北京大学
PEKING UNIVERSITY

智能硬件体系结构

第十一讲：人工智能加速器架构

主讲：陶耀宇、李萌

2024年秋季

注意事项

北京大学 PEKING UNIVERSITY

- **课程作业情况**
  - 第4次作业**12月2日**截止（缓存设计）
  - 未来**2次作业（12.1-12.15、12.15-12.30）**
    - AI芯片架构、软硬加协同优化
  - **Take Home考试**
    - **推迟到11月30日00:00 – 12月1日11:59（周六、日）**
    - 每迟交一天扣10分
  - **Paper Review（12月25号/27号1-3点，每个同学10分钟）**
    - 时间有疑义请尽快联系老师或助教提出更换
    - 请自行阅读近5年内的体系结构相关论文1-2篇
    - 做8页左右的PPT，**每个同学10分钟演讲 + 1~2分钟答疑**
  - 第二次实验将在**12月1号**放出，截止**12月31号**

目 录

CONTENTS

A New Golden Age for Computer Architecture

Innovations like domain-specific hardware, enhnaced security, open instruction sets, and agile chip development will lead the way.

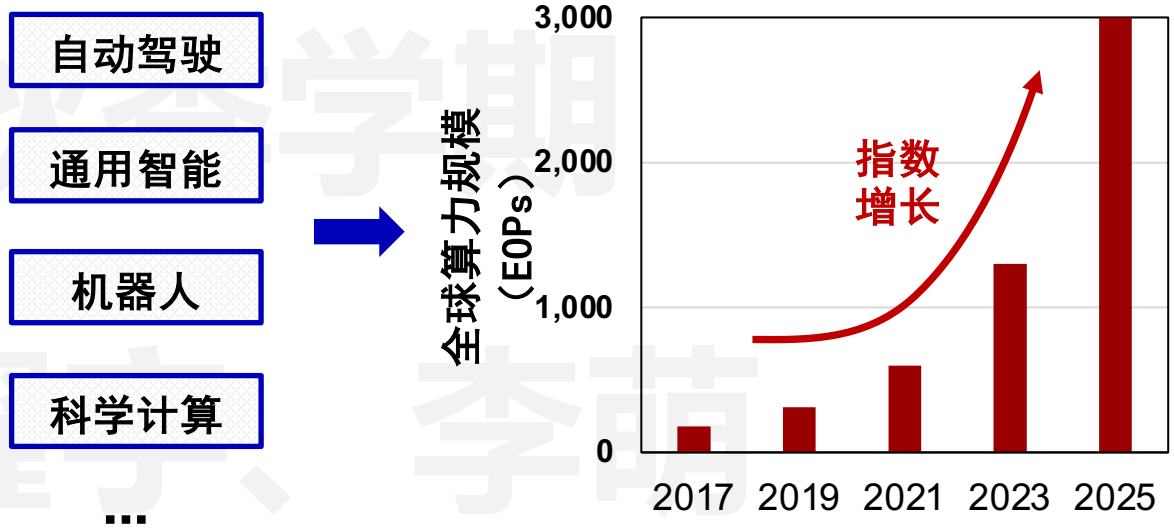
-- By John L. Hennessy and David A. Patterson

目标应用 —— 深度学习与深度神经网络

- 过去10年，得益于**算力的指数级**提升，以深度学习为代表的人工智能快速发展
  - 从全连接网络（MLP）到卷积神经网络（CNN）、循环神经网络（RNN）再到注意力网络（Transformer）和大模型
- 人工智能已经成为驱动半导体和硬件体系结构发展的重要驱动力

时间	模型	模型参数量 (GB)	模型算力需求 (TFLOPs-day)
2012	AlexNet	10^{-2}	10^{-3}
2015	ResNets	10^{-1}	10^{-1}
2017	Transformer	1	10^{1}
2020	GPT-3	10^{2}	10^{6}
2023	ChatGPT	10^{3}	10^{7}

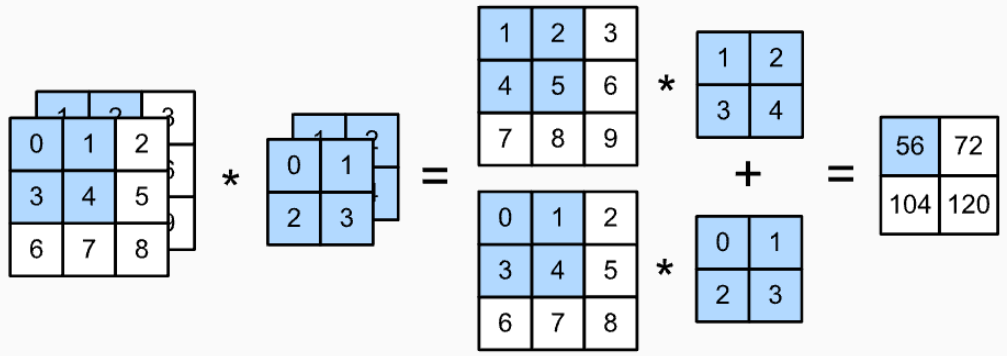
AI算法参数量与算力需求增长迅猛

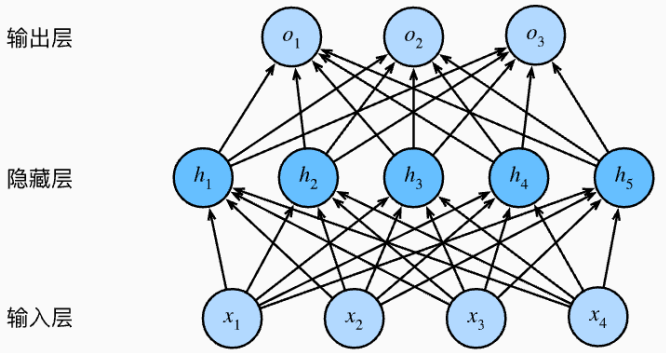


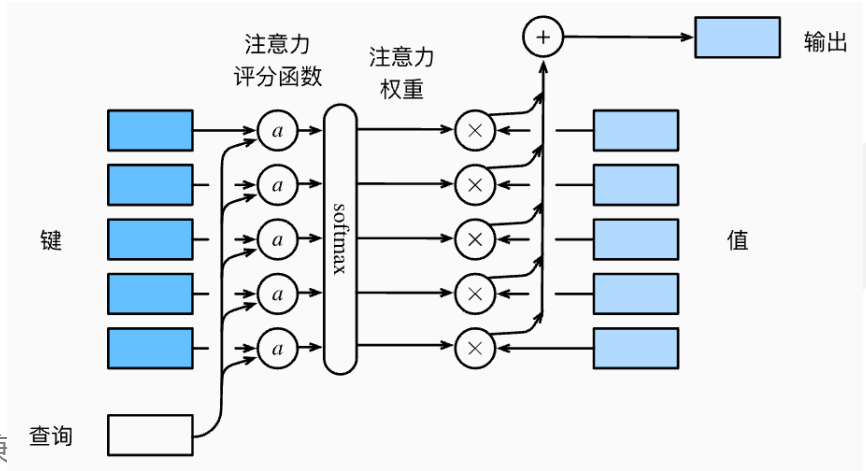
算力需求驱动硬件体系结构发展

目标应用 —— 深度学习与深度神经网络

- 典型神经网络**算子**和**拓扑**结构
  - **算子**：卷积、全连接、注意力机制、归一化函数、非线性函数等，主要操作对象为张量、向量等

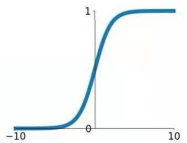






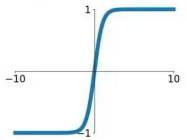
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



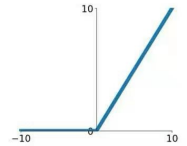
tanh

$$\tanh(x)$$



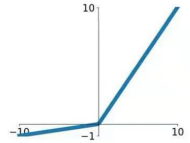
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

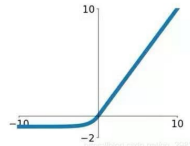


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

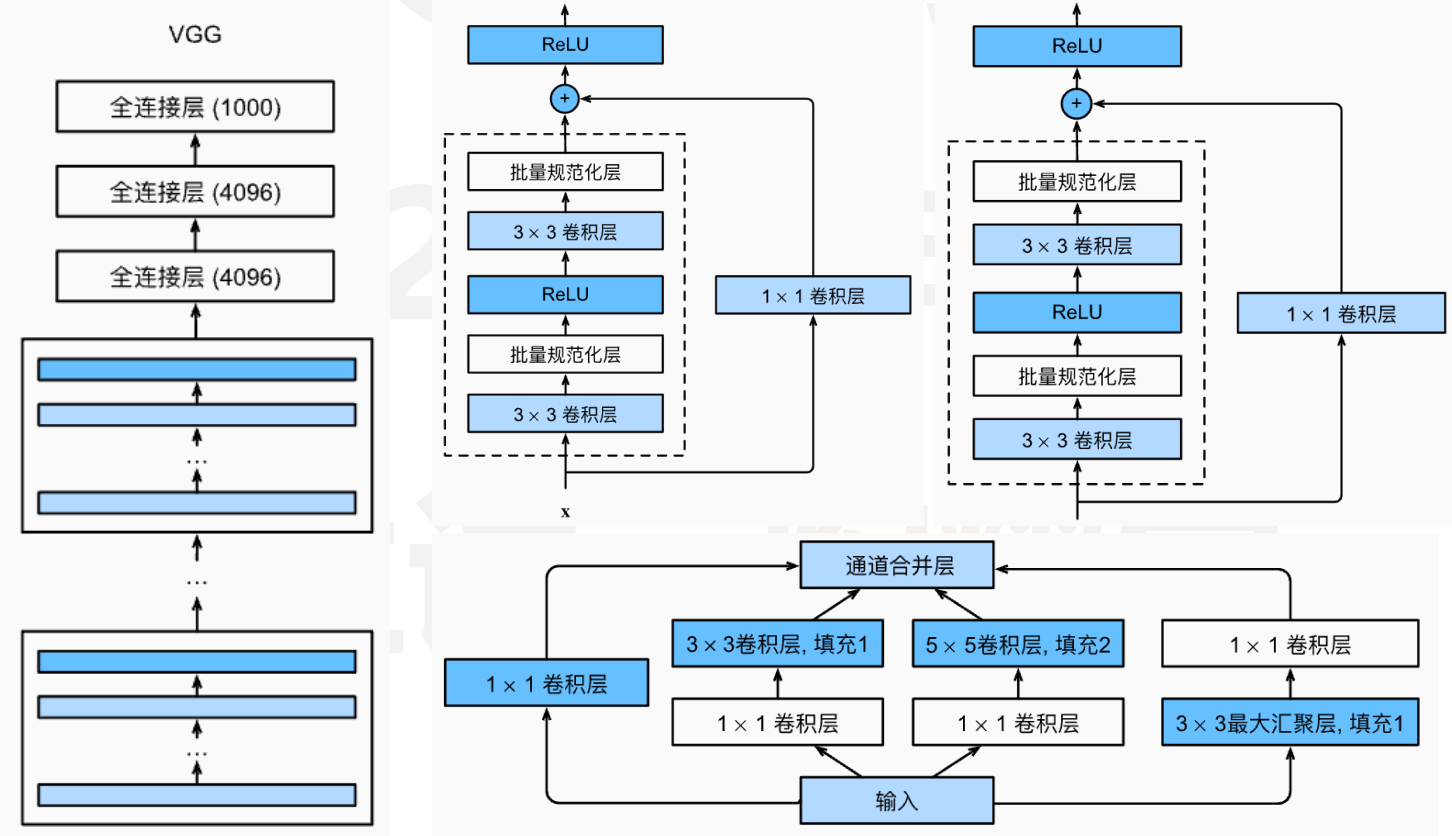
ELU

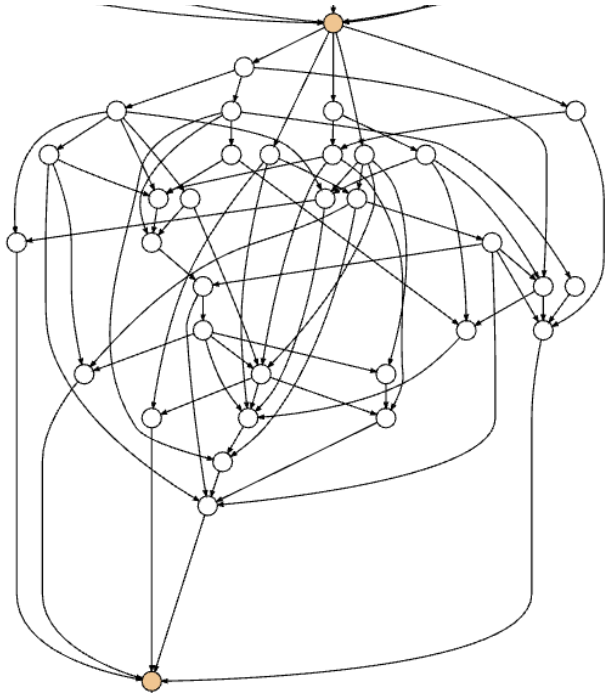
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



目标应用 —— 深度学习与深度神经网络

- 典型神经网络**算子**和**拓扑**结构
  - **算子**：卷积、全连接、注意力机制、归一化函数、非线性函数等，主要操作对象为张量、向量等
  - **拓扑**：线性结构、残差结构、多分支结构、随机结构



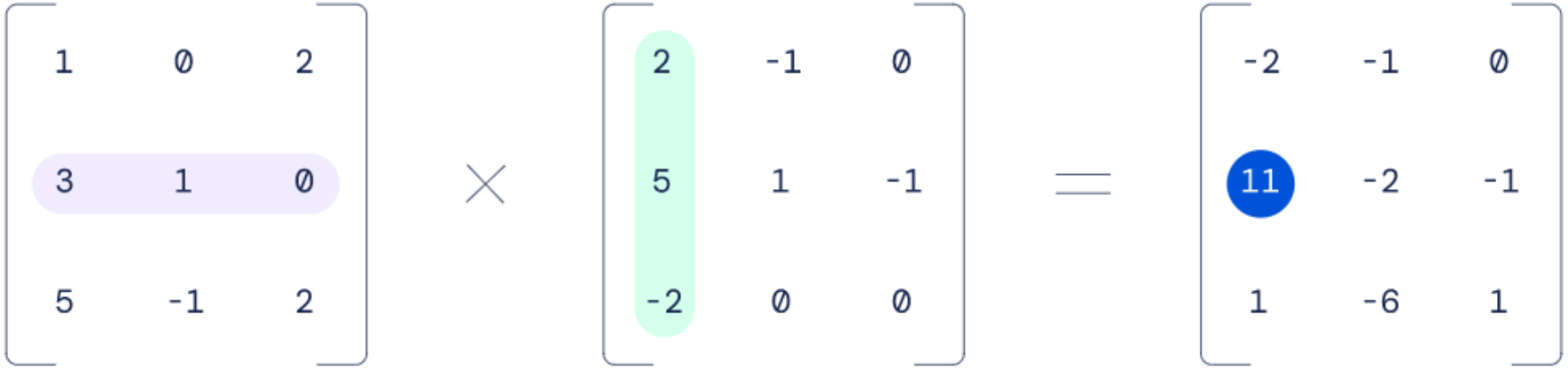


目标应用 —— 深度学习与深度神经网络

北京大学 PEKING UNIVERSITY

“没关系，AI芯片都是在加速线性算子，而线性算子都可以视为矩阵乘法”

