1）论文阅读 三段式：

研究问题的描述(层次、重要性、和别人研究的同异)

有哪些挑战和解决挑战的方法

和自己的工作相比可以借鉴的点是什么:

1. 自己的工作，主要解决了哪些问题，下周的计划是什么:

几个问题 需要细化深入：

1. 文章1/2/5都是软硬协同创新，具体的硬件层次是哪里，亮点创新是什么？

* Fast：系统全面，BFP-MAC（具体电路），转换器（硬件找共享指数，考虑了梯度 增加随机舍入），存储，数据流、映射分析（前向 反向 多向脉动阵列）

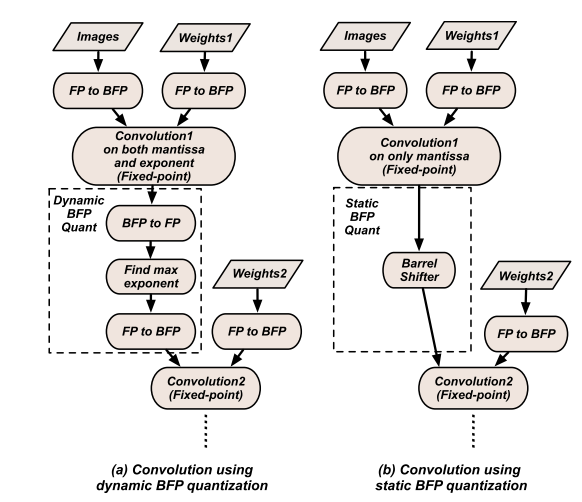
亮点：（1）BFP进行可变精度训练（可变支持2^n尾数）和（2）随机舍入的BFP

* Buket：硬件在MAC层次，如何优化高耗能的FP-ACC

亮点：1.将指数相近的放在一个桶里累加，减少移位。2.为了减少进位导致FP-ACC工作，允许进位传递到旁边的桶；3。对于桶累加后一些很小的值，可以不在FP-ACC中加，如何维护，只保留TOPK最大的一些桶。

* 3D：NNandLS 在之前的论文ICFPT基础上，提出新场景3D CNN，硬件创新较少 转换器直接移位（软件通过最小KL散度找共享指数指数） 尾数乘加树在通道数少时，分成两个。

亮点1.更省的量化：静态块浮点，为每层找一个固定共享指数，；2.如何将2D 3D统一成一种计算pattern，



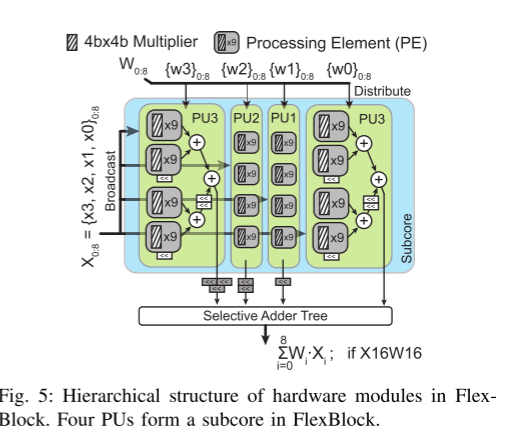
1. 第2篇的mac可以深入借鉴，是不是就可以基于这个搞MX的支持，要深挖

采用分桶累加的策略，需要指数差异较大的场景；另外分析如果稀疏的时候不需要加0，也减少了FP-ACC活动性

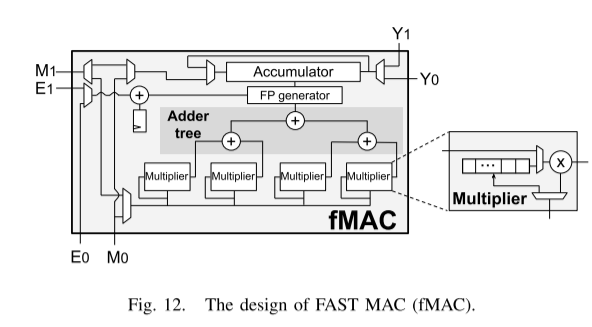
1. 第3篇的硬件层次是什么，多精度体现在哪个层次描述不清晰

硬件层次在PE阵列，有三层，PE

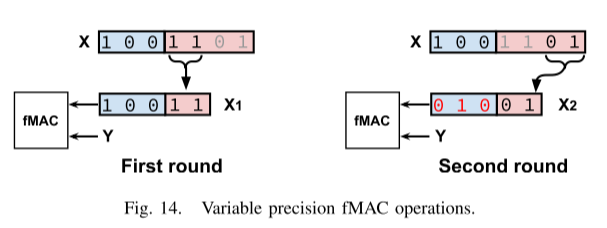
PU（实现X上的子字并行） SUBCORE（实现W上的子字并行），分层子字并行，减少传递的部分积累加的数目



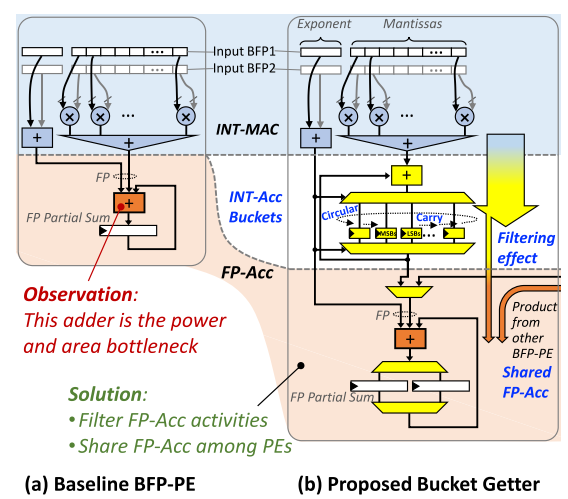
1. 这几篇的相同硬件层次上，例如mac上的异是什么



* FAST：设计的2bit的基本MAC，进行4bit计算两轮，时间换空间；需要支持多向；



* BUKET：讨论的MAC里面的FP accumulator



* FLEX：MAC不做浮点累加，让FlexBlock的处理核心只处理定点计算，分层子字并行
* 2019:：卷积层采用16（输入通道）×64（输出通道）BFP PU。16个PE对应16个输入通道，PE内64个PU共享输入图，用不同通道的权值，16个PU的乘法结果在An中累加，结果与前面输入通道的Sn累加。
* 3D： 根据通道数大小分为两种模式，通道数少时，将乘加树分成两个独立的树。

