

引用格式:王巍,周凯利,王伊昌,等. 卷积神经网络(CNN)算法的 FPGA 并行结构设计 [J]. 微电子学与计算机,2019,36(4): 57-62. [WANG W, ZHOU K L, WANG Y C, et al. FPGA parallel structure design of convolutional neural network (CNN) algorithm[J]. Microelectronics & Computer, 2019, 36(4): 57-62.]

# 卷积神经网络(CNN)算法的 FPGA 并行结构设计

王 巍,周凯利,王伊昌,王 广,杨正琳,袁 军  
(重庆邮电大学 光电工程学院/国际半导体学院,重庆 400065)

**摘 要:**本文进行了 CNN 算法的 FPGA 并行结构设计. 该设计首先利用 CNN 的并行计算特征以及循环变换方法, 实现了可高效进行并行流水线的卷积计算电路, 然后利用能够减少存储器访存时间的双缓存技术, 在输入输出部分实现了缓存阵列, 用于提高电路的计算性能(GOPS, 每秒十亿次运算数). 同时本文还对激活函数进行了优化设计, 利用查找表和多项式结合的分段拟合方法设计了激活函数(sigmoid)的硬件电路, 以保证近似的激活函数的硬件电路不会使精度下降. 实验结果表明: 输入时钟为 150 MHz 时, 整体电路在计算性能上由 15.87 GOPS 提高到了 20.62 GOPS, 并在 MNIST 数据集上的识别率达到了 98.81%.

**关键词:** 卷积神经网络; 现场可编程门阵列(FPGA); 并行结构; 流水线;

**中图分类号:** TP333.5      **文献标识码:** A      **文章编号:** 1000—7180(2019)04—0057—06

## FPGA Parallel Structure Design of Convolutional Neural Network (CNN) Algorithm

WANG Wei, ZHOU Kai-li, WANG Yi-chang, WANG Guang,  
YANG Zheng-lin, YUAN Jun

(College of Electronics Engineering / International Semiconductor College, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** In this paper, the FPGA parallel structure design of CNN algorithm is carried out. The design first uses the parallel computing features of CNN and the cyclic transformation method to realize a convolution calculation circuit that can efficiently perform parallel pipelines. Then, using the double-buffer technology that can reduce the memory access time, a cache array is implemented in the input and output sections to improve the computational performance of the circuit (GOPS, one billion operations per second). At the same time, the activation function is optimized. The hardware circuit of the activation function (sigmoid) is designed by using the segmentation fitting method of lookup table and polynomial to ensure that the hardware circuit of the approximate activation function will not reduce the accuracy. The experimental results show that when the input clock is 150MHz, the overall performance of the circuit is improved from 15.87 GOPS to 20.62 GOPS, and the recognition rate on the MNIST data set reaches 98.81%.

**Key words:** convolution neural network; field programmable gate array (FPGA); parallel structure; pipeline;

### 1 引言

近年来,神经网络领域依靠深度学习技术使其

算法性能方面的优异表现被广泛关注,其中卷积神经网络表现突出,它的一个典型应用场景是图像中手写数字的识别.但是卷积神经网络中存在大量独

立、重复的乘法和加法运算,并且这些计算过程涉及大量的数据访问以及中间数据的存储,这使得研究人员非常关注计算的性能,并行度和功耗.而现有的通用计算平台(CPU、GPU<sup>[1]</sup>)具有一些不足,所以研究人员在算法实现上提出了许多基于FPGA<sup>[2]</sup>,SOC<sup>[3]</sup>和ASIC<sup>[4]</sup>的硬件加速实现方案.

为了提高CNN算法的硬件电路性能,采用具有很多专用算术功能单元、大量通用逻辑资源、片上存储资源、外围I/O接口和高速网络接口的FPGA器件是一个非常好的选择<sup>[5]</sup>.目前,相关方面的研究也得到了越来越多的关注<sup>[6,7,8]</sup>.文献[6]提出了CNN卷积核的硬件电路,并利用了流水线和循环展开来优化不同的循环迭代中重叠循环体的执行,使得数据共享的使用程度大大影响吞吐量,但是在嵌套循环重构方面的设计并不合理,虽然节省了硬件资源,但是在计算性能(GOPS,每秒十亿次运算数)方面没有太大提升.文献[8]提出增加足够的计算资源来并行执行卷积计算从而产生输出特征图的一行值,并且在不影响识别精度的情况下探究了合理的定点表示<sup>[9]</sup>,但是这种计算模式会造成计算单元的利用率不高,虽然提高了计算性能(GOPS),但是消耗了太多的硬件资源.尽管这些FPGA的实现报告良好的计算性能,但是还存在改进的空间.