—— 常见误区



AI加速器架构基础

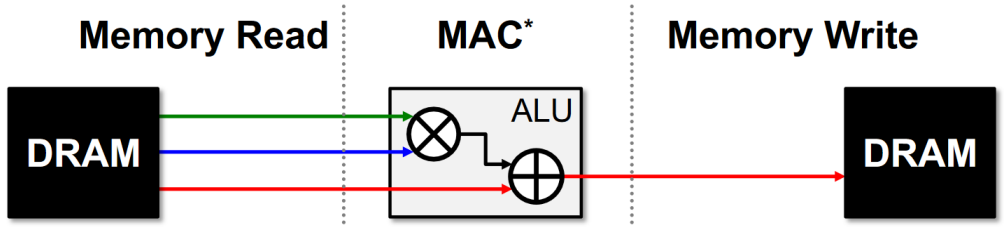
- **与CPU中的很多负载相比，深度神经网络计算的核心差别有哪些?**
- **是否还有其他？对于硬件的影响又是什么样子的呢?**

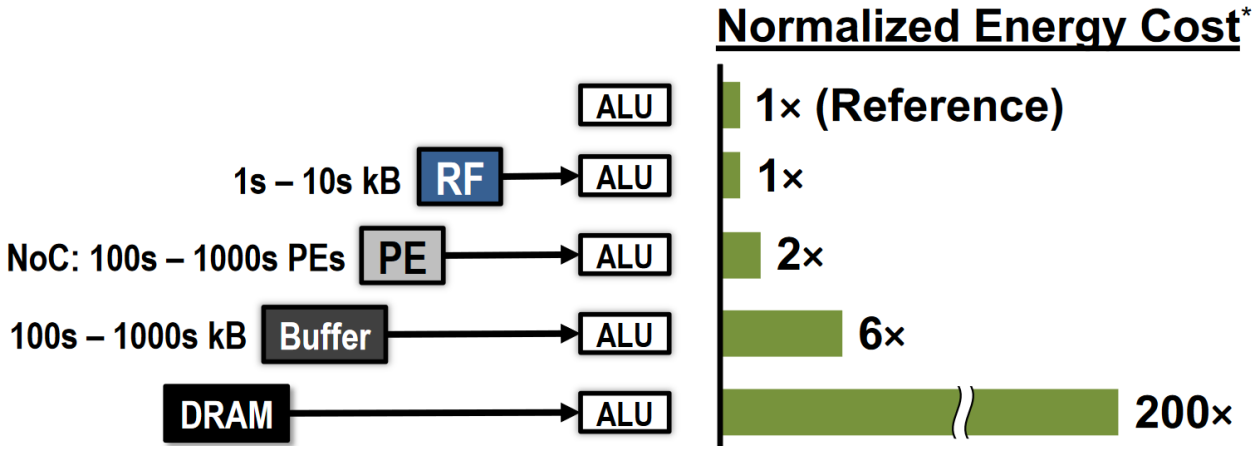
	CPU常见负载	神经网络
操作对象	数据种类多样	张量、向量为主
控制流	控制流复杂灵活	静态固定数据流为主
数据流	复杂不规则	具有较为固定形式

AI加速器架构基础

北京大学 PEKING UNIVERSITY

- 为什么传统CPU进行深度神经网络计算是不高效的？CPU的向量指令呢?
- 需要开发专用AI加速器，提升神经网络的计算效率

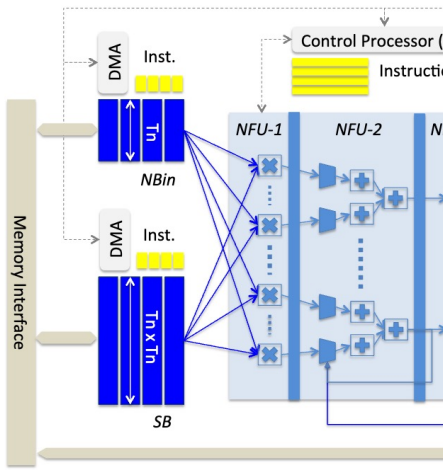




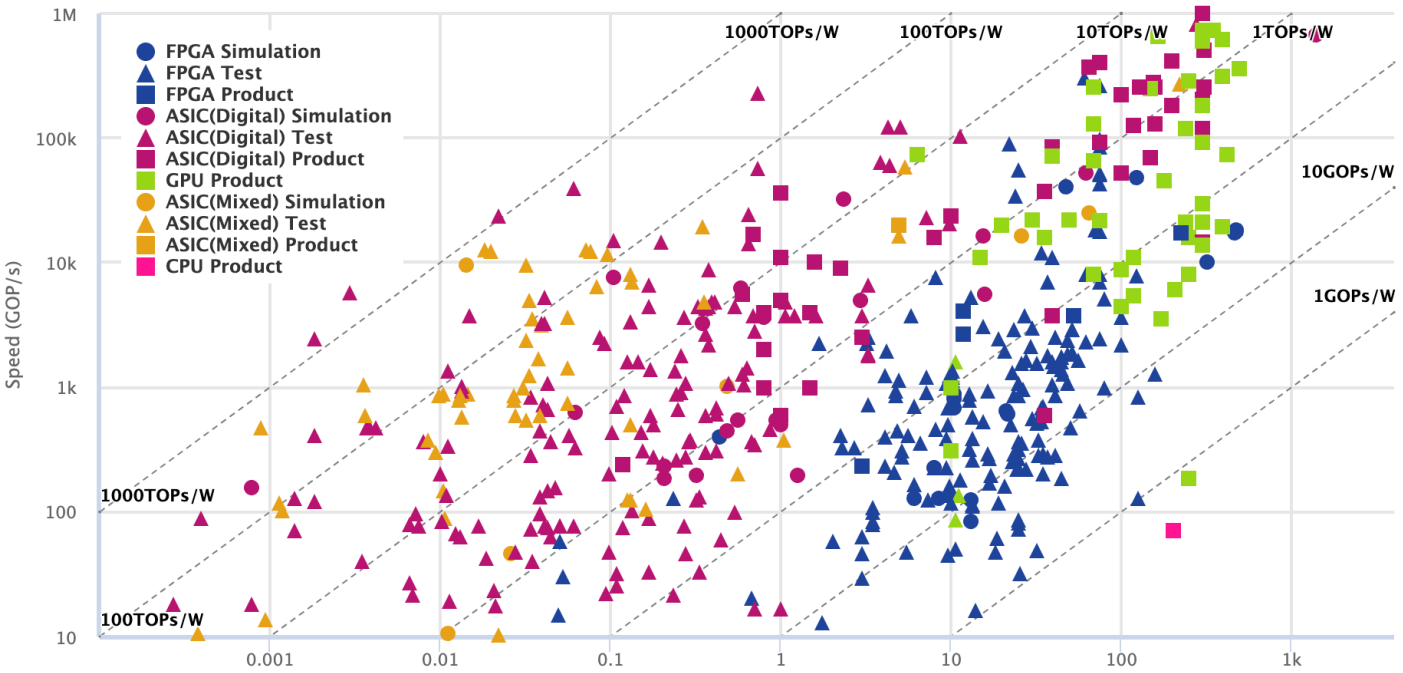
AI加速器架构基础

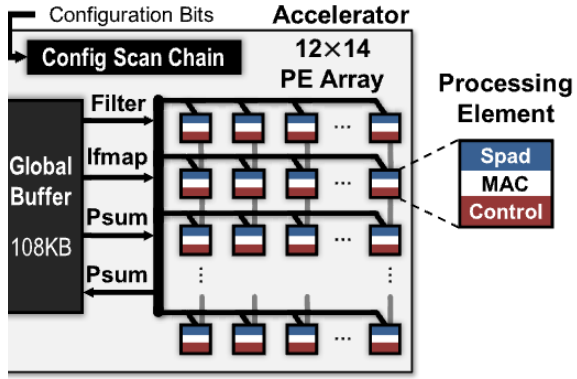
北京大学 PEKING UNIVERSITY

- 在深度神经网络的计算需求驱动下，学术界和业界都在专用AI加速器领域开展了广泛的研究



DianNao



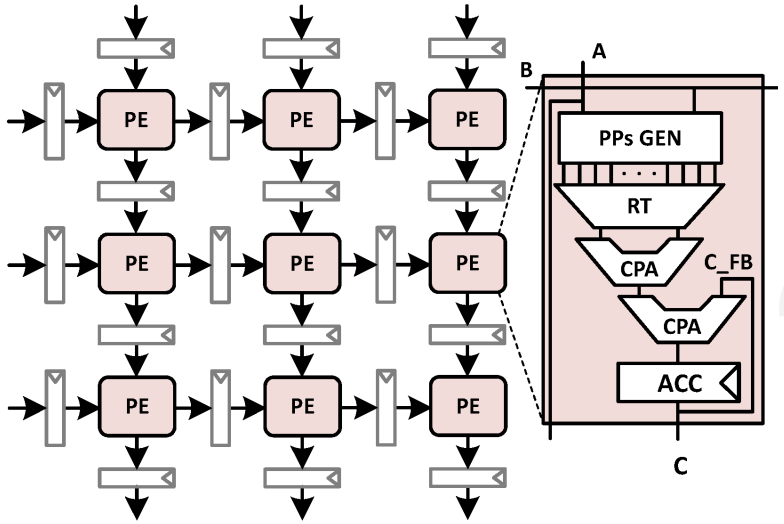


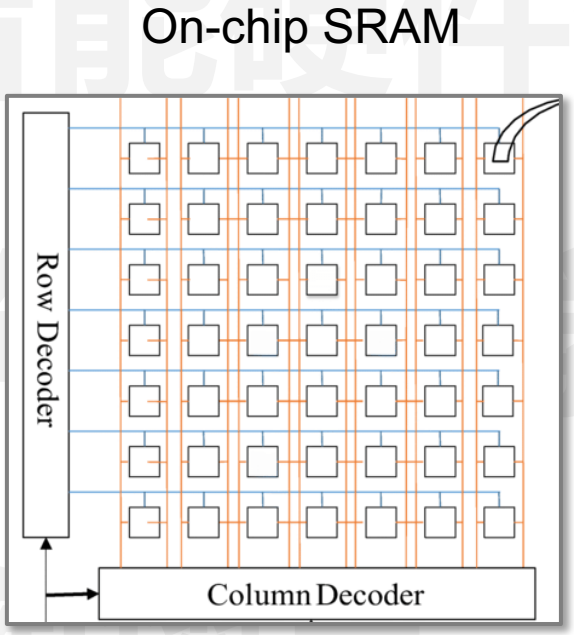
yeriss

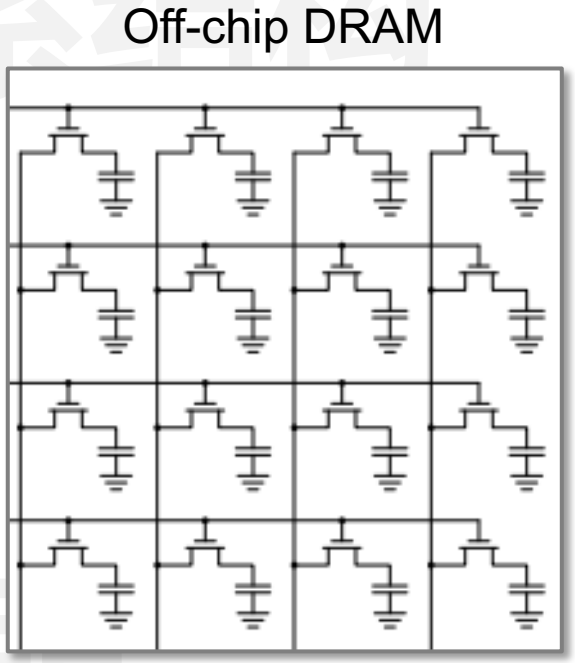
Meta Google amazon TESLA 理想 Alibaba

AI加速器架构基础

- AI加速器整体架构







计算 **访存** **通信**

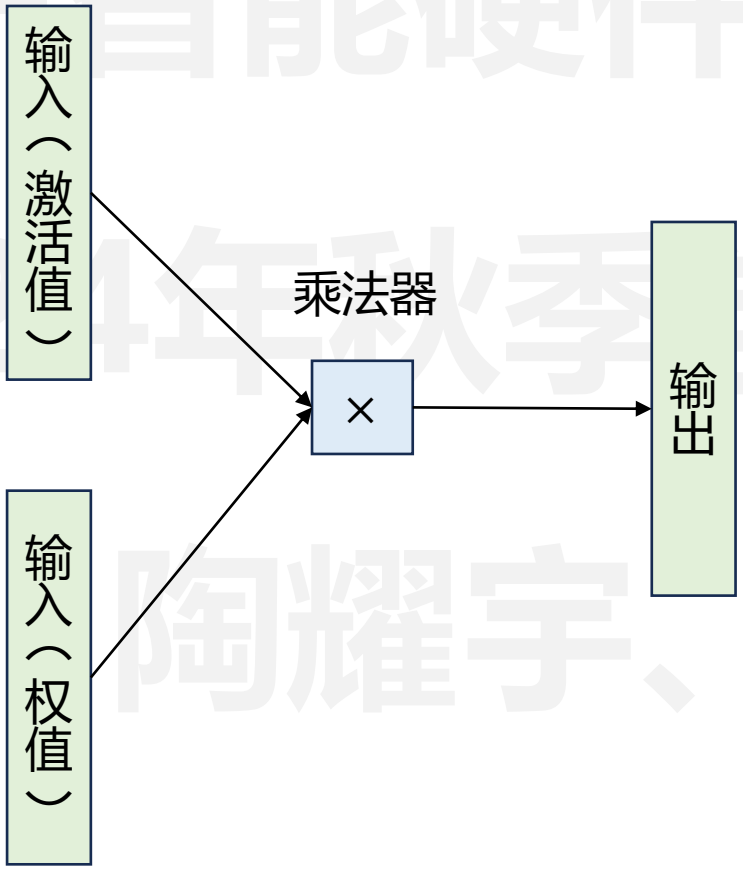
AI加速器架构基础——计算单元

- 计算单元：主要包含面向矩阵、向量和标量三种
  - 分别对应神经网络计算过程中矩阵、向量和标量计算算子
- 针对计算单元，我们经常关心的指标是**计算访存比**，高计算访存比，通常能带来更高的能效
- 计算访存比同时受到**计算负载**和**硬件架构**的影响