MX虽然有两级，但是在单mac上可以以一级体现 或者2级（那可能比较宽），有一种搞法就极致了 支持一种很低的 在某些场景可以 ；也有的搞法 是可配置的 但同时硬件开销可控 这种适应的场景应该要多些

文章1 FAST: DNN Training Under Variable Precision Block Floating Point with Stochastic Rounding

KUNG组HPCA2022

* 研究问题：如何通过使用可变精度的块浮点数和随机舍入来实现快速和准确的深度神经网络训练。可变精度意味着（1）系统在训练期间有效地支持尾数宽度范围内的BFP格式（2）允许具有不同尾数宽度的BFP之间的点积。
* 研究层次：软硬件协同，设计了整系统包括FAST-MAC、BFP转换器、存储，分析了前向反向的脉动阵列计算映射。
* 研究重要性：BFP能够实现比FP更高的点积计算效率，不同层需要不同的BFP精度。随机舍入对模型精度至关重要，特别是当使用BFP用于具有低精度尾数的梯度时
* 研究差异：

同：都是用BFP格式

异：在格式层面，微软MSFP12 PTQ进行推理 指数宽8b，而本文使用BFP感知的DNN训练，指数宽4b，同时实现类似的推理精度；对于训练使用可变精度减少时间和能耗。在加速器层面：动态调整DNN训练的精度；FAST对BFP组中的每个部分乘积执行整数MAC，2b,4b尾数相当于实现混合精度

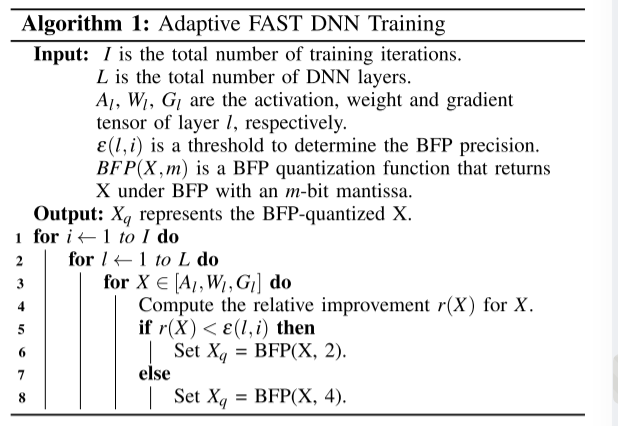
* 挑战：

1）如何有效地使用块浮点进行训练，同时保持模型的准确性。这需要在每个DNN层和每次迭代中，自适应地选择权重、数据和梯度的最佳精度。

2）还需要处理低精度尾数(例如2或4位)的BFP梯度更新，以最小化梯度下降中的损失。

* 解决：

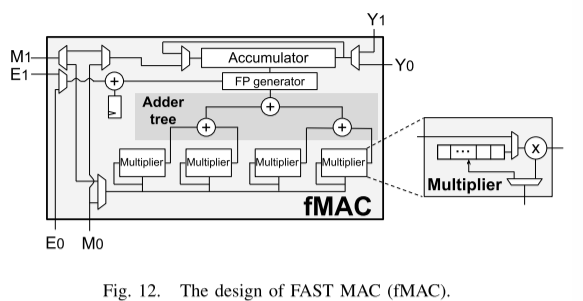
1）提出可变精度训练算法，通过自适应地选择权重、数据和梯度的最佳精度，从而减少总的训练时间。支持用2位（低精度）或4位（高精度）表示BFP尾数，4位尾数与2位BFP尾数相比的相对改进r（X）。



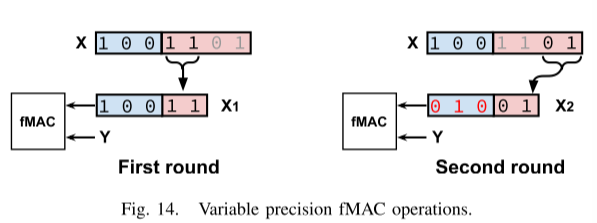
2）提出了使用BFP进行可变精度训练和使用随机舍入的BFP的方法。通过对权重梯度更新应用随机舍入的影响进行了分析。

3）设计了一个模块化的架构，包括FAST乘法累加器(fMAC),用于处理一组BFP值。

前向为例：WS，首先使用E0和M0端口将W的BFP共享指数和尾数分别预存到fMAC中 从下到上，激活通过相同的E0和M0端口进入以执行DP计算。输出O通过Y0向右传递。



