文章编号:1671-4598(2023)05-0221-07 DOI:10.16526/j.cnki.11—4762/tp.2023.05.033 中图分类号:TP391 文献标识码:A

基于 FPGA 加速的低功耗的 MobileNetV2 网络识别系统

孙小坚，林瑞全，方子卿，马 驰
（福州大学 电气工程与自动化学院，福州 350000）

摘要：近年来，卷积神经网络由于其出色的性能被广泛应用在各个领域，如图像识别、语音识别与翻译和自动驾驶等；但是传统卷积神经网络（CNN，convolutional neural network）存在参数多，计算量大，部署在 CPU 与 GPU 上推理速度慢、功耗大的问题；针对上述问题，采用量化感知训练（QAT，quantization aware training）的方式在保证图像分类准确率的前提下，将网络参数总量压缩为原网络的 1/4；将网络权重全部部署在 FPGA 的片内资源上，克服了片外存储带宽的限制，减少了访问片外存储资源带来的功耗；在 MobileNetV2 网络的层内以及相邻的点卷积层之间提出一种协同配合的流水线结构，极大地提高了网络的实时性；提出一种存储器与数据读取的优化策略，根据并行度调整数据的存储排列方式及读取顺序，进一步节约了片内 BRAM 资源。最终在 Xilinx 的 Virtex-7 VC707 开发板上实现了一套性能优、功耗小的轻量级卷积神经网络 MobileNetV2 识别系统，200 MHz时钟下达到了 170.06 GOP/s 的吞吐量，功耗仅为 6.13 W，能耗比达到了 27.74 GOP/s/W，是 CPU 的 92 倍，GPU 的 25 倍，性能较其他实现有明显的优势。

关键词：硬件加速；量化感知训练；MobileNet；并行计算；流水线结构

Low-power MobileNetV2 Network Identification System Based on FPGA Acceleration

SUN Xiaojian, LIN Ruiquan, FANG Ziqing, MA Chi
(School of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350000, China)

Abstract: In recent years, convolutional neural networks have been widely used in various fields, such as image recognition, speech recognition and translation, and autonomous driving; However, traditional convolutional neural network (CNN) has the problems of many parameters, large computation, slow inference speed and high power consumption on CPU and GPU. To address above problem, quantization aware training (QAT) is used to compress the total number of network parameters to 1/4 of the original network while ensuring the accuracy of image classification. All the network weights are deployed on the on-chip resource of FPGA, which overcomes the limitation of off-chip storage bandwidth and reduces the power consumption caused by accessing off-chip storage resources. A cooperative pipeline structure is proposed within the layers of the MobileNetV2 network and between the adjacent point convolutional layers, which greatly improves the real-time performance of the network. An optimization strategy for memory and data reading is proposed to adjust the data storage arrangement and reading order by the parallelism degree, further saving on-chip BRAM resources. Finally, a lightweight convolutional neural network MobileNetV2 recognition system with excellent performance and low power consumption was implemented on the Xilinx's Virtex-7 VC707 development board. The throughput of 170.06 GOP/s was realized under the clock of 200 MHz, with the power consumption of only 6.13 W, the energy consumption ratio of 27.74 GOP/s/W, 92 times that of CPU and 25 times that of GPU. The performance has obvious advantages over other implementation modes.

Keywords: hardware acceleration; quantization aware training; MobileNetV2; parallel computing; pipeline structure

0 引言

卷积神经网络善于从海量数据中提取目标的特征，由于其出色的分类识别能力，已被广泛的应用于图像分类、语音识别、医疗诊断、国防安全等领域。庞大的参数量使得卷积神经网络部署在资源有限的嵌入式设备上存在困难。轻量级网络的诞生，很大程度上降低了网络的参数量，适合部署在小容量的嵌入式设备上，应用于目标分类的场景[1-3]，轻量级网络如 MobileNetV2 [4]，采用深度可分离的卷积结构，在减少了网络参数量的同时又能保证识别精度。

目前将 CNN 部署在嵌入式设备上主要有几种方式，如

收稿日期:2022-10-13； **修回日期**:2022-11-04。

作者简介:孙小坚(1992-)，男，福建宁德人，硕士研究生，工程师，主要从事图像处理、FPGA 方向的研究。
通讯作者:林瑞全(1971-)，男，福建莆田人，博士研究生，教授，主要从事 Delta 算子理论，有源电力滤波方向的研究。

引用格式:孙小坚，林瑞全，方子卿，等.基于 FPGA 加速的低功耗的 MobileNetV2 网络识别系统[J].计算机测量与控制，2023，31(5)：221-227.

