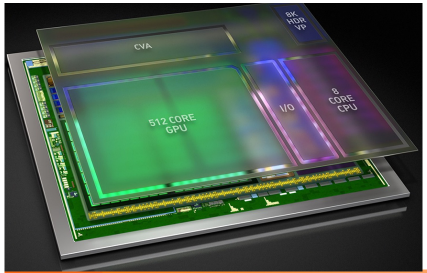
The Neural Processing Unit in FSD



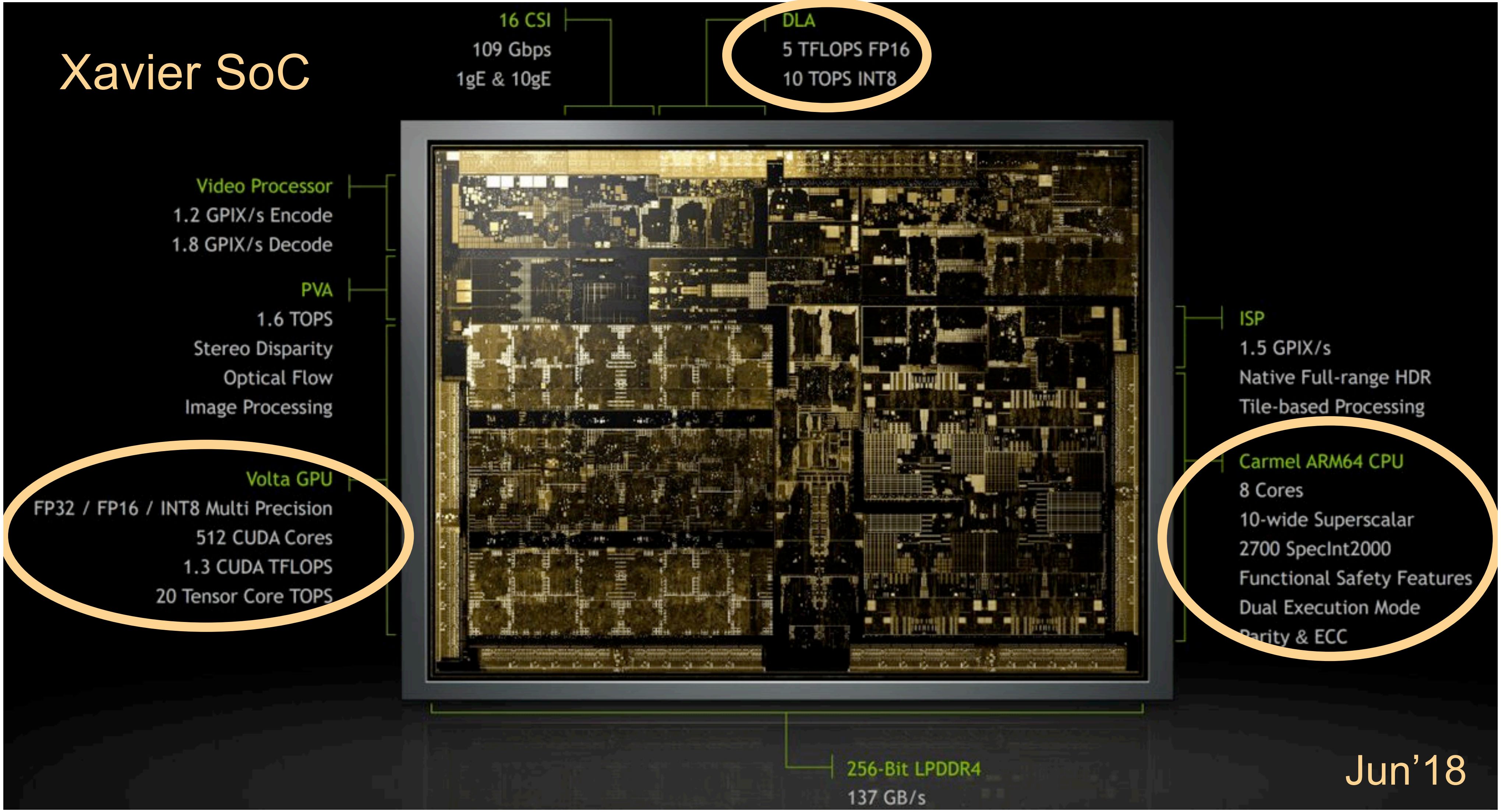


NVidia roadmap for Drive systems





NVidia Xavier SoC



Xavier SoC

16 CSI
109 Gbps
1gE & 10gE

DLA
5 TFLOPS FP16
10 TOPS INT8

Video Processor
1.2 GPIX/s Encode
1.8 GPIX/s Decode

PVA
1.6 TOPS
Stereo Disparity
Optical Flow
Image Processing

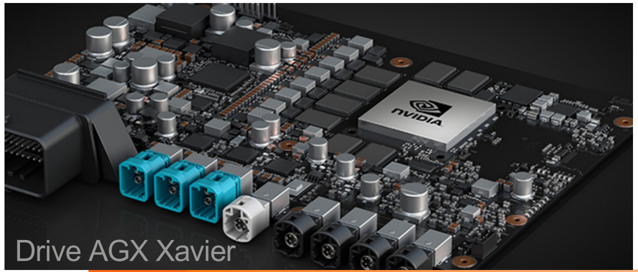
Volta GPU
FP32 / FP16 / INT8 Multi Precision
512 CUDA Cores
1.3 CUDA TFLOPS
20 Tensor Core TOPS

ISP
1.5 GPIX/s
Native Full-range HDR
Tile-based Processing

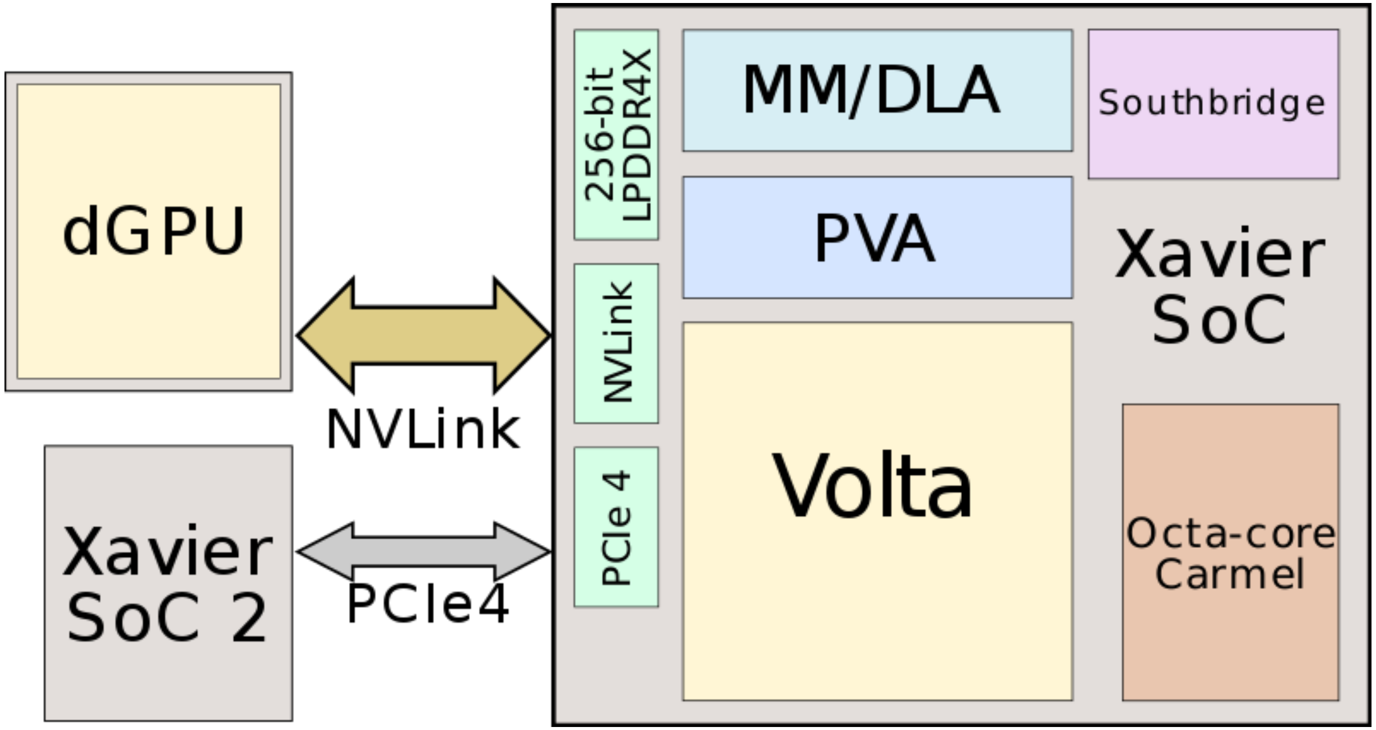
Carmel ARM64 CPU
8 Cores
10-wide Superscalar
2700 SpecInt2000
Functional Safety Features
Dual Execution Mode
Parity & ECC

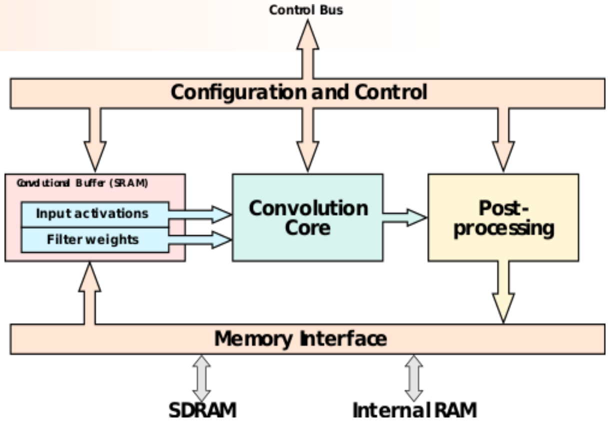
256-Bit LPDDR4
137 GB/s

Jun’18



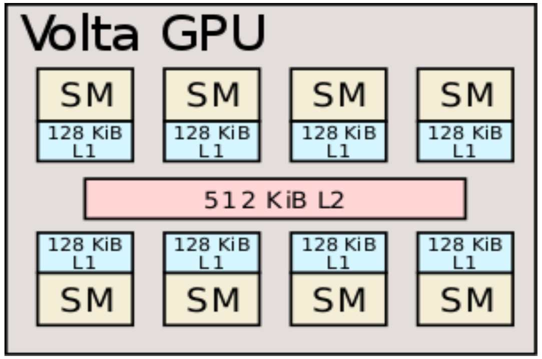
NVidia Xavier SoC

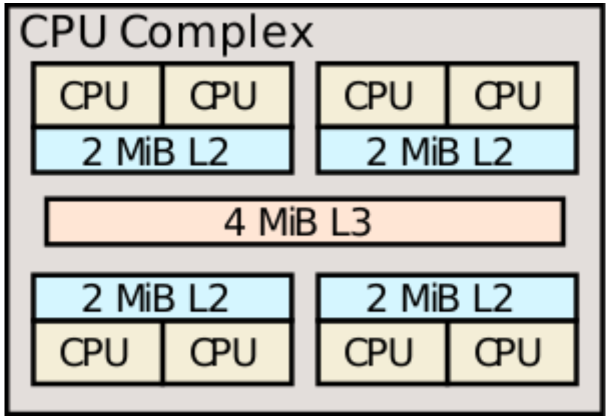




Deep **L**earning **A**ccelerator

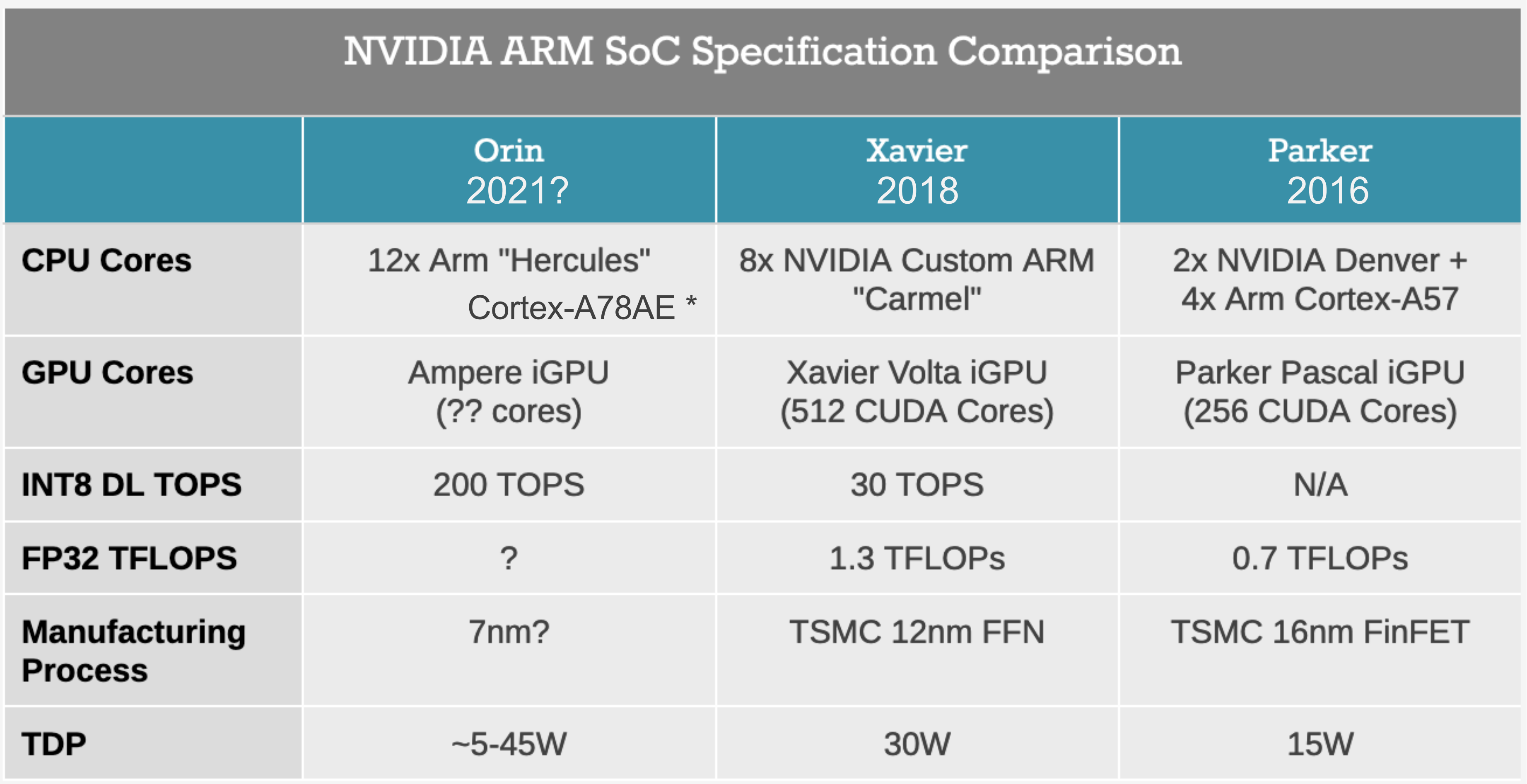
Large array of multiply-accumulate units optimized for CNNs