将计算细分为2位块，允许通过简单地运行多遍fMAC来实现涉及更高精度尾数的算术运算。



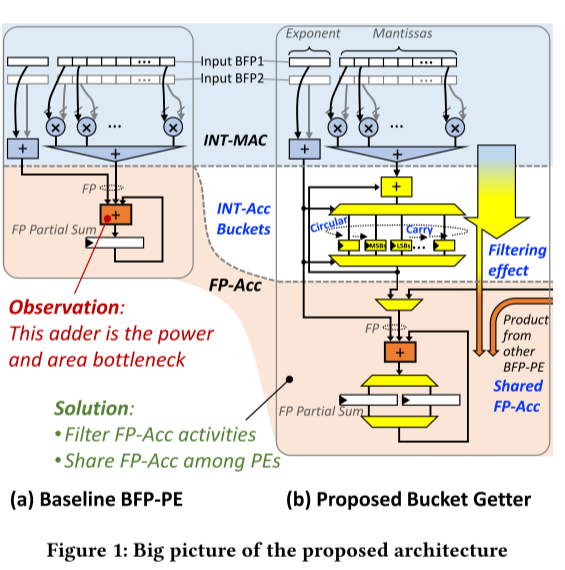
文章2. Bucket Getter: A Bucket-based Processing Engine for Low-bit Block Floating Point DNNs

国立清华MICRO2023

* 研究问题：浮点加法器（FP-Acc）（其包含用于归一化、对齐、加法和定点到浮点（FXP 2FP）转换的模块）主导功率和面积开销，因此妨碍BFP-PE的硬件效率。
* 研究层次： 切入点小，PE中FP-ACC的优化
* 研究重要性：解决了块浮点计算中浮点加法器的功耗和面积的瓶颈
* 研究差异：

同：是研究BFP的计算问题

异：是第一个针对耗电的FP-Acc问题的解决方案



* 挑战：

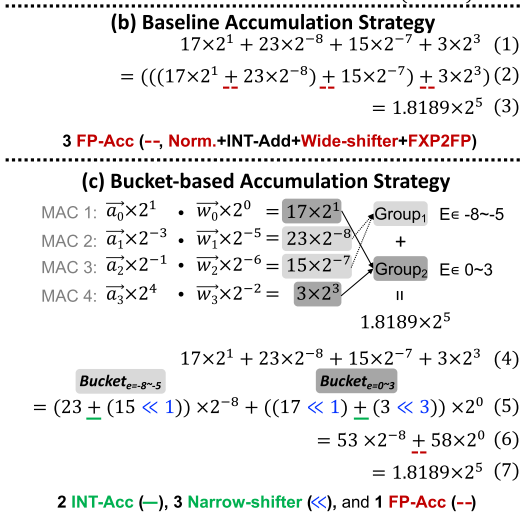
1）FP-Acc主导BFP-PE的功耗，因为除了加法器，FP-Acc还需要归一化、对齐和定点到浮点转换（FXP 2FP）单元，如何降低FP-ACC的活动性。

2）每个桶独立地累积小范围指数值中的部分和，第一级桶累加器和第二级FP累加器类似于L1 L2cache，但是当桶累加器溢出时，仍然需要FP-ACC累加。

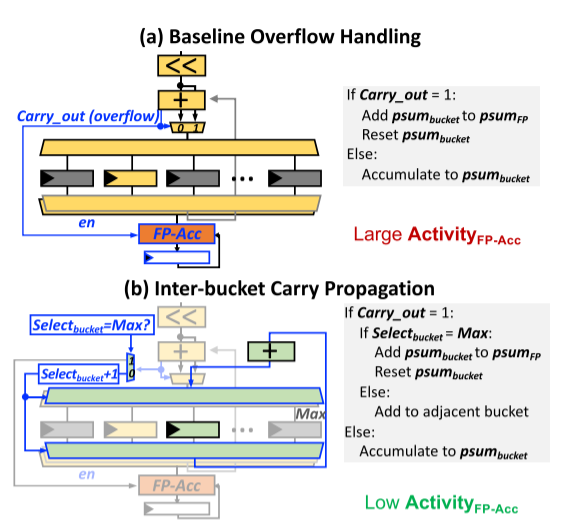
* 解决：

1. 基于桶的累加器不是只使用FP域中的一个寄存器来执行累加，而是利用一组小的累加器(桶)，并且每个累加器只负责累加具有小范围指数值的部分和，因此可以大大简化移位器对准单元。

图中分为两组，每个组的动态范围很小（4位）;因此，累加这些项需要简单的运算



1. 桶间进位传播：允许每个桶将溢出转移到相邻桶，以减少FP-ACC的活动
2. 当黄色桶累加器发生溢出时，灰色的其他桶不能辅助黄色桶，因此触发FP-Acc活动
3. 添加了一组额外的多路复用器和多路分解器电路以及相应的右侧桶选择逻辑。这种硬件支持有助于将溢出项添加到指定的桶中，该桶在(b)中显示为绿色桶

