申请人：浙江大学

发明人：黄科杰 杨树园 沈海斌

第一发明人身份证号：330222198007126952

联系人电话及邮箱：17706443800 huangkejie@zju.edu.cn

寄信地址及邮编：浙江大学玉泉校区老生仪楼3楼    邮编310027

**专利申请提交的技术交底材料**

一、名称

一种低功耗存储器内计算处理器架构  
二、技术领域

本发明公开了一种低功耗存储器内计算处理器架构，涉及存储器内计算技术和软硬件协同设计技术。  
 三、背景技术

传统冯诺依曼架构中，存储器与计算单元分离，完成计算需要从存储器读取数据，送入计算单元执行计算，最终把结果写回存储器，其中存储器访问操作消耗了大部分能量。存储器内计算技术中，存储器除原有的存储功能外，嵌入了计算功能，从而大大减少了存储器访问的次数，降低了数据搬移的功耗。并且存储器内计算采用模拟电路执行计算，与传统由数字电路实现的计算单元相比，计算功耗更低。虽然存储器内计算技术在能效比等方面取得了突破性优势，但是如何利用存储器内计算技术灵活地完成不同算法的执行还是一个挑战。

网络映射技术通过设计数据流，规定了神经网络算法中循环嵌套的顺序和并行度展开情况，使得硬件可以高效地执行神经网络算法。目前主流的数据流包括行固定数据流，输入固定数据流，输出固定数据流等。存储器内计算技术中，由于权重存储在存储器中，因此具有权重固定的特性。高效的数据流可以充分利用数据重用和硬件的并行度，提高硬件性能。网络映射技术同时也规定了网络参数与存储器内计算核指定位置的对应关系。

流水线技术可以实现多条指令的重叠执行，与单周期指令相比，流水线可以增加指令的吞吐率，提高处理器的性能。

四、发明内容

本发明旨在利用存储器内计算低功耗、高能效比的优势，设计低功耗存算处理器。

本发明基于存储器内计算技术，设计了一种低功耗存算处理器架构，使用存储器内计算核、向量加法器和除法器组成向量算术处理单元。片上存储器全部使用寄存器组实现。设计了专用指令集，指令采用4级流水线，包括取指、译码/读取、执行、写回4个阶段，可以实现矩阵向量乘法、向量加法、向量常数除法和激活等基本操作。支持卷积网络、全连接网络、残差网络等多种网络结构。处理器通过使用power gating、clock gating和低电压供电等低功耗技术，具有低功耗休眠模式，可以结合使用场景，在休眠模式和正常工作模式之间灵活切换。

本发明一种低功耗存算处理器工作流程，如图1：

S01:输入网络模型，包括网络各个层的结构，以及网络各层大小。

S02:对网络进行优化、映射。优化包括在保证网络推理准确度的前提下，使用神经网络量化技术进行低比特量化；若网络存在残差层，进一步统一残差层量化因子；结合处理器的数据位宽，对网络各层大小进行调整，使网络结构更加规整。映射即完成数据流的设计，并完成网络各层参数到存储器内计算核指定位置的排列，最终结果保存到参数文件中。

S03：网络编译，根据专用指令集，由编译脚本把网络编译为指令，保存到指令文件中。

S04：指令、参数载入处理器，把S02得到的参数文件，以及S03中得到的指令文件从输入接口载入处理器，其中指令存储在指令寄存器中；参数在经过擦除，读取，写入3个阶段后最终以阻值的形式存储在存储器内计算核中。

S05：处理器正常工作。

具体技术方案如下：

低功耗存算处理器结构如图2所示，包括输出接口、输入接口、指令寄存器、译码器、寄存器组1和向量处理单元。

指令寄存器存放S03步骤中网络编译后的指令，由于神经网络算法具有重复执行的特性，因此编译后的指令具有周期性。在指令寄存器中只需保存一个周期长度的指令。

译码器每个周期从指令寄存器中循环读取指令，对指令进行译码，输出控制信号。

向量处理单元包括：存储器内计算核、向量加法器和除法器。

存储器内计算核如图3所示，包括寄存器组2、MUX1、MUX2、MUX3、存储器内计算阵列、计算和读取电路。存算核可以完成向量矩阵乘法形式的计算，寄存器组2存储向量，对应网络的特征图，MUX1实现位选择，从reg的多位中选择其中一位，实现输入特征的串行输入；MUX2将数字电平转换为模拟电平；矩阵存储在存储器内计算阵列中，对应网络的参数值，存储器类型可以为Flash、SRAM、RRAM、PCRAM等，规格为,即p根字线，q根位线。MUX3为模拟选通器，从存算阵列的多块输出中选择其中一块；计算和读取电路负责完成积分等运算，并把计算结果以数字信号的形式输出。

向量加法器和除法器由数字电路实现，向量加法器支持向量的加法、减法操作。除法器支持向量与常数的除法操作。

五、附图说明

图1低功耗存算处理器工作流程图

图2低功耗存算处理器整体架构图

图3存储器内计算核结构图

图4指令流水线图

图5网络映射图

图6 TC-resnet8网络映射图

图7 TC-resnet8编译执行图   
六、具体实施方式

指令包含4级流水线：取指，译码/读取，执行，写回。如图4所示。

取指阶段：根据指令计数器，从指令寄存器中读取对应的指令，送入译码器中；

译码/读取阶段：译码器根据指令集，对指令进行解码，得到各个模块的控制信号；若此指令中包含读寄存器组操作，则发出读请求信号和读地址信号。

执行阶段：共3种操作类型，向量矩阵乘法、向量加/减法和向量常数除法，由译码得到的控制信号决定执行哪一种操作。

写回阶段：把指令执行的结果写回到寄存器组中，即发出写请求信号和写地址信号。

使用流水线技术，流水线每级执行时间需要相同，由于向量除法操作以及向量矩阵乘法占用的周期数最长，因此采用分频技术，令取指、译码/读取、写回和执行阶段的加法操作使用低频时钟，执行阶段的向量矩阵乘法、除法操作使用高频时钟，从而使得流水线各阶段均可以在一个统一的低频时钟周期内完成。

存算阵列分为多个存算块，一个或多个存算块可以容纳一层网络的权重。由于整个存算阵列共用寄存器组2，如图3所示，因此同一时间只有一个存算块处于工作状态，其它块处于低功耗的休眠状态。通过MUX3的选择信号决定当前工作的存算块。每个块可以支持个权重，即最大可以支持大小的向量与大小的矩阵进行乘法。

本处理器中的向量采用统一位宽,因此在S02步骤中，对网络大小进行优化时，会尽量使得网络向量位宽为处理器统一位宽的整数倍，网络中的某一向量在处理器中占用的存储地址个数为, 如公式(1)所示;经过优化后，若网络仍有部分层的向量位宽不是统一位宽的整数倍，具体处理办法为：对于位宽小于统一位宽的向量，使用补零的方式使其位宽增加至统一位宽；对于位宽大于统一位宽的向量，则采用切分+补零的方式，存放到多个地址中。

(1)

映射方案为采用权重固定数据流，对网络进行im2col变换，把卷积计算转换为向量矩阵乘法,如图5所示，输入特征图大小为的3维矩阵,权重大小为的4维矩阵,经过im2col变换后，权重转换为大小为的2维矩阵，输入特征图的每个卷积窗口都转换为大小为的向量。一个卷积窗口的计算转换为向量矩阵乘法，而存储器内计算核支持的计算形式为向量矩阵乘法，因此经过im2col变换后，卷积计算可以方便地映射到存储器内计算核上。若一块存算核中无法容纳权重矩阵，则采用切分的方式，把其分到多块存算核中，某一层权重需要的块数由公式（2）决定,若k维度超出，则直接使用多块存算核；若维度超出，则除了使用多块存算核外，还需要使用加法器完成块之间的加法计算。

(2)

残差层的优化方法为：残差层可以分为主路径和分支路径，主路径和分支路径的结果需要进行元素级相加，得到残差层的最终结果。由于主路径和分支路径的量化因子一般不同，所以需将主路径和分支路径映射到不同的存算块中，元素级相加需要使用加法器执行；S02中可以通过网络训练的优化，使得主路径和分支路径的量化因子相同，如公式所示，公式中的f函数代表量化操作，分别代表主路径和分支路径的输入特征向量，分别代表主路径和分支路径的权重矩阵，量化因子由原来不同和调整为相同的后，在存算核大小允许的条件下，主路径和分支路径可以映射到同一存算块中，直接利用存算核完成元素级相加，避免了使用向量加法器，从而降低功耗。

(1)

本处理器可以灵活运行多种神经网络算法，下面运行语音关键词识别网络TC-resnet8为例，进一步阐述处理器工作流程。

TC-resnet8的结构如表一所示，该网络为时序卷积网络，可以实现语音关键词识别功能。

该网络向存储器内计算核的映射情况如图6所示，图中阴影部分表示映射后实际使用的部分，空白部分表示闲置部分，由于CONV1的输入向量位宽大于统一位宽且不能整除，因此采用了切分+补零的方式，3个残差层在S02步骤中分别调整为量化因子相同，因此CONV2\_2和 CONV2\_b、CONV3\_2和 CONV3\_b、CONV4\_2和 CONV4\_b均映射到了同一块存算核中。

在S03步骤网络编译过程中，TC-resnet8具有以8帧（每帧为10ms）为周期重复执行的特性。如图7所示，横轴代表帧数方向，从左往右帧数递增；纵轴代表每帧展开情况，从上到下时间递增，表示一个10ms时间段内处理器先后执行的任务情况。其中方框代表某一次卷积计算，上面的字母和数字代表了执行计算的层的编号；阴影代表了处理器进入休眠状态。如第二列表示在第二帧内，处理器需要首先完成CONV1层的一次计算，然后依次执行CONV2\_1的一次计算、CONV2\_2+ CONV2\_b的一次计算，然后进入休眠状态。对于TC-resnet8网络来说在每个10ms间隔内，低功耗的休眠状态占据了绝大部分时间。因此处理器可以通过运行TC-resnet8网络实现低功耗的实时关键词检测。具体帧数的时长和休眠时间的占比与具体的网络有关。

|  |  |  |
| --- | --- | --- |
| 层数 | 编号 | 层结构 |
| 1 | CONV1 | 卷积层 |
| 2 | CONV2\_1 | 残差层1主路径第一个卷积层 |
| 3 | CONV2\_2 | 残差层1主路径第二个卷积层 |
| 4 | CONV2\_b | 残差层1分支路径卷积层 |
| 5 | CONV3\_1 | 残差层2主路径第一个卷积层 |
| 6 | CONV3\_2 | 残差层2主路径第二个卷积层 |
| 7 | CONV3\_b | 残差层2分支路径卷积层 |
| 8 | CONV4\_1 | 残差层3主路径第一个卷积层 |
| 9 | CONV4\_2 | 残差层3主路径第二个卷积层 |
| 10 | CONV4\_b | 残差层3分支路径卷积层 |
| 11 | POOL | 平均池化层 |
| 12 | FC | 全连接层 |

表一



图1



图2



图3



图4



图5



图6



图7