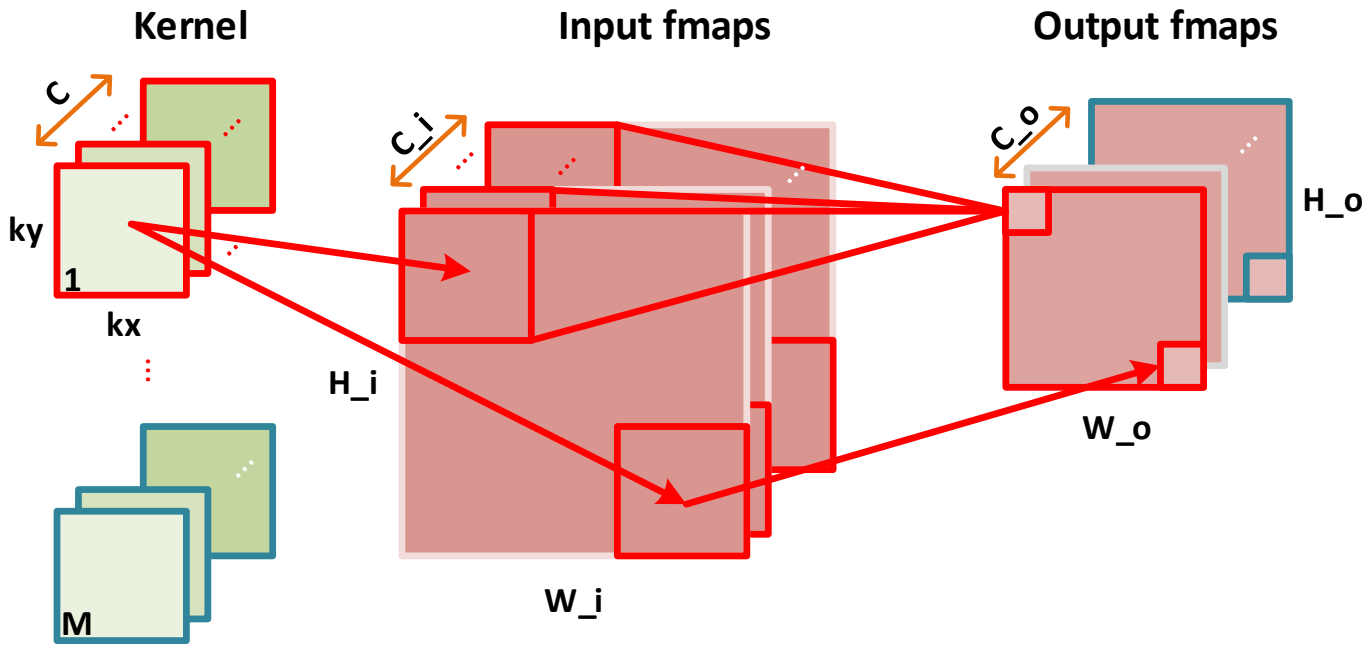
■小型卷积核设计

简述

- 本课程设计实现一个简单的卷积核处理单元，根据给定的指标，给定的存储块，优化卷积计算的方式，实现一层卷积层的计算。
- 设计需要从算法层面出发，设计整体的硬件代码，然后根据实验课上所学的内容，实现硬件代码的仿真，综合，布局布线，最终生成没有问题的GDSII文件。
- 课程设计根据实现的卷积核单元的面积，速度，功耗等来决定最终成绩。

卷积计算

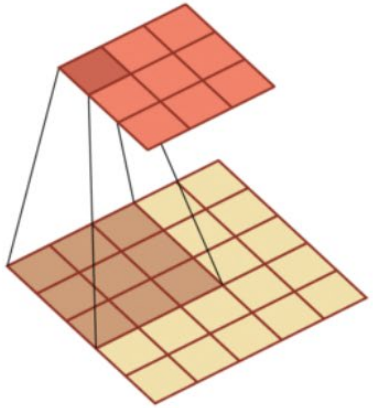


$$O[z][y][x] = \mathrm{Re}LU\left(B[z] + \sum_{k=0}^{c_i-1}\sum_{j=0}^{ky-1}\sum_{i=0}^{kx-1} I[k][Uy+j][Ux+i] \times W[z][k][j][i] \right)$$

