

CiM(Compute-in-Memory)

fassial

更新: July 9, 2020

摘 要

本文为 Alveo-CiM 项目的 CiM(Compute-in-Memory) 模块介绍。

关键词: Alveo-CiM, Compute-in-Memory

1 声明

1. 本文档系 Alveo-CiM 项目¹CiM(Compute-in-Memory) 子模块的说明文档。
2. 本文档只允许无修改原样分发, 必须署名。

2 模块层次

CiM(Compute-in-Memory) 模块主要包括三层: `cim_top`、`cim_cell_group`、`cim_cell`。目前 `cim_cell_group` 层已被实现所有功能。三者间的关系如下:

1. `cim_cell` 系底层单元, 仅支持 `rst` 复位和 `update` 驱动的累加, 并没有包含 CiM 的所有功能。
2. `cim_cell_group(ccg)` 实现了 CiM 的所有功能, 可以配置其进行 XNOR-ACC 或者 MUL-ACC 运算。一个 `ccg` 包含多个 `cim_cell` 和一个 `alu` 单元, 其一个完整周期为内含 `cim_cell` 的数量与时钟周期的乘积。

$$SystemCycle = N_{cim_cell} * ClockCycle$$

3. `cim_top` 主要是由 `cim_cell_group(ccg)` 集成而来, 是一个更大规模的硬件实现神经网络实体。

3 `cim_top`

// TODO

¹<https://github.com/Fassial/Alveo-CiM>

4 cim_cell_group

4.1 模块简图

cim_cell_group 设计简图如图??所示。

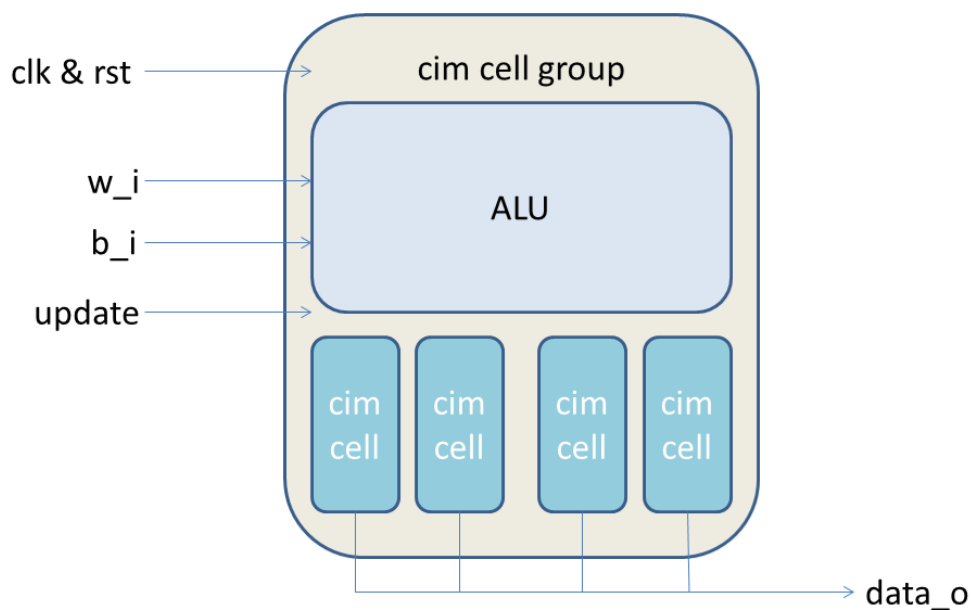


图 1: cim_cell_group 模块简图

4.2 端口说明

4.2.1 IN

- **clk & rst:** external signals. 固有信号，系统时钟与系统重置信号。
- **w_i:** 输入的权重，作为 ALU 的输入。
- **b_i:** 输入的待运算数据，作为 ALU 的输入。
- **update:** 更新信号，指示 cim_cell 进行累加操作，具体 cim_cell 编号依据内部计数器 (count) 决定。

