分 类 号： TN79　　密　　级：公开

论文编号：2016021654

贵　州　大　学

2019 届 硕士研究生学位论文

深度学习中的卷积神经网络硬件加速系统设计研究

学科专业：电路与系统

研究方向：软硬件协同设计

导　　师：周　骅

研 究 生：王　昆

中国 ▪ 贵州 ▪ 贵阳

2019 年 6 月

目录

摘要 I
ABSTRACT II
第一章 绪论 1
1.1 课题研究背景及意义 1
1.2 国内外研究现状 2
1.3 本文主要工作 5
1.4 论文章节安排 7
第二章 卷积神经网络与本文硬件加速系统设计 9
2.1 深度学习简介 9
2.1.1 基本概念 9
2.1.2 前向传播反向传播 13
2.2 卷积神经网络结构及模型 14
2.2.1 卷积神经网络的结构 16
2.2.2 卷积神经网络并行性 18
2.2.3 常见卷积神经网络模型 18
2.3 硬件加速技术 18
2.3.1 硬件加速平台简介 18
2.3.2 FPGA 芯片结构及特点 19
2.3.3 硬件加速优化方法 20
2.4 本文卷积神经网络硬件加速系统设计 20
2.5 本章小结 22
第三章 卷积神经网络硬件结构设计 23
3.1 本文卷积神经网络模型 23
3.2 卷积神经网络硬件设计思路 24
3.3 卷积神经网络硬件结构设计 24
3.3.1 卷积层硬件设计 24
3.3.2 池化层硬件设计 26
3.3.3 激活函数硬件设计 28
3.3.4 函数分类器硬件设计 29

3.3.5 权值存储硬件设计 29
3.4 硬件优化设计 32
3.5 本章小结 36
第四章 基于 ZYNQ 的实时识别框架设计 38
4.1 图像采集及显示 38
4.2 数据存储结构设计 44
4.2.1 VDMA 与 DDR 44
4.2.2 视频时序控制模块 48
4.2.3 降采样模块与 BRAM 48
4.3 实时识别框架系统实现 49
4.4 本章小结 50
第五章 硬件加速系统测试与验证 51
5.1 卷积神经网络各模块测试和验证 51
5.2 卷积网络模型权值训练 54
5.3 卷积神经网络硬件加速实时识别框架系统测试 57
5.4 本章小结 59
第六章 总结与展望 61
6.1 本文总结 61
6.2 工作展望 62
致谢 63
参考文献 64
附录 67

摘要

近几年来伴随着深度学习所带来的新的机器学习热潮，深度神经网络已经广泛的应用于图像识别、图像分类、目标检测和自然语言处理等不同的大规模机器学习问题当中，并且已经取得了一系列突破性的实验结果与实际应用，如今深度学习其强大的特征学习能力与识别分类能力被广泛的研究与关注。但由于深度学习中的卷积神经网络模型通常具有深度高、层次复杂、数量级大、并行度高、计算和存储密集的特征，从而使得大量的卷积计算操作和池化计算操作在具体应用中成为巨大的瓶颈，并且大量层间计算结果的存储对于计算机的存储结构也提出了较高的要求，使其在实时的应用场景下面临着巨大的挑战。现场可编程阵列 FPGA（Field-Programmable Gate Array），是一种电路密集度大的运算加速器件，它集成了丰富的内部存储硬件资源、灵活的可编程逻辑资源以及高性能的计算资源，能够充分发挥卷积神经网络结构并行特性，并且能够在尺寸要求小、功耗限制低情况下实现卷积神经网络的高速运算，是实现卷积神经网络运算的理想平台。

本论文主要针对深度学习中的图像识别任务进行了硬件加速系统设计研究。文章主要根据卷积神经网络的结构特点，在基于 ZYNQ 系列芯片的 FPGA 上将卷积神经网络进行了硬化实现，利用 FPGA 的并行计算特性与流水线技术减少了卷积神经网络的计算时间，从而实现了卷积神经网络的硬件加速；同时为了满足实时场景下对图像识别的应用需求，本文设计出了一种实时识别硬件系统框架，采用软硬件协同的方式，使用 ZYNQ 系列芯片的 ARM 完成对输入图像数据的实时采集、存储和显示，将采集存储的数据通过 AXI4 总线传输至 FPGA 中硬化后的卷积神经网络来完成对图像的实时识别，并且该系统框架还可以替换不同的硬化卷积神经网络模型，满足多场景下的实时识别任务需求。

实验结果表明，本文设计的硬化卷积神经网络模型能够在单个时钟周期内完成 528 次卷积运算，相较于通用 CPU 的计算效率得到了显著提升；在对权值参数进行 11 位定点量化后网络的识别率为 97.8%，具有较高的准确率；并且本文设计出的实时识别硬件系统框架能够实现对摄像头采集图像的实时识别，同时结合 ZYNQ 器件中高度模块化设计使得整个系统框架具有移植性高的特性，且系统整体运行时所需的功耗低。

关键词:深度学习；卷积神经网络；硬件加速；实时识别；

ABSTRACT

In recent years, with the new machine learning boom brought about by deep learning, deep neural networks have been widely used in different large-scale machine learning problems such as image recognition, image classification, target detection and natural language processing, and have been obtained. A series of breakthrough experimental results and practical applications, today's deep learning of powerful feature learning ability and recognition and classification capabilities have caused extensive research and attention. However, the convolutional neural network model in deep learning usually has the characteristics of high depth, complex hierarchy, large order of magnitude, high degree of parallelism, computational intensiveness and storage intensiveness, which makes a large number of convolution calculation operations and pooled computation operations in specific applications. It has become a huge bottleneck, and the storage of a large number of inter-layer calculation results also puts high demands on the storage structure of the computer, making it face enormous challenges in real-time application scenarios. As a highly intensive computing acceleration device, Field-Programmable Gate Array (FPGA) contains a large number of programmable logic resources, storage resources and computing resources, which can fully utilize the parallel characteristics of the convolutional neural network structure. The high-speed operation of the convolutional neural network can be completed under the constraints of small size and low power consumption, and is an ideal platform for realizing convolutional neural network operations.

In this thesis, the hardware acceleration system design research is carried out for the image recognition task in deep learning. According to the structural characteristics of the convolutional neural network, the convolutional neural network is hardened on the FPGA based on ZYNQ series chip. The parallel computing feature and pipeline technology reduce the computation time of the convolutional neural network, thus achieving the hardware acceleration of the convolutional neural network. At the same time, in order to meet the application requirements of image recognition in real-time scenarios, this paper designs a real-time recognition hardware. The system framework adopts the software and hardware coordination method, and uses the ARM of ZYNQ series chip to complete the real-time acquisition, storage and display of the input image data, and transmits the collected data to the hardened convolutional neural network in the FPGA through the AXI4 bus. Real-time

recognition of images, and the system framework can replace different hardened convolutional neural network models to meet real-time identification task requirements in multiple scenarios.

The experimental results show that the hardened convolutional neural network model designed in this paper can complete 528 convolution operations in a single clock cycle, which is significantly improved compared with the general CPU. After the 11-bit fixed-point quantization of the weight parameters The recognition rate of the network is 97.14%, which has a high accuracy rate. The real-time identification hardware system framework designed in this paper can realize the real-time recognition of the captured image of the camera. At the same time, combined with the highly modular design of ZYNQ device, the whole system framework has the transplant. High performance and low power consumption required for overall system operation.

Keywords: Deep learning; CNN; hardware speedup; Real-time recognition;

第一章　绪论

1.1 课题研究背景及意义

随着深度学习的广泛应用与发展，卷积神经网络（Convolutional Neural Network）在越来越多的场景中被使用[1-4]，特别是在图像识别领域，已经实现了突破性的发展。在 2012 年 Alex Krizhevsky 设计的 AlexNet 网络将图像识别的错误率降低到 16.4%，2014 年 VGGNet 网络将图像识别的错误率降低至 7.3%，2015 年 Kaiming He 提出的 ResNet 网络更是将图像识别的错误率降低至 3.57%，这些发展使得图像特征提取的方式逐渐由传统的人工提取变为使用深度卷积神经网络学习特征方式所取代。

深度卷积神经网络是一种多层神经网络结构，具有很强的容错性，学习和并行处理能力。它是一种具有多层感知器，本地连接和权重共享的网络结构，它减少了网络结构、网络模型的复杂性和网络连接权重的数量[5-8]。而且使用深度卷积神经网络在进行图像识别时可以直接将图像数据通过网络的输入层进入网络，利用网络中每一层的卷积核进行相应的特征提取，避免了传统算法对图像进行建模以提取特征[9-11]。因此，近年来卷积神经网络结构在也视频监控、机器视觉、模式识别、图像搜索等领域得到越来越广泛的应用。

随着深度学习的高速发展之下，深度学习中卷积神经网络模型的规模向着更加深层次的方向进行发展，使得其计算的复杂性也呈指数形式上升。当前最大的卷积神经网络模型之一每进行 224x224 分辨率图像识别分类则需进行高达 390 亿次的运算，并且卷积神经网络模型的运算量是和图像数据的尺寸大小成正比，图像的分辨率越高则所需的计算时间越长。在深度卷积神经网络中，卷积层中的运算耗时大，其基本占据了整个卷积神经网络运算时间的 90%，使得其计算速度影响整个卷积神经网络的性能。但是在现如今将深度卷积神经网络进行实现的硬件载体主要是使用商用的 GPU 图形处理器与商用的通用处理器 CPU 来实现的，GPU 虽然灵活但是其功耗较大，而通用 CPU 的设计目的是解释计算机软件中的计算机指令和过程数据，因此其自身的特性是不够充分利用卷积神经网络内部结构的并行性的，使得计算效率低，并且在很多场景应用中还需要对视频的内容进行实时的识别与分析。

鉴于此背景，为了提升卷积神经网络的计算效率、加速整体系统性能和满足实时识

别的场景需求，有必要设计出一种卷积神经网络硬件加速器和实时识别硬件框架，可将硬化后的卷积神经网络部署在实时识别硬件框架中，达到硬件加速从而来提升系统整体计算性能和实时识别的目的。

1.2 国内外研究现状

深度卷积神经网络是机器学习方法的一种，在计算机科学领域，神经网络指由多个简单计算元素（神经元）层所组成的系统。这些神经元仅仅大致地模仿了人脑中的神经元，但却能通过加权连接互相影响，通过改变连接的权重来改变神经网络所执行的计算。它的起源要追溯到 1943 年麦卡洛克(McCulloch)和皮茨(Pitts)的工作，他们所设计出的神经元模型构成的网络模型原则上可以计算任何函数；1949 年学者赫布(Hebb)的工作阐述了突触修正的生理学模型；1958 年学者罗森布拉特(Rosenblatt)第一个提出了感知器作为有监督学习的模型；在 1962 年经过表皮层的深入研究，有关学者还提出了感受视野的基本概念为神经网络的发展做出了一定的贡献[12-13]；80 年代霍普菲尔德(Hopfield)发表了一篇引起巨大争论的论文，也推动了神经网络研究的高速发展；同年代，Geoffrey Hinton 就已经开始提倡使用机器学习方法进行人工智能研究，他希望通过人脑运作方式探索机器学习系统。在受人脑的启发后，他和其他研究者提出了人工神经网络（artificial neural network），为机器学习研究奠定了基石[14-15]。

在 1986 年，Hinton 与 David Rumelhart 和 Ronald Williams 提出了反向传播，Hinton 等研究者表示反向传播算法允许神经网络探索数据内部的深层表征，因此神经网络才能解决以前被认为无法解决的问题，现阶段反向传播算法目前已经成为了训练深度神经网络所必需的算法，同时 Hinton 和 Terrence Sejnowski 还提出了玻尔兹曼机，它是第一个能学习神经元内部表征的深度神经网络，这种表征既不是输入也不是输出的一部分[16-18]。

同样在 80 年代，LeCun 构建了卷积神经网络，这是该领域的一项重要理论，对于提高深度学习效率至关重要。在 80 年代后期，LeCun 利用手写数字图像训练了第一个卷积神经网络系统。现如今，卷积神经网络已成为计算机视觉、语音识别、图像识别、图像合成和自然语言处理领域的行业标准，并且卷积神经网络有着广泛的应用，如目标检测、医学图像分析、语音分析和信息过滤等。

在 90 年代，Bengio 提出了将神经网络与序列的概率建模相结合，例如隐马尔可夫

模型这种序列的概率建模方法，这些创新观点被 AT&T/NCR 所接受，并被用于阅读手写支票，该系统被认为是九十年代神经网络研究的巅峰之作，现代基于深度学习的语音识别系统都是在这些概念上继续扩展的[19]。

在 2000 年 Bengio 等研究者提出了高维词嵌入作为词义的表征方法，对自然语言处理任务产生了巨大而持久的影响，包括机器翻译、知识问答、视觉问答等。在 2006 年，Hinton 基于深度信念网提出了一种快速逐层非监督的训练算法，为多层神经网络训练方法做出了革命性的进展。2010 年 Bengio 和 Ian Goodfellow 等研究者提出的生成对抗网络（GAN），这项研究引起了计算机视觉和计算机图形学的革命，使得计算机能生成与原始图像相媲美的图像。

2012 年 AlexNet 在 ILSVRC 图像识别竞赛中所带来的惊人表现，让学术界看到了深度学习所蕴含的巨大潜力[20-23]。在随后的几年中，随着 GoogLeNet、VGGNets、ResNets 等模型的提出，CNNs 的识别准确率持续提高，让计算机拥有了超越人类的图像识别能力。

同时，深度学习在自然语言处理（Natural Language Processing, NLP）方面同样取得了巨大的成功，不但有 Siri 和 Cortana 这样可以与人类正常交流的对话机器人，甚至还能够进行写诗和作曲的创作。2016 年由谷歌公司提出设计的的 AlphaGo 击败人类标志着深度学习的发展已经取得了不可思议的成果，AlphaGo Zero 更是采用无监督学习，以难以置信的战绩碾压人类。

在图像检测领域当中， Jitendra Malik 等人创造性的提出了 R-CNN，利用网络模型生成候选区域，再对候选区域输入网络中进行特征提取，使得图像目标检测技术取得重大进展。2015 年 Ross GirshickFast 等人提出了 Fast R-CNN，针对 R-CNN 效率低下的问题做出了感兴趣区域池和整合模型的改进，将图像检测识别训练时间相较于 R-CNN 提升了近 9 倍。2015 年，Wei Liu 等人提出了 SSD（Single Shot MultiBox Detector）使得对图像检测的效率更加迅速。2016 年 Jian Sun 提出的 Fater R-CNN 对 Fast R-CNN 的区域建议做出了改进，使得其在图像检测准确率达到了 78.8%。

而以上的研究主要是依据于卷积神经网络，作为典型的密集型算法，卷积神经网络中包含着大量的独立重复的乘法和加法运算，目前卷积神经网络主要运行在 CPU 或高性能 GPU 搭建而成的平台及服务器上，并不能完全开发其内部并行的特点，而且成本、

功耗、体积等方面并不如人意。但现场可编程门阵列 FPGA 其自身集成了丰富的内部存储硬件资源、灵活的可编程逻辑资源以及高性能的计算资源，能够充分发挥出卷积神经网络的并行性，以较低功率完成密集的卷积神经网络运算。因此，在 FPGA 上实现卷积神经网络算法或利用 FPGA 进行卷积神经网络计算加速已经成为研究的一大热门。

Laure Vetan 等人设计出了一种了基于 FPGA 的卷积神经网络硬件加速设计。该设计中指出在 FPGA 内实现卷积神经网络的最大限制就是 FPGA 芯片的存储带宽资源，该设计旨在利用最少的片上存储资源完成卷积神经网络的计算。其工作原理是就是将输入图像数据和权重存入不同的片上存储单元当中，每个运算单元通过将地址的选择器读取输入图像的数据，从而来完成一次计算后在更新权重和存储地址，直到将所有的权重与输入图像运算完毕后再从片外存储读取新的输入图像。这种设计能够复用输入图像的数据，使其能够在最少的存储资源下完成密集的运算[24-26]。

Srivastava N 等人使用了 ASIC 专用芯片完成机器学习中的通用计算部分，并以卷积神经网络 CNNs 为例，设计出一个具有高数据吞吐率的通用加速器。Alexandre Alahi 等人利用 GPU 器件提供的 CUDA 编程框架对深度可信网络内的预训练与全局训练算法进行了加速，该加速器相较于通用 CPU，本地的预训练效率可达到 22 倍的加速效果，全局训练更是有 33 倍左右的性能提升。

Yu Chen 等人提出了基于 FPGA 的可配置的 CNN 算法实现方法，其主要工作是将 FPGA 芯片上的存储单元与运算单元配置成阵列，然后根据使用者不同的需求来进行阵列中元素的激活，来达到计算不同大小的卷积内核、移动步长、输入通道、输出通道的卷积神经网络的目的。

余子健等人通过优化卷积神经网络算法，压缩节点和权值数据位宽，来设计出了一种 FPGA 的加速器。Stefano Carlo 等人提出了一种基于 FPGA 的 2D 卷积计算方法[27-30]。该设计中使用了缓存器来规避 FPGA 上运算单元与存储单元的直接连接，通过加法器的树形结构来进行卷积神经网络的卷积运算，在保持高性能的同时尽可能减少芯片了的使用面积，从而满足了嵌入式系统的需求。

Yongmei Zhou 等人提出了在 FPGA 上实现 5 层的 CNN 网络结构。其内部并行方式为输出图像并行，输入图像存储在外存储设备中，只在需要使用输入图像进行计算时才将其读取进片内存储单元。这种设计能达到峰值为 16.58GMAC 的最高性能，而且所占

用的硬件资源相对较少。

魏小松等人设计出了一种基于 FPGA 训练卷积神经网络加速器，该加速器的计算性能是通用 CPU 平台的 6.8 倍，能量效能是 GPU 平台的 9.7 倍，充分发挥出了硬件加速的优势。Fan Sun 等人利用 PiPe 技术设计出了一种高效能的卷积神经网络加速器，相较于 CPU 计算速度提升了 3.8 倍，能效更是提升了 14.3 倍。

在基于 FPGA 的卷积神经网络硬件加速系统研究中，Xiaosong Wei 和 Fan Sun 的卷积神经网络加速器的效果较为明显，但是他们的研究仅限于将卷积神经网络模型进行了实现，并未对其硬件结构进行充分的优化设计，不能够充的发挥出 FPGA 的并行运算性能；同时也不能够灵活的完成对实时场景下的识别任务以满足不同需求，并且通用性低、移植性差。

本文在深入研究卷积神经网络结构后，针对卷积神经网络的结构进行了优化的硬件实现，并且设计出了一种实时识别的系统框架，能够将硬件实现的卷积神经网络嵌入其中以达到实时识别场景任务的需求，从而充分发挥出 FPGA 可编程的灵活性和并行特性的潜能，且具有一定的移植性。

1.3 本文主要工作

通过对本论文课题的深入研究，文章中首先对本文所设计的卷积神经网络结构模型进行了硬化实现，对本文所设计的卷积神经网络模型系统结构中，包括卷积层、采样层、输出层、激活函数实现了硬件结构设计，并且完成了将卷积神经网络在 FPGA 上进行部署运行；其次设计并实现了一种基于 ZYNQ 的实时识别硬件系统框架设计，该硬件系统框架设计能够实现对输入图像数据进行实时采集显示，并将图像数据内容传输至硬化实现后的卷积神经网络进行图像内容识别并给出识别出的结果及其分类。同时文章还主要研究了卷积神经网络中卷积运算的硬件优化实现方法，通过研究卷积神经网络的结构对并行计算进行了硬件实现优化处理，进而实现加速整个系统模型的卷积运算效率；文章中还提出了一种使用硬件实现的函数分类器来完成对整个卷积神经网络识别结果的输出。文章最终所采用 Xilinx 公司低成本 ZYNQ 系列芯片硬件平台来验证实现整个系统设计方案的功能性和性能，通过使用 MINIST 手写数据集分类的实时识别场景对系统进行了功能验证、系统整体运行的性能和系统整体设计方案的可行性。本文中主要涉及

到的研究和工作内容包括以下 6 个方面：

1）深度学习中卷积神经网络总体方案的硬件设计

针对卷积神经网络结构，文章设计出了一种深度学习中的卷积神经网络硬件系统。卷积神经网络是整个系统设计的主要组成部分，而其中的卷积运算是卷积神经网络的核心部分，是整个系统后续执行的重要一步，本课题将会根据常见的卷积运算硬件实现方法来分析典型的输入数据卷积运算，并针对本论文设计的数据缓存结构设计一种合理的卷积运算硬件结构。系统硬件设计根据数据流的输入，使用内置寄存器模板完成对数据的存储，将存储的数据与训练好的权值参数使用乘法器完成图像特征提取的卷积运算；在提取到图像特征参数后，使用内置的模板完成特征图像的降采样处理，从而减少后级的权值参数；再通过研究函数分类器的实现方案，完成对输入图像的识别分类。

2）卷积网络图像输入数据缓存处理

由于文章输入数据采用的是 MINIST 数据集进行系统的功能验证，该数据为图像数据，若采用并行输入则会消耗大量的逻辑资源，为了减少硬件逻辑资源的消耗，文章输入数据的方式采用的是串行输入，再对其进行内部重新构建，必须对其进行处理才能够进行后续的卷积计算，从而才能够完成整个系统的识别任务。因此如何有效的对输入数据进行缓存处理显得至关重要，合理的存储规则能够有效地提高整个系统的运行效率，从而提升整体硬件系统的数据吞吐量，加速系统运行的识别效果，所以本课题根据设计出的卷积网络模型选择了一种合适的数据缓存处理方法来提高数据的吞吐量。

3）卷积的并行化运算硬件结构研究

使用 FPGA 作为实现平台是因为其硬件结构具有天然的并行运算特性，从而能够加速整个系统中卷积运算的执行速度。为了加速卷积神经网络模型的卷积运算效率，必须充分的利用 FPGA 的硬件电路特性，设计出合理的并行卷积运算硬件结构，会使得整个卷积神经网络硬件系统能够在进行卷积运算时高效并行的执行当前阶段的卷积运算，从而来达到整个系统的加速处理，提升系统性能。本课题将深入研究并对比现有的常见卷积并行运算结构实现方法的优缺点，提出了一种卷积神经网络卷积运算的并行化硬件结构设计方案，并对其功能性进行了检验，达到了卷积运算并行加速的目的。

4）函数分类器的硬件实现研究

在卷积神经网络系统输出层的函数分类器中多数是不利于或难以硬件实现的，其主

要函数使用的是 Softmax、Sigmoid 等，这些函数是连续函数，不利于离散处理，并且为了完成最终输出结果，还必须对卷积运算结果进行概率输出，而概率分类器涉及到的指数运算不利于硬件实现，设计出一种硬件实现的函数分类器对于整个系统设计必不可少。为此本论文通过结合软硬件设计的方式，通过使用 Matlab 来验证论文提出函数分类器的硬件设计合理性、功能性，再使用硬件描述语言来完成具体的函数分类器硬件实现，以达到系统识别的预期目标。

