AI 芯片 – AI 芯片基础

计算体系架构黄金10年

ZOMI

Talk Overview

1. AI 计算体系

- 深度学习计算模式
- 计算体系与矩阵运算

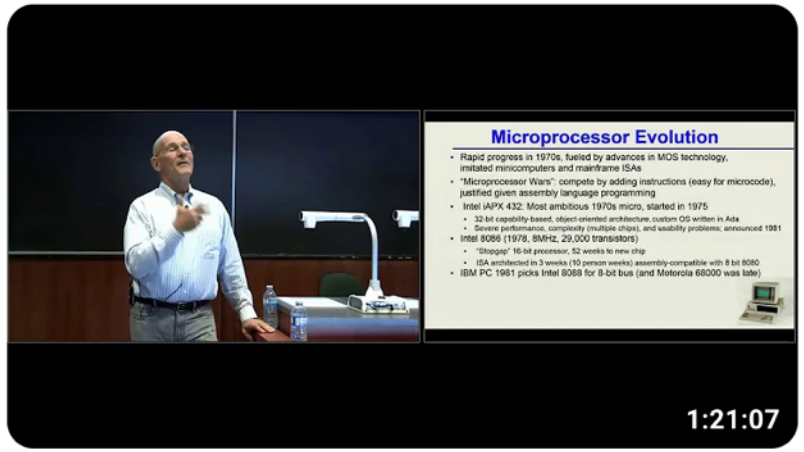
2. AI 芯片基础

- 通用处理器 CPU
- 从数据看 CPU 计算
- 通用图形处理器 GPU
- AI专用处理器 NPU/TPU
- 计算体系架构的黄金10年

计算机架构的新黄金时代

- A New Golden Age for Computer Architecture: History, Challenges and Opportunities

  https://www.youtube.com/watch?v=kFT54hO1X8M



David Patterson - A New Golden Age for Computer Architecture: History, Challenges and Opportunities

7.1万次观看 · 3年前

UBC Computer Science

Abstract: In the 1980s, Mead and Conway democratized chip design and high-level language programming surpassed assembly ...

Turing Awards | What is Computer Architecture | IBM System360 | Semiconductors | Microprocessor... 44 个章节

编译器的黄金时代

- The Golden Age of Compiler Design in an Era of HW/SW Co-design
- https://www.youtube.com/watch?v=4HgShra-KnY

ASPLOS Keynote: The Golden Age of Compiler Design in an Era of HW/SW Co-design by Dr. Chris Lattner

2.7万次观看 · 1年前

SiFiveInc

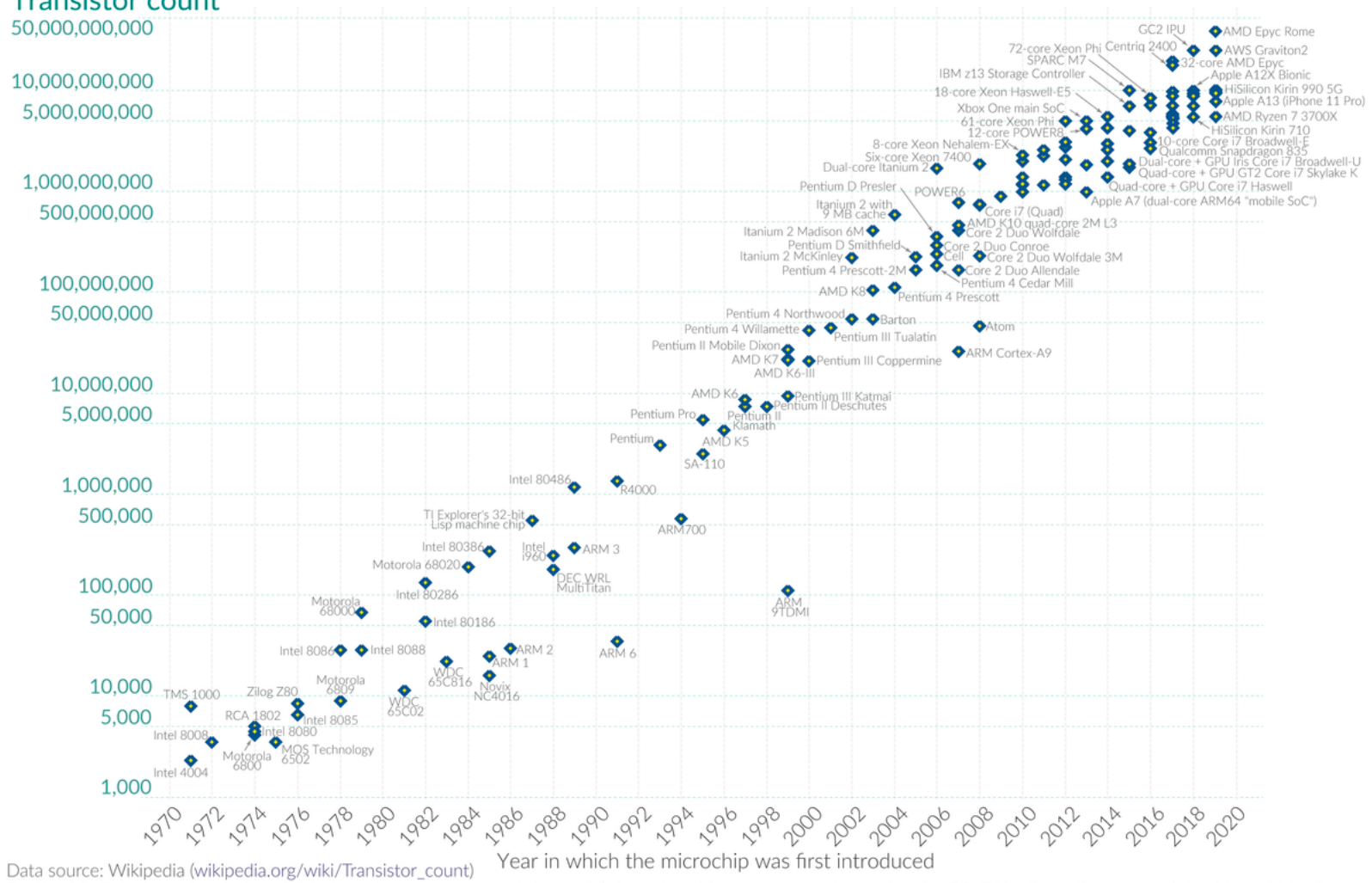
This week at the ASPLOS 2021 conference, Dr. Chris Lattner gave the keynote address to open the event with a discussion of the ...

A New Golden Age for Computer Architecture John L. Hennessy, David A. Patterson June 2018 End o... 22 个章节

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World in Data



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

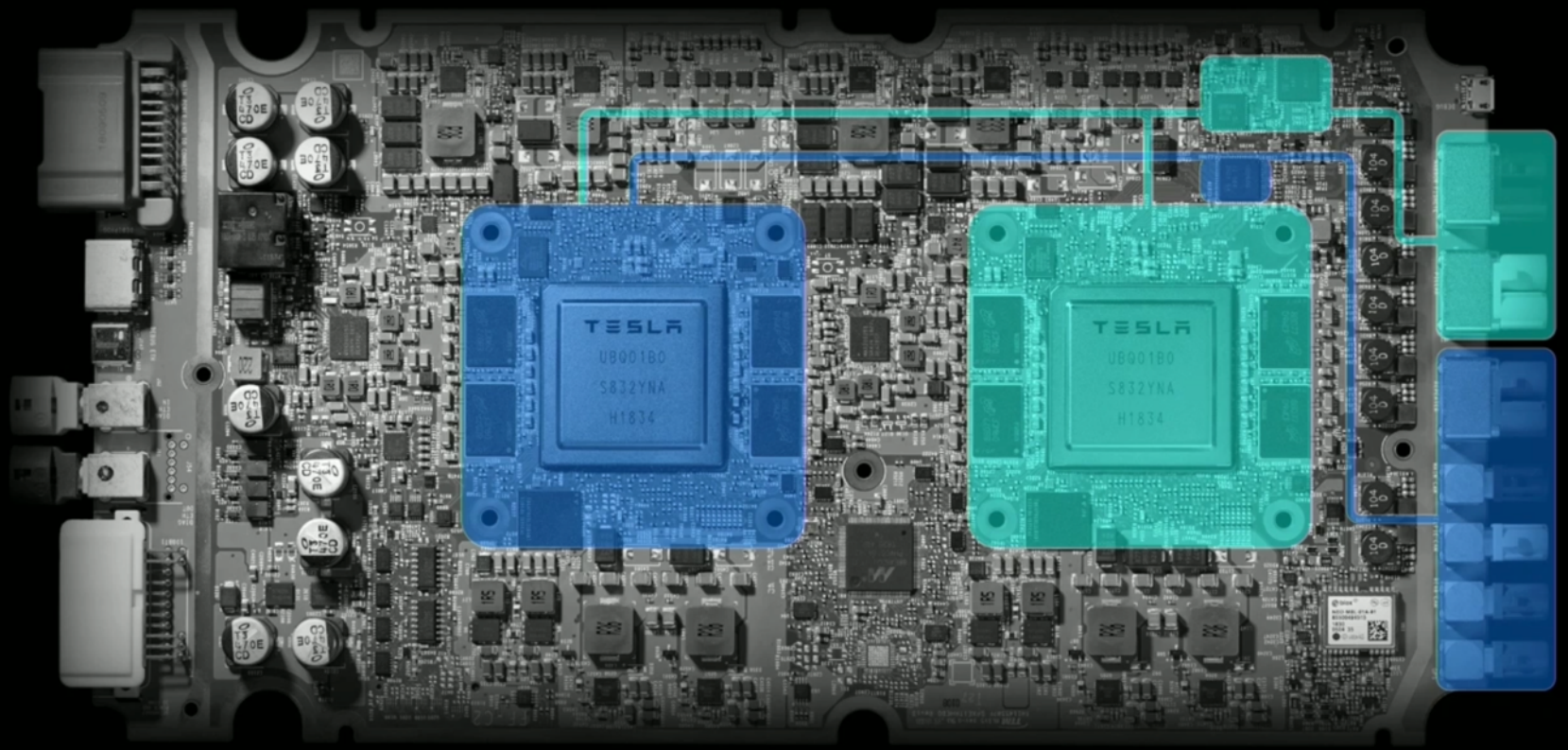
AI 芯片发展

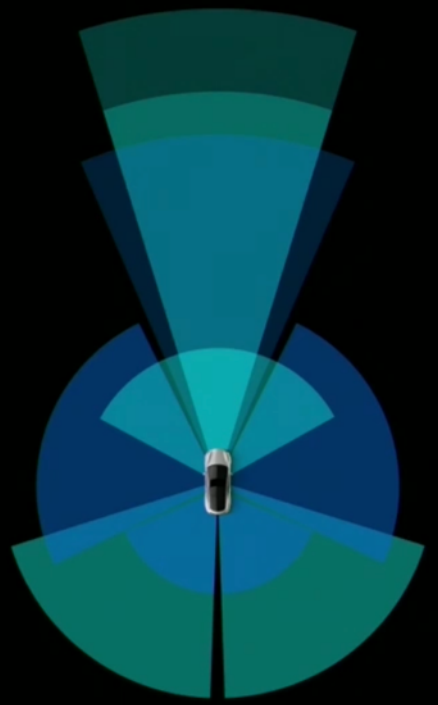
AI芯片算力三阶段

1. 第一阶段：芯片算力不足，神经网络没有被受到重视；
2. 第二阶段：CPU 算力大幅提升，但仍然无法满足神经网络增长需求；
3. 第三阶段：GPU 和AI芯片新架构推动人工智能快速落地；

