哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

深度学习体系结构（实验）

实验四 基于脉动阵列的CNN加速器设计

实验目的

- 了解脉动阵列的基本原理，熟悉脉动阵列的基本结构
- 掌握脉动阵列的HLS实现方法，掌握如何利用脉动阵列加速矩阵乘法和卷积运算
- 进一步熟悉使用HLS搭建硬件加速系统的方法和流程

实验内容

- 使用HLS编写脉动阵列，并对编写的代码进行CSim、综合并打包成IP核
- 构建Block Design电路图，生成并导出Overlay
- 对脉动阵列IP核进行GEMM测试
- 使用HLS Directive优化脉动阵列，并运行卷积、CNN测试

实验原理 — 脉动阵列简介

Systolic Array

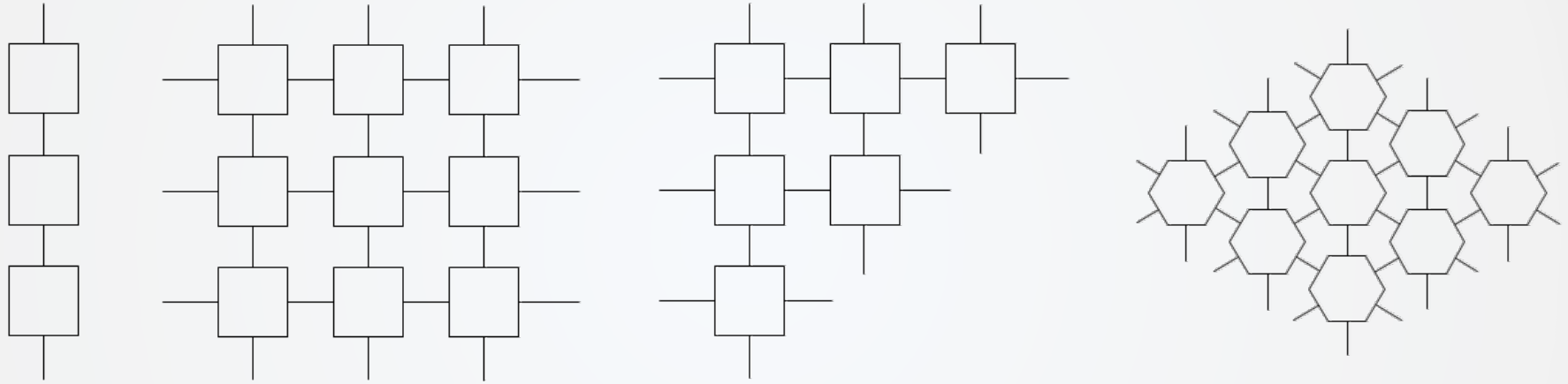
- 1982年由H. T. Kung提出，最初用于解决VLSI片上通信的性能瓶颈问题

