



ZOMI

AI Core
计算模式

Ascend



About

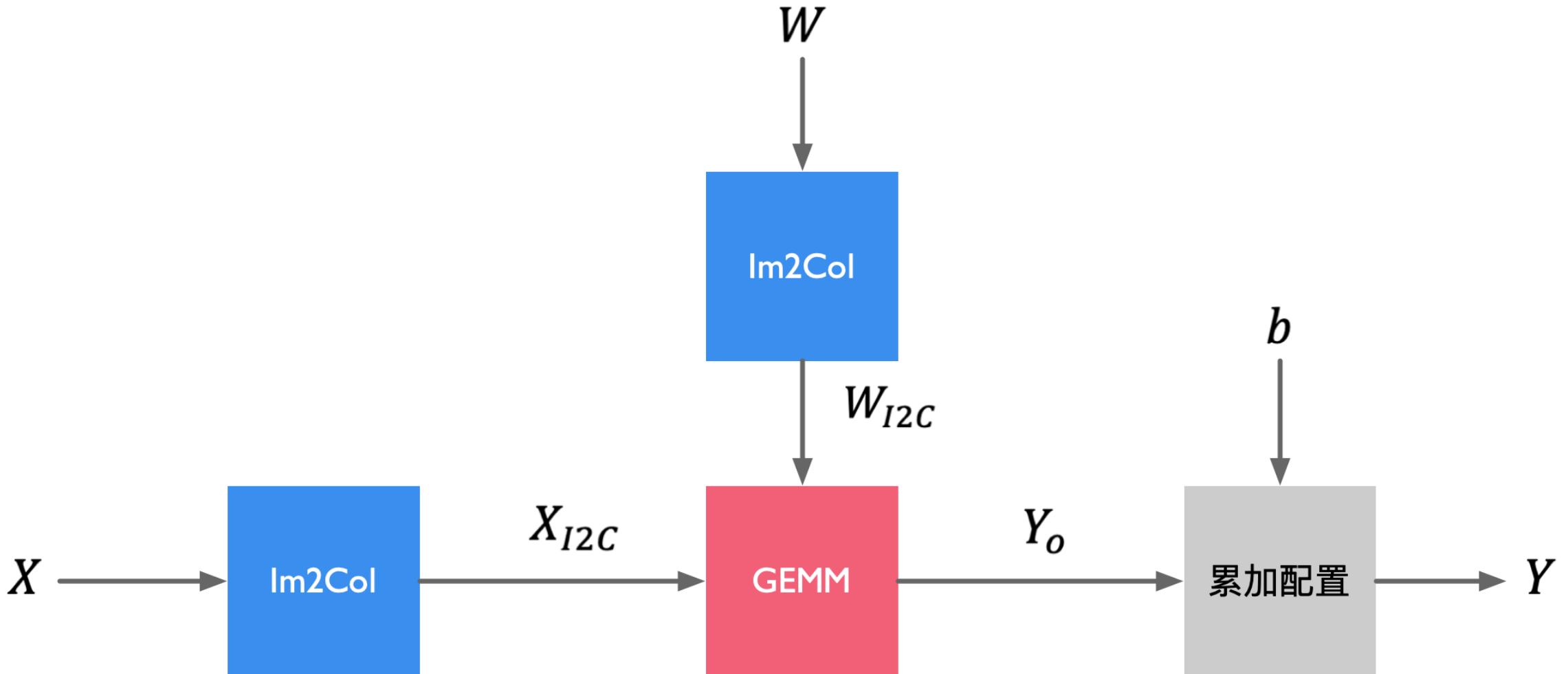
- **昇腾 SOC 架构:** 昇腾 310 芯片 - 昇腾 910 芯片
- **AICore 的灵魂:** 达芬奇架构内部细节
- **AICore 计算模式:** Vector 和 Cube 计算方法
- **服务器爆炸图:** 从芯片到服务器



