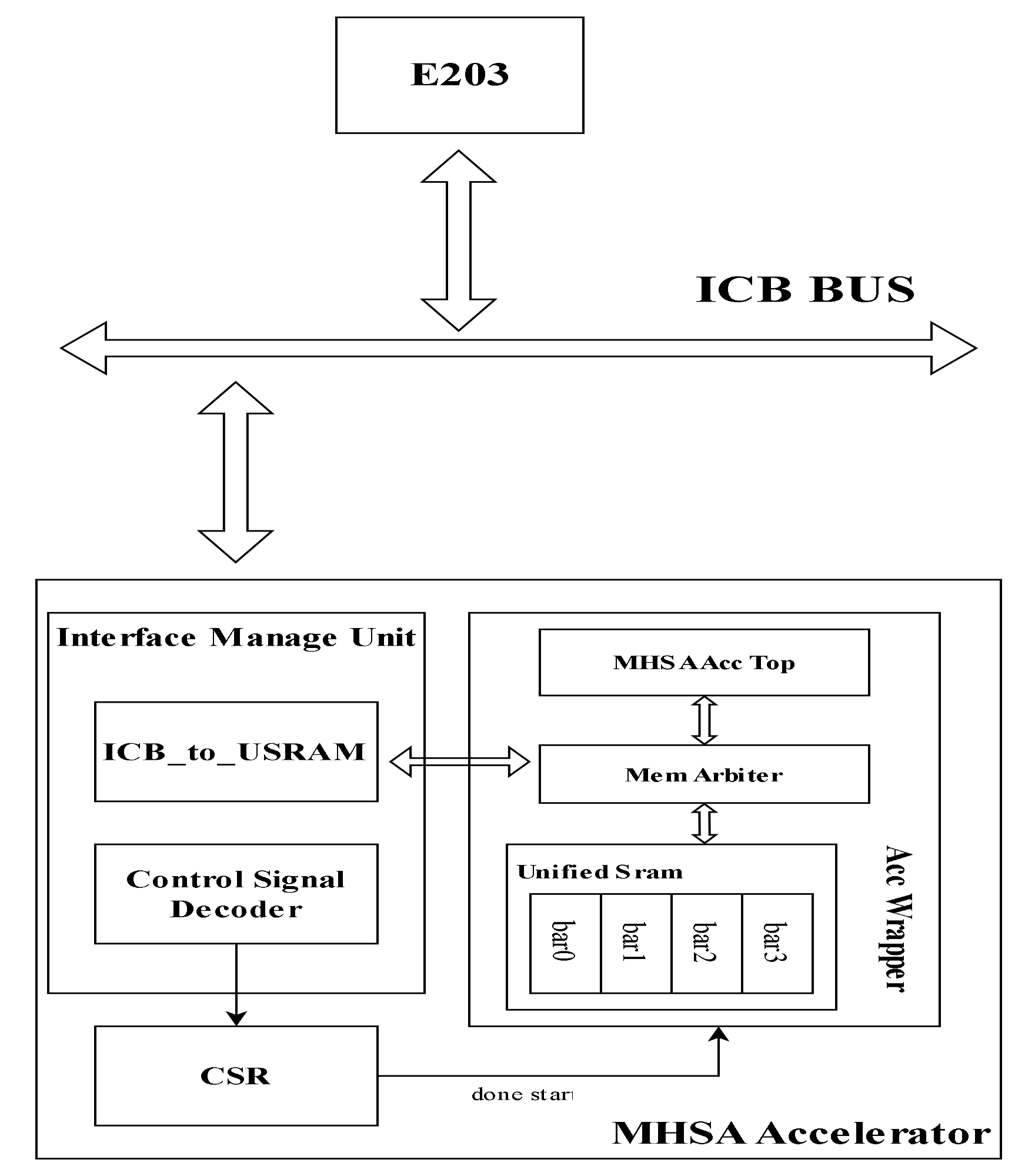
E203-MHSA-Accelerator SPEC

|  |  |
| --- | --- |
| 版本 | V1 |
| 作者 | 汪子尧 |
| 邮箱 | [wangziyao1@sjtu.edu.cn](mailto:Jiangjianfei@sjtu.edu.cn) |
| 修改时间 | 2025.4.3 |

