

上海交通大学硕士学位论文

## 5G 智能通信系统中关键技术研究及实现

硕 士 研 究 生：宫宇

学 号：117034910076

导 师：宫新保

申 请 学 位：工学硕士专业学位

学 科：信息与通信工程

所 在 单 位：电子信息与电气工程学院

答 辩 日 期：2020 年 1 月

授予学位单位：上海交通大学

Dissertation Submitted to Shanghai Jiao Tong University  
for the Degree of Master

**RESEARCH AND IMPLEMENTATION ON KEY  
TEHCHNOLOGIES IN 5G INTELLIGENT  
COMMUNICATION SYSTEM**

<b>Candidate:</b>	Gong Yu
<b>Student ID:</b>	117034910076
<b>Supervisor:</b>	Gong Xinbao
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Electronic and Communication Engineering
<b>Affiliation:</b>	School of Electronic Information and Electrical Engineering
<b>Date of Defence:</b>	Jan, 2020
<b>Degree-Confering-Institution:</b>	Shanghai Jiao Tong University

# 上海交通大学

## 学位论文原创性声明

本人郑重声明：所呈交的学位论文《5G 智能通信系统中关键技术研究及实现》，是本人在导师的指导下，独立进行研究工作所取得的成果。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作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：        年    月    日

# 上海交通大学

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并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。本人授权上海交通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、缩印或扫描等复制手段保存和汇编本学位论文。

保密☐，在\_\_\_年解密后适用本授权书。

本学位论文属于

不保密☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期：      年    月    日

日期：      年    月    日

## 5G 智能通信系统中关键技术研究及实现

### 摘 要

在 5G 通信系统中，一方面需要扩大带宽以提供更高的通信质量和更低的通信延迟，另一方面需要与人工智能结合以支撑更广泛的应用场景。然而，宽带通信会大大增加 5G 系统中数字多波束合成的计算复杂度和信号处理复杂度；另外，以深度学习为代表的人工智能技术的计算复杂度很高，很难直接部署在 5G 通信的边缘应用场景；这些都大大增加了 5G 智能通信系统的实现难度。针对上述问题，本文提出了一种宽带数字多波束合成的实现方案，并提出了一种高速率、高吞吐率的深度神经网络加速方案，在此基础上设计并实现了一套适用于 5G 智能通信系统的高速数字信号处理硬件平台。

本文的主要研究内容包括：

（1）设计了一种适用于大带宽 5G 智能通信系统的低延迟低复杂度和多通道可扩展的数字多波束合成技术方案。该技术方案基于先抽取再滤波和多相滤波的方法，对传统数字波束合成方案做了改进，降低了系统的计算复杂度和延迟。该技术方案相比于传统的数字波束合成方案，能够在处理大带宽的高速信号时达到 100% 的数据吞吐率，在 8 通道合成 2 波束时，相较分数延迟滤波器算法，计算复杂度只有其 47.9%；在单通道单波束时，延迟只有其 64.53%。并且随着通道数目和合成波束数目的增加，延迟和计算复杂度的优势更加明显。

（2）设计了一种低延迟、实时的、大数据吞吐率的神经网络硬件加速方案。该技术方案基于 FPGA 平台，通过对乘法器的合理分配和乘法器分时复用实现了多通道卷积核并行计算。同时对池化层和全连接层的实现做了优化，降低了计算延迟的同时，保证了工程的可实现，实现了神经网络推断的实时性。对每层网络单独的定点量化策略保证

了计算的精度和结果的准确性。实验表明，本方案的神经网络推断可以达到 95.4% 的准确率，且速度可以达到 1060FPS，远高于 CPU，略低于 GPU。同时由于 FPGA 功耗低的特性，本方案推断每张图的功耗仅为 0.01188W，远远低于 GPU 和 CPU。

(3) 设计并实现了一套适用于大带宽 5G 智能通信系统的数字信号处理硬件平台。该硬件平台采用模块化的设计方式，支持阵列通道数和数字信号处理、传输能力的扩展。单个模块集成了 8 个 1Gsp/s 采样率的 ADC 通道。该平台以 FPGA 作为数字信号处理的核心，不仅提供了对多通道宽带数字信号的并行高速实时处理能力，而且其强大的可编程性也支持数字信号处理算法的不断迭代和升级。此外该平台还配备了两个 6.25Gbps QSFP+ 光收发器、千兆以太网和 PCIe 高速数据传输接口用于基带信号的传输。该硬件平台很好地满足了大带宽 5G 智能通信系统对信号的接收、处理和传输的需求，同时也为数字信号处理算法的实现提供了良好的硬件基础。

**关键词：**宽带数字多波束合成，第五代移动通信技术，神经网络，FPGA

# **RESEARCH AND IMPLEMENTATION ON KEY TECHNOLOGY IN 5G INTELLIGENT COMMUNICATION SYSTEM**

## **ABSTRACT**

In the 5G communication system, on the one hand, it is necessary to expand the bandwidth to provide higher communication quality and lower communication delay, and on the other hand, it needs to be combined with artificial intelligence to support a wider range of application scenarios. However, broadband communication will greatly increase the computational complexity and signal processing complexity of digital multi-beam synthesis in 5G systems. In addition, the computational complexity of artificial intelligence technologies represented by deep learning is very high, and it is difficult to directly deploy on the edge application scenarios of 5G communication. These have greatly increased the difficulty of implementing 5G intelligent communication systems. Aiming at the above problems, this paper proposes a broadband digital multi-beam forming implementation scheme, and proposes a high-speed, high-throughput deep neural network acceleration scheme. Based on this, a set of high-speed digital signal processing hardware platform suitable for 5G intelligent communication system is designed and implemented.

The main research contents of this article include:

(1) A low-latency, low-complexity and multi-channel scalable digital multi-beam forming technology solution suitable for large-bandwidth 5G intelligent communication systems is designed. This technical solution is based on the methods of first-sampling-then-filtering and polyphase filtering. It improves the traditional digital beam combining scheme and reduces the computational complexity and delay of the system. Compared with the

traditional digital beam forming scheme, this proposed scheme can achieve a 100% data throughput rate when processing high-speed signals with large bandwidths. When forming 2 beams in 8 channels, the delay is only 64.53% compared with the fractional delay filter algorithm, the calculation complexity is only 47.9%, and the advantages become more obvious as the number of channels and the number of beams increase.

(2) A low-latency, real-time, large-data throughput neural network hardware acceleration scheme is designed. The technical solution is based on the FPGA platform, and multi-channel convolution kernels are calculated in parallel by rational allocation of multipliers and time-division multiplexing of multipliers. At the same time, the implementation of the pooling layer and the fully connected layer is optimized, while reducing the calculation delay, ensuring the realization of the project, and realizing the real-time performance of neural network inference. A separate parameter quantization strategy for each layer of the network ensures the accuracy of the calculation and the accuracy of the results. Experiments show that the neural network inference of this scheme can reach 95.4% accuracy, and the speed can reach 1060FPS, which is much higher than the CPU and slightly lower than the GPU. At the same time, due to the low power consumption of FPGA, the power consumption of each image is only 0.01188W, which is far lower than GPU and CPU.

(3) A digital signal processing hardware platform suitable for large-bandwidth 5G intelligent communication systems is designed and implemented. The hardware platform adopts a modular design, and supports the expansion of the number of array channels and digital signal processing and transmission capabilities. A single module integrates eight ADC channels with 1Gsps sampling rate. The platform uses FPGA as the core of digital signal processing, which not only provides parallel high-speed real-time processing capabilities for multi-channel wideband digital signals, but also its strong programmability supports continuous iteration and upgrade of digital signal processing algorithms. In addition, the platform is also



equipped with 2.5Gbps QSFP+ optical transceivers, Gigabit Ethernet and PCIe high-speed data transmission interfaces for the transmission of baseband signals. The hardware platform satisfies the needs of large-bandwidth 5G intelligent communication systems for signal reception, processing, and transmission, and also provides a good hardware foundation for the implementation of digital signal processing algorithms.

**Key Word:** Broadband Digital Multi-Beam Forming, 5G, Neural Network, FPGA

# 目 录

第一章 绪 论 .....	1
1.1 选题的目的和意义 .....	1
1.2 波束合成与神经网络边缘计算加速研究现状和发展趋势 .....	2
1.3 本文章节安排 .....	3
第二章 波束合成与神经网络加速技术概述 .....	5
2.1 波束合成技术概述 .....	5
2.1.1 波束合成基本原理 .....	5
2.1.2 模拟波束合成技术 .....	7
2.1.3 数字波束合成技术 .....	8
2.2 神经网络边缘计算硬件加速技术概述 .....	9
2.2.1 神经网络简介 .....	9
2.2.2 神经网络的边缘计算 .....	11
2.3 本章小结 .....	14
第三章 低复杂度的多波束合成技术方案 .....	15
3.1 低复杂度数字波束合成系统设计 .....	15
3.2 低复杂度数字波束合成系统实现优化 .....	18
3.3 系统方案分析 .....	25
3.3.1 子信道数目 $K$ 分析 .....	25
3.3.2 系统计算复杂度分析 .....	25
3.3.3 系统延迟分析 .....	28
3.4 波束合成分析 .....	30
3.4.1 覆盖范围分析 .....	30
3.4.2 旁瓣抑制分析 .....	31
3.5 本章小结 .....	34
第四章 神经网络边缘计算加速 .....	36
4.1 基于 FPGA 的神经网络硬件加速系统架构设计 .....	36
4.2 基于 FPGA 的神经网络硬件加速系统实现优化 .....	40
4.2.1 卷积层 .....	41

4.2.2 池化层 .....	47
4.2.3 全连接层 .....	48
4.3 定点量化策略 .....	51
4.4 系统性能分析 .....	52
4.5 本章小结 .....	56
<b>第五章 多通道宽带数字信号处理硬件平台设计与实现 .....</b>	<b>58</b>
5.1 系统设计目标 .....	58
5.2 系统硬件设计 .....	59
5.3 模块化单板硬件设计 .....	60
5.3.1 单板硬件设计 .....	60
5.3.2 接收功能设计 .....	62
5.3.3 数字信号处理功能设计 .....	63
5.3.4 基带信号传输功能设计 .....	64
5.4 系统性能测试及分析 .....	66
5.5 本章小结 .....	68
<b>第六章 总结与展望 .....</b>	<b>69</b>
6.1 本文总结 .....	69
6.2 未来工作展望 .....	70

## 缩略语对照表

ADC(Analog-to-Digital Converter)	模/数转换器
DDR(Double Data Rate)	双倍数据速率
DSP(Digital Signal Processor)	数字信号处理器
FFT(Fast Fourier Transform)	快速傅里叶变换
FPGA(Field-Programmable Gate Array)	现场可编程门阵列
IEEE(Institute of Electrical and Electronic Engineers)	电子电气工程师协会
IFFT(Inverse Fast Fourier Transform)	逆向快速傅里叶变换
ITU(International Telecommunication Union)	国际电信联盟
I/O(Input/Output)	输入/输出
MAC(Media Access Control)	介质访问控制
MAX(MAXimum)	最大值
MIMO(Multiple-Input Multiple-Output)	多输入多输出
MIN(MINimum)	最小值
PLL(Phase-Locked Loop)	锁相环
RGMIIR(Reduced Gigabit Media Independent Interface)	精简千兆介质无关接口
SNR(Signal-to-Noise Ratio)	信噪比
SoC(System on Chip)	片上系统
WLAN(Wireless Local Area Network)	无线局域网

## 第一章 绪论

### 1.1 选题的目的和意义

中国 5G 网络通讯研究实验的第一阶段和第二阶段分别于 2016 年 9 月和 2017 年 9 月完成。进一步的，中华人民共和国工业和信息化部于 2017 年 11 月正式宣布启动 5G 网络通讯技术研究发起实验的第三阶段进程，并且第三阶段实验的测试于 2019 年 1 月顺利完成。在即将到来的 2020 年，5G 将会全面商用。5G 网络在理论上，频带可达 6GHz 以上，带宽可达几 GHz 甚至十几 GHz，速度甚至可达 10Gb/s<sup>[1]</sup>，除了高下载速率以外，5G 网络同时也具有低时延和高可靠的特点，更加智能、开放和灵活<sup>[2]</sup>，成为新一代科技热潮的基础和领头羊。

人工智能是当下最尖端最有潜力的新技术，融合了生物学、计算机、大数据等多种科学，是一门综合性极强的跨领域学科<sup>[3]</sup>。人工智能可以凭借计算机本身的学习能力对海量冗繁的数据进行分析、计算和分类整理，并且从中获取知识信息，获取经验，从而不断完善数据库和学习能力。其本身短时间数据传输、整合、运算以及其对于庞大数据库的要求<sup>[4]</sup>，和 5G 带宽大、速度快、时延小、可靠性高的特点不谋而合。同时，人工智能的发展可以更好地规划通信网络<sup>[5]</sup>，提高网络的传输效率。因此 5G 和人工智能之间相辅相成，彼此协助，互为应用。5G 时代的通信系统一定是与人工智能相结合的 5G 智能系统。

但是 5G 智能系统的大规模推广和商用仍有许多的阻碍。一方面是 5G 通信系统本身仍有不足，大带宽和大数据吞吐导致波束合成处理延迟过高，不满足 5G 通信的要求；另一方面，庞大数据导致人工智能的运算量非常大，延迟很高，无法满足智能系统中的实时性。

为了解决上述存在的问题，一种应对波束合成处理高延迟的方法是将模拟波束合成和数字波束合成相结合<sup>[6][7][8]</sup>：使用相控阵列对模拟波束做预处理，然后对预处理过的波束做数字波束合成。这种技术手段可以增加覆盖距离，提高频谱效率和用户容量。然而在 5G 通信中采用了大规模 MIMO 阵列，通道多，带宽大，再加上多波束的需求，这种技术手段仍然有计算复杂和部署成本高的特点。因此，研究适用于 5G 智能通信系统的低延迟、低复杂度数字多波束合成技术具有重要的意义。

解决 5G 智能系统中人工智能计算时延高的问题的手段是利用边缘计算，将

数据处理从云端转移到客户端<sup>[9][10][11]</sup>。然而，传统硬件平台的加速技术不能满足 5G 智能通信系统低延迟、低功耗的要求：CPU 结构具有先天的劣势，而 GPU 功耗大，笨重且难以工程实现。因此，研究一种适合 5G 智能通信系统的神经网络边缘计算硬件加速技术具有重要意义。

## 1.2 波束合成与神经网络边缘计算加速研究现状和发展趋势

数字波束合成技术能够实现射频信号功率在地面的灵活分配，且利用收/发数字波束合成技术提供的空域-时域-频域三维信息，提升通信系统性能，已成为一种发展趋势。在无线通信技术中，波束形成的研究是重要的研究方向之一。

在 WLAN 技术标准 IEEE 802.11n/ac<sup>[12]</sup>以及移动网络技术标准 LTE/LTE-A<sup>[13]</sup>中，多通道的信号传输已经获得支持。由于在 5G 通信中，应用毫米波大规模 MIMO 使得天线的数目大幅增长，较高复杂度的数字波束合成技术已经不再适合工程应用，因此设计一款复杂度较低的数字波束合成方案是当下波束合成的重点研究方向。文献[14]给出了一种基于最小二乘法的分数延迟滤波器设计，对分数延迟滤波器算法进行了优化，使滤波器的精度更高、计算复杂度更低，并且基于分布式算法在 FPGA 上做了工程实现， $f_{max}$ 达到了 83.98MHz。文献[15]设计了一种低复杂度、高精度的可变分数延迟滤波器，可以灵活调整滤波器的截止频率，其设计复杂度相比加权最小二乘法设计法降低了 1 个数量级。因此，在保证精度的同时，未来的波束合成算法以及工程实现中，低复杂度和低延迟是一个不可阻挡的趋势。

随着高带宽、低延迟 5G 网络的到来，各类新兴的产业如智能城市、智能家居、虚拟现实/增强现实（VR/AR）等万物互联的网络也渐渐受到学术界和工业的关注<sup>[16]</sup>。这些新兴产业应用所产生的数据量将会呈爆炸式增长，对数据计算的实时性、网络依赖性和数据安全性提出了更高的要求<sup>[17]</sup>。为了解决这些问题，国内外学者们提出了边缘计算的概念。边缘计算的“边缘”与“云计算”的概念相对立，指的是在数据源与云端中心之间的任何计算及网络资源。边缘计算实现了物联网技术前所未有的连接性、集中化和智能化，是计算系统从扁平到边缘，以及面向 5G 网络架构演进的必然技术。边缘计算可以满足数据和应用之间敏捷连接、实时业务、安全与隐私保护等方面的需求。