AI Core

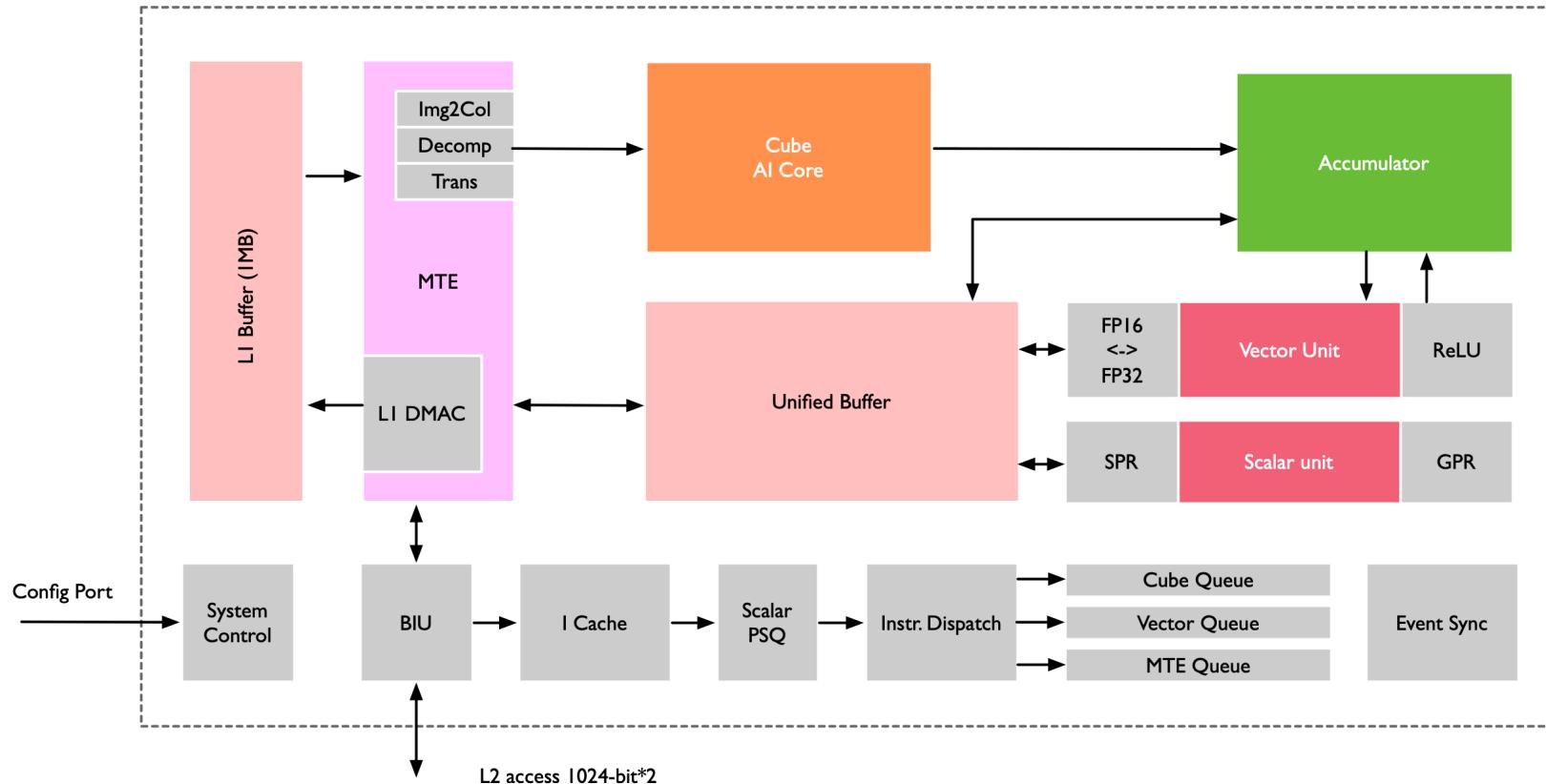
计算数据通路

矩阵乘与卷积计算



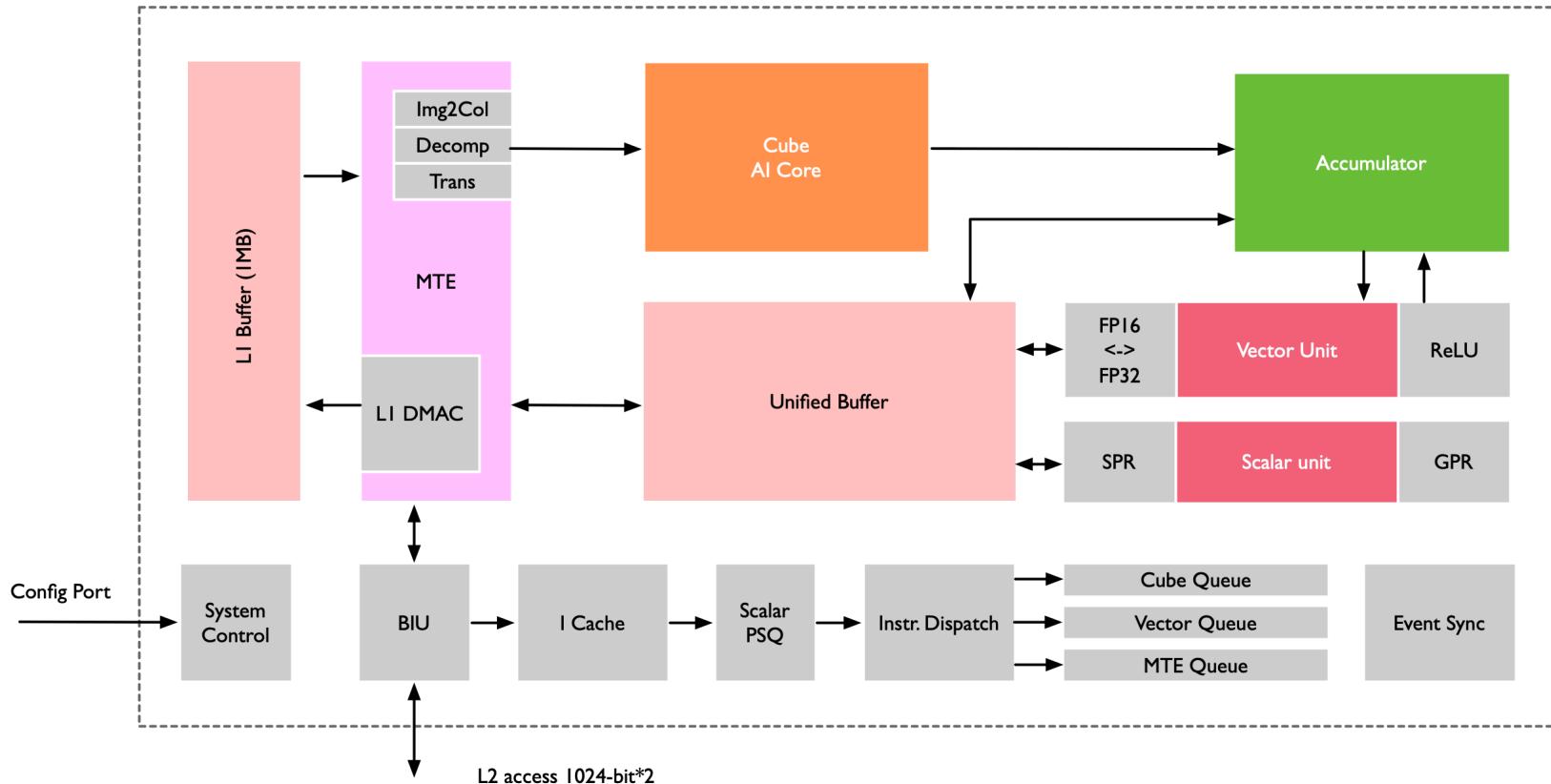
利用 AI Core 来加速 GEMM 计算

1. 总线接口从核外 L2 缓冲区/内存中读取计算指令，送入指令缓存，完成指令预取等操作；
2. 等待标量指令处理队列进行译码，当前无执行指令则读入指令，并进行地址和参数配置；