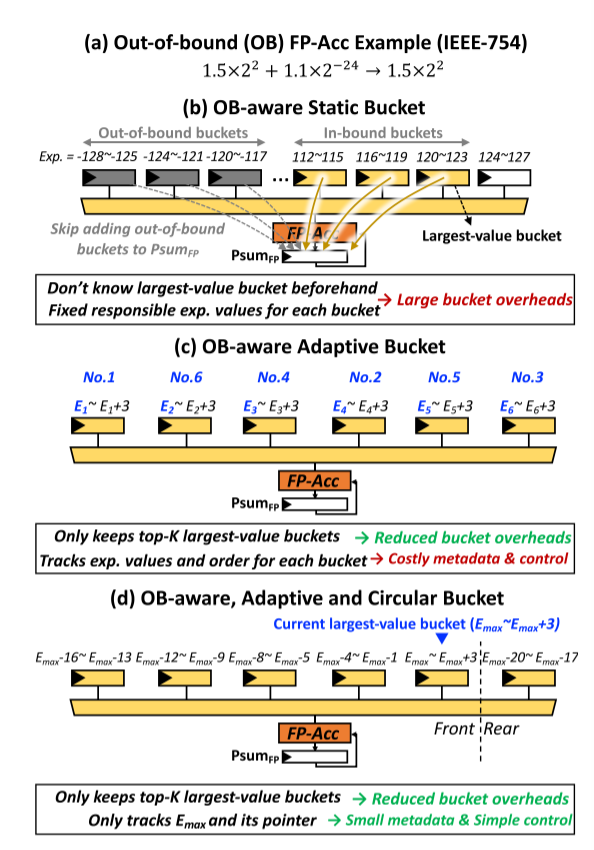


1. a)非常小的部分和（out-of-bound）添加到原始FP部分和不会影响结果

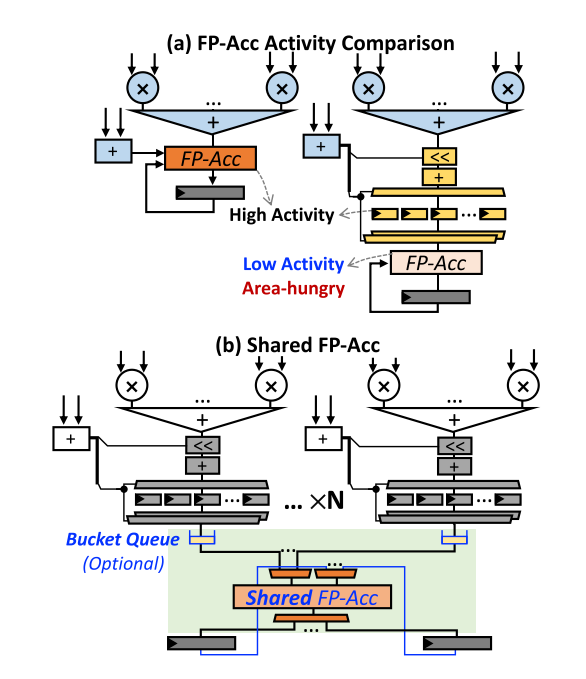
b)完成桶累加后识别，跳过很小的值，不在FP-ACC中加，但它仍然需要维护所有的桶. 图中分配64个桶，每个桶负责四个指数值

c) track每个桶的负责指数值，并仅保留前topK个in-bound桶，如果检查来的部分和大于当前最大值的桶则检查顺序并替换。分配6个桶，但是需要复杂的存储桶控制逻辑

d)自适应循环桶结构：以减少所需的桶数。第一次随机选一个桶放在循环队列的前面，只需要记住这一个桶和指数，当部分和指数大于这个时，占据最小桶，如果部分积out-of-bound删除最小桶的部分积。



4）共享FP-ACC：在多个PE之间共享FP-ACC以减少FP-ACC的面积开销。



* 启发：MX的累加时，MAC考虑稀疏，定位零输入值或零权重以跳过对应的无效MAC操作

文章3：FlexBlock: A Flexible DNN Training Accelerator with Multi-Mode Block Floating Point Support

韩国科学院IEEE Transactions on Computers  二区CCF-A 2022

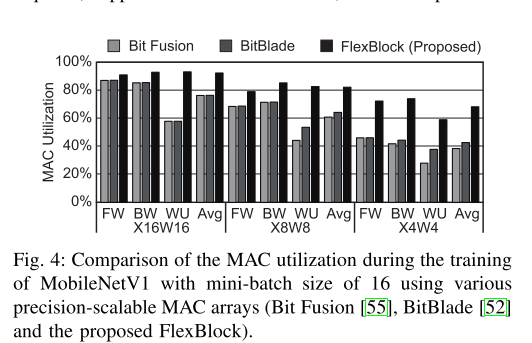
* 研究问题：
* 研究层次： 主要对MAC多精度的优化
* 研究重要性： 提高多精度支持的加速器的利用率
* 研究差异：

同：支持多种精度

异：基于BFP的DNN加速器的先前工作依赖于特定的BFP表示，使得它们的通用性较低。本文支持三种不同的BFP精度模式，并且确保资源高效利用。

* 挑战：

1. 先前的工作通用性低，训练移动场景的 DNN 需要低精度和高精度的算术运算，这促使了对具有多精度支持的加速器架构的需求。
2. 先前的多精度大多关注推理，当训练时固定精度和特定层的MAC利用率是次优考虑，比如bit-fusion,bit-blade，在精度降低时利用率下降严重，16位76.5% 8位降低13.8% 4位降低22%，这是因为在降低的精度下需要更大数量的累积器。