目前边缘计算提供了一种新的生态系统和价值链。第三方数据分析机构 IDC 预测，到 2020 年，全球将有约 500 亿的智能设备接入互联网，其中主要涉及智能手机、可穿戴设备、个人交通工具等，其中 40% 的数据需要边缘计算服务。边缘

计算有着强大市场潜力，也引起了各研究机构、标准组织、服务提供商和产业界极大的关注。在工业上，2017 年，亚马逊携 AWS Greengrass<sup>[18]</sup>进军边缘计算领域，走在了行业的前面。微软发布了 Azure IoT Edge<sup>[19]</sup>解决方案，该方案“将云分析扩展到边缘设备”，支持离线使用。谷歌 2017 年已发布硬件芯片 Edge TPU 和软件堆栈 Cloud IoT Edge<sup>[20]</sup>，意在帮助改善边缘联网设备的开发。文献[21]设计了基于 FPGA 的卷积神经网络加速方案，定量分析其计算吞吐量和所需的内存带宽，给出了 FPGA 资源需求最低的解决方案，实现了在 100MHz 工作频率下 61.62 GFLOPS 的峰值性能，明显优于以前的方法。文献[22]基于并发和流水线的方法加速 Viola-Jones 人脸检测算法，提高了数据吞吐量，通过并发卷积操作和流水线特征图加速了卷积神经网络计算过程，实现了 2.9 倍的加速比。因此不管在学术还是在工业上，边缘计算都是未来智能系统的研究方向。

### 1.3 本文章节安排

本文各章节的主要内容安排如下：

第二章介绍了波束合成技术与神经网络硬件加速的概况，包括基本概念、基本原理和在工程实践中常用的技术手段。

第三章提出了一种适用于 5G 智能通信系统的大带宽、低延迟、低计算复杂度且可扩展的多波束数字波束合成技术方案。首先提出了一种针对 5G 大带宽智能通信系统的数字波束合成的系统设计。接着，对该系统方案的硬件实现做了优化，降低了数字波束合成技术在处理大带宽和大数据吞吐量时的计算复杂度。然后，论证分析了本文提出的数字波束合成系统的计算复杂度和系统延迟，以及子信道数目参数  $K$  对系统的影响。最后分析了合成的多波束在覆盖范围、波束间的正交性与旁瓣抑制的关键性能。结果表明，本文所设计的适用于 5G 智能通信系统的多波束数字波束合成技术方案不仅在计算复杂度、计算延迟、实时性方面具有优势，并且具有高旁瓣抑制和覆盖范围广的特点。

第四章提出了一种设计了一种低计算复杂度的、低延迟的、大数据吞吐率的神经网络硬件加速方案。首先，给出了神经网络硬件加速的系统设计。然后，详细介绍了提出的神经网络硬件加速方案中卷积层、池化层和全连接层硬件加速的关键技术。接着，给出了该方案中从浮点数到定点数的量化策略。最后，对该系统方案的系统时延、功耗等方面做了分析，并通过计算机仿真证明了本文提出的神经网络硬件加速方案的准确率可达 95.4%。结果表明，本文所设计的神经网络

硬件加速方案不仅系统延迟低,处理速度快,且效能仅为 0.01188W,优于传统 CPU、GPU 计算平台,而且由于其他 FPGA 边缘计算加速架构,易于工程实现。

第五章设计并实现了一套适用于 5G 智能通信系统的多通道宽带数字信号处理硬件平台。首先,讨论分析了 5G 智能通信系统对数字信号处理硬件平台的功能与性能的需求,并在此基础上制定了硬件平台的总体设计目标。然后,根据设计目标给出了一种模块化、可扩展的系统硬件设计。接着,进一步给出了单个模块的硬件设计。最后,对硬件平台的性能进行了分析和总结。结果表明,本文所设计的数字信号处理硬件平台具备多通道和大带宽的特点,不仅很好地满足了 5G 智能通信系统对信号的接收、处理和传输的功能需求,而且也为数字信号处理算法的实现提供了良好的硬件基础。

第六章总结了本文的主要内容并展望了进一步的工作。本章概括了本文在多波束数字波束合成系统设计、神经网络硬件加速系统设计以及数字信号处理硬件平台设计与实现上的工作与成果;此外还展望了在本文工作的基础上进一步的研究方向。



## 第二章 波束合成与神经网络加速技术概述

波束合成技术是一种利用传感器阵列定向发射/接收信号的信号处理技术。波束合成技术通过控制传感器阵列中每一个单元上的信号的幅度和相位，在空间上形成干涉相长和相消，从而在不同方向上获得不同的增益，增强信号发射/接收的指向性，抑制噪声和干扰，提升了无线通信系统的信道容量和频谱效率，是 5G 通信系统中必不可少且极为重要的一部分。

神经网络通过训练之后可以实现多种非常复杂的任务，是当前人工智能中应用最广泛的技术。在 5G 智能通信系统中，人工智能也扮演越来越重要的角色，神经网络的应用必不可少。而在 5G 智能通信系统中，大带宽带来了更大的数据量，同时实时性和低延迟对高计算复杂度的神经网络提出了新的要求。

本章 2.1 节介绍了波束合成技术的基本概念及原理，建立了波束合成的信号处理模型，并分别介绍了传统的模拟和数字波束合成技术及其特点；2.2 节介绍了神经网络的基本概念及原理，并分别介绍了神经网络在现有硬件平台的表现和特点；2.3 节对本章的主要内容进行了总结。

### 2.1 波束合成技术概述

#### 2.1.1 波束合成基本原理

波束合成是利用传感器（例如天线单元）阵列建立系统的空间辐射图来进行信号的定向发送或接收的信号处理技术。波束合成过程的本质是电磁波在空间中的不同方向上形成干涉相长或相消，这个过程通过调整每个传感器单元接收到的电磁波的幅度和相位来实现。波束合成的结果是使信号在空间域中在不同的方向上有出不同的强度，形成波束图。因此，波束合成技术在本质上构建了一个信号空间域的滤波器，利用合成的波束图实现空间滤波。波束合成技术提高了系统的空间选择性，增强了所需要信号而抑制了不希望有的信号，从而提高系统的功能和性能。

在波束合成中，最简单的模型是均匀线形天线阵列。图 2-1 是均匀线形天线阵列波束合成的示意图。设阵列单元数为  $N$ ，则  $N$  个天线单元以  $d$  为间距排成一条直线，所需信号的入射方向与线阵的法向量方向的夹角为  $\theta_0$ ，入射波的入射方

向与线阵的法向量方向的夹角为 $\theta$ ，载波的波长为 $\lambda$ ，入射信号的复包络为 $a$ ，为了避免合成波束中栅瓣的出现，设定 $d \leq \frac{\lambda}{2}$ 。忽略掉接收单元和通道的幅度和相位不平衡以及噪声，第 $i$ 个阵列单元的接收信号可以表示为：

$$x_i = ae^{-j(i-1)\frac{2\pi}{\lambda}\sin\theta} = ae^{-j(i-1)\beta} \quad i = 1, 2, 3, \dots, N \quad (2-1)$$

然后通过相位补偿得到相应合成波束。

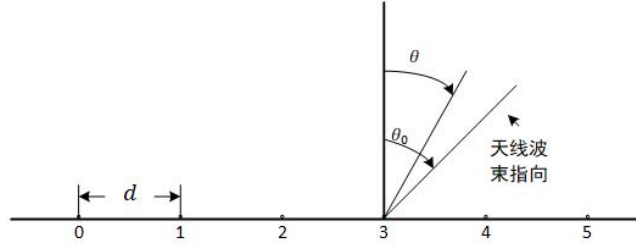


图 2-1 N 单元线阵波束合成

Figure 2-1 N-unit linear array beamforming

在实际工程中，天线阵列多为平面阵列，其结构相比曲面阵更简单，应用也更为广泛。如图 2-2 是矩形平面阵列波束合成坐标图。在坐标系 $O - xyz$ 中，设有 $M \times N$ 个阵元按照如图 2-2 的矩形形式排列， $x$ 、 $y$ 方向的阵元间距分别为 $d_x$ 、 $d_y$ ，俯仰角为 $\theta$ ，方位角为 $\phi$ 的波束，则在直角坐标系中，该波束的指向可以用一个三维单位矢量 $\vec{r}$ 表示：

$$\vec{r} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta) \quad (2-5)$$

阵列中位置为 $(0,0)$ 的参考辐射单元到位置为 $(m,n)$ 的辐射单元的距离矢量 $\vec{l}$ 表示为：

$$\vec{l} = (md_x, nd_y, 0); \quad m = 0, 1, \dots, M-1; \quad n = 0, 1, \dots, N-1 \quad (2-6)$$

位置为 $(m,n)$ 的辐射单元发射的电磁波与位置为 $(0,0)$ 的参考辐射单元发射的电磁波的波程差 $\Delta x$ 为距离矢量 $\vec{l}$ 在波束指向 $\vec{r}$ 上的投影的长度：

$$\Delta x = \vec{l} \cdot \vec{r} = md \sin \theta \cos \phi + nd \sin \theta \sin \phi \quad (2-7)$$

相应的相位差 $\Delta\varphi$ 为：

$$\Delta x = \vec{l} \cdot \vec{r} = md \sin \theta \cos \phi + nd \sin \theta \sin \phi \quad (2-8)$$

然后通过相位补偿得到相应合成波束。

在波束合成中，根据波束合成的技术，可以分为模拟波束合成和数字波束合成。

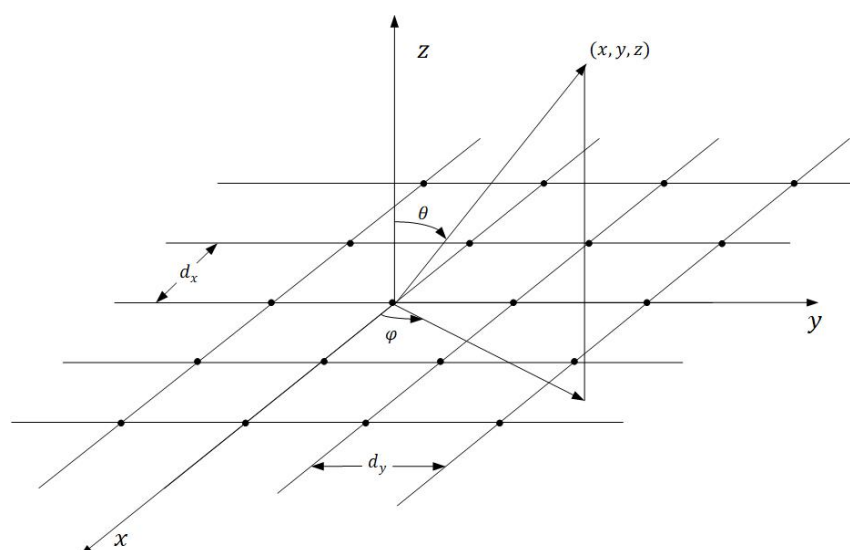


图 2-2 矩形天线平面阵列  
Figure 2-2 Rectangular antenna planar array

### 2.1.2 模拟波束合成技术

模拟波束合成是在射频（RF）或者中频（IF）中构建波束合成器，实现对阵列中各个辐射单元上的模拟信号的幅相补偿，然后经过 A/D 转换将波束合成的输出数字化。如图 2-3 所示，在模拟波束合成技术中，各天线单元的接受信号通常使用可变增益放大器实现幅度补偿，使用可调的移相器、延时线或变频本振实现相位补偿。

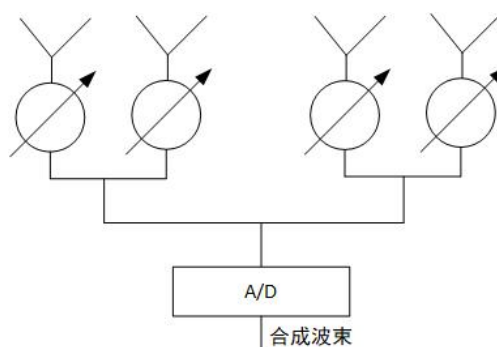


图 2-3 模拟波束合成结构  
Figure 2-3 Analog beamforming structure

模拟多波束合成的结构更加复杂，有三种基本类型：反射式、投射式和直接辐射式<sup>[23]</sup>。反射式多波束合成包括 Blass 矩阵等方式，Blass 矩阵方法结构包含定向耦合器、传输线和电阻负载，利用传输线的长度来扫描相位，利用定向耦合器来

激励产生实时延迟，从而生成多个波束。但是这种方法会引起网络损耗，同时需要大量的硬件设备。Dome 透镜属于透射式模拟多波束合成方式，由一个平面相控阵和一个半球形的 Bootlace 透镜组成，透镜折射平面阵列的扫描波束，获得更大覆盖范围，因此可以有更灵活的设计条件，但是系统非常笨重。直接辐射式，如 McFarland 二维矩阵，可以得到更好的可靠性、较少的泄露以及更少的口面遮挡，但是质量很大，制造工艺复杂。

### 2.1.3 数字波束合成技术

数字波束合成技术是一种目前最先进的相控雷达波束合成和控制技术。在数字波束合成系统中，每一个天线单元都有一组数据转换器，其相位调整是在数字信号器单元或者 FPGA 中完成。在数字波束合成系统中，每个天线单元或者每一个子阵都将射频信号数字化，然后通过数字信号处理例如加权、组合形成所需要的单个波束或者多个波束。一种最简单的数字波束合成如图 2-4 所示。

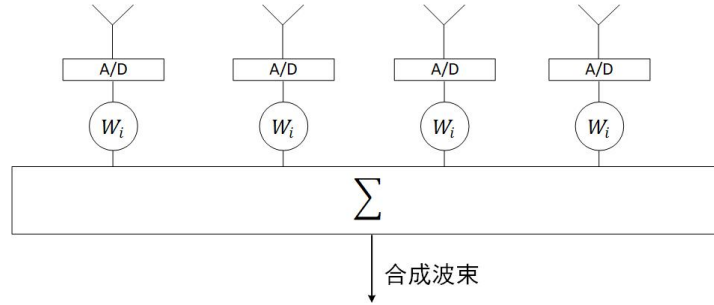


图 2-4 数字波束合成结构

Figure 2-4 Digital beamforming structure

以均匀线阵单波束为例，根据 (2-1) 式，各阵列单元接收信号经过加权矢量  $W_i$  加权之后，进行波束合成，则合成后的波束为

$$Y = \sum_{i=1}^N X_i W_i = \sum_{i=1}^N a e^{-j(i-1)\beta} W_i \quad (2-9)$$

假设权值为等幅，即有

$$W_i = e^{j(i-1)\frac{2\pi d}{\lambda} \sin \theta_0} = e^{j(i-1)\beta_0} \quad (2-10)$$

其中， $\beta_0 = \frac{2\pi d}{\lambda} \sin \theta_0$ 。则有

$$Y = \sum_{i=1}^N a e^{-j(i-1)\beta} \cdot e^{j(i-1)\beta_0} = a e^{-j\frac{N-1}{2}(\beta-\beta_0)} \cdot \frac{\sin \frac{N}{2}(\beta-\beta_0)}{\sin \frac{1}{2}(\beta-\beta_0)} \quad (2-11)$$

对其取绝对值，做归一化，可得：

$$|Y| = \left| \frac{\sin \frac{N}{2}(\beta - \beta_0)}{N \sin \frac{1}{2}(\beta - \beta_0)} \right| \quad (2-12)$$

根据式子可知，在 $\beta = \beta_0$ ，即 $\theta = \theta_0$ 处，数字波束合成的输出取最大值，此时波束指向为 $\theta_0$ ，另外可以得知，只要控制权值，就可以控制所合成波束的指向。

数字波束合成有极强的灵活性，可以轻松传输多条波束，并且可以改变波束的数量，灵活便利<sup>[24]</sup>。这种灵活性在许多应用中有巨大的吸引力。但是数字波束合成系统中每一个天线单元都要配备一个数据转换器，由于数据转换器功耗大，所以整个系统的功耗比较大。此外，由于在 5G 通信系统中，大带宽大数据对数据转换器提出了更高的要求，数据转换器必须适合更高的频率和更大的数据精度。且 5G 通信的大带宽数据也大大增加了系统的计算复杂度和数据吞吐量，波束合成处理器的计算复杂度陡升，延迟增加，功耗进一步增大。

## 2.2 神经网络边缘计算硬件加速技术概述

### 2.2.1 神经网络简介

神经网络是现在深度学习的基础，也是人工智能中重要的机器学习技术，神经网络通过模拟大脑中的神经元以达到机器学习的目的。而在 2012 年，多伦多大学 Geoff Hinton 实验室提出的 AlexNet 网络更是具有历史意义<sup>[25]</sup>，开启了深层卷积神经网络，把沉寂了多年的深度学习带入了新的大爆发。