利用 AI Core 来加速 GEMM 计算

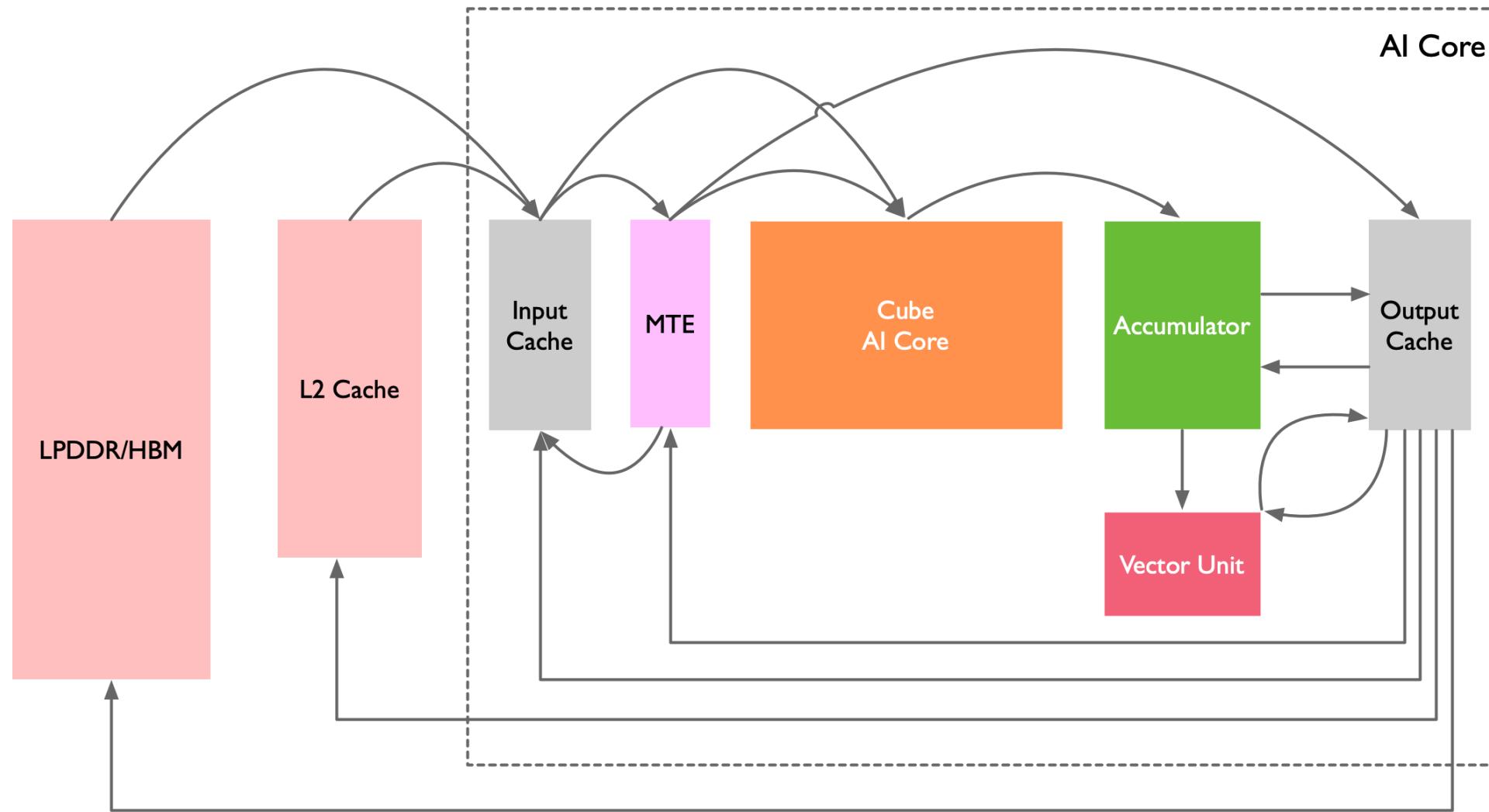
3. 指令发射模块按照指令类型，分别送入相应指令队列进行执行；
4. GEMM 计算首先发射数据搬运指令，被发送到存储转换队列中，最终转发到存储转换单元。



利用 AI Core 来加速 GEMM 计算

1. 总线接口从核外 L2 缓冲区/内存中读取计算指令，送入指令缓存，完成指令预取等操作；
2. 等待标量指令处理队列进行译码，当前无执行指令则读入指令，并进行地址和参数配置；
3. 指令发射模块按照指令类型，分别送入相应指令队列进行执行；
4. GEMM 计算首先发射数据搬运指令，被发送到存储转换队列中，最终转发到存储转换单元。