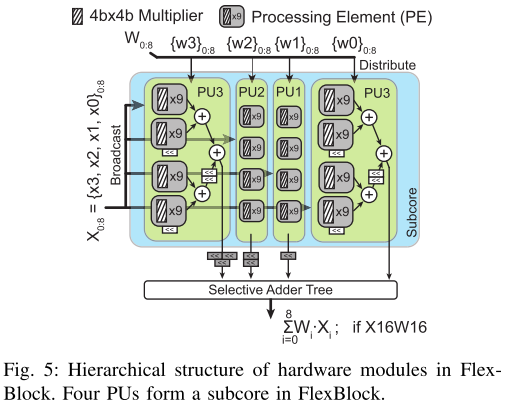


* 解决：

1）两种设计技术来最大化所有训练步骤和各种层类型的利用率：

i）以分层方式将张量维度映射到计算单元，

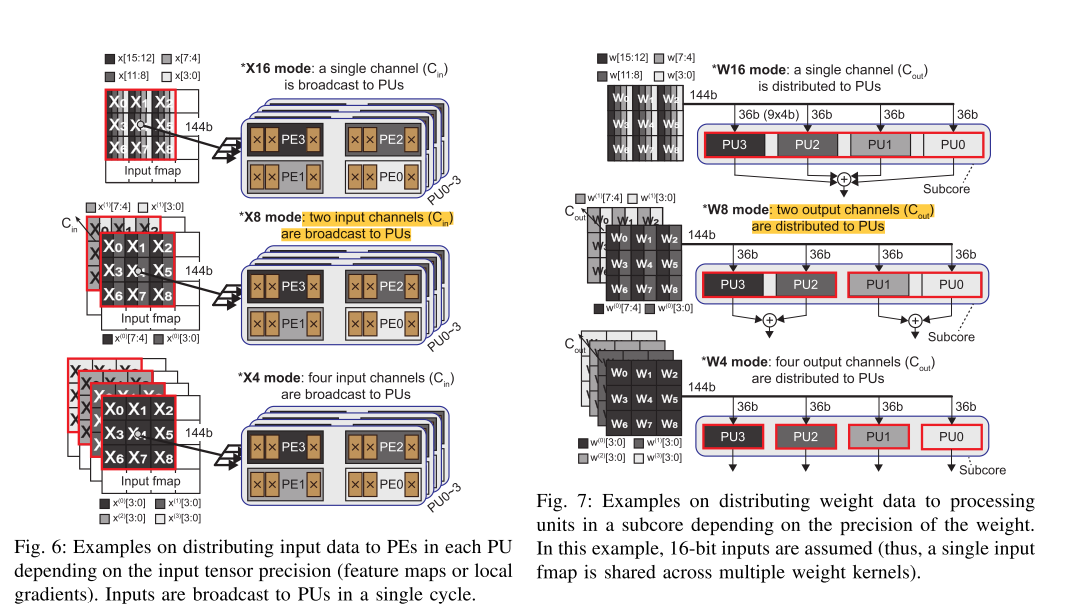
乘法器→处理元件（PE）→处理单元（PU）→子核



X上的子字并行性在PU中的四个PE上实现。对于16b，每个4b（x3、x2、x1或x0）被映射到对应的PE。然后，四个PE被聚集以形成PU。W上的另一子字并行性跨子核中的四个PU实现。对于16位b，则仅将4位子字（w3）映射到PU 3并传送到PU内的所有PE。w2、w1和w0分别被分配给对应的PU。

2）提出支持多种BFP精度模式和层类型。FlexBlock旨在支持4位、8位和16位（符号+尾数），并具有8位共享指数。(与BF16和FP32的指数位宽相同)。此外还支持不同尾数，比如8位输入 4位权值，16位梯度，总共27种组合

用4位算法（FB 12 X4W4）来计算特征图和局部梯度，使用8位/16位算法（FB 16 X8W8/FB 24 X16W16）来计算权重梯度



X：16x，每个fmap元素由四个4b子字组成，每个子字映射到对应颜色的PE 4x4乘数，输入广播到PU

8x 每个元素二个4b，广播两个channel的；4x广播四个channel

W：（W16）将每一权重参数的4位子字递送到对应PU。其他三个子字被分配给子核中的剩余PU。在这种模式下，来自所有PU的输出由选择性加法器树累加。

（W8），PU 2 -3和PU 0 -1分别提供Cout = k和Cout = k + 1的部分和。在reduction单元末端的选择性4路加法器树产生两个部分和。

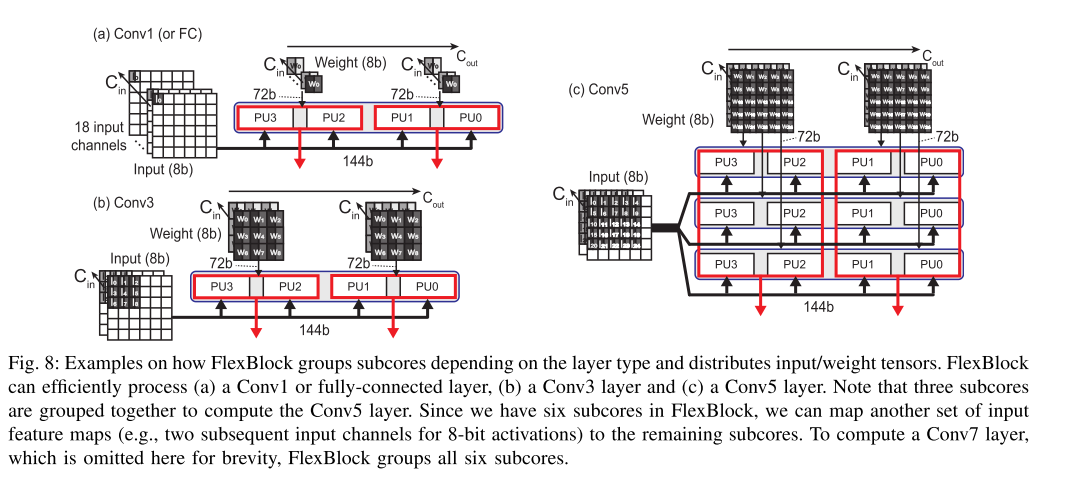
W4 4个输出通道分给4个PU，绕过选择性加法树

ii）设计约简单元。

3D操作映射：来自PU的输出由3D模式的缩减单元垂直累积

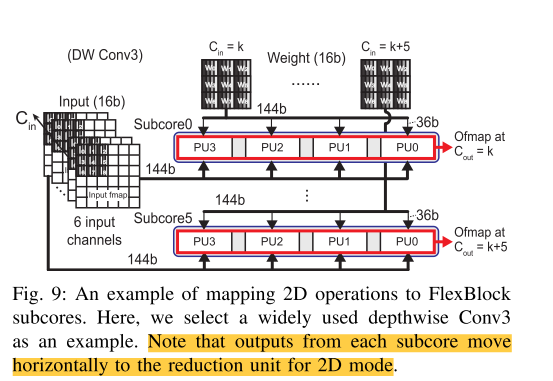
W1x1,Subcore 0,1分别负责18个输入通道

FlexBlock中放置了六个子核心。对于Conv 5层，两个集群，每个集群有三个子核心，核心利用率变为（5 × 5）/（3 × 9）= 0.93。对于Conv 7层，我们使用所有六个子核构建一个集群，（7 × 7）/（6 × 9）= 0.91