  —— domain-specific架构的典型例子

- 设计专用系统需要考虑的3个因素：
  - 出于成本考虑 —— **简单性和规律性**
  - 出于性能考虑 —— **并行和通信、平衡计算和I/O**

实验原理 — 脉动阵列结构

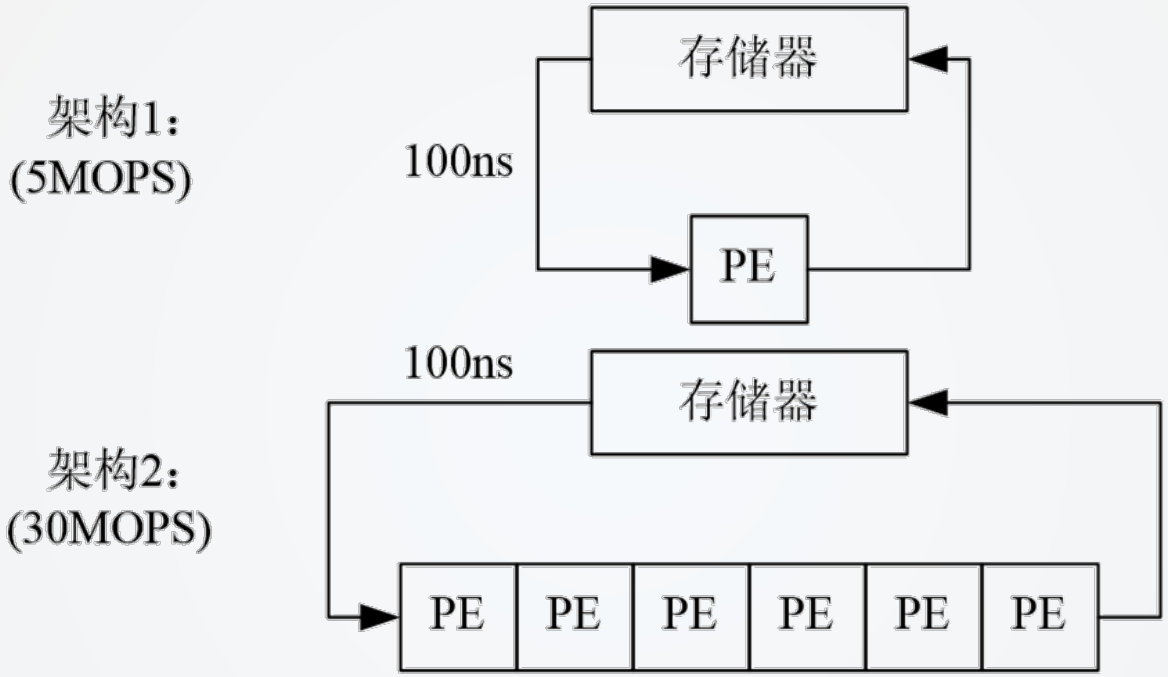
- 由许多PE按照特定规律连接起来的流水式同构多处理器架构
  - 每个PE仅与相邻的PE进行连接