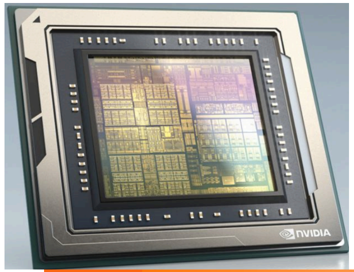
https://fuse.wikichip.org/news/1618/hot-chips-30-nvidia-xavier-soc/

NVidia Drive SoC’s: Parker, Xavier, Orion



NVIDIA ARM SoC Specification Comparison			
	Orin 2021?	Xavier 2018	Parker 2016
CPU Cores	12x Arm "Hercules" Cortex-A78AE *	8x NVIDIA Custom ARM "Carmel"	2x NVIDIA Denver + 4x Arm Cortex-A57
GPU Cores	Ampere iGPU (?? cores)	Xavier Volta iGPU (512 CUDA Cores)	Parker Pascal iGPU (256 CUDA Cores)
INT8 DL TOPS	200 TOPS	30 TOPS	N/A
FP32 TFLOPS	?	1.3 TFLOPs	0.7 TFLOPs
Manufacturing Process	7nm?	TSMC 12nm FFN	TSMC 16nm FinFET
TDP	~5-45W	30W	15W

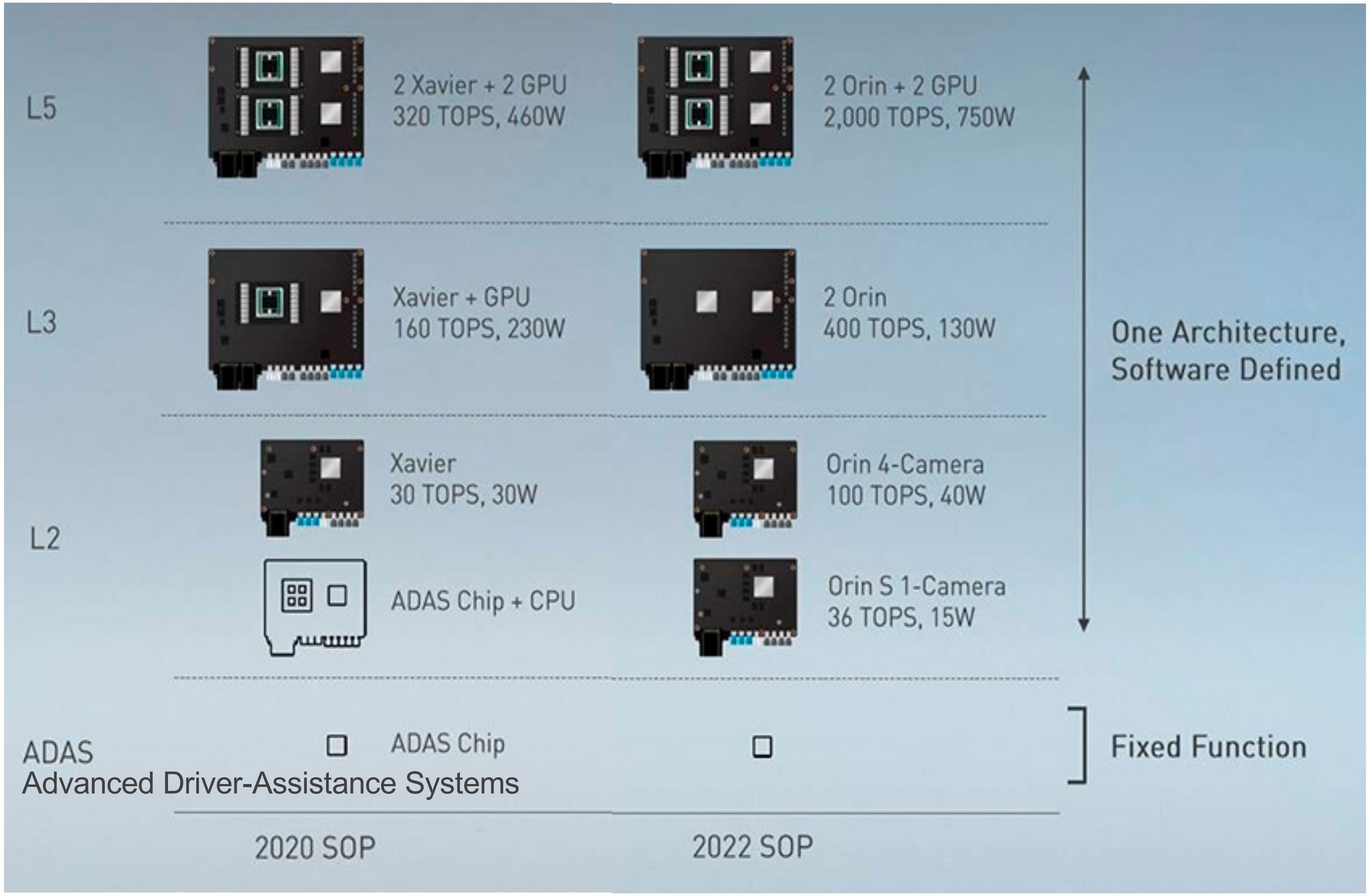
* AE, *Automotive Enhanced*: improved functional security and safety



NVidia Orin: a SoC for Autonomous Driving

NVidia Partners

CARS



L5

2 Xavier + 2 GPU
320 TOPS, 460W

2 Orin + 2 GPU
2,000 TOPS, 750W

L3

Xavier + GPU
160 TOPS, 230W

2 Orin
400 TOPS, 130W

L2

Xavier
30 TOPS, 30W

Orin 4-Camera
100 TOPS, 40W

ADAS Chip + CPU

Orin S 1-Camera
36 TOPS, 15W

One Architecture,
Software Defined

ADAS
Advanced Driver-Assistance Systems

ADAS Chip

Fixed Function

2020 SOP

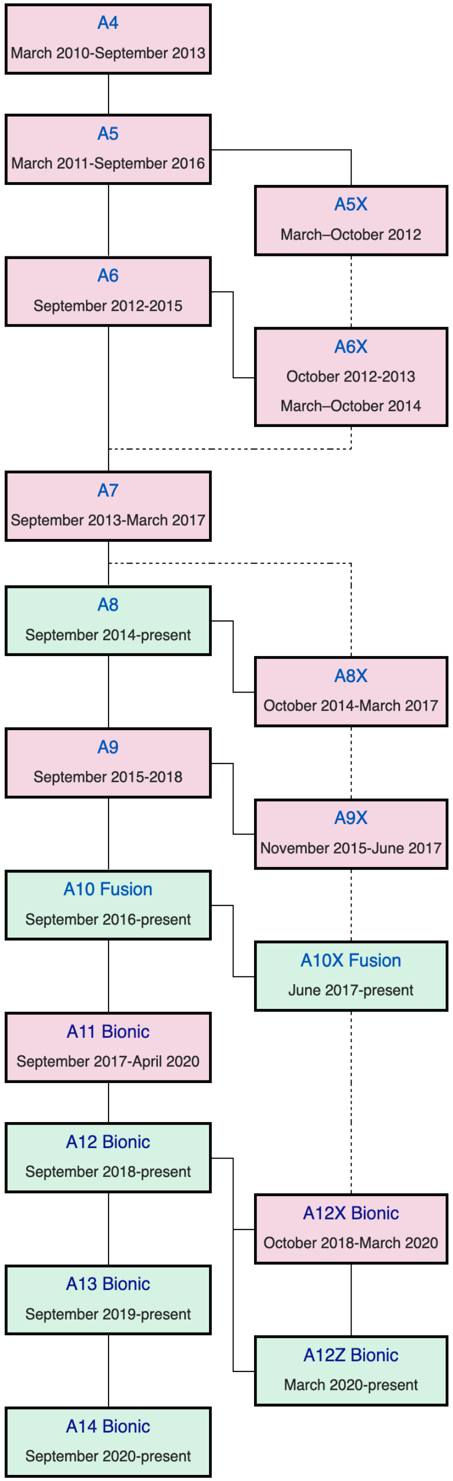
2022 SOP

Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array

- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, **…**
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

