智能芯片设计

第5章 智能芯片架构设计

华中科技大学 人工智能与自动化学院
多谱信息智能处理技术全国重点实验室

本章内容

5.1 时域与空域计算架构

5.2 DianNao系列智能芯片架构

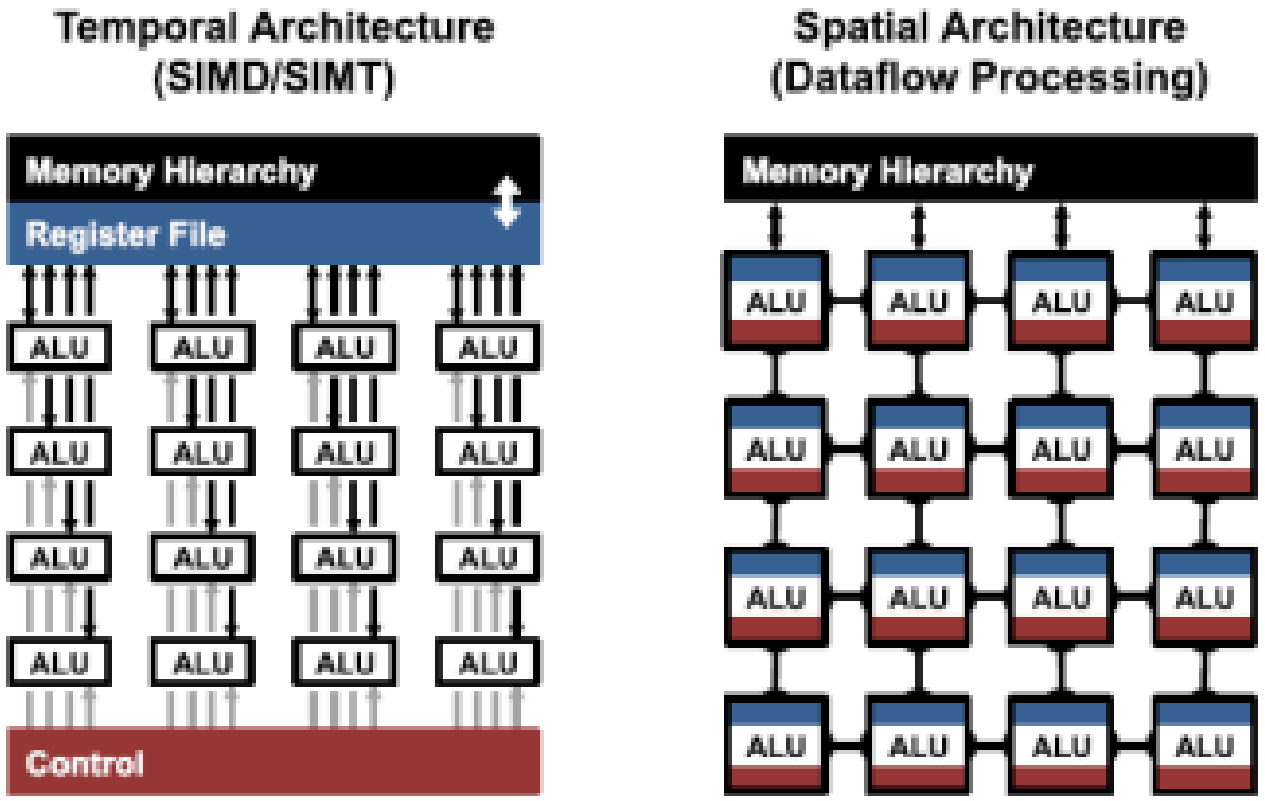
5.3 Thinker智能芯片架构

5.4 DNPU智能芯片架构

5.5 A310智能芯片架构

5.1 时域与空域架构

□ 智能芯片通常采用高度并行的计算架构：时域 和 空域 架构



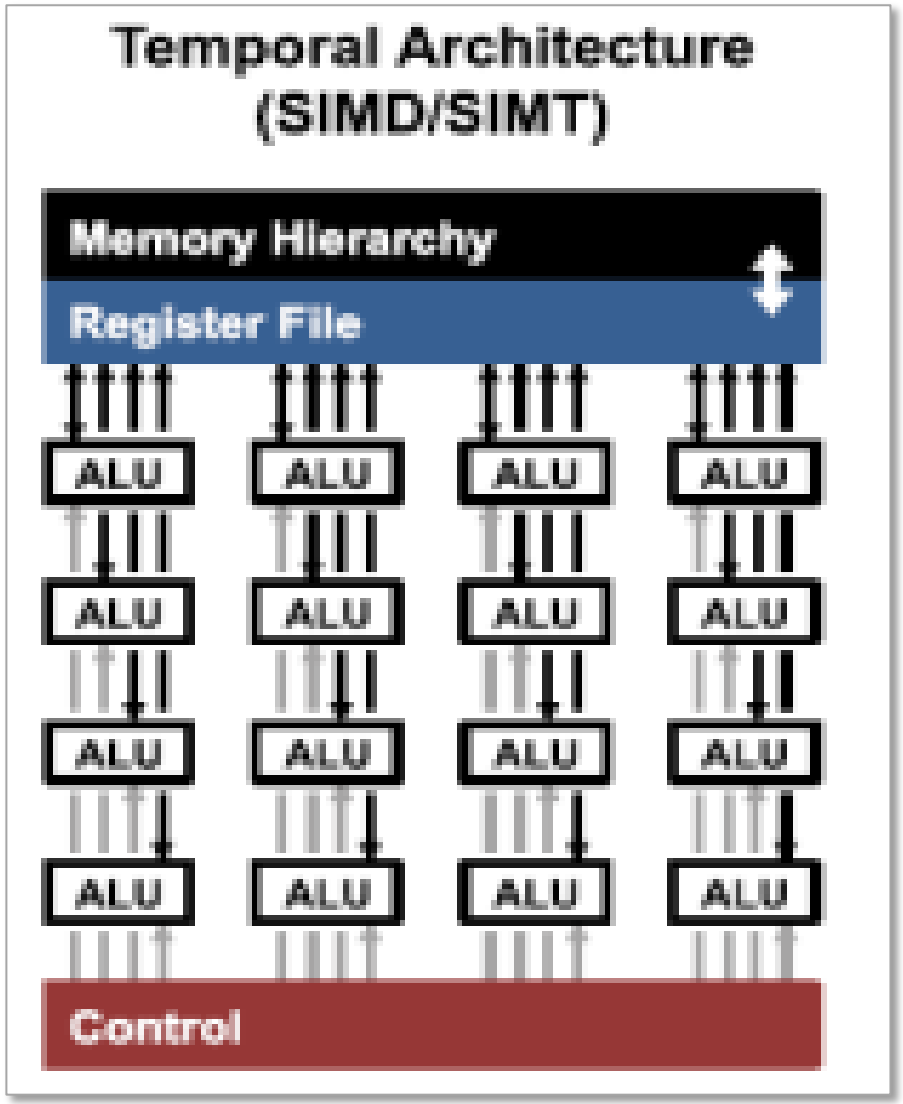
Sze V, Chen Y H, *etal. Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the, 2017, 105(12):2295-2329*

5.1 时域与空域架构

口时域架构

- 专门针对DNN**定制指令集**的专用处理器架构
- 基于指令流集中控制ALU和存储资源
- 每个ALU都从集中式存储器获取运算数据，并向其写回结果
- AI处理器发展的早期：**标量处理器**，通过系列标量运算完成大型矩阵和张量的计算

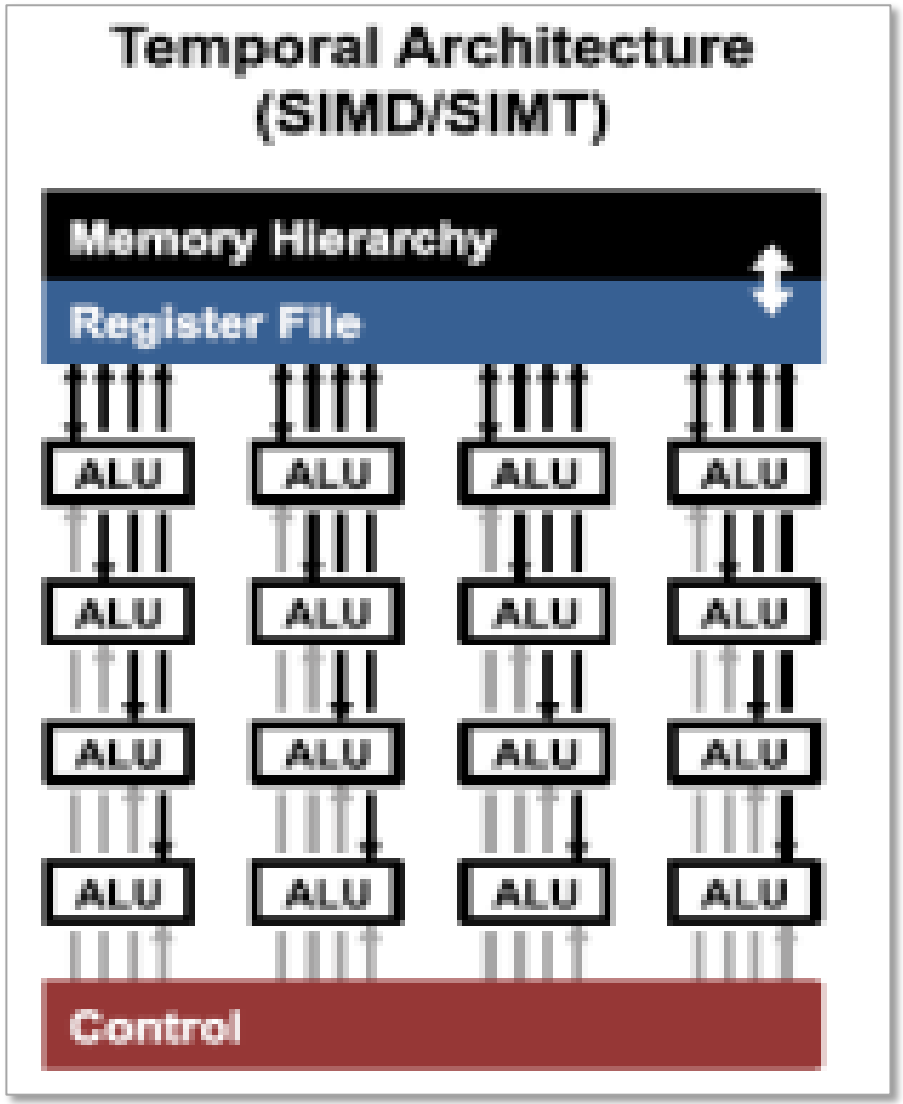
- 架构改进：**矢量处理器**，SIMD，同一指令同时针对多个数据元素执行相同的操作
- CPU → GPU