# 系统架构

E203-MHSA-Accelerator是一款基于平头哥E203 RISC-V处理器自主开发的硬件加速模块。该模块作为独立外设直接挂载在E203 SoC的ICB总线上，专门针对Transformer模型中的多头自注意力（MHSA）计算单元进行硬件加速。该方案通过在SoC系统中集成专用硬件加速模块，针对当前Transformer模型在边缘设备部署时面临的计算瓶颈问题，提出了轻量化异构计算解决方案。



# **加速器架构设计**

为提升加速器系统的硬件兼容性，外设采用标准化的接口控制体系。核心组件Interface Management Unit（IMU）负责桥接E203主控核的ICB总线协议与加速器内部逻辑，该单元将ICB总线输入的读写请求实时解析为两类信号：面向SRAM存储体的Unified Sram Interface访存通道，以及包含启动/状态监测的硬件控制通道。Accelerator Wrapper接口信号如表1所示：

表1 加速器顶层接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号类型 | 信号名称 | 方向 | 位宽 | 描述 |
| Global signal | clk | input | 1 | 系统时钟信号 |
| rst\_n | input | 1 | 低有效异步复位信号 |
| Control signal | done | output | 1 | 加速器计算完成标志 |
| start | input | 1 | 加速器启动触发信号 |
| input\_base | input | [31:0] | 输入数据基地址（32位地址空间） |
| output\_base | input | [31:0] | 输出数据基地址（32位地址空间） |
| Unified Sram Interface | soc\_write\_en | input | 1 | SRAM接口写使能信号： 1 - 写操作 0 - 读操作 |
| soc\_data\_in | input | [WIDTH-1:0] | 输入数据总线（位宽由WIDTH参数定义） |
| soc\_addr | input | [31:0] | 统一寻址总线（32位地址空间） |
| soc\_data\_out | output | [WIDTH-1:0] | 输出数据总线（位宽由WIDTH参数定义） |