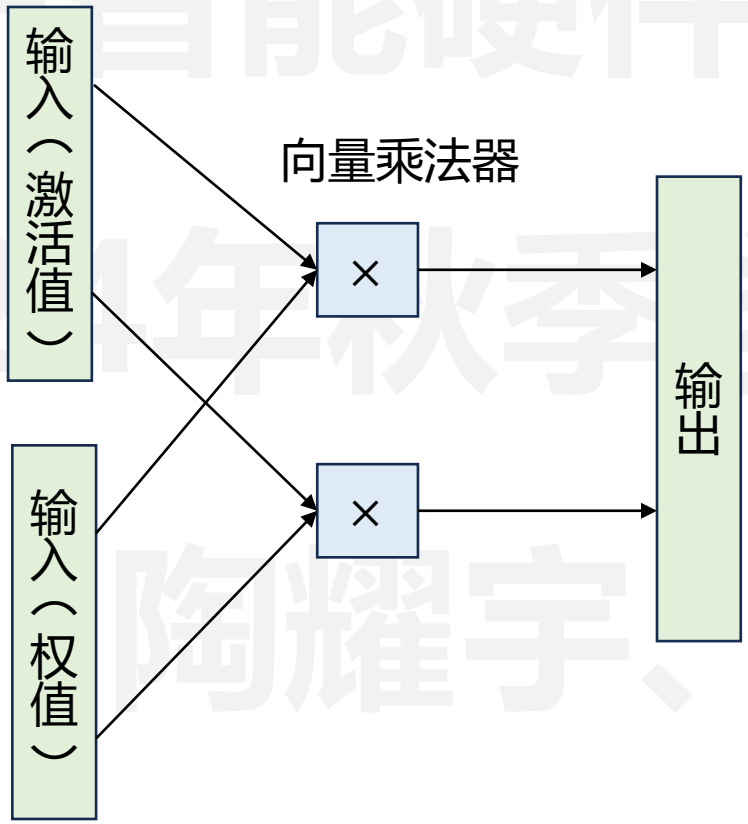
$$\begin{bmatrix} 1 & 0 & 2 \\ 3 & 1 & 0 \\ 5 & -1 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & -1 & 0 \\ 5 & 1 & -1 \\ -2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -1 & 0 \\ 11 & -2 & -1 \\ 1 & -6 & 1 \end{bmatrix}$$