5.1 时域与空域架构

口时域架构

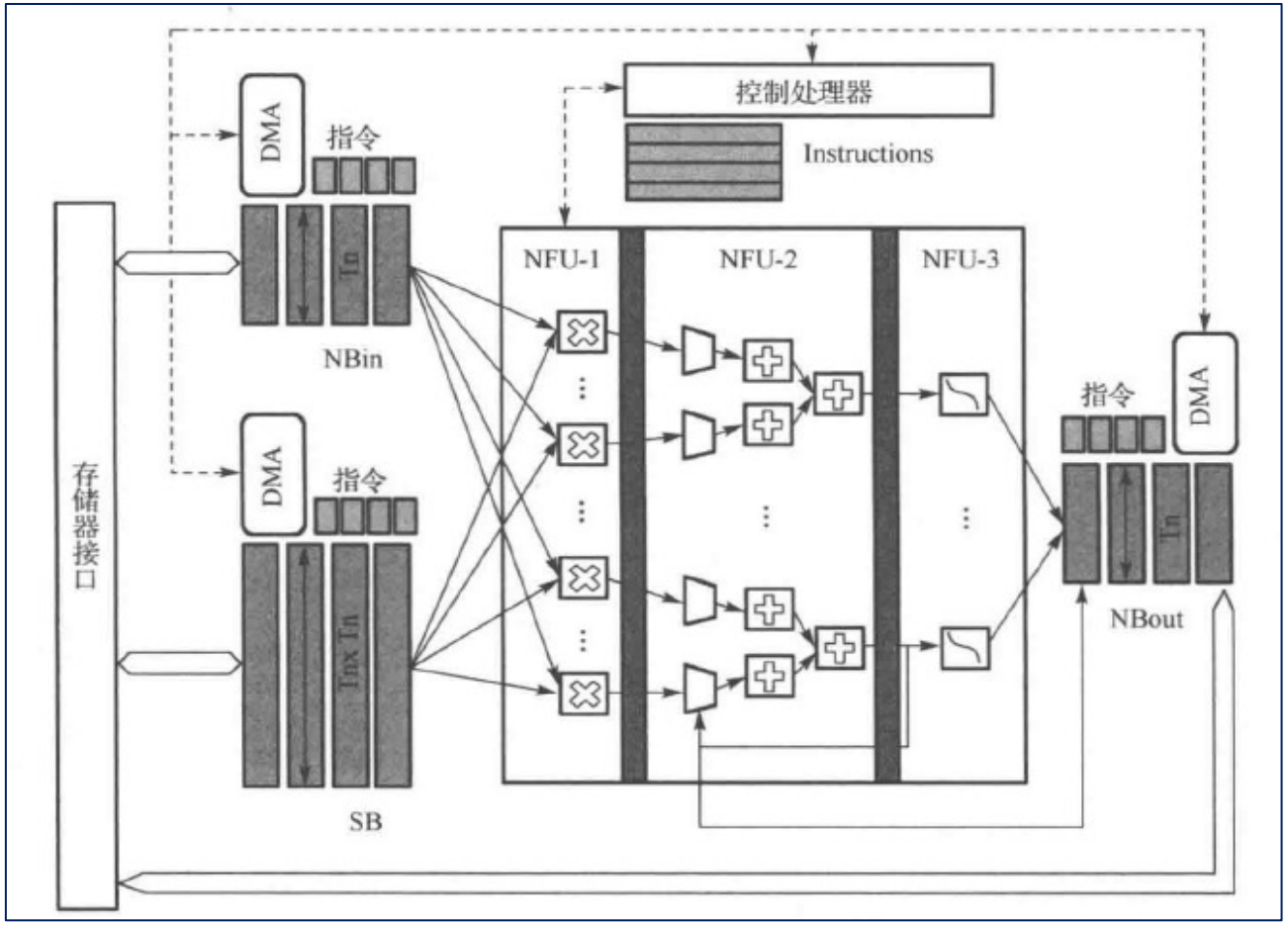
- 计算单元配置灵活，通用性较强
- 每一步操作需要精确的指令控制存储器访问和具体计算单元的操作类型
- 集中式存储导致片外存储交互频繁，性能和能效受限
- 标量处理器→矢量处理器



5.1 时域架构代表性设计

- 中科院计算所的DianNao(2014)

- NPU、缓存（NBin/Nbout/SB）和控制器
- NPU和缓存在指令流驱动下受控制器统一组织调度
- NPU三级流水：NFU1（乘法）、NFU2（加法）、NFU3（激活）、
- 较128bits的SIMD处理器，达到117倍加速比、21倍降能耗

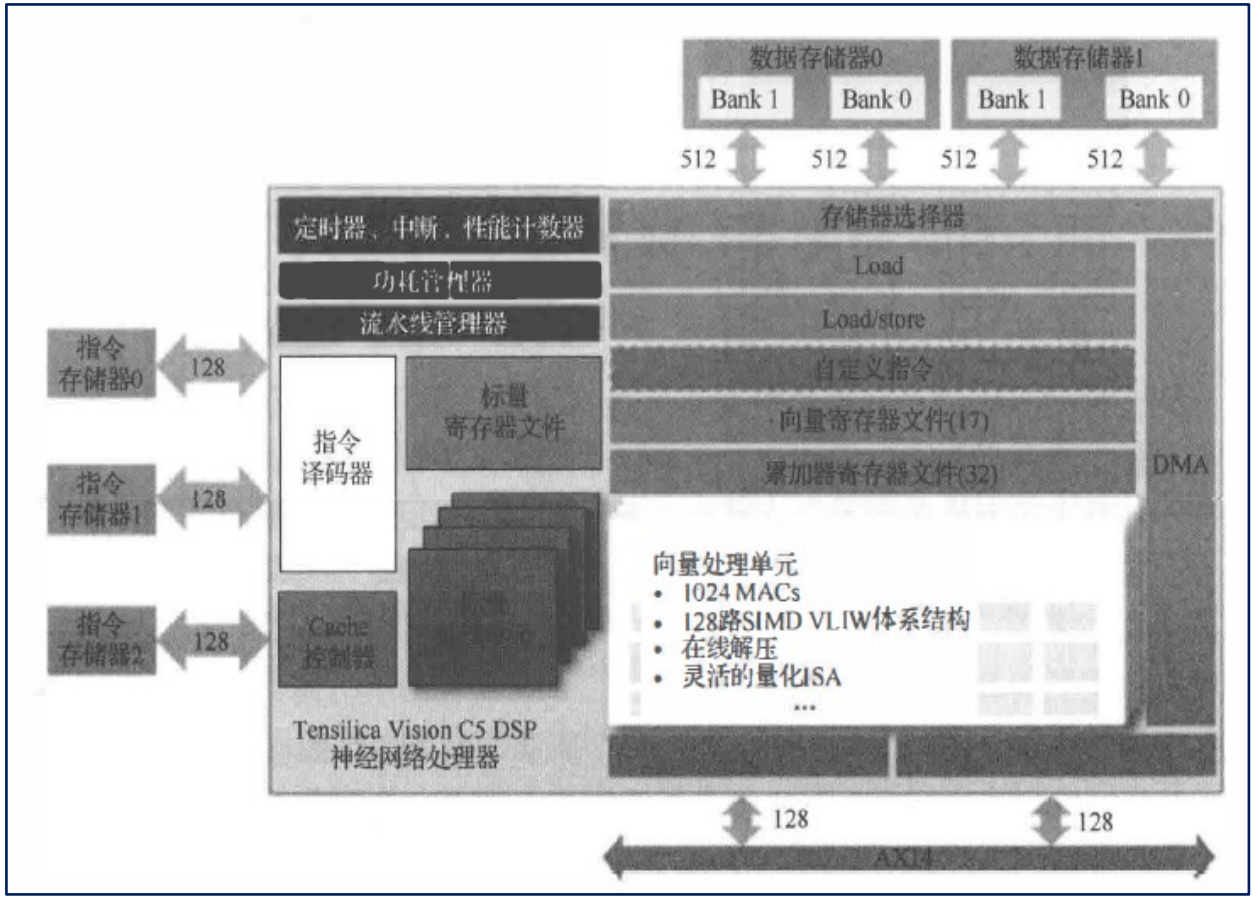


Chen Y, Chen T, *etal. DianNao family: energy-efficient hardware accelerators for machine learning. Research highlight, Communications of the ACM, 2016, 59(11):105-112*

5.1 时域架构代表性设计

- Cadence 公司的Vision C5 DSP (2017)