  - 简单性和规律性：一种PE、连接规律
  - 并行和通信：无相关流水、最简通信（所有PE仅与相邻PE通信）

实验原理 — 脉动阵列结构

- 平衡计算与I/O —— 解决或缓解I/O带来的性能瓶颈问题

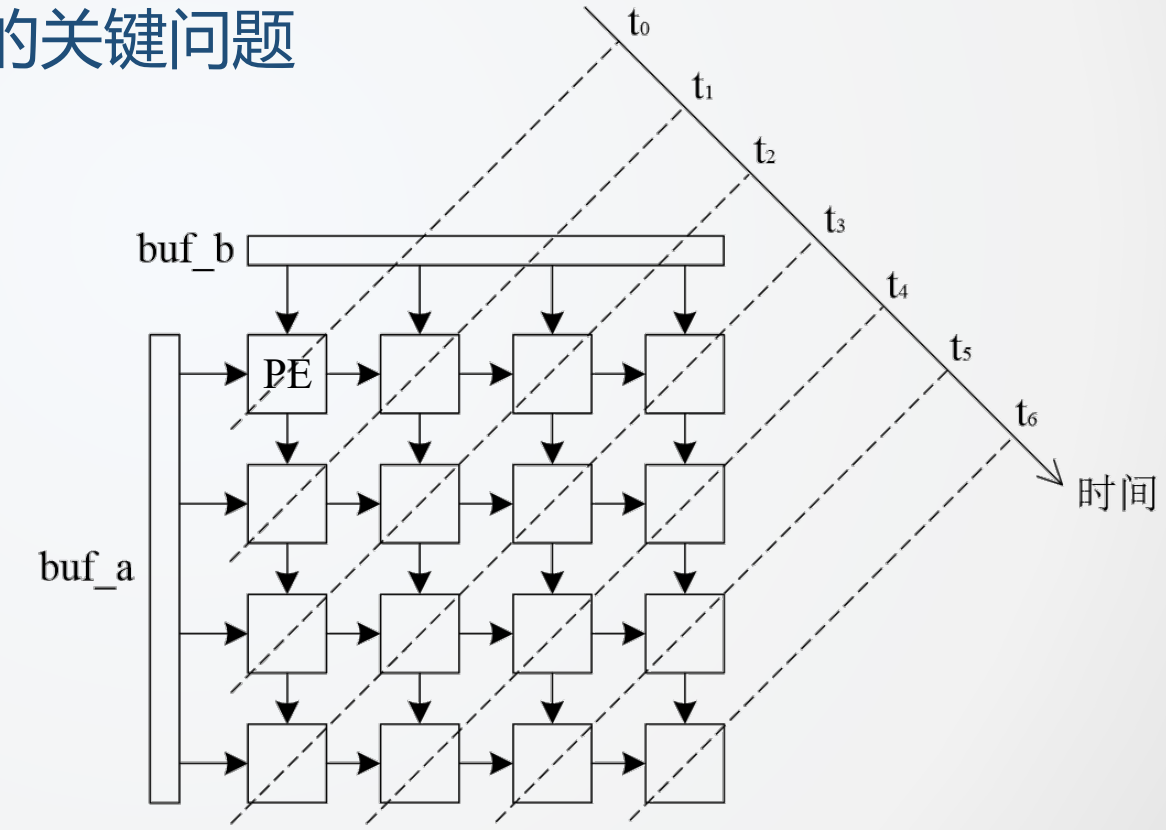


- I/O带宽不变，计算能力翻倍
  - PE再多，带宽都不变，因此只适合对**带宽要求较低**，并且**运算具有规律性**的计算密集型应用

实验原理 — 脉动阵列工作原理

“脉动”的阵列

- 数据在时钟的驱动下，像脉搏一般在阵列中向前跳动
- 设计和使用脉动阵列需要考虑的关键问题
  - PE实现什么运算?
  - 数据如何传递?

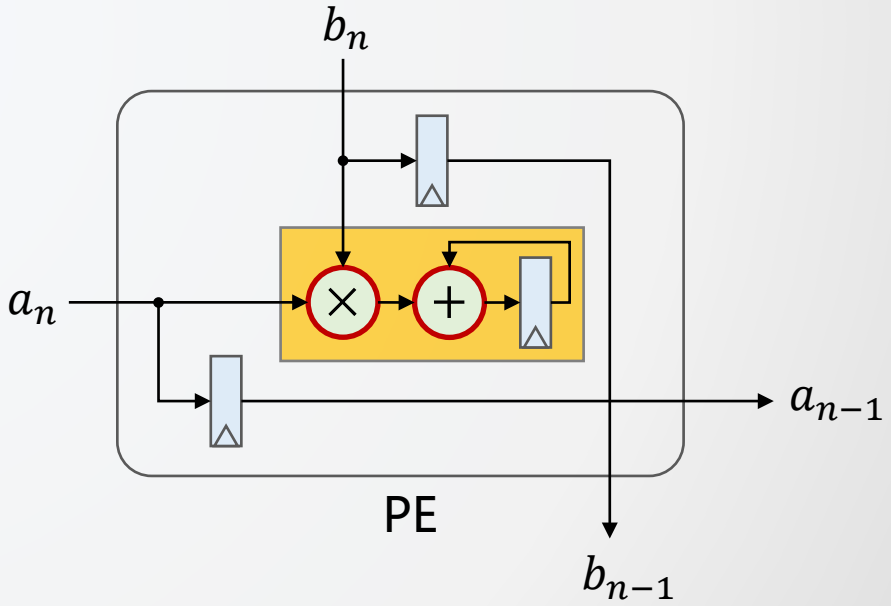


实验原理 — 脉动阵列计算GEMM

- 设有矩阵 $\boldsymbol{A}_{3\times3}$、 $\boldsymbol{B}_{3\times4}$，计算矩阵 $\boldsymbol{C}=\boldsymbol{A}\times\boldsymbol{B}$ ：

$$\boldsymbol{A}=\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}, \boldsymbol{B}=\begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \end{bmatrix}$$