矩阵乘的数据通路



AI Core

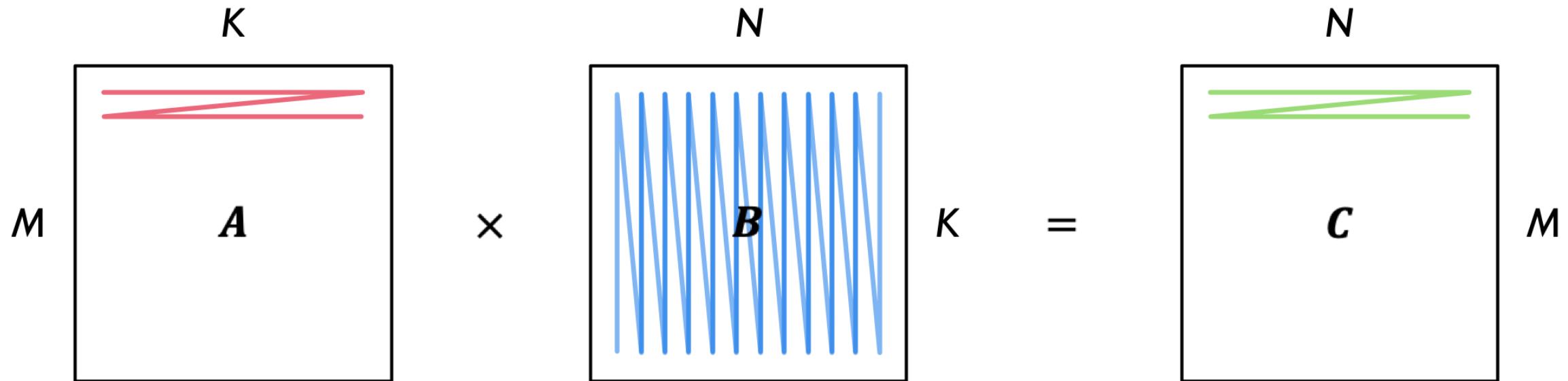
计算本质

Cube Core



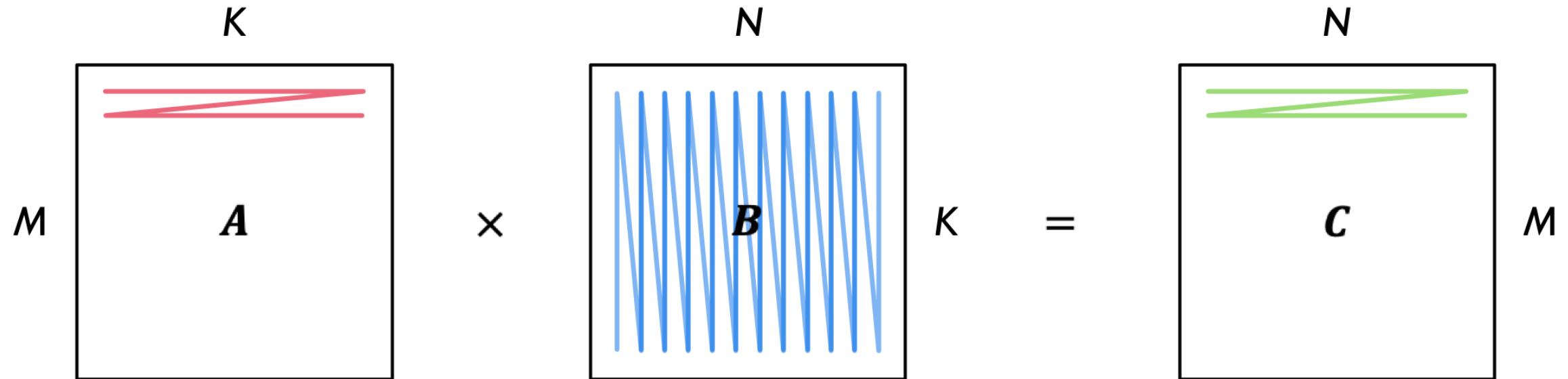
CPU 矩阵乘法

- 要用到 3 个循环进行一次完整矩阵相乘，在 SISD CPU 上执行至少需要 $M*K*N$ 个指令周期才能完成，当矩阵非常庞大时执行过程极为耗时。



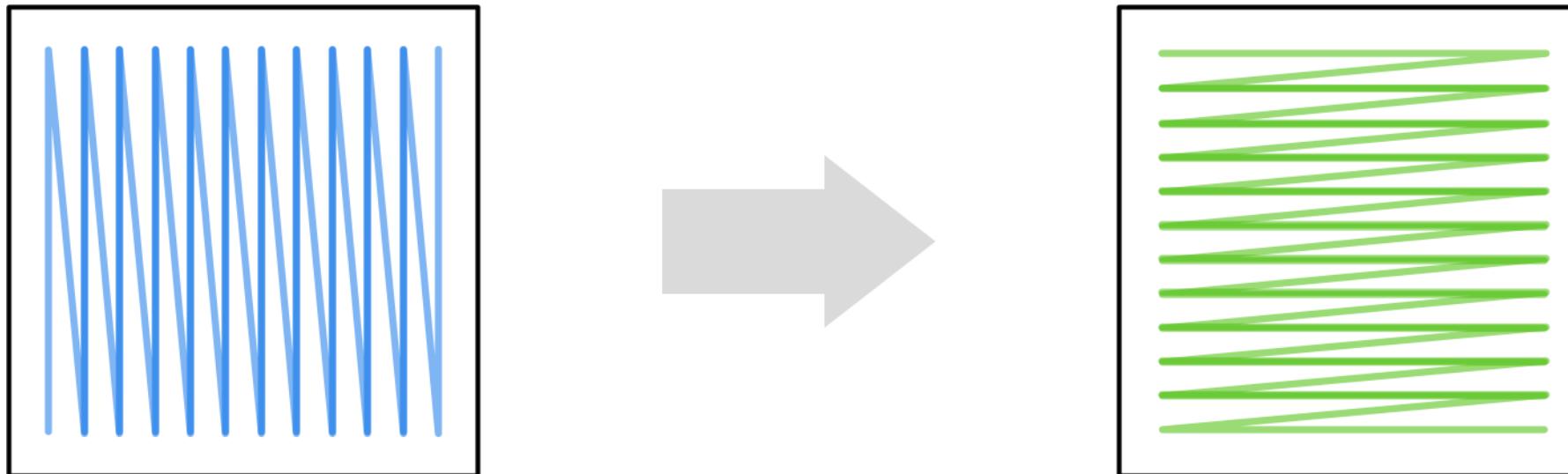
CPU 矩阵存储格式

- CPU 计算过程中，矩阵 A 按照行扫描，矩阵 B 按列扫描；典型矩阵存储矩阵 A & 矩阵 B 都按照行方式进行存放，即使 Row-Major 方式。



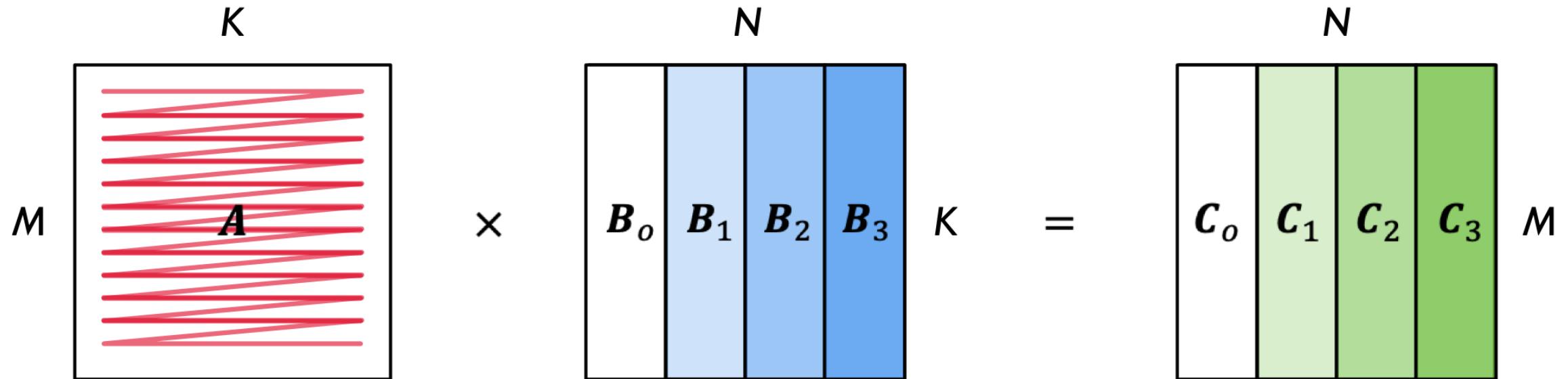
CPU 矩阵存储格式

- 内存读取按行读更方便，因此对 A 矩阵高效，B 矩阵低效。为此需要将矩阵 B 存储方式转成按列存储，即 Column-Major 矩阵计算，NPU 通过改变矩阵存储方式来提升矩阵计算的效率。