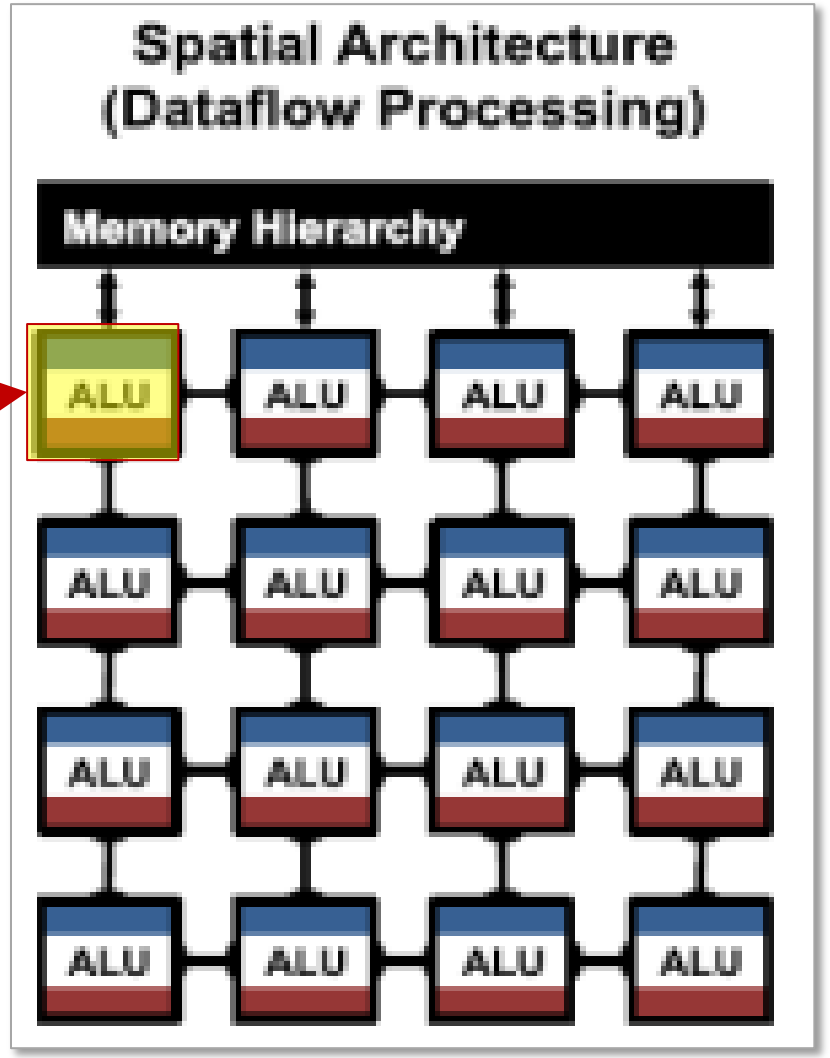
- SIMD VLIW体系结构
- 支持卷积/全连接/池化等计算
- 1mm^{2}面积 1 TMAC/s
- 较2017年GPU： AlexNet基准最快提高6倍，Inception V3基准最快提高9倍



5.1 时域与空域架构

口空域架构

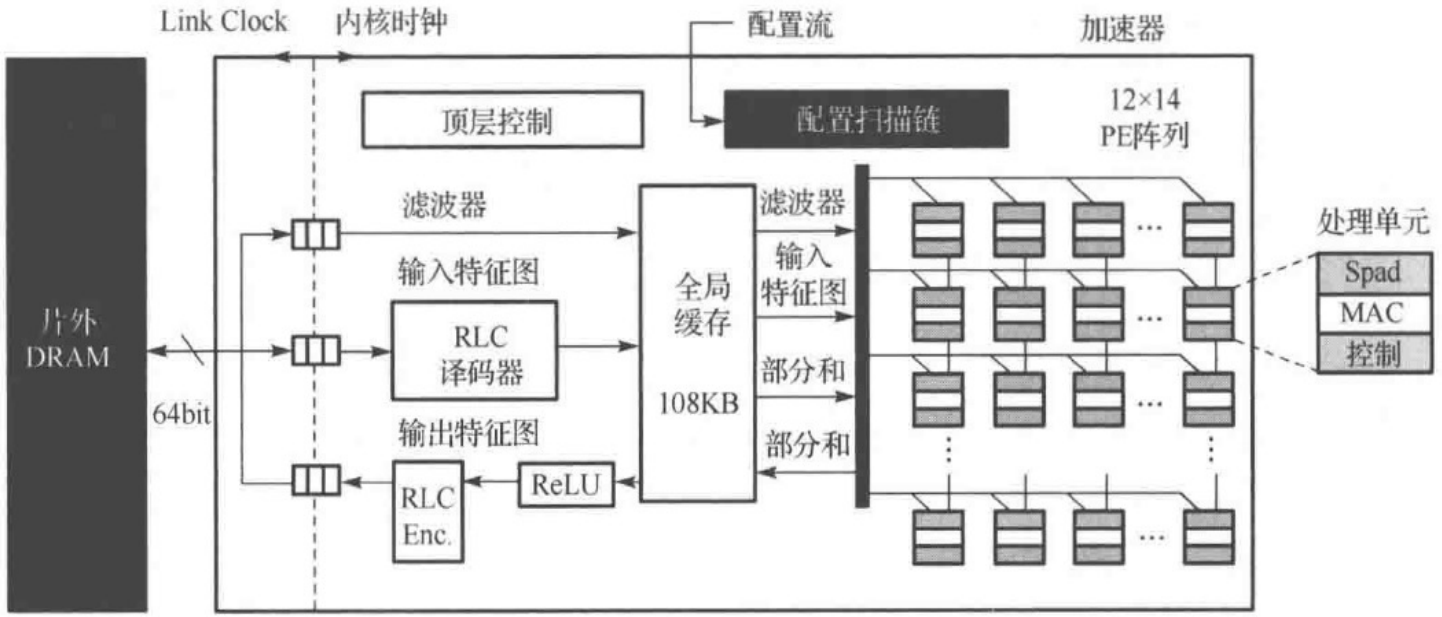
- 每个ALU都有独立的控制逻辑
- 每个ALU都有本地存储器，如本地缓存或寄存器组
- **ALU+本地存储器 = 计算单元PE**
- 基于数据流控制：所有ALU形成处理链，数据在ALU间传递
- 有效提高计算任务的执行性能，控制能耗；但缺少时间域上的可变性，灵活性不足



5.1 空域架构代表性设计

MIT的Eyeriss (2016)

- 1个12×14的PE阵列
- 每个PE自带控制器和本地存储器，PE之间可直接传递数据
- PE本地Reg、片上全局缓存、片外DRAM多级存储架构
- 行复用（RS）数据流提升片上数据复用率
- 计算能效较传统方法提升10倍

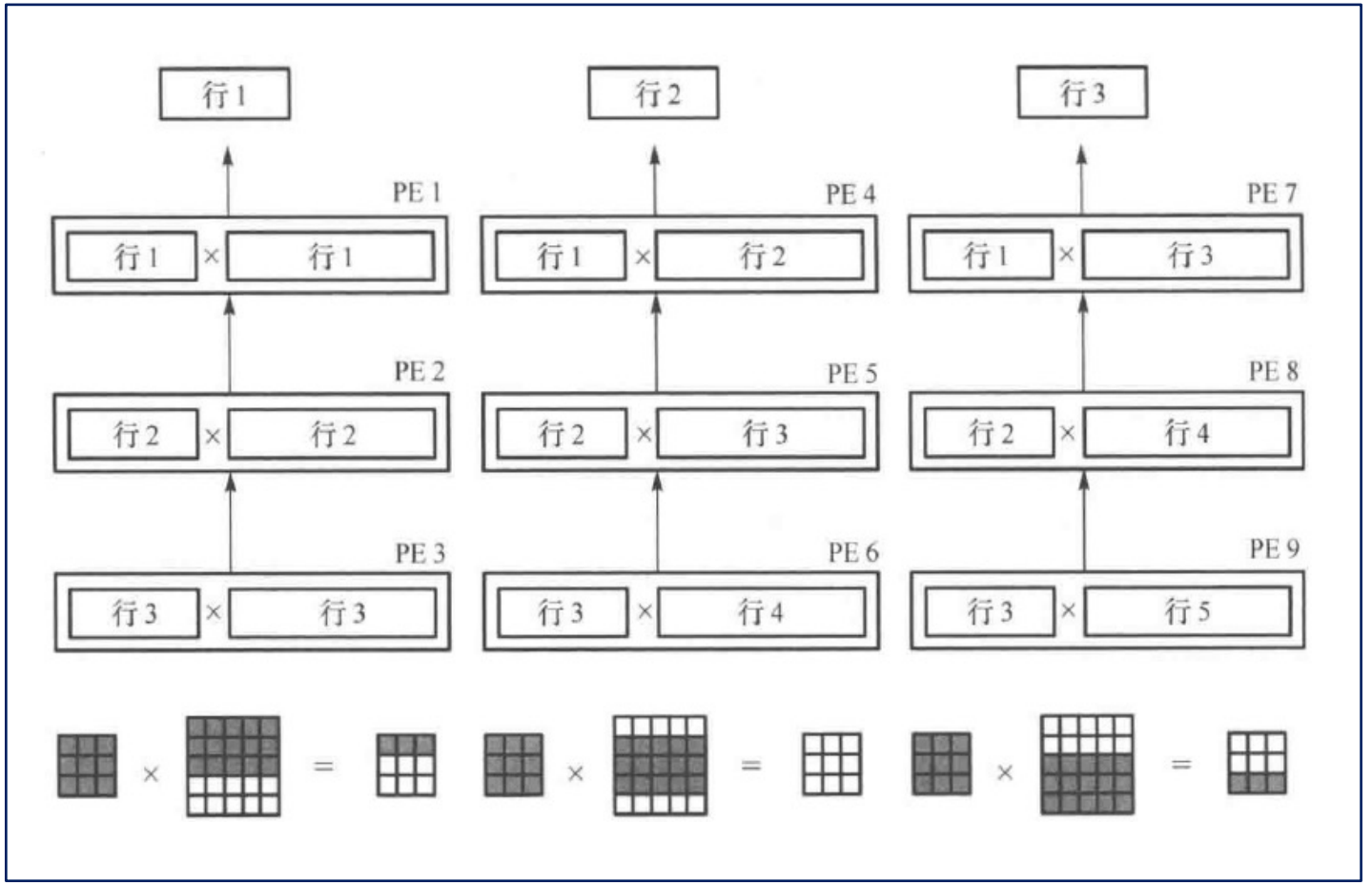


Chen Y H, Krishna T, *etal. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits, 2017, 52(1): 127-138*

5.1 空域架构代表性设计

- MIT的Eyeriss (2016)

- 3×3核3行 → 同列的3个PE
- 流过PE的IFM相乘，与邻行传递的部分和累加，再行间传递
- 权重数据及部分和数据不需要与全局缓存交互，实现了权重数据和输出数据的重用
- 低成本存储访问提升能效
- 计算能效较传统方法提升10倍

