CIM-PQC：存算一体电路

本项目实现了一种**基于SRAM的数字存算一体计算宏（DCIM macro）**，采用TSMC在ISSCC 2023中的思想[1]，可以在单个宏中处理可变的12/24b的整数权重与12/24b的整数输入。

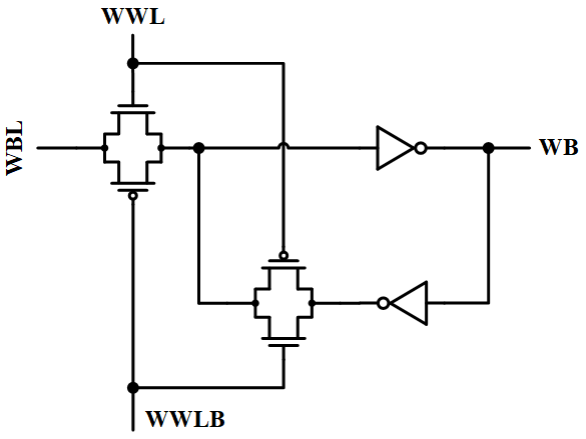
本文介绍了电路的具体实现，将从存算一体阵列、架构、实现、后端4个方面展开。

1. 存算一体阵列

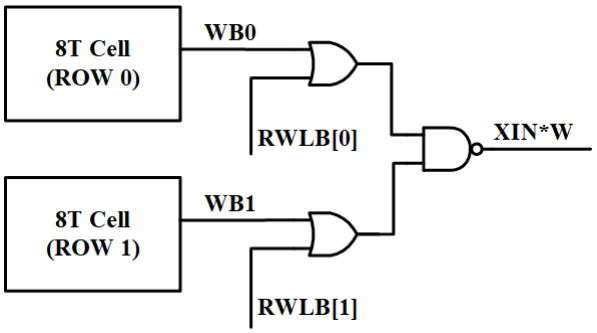
位单元

本项目中采用了**8T 2b OAI (or-and-invert)**单元，基于**SRAM DCIM**范式。

存储单元使用了紧凑的**8T方案**，省去了单独的读出逻辑，减少了总晶体管数量与信号数量。



进一步地，引入了**ping-pong**结构，阵列具有偶数行，通过设计行选择逻辑，可以实现**并行的权重写入与MAC操作**。行选择信号由读字线驱动器（RWLDRV）产生，并通过RWLB传递给OAI。



具体而言，该结构采用了**2个8T cell和1个OAI**：

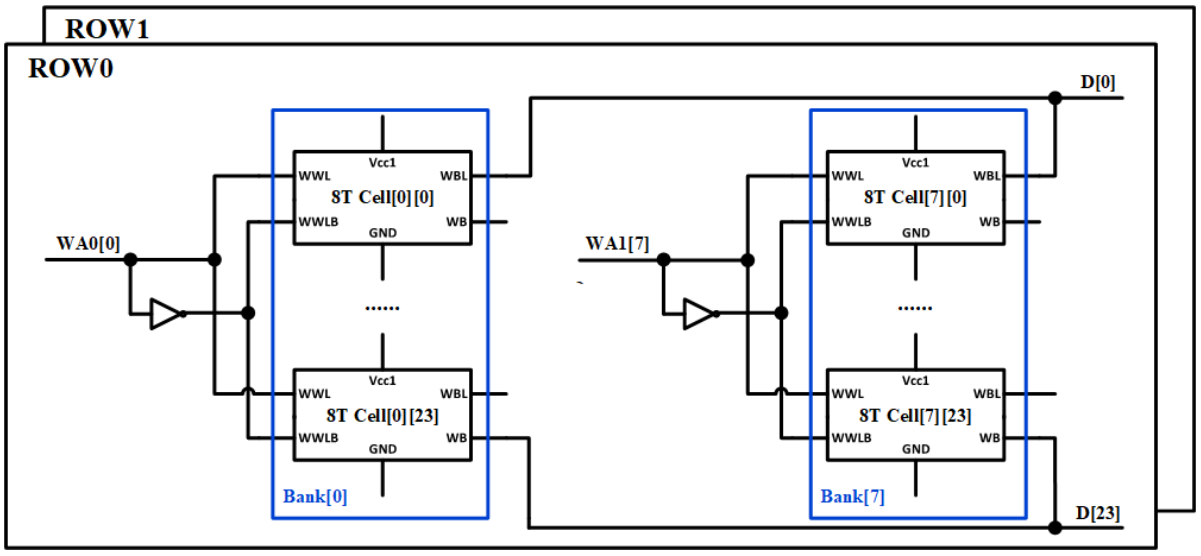
- 8T单元：作为存储器，进行数据存储和行选择写操作。
- OAI：对MAC操作进行行选择与按位乘法。