- 矩阵乘法分析：
  - 带宽要求较低？ — 数据可重复使用
  - 运算有规律性？ — 核心运算：MAC



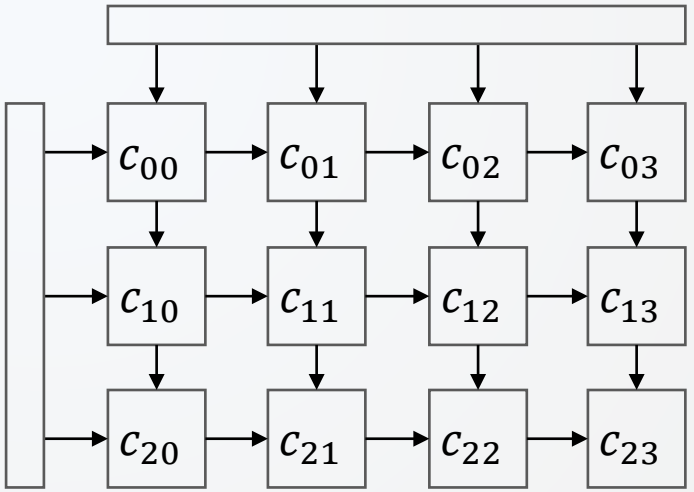
实验原理 — 脉动阵列计算GEMM

脉动阵列的设置

- PE存储计算结果
- $A_{3\times3}$矩阵从左流入，
  $B_{3\times4}$矩阵从上流入

⑤	④	③	②	①
		a_{02}	a_{01}	a_{00}
	a_{12}	a_{11}	a_{10}	
a_{22}	a_{21}	a_{20}		

⑥				b_{23}
⑤			b_{22}	b_{13}
④		b_{21}	b_{12}	b_{03}
③	b_{20}	b_{11}	b_{02}	
②	b_{10}	b_{01}		
①	b_{00}			



实验原理 — 脉动阵列计算GEMM

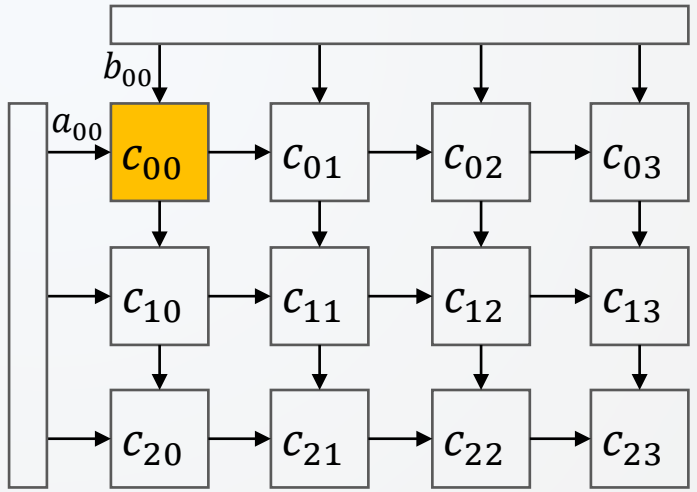
第1拍

- a_{00}、b_{00}流入阵列，开始计算c_{00}：

$$c_{00} \mathrel{+}= a_{00} \times b_{00}$$

⑥			b_{23}
⑤		b_{22}	b_{13}
④	b_{21}	b_{12}	b_{03}
③	b_{20}	b_{11}	b_{02}
②	b_{10}	b_{01}	