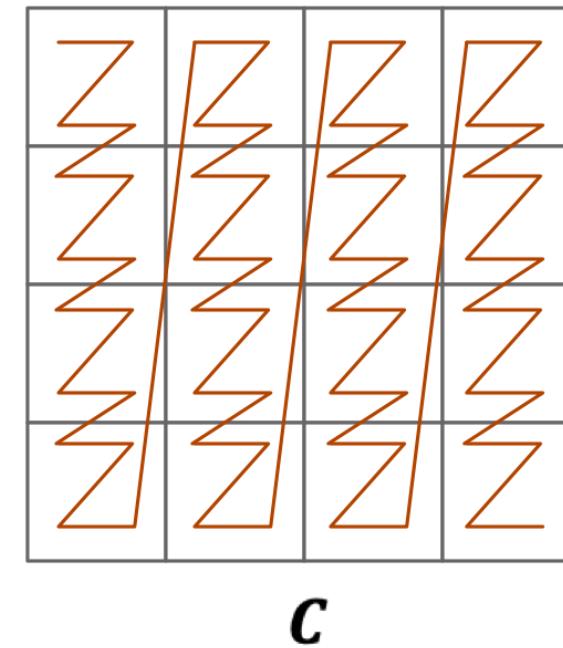
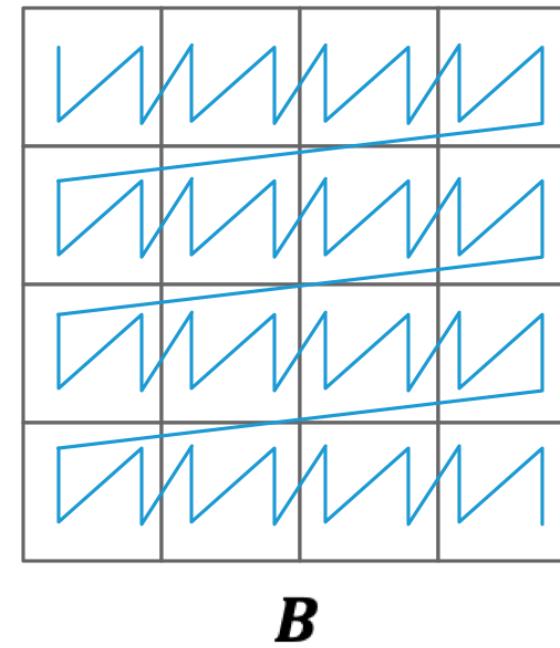
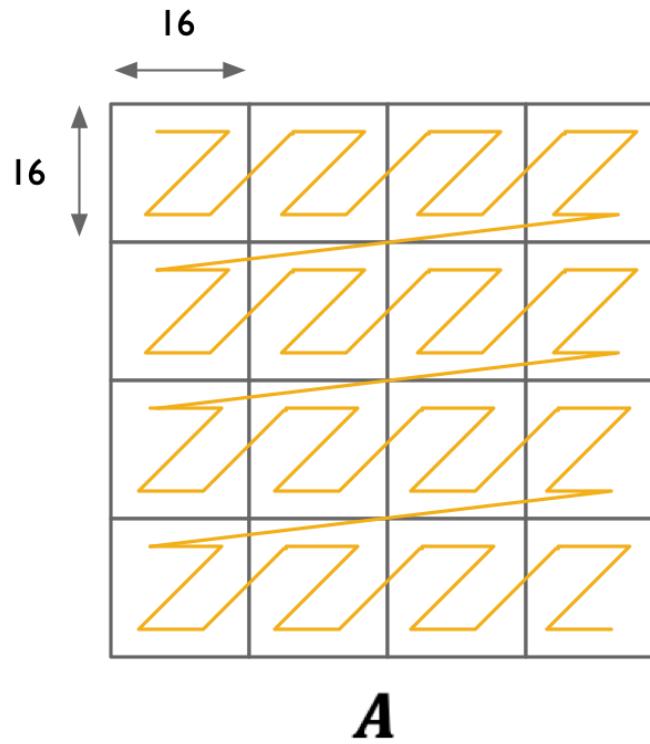
矩阵分块 Tiling

- 受限于片上缓存容量，一次难以装下整个矩阵 B ，将 B 划分成为 $B_0, B_1 \dots$ 等多个子矩阵；如此往复，依次将所有子矩阵搬运到缓存中，完成计算全过程，得到结果矩阵 C 。



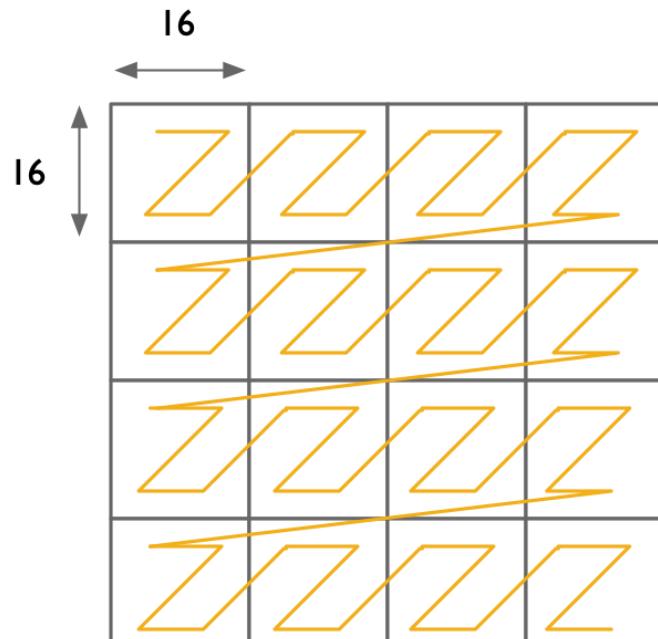
矩阵分块 Tiling

- 受限于片上缓存容量，一次难以装下整个矩阵 B ，将 B 划分成为 $B_0, B_1 \dots$ 等多个子矩阵；如此往复，依次将所有子矩阵搬运到缓存中，完成计算全过程，得到结果矩阵 C 。