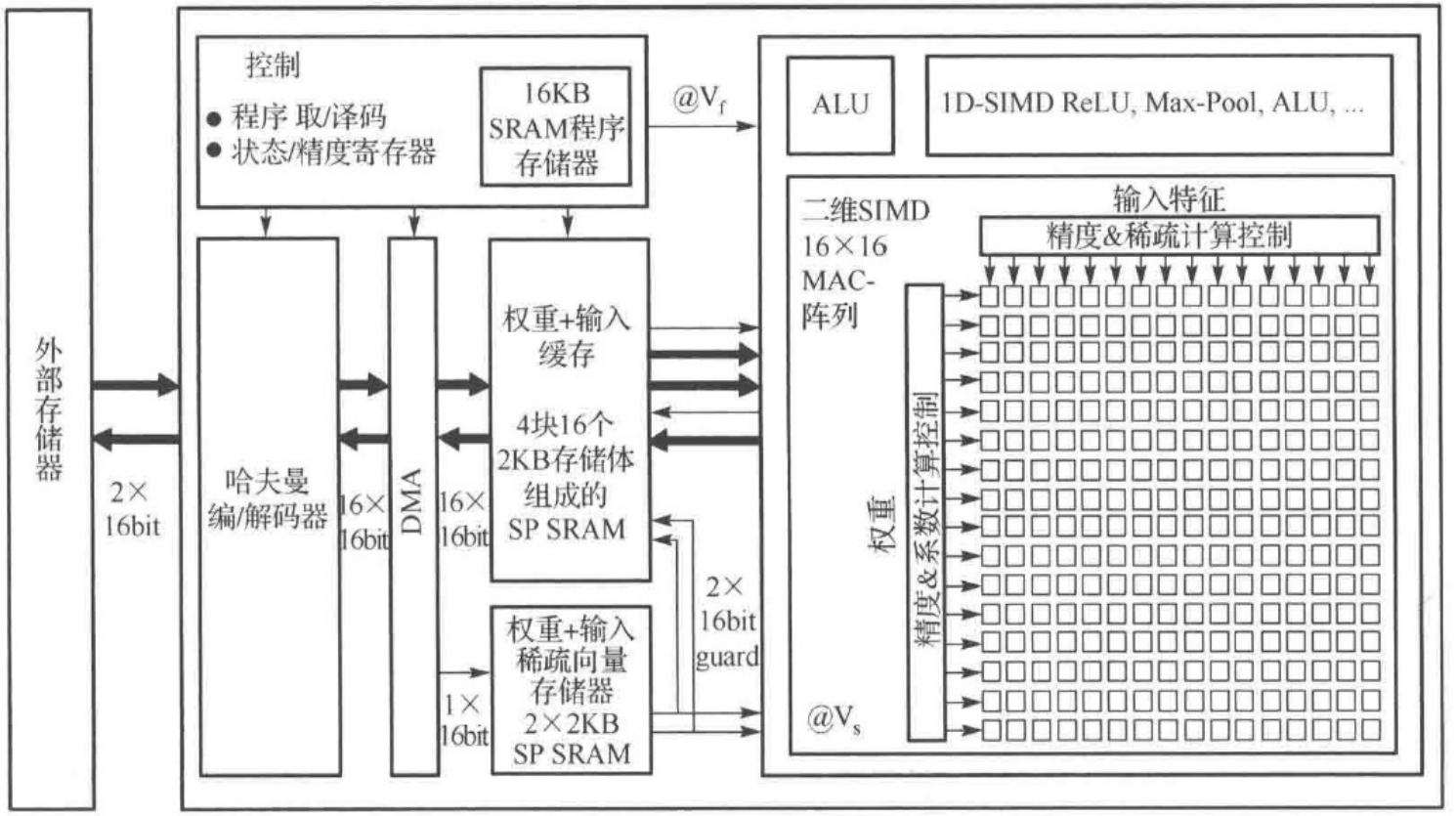


Chen Y H, Krishna T, *etal. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits, 2017, 52(1): 127-138*

5.1 空域架构代表性设计

口 比利时鲁汶大学的ENVISION (2017)

- 16bit RISC核，专用ReLU和池化一维SIMD处理单元
- 卷积层/全连接层MAC阵列16×16,电压-精度-频率动态缩放DVAFS
- DNN内在并行性/重用性提高MAC阵列数据效率
- 稀疏化网络压缩与计算，提高DNN硬件执行效率，10TOP/W

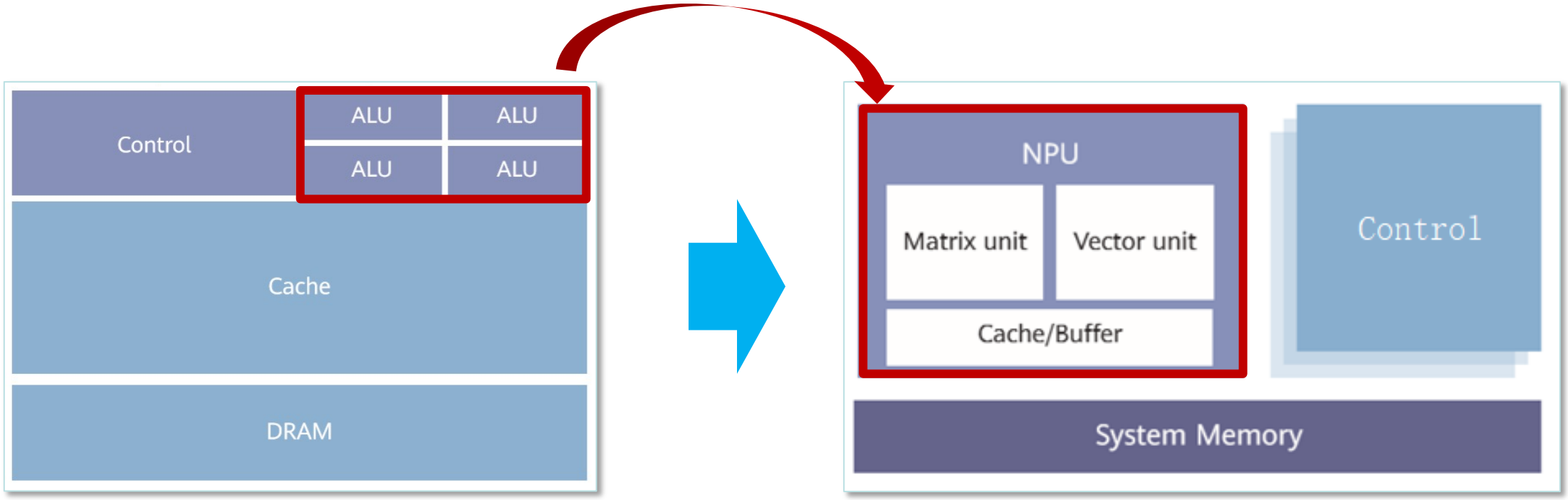


5.1 时域与空域架构

- 一种DLP基本架构：设计运算、存储、控制高效专用架构 (DSA)

DSA: Domain Specific Architecture

ALU → NPU：专注于向量/矩阵/张量运算

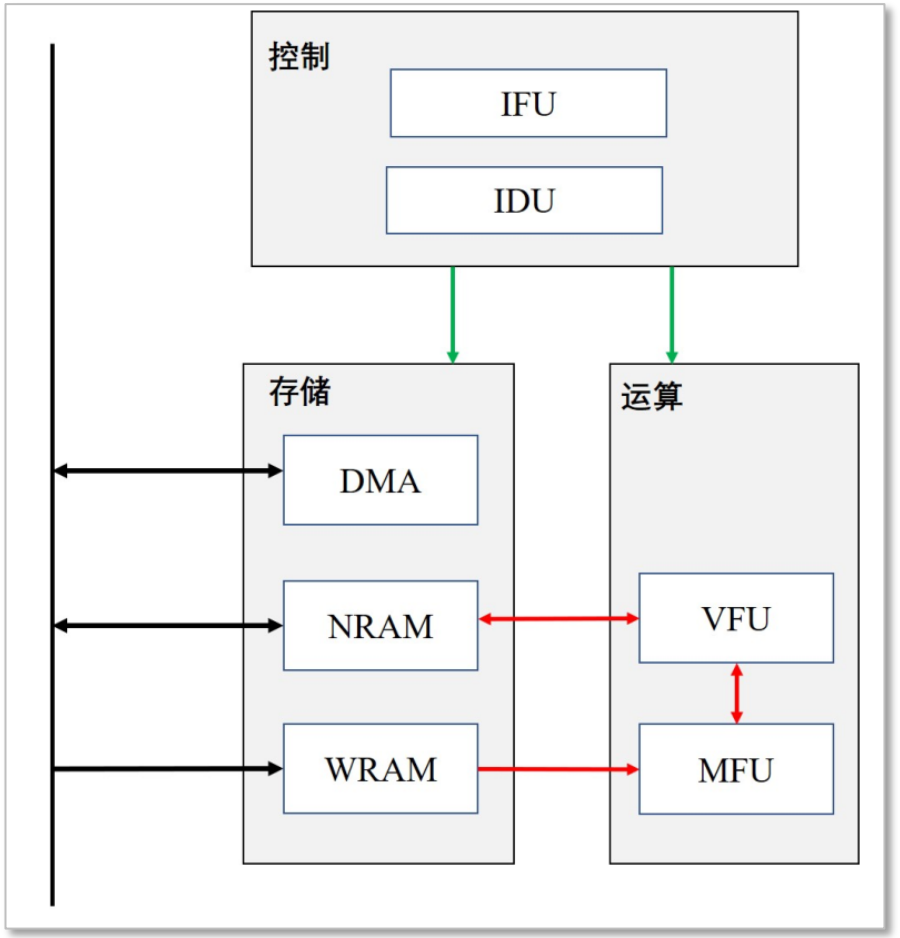


CPU　　DLP

5.1 时域与空域架构

口 一种DLP基本架构

- **控制模块**
  - ◆ 指令的语义粒度（提供专用指令，操作粒度为tensor）
  - ◆ 领域专用指令 vs. RISC vs. CISC
- **运算模块**
  - ◆ 基于tensor语义设计运算模块
- **存储模块**
  - ◆ 基于tensor语义设计存储模块