处理器 CPU、GPU、专用集成电路（ASIC，application specific integrated circuit）和现场可编程门阵列（FPGA，field programmable gate array）等。将 CNN 部署于传统的处理器 CPU 上，会带来高延迟。若是使用并行计算 GPU 会存在较大的功耗。使用专用的集成电路 ASIC 虽然性能好，但是成本较高。FPGA 具有实时性、低功耗、可重构的优点，特别适用于作为 CNN 的硬件加速器。将 AI 领域主流的 CNN 网络部署在 FPGA 上的工作仍然面临着内存带宽不足和计算并行度低的限制。由于 FPGA 片内存储资源有限文献［5］将卷积计算中间缓存放在了片外 DDR 上，文献［6］则把 DDR 当做权重数据的存储区，这样虽能缓解片内存储的压力，但是片外的 DDR 存在带宽的限制，同时也增加了额外的功耗。而文献［7－8］将参数均存在片内存储资源上，无需访问片外 DDR，虽然降低了系统功耗，但是需要足够大的片内存储资源。在探索提升计算并行度的方案上，文献［9－11］采用单一的计算结构，不同的卷积层复用该计算结构，可以将大部分资源分配给该计算结构，最大程度提升该计算阵列的并行度，但是难以适应不同卷积的计算方式，导致效率低下。文献［12－13］以全流水线的形式将 CNN 网络展开，层与层之间流水计算，每层分别占用不同的资源，资源消耗较大，适用于浅层次的网络，无法部署深层次的网络。文献［14］综合了单一计算结构与全流水线结构的优点，针对 MobileNetV2 网络提出了一种半流式结构，该结构只针对专用的 CNN 设计，结构固定，通用性不强。

针对上述问题，本文采用量化感知训练的方式在量化的过程中对模型进行二次训练将模型精度由静态量化的 73.33%提升到了 93.89%，压缩模型尺寸为原来的 1/4，用 8 位定点整数表示权重与缓存。将压缩后的权重参数全部存放在片内，克服了片外存储带宽限制的同时，降低了功耗，硬件加速部分功耗仅为 6.13 W；并且优化了存储器的存储结构与数据的读取方式，根据不同层的卷积计算方式与并行度设计权重和输入数据在 BRAM 中的排列方式，进一步节约了内存，卷积运算中对输入图片数据进行复用减少了对内存的访问次数；提出了一种层内、层间协同配合的流水线结构，在卷积层内，相邻 PW 层间流水线展开，采用十二级流水线的设计方法，极大的提升了网络的实时性，在短暂延时后，能够实现每 0.64 ms 推断一张图片。通过实验分析与对比，本文的方法在一定程度上解决了将 CNN 网络部署在 FPGA 上时内存带宽不足和计算并行度低所带来的限制，同时还降低了功耗。

1 轻量级网络 MobileNetV2 及其量化处理

1.1 轻量级网络 MobileNetV2

MobileNetV2 由 gogle 团队于 2018 年提出，采用深度可分离卷积，即将一个标准的卷积层拆分为逐层卷积（DW，depthwise convolution）与逐点卷积（PW，point-wise convolution）操作。使得网络的参数量与计算量大为降低，但是却能保持较高的准确率，因此被广泛的应用于小容量嵌入式应用场景。MobileNetV2 的网络结构如表 1 所示，其中 t 代表着缩放系数，c 代表输出通道数，n 代表重复次数，s 代表着步长。

表 1 MobileNetV2 网络结构

输入	操作	t	c	n	s
224×224×3	Conv2d 3×3	—	32	1	2
112×112×32	Bottlenect	1	16	1	1
112×112×16	Bottlenect	6	24	2	2
56×56×24	Bottlenect	6	32	3	2
28×28×32	Bottlenect	6	64	4	2
14×14×64	Bottlenect	6	96	3	1
14×14×96	Bottlenect	6	160	3	2
7×7×160	Bottlenect	6	320	1	1
7×7×320	Conv2d 1×1	—	1 280	1	1
7×7×1 280	Avgpool 7×7	—	—	1	—
1×1×1 280	Conv2d 1×1	—	k	—	

与 MobileNetV1［15］相比 MobileNetV2 参数更少，准确度却更高，改进的地方在于它采用了倒残差瓶颈模块（Inverted residual block）先用 1×1 conv 将 k 维网络进行升维到 tk 维，后通过 1×1 DW 层进行特征的提取，最后由 1×1 conv 操作重新降低为 k 维。倒残差瓶颈模块输出部分借鉴了 RseNet［16］的 short-cut 操作，当中间 DW 层进行下采样时（stride＝2）时，直接输出，当中间 DW 层不进行下采样（stride＝1）时，瓶颈模块输入与输出相加，具体操作如图 1 所示。

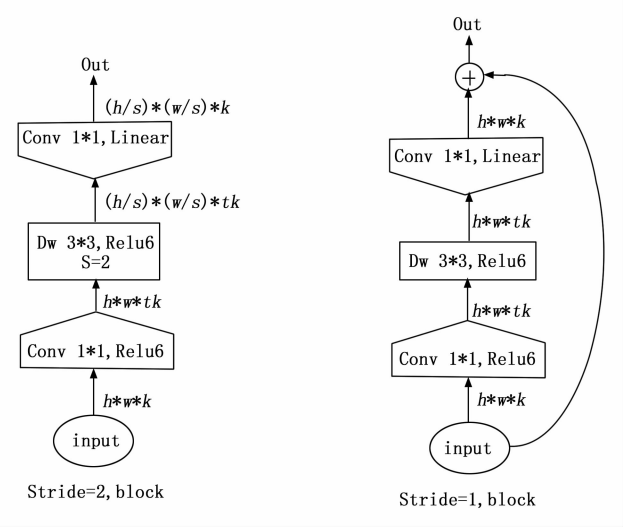

图 1 倒残差瓶颈模块具体操作

1.2 网络的量化处理

通常 PC 端保存训练好的卷积神经网络模型参数为 32 位的浮点数。MobileNetV2 网络的参数量为 3.4 M，FPGA 的片内 BRAM 资源十分的有限，如果网络的参数均采用 32 浮点数表示存入 FPGA 中，FPGA 内存将严重不足。这就