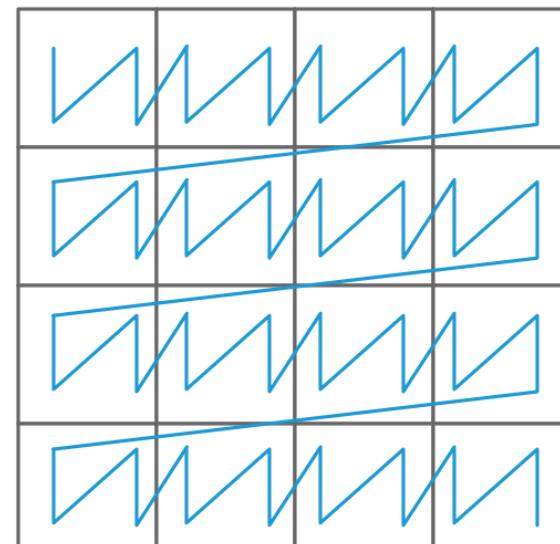


矩阵分块 Tiling

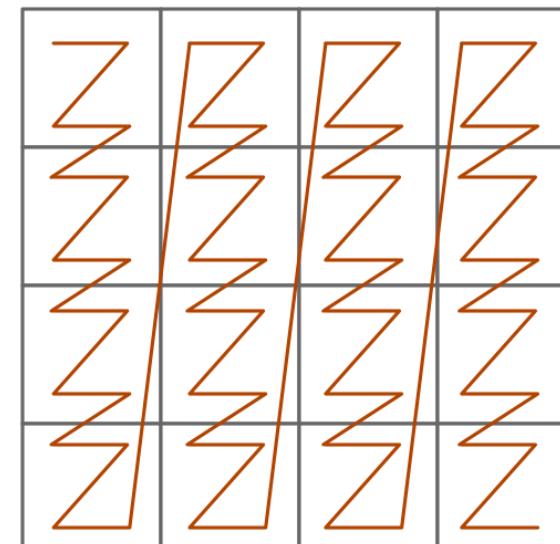
大 Z 小 z



大 Z 小 N



大 N 小 Z



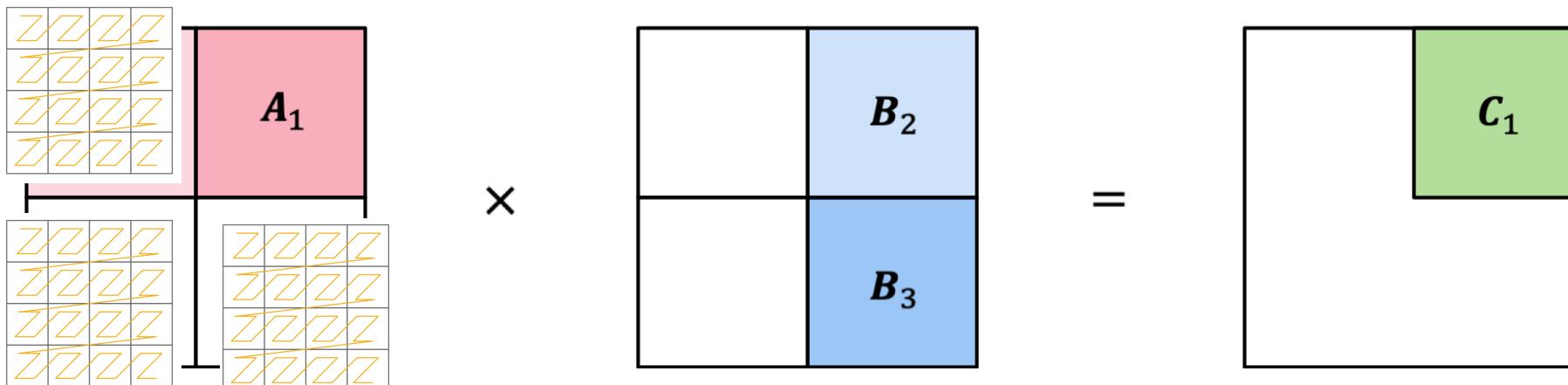
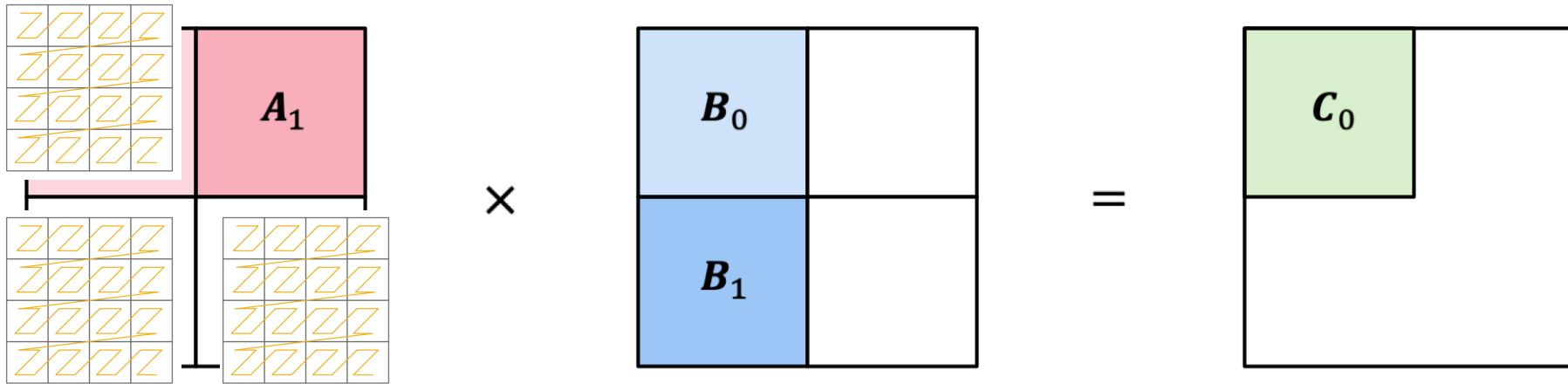
A