Evolution of Apple A series



A4
March 2010-September 2013

A5
March 2011-September 2016

A5X
March–October 2012

A6
September 2012-2015

A6X
October 2012-2013
March–October 2014

A7
September 2013-March 2017

A8
September 2014-present

A8X
October 2014-March 2017

A9
September 2015-2018

A9X
November 2015-June 2017

A10 Fusion
September 2016-present

A10X Fusion
June 2017-present

A11 Bionic
September 2017-April 2020

A12 Bionic
September 2018-present

A12X Bionic
October 2018-March 2020

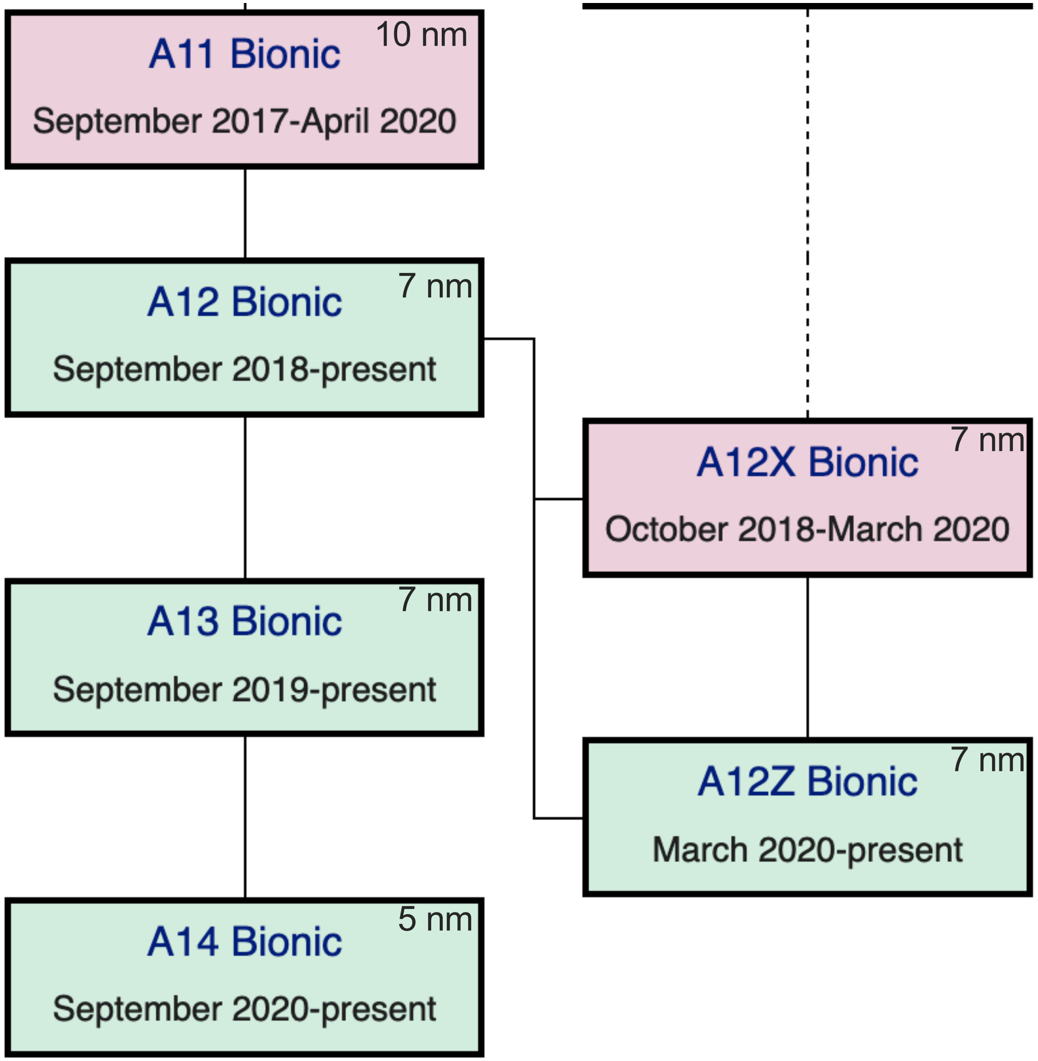
A13 Bionic
September 2019-present

A12Z Bionic
March 2020-present

A14 Bionic
September 2020-present

Bionic

Bionic => with a neural engine



A11 Bionic 10 nm
September 2017-April 2020

A12 Bionic 7 nm
September 2018-present

A12X Bionic 7 nm
October 2018-March 2020

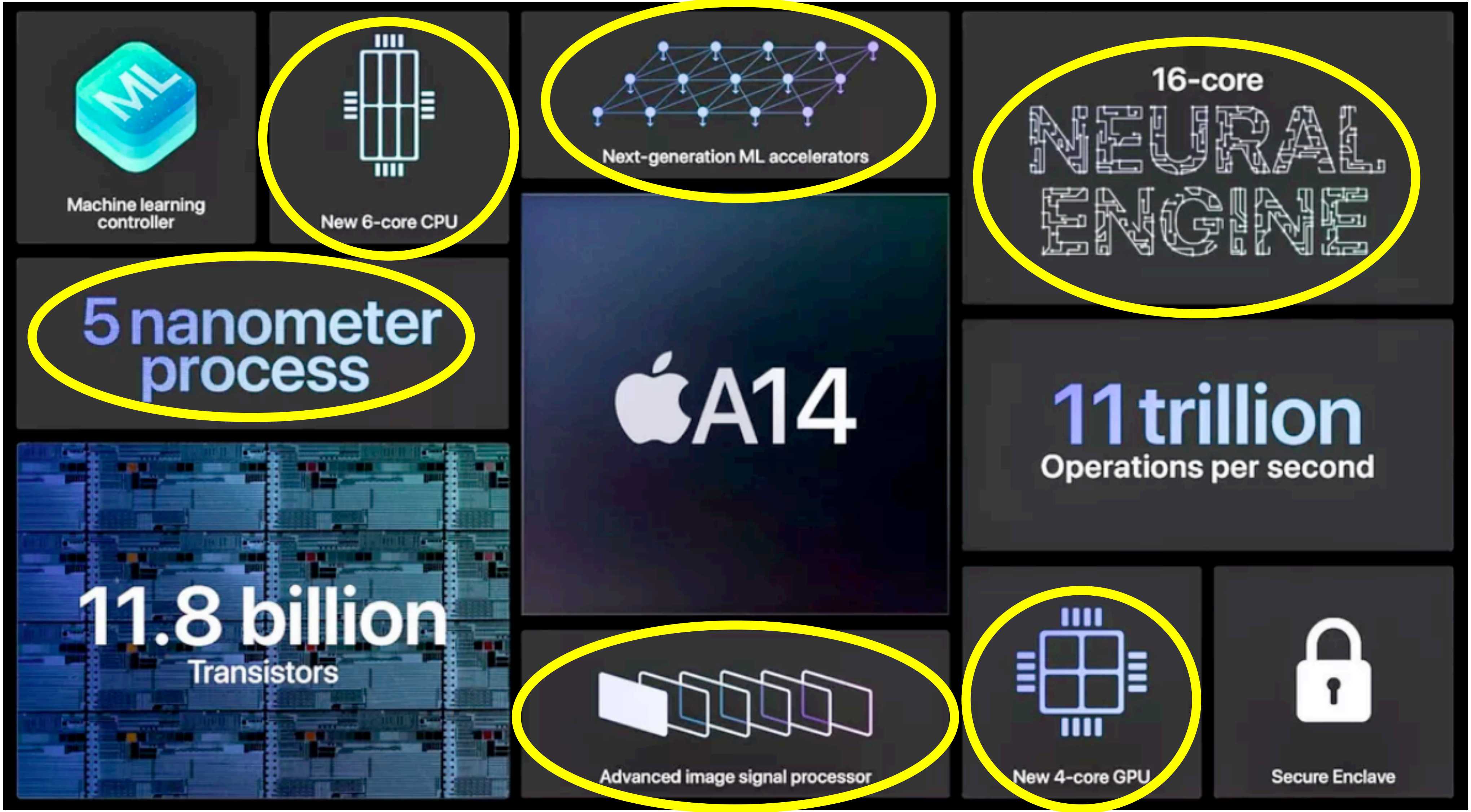
A13 Bionic 7 nm
September 2019-present

A12Z Bionic 7 nm
March 2020-present

A14 Bionic 5 nm
September 2020-present

https://en.wikipedia.org/wiki/Apple-designed_processors

Apple A14 Bionic SoC



Machine learning controller

New 6-core CPU

Next-generation ML accelerators

16-core NEURAL ENGINE

5 nanometer process

A14

11 trillion Operations per second

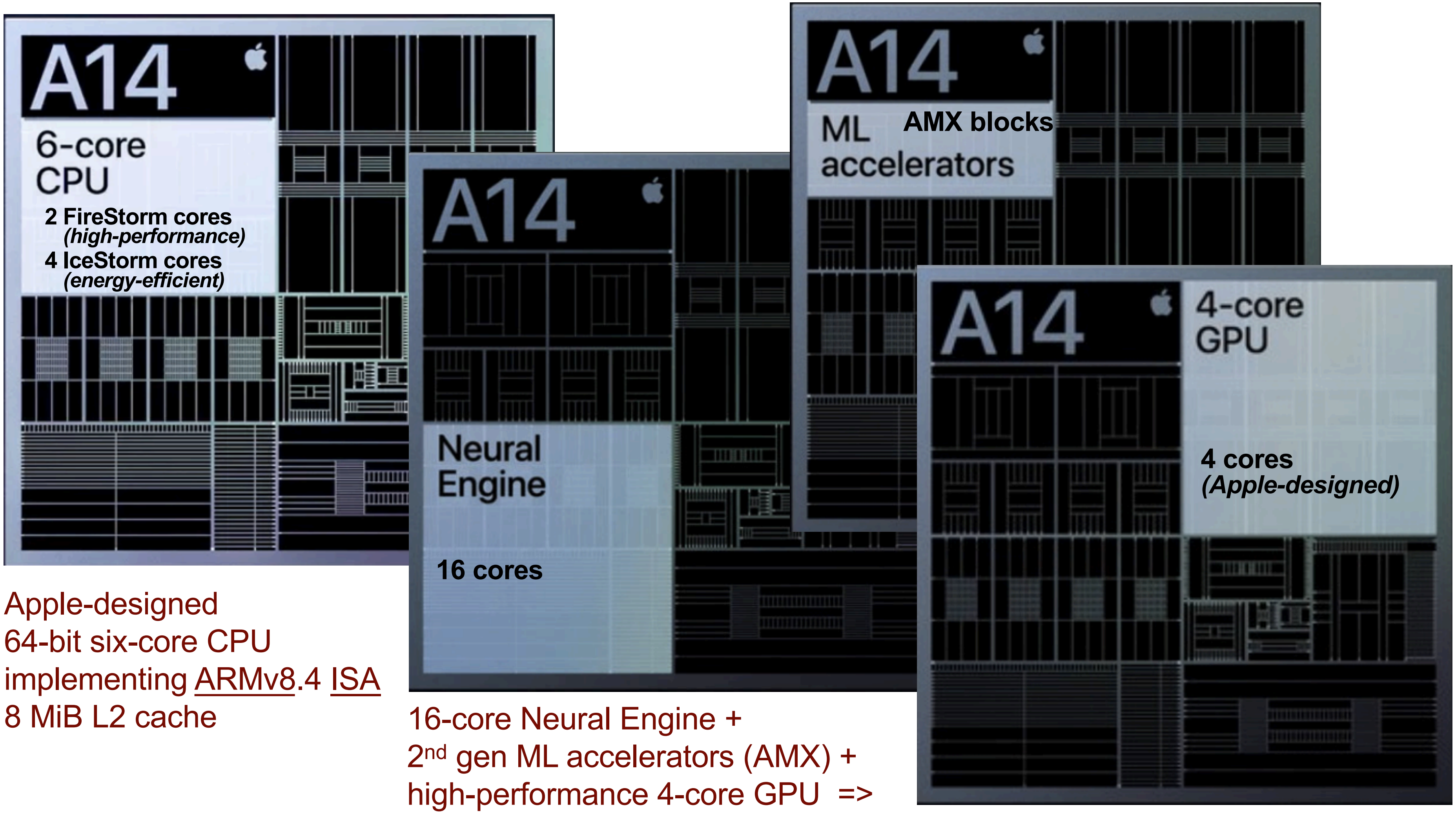
11.8 billion Transistors

Advanced image signal processor

New 4-core GPU

Secure Enclave

Apple A14 Bionic SoC



A14

6-core CPU

2 FireStorm cores
(high-performance)

4 IceStorm cores
(energy-efficient)

A14

Neural Engine

16 cores

A14

ML accelerators

AMX blocks

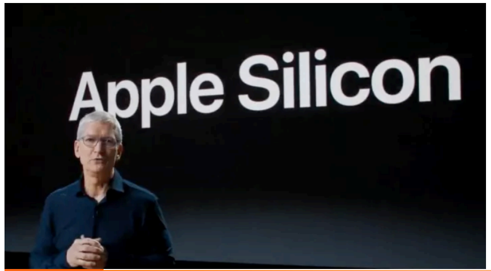
A14

4-core GPU

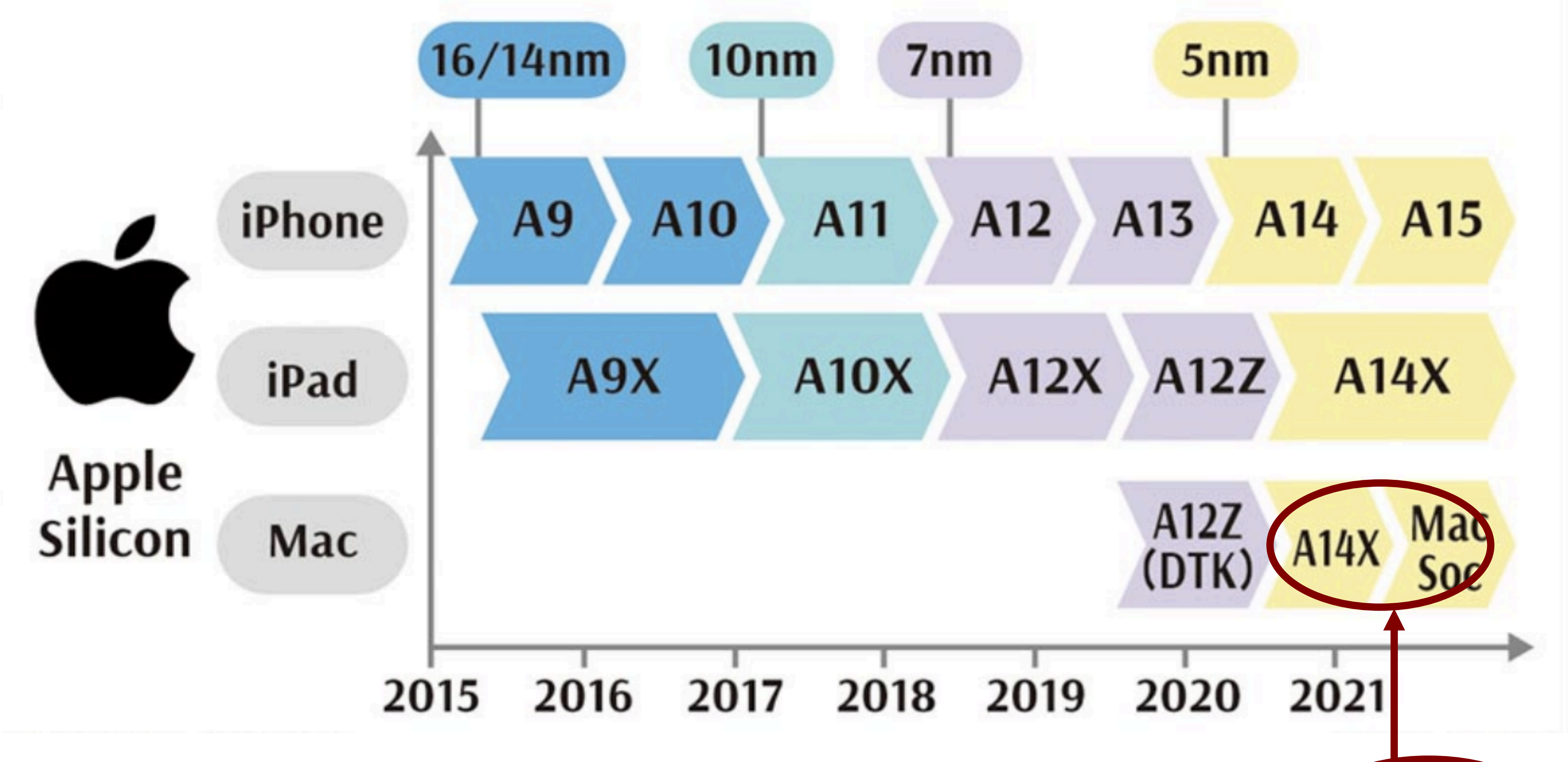
4 cores
(Apple-designed)

Apple-designed
64-bit six-core CPU
implementing ARMv8.4 ISA
8 MiB L2 cache

16-core Neural Engine +
2nd gen ML accelerators (AMX) +
high-performance 4-core GPU =>
powerful image recognition, natural language learning, motion analysis, …



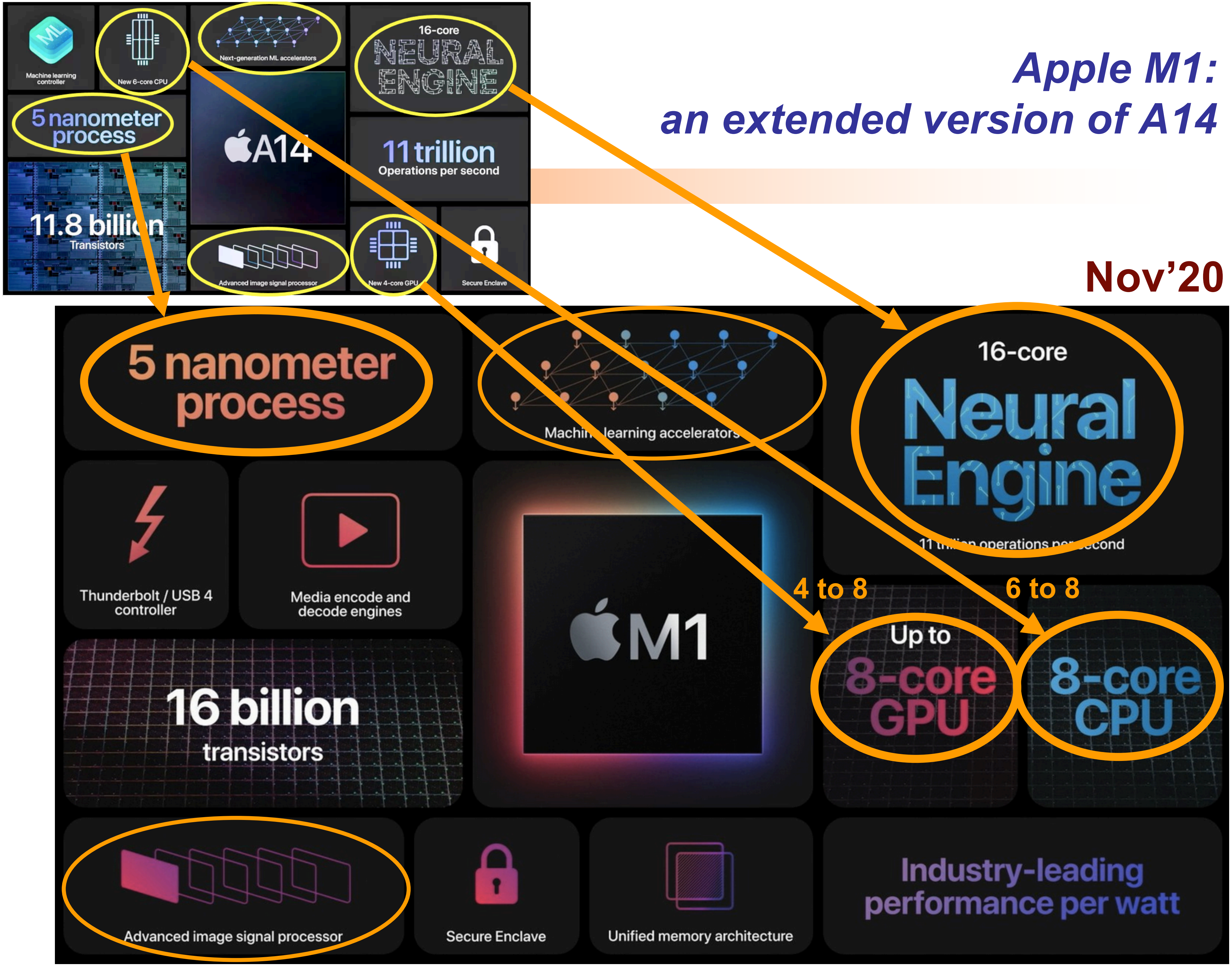
Roadmap of Apple Silicon



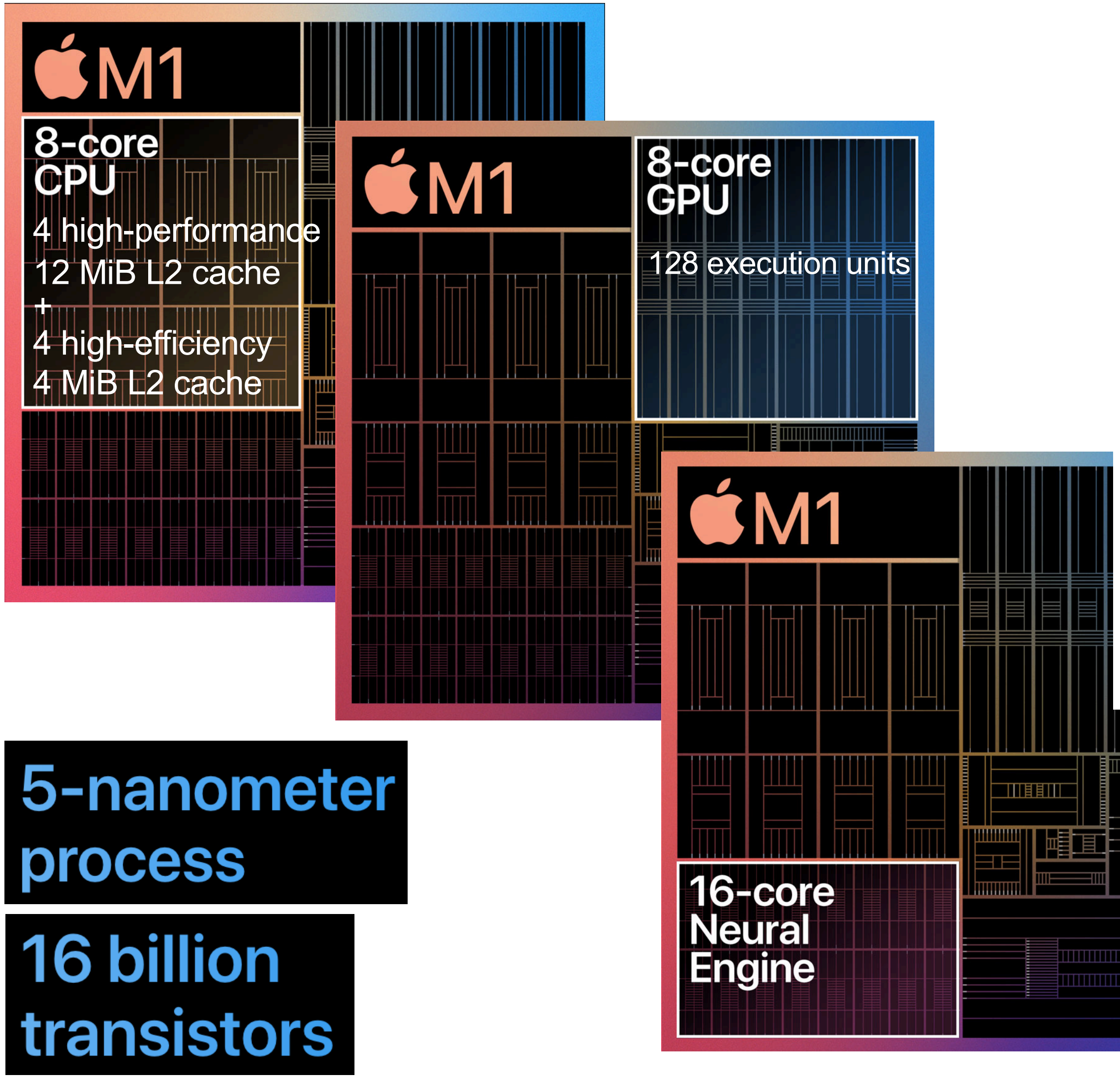
July 2020

Apple M1:
an extended version of A14

Nov’20


Machine learning controller
New 6-core CPU
Next-generation ML accelerators
16-core
NEURAL ENGINE
5 nanometer process
A14
11 trillion
Operations per second
11.8 billion
Transistors
Advanced image signal processor
New 4-core GPU
Secure Enclave
5 nanometer process
Machine learning accelerators
16-core
Neural Engine
11 trillion operations per second
Thunderbolt / USB 4 controller
Media encode and decode engines
M1
4 to 8
6 to 8
Up to
8-core GPU
8-core CPU
16 billion
transistors
Advanced image signal processor
Secure Enclave
Unified memory architecture
Industry-leading performance per watt

Apple M1 SoC



M1

8-core CPU

4 high-performance
12 MiB L2 cache
+
4 high-efficiency
4 MiB L2 cache

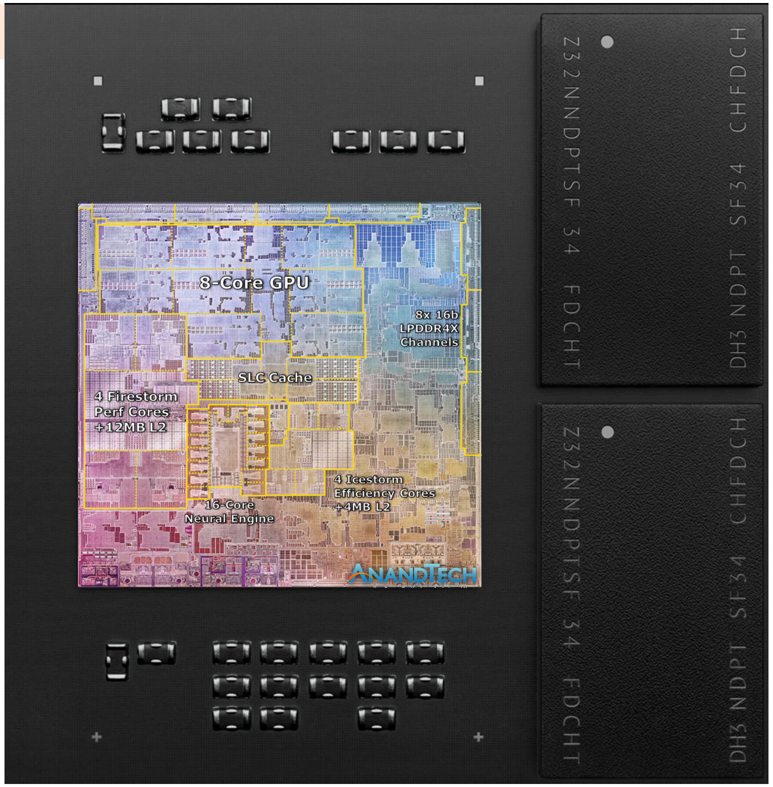
M1

8-core GPU

128 execution units

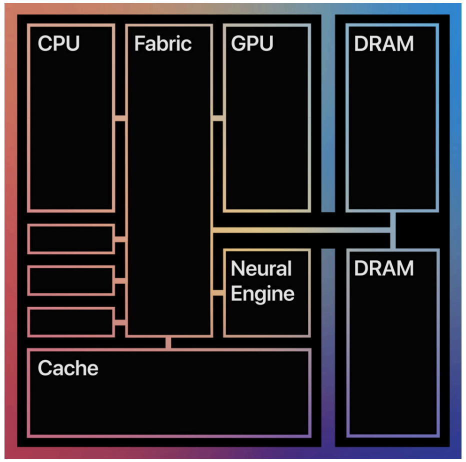
M1

16-core Neural Engine

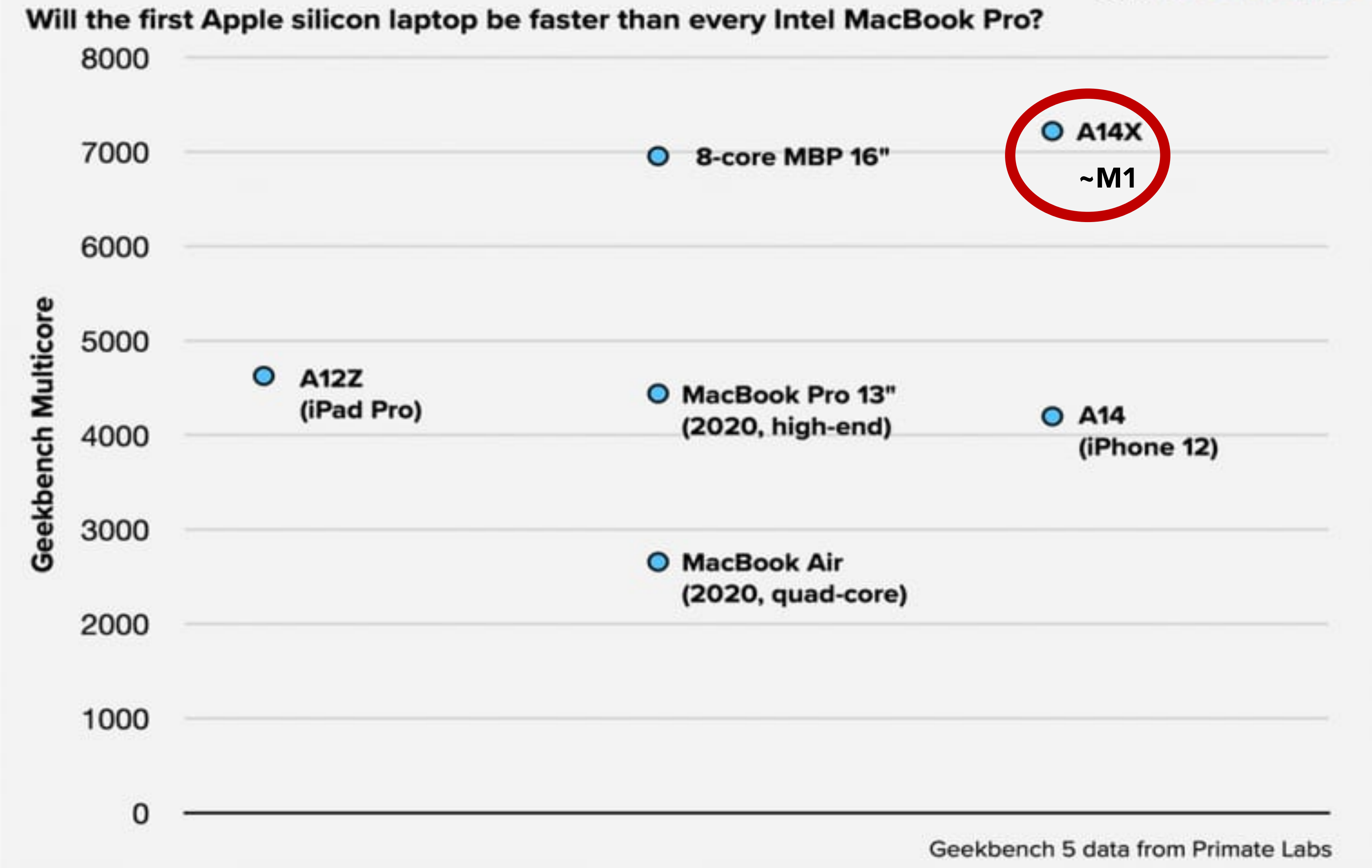


5-nanometer process

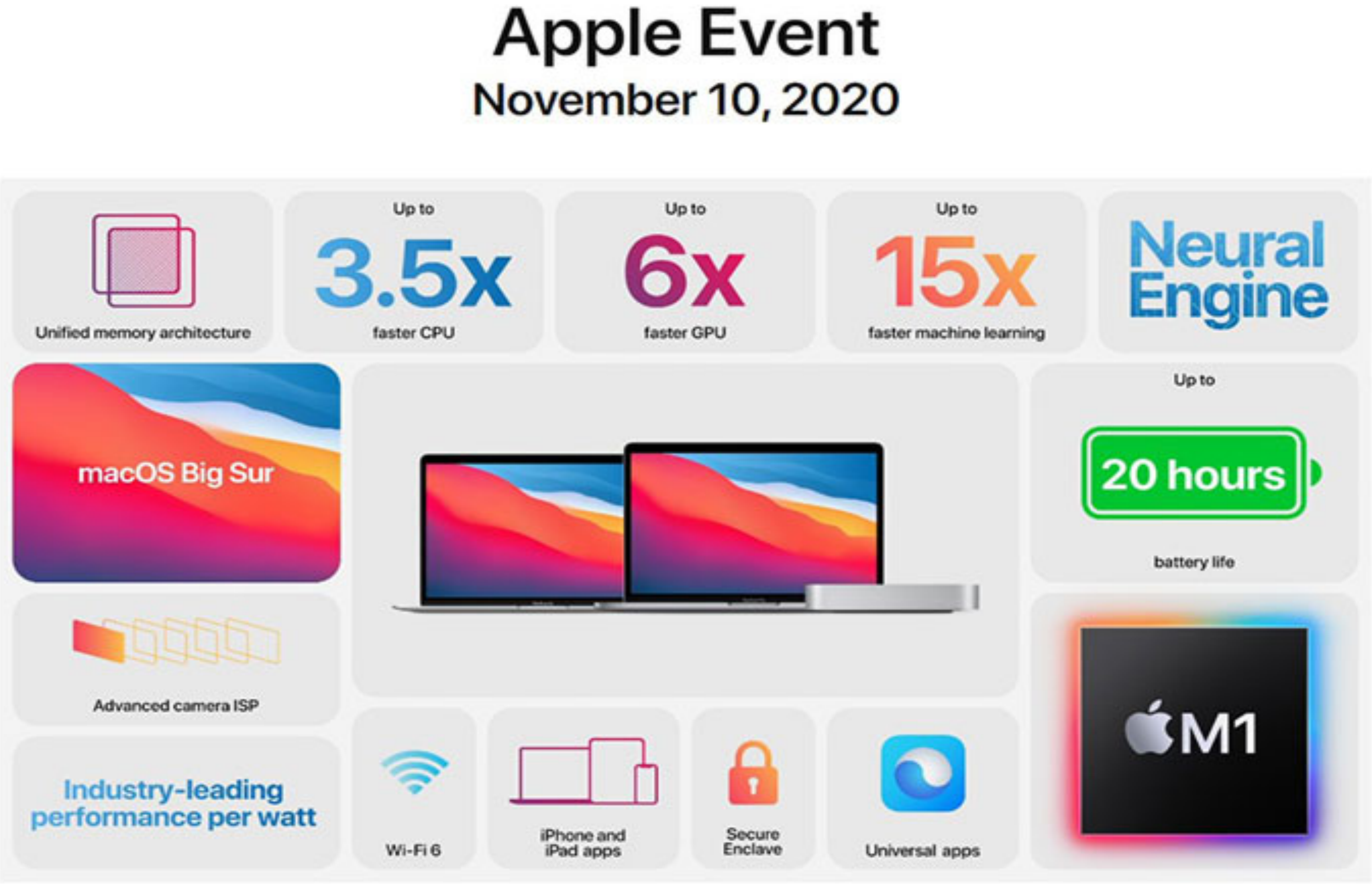
16 billion transistors



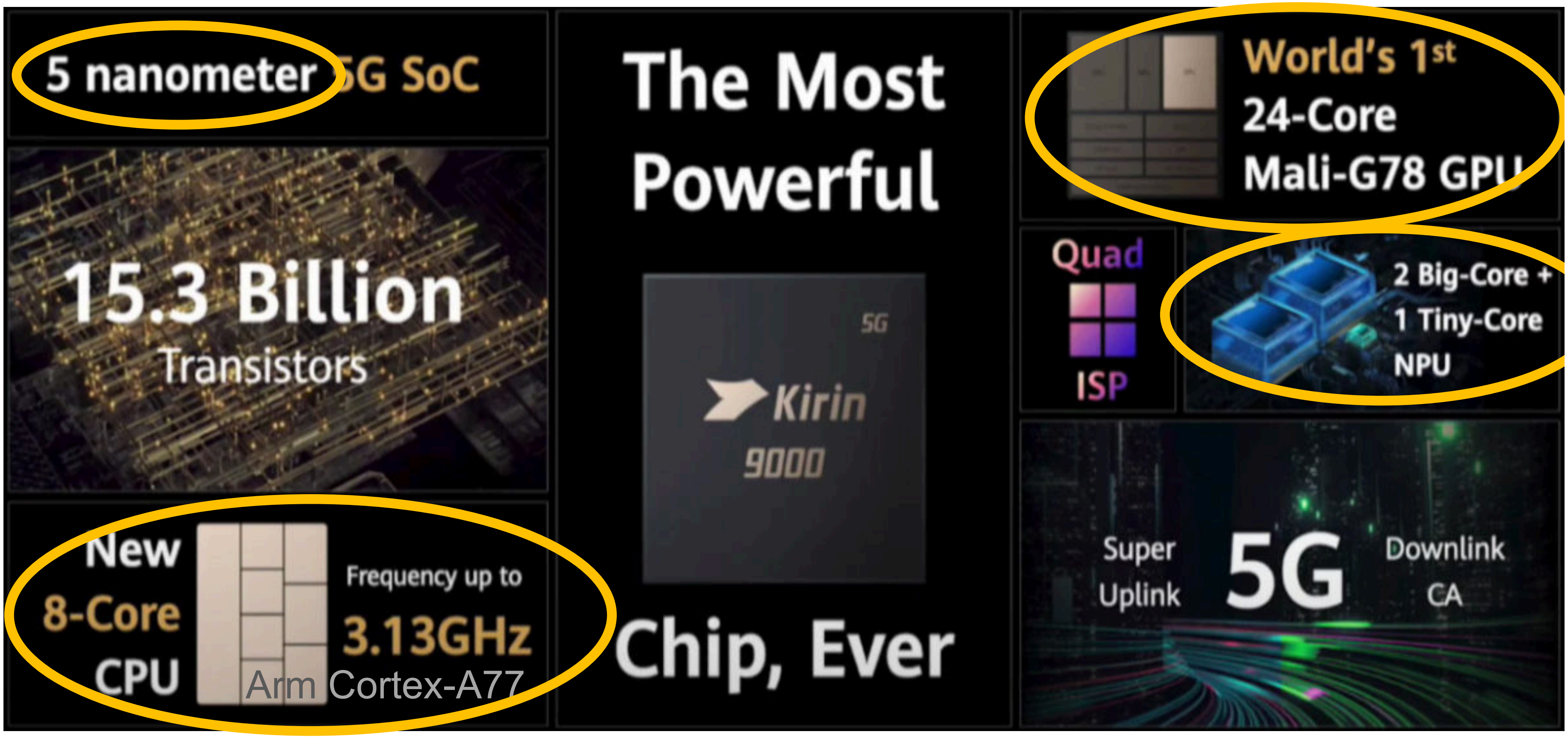
Apple M1 Performance



The new MacBooks *(Nov’20)*

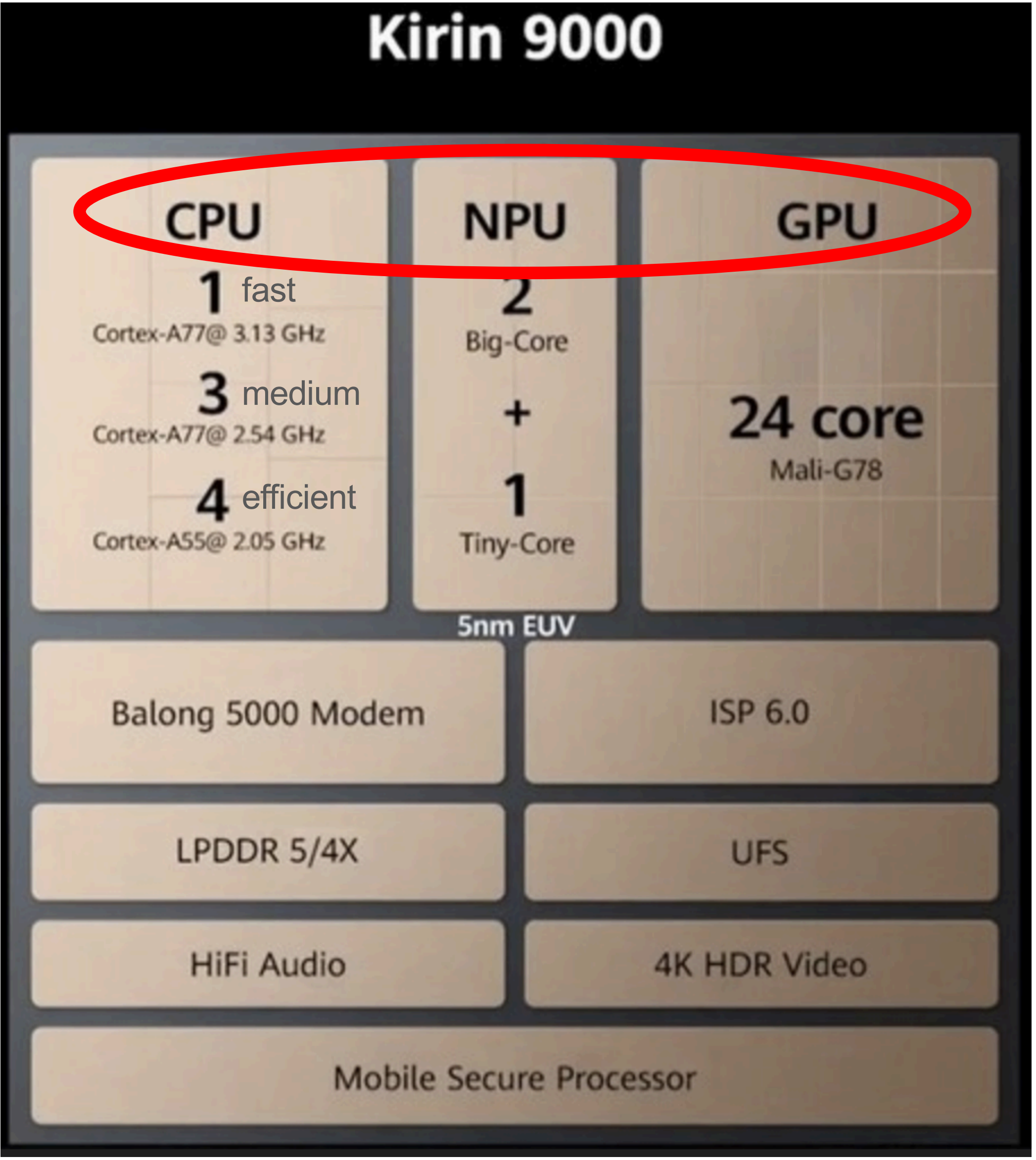


Huawey Kirin 9000



Huawey Kirin 9000

A77: old ARM generation…



Kirin 9000

CPU

1 fast
Cortex-A77@ 3.13 GHz

3 medium
Cortex-A77@ 2.54 GHz

4 efficient
Cortex-A55@ 2.05 GHz

NPU

2
Big-Core
+
1
Tiny-Core

GPU

24 core
Mali-G78

5nm EUV

Balong 5000 Modem

ISP 6.0

LPDDR 5/4X

UFS

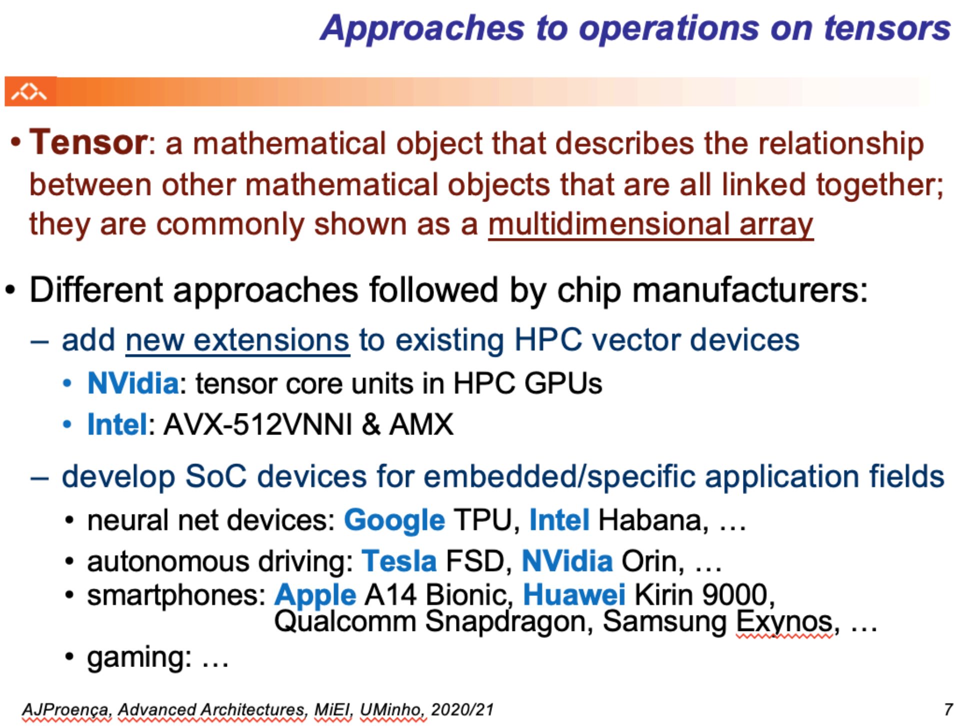
HiFi Audio

4K HDR Video

Mobile Secure Processor

Approaches to operations on tensors

And after this…



Approaches to operations on tensors

- **Tensor**: a mathematical object that describes the relationship between other mathematical objects that are all linked together; they are commonly shown as a multidimensional array
- Different approaches followed by chip manufacturers:
  - add new extensions to existing HPC vector devices
    - **NVidia**: tensor core units in HPC GPUs
    - **Intel**: AVX-512VNNI & AMX
  - develop SoC devices for embedded/specific application fields
    - neural net devices: **Google** TPU, **Intel** Habana, …
    - autonomous driving: **Tesla** FSD, **NVidia** Orin, …
    - smartphones: **Apple** A14 Bionic, **Huawei** Kirin 9000, Qualcomm Snapdragon, Samsung Exynos, …
    - gaming: …

AJProença, Advanced Architectures, MiEI, UMinho, 2020/21 7

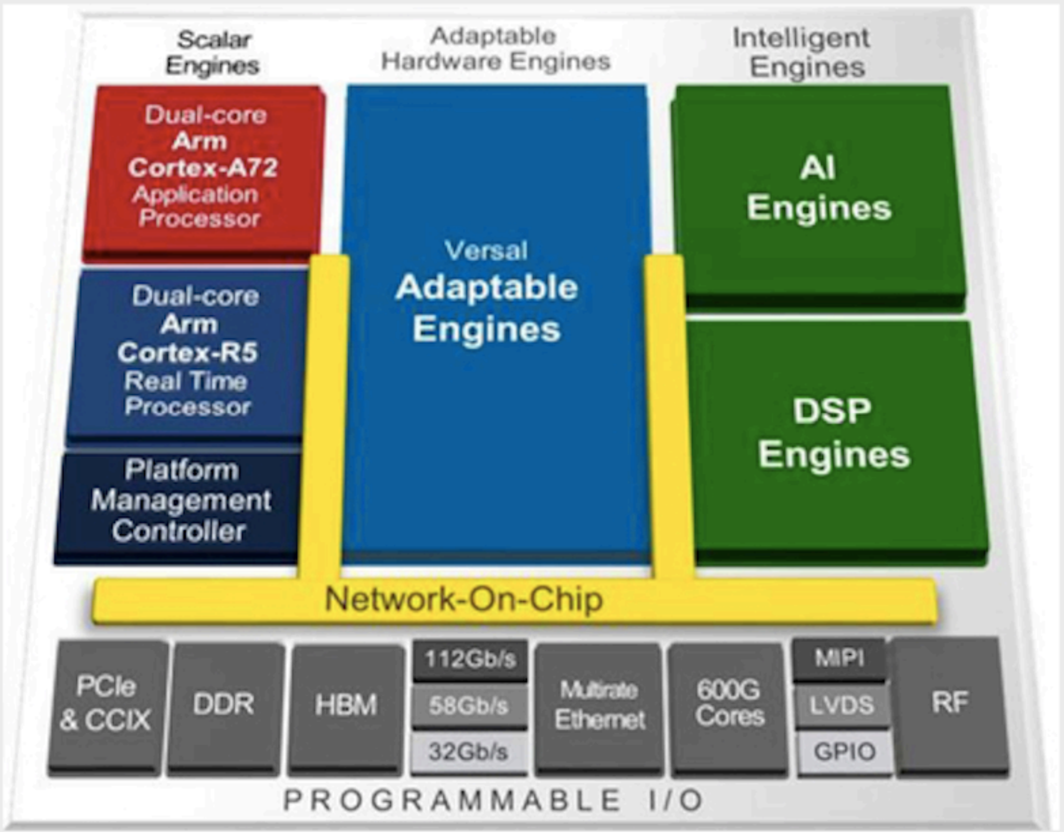
Yet some more odd approaches:

- SoC with reconfigurable components: **Xilinx** ACAP
- a system based on a very large "chip": **Cerebras**

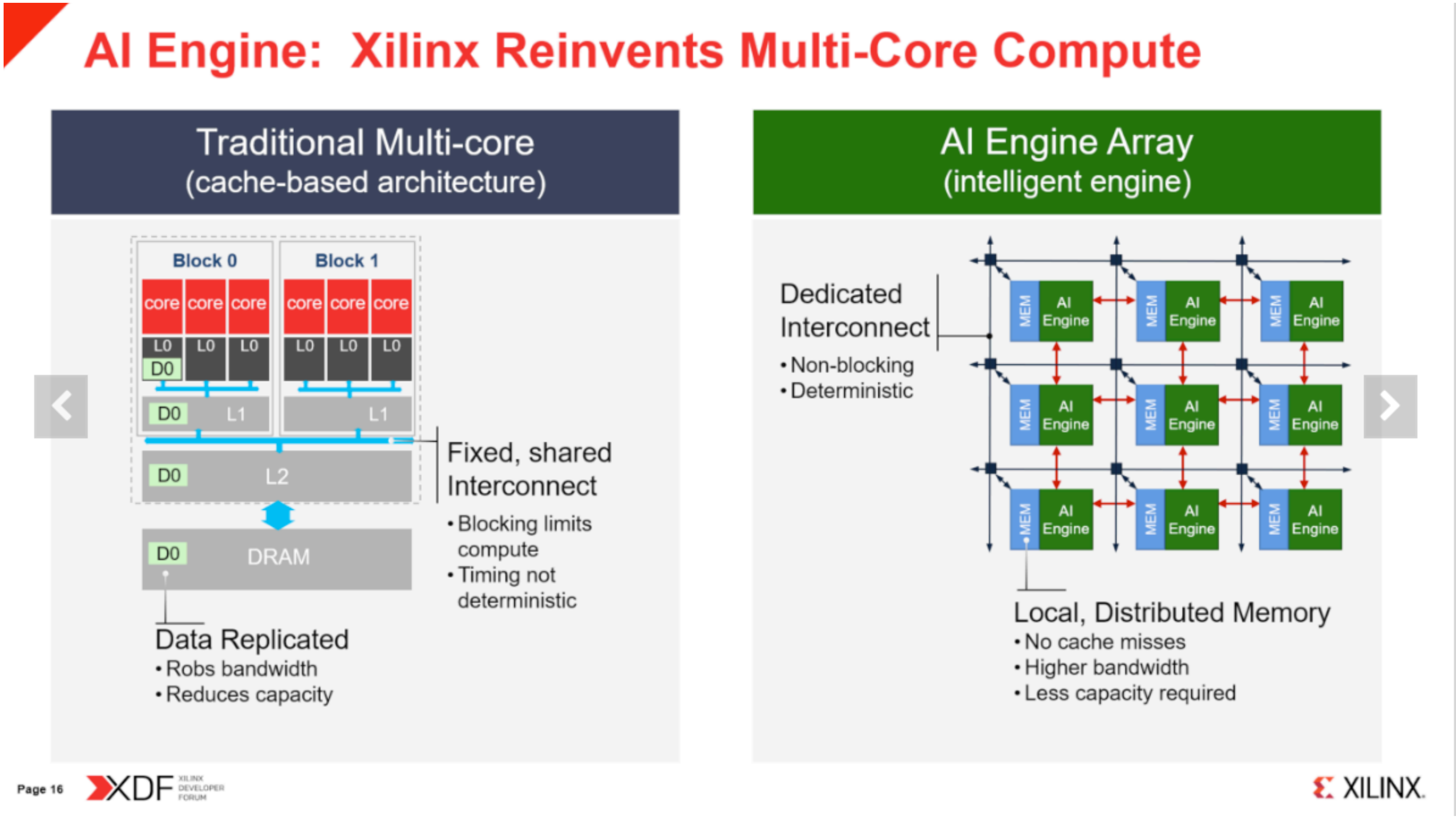
Xilinx Versal:
an Adaptive Compute Acceleration Platform (ACAP)

ACAP die, an adaptable accelerator-fabric ecosystem with:

- multicore ARM SoC's
- an FPGA fabric with distributed memory
- hw-programmable DSP engines
- AI Engines with vector units
- other specialized accelerators
- a flexible NoC interconnection



The AI Engine Array in Xilinx Versal (1)



AI Engine: Xilinx Reinvents Multi-Core Compute

Traditional Multi-core
(cache-based architecture)

Fixed, shared Interconnect
- Blocking limits compute
- Timing not deterministic

Data Replicated
- Robs bandwidth
- Reduces capacity

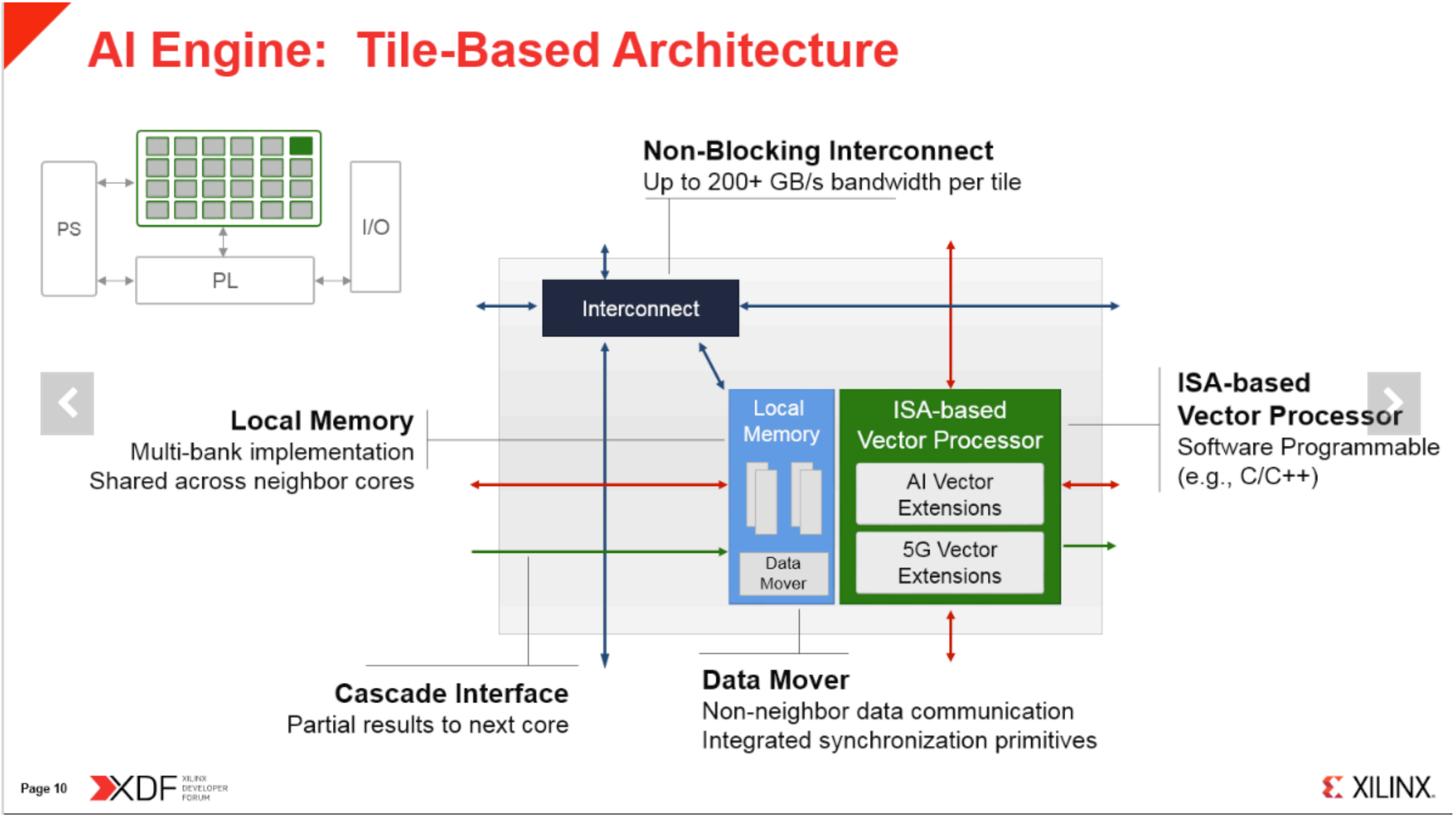
AI Engine Array
(intelligent engine)

Dedicated Interconnect
- Non-blocking
- Deterministic

Local, Distributed Memory
- No cache misses
- Higher bandwidth
- Less capacity required

https://forums.xilinx.com/t5/Adaptable-Advantage-Blog/Xilinx-Unveiled-the-Secret-Sauce-of-the-Ultimate-AI-Inference/ba-p/898197

The AI Engine Array in Xilinx Versal (2)



AI Engine: Tile-Based Architecture

Non-Blocking Interconnect
Up to 200+ GB/s bandwidth per tile

Local Memory
Multi-bank implementation
Shared across neighbor cores

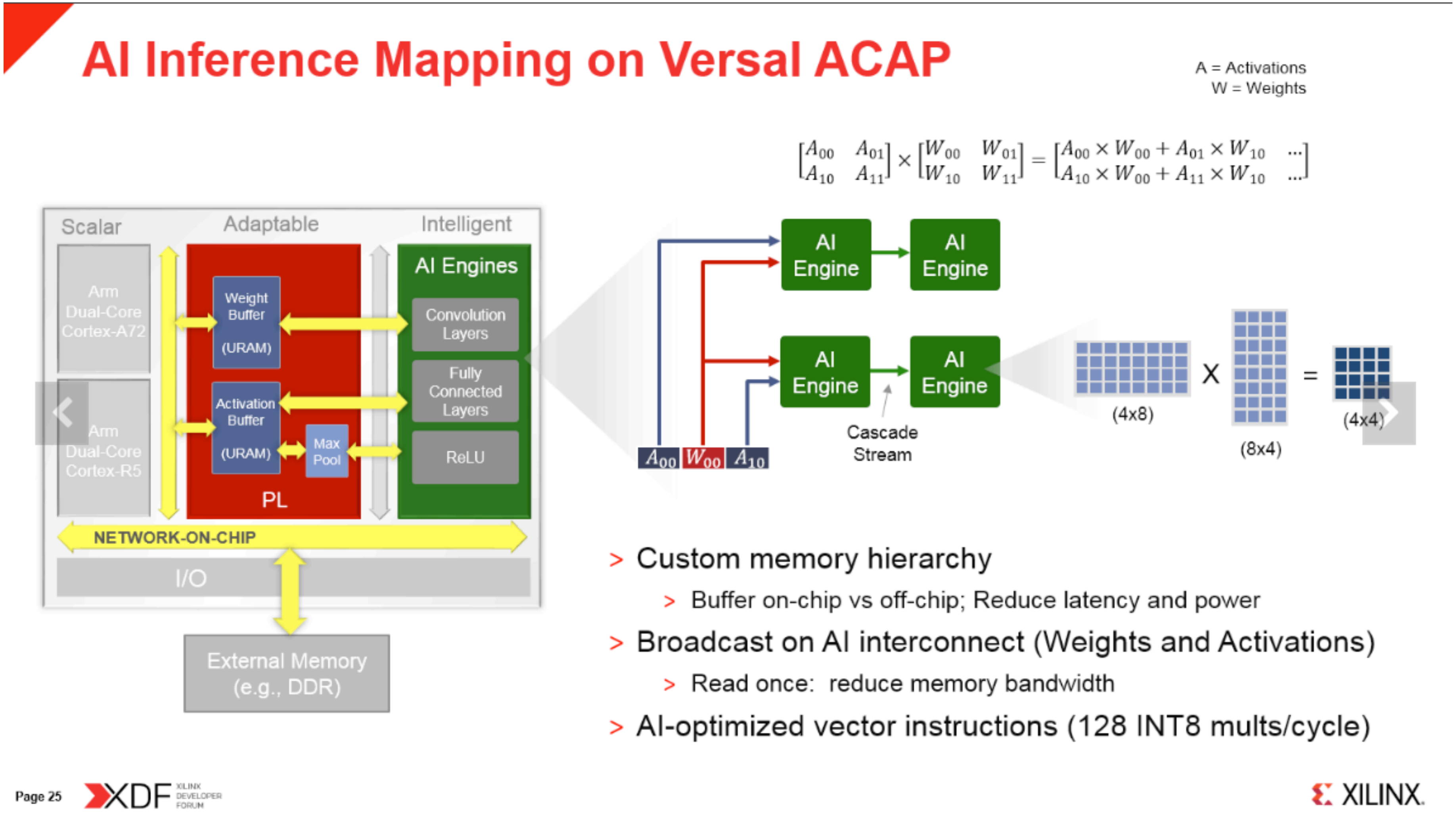
ISA-based Vector Processor
Software Programmable
(e.g., C/C++)

Cascade Interface
Partial results to next core

Data Mover
Non-neighbor data communication
Integrated synchronization primitives

https://forums.xilinx.com/t5/Adaptable-Advantage-Blog/Xilinx-Unveiled-the-Secret-Sauce-of-the-Ultimate-AI-Inference/ba-p/898197

The AI Engine Array in Xilinx Versal (3)



AI Inference Mapping on Versal ACAP

A = Activations
W = Weights

$$\begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \times \begin{bmatrix} W_{00} & W_{01} \\ W_{10} & W_{11} \end{bmatrix} = \begin{bmatrix} A_{00} \times W_{00} + A_{01} \times W_{10} & \dots \\ A_{10} \times W_{00} + A_{11} \times W_{10} & \dots \end{bmatrix}$$

- Custom memory hierarchy
  - Buffer on-chip vs off-chip; Reduce latency and power
- Broadcast on AI interconnect (Weights and Activations)
  - Read once: reduce memory bandwidth
- AI-optimized vector instructions (128 INT8 mults/cycle)

Page 25 XDF XILINX DEVELOPER FORUM

XILINX.

https://forums.xilinx.com/t5/Adaptable-Advantage-Blog/Xilinx-Unveiled-the-Secret-Sauce-of-the-Ultimate-AI-Inference/ba-p/898197

Cerebras: a Wafer Scale Engine (WSE) *(Aug'19)*

Cerebras Wafer Scale Engine (WSE):
the largest chip ever built

46,225 mm^2 chip

56x larger than the biggest GPU ever made

400,000 core

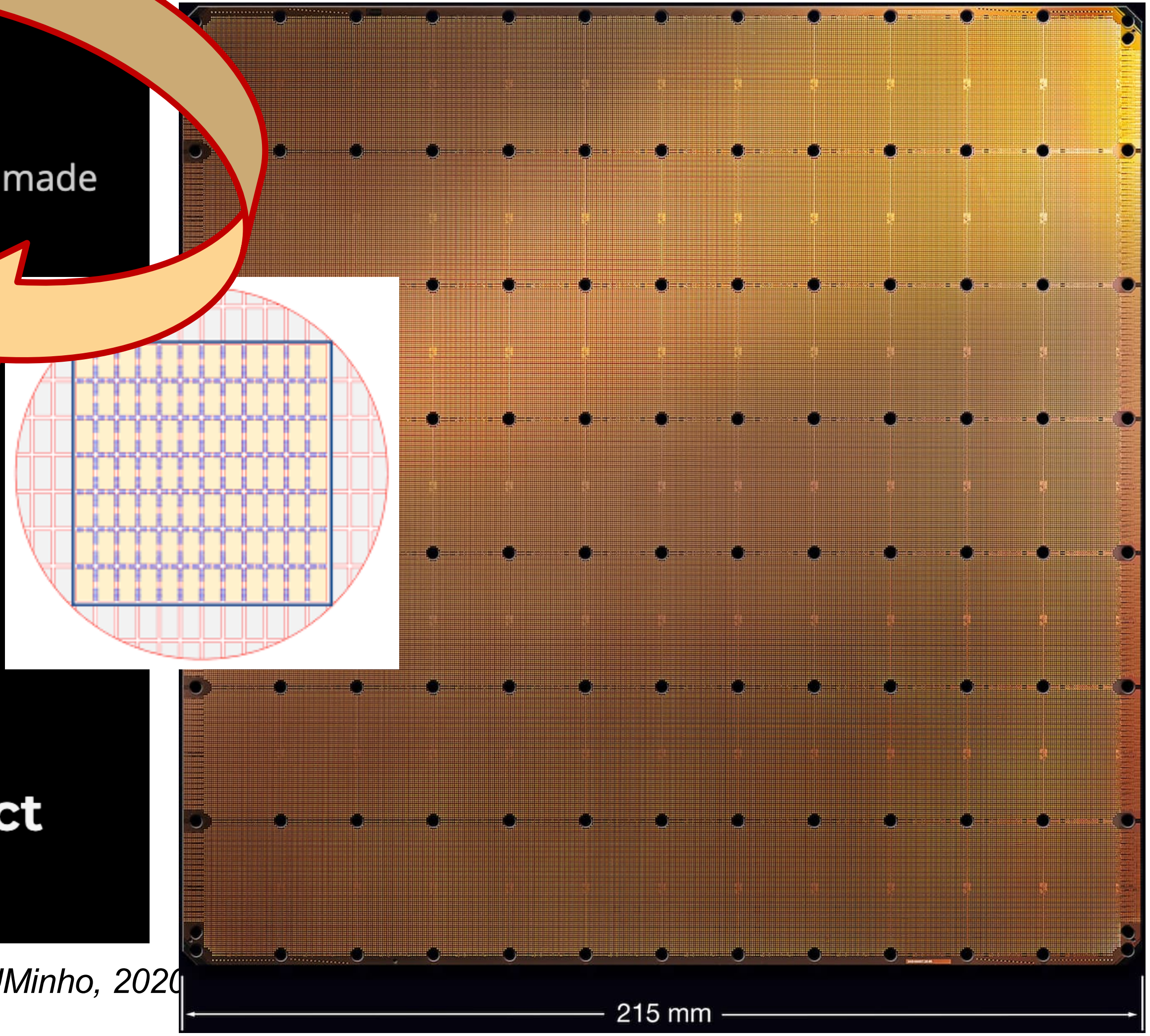
78x more cores

18 GB on-chip SRAM

3000x more on-chip memory

100 Pb/s interconnect

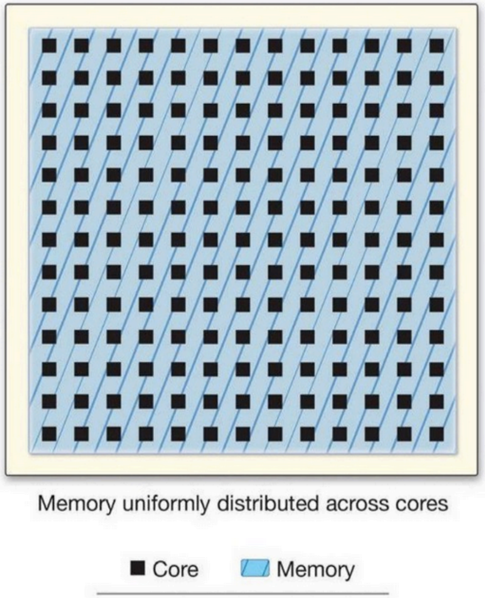
33,000x more bandwidth



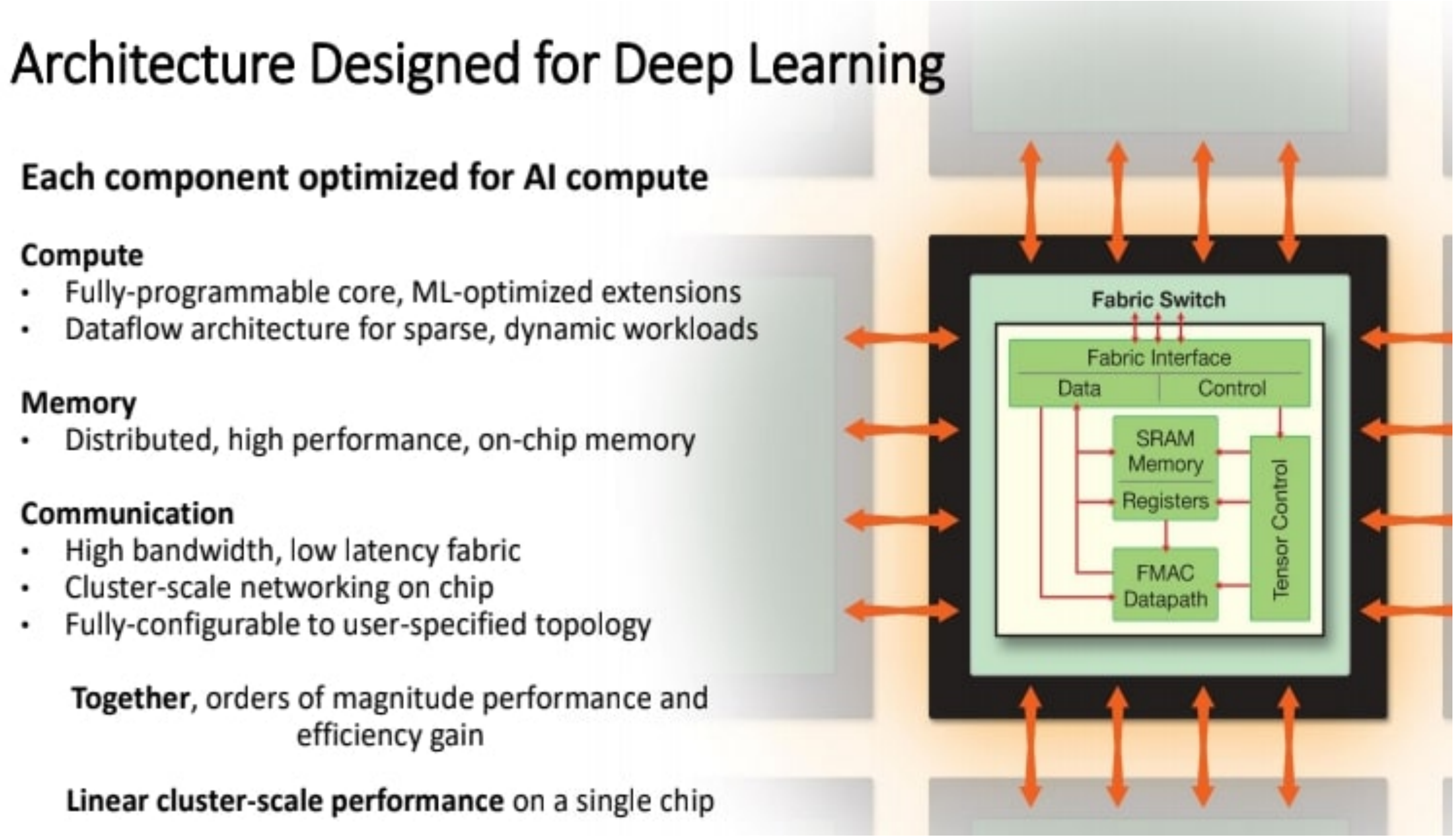
Cerebras: additional data

Some additional data:

- 16 nm process
- 215 mm x 215 mm, ~15 kW consumption !
- 84 individual chips (12 wide by 7 tall)
- each chip:
  - 225 MiB SRAM
  - 54 x 94 = 5,076 Sparse Linear Algebra cores (SLA)
    (2 cores per row/column unused due to repair scheme leaving 4,888 usable cores)
- each core:
  - 47 kiB SRAM
  - Zeros not loaded from memory and zeros not multiplied
  - FP32 precision and scalar execution (can't filter zeros from memory with SIMD)
  - FMAC datapath with peak 8 operations per cycle)
  - Tensor control unit feeds the FMAC datapath with strided accesses
    (from memory or inbound data from links)
  - 4x 8 GB/s bidirectional links to its neighbours


Memory uniformly distributed across cores

Cerebras: flexible cores optimized for tensor op's



Architecture Designed for Deep Learning

Each component optimized for AI compute

Compute

- Fully-programmable core, ML-optimized extensions
- Dataflow architecture for sparse, dynamic workloads

Memory

- Distributed, high performance, on-chip memory

Communication

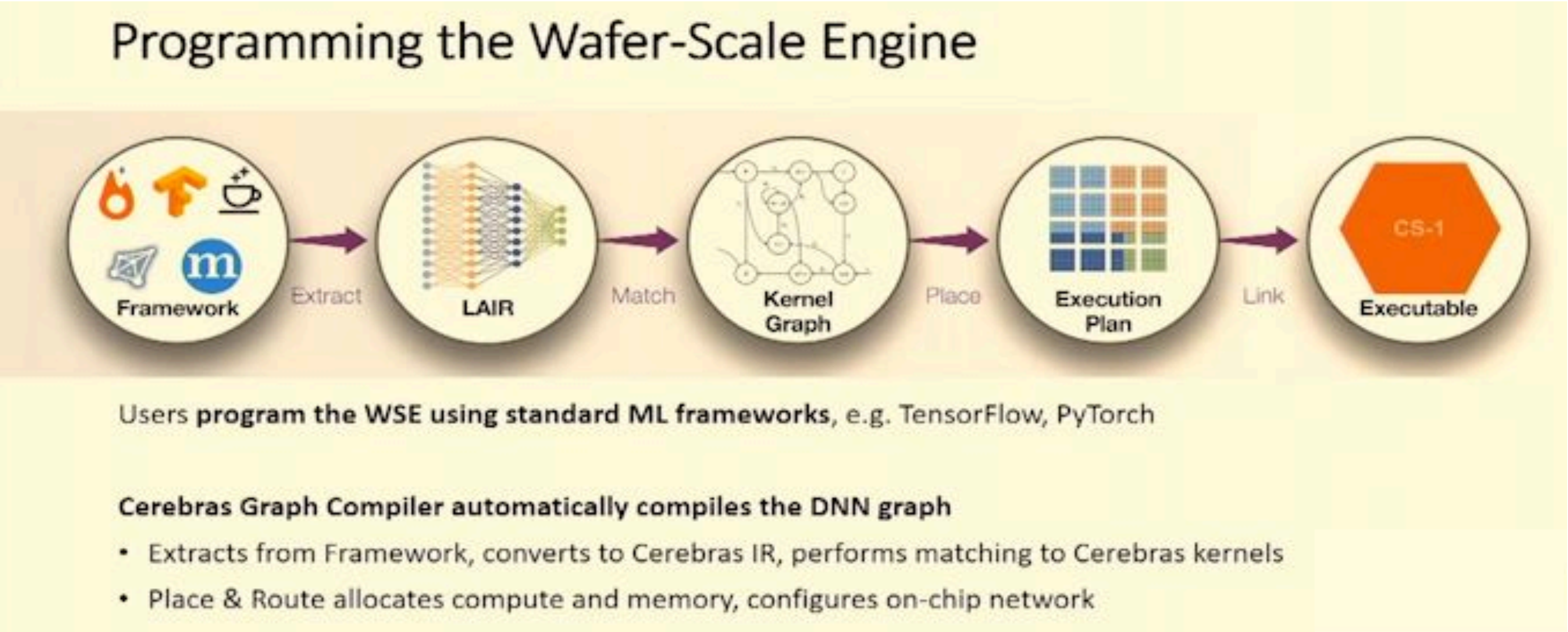
- High bandwidth, low latency fabric
- Cluster-scale networking on chip
- Fully-configurable to user-specified topology

Together, orders of magnitude performance and efficiency gain

Linear cluster-scale performance on a single chip

https://www.anandtech.com/show/16006/hot-chips-2020-live-blog-cerebras-wse-programming-300pm-pt

Cerebras: programming WSE



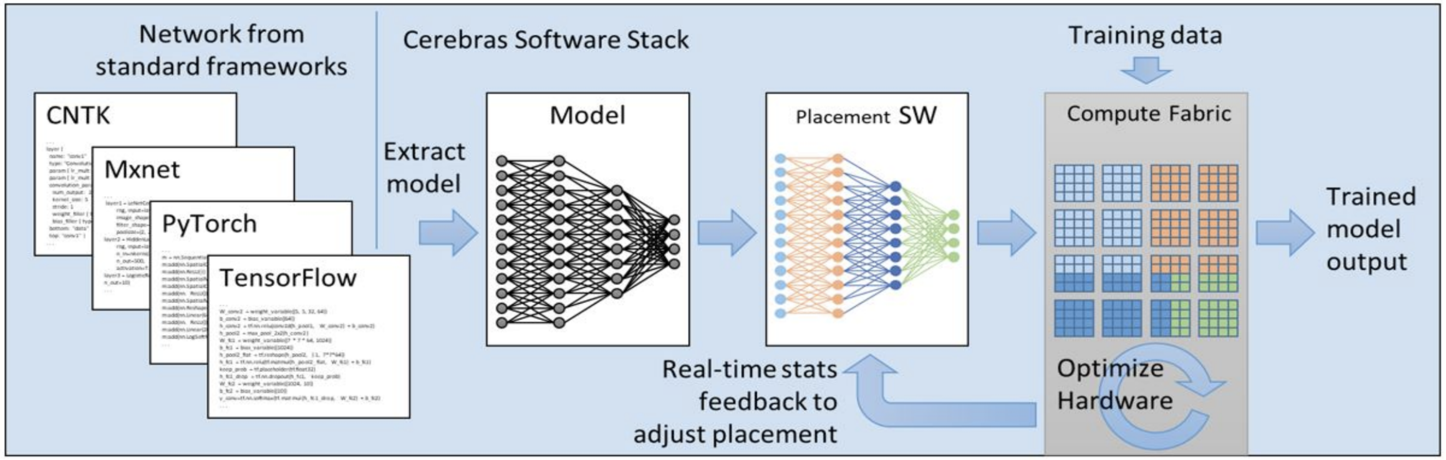
Programming the Wafer-Scale Engine

Framework
Extract
LAIR
Match
Kernel Graph
Place
Execution Plan
Link
CS-1
Executable

Users **program the WSE using standard ML frameworks**, e.g. TensorFlow, PyTorch

Cerebras Graph Compiler automatically compiles the DNN graph

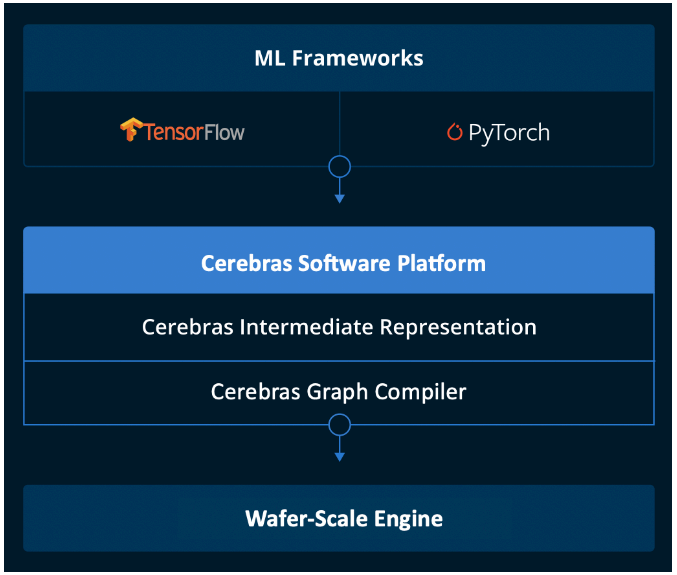
- Extracts from Framework, converts to Cerebras IR, performs matching to Cerebras kernels
- Place & Route allocates compute and memory, configures on-chip network



https://www.anandtech.com/show/16006/hot-chips-2020-live-blog-cerebras-wse-programming-300pm-pt

Packing the Cerebras WSE: clusters of CS-1s (Argonne National Lab)

SOFTWARE PLATFORM



ML Frameworks

TensorFlow

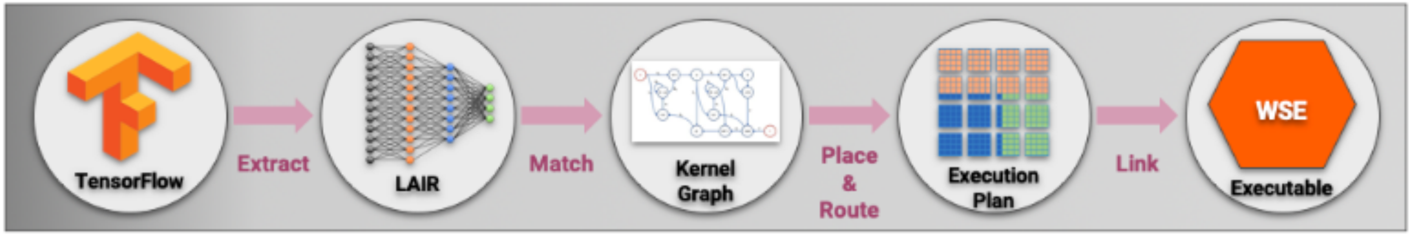
PyTorch

Cerebras Software Platform

Cerebras Intermediate Representation

Cerebras Graph Compiler

Wafer-Scale Engine

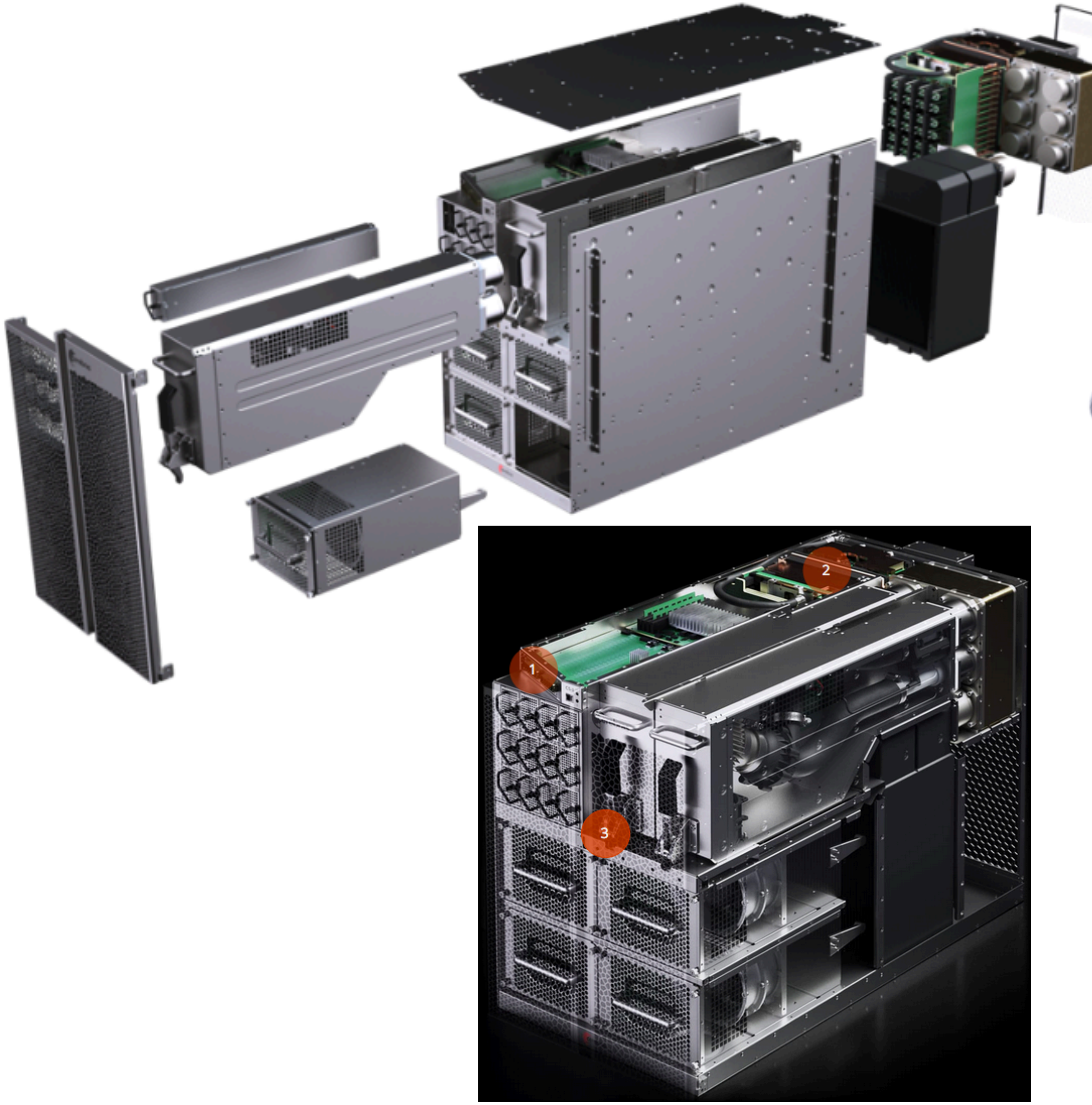


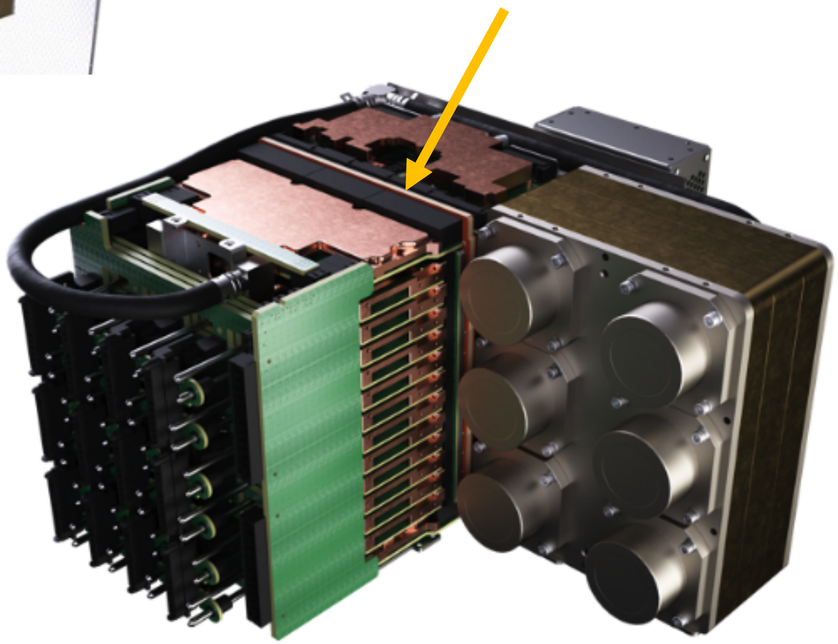
TensorFlow → Extract → LAIR → Match → Kernel Graph → Place & Route → Execution Plan → Link → WSE Executable

Figure 8: A high-level overview of the compilation process for the WSE

https://secureservercdn.net/198.12.145.239/a7b.fcb.myftpupload.com/wp-content/uploads/2019/11/The-Cerebras-CS-1-Product-Overview.pdf

The CS-1 internal structure





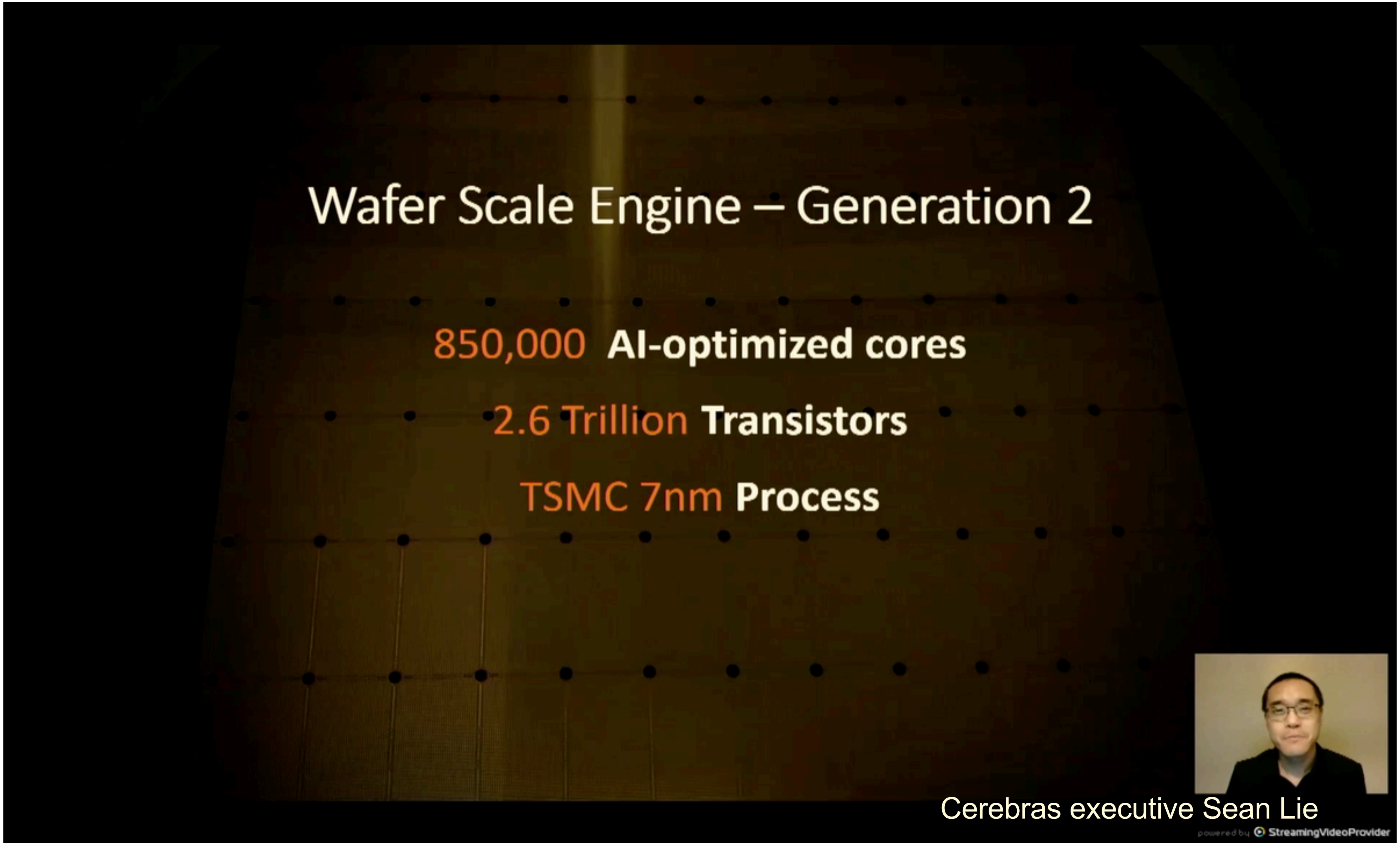
2. Engine Block

Power, cooling and packaging solution for the Wafer-Scale Engine

https://secureservercdn.net/198.12.145.239/a7b.fcb.myftpupload.com/wp-content/uploads/2019/11/The-Cerebras-CS-1-Product-Overview.pdf

Cerebras:

Generation 2 *(in-house tests in Aug'20)*



Wafer Scale Engine – Generation 2

850,000 **AI-optimized cores**

2.6 Trillion **Transistors**

TSMC 7nm **Process**

Cerebras executive Sean Lie

https://www.anandtech.com/show/16006/hot-chips-2020-live-blog-cerebras-wse-programming-300pm-pt