自此，上述位单元实现了1b的权重存储。

阵列搭建

根据4-step NTT的拆分思想，本项目中所需的最小阵列规模为16 * 8 * 12/24b。

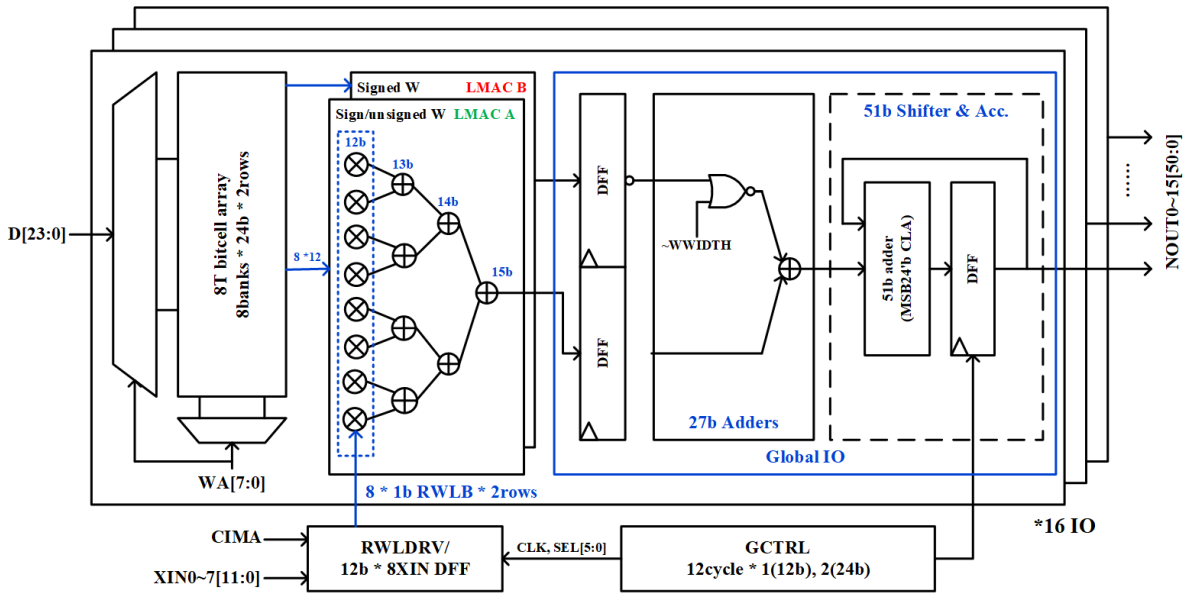
这里设置m = 16, n = 8，搭建了8banks * 24b * 2rows的存算一体阵列，结构如下：



阵列的输入输出信号如下：

```
module cim_array(
    input [23:0] D1,
    input [7:0] WA0, WA1,
    output [95:0] wb0_a, wb1_a,
    output [95:0] wb0_b, wb1_b
);
```

2. 电路架构



DCIM支持**12/24b XIN**和**12/24b W**的MAC，具有16个输出通道（m = 16）。

电路组成

电路主要由以下几个部分组成：

- **SRAM存储阵列（8T bitcell array）**：存储权重，权重通过写端口在MAC前预加载（同正常SRAM的写操作），由WBL输出至LMAC
- **局部乘累加电路（Local MAC）**：完成12b权重与1b输入的乘法，每个LMAC包含8个按位乘法器、加法树中的13~15b加法器
- **全局IO（Global IO）**：括移位寄存器、27b加法器和51b移位累加器，前者完成12b/24b权重的累加，后者完成12b/24b输入的累加
- **读字线驱动（RWLDRV）**：将XIN并行信号转变为串行信号，在与地址解码信号（CIMA）结合后传播到RWLB，输入LMAC
- **全局控制器（GCTRL）**：配置全局时钟与位选信号，设置MAC规模，WWIDTH指定当前W位宽，INWIDTH控制XIN位宽

此外，还引入了符号扩展进位超前加法器（signed-CLA）和加法树流水线来提高吞吐率。

处理过程

当前设置的计算规模为`16 * 8 * 12/24b W` X `8 * 1 * 12/24b XIN`，在1次计算过程中：

1. 权重经由WBL输出至LMAC，与串行输入的XIN信号执行乘法操作（通过OAI实现），得到8个`1b XIN`与`12b W`的乘积结果。
2. 加法树对上述乘积进行累加，得到`8 * 1b XIN`与`8 * 12b W`的乘累加结果。
3. 2个LMAC输出的数据经由移位寄存器与加法器，得到`8 * 1b XIN`与`8 * 12/24b W`的乘累加结果。
4. 移位累加器将多周期的XIN串行信号与W的MAC结果累加，得到`8 * 12/24b XIN`与`8 * 12/24b W`的乘累加结果。
5. 整个电路由16个这样的模块组成，每个模块的W不同，XIN相同，从而实现`16 * 8 * 12/24b XIN`与`8 * 12/24b W`的乘累加。