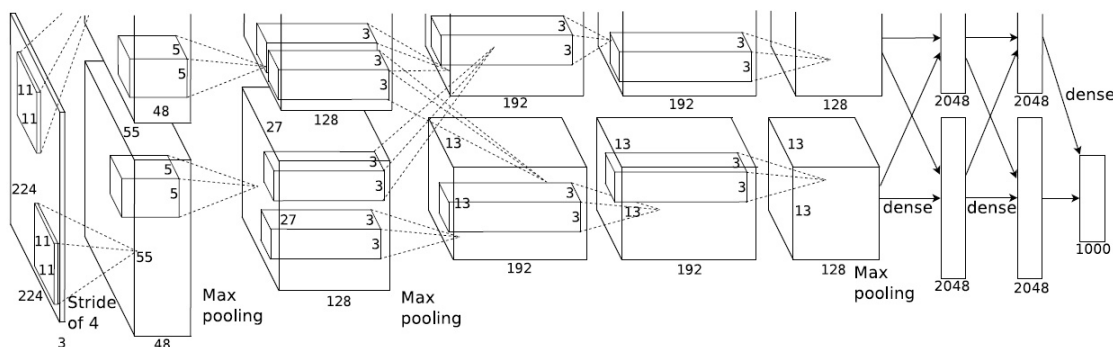


图 2-5 AlexNet 网络结构  
Figure 2-5 AlexNet network structure

图 2-5 是 AlexNet 的网络结构。AlexNet 一共分为 8 层，其中前 5 层为卷积层，后 3 层为全连接层。下面介绍每一层的计算过程。

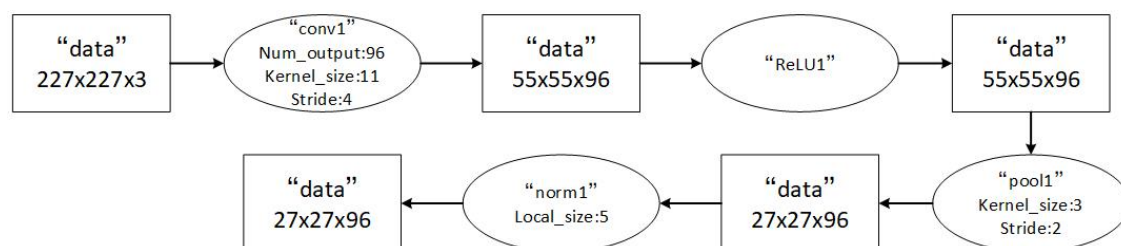


图 2-6 AlexNet 第 1 层卷积层计算流程

Figure 2-6 AlexNet 1<sup>st</sup> convolutional layer calculation process

第 1 层卷积层的输入为尺寸为  $224 \times 224 \times 3$  的原始图像，在进入卷积之前补零变成  $227 \times 227 \times 3$  大小的图像，然后与 96 个尺寸为  $11 \times 11 \times 3$  的卷积核做卷积，第 1 层的滑窗的步长为 4 个像素点，因此卷积核经过移动计算之后得到的每一个二维平面的特征图大小为  $(227 - 11)/4 + 1 = 55$  个像素点，第 1 层卷积层卷积之后得到的特征图大小为  $55 \times 55 \times 96$ 。随后卷积之后的特征图经过 ReLU 单元进行非线性激活处理，得到大小仍为  $55 \times 55 \times 96$  的激活像素点。ReLU 之后的像素点经过池化层作做最大池化处理，池化层的尺寸为  $3 \times 3$ ，步长为 2，因此每一个二维平面得到的像素点的大小为  $(55 - 3)/2 + 1 = 27$  个像素点，因此池化后的像素点的大小为  $27 \times 27 \times 96$ 。池化后的特征图像素层再进行归一化处理，归一化处理的尺寸大小为  $5 \times 5$ ，处理结果的像素规模不发生变化。因此在第一层卷积层之后的输出为  $27 \times 27 \times 96$ 。AlexNet 第一层计算的过程如图 2-6 所示。

对于 AlexNet 中的第 2、3、4、5 卷积层，运算过程也是相似的，只是输入图像的尺寸和卷积核大小数目不同。另外，第三层和第四层没有池化层。表 2-1 给出了各个卷积层输入图像的大小、卷积核的大小和数目以及输出的特征图像的大小。

表 2-1 卷积层处理特征

Table 2-1 Processing characteristics of convolutional layer

卷积层数	输入尺寸	卷积核尺寸	卷积核个数	池化层尺寸	池化层输出尺寸	输出尺寸
第 1 层	224x224x3	11x11x3	96	3x3	27x27x96	27x27x96
第 2 层	27x27x96	5x5x96	256	3x3	13x13x384	13x13x384
第 3 层	13x13x256	3x3x256	384	无	13x13x384	13x13x384
第 4 层	13x13x384	3x3x384	384	无	13x13x384	13x13x384
第 5 层	13x13x384	3x3x384	256	3x3	6x6x256	6x6x256

第 6 层是全连接层，其数据输入是第 5 层的输出。在第 6 层中，一共由 4096 个卷积核，每个卷积核的尺寸为  $6 \times 6 \times 256$ 。在全连接层，卷积核中的每一个系数都只与输入尺寸中的一个像素点一一对应相乘，因此卷积核相乘之后的输出是

$4096 \times 1 \times 1$ ，即有 4096 个神经元。之后，这 4096 个数据再通过 ReLU 函数，生成 4096 个值。全连接层的处理过程如图 2-7 所示。

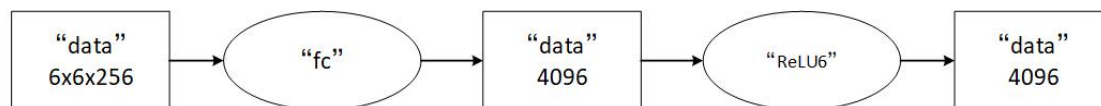


图 2-7 第 6 层全连接层处理过程

Figure 2-7 6<sup>th</sup> layer fully connected layer processing

第 7 层和第 8 层和第 6 层的处理过程大致相似，除了输入尺寸和卷积核尺寸不同。表 2-2 给出了各个全连接层的处理特征。

表 2-2 全连接层处理特征

Table 2-2 Processing characteristics of fully connected layer

全连接层	卷积核尺寸	卷积核个数	输出尺寸
第 1 层	6x6x256	4096	1x4096x1
第 2 层	4096x1x1	1	1x4096x1
第 3 层	4096x1000	1	1x1000x1

## 2.2.2 神经网络的边缘计算

第五代通信技术带来的低延迟和大带宽大数据给多客户端的及时通讯带来了可能，也推动了物联网的发展。因此，5G 时代也代表着物联网时代的到来，彼时将会有无数的物联网设备接入网络。在 5G 智能通信系统中，这些物联网设备将是智能设备。传统的智能领域在通信系统的设计是由服务器和客户端组成的：客户端负责收集图片并上传到服务器，服务器负责对图片运用人工算法进行分析。但是随着客户端的指数型爆发增长，客户端所产生的数据量也随着加大，如果这些数据全都交由云端服务器来处理，将会造成网络传输和服务器端的巨大压力。同时，设备之间的性能不同会导致实时的协同工作难以保证，且数据泄露的风险也随之加大。而在 5G 智能通信系统中，在图书馆、自动驾驶、智慧城市这样的应用场景下，边缘数据迎来了爆发性的增长，为了解决面向数据的传输、计算和存储中的计算负载，以及数据传输带宽等问题，边缘计算，即客户端的处理能够解决网络堵塞、服务器计算压力大和数据安全等问题，将是下一个重点研究的问题。在神经网络的计算中，目前主流的硬件平台有 CPU、GPU 和 FPGA。

神经网络的计算在不同层之间保持了高度的独立性，各层的网络计算独立，层



间无反馈的数据，因此，神经网络也是一种高度并行的网络结构。而如图 2-8 CPU 的结构图所示<sup>[26]</sup>，CPU 的硬件结构决定了 CPU 能更好的解决逻辑处理和事务处理，而繁多复杂的控制指令和寻址操作难以满足神经网络在边缘计算中的实时性，且不满足实际应用的小型化的需求。

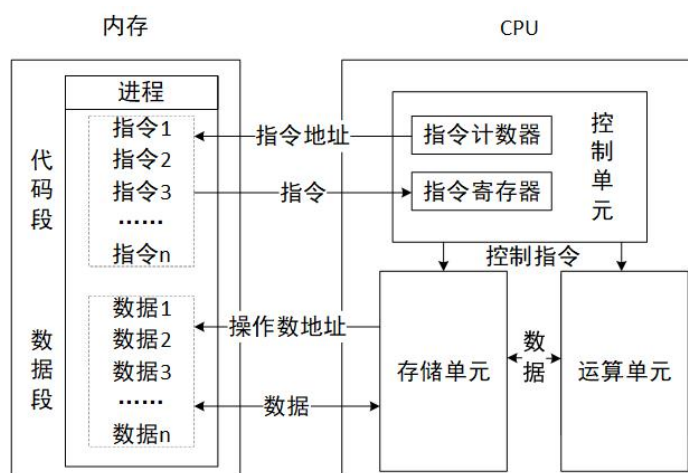


图 2-8 CPU 结构图

Figure 2-8 CPU structure

当前的神经网络加速器的研究很多在 GPU 上展开。GPU 的 MIMD（多数据多指令）结构上有先天的计算优势<sup>[27]</sup>，有多核并行计算的基础结构，且核心数非常多，可以支撑大量数据的并行计算，且拥有更高的访存速度和更高的浮点运算能力。GPU 进行数据处理的过程可以描述成：GPU 从 CPU 处得到数据处理的指令，把大规模、无结构化的数据分解成很多独立的部分然后分配给各个流处理器集群。每个流处理器集群再次把数据分解，分配给调度器所控制的多个计算核心同时执行数据的计算和处理。尽管每个线程的计算性能、效率与 CPU 中的 Core 相比低了不少，但是当所有线程都并行计算，那么累加之后它的计算能力又远远高于 CPU。但是 GPU 的功耗非常大，相比其他处理器也非常笨重，在大多数情况下并不适合边缘计算，因为边缘计算很多客户端小巧且仅使用电池供电。



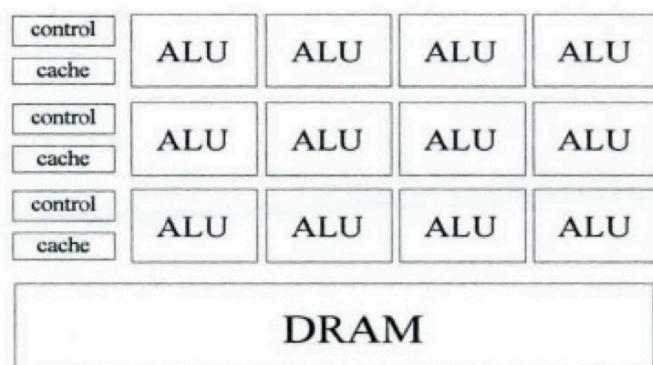


图 2-9 GPU 结构图

Figure 2-9 GPU structure

FPGA 在硬件层面上，其内部集成大量的数字电路基本门电路和存储器，用户可以通过烧入配置文件来定义它们之间的连线，从而达到定制电路的目的；逻辑层面上，它不依赖于冯诺依曼结构<sup>[28]</sup>，一个计算得到的结果可以被直接馈送到下一个无需在主存储器临时保存，因此不仅存储器带宽需求比使用 GPU 或者 CPU 实现时低得多，而且还具有流水处理和响应迅速的特点。可以获得和计算性能的最佳平衡。同时，与固定的 ASIC<sup>[29]</sup>芯片相比，FPGA 又具有充分的灵活性，只需要使用硬件语言重新改写网络结构和参数，就能得到一个适用于新场景的硬件平台。因为神经网络的训练过程中的计算非常复杂 FPGA 难以实现，所以 FPGA 在神经网络加速领域主要着眼于推断过程。但是在未来的深度学习中，大约有 95% 的应用是数据的推断。而且 FPGA 相较于 GPU/CPU 无论在研发还是产出上的成本都明显降低。

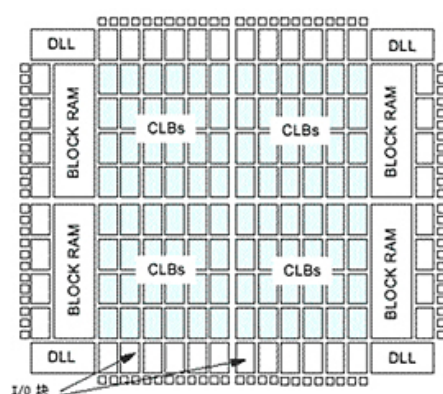


图 2-10 FPGA 内部结构图

Figure 2-10 FPGA internal structure

## 2.3 本章小结

本章介绍了波束合成技术与时钟同步技术的概况，包括基本概念、基本原理和在工程实践中常用的技术手段。本章的主要内容包括：

（1）基于均匀线阵场景，建立了波束合成的信号处理模型，推导出了均匀直线阵列合成波束的方向图函数。然后介绍了传统的模拟和数字波束合成技术及其典型的系统架构，并分析总结了上述几种多波束合成架构的特点。分析结果表明，在 5G 通信系统的大带宽大数据量的场景下，传统的多波束合成架构在工程实现上有计算复杂度过大和计算时延过高的问题，需要进行优化和改进。

（2）首先介绍了神经网络的基本概念和基本框架。然后介绍了神经网络 AlexNet 的在一些通用硬件平台上的实现，并对比了神经网络在这些硬件平台上的实现特点。进一步的，介绍了神经网络的在 FPGA 硬件平台的加速技术和方案，并分析了这些硬件加速技术的特点。分析结果表明，在 5G 智能系统的大带宽大数据量的条件下，基于通用平台的 AlexNet 硬件加速技术难以解决计算复杂度过高，计算延迟过大的难题，而基于 FPGA 硬件平台的神经网络加速利用 FPGA 本身的优势能解决系统计算延迟过高的问题，在工程实践中需要一种低计算复杂度、灵活性好且易于移植的神经网络硬件加速方案。

因此，接下来本文将重点研究针对大带宽大数据量的 5G 智能通信系统中的低延迟、低复杂度的多波束数字波束合成技术和低延迟、低计算复杂度、易于移植的 FPGA 硬件加速方案。

### 第三章 低复杂度的多波束合成技术方案

在 5G 智能通信系统中，带宽相比上一代通信系统变大，达到 500MHz 以上，数据量也成比例变高，大带宽大数据吞吐给实时的数字波束合成带来了巨大的压力；同时，数据处理的计算复杂度过高，不满足 5G 系统中低延迟的要求，给波束合成技术带来了巨大的挑战，传统的波束合成技术已经难以满足需求。从降低数据处理的速度和降低计算复杂度两个方面出发，本章设计一种低计算复杂度的实时数字波束合成系统，并对其硬件实现做了优化。

本章 3.1 节提出了一种低复杂度低延迟的多波束合成系统架构设计；3.2 节对该系统方案的实现做出了优化；3.3 节从子信道数目、复杂度和延迟三个方面对该系统的性能做了分析；3.4 节从覆盖范围、旁瓣抑制两个方面对合成的波束做了分析；3.5 节对本章的主要内容进行了总结。

#### 3.1 低复杂度数字波束合成系统设计

本节设计以最简单的均匀线阵为例，可将其扩展到面阵等其他天线阵列。按照 2.1 中的建立坐标系的方法，假设阵元中某一阵元  $n$  的坐标为  $(x_n, 0)$ ，则上述平面波到达该阵元相对于到达中心阵元的波程差为：

$$\Delta\lambda = -(x_n \cdot \sin\theta) \quad (3-1)$$

上述阵元接收到的该平面波信号相对于中心阵元接收到的该平面波信号的时间差为（ $c$  表示光速）

$$\Delta t = \frac{\Delta\lambda}{c} \quad (3-2)$$

$\Delta t$  的值为负表示超前，为正表示滞后。假设中心阵元接收到的上述平面波信号表示为  $x(t)$ ，则该阵元接收到的上述平面波信号为  $x(t - \Delta t)$ 。补偿上述不同阵元的时间差的过程，在频域的角度来看，可表示为：

$$Y(e^{j\omega}) = X(e^{j\omega}) \cdot e^{-j\omega \frac{\Delta t}{T_s}} \cdot H(e^{j\omega}) \quad (3-3)$$

其中  $-\frac{\Delta t}{T_s} = D + d$ ， $H(e^{j\omega}) = e^{-j\omega d}$ ， $T_s$  表示采样间隔， $D$  表示整数部分的延时， $d$  表示分数部分的延时。