5）实时识别框架高速数据存储交互

为了实现实时识别的系统框架设计，需要将摄像头采集的图像视频数据高速的存储至系统当中并且传输至硬化的卷积神经网络结构中进行识别与分类，最终将分类结果进行显示。因此必须设计出一种高速的数据存储交互系统，文章主要研究了 ZYNQ 系列芯片内部的 DMA 存储与 VDMA 存储系统，实现了将采集图像数据通过 VDMA 与 AXI4 总线存储至高速存储器 DDR 中，并将 DDR 中的图像数据传输至硬化后的卷积神经网络中进行识别与分类，从而满足实时识别的任务需求。

6）数据流时序控制

文章中当图像数据通过硬化的卷积神经网络系统中时每次处理运行识别分类需消耗一定的时钟周期，且外部图像数据的采集与内部 DDR 数据交互的访问也需要一定的时钟周期。为了使整个系统合理的运行，因此需要对模块间的数据流进行时序控制，才能确保系统的稳定工作。文章主要研究了不同时钟域的数据交互方法，针对文章中的实时识别框架系统和高速存储硬件结构进行了跨时钟域的数据交互处理，使得整个系统的数据流时序能够准确无误的运行，从而确保实时识别系统框架对数据的高速正确处理。

1.4 论文章节安排

围绕文章的主要工作内容，本文的章节安排内容如下：

第一章：绪论，首先简要的介绍了当前深度学习中卷积神经网络的发展与应用，同时对研究 FPGA 实现卷积神经网络硬件加速系统的背景和意义进行了详细的介绍。然后对当前的 FPGA 硬件加速系统研究情况进行了说明与介绍。最后对本文所做的主要工作进行了详细的阐述说明，并且给出了文章的主要章节安排。

第二章：卷积神经网络与本文硬件加速系统设计，首先简要的阐述了深度学习，包

括了其相关的知识、基本理论和使用的训练方法；对卷积神经网络的组成进行了介绍，对其内部组织结构予以详细说明，同时给出了典型的卷积神经网络模型；然后分析了多平台的硬件加速技术、FPGA 器件的硬件电路优势与实现卷积神经网络的可行性和硬件加速优化方法；最后对本文研究设计的卷积神经网络硬件加速系统进行了阐述。

第三章：卷积神经网络硬件结构设计，首先基于 ZYNQ 系列芯片内的 FPGA 设计出了本文硬化的卷积神经网络模型，并分析了模型的主要设计思路，包括了对整体硬件卷积神经网络结构、功能实现、结构特点的介绍；然后对卷积神经网络中的每层设计进行了详细阐述说明，包括卷积层、池化层、激活函数、权值存储和函数分类器的实现；最后对卷积运算的硬件优化实现进行了详述，完成了本文所设计的卷积神经网络硬化实现。

第四章：基于 ZYNQ 的实时识别框架设计，首先阐述了实时识别框架的结构设计，对图像数据采集与显示进行了详细说明；然后对实时识别框架中的高速数据存储结构实现进行了介绍，包括对数据时序控制、图像降采样存储和高速数据间的交互；最后完成了整个实时识别硬件框架的构建。

第五章：硬件加速系统测试与验证，首先对文章中设计出的硬化卷积神经网络各模块进行了功能测试；然后根据文章在硬件上设计好的卷积神经网络模型，在软件上对其进行模型构建，训练出卷积神经网络权值参数将其定点量化后将其存储至权值存储模块中；最后将硬化后具有权值参数的卷积神经网络模型嵌入实时识别系统框架中进行测试，从而达到硬件加速和实时识别的目的。

第六章：总结与展望，总结了本文所做的研究内容与工作成果，同时对今后的工作和后续的改进做出了展望。

第二章　卷积神经网络与本文硬件加速系统设计

在本章节中对深度学习、卷积神经网络和 FPGA 的相关硬件加速技术与相关知识进行了简要介绍。首先简要阐述了深度学习中的基本概念和知识，包括基本理论和使用的训练方法；对卷积神经网络的结构进行了详细的介绍说明，并且给出了常见的卷积神经网络模型结构；然后分析了卷积神经网络的硬件加速相关技术与 FPGA 具有的特点；最后对本文研究的卷积神经网络硬件加速系统设计进行了阐述。

2.1 深度学习简介

2.1.1 基本概念

深度学习是当前机器学习学科中最繁荣的一个分支，也是整个人工智能领域中应用前景最为广阔的技术[31-34]。近几十年来，构建模式识别和机器学习系统都要求技艺高超的工程师和富有充足经验的领域专业人员来设计出特征提取器（Feature Extractor），并且需要将原始的输入数据（如图像的像素值）转化成合适的中间表示形式或者具有一定的特征向量（Feature Vector），再使用训练系统（通常称之为分类器）对输入数据或者模式进行检测和分类，其基本流程如图 2.1 所示。

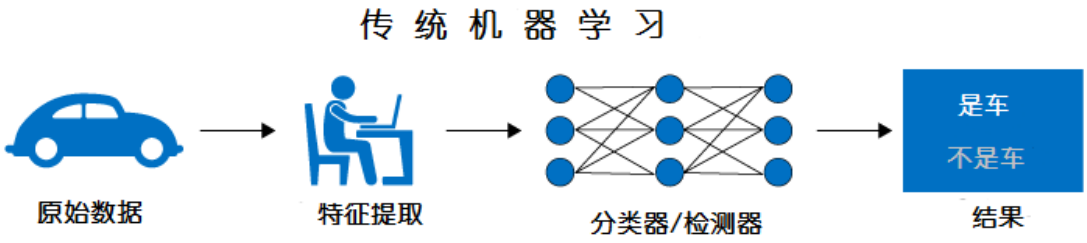


图 2.1 传统机器学习处理模式

随着近几年的发展，现如今机器学习学科的发展已经成为最为重要的一个研究分支，也是整个深度学习领域当中应用前景最为广阔的一门技术。通过机器学习技术可以为当代的许多方面提供智能的社会发展动力：对于网络资料的查阅、论坛里的内容屏蔽、网上商城内部的推荐广告，以及它也越来越多地出现在消费类产品中，如最新款式相机和智能移动设备。机器学习系统还被用于识别图像或视屏当中的待检测物体，并且还能够将已经录制好的语音自动翻译为文本的形式，对不同搜索内容还会进行匹配与之相关的

新闻项目。在社交论坛中的帖子或者搜索用户具有感兴趣的使用产品，并给其选择相应的搜索结果，这些应用程序被开发者和供应商越来越多地使用在日常生活当中。

在深度学习和人工智能发展的早期阶段里，随着计算机在智能化和自动化方面所取得的成功，其的主要实现方法和实施思路就是通过将人类所总结的相关理论知识通过使用一系列已经预定好的、具有规范形式及形式化的数学描述方法和规则来进行表示，之后将所需要结局的实际问题使用自动化和智能化的相关程序来代替人类去解决相关的问题[35-37]。在早期，以知识为根基的一种专家系统（knowledge-based expert system)就是这方面的典型代表，该系统是利用将某个学科领域和研究领域中专业人员的专家经验通过规定好的、具有规范化的知识表示方法通过计算机写成相关的规则，计算机系统依照相应的规范规则去计算处理从而实现出专家相应的思维方式。

不过在实际的应用中，此类的专家系统都没有获得太大的成功，其中最主要的原因是该系统只能够根据已知的、已经成熟的方案和理论知识去解决实际的问题，但是其局限性体现在系统已经被写入了大量的数学规则，且该规则一般具一定的数量限制，而规则数量则决定了系统对不同情况的适应程度，如果在处理相应的场景时该系统内部有相应的数学规则，则能够处理相应的任务需求，如果该系统没有相应的相关数学规则，则不能够处理相应的任务需求。然而问题的发生状况是无限多样的，专家系统通过用有限的规则去处理完成无限的可能，注定是难以成功的。

在早期人工智能方面取得成功的项目多数解决的是具有明确规则和条件的问题，例如经典的西洋跳棋。在 1997 年 IBM 公司利用深度学习所设计出的“深蓝”计算机系统在国际象棋上战胜当时的人类世界冠军卡斯帕罗夫就是这个方面最典型的例子。对于普通的人类来说下象棋是非常具有挑战的项目，但是该项目和真实世界的复杂程度和多样性相比较而言，国际象棋其实终究只是一个简单问题。因为整个棋盘上只有 32 个棋子和 64 个可以落子的位置，其每一步的落子规则也是非常明确的，所有可能的落子方法组合是有限的，可以通过数学相关公式进行描述从而被挪列出来，再利用计算机系统的强大高速的计算能力来辅助以启发式搜索等算法，在电子芯片的摩尔定律作用下，使用该系统战胜人类只是时间问题。通常在这种类型的问题中，问题的表示通常都不是一个重要的难题，一般专业的程序开发人员也可以在很短时间内完成一个相应的专家系统[38-42]。但是许多真实世界的问题却并不都是那么容易能够用计算机语言表达清楚的，比

如视频流和图像内容的识别和语音识别，这些问题的可能性与复杂性都有着比国际象棋大得多的问题，即使对于人类自身而言来说，也有很多时候是不能够确定、无法选择的时刻，所以用数学规范与标准来描述实际问题是不现实的。

深度学习则是一种由其内部结构组成，能够进行表示学习的一种智能方式，其可以使用非线性的模块进行组成与搭建，将这些组成部分的前一级表示（从最初的输入数据开始）转化为更高层、更抽象的表示，只要构建出来的一个学习系统拥有大量这种简单的非线性模块构造，其就可以达到专家系统难以实现的目的，从而去完成丰富的学习以处理非常复杂的任务需求。

深度学习（Deep Learning，也曾被叫作 Feature Learning）是通过在人工神经网络(Aritificial neural network, ANN)基础上发展而来的一种表示学习（Representation Learning）方法，也是一种机器学习方法，而且是目前人工智能领域当中最具有发展前景的一个研究分支[43-45]，其结构关系如图 2.2 所示。

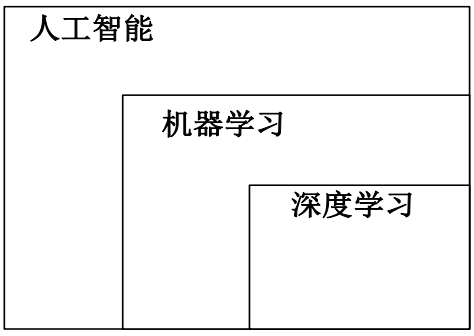


图 2.2 人工智能结构关系

其主要的模型结构是各种深度卷积神经网络（Deep Convolution Neural Network）。表示学习是近年来机器学习领域当中发展最为迅猛且最受学术界追捧的方向之一。所谓表示学习，就是通过将算法在少量人为设定的先验经验情况下，能够通过系统自身自动的从现有的数据当中提取出合适的特征数据，来完成原本需要通过数学方法进行特征提取算法才能得到的结果。对于分类问题来说，对于分类问题，高级表示可以突出显示重要的类别信息，并且还可以减少不相关的相应背景信息。例如把一幅图像以像素流的数组方式提供给网络作为数据输入，在系统网络中的输入层学习将会得到特征的边缘信息，该边沿信息是所输入图像内某个位置是否具有相关特定朝向的边缘；在网络模型内部结构中，通过卷积层将输入信息的边沿检测出来，并且能够识别出其构成的基本形状和它的排列方式，在整个过程当中并没有对输入信息的位置改变进行过多的注意；在网络内

部的输出层再将输入图案的基本图形进行组合拼接起来，从而对应出典型物体的相应位置，最后通过分类器层来判断由部件组成的图像内容信息，如图 2.3 所示。

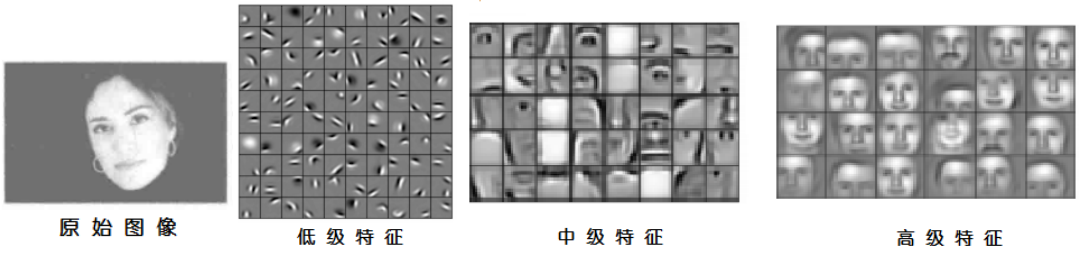


图 2.3 表示学习中间特征

深度学习中最为关键的方面是这些对输入信息内容抽取的特征层不是通过由专家而预先设置的数学规则，而是使系统网络通用学习和训练方法自动的从输入数据学习得到的。这种对实际问题的多样性而从低层次的表示到高的特征表示是目前人类无法通过数学规则和一定的规范能够预估的，完全由系统网络去判定和学习决定哪些信息特征是其内部需要的，哪些是可以进行减少的。在表示学习的范畴中，深度学习则是通过对其内部结构多层非线性变换的组合方式从而得到更为抽象也更有效的特征表示。早起所使用的机器学习解决问题的主要内容包括需要大量的研究人员去提取出数据的相关特征工程，然后在通过加上一个可以被训练的分类器来完成对输入数据的检测。在深度学习的时代，特征工程的提取已经被大量可用于训练的特征提取器所替代，并且在准确性和应用效率上有了较为显著提高。更为重要的是，人工智能的目标就是希望通过机器计算系统的能力理来解我们生活中的不同场景任务需求，机器系统需要能够自动的去感受和辨别大量未知数据背后的多种隐含因素时才能完成这个目的。

如今深度学习的应用已经在大量的场景应用上取得了突破性的进展。其中的卷积神经网络是所起的作用不容小觑[46-49]。例如 2012 年研究人员所设计的 AlexNet 网络结构在 ILSVRC 图像识别竞赛中所带来的惊人表现，让学术界看到了深度学习所蕴含的巨大潜力。在以后的几年当中，随着 GoogLeNet 网络模型结构、VGGNets 网络模型结构、ResNets 网络模型结构的提出，使得 CNNs 的识别准确率持续提高，目前已经让计算机拥有了超越人类的图像识别能力。

随着深度学习的快速发展与应用，其内部结构变得越来越大，所包含的参数成倍的增长，同时也对其实现的硬件器件提出了越来越高的需求。传统的 CPU 与 GPU 已经难

以满足数据量庞大和有实时性需求场景的深度卷积神经网络模型，为此有必要完成在其它器件上实现深度卷积神经网络已满足更多的需求，这也正是本文的研究意义所在。

2.1.2 前向传播反向传播

在深度卷积神经网络中最为普遍的训练算法就是 BP 算法，该训练算法就是通过将数据前向的输入整个深度卷积神经网络当中，经过网络内部结构中的每一层特征提取层来完成对输入数据的特征信息提取，在与预先已知的结果进行对比得出误差结果；在经过反向传播从而修正网络内部结构中的权值参数已达到整个网络模型的建立，其具体流程如图 2.4 所示。

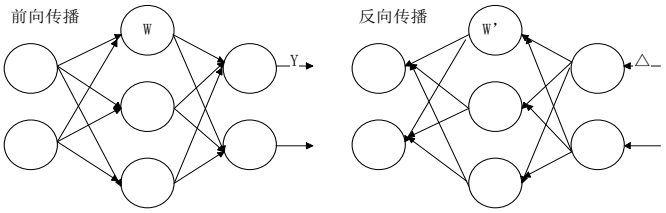


图 2.4 前向传播和反向传播

反向传播算法训练出神经网络模型的经典算法曾经在 20 世纪 70 年代到 80 年代被多次重新定义。传播算法的基本理论依据的是控制理论相关理论知识，该算法对于深度神经网络的发展至关重要，在整个网络内部的权值参数训练中必不可少。在传播训练的过程中，具体使用了复合函数知识当中的主要内容：链式法则 [50]。在输入数据固定的情况下，反向传播算法利用神经网络的输出敏感度来快速计算出神经网络中的各种超参数。

在前向传播的过程中，输入数据被指定为 X，W_{jk}^{l} 被记录为第 1 层中从第 1 层到第 1 层，第 j 个神经元到第 k 个神经元的权重，并且 b_j^{l} 是第 1 层的第 j 个神经元的偏移量。a_j^{l} 是第 1 层第 j 个神经元的激活输出值，即为激活函数的输出值，因此 a_j^{l} 的计算结果是从神经元上层的激活中获得的，可以表达为：

$$a_j^{l}=\sigma(\sum_{k} w_{jk}^{l} a_{jk}^{l-1}+b_j^{l}) \tag{2.1}$$

可以将式（2.1）重新改写为矩阵的形式有：

$$a^{l}=\sigma(w^{l} a^{l-1}+b_j^{l}) \tag{2.2}$$

为了便于表示，通常将该表达式记为 $z^{l}=w^{l}a^{l-1}+b^{l}$ 的形式，该式为每一层的权值输入，

则可将式（2.2）改写为 a^l 是=$\sigma(z^l)$。从而可以使用该式子分别计算出卷积神经网络当中每一层网络结构的激活函数值，从而最终根据输入的数据 X 计算算出与之相的输出 Y。

在反向传播中，它将会计算输出误差值对所有的参数 W 的偏微分，即如下所示：

$$x_j^l = f(\sum_{i \in M_j} x_j^{l-1} * k_{ij}^l + b_j^l) \tag{2.3}$$

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l) \tag{2.4}$$

其中 x 是卷积网络中第 l 层的第 j 个数出输出结果，k 为第 l 层的第 i 组的第 j 个特征卷积核，参数 b 是第 l 卷积层的第 j 个输出的偏移量，f(.)是每个卷积层所使用的激活函数，down(.)为卷积网络中池化层采用的降维函数。并且在神经元中所使用的激活函数其具体细节并不重要，它可以是非线性的 Sigmoid 函数或者是 RELU 函数。只需要每次可以使用梯度下降法对网络进行训练更新出新的误差，即每次沿着梯度的负方向移动设置的步长单位，不断的进行重复训练调整网络当中的权值参数，直到整个卷积神经网络的输出误差与预期值达到最小即可[52]。

但是在整个卷积神经网络的训练过程中，我们需要注意的是，反向传播算法不仅需要准确计算梯度，还需要使用一些数学优化方法来完成对整个卷积神经网络的训练。反向传播算法之所以重要，是因为它具有效率高的特点。假定对现有的一个节点求出其偏导数需要的时间作为单位时间，且其运算时间是呈线性关系的，那么整个卷积神经网络的时间复杂度可以表示为：O=O(V+E)，其中 V 为节点数、E 为连接边数。并且在标准的神经网络训练中，我们实际上关心的是训练损失函数的梯度，这仅仅是一个与网络输出有关的简单函数，但是卷积神经网络是可以增加新的输出节点的，此时我们要求的就是新的网络输出与整个卷积神经网络超参数的偏微分。而反向传播算法正是反向的(自上往下)来寻找最有路径的。它能够对于每一个路径只访问一次就能求顶点对所有下层节点的偏导数值，有效的地避开了一定冗余计算。

2.2 卷积神经网络结构及模型

神经网络领域最初主要受到生物神经系统建模目标的启发，但后来发生了变化，它为工程问题在机器学习任务中取得了良好的效果。在人类大脑构成中，神经元是它的主要构成单位。并且在人类神经系统中含有丰富的神经元，研究表明约为 860 亿个，其通过大量的突触互相组成，这些突触具有 10^14-10^15 个之多，如图 2.5 所示。

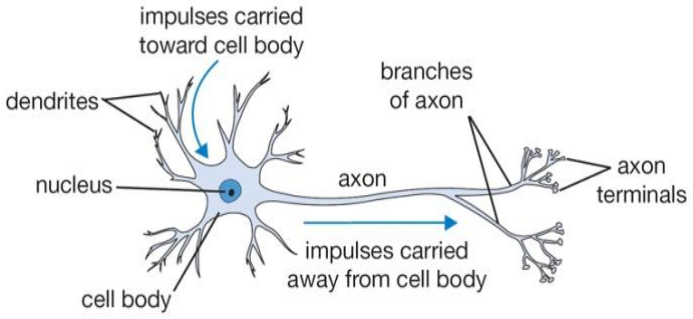


图 2.5 神经元构成

神经网络的结构设计是受到了动物视觉神经系统的启发，所以首先简单了解一下视觉神经系统。眼睛是一个成像系统，外界图像信息反射出相应的光照，光线进入瞳孔，然后映射在晶状体上，经过神经的传递，最终实现外界信息被成像在视网膜上，这一部分是由生物体的视觉所构成的一套完善的光学结构组织。在视网膜内布存在有丰富的细胞，这些细胞能够将外界接收的光线激励用于内部，使相应的突触造成神经冲动，从这些细胞开始，就进入了视觉神经系统[51]。神经冲动通过视觉通路（视觉神经→视觉交叉→视觉神经→视觉外侧膝状体→视觉的放射）传递到大脑的初级视觉皮层。从视觉皮层开始，对图像信息的处理也是分层的，这些层分别是从 Vl 到 V5（V 代表 Visual）[52]。每一层处理一部分特定的信息，比如 Vl 主要对一些边缘纹理响应，V4 则对一些简单的形状响应，而信号传递的大概顺序也是从 Vl→V5，需要注意的是这种大致顺序并不是一层层的简单连接，比如 V2 和其他所有层都有连接。

上面只是简要的介绍了一下神经科学中的分层结构，真正在对现在的计算机视觉尤其是对卷积神经网络启发很大的发现是加拿大科学家大卫休伯尔（David Hubel）瑞典科学家托斯坦•维厄瑟尔西（Torsten Wiesel）从 1958 年起对猫视觉皮层的研究。他们在 Vl 皮层里发现了两类特有的细胞，即简单细胞（Simple Cells）和复杂细胞（Complex Cells），每种细胞都具有这样的特征：每个细胞仅响应特定方向上所产生生成的条形激励，也就是说，这些细胞是有方向性选择的。这两种细胞的主要区别是简单细胞对应的视网膜上的光感受细胞所在的区域很小，而复杂细胞则对应更大的区域，这个区域被称作感受野（Receptive Field）。模拟这种概念的结构早在 1980 年计算机界已经有人提出来了，是日本学者 Kunihiko Fukushima 提出的。现在的卷积神经网络结构深受其的影响，简单来说就是卷积层用来模拟对特定图案的响应，而池化层用来模拟感受视野。

2.2.1 卷积神经网络的结构

卷积神经网络的结构与神经网络非常相似：它们都是由具有可以学习的权重和偏差的神经元组成。每个简单的神经元通过接收一些外部的数据输入，并且相应的执行点积运算再可选地以非线性跟随数据。整个网络仍然可以展示单个可区分的分数函数：从一端的原始图像像素数据到另一端的类分数的输入，并且它们仍然在最后一个（完全连接的）层上传递函数分类器（例如 SVM 和 Softmax 函数）完成后，其基本组成如图 2.6 所示。