AI加速器架构基础——计算单元

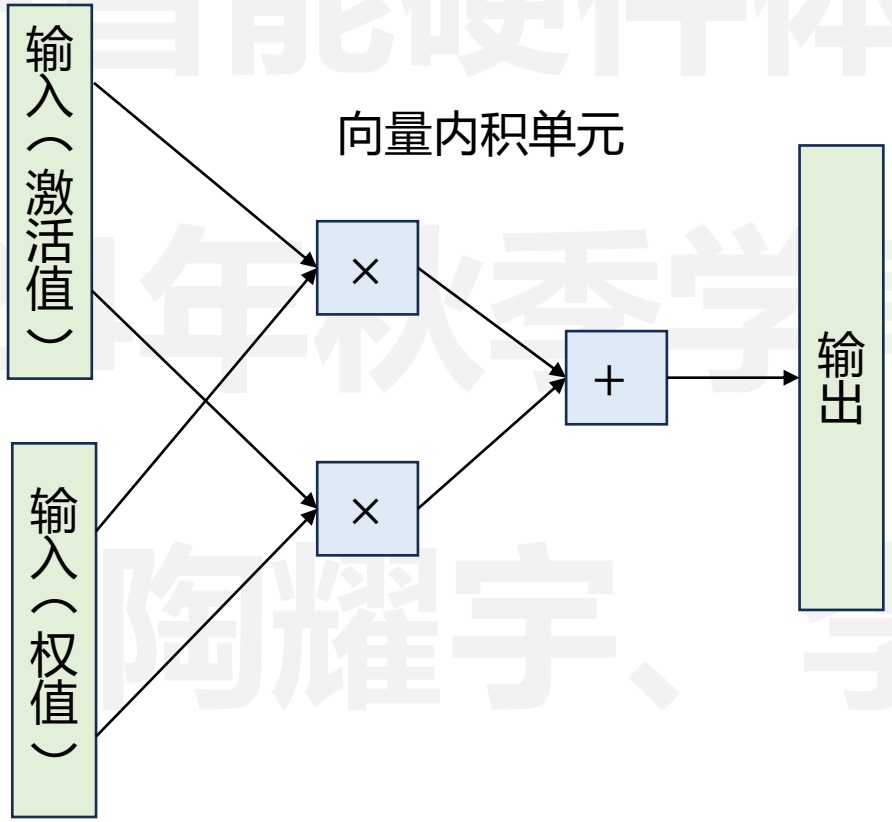
- **矩阵计算单元：通过内积计算，最大化数据复用**



- 矩阵计算单元：通过内积计算，最大化数据复用

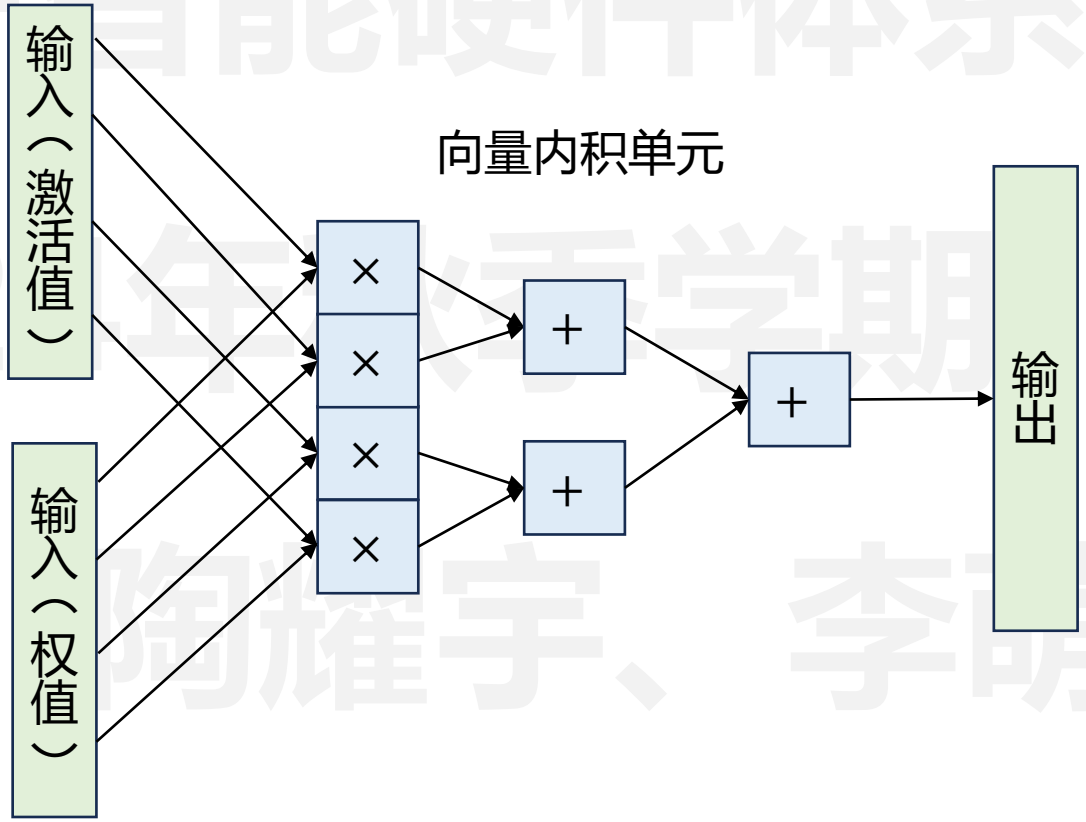


- **矩阵计算单元：通过内积计算，最大化数据复用**



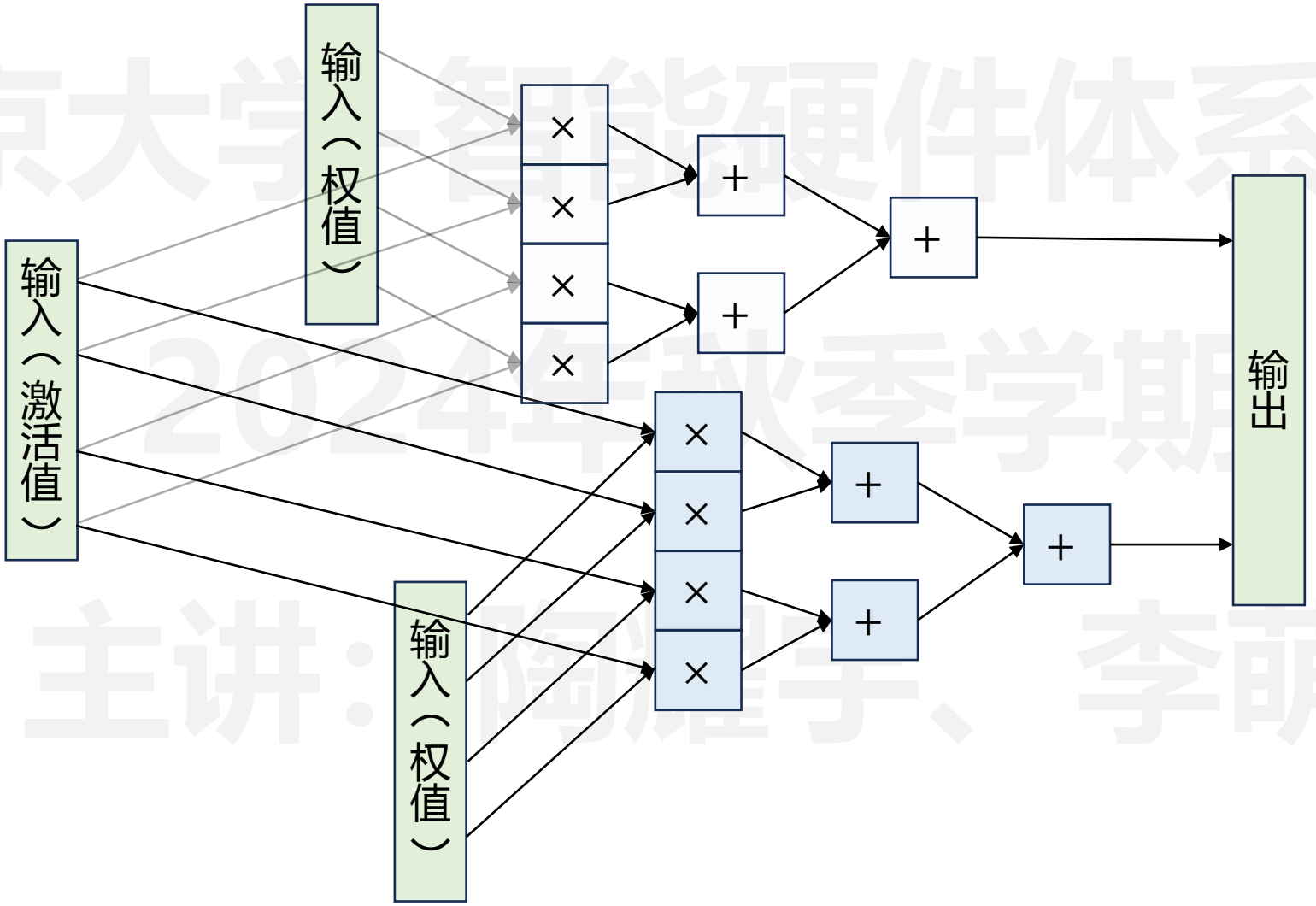
AI加速器架构基础

- **矩阵计算单元：通过内积计算，最大化数据复用**



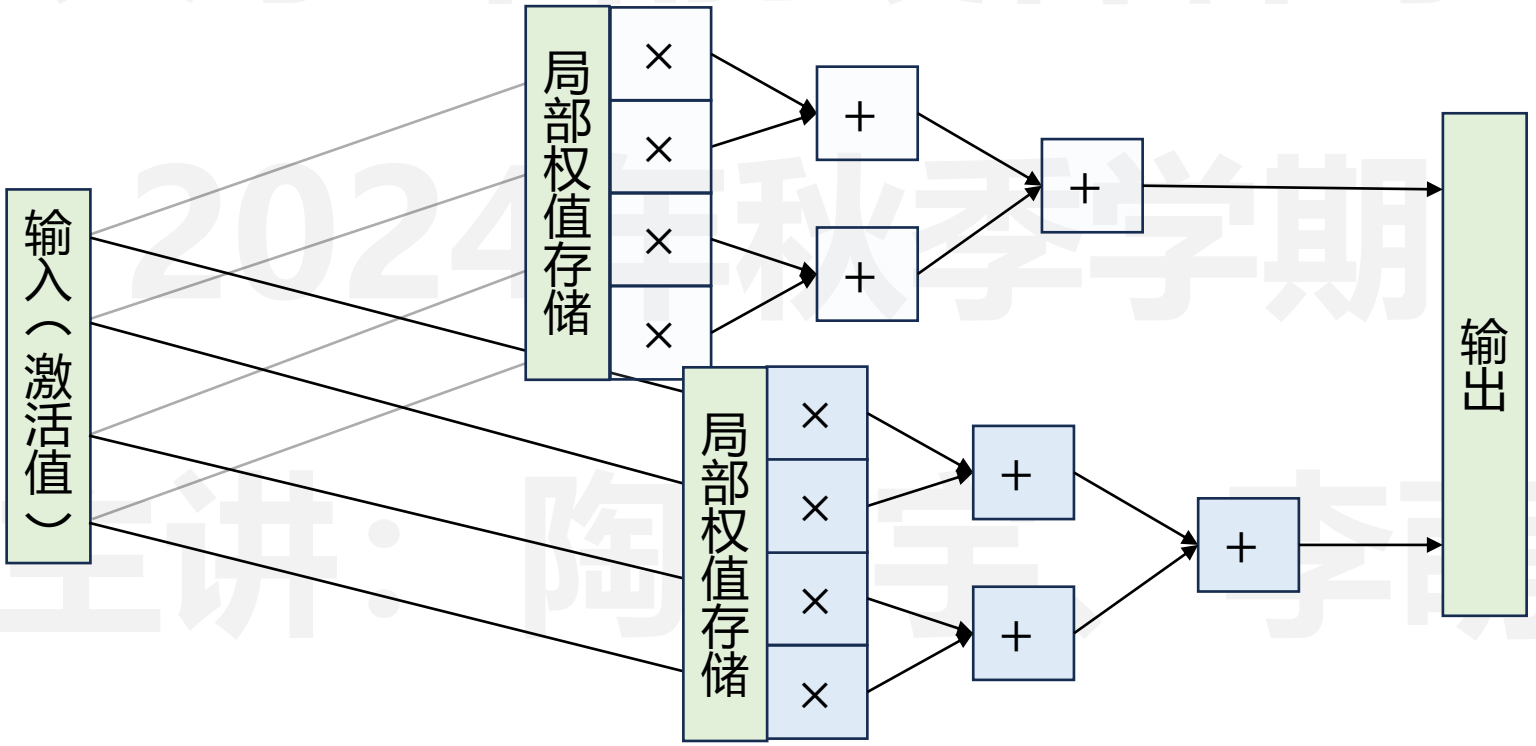
AI加速器架构基础

- 矩阵计算单元：通过内积计算，最大化数据复用



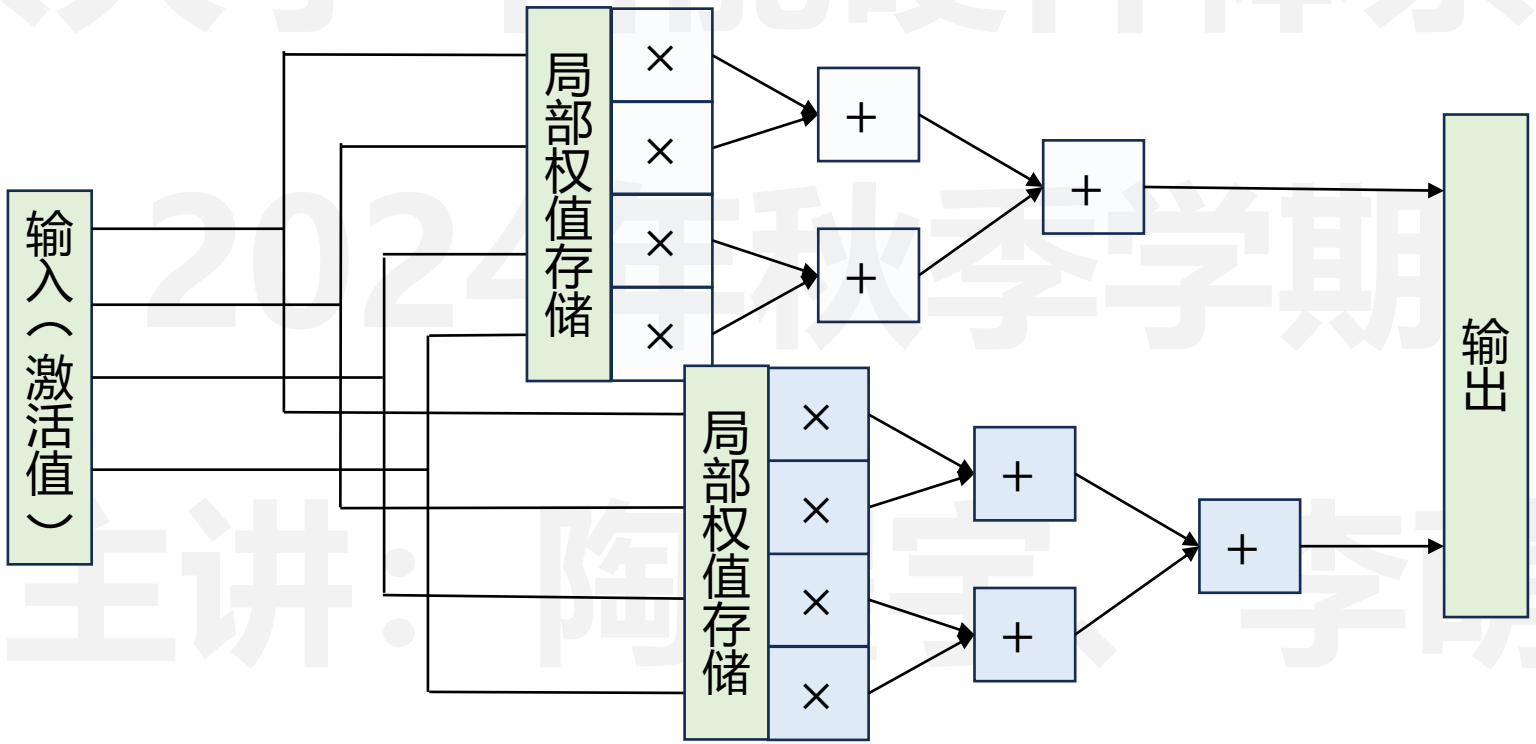
AI加速器架构基础

- **矩阵计算单元：通过内积计算，最大化数据复用**
- **权重采用局部小而快的存储器，直接存储在内积单元附近的电路中**

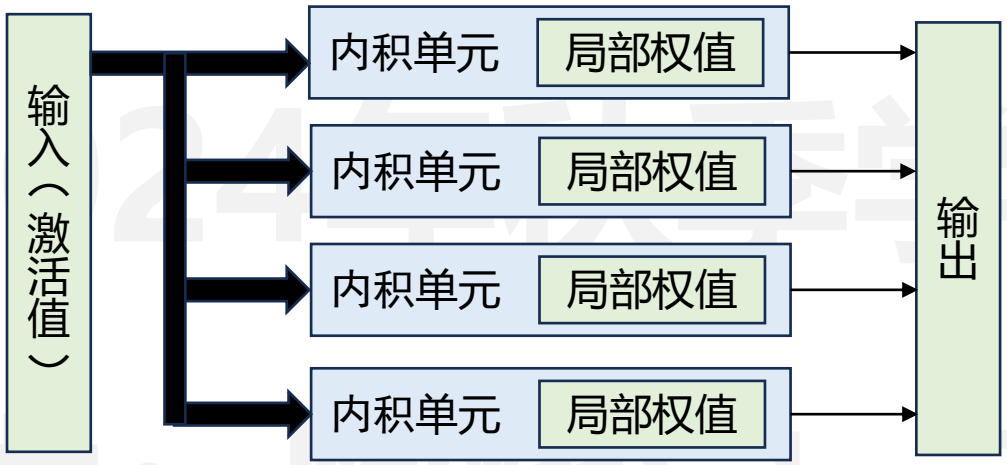


AI加速器架构基础

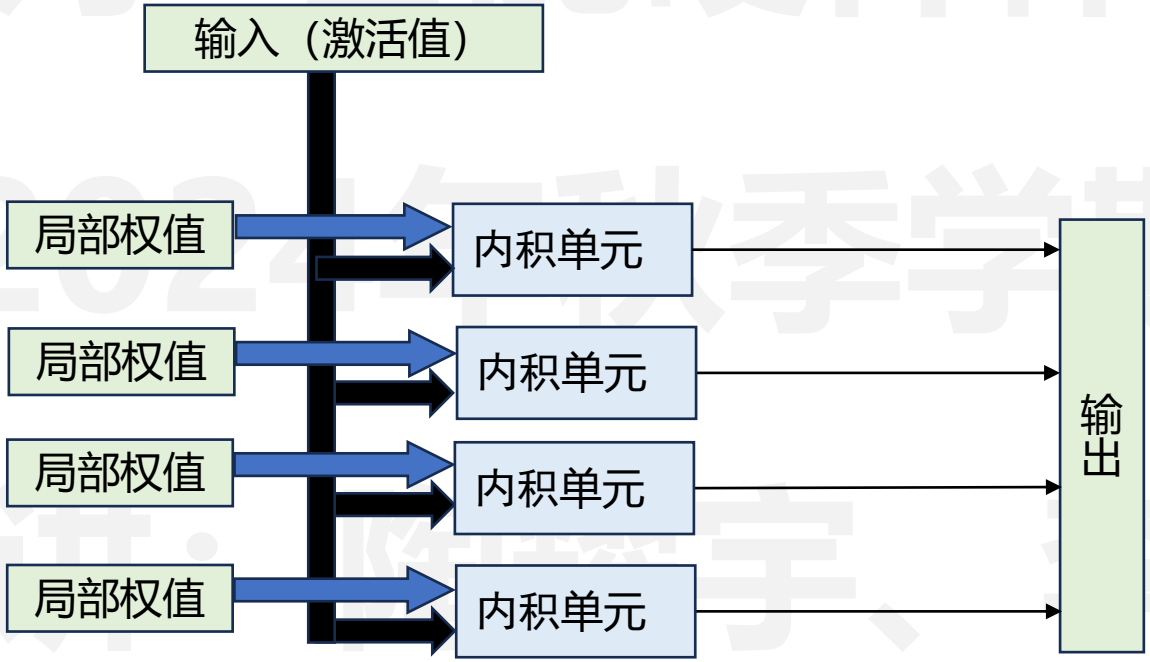
- **矩阵计算单元：通过内积计算，最大化数据复用**
- **所有内积单元共享激活值，采用广播的形式**