本文介绍一种卷积神经网络(CNN)算法的FPGA并行架构设计,其首先利用CNN的并行计算特征以及循环变换方法,实现了可高效进行并行流水线的卷积计算电路,用于提高电路的整体计算性能(GOPS),并减少硬件资源的使用;然后在输入输出部分还加入了双缓冲用于减少存储器的访存时间.同时还对激活函数进行了优化设计,以保证近似的激活函数的硬件电路不会使精度下降.

2 卷积神经网络算法分析

2.1 卷积神经网络拓扑结构

图1所示为本文采用的应用于手写数字识别的卷积神经网络模型结构拓扑图.卷积神经网络利用了图像的空间关系,使得在网络结构上有三个特性:局部连接,权重共享以及空间或时间上的次采样.这使得它能够对大量有标签的数据进行自动学习并从中提取复杂特征.CNN模型中包括一个输入层(input)、两个卷积层(C1和C2)、两个池化层(S1和S2)和一个全连接层(FC).其中每层由多个二维平面组成,二维平面里包含多个独立的神经元.

2.2 卷积神经网络前向传播模型

本文在LeNet卷积神经网络的基础上优化了激活函数和池化计算方式,使得算法能够更有利于硬件设计.同时在训练阶段,优化了学习率,以便算法能够更好地寻找最优值,从而提高手写数字识别的识别率.在训练开始后,随着循环迭代次数的增加,学习率的设置不再采用LeNet卷积神经网络中固定值的方式而是以变化的方式逐渐减小.MATLAB仿真该优化模型的实验结果显示对于手写数字MNIST测试集的识别率达到了98.81%.在该模型中涉及的权重系数(weights)以及偏置系数(biases)个数如表1.

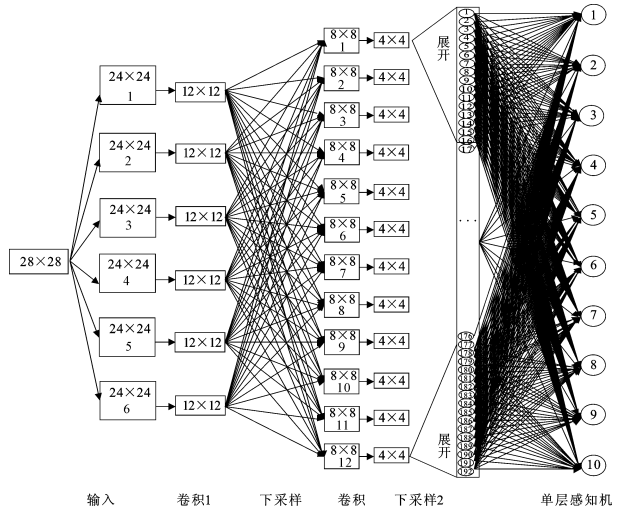


图1 卷积神经网络结构拓扑图

表1 sigmoid函数正区间内分段逼近函数表达式

| 层类型  | 特征图尺寸   | 参数个数(weights+biases) |
|------|---------|----------------------|
| 卷积层1 | 28×28×6 | 5×5×6+6=156          |
| 池化层1 | 14×14×6 | —                    |
| 卷积层2 | 8×8×12  | 5×5×12+12=312        |
| 池化层2 | 4×4×12  | —                    |
| 全连接层 | 10      | 4×4×12+10=202        |

卷积层用来将卷积核与前一层输出特征图的局部感受野相连,从而提取该局部特征.卷积运算如公式(1)表示.同时优化后的激活函数,选用了更有利于硬件实现的修正线性单元(relu)函数,如公式(2).

$$h_j^l = f(\sum_{i=M_l} h_i^{l-1} * k_{ij}^l + b_j^l) \tag{1}$$

$$f(x) = relu(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{2}$$

式中, $h_j^l$ 为第l隐含层的第j个输出特征图; $k_{ij}^l$ 为第

$l$ 层的第 $j$ 个卷积核的第 $i$ 个数; $b_j^l$ 为第 $l$ 卷积层的第 $j$ 个输出特征对应的偏置系数; $M_l$ 为第 $l$ 层卷积运算对输入特征图的选择; $f(x)$ 为激活函数。