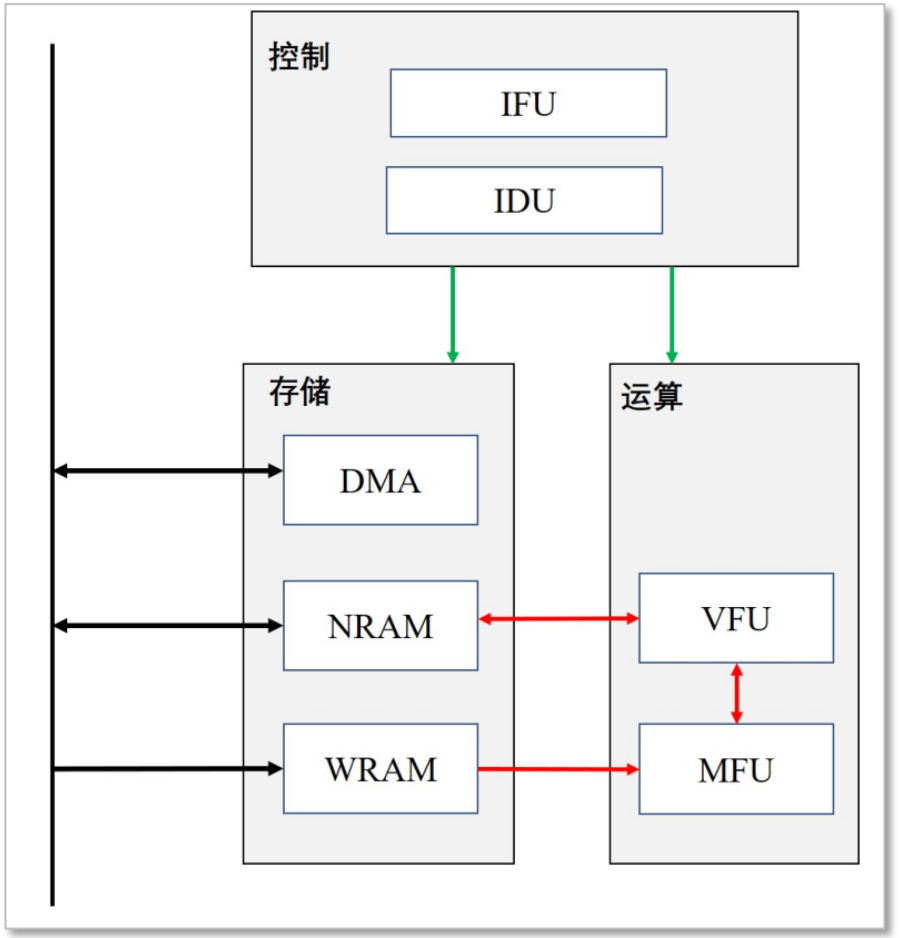


DLP基本架构

5.1 时域与空域架构

一种DLP基本架构

- **控制模块**
  - 取指单元IFU（Instruction Fetch Unit）
  - 指令译码单元IDU（Instruction Decode Unit）
- **运算模块**
  - 向量运算单元VFU（Vector Function Unit）
  - 矩阵运算单元MFU（Matrix Function Unit）
- **存储模块**
  - 权重存储单元WRAMW（Weight RAM）
  - 神经元存储单元NRAM（Neuron RAM）
  - 直接内存存取单元DMA（Direct Memory Access）