考虑到主机端与加速器运算单元都存在对Unified Sram的访问，这里增加仲裁单元Mem Aribiter。

* **加速器运行流程：**

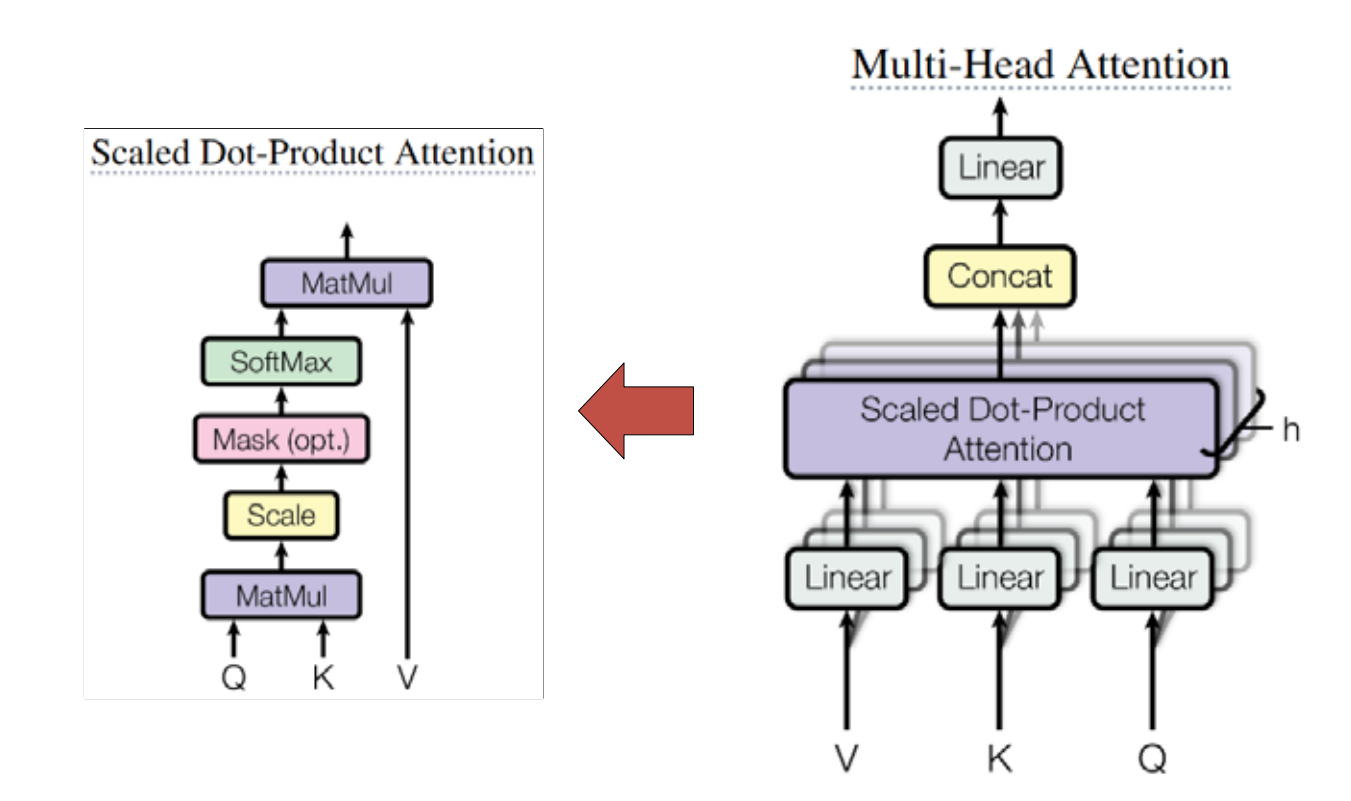
1. 主机端完成任务定义与划分
2. 主机端通过DMA或DIRECT-IO的方式通过ICB总线向加速器Unified Sram写入初始数据（输入参数/输入权重）
3. 主机端通过配置加速CSR使能加速器
4. 加速器进行运算
5. 主机间隔一定时间轮询读取加速器状态寄存器结束位、

* **Memory Map**

|  |  |  |
| --- | --- | --- |
| Offset Address | Memory Type | 功能描述 |
| 0x00000000 | Control Register | 写1启动计算，自动清零 |
| 0x00000004 | Status Register | 只读状态寄存器，bit0=done |
| 0x00000008 | Configuration Register | 输入数据起始地址 |
| 0x0000000C | Configuration Register | 输出数据起始地址 |
| 0x00001000-0x10000000 | Data Buffer | 统一编址的SRAM存储体 |
| 0x10000000-0xFFFFFFFF | Reserved Area | 保留地址空间 |