异构与超异构场景

PERCEIVE

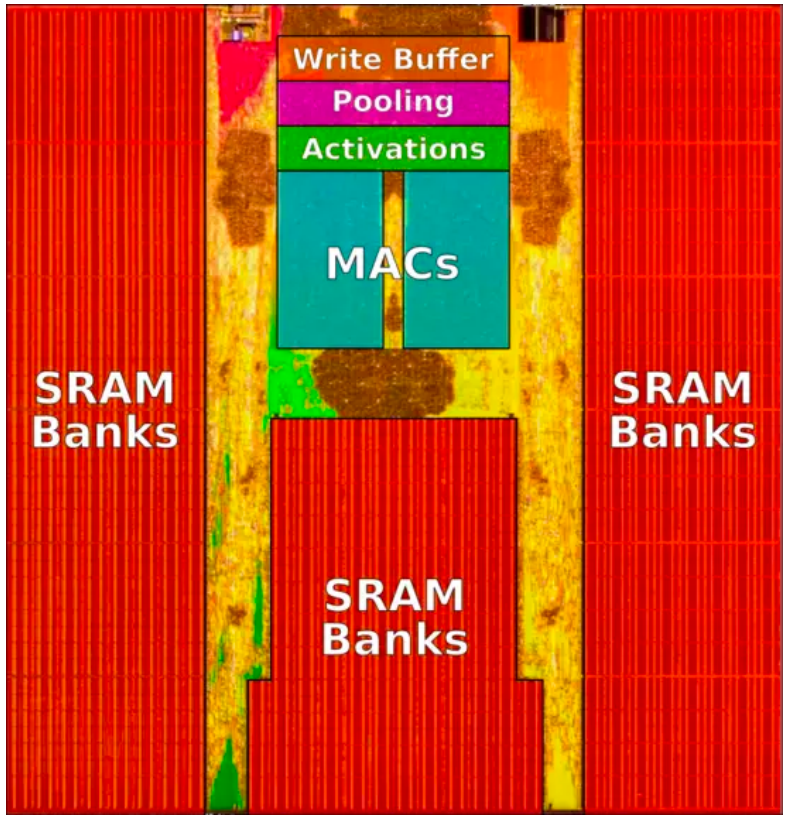




TESLA LIVE

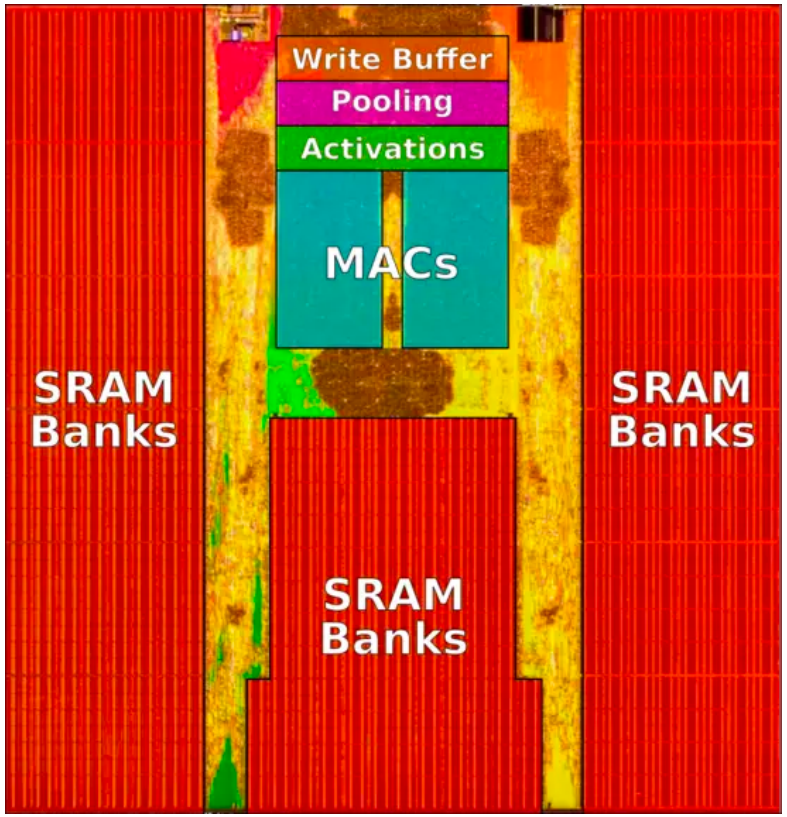
Radar – GPS – Maps – IMU – Ultrasonic – Wheel Ticks – Steering Angle

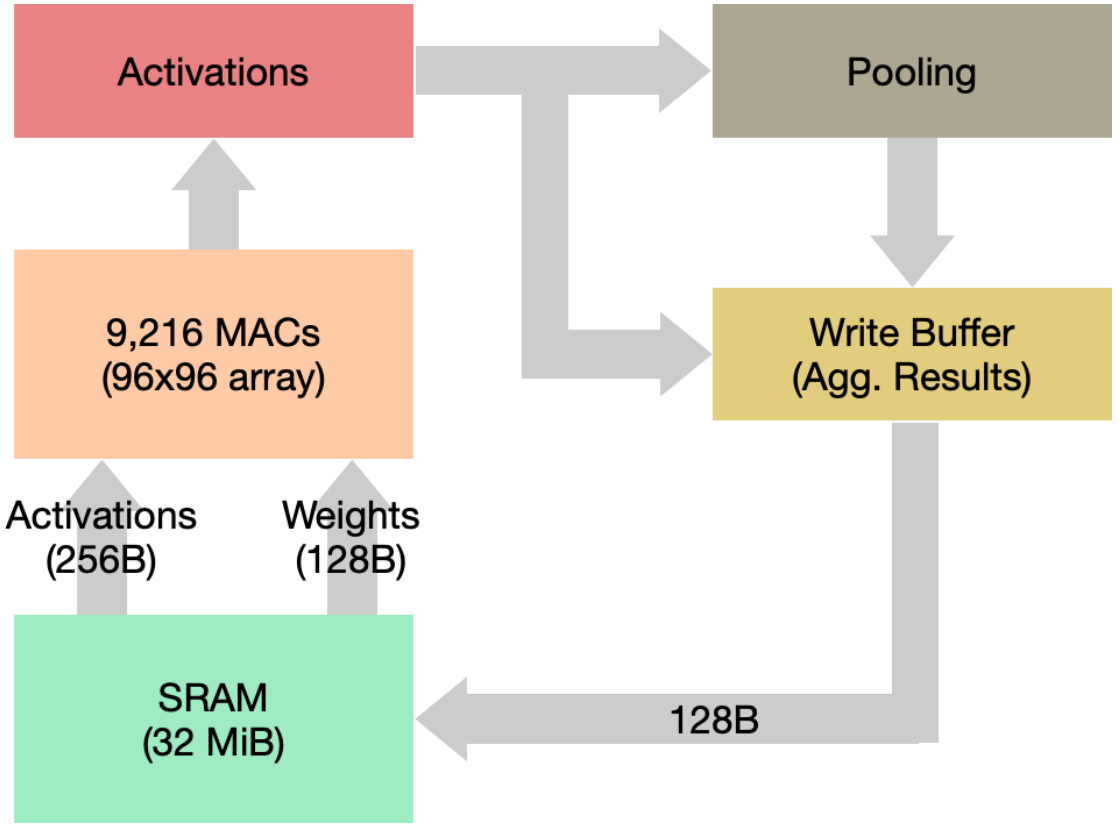
特斯拉 HW3 FSD 芯片



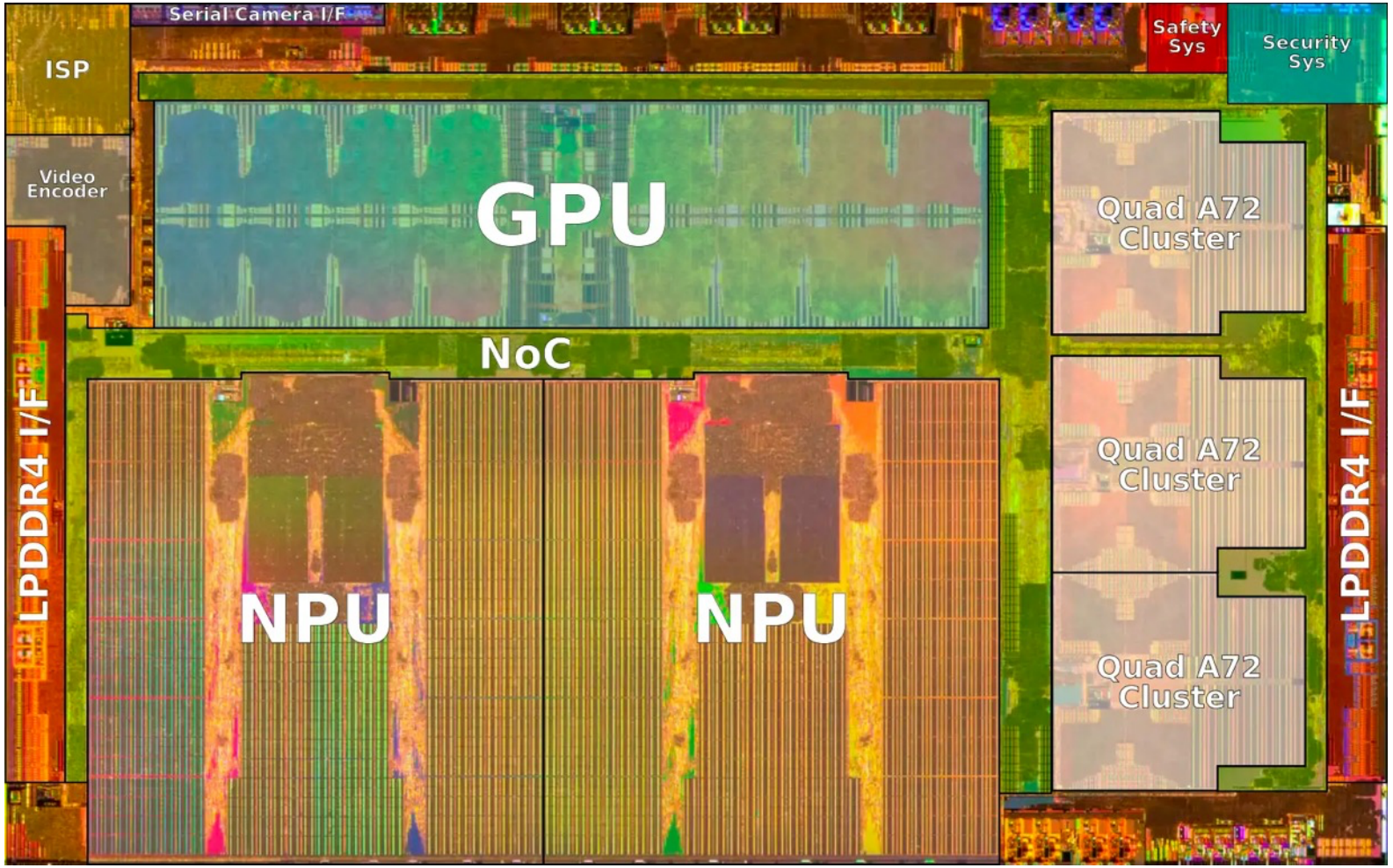
Chip	Tesla - FSD Chip	Qualcomm - Snapdragon 865 (Galaxy S20, March 6 2020)
Technology Node	Samsung 14 nm process	TSMC's advanced 7nm (N7P)
CPU	3x (4-core) Cortex-A72	4x Cortex-A77, 4x Cortex-A55 (4 high power, 4 low power)
GPU	Custom GPU, 0.6 TFLOPS @ 1 Ghz	Adreno 650, 1.25 TFLOPS @ 700 MHz -ish
NPU (AI accelerator)	2x Tesla NPU, each 37 TOPS (total 74 TOPS)	Hexagon 698 @ 15 TOPS
Memory (Cache)	2x 32MB SRAM for NPUs	1 MB L2, 4 MB L3, and 3 MB system wide cache
Memory (RAM)	8GB LPDDR4X, 2x 64-bit, Bandwidth 111 GB/s	16GB LPDDR5, 4x 16-bit , Bandwidth 71.30 GB/s
ISP (Image signal processor)	24-bit? 1 billion pixels per second	Spectra 480, dual 14-bit CV-ISP 2 Gpixel/s, H.265 (HEVC)
Secure Processing Unit	"Security system", verify code has been signed by Tesla.	Qualcomm SPU230, EAL4+ certified
"Safety System"	Dual-core CPU that checks congruency between the NPUs	None
TDP	36 Watt	5 Watt