2D操作映射：来自每个子核的PU的所有输出由2D缩减单元中的4路加法器树累加

每个子核输出一个通道的结果

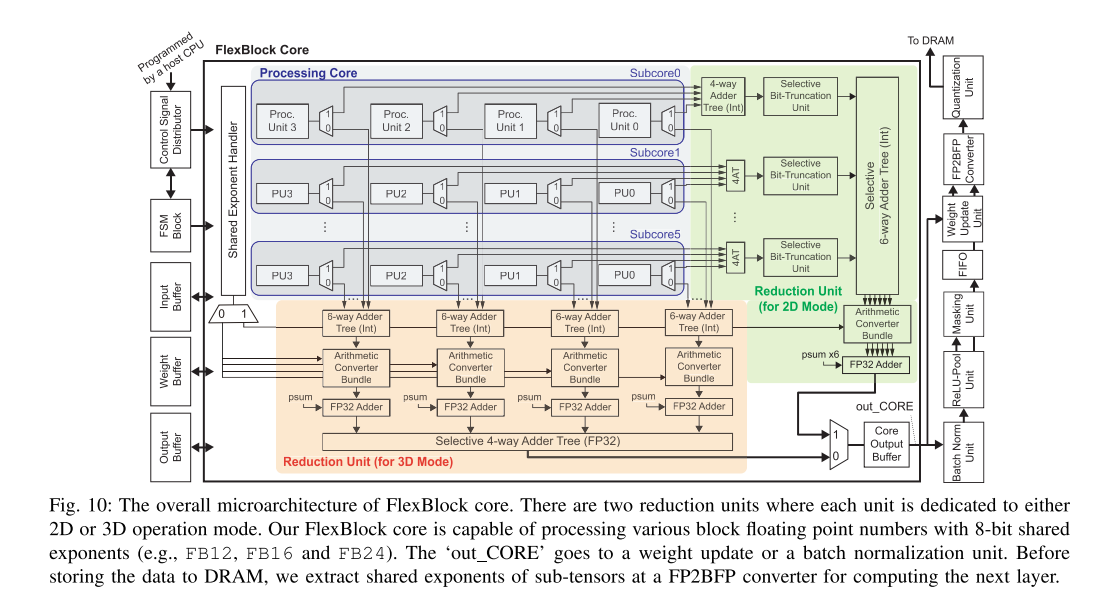


整体结构：

支持FlexBlock中的块浮点运算来处理的，FlexBlock具有共享的指数处理程序、放置在FP32累加单元之前的算术转换器和FP2BFP转换器。

通过将用于2D操作的专用归约单元与用于3D操作的默认归约单元一起放置来解决core利用率不足。

每个处理核心由六个子核心组成，以最大限度地提高各个DNN层的核心利用率。



文章4：High-Performance FPGA-Based CNN Accelerator With Block-Floating-Point Arithmetic

清华TVLSI期刊 CCF-B 2019

* 研究问题：
* 研究层次： 在之前工作基础上提出系统级的FPGA实现方案。
* 研究重要性： 本文研究了有限精度数据表示和计算对神经网络训练的影响，并提出了一种实现低精度定点算术和随机舍入的能效硬件加速器，以克服深度学习中计算资源受限的问题。
* 研究差异：

同：BFP量化

异：2019年较早的，BFP在硬件实现，提出了一种BFP的误差传播数学模型。

* 挑战：

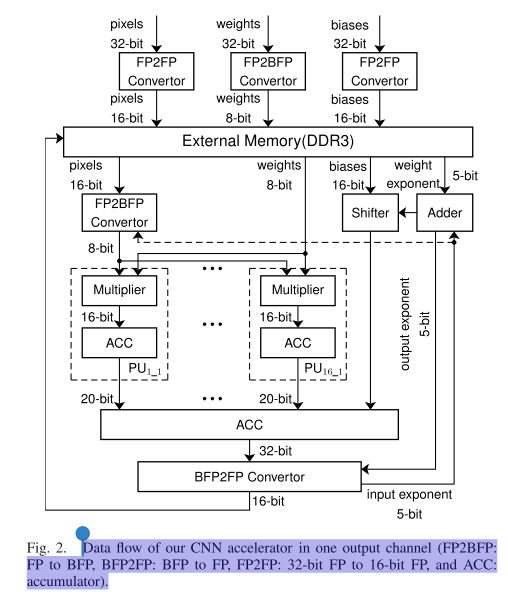
1. 低精度量化需要耗时的再训练来修改权重QAT fine-tune，而NIVIDAint8 TENSORRT在量化和去量化阶段中需要FP除法和乘法 PTQ。
2. 精度损失主要来自BFP 2FP和FP 2BFP转换。加法和乘法的加速是以比浮点运算更低的计算精度为代价的，因为块中的小数字在基于块的对齐过程中牺牲了更多的有效位，截断和舍入是处理移位的两种常用方法。由于截断模式引入了有偏误差，误差会在层间累积。

对于低位尾数BFP，普通置零模型（当权值FP32toBFP 移位>=24 尾数位24包括隐藏的1/输入图FP16toBFP 移位>=11时，尾数置0)导致高舍入误差。