输入信号  $x(t)$  是高速大带宽信号，首先要去除冗余信号减少带宽，同时降采样降低数据处理速度。本文系统采用数字正交变换处理对原始信号进行处理，得到

解析信号。数字正交变换处理使用希尔伯特变换实现实信号到 I/Q 复信号的转化，将原始实信号变成解析信号。原始信号通过希尔伯特变换器得到 Q 路正交信号，这个过程在频域上表现为将频谱相位推移 90 度。同时把原始信号作为 I 路同相信号，得到解析信号  $y(n) = I + jQ$ ，其中  $I = x(n)$ ， $Q = h(x(n))$ 。然后对解析信号进行 2 倍降采样，在时域中降低了数据处理的速度，同时，在频域中，由于实信号的频谱在  $[0, 2\pi]$  上是共轭对称，知其一半，则可知道另一半，差别仅在相位，因此信号通过降采样处理在频域中只保留了  $[0, 2\pi]$  中单边谱，去除了冗余的频谱。

设  $x(n)$  为输入信号， $h(n)$  为希尔伯特滤波器，则 Q 路信号的正交变换处理可表示为(3-4)：

$$\begin{aligned} y(m) &= x(n) * h(n)|_{n=2m} \\ &= x(2m) * h(2m) \\ &= \sum_{l=0}^L h(l)x(2m-l) \end{aligned} \quad (3-4)$$

整个过程从频域的角度来看，就是实信号在  $[0, 2\pi]$  上的对称谱变成了单边谱，如图 3-1 所示。

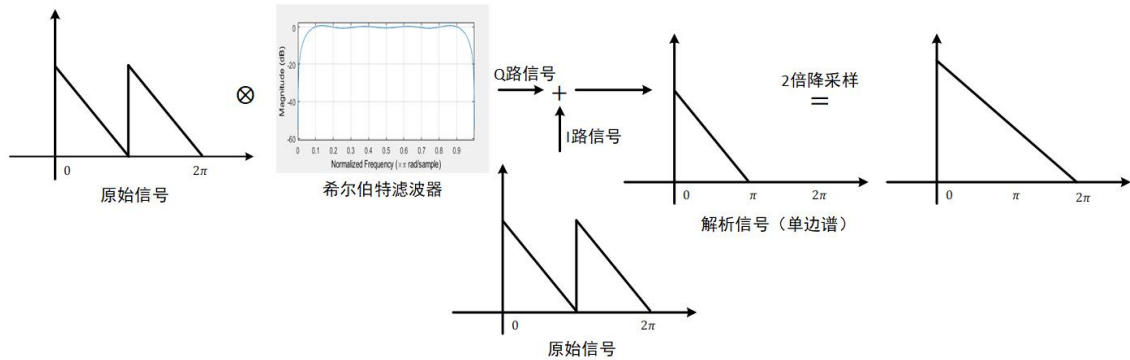


图 3-1 数字正交变换原理

Figure 3-1 Principle of digital orthogonal transformation

经过数字正交变换处理之后，5G 通信系统中的高速大带宽信号频谱从双边谱变成单边谱，避免了频谱浪费，同时降采样也降低了数据处理的速率，缓解数据处理的压力。

数字正交变换处理之后，虽然降低了数据的传输速度，减半了频谱宽度，信号仍为大带宽的高速信号。对于宽带信号，延迟补偿因子不能直接与宽带信号相乘，只能通过分数延迟滤波器与之做卷积运算。这种卷积运算不仅计算复杂度高，而且资源浪费严重，因为每合成一个波束，就要设计一套分数延迟滤波器与之相卷

积。本节提出的系统设计中，将宽带信号划分为若干个窄带信号，做窄带信号处理。下面介绍将宽带信号处理变为窄带信号处理的方式。

将宽带信号在 $[0, 2\pi]$ 内分解为 $K$ 个窄带子信号，每个子信道的带宽为 $\frac{2\pi}{K}$ ，当 $K$ 取值比较大时，这些子信道的窄带信号就可以看作是点频信号。因此设计通过频率为 $\frac{\pi}{K}$ ，截止频率为 $\frac{2\pi}{K}$ ，过度带宽为 $\frac{\pi}{K}$ 的低通滤波器 $h_{LP}(n)$ ，将宽带信号与 $h_{LP}(n)$ 卷积，得到窄带信号，最后对子信道的信号做 $K$ 倍降采样，频域上将每个子信道的窄带频谱扩展到 $[0, 2\pi]$ ，时域上表现为数据处理速率降低了 $K$ 倍。单个子信道的处理过程如图 3-2 所示。为了得到 $K$ 个窄带子信号，需要将宽带信号依次乘以相移因子 $\omega_k = 2\pi \cdot \frac{k}{K}$ 对宽带信号做调制调整相位，然后依次与该滤波器 $h_{LP}(n)$ 做卷积。第 $k$ 个数字信道化处理可表示为(3-5)

$$\begin{aligned} z_k(m) &= [y(n)e^{-j\omega_k n}] * h_{LP}(n)|_{n=Km} \\ &= \sum_{l=0}^{L-1} y(Km-l)e^{-j\omega_k(Km-l)} \cdot h_{LP}(l) \end{aligned} \quad (3-5)$$

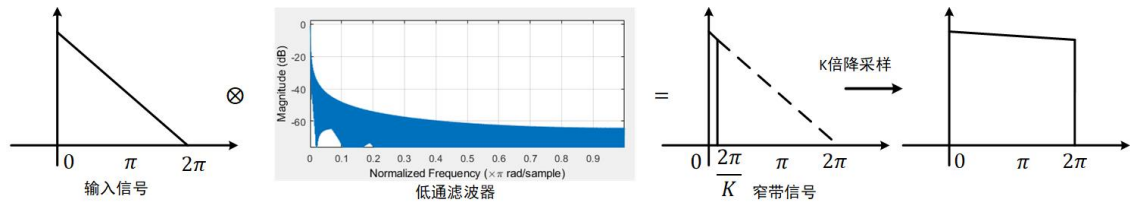


图 3-2 单个子信道的处理过程

Figure 3-2 Processing of a single subchannel

整个数字信道化处理过程如图 3-3 所示。

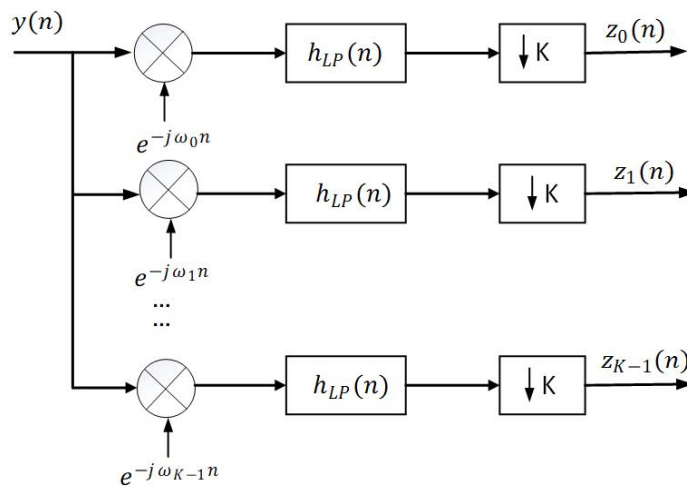


图 3-3 数字信道化处理流程图

Figure 3-3 Digital channelization processing flowchart

信号通过数字信道化处理之后，宽带信号变成了 $K$ 路窄带信号，同时速度变成了原始输入信号速度的 $\frac{1}{2K}$ 。之后这 $K$ 路窄带信号进入加权求和处理部分。这些窄带信号可以看作是点频信号，可以直接与相位补偿因子相乘，直接完成延迟补偿，避免了滤波器的使用，降低了计算复杂度。并且随着合成波束数目的增加，资源更加节约，在 3.3 节将会给出证明。

系统设计的总体架构如图 3-4 所示。经过 AD 采集的信号经过该系统，完成数字波束合成，生成想要的多波束。但在实际工程上，该系统的设计仍然计算量比较大，不适应 5G 通信系统中的大带宽低延迟的特点。本文在对算法进行设计之外，同时在 3.2 节对系统的工程实施上做了优化

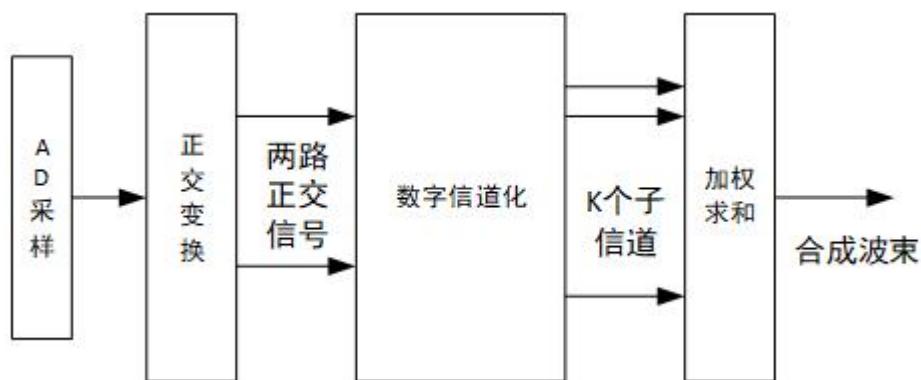


图 3-4 系统设计总体架构

Figure 3-4 Overall architecture of system design

### 3.2 低复杂度数字波束合成系统实现优化

对于 3.1 节中使用的数字正交变换，传统的希尔伯特变换处理一般采用的方法是使用 FIR 滤波器实现，如 (3-4) 中的“先滤波后抽取”的实现结构，先对原始信号做滤波处理，然后 2 倍降采样抽取。这种实现方法计算复杂度很高，对信号处理速度的要求也很高。为了优化算法的计算复杂度和系统处理的延迟，采用等价且更为高效的多相结构实现。

优化过程如下：设 $x(n)$ 为输入信号， $h(n)$ 为希尔伯特变换器，新型的数字正交变换的过程可以表示如下：

令 $l = 2i + k$ ，(3-4) 可以改写为：

$$\begin{aligned}
y(m) &= x(n) * h(n)|_{n=2m} \\
&= x(2m) * h(2m) \\
&= \sum_{l=0}^L h(l)x(2m-l) \\
&= \sum_{k=0}^1 \sum_{i=0}^{\frac{L}{2}-1} h(2i+k)x(2(m-i)-k)
\end{aligned} \tag{3-6}$$

定义  $h_k(i) = h(2i+k)$ ,  $x_k(i) = x(2i-k)$ , 则有:

$$\begin{aligned}
y(m) &= \sum_{k=0}^1 \sum_{i=0}^{\frac{L}{2}-1} h_k(i)x_k(m-i) \\
&= \sum_{k=0}^1 x_k(i) * h_k(i)
\end{aligned} \tag{3-7}$$

(3-7)就是等价的“先抽取后滤波”的多相结构实现方式。其处理过程可用图 3-5 表示。

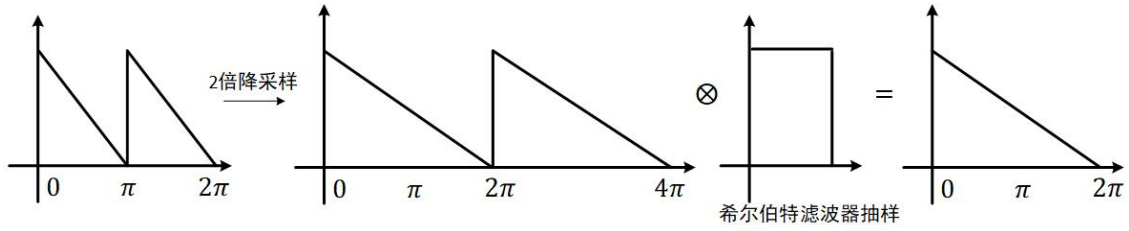


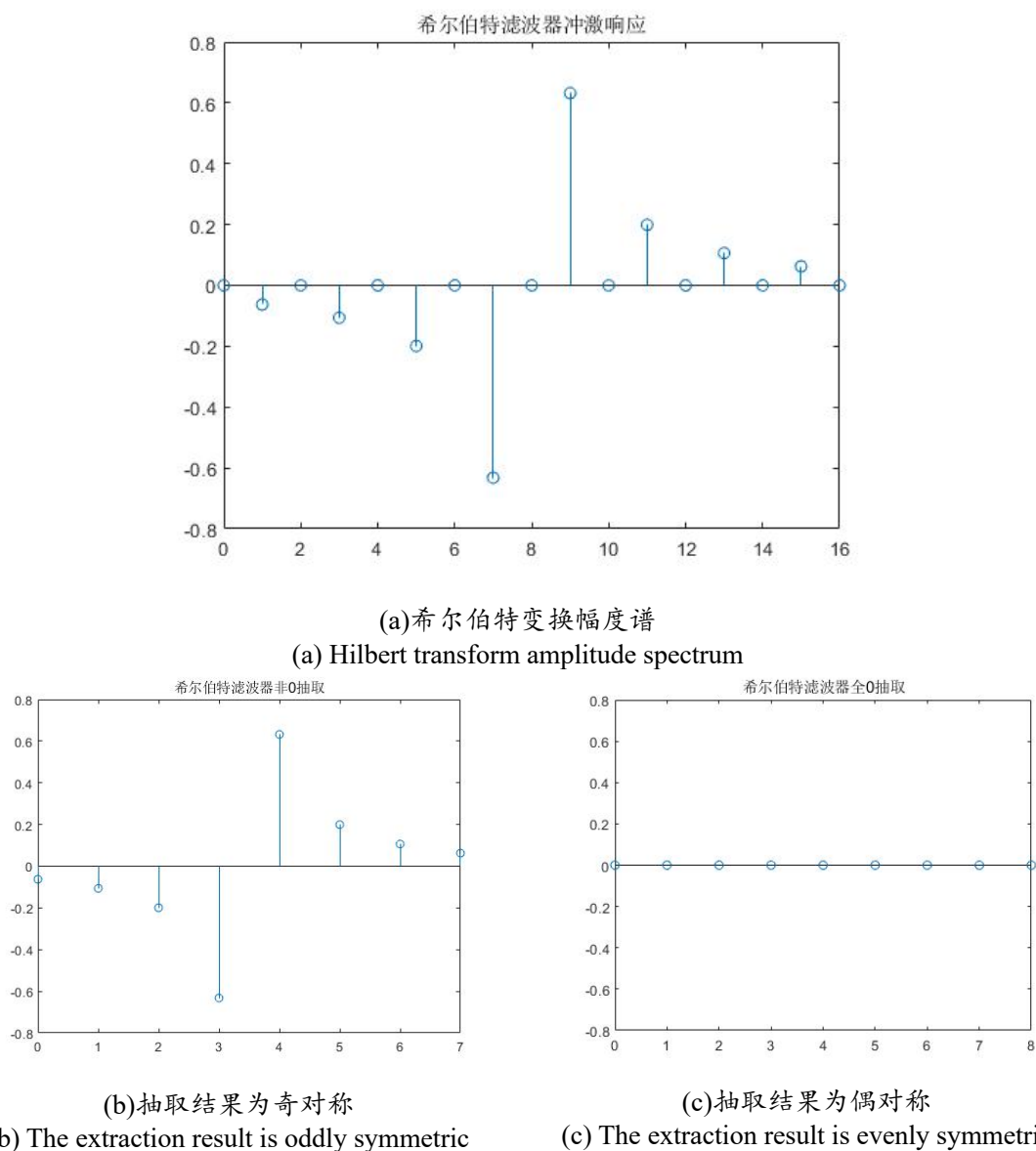
图 3-5 先抽取后滤波的多相结构

Figure 3-5 Polyphase structure of first-sampling-then-filtering

其中, 3-5 中的滤波器的原型是希尔伯特滤波器, 是由原来的希尔伯特变换器 2 倍降采样之后得到的新滤波器。希尔伯特变换器, 取其阶数  $M$  选取为偶数, 冲激响应表示为

$$h(n) = \begin{cases} \frac{2}{\pi} \cdot \frac{\sin^2\left(\frac{\pi\left(n-\frac{M}{2}\right)}{2}\right)}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases} \quad n = ,1, \dots, M \tag{3-8}$$

因此希尔伯特变换器的冲激响应的幅度谱可以用图 3-6(a)来表示:

图 3-6 希尔伯特滤波器冲激响应( $M=16$ )及其抽取Figure 3-6 Hilbert filter impulse response ( $M=16$ ) and its extraction