3.电路实现

电路前端搭建：基于Cadence Virtuoso，存算一体阵列直接通过原理图搭建，外围数字逻辑电路通过Verilog进行了RTL设计。

RTL部分

进行了时序验证，实现 `8 * 12/24b XIN` 与 `8 * 12/24b W` 的MAC，cim_array.v为SRAM阵列的逻辑替代，不会被例化在Schematic中

```
top.v
├── cim_array.v
|   └── cim_bank.v
└── digital_circuit.v
    ├── cim_array_ctrl.v
    ├── global_io.v
    |   ├── add.v
    |   └── accumulator.v
    |       └── se_cla.v
    |           ├── s_cla.v
    |           └── add.v
    ├── local_mac.v
    |   ├── add.v
    |   └── oai_mult.v
    ├── rwldrv.v
    └── gctrl.v
```

顶层模块 (top)

```
module top(
    input [23:0] D,
    input clk,rstn,cima,acm_en,
    input [7:0] WA,
    input inwidth,
    input wwidth,
    input start,
    input [191:0] xin0,
    output [50:0] nout,
    output wire st
);
```

子模块

- **global_io**
    - 功能：全局输入输出控制

```
module global_io(
    input [14:0] macout_a,
    input [14:0] macout_b,
    input clk,acm_en,rstn,
    input st,
    input wwidth,
    output [50:0] nout
);
```

- **cim_array_ctrl**
  - 功能：CIM阵列控制，生成处理后的数据和地址信号

```
module cim_array_ctrl(
    input [23:0] D,
    input [7:0] WA,
    input clk, rstn, cima,
    output reg [23:0] D1,
    output reg [7:0] WA0, WA1
);
```

- **cim_array**
  - 功能：CIM存储阵列

```
module cim_array(
    input [23:0] D1,
    input [7:0] WA0, WA1,
    output [95:0] wb0_a, wb1_a,
    output [95:0] wb0_b, wb1_b
);
```

- **local_mac (lmaca 和 lmacb)**
  - 功能：两个本地乘累加器（MAC），lmaca的sus固定为0，lmacb的sus为~wwidth

```
module local_mac(
    input [95:0] wb0,
    input [95:0] wb1,
    input [7:0] rwlb_row1,
    input [7:0] rwlb_row0,
    input sus,
    output wire [14:0] mac_out
);
```

- **rwldrv**
  - 功能：行驱动控制，生成行选择信号

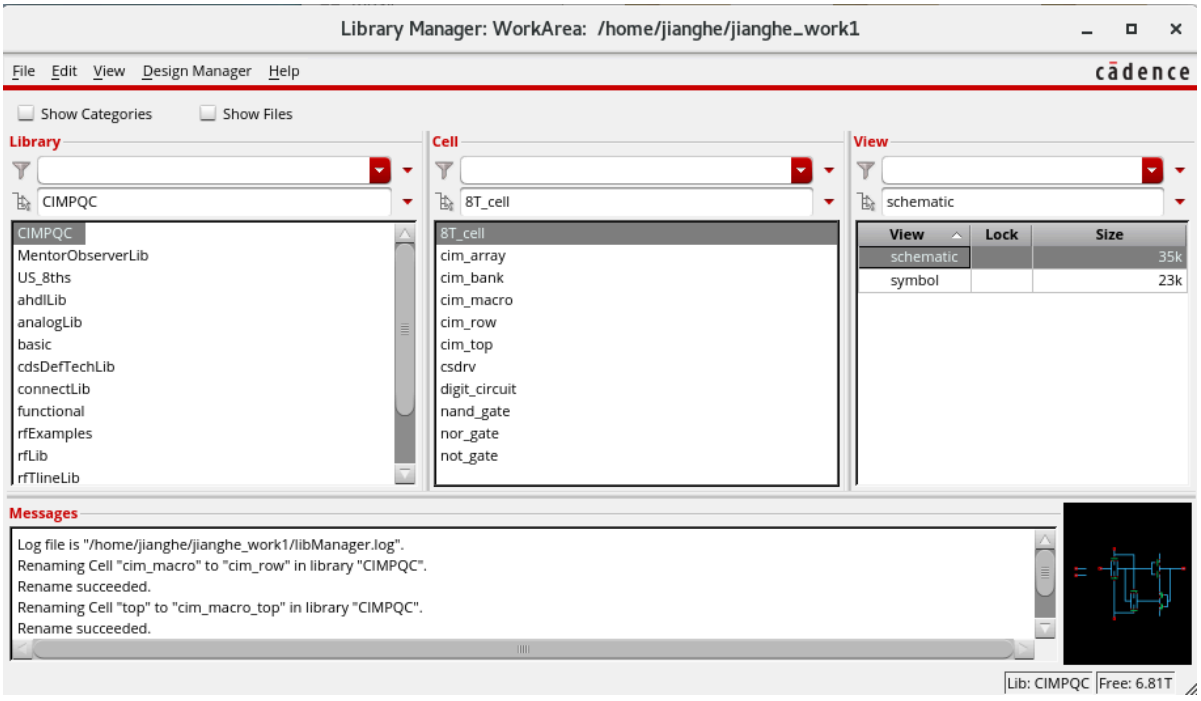
```
module rwldrv(
    input cima,
    input [191:0] xin,
    input clk,rstn,
    input inwidth,
    input [5:0] sel,
    output reg [7:0] rwlb_row1,
    output reg [7:0] rwlb_row0
);
```