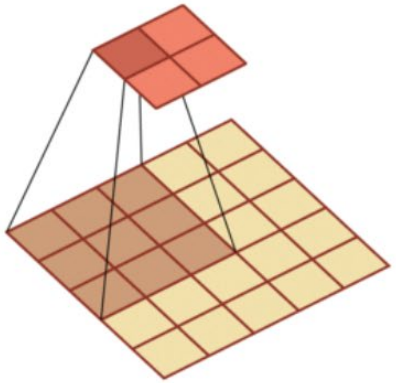
$$0 \le z \le C_o, 0 \le y \le H_o, 0 \le x \le W_o \qquad \mathrm{Re}LU(x) = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases}$$

U is stride. B is bias.

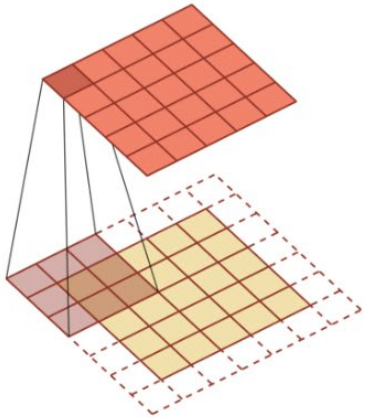
pad的概念



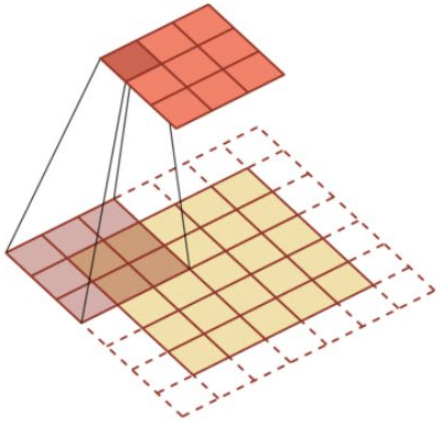
步长1，零填充0



步长2，零填充0



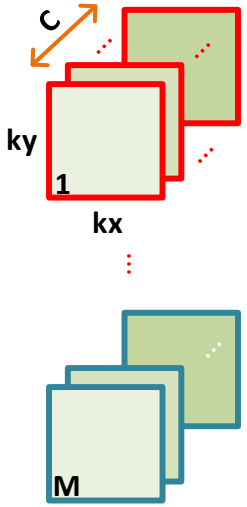
步长1，零填充1



步长2，零填充1

设计指标

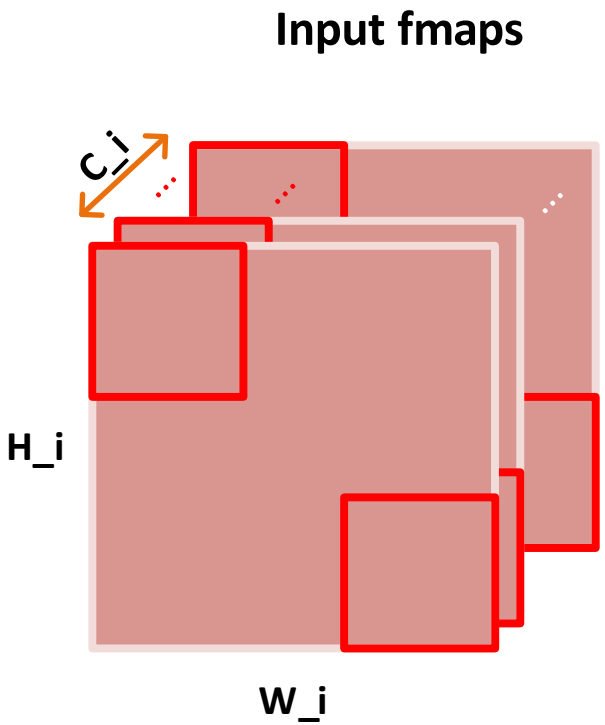
- Kernel



- 卷积核的宽 kx 与高 ky 相等，固定为4
- 卷积核的通道数 C 在8~32之间可调，以8为步长调节，包括8 及32
- 卷积核的数目 M 也在8~32之间可调，以8为步长调节，包括8及32

设计指标

- ifmap 及 ofmap



- 本设计使用 stride 为1，并且周边不补 pad，以简化设计，从而输出的 ofmap 的尺寸通过 ifmap 的尺寸与公式计算得到
- 如果估算，觉得加 Pad 对自己有好处，可以自行设计，并且最后展示的时候需要注明清楚
- 输入特征图的宽与高固定为64
- 输入特征图的通道数与 Kernel 的通道数设置一致，即8~32之间可调，以8为步长，包括8及32
- 输出特征图的通道数与 Kernel 的数目设置一致，即8~32之间可调，以8为步长，包括8及32

设计指标

- 其他说明

- 设计中输入与权重值的位宽均为 8 bit
- 输出保持最大位宽精度，即25bit
- 设计中不考虑 bias 项，即不计算 bias 的累加
- 设计中只使用 ReLU 函数作为激活函数
- ~~设计中可以使用指定大小的 memory 块进行设计~~
- 设计中如果需要用到存储，请使用寄存器作为存储