从图 3-6 中可以看出, 对希尔伯特变换器进行 2 倍降采样抽取, 可以得到两路滤波器系数, 其中一路全为 0, 如图 3-6(c)所示; 另一路具有奇对称的特点, 如图 3-6(b)。因此在实际的工程实施过程中, 可以将系数全为零的那一路信号直接丢弃不算, 另一路信号可以先做减法再做乘法, 过程如(3-9)所示, 这样, 在传统处理方法中需要  $M$  个乘法器的正交信号现在只需要  $\frac{M}{4}$  个乘法器。



$$\begin{aligned}
y(n) &= \sum_{n=1}^{\frac{M}{2}} x(n)h(2n) \\
&= \sum_{n=1}^{\frac{M}{4}} [x(n) - x(2n)]h(2n)
\end{aligned} \tag{3-9}$$

没有经过希尔伯特变换器处理的信号支路称为同相支路。对于同相支路(I路), 仍需要一个原型滤波器为整数倍单位延时器, 用于补偿正交支路滤波器群延迟特性导致的同相支路与正交支路信号相位不一致。从(3-8)中可以看出, 希尔伯特变换器的相位延迟为 $\frac{M}{2}$ , 因此将同相支路的原型滤波器冲激响应设计为 $\delta(n - \frac{M}{2})$ 。其冲激响应如图 3-7 所示。

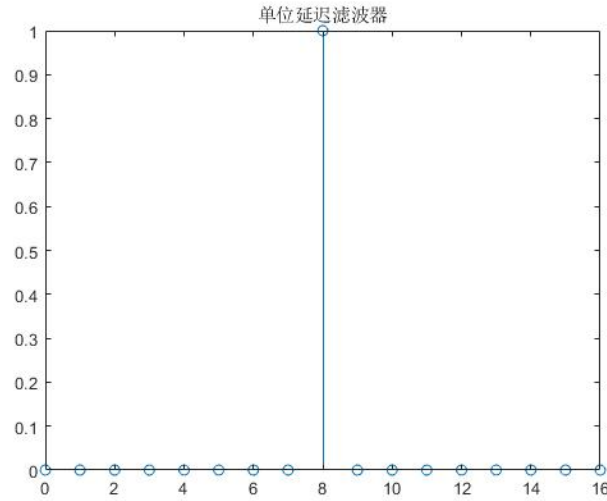


图 3-7 同相支路的原型滤波器冲激响应

Figure 3-7 Impulse response of prototype filter of in-phase branch

从上述的推算中可以看出, 同相支路和正交支路的处理都采用多相结构实现。同相支路的原型滤波器系数经过抽取后其中一路系数为全 0, 另一路仍为整数倍单位延时器, 冲激响应表示为 $h_I(n)$ ; 正交支路的原型滤波器系数经过抽取后其中一路系数为全 0, 另一路系数具有奇对称的特点, 冲激响应表示为 $h_Q(n)$ 。值得注意的是, 同相支路和正交支路的滤波器系数经过抽取之后哪一路系数全为 0 与 $\frac{M}{2}$ 是奇数还是偶数有关, 同样的, 对原始信号的抽取顺序也与之有关。数字正交变换处理的实现的原理框图如图 3-8 结构所示。

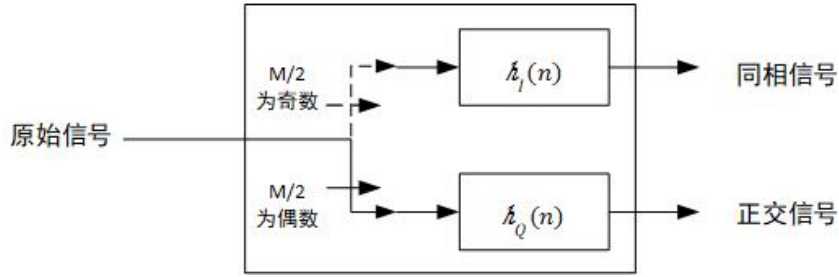


图 3-8 数字正交变换处理的实现的原理框图

Figure 3-8 Principle block diagram of digital orthogonal transformation processing

同样的，“先抽取后滤波”的思想同样可以用于数字信道化处理。数字信道化处理中，对宽带信号先调制，然后调制之后的宽带信号通过低通滤波器得到 $K$ 路窄带信号，最后对窄带信号做 $K$ 倍降采样得到 $K$ 路子信道。现在对其做优化，采用先“抽取后滤波”的形式，先对低通滤波器和输入信号做 $K$ 倍降采样抽取，然后再去做卷积运算。对 (3-5)，令 $l = K_i + p$ ，则

$$\begin{aligned}
 y_k(m) &= [x(n)e^{-j\omega n}] * h_{LP}(n)|_{n=Km} \\
 &= \sum_{l=0}^L x(Km - l)e^{-j\omega_k(Km-l)} \cdot h_{LP}(l) \\
 &= \sum_{p=0}^{K-1} \sum_{i=0}^{\frac{L}{K}-1} x(K(m-i) - p)e^{-j\omega_k(K(m-i)-p)} \cdot h_{LP}(Ki + p)
 \end{aligned} \tag{3-10}$$

定义 $x_p(m) = x(Km - p)$ ,  $h_p(m) = h_{LP}(Km + p)$ , 则有

$$\begin{aligned}
 y_k(m) &= \sum_{p=0}^{K-1} \left[ \sum_{i=0}^{\frac{L}{K}-1} x_p(m-i)e^{-j\omega_k K(m-i)} \cdot h_p(i) \right] e^{j\omega_k p} \\
 &= \sum_{p=0}^{K-1} [x_p(m)e^{-j\omega_k Km} * h_p(m)] \cdot e^{j\omega_k p}
 \end{aligned} \tag{3-11}$$

因为 $e^{-j\omega_k Km} = e^{-j(2\pi\frac{k}{K})Km} \equiv 1$ ，所以有

$$y_k(m) = \sum_{p=0}^{K-1} [x_p(m) * h_p(m)] \cdot e^{j\omega_k p} \tag{3-12}$$

上式可以等效为多相滤波加 IFFT 的结构,对其进一步变换:

$$\begin{aligned}
y_k(m) &= \sum_{p=0}^{K-1} [x_{K-1-p}(m) * h_{K-1-p}(m)] \cdot e^{j\omega_k(K-1-p)} \\
&= \left\{ \sum_{p=0}^{K-1} [x_{K-1-p}(m) * h_{K-1-p}(m)] \cdot e^{-j\omega_k p} \right\} \cdot e^{j\omega_k(K-1)} \\
&= \left\{ \sum_{p=0}^{K-1} [x_{K-1-p}(m) * h_{K-1-p}(m) \cdot e^{-j\omega_k p}] \right\} \cdot e^{-j\omega_k}
\end{aligned} \tag{3-13}$$

(3-13)是多相滤波加 FFT 的高效实现结构。最终整个数字信道化处理部分转换为等价的且更为高效的“先抽取后滤波”的多相结构。再加上前面的调制部分，最终整个数字信道化处理可以采用多相滤波加 FFT 的结构实现。整个数字信道化的结构如图 3-9 所示。

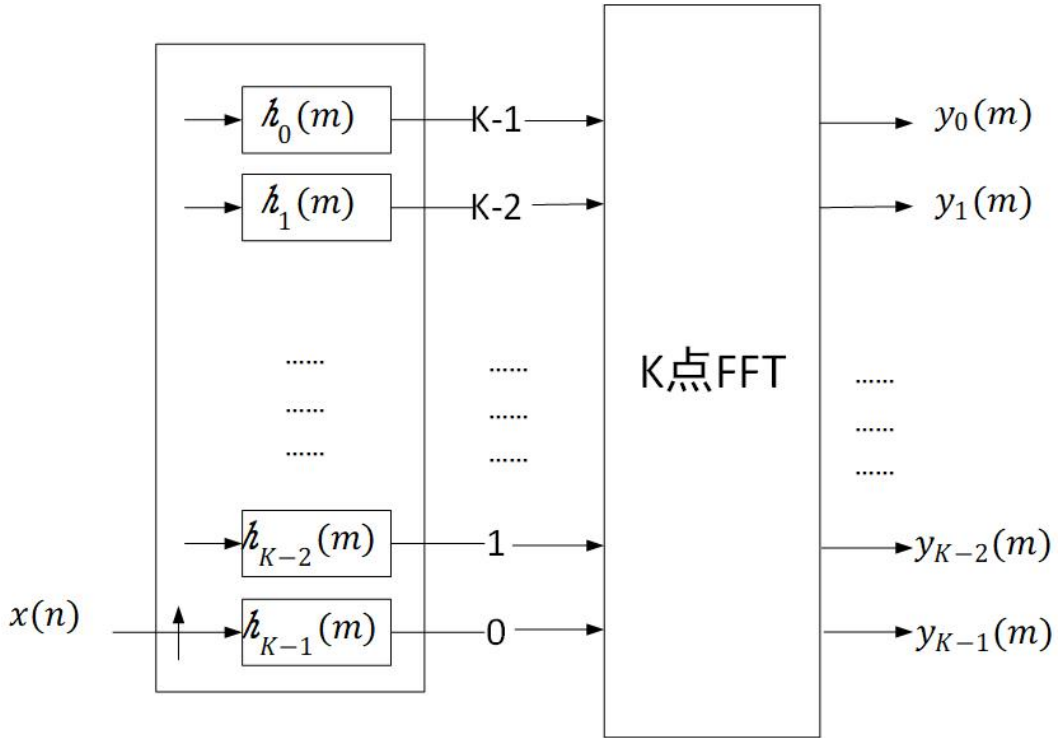


图 3-9 数字信道化结构图  
Figure 3-9 Digital channelization structure

从(3-13)可以看出，还需要对 FFT 的输出结果（各子信道的信号）乘以一个补偿因子  $e^{-j\omega_k}$ ，补偿因子可以折算入复加权系数中，因此乘以补偿因子的操作可以并入加权求和单元来处理。所以在加权求和的过程中，加权因子需要包含三个部分，分别是延时补偿系数  $W_i$ ，幅度加权因子  $B(\varphi)$  以及补偿因子  $e^{-j\omega_k}$ 。所以每

个信道的补偿因子可以表示为：

$$W_{(k)} = W_i \times B(\varphi) \times e^{-j\omega_k} \quad (3-14)$$

统一的， $W_{(k)}$ 可以表示成 $W_{(k)} = a(k)e^{-j\omega_k(D+d)}$ ，其中 $a(k)$ 是加权求和系数的包络， $D$ 是整数部分的延迟补偿， $d$ 是分数部分的延迟补偿。为了提升算法的性能，减少乘法器的使用和延迟压力，简化算法，将整数部分的延迟和分数部分的延迟采用不同的方式实现。整数部分的延迟采用时域补偿的方法来实现，即在信号的数字信道化处理之前将原信号与 $\delta(n-D)$ 卷积，工程上可以采用 $D$ 个时钟周期的单位周期延迟实现。分数部分的延迟的处理采用将数字信道化后各子信道的窄带信号乘以补偿因子 $a(k)e^{-j\omega_k d}$ 实现。在实现的过程中，将补偿因子表示为 $a(k)e^{-j\omega_k d} = w_I + jw_Q$ ，将输入到加权求和的数字信号表示为 $I + jQ$ ，则在加权求和的复数乘法为：

$$\begin{aligned} (I + jQ) \times (w_I + jw_Q) &= Iw_I - Qw_Q + j(Qw_I + Iw_Q) \\ &= Iw_I - Qw_I + Iw_Q - Qw_Q - Iw_Q + Qw_I + j(Qw_I + Iw_Q) \\ &= (I - Q) \times (w_I + w_Q) - Iw_Q + Qw_I + j(Qw_I + Iw_Q) \end{aligned} \quad (3-15)$$

观察上式， $Qw_I$ 和 $Iw_Q$ 可以共用，因此只要算一遍即可，因此上式复数乘法的实现由传统的四乘法器架构可变为先预处理然后三乘法器架构。如图 3-10，表明了加权求和处理三乘法器的实现原理框图。

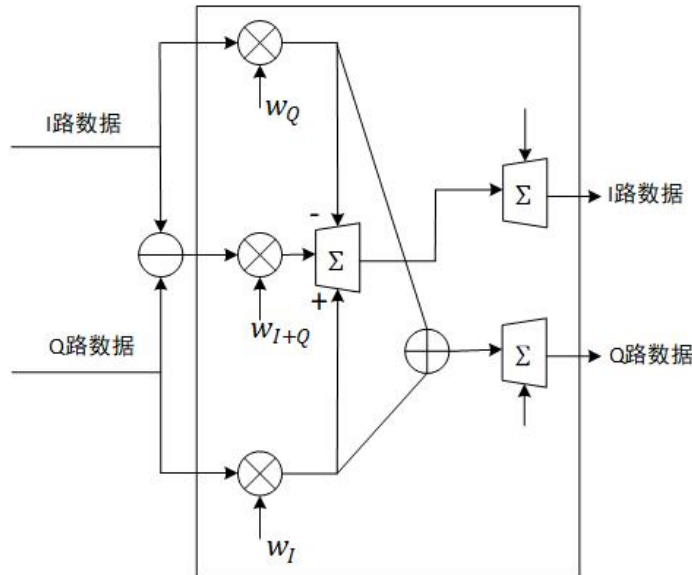


图 3-10 加权求和处理单元框图

Figure 3-10 Block diagram of weighted sum processing unit

### 3.3 系统方案分析

#### 3.3.1 子信道数目 $K$ 分析

数字信道化处理过程中，窄带子信道数目 $K$ 的选取不仅会影响多相滤波器、FFT 处理的资源消耗，而且也会对时间分辨率和频率分辨率造成影响。

数字信道化处理的调制和低通滤波处理可以等效看成是在原来的宽带信号上加了  $K$  个矩形窗，如图 3-11 所示。 $K$  数目的增加会在多项滤波器和 FFT 处理时增加资源消耗（分析可见 3.3.2 节），同时会提高频率分辨率，降低时间分辨率。设带宽为  $B$ ，系统处理主频为  $\frac{1}{T}$ ，那么数字信道化处理的频率分辨率为  $\frac{B}{K}$ ，时间分辨率为  $KT$ 。因此当  $B$  和  $T$  固定不变时， $K$  越大，信道化处理的频率分辨率越好，而时间分辨率会变差。以宽带 500MHz 信号，处理主频为 500MHz 为例，如果选取分别选取  $K=256$  和  $K=1024$ ，则其时间分辨率和频率分辨率对比如表 3-1 所示。

因此在实际工程中，需要结合系统的时、频分辨率要求来做折中考虑。

表 3-1 取不同  $K$  值时，系统时、频分辨率对比  
Table 3-1 Comparison of time and frequency resolution of the system with different  $K$

$K$	频率分辨率	时间分辨率
256	1.95MHz	512ns
1024	0.49MHz	2048ns

#### 3.3.2 系统计算复杂度分析

由于在系统实现的过程中，乘法器的资源最有限，并且计算需要的时延最多，因此在本节的分析中，用实现过程中所需要乘法器的数目作为评价系统的计算复杂度的一个指标。并且，在本节的分析中将本文提出的实现方案与传统分数延迟滤波器算法作对比，进而分析本文提出大带宽波束合成实现方案的优点。

在本文提出的系统中，数字正交变换处理过程乘法器的消耗数目与数字正交变换器的阶数 $M$ 有关，由于先抽取后滤波，因此抽取之后的滤波器的阶数为  $\frac{M}{2}$ ，并且由于抽取之后的系数由奇对称的特点，可以先做减法再做乘法，因此实际工程实现中使用滤波器的阶数仅为  $\frac{M}{4}$ 。相比之下，在传统分数延迟滤波器算法的数字正交化处理过程中，需要使用 $M$ 阶希尔伯特滤波器才能达到同等效果。

在数字信道化的处理过程中，对原信号的整数部分的延迟采用时域处理的方法实现，即在数字信道化之前将宽带信号与 $\delta(n-D)$ 卷积。而宽带信号与 $\delta(n-D)$ 的卷积可以直接使信号通过 $D$ 个单位的移位寄存器组实现。因此乘法器的消耗仅包括多项滤波器组和 FFT 两部分。多项滤波器组消耗的乘法器与原型低通滤波器的阶数  $L$  与阶数和子信道的个数  $K$  有关，每个子信道需要消耗的乘法器的个数为 $\frac{L}{K}$ 。由于抽取后的信号速率非常低，各路滤波器组可以分时使用同一组滤波器组，只需要改变系数即可。FFT 消耗的乘法器的个数与实现架构和信道数  $K$  有关，在 FPGA 的实现平台上，可以通过 Xilinx 官方 Fast Fourier Transform IP Core 实现，如图 3-11，以子信道数 $K = 1024$ 为例，可以看到1024点的 FFT 消耗的乘法器数目为 12 个。