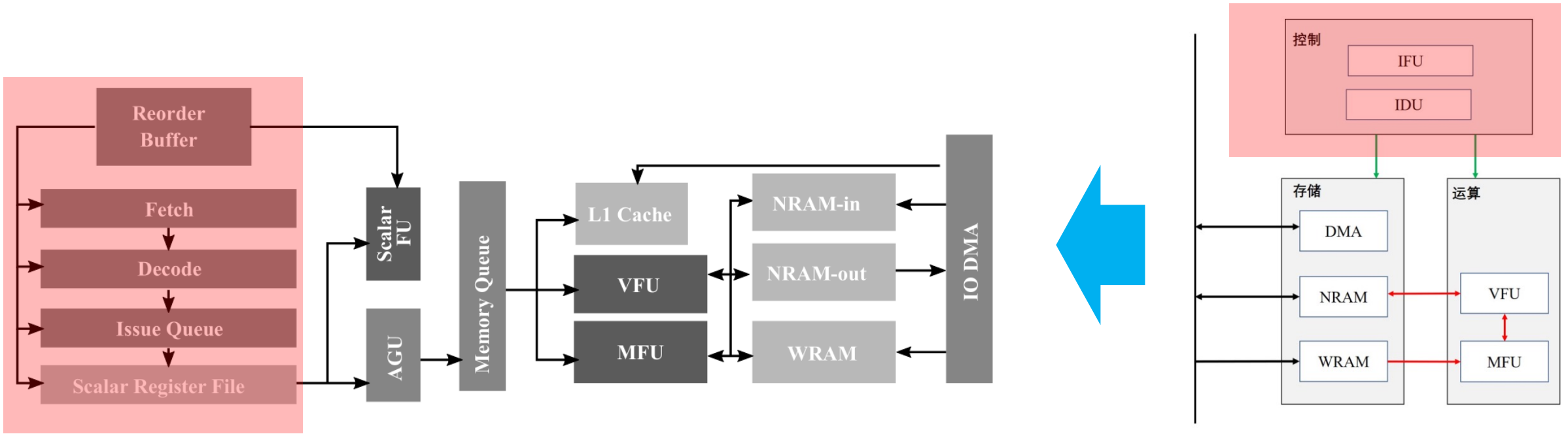
DLP基本架构

5.1 时域与空域架构

口 一种DLP基本架构

控制：

多发射队列，支持指令级并行，寄存器重命名



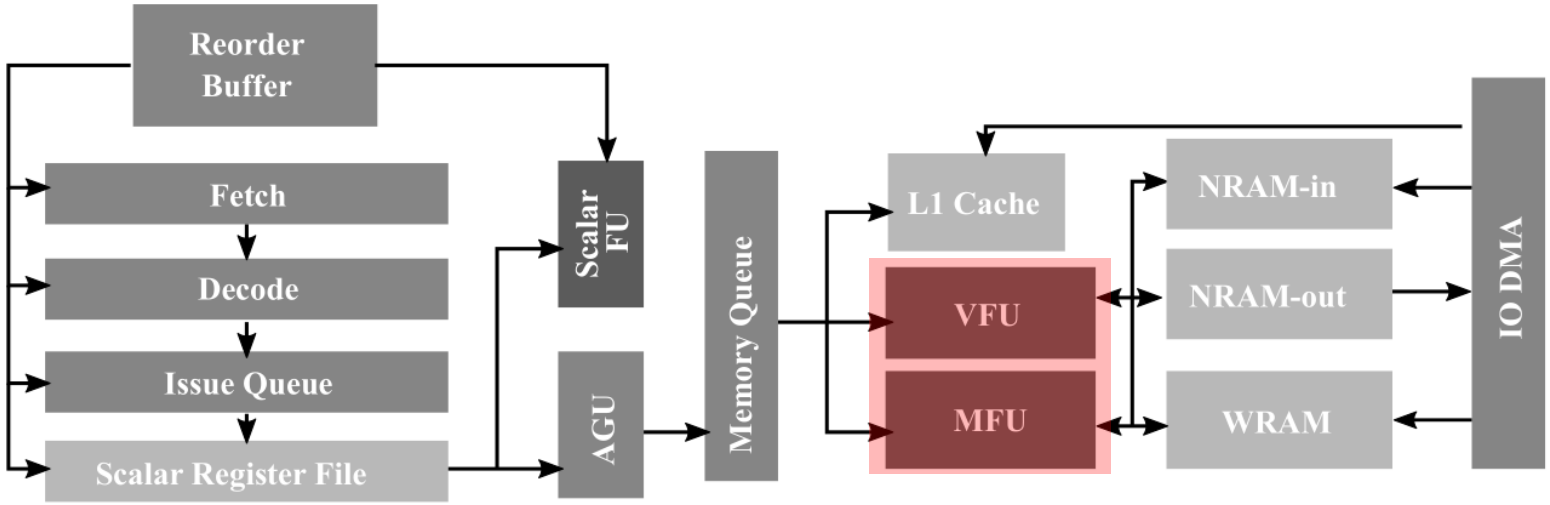
5.1 时域与空域架构

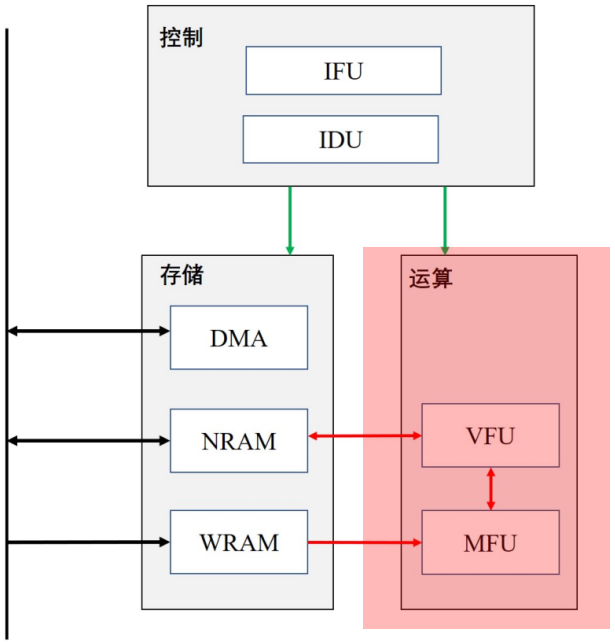
口 一种DLP基本架构

运算：

增加运算器中的操作，支持硬件高效执行的操作

低位宽运算器（量化），提高执行能效





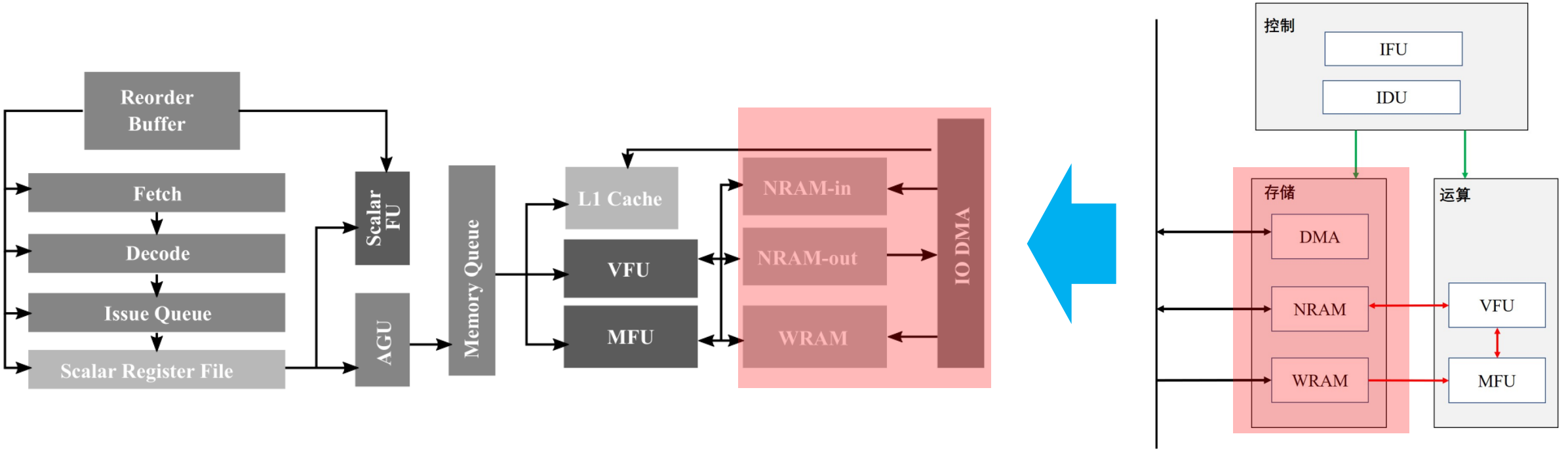
5.1 时域与空域架构

口 一种DLP基本架构

访存：

多级存储架构，降低访存延迟

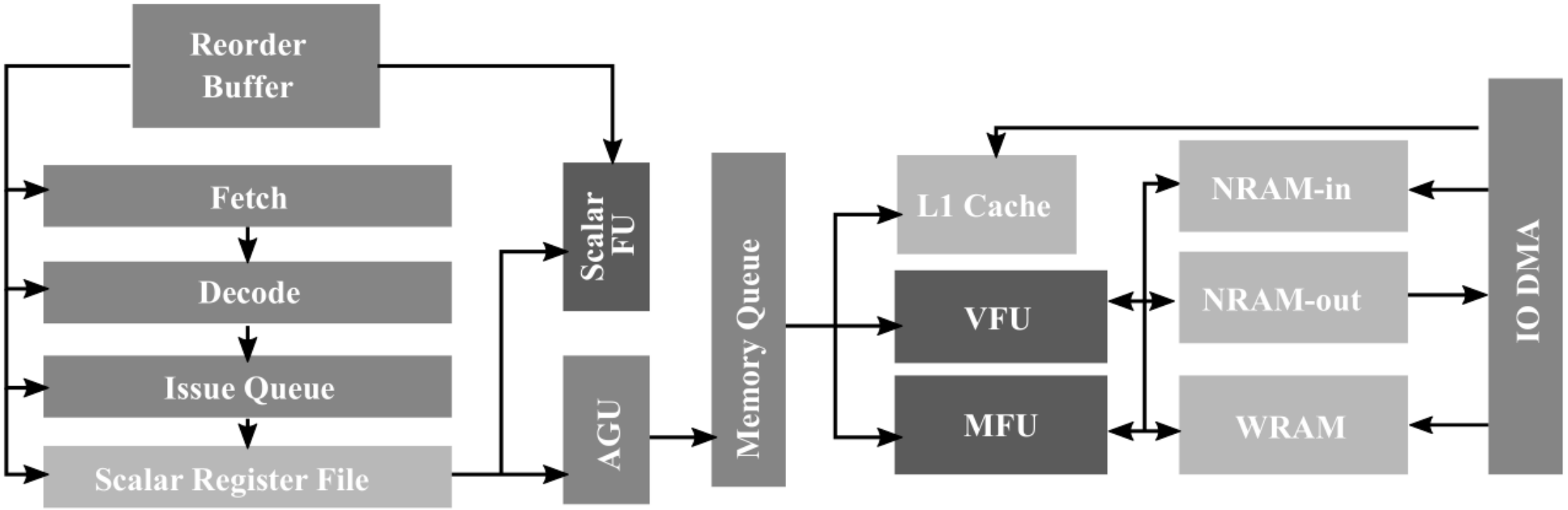
多通道存储架构，扩增访存带宽，减少冲突



5.1 时域与空域架构

□ 一种DLP基本架构

- 7段流水：取指、译码、发射、读寄存器、执行、写回、提交

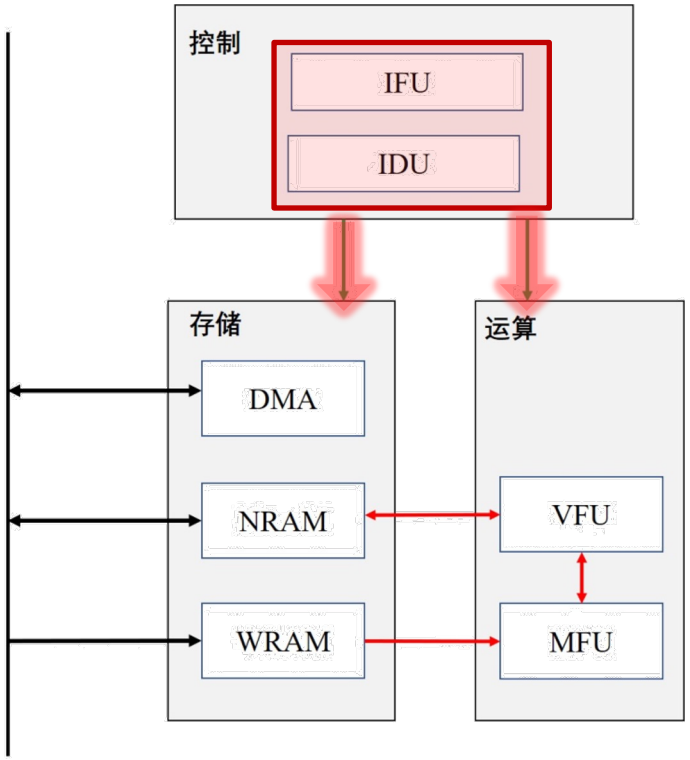


5.1 时域与空域架构

口 执行流程

Step #1：IFU 通过 DMA 从 DRAM 中读取程序指令，然后经过 IDU 进行译码后分发给DMA、VFU 和 MFU

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA



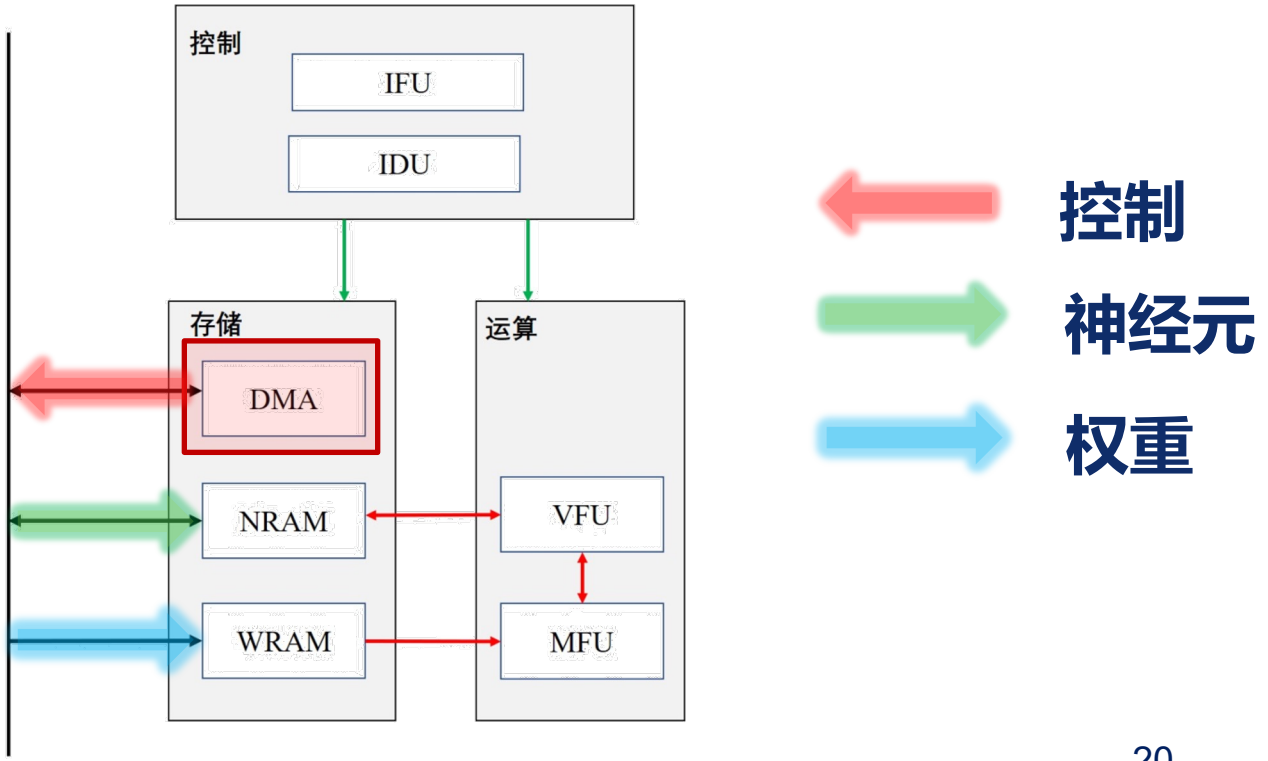
控制

5.1 时域与空域架构

口 执行流程

Step #2：DMA 接收到访存指令（读tensor指令，包括地址，数据量等信息）后从 DRAM 读取神经元tensor至 NRAM，读取权值tensor至 WRAM