- **gctrl**
  - 功能：全局控制逻辑，生成选择信号和状态信号

```
module gctrl(
    input inwidth,
    input clk,rstn,
    input start,
    output reg [5:0] sel,
    output reg st
);
```

Schematic

Library内模块如下：

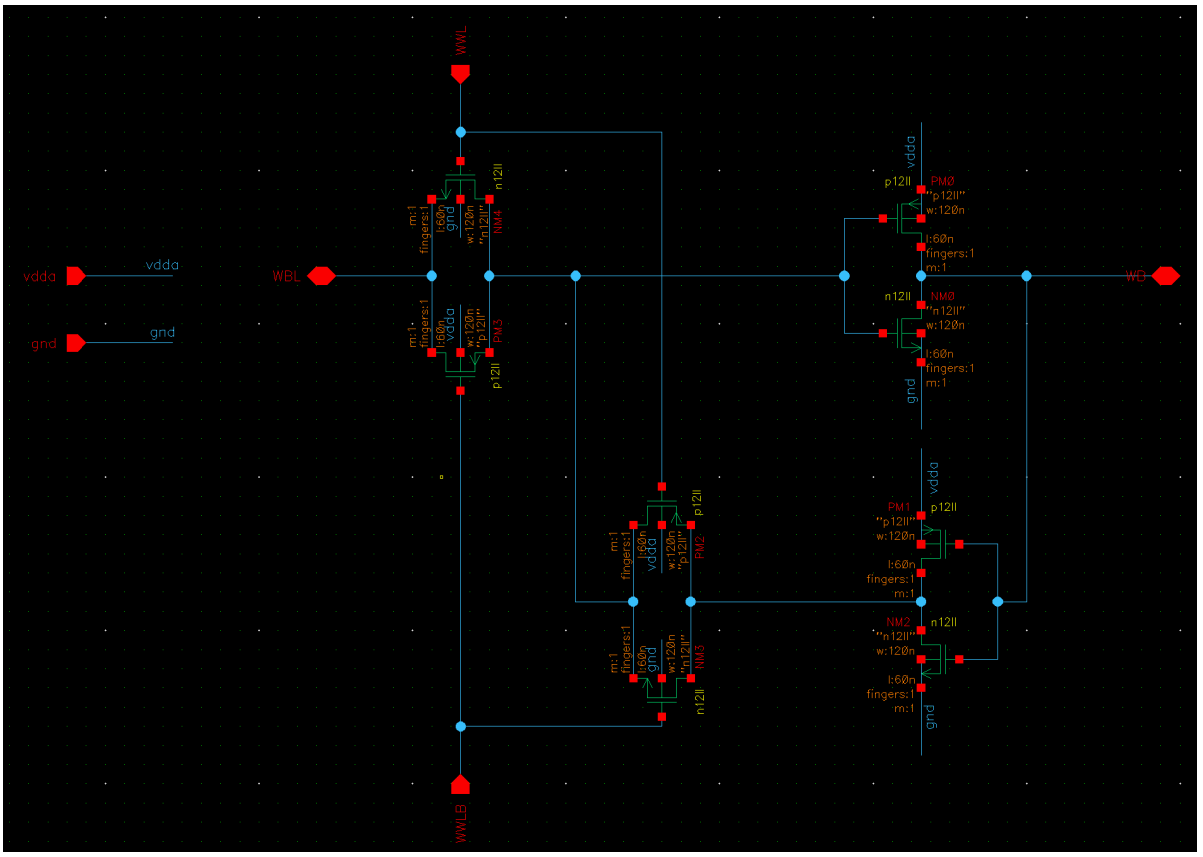


- **8T_cell**：存储1-bit数据

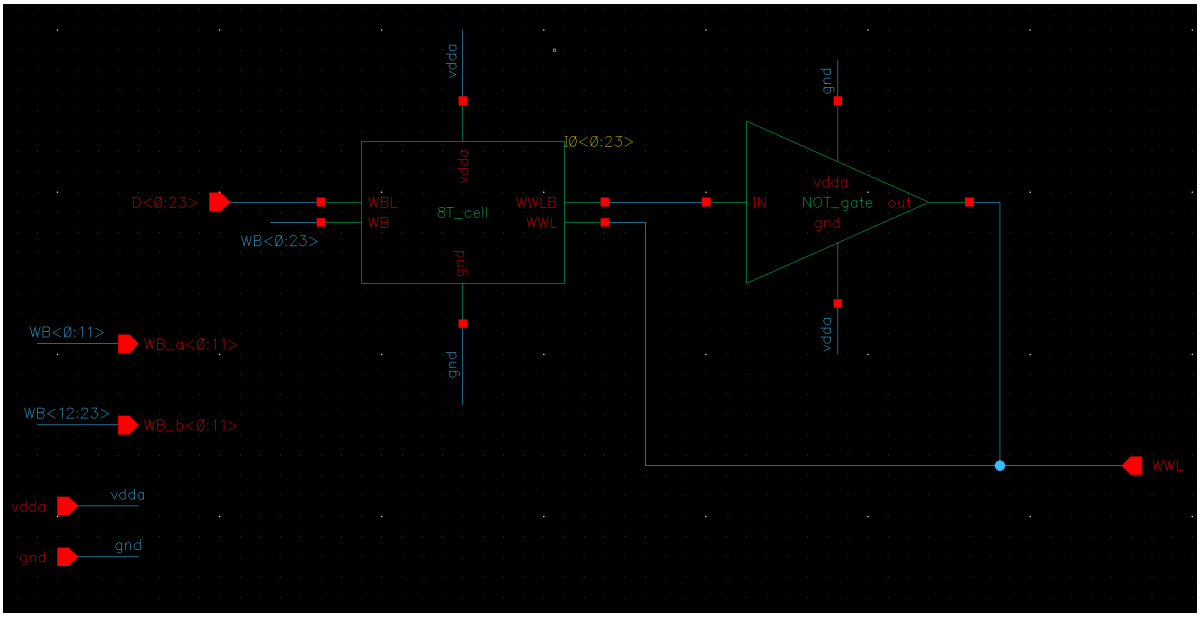
由于该部分仍为数字电路，晶体管尺寸选择默认，未进行修改

工艺库：smic55ll_121825

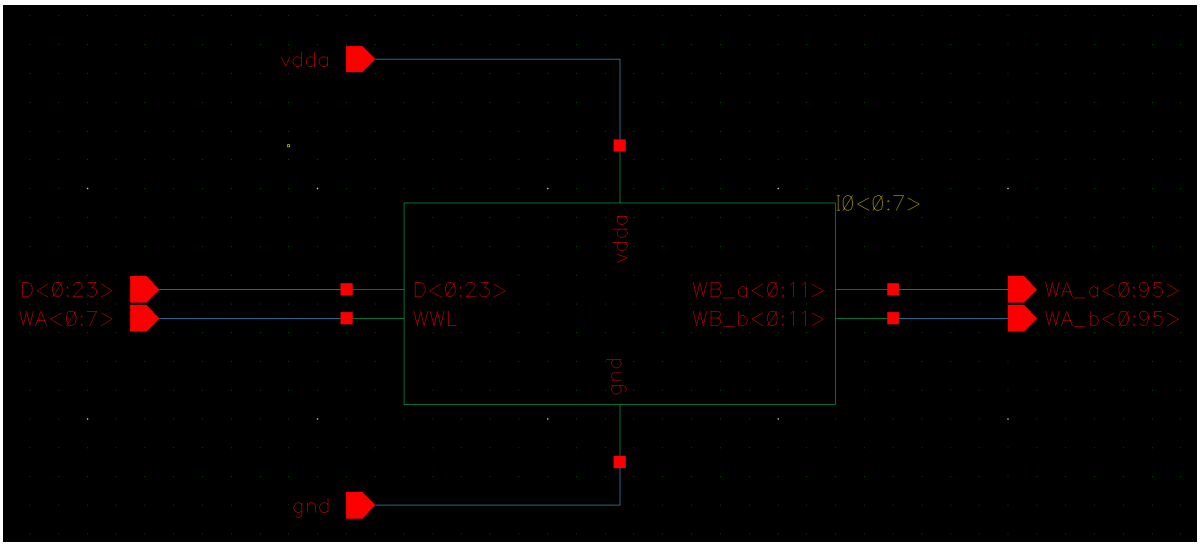
Instance：NMOS（n12ll），PMOS（p12ll）



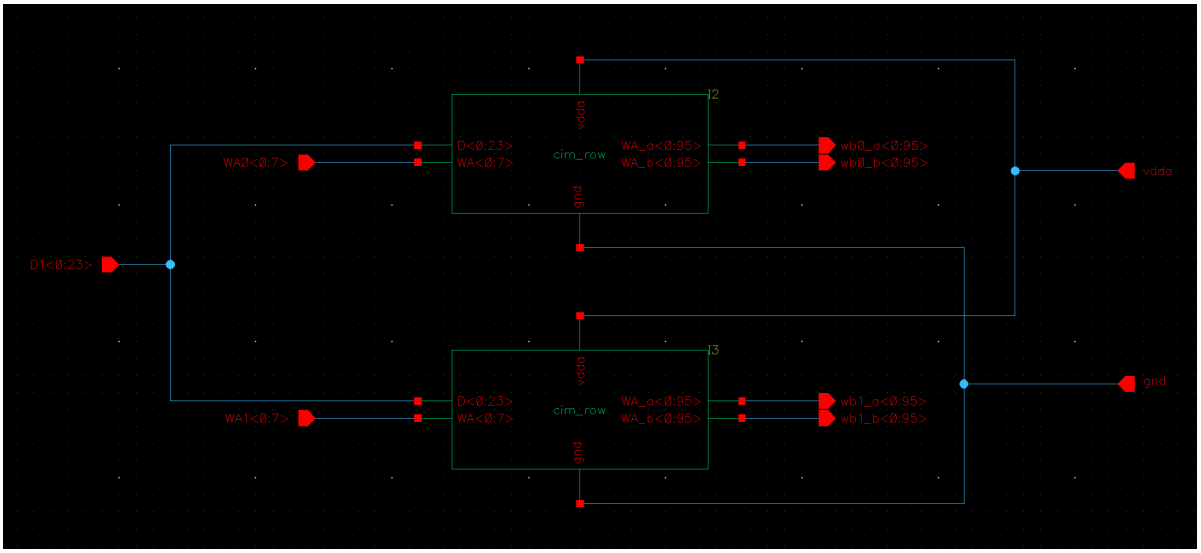
- **cim_bank（24-bit）**：存储24bit数据



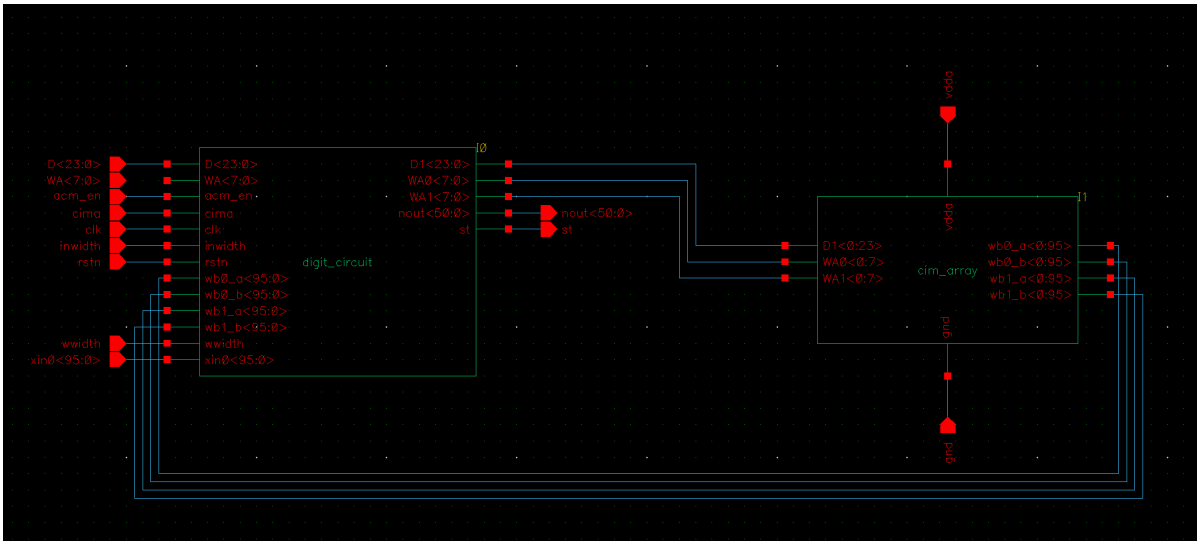
- **cim_row（8banks）**：存储8*24b数据



- **cim_array（2rows）**：存储2*8*24b数据

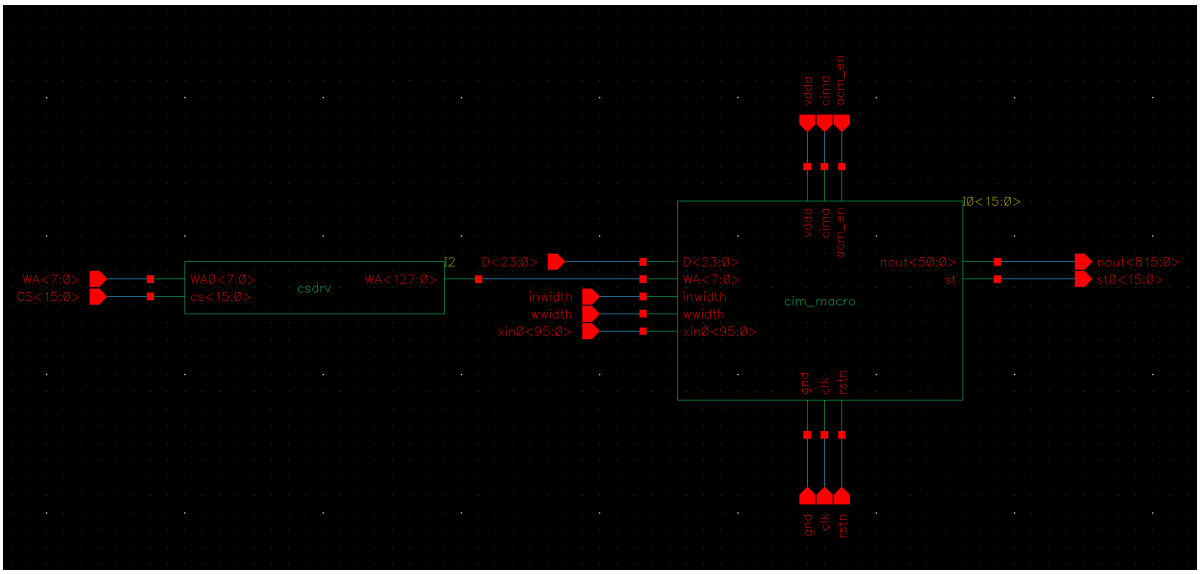