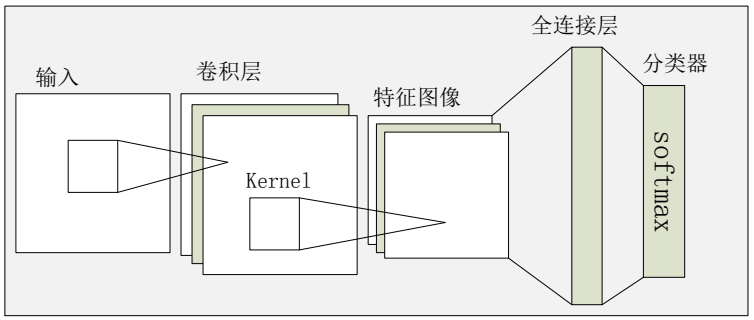


图 2.6 卷积神经网络结构

由结构图能够得出每个卷积神经网络模型的基本结构可以分为：卷积层，池化层与激活层。通过该结构，外部的输入数据首先经过卷积层得到响应，然后经过下一层的激活层完成对数据的非线性变换处理，而卷积本身也可以看成是一种线性变换，再考虑到有偏置项的存在，使得这种放射变换非线性变换的操作实际上是一般神经网络中基本操作的特例。卷积神经网络是一种特殊的深度神经网络模型结构。其特殊性主要体现在两个方面，一方面它在神经元之间的连接没有进行完全连接，另一方面共同构成出同一层中的一些神经元，它们之间的连接权重是共享的（即相同）。其非完全连接和权重共享网络结构使其更类似于生物神经网络，降低了网络模型的复杂性（适用于大规模深度卷积神经网络结构而言是非常重要的），减少了权值的数量。

卷积神经网络在卷积层中的主要工作任务是进行卷积神经网络结构中复杂且大量的卷积计算。在卷积层中，每个卷积层是通过多组可以进行学习的过滤器模板组成，每个过滤器模板在空间上都拥有较小的宽度和高度。典型过滤器模板可能会具有 5x5x3 的大小（即 5 像素宽度和高度，拥有图像所具有的 3 个颜色通道）。在前向传递过程中，在输入图像数据的宽度和高度上进行滑动（即卷积运算）每个滤波器模板，从而来计算

出滤波器模板与输入图像的任何位置之间的点积。当在输入图像数据的宽度和高度上滑动模板后，将会生成一个当前输入图像数据的激活特征图，该特征能够在输入图像的每个空间位置给出当前对应滤波器模板的响应。直观地说，卷积神经网络将会学习到当它们看到某种类型的视觉特征时从而去激活的相对应的过滤器模板产生出特征图，例如某些方向的边缘信息、某种颜色块或者在卷积神经网络较高层上的整个矩形及轮状图案，该过程如图 2.7 所示。

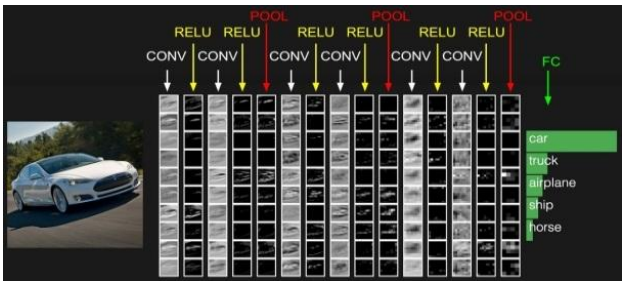


图 2.7 输入图像特征图提取

在卷积神经网络中的池化层对卷积层所提取出的特征图后虽然对特征图的尺寸及维度进行了减少，但还是能够将特征图的大量信息进行保存。常见的池化方法有最大化法、平均化池化法、加和池化法等。对于最大池化法而言（Max Pooling），将会定义一个空间邻域（例如 2x2 的过滤器模板），并从过滤器模板内的待修正特征映射中提取出最大的元素。除了对获取最大元素进行之外，我们还可以使用平均池化方法（Average Pooling）来完成对于过滤器窗口中每个数据的计算。在具体的使用当中，不同的池化方法会有不同的效果，但是综合而言最大池化方法效果最好，它的基本池化过程如图 2.8 所示。

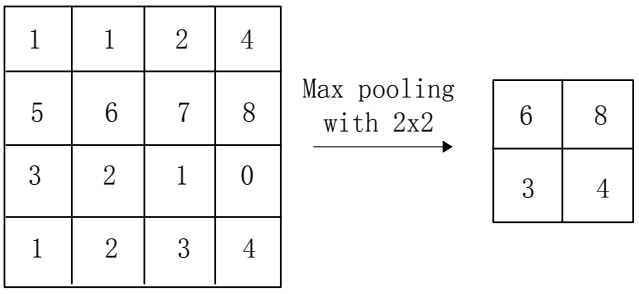


图 2.8 最大池化法

在卷积神经网络的激活层中会经常使用非线性函数来完成，常见的函数包括 Sigmoid 函数、ReLu 函数等。其中 Sigmoid 函数能够将特征数值归一化至 0 到 1 之间的范围内，并且能够取得不错的激活效果。ReLu 函数能够在特征数值小于 0 的时候将其

置零，而在输入数值大于 0 时使之呈斜率为 1，大大的提高了收敛性。

2.2.2 卷积神经网络并行性

通过对卷积神经网络的简要介绍，可以看出在卷积神经网络的结构中主要组成部分就是其中的卷积层，并且每一层间的计算是互不相关的，这为卷积神经网络的并行运算提供了可能。

首先由于每个过滤器模板对输入图像的不同特征进行提取是互不相关的，因此当卷积神经网络在进行计算时首先可以将卷积层中的过滤器模板进行并行处理，从而达到一定的加速效果；然后卷积神经网络结构中后一层级之间的运算是根据上一层过滤器模板所提取出来的，在卷积神经网络系统运行的周期内则可以进行不同层间的并行流水线处理，以达到卷积神经网络的高效利用率。

2.2.3 常见卷积神经网络模型

AlexNet 卷积神经网络模型结构的提出具有深远的意义，被学术界普遍认为是一个经典的模型结构，它的设计思路与优化方法对于其它的卷积神经网络模型结构至关重要，无论是后来 2013 年的冠军结构 ZF-Net，还是 VGGNet。AlexNet 针对的是 ILSVRC 的分类问题，输入图片是 256x256 的三通道彩色图片。为了增强其泛化能力，训练的时候 AlexNet 采用的数据增加手段包含了随机位置裁剪，卷积运算的时候还采用了分组的方法。

VGGNet 卷积神经网络结构也是常见的模型，该网络从网络结构设计上来说基本是延续了 AlexNet 的设计理念，它通过 AlexNet 构造了一个具有更多层级和更深的结构。其具体的网络结构是通过 8 个网络层次所构成，包括了 5 组卷积层与 3 层全连接层。

LeNet-5 是成功的一种卷积神经网络结构模型，上个世纪被用于商业应用当中。该网络结构非常简单，仅包括了 7 个网络层，但却对打印字符的智能识别起到了重要作用。

2.3 硬件加速技术

2.3.1 硬件加速平台简介

随着深度卷积神经网络的规模越来越大，计算量的提升，产生了众多的硬件加速平台，其中主要包括了基于专用集成电路（Application Specific Integrated Circuit，ASIC）平台、图形图像处理器（Graphics Processing Unit, GPU）平台以及现场可编程阵列（Field Programmable Gate Array, FPGA）平台来实现深度卷积神经网络。虽然 ASIC 专用芯片平台具有强大的数据运算效能，但是开发出一款这样的芯片需要耗费大量的时间积累，并且专一性强、研发成本高，如果改变整个网络结构则需要完全重新设计实现；GPU 芯片平台具有高度的灵活性及可编程性，能够应对不同的卷积神经网络结构，但是其功耗高，对于低功耗的场合下难以处理；而 FPGA 芯片平台则具备了可定制性强、自由度大以及能耗低的特点，使其成为了卷积神经网络硬件加速平台的理想选择。

2.3.2 FPGA 芯片结构及特点

FPGA 可编程芯片其内部系统架构了使用的是模块化架构设计，它是通过使用高级硅模组合模块(ASMBL)架构将功能模块分布成可互换的列，在这种结构中，FPGA 器件只需要更换成列状的 IP 模块，从而可以加快针对特定应用开发出能满足其独特需求的产品。在物理制造层上，其内部的 ASMBL 模块被设计为具有相同宽度的模块，并且将其垂直进行排列组合，生产出的高度与芯片封装相同。每列可能包含可编程逻辑结构或硬 IP，如存储器、DSP 模块、高速 I/O、串行/解串行模块、通用可编程 I/O、微处理器内核或其它任何能够被布置在这种纵横比(指高与宽之比)空间中的 IP。并且使用 ASMBL 模块架构的 FPGA 芯片还采用了九层的金属工艺，在九个金属层中，有 1 至 4 金属层专门用于列中的电路，但上面的金属层是供 FPGA 的分段式可编程互连结构使用的。I/O、电源和时钟连接不是置于裸片的外围，而是被制作在列内的基板栅格阵列上，通过倒装芯片技术绑定到封装上。列电路在被创建的同时将走向设计收敛，而且这与它们在特定芯片内的布局无关。

因此 FPGA 芯片具有非常高度的自由定制化，能够根据实际的应用场景开发出适合的电路设计，由于内部架构的设计，将常用的电路模块 IP 化，从而大大减少了开发周期，节约了成本，且内部的电路按照一定规则排列有效的减少了能耗与体积，使其具也备了微型化和小型化的特点。

2.3.3 硬件加速优化方法

通过 FPGA 内部结构，可以看出其本身计算单元是电路，而电路之间的运算是独立进行的，因此首先可以在进行数据计算时对其进行并行化处理，使用软件对算法进行并行化分析，把能够同步进行的数据计算使用不同的电路模块并行进行运算处理，从而达到加速的目的；其次是对数据流进行流水线处理，通过将电路进行逻辑组合设计出合理的流水线结构加速数据的运算。

在卷积神经网络中，每个滤波器模板的权值虽然是独立互不相关的，但是依然会存在相同的权值参数，因此可以通过将同样的参数值使用单个寄存器进行存储，以达到减少外部存储器对于过滤器模板权值参数的访问、降低硬件资源的消耗和缩短时序路径的目的。

2.4 本文卷积神经网络硬件加速系统设计

文章经过对相关理论知识研究，使用 ZYNQ-7000 系列 FPGA 芯片，以 VGG 卷积神经网络模型为基础和 MINIST 手写数据集为验证数据集，对其卷积神经网络模型进行了修改与裁剪，设计出了一种 12 层的硬化卷积神经网络模型。

为了能够满足对于实时识别任务场景的需求，同时还设计出了一种基于 ZYNQ 的实时识别硬件系统框架，该系统框架为本文设计出的硬化卷积神经网络提供了符合 AXI4 总线数据流的高速数据通信接口，在实际应用中能够通过使用 Vivado 开发设计工具将符合实时识别框架接口的硬化卷积神经网络进行接入，通过工具综合后生成对应电路并部署至器件当中运行以满足具体的实时识别任务需求。本文硬件加速系统整体示意图如图 2.9 所示。

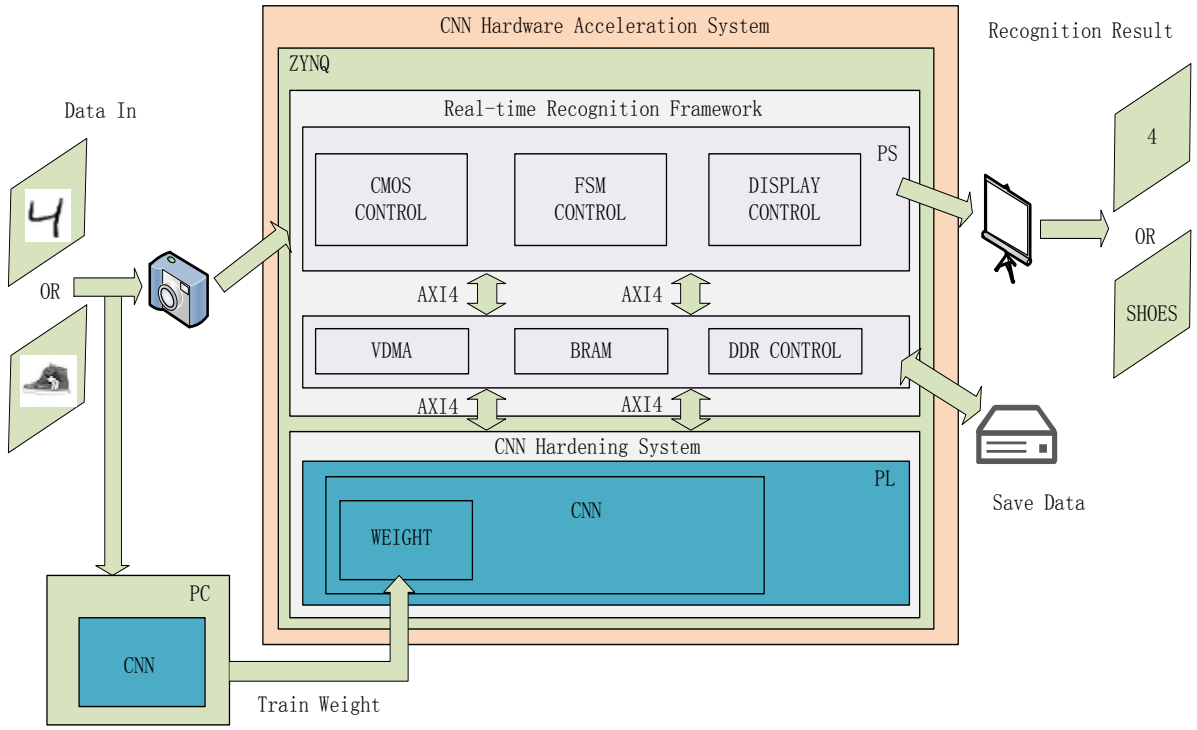


图 2.9 本文硬件加速系统整体示意图

在卷积神经网络硬件实现研究中，本文主要以卷积神经网络的结构为基础，针对手写数据集内容的识别，研究输入图像数据缓存、重构的硬件模块实现来完成对图像数据的存储；设计了并行优化的滤波器模板完成对存储图像数据的卷积运算操作从而实现对映射特征图信息的提取；硬件实现函数分类器模块以完成对图像内容的识别，从而使设计出的硬化卷积神经网络能够对运算进行硬件加速。同时对整个硬化卷积神经网络模型设计了卷积运算和数据存储状态机控制器，确保了卷积网络运算中的数据流准确；为了能够在不同识别场景任务中应用本文设计实现的硬化卷积神经网络，在硬件设计实现中将权值存储模块进行了独立的硬件模块实现，从而对硬化的卷积神经网络模型进行替换不同的权值参数以完成对多场景应用的识别任务。

在实时识别系统框架研究中，本文对输入图像数据的实时采集与存储模块进行设计，使用 CMOS OV7670 摄像模块和高速 VDMA 控制器相结合完成对实时图像数据的采集与存储；为了针对多分辨率图像的显示，使用了 VTC 视频时序控制模块，并且设计了一种降采样模块来完成对视频存储数据多分辨率显示的实现；设计了与硬化卷积神经网络模块的高速数据传输接口 AXI4 总线接口，从而可以使用设计出的实时识别系统框架，对硬化卷积神经网络模型的性能与功能可以进行快速的验证和部署实现，以大大减少开

发所需周期，并且该系统还具有可移植性高、消耗能源低的特点。

经过对以上两个内容的设计研究，最终实现本文的卷积神经网络硬件加速系统设计。

2.5 本章小结

本章主要简单的介绍了深度学习的基本概念与理论基础，介绍了深度学习中重要的反向传播算法与原理，并给出了相关计算公式；对卷积神经网络的结构进行了详细说明，对卷积神经网络当中每层的运算和其结构的并行特性进行了详细分析，并简要的介绍了几种常见的卷积神经网络模型；然后分析了多平台的硬件加速技术，对比了不同平台之间的优缺点，突出了 FPGA 芯片平台的优势和特点，对 FPGA 芯片器件平台实现卷积神经网络的可行性做出了分析，同时还简要介绍了常用的硬件加速优化方法；最后对本文所研究的卷积神经网络硬件加速系统进行了阐述。

第三章　卷积神经网络硬件结构设计

前面章节对深度学习和卷积神经网络的基本结构进行了详尽的介绍。本章将会根据上一章节所介绍的卷积神经网络结构，对卷积神经网络的各模块功能进行硬件设计实现。本章的主要内容包括了卷积神经网络结构的硬件整体结构设计，对本文模型结构设计思路进行了详细说明。在网络的内部结构中分别实现了卷积硬件、池化硬件层、激活函数硬件、权值存储结构硬件、分类器硬件的结构设计。此外，在整体的硬件结构设计中，利用 FPGA 硬件电路的特点，完成了对卷积运算层的并行优化，根据整体系统结构设计做了流水线的结构处理，从而提升了模型的运算速度，加速了系统的计算效率。

3.1 本文卷积神经网络模型

本文的卷积神经网络模型结构以经典的 VGG 卷积神经网络模型结构为基础，对其进行了一定的精简及修改，从而所设计出来的卷积神经网络模型结构组成如下图 3.1 所示。

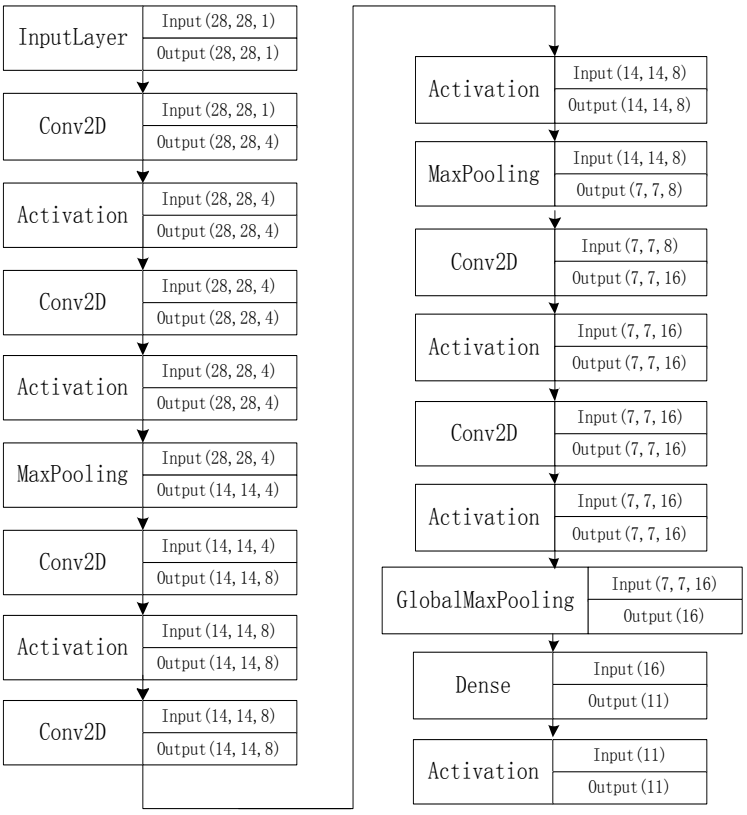


图 3.1 本文所设计出的卷积神经网络结构模型组成

由于在原本 VGG 模型结构中具有大量的权值参数，需要大量的存储空间，而本文所采用的 ZYNQ 7000 系列芯片的存储空间有限，因此对 VGG 模型进行了裁剪来降低权值参数的数量，而且删除了大量的全连接层和偏置项，从而便于本文最终使用 MNIST 手写数据集来进行网络模型的功能验证。

3.2 卷积神经网络硬件设计思路

卷积神经网络的数据流输入是按照一定方向进行的，因此可以根据这一特点完成其内部每一层的硬件结构设计，通过结构图可以看出该卷积神经网络模型结构总共由 12 层构成，包括了一个输入层，一个输出层。

本文最终所验证的数据集采用的是 MINIST 手写数据集，因此按照精简后的 VGG 网络的结构对该网络结构中的输入层构建为 28x28 的输入窗口，所以网络中总共输入的数据是 784 个。在输出层与其进行连接的权值个数是 16x11=176 个，最终的识别结果共有 11 种，其中一种是输入数据无效的结果。

在网络内部结构的第 2、3、5、6、8、9 层是该网络的卷积层。经过滤波器模板进行输出激活特征图像的个数分别为 4、4、8、8、16、16 个，将每个滤波器模板的大小层卷积核大小设置为 3x3，滑动距离均为 1，并且没有改变输出特征图的尺寸大小，与输入图像保持为一致。每一层的参数分别为：第 2 层：权重为 3x3x4=36 个；第 3 层：权重为 3x3x4x4=144 个；第 5 层：权重为 3x3x4x8=288 个；第 6 层：权重为 3x3x8x8=576 个；第 8 层：权重为 3x3x8x16=1152 个；第 9 层：权重为 3x3x16x16=2304 个，所以在本文设计的卷积神经网络模型结构中卷积层总共有拥有 4500 个参数，基本满足本文所使用的 ZYNQ 7000 系列芯片的存储容量。

在卷积神经网络模型内部结构中的第 4，7，10 层均是池化层。由于该层是对上一层的输出特征图进行降维运算处理，因此这里本文也采用了卷积运算，且滤波器模板的尺寸为 2x2，进行了最大池化方法。

3.3 卷积神经网络硬件结构设计

3.3.1 卷积层硬件设计

由于输入的数据为图像数据，而图像数据在 FPGA 内部一般的是通过串行方式进行数据交互，因此要实现卷积运算的硬件实现，必须对其输入的数据根据每个卷积层的滤波器模板的尺寸大小来进行与原始输入图像数据相对应大小的进行匹配。本文在整个卷积神经网络系统结构设计中是通过使用了移位寄存器（Shift Register）来完成对输入图像的缓存数据处理，移位寄存器还能够将输入图像的数据在进行数据存储的同时进行移位操作处理，该模块能够根据所设定的大小来完成对输入图像数据进行尺寸匹配处理操作，当需要数据并行的且单行进行输出时，仅在其末端增添相应的抽头信号即可将图像数据进行输出操作，通过使用移位寄存器能够使得卷积神经网络运行后在 1 个时钟周期内完成 1 次卷积运算，其具体结构如图 3.2 所示。

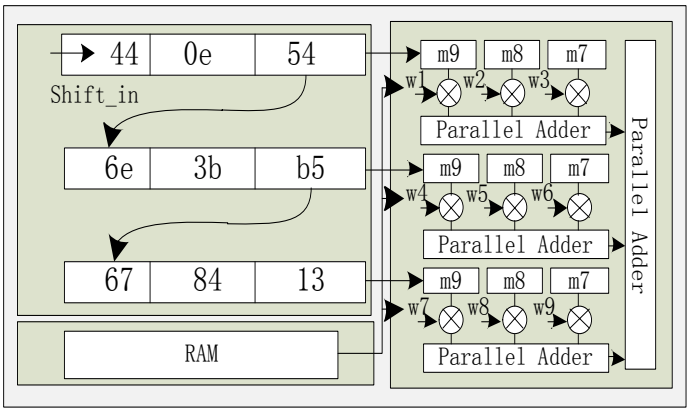


图 3.2 卷积层硬件结构

该卷积核层硬件结构是过滤器模板的尺寸为 3x3 时的运算过程，其中 Shift_in 信号是当前的数据图像输入，权值参数 w_i 为当前过滤器模板的权值，m_i 为经过移位寄存器的输出图像像素值得缓存寄存器。