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA

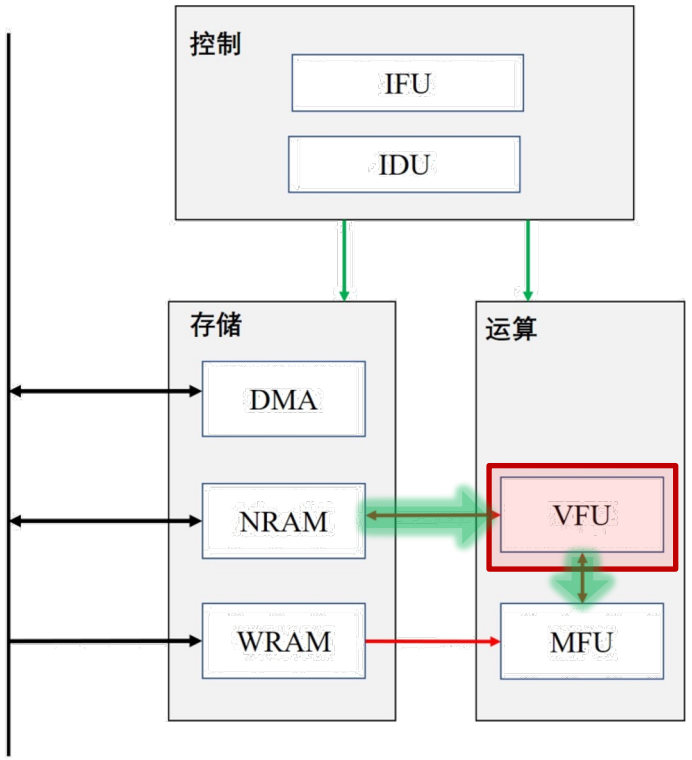


5.1 时域与空域架构

口 执行流程

Step #3：VFU 接收到指令后从 NRAM 中读取神经元tensor，并对神经元tensor进行预处理（如边界扩充等），然后发送给 MFU

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA

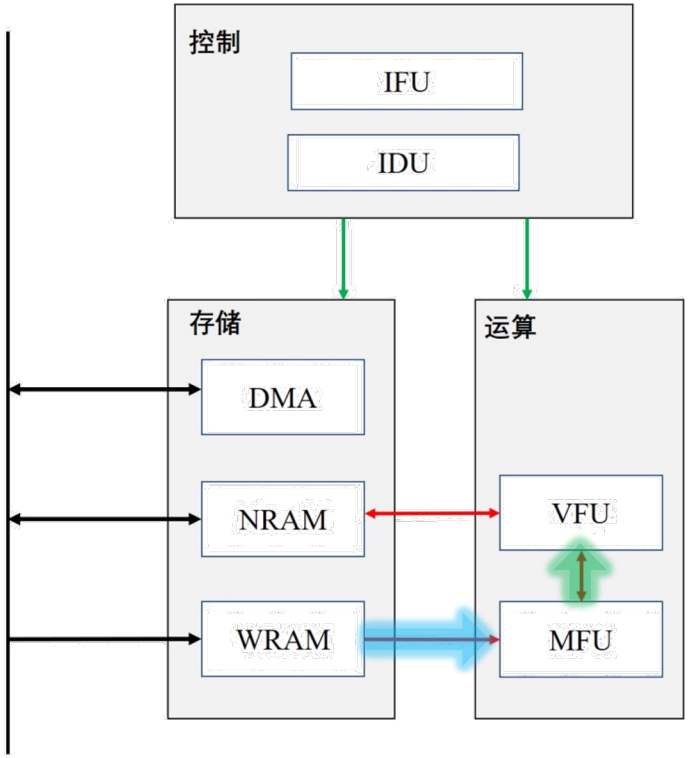


5.1 时域与空域架构

执行流程

Step #4：MFU 接收到指令后从 VFU 接收经过预处理的神经元tensor，并从 WRAM 中读取权重tensor，完成矩阵运算后将结果发送给 VFU

- 取指单元IFU
- 指令译码单元IDU
- 向量运算单元VFU
- 矩阵运算单元MFU
- 权重存储单元WRAMW
- 神经元存储单元NRAM
- 直接内存存取单元DMA

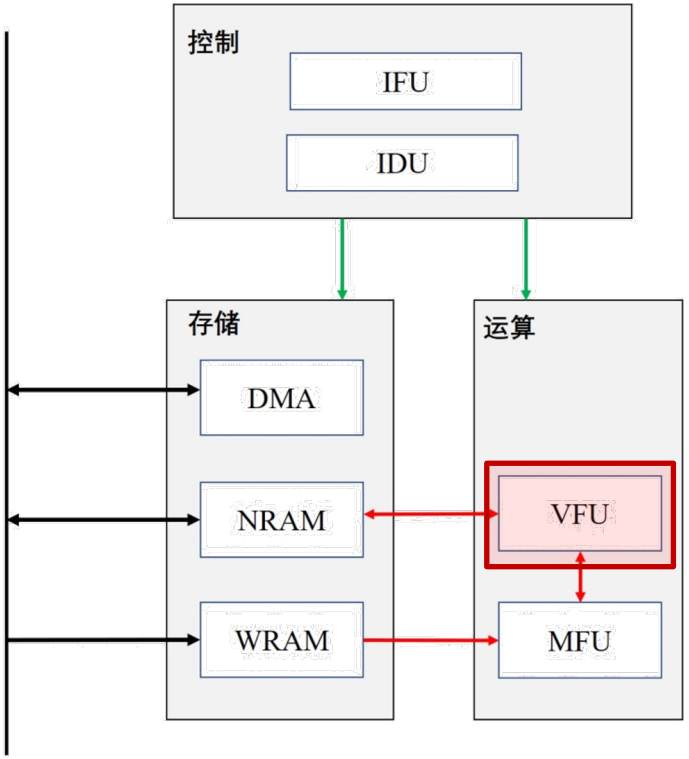


5.1 时域与空域架构

口 执行流程

Step #5：VFU 对输出神经元tensor进行后处理（如激活、池化等）

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA

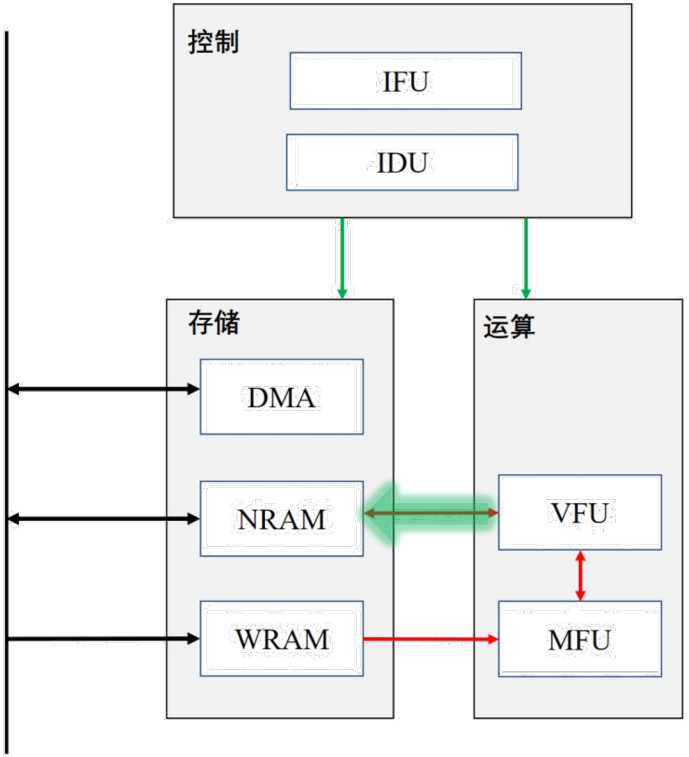


5.1 时域与空域架构

口 执行流程

Step #6：VFU 将运算结果tensor写回 NRAM

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA

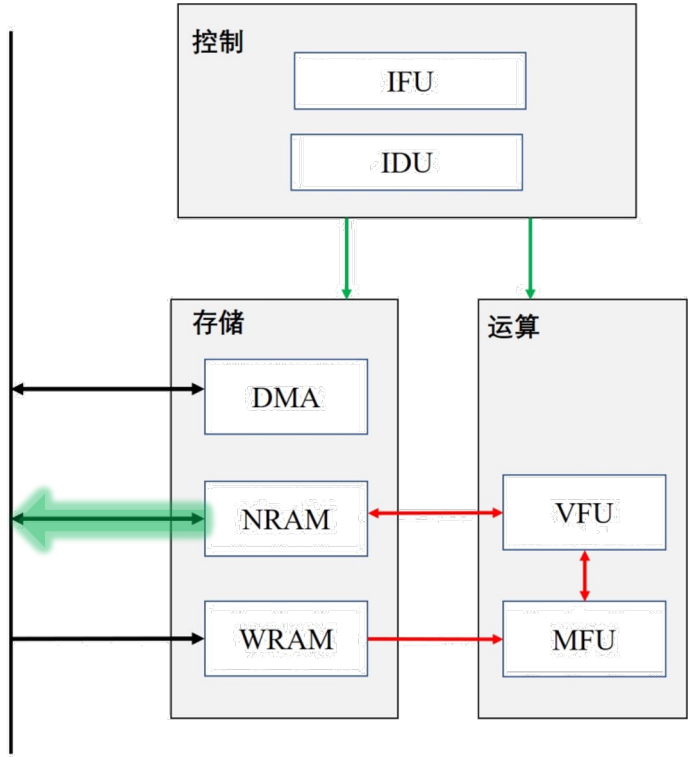


5.1 时域与空域架构

口 执行流程

Step #7：DMA 将输出神经元tensor从 NRAM 写回到 DRAM

- ◆ 取指单元IFU
- ◆ 指令译码单元IDU
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU
- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA



5.1 时域与空域架构

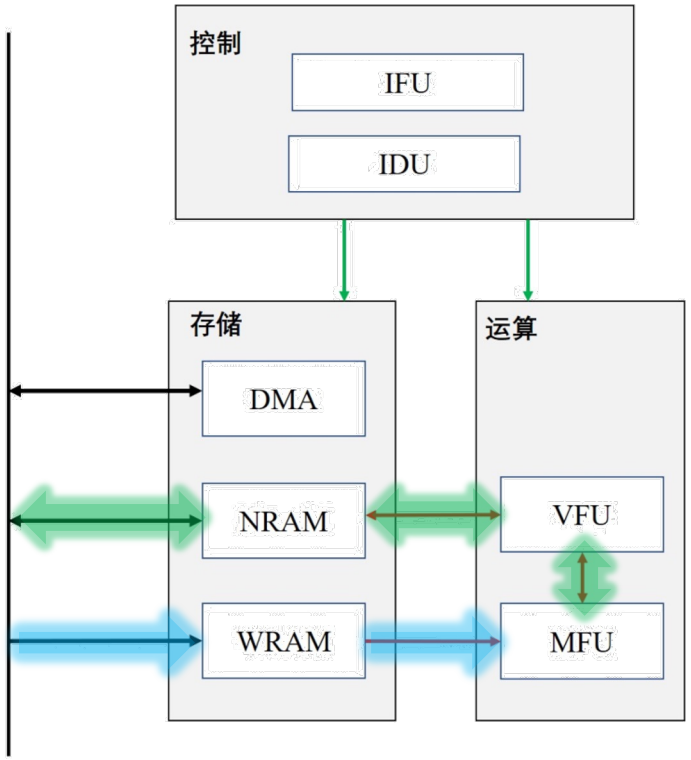
口 执行流程中的数据流

■ 神经元tensor数据流

DRAM->NRAM->VFU->（MFU->VFU->）NRAM->DRAM

■ 权值tensor数据流

DRAM->WRAM->MFU



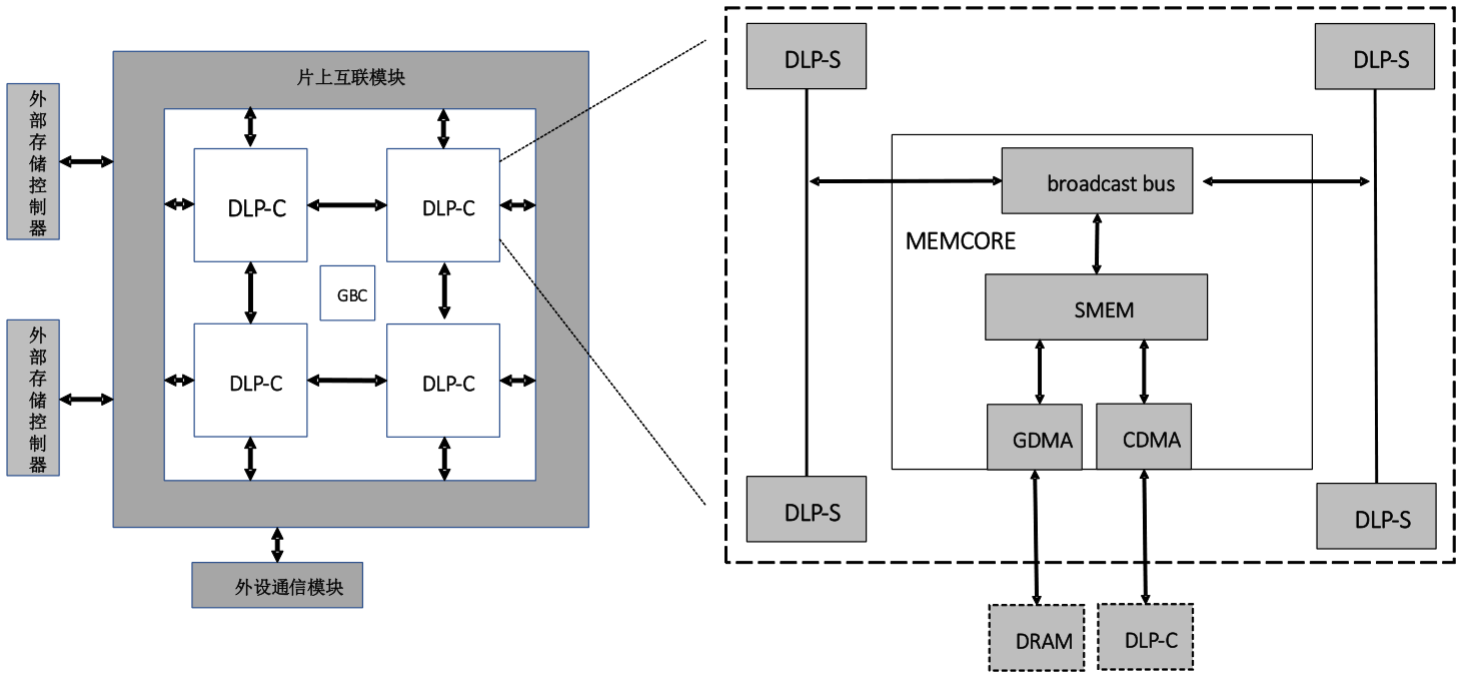
- ◆ 向量运算单元VFU
- ◆ 矩阵运算单元MFU

- ◆ 权重存储单元WRAMW
- ◆ 神经元存储单元NRAM
- ◆ 直接内存存取单元DMA

5.1 时域与空域架构

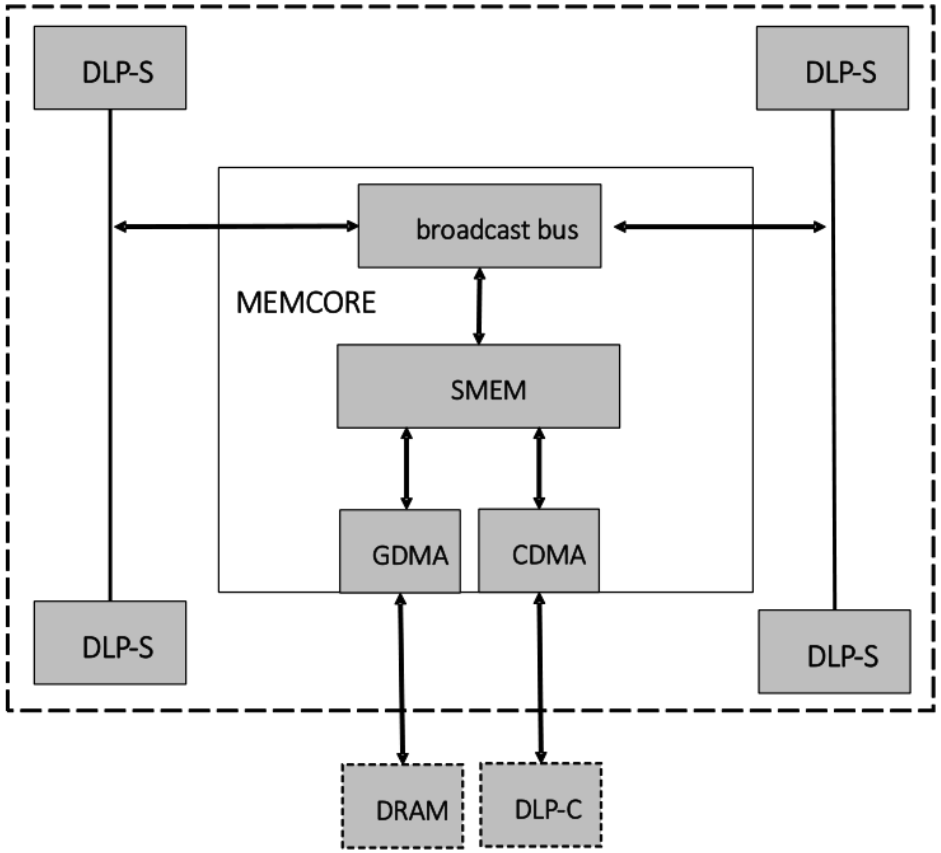
□ 多核处理器DLP-M分层架构

- ■ 一个DLP-M由多个DLP-C构成
- ■ 一个DLP-C（簇）由多个DLP-S (单核) 构成
- ■ 分层结构设计-减少NoC的负载核开销



5.1 时域与空域架构

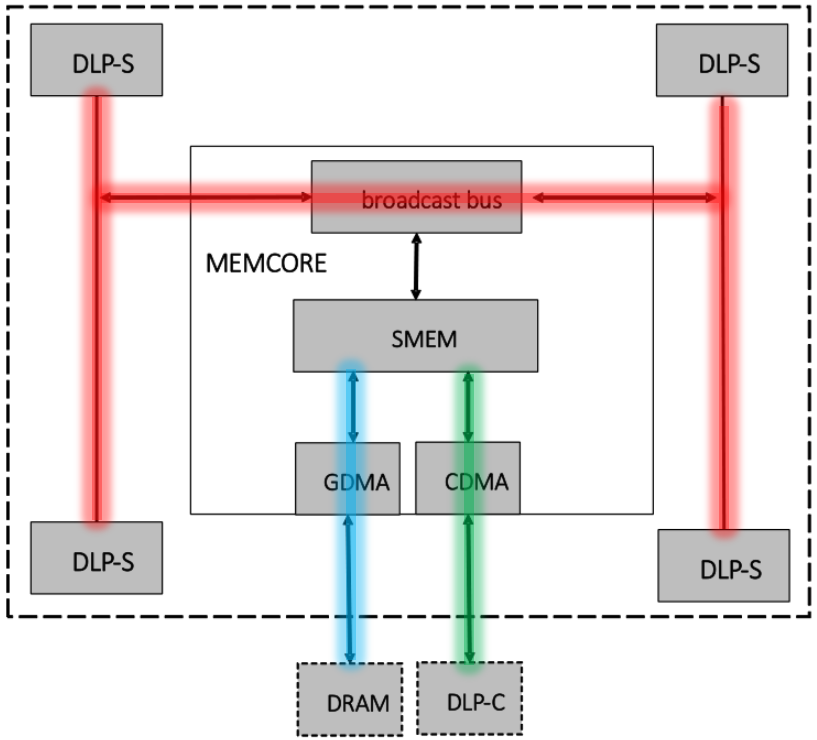
- DLP-C架构
  - 四个DLP-S
  - 存储核MEMCORE
  - ◆存储：DLP-S共享数据
  - ◆通信：DLP-C与片外DRAM，DLP-C之间，多个DLP-S之间



5.1 时域与空域架构

MEMCORE架构

- 共享存储模块SMEM（Shared Memory）
- 广播总线（Broadcast Bus）
- Cluster 直接内存访问CDMA（Cluster Direct Memory Access）
- 全局直接内存访问GDMA（Global Direct Memory Access）



5.1 时域与空域架构

- DLP-M架构
  - 外部存储控制器
  - 外设通信模块
  - 片上互联NoC模块
  - 同步模块GBC（Global Barrier Controller）
  - 四个DLP-C