**数据是如何存储的？不只是最开始的输入图像，还有中间每一层输出的数据是怎么存储的，并且其地址又是怎么存放的？**

从system top中可以看到wire的数据和地址。分为data和kernel(M1),kernel的只有读出来的数据和地址，data的有读出的数据和地址、写入的数据和地址。

其中kernel的部分接到了kernel\_bram\_dma这个模块以及对应的Kernel\_Bram模块。Data的部分接到了BRAM\_DMA这个模块以及对应的DATA\_BRAM。

Data的bram\_write\_value没有用上？

每个DATA\_BRAM的位宽为72bits，存储9个8bit的**ifmap**，深度为10240；

阅读D:\Postgraduate\project\MobileNet\MobileNet\_VC709\_v2\tcasii\tcasii\layer1的python代码，它读取D:\\Spyder Project\\mobilenetv2\_demo\\layer1\_to\_53\_M\_V1\_bias32b存储的pickle对象，并按照**一定的顺序**生成用于配置FPGA片上BRAM的.coe文件。

每个Kernel\_BRAM的位宽为72bits，存储9个8bit的weight，深度为43008；

M1\_Bram的位宽为288bits，可以存储9个32bit的M1，深度为5120；

先看kernel\_bram\_dma

功能：1.将kernel\_read\_input直接连到kernel\_input作为输出

2.计算出kernel的读取地址：地址 = Kernel\_base\_addr + kernel\_read\_base\_p + kernel\_base\_channel = Kernel\_base\_addr + kernel\_pw\_p\*Ifmap\_channel\_num

+ kernel\_base\_channel;所有kernel读取地址都是一样的？

3.将M1\_out直接连到M1\_bram\_in

4.将kernel\_pw\_p延迟一拍用于M1读取地址的计算

5.M1\_bram\_en和kernel\_bram\_en都只在bram\_st == 1 时拉高

简单地说，kernel\_bram\_dma模块的功能是计算读取地址，输出使能信号

数据的存储和搬运：

Data, weight,M1,M0,zero point是如何存储在bram中的？每一个数据的地址如何确定？当计算完成后，Ofmap的数据又应该存储到哪个位置？

为了研究数据在bram中的存储方式，我首先需要研究将bram初始化的coe文件是如何生成的。

Coe文件是由txt文件直接修改后缀得到，而 txt则是由D:\Postgraduate\project\MobileNet\MobileNet\_VC709\_v2\tcasii\tcasii的python代码生成。

首先使用numpy的模型将M1、input、weight这些数据存储为.p格式（这种格式与pkl一样），然后运行layer1\_coe.py将数据写入txt文件。

成功运行后，研究这个layer1\_coe.py。

Data、weight的存储方式画在***bram初始权重和输入存储方式.pdf***这个文档中。

BRAM\_DMA

Ifmap\_size = 225, ofmap\_size = 112, ifmap\_channel = 3

Ifmap\_hori\_num &ifmap\_vert\_num分别代表了此时需要读出的3\*3窗口的数据的中心位置（4）在输入特征图的坐标。Ofmap\_hori\_num &Ofmap\_vert\_num类似。

先将Ifmap\_hori\_num &ifmap\_vert\_num分别除3得到其在哪个3\*3窗口以及具体的位置索引。其中Ifmap\_hori\_num除3得到的商为read\_h\_block\_idx（窗口行坐标），余数为read\_local\_hori\_cnt。Ifmap\_vert\_num除3得到的商为read\_v\_block\_idx（窗口列坐标），余数为read\_local\_vert\_cnt。

Read\_scenarios = {read\_local\_vert\_cnt, read\_local\_hori\_cnt} 此为指示信号，说明了此时读出的窗口属于哪一种case。

再将Ifmap\_size = 225, ofmap\_size = 112分别除3得到每行（列）有多少个3\*3窗口。结果分别为row\_block\_num，ofmap\_row\_block\_num。

根据read\_scenarios，将输出Ifmap0\_input ~ Ifmap8\_input 连接到相应的bram数据。

接下来计算每个数据的水平块偏移 + 垂直块偏移。由于窗口中有些数据和中心位置的数据处于不同的行（列），因此需要逐个计算。计算完成后相加得到对应的地址。

然后计算写入地址。

Ofmap\_hori\_num除3得到写入位置的窗口行坐标write\_h\_block\_idx，余数write\_local\_hori\_cnt。Ofmap\_vert\_num除3得到写入位置的窗口列坐标write\_v\_block\_idx，余数write\_local\_vert\_cnt。

write\_scenarios = {write\_local\_vert\_cnt, write\_local\_hori\_cnt};说明了写入位置在窗口内的哪一个位置。基于此输出bram\_write\_wea，并送到每一个bram端口，从而控制写入哪个bram。

接下来用和读取类似的方式计算写入坐标。

疑问：为什么计算偏移的时候需要乘ifmap\_channel？

Counter\_dw

用户设置、状态机、counter

Config\_data中，general config的ifmp\_channel&ofmap\_channel的含义是什么？在central\_control中，ifmap\_channel使用了10bit来存储，在counter\_dw中却只定义了一个8bit的wire来进行赋值？

为什么ifmap\_size后面要跟着一个no\_first\_layer?

后续语句：assign ifmap\_size = (Mode==3'd1)? 224: ifmap\_size\_no\_first\_layer;

只要是普通卷积就设置为224作为ifmap\_size？单独这样设置有什么意义吗？不能直接在config\_data里面写吗

因为config\_data只用了7位来表示，224放不下。

Pos x,pos y为什么是从零开始的？如果是nc或者dw，那么应该从1，1的位置开始卷积。

为什么write channel要到最后一个channel才能使ofmap的hori和vert自增？