- **矩阵计算单元：通过内积计算，最大化数据复用**
- **所有内积单元共享激活值，采用广播的形式**

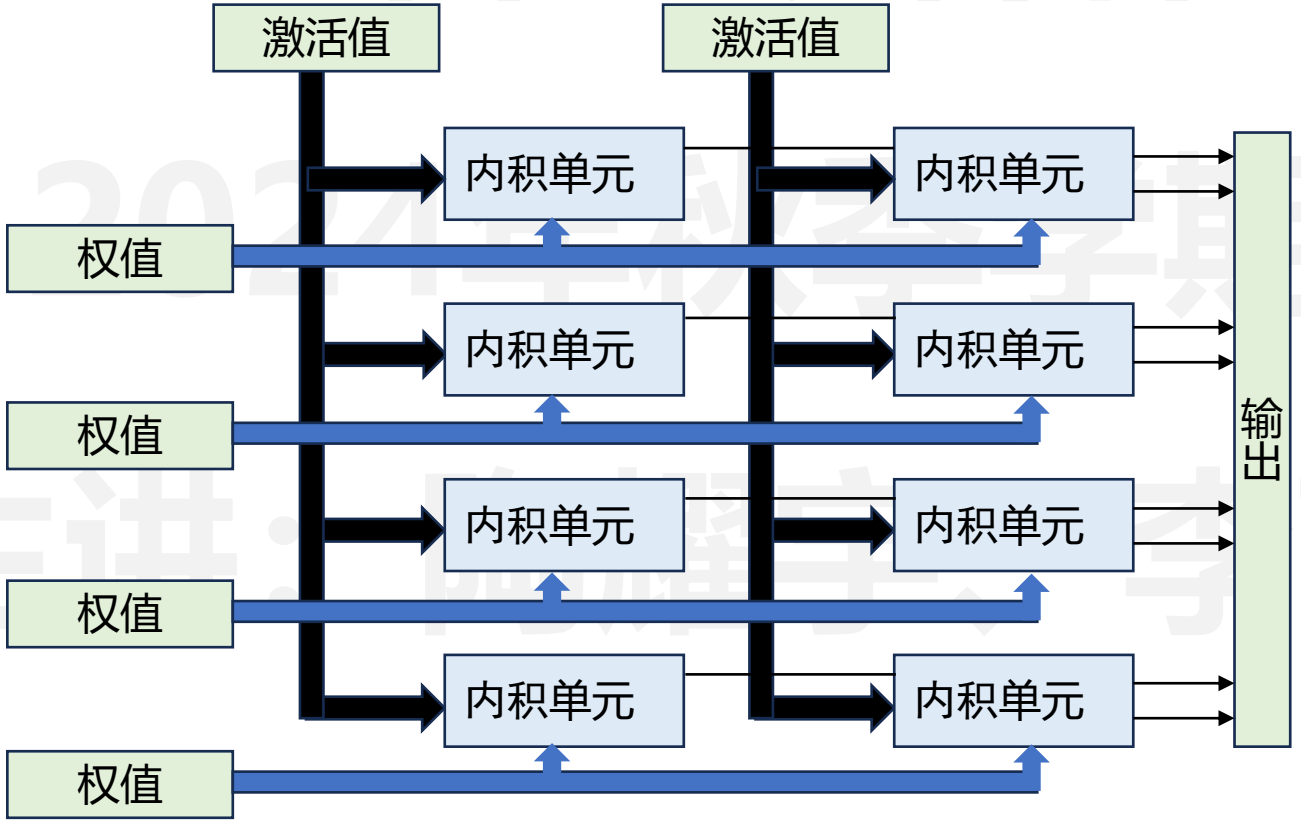


- **矩阵计算单元：通过内积计算，最大化数据复用**
- **如果我们把权值提取出来**



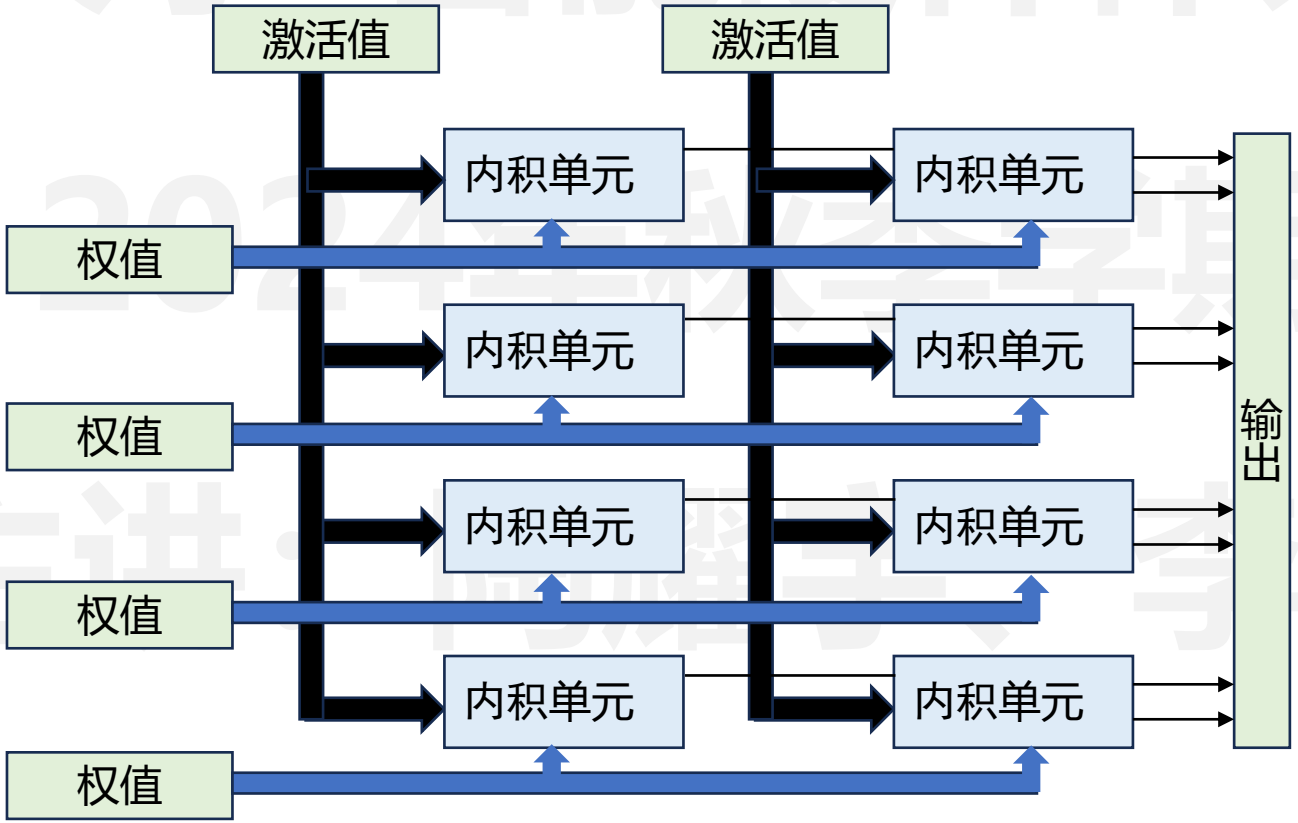
AI加速器架构基础

- **矩阵计算单元：通过内积计算，最大化数据复用**
- **如果我们把权值提取出来，并进一步扩大规模，能够显著增加数据复用**



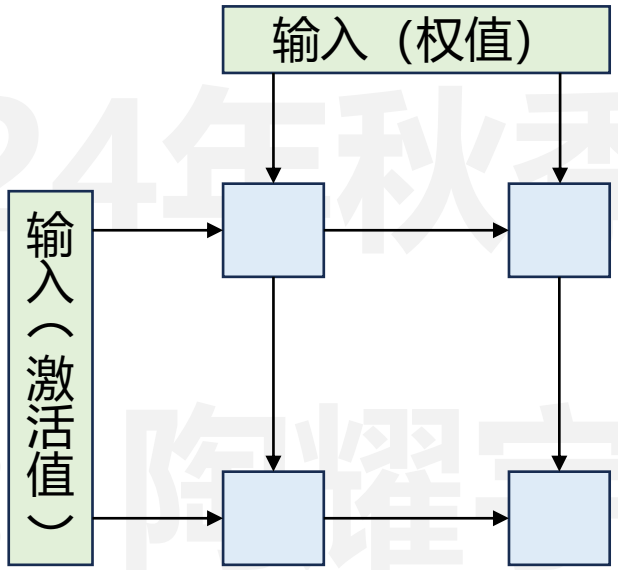
AI加速器架构基础

- 矩阵计算单元：通过内积计算，最大化数据复用
- 矩阵乘法单元在规模大时，能够实现很大的计算访存比
- 但是，面临连线复杂、距离远、扇出多等问题



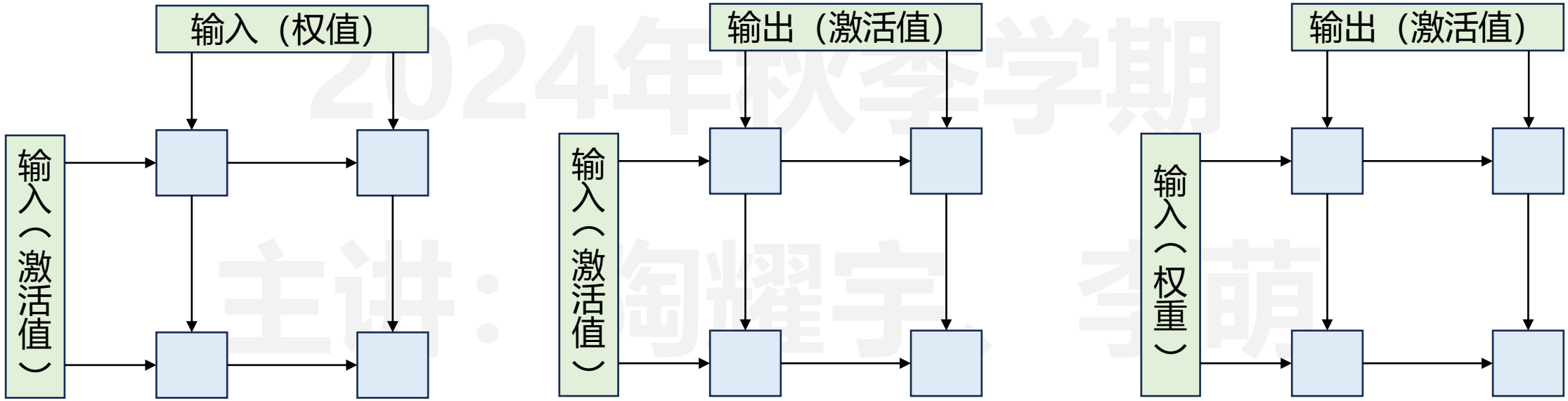
AI加速器架构基础

- 针对连线距离复杂、距离远、扇出多的问题，提出**脉动阵列**进行改进
- 只存在相邻计算单元之间的数据传输，连线短、扇出少
- 但是，脉动阵列的延迟高（需要等待启动/排空），专用性更强，难以改造支持其他功能



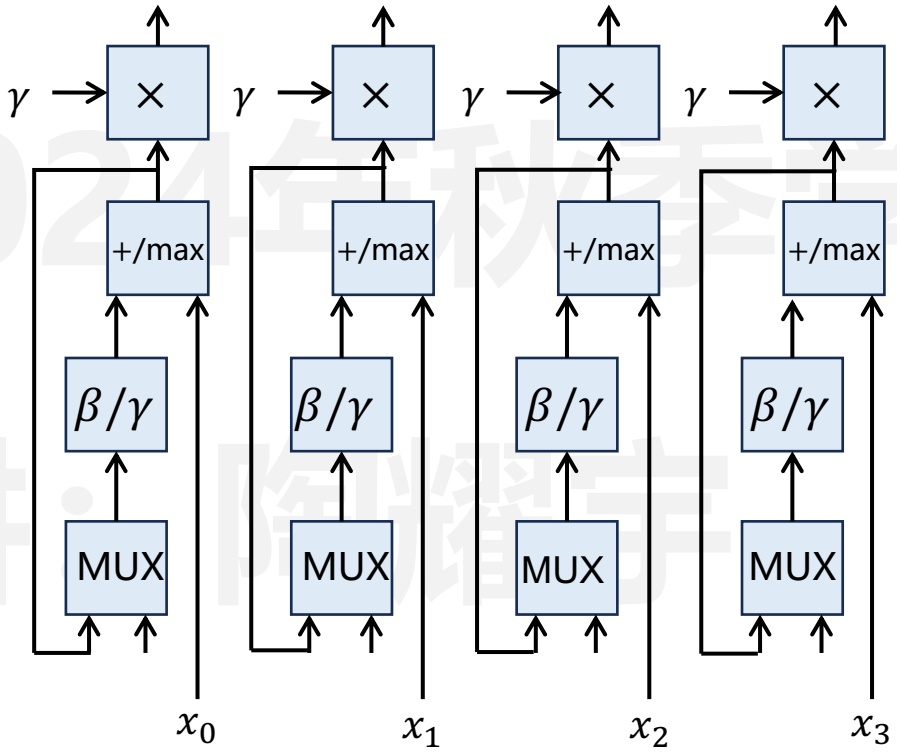
AI加速器架构基础——张量单元

- 矩阵乘法阵列设计核心思想：设计数据流架构，利用计算模式本身的数据复用，提升计算效率
- 针对数据流架构，核心是明确哪些数据是**移动**的，哪些数据是**固定不动**
- **典型数据流**：output stationary、weight stationary、input stationary
  - 不同的数据流，具有不同的神经网络算子适用性



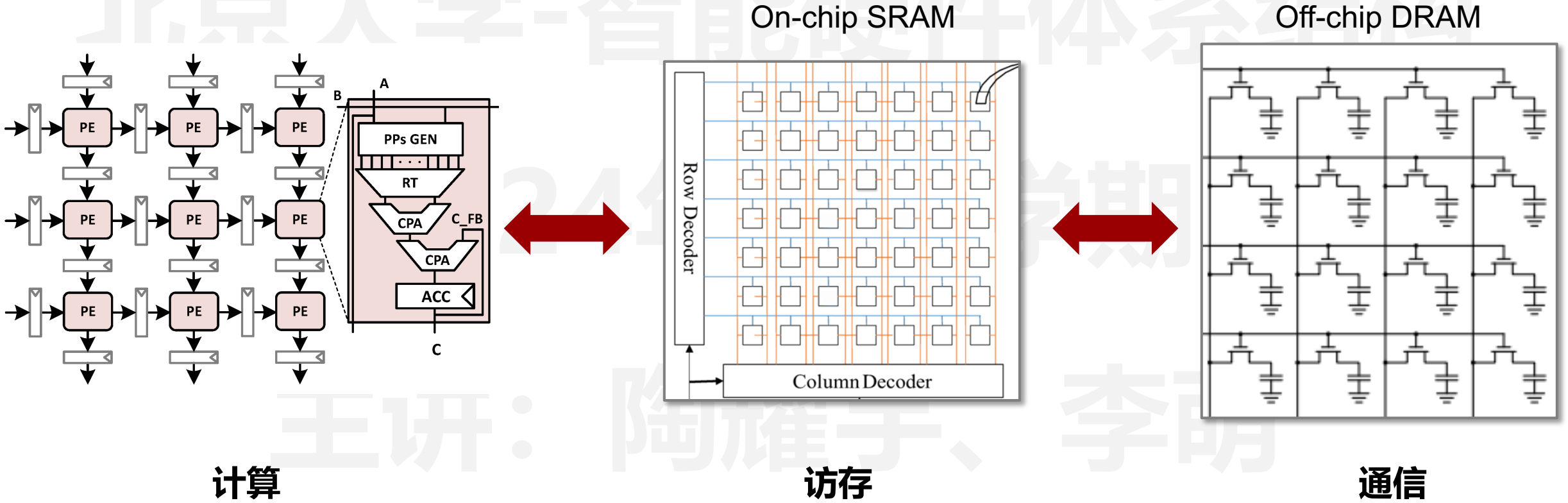
AI加速器架构基础——向量和标量单元

- 向量计算单元主要用于计算池化、归一化、ReLU、Sigmod、Softmax等特殊函数
- 尽管向量运算占神经网络总运算量并不大，但是，如果没有合适的加速硬件，仍然可能**成为瓶颈**
- 思考：向量计算单元和矩阵计算单元都有哪些区别？



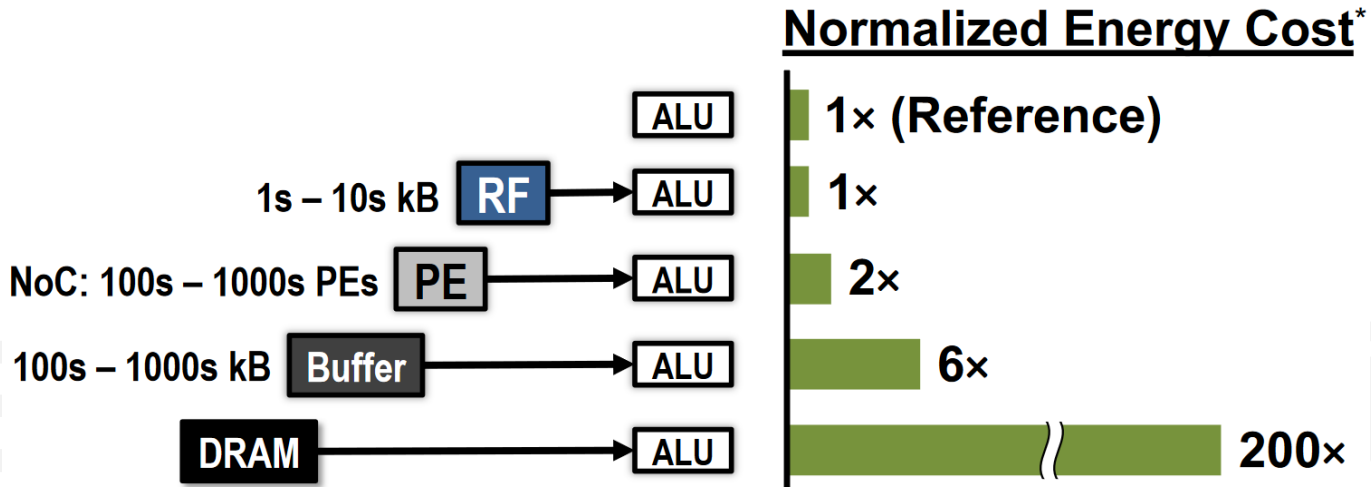
AI加速器架构基础

- AI加速器整体架构



AI加速器架构基础——片上访存

- 针对AI芯片，通过引入片上SRAM，能够有效提升数据复用，降低DRAM访存开销
- 思考：AI芯片的片上存储是否可以延用传统CPU的缓存?
  - 可以但是没必要，由于AI芯片往往采用确定数据流，数据复用明确，无需借助类似CPU的缓存，仅使用便签存储器（Scratch Pad Memory）即可



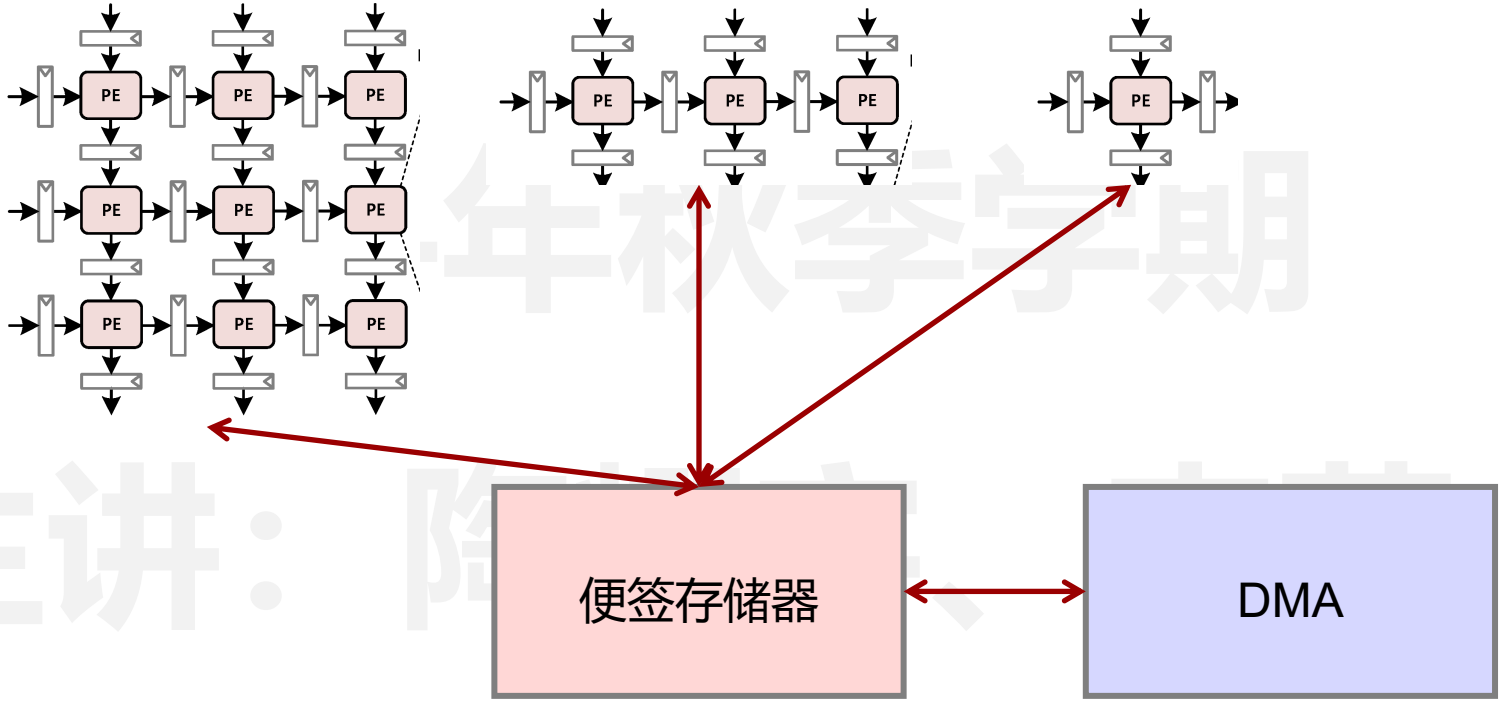
AI加速器架构基础——片上访存

- **便签存储器 VS CPU缓存**
- **同等工艺下，CPU缓存面积约为便签存储器的1.5倍**

便签存储器	CPU缓存
软件控制	硬件控制
软件编程	复杂的缓存管理策略 + 相应硬件支持
访存方式较为固定，多为串行	访存方式多样、随机
访问速度快、功耗低、轻量化	功耗较高
以读取为主	读写比例不确定