特斯拉 HW3 FSD 芯片





特斯拉 HW3 FSD 芯片



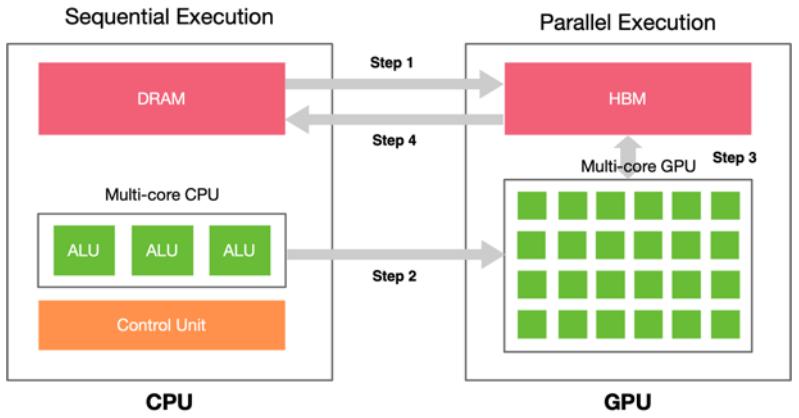
计算体系迎来异构

异构计算的出现

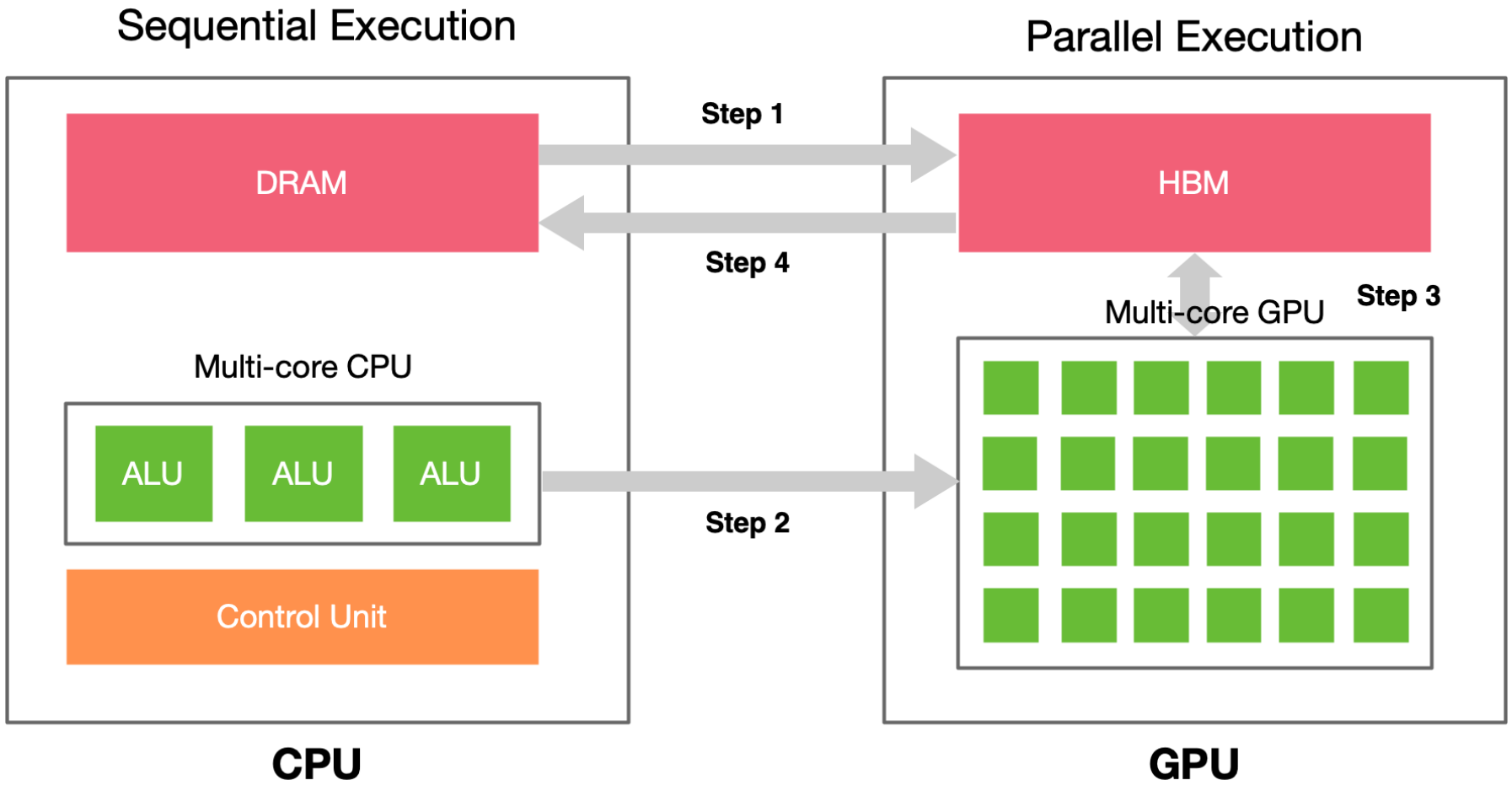
- 目前主流计算机微处理器普遍采用冯•诺依曼结构，受半导体物理技术和功耗（CPU主频达到制造工艺的极限，随着主频提高，功耗上升）、计算效率的限制，微处理器运算速度的提高已经趋于缓慢，**基于多核处理器或者集群计算机的并行计算技术逐渐成为提高计算机运算性能的主要手段。**
- 并行计算中的微处理器同样受冯•诺依曼瓶颈制约，在处理数据密集型计算时，计算速度和性价比不理想（深度学习计算领域，虽然NPU性能优于CPU，但NPU上层软件依然滞后），各大科研机构和芯片厂商都在探索基于异构计算 AI 芯片，希望 NPU 未来像 GPU 一样成为计算机体系的标配。
- **由于系统功耗限制、欠缺和芯片结构匹配的上层基础软件，想要在HPC中发挥AI芯片计算效率优势，只有在异构计算模式下采用新的计算原理、开发高性能算法和大规模并行基础软件。**

CPU 与 GPU 的异构工作流程 Workflow

- CPU把数据准备好，并保存在CPU内存中；
- 将待处理的数据从CPU内存复制到GPU内存（处理①）；
- CPU指示GPU工作，配置并启动GPU内核（处理②）；
- 多个GPU内核并行执行，处理准备好的数据（图中的③处理）；
- 处理完成后，将处理结果复制回CPU内存（处理④）；
- CPU把GPU的结果进行后续处理。