池化层用来控制过拟合,通过对特征进行二次提取,使模型有较高的畸变容错能力,但不改变特征图的数目。池化操作把输入特征图分割为不重叠的 $2 \times 2$ 像素小块,每块对应的输出结果为这四个像素的均值或最大值,因此也被称为“平均池化”或“最大池化”。其中,最大池化操作不需要算术计算单元,有利于FPGA实现,所以本层被设置为最大池化而不再是平均池化,公式如(3)。

$$\text{pool}_{\max}(G_{\lambda, \tau}^A(i, j)) = \max_{\lambda, \tau}((a_s)_{\lambda \times \tau}) \quad (3)$$

式中,特征图 $A$ 的不重叠块大小为 $\lambda \times \tau$ ;第 $ij$ 个块为 $G_{\lambda, \tau}^A(i, j)$ ;  $s$ 和 $t$ 为特征图 $A$ 中像素 $a$ 的位置。

全连接层主要用来分类,它可以把所有局部特征结合起来抽象成全局特征。该层的激活函数采用sigmoid函数,其输出公式如式(4),其中的激活函数如公式(5)。

$$o_j^l = f(\sum w^l \cdot h^{(l-1)} + b_j^l) \quad (4)$$

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

式中, $o_j^l$ 表示第 $l$ 输出层的第 $j$ 个输出神经元; $w^l$ 表示第 $l$ 输出层的连接权值; $h^{(l-1)}$ 表示第 $l-1$ 层的第 $i$ 个输入特征图; $f(x)$ 为激活函数; $\sigma(x)$ 为sigmoid函数的表示形式。

### 3 FPGA的设计实现

#### 3.1 CNN的并行结构分析

论文[3]指出在卷积神经网络的计算中,卷积层的输出是由多个嵌套循环卷积计算组成,这使得卷积层的计算消耗大量的乘加运算单元并且耗时最多。而这种情况很适合将算法映射到FPGA的体系结构中实现硬件加速,从而在最大程度上挖掘网络模型的并行性特征。其中的并行特征可以概括4种:层间并行性、输出间并行性、卷积核间并行性和卷积核内并行性。由于FPGA片上资源是有限的,无法设计完全并行的计算结构,因此只能对部分卷积核进行并行结构计算。首先,我们需要利用输出间并行性对多个循环嵌套卷积计算进行计算划分,其中使用了循环变换方法,主要包括采用循环分块优化全局多个循环迭代的分块划分,以及采用循环展开实现分块循环结构到局部并行结构的设计,来构建直接用于卷积神经网络卷积层的计算电路。其中,我们只对输入特征图数量 $N$ 和输出特征图数量 $M$ 这两

个维度进行循环分块,循环分块尺寸用 $\langle \text{输出特征图 } T_m, \text{输入特征图 } T_n \rangle$ 表示。我们采用均匀分块对多个卷积运算循环进行处理,第一个卷积层的循环分块尺寸为 $\langle T_m, T_n \rangle = \langle 3, 1 \rangle$ ,第二个卷积层的循环分块尺寸为 $\langle T_m, T_n \rangle = \langle 6, 3 \rangle$ 。这种循环分块方法能够很自然的组成多输入多输出内部并行结构,从而达到充分开发网络模型的并行性特征。

而循环展开则利用了卷积核间的并行性,对块内循环的多输入单输出的特征提取过程,进行多个卷积运算的并行处理。这种多输入单输出并行更有利于减少提取一张输出特征图的执行时间。对于层间并行性,主要体现在系统初始化时参数的装载阶段,即网络中各层的权值参数可以同时并行加载。这是因为卷积神经网络是一种多层的网络结构,其中整个网络的数据是层层递进传播和处理的,前后层之间的运算有很强的相关性,使得层间计算的并行性很低。对于卷积核内并行性,主要是利用输入数据之间不存在数据相关的特性,把大小为 $K \times K$ 的卷积核中全部的 $K^2$ 次乘累加运算都并行实现。