⑤	④	③	②
		a_{02}	a_{01}
	a_{12}	a_{11}	a_{10}
a_{22}	a_{21}	a_{20}	

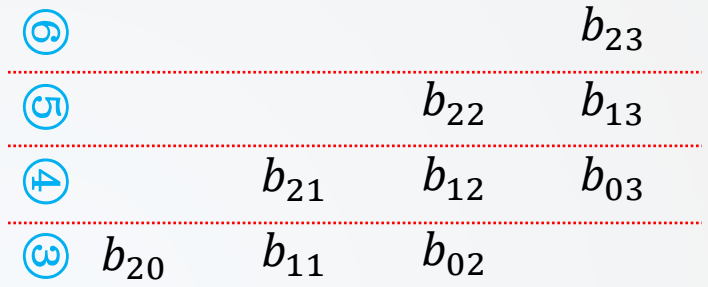


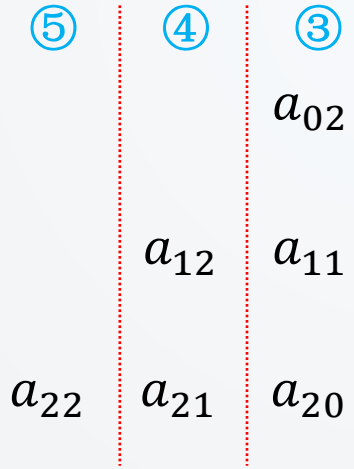
实验原理 — 脉动阵列计算GEMM

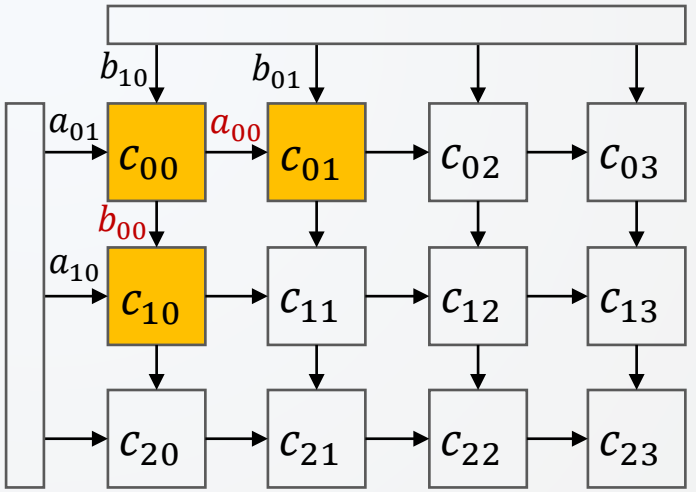
第2拍

- a_{01}、a_{10}、b_{10}、b_{01}流入阵列，开始计算c_{01}、c_{10}，继续计算c_{00}：

$$c_{00} \mathrel{+}= a_{01} \times b_{10}$$
$$c_{01} \mathrel{+}= a_{00} \times b_{01}$$
$$c_{10} \mathrel{+}= a_{10} \times b_{00}$$