CPU 与 GPU 的异构工作流程 Workflow

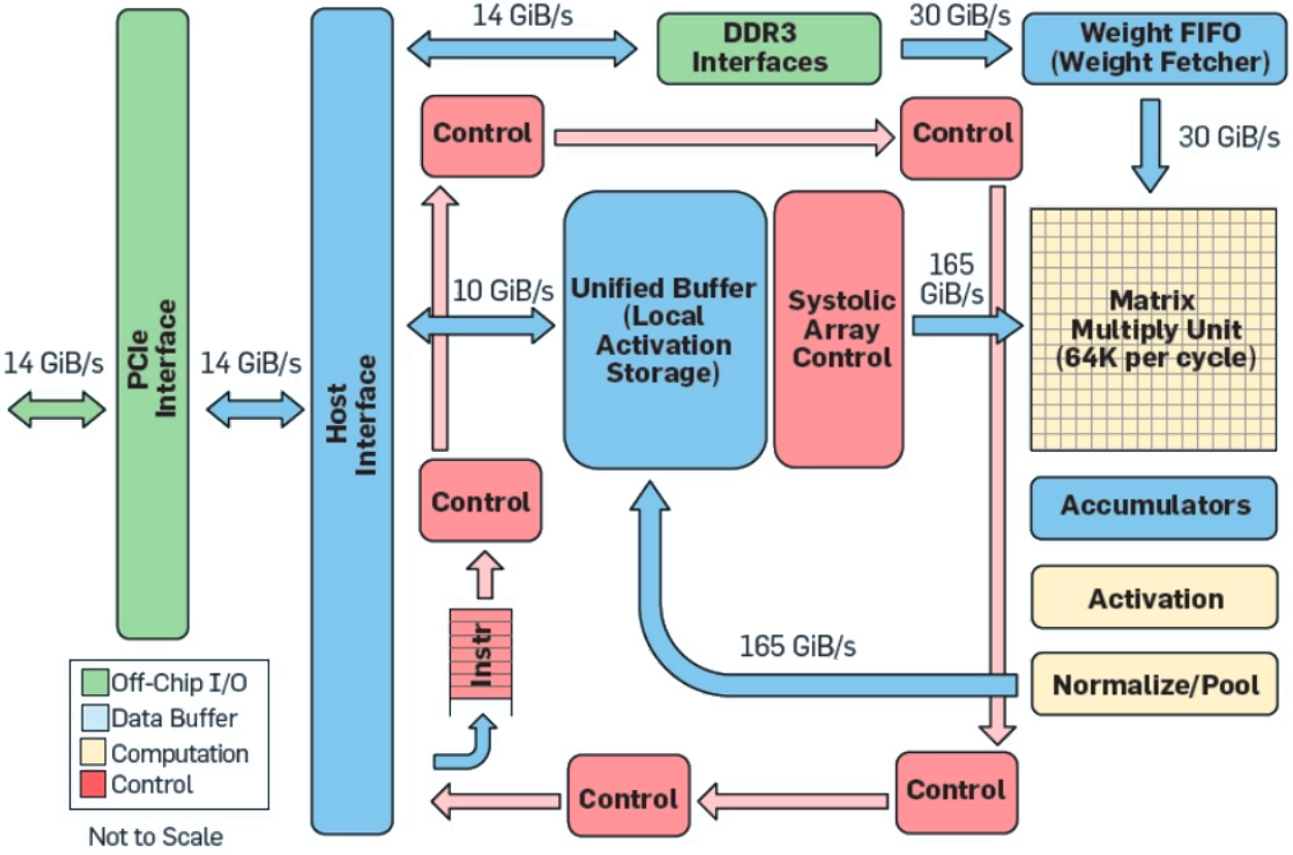


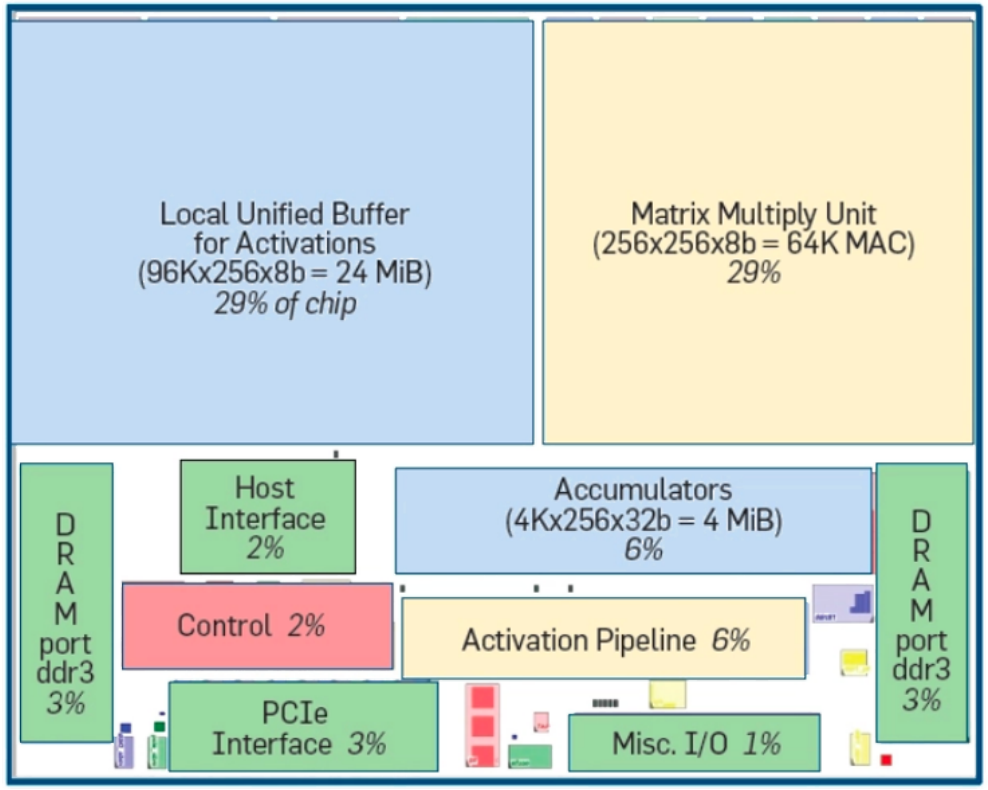
AISC

通过驱动程序和CSR和可配置表项交互，以此来控制硬件运行。和GPU类似，ASIC的运行依然需要CPU的参与：

- **数据输入**：数据在内存准备好，CPU控制ASIC输入逻辑，把数据从内存搬到处理器；
- **数据输出**：CPU控制ASIC输出逻辑，把数据从处理器搬到内存，等待后续处理。
- **运行控制**：控制CSR、可配置表项、中断等；

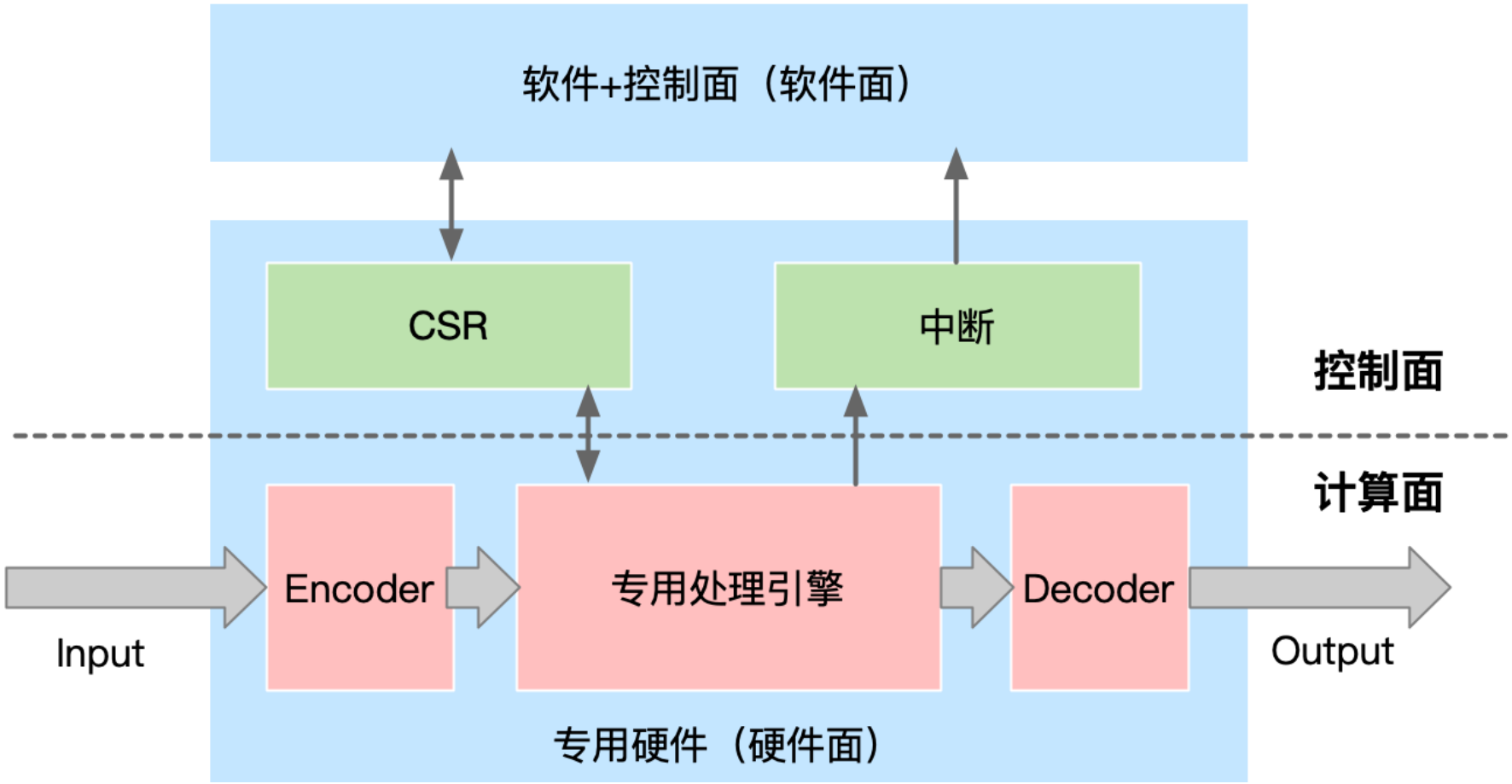
ASIC





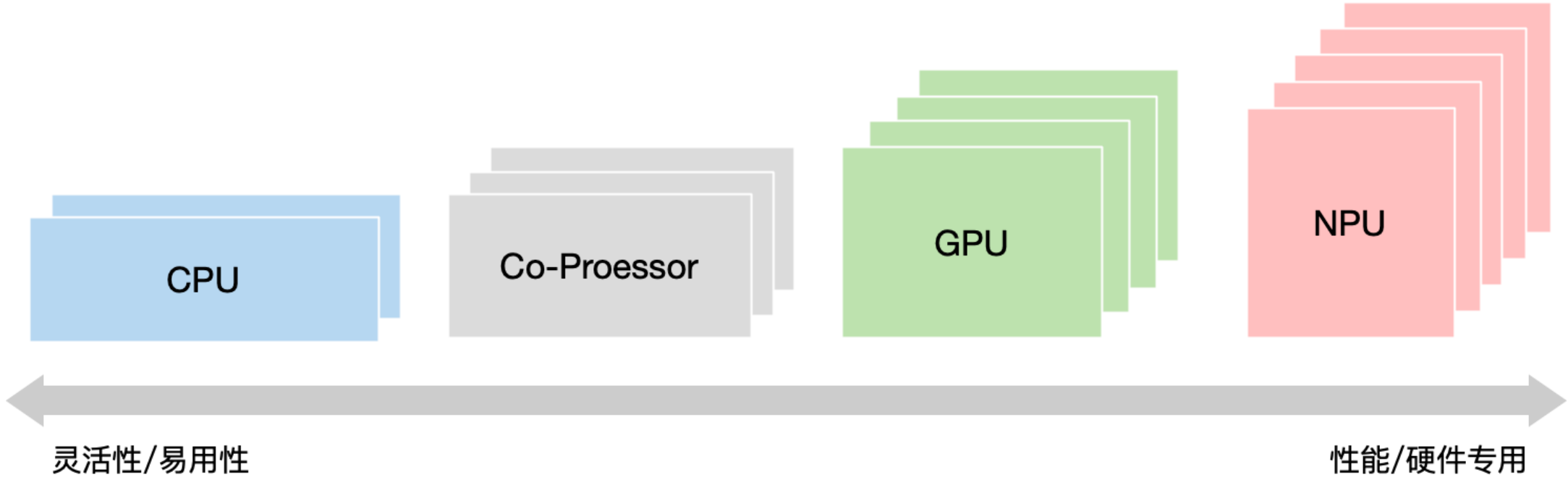
Google TPU Architecture

ASIC



从CPU到ASIC，架构越来越碎片化

- 宽度表示场景覆盖度；
- 高度代表专用性能；



从CPU到ASIC，架构越来越碎片化

- 指令是软件和硬件的媒介，指令的复杂度（单位计算密度）决定了系统的软硬件解耦程度，典型的处理器平台大致分为CPU、协处理器、GPU、FPGA、DSA、ASIC。指令复杂度越高，单个处理器器覆盖的场景就会越小，处理器器的形态就会越多。

- 从CPU到ASIC，处理器越来越碎片化，构建生态越来越困难。

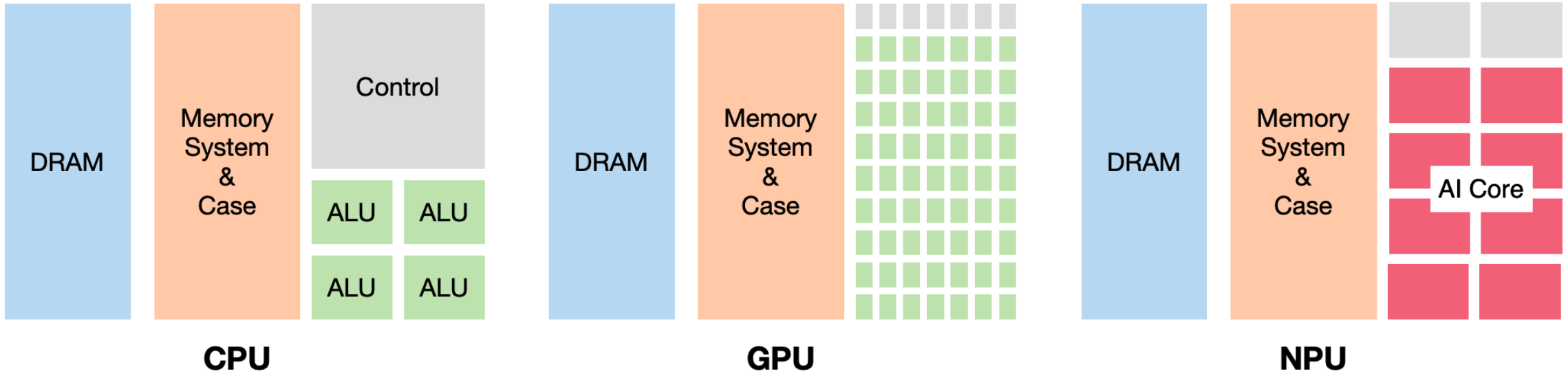
异构计算的问题

CPU + XPU 异构计算中的 XPU ，决定了整个系统的性能/灵活性特征：

1. GPU 灵活性较好，但性能效率不够极致；
2. DSA 性能好，但灵活性差，难以适应复杂计算场景对灵活性的要求。
3. FPGA 功耗和成本高，需要一些定制开发，落地案例不多。
4. ASIC 功能完全固定，难以适应灵活多变复杂计算场景。