#### 3.2 FPGA整体架构设计

本文在只考虑前向传播的情况下,利用卷积神经网络的并行计算特征及循环变换方法来确定整体架构中的设计划分,并在FPGA平台上实现了由若干并行或串行子系统构成的层级结构。图2所示为CNN卷积神经网络硬件架构图,其中,各子系统层对应了卷积神经网络中的特征映射图层。整体架构的子系统主要包括输入数据缓存模块、权值数据的存储模块、卷积计算模块、池化计算模块、全连接层内积运算模块、激活函数(sigmoid)模块以及控制模块。其中控制模块控制数据流在架构内各模块之间传递的先后关系以及模块之间互联通信的控制、握手信号。首先,控制逻辑产生数据请求信号,在该信号的控制下输入的像素数据流与系统时钟同步,并在每个周期后更新数据。然后输入的像素数据被送入到系统中各个并行或串行子系统的功能模块,完成若干个特征映射图的运算。最后,在经过逐层特征提取后,需要进行特征分类,它把多路局部特征数据流组合成全局特征送入到全连接层内积运算模块进行分类。

#### 3.3 卷积运算模块设计

卷积运算在数学形式上可以看作是由若干个乘累加运算部件组成,但在硬件设计上还需要考虑不同的数据缓存结构对卷积计算模块的性能影响,以

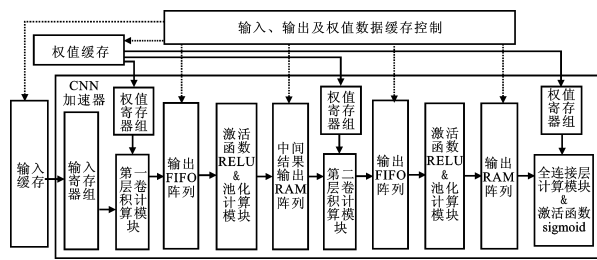


图2 CNN卷积神经网络硬件架构图

及不同的硬件结构需要合适的存储方式来配合计算过程中的流水线设计. 为了满足性能要求, 我们不但要考虑卷积层中通用卷积运算的设计, 还要对循环分块后每个卷积层中多输入多输出并行结构进行合理的设计. 在卷积核内及层间可以采用流水线式数据通路来提高数据运算的计算性能(GOPS).

首先, 本文的卷积计算模块是关于  $5 \times 5$  卷积核的卷积运算. 这里为了更好的显示电路结构, 我们简化地给出  $3 \times 3$  卷积运算电路, 如图3所示. 图3的设计通过行缓存结构搭配转置后的流水线卷积结构组成了一种新的能高效进行并行流水线操作的卷积计算电路. 其中行缓存结构实现图像的行列对齐, 并保证流水线的通畅, 提高数据运算的计算性能. 而转置后的流水线结构使得我们不用再给电路的输入数据提供额外的移位寄存器, 也不用为达了获得高通量给乘积的加法器添加额外的流水线级. 同时该电路的存储结构能够充分开发卷积窗口内的数据重用.

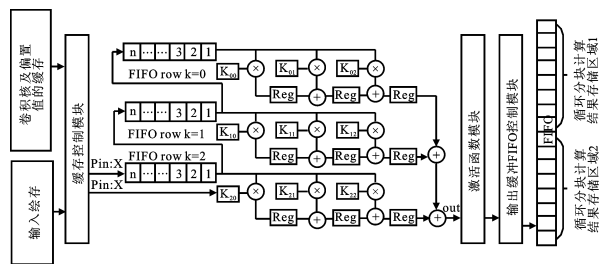


图3 带行缓存的并行流水线卷积计算模块

在图3的电路中, 为了不破坏流水线的流通性, 电路单元需要一定的时间将数据装满整个行缓存结构, 然后转置电路结构的流水线开始流通. 具体的电路组成主要包括  $K$  (卷积核大小) 个长度为  $N$  (输入图像的宽度) 的 FIFO 缓存阵列作为输入的行缓存; 以及  $K^2$  个乘法器和  $K^2$  个加法器用作卷积计算. 其中  $K^2 - 1$  个加法器用作二维卷积基本计算单元自身的卷积运算, 另外一个加法器作为多个二维卷积计算单元的累加单元. 在开始的时候, 输入图像的数

据首先按照逐行扫描的顺序进入 FIFO 阵列进行缓存, 在保证数据装满整个行缓存后, 每行电路开始同时读取 FIFO 阵列上的输入数据, 并将 FIFO 阵列上的每行输入数据分别广播到对应的  $K$  个乘法器上, 用于与存储器中的权值相乘. 接着将得到的乘积结果通过加法器链逐级累加, 其中两个临近加法器之间的寄存器用于流水线的时钟同步, 并且在计算下一个输入像素时, 为后面的加法器提供前一个时钟周期的部分累加结果. 直到所有的卷积窗口内的像素数据移动完毕.