B

C

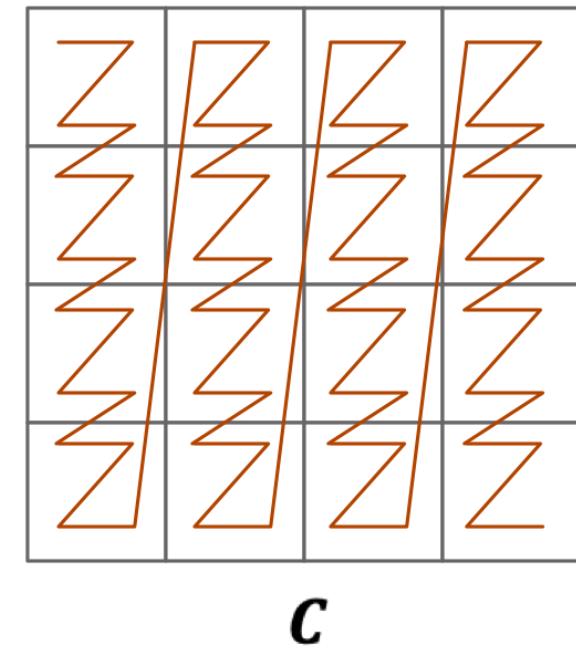
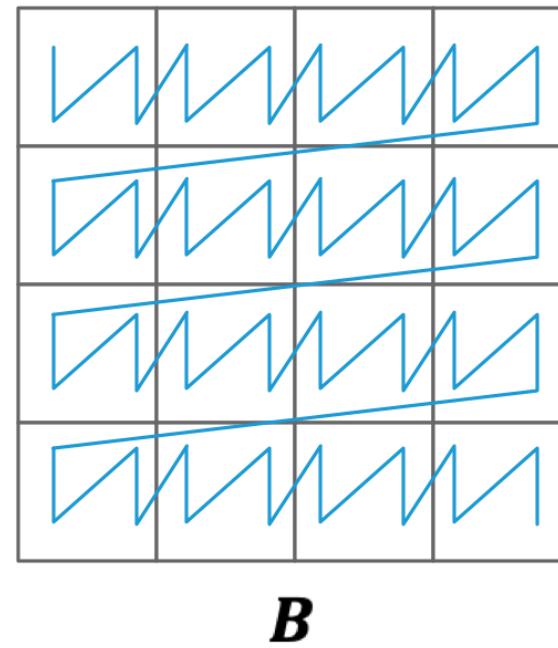
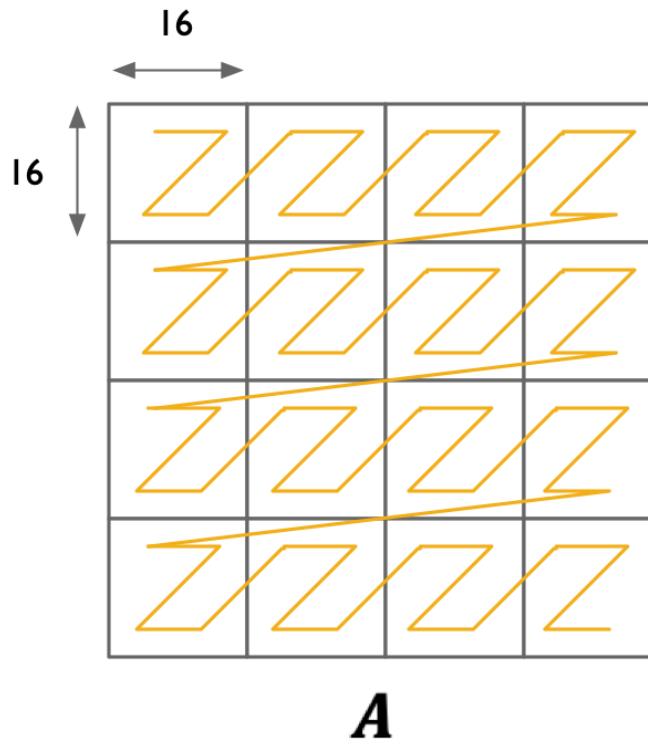
矩阵分块 Padding