需要对网络模型进行量化处理，压缩模型参数，缓解 FPGA 内存不足的压力。量化方式有静态量化、动态量化、量化感知训练 3 种方式，前两种方法是针对训练好的模型进行量化，而量化感知训练则会对模型进行二次训练。小型的网络，如果采用前两种量化方式会带来较大的精度损失[17]，而量化感知训练不仅可以压缩模型参数，且模型的精度损失还较小，与此同时相较于浮点数运算，FPGA 更加擅长处理定点数运算。因此本文采用量化感知训练的方式，将 32 位浮点数参数用 INT8 型表示。

假设卷积的权重参数为 w，偏置为 b，输入为 x，输出激活值为 a，卷积运算可以表示为下式所示：

$$a = \sum_{i}^{N} w_i x_i + b \tag{1}$$

用 X_{float} 表示浮点实数，q 表示量化后的定点数，S 为量化的尺度因子，Z 则表示 0 经过量化后对应的数值。则浮点数与定点数的转化公式如公式（2）与公式（3）所示：

$$X_{float} = S(q - Z) \tag{2}$$

$$q = round(\frac{X_{float}}{S}) + Z \tag{3}$$

将公式（1）中的 w、b、x 分别用量化后的定点数表示，同时偏置的量化尺度因子取为 $S_b = S_w S_x$ 就得到了公式（4）。

$$q_a = \frac{S_w S_x}{S_a}\left(\sum_{i}^{N}(q_w - Z_w)(q_x - Z_x) + q_b\right) + Z_a \tag{4}$$

参数量化后，需要用新的定点数计算公式替代原来的浮点数卷积运算公式在硬件上实现。整个公式（4）只有 $\frac{S_w S_x}{S_a}$ 是浮点数，其中 S_w 为权重的量化尺度，S_x 为输入的量化尺度，S_a 为输出的量化尺度。假设 $M = \frac{S_w S_x}{S_a}$，只要找到合适的 n 与 M_0 使得 $M \approx 2^{-n} M_0$，公式（4）就能全部转化为定点数运算。

量化感知训练在对网络进行二次训练的过程中将式（2）加入到了它的前向传播的训练过程，卷积计算时将浮点数转化为定点数。计算完成后进行反量化操作，通过式（3）将定点数重新转化为浮点数。在网络训练的反向传播过程中则按浮点数的计算方式进行，以此来提高模型对量化效应的适应度，最终提高量化后模型的精度每一层的量化操作按照式（5）进行

$$clamp(r,a,b) \approx \min(\max(x,a),b)$$

$$s(a,b,n) \approx \frac{b-a}{n-1}$$

$$q(r,a,b,n) \approx$$

$$round\left(\frac{clamp(r,a,b) - a}{s(a,b,n)}\right)s(a,b,n) + a \tag{5}$$

其中：r 代表要量化的实数值，a，b 是该实数的量化范围，即该层 Tensor 里的最小值与最大值，n 则代表量化级数，若最终需量化为 8 位则 $n = 2^8 = 256$。

在 PyTorch 框架下对东北大学发布的热轧带钢表面缺陷数据集进行试验测试，输入图片尺寸为 96×96×3，运用静态量化将模型参数压缩至 INT8 型后缺陷分类正确率下降到了 73.33%，采用量化感知训练方式将模型参数压缩至 INT8 型的同时进行再次训练，在未采用按通道（per-channel）优化的情况下 8 轮 EPOCH 后，正确率可以达到 93.89%，如图 2 所示。

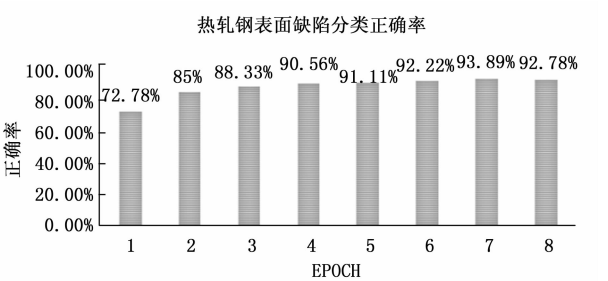

图 2　热轧带钢表面缺陷分类正确率

未量化前浮点数模型的精度为 96.10%，量化后精度损失为 2.21%，满足精度要求。

2　MobileNetV2 网络识别系统的硬件设计

2.1　系统整体结构

系统的整体架构如图 3 所示，由 Host 上位机、硬件加速模块（Accelerator）、显示模块（LCD）组成。其中网络的权重，经量化后根据 DW、PW 层的并行度以及数据读取规则进行有序排列，提前写入 On-chip Memory 中。上位机只负责通过 PCIE 总线传输输入图片数据到 Input-buffer 中。