然后, 对于循环分块后卷积层的多输入多输出并行结构, 我们也采用了流水线式数据通路和输入输出缓存结构来提高数据运算的计算性能(GOPS). 图4所示为第二个卷积层的部分并行电路设计及缓存设计. 其中, 设计为了使存储器的访问不影响计算性能, 保证计算单元持续工作, 在每层循环分块后的输入和输出部分都进行了乒乓缓存设计用于实现数据存储, 即当前的缓存在工作完成时, 另一块缓存已经存储好下一次需要的数据.

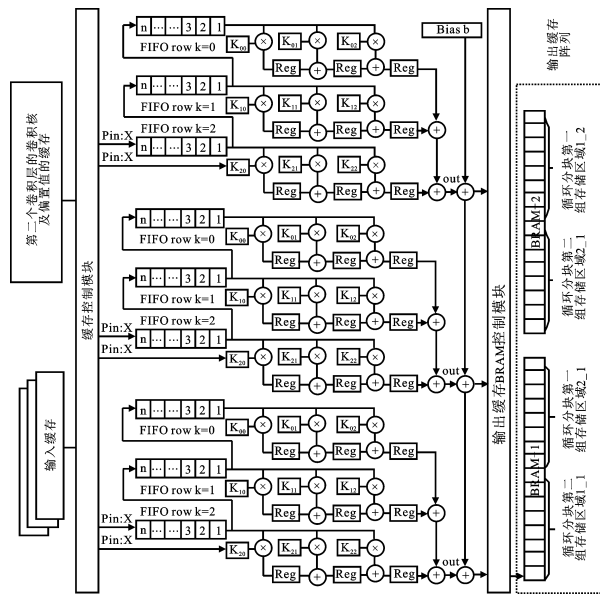


图4 第二个卷积层的部分并行电路设计及缓存设计

在图4的电路中, 我们使用图3中的卷积计算模块组成了图4中对应的3个并行卷积电路, 也就是同时对3张输入特征图进行卷积计算. 这种多输入单输出结构是因为, 对于第二个卷积层, 实际的输入特征图的数量有6个, 输出特征图的数量有12个, 但是在利用了循环分块进行硬件设计后, 输入特征图的数量变成3个. 所以图4是结合卷积间并行特征, 利用卷积计算模块中的多个二维卷积累加单元, 累加输出1张输出特征图的中间结果. 其中输出

的乒乓缓存 BRAM-1 用于缓存循环分块中的第一组循环结果,BRAM-2 用于缓存循环分块中的第二组循环结果. 又因为输出特征图的数量变成了 6 个,而图 4 只能用于生成 1 张特征图,所以本文使用 6 个图 4 的电路结构组成了第二个卷积层的多输入多输出并行运算电路,即需要 18 个卷积计算模块. 而对于第一个卷积层的并行设计,其设计思路与第二层类似,只是输入特征图的数量为 1 张,但有 3 个并行卷积电路,同时对同一张输入特征图做卷积运算,并且在卷积运算结束后直接把卷积结果存入上缓存中,而不需要累加多个二维卷积结果.

综上可知,带行缓存的并行流水线卷积电路,在流水线填充完毕后能够保证每个时钟周期内,可以输出相关特征图中某一个像素点上的值. 而每一层的多输入多输出并行电路,在块循环全部结束后,每一层可以输出相关特征图中全部的像素点上的值. 这有利于提高电路的整体计算性能(GOPS).

3.4 sigmoid 函数运算模块设计

激活函数是实现神经网络计算的关键环节,它主要用来加入非线性因素,通过线性变化解决线性不可分问题. 其中,relu 函数可以直接在 FPGA 上实现. 而 S 型函数 sigmoid,则不能在 PFGA 上直接实现,它需要使用一些函数逼近的方法来近似. 常用的逼近方法有:查表法、泰勒级数展开法、分段线性函数逼近法、多项式拟合法和 CORDIC 算法. 其中,查表法属于最简单的方法,在精度要求高时,会消耗大量的存储资源. 泰勒级数展开法需要很多阶数,涉及大量乘法操作. 多项式拟合法也存在消耗大量乘法器的问题. CORDIC 算法的运算过程消耗的时间长,并且精度不高,但是占用的资源少.