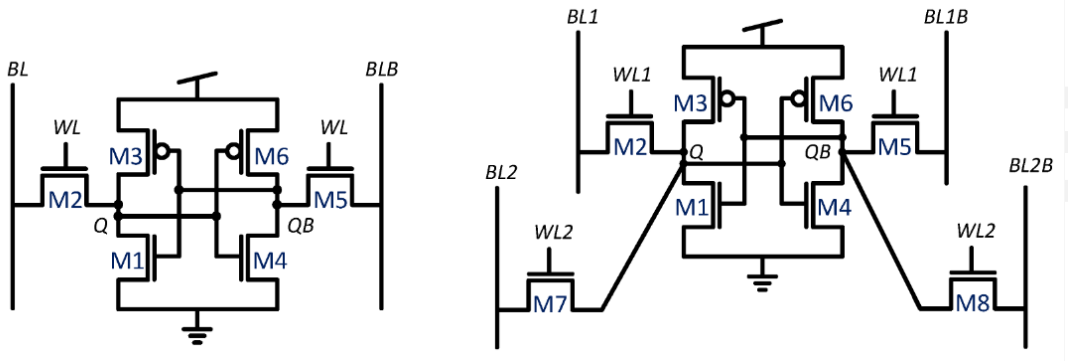
AI加速器架构基础——片上访存

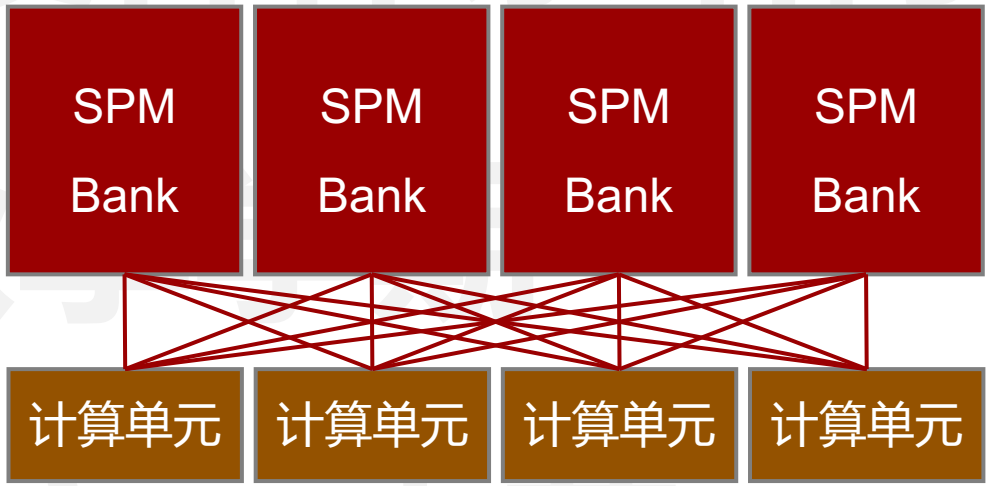
- AI芯片中，便签存储器需要连接矩阵运算单元、向量运算单元、寄存器、DMA/外存等
- 便签存储器是**数据传输的枢纽**，也非常容易造成读写冲突



AI加速器架构基础——片上访存

- AI芯片中，便签存储器需要连接矩阵运算单元、向量运算单元、寄存器、DMA/外存等
- 便签存储器是**数据传输的枢纽**，也非常容易造成读写冲突
- 为了缓解便签存储器的读写冲突，有以下方法





多端口SRAM：

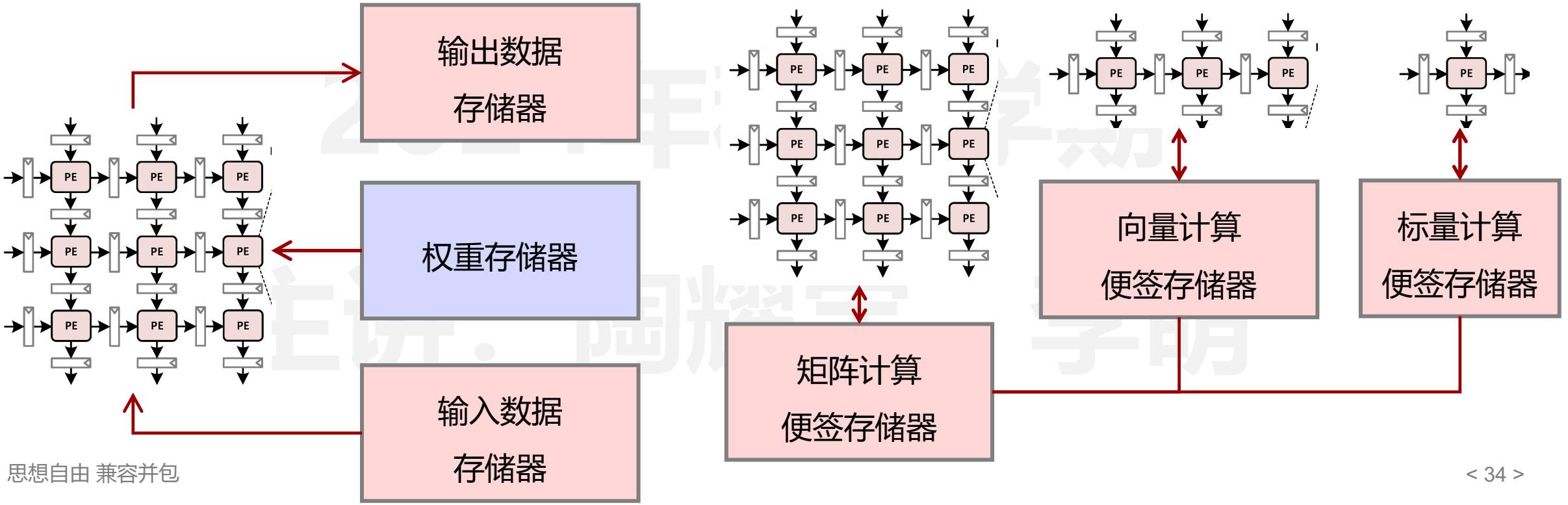
- 增加一个端口，面积增长50%~100%
- 面积进一步影响成本、能效、演示

分组SRAM：

- 仍然存在bank conflict
- 连线复杂度提升~ $O(\text{分组数量}^2)$

AI加速器架构基础——片上访存

- 此外，还可以模仿CPU中数据和指令分开存储（**哈佛架构**）的方式，采用**分离式**便签存储器
- 按照**数据**划分，例如输入数据/输出数据/权重、输入输出数据/权重
- 按照**功能单元**划分，例如矩阵/向量/标量
- 按照**处理阶段**划分，例如输入数据/累加器



AI加速器架构基础——片上访存

北京大学 PEKING UNIVERSITY

- 分离式便签存储器对于数据进行了分流，从而提高了处理效率
  - 对于不同便签存储器的使用方式进行了约束，损失了架构的通用性

“Computer architecture is all about making trade-offs”

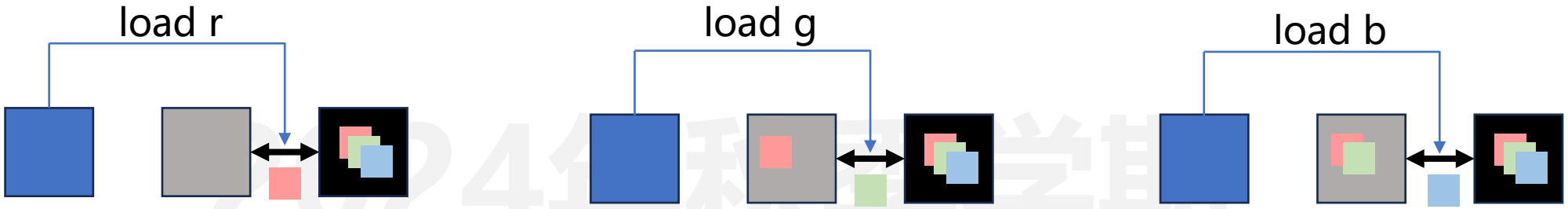
AI加速器架构基础——外部存储器访问

- AI芯片计算过程中，往往需要访问外部存储器（主要为DRAM）
  - 输入数据（例如ImageNet数据集$224\times224\times3$）的传输
  - 模型权重的传输（ResNet50模型参数为25.6M，难以完全存储于片上）
  - 神经网络中间计算结果
- 对于传统CPU，需要执行30万条load/store指令，才能完成ImageNet数据读取
- 对于AI处理器，我们希望
  - 1条load指令装载一整张图片，1条指令完成计算，1条store指令送回内存

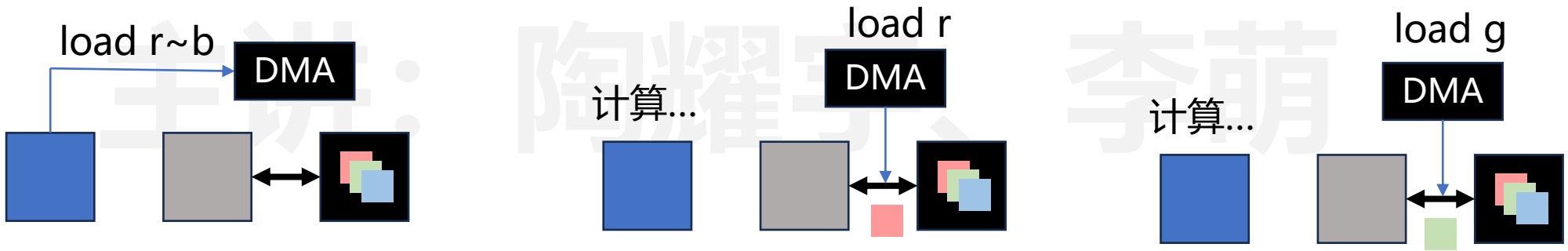
AI加速器架构基础——外部存储器访问

- 如何实现“1条load指令装载整张图像”？

- 处理器控制

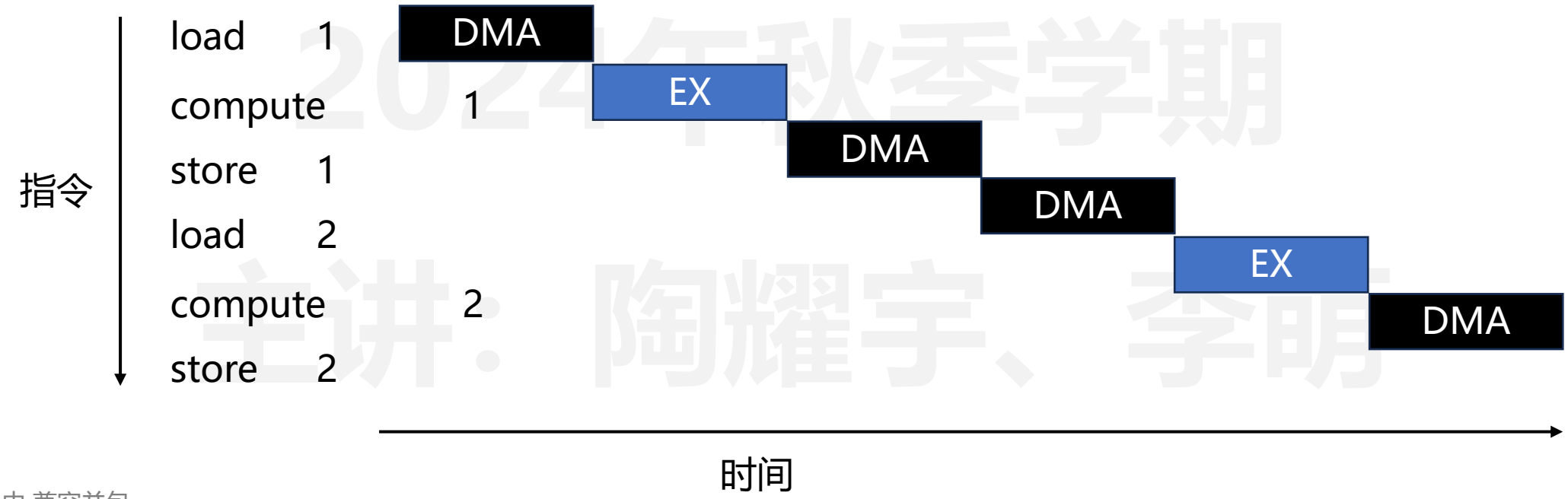


- DMA控制



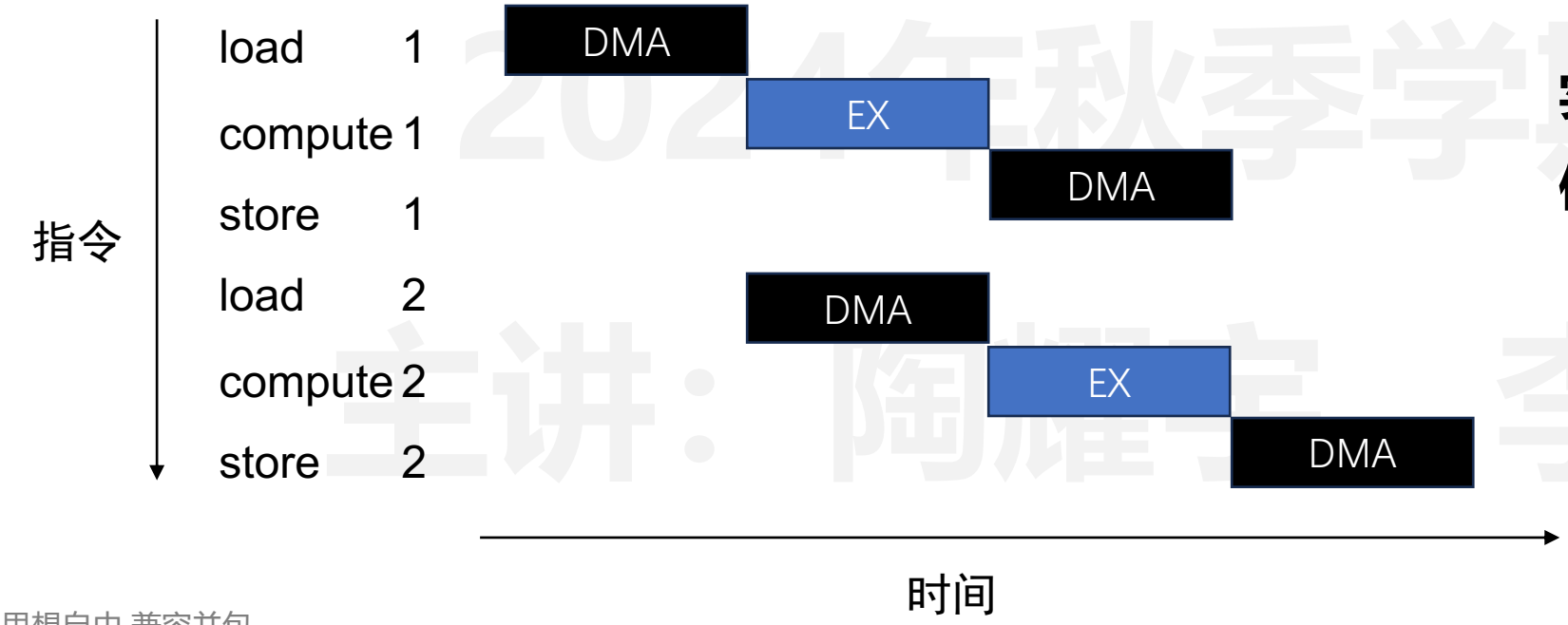
AI加速器架构基础——外部存储器访问

- DMA（direct memory access）：一种计算机系统中的数据传输机制，允许硬件子系统在**不经过CPU干预**的情况下，直接与主存进行数据交换
  - 能够显著减轻CPU负担，并且提高数据传输效率
- 针对AI处理器，单次DMA操作可能持续数百到数十万个周期