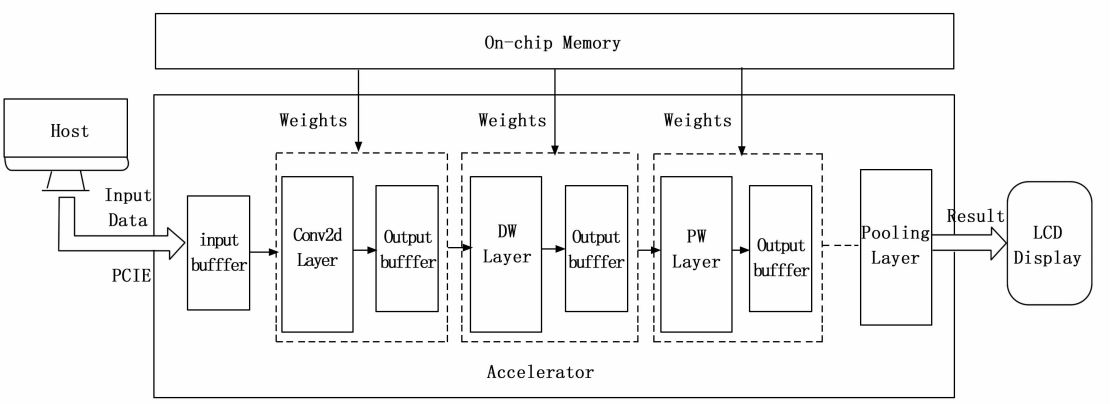

图 3　MobileNetV2 网络识别系统整体结构

硬件加速部分在监测到图片数据已经加载完成后，从 On-chip Memory 读入权重，即开始前向推理加速。硬件加速模块对应着 MobileNetV2 网络结构进行部署，主要由 Conv2d 层、DW 层、PW 层、池化层（Pooling Layer）组成。最终分类结果显示在 Virtex-7 FPGA 挂载的 LCD 上。

2.2 存储器与数据读取的优化策略

针对片内资源短缺的问题，为了进一步的节约片内 BRAM 的存储资源，本文提出了一种存储器与数据读取的优化策略，优化了权重与缓存的存储结构以及数据的读取方式。

将一个最基本的卷积运算展开，要做到多通道并行，最简单的方法就是将每一路通道上的输入图片与权重数据，分别用一个 BRAM 存储。这样做程序实现简单，但由于 MobileNetV2 网络输入通道数量最多可达 1 280，这样的实现方法会造成片内 BRAM 资源的严重浪费，本文根据 FPGA 片内 BRAM 资源的构成特点，采用一个地址拼接多个通道数据的方式将卷积计算合理的在输入、输出通道进行展开，根据卷积计算的特点，上一层卷积的输出通道并行度需与下一层输入通道的并行度保持一致，且该数值应为 MobileNetV2 网络每层通道数的公因数，18 K BRAM 空间按 512×36 进行配置则每层网络中权重消耗的 18 K BRAM 资源与输入通道、输出通道并行度的关系如式（6）所示：

$$Data_w = P_{in} * P_{out} * n_w$$

$$Depth = \frac{C_{in} * C_{out} * K * K}{P_{in} * P_{out}}$$

$$BRAM_{num} = \left\lceil \frac{Depth}{512} \right\rceil * \left\lceil \frac{Data_w}{36} \right\rceil \tag{6}$$

其中：$Data_w$ 表示权重数据在 BRAM 中每个地址的数据位宽，P_{in}、P_{out} 则表示卷积运算输入通道与输出通道设置的并行度，n_w 代表量化的位宽，由于采用的是 8 位量化所以这里的 $n_w = 8$。$Depth$ 表示 BRAM 的数据深度，C_{in}、C_{out} 为本层网络的输入与输出通道数，卷积核的尺寸大小为 $K * K$。最终可由 $Depth$ 与 $Data_w$ 的数值大小确定本层网络消耗的 18 K BRAM 资源数。结合 Virtex-7 FPGA 片内资源情况通过计算可知当每层的 P_{in}、P_{out} 都取 4 时，在兼顾网络推理速度时能够留有一定的存储裕度，为了进一步提升处理速度，在 MobileNetV2 网络计算密集的层增大了输入与输出通道的并行度。每层网络权重消耗的 BRAM 资源数值情况与并行度如表 2 所示。

以 PW 层为例，数据的存储结构与读取方式如图 4 所示。假设输入图片的尺寸为 Map（$k \times k \times n$），权重则为 Weight（$n \times m$）。在 PW 层设计的并行

表 2 MobileNetV2 每层网络存储资源消耗情况

层级	权重数	P_{in}	P_{out}	BRAM（18K）	层级	权重数	P_{in}	P_{out}	BARM（18K）
0	864	3	4	3	27	24 576	4	4	12
1	288	4	4	1	28	3 456	4	4	2
2	512	4	8	8	29	24 576	4	4	12
3	1 536	8	4	8	30	24 576	4	4	12
4	864	4	4	1	31	3 456	4	4	2
5	2 304	4	4	4	32	36 864	4	4	20
6	3 456	4	4	4	33	55 296	4	4	28
7	1 296	4	4	1	34	5 184	4	4	3
8	3 456	4	4	4	35	55 296	4	4	28
9	3 456	4	4	4	36	55 296	4	4	28
10	1 296	4	4	1	37	5 184	4	4	3
11	4 608	4	4	4	38	55 296	4	4	28
12	6 144	4	4	4	39	55 296	4	4	28
13	1 728	4	4	1	40	5 184	4	4	3
14	6 144	4	4	4	41	92 160	4	4	48
15	6 144	4	4	4	42	153 600	4	4	76
16	1 728	4	4	1	43	8 640	4	4	5
17	6 144	4	4	4	44	153 600	4	4	76
18	6 144	4	4	4	45	153 600	4	4	76
19	1 728	4	4	1	46	8 640	4	4	5
20	12 288	4	4	8	47	153 600	4	4	76
21	24 576	4	4	12	48	153 600	4	4	76
22	3 456	4	4	2	49	8 640	4	4	5
23	24 576	4	4	12	50	307 200	4	8	152
24	24 576	4	4	12	51	409 600	8	4	200
25	3 456	4	4	2	52	—	—	—	—
26	24 576	4	4	12	53	7 680	4	6	6