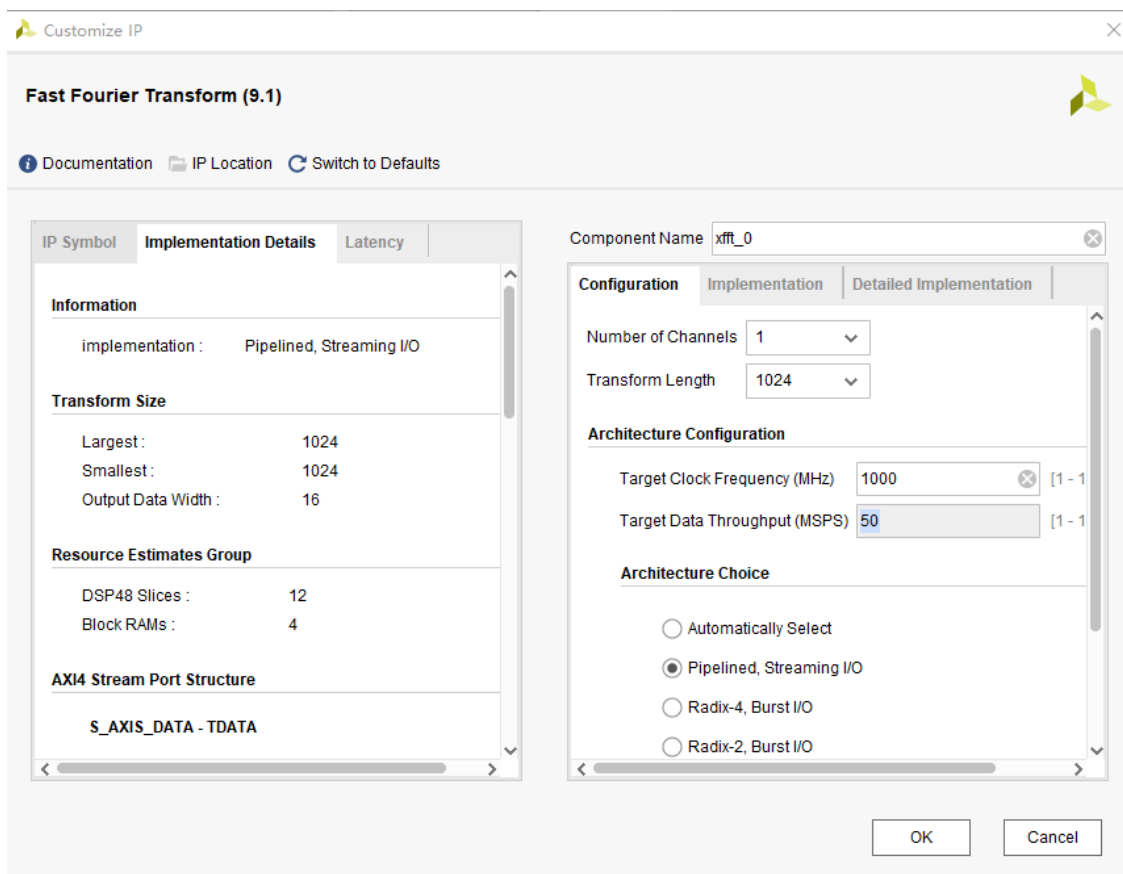


图 3-11 Fast Fourier Transform IP Core 设置  
Figure 3-11 Fast Fourier Transform IP Core settings

在传统分数延迟滤波器算法中，消耗的乘法器的数目与分数延迟滤波器的阶数  $M$  有关，若分数延迟滤波器的阶数为  $M$ ，则消耗的乘法器的数目是 $(M + 1) \cdot 2$ ，分数延迟滤波器的阶数越高，则算法的性能越好，同时消耗的乘法器的数目也更

多。表 3-1 显示了分数延迟滤波器阶数与乘法器资源消耗以及性能之间的对比。

表 3-1 分数延迟滤波器乘法器消耗和性能对应关系表  
Table 3-1 Fractional delay filter multiplier consumption and performance

分数延迟 FIR 滤波器阶数	乘法器消耗	绝对延迟误差	幅度误差
5	12	$\leq 5ps$	$\leq 0.1dB$
7	16	$\leq 1ps$	$\leq 0.1dB$
9	20	$\leq 1ps$	$\leq 0.1dB$

对比采用 16 阶的希尔伯特变换器，1024 阶低通滤波器，宽带信号分成 1024 个子信道，5 阶分数延迟滤波器。合成单通道、单波束的波束时，两种方案的乘法器数目对比如表 3-2 所示。

表 3-2 本文提出系统设计与传统分数延迟滤波器所需乘法器数目比较  
Table 3-2 Comparison of the number of multipliers required for system design and traditional fractional delay filters

比较项目	本文提出系统	传统分数延迟滤波器算法
数字正交变换	4	16
数字信道化	13	0
分数延迟滤波器	0	12
加权求和	3	4

系统乘法器的实际消耗与通道数 $n_1$ 和合成波数 $n_2$ 有关。在传统分数延迟滤波器算法中，每增加一个通道就要增加一条并行流水线，同时，每增加一个合成波束，就需要增加一个分数延迟滤波器和加权求和单元；在本文提出的系统方案中，每增加一个通道就要增加一条并行流水线，同时，每增加一个合成波束，几个波束可以共用一套数字信道化处理模块，只需要增加一套加权求和处理流水线。因此传统分数延迟滤波器算法和本文提出系统的系统复杂度可以分别用 $O_1$ 和 $O_2$ 表示：

$$\begin{aligned} O_1 &= n_1 \cdot [16 + n_2 \cdot (12 + 4)] \\ O_2 &= n_1 \cdot (4 + 13 + n_2 \cdot 3) \end{aligned} \quad (3-16)$$

因此，以 8 通道 2 波束为例，分数延迟滤波器算法需要 384 个乘法器，而本文提出的系统方案需要 184 个乘法器，系统复杂度仅为分数延迟滤波器算法的 47.9%。同时，随着通道数 $n_1$ 和合成波数 $n_2$ 的增加，系统复杂度的降低更加明显。

图 3-12 给出了两种方案所消耗的乘法器随着通道数和合成波束数目增加的变

化。从图中可以看出，本文提出的系统在所消耗乘法器方面优于传统分数延迟滤波器，并且随着通道数和合成波束数目的增加，这种优势更加明显。因此本文提出的宽带波束合成方案具有计算复杂度低的特点。

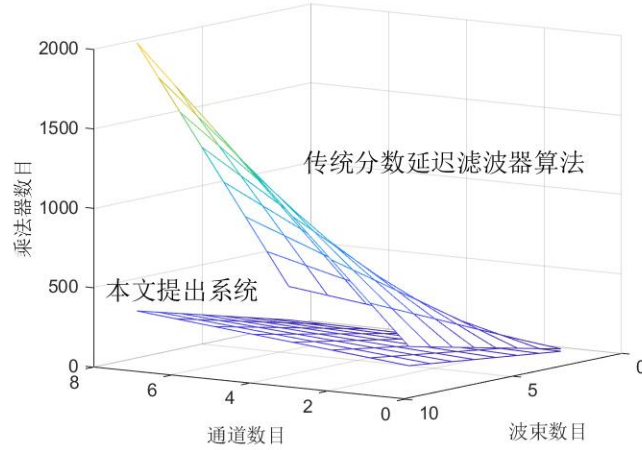
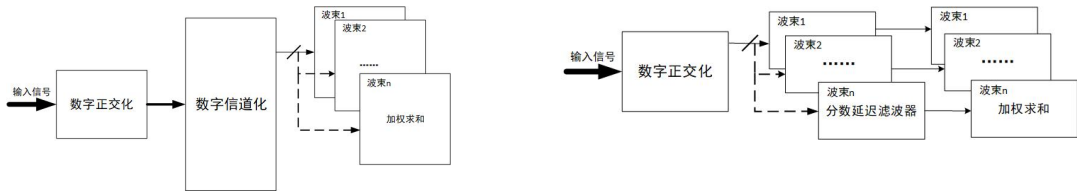


图 3-12 两种方案消耗乘法器与通道数目和波束数目的关系

Figure 3-12 The consumption multiplier of two schemes between the number of channels and beams

### 3.3.3 系统延迟分析

在本文提出的系统中，单通道合成多波束时，单通道接收的数据在数字正交化和数字信道化的处理过程是相同的，因此可以共用，只在加权求和时分别与不同的加权因子相乘。于是，数字多波束合成在单通道时增加的时间延迟消耗仅在加权求和部分。而分数延迟滤波器算法在多波束合成时，需要依次通过多个分数延迟滤波器，仅在数字正交化部分可以共用。因此当合成波束数目增加时，增加的系统延迟包括分数延迟滤波器的延时和加权求和的延迟。图 3-13 给出了两种系统的延迟分析。



(a) 本文提出系统延迟分析

(a) Proposed system delay analysis

(b) 分数延迟滤波器延迟分析

(b) Fractional delay filter delay analysis

图 3-13 系统延迟分析

Figure 3-13 System delay analysis



系统的计算延迟主要跟乘法器的使用数目有关，其他信号处理部分的速度都非常快，彼此差别不大。以每个乘法器需要三个时钟周期为例，在单波束、单通道的数字波束合成时，分数延迟滤波器和本文提出系统主要延迟的对比如表 3-3 所示。

表 3-3 本文提出系统设计与传统分数延迟滤波器系统延迟比较（单位：个时钟周期）  
Table 3-3 Comparison of system design and fractional delay filter system delays (unit: clock cycles)

比较项目	本文提出系统	传统分数延迟滤波器算法
数字正交变换	12	48
数字信道化	39	0
分数延迟滤波器	0	36
加权求和	9	12

因此，以单通道单波束为例，分数延迟滤波器算法需要 93 个时钟周期，而本文提出的系统方案需要 60 个时钟周期，需要的系统延迟仅为分数延迟滤波器算法的 64.53%。当通道数 $n_1$ 和合成波数 $n_2$ 增加时，系统延迟也会增加。根据图 3-13 的系统延迟分析，给出了本文提出系统的计算延迟 $T_1$ 和传统分数延迟滤波器算法的计算延迟 $T_2$ ：

$$\begin{aligned} T_1 &= n_1 \cdot (12 + 39 \times n_2 \cdot 9) \\ T_2 &= n_1 \cdot [48 + n_2 \cdot (36 + 12)] \end{aligned} \quad (3-17)$$

图 3-12 给出了两种方案所消耗的乘法器随着通道数和合成波束数目增加的变化。从图中可以看出，本文提出的系统在所消耗乘法器方面优于传统分数延迟滤波器，并且随着通道数和合成波束数目的增加，这种优势更加明显。因此本文提出的宽带波束合成方案具有系统延迟低的特点。

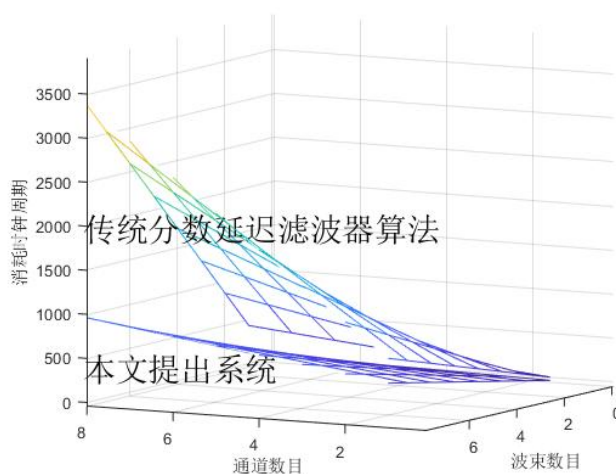


图 3-12 两种方案系统时延与通道数目和波束数目的关系  
Figure 3-12 System delay between channel and beam number

### 3.4 波束合成分析

#### 3.4.1 覆盖范围分析

从 $-90^{\circ}$ 到 $90^{\circ}$ 的多波束合成如图所示。理论上本系统的合成的多个波束可以在空域中覆盖 $180^{\circ}$ 的范围，但是从图中可以看出，偏离天线阵列法向量主瓣太宽，且由于部分主瓣与覆盖区域的另一侧的主瓣发生了混叠而在实际中无法有效应用。对于无线通信系统中的蜂窝网络而言，基站通常架设在每个六边形小区的顶点上，通过三个扇区分别对相邻的三个小区进行覆盖，每个扇区对应的方位角为 $120^{\circ}$ <sup>错误！未找到引用源。</sup>，因此可以剔除上述无效波束（在实际应用中将相应的波束端口置空），如图所示，剩下的波束完全可以覆盖方位角上 $120^{\circ}$ 的范围。

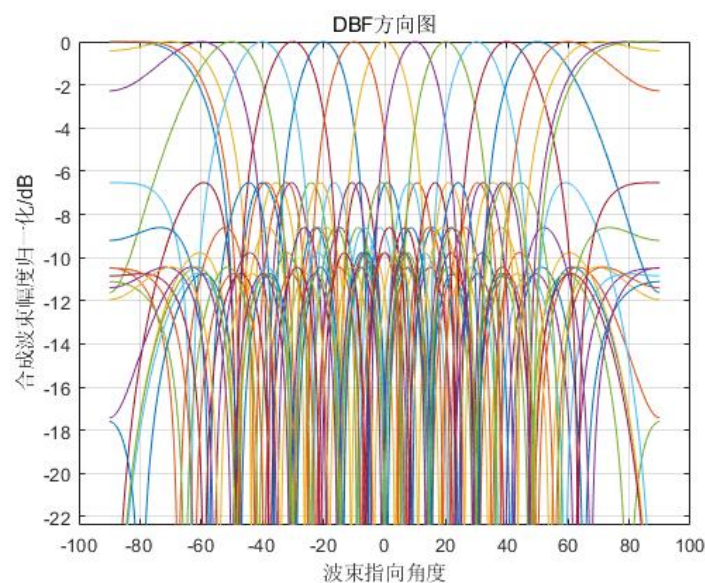
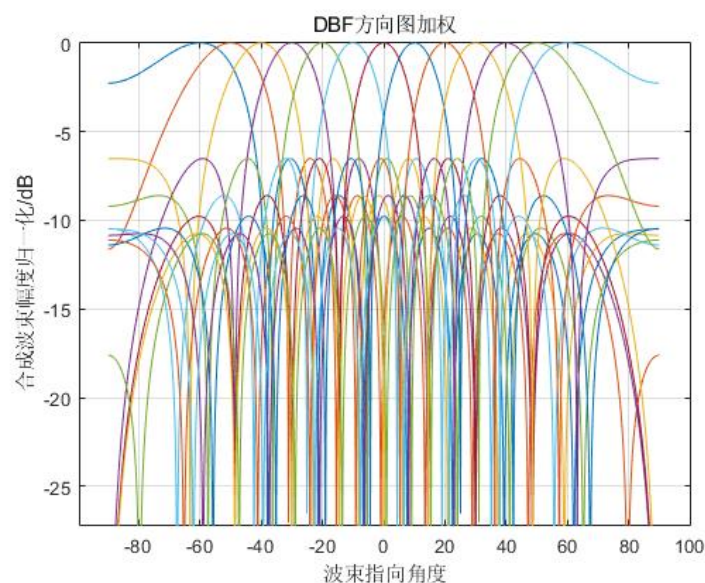
(a) 全空域 $180^\circ$ 覆盖波束合成图(a)  $180^\circ$  coverage beamforming map in full airspace(b) 全空域 $120^\circ$ 覆盖波束合成图(b)  $120^\circ$  coverage beamforming map in full airspace

图 3-13 合成多波束覆盖分析

Figure 3-13 Analysis of multi-beamforming coverage

### 3.4.2 旁瓣抑制分析

在实际的 DBF 方案中，两个关键的指标分别是合成波束对旁瓣的抑制以及主瓣的宽度，要求满足旁瓣抑制的条件下使合成波束的主瓣尽可能窄，实现窄波束、

低旁瓣。波束主瓣的宽度主要与天线阵列的规模有关：天线规模越大波束主瓣越窄。下面给出本文中的旁瓣抑制方法。由于 Dolph-Chebyshev 方向图具有给定旁瓣抑制条件下主瓣宽度最窄的特点<sup>[30]</sup>，应用其作为理想的合成波束方向图。本方案中使用最小二乘算法求解阵列中各阵元的加权系数对其进行拟合。