为了更好的实现 sigmoid 函数,本文结合 sigmoid 函数关于点(0,0.5)对称的性质,以及函数曲线在 y 轴附近弧度明显,而在远离 y 轴时,曲线比较平坦的性质,采用了查表法和多项式拟合相结合的方法对 sigmoid 函数进行分段逼近,而且只计算[0,+∞)区间的函数值. 这样做是为了更好地平衡近似的激活函数的硬件电路在计算精度与资源消耗之间的矛盾. 首先,借助 MATLAB 软件分段逼近 sigmoid 函数,确定分段区间、分段数目及分段区间内的拟合多项式和近似常数值,并保证误差的数量级在 0.001 以下. 在 MATLAB 中,使用函数 polyfit 来求拟合多项式中每一阶的系数. 然后,对分段拟合后的 sigmoid 函数的函数值与理论值之间的误差绝

对值进行分析,其中最大的误差绝对值为  $8.62 \times 10^{-4}$ ,误差的数量级满足 0.001 以下. 表 2 显示了 sigmoid 函数正区间内分段逼近函数的表达式. 正区间被分为了 12 个子区间.

表 2 sigmoid 函数正区间内分段逼近函数表达式

| 区间                       | 逼近函数表达式                                              |
|--------------------------|------------------------------------------------------|
| $x_{13} \in [0,1)$       | $y = -0.0279 * x_{13}.^2 + 0.2605 * x_{13} + 0.4992$ |
| $x_{14} \in [1,2)$       | $y = -0.0467 * x_{14}.^2 + 0.2896 * x_{14} + 0.4882$ |
| $x_{15} \in [2,3)$       | $y = -0.0298 * x_{15}.^2 + 0.2202 * x_{15} + 0.5600$ |
| $x_{16} \in [3,4)$       | $y = -0.0135 * x_{16}.^2 + 0.1239 * x_{16} + 0.7031$ |
| $x_{17} \in [4,5)$       | $y = -0.0054 * x_{17}.^2 + 0.0597 * x_{17} + 0.8297$ |
| $x_{18} \in [5,5.02)$    | $y = 0.9930;$                                        |
| $x_{19} \in [5.02,5.19)$ | $y = 0.9940;$                                        |
| $x_{20} \in [5.19,5.39)$ | $y = 0.9950;$                                        |
| $x_{21} \in [5.39,5.64)$ | $y = 0.9960;$                                        |
| $x_{22} \in [5.64,5.97)$ | $y = 0.9970;$                                        |
| $x_{23} \in [5.97,6.47)$ | $y = 0.9980;$                                        |
| $x_{24} \in [6.47,7.55)$ | $y = 0.9990;$                                        |

在 FPGA 中,卷积神经网络识别的准确性与数据精度高度相关. 为了不影响识别结果,本文把输入输出数据的定点位宽设置为 16 位,其中第一位为符号位<sup>[9]</sup>. 由表 2 可知,本文首先需要对二次函数  $y = a * x^2 + b * x + c$  进行硬件设计. 为了优化硬件设计,对二次函数的运算表达式进行重新组合,即  $y = (a * x + b) * x + c$ ,其中多项式的系数  $a, b, c$  经过量化存入寄存器. 然后,对用于拟合曲线的常数经过量化后放入查找表,硬件电路如图 5 所示. 图 5 中的多路器(MUX)用于判定量化后的输入数据所在的区间.

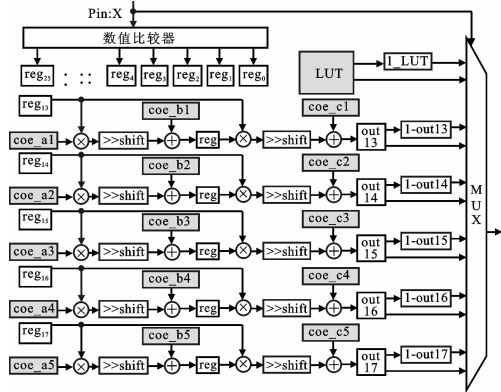


图 5 sigmoid 函数的硬件电路模块

4 结果分析与讨论

在实验中,首先将上节得到的卷积神经网络算法的 FPGA 设计进行硬件仿真分析,然后使用 Xilinx Virtex-7 平台进行硬件实现. 实验中将 MNIST 测试集(10000 张图片)中的图像像素作为卷积神经网络算法的输入特征图,将最后每一类的得分作为输出. 如图 6 所示的仿真图,它显示了处理一张图片后会输出每一类的得分.