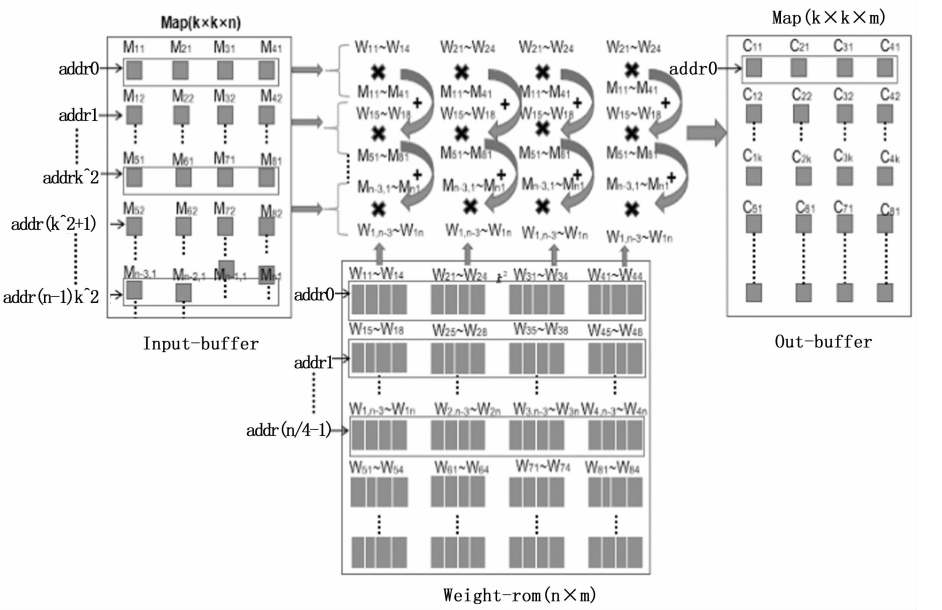

图 4 数据的存储结构与读取方式

度为 16，即在一个时钟周期完成 16 组的 1×1 的逐点卷积运算。需要在一个周期内读入 4 个通道的输入数据，ram 里的每个地址存入由 4 个通道拼接成的 32 位宽的数据，同理 Weight-rom 中的每个地址拼接了 4 个通道的各 4 个权重，即一个地址存入 128 位宽的数据。

数据的读取方式按输入图片尺寸大小进行地址跳转读取，如输入图片的尺寸为（$k\times k\times n$）则每时钟周期按 addr0、addrk^2、…addr（n-1）k^2 进行跳转读取。频繁的数据访问会带来额外的功耗，因此在卷积计算过程中应加强数据的复用来减少访问的次数，降低功耗。可采用多个卷积核共享输入图片数据的方法减少对内存的访问次数[10]，具体实现如下：在第一个 CLK 读入层间缓存中的 M11～M14，同时读入 4 个通道的各 4 个权重值，W11～W14、W21～W24、W31～W34、W41～W44 进行相乘后各自累加，n 个通道累加完毕后，同时输出 4 个输出通道的同一位置的数，拼接后存入一个地址。

2.3　量化模块的硬件实现

量化模块依据上文公式（4）进行实现，卷积计算累加的结果，需要乘上对应的量化乘数，量化乘数为浮点数的表示形式，要变成定点数的表示形式则需先进行放大，找出误差合适的 M_0 乘数。使得乘法在定点数上进行，乘法结果通过右移位进行缩小还原，最终重新截成 int8 型数据输出，整个过程流水线排开，具体实现过程如图 5 所示。

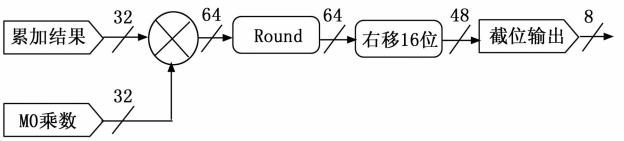

图 5　量化模块的硬件结构

2.4　DW 层设计与实现

DW 卷积不同于一般的常规卷积，它的一个卷积核只负责一个通道，输入图片的通道数经过 DW 层后，通道数不变，在 DW 层并行度上，按 4 输入通道与卷积核内并行相结合的策略，DW 层的卷积过程如图 6 所示。