对于阵元数目为 $N$ 的均匀线阵，为了避免栅瓣的出现，取阵元间距 $d = \frac{\lambda}{2}$ ，采用 Dolph-Chebyshev 所综合出的方向图为：

$$B_C(\theta) = T_{N-1}(\theta) \quad (3-18)$$

其中， $\theta$ 为观察方向与线阵法线的夹角， $\theta_0$ 为入射波与线阵法向量的夹角， $T_M$ 为  $M$  阶 Chebyshev 多项式：

$$T_M(z) = \begin{cases} \cosh(\text{Marccosh}(z)) & z \geq 1 \\ \cos(\text{Marccos}(z)) & |z| < 1 \\ -\cosh(\text{Marcoh}(z)) & z \leq -1 \end{cases} \quad (3-19)$$

$$z = z_0 \cos\left(\frac{\pi d}{\lambda} \sin(\theta)\right) \quad (3-20)$$

$$z_0 = \cosh\left(\frac{1}{M} \text{arcosh}(R)\right) \quad (3-21)$$

$R$  为主旁瓣抑制比。

若设 $\varphi = \frac{2\pi}{\lambda} \cdot d \cdot (\sin\theta - \sin\theta_0)$ ，则有  $N$  元线阵的加权因子为：

$$B(\varphi) = \sum_{n=1}^N A_n e^{j(2n-N-1)\frac{\varphi}{2}} \quad (3-22)$$

为了求解 $A_n$ ，可用傅里叶级数展开的方式计算，可得：

当 $N = 2L + 1$ 时，

$$A_n = r + 2 \sum_{l=1}^L T_{N-1}\left(z_0 \cos\left(\frac{l\pi}{N}\right) \cos\left(\frac{2nl\pi}{N}\right)\right) \quad (3-23)$$

当 $N = 2L$ 时，

$$A_n = r + 2 \sum_{l=1}^L T_{N-1}\left(z_0 \cos\left(\frac{l\pi}{N}\right) \cos\left(\frac{(2n-1)l\pi}{N}\right)\right) \quad (3-24)$$

这样所求解 Dolph-Chebyshev 加权系数计算量非常大，需要多次迭代，于是本文采用最小二乘法拟合的形式来求解加权系数。

直接设  $N$  元线阵的加权因子为：

$$B(\varphi) = \sum_{n=1}^N W_n e^{jn\varphi} \quad (3-25)$$

接下来使用最小二乘算法, 求解阵列中各阵元的加权系数 $w_n$ , 使得合成波束的方向图 $B(\theta)$ 与 $B_c(\theta)$ 的误差最小, 从而确定加权系数。

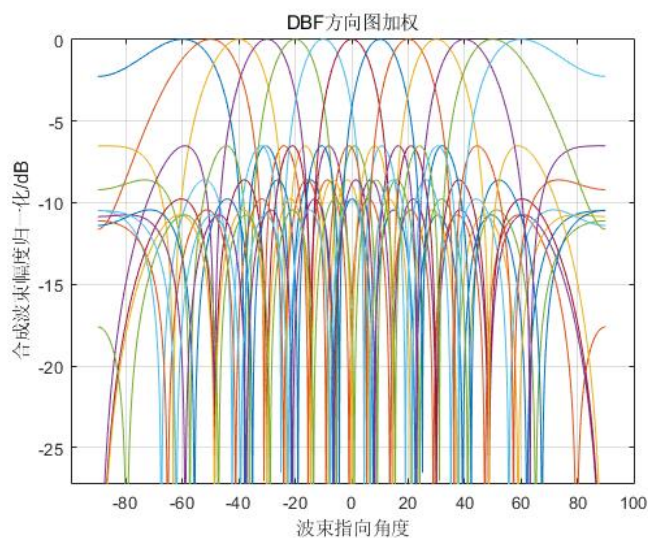
$$J = \sum_{\omega \in [\omega_1, \omega_N]} |B(\varphi) - B_c(\theta)|^2 \cdot W(\omega) \quad (3-26)$$

这样只需要在所关注的 $\omega$ 的范围内令误差最小就可以了。最小二乘法往往需要不断的迭代系数来得到更好的结果, 而在这里只需要一次迭代, 就能取得不错的近似。

针对 8 阵元天线阵列, 合成两个波束的场景, 取抑制比 $R = 20\text{dB}$ , 可得 Dolph-Chebyshev 窗系数 $w(n_1)$ 和相应的 PA 增益设置 $20 \lg B(\varphi)$ 如表 3-4 所示。展示了相应的波束方向图, 对比 3-17 可知经旁瓣抑制后波束旁瓣高度由 $-6.5\text{dB}$ 下降为 $-20\text{dB}$ 。

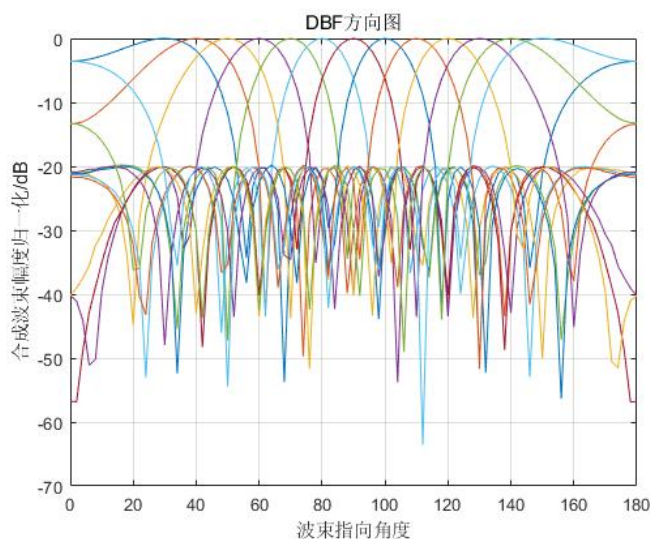
表 3-4 Dolph-Chebyshev 窗系数及相应的 PA 增益设置  
Table 3-4 Dolph-Chebyshev window and corresponding PA gain configurations

天线端口	Dolph-Chebyshev 窗系数 $B(\varphi)$	PA 增益 $20 \lg B(\varphi)$ [dB]
天线#1	0.5799	-4.7329
天线#2	0.6603	-3.6051
天线#3	0.8751	-1.1586
天线#4	1	0
天线#5	1	0
天线#6	0.8751	-1.1586
天线#7	0.6603	-3.6051
天线#8	0.5799	-4.7329



(a) 未旁瓣抑制合成多波束

(a) Non-sidelobe suppressed multi-beamforming



(b) 旁瓣抑制合成多波束

(b) Sidelobe suppression multi-beamforming

图 3-17 全空域 $120^\circ$ 覆盖波束合成图Figure 3-17  $120^\circ$  coverage beamforming map in full airspace

### 3.5 本章小结

本章提出了一种适用于大带宽大数据吞吐的低复杂度低系统延迟的多波束数字波束合成技术方案，且兼顾了灵活性和可扩展性，不仅达到了良好的性能，而且具备较强的工程可实现性。本章的主要内容包括：

(1) 为了应对 5G 智能通信系统中的高速大数据带宽的问题，设计了一个低

延低系统复杂度的 DBF 系统。采用数字正交变换处理和数字信道化处理对高速宽带信号进行降速和宽带分割，最终通过加权求和最终实现多波束数字合成。

(2) 对设计出的 DBF 系统做了实现优化。以“先抽取后滤波”的实现思路代替了传统的“先滤波后抽取”的实现思路。同时在数字信道出处理的部分，使用多项滤波器和 FFT 结构代替了设计中的调制和低通滤波设计。最后在加权求和部分把多项加权系数合并在一起计算，整数部分延迟使用时域处理方法，分数部分延迟采用频域相位补偿的处理方法。大大降低了系统实现的计算复杂度。

(3) 分析系统方案。首先分析了在实际实现时，子信道划分  $K$  选取的两个重要标准。接着分析了系统的计算复杂度，与传统分数延迟滤波器算法做对比，在 8 通道 2 波束的情况下，本文提出系统复杂度仅为传统分数延迟滤波器算法的 47.9%，且随着通道数和合成波束的增加，这种优势会更加明显。最后分析了系统的实时性，在单通道单波束情况下，与传统分数延迟滤波器算法相比，本文提出方案延迟更低，仅为其 64.53%，且当通道数和合成波束增加时，优势会更加明显。

(4) 分析了合成多波束。首先分析了合成波束能良好覆盖空域  $120^\circ$  的范围，能实际应用于工程实现。最后分析了合成波束的旁瓣抑制，结果表明合成波束在主瓣最窄的情况下使得旁瓣抑制达到 20dB。

综上所述，本章提出的多波束合成技术方案不仅保持了良好的性能，并且还降低了工程实现的复杂度和功耗，具有较强的工程可实现性。

## 第四章 神经网络边缘计算加速

作为 5G 智能系统中不可缺少的重要组成，智能物联网正在飞速发展。但 5G 智能通信系统中的大带宽和大数据量的特点给智能物联网中的神经网络的数据处理带来巨大的压力，神经网络需要加速才能适用于 5G 通信系统中的低延迟特点。脱离云计算的边缘计算是目前神经网络加速的前沿方向，但是基于 GPU 和 CPU 的边缘计算加速具有局限性且难以实际应用。针对上述问题，本文提出了一种基于 FPGA 的神经网络边缘计算硬件加速系统，该方案具有准确率高、延时低、功耗低的特点，易于实际工程应用。

本章 4.1 节提出了一种基于 FPGA 的神经网络硬件加速系统架构设计；4.2 节详细介绍了对卷积层、池化层、全连接层的优化和关键技术；4.3 节介绍了系统中的定点量化策略；4.4 通过计算机仿真分析了本章提出的基于 FPGA 的神经网络硬件加速性能；4.5 节对本章的主要内容进行了总结。

### 4.1 基于 FPGA 的神经网络硬件加速系统架构设计

由于 5G 智能通信系统的大带宽和低延迟等特性，使得需要及时通讯的低延迟物联网系统成为可能。在智能物联网中，文字识别是十分重要的一个应用场景；在神经网络领域中，AlexNet 网络因为使用了 ReLU 激活函数、重叠池化等特点，具有划时代的意义，且后来的许多神经网络都以 AlexNet 神经网络为启发。因此在本文中采用 AlexNet 作为文字识别的网络，以 MNIST 手写数据集作为训练和测试数据，在两片当前主流的 FPGA 芯片 xc7k325tffg900-2 上完成测试。

在基于 FPGA 的神经网络硬件加速系统中，需要解决传统 CPU、GPU 等硬件不能解决的高延迟和计算复杂度高的问题，满足神经网络边缘计算的实时性和低功耗需求。为了解决上述问题，本文充分利用了 FPGA 的并行计算和流水线结构。同时，由于 FPGA 内部可以直接存储数据，而不需要访问外部存储器，免去了因为系统指令和存储器访问带来的时间消耗，从而能降低系统的延迟。基于 FPGA 的神经网络硬件加速系统的需求和解决方案如表 4-1 所示。



表 4-1 神经网络硬件加速系统系统需求和解决方案

Table 4-1 Neural network hardware acceleration system requirements and solutions

系统需求	解决方案
降低延迟	多通道并行计算
	FPGA 内部存储参数
	流水线结构无需指令
低功耗	FPGA 自身优势

本文设计的系统架构如图 4-1 所示。系统采用多层彼此之间相互独立、多通道并行运算结构，每一层单独设计，每一层中的每一个卷积核并行运算。因此系统的设计需要考虑到乘法器的分配方式。下面介绍本系统如何分配乘法器。

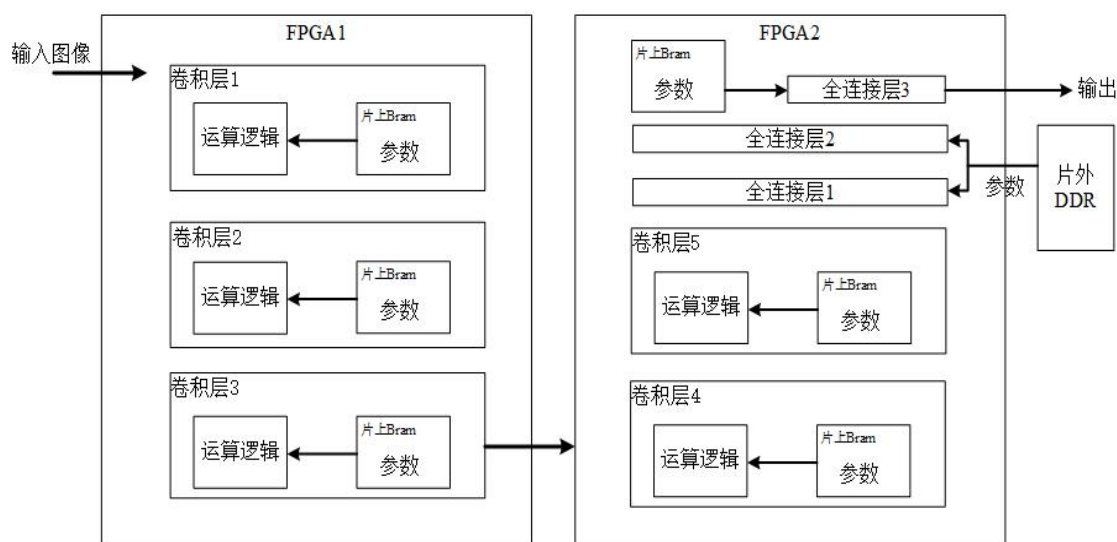


图 4-1 本文设计系统架构

Figure 4-1 System architecture

MNIST 手写数据集每张图的尺寸大小为  $28 \times 28 \times 1$ ，因此，根据第二章的内容，本文系统的中 AlexNet 的系统结构如表 4-2 所示。

表 4-2 本文中 AlexNet 系统结构  
Table 4-2 Proposed AlexNet system architecture

网络层	输入		输出		过程	
	图像大小	补零后图像尺寸	图像大小	输出数据量	卷积核大小	卷积核个数
卷积层 1	$28 \times 28 \times 1$	$38 \times 38 \times 1$	$14 \times 14 \times 96$	18816	$11 \times 11 \times 1$	96
卷积层 2	$14 \times 14 \times 96$	$18 \times 18 \times 96$	$7 \times 7 \times 256$	12544	$5 \times 5 \times 96$	256
卷积层 3	$7 \times 7 \times 256$	$9 \times 9 \times 256$	$7 \times 7 \times 384$	18816	$3 \times 3 \times 256$	384
卷积层 4	$7 \times 7 \times 384$	$9 \times 9 \times 384$	$7 \times 7 \times 256$	12544	$3 \times 3 \times 384$	256
卷积层 5	$7 \times 7 \times 256$	$9 \times 9 \times 256$	$4 \times 4 \times 256$	4096	$3 \times 3 \times 256$	256
全连接层 1	4096		4096			
全连接层 2	4096		1024			
全连接层 3	1024		10			

根据表 4-1 可知，卷积层 1 有 96 个  $11 \times 11 \times 1$  个卷积核，即 11616 个参数。分析卷积层 1 可知，在 AlexNet 的卷积方式下，对于输入的  $28 \times 28$  的特征图的每一个像素点，都在卷积核中心所在的位置进行了一次卷积运算，所以应当保证每输入一个像素点，96 个滤波器都能完成一次卷积，也就是每输入一个像素点，需要这 11616 个参数都与以输入像素点为中心的窗内的像素点进行一次乘法，也就是需要 11616 次乘法。对于卷积层 1，进入 1 个像素点的同时，由于池化层的降采样过程，会输出 96 个通道的  $1/4$  像素点。也就是每当有 1 个像素点进入卷积层 1 时，会有 96 个通道的  $1/4$  像素点进入到卷积层 2。

这 96 个通道的  $1/4$  像素点进入卷积层 2 以后，卷积层 2 需要进行的运算是 256 个  $5 \times 5 \times 96$  的卷积核与特征图相对应像素点相卷积，但要注意，要等 4 个像素点进入卷积层 1 之后，他们才必须进行一次这样的操作。所以在整个系统输入一个像素点后，卷积层 2 需要进行  $256 \times 5 \times 5 \times \frac{96}{4} = 153600$  次乘法，然后，卷积层 2 经过池化，输出 256 通道的  $1/16$  像素点。

同样的道理，当整个系统输入了一个像素点后，卷积层 3 需要进行  $256 \times 3 \times 3 \times \frac{384}{16} = 55296$  次乘法，卷积层 4 由于没有池化层，需要进行  $384 \times 3 \times 3 \times \frac{256}{16} = 55296$  次乘法，卷积层 5 由于没有池化层，需要进行  $256 \times 3 \times 3 \times \frac{256}{16} = 36864$  次乘法。

对于卷积层 5，它在进行完卷积以后，还是  $7 \times 7$  的图，但是通过补零，变成  $8 \times 8$  的图，再进行池化，变成  $4 \times 4$  的图，所以这一步是把  $7 \times 7$  降采样成了  $4 \times 4$ ，比例为  $\frac{49}{16} = 3.0625$ ，在设计阶段近似为 3。所以，当整个系统输入一个像素点后，