AI加速器架构基础——外部存储器访问

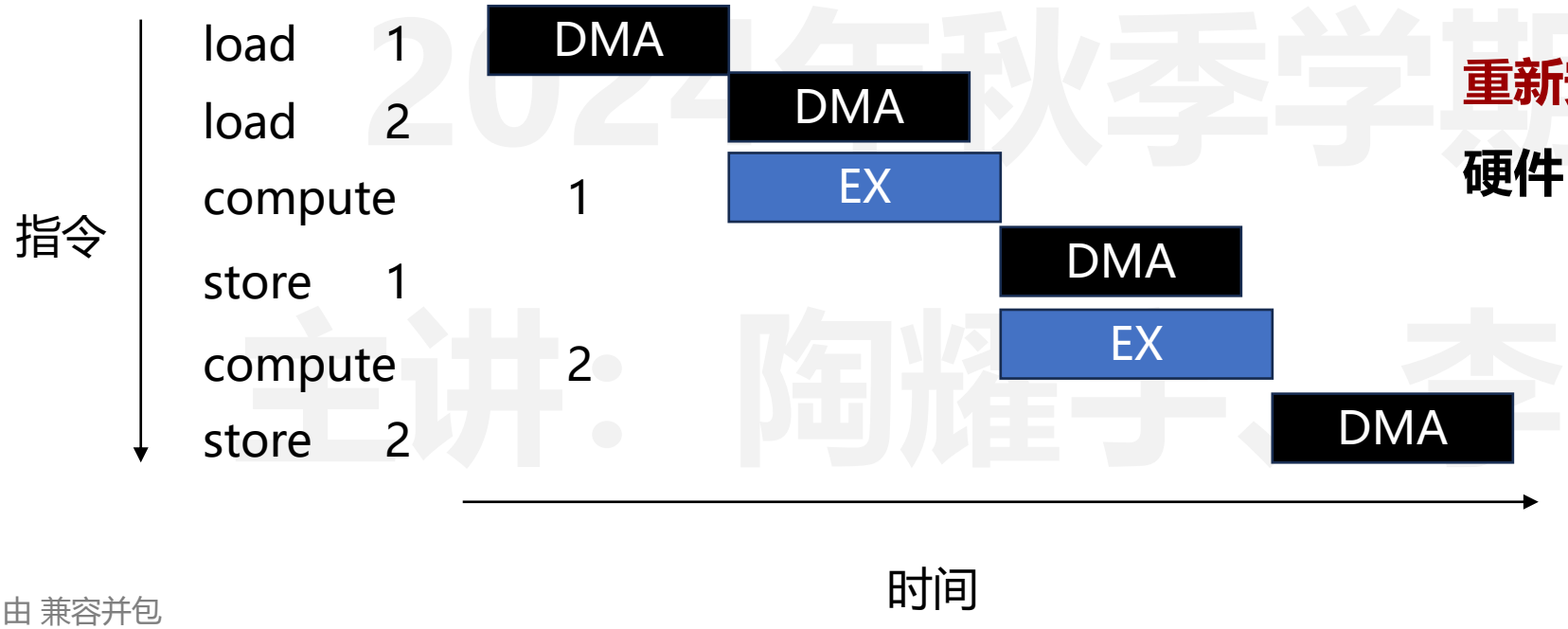
- DMA（direct memory access）：一种计算机系统中的数据传输机制，允许硬件子系统在**不经过CPU**干预的情况下，直接与主存进行数据交换
  - 能够显著减轻CPU负担，并且提高数据传输效率
- 针对AI处理器，单次DMA操作可能持续数百到数十万个周期



实现乱序执行提升不同硬件部分的利用率，有必要吗？有哪些硬件代价？

AI加速器架构基础——外部存储器访问

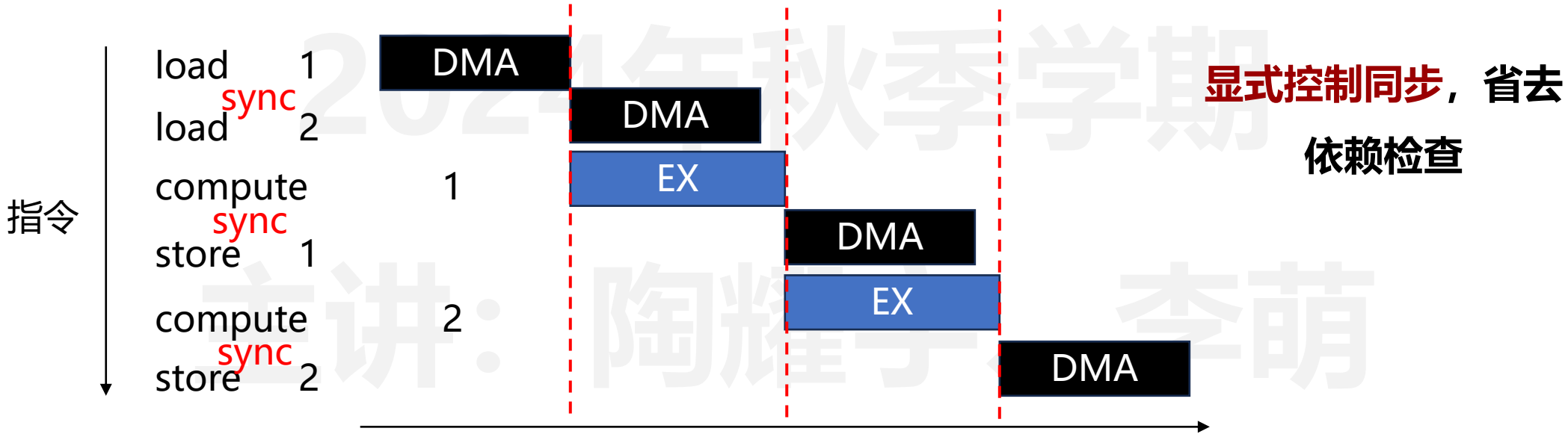
- DMA（direct memory access）：一种计算机系统中的数据传输机制，允许硬件子系统在不经过CPU干预的情况下，直接与主存进行数据交换
  - 能够显著减轻CPU负担，并且提高数据传输效率
- 针对AI处理器，单次DMA操作可能持续数百到数十万个周期



重新安排指令顺序，简化硬件，避免乱序执行的相关硬件开销

AI加速器架构基础——外部存储器访问

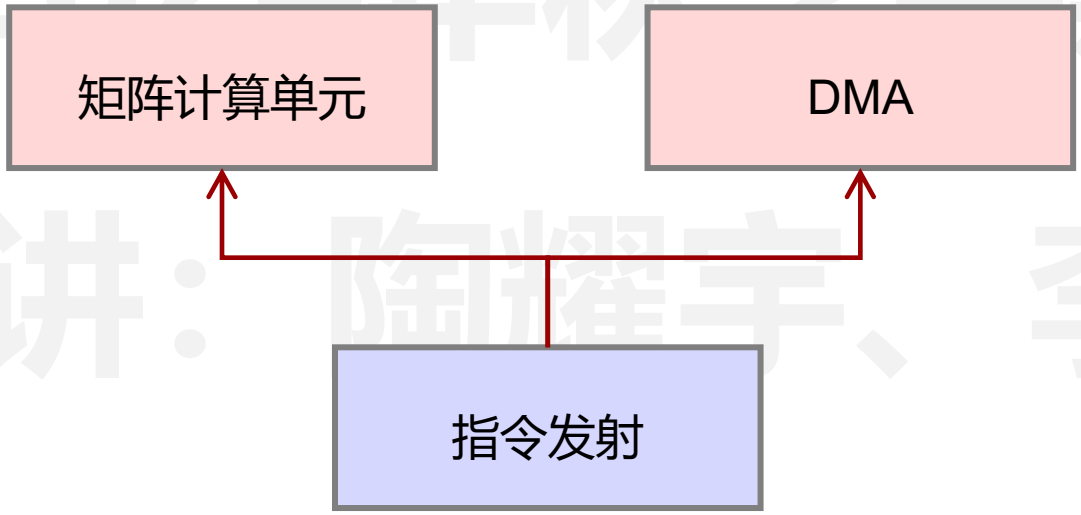
- DMA（direct memory access）：一种计算机系统中的数据传输机制，允许硬件子系统在不经过CPU干预的情况下，直接与主存进行数据交换
  - 能够显著减轻CPU负担，并且提高数据传输效率
- 针对AI处理器，单次DMA操作可能持续数百到数十万个周期



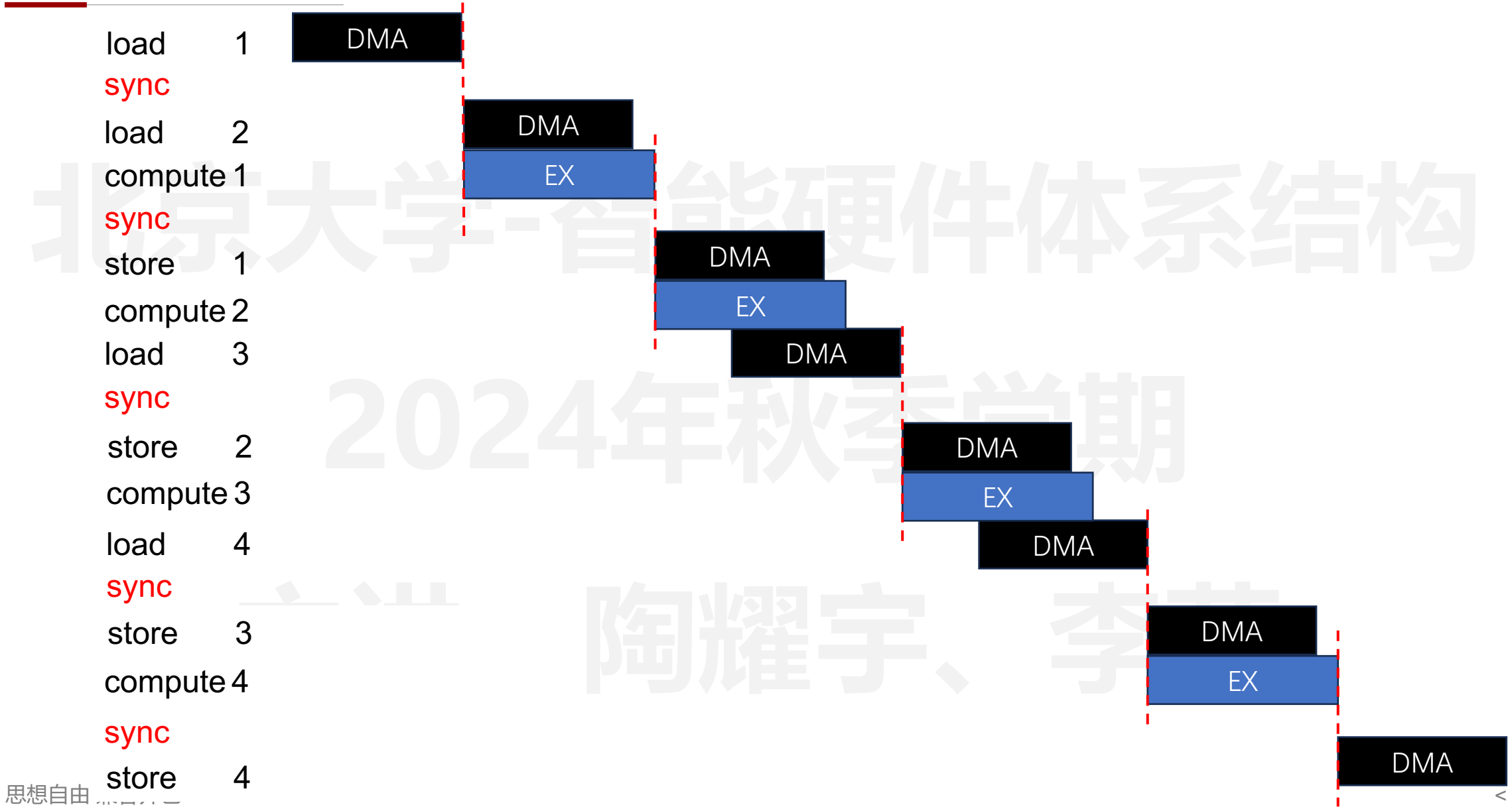
显式控制同步，省去依赖检查

AI加速器架构基础——外部存储器访问

- 软件定义数据流
- 如何实现同步指令（sync）？
  - 计算模块：随时执行收到的指令
  - DMA模块：随时执行收到的指令
  - 指令发射模块：将计算指令发射到计算模块，将访存指令发射到DMA模块，遇到sync时，阻塞直到整个处理器空闲下来