在卷积层运算开始时，三块输入缓存依次载入输入特征图的第 1、2、3 行像素值，然后每周期每行依次输入一个像素值，相当于卷积窗口向右滑动，对于每行来说，该像素数值会同时的输入到三个乘法器与三个权值相乘，达到了像素复用的效果，经过三个周期后，会得到单个卷积窗口的部分和，并与上一输入特征图对应的部分和累加，产生一个新的部分和。当输入特征图的第 1、2、3 行的卷积运算完成后，输入缓存再依次载入第 2、3、4 行的数据进行运算，以此类推。

利用这样的硬件设计，使得系统能够进行流水线的数据流计算，该运算单元每个周期都会输出一个有效的卷积部分和。同时我们可以重复生成这样的运算单元结构，实现

不同输入特征图或不同输出特征图的并行。

通过使用硬件描述语言根据卷积层的结构特点进行描述建模，最终生成的卷积层模块信号如表 3.1 所示，生成的卷积层 RTL 如图 3.3 所示。

表 3.1 卷积层模块接口定义

信号名	说明
CLK	卷积层模块工作时钟信号
RST_N	卷积层模块复位信号
input_weight_en	权值使能
weight_ab	单时刻写入的权值
write_done_weight_bias	权值写完的信号
input_fmap_en	输入特征数据使能信号
output_en	输出特征数据使能信号

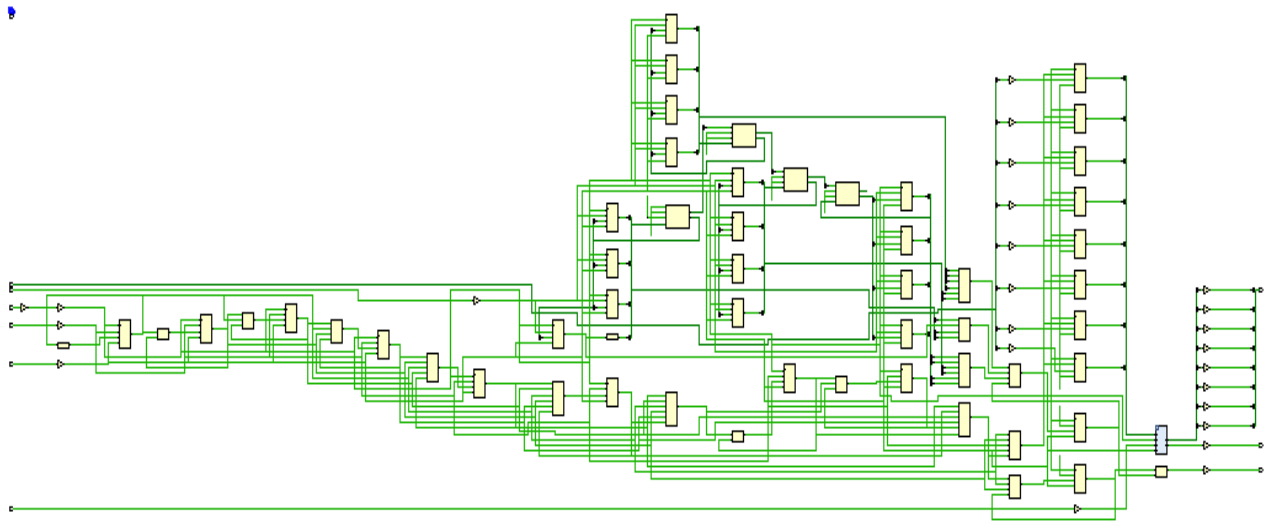


图 3.3 卷积层硬件 RTL

3.3.2 池化层硬件设计

对于文章中的卷积神经网络硬件结构设计而言，可以将整个硬件实现设计为在一个硬件模块当中，但是为避免耦合的产生，并且为以后设计的可延展性，本文将整个硬件设计分为了三大模块：顶层，存储层，以及池化层。

1）对于顶层，可实现简单的调度逻辑，并为以后的延展留出余量。在本模块的设计时，重点分离了时序逻辑以及组合逻辑的关系，避免了数据的混淆。且由于最大池化层可能会存在与存储模块时钟不匹配的问题，所以在池化层硬件实现时，本文对其时钟域进行重点的考虑，避免了时序间的数据冲突。

2）对于存储模块，本文采用了外部存储 DDR 和片内存储器 BRAM 存储模块，对于待处理图像进行缓存采用了片内的 BRAM 模块，由于不同模块间的数据交互将会产生不同的数据路径延迟，因此还匹配了相应的工作时钟。

3）本文所设计的池化层硬件模块，首先是将图像数据进行传入，根据系统需要池化的最小分割图来判断传入数据的多少，以 3x3 为例，首先传入前三位像素点，进行第一次比较，较大者缓存入 buffer，之后再传入三次数据，依次比较，得出这个 3x3 方格内最大的数。本次池化操作进行完毕后，系统将池化后的结果放入新的地址，之后使每个过滤器模板以 1 的步长进行移动，来完成对新构成的 3x3 模板进行池化运算，以此类推，直到图像池化完毕，从而构成新的输出特征图，其具体数据交互过程如图 3.4 所示。

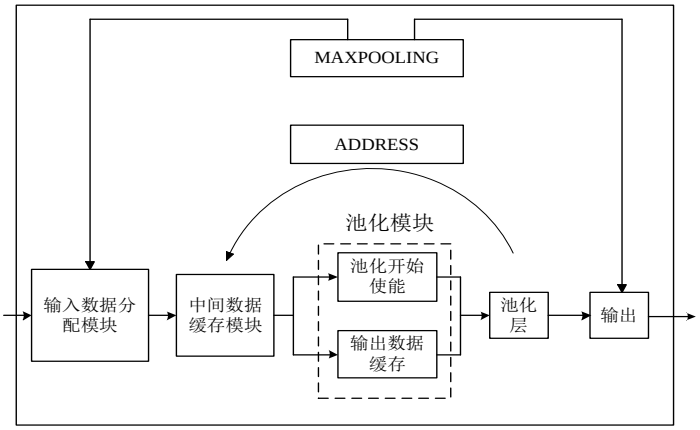


图 3.4 池化层与整体系统数据交互

本文系统设计通过使用硬件描述语言根据池化层结构特点进行描述建模，最终生成的池化层模块信号如表 3.2 所示，生成的池化层 RTL 如图 3.5 所示。

表 3.2 池化层模块接口定义

信号名	说明
CLK	池化层模块工作时钟信号
RST_N	池化层模块复位信号

续表 3.2 池化层模块接口定义

data_vaild	输入数据使能
data	单时钟的输入数据
data_done	池化完成信号
mem	池化层模板数据信号
max_en	池化层模模板使能信号

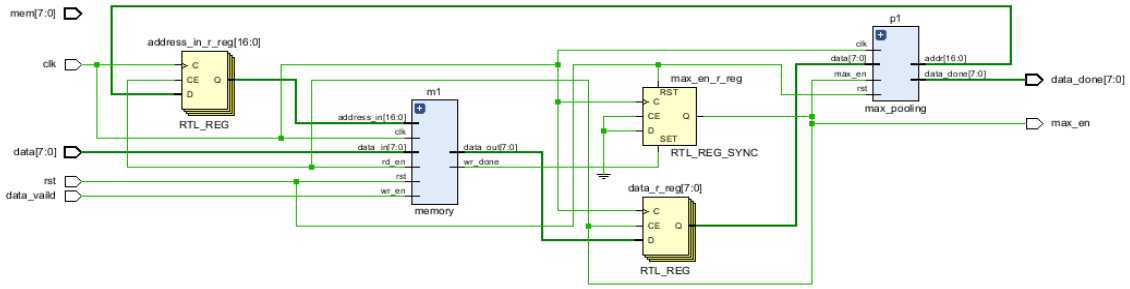


图 3.5 池化层硬件 RTL

3.3.3 激活函数硬件设计

本文使用了便于硬件实现的 ReLu 函数作为特征输出图的激活函数，该函数的实现较为简单，因为当其输入值小于零时则函数输出零，当其输入值大于零时则输出其本身值，因此在具体的 FPGA 实现中使用了比较器来进行实现，在系统中的结构如图 3.6 所示。

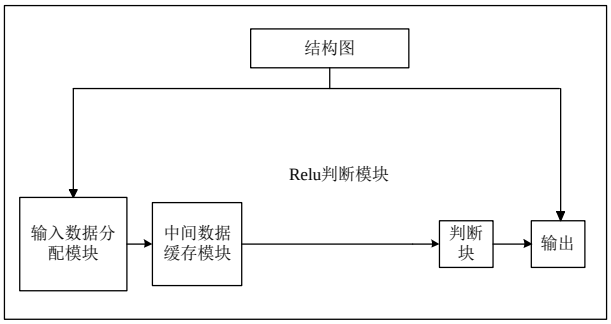


图 3.6 激活函数层硬件与整体系统数据交互

本文系统设计通过使用硬件描述语言根据激活函数结构特点进行描述建模，最终生成的激活层模块信号如表 3.3 所示，生成的激活函数 RTL 如图 3.7 所示。

表 3.3 激活函数模块接口定义

信号名	说明
CLK	激活函数模块工作时钟信号
RST_N	激活函数模块复位信号
data	单时钟输入数据
data_done_w	激活函数运算完成信号

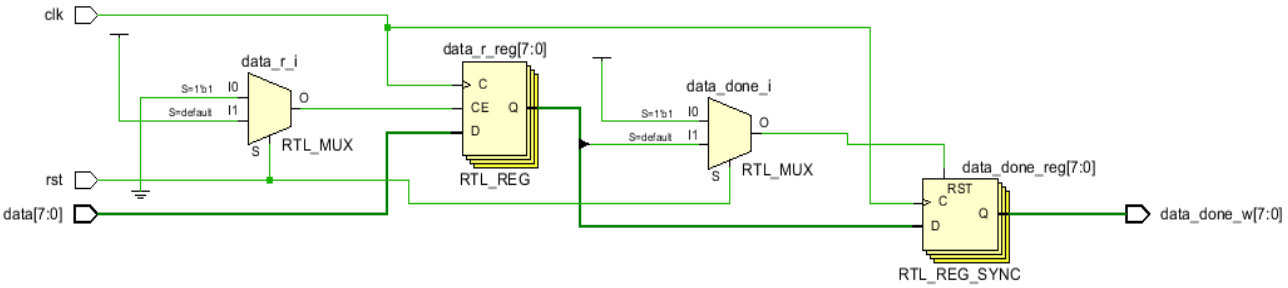


图 3.7 激活函数层硬件 RTL

3.3.4 函数分类器硬件设计

在本文设计的卷积神经网络模型结构中，输入图像数据会通过卷积神经网络模型内部中的每一层卷积层进行运算，从中提取出当前图像的激活特征图，然后这些特征图被池化层完成降维处理后再和卷积网络模型中的输出层进行全连接，从而识别输入图像的具体分类结果，则必须要实现卷积网络模型中的最后一层概率函数分类器硬件的实现。在本文系统设计中的分类器所采用的是 Softmax 函数来将特征值进行分类并且完成最后结果的输出，但是由于 Softmax 函数连续的曲线，并且是把所有输入的特征值数据通过 e 指数来完成运算从而得出输出输入特征值的概率分布，并且该概率分布的数值范围是从数值 0 至数值 1 之间的小数形式，但是 FPGA 其本身是不利于完成小数浮点数的数

值运算，并且要进行 e 指数的浮点数运算是需要消耗大量的时间。因此在本文的分类器硬件实现中系统设计采用的是查表法，通过地址寻址器来访问查找表从而完成 e 指数的浮点运算结果。具体是利用已经将计算后的指数运算结果存储至片内的 BRAM 当中，然后将输入图像数据的特征值来当作地址以便查找出所对应的指数运算结果。

本文的系统设计中通过计算机软件将处理运算后的输入特征值计算数值结果进行处理，通过将其数值范围缩小至-28 至 38 之间并进行取整处理，由于该处理方法会在一定程度上对特征值的比重进行增大或者消减，但其降低了 FPGA 芯片内部所存储的查表需要存储具体数值 ROM 的存储容量，并且还减少了一定的硬件资源消耗。最终经过硬件设计后实现了 Softmax 函数分类器，其结构图如图 3.8 所示。

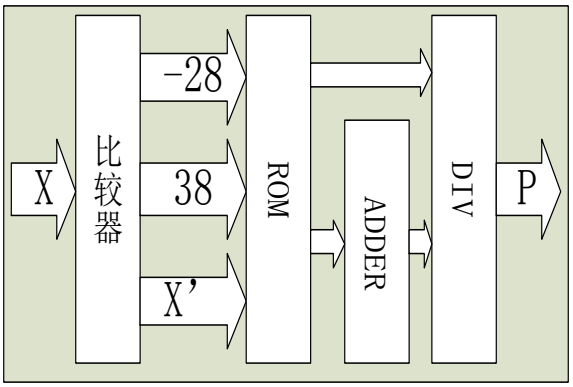


图 3.8 分类器函数硬件实现电路结构图

本文系统设计通过使用硬件描述语言根据分类器模块结构特点进行描述建模，最终生成的分类器模块信号如表 3.4 所示。

表 3.4 分类器模块接口定义

信号名	说明
CLK	分类器模块工作时钟信号
RST_N	分类器模块复位信号
data	单时钟输入数据
Address_i	地址读取信号
Address_i_done	地址读取完成信号
done	计算完成信号
Softmax_busy	计算忙信号

3.3.5 权值存储硬件设计

本文所设计的卷积神经网络结构中总共包含有 4500 个权值参数，需要对其进行一定规则的存储才能够确保整个系统的卷积运算。因此本文利用状态机设计了状态控制器和地址控制器来完成对卷积神经网络当中权值参数进行读取，使权值参数能够按照一定的规则参与卷积运算。

该状态机首先上电后保持为空闲态，当收到每层卷积运算开始信号时，跳转至特征计算状态进行特征图运算输出，在每层的特征图运算完成后，进行降采样处理即进入卷积运算状态；并且每次进行特征运算和卷积运算的时候都会进行地址搜索，以确保与之对应的权值被正确读取，该状态机的状态转移图如图 3.9 所示。

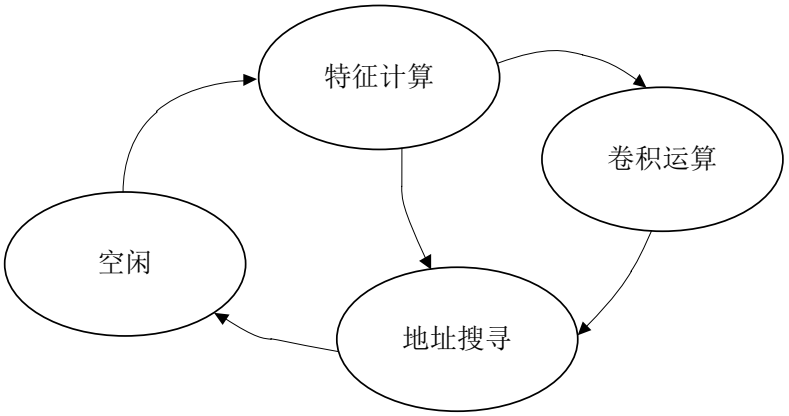


图 3.9 状态机状态转换图

本文系统设计通过使用硬件描述语言根据权值存储模块结构特点进行描述建模，最终生成的权值存储模块信号如表 3.5 所示。

表 3.5 权值存储模块接口定义

信号名	说明
CLK	权值存储模块工作时钟信号
RST_N	权值存储模块复位信号
Data_address	数据地址信号
Weight_address	权值地址信号
State_i	状态机状态信号
Data	输入数据信号

续表 3.5 权值存储模块接口定义

weight	输入权值信号
Data_busy	数据等待信号
done	初始化完成信号

至此完成了卷积神经网络结构中的所有模块设计，然后将各个模块进行组合，加入本文设计的状态控制器和地址控制器模块，最终本文所设计的卷积神经网络硬件整体系统结构如图 3.10 所示。

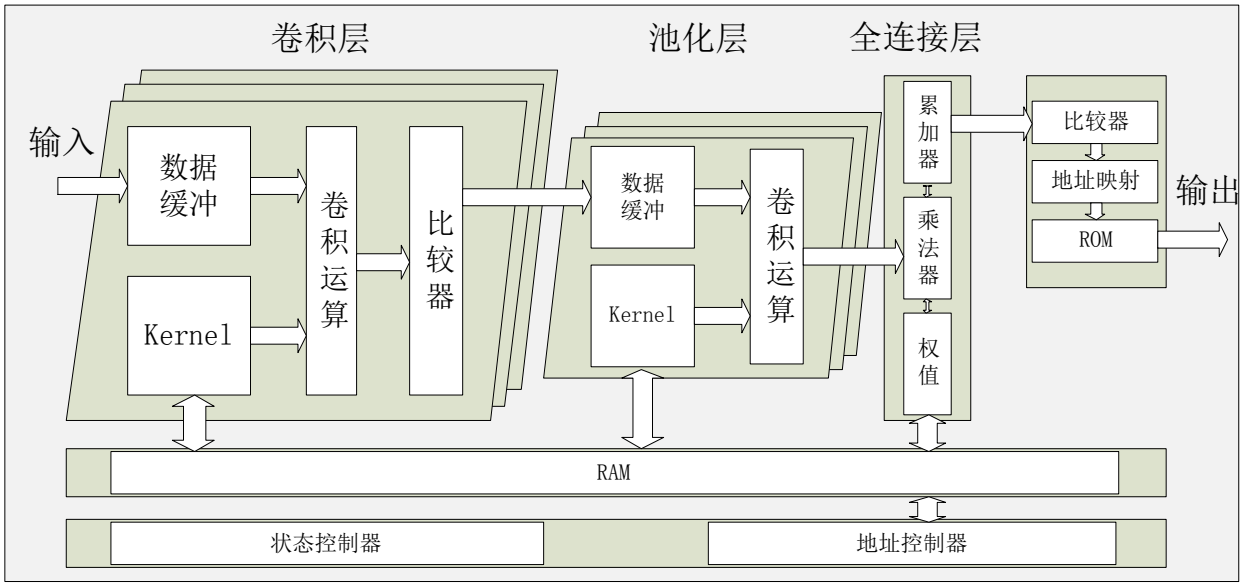


图 3.10 系统整体硬件结构

3.4 硬件优化设计

因为卷积神经网络模型结构中的每一个模块的运算是不相关的并且相互独立，充分展现出其组成结构能够来实现并行运算的特点，特别是当在进行卷积运算的时候，在每一层的卷积层进行卷积运算中，其乘法和加法运算都是独立的，因此是可以并行执行。

通常情况下，利用 FPGA 实现的计算单元都会引入流水线结构，使其能在一个时钟周期内完成卷积核大小的乘累加操作。如图 3.11 的结构来实现卷积窗口内部的并行，每一个时钟周期完成 9 次乘法和 8 次加法。

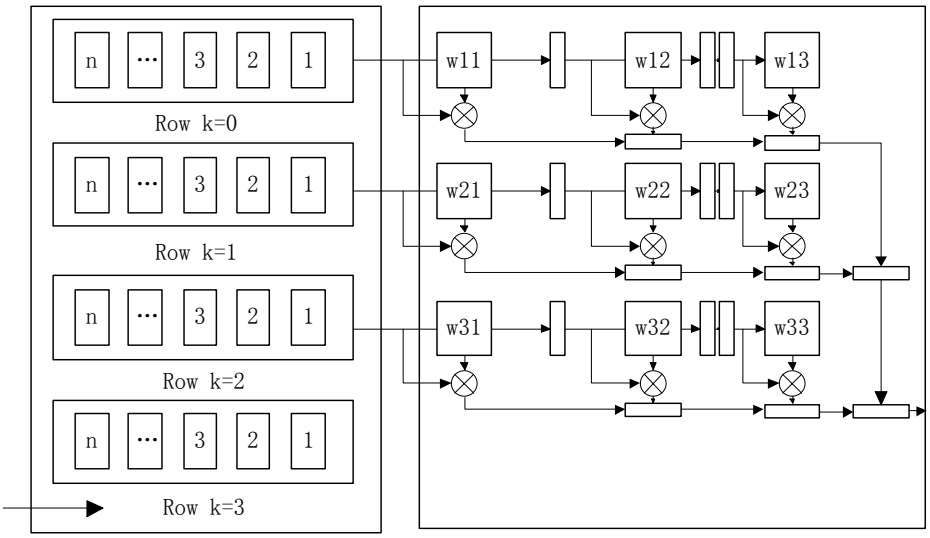


图 3.11 卷积窗口并行运算硬件优化结构图

图中 W_{mn} 代表卷积核中第 m 行第 n 列的权值，图中左侧区域为输入图像像素存储区域，每个时钟周期依次向计算区域输入一个像素值，三行所得结果相加就是一个卷积窗口的部分和。当一行的像素值全部输入到计算区域后，所有行向上移动一行，进行下一次计算。以第一行的计算为例，假设 X_{kn} 表示第 k 行第 n 个像素，在第一个时钟周期，计算 X_{01}•W_{11}，然后将结果流水至下一级，后两个乘法器计算得到的结果为无效值；当在下一个时钟运算的时候，计算 X_0W_{12}，第一个乘法单元同时计算 X_{02}•W_{11}，加法器计算 X_{01}•W_{11}+ X_{02}•W_{12}，第三个乘法单元计算得到的为无效值；再下一个运算周期时，计算 X_{03}•W_{13}，同时加法器得到 X_{01}•W_{11}+ X_{02}•W_{12}+ X_{03}•W_{13} 的结果，第一个乘法单元计算 X_{03}•W_{11}，第二个加法单元得到 X_{02}•W_{11}+ X_{03}•W_{12}。因此，该计算单元在开始工作后需要经过三个时钟周期才能得到一个正确的输出特征图像素，而在得到第一个正确的像素后，每过一个时钟周期能达到一个正确的像素，达到流水计算的目的。

在不考虑相同输入特征图卷积窗口并行的情况下，卷积窗口在完成一次运算后需要向右或向下移动。假设 FPGA 上的计算资源足够多，能够满足卷积窗口覆盖整幅输入特征图的需求，那么在不更新卷积核的情况下，FPGA 可以在三个时钟周期后得到与该卷积核对应的输出特征图。然而 FPGA 的片上资源和带宽是有限的，要在片上计算资源和带宽的限制下减少计算一幅图所用的时间，就需要考虑如何合理复用输入特征图中的像素。