- **cim_macro（array+digit）**：完成2rows可配置的8*24b数据的存储与计算



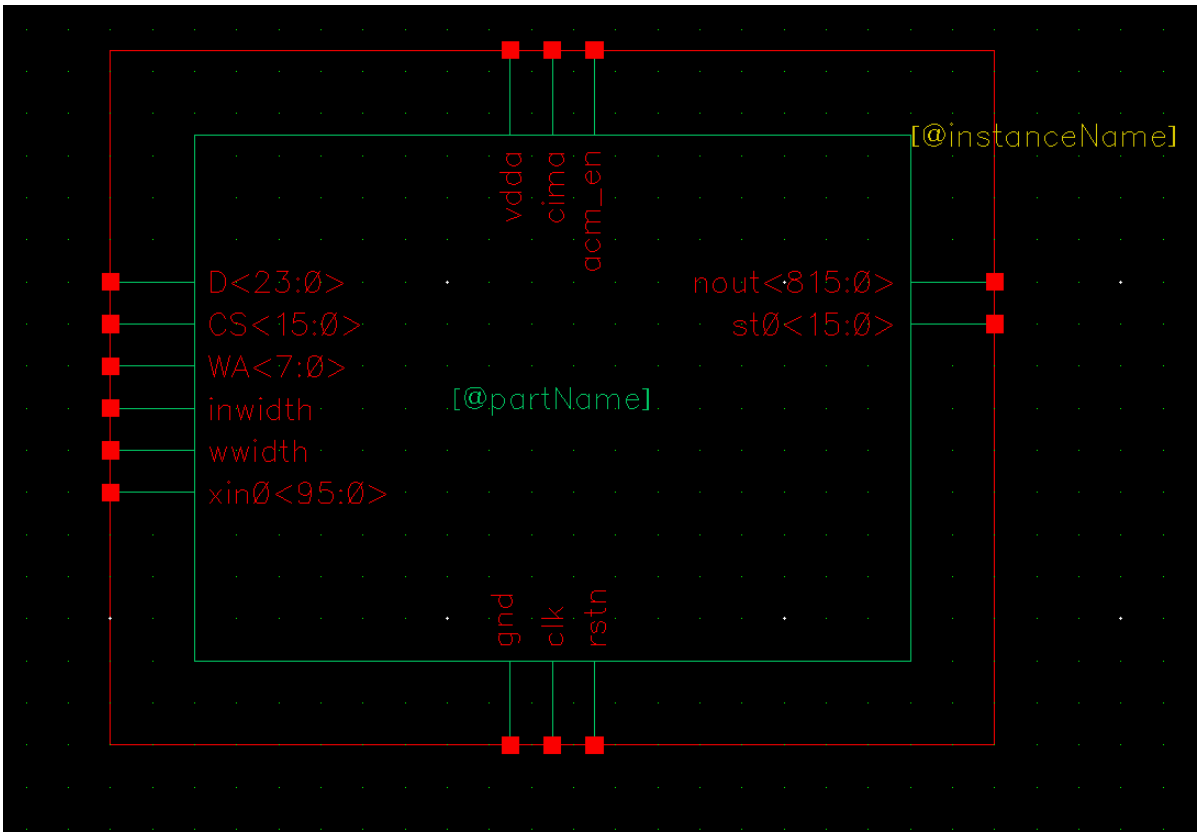
- **cim_top（csdrv+8macros）**：完整电路顶层

该部分在例化16个macro的同时，增加了片选信号输入与驱动



接口设置如下，通过CS与WA信号确定要写入的

也可以改变D的位宽，将权重写入形式修改为整行输入



4. 后端设计（未实现）

本部分转述了[1]中的内容，为后续实现提供参考

MAC阵列混合Vt设计

将混合Vt设备放置在加法器树中，平衡性能和泄漏

- 由于加法器的树形结构，SRAM存储单元阵列和加法器在加法器树的早期阶段占据了加法器树中总器件的75%以上。因此，我们在SRAM阵列和本地加法器中使用**高Vt器件**来减少泄漏。
- 在加法器的其他部分：半全局和全局加法器、移位器和累加器中使用了**低Vt器件**。

在第一级流水线中，采用混合Vt设计导致36.7 %的延迟来自于高Vt器件，63.3 %来自于低Vt器件。泄漏均匀地分布在高Vt (49.5 %)和低Vt (50.5 %)器件之间；MAC阵列漏电流比单纯的低Vt设计小38.6 %。为了获得更好的MAC吞吐量，引入了加法树流水线。高Vt器件在低电压时，由于过驱动电压(Vgs - - Vt)较小，信号走线较慢；在高电压时，由于导线电阻引起的导线延时，信号走线较慢。