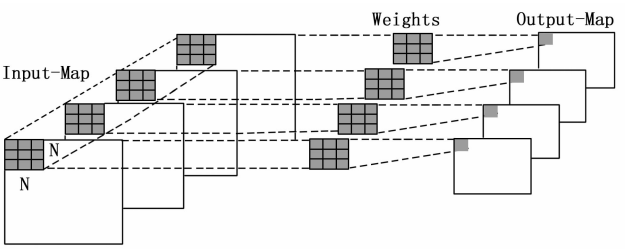

图 6　DW 层并行结构

DW 层硬件架构如图 7 所示，在并行计算上，将卷积计算从输入通道展开。每个输入通道将 3×3 的窗口数据分别送入每组由 9 个乘法器组成的乘法阵列中，在下一个周期通过加法树将乘结果累加起来，存入 32 位的 Sum 寄存器中，经过量化单元量化及 Relu 后，截成 8 位的整型数据。最后对每个通道的卷积结果进行拼接，存入 BRAM 的地址中。

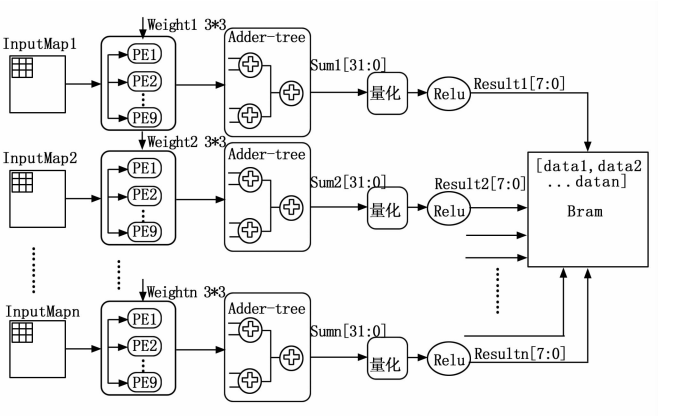

图 7　DW 层的硬件架构

2.5　PW 层设计与实现

PW 为点卷积操作，用于对上一层的输出 map 在深度方向进行加权组合。在 PW 层采取了 4 输入通道与 4 输出通道相结合的并行方式。在一个 CLK 同时读取 4 输入通道同一个位置的数据与 4 组通道上的权重进行计算，依次按输入通道方向读取输入 Map 数据，经过 n 个 CLK 后遍历整个输入通道，输出 4 个输出通道同一位置的数据。PW 层的并行设计如图 8 所示。

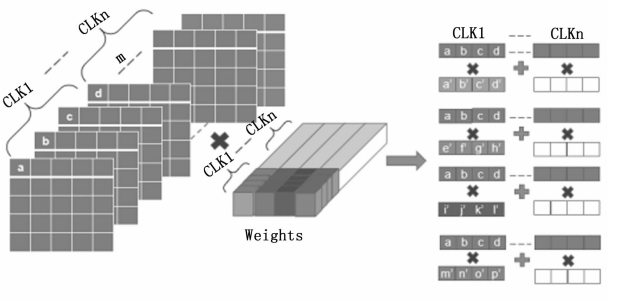

图 8　PW 层并行结构

每个时钟周期输入由 4 个输入通道拼接而成的 32 位数据进入乘累加阵列，当前 4 通道累加结果存入 Sum1 _ 1 寄存器中与先前通道的累加结果 Sum1 相加，1×1 的点卷积将输入通道遍历后，输出最终的累加结果给量化模块，Relu 操作后输出 8 位整型数，最后将 4 个输出通道结果拼接成 32 位数存入 Bram 中，PW 层的硬件实现如图 9 所示。

2.6　层内层间协同配合的流水线结构

构成 MobileNetV2 网络的卷积层类型为标准卷积层、*DW* 卷积层、*PW* 卷积层，其中标准卷积层的计算量可由公式（7）得到，*DW* 卷积层、*PW* 卷积层的计算量可由公式（8）、（9）计算得出。

$$O_{sc}=h*w*C_{in}*C_{out}*K*K \tag{7}$$

$$O_{dw}=h*w*C_{in}*K*K \tag{8}$$

$$O_{pw}=h*w*C_{in}*C_{out} \tag{9}$$

h 和 w 分别为输出特征图的行数与列数，C_{in} 和 C_{out} 为输

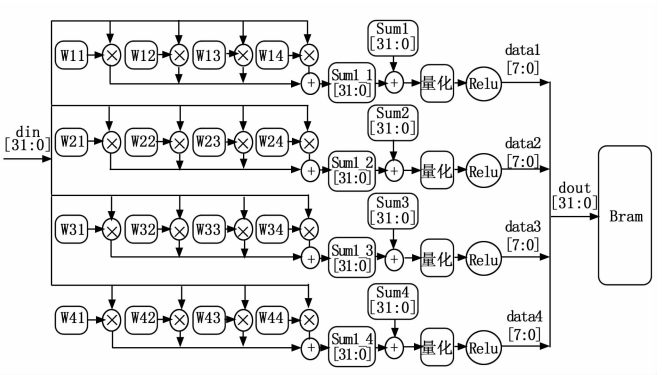

图 9 PW 层的硬件架构

入与输出的通道数，K 为卷积核的尺寸。

当输入为 96 * 96 * 3 时可得 MobileNetV2 网络各部分的计算量如表 3 所示。PW 层的计算量占到了总网络计算量的 89.79%，因此在相邻两个 PW 层间设计层间流水计算结构可以大幅度提升网络的实时性。