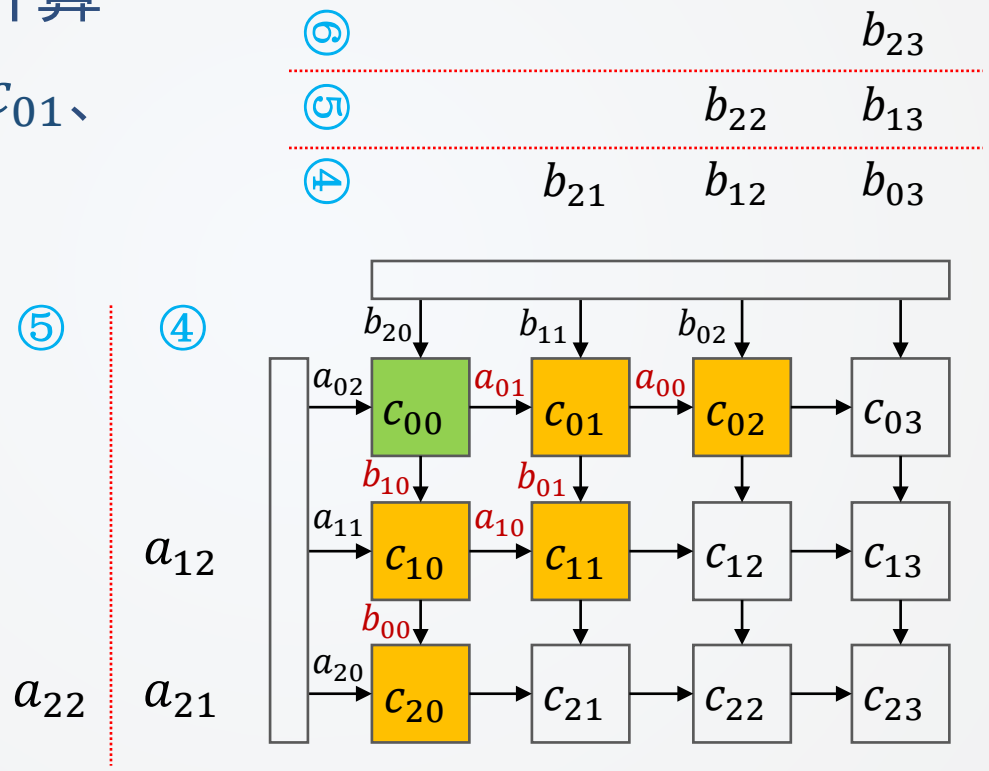


实验原理 — 脉动阵列计算GEMM

第3拍

- 数据继续流入阵列，开始计算c_{02}、c_{11}、c_{20}，继续计算c_{01}、c_{10}，c_{00}计算完成:

$$c_{00} \mathrel{+}= a_{02} \times b_{20}$$
$$c_{01} \mathrel{+}= a_{01} \times b_{11}$$
$$c_{10} \mathrel{+}= a_{11} \times b_{10}$$
$$c_{02} \mathrel{+}= a_{00} \times b_{02}$$
$$c_{11} \mathrel{+}= a_{10} \times b_{01}$$
$$c_{20} \mathrel{+}= a_{20} \times b_{00}$$



实验原理 — 脉动阵列计算卷积

三维卷积

- 输入特征图有3通道，卷积核有2个

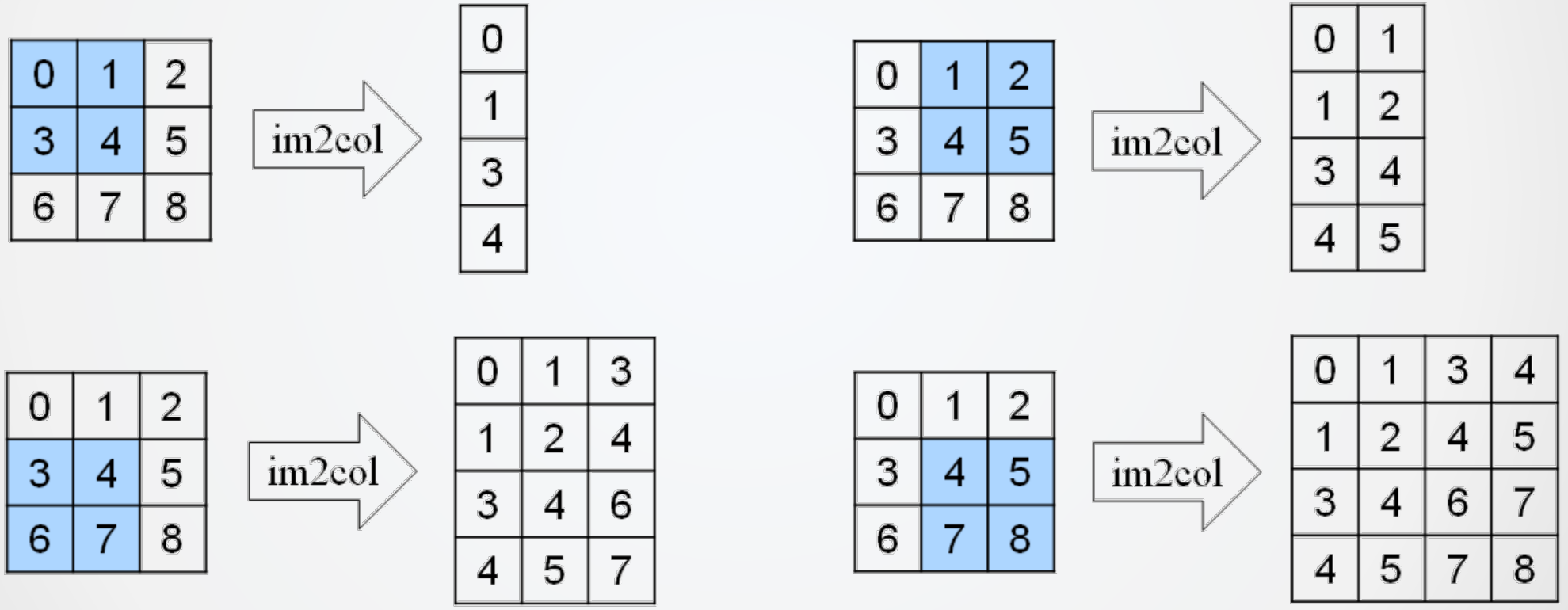

```
for (m = 0; m < OUT_CH; m++)
  for (r = 0; r < OUT_ROW; r++)
    for (c = 0; c < OUT_COL; c++)
      for (n = 0; n < IN_CH; n++)
        for (y = 0; y < KERN_R; y++)
          for (x = 0; x < KERN_C; x++)
            out(m, r, c) += in(n, r*S+y, c*S+x)*w(m, n, y, x);
```