AI加速器架构基础——外部存储器访问

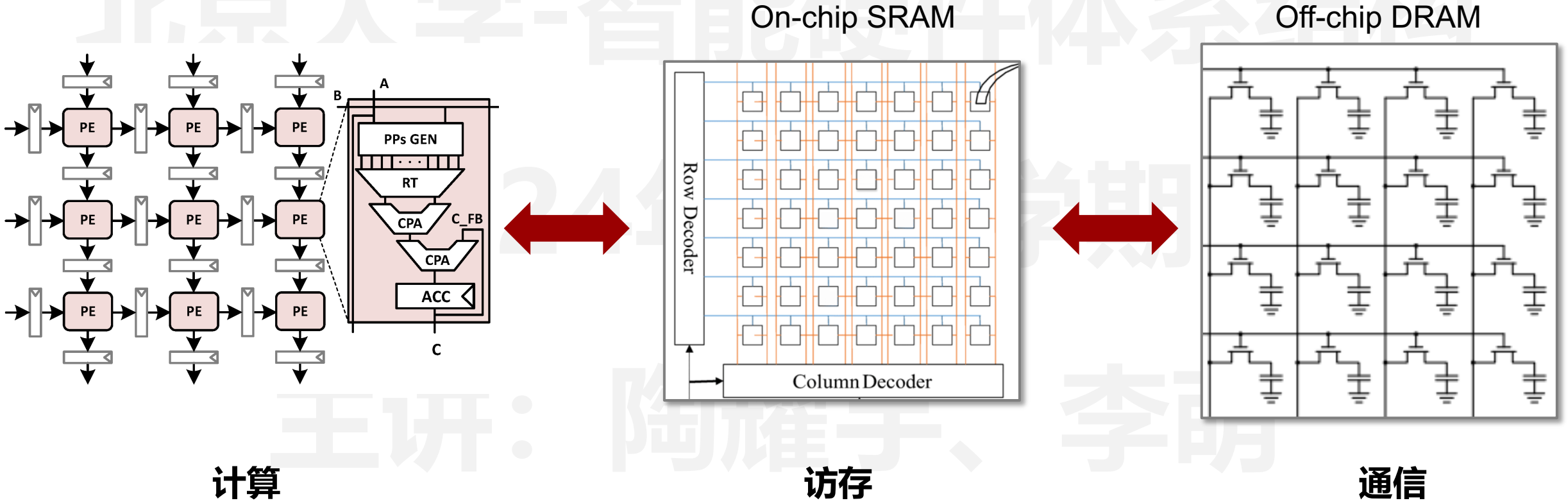


AI加速器架构基础——访存小结

- 便签存储器是数据传输的枢纽
  - 访问便签存储器成为瓶颈
  - 可以通过增加端口、设计分组SRAM的方法，实现更多读取支持（硬件开销增加）
  - 也可以根据算法特征，采用分离式设计
- 针对外部访存，设计DMA，并通过软件流水线，将计算和访存并行起来
  - 指令重新排序，不再需要乱序执行
  - 显示控制同步，省去依赖检查

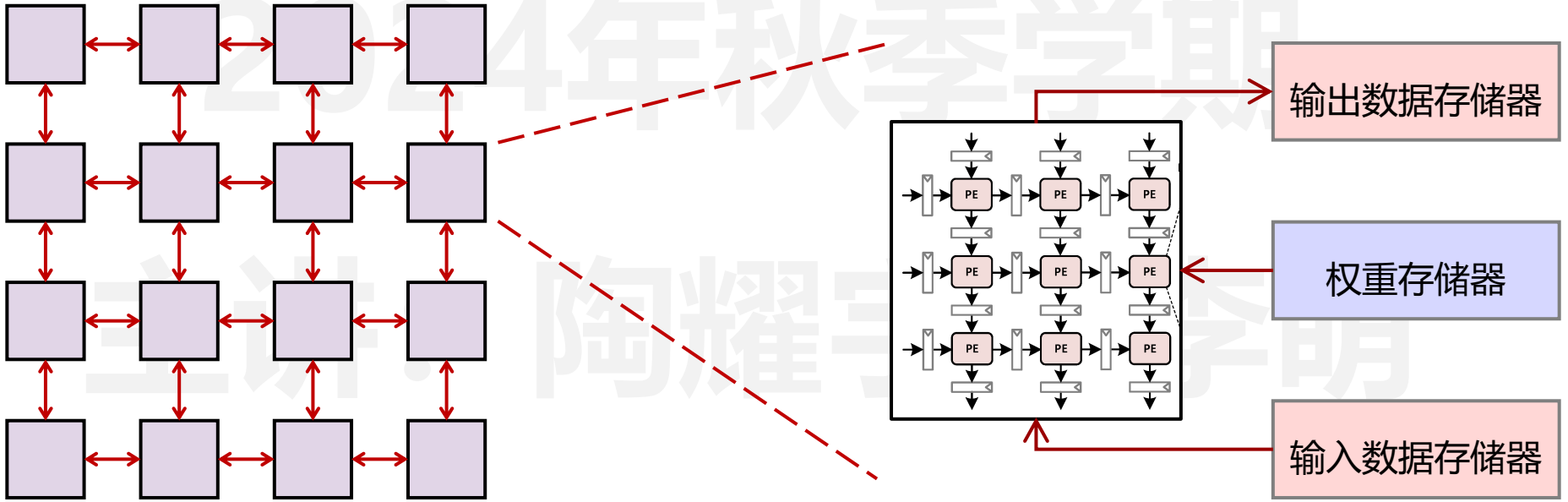
AI加速器架构基础

- AI加速器整体架构



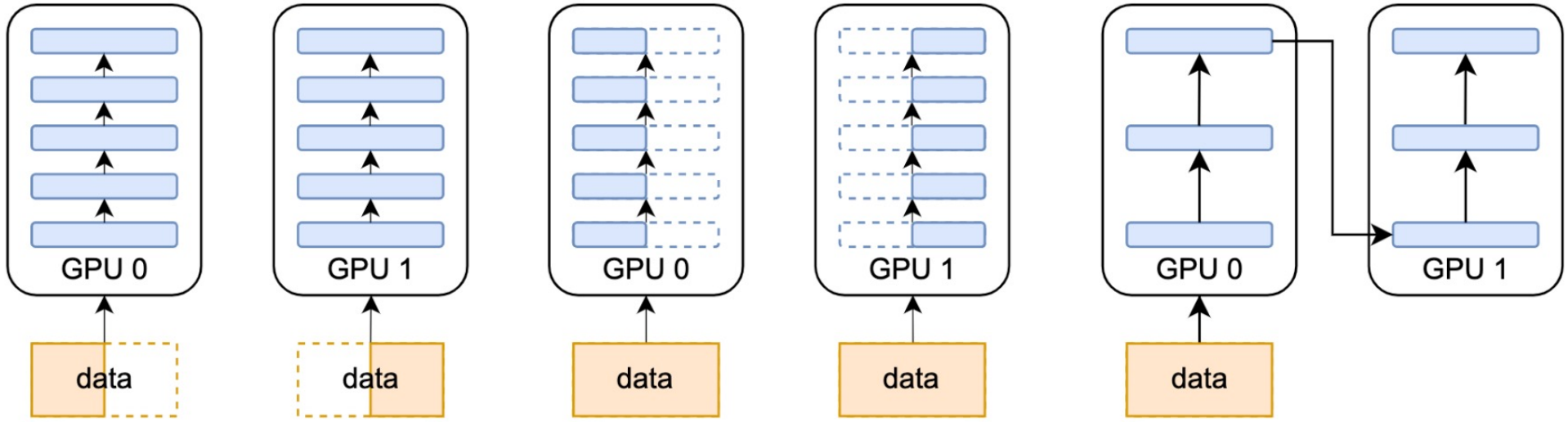
AI加速器架构基础 —— 通信

- 通信即包括计算核与**外存**的通信，也可以包括多核扩展中**计算核间**的通信
- 在上一讲中，我们介绍了不同的片上网络拓扑、路由设计等，本节课，我们重点说明在AI加速器中需要哪些通信算子，以及为什么会有此类需求
- 计算核间的通信需求主要来源于**并行计算**，即需要多计算核共同完成神经网络的计算



典型模型并行计算方式简介

- 面向神经网络，典型的并行计算方式有以下几种

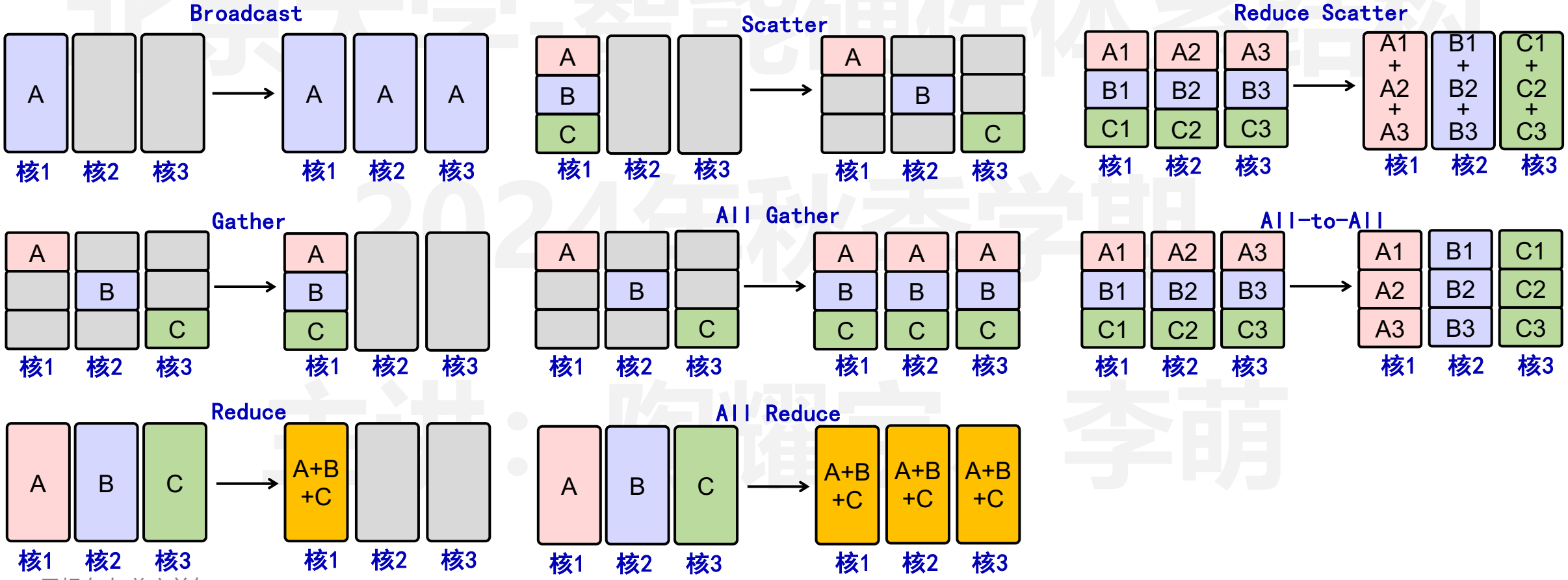


数据并行　　张量并行　　流水线并行

	数据并行	张量并行	流水线并行
部署方式	模型重复存储 各组部署相互独立	单个weight分配至多个芯片 各计算单元结果需要累加	不同weight分配到不同芯片阵列 各计算单元结果存在数据依赖
硬件特点	存算芯片算力较大模型较小	存算芯片算力较小模型较大	常规部署策略
推理特点	Batch 较大/卷积	无固定需求	Batch较大/算力有限

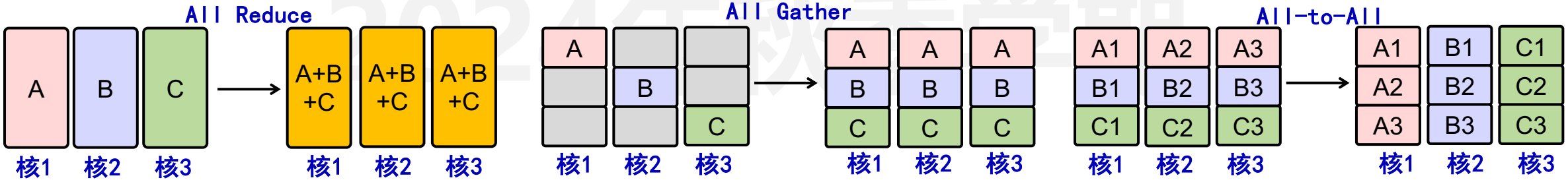
典型集合通信算子

- **集合通信**：并行计算领域中的一种通信操作类型，用于多个并行计算单元（多个处理器、多个计算节点）间的数据交互和同步



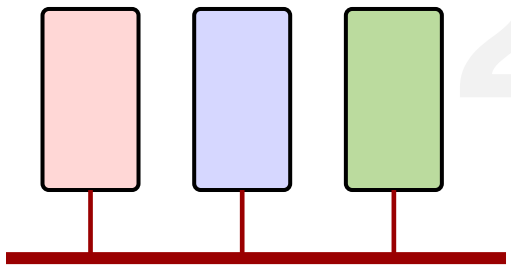
典型集合通信算子

- 大部分的集合通信操作，可以通过二维环（Ring）互连拓扑实现
- 是否说明二维环就足够了呢?

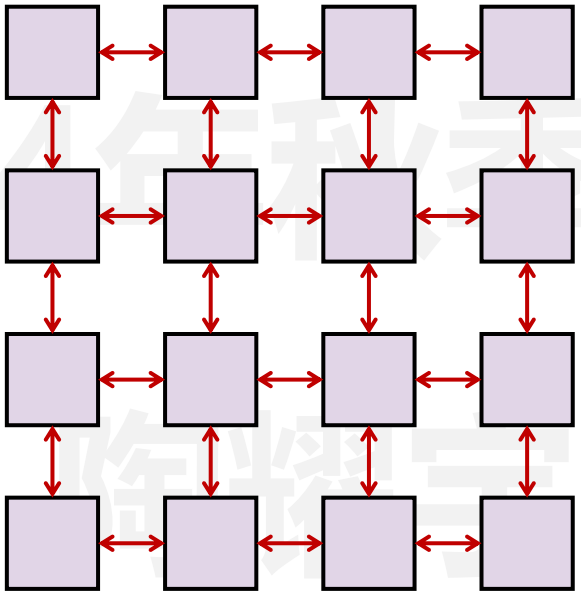


AI加速器架构基础 —— 通信

- 常见的物理链路设计：
  - 总线：常用AXI、PCIe等标准，性能差、成本低
  - 片上网络：常用胖树、二维环、二维网格等拓扑，性能较好、成本可控
  - 交叉开关：性能最佳，但是成本很高，难以拓展



总线



片上网络

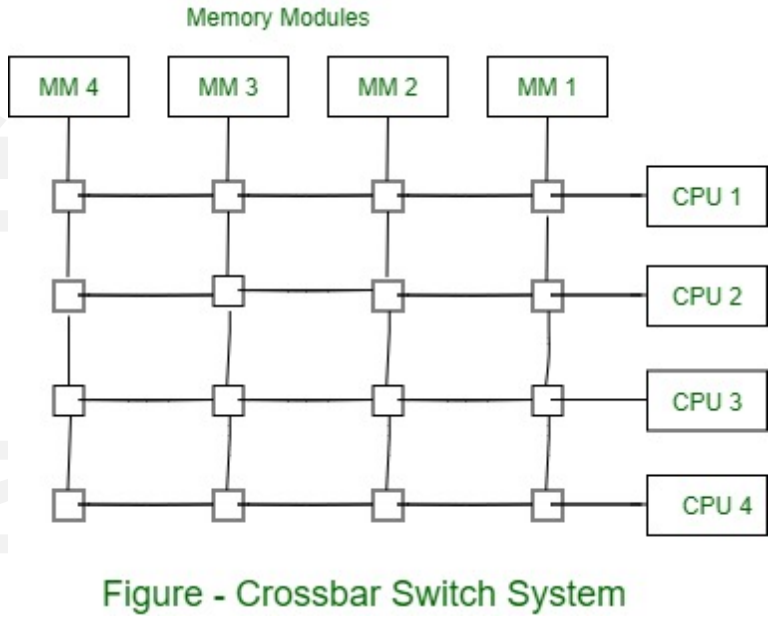


Figure - Crossbar Switch System

交叉开关

通信小结

北京大学 PEKING UNIVERSITY

- 通信结构的设计原则
  - 逻辑上：环状链路足以高效完成通信
  - 物理上：链路设计适当增加冗余，按需配成环路
  - 需要综合考虑性能和成本约束，做出选择