* 解决：

1. 无需再训练和微调的BFP算法加速器。BFP量化的精度取决于尾数的位长和块划分模式。

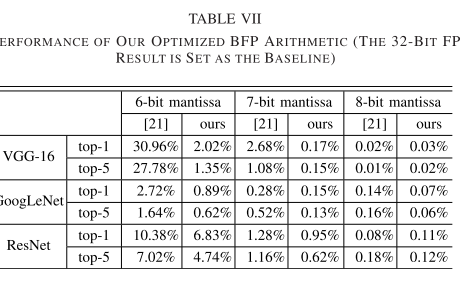
输入图和偏置转换成FP16存在DDR，偏置尾数移位shifter（偏置和输出的指数差）



1. 提出了一种优化的舍入和置零模型，在保持计算精度的同时有效地减少了尾数位长度

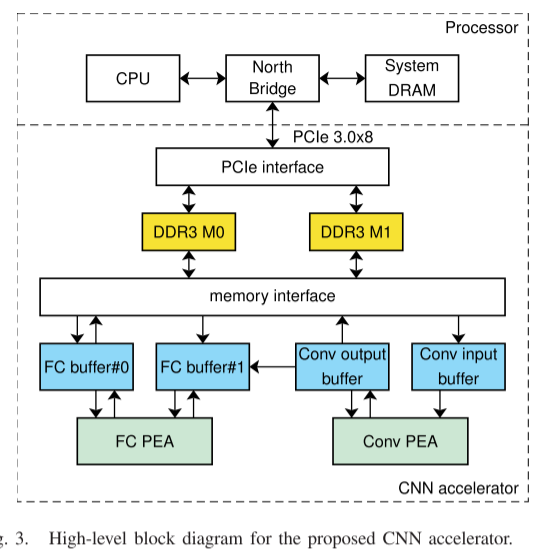
RN就近舍入，RZ向零舍入，RU向正无穷舍入，RD向负无穷舍入，选择RN

当权重FP2BFP模块中右移长度di大于24时，则BFP格式化的数字xbi被设置为零。



1. 并行卷积引擎，乒乓存储器访问模式和优化的片上缓冲区方案

当前图像的FC层由FC PEA处理时，CONV PEA处理下一图像的卷积计算。通过分析卷积层和FC层的计算量，卷积层采用16（输入通道）×64（输出通道）BFP PU，而FC PEA仅采用2个16位FP乘累加（MAC）



通过实验选取8位尾数

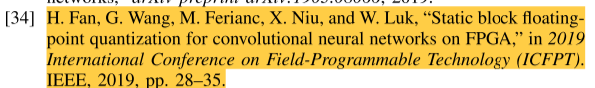
**文章5：High-Performance Acceleration of 2-D and 3-D CNNs on FPGAs Using Static Block Floating Point---**

**英国伦敦帝国理工学院 IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS 一区2022**

* 研究问题： 主要研究了如何使用静态块浮点数在FPGA上高效加速2D和3DCNNs
* 研究层次： 偏向量化方法的研究和应用，硬件设计很少
* 研究重要性：Lian等人[13]文章4 虽然采用8位尾数BFP进行主计算，但片上片外通信所用的精度仍为16位，且需要频繁进行BFP与FP之间的转换，这给内存占用和带宽资源带来了沉重的负担。本文是静态量化，为每层找一个固定的共享指数。
* 研究差异：

同：硬件架构类似

异：在之前的论文基础上，提出新场景3D CNN，之前论文仅评估了2D卷积的内核设计，而没有运行实际的CNN模型，并且没有研究2D和3D CNN的统一硬件架构。



[os-hxfan/Static\_BFP\_CNN: Static Block Floating Point Quantization for CNN (github.com)](https://github.com/os-hxfan/Static_BFP_CNN/tree/master)

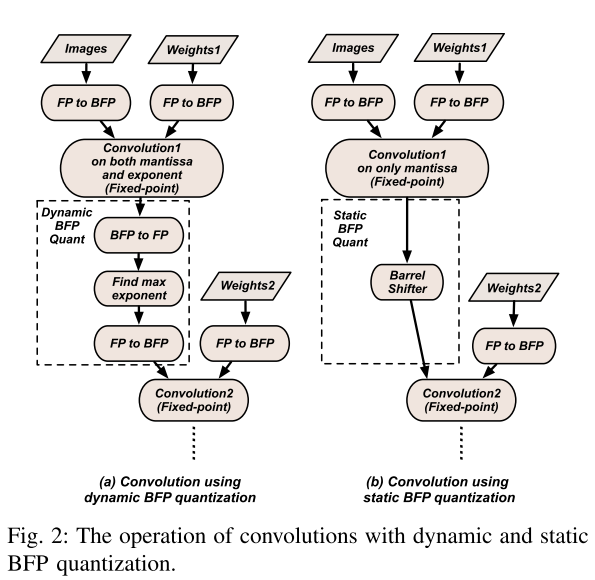
* 挑战：

1. 多数加速器只专注于加速2D CNN，而没有考虑它们在3D CNN上的计算效率。因为它们具有不同的数据局部性，3D卷积的内存和计算密集度更高，统一到一个硬件架构上很难
2. 为了提高硬件性能，基于 FPGA 的加速器倾向于使用低精度的权重或激活。然而，以前的方法要么导致显著的精度损失，要么需要大量的硬件资源来实现量化模块，例如线性量化、动态BPF量化。

