

项目说明version2

项目借鉴 TSMC 在 2023 年 IEEE 国际固态电路会议 (ISSCC 2023) 中提出的基于 SRAM 的数字存算一体计算宏 (DCIM macro) 思想，需要实现一种支持可变位宽与并行操作的高效计算架构，可处理 12/24b 整数权重与 12/24b 整数输入。

电路架构

- 核心组成：电路由 SRAM 存储阵列、局部乘累加电路 (LMAC)、全局 IO (Global IO)、读字线驱动 (RWLDRV) 和全局控制器 (GCTRL) 构成。其中，SRAM 阵列预加载权重；LMAC 包含按位乘法器与加法树，完成 1b 输入与 12b 权重的乘累加；Global IO 集成移位寄存器、27b 加法器与 51b 移位累加器，处理多周期结果累加；GCTRL 通过 WWIDTH 和 INWIDTH 信号配置权重与输入位宽。
- 处理流程：1) 权重经 WBL 传输至 LMAC，与串行输入的 XIN 通过 OAI 执行乘法；2) 加法树累加 8 个 1b×12b 乘积结果；3) Global IO 对多周期结果移位累加，最终得到 8×12/24b 输入与 8×12/24b 权重的乘累加结果；4) 16 个相同模块并行工作，实现 16×8×12/24b 的全局计算。
- 位宽配置：支持 12/24b 权重 (WWIDTH 信号控制) 与 12/24b 输入 (INWIDTH 信号控制)，通过符号扩展机制兼容有符号运算：前 4 个周期执行 4b 有符号操作，其余周期为 4b 无符号操作，确保位宽切换时的精度一致性。
- RTL 设计：基于 Verilog 完成模块化设计，顶层模块 (top) 包含存算阵列 (cim_array)、数字逻辑 (digital_circuit) 等子模块，其中 cim_array 负责存储与基础运算，global_io 处理累加与输出，rwldrv 实现输入信号串并转换，gctrl 生成控制时序与状态信号。
- 同时 MAC 与权重更新：采用 ping-pong 结构，阵列设计为 2rows，通过行选择逻辑实现并行的权重写入与 MAC 操作（一个行用于写操作时，另一个行可同时进行计算）。

RTL架构设计

- 顶层结构：以 top.v 为核心，包含存算阵列 (cim_array.v)、数字逻辑 (digital_circuit.v) 等子模块，层级关系如下：
- plaintext

代码块

```
1  top.v
2  |—— cim_array.v (含cim_bank.v)
3  |—— digital_circuit.v
```

```
4      |—— cim_array_ctrl.v (阵列控制)
5      |—— global_io.v (全局IO, 含累加器)
6      |   |—— add.v
7      |   |—— accumulator.v (含se_cla.v、s_cla.v)
8      |—— local_mac.v (局部MAC, 含oai_mult.v)
9      |—— rwldrv.v (读字线驱动)
10     |—— gctrl.v (全局控制)
```

处理流程实现

1. 权重加载: 外部通过 D 和 WA 端口将权重数据写入到 cim_array 中的 cim_bank。cim_array_ctrl负责将地址和数据路由到正确的 cim_bank。
2. 输入加载: 外部输入向量通过 xin0 端口送入系统。
3. 计算开始:
 - gctrl 模块开始计数, 生成 sel 信号。
 - rwldrv 根据 sel 信号, 从 xin0 中选择一小部分位, 生成行驱动信号 rwlb_row0 和 rwlb_row1。
 - cim_array 中的 cim_bank 被 rwlb 信号激活, 输出存储的权重位到 wb 总线。
 - local_mac 模块接收来自 cim_array 的权重位 (wb*) 和来自 rwldrv 的输入位 (rwlb_*), 通过内部的 oai_mult 阵列和加法器树, 计算出一个部分的乘累加结果 mac_out。
4. 结果处理:
 - global_io 模块接收两个 local_mac 的输出 macout_a 和 macout_b, 进行一次加法/减法操作。
 - accumulator 模块将 global_io 的输出进行累加。gctrl 产生的 st 信号控制累加的开始和结束。
5. 输出: 最终的累加结果通过 nout 端口输出。这个过程会根据 inwidth 的设置, 循环执行 12 或 24 次, 以完成一次完整的向量-矩阵乘法。