## Flow of a conv layer

1. 确定卷积层的参数，诸如feature大小，weight大小等。
2. 设置buffer\_copy函数的模板。
3. 根据(分配的)global BRAM的数量来决定将一个layer分为多少个block。
4. 计算当前block需要拷贝的feature和weight在DRAM内的位置(考虑padding)并进行拷贝。
5. 根据PE的数量决定一个block内的循环次数，执行计算。
6. 计算当前PE需要拷贝的feature和weight在global BRAM的位置并进行拷贝。
7. 为local BRAM初始化bias；调用PE进行计算；在local BRAM内做ReLU。
8. 将计算结果拷贝回global BRAM
9. Global BRAM内的feature拷贝回DRAM

## Padding的处理

Max pooling，average pooling和conv都需要padding，需要填充的数值是不同的

是在DRAM到global BRAM阶段还是在global BRAM到local BRAM的阶段处理

* DRAM2global：inception block里面我们可能要共用input\_feature
* global2local：控制逻辑比较复杂

## 某些判断是否有必要(不能整除的情况)

主要是不能整除的情况。比如第一层的input feature的channel数量是3，但是global BRAM的channel数量有64。是仍然进行64个通道的计算，还是加一个控制逻辑，只做前三个通道的计算。

## 自动生成代码

现阶段相同算子大部分内容都是相同的，只需要填上参数即可。但是需要填的参数比较多，所以现阶段是否考虑写一些自动生成的代码。