- A 和 B 都等分成同样大小块，每一块是 16×16 子矩阵，排不滿的地方可以通过补零实现。

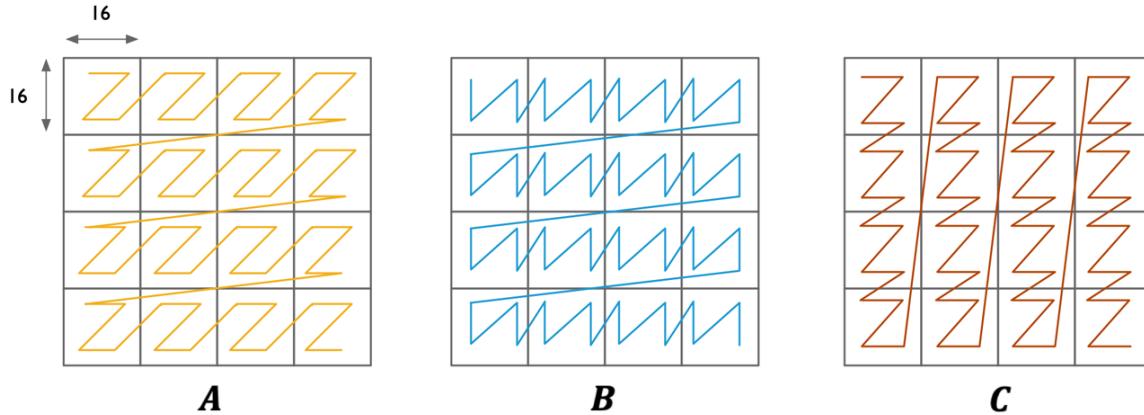


Cube Core 计算 MAC 16^3

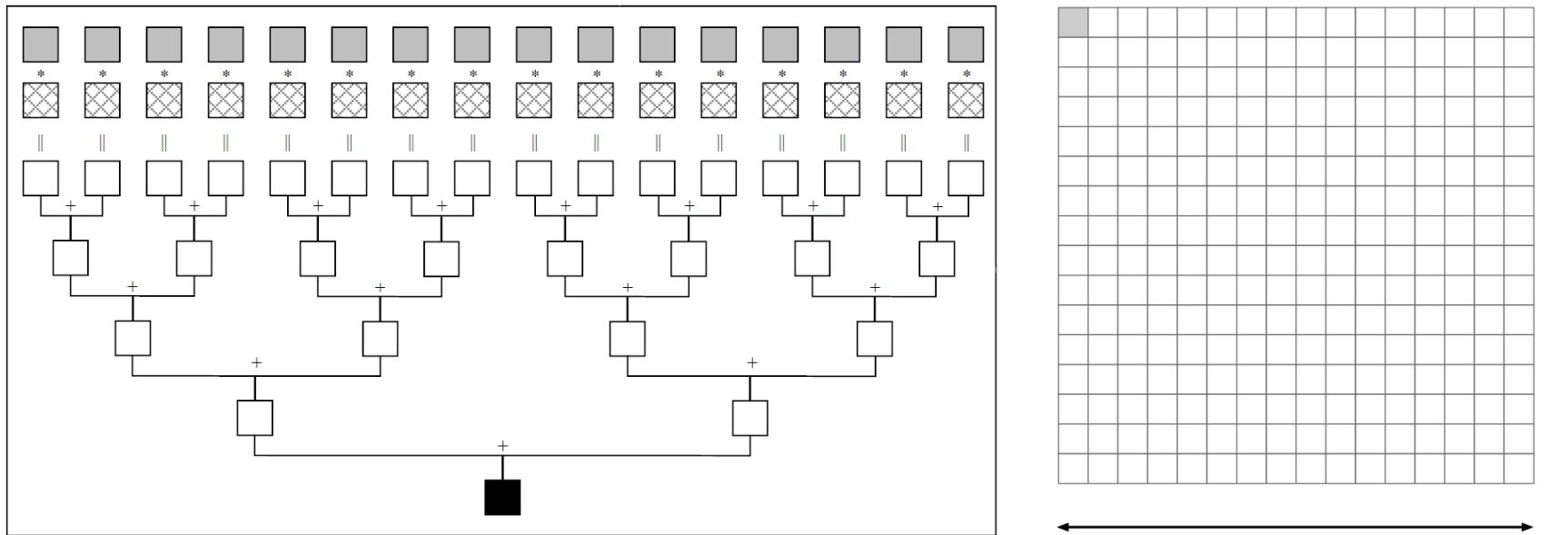
- Cube Core 一条指令完成两个 16×16 矩阵 MAC，等于一个时钟周期进行 $16^3 = 4096$ 个 MAC 计算；执行前将 A 按行 & B 按列存放在 Input buffer，通过 Cube Core 计算得到 C 按行存放在 Output Buffer。



Cube Core 计算 MAC 16^3



- C 第一元素由 A 第一行 16 个元素 & B 第一列 16 个元素通过 Cube Core 电路进行 16 次乘法 & 15 次加法计算得到。
- Cube Core 共有 256 个矩阵计算子电路组成，一条指令并行完成矩阵 C 256 个元素计算。

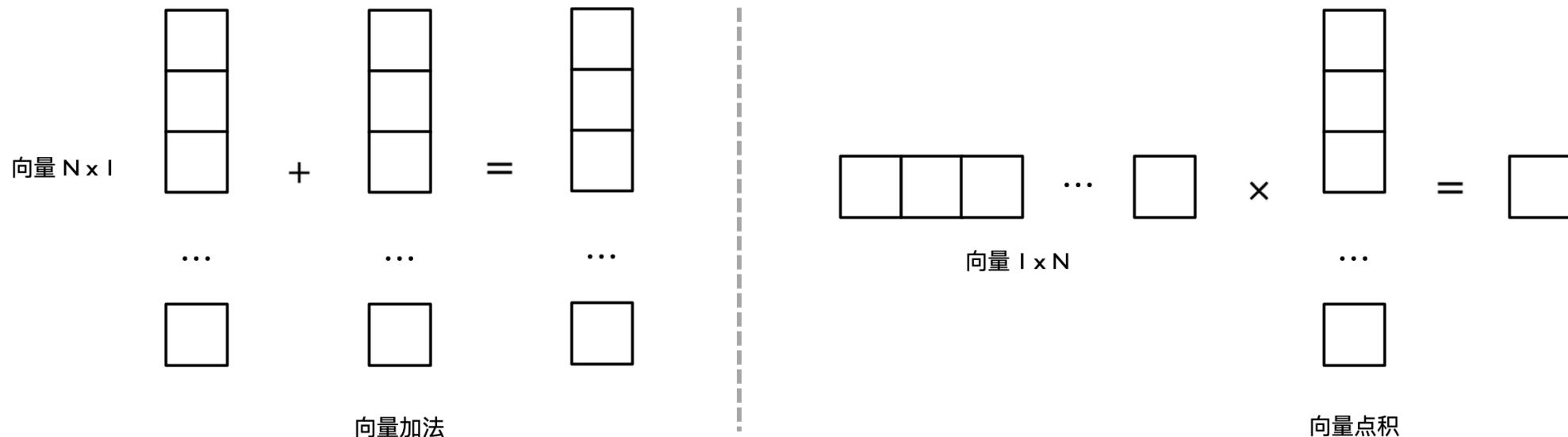


Vector Core



AI Core: 计算单元 Vector Unit

- Vector Unit 主要负责完成和向量相关运算，能够实现向量/标量/双向量间计算，功能覆盖各种基本和多种定制计算类型，包括FP32、FP16、INT32和INT8等数据类型计算。
 - e.g. 可快速完成两个 FP16 向量相加或者相乘。源操作数和目的操作数，通常保存在output buffer。
 - imp. 对Vector Unit 输入数据可以不连续，这取决于输入数据寻址模式。



小结与思考

思考

1. 针对大模型场景，你觉得 NPU 可以往哪个方向改进呢？
2. Cube Core 计算的时候要注意数据排布、矩阵分块、数据尾数补零对齐。





Thank you

把AI系统带入每个开发者、每个家庭、
每个组织，构建万物互联的智能世界

Bring AI System to every person, home and
organization for a fully connected,
intelligent world.

Copyright © 2023 XXX Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.



Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem