DOI:10.16136/j.joel.2022.12.0089

基于近存储计算的手写数字识别实时检测阵列结构设计

霍紫晴,山 蕊*,冯雅妮,高 旭,冯 煜

(西安邮电大学 电子工程学院,陕西 西安 710121)

摘要:卷积神经网络(convolutional neural network,CNN)作为传统神经网络的改进,已经得到了广泛的应用。然而,在CNN性能提升的同时其模型的规模不断扩大,对存储及算力的要求越来越高,基于冯·诺依曼体系结构的处理器难以达到令人满意的高处理性能。为了提升系统性能,近存储计算(near memory computing,NMC)成为了一个具有发展前景的研究方向。本文利用一种支持NMC的可重构阵列处理器实现手写数字识别,并行地实现了卷积运算;同时利用共享缓存阵列结构,减少片外存储的频繁访问。实验结果表明,在110 MHz的工作频率下,执行单个5×5卷积运算的计算速度提升了75.00%,可以在9960 μs内实现一个手写数字的识别。

关键词:卷积神经网络(CNN);手写数字识别;可重构阵列处理器;近存储计算(NMC);共享缓存阵列

中图分类号:TP391.4 **文献标识码**:A **文章编号**:1005-0086(2022)12-1315-08

Real-time detection array structure design based on near memory computing for handwritten digit recognition

HUO Ziqing, SHAN Rui*, FENG Yani, GAO Xu, FENG Yu

(College of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China)

Abstract: Convolutional neural network (CNN) has been widely used to improve traditional neural networks. However, as the performance of the CNN improves the size of its model increases, which requires larger storage and more computing power, so the processors based on the von Neumann architecture is difficult to achieve satisfactory processing performance. In order to improve the performance of the system, near memory computing (NMC) is a promising research direction at present. In this paper, a reconfigurable array processor-based NMC structure is used to implement handwritten digit recognition, and convolution operations are realized in parallel; at the same time, the shared cache array structure is used to reduce off-chip storage access. Experimental results show that with the operating frequency of 110 MHz, the calculation speed of a single 5×5 convolution operation is increased by 75.00%, and a handwritten digit can be identified within 9960 μs.

Key words: convolutional neural network (CNN); handwritten digit recognition; reconfigurable array processor; near memory computing (NMC); shared buffer array

1 引 言

实时手写数字识别在银行、教育、邮政等实际生活中的广泛应用使之成为图像处理和模式识别领域的一个研究热点。手写数字识别是一个图像分类问题,目前图像分类的方法大致可以分为传统图像分类方法、经典机器学习方法以及深度学习方法3大类,其中深度学习方法不用和传统图

* E-mail:shanrui0112@163.com

收稿日期:2022-02-17 修订日期:2022-03-30

基金项目:国家自然科学基金(61802304,61834005,61772417,61602377)资助项目

像分类方法一样需要人工提取复杂的特征和重建数据；与经典机器学习方法相比，显著提高准确性[1]。卷积神经网络（convolutional neural network，CNN）作为深度学习的主要内容，因其具有较强的函数表达能力和网络泛化能力，识别率往往能够超过传统图像分类方法，因此，通过CNN实现实时手写数字识别的研究具有重要的意义。但是，随着算法的精进和应用场景日益多元化，CNN在识别准确率提高的同时，其结构变得越来越复杂、深度也不断加深[2]。CNN计算密集型和存储密集型的特点逐渐成为限制其快速、高效实现的主要原因[3]。

为进一步提高CNN的计算性能，基于中央处理器（central processing unit，CPU）、图形处理器（graphics processing unit，GPU）、专用集成电路（application specific integrated circuit，ASIC）、现场可编程门阵列（field programmable gate array，FPGA）的加速器相继被提出。文献[4]中采用并行化的多核CPU实现了3种不同的CNN模型，减少了计算时间，提高了吞吐量；但在面对大量的数据时，CPU的处理速度仍然不能满足需求。GPU并行计算能力强且速度快，文献[5]中实现了基于GPU的二值化神经网络，减少了计算量和资源占用，显著减少了运行时间；然而，基于GPU的加速器存在着功耗大的问题，且GPU硬件结构固定，限制了CNN在嵌入式结构上的应用。ASIC加速器具有低功耗、计算性能高的特点，因此，基于ASIC实现CNN成为了主要研究方向之一。文献[6]设计了一种面向神经网络的芯片。该芯片在28 nm互补金属氧化物半导体（complementary metal oxide semiconductor，CMOS）技术中实现，它主要由神经网络和特征提取电路组成。实验结果证明：在40 kHz频率下功耗仅为0.51 μW；然而ASIC芯片灵活性极低，研发成本极高，开发周期较长，而不同CNN输入特征图的形状和核大小在不同的卷积层中是有区别的。因此，基于ASIC的加速器不适用于结构灵活多变的CNN。对比于ASIC，FPGA具有计算速度快、灵活配置的特点，文献[7]采用软硬件协同设计的方式实现了一种基于FPGA的网络加速器；在确保识别精度不变的情况下，推理速度有了大幅度提升，但将输入特征图加载到片外存储中，存在频繁访问存储的问题。而可重构阵列处理器兼顾通用处理器的灵活性和ASIC的计算能力，逐渐成为实现CNN高效能计算的一个具有发展前景的研究方向。文献[8]提出了一种具有72个处理元（process element，PE）的可重构的CNN硬件加速器结构，该加速器是一种基于层的体系结构，为方便适应不同的CNN参数可以重新配置，结果表明：该结构有效降低了带宽利用率，提高了数据重用效率。

虽然可重构阵列处理器的计算高效性以及功能灵活性可以满足CNN这类数据密集型应用的需求，但采用传统的存储结构实现CNN时需要与外部存储进行大量的数据通信，造成较大访存开销，因此，近存储计算（near memory computing，NMC）将逐步兴起[9]。本文在已有的可重构阵列处理器的基础上，利用NMC阵列和数据并行化计算，提高了CNN推理过程的计算速度。利用共享缓存阵列有效降低了处理器与主存之间的数据通信，同时实现了手写数字的实时检测。

2 基于可重构阵列处理器的NMC结构

基于可重构阵列处理器的近数据计算结构如图1所示。本文在此结构上实现手写数字的实时检测，该结构可分为两部分：NMC阵列结构和缓存阵列结构。NMC阵列结构由主处理器和协处理器（coprocessor，cope）构成，主处理器端为可重构阵列处理器，协处理器端为NMC单元。共享缓存阵列和双倍速率同步动态随机存储器（double data rate，DDR）用于实现数据高速并行访问。

每个主处理器对应一个协处理器。上位机通过H树网络下发配置信息，在主处理器中对下发的32 bit配置信息进行译码，由指令的高6 bit为标志判断该指令是否需要下发到协处理器，若高6 bit未指向协处理器，则在主处理器中进行操作，协处理器不工作，若指向协处理器，则将除高6 bit的其他信息下发给协处理器进行处理。相邻PE之间通过邻接互连的方式，将东、南、西、北4个方向的邻接寄存器互连，通过邻接寄存器的数据拷贝实现数据共享。

可重构阵列处理器由4×4个PE组成阵列处理器，16个PE均采用Load/Store模式的精简指令集（reduced instruction set computer，RISC）结构进行设计。每个PE由寄存器、算术逻辑单元、数据/指令存储器和程序计数器更新单元组成，数据存储器容量为16×32 bit，指令存储器容量为32×512 bit。

协处理器端由16个cope单元组成，每个cope采用三级流水线结构，由指令寄存单元、译码取数单元和计算单元组成。cope通过和可重构阵列内

PE 进行数据通信实现功能重构，并且能够从存储中取数进行计算，对存储内数据进行更新，实现加速计算的功能。同时，协处理器支持的大部分指令都可以实现一条指令完成从存储中取数、计算和写回，相比于 PE 采用的 Load/Store 模式进一步提升了计算速度；针对神经网络中的卷积运算，在协处理器中设计专用的乘累加指令以及累加寄存器可以有效提高卷积的计算速度。表 1 列出了部分特殊指令，其中 M[Rs]和 M[Rt]表示直接从存储中读取操作数；M[Rd]表示直接将计算结果写回存储中；RM 表示乘累加寄存器。

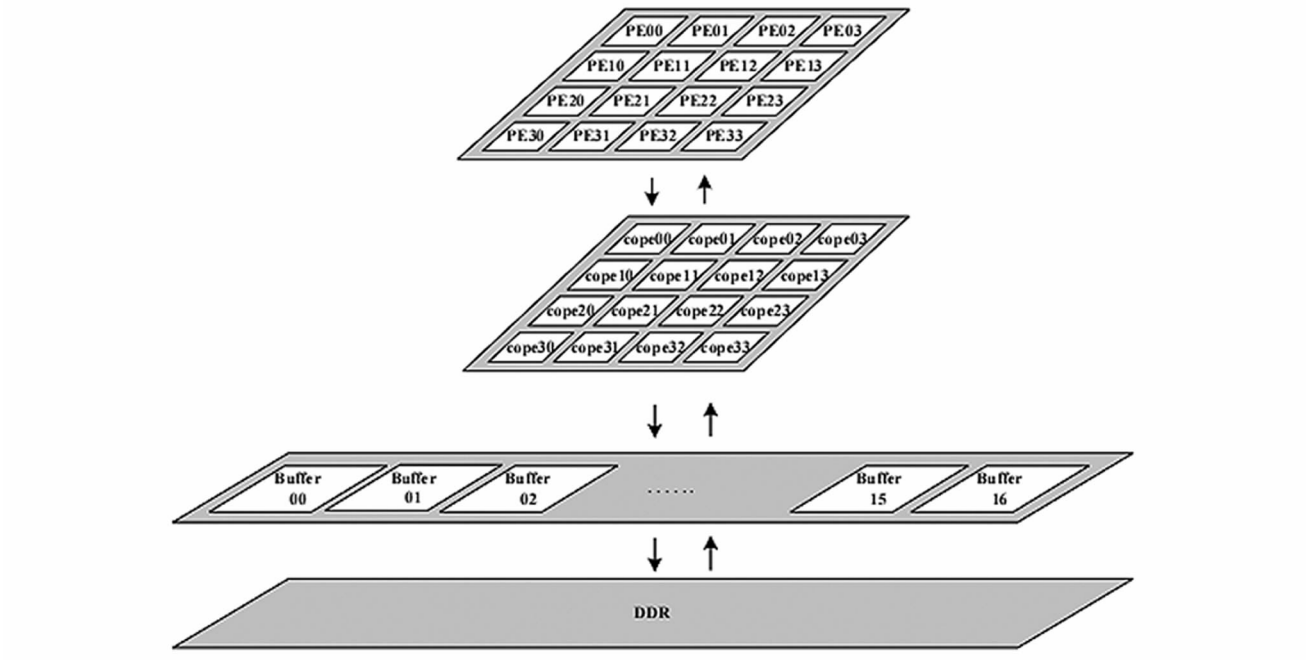

图 1　基于可重构阵列处理器的 NMC 结构

Fig. 1　Near memory computing structure based on reconfigurable array processor

表 1　协处理器支持的部分特殊指令

Tab. 1　Some special instructions supported by the coprocessor

Operation	Description	Function
ADDM0	Rd←Rs+M[Rt]	Register addition
ADDM1	Rd=M[Rs]+M[Rt]	Register addition
ADDM2	M[Rd]←Rs+M[Rt]	Register addition
ADDM3	M[Rd]=M[Rs]+M[Rt]	Register addition
MAC2	RM←M[Rs]macM[Rt]	Multiply and accumulate
STRM2	M[Rd]←RM	Write to memory

共享缓存阵列结构由 buffer 和 DDR 组成。共享缓存阵列包含 17 个支持非对齐存储访问的 buffer，每个 buffer 可缓存 16×32 bit 的数据且共享缓存阵列可同时接收 16 个协处理器的访问请求。当协处理器发出访问请求，共享缓存将缓存以访问地址为首地址的连续 16 个数。由于计算过程中数据访问的局部性，当缓存完成后，协处理器发出访问请求时仅需从共享缓存阵列中获取操作数，无需频繁访问存储。算法实现过程中的原始数据、中间计算结果和最终数据都存储在 DDR 中，DDR 可以在时钟上升沿和下降沿各传输一次数据，在一个时钟周期内传输两次数据，因此，与其他的片外存储相比，其有较高的数据传输率。

3　手写数字识别

3.1　CNN 网络结构

本设计所采用的 CNN 基本结构如图 2 所示，依次包含输入层、第一个卷积层 C1、第一个池化层 S2、第二个卷积层 C3、第二个池化层 S4 和全连接输出层 F5。输入分辨率为 28×28 的灰度图像，经过 C1、S2、C3、S4 各层的卷积池化操作，最后经过 F5 层输出预测结果。输入图像在 C1 层分别与 6 个 5×5 的卷积核进行卷积，卷积过程中步长为 1，第一层卷积完成后得到 6 个 24×24 的特征图，将 6 张图像中每个像素点与对应的偏置相加并通过线性整流函数(rectified linear unit，ReLU) 进行激活，得出第一个卷积层 C1 的 6 个 24×24 的输出特征图。S2 为池化层，采用最大池化方式，既可以减少预算复杂程度又避免丢失神经元信息。采样模板大小为 2×2，采样步长设为 2，对输入图像的每个 2×2 区域进行最大

值采样，由于采样步长为2，相邻采样窗口，无重叠区域。该层输出6个12×12的特征图像。C3层与C1层类似，6个12×12的特征图像中的每一个，分别与12个5×5的卷积核进行卷积操作，卷积步长为1，得到12个6×8×8的卷积图像，将相同通道的图像中像素点依次相加，再加上偏置后，经过激活函数，从而得该层12个8×8的输出特征图像。S4层与S2层类似，采样模板大小为2×2，采样步长设为2。该层输出12个4×4的特征图像。在F5层，将S4输出的192个神经元展成一维向量的形式，作为F5层的输入，输出为10个神经元的全连接单层神经网络。具体网络结构如表2所示。

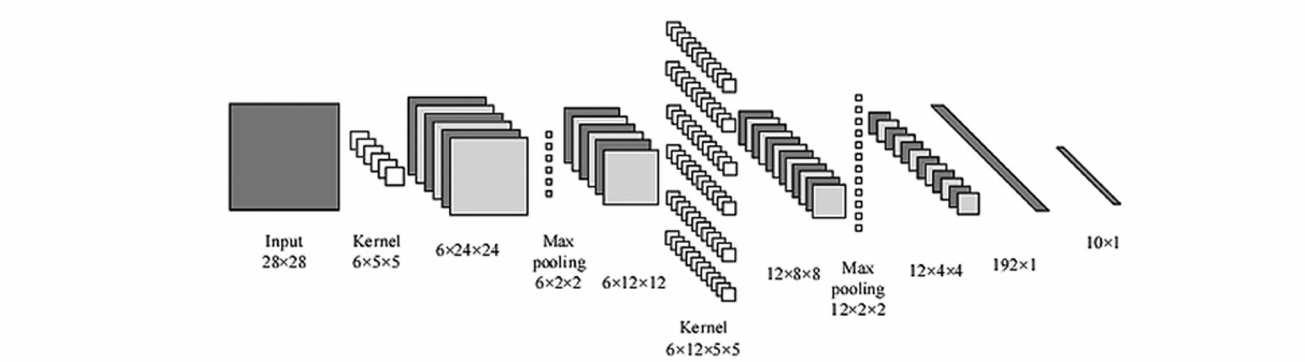

图2　CNN网络结构

Fig. 2　CNN network structure

表2　网络具体结构

Tab. 2　Network structure

Layer	Layer type	Input size	Weight	Stride
C1	Convolution	28×28	6×5×5	1
S2	Max pooling	24×24×6	2×2	2
C3	Convolution	12×12×6	6×12×5×5	1
S4	Max pooling	8×8×12	2×2	2
F5	Fc	192	192×10	—

3.2　手写数字识别结果

手写数字识别数据集选用MNIST数据集，其中训练集有60 000幅图像，测试集有10 000幅图像，每幅图像均是像素为28×28的灰度图像，在PC机上迭代5次后准确率达到95.37%，图3为部分数字的识别结果，图右下角为其对应输入图像的预测值。

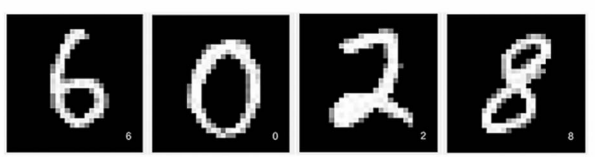

图3　手写数字识别结果

Fig. 3　Handwritten digit recognition results

由于神经网络中通常有大量的参数，一般模型内部的计算都采用了浮点数计算，浮点数的计算会消耗比较大的计算资源，如果在不影响模型准确率的情况下，模型内部可以采用其他简单数值类型进行计算的话，可以减小存储空间的使用，计算速度会提高很多，消耗的计算资源会大大减小，通过分析卷积核的数据范围，数据多集中在[−1，1]区间内，因此，采用对称量化的方式，将训练后的32 bit浮点数表示的卷积核和偏置量化为8 bit有符号整数。

4　手写数字识别实现过程

4.1　手写数字识别并行化设计

卷积运算过程是指用较小的卷积核对图像进行遍历，卷积核覆盖部分对应点相乘，结果进行累加，然后对输入图像按照特定步长滑动，继续进行对应点的乘累加。卷积过程就是从输入特征图中提取出各种特征，随着卷积操作次数的增加，可以从输入特征图中提取出更复杂的特征，图4是第一层卷积层中6个通道的卷积核对数字5提取的6张特征图。

图4　数字5的6张特征图

Fig. 4　Feature maps of the number 5

根据卷积操作及NMC阵列的特点，由协处理器执行卷积过程中的访存计算指令，主处理器负责卷积运算中的循环控制。因为协处理器中执行的访存计算指令操作数来自于主存，且主存存储容量太大，所以采用寄存器间接寻址的方式对主存存储空间进行访问。在执行CNN网络卷积运算之前，原始图像数据、权重和偏置都被存放于DDR中。所有计算指

令和配置信息会由 H 树下发至主处理器,主处理器根据配置信息将对应指令下发至协处理器的指令寄存单元,由指令寄存单元缓存主处理器下发给协处理器的指令;当译码单元接收到指令后会对其进行译码,根据译码结果读出指令执行所需要的源操作数;执行写回单元接收到译码单元发送的源操作数并执行计算指令,最后将计算的结果写回主存或共享缓存阵列中。

CNN 中的 90%的计算为卷积运算且卷积运算过程是整个网络计算中最复杂、最耗费计算资源的部分,当卷积计算过程可以并行化实现时,计算效率将会得到大幅度提高。因此,本文根据 NMC 阵列结构特点和 CNN 的潜在并行性设计了一种并行计算方式:由 12 个协处理器同时处理两个 5×5 的卷积运算,其中 10 个协处理器执行 1×5 卷积运算,2 个协处理器对计算的中间结果进行累加,从而 CNN 卷积计算的并行度,达到提高处理器的计算效率的目的。

4.2　手写数字识别并行化实现

CNN 的经典网络模型由卷积层、池化层以及全连接层等结构组成。本文通过分析 CNN 结构特点来实现卷积过程并行化计算。C1 层输入图像大小为 28×28,卷积核大小为 5×5,通道数为 6。映射结构如图 5 所示,PE00 单元到 PE21 单元主要做的是当前通道特征图与卷积核的卷积计算,而 PE30 和 PE31则是将中间计算数据累加得到最终的卷积结果。

当指令被下发,且指令所需要的操作数均已准备好后,PE00-PE10 将对该层输入图像 1 个卷积核进行乘累加操作:PE00 进行该层输入第 1 行的 1×5 的乘累加操作,PE01 进行输入图像第 2 行的 1×5 的乘累加操作,以此类推,PE10 完成输入图像的第 5 行的 1×5 的乘累加操作;同时,PE11-PE21 完成该层输入图像与第 4 个卷积核的乘累加操作。当 PE00-PE10 分别做完第一个 1×5 的卷积运算后,会依次向 PE30 发送握手信号。PE30 依次与 PE00-PE10 握手并读取当前 PE 中的 1×5 卷积结果,将卷积结果进行累加后给对应的 PE 发出握手信号,直到一个 5×5 卷积结果累加完成,由 PE30 将累加结果写回主存。当 PE00-PE10 接收到来自 PE30 的握手后,向右滑动进行下一次 1×5 的卷积。以此类推,PE11-PE21 会与 PE31 进行握手并交互数据。PE00-PE21 这 10 个 PE 中用于暂存中间计算数据的地址可以被重复覆盖。当得出 24 个卷积结果后,PE00-PE21 会将卷积窗口向下滑动 1 行,继续进行卷积运算,6 个通道依次进行卷积最终得出 24×24 大小的 6 张特征图。PE30 和 PE31 将所有卷积结果写回存储后,由 PE22 对卷积结果加上对应的偏置;PE23 对加上偏置的结果进行 ReLU 函数激活,将小于零的部分抛弃;PE32 对激活后的数据进行池化并写回存储,最终得到 12×12 大小的 6 张特征图。

C3层输入图像大小为12×12,卷积核大小为 5×5,通道数为 12,每个通道都有 6 个卷积核。映射结构见图 6,PE00-PE10 和 PE11-PE21 分别计算完前 6 个通道和后 6 个通道的卷积,当得出第一行的 8 个卷积结果后,PE00-PE21 会将卷积窗口向下滑动 1 行,继续进行卷积运算,12 个通道中每个通道的 6 个卷积核都和 6 张特征图相对应进行卷积,一个通道计算完后得到 8×8 大小的 6 张特征图,直到 PE00-PE10 和 PE11-PE21 分别计算完前 6 个通道和后 6 个通道的卷积,最终得到 72 张特征图,PE30 和 PE31 将所有卷积结果写回存储后;由 PE33 将 12 个通道各自的特征图进行融合得到 12 张 8×8 大小的特征图;由 PE22 对卷积结果加上对应的偏置;PE23 对加上偏置的结果进行 ReLU 函数激活,将小于零的部分抛弃;PE32 对激活后的数据进行池化并写回存储,最终得到 4×4 大小的 12 张特征图。

F5 层为全连接层,输入大小为 1×192,权重大小为 192×10,具体计算为简单的向量乘操作,由 PE33 执行,最终将得到的 10 分类结果写回存储的 0—9 号地址,对 10 个数依次进行比较,将数值最大的数据对应的地址输出作为最终识别结果。

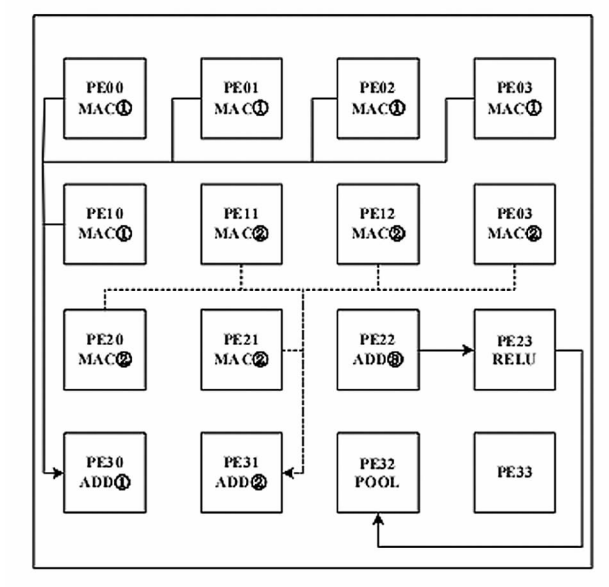

图 5　第一层卷积,池化映射图

Fig. 5　Level 1 convolution, pool operation map

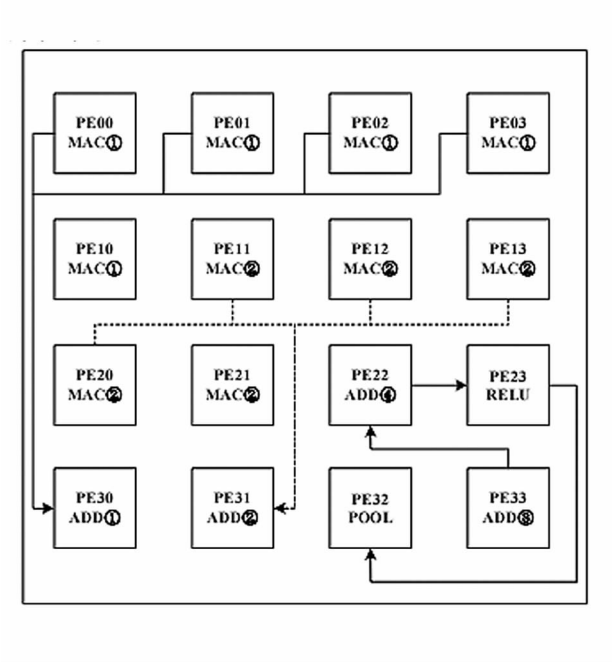

图 6　第二层卷积,池化映射图

Fig. 6　Level 2 convolution, pool operation map

5　实验结果及分析

本实验使用 Xilinx 公司的 ISE 进行硬件开发,使用 Modelsim SE-64 10.5 进行功能仿真验证,5×5 卷积仿真波形仿真结果如图 7 所示,输入图像为手写数字 5,在 1 095 624 个时钟内完成数字识别,识别结果为数字 5 仿真结果正确。

通过 Xilinx ISE 对结构进行综合,表 3 列出了综合后的 FPGA 芯片的资源使用情况。

完成一个 5×5 卷积消耗 107 个时钟周期,与文献[10]相比,性能提升了 75.00%。

表 4 将本文结构与文献[11]－[13]进行比较。文献[11]提出了一种阵列结构的 CNN 硬件加速器,并且可重新配置层参数以适应不同的 CNN 结构。该体系结构同时对 3 行输入特征图进行卷积,最后生成一行输出特征图,通过采用多个 PE 同时进行卷积操作的方法提高计算并行度,进一步提高卷积处理速度,该结构下电路工作频率可达 100 MHz。文献[11]提出了一种用于卷积计算的粗粒度可重构

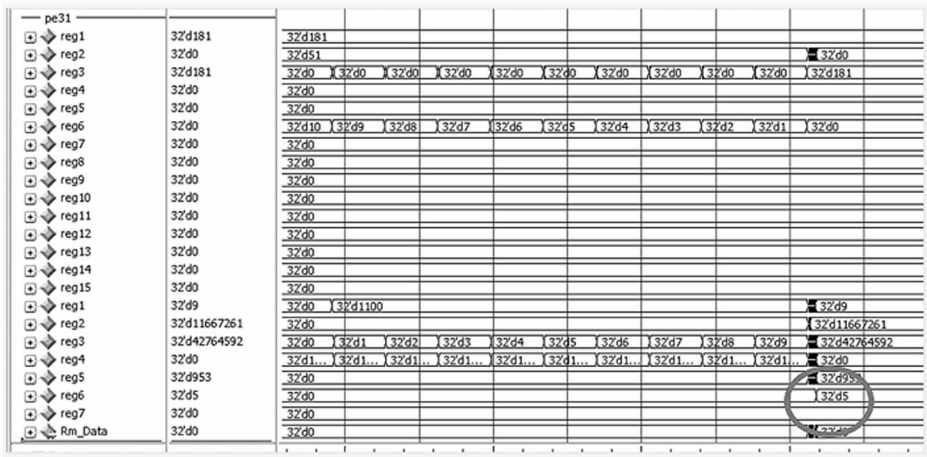

图 7　数字 5 识别仿真结果

Fig. 7　Simulation results of digit 5 recognition

表 3　硬件资源使用情况

Tab. 3　Hardware resource usage

Logic devic	Number of resource occupied	Resource utilization/%
Slice LUTs	120 317	35
Slice registers	39 290	5
LUT-FF pairs	22 992	16

计算结构,通过对数据量化过程进行配置降低能耗和片内存储器需求,但并不具有通用性。文献[13]设计了一种高效、可重用的 CNN FPGA 加速器,通过优化 CNN 内的多级数据并行以及细粒度和粗粒度流水线并行,最大限度地提高了 FPGA 的计算能力,且电路最高工作频率可达 150 MHz,通过 8 bit 定点操作实现了卷积层,但同时资源消耗与本文相比也占用更大,本文的最高工作时钟频率可达 110 MHz,相比于文献[11,12]有大幅度提升,且精度较高。

图 8 对本文手写数字识别时间进行比较。文献[14]提出了一种在加快网络预测阶段的同时保持其准确性的架构,对 CNN 中的每一种层都有固定的结

构,其整体结构可扩展,更容易用于较大的网络,结合并行化 Form2 实现方案,可以在大约 50 693 个 μs 中得出预测结果。文献[15]提出了一种名为 NASH-CNN1 的 CNN 硬件架构,该架构在延迟、精度和硬件复杂性之间实现很好的平衡,达到一次手写数字识别仅需 7.58 μs。

表 4　工作频率、位宽、功耗比较

Tab. 4　Comparison of frequency, precision and power

	Ref.[11]	Ref.[12]	Ref.[13]	This paper
Frequency MHz	110	100	150	110
Precision/bit	16	8	8	32
Power	—	107 mW	26 W	6.444 3 W

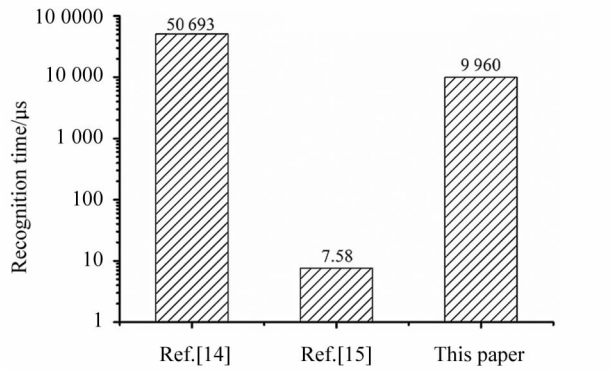

图 8　手写数字识别时间对比

Fig. 8　Comparison of handwritten digit recognition time

文献[16]与本文对比,识别时间大幅缩短,但同时资源消耗也更大,见表 5。对比文献[16],本文采用较少的硬件资源实现了并行的卷积计算,且可以在 9 960 μs 内得出识别结果,充分实现手写数字的实时检测。

表 5　资源占用对比

Tab. 5　Comparison of resource usage

Logic device	Ref.[16]	This paper
Slice LUTs	265 460	120 317
Slice registers	358 848	39 290
LUT-FF pairs	—	22 992
DSP	3 599(48E1)	48(48E1)

6　结　论

本文基于可重构阵列处理器的 NMC 结构,设计了 CNN 并行计算的方法,实现了手写数字识别。实验结果表明,在提高了卷积运算速度的同时降低了访存延迟。最高工作时钟频率可达 110 MHz,与以往的研究相比,执行单个卷积运算的计算速度平均提升了 75.00%,从图像输入到产生识别结果共用时 9 960 μs,满足实时检测的要求。

参考文献:

[1] BARI M, AMBAW A, DOROSLOVACKI M. Comparison of machine learning algorithms for raw handwritten digits recognition[C]//2018 52nd Asilomar Conference on Signals, Systems, and Computers, October 28-31, 2018, Pacific Grove, CA, USA. New York: IEEE, 2018: 1512-1516.

[2] LI Z W, LIU F, YANG W J, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J/OL]. IEEE Transactions on Neural Networks and Learning Systems (Early Access). (2021-06-10)[2022-02-17]. https://ieeexplore.ieee.org/document/9451544.

[3] WANG H L, CHNEG J F. Design of CNN accelerator based on embedded device application[J]. Chinese Journal of Electron Devices, 2021, 44(4): 5.
王红亮,程佳风.基于嵌入式设备应用的 CNN 加速器的设计研究[J].电子器件,2021,44(4):5.

[4] DATTA D, MITTAL D, MATHEW N P, et al. Comparison of performance of parallel computation of CPU cores on CNN model[C]//2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), February 24-25, 2020, Vellore, India. New York: IEEE, 2020: 1-8.

[5] KHAN M, HUTTUNEN H, BOUTELLIER J. Binarized convolutional neural networks for efficient inference on GPUs [C]//2018 26th European Signal Processing Conference (EUSIPCO), September 3-7, 2018, Rome, Italy. New York: IEEE, 2018: 682-686.

[6] SHAN W, YANG M, WANG T, et al. A 510-nW wake-up keyword-spotting chip using serial-FFT-based MFCC and binarized depthwise separable CNN in 28-nm CMOS[J]. IEEE Journal of Solid-State Circuits, 2020, 56(1): 151-164.

[7] CHEN Y H, FAN C P, CHANG C H. Prototype of low complexity CNN hardware accelerator with FPGA based PYNQ platform for dual-mode biometrics recognition [C]//2020 International SoC Design Conference (ISOCC), October 21-24, 2020, Yeosu, Korea (South). New York: IEEE, 2020: 189-190.

[8] WU C B, WANG C S, HSIAO Y K. Reconfigurable hardware architecture design and implementation for AI deep learning accelerator[C]//2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), October 13-16, 2020, Kobe, Japan. New York: IEEE, 2020: 154-155.

[9] MAO H N, SHU J W, LI F, et al. Development of process-

ing-in-memory[J], Science in China (Information Sciences), 2021, 51: 173-206.
毛海宇, 舒继武, 李飞, 等. 内存计算研究进展[J]. 中国科学: 信息科学, 2021, 51: 173-206.

[10] JIANG L, WANG X J, LIU Z T, et al. Design and implementation of convolutional neural network based on FPGA [J], Microelectronics and Computer, 2018, 35(8): 138-142.
蒋林, 王喜娟, 刘镇弢, 等. 基于FPGA的卷积神经网络设计与实现[J]. 微电子学与计算机, 2018, 35(8): 138-142.

[11] SHEN Y, FERDMAN M, MILDER P. Escher: a CNN accelerator with flexible buffering to minimize off-chip transfer [C]//2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), April 30-May 2, 2017, Napa, CA, USA. New York: IEEE, 2017: 93-100.

[12] YUAN Z, LIU Y P, YUE J S, et al. CORAL: coarse-grained reconfigurable architecture for convolutional neural networks[C]//2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), July 24-26, 2017, Taipei, Taiwan, China. New York: IEEE, 2017: 1-6.

[13] ZHANG C, SUN G G., FANG Z M, et al. Caffeine: toward uniformed representation and acceleration for deep convolutional neural networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 38(11): 2072-2085.

[14] GHAFFARI S, SHARIFIAN S. FPGA-based convolutional neural network accelerator design using high level synthesize[C]//2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), December 14-15, 2016, Tehran, Iran. New York: IEEE, 2016: 1-6.

[15] VU T H, MURAKAMI R, OKUYAMA Y, et al. Efficient optimization and hardware acceleration of CNNs towards the design of a scalable neuro inspired architecture in hardware[C]//2018 IEEE International Conference on Big Data and Smart Computing (BigComp), January 15-17, 2018, Shanghai, China. New York: IEEE, 2018: 326-332.

[16] ZHOU Y, WANG W, HUANG X. FPGA design for PCANet deep learning network[C]//2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines, May 2-6, 2015, Vancouver, BC, Canada. New York: IEEE, 2015.

作者简介:

山　蕊 (1986—), 女, 博士, 副教授, 主要研究领域为集成电路设计.