为了平衡这些影响，第一级流水线在混合Vt和长导线的加法树中包含较短的门级，而第二级流水线在低Vt和短导线的加法树中包含较长的门级。静态时序分析(STA)用于检查第一级和第二级管道之间的路径延迟和最大工作频率(fMAX)差异。从fMAX -电压图可以看出，第1级和第2级之间的fMAX得到了很好的平衡。第一级流水级在0.9 V时由于栅极级数较短而快2 %，第二级流水级在0.5 V时由于器件电压较低而快4 %。STA结果表明，在0.5和0.9 V时，SSG和- 40 ° C的fMAX分别为0.41和1.49 GHz。

累加器布局优化和电路优化

2个8T细胞和OAI的总面积为0.161μm2。如果增加XIN的个数和W的宽度，DCIM设计需要更多的水平方向的信号轨道，因为加法树变得更深和更宽。为了减轻路由拥塞，对所提出的比特单元采用了三单元高度风格。由于导线在水平方向的走线更短，导线时延和动态功率降低。

Signed-CLA优化效果测量

由在SSG、0.9 V和- 40 ° C角下的STA结果表明，优化后的符号CLA比传统CLA快9.6 %；所需的面积也较小，与纹波进位加法器(RCA)相比，带符号的CLA在宏中的面积开销为0.9 %。

补充

参考文献

[1] MORI H, ZHAO W C, LEE C E, 等. A 4nm 6163-TOPS/W/b \mathbf4790-TOPS/mm^2/b SRAM based digital-computing-in-memory macro supporting bit-width flexibility and simultaneous MAC and weight update[C/OL]//2023 IEEE International Solid-State Circuits Conference (ISSCC). 