表 3 MobileNetV2 网络各部分计算量

卷积类型	计算量	百分比
标准卷积层	390 758 4	3.59%
全连接层	15 354	0.01%
DW 卷积层	718 732 8	6.61%
PW 卷积层	977 365 44	89.79%

本文在 MobileNetV2 网络相邻的 PW 层内及层间以流水线的方式运行计算，流水结构如图 10 所示。数据从 PW1 层输入直到 PW2 层结果输出，一共采用了十二级流水线的设计方式。

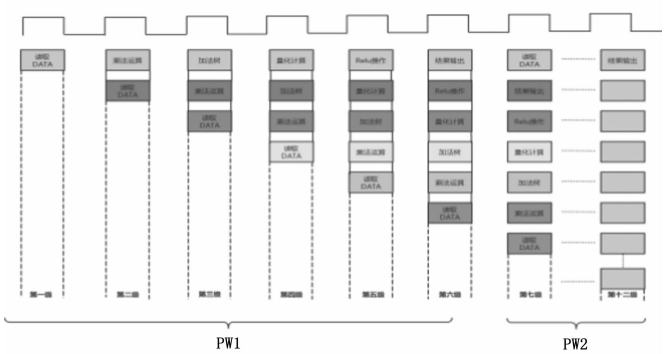

图 10 十二级流水线

在相邻的 PW 层间流水线计算结构设计如图 11 所示，点卷积操作只需单个点的像素就能进行卷积操作，但是完整输出一次操作结果需要遍历整个输入通道。PW1 的权重为（m，n）的二维数组形式，这里 m 为输出通道数，n 为输入通道数。PW 层并行度设置为 $k=4$，则 PW1 层在一个周期读取前一层输出的 4 张图片的 4 个数据后，同时读取 4 组 n 输入通道上的 4 个数，即 4×4 个权重数据，进入 PW1 层计算。在下一个周期 PW1 模块读取下一组输入数据后，权重则按输入通道的方向读取下一组 4×4 个权重数据，以此类推遍历整个输入通道完成一组运算。PW1 层的输出图片尺寸为（y，y，m），当遍历完整个输出通道，m 张图片的首位数据全部写入缓存时，PW1 层发给 PW2 一个 start 信号，PW2 开始读入数据进行运算，PW2 层的权重读取顺序与 PW1 层一致。点卷积运算不改变输入图片的尺寸大小，由于 PW2 的输入、输出通道数的乘积大于 PW1，即 $x \times m \geqslant m \times n$，所以 PW2 遍历完整个输入、输出通道的时间要大于等于 PW1。PW2 只需延迟等待 PW1 层写入缓存的 m 张图片的首位数据后启动，PW1 的写和 PW2 读就不会发生冲突，两层就能以流水的计算方式同时运行计算。

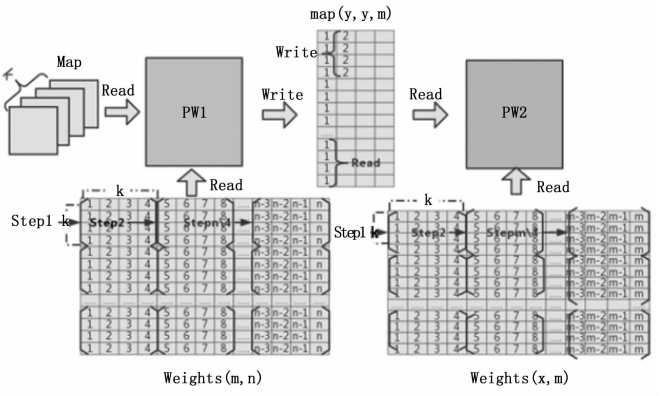

图 11 层间流水计算结构

3 结果分析

3.1 实验环境

本文采用 Xilinx Virtex-7 VC707 开发板作为实验平台，该开发板芯片具有 485 760 个逻辑单元、2 800 DSP、37.08 MB 的 Block RAM，所用开发工具为 Vivado 2018.3，采用 Verilog 语言进行编程，PC 端网络的训练与量化采用了基于 Python 的 PyTorch 框架，CPU 型号为 Inter Core i5 -6200U，主频为 2.3 GHz。下面讨论实现的效果及性能比较。

3.2 资源与能耗

主要资源消耗情况如表 4 所示，其中 LUT 资源表示查找表，BRAM 代表片内存储资源，FF 为触发器资源，DSP 为计算单元。LUT、FF、DSP 资源主要用在了卷积功能模块以及数据流的实现上，消耗量均不大。由于本文为了减少片外 DDR 的访问降低系统功耗，将所有的权重及层间缓存都布置在了片内 BRAM 中，所以片内 BRAM 资源消耗较大。

表 4 资源占用情况

资源类别	LUT	BRAM	FF	DSP
消耗资源	113 295	993	69 429	717
资源总数	303 600	1 030	607 200	2 800

整个系统的能耗如图 12 所示，在 200 MHz 时钟下整体功耗为 8.24 W，扣除数据传输接口 PCIE 功耗 2.108 W，整个硬件加速部分的功耗仅为 6.13 W。