* 解决：

1. 所提出的静态BFP量化将2D和3D CNN中的权重和激活都转换为8位尾数，同时保持相同的准确度。

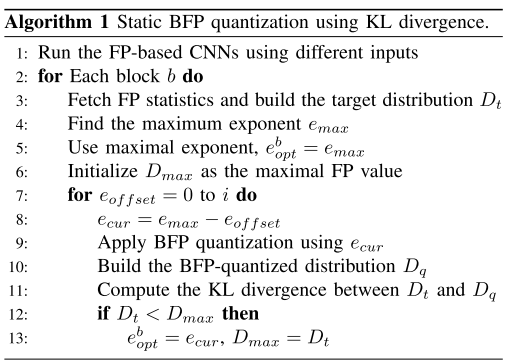
动态量化需要额外的硬件资源来计算和存储指数部分，



静态量化为不同输入固定共享指数，以消除BFP和FP之间的频繁转换，并在CNN推理之前确定共享指数。简单地使用收集的中间统计数据中的最大指数是确定共享指数的一种方法，但是可能导致模型(mobilenet)准确率下降

本文通过最小化描述两个分布之间差异的KL散度来确定共享指数。在开始时，根据块大小将中间结果分成几个块。（由于空间KxW和通道Nc数据要累加，这样分块需要频繁的指数对齐，所以本文沿时间维度分块激活 NI个 这样不需要对单个卷积层执行指数重新对齐 ）。对于每个块，可以通过在示例输入上运行CNN来基于收集的中间FP统计数据绘制直方图，该示例输入用于记录原始分布。然后，基于i个不同的指数emax至emax−i + 1对i个块应用BFP量化，这产生i个不同的BFP量化分布。我们的实验表明，在大多数情况下，i = 3足以找到适当的指数。当完成BFP量化时，我们计算针对i个不同指数的BFP量化分布和FP分布之间的KL散度。然后，将共享指数设置为具有最小KL散度的指数。我们在每一层上迭代地执行这个过程，以获得整个网络的共享指数。

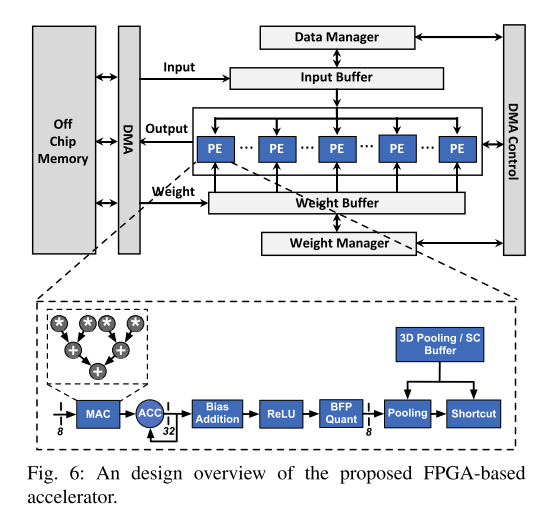
对有共享指数a和b的两个连续卷积层，它只需要执行将尾数部分移位a-b位的指数重新对齐，这避免了琐碎的数据转换。·由于共享指数是已知的，因此可以在运行时间之前预先计算每个层所需的移位比特。



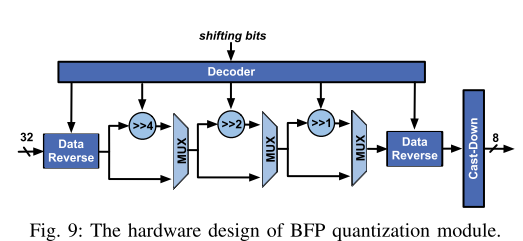
1. 提出了一种统一的硬件架构，以高硬件效率加速2D和3D CNN。

通过在空间上计算多通道PC和滤波器PF，并在时间上映射kt（时间维度）、kh和kw循环中的计算，来利用通道和滤波器并行性，从而将2D和3D卷积都统一到一个硬件设计上，因为与第三维相关的所有计算都是在时间上执行的。

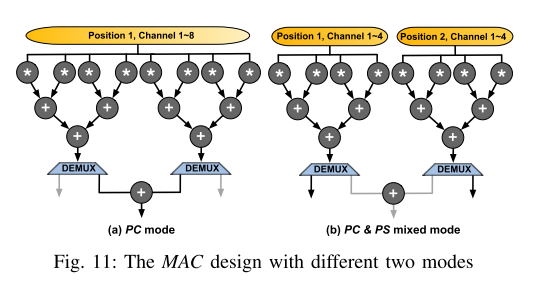
每个MAC包含PC数量的乘法器，后面是log 2 PC级加法器树。因为静态BFP量化在运行时使用固定的共享指数，所以我们只需要对相应的尾数部分执行计算。



由于共享指数在不同的层中可能会有所不同，因此我们需要使用属于下一层的共享指数来对齐当前卷积层生成的输出。此外，由于当前层的共享指数可能大于或小于下一层中使用的指数，因此我们的BFP量化模块需要支持右移和左移操作。在本文中，我们采用基于掩码的数据反转桶形移位器设计。为了将32位移位结果截断为8位，我们在BFP量化模块的末尾放置了一个castdown模块以生成8位结果。向下投射模块简单地从32位移位结果中提取高8位



硬件优化：当通道数小时，大多数乘法器和加法器是空闲的，因为PC总是被设置为相对较大的值，例如32或64。设计MAC支持具有不同并行组合的两种模式，将MAC设计划分为几个子树，并在每个子树的尾部放置DEMUX模块。当信道号Nc较大并且可被PC整除时，PC模式 每个子树的数据聚合在一起。当通道数小于PC时，PC&PS模式 不同位置的数据馈送到不同的子树中，直接输出来自每个子树的结果



3）自动化工具包括两个优化：1）硬件优化，它根据硬件平台的资源模型、延迟模型和可用资源来确定硬件参数，例如包括PC和PF的并行级别。2)软件优化，确定软件参数，如每层的共享指数，以优化给定数据集的准确性。