异构计算的问题

- **复杂计算的挑战**：系统越复杂，需要选择越灵活的处理器；性能挑战越大，需要选择越偏向定制的加速处理器。

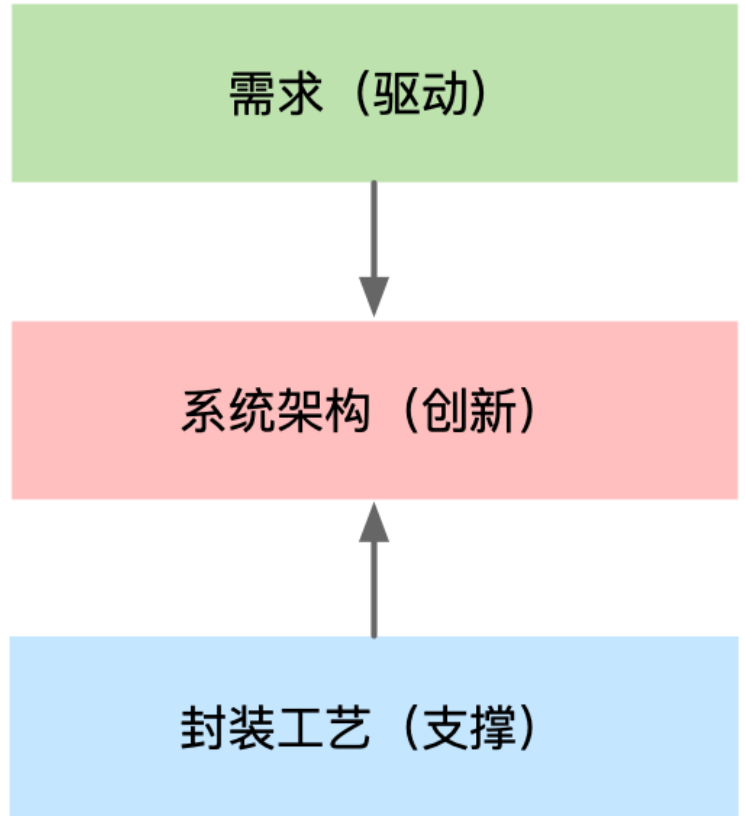


- **本质矛盾**：单一处理器无法兼顾性能和灵活性。

从异构到超异构

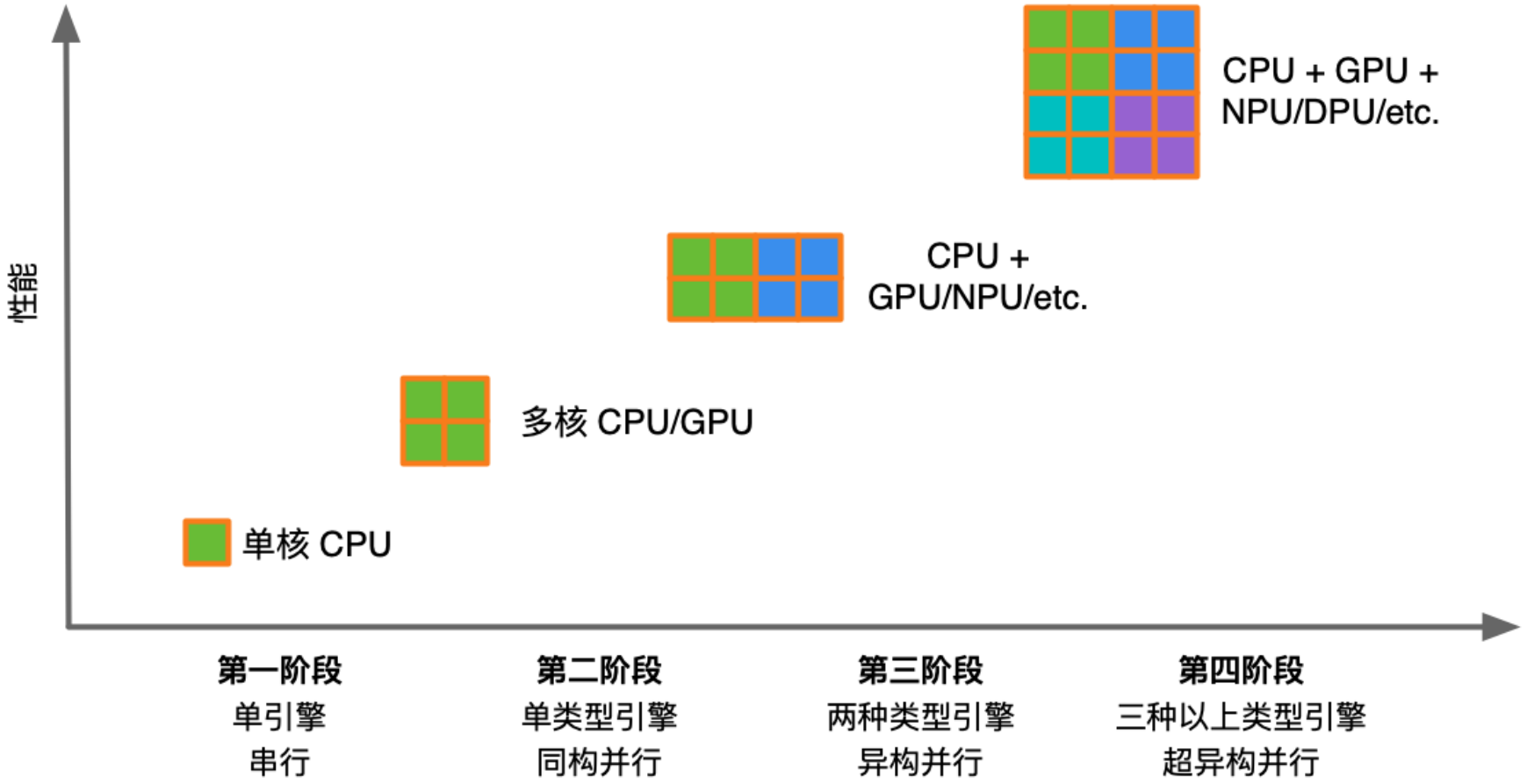
why now ?

1. **需求驱动**：软件新应用层出不穷，两年一个新热点；并且，已有的热点技术仍在快速演进。元宇宙是继互联网和移动互联网之后的下一个互联网形态，要想实现元宇宙级别的体验，需将算力提升1000倍。
2. **工艺和封装支撑**：工艺封装持续进步，10nm以下芯片从2D->3D->4D。Chiplet使得在单芯片层次，可以构建规模数量级提升的超大系统。系统规模越大，超异构的优势越明显。
3. **系统架构持续创新**：通过架构创新，在单芯片层次，实现多个数量级的性能提升。挑战：异构编程很难，超异构编程更是难上加难；如何更好地驾驭超异构，是成败的关键。