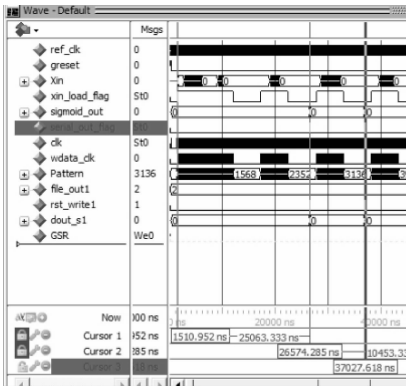


图 6 卷积神经网络算法的仿真结果

在结果分析中,本文主要以卷积神经网络消耗的 MAC 算术运算单元的数量来衡量 FPGA 的硬件开销. 同时,我们通常用 GOPS(每秒十亿次运算数)的单位来描述性能,它是一个非常好地用于衡量计算性能的指标. 其中 GOPS 以文献[3]给出的公式为参考:

计算性能 = 操作总数 / 执行周期数

操作总数=2×R×C×M×N×K×K;执行周期数为完成计算所需的时钟周期,一般取实际测量值. 其中 N 和 M 分别为输入和输出特征图的数量, R 和 C 分别为输出特征图的行列数, K 为卷积核的尺寸.

表 3 给出了卷积神经网络 FPGA 实现的性能比较,其中比较了在 MNIST 测试集(10000 张图片)中预测一张图片的时间,识别率,FPGA 资源占用情况和计算性能(GOPS). 由于本文利用了循环分块方法重新设计了循环操作以及为了较少数据存取延迟使用了双缓存技术,使得本文的实验性能较好. 通过表 3 的性能比较发现与文献[8]相比,虽然文献[8]由于采用片外存储器节省了 BRAM 的使用,并且 FF 的使用也比本文占用的资源少,但是本文由于使用了新的循环计算节省了主要的 DSP 资源 73 个,同时本文的计算性能 GOPS 与文献[8]相比也提

高了 4. 75GOPS, 因此有利于整体性能. 而与文献[2,6]相比,在计算性能上得到明显提升,其 GOPS 提升了 7. 5 倍,不过在主要的硬件资源消耗上相对较多.

表 3 卷积神经网络 FPGA 实现的性能比较

| 参数     | 文献[2]              | 文献[6]      | 文献[8]              | 本文                 |
|--------|--------------------|------------|--------------------|--------------------|
| FPGA   | Virtex-7 xc7vx485t | Zynq-ZC702 | Virtex-7 xc7vx485t | Virtex-7 xc7vx485t |
| 频率(M)  | 100                | 166        | 150                | 150                |
| 时间/ms  | 2. 636 8           | 0. 151     | 0. 025 4           | 0. 019 7           |
| BRAM   | 27                 | 96         | 0                  | 89                 |
| DSP    | 20                 | 95         | 638                | 565                |
| FF     | 54 075             | 27 664     | 66 346             | 66 747             |
| LUT    | 14 832             | 38 836     | 51 125             | 35 295             |
| 识别率(%) | 98. 62             | 99. 01     | 96. 8              | 98. 81             |
| GOPS   | 1. 58              | 2. 70      | 15. 87             | 20. 62             |

5 结束语

本文在对卷积神经网络(CNN)算法深入研究的基础上进行了 CNN 算法的 FPGA 并行结构设计. 该设计实现了卷积计算电路和激活函数(sigmoid)的硬件电路. 并行结构的设计实现使电路的整体计算性能得到提高,并减少了硬件资源的使用. 实现结果在 MNIST 数据集上获得了 98. 81% 的识别率以及在输入时钟为 150 MHz 时,整体电路在计算性能(GOPS)上由 15. 87 GOPS 提高到了 20. 62 GOPS.

参考文献:

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[2] GHAFARI S, SHARIFIAN S. FPGA-based convolutional neural network accelerator design using high level synthesize[C]// International Conference of Signal Processing and Intelligent Systems. IEEE, 2017: 1-6.

[3] GOKHALE V, JIN J, DUNDAR A, et al. A 240 Gops/s Mobile Coprocessor for Deep Neural Networks [C]// Computer Vision and Pattern Recognition Workshops. IEEE, 2014:696-701.