在相同输入特征图卷积窗口并行方式下，卷积核在完成整幅输入特征图的计算前不

需要重新载入，假设卷积核大小为 3x3,在第一个时钟周期载入输入特征图第一行像素，在第下一个时钟周期加载输入特征图第二行像素值，最后一个时钟周期加载完成输入特征图第三行像素值。然后各个计算单元根据片上存储单元的地址读取相应的像素数据，再与卷积核中的权值进行卷积运算，然后将得到的部分和送回片上存储单元中，当 3 行像素全部计算完毕后再从片外存储单元读取一行新的像素数据，直至将输入特征图的最后一行数据读入片内存储单元。然而，卷积窗口的数量并不一定每次都能被输入图像的大小整除，在不能够被整除的情况下，就一定会出现计算单元空置的情况，此时就需要考虑跨性调度的并行方式以充分利用 FPGA 上的计算资源。所谓跨行调度就是在计算过程中允许卷积窗口处于输入特征图的不同行，这样的方式比卷积窗口向下移动拥有更好的资源利用率。使用相同输入特征图内多卷积窗口并行的方式能够充分复用载入至片上存储单元的输入特征图数据，减少访问片外存储的次数；这种方式一次需要更新的数据与重新载入一幅完整的输入特征图相比也不多，消耗很小的带宽；同时计算单元是通过寻址的方式读取片上的数据，访问方式简单，容易控制，不易出错。在硬件资源充足的条件下，相同特征图内多卷积窗口并行的方式能够与其他卷积计算并行方式共同使用，进一步减少完成一层卷积运算所需的时间，其具体实现结构如图 3.12 所示。

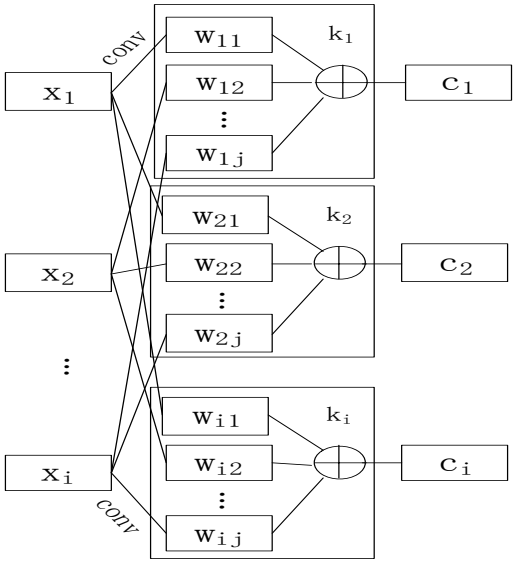


图 3.12 不同过滤器模板并行运算硬件优化结构图

在本卷积神经网络系统结构中也采用了这样的优化方式，使得本文系统能够在一个时钟周期内能够进行 528 次的卷积运算，从而大大的提高了卷积神经网络的运行效率，达到了加速运算的目的。

同时本文还完成了对输出特征图的并行优化硬件设计以达到硬件加速的目的，其具体硬件实现结构图如图 3.13 所示。

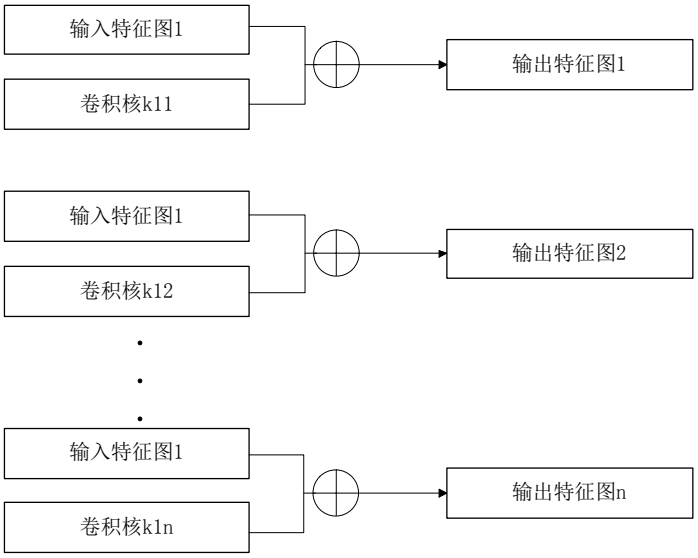


图 3.13 输出特征图并行运算硬件优化结构图

在多个生成的图像激活特征图上，相同区域的神经节点会连接在相同输入特征图上的相同位置，因而要使用不一样的过滤器模板，显然只能复用输入特征图的像素，而不能复用卷积核。该结构会产生大量的中间数据，且随着输入通道和输出通道的增加，会需要越来越多的缓存空间来存储计算过程中得到的中间值。与不同输入特征图卷积窗口并行一样，当其受到 FPGA 硬件资源限制的时候，就需要对卷积核进行分组计算。其卷积窗口移动方式有两种，第一种是对所有卷积窗口输入特征图的同一位置进行完卷积运算后，再移动卷积窗口进行下一个位置的运算；另一种方式就是第一组的卷积窗口先对整幅输入特征图计算完毕，将计算结果送至缓存，再切换至第二组的卷积窗口进行计算，直到完成所有卷积核的计算。但是该硬件设计需要大量的缓存来存储中间值是输出特征图，这是并行优化无法避免的缺点，因此需要在采用这种方法进行卷积计算单元设计时需要非常慎重地考虑计算资源、存储资源和带宽之间的关系。

在进行了对本文系统设计进行硬件优化处理后，通过使用硬件描述语言进行描述建模，最终生成的并行运算优化的卷积神经网络硬件加速器顶层模块信号如表 3.6 所示。

表 3.6 卷积神经网络加速模块顶层接口定义

信号名	说明
clk	卷积神经网络工作时钟信号
GO	卷积神经网络开始信号
RESULT	识别结果输出信号
we_database	图像数据使能信号
dp_database	图像输入数据信号
address_p_database	图像地址信号
STOP	卷积神经网络模块停止信号
rst_n	卷积神经网络模块复位信号

最终本文设计所生成的并行运算优化的卷积神经网络硬件加速器 RTL 结构图如图 3.14 所示。

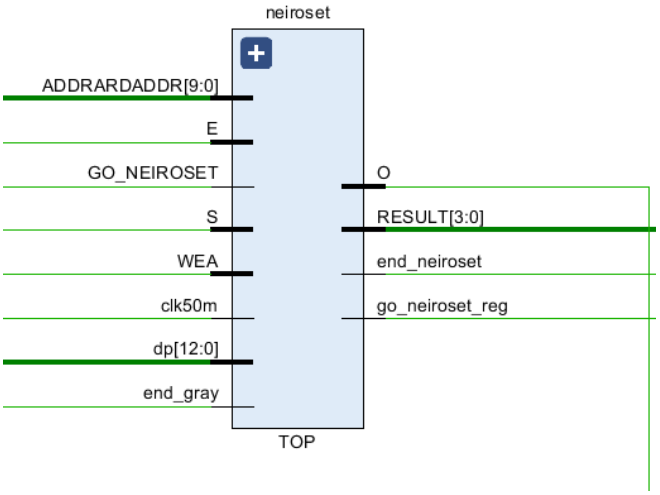


图 3.14 并行计算优化的卷积神经网络硬件 RTL 结构图

3.5 本章小结

本章主要介绍了文章所设计的精简 VGG 卷积神经网络的硬件结构设计，以 VGG 卷积神经网络模型结构为基础，为了满足本文所使用的 MNIST 手写数据集进行功能验证本文和本文所使用的 ZYNQ 7000 芯片，对 VGG 原有结构进行了一定的修改：删除

了一些卷积层、池化层及全连接层，丢弃了每个神经元中的偏置参数，去掉了一些在软件中使用的优化算法。基于 ZYNQ 7000 系列芯片内的 FPGA 设计出了本文所使用的硬化卷积神经网络模型，并分析了所设计的硬件模型主要思路，包括了对整体硬件网络结构、功能实现、结构特点的介绍；然后对卷积神经网络中的每层硬件设计实现进行了详细阐述说明，包括卷积层、池化层、激活函数、权值存储和函数分类器的实现,并给出了通过硬件描述语言实现的相关 RTL 结构图；最后对卷积运算的硬件优化实现进行了详述，包括了卷积层间的并行优化运算，输入特征图之间的并行优化运算，输入特征图之间的并行优化运算，从而最终完成了本文所设计具有并行计算优化的卷积神经网络硬件加速模块。

第四章　基于 ZYNQ 的实时识别框架设计

前一章节主要完成了对卷积神经网络的硬化实现，但是仅有硬化的卷积神经网络模型是不能够在实际场景中应用的，特别是在实时场景下。本章节主要基于 ZYNQ 系列芯片通过结合上一章节的内容完成一种实时识别的系统框架设计，通过使用 ZYNQ 芯片中的 ARM 进行图像视频采集存储，并将存储的图像视频数据通过 AXI 总线传输至 ZYNQ 芯片中的 FPGA 中，使用硬化的卷积神经网络完成对实时采集的图像数据进行识别分类，从而真正意义实现具有实际应用价值的卷积神经网络硬件加速系统。本章节的内容主要包括了实时识别框架的图像数据采集模块设计实现、图像数据存储模块架构设计实现以及总体实时识别框架的实现，本文实现的实时识别总体框架的结构设计如图 4.1 所示。

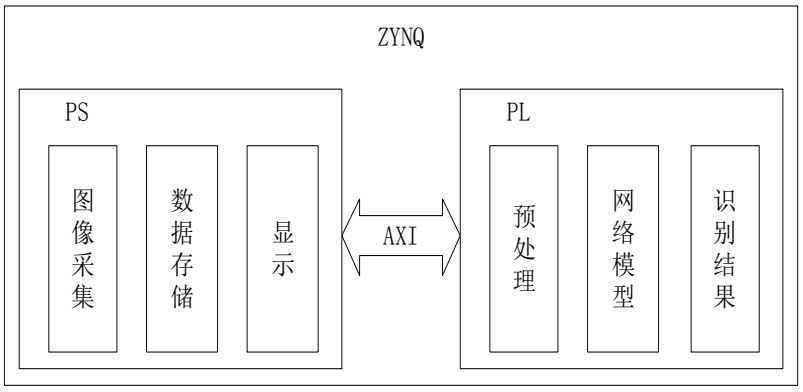


图 4.1 基于 ZYNQ 的实时识别框架结构

4.1 图像采集及显示

本文的实时识别框架系统设计中图像采集部分采用 OV7670 的 CMOS 型号的摄像头来完成外部图像数据的采集与获取，摄像头具有高度的灵敏度，非常适合在弱光环境下的图像视频采集应用，并且所需要的工作电压低使得其在低功耗设备中的使用成为了可能，此外还具备消除噪声和坏点补偿的功能，能够传输 30FPS 帧率的图像数据，从而实现实时性的需求。其硬件电路主要接口如表 4.1 所示。

表 4.1 OV7670 摄像头接口定义

信号名	说明
ADD	外部输入电压
SIO_D	SCCB 数据接口
SIO_C	SCCB 时钟接口
DATA	输出数据
PWDN	能效模式使能
VREF	参考电压
VSYNC	帧同步
HREF	行同步
PCLK	像素时钟
XCLK	系统时钟

该摄像头的图像采集流程基本是通过 CMOS 感光元器件内部的感光阵列来完成对外界图像光强的感知，根据光照强度的强弱来产生相应的电荷，通过模拟电路完成对模拟信号的转变，再将获取到的数据信号进行相关的算法处理得出最终转换后的图像数据值。该摄像头能够设置为两种分辨率的工作模式，分别为 640x480 和 320x240 的图像像素，在本文系统设计中是将该 CMOS 摄像头的分辨率设置为 640x480 的图像采样像素。OV7670 支持的图像输出数据格式非常丰富，包括了 RGB888、RGB444、RGB565 等数据格式，由于本文所使用的是 MNIST 数据集来进行系统功能验证的，因此选择采集图像的输出数据格式为 RGB565。

OV7670 摄像头的数据传输是通过 SCCB 总线接口进行的，该总线接口与 I2C 总线接口基本相同，包括了一根数据线和一根时钟信号，可以支持 100Kb 每秒或者 400Kb 每秒的数据传输速度。在 OV7670 中，CMOS 摄像机通过场同步信号与行同步时序操作来完成捕获图像数据的输出，其具体的数据操作时序图如图 4.2 所示[51]。

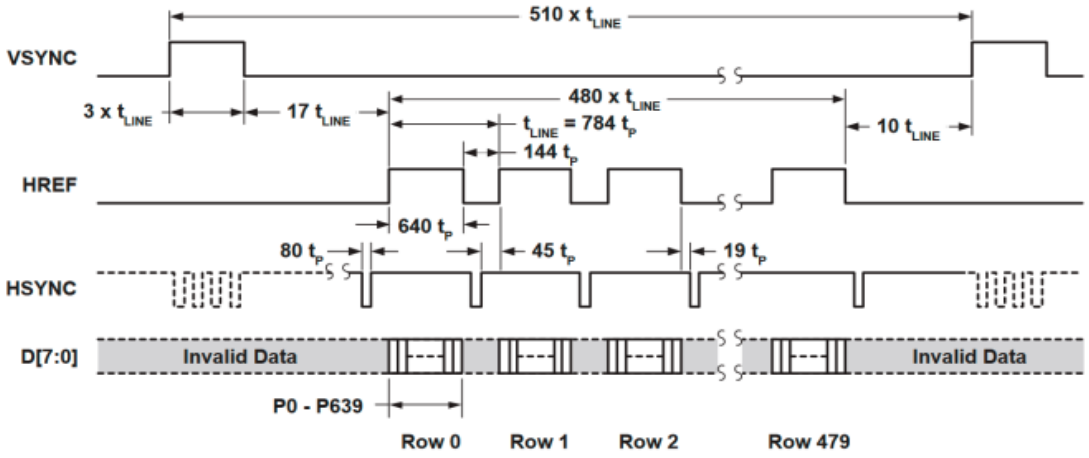


图 4.2 OV7670 器件通讯操作时序

由于 OV7670 的数据接口只有 8 根数据线，因此只能够每次完成 8bit 的数据传输，而本文所设置图像数据格式为 RGB565 格式，因此需要将第一次传输的图像数据进行缓存，再与第二次传输的数据进行拼接从而获得正确的图像数据，其具体的数据通信传输时序如图 4.3 所示[52]。

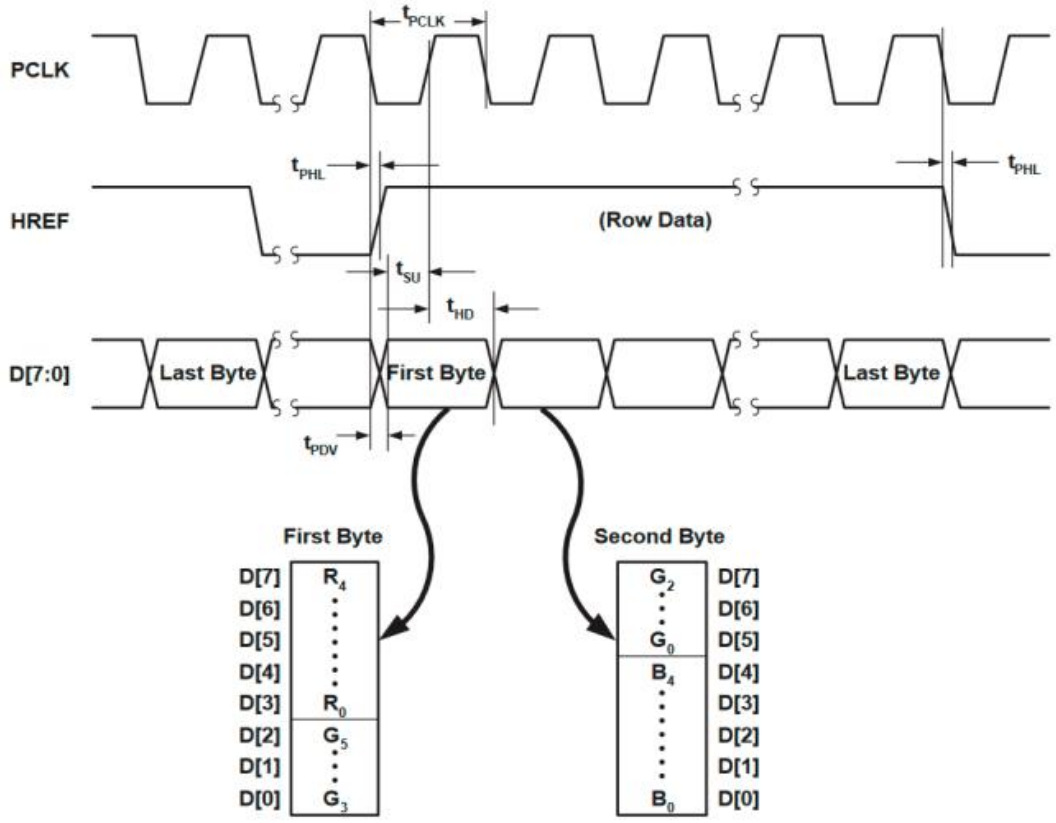


图 4.3 OV7670 器件图像数据通信时序

本文系统设计通过 OV7670 的工作原理与所需配置，通过使用硬件描述语言完成了摄像头采集模块的构建，并将其封装为 IP 核的形式如图 4.4 所示。

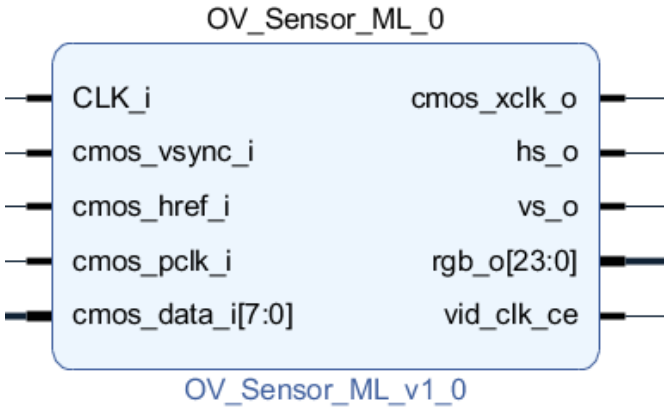


图 4.4 OV7670 IP 核

为了直观的观察采集图像数据，本系统框架使用了多分辨率显示设计方案来将通过摄像头采集的实时图像数据进行显示。分别使用了以 ILI9341 作为主控芯片的 2.8 英寸液晶显示屏和具有 HDMI 接口的显示器，从而满足不同的分辨率显示的需求。其中 HDMI 接口采用了三对差分信号以满足高速的视频流数据图像，其基本的时序原理可以根据 VGA 接口时序进行修改以达到传输数据的需求，本文所设计的实现的 HDMI 控制器模块及接口信号如图 4.5 所示。

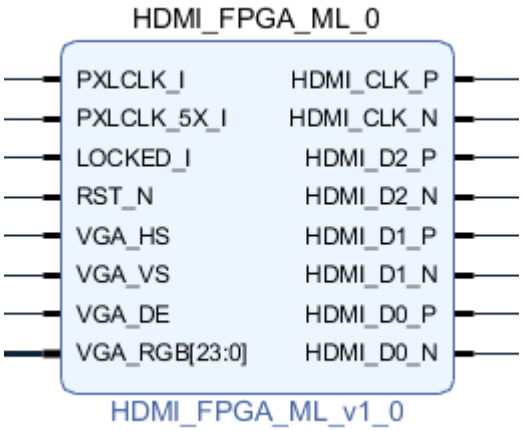


图 4.5 HDMI 控制模块及接口

TFT 显示屏的 ILI9341 主控芯片能够进行 262 和 144 色彩的图像显示，它的图像分辨率为 320x240，还包括了 320 个栅极驱动器通道和 720 个源驱动器。它内部配备 172800 字节的图形内存，并且具有外部模拟电源电路。ILI9341 芯片器件能够支持多位数据总

线接口，包括了对微处理器具有 8/9/16 和 18 位接口，与 RGB 图像格式数据拥有多达 6/18 位的接口，同时还能够对 SPI 总线进行 3/4 位的通讯接口。它根据存储输入图像来移动内部的图像窗口改变 GRAM 的地址索引从而更新出待显示的图像数据，所以能够对图像进行独立的现实而不影响其内部的图像存储区域，且工作电压低是理想的显示实现器件。其主要内部结构如图 4.6 所示。

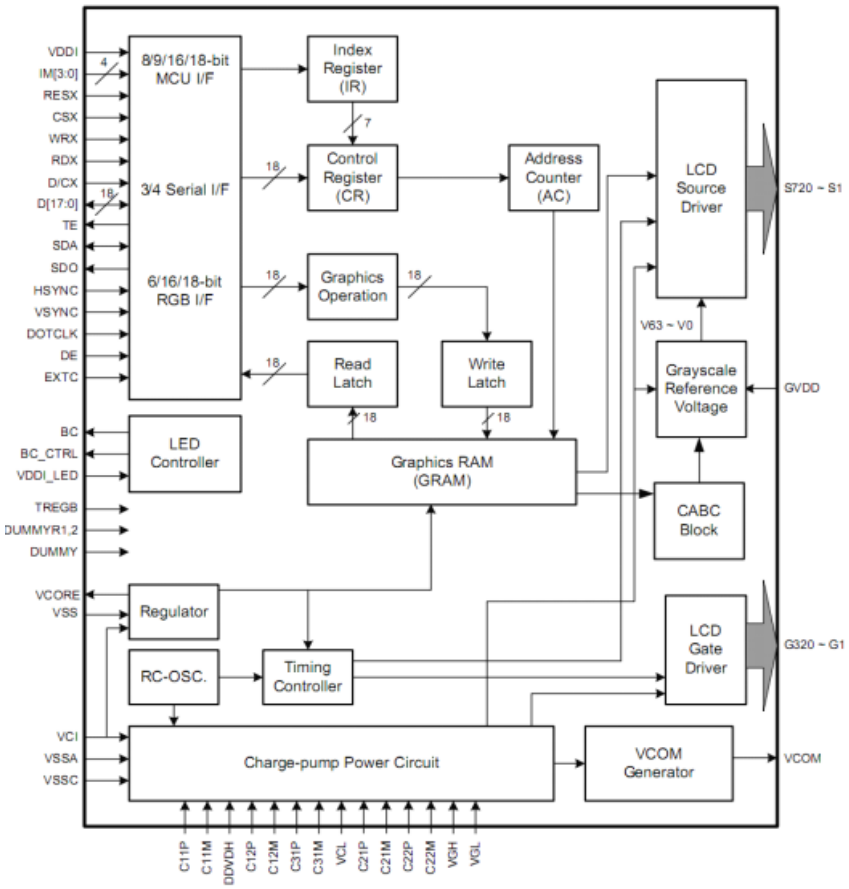


图 4.6 ILI9341 内部电路结构

ILI9341 芯片内部的关键寄存器如表 4.2 所示。

表 4.2 ILI9341 关键寄存器

寄存器地址	功能说明
04H	设备 ID 信息
0CH	显示像素格式
0DH	显示图像格式
21H	图像反转控制

续表 4.2 ILI9341 关键寄存器

26H	显示伽马值控制
2AH	列地址设置
2BH	页地址设置
2CH	写存储器
36H	访问控制寄存器
52H	显示亮度控制

本文所使用的 TFT 显示屏已经将 ILI9341 芯片进行了内部集成，因此只需关注该模块给出的外部引脚信号即可，该显示屏通过 4 线 SPI 总线进行数据传输，其数据传输时的通讯协议操作时序如图 4.7 所示。