# 计算原理

# 基本计算原理



MHSA（Multi-Head Self-Attention，多头自注意力）计算是Transfomer中重要计算单元。基本计算原理简化为以下三个步骤：

1、输入投影与分解

特征维度（B，L，C）的输入向量X通过三组可训练参数矩阵（WQ/WK/WV）生成查询（Q）、键（K）、值（V）三元组，沿特征维度拆分为h个子矩阵。

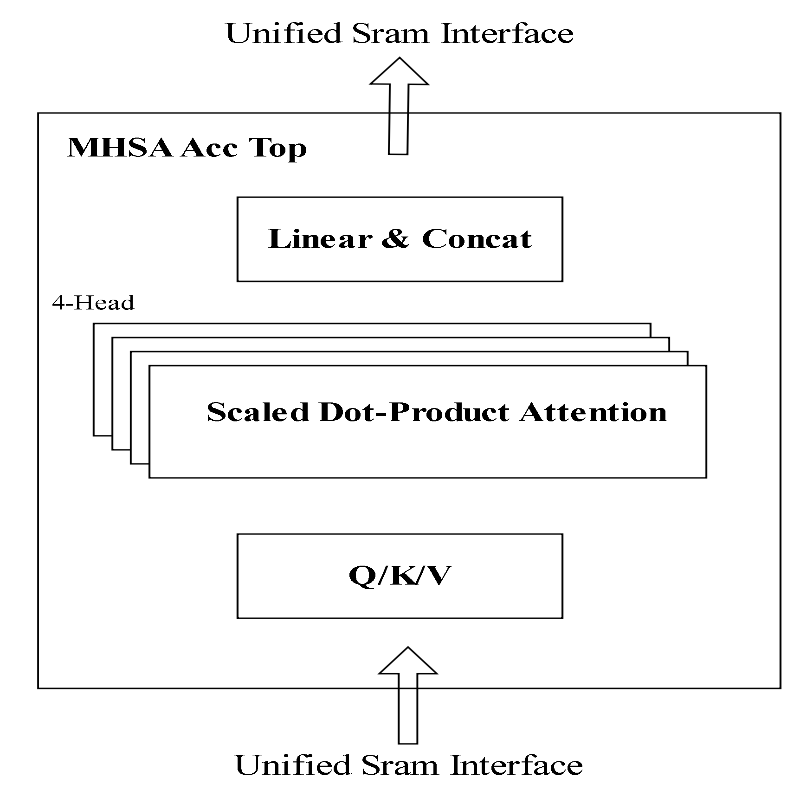
1. 注意力权重生成

3、加权融合与重构

注意力权重与V矩阵相乘后，通过拼接单元将h个子头输出沿特征维度重组，最终线性变换层后输出。

# 加速计算原理

硬件架构上，加速器采用三级流水线结构实现MHSA核心运算：第一级完成序列拆分与Q/K/V矩阵计算，第二级执行Scaled Dot-Product Attention的并行计算，第三级处理多头结果的拼接与线性变换。



在MHSA加速器设计中，我们重点关注了计算过程中最耗时的部分，即Q/K/V矩阵计算以及注意力计算。对于输入特征维度（1，32，128）的情况，**Q/K/V三路矩阵乘运算占据整体计算量62%**。为了加速这一关键步骤，我们采用了脉动阵列架构来实现矩阵乘法的高效计算。具体而言，Q/K/V矩阵乘法的维度为（32，128）×（128，32），而注意力计算的矩阵乘法维度为（8，32）×（32，8）。因此，我们设计了一个8×8规模的脉动阵列作为最基本的矩阵乘运算单元，对于大规模矩阵，可以进行矩阵分块，使用脉动阵列分时复用完成运算。此外，考虑乘法器资源，8×8规模的脉动阵列共利用64个乘法器资源，因此我们在例化6个脉动阵列的情况下，乘法器资源总数仍少于480，符合设计初衷。

# 近似加速比

由于加速器整体加速比需要考虑访存与脉动阵列调度策略等其他多方面因素，这一部分将在后续设计中逐步完善，这里仅考虑脉动阵列对Q/K/V矩阵的近似加速比。考虑类似于CPU进行运算的全折叠的矩阵乘运算方式，对于单个（8，128）×（128，8）的矩阵乘，需要的时钟周期数为：

Clock Cycle ( CPU ) = 128×8×8 = 8192

MHSA加速器中，可以使用8×8的脉动阵列的方式对其进行加速，总共需要的时钟周期数为：

Clock Cycle ( ACC ) = 7 + 128 + 8 = 143

理论加速比为：

Acc Ratio = 8192 / 143 ≈ 57