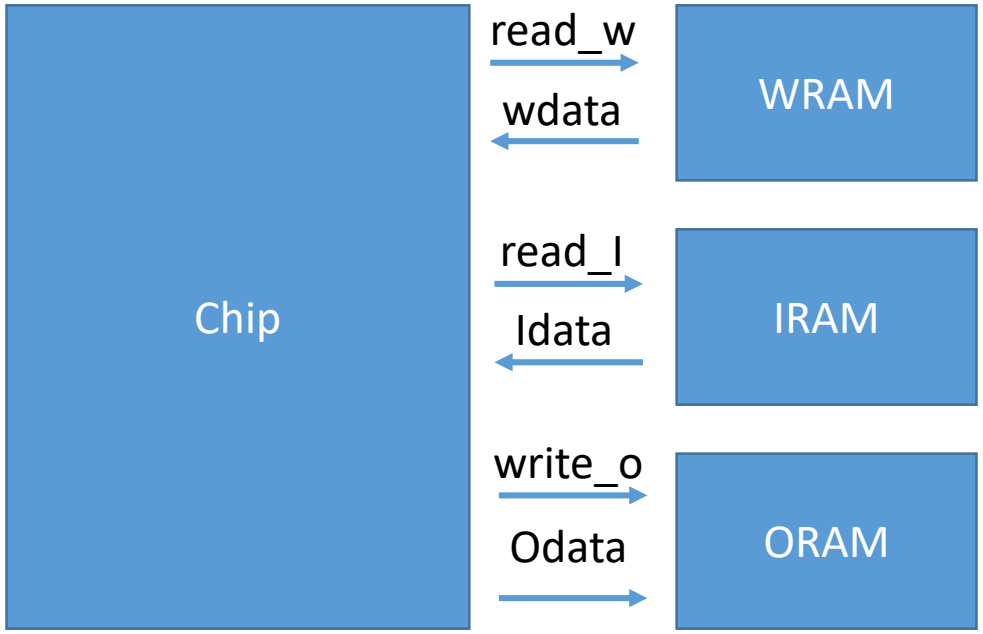
端口信号

- 控制型端口

信号	方向	说明	备注
start_conv	Input	卷积起始信号	-
~~cfg_kx [1:0]~~	~~Input~~	~~卷积核的尺寸~~	~~00 -> kx = 1, 01 -> kx = 2~~ ~~10 -> kx = 3, 11 -> kx = 4~~
cfg_ci [1:0]	Input	输入的通道数	00 -> ci = 8, 01 -> ci = 16 10 -> ci = 24, 11 -> ci = 32
cfg_co [1:0]	Input	输出的通道数	00 -> ci = 8, 01 -> ci = 16 10 -> ci = 24, 11 -> ci = 32
end_conv	output	卷积计算任务完成信号	-

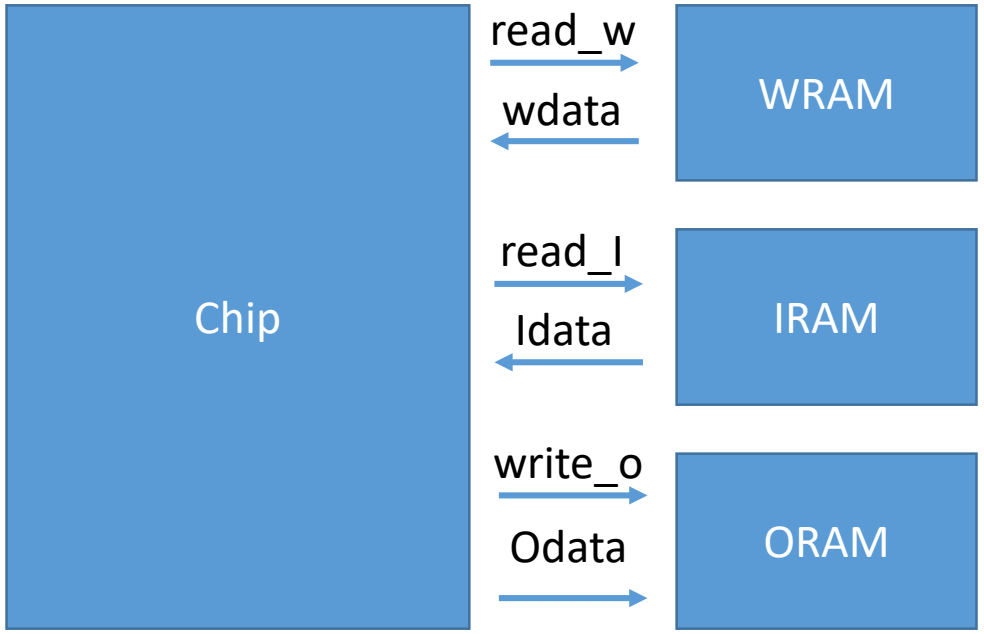
端口信号

- 数据端口



- 为了简化设计，外部数据交互不需要地址控制，外部数据假设只要给了控制信号，在下一个时钟的上升沿就会给入你所需要的数据。
- 例如，只要给出 read_w 信号，在下一个时钟上升沿，就会从 wdata 端口中取到需要的数据。write 信号也是类似。
- 对于输入与权重值，其数据位宽为8*8bit，输入与权重值不共用数据总线
- 对于输出数据，数据位宽为2*25bit

补充说明



- 为了简化设计，本设计核心模块只是实现一个计算阵列，所有的数据都来自于外部的存储
- 假设在外部已经将需要放入计算阵列的数据排布好了，内部只需要实现对输入数据的读取操作，对内部卷积计算的控制操作以及对输出数据的输出操作
- 前一条所提到的数据的排布，请在testbench里面合理排布，以保证核心计算正确
- 我们只提供一个测试数据样本，也就是输入特征图，卷积核，以及输出特征图，请在testbench中根据自己的设计合理排布自己的数据


上海科技大学
ShanghaiTech University
Thanks

上海科技大学
ShanghaiTech University