## 参考文献:

- [1] ZHOU G, XU J, WANG J, et al. Comparison study on digital peak current, digital peak voltage, and digital peak voltage/peak current Controlled Buck Converter [C]// Industrial Electronics and Applications, 2009. Iciea 2009. IEEE Conference on. IEEE, 2009; 799-804.
  - [2] CHEN J, PRODIC A, ERICKSON R W, et al. Predictive digital current programmed control[J]. Trans on IEEE Power Electronics, 2003, 18(1):411-419.
  - [3] 王志勇,柯曾喜,王云. 电流型控制开关变换器的分析与设计[J]. 微电子学与计算机, 2007, 24(2):96-98.
  - [4] LAI Y S, YEH C A. Predictive Digital-Controlled Converter With Peak Current-Mode Control and Leading-Edge Modulation[J]. IEEE Transactions on Industrial Electronics, 2009, 56(6):1854-1863.
  - [5] DAVOUDI A, JATSKEVICH J. Parasitics Realization in State-Space Average-Value Modeling of PWM DC-DC Converters Using an Equal Area Method[J]. IEEE Transactions on Circuits & Systems I Regular Papers, 2007, 54(9):1960-1967.
  - [6] MATTAVELLI P. Digital control of DC-DC boost converters with inductor current estimation[C]// Applied Power Electronics Conference and Exposition, 2004. Apec '04. Nineteenth IEEE. IEEE, 2004:74-80 Vol. 1.
  - [7] 童乔凌,陶志威,谢文,等. 基于 EKF 的 DC/DC 转换器电感电流观测器[J]. 华中科技大学学报(自然科学版), 2013, 41(5):48-51.
  - [8] 陈晨,童乔凌,谢文,等. 基于改进 EKF 的 DC/DC 转换器电感电流观测器[J]. 微电子学与计算机, 2014, 31(6):125-128.
- 作者简介:**
- 王力志** 男, (1994-), 硕士研究生. 研究方向为集成电路工程设计. E-mail: 758385315@qq. com.
- 薛海斌** 男, (1983-), 博士, 副教授, 硕士生导师. 研究方向为纳米器件、单分子器件的电流噪声特性.
- 吕玉祥** 男, (1964-), 硕士, 教授. 研究方向为测试计量.
- 童乔凌** 男, (1981-), 博士, 副教授, 博士生导师. 研究方向为超大规模集成电路设计.
- 
- (上接第 62 页)
- [4] CHEN Y H, KRISHNA T, EMER J, et al. 14.5 Eyerriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks[C]// IEEE International Solid-State Circuits Conference. IEEE, 2016:262-263.
  - [5] FARABET C, POULET C, HAN J Y, et al. CNP: An FPGA-based processor for Convolutional Networks [C]// International Conference on Field Programmable Logic and Applications. IEEE, 2009:32-37.
  - [6] FENG G, HU Z, CHEN S, et al. Energy-efficient and high-throughput FPGA-based accelerator for Convolutional Neural Networks[C]// IEEE International Conference on Solid-State and Integrated Circuit Technology. IEEE, 2017:624-626.
  - [7] CHEN ZHANG, PENG LI, GUANGYU SUN, et al. Optimizing fpga-based accelerator design for deep convolutional neural networks[C]// in Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2015:161-170.
  - [8] ZHOU Y, JIANG J. An FPGA-based accelerator implementation for deep convolutional neural networks [C]// International Conference on Computer Science and Network Technology. IEEE, 2016:829-832.
  - [9] JIANG J, HU R, LUJAN M. A Flexible Memory Controller Supporting Deep Belief Networks with Fixed-Point Arithmetic[C]// Parallel and Distributed Processing Symposium Workshops & Phd Forum. IEEE, 2013:144-152.
- 作者简介:**
- 王巍** 男, (1967-), 博士后, 教授. 研究方向为集成电路设计.
- 周凯利(通讯作者)** 女, (1991-), 硕士研究生. 研究方向为数字集成电路设计. E-mail: 2508005354@qq. com.
- 王伊昌** 男, (1996-), 硕士研究生. 研究方向为模拟集成电路设计.
- 王广** 男, (1994-), 硕士研究生. 研究方向为半导体光电器件设计.
- 杨正琳** 男, (1980-), 博士, 副教授. 研究方向为模拟集成电路设计.
- 袁军** 男, (1984-), 博士, 副教授. 研究方向为数模混合集成电路设计.