- 如何用脉动阵列实现?
  - 维度不同
  - 算法操作复杂

实验原理 — 脉动阵列计算卷积

卷积降维：im2col

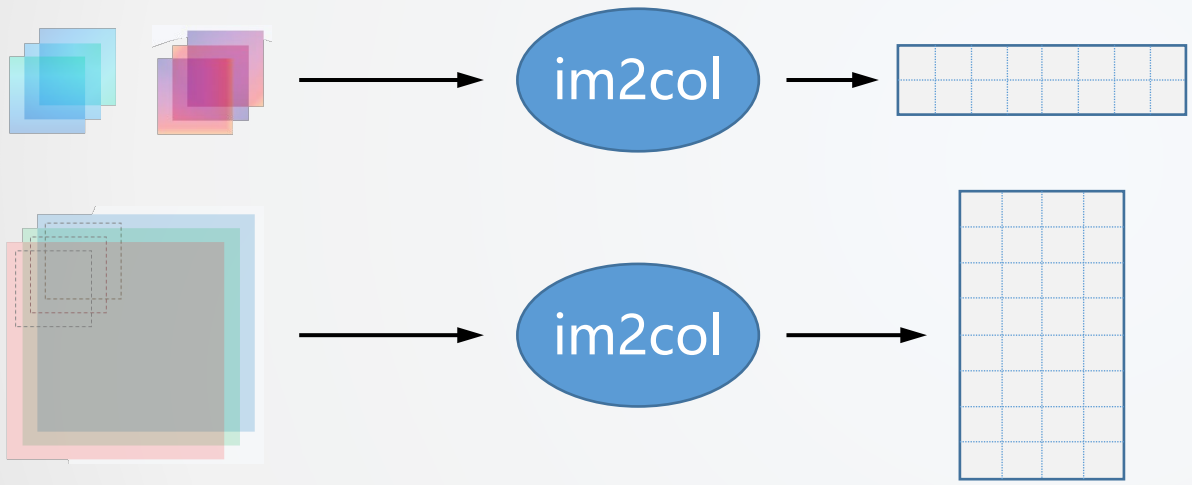
- 将3维特征图和卷积核展开成2维



实验原理 — 脉动阵列计算卷积

卷积的计算方法

- 先降维，再输入脉动阵列



方法1：

- 卷积核存储在PE中
- 输入特征图向右流动
- 卷积中间结果向下流动

方法2：

- 当成矩阵乘法处理
- 卷积中间结果存储在PE中
- 卷积核向右流动
- 输入特征图向下流动

实验步骤

1. 使用HLS编写脉动阵列IP核
2. 构建Overlay
3. GEMM测试
4. HLS优化
5. 运行卷积测试
6. 运行CNN

验收与提交

验收内容

序号	验收项目	分值
1	通过CSim	1分
2	通过GEMM测试	1分
3	HLS优化效果	2分
4	通过卷积测试	1分
5	成功运行CNN	2分

加速比≥130得1分；
≥250得2分

将源码、运行结果、实验报告打包提交（9分）

- 命名规则：**学号_姓名_DLA实验4.zip**
- 提交方法：https://hitsz-cslab.gitee.io/dla/ojguide
- DDL：下周同一上课时间前

哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

开始实验