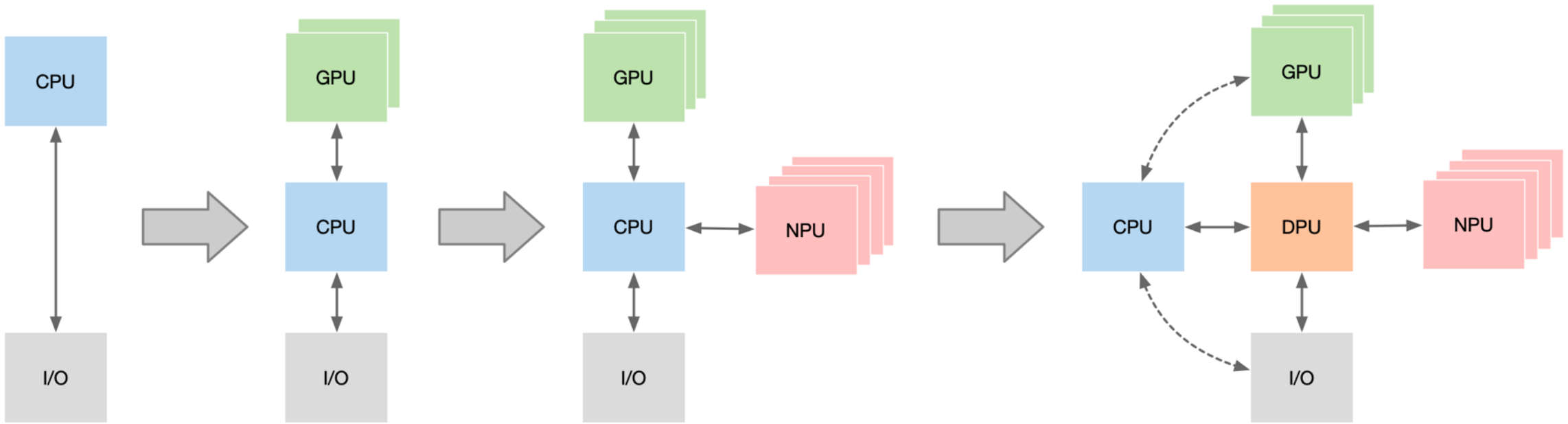


计算架构演进

- 计算从单核的串行走向多核的并行；又进一步从同构并行走向异构并行。

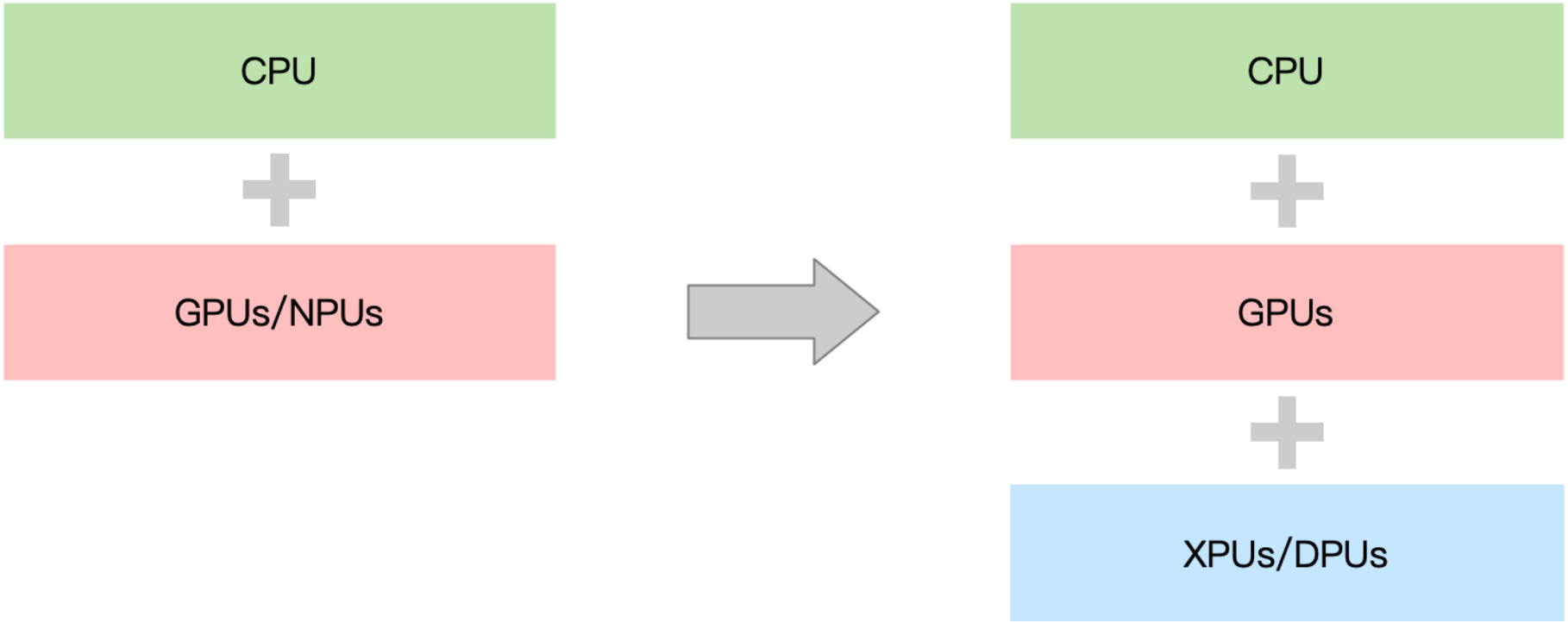


异构计算的发展



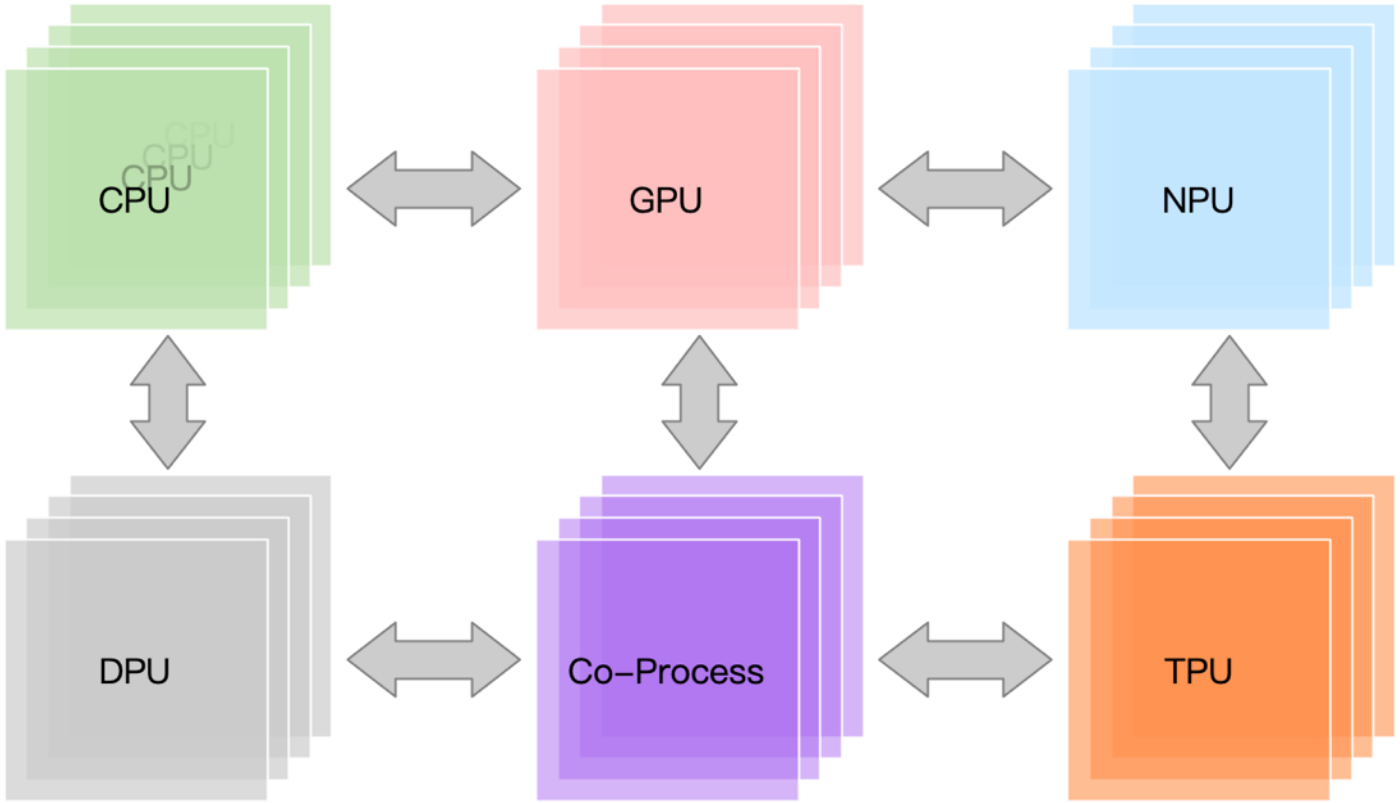
从异构并行，走向超异构并行

- 计算需要进一步从异构并行走向超异构并行。异构计算是 CPU + XPU 的两个层次的处理器类型，而超异构计算则是 CPU + GPU + DSA 三个层次的处理器类型。



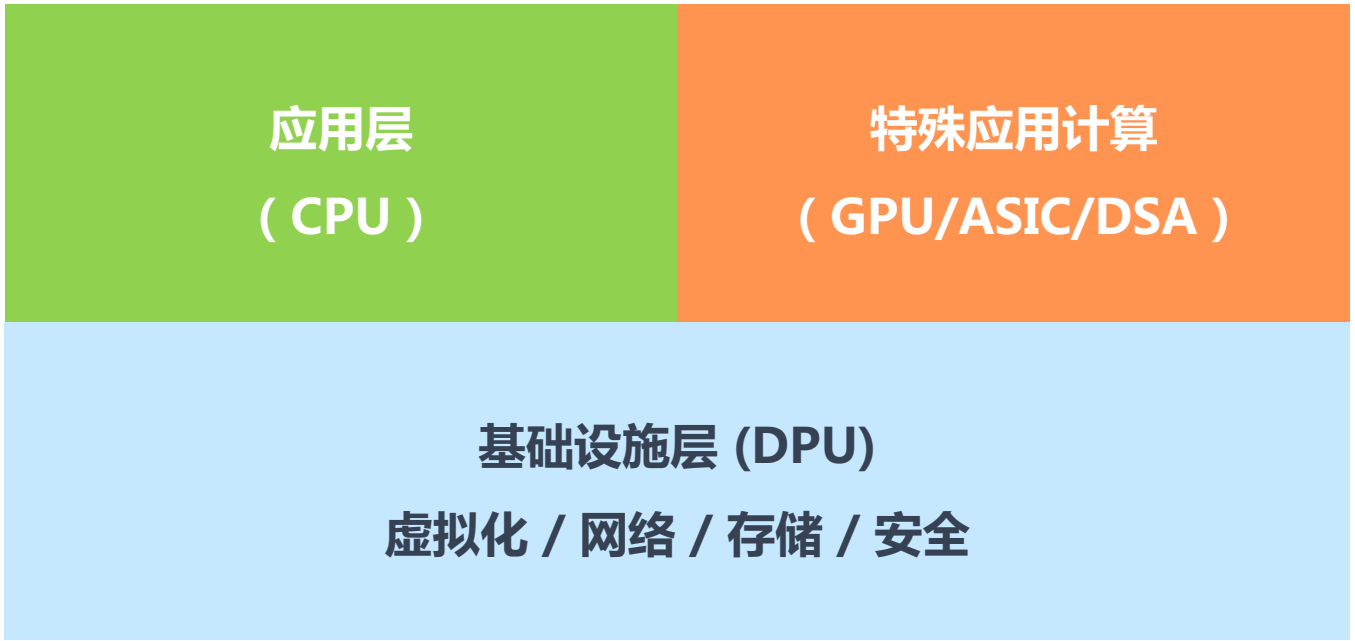
超异构计算

- 超异构计算，非简单的集成，而是把更多的异构计算整合重构，各类型处理器间充分、灵活的数据交互，形成统一的超异构计算体系。



基础特征

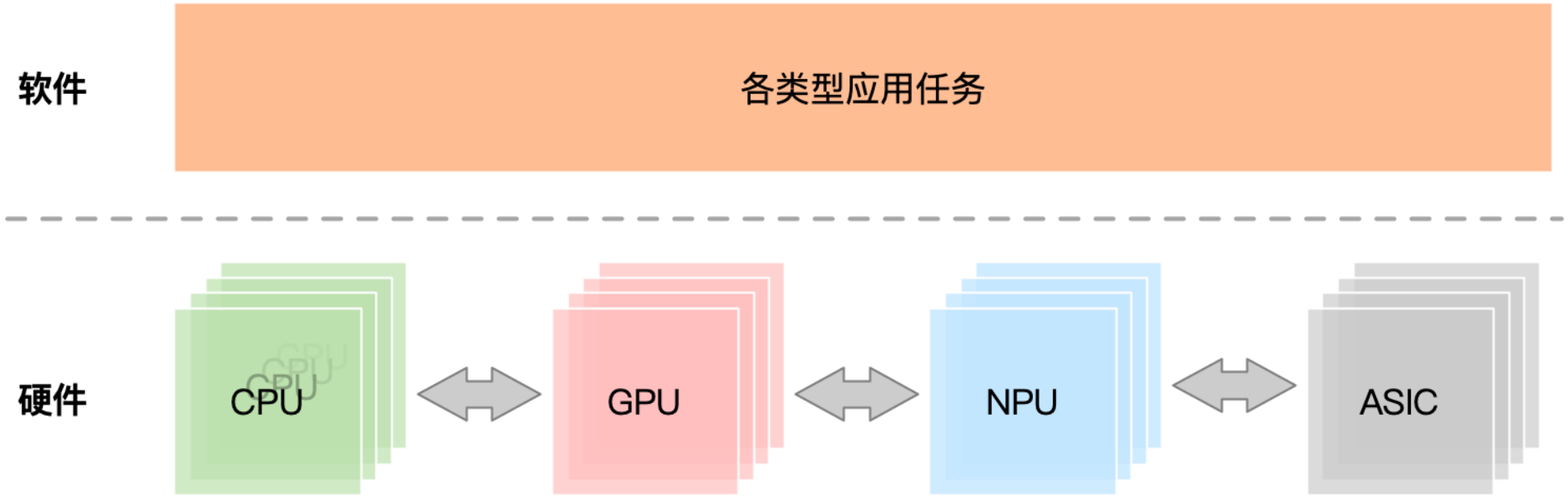
1. 超大规模的计算集群；
2. 复杂计算系统，由分层分块组件组成；



超异构
挑战与思考

超异构的软件层

- 软件需要跨平台复用：跨架构、跨不同处理器类型、跨厂家平台、跨不同位置、跨不同设备类型。因此软件架构的复杂性增长，会成为一个最大的挑战。



Question?

- 如此复杂的超异构该如何驾驭？

开放生态、接入社区

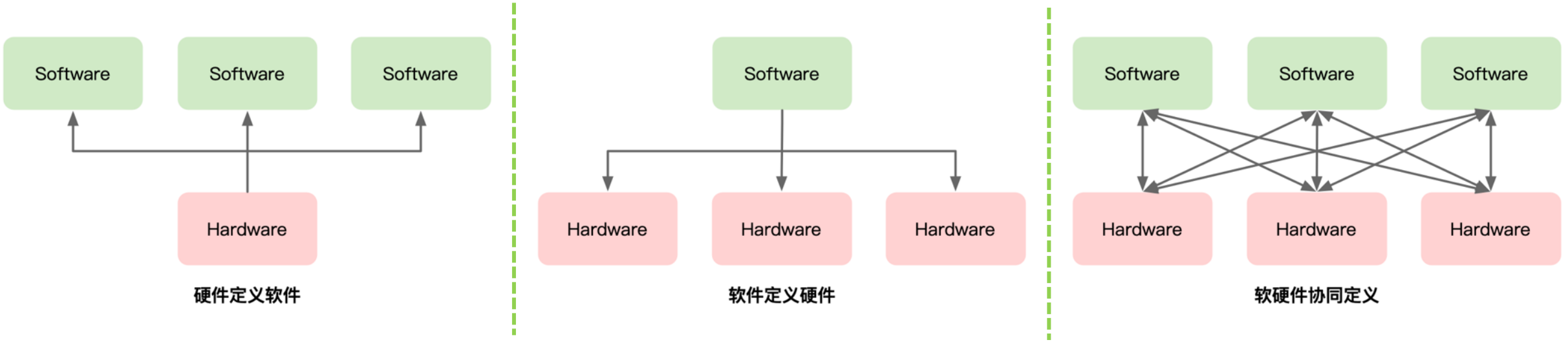
- **开放接口/架构及生态：**形成标准的开放接口/架构；开发者遵循接口/架构开发产品和服务，从而形成开放生态。
- **软件兼容：**尽可能减少针对已有应用的定制化开发，兼容已有软件生态，通过基础软件（如编译层）对接加速应用软件；
- **编程体系：**提供门槛更低的编译体系，通过编程体系构建上层加速库从而对接领域应用，即提供门槛较低的标准领域编程语言（如CUDA）；
- **开放架构：**进一步开放软硬件架构，防止架构过多导致的市场碎片化，如90年代编译器风起云涌到目前聚焦2/3个编译器。

Question?

- 硬件定义软件，还是软件定义硬件？

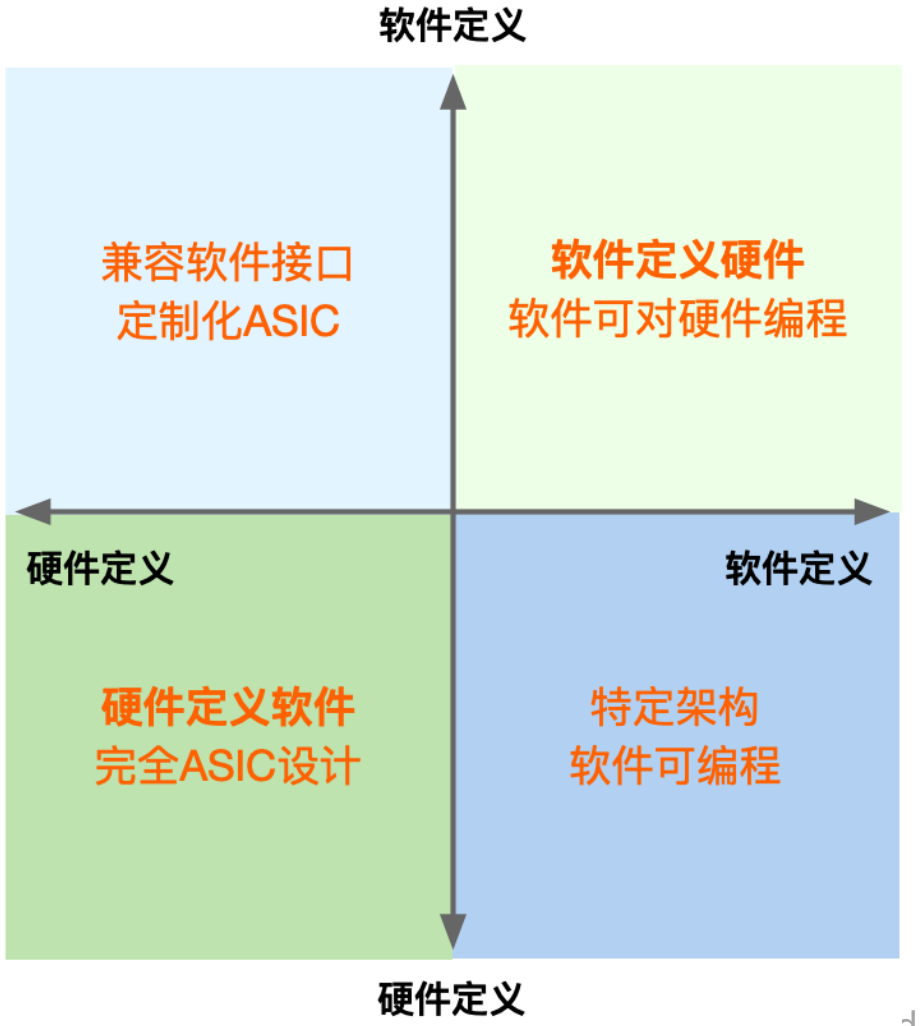
Question?

- 硬件定义软件，还是软件定义硬件？



Hardware vs Software define

- **硬件定义软件：**
  - 系统业务逻辑以硬件实现为主，软件实现为辅；软件依赖于硬件提供的接口构建（e.g. 早期的操作系统）。
- **软件定义硬件：**
  1. 系统业务逻辑以软件实现为主，硬件实现为辅（e.g.，AI4SIC模拟仿真软件）；
  2. 硬件对于软件可编程，硬件按照软件编程逻辑执行操作（e.g.，带有渲染管道Pipeline的GPU）；
  3. 硬件依赖于软件提供的接口构建（e.g. AI 算法）



Hardware vs Software define

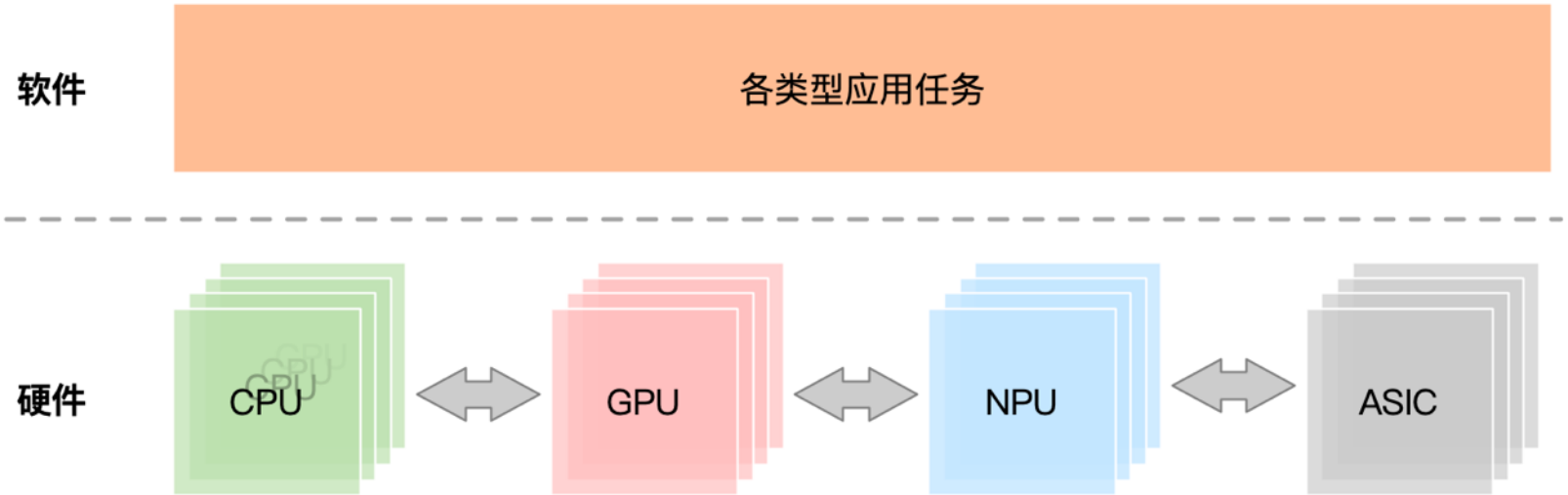
硬件定义软件，还是软件定义硬件，跟系统复杂度是休戚相关：

- **CPU**：系统复杂度较小，迭代较慢。可以快速设计优化的系统软硬件划分，先硬件开发，然后开始系统层和应用层的软件开发（Windows操作系统等软件）。
- **GPU**：量变引起质变，随着系统复杂度上升，系统迭代快，直接实现一个完全优化的设计难度很大。系统实现变成了演进式：1）前期系统不够稳定，算法和业务逻辑在快速迭代，需要快速实现想法。2）随系统发展，算法和业务逻辑逐渐稳定，后续逐步优化到GPU、DSA等硬件加速来持续优化性能。

本质上是系统定义：系统复杂度过高，实现难以一次到位，系统实现，变成了持续优化和迭代的过程。

计算体系 vs 编译体系

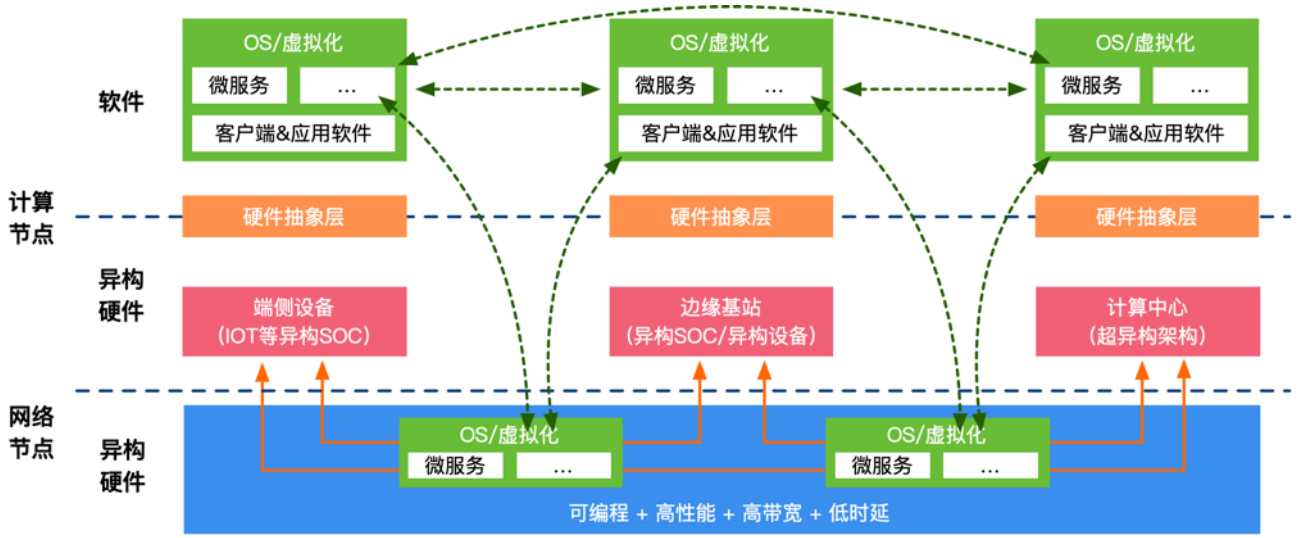
- 超异构架构的处理器越来越多，需要构建高效、标准、开放接口和架构体系（e.g.，OpenCL），才能构建一致性的宏架构（多种架构组合）平台，才能避免场景覆盖的碎片化。
- 现在正处于计算体系变革和编译体系变革10年，避免为了某个应用加速而去进行非必要大量上层应用迁移对接到硬件API，应交由一致性的宏架构（多种架构组合）平台（编译/操作系统）。



计算资源中心化，提升算力利用率

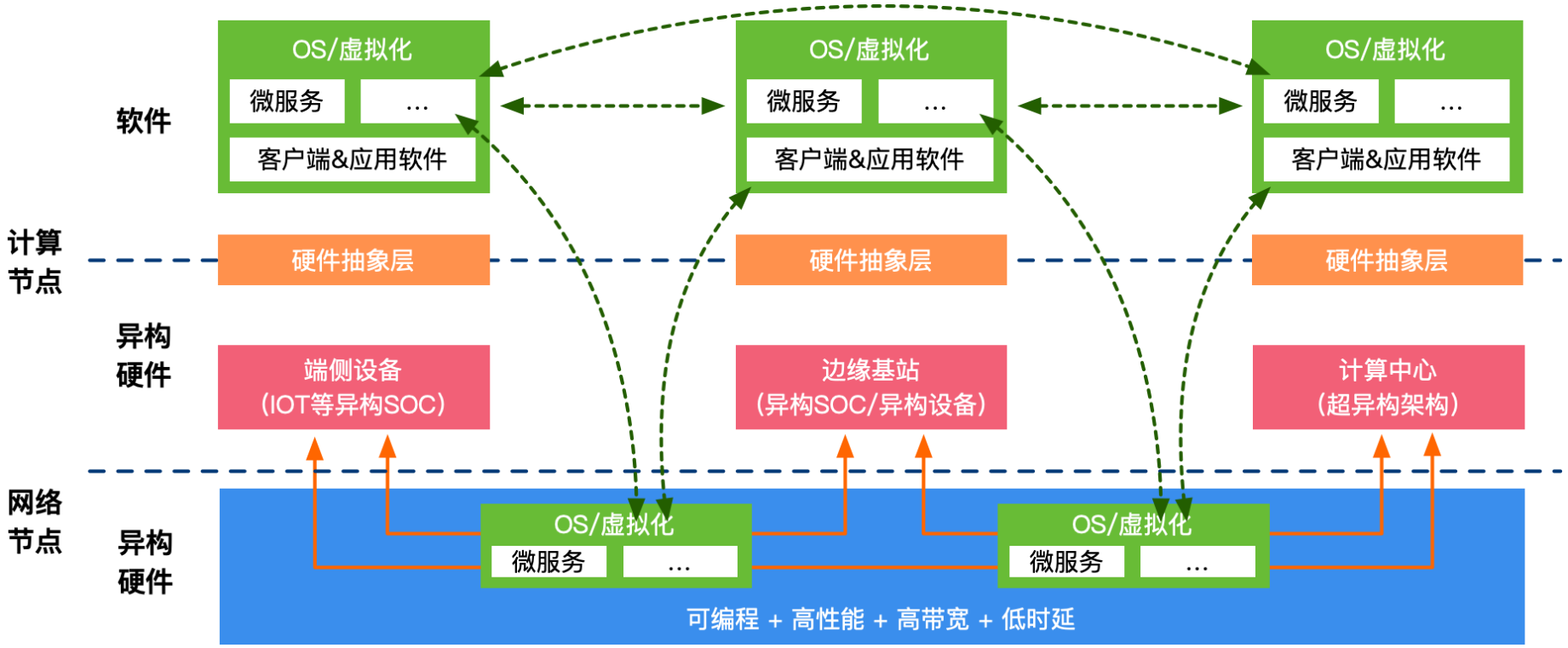
跨平台统一计算架构，把孤岛计算资源连接起来，实现计算资源池化，提升算力利用率：

1. **跨同类处理器架构**：应用软件可跨x86、ARM和RISC-V等CPU运行。
2. **跨不同类处理器架构**：软件跨CPU、GPU、FPGA和DSA等处理器运行。
3. **跨芯片平台**：软件在 Intel、Huawei、NVIDIA 等不同公司芯片运行。
4. **跨云边端**：计算根据应用场景的部署情况，自适应选择运行在云边端中。



计算资源中心化，提升算力利用率

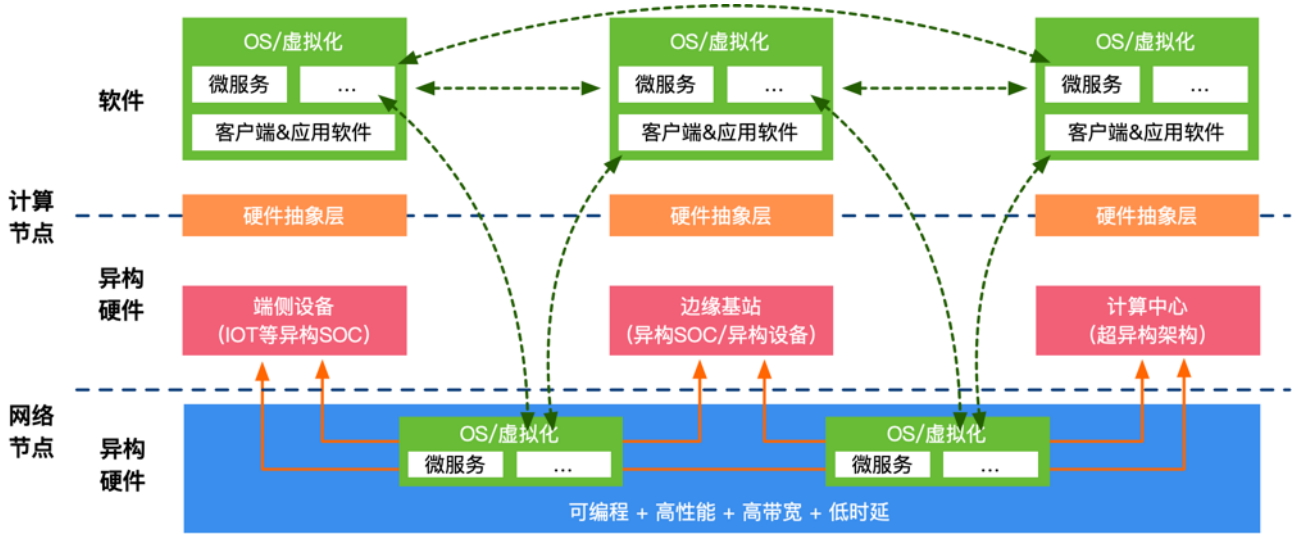
- 跨平台统一计算架构，把孤岛计算资源连接起来，实现计算资源池化，提升算力利用率。
- 超异构时代，形成开放生态，让计算资源形成资源池，满足更复杂的应用场景对算力无限的需求。



Summary：软硬件共同定义，超异构开放生态

软件应用算法支持硬件加速：

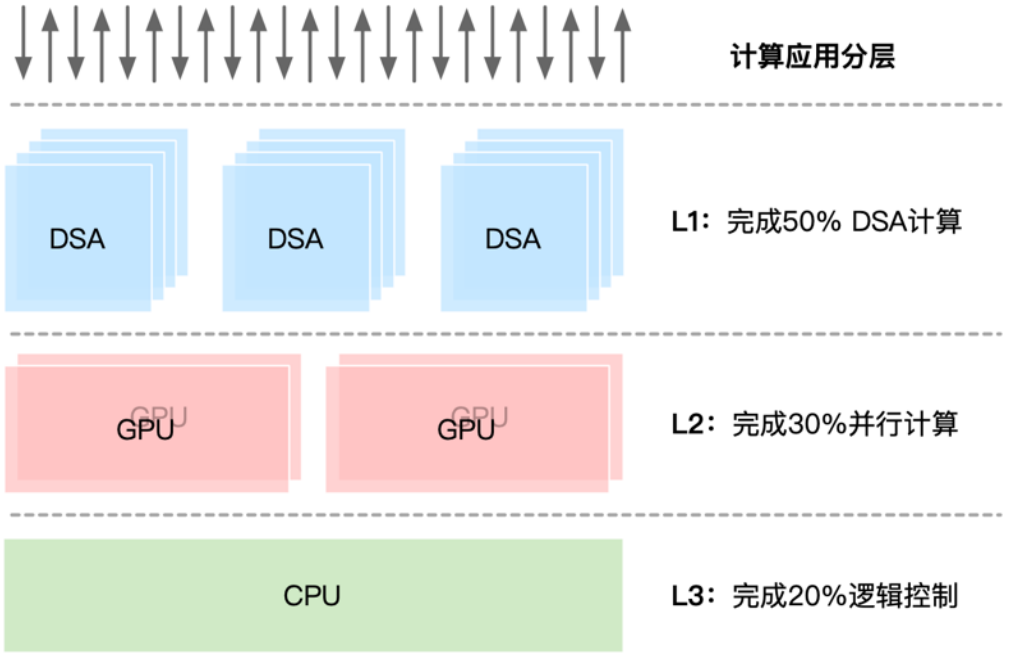
- 软件架构调整，控制面和计算/数据面分开，并接口标准化；
- 加速硬件的资源触发，底层基础软件（编译器）自适应选择计算/数据；
- 数据输入/输出来源于软件，也可以来源于硬件，更多可以下沉到硬件独立传输计算；



Summary：软硬件共同定义，超异构开放生态

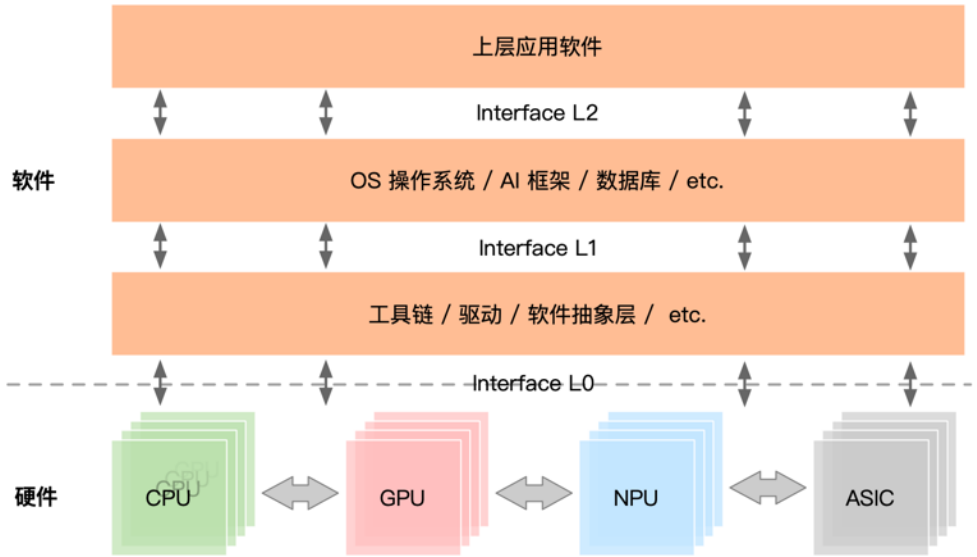
极致性能优化的分层可编程体系架构：

- 在超异构系统中，绝大部分计算交由给DSA进行极致计算，因此系统整体性能效率接近DSA；
- 用户角度应用运行在CPU，开发者感知的是CPU可编程，通过操作系统和编译器区分异构；
- Chiplet + 超异构，系统规模数量级提升，使得整体超异构系统性能数量级提升；



Summary：体系异构、平台化、开放生态

1. **超异构计算架构**：CPU+GPU+FPGA+DSA 多架构处理器组成的超异构计算。目标是接近CPU的灵活性，接近ASIC的性能效率，实现不影响开发效率下的数量级整体性能提升。
2. **平台化 & 可编程**：目标软件定义一切，硬件加速一切。完全可软件编程的硬件加速平台，完全由软件编程决定业务逻辑。足够通用性，满足多场景、多用户需求，满足业务演进。
3. **建立标准 & 开放生态**：架构/接口标准的开放，持续演进，拥抱开源开放的生态，支持云原生、云网边端融合，实现用户无（硬件/框架等）等平台依赖。



引用

1. https://www.youtube.com/watch?v=kFT54hO1X8M
2. https://www.youtube.com/watch?v=4HgShra-KnY
3. https://www.youtube.com/watch?v=NJVcsvQ30AQ&t=369s
4. https://www.nextbigfuture.com/2021/03/nvidia-drive-uses-1000-watts-but-tesla-hw3-fsd-chip-uses-36-watts.html
5. https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip
6. https://mp.weixin.qq.com/s/mYzBANBjGqjEhKgBC9mSbA