2023: 132-134[2024-12-04]. https://ieeexplore.ieee.org/document/10067555. DOI:10.1109/ISSCC42615.2023.10067555.

TSMC DCIM的解读

CIM技术经典导读之数字SRAM CIM技术 - sasasatori - 博客园

其他的TSMC DCIM工作

ISSCC 2024：MORI H, ZHAO W C, LEE C E, 等. A 4nm 6163-TOPS/W/b \mathbf4790-TOPS/mm^2/b SRAM based digital-computing-in-memory macro supporting bit-width flexibility and simultaneous MAC and weight update[C/OL]//2023 IEEE International Solid-State Circuits Conference (ISSCC). 2023: 132-134[2024-12-04]. https://ieeexplore.ieee.org/document/10067555. DOI:10.1109/ISSCC42615.2023.10067555.

ISSCC 2022：FUJIWARA H, MORI H, ZHAO W C, 等. A 5-nm 254-TOPS/W 221-TOPS/mm2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations[C/OL]//2022 IEEE International Solid-State Circuits Conference (ISSCC): 卷 65. 2022: 1-3[2024-12-11]. https://ieeexplore.ieee.org/document/9731754/?arnumber=9731754. DOI:10.1109/ISSCC42614.2022.9731754.

ISSCC 2021：CHIH Y D, LEE P H, FUJIWARA H, 等. 16.4 an 89TOPS/W and 16.3TOPS/mm2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications[C/OL]//2021 IEEE International Solid-State Circuits Conference (ISSCC): 卷 64. 2021: 252-254[2024-12-11]. https://ieeexplore.ieee.org/document/9365766/?arnumber=9365766.

DOI:10.1109/ISSCC42613.2021.9365766.