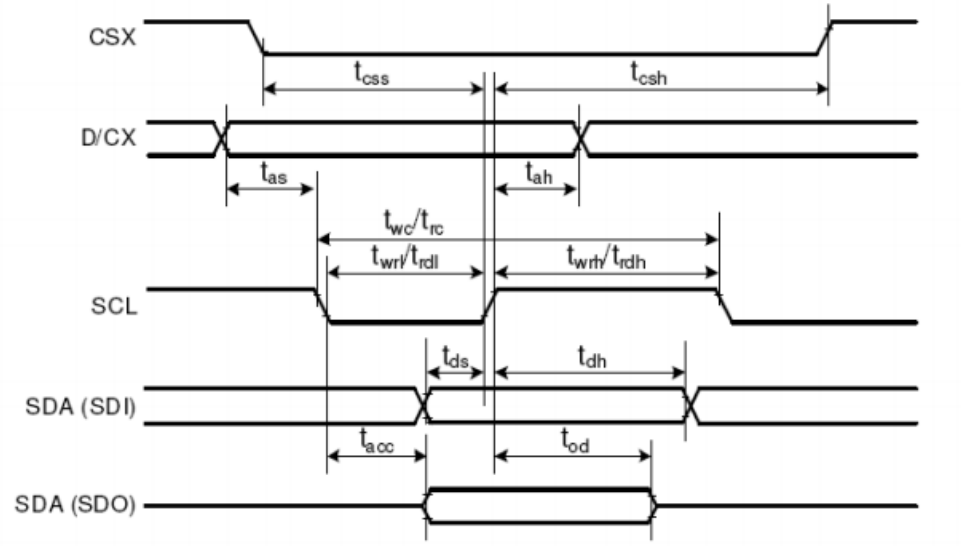


图 4.7 数据通信时 4 线 SPI 时序操作图

该时序会根据模块给出当前数据是命令数据还是图像数据，如果是命令数据将会把信号控制线拉高，否则拉低，并保持一定的时间以确保信号稳定的输入到 ILI9341 芯片中，然后再将时钟信号拉低开始进行数据的传输。根据时序操作和 ILI9341 芯片配置本系统设计通过使用硬件描述语言完成对显示模块的构建，最终使用编译工具所生成的 TFT 显示硬件模块和其接口信号结构图如图 4.8 所示。

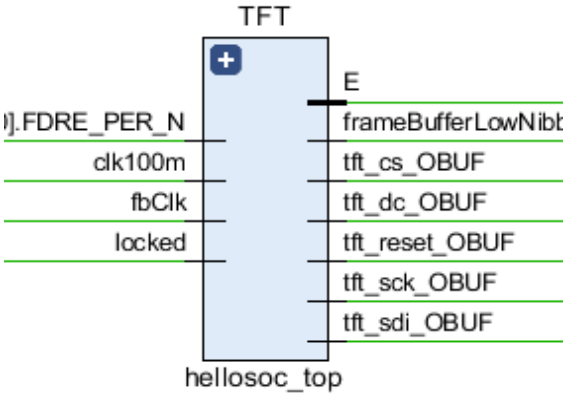


图 4.8 卷积神经网络 TFT 模块与接口信号

4.2 数据存储结构设计

由于本系统框架需要满足一定的实时性需求，因此需要高速的数据存储结构来完成对摄像头采集的数据进行存储与读取，并且还要配合硬化的卷积神经网络进行识别分类任务。因此本文系统框架设计使用了 Xilinx 公司提供的 VDMA 高速视频流控制模块和高速存储器模块 DDR 来完成系统框架的数据存储与读取。

4.2.1 VDMA 与 DDR

VDMA 是 Xilinx 公司为了满足对视频应用程序需要进行帧缓冲区来处理帧速率变化或图像尺寸的变化（缩放或裁剪）的需求所设计出的高性能图像视频控制模块，该模块通过 AXI4 总线实现接入，利用 AXI4-Stream 视频接口和 AXI4 接口之间的高效高带宽进行图像数据的交互。该模块其具体实现原理与结构如下图 4.9 所示。

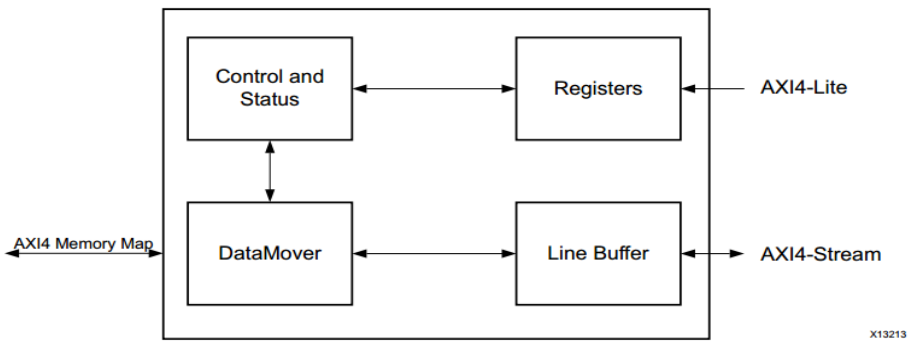


图 4.9 模块 VDMA 原理与结构图

其具体工作原理是通过 AXI4-Lite 接口首先对寄存器进行编程后，控制/状态逻辑模块为数据传输模块生成适当的命令，以启动 AXI4 主机接口上的写入和读取命令，以及用于将像素数据写入 AXI4-Memory Map 所使用的可配置不同传递缓冲区，该缓冲区能够将暂时保存像素数据在接口当中，或者也可以将其写入 AXI4-Stream 接口；在写路径中，AXI 接口的 VDMA 将会接受来自于 AXI4-Stream Slave 接口上的视频图像帧,并使用 AXI4 主接口将该数据存入系统的存储单元模块当中，在读取路径中，AXI 接口的 VDMA 使用 AXI4 主接口从系统存储器读取图像数据帧并在 AXI4-Stream 主接口上完成对其输出，并且此写入与读取路径均是独立运行的。此外 AXI 接口的 VDMA 还提供了一个选项，可用于将输入/输出帧与外部同步信号同步以满足更多的需求,在实际的硬件工程中该模块的封装结构形式及接口信息如图 4.10 所示。

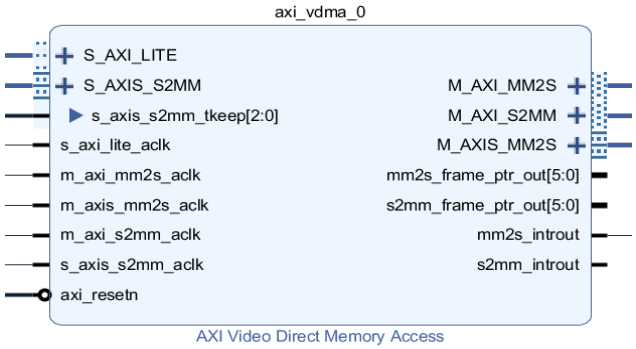


图 4.10 VDMA 模块结构与接口信息

该模块数据的交互通信主要是通过 AXI4 总线完成的，相关信号定义如表 4.3 所示。

表 4.3 VDMA 信号定义

信号名称	说明
mm2s_fsync	MM2S 帧同步输入。使能该信号后，VDMA 操作开始于 fsync 每个下降沿。该信号至少要持续一个 m_axis_mm2s_aclk 时钟周期
s2mm_fsync	S2MM 帧同步输入。使能该信号后，VDMA 操作开始于 fsync 每个下降沿。该信号至少要持续一个 s_axis_s2mm_aclk 时钟周期
mm2s_frame_ptr_in	主机到从机输入的帧图像编号
mm2s_frame_ptr_out	主机到从机输出当前图像帧的编号
s2mm_frame_ptr_in	从机到主机输入的帧图像编号
s2mm_frame_ptr_out	从机到主机输出当前帧图像的编号

为了能够完成对该模块的控制还需要对其时序进行分析，首先对于主机到从机的读取时序如图 4.11 所示。

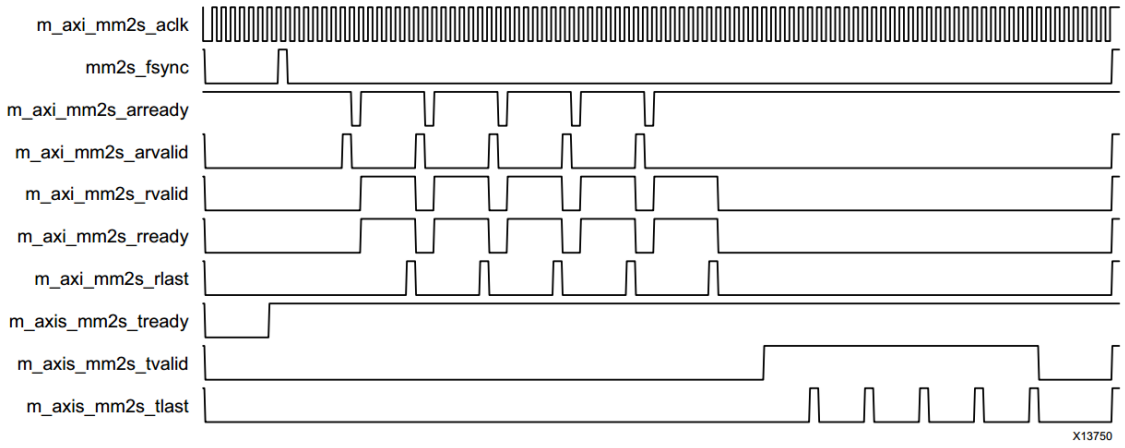


图 4.11 VDMA 主机到从机的读时序

从图中可以看出，当 VDMA 模块从 mm2s_fsync 接收到信号时，VDMA 将在 m_axi_mm2s_araddr 的初始地址产生出 m_axi_mm2s_arvalid 信号。并且 M_axi_mm2s_arvalid 信号的使能次数为 5 次，在该信号使能期间将会分别获取视频数据每帧的 5 行图像数据，从 MM 读取的数据会放入行缓冲器当中进行存储。当接收到来自 axi-stream 的 m_axis_mm2s_tvalid 信号时，数据会被发送至 axi-stream 信号线上。在每一行的末尾，axi-stream 将使 m_axis_mm2s_tlast 信号有效。从机到主机的写时序如图 4.12 所示。

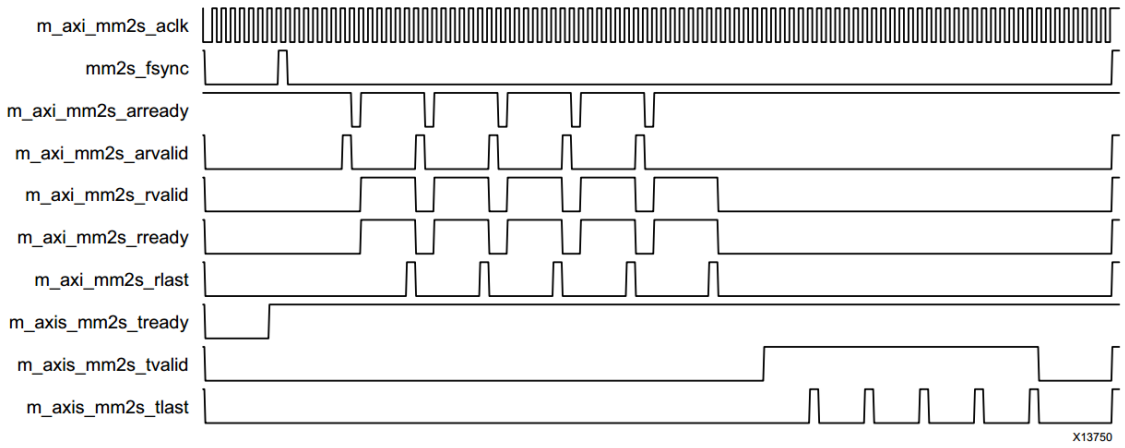


图 4.12 VDMA 从机到主机的写时序

从图中可以看出，当 VDMA 模块接收到 s2mm_fsync 信号时，VDMA 模块将产生出 s2mm_fsync_out 和 s_axis_s2mm_tready 信号以指示它已经准备好从 axi-stream 信号接

口来进行接收数据。读取的数据存将会被存储在行缓冲区当中。在 m_axi_s2mm_awvalid 信号有效时，m_axi_s2mm_wvalid 信号将会变为有效，并且会把数据放置在 m_axi_s2mm_wdata 当中以完成数据交互。

在 VDMA 模块中为了与实际的视频流数据格式相匹配还需要对其进行相关寄存器配置，其中主要的寄存器如表 4.4 所示。

表 4.4 VDMA 主要寄存器

寄存器偏移地址	说明
00H	MM2S VDMA 控制寄存器
04H	MM2S VDMA 状态寄存器
14H	MM2S 寄存器索引
30H	S2MM VDMA 控制寄存器
34H	S2MM VDMA 状态寄存器
44H	S2MM 寄存器索引

为了完成输入图像数据的高速存储本文使用DDR3外部高速存储器来将获取的图像数据进行缓存，DDR3 具有 4bit 的突发长度，简洁的寻址时序，自动的 ZQ 校准功能，低功耗高效率的特点被广泛的使用在计算机系统当中。关于其具体工作原理、接口定义及时序信息由于篇幅限制不再赘述，本文系统框架设计使用了 ZYNQ 7000 系列芯片 ARM 端内置的 DDR 控制器完成对外部 DDR 存储器的访问，以达到高速的数据交互，在 ZYNQ 芯片中 DDR 控制器的配置较为简便，只需明确外部所使用的 DDR 芯片型号即可，经过对 ZYNQ 芯片进行设置后生成的 ZYNQ IP 模块及接口如图 4.13 所示。

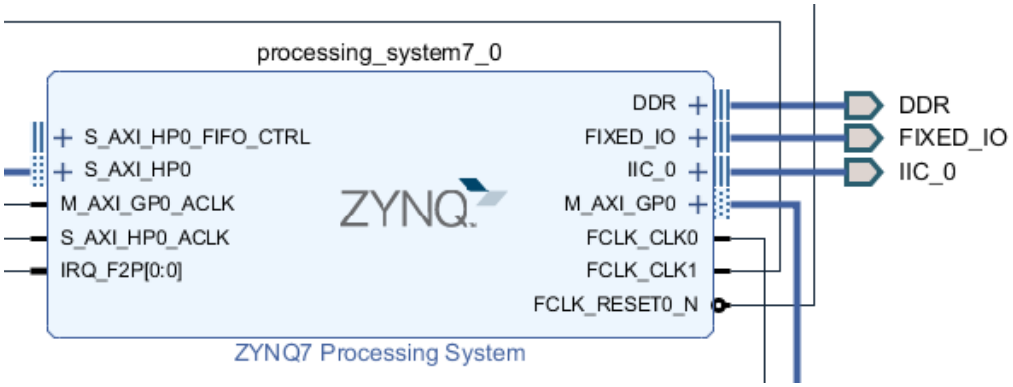


图 4.13 ZYNQ DDR 配置结果

4.2.2 视频时序控制模块

由于视频的分辨率会有一定的时序需求，因此需要使用视频时序控制器来完成对当前输入图像数据的时序控制，本文实时识别系统框架中使用了 VTC 视频时序控制器 IP 来完成对输入图像的时序控制，该模块的主要作用就是根据系统设定的视频分辨率产生标准的场同步信号和行同步信号时序，该时序控制模块的具体配置与最终实现的封装模块结构信号如图 4.14 所示。

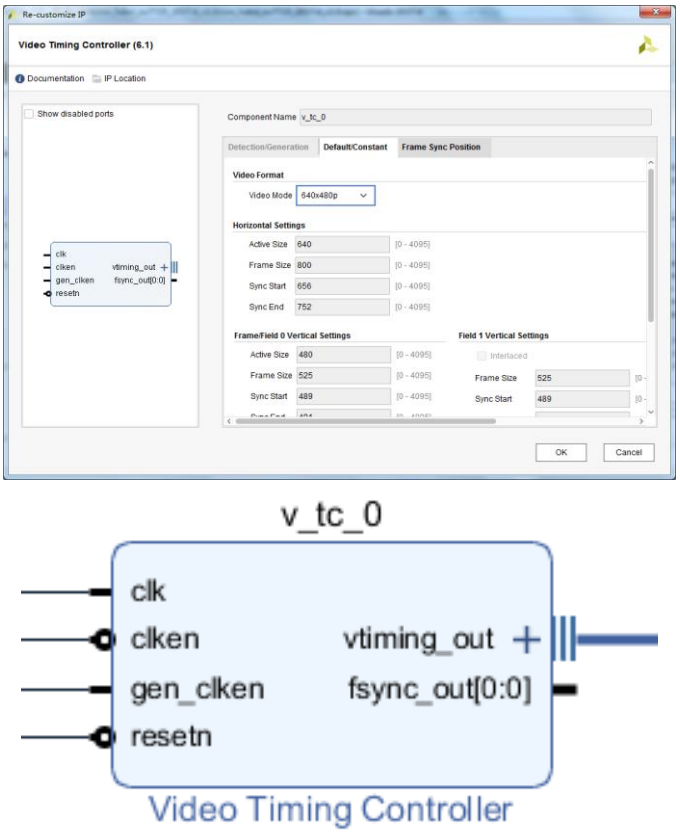


图 4.14 VTC 配置与结构信号

4.2.3 降采样模块与 BRAM

在摄像头数据采集模块配置本文将视频的分辨率设置为了 640x480，但是由于采用 ILI9341 芯片的 TFT 显示屏最高只能支持 320x240 的分辨率，因此本文设计了一种降采样模块，将 640x480 分辨率降低为 320x240 分辨率，主要利用的是对视频流数据输入间隔地对输入数据进行有效使能，从而到达降采样的目的。通过使用硬件描述语言对其进

行编码生成的结构图及信号如图 4.15 所示。

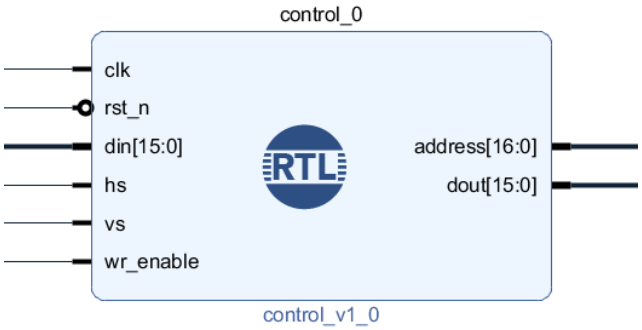


图 4.15 降采样结构及信号

为了减少图像数据的存储容量，本文输入硬化卷积神经网络的数据图像分辨率为 320x240，因此在获取到降采样后的图像数据后，可以将其存储至 ZYNQ 的片内 BRAM 当中，减少了外部存储器件与硬化卷积神经网络间的数据交互，从而一定程度上加速了系统的实时识别效率。由于该分辨率下的图像数据共有 76800 个，并且对于每一个数据而言，它们是由 16 位组成的，基本能够满足 ZYNQ 的存储容量，因此在设计中使用了双端口 BRAM，将它的深度与数据位宽之设置成一样即可，并且能够进行不同时钟域间的数据传输。

4.3 实时识别框架系统实现

在完成了图像视频信息采集模块和高速数据存储结构设计后，将每个模块进行组合后从而完成了本文基于 ZYNQ 的实时识别框架设计实现，最终设计生成的具体实现模块子系统如图 4.16 所示。

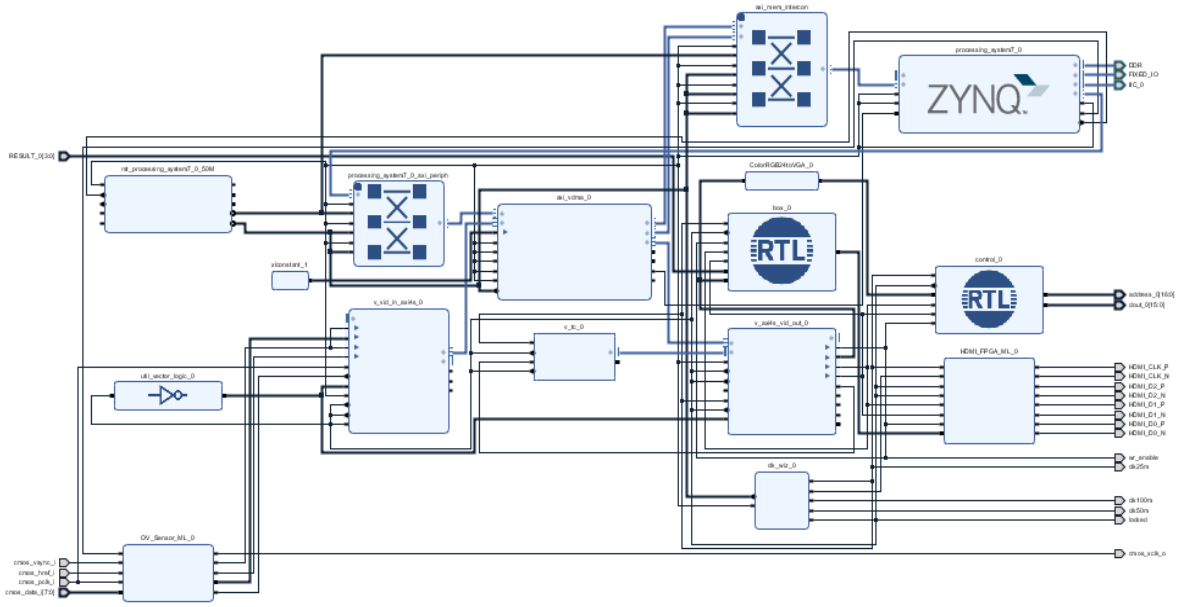


图 4.16 实时识别系统框架子系统

通过该实时识别系统结构能够完成对摄像头采集的数据进行实时的采集，将采集后的图像数据通过 AXI 总线传输至 VDMA 控制模块，该模块再根据 ARM 端的控制信号将图像数据存储至外部器件 DDR 内，经过 AXI 总线读取 DDR 内部图像数据使用 HDMI 接口传输图像进行显示，并且同时将 DDR 内部的图像数据进行降采样传输至片内的 BRAM 当中使用 ILI9341 控制芯片进行 TFT 显示。实时识别框架还为硬化的卷积神经网络模块留下了相关信号接口，当将硬化的卷积神经网络连接时，从而实现卷积神经网络的硬件加速系统设计，进行实时图像内容识别分类。