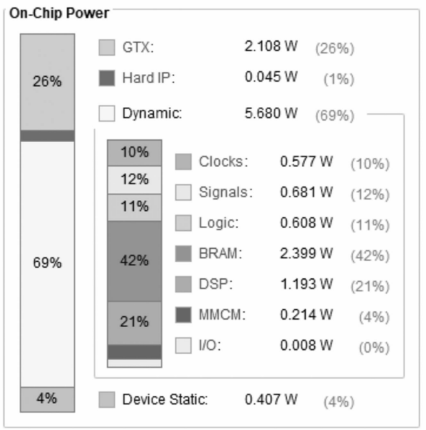

图 12　系统功耗

表 5 列出了本文实现与 CPU、GPU 平台性能的对比。200 MHz 时钟下本文的硬件实处理单张图片仅耗时 0.64 ms，能耗比为 27.74 GOP/s/W 是 CPU 平台的 92 倍，GPU 平台的 25 倍。可见将 CNN 部署在 FPGA 上无论从推理时间还是功耗上都具有极大的优势。

表 5　不同平台性能对比

平台	单图时间/ms	性能/(GOP/s)	功耗/W	能耗比/(GOP/s/W)
i5－6200U CPU	27.07	4.02	13.49	0.30
NVIDIA GTX 2070	1.60	68.03	60.2	1.13
本文 Virtex－7 VX485T	0.64	170.07	6.13	27.74

表 6 列出了在不同的 FPGA 平台实现 CNN 的性能对比，可以看出本文较文献［14］、［19］和文献［20］在计算性能上存在一定的优势，虽然计算性能上较文献［18］还有一定差距，但是在功耗上优势明显，能效比（Gop/s/W）较高。在追求高计算性能的同时能够兼顾功耗，达到计算性能与功耗两者的平衡，更加适用于实际的应用场景。

表 6　不同 FPGA 实现性能对比

	［14］	［18］	［19］	［20］	本文
平台	XCZU7EV	Arria 10	VC707	XC7Z045	VC707
实现网络	Mobile NetV2	Mobile NetV2	Shuffle Net	VGG16	Mobile NetV2
计算性能/(GOP/s)	89.60	382.96	107.9	137	170.07
功耗(W)	6.2	29.4	9.26	9.63	6.13
能效比/(GOP/s/W)	14.45	13.02	11.7	19.50	27.74

4　结束语

本文提出的基于 FPGA 加速的低功耗的 MobileNetV2 网络识别系统，将权重和缓存均放入片内存储，克服了片外存储带来的带宽限制与额外的功耗，小型网络量化会带来较大的精度损失，因此采用量化感知训练的方式进行二次训练提高网络的精度。为了进一步节约片内存储资源，优化了数据存储结构，复用了输入特征图。在相邻的 PW 卷积层内与层间以流水线的方式展开，极大的提升了网络的实时性。

本文设计的 MobileNetV2 网络虽然将权重和层间缓存均放入片内存储，在一定程度上克服了片外存储带来的带宽限制，但是受限于片内 BRAM 资源，每层的并行度并不能设计的足够高，如何在有限的片内 BRAM 资源下，进一步提高网络的并行度是需要在未来的研究中考虑的问题。

参考文献：

［1］赵大贺，姚晓通．基于轻量级卷积网络的复杂背景下接触网绝缘子识别［J］．电瓷避雷器，2022（3）：172－178.

［2］卢海燕，赵红东，王添盟，等．基于轻量级卷积神经网络的红外行人行为识别［J］．传感器与微系统，2022，41（9）：129－131.

［3］章　琦，朱鸿泰，程　虎，等．轻量级红外弱小目标检测算法［J］．激光与光电子学进展，2022，59（16）：282－288.

［4］SANDLER M，HOWARD A，ZHU M，et al. MobileNetV2：Inverted Residuals and Linear Bottlenecks［C］//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition，2018：4510－4520.

［5］崔江伟，周勇胜，张　帆，等．基于流水线架构的卷积神经网络 FPGA 实现［J］．北京化工大学学报（自然科学版），2021，48（5）：111－118.

［6］黄　瑞，金光浩，李　磊，等．轻量化神经网络加速器的设计与实现［J］．计算机工程，2021，47（9）：185－190.

［7］CHEN Y，LUO T，LIU S，et al. DaDianNao：A Machine-Learning Supercomputer［C］//2014 47th AnnualIEEE/ACM International Symposium on Microarchitecture，2014：609－622.

［8］DU Z，FASTHUBER R，CHEN T，et al. ShiDianNao：Shifting vision processing closer to the sensor［C］//2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture（ISCA），2015：92－104.

［9］BAI L，ZHAO Y，HUANG X. A CNN Accelerator on FPGA Using Depthwise Separable Convolution［J］. IEEE Transactions on Circuits and Systems II：Express Briefs，2018，65（10）：1415－1419.

［10］LI Z，WANG L，YU J，et al. The Design of Lightweight and Multi Parallel CNN Accelerator Based on FPGA［C］//2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference（ITAIC），2019：1521－1528.

［11］ZHANG C，LI P，SUN G，et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks［C］//FPGA 2015－2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays，2015：161－170.

［12］LI H，FAN X，JIAO L，et al. A high performance FPGA-based accelerator for large-scale convolutional neural networks［C］//2016 26th International Conference on Field Programmable Logic and Applications（FPL），2016：1－9.

（下转第 234 页）