4.2.2 OUT

- **data_o:** 输出 cim_cell 内部数据，以数组的形式。

5 cim_cell

5.1 模块简图

cim_cell_group 设计简图如图??所示。

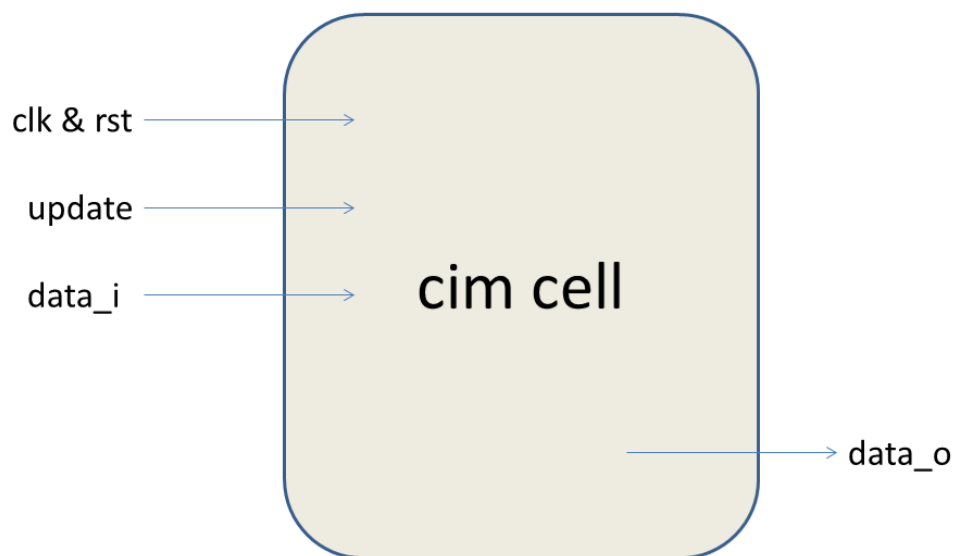


图 2: cim_cell 模块简图

5.2 端口说明

5.2.1 IN

- **clk & rst:** external signals. 固有信号，系统时钟与系统重置信号。
- **data_i:** 输入的数据，待 update 信号为高电平与 cim_cell 内部值相加更新原值。
- **update:** 更新信号，指示 cim_cell 进行累加操作，具体 cim_cell 编号依据内部计数器 (count) 决定。

5.2.2 OUT

- **data_o:** 输出 cim_cell 内部数据。

6 资源占用

由于 cim_cell_group(ccg) 是可以参数化配置的,我们修改 cim_cell_group 的可配置参数??(DATA_WIDTH、N_GROUP、ALU_KIND) 对其占用 FPGA 板载资源进行了较为丰富的实验??(暂不考虑 Bonded IOB、

BUFGCTRL 资源)。cpg 内部的 ALU 主要受 ALU_KIND、DATA_WIDTH 参数影响，cim_cell 主要受 DATA_WIDTH、N_GROUP 参数影响。

- **DATA_WIDTH**: cim_cell 内部数据的宽度，单位是 bit。
- **N_GROUP**: cim_cell_group 内部包含 cim_cell 的数量。
- **ALU_KIND**: cim_cell_group 内部 ALU 的运算行为。当值为 0 时，ALU 进行 XNOR 运算；当值为 1 时，ALU 进行 MUL 运算。

表 1: cim_cell_group 板载资源占用实验

No.	<ALU_KIND, DATA_WIDTH, N_GROUP>	Slice LUTs	Slice Registers	DSPs
1	<0, 32, 1>	34/31/33	33/0/32	0/0/0
2	<1, 32, 1>	49/15/33	33/0/32	3/3/0
3	<0, 32, 4>	134/124/33	131/0/32	0/0/0
4	<1, 32, 4>	149/15/33	131/0/32	3/3/0
5	<0, 32, 12>	401/372/33	389/0/32	0/0/0
6	<1, 32, 12>	416/15/33	389/0/32	3/3/0
7	<0, 32, 1024>	33854/31744/33	389/0/32	0/0/0
8	<1, 32, 1024>	33869/32783/1	32811/0/32	3/3/0