4.4 本章小结

本章主要阐述了实时识别框架的结构设计，对本文所使用的图像采集与两种显示方案硬件结构实现进行了详细说明；然后对实时识别框架中的高速数据存储结构实现进行了介绍，包括了对图像视频数据的时序控制、输入图像数据降采样存储至片内 BRAM 和 VDMA 与 DDR 之间高速数据间的交互；最后通过 Vivado 设计工具完成了整个实时识别硬件框架的构建，并且为硬化的卷积神经网络的接入留下了相应的接口，以便最终实现卷积神经网络的硬件加速系统设计。

第五章　硬件加速系统测试与验证

前两章主要完成了对卷积神经网络的硬化实现和实时识别系统框架的设计。为了测试其功能性，本章则对其进行实际场景的功能验证。首先对硬化卷积神经网络中的每一层进行了仿真验证；然后通过深度学习库 keras 和 MNIST 手写数据集完成了对本文设计的卷积神经网络模型的权值训练，并且将权值存储至硬化卷积神经网络的权值存储模块中；最后将具有权值的硬化卷积神经网络接入实时识别系统框架中进行测试验证，从而实现了卷积神经网络硬件加速系统设计。

5.1 卷积神经网络各模块测试和验证

卷积神经网络模块中卷积层的硬件实现测试与验证。在硬件实现的卷积层中编写测试平台，然后通过 modelsim 逻辑仿真测试平台，使用其自带的相关函数对设计好的卷积神经网络中的卷积层进行逻辑仿真测试。首先进行了单层卷积层仿真验证，仿真结果如图 5.1 所示。

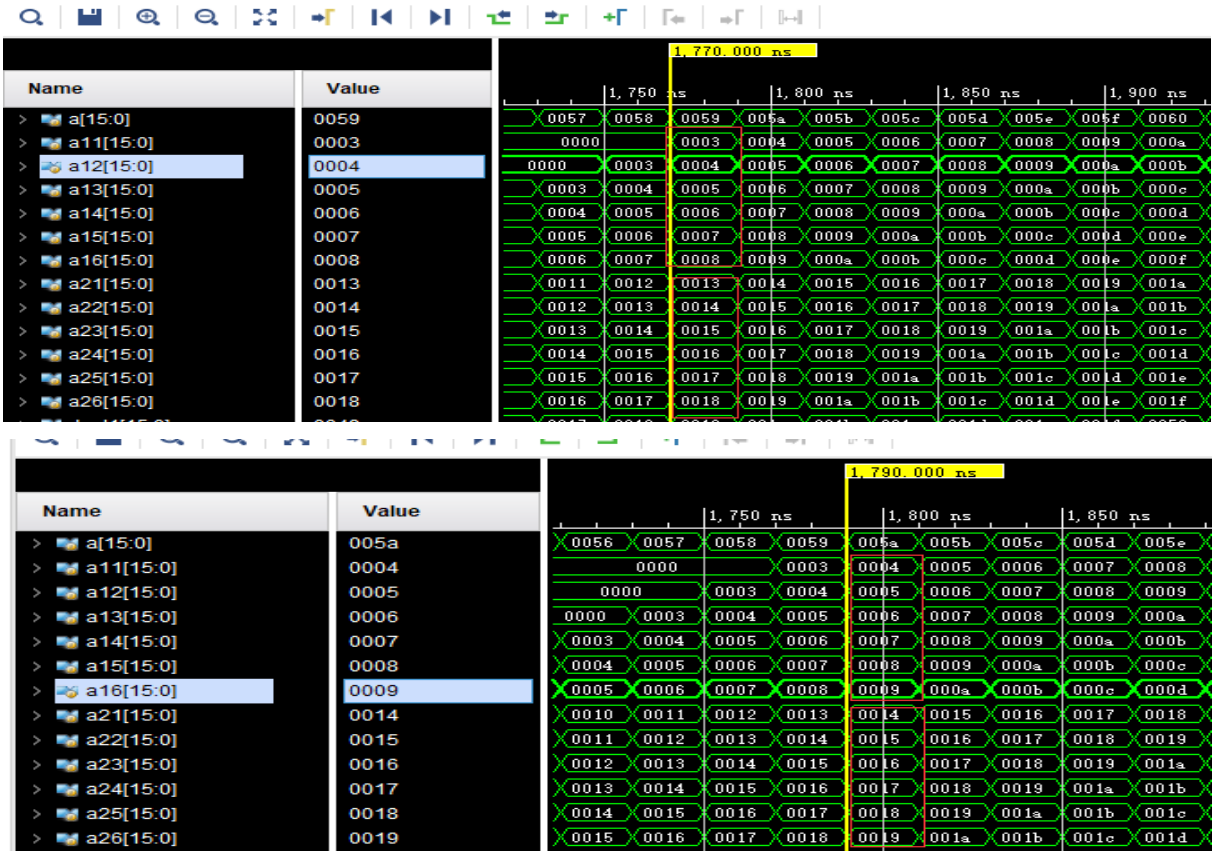


图 5.1 单层卷积层硬件仿真测试

从图中可以看出单层卷积层的运算仿真结果与预期一致，从而验证了单层卷积层的功能，然后将多个卷积层进行组合后再使用 modelsim 进行仿真，仿真结果如图 5.2 所示。

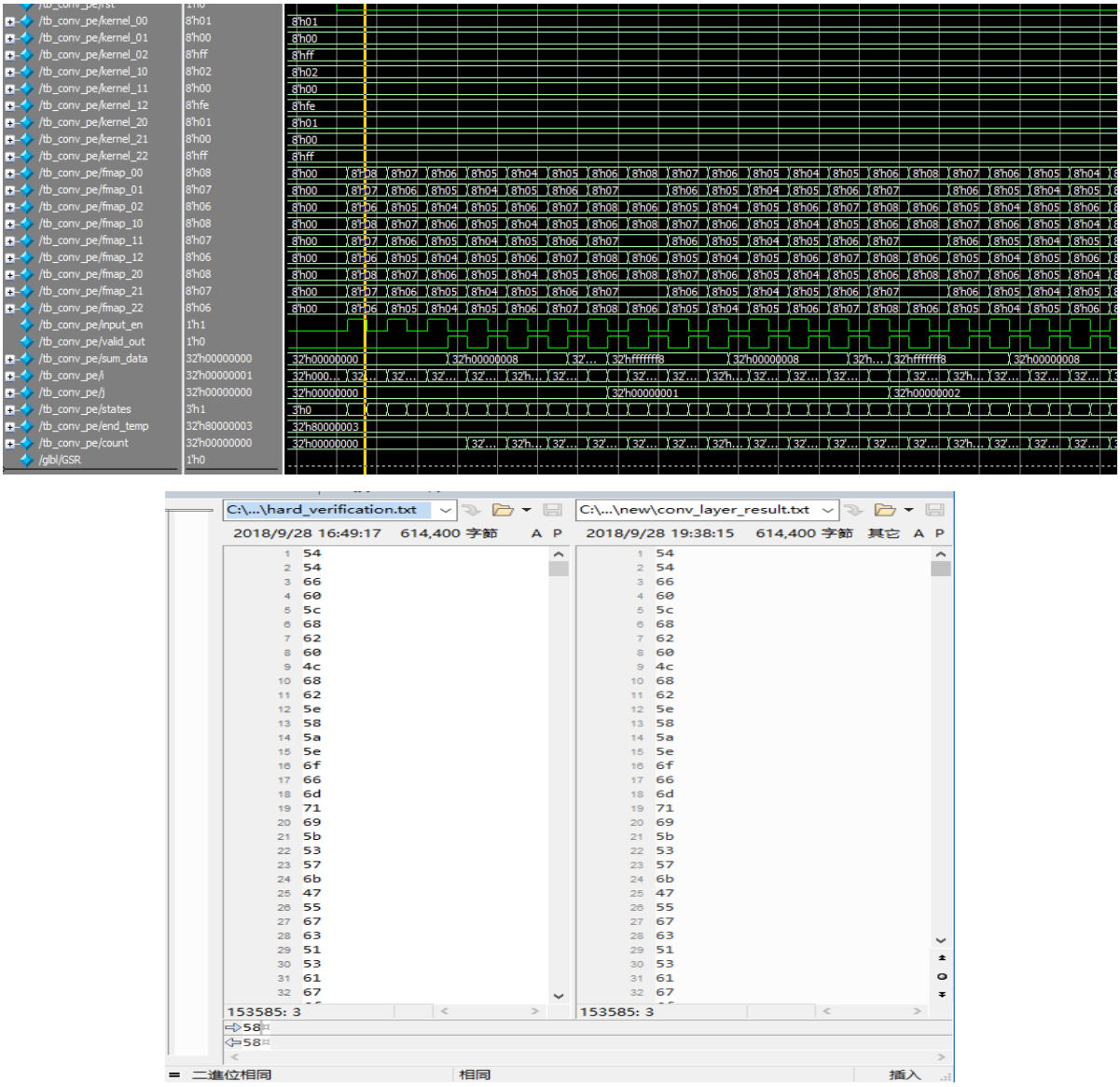
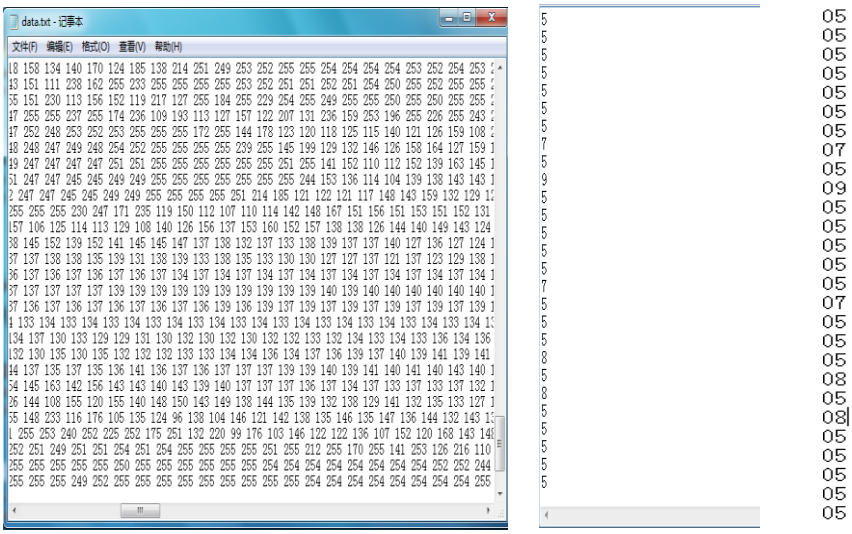


图 5.2 卷积层硬件仿真测试

可以看整个卷积层的运算功能满足设计要求，从而为实现卷积神经网络的硬化提供了基础。

卷积神经网络模块中池化层的硬件实现测试与验证。在测试中通过将尺寸为 128x128 像素的图像输入池化层中，将计算的结果再进行导出与软件运行进行对比，输入图像数据如图 5.3(a)所示，硬件仿真结果如图 5.3(b)所示，软件仿真结果如图 5.3(c)

所示。

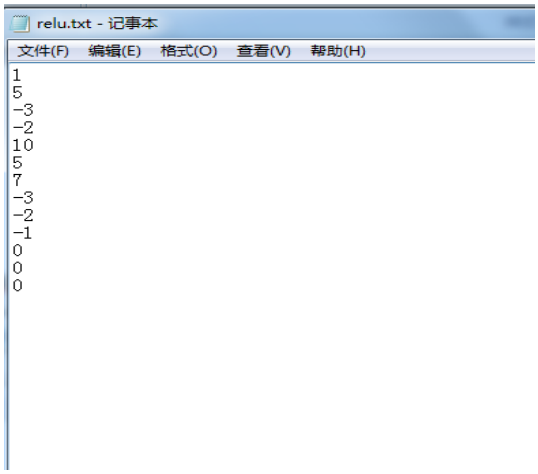


(a) (b) (c)

图 5.3 池化层硬件仿真测试

通过硬件仿真结果可以看出硬件设计的池化层运算结果与软件仿真的运算结果一致，从而确保了池化层的功能。

卷积神经网络模块中激活函数的硬件实现测试与验证。由于激活函数的硬件设计非常简单，仅使用一级比较器来实现，因此直接使用 modelsim 软件进行对激活函数硬件层的仿真测试验证，仿真结果如图 5.4 所示，其中 5.4(a)为输入数据，5.4(b)为输出数据。



(a)

(b)

图 5.4 激活函数硬件仿真测试

通过仿真可以看出在输入值小于 0 时输出为 0，大于 0 时输出其本身，该激活函数功能准确，从而为实现卷积神经网络的硬化提供了基础。然后将硬化的卷积神经网络模型使用 Vivado 工具进行编译综合，最终生成的电路图如图 5.5 所示。

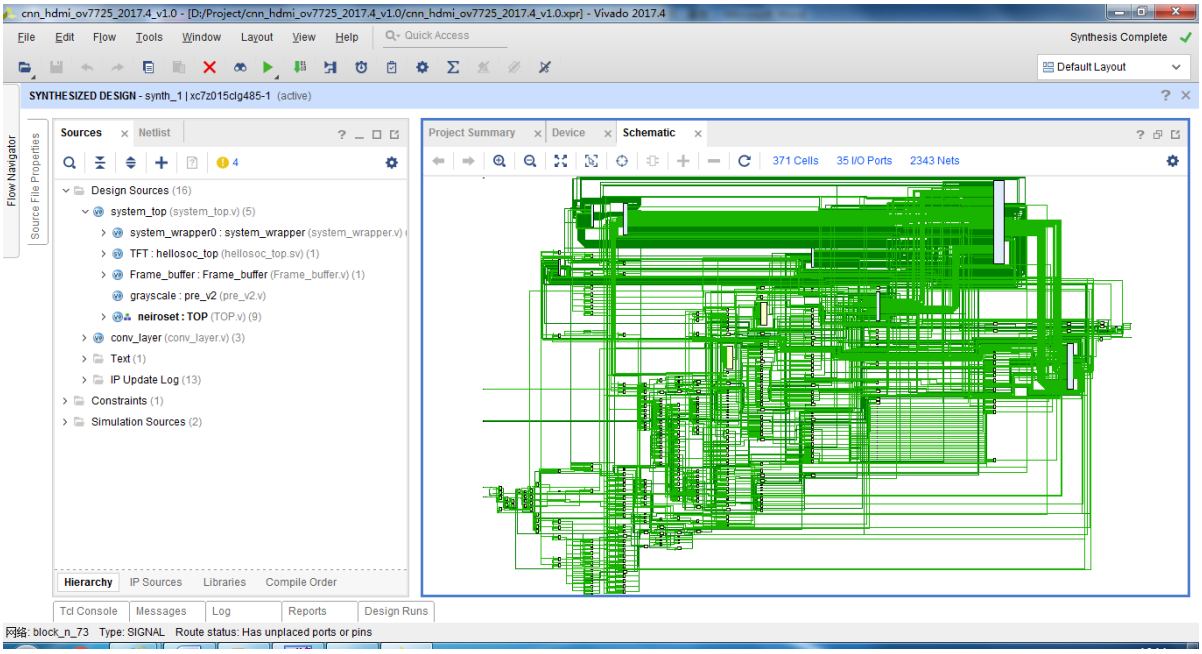


图 5.5 硬化卷积神经网络电路图

最终生成的硬化卷积神经网络电路所占用的器件资源如表 5.1 所示。

表 5.1 逻辑资源消耗情况

Resource	Estimation	Available	Utilization %
LUT	2205	46200	4.77
FF	1334	92400	1.44
BRAM	12	95	12.63
DSP	16	160	10.00
IO	34	150	22.67
BUFG	1	32	3.13

从硬件资源消耗可以看出本文设计的硬化卷积神经网络基本满足器件资源的容量。

5.2 卷积网络模型权值训练

本文使用了 Keras 深度学习库完成了对本文所设计出的卷积神经网络模型权值训练。首先通过 keras 相关的 API 函数接口完成了本文的卷积神经网络结构模型的搭建，相关代码为：

```
x = Conv2D(4, (3, 3), activation=None, padding='same', name='conv1', use_bias=use_bias)(inputs1)
x = Activation('relu')(x)
x = Conv2D(4, (3, 3), activation=None, padding='same', name='conv2', use_bias=use_bias)(x)
x = Activation('relu')(x)
x = MaxPooling2D((2, 2), move=(2, 2), name='pool1')(x)
x = Conv2D(8, (3, 3), activation=None, padding='same', name='conv3', use_bias=use_bias)(x)
x = Activation('relu')(x)
x = Conv2D(8, (3, 3), activation=None, padding='same', name='conv4', use_bias=use_bias)(x)
x = Activation('relu')(x)
x = MaxPooling2D((2, 2), move=(2, 2), name='pool2')(x)
x = Conv2D(16, (3, 3), activation=None, padding='same', name='conv5', use_bias=use_bias)(x)
x = Activation('relu')(x)
x = Conv2D(16, (3, 3), activation=None, padding='same', name='conv6', use_bias=use_bias)(x)
x = Activation('relu')(x)
x = GlobalMaxPooling2D()(x)
x = Dense(11, activation=None, use_bias=use_bias)(x)
x = Activation('softmax')(x)
model = Model(inputs=inputs1, outputs=x)
```

然后使用 PC 端运行 keras 代码，训练方法选择随机梯度下降法，最终完成对网络权值的训练，训练结果如图 5.6 所示。

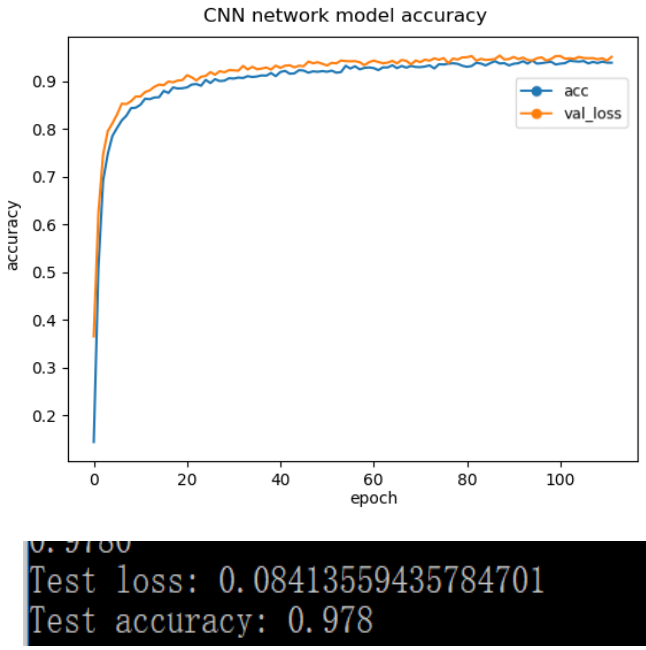


图 5.6 软件训练结果

可以看出本文所设计出的卷积神经网络模型的准确率为 97.8%，能够实现对 MNIST 手写数据集的识别分类。然后将训练好的网络模型中的权值参数提取，对其进行 11bit 定点量化处理，该处理通过循环算法的方式寻找出最优的量化数值，具体量化算法伪代码如下：

```
def best_weight()
    while err !=0
        for i = 8; ;i++
            exchange(weight);
            test(weight);
            err= compare();
```

算法首先从 8bit 量化开始，将当前的权值进行 8bit 数值量化，然后将该数值放入网络模型中替换未量化的权值数值，然后使用测试集进行验证，将误差结果与未量化的权值网络结果进行对比，得出误差值，如果误差不为 0 则增大量化位宽，从而寻找无损量化位宽，本文最终生成的是 11bit 无损定点量化结果。生成的定点量化权值数值如图 5.7 所示。

```
storage[784] =  11'b00010010100; // 148
storage[785] =  11'b00100111101; // 317
storage[786] = -11'b00001101011; // -107
storage[787] = -11'b00000101001; // -41
storage[788] = -11'b00000011010; // -26
storage[789] = -11'b00000011100; // -28
storage[790] = -11'b00011000001; // -193
storage[791] =  11'b00010101100; // 172
storage[792] = -11'b00011000001; // -193
storage[793] = -11'b00001010111; // -87
storage[794] = -11'b00000111011; // -59
storage[795] =  11'b00100010000; // 272
storage[796] = -11'b00001011011; // -91
storage[797] = -11'b00000100100; // -36
storage[798] =  11'b00011100100; // 228
```

图 5.7 定点量化生成的权值数值

5.3 卷积神经网络硬件加速实时识别框架系统测试

文章中具体使用的是 ZYNQ-7015 芯片来实现硬件加速系统，首先将硬化的卷积神经网络接入至上一章所设计的实时识别框架结构中，最终系统设计工程的顶层结构如图 5.8 所示。

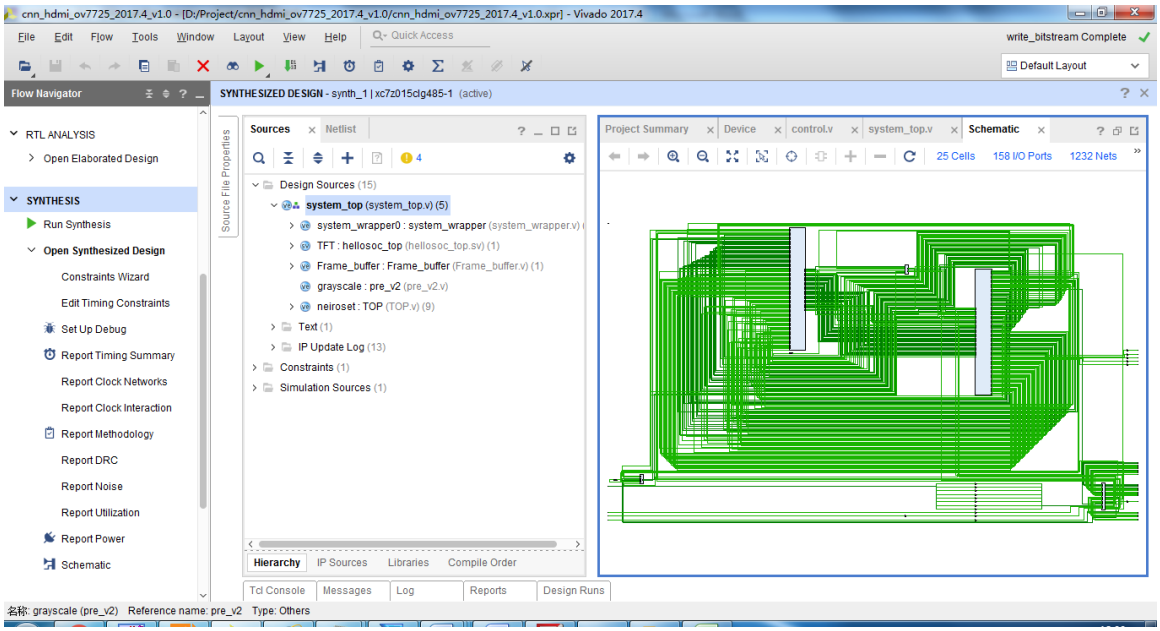


图 5.8 卷积神经网络硬件加速实时识别系统

然后在该芯片上使用 Vivado 工具进行综合实现，最终生成的硬件电路所占用的器

件资源如表 5.2 所示。

表 5.2 逻辑资源消耗情况

Resource	Utilization	Available	Utilization %
LUT	7774	46200	16.83
LUTRAM	246	14400	1.71
FF	8134	92400	8.80
BRAM	55	95	57.89
DSP	16	160	10.00
IO	27	150	18.00
BUFG	9	32	28.13
MMCM	1	3	33.33

可以看出占用逻辑资源最多的是片内存储器 BRAM，因为本文对输入视频图像进行了降采样处理并且存储在了芯片内部，但是其他的资源使用较少，再使用 VIVADO 工具的对芯片的能耗分析工具结果如图 5.9 所示，

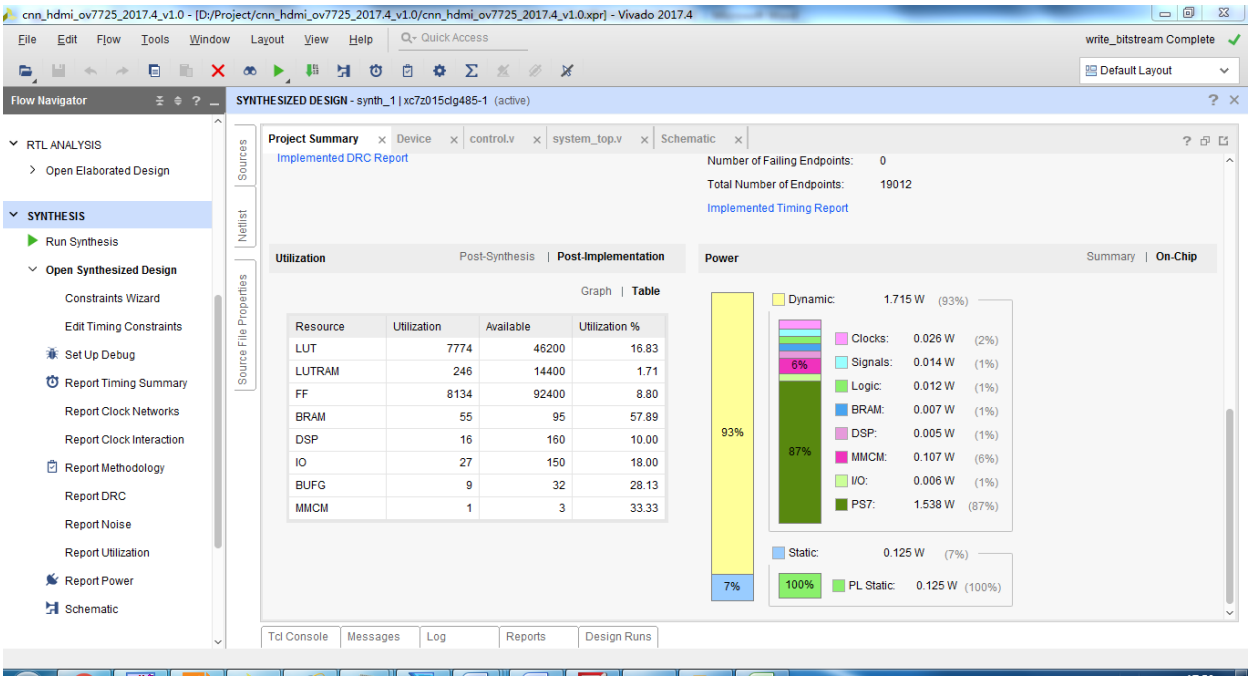


图 5.9 能耗结果

可以看出总体能耗仅有 0.125W，因此本系统设计具有低功耗的特点。

然后将生成的电路部署至 ZYNQ 芯片中进行 MINIST 手写识别的实际场景的功能测试，测试结果如图 5.10 所示。

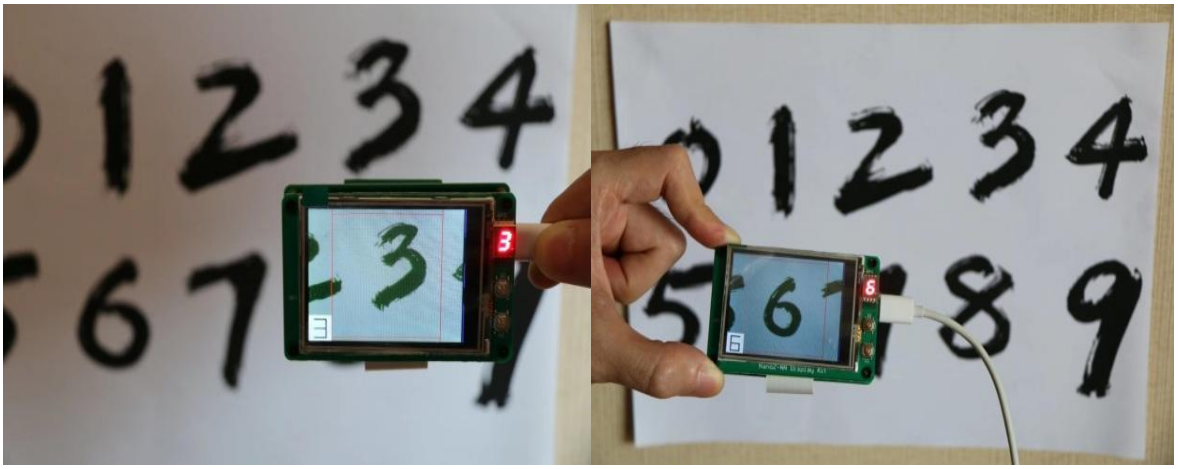


图 5.10 卷积神经网络硬件加速实时识别系统测试结果

为了测试应用在不同场景中，本文还通过使用 keras 深度学习库训练对 Fashion-MNIST 数据集完成了网络模型的训练，并且将训练出的权值导出至本文设计的硬化卷积神经网络结构当中，再将硬化的卷积神经网络接入实时识别系统框架中进行测试，测试结果如图 5.11 所示。

图 5.11 替换硬化神经网络后实时识别系统测试结果

通过实际场景的实验，可以看本文所设计的系统能够完成对实际场景内容的实时识别，并且可以根据实际需求任务快速的完成硬件卷积神经网络的构建，再将其接入至实时识别系统框架中，从而实现了卷积神经网络硬件加速系统的设计研究。

5.4 本章小结

本章主要完成了硬化的卷积神经网络和实时识别系统框架的仿真与测试，并且展示了最终的实时场景中的测试结果。本章首先对文章中所设计出的硬化卷积神经网络中的

各模块进行了仿真测试与功能测试；然后根据文章在硬件上设计好的网络模型，使用深度学习库 Keras 来对其进行模型构建，训练出本文章的卷积神经网络权值参数，再将其进行了 11 位定点量化后存储至硬化的卷积神经网络权值存储模块中；最后将硬化后具有权值参数的卷积神经网络模型接入至实时识别系统框架中进行实际测试，同时还进行了不同应用场景下的效果测试，实验结果表明本文设计的卷积神经网络硬件加速系统具有实时识别的功能，能够根据不同场景任务需求快速完成系统的搭建，并且移植性高且所需功耗低。

第六章　总结与展望

6.1 本文总结

本文对深度学习中的卷积神经网络硬件加速系统设计进行了深入的研究，主要围绕卷积神经网络的硬件设计实现和实时识别系统框架进行了详细阐述。文章从对深度学习和卷积神经网络的发展现状开始，对当前卷积神经网络硬化实现的难点和现阶段实现的不足进行了简要的分析与说明，从中汲取了优势设计出了本文的卷积神经网络硬件结构设计。并且对本文的硬件结构实现进行了详细的说明，包括了具体的卷积层硬件实现、池化层硬件实现、权值存储硬件实现、函数分类器硬件实现，同时还在基于 ZYNQ 系列芯片的 FPGA 上将卷积神经网络进行了具体硬化实现，并且利用了 FPGA 的并行计算特性与流水线技术减少了卷积神经网络的计算时间，从而实现了卷积神经网络结构的硬件加速目的。此外为了满足一定实时场景下对于图像识别的应用需求，本文还提出了一种实时识别硬件系统框架，通过使用摄像头进行数据采集，并将采集到的实时图像数据通过 VDMA 控制器存储至片外的 DDR 中，通过采用软硬件协同的方式，使用 ZYNQ 系列芯片的软件完成图像数据的显示，并且进行了两种分辨率的实现，同时将图像数据通过 AXI4 总线传输至 FPGA 中硬化后的卷积神经网络来完成对图像数据的实时识别，并且文章在最后还使用了不同的数据集进行训练权值参数，通过替换权值参数使得本文的系统硬件框架能够满足多场景下的实时识别任务需求。

在最后章节的实验结果表明，本文所研究的硬化卷积神经网络模型能够的各个模块均能够完成相应的功能，并且对卷积运算模块则能够在单个时钟周期内完成 528 次卷积运算，相较于通用 CPU 的计算效率得到了显著提升，并且使用 keras 深度学习库进行本文设计的卷积神经网络模型权值的训练，将获取的权值参数进行 11 位定点量化后整体网络的准确率为 97.8%，基本达到了无损量化，具有较高的识别率。最后对本文设计出的卷积神经网络硬件加速实时识别系统进行了实际场景的实时识别测试，测试结果表明该系统能够达到实时识别分类的目的，完成了本文所研究的目的。同时结合 ZYNQ 器件中高度模块化设计使得整个系统框架具有移植性高的特性，并且对系统整体运行时所需的功耗进行分析后，该系统整体运行功耗低。

6.2 工作展望

在今后的时间，深度学习中卷积神经网络硬化的实现将是学术界研究的热点之一，也是推动深度学习更加广泛实际应用的关键。本文设计的卷积神经网络硬件加速系统由于时间有限，后续还存在着许多改进之处有待完成，主要有以下几个方面：

第一，增加硬化卷积神经网络结构规模。现如今使用广泛的卷积神经网络模型基本都包含大量的层数，并且深度较深，而本文所设计的网络结构较小，虽然具有一定的实用性，但是还是不能够满足更多的场景需求，因此后续设计将会对硬件结构进行改进，使其网络模型的结构更大。

第二，减少量化位宽，现阶段主流的定点量化位宽主要是 8bit 的位宽，这样能够节省更多的逻辑存储资源，从而可以使得卷积神经网络硬件结构变大，适用于更多的实际应用中，后续将会对本文所使用的 11bit 量化位宽进行缩减。

第三，图像视频采集部分硬件实现，图像视频数据在分辨率更高的时候数据量也较大，对于实时识别的速率要求较高，而使用软件完成对图像视频数据进行处理将会大大降低了实时性，所以将会对本文设计实时识别系统中的图像采集和存储模块进行改进，以完全使用硬件来进行处理，获取更高的处理效率。

致谢

时光飞逝，随着完成毕业论文的设计我的三年研究生学习生涯也即将画上圆满的句号，经过这段美好且充实丰富的学习经历，使我的专业技能和理论知识得到了较为明显的提升，为今后的工作与发展奠定了坚实的基础。在此，与帮助、指导、支持和关注过我的老师、同学和朋友表示由衷的感谢与感恩。

首先，对于我的指导老师周骅副教授表示衷心的感谢，感谢周骅老师一直以来对我学业上的悉心教导与教诲，周老师专业知识丰富，平易近人，通情达理，对我的科研内容给予了非常大的帮助，并非常严谨且耐心的对我所做的实验内容进行指导，还会传授给我相关的技术经验，以帮助我能够高效的掌握相关知识，在平时的生活中，周老师也对我表现出了关心。周老师严谨的科研态度和创新精神也给我留下了深刻的印象，使我在今后的生活中以他为榜样，在此向周骅老师表示衷心的感谢。

此外还有实验室的赵麒博士，非常感谢您无私的对我工程设计方面的谆谆教导，给予了我这方面非常丰富的实际经验与技能，指导我完成了相关的工程设计实验，给予了我非常大的帮助，并且还给我讲述了很多生活中和工作中的道理，为我今后步入社会打下了一定的基础，在此向赵麒博士表示由衷的感恩。

同样还要感谢实验室的师兄师姐还有师弟师妹对我的帮助，感谢你们对我的研究方向做出的帮助，为我传授了前沿的研究课题，帮助我解决了我在实验中所遇到的问题，并且提供了相关的实验文档和工具使用文档，使我快速的掌握相关开发工具的使用，并且为我们实验室构建了一个科研氛围浓厚且和谐、有爱的大家庭，在这样的环境中我完成了我的三年研究生学习生涯，祝你们在工作的道路上一帆风顺。

同时还要感谢我的亲朋好友对我的支持与理解，是你们给予了我坚强的后盾，特别是我的父母，在生活上对我起到了很大的帮助，在我低落的时候鼓舞我，从而能够在人生的道路上不断的成长，使我顺利地完成了学业。

最后还要感谢我的学校为我提供了科研所需要的资料和平台，从而促成了我的科研顺利完成。

参考文献

[1] 魏小淞.FPGA 加速卷积神经网络训练的研究与实现[D].硕士学位论文,西安电子科技大学 2018,06.19.

[2] LeCun Y, Bengio Y, Hinton G. Deep learning[J].Nature,2015,521(7553):436-444.

[3] Szegedy, Christian, Yangqing Jia，et al. Going deeper with convolutions[C].Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2015：1-9.

[4] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[5] 杨军，余江.基于 FPGA 的数字系统研究与设计[M].科学出版社, 2016.05.01.

[6] 张文爱.EDA 技术与 FPGA 应用设计[M].电子工业出版社, 2016.06.01.

[7] M.Zeiler,S.Zhang,Y.LeCun,et al.Regularization of neural networks using dropconnect[C].International Conference on Machine Learning,2013:1058–1066.

[8] 赵博然.FPGA 实现的可编程神经网络处理器[D].硕士学位论文，西安电子科技大学, 2018.05.01.

[9] 王金兰.基于 FPGA 的卷积人工神经网络加速方法与实现研究[D].硕士学位论文，兰州大学, 2017.09.01.

[10] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C].Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2016: 779-788.

[11] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[J]. Computer Science, 2015:2818-2826.

[12] 殷伟.基于 FPGA 的卷积神经网络并行加速体系架构的研究[D].硕士学位论文，西安电子科技大学, 2018.06.01.

[13] 王思阳.基于 FPGA 的卷积神经网络加速器设计[D].硕士学位论文，电子科技大学, 2017.05.19.

[14] Lifeng Miao, Jun Jason Zhang, Chaitali Chakrabarti, et al. Efficient Bayesian Tracking of Multiple Sources of Neural Activity: Algorithms and Real-Time FPGA Implementation[J].IEEE Transactions on Signal Processing, 2013,61(3):633-647.

[15] M. Bolic, P. M. Djuric, Sangjin Hong. Resampling algorithms and architectures for distributed particle filters [J]. IEEE Transactions on Signal Processing, 2005,53(7):2442-2450.

[16] 董振兴.基于 FPGA 平台的深度学习应用研究[D].硕士学位论文，西安电子科技大学, 2018.05.20.

[17] Sarikaya R,Hinton G E,Deoras A. Application of Deep Belief Networks for Natural Language Understanding[J]. IEEE/ACM Transactions on Audio Speech & Language Processing,2014,22(4):778-784.

[18] Ivan Sutherland,Marly Roncken,Navaneeth Jamadagni,Chris Cowan,and Swetha Mettala Gilla. The Weaver, an 8x8 Crossbar Experiment[C].ARC report,2015,11.17.

[19] R.Zhao,W.Ouyang,H.Li,X.Wang.Saliency detection by multi-context deep learning[C],CVPR,2015.

[20] Li Yanbin, Cao Zuoliang, Liu Changjie. Particle filter algorithm for target tracking based on DSP[J]. Journal of Optoelectronics Laser,2009,20(6):771-774.

[21] Roncken M,Gilla S M,Park H,et al. Naturalized Communication and Testing[J].IEEE International Symposium onAsynchronous Circuits and Systems,2015:77-84.

[22] Krizhevsky Alex, Ilya Sutskever, Geoffrey E. Hinton.Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems,2012.

[23] 余子健，马德，严晓浪，等.基于 FPGA 的卷积神经网络加速器［J］. 计算机工程，2017，43(1): 109-114，119.

[24] 方睿，刘加贺，薛志辉，等. 卷积神经网络的 FPGA 并行加速方案设计[J].计算机工程与应用，2015，51(8):32-36.

[25] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve ，et al.Target-driven visual navigation in indoor scenes using deep reinforcement learning[C].IEEE ICRA,2017:3357-3367.

[26] Alexandre Alahi, Judson Wilson, Li Fei-Fei F,et al.Unsupervised camera localization in crowded spaces[C].IEEE ICRA,2017:2666-2673.

[27] 蒋雨欣，李松斌，刘鹏，等.基于多特征深度学习的人脸性别识别[J].计算机工程与应用,2016,1(43):226-231.

[28] 李倩玉，蒋建国，齐美彬.基于改进深层网络的人脸识别算法[J].电子学报,2017，3：619-625.

[29] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C].Advances in neural information processing systems,2015: 91-99.

[30] Hasitha Muthumala Waidyasooriya， Masanori Hariyama. FPGA-based deep-pipelined architecture for FDTD acceleration using OpenCL[C]. IEEE，ICIS,2016:1-6.

[31] 王小雪.基于 FPGA 的卷积神经网络手写数字识别系统的实现[D].硕士学位论文，北京理工大学, 2016.06.01.

[32] 李泽坤.基于 FPGA 的卷积神经网络系统的设计与实现[D].硕士学位论文，哈尔滨工业大学, 2017.06.01.

[33] Akane Tahara, Yoshiki Hayashida, Theint Theint Thui, et al. FPGA-based Real-Time Object Tracking Using a Particle Filter with Stream Architecture[C]. Fourth International Symposium on Computing and Networking, 2016:422-428.

[34] Amin Jarrah, Mohsin M. Jamali, Seyyed Soheil Sadat Hosseini. Optimized FPGA based implementation of particle filter for tracking applications[C]. IEEE National Aerospace and Electronics Conference, 2014:233-236.

[35] 王尔申，范云飞，庞涛. 基于 FPGA 的粒子滤波算法研究与实现[J].微电子学与计算机,2015，32(8):58-61.

[36] 何康，陆小峰，路亨立. 一种改进的粒子滤波算法及其 FPGA 硬件实现[J].计算机工程应

用,2015,51(24):45-49.

[37] Pinalkumar Engineer, Rajbabu Velmurugan, Sachin Patkar. Parameterizable FPGA Framework for Particle Filter Based Object Tracking in Video[C]. 28th International Conference on VLSI Design, 2015:35-40.

[38] Alfonso Rodr ́guez, F ́elix Moreno. Evolutionary Computing and Particle Filtering: A Hardware-Based Motion Estimation System[J]. IEEE Transactions on Computers, 2015,64(11):3140-3152.

[39] Qing Ming, Jo Kang-hyun.A novel particle filter implementation for a multiple-vehicle detection and tracking system using tail light segmentation[J]. International Journal of Control, Automation and Systems,2013,11(3):577-587.

[40] 鲍贤亮.一种高性能 CNN 专用卷积加速器的设计与实现[D].硕士学位论文，南京大学, 2018.05.27.

[41] Lifeng Miao, Jun Jason Zhang, Chaitali Chakrabarti, et al. Efficient Bayesian Tracking of Multiple Sources of Neural Activity: Algorithms and Real-Time FPGA Implementation[J].IEEE Transactions on Signal Processing, 2013,61(3):633-647.

[42] Akane Tahara, Yoshiki Hayashida, Theint Theint Thui, et al. FPGA-based Real-Time Object Tracking Using a Particle Filter with Stream Architecture[C]. Fourth International Symposium on Computing and Networking, 2016:422-428.

[43] 乐毅.深度学习 Keras 快速开发入门[M].电子工业出版社, 2017.08.01.

[44] 王琛，胡振邦，高杰.深度学习原理与 TensorFlow 实践[M].电子工业出版社, 2016.05.01.

[45] Amin Jarrah, Mohsin M. Jamali, Seyyed Soheil Sadat Hosseini. Optimized FPGA based implementation of particle filter for tracking applications[C]. IEEE National Aerospace and Electronics Conference, 2014:233-236.

[46] Alfonso Rodr ́guez, F ́elix Moreno. Evolutionary Computing and Particle Filtering: A Hardware-Based Motion Estimation System[J]. IEEE Transactions on Computers, 2015,64(11):3140-3152.

[47] PHAN H,HERTEL L,MAASS M,et al.Robust audio event recognition with 1-max pooling convolutional neural networks[J].arXiv preprint arXiv,2016,99:168-174.

[48] Brady J.Radiation-hardened delay-insensitive asynchronous circuits for multi-bit seumitigation and data-retaining sel protection[M].Dissertations & Theses - Gradworks,2014.

[49] Fengbin Tu,Shouyi Yin,Peng Ouyang,Shibin Tang,Leibo Liu,Shaojun Wei.Deep Convolutional Neural Network Architecture With Reconfigurable Computation Patterns[J].IEEE,TVLSI,2017.

[50] K.He,X.Zhang,S.Ren,J.Sun.Delving deep into rectifiers:Surpassing human-level performance on imagenet classification[C]IEEE Conference on Computer Vision and Pattern Recognition,2015,6:1026–1034.

[51] 孙凡.卷积神经网络加速器的实现与优化[D].硕士学位论文，中国科学技术大学, 2018.05.01.

[52] C.Szegedy,et al.Going deeper with convolutions[C]IEEE Conference on Computer Vision and Pattern Recognition,2015,6:1–9.

附录

硕士期间参加的科研项目：

[1]贵州省科技厅联合基金项目《电子元器件远程检测适配器的设计实现》黔科合LH字【2014】7630；

[2]贵州省功率元器件可靠性重点实验室开发基金项目《基于可信计算的物联网感知层安全机制研究》编号：KFJJ201602；

[3]贵州省普通高等学校智能物联网工程中心建设项目，编号：黔教合KY字[2016]016；

[4]贵州大学引进人才科研项目《嵌入式系统可信计算的安全机制研究》贵大人基合字（2015）53号；

硕士期间发表论文目录：

[1]王昆，周骅.基于深度学习的实时识别硬件系统框架设计[J].电子技术应用，2018，44(10)：11-14；

[2]王昆，周骅.深度学习中的卷积神经网络系统设计及硬件实现[J].电子技术应用，2018，44(5)：56-59；

硕士期间参加的比赛：

[1] “兆易创新杯”第十三届中国研究生电子设计竞赛西南赛区二等奖；

[2] “华为杯”第十二届中国研究生电子设计竞赛西南赛区三等奖；

附：贵州大学学位论文原创性声明和使用授权声明

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究在做出重要贡献的个人和集体，均已在文中以明确方式标明。本人在导师指导下所完成的学位论文及相关的职务作品，知识产权归属贵州大学。本人完全意识到本声明的法律责任由本人承担。

论文作者签名：王昌 日 期：2019年6月11日

关于学位论文使用授权的声明

本人完全了解贵州大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权贵州大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

本学位论文属于：

保 密（ ），在____年解密后适用授权。

不保密（√）

（请在以上相应方框内打“√”）

论文作者签名：王昌 导师签名：周琴

日 期：2019年6月11日