卷积层 5 输出 256 个通道的  $\frac{1}{16 \times 3} = 1/48$  个像素点。这样，相当于当整个系统输入一个像素点后，有  $\frac{256}{48} = 5.33$  个像素点进入了全连接层。

对于全连接层 1，每当输入 4096 个点时，需要进行  $4096 \times 4096$  次乘法，所以每输入一个像素点，需要进行 4096 次乘法。所以当整个系统输入了 1 个像素点时，全连接层 1 需要进行  $4096 \times 5.33 = 21832$  次乘法。同理，对于全连接层 2，需要进行  $1024 \times 5.33 = 5458$  次乘法，对于全连接层 3，则需要进行  $\frac{10}{48 \times 4} \times 256 = 13.3$  次乘法。

通过这样的分析可知，整个系统输入一个像素点，整个系统需要的乘法次数为 339975 次，其中所有卷积层总共需要 312672 次乘法，所有全连接层需要 27303 次乘法，相差十倍。由此判断，卷积层的最大特点是运算密集，而全连接层最大的特点是参数密集。考虑到卷积层和全连接层有着不同的特点，因此在系统设计上，将卷积层的参数放在 FPGA 片上 Bram 中，而把全连接层的参数放在外部 DDR 中进行调用。

由于卷积层的参数约为 3MB，选用的单片 FPGA 的 Bram 存储空间约为 1.95M，同时考虑到单片 FPGA 的乘法器个数，所以需要两片 FPGA，可以将卷积层的前三层放在第一片 FPGA 上，于是第一片不需要与内存相连；将后两层卷积层和所有全连接层放在第二片 FPGA 上，第二片 FPGA 需要与存储全连接层参数的内存相连。为了按比例地给各层分配乘法器资源，进行如下计算：

首先计算输入一个像素点所需的总运算次数与总乘法器个数之比。单片 xc7k325tffg900-2 型号的 FPGA 芯片的乘法器数目为 840 个，两片共有 1880 个乘法器，为了保证工作主频足够高，按照每片最多使用 75% 左右的乘法器的标准，本次系统共使用 1400 个乘法器，所以比值为  $\frac{339975}{1400} = 243$ 。将各层每输入一个像素点所需要的乘法器次数除以这一比值，则得到各层建议的乘法器数。得到表 4-3

表 4-3 各层乘法器数量分配与处理速率  
Table 4-3 Numbers of multiplier allocation and processing rate in each layer

网络层	乘法次数	计算乘法器数目	建议乘法器数目	处理一个像素需要周期数
卷积层 1	11616	48	48	242
卷积层 2	153600	632	512	300
卷积层 3	55296	228	192	288
卷积层 4	55296	228	256	216
卷积层 5	36864	152	128	288
全连接层 1	21832	90	128	171
全连接层 2	5458	22	32	171
全连接层 3	13	0.06	1	13

根据表 4-3，卷积层第一层共有 96 个卷积核，每个卷积核需要做的乘法数目为  $11 \times 11 = 121$  个，但是第一层只分配了 48 个乘法器，因此两个卷积核共用一个乘法器，因此一共需要  $121 \times 2 = 242$  个周期。同理，卷积层第二层共有 256 个卷积核，每个卷积核需要做的乘法数目为  $5 \times 5 \times \frac{96}{4} = 600$ ，由于第二层分配了 512 个乘法器，因此每个卷积核有两个乘法器，一共需要的乘法周期为  $\frac{600}{2} = 300$ 。同理，按照这种分析，每层处理一个像素的乘法周期数如表 4-3 所示。通过表 4-3 可以看出，这种设计方式下，计算的瓶颈为卷积层第二层，因此，系统的计算速度为每 300 个周期处理一个像素点。

此外，在两片 FPGA 之间，本次设计考虑用网口进行单向通信。当整个系统输入了一个像素点时，卷积层 3 输出 384 通道的  $\frac{1}{16}$  个像素点，即  $\frac{384}{16} = 24$  个像素点。设本次设计系统工作时钟频率为 300MHz，那么可以达到 1M 个像素点每秒的速度输入，一个像素点为 8bit，所以输入端吞吐速率可达 1MB/s，所以在卷积层 3 输出后，数据速率为 24MB/s。因此两片 FPGA 的数据传输用千兆网口就可以实现。

## 4.2 基于 FPGA 的神经网络硬件加速系统实现优化

系统对每一层做独立设计，本节将分别介绍卷积层、池化层和全连接层的设计，以及实现优化策略。

### 4.2.1 卷积层

神经网络中的卷积层主要进行如图 4-2 所示的高维卷积运算。若干张输入特征图(输入特征图用 ifmap 表示,输出特征图用 ofmap 表示)与若干个卷积核(Kernel, 用  $w$  表示)进行卷积后得到若干张输出特征图。

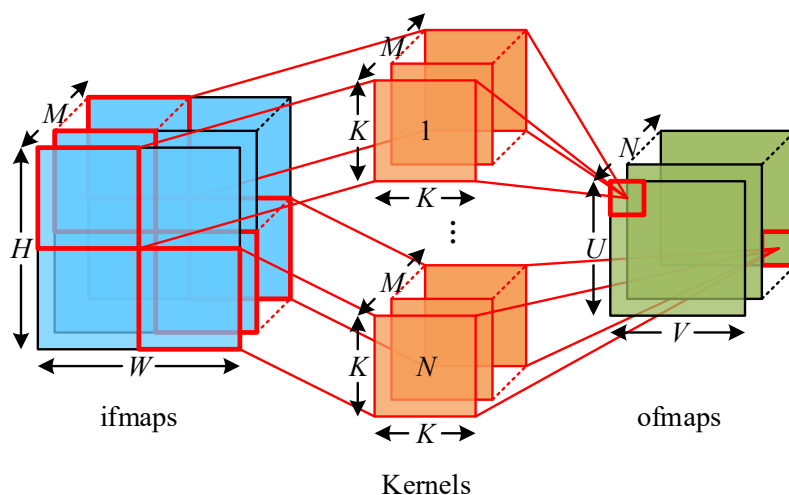


图 4-2 高维卷积示意图

Figure 4-2 High-dimensional convolution

先简化高维的卷积运算为平面二维卷积运算。假设有一张二维的输入特征图,其大小为  $4 \times 4$ , 看作是输入, 用  $X$  矩阵来表示, 对其进行卷积运算的卷积核大小为  $3 \times 3$ , 看作是系统的冲激响应  $H$ , 如图 4-3 所示。

$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$
$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$
$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$
$x_{41}$	$x_{42}$	$x_{43}$	$x_{44}$

$h_{11}$	$h_{12}$	$h_{13}$
$h_{21}$	$h_{22}$	$h_{23}$
$h_{31}$	$h_{32}$	$h_{33}$

图 4-3 输入特征图  $X$  矩阵与卷积核  $H$

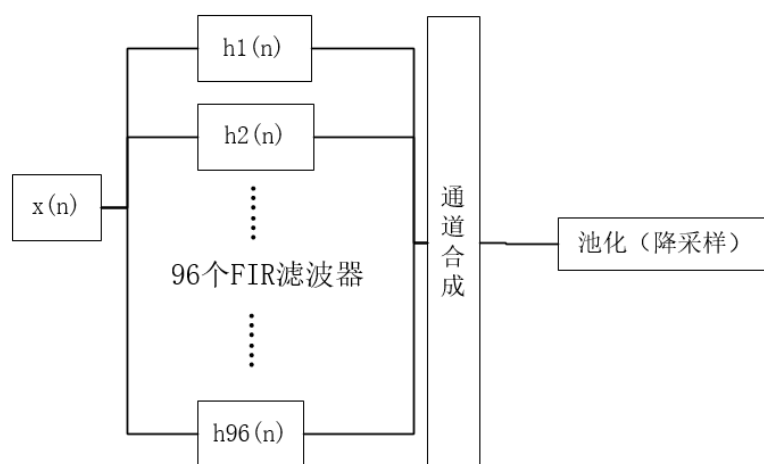
Figure 4-3 Input feature map  $X$  matrix and convolution kernel  $H$

那么用该卷积核  $H$  与  $X$  相卷的运算就应该将  $H$  从  $X$  的左上角滑到右上角，然后向下移动一行，再从最左侧滑到最右侧，这样  $H$  就把整个  $X$  扫过了一遍。在每个位置都让  $H$  的每个数值和其下面  $X$  上的对应位置的数值相乘，并将得到的 9 个积相加，得到输出图的一个像素点。在 AlexNet 中，由于所有卷积卷完以后得到的输出图的大小都与输入图的大小相同，所以要进行补零操作。如图 4-4 所示。

0	0	0	0	0	0
0	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	0
0	$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$	0
0	$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$	0
0	$x_{41}$	$x_{42}$	$x_{43}$	$x_{44}$	0
0	0	0	0	0	0

图 4-4 补零后的  $X$  矩阵Figure 4-4  $X$  matrix after zero padding

通过上述描述可知，所进行的卷积运算与传统信号处理过程中所用到的  $y(n) = x(n) * h(n)$  是非常类似的，所以可以直接把一个卷积核看作是一个 FIR 滤波器，因此，每一个卷积核都相当于传统信号处理过程中的一个 FIR 滤波器，只不过神经网络中的滤波器阶数相当高。比如对于第一层卷积运算，我们可以将它视为是如图 4-5 所示的一种信号处理手段。



每个 $h(n)$ 均为 $11 \times 11 = 121$ 阶FIR滤波器

图 4-5 卷积核的等效 FIR 结构图

Figure 4-5 Equivalent FIR structure of the convolution kernel

二维卷积模块由滑窗、乘法器复用、多通道融合以及非线性激活函数四个子模块组成。

#### (1) 滑窗

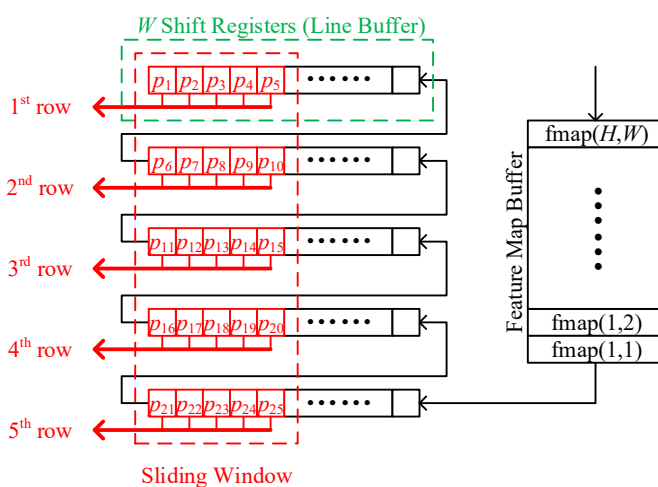


图 4-6 滑窗的硬件实现

Figure 4-6 Hardware implementation of sliding window

滑窗的硬件实现如图 4-6 所示。以  $5 \times 5$  大小的二维卷积窗口为例，上图所示的滑窗模块共有 5 个由移位寄存器链构成的行缓存（Line Buffer，图中绿色虚线框内），每一行 Line Buffer 存储了  $fmap$  中的一行（ $W$  个）像素点，取出每一行 Line Buffer 的前 5 个数据（图中红色虚线框内），相当于取出了  $5 \times 5$  窗口内的 25 个像

素点；每过一个时钟周期，移位寄存器移位一次，相当于  $5 \times 5$  的窗口以 1 为步进滑动一次。 $5 \times 5$  的卷积窗口在  $32 \times 32$  的 fmap 上滑动的效果如图 4-7 所示。

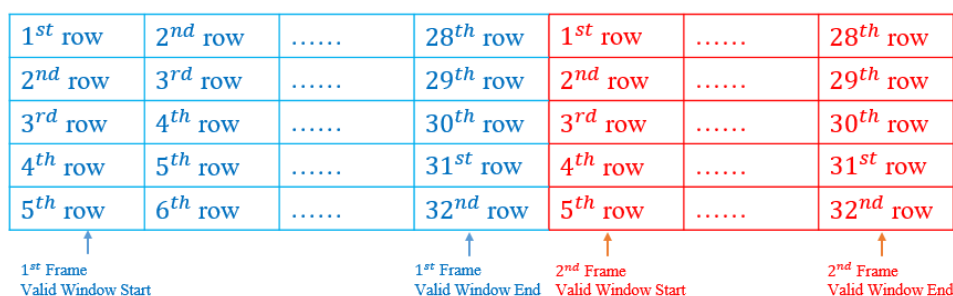


图 4-7 滑窗效果图

Figure 4-7 Result of sliding window

从算法的角度讲，为了配合步进为 1 的滑窗操作，或者从硬件的角度讲，为了配合滑窗模块中的移位寄存器的“移位”结构，卷积层中的 fmap 缓存采用了每个时钟周期吞入/吐出 1 个数据的 FIFO (First In, First Out) 结构，在 FPGA 中可以用 BRAM 或移位寄存器实现。

把 fmap 缓存吐出 1 张 fmap 的过程称为 1 帧 (Frame)，完成这 1 帧需要  $H \times W$  个时钟周期，与此同时窗口也滑动了  $H \times W$  (即 ifmap 的尺寸) 次，其中有  $H \times W$  个窗口位置是有效的。上述过程如图 4-7 所示，图中用蓝色表示第 1 帧数据，用绿色表示紧接着的第 2 帧数据。由图 4-7 可知，滑窗的效率最高可为 100%。

## (2) 乘法器复用

从表 4-3 可以看出，第一层、第三层和第五层卷积层因为乘法器的资源有限只分配了通道数一半的乘法器，所以乘法器需要两个通道分时复用。以两个卷积核共用一个乘法器为例，不同的周期两个卷积核分时复用一个乘法器，然后将计算的结果分别累加起来。整个过程如图 4-8 所示。

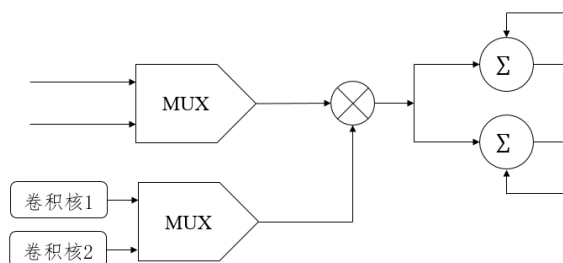


图 4-8 乘法器复用

Figure 4-8 Multiplier multiplexing



### (3) 多通道融合

多通道融合模块仅在有多张( $M > 1$ ) ifmap 的卷积层中存在, 例如对于 AlexNet 而言, C1 层仅有 1 张 ifmap, 在硬件实现时不需要多通道融合模块; 而 C2 层有 96 个通道的 ifmap, 在硬件实现时就需要多通道融合模块。

多通道融合实现的功能就是把多个二维卷积的结果进行求和和累加, 在传统的硬件实现时, 需要加法器树和累加器两个模块。以 25 个通道的加法器数为例, 加法器树的硬件实现如图 4-9 所示。累加器的硬件实现如图 4-10 所示, 第一级寄存器 (REG1) 用来存储累加过程的中间结果, 第二级寄存器 (REG2) 用来存储累加结束时的最终结果。

本文系统的实现过程中, 由于在点乘的过程中没有用到乘法器阵列而是乘法器复用, 每一个卷积核只分配了一个或者两个乘法器, 因此在同一个时钟周期同一个通道计算输出的像素点并不多, 因此可以直接只使用累加器模块将单个通道的像素点相加起来。免去了大规模加法器树的使用, 节省了资源消耗和系统时延。

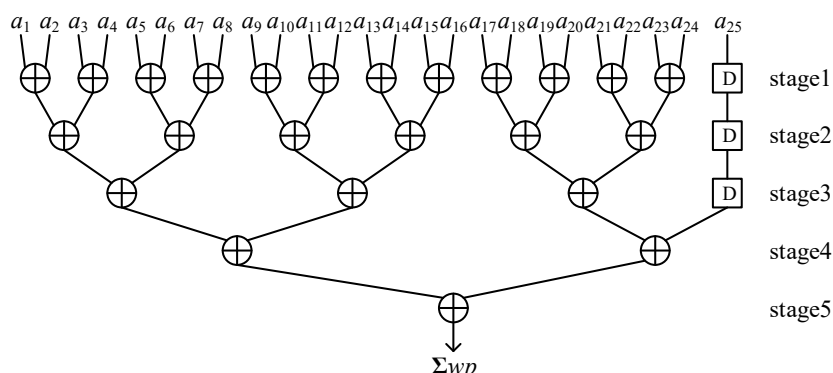


图 4-9 加法器树硬件结构

Figure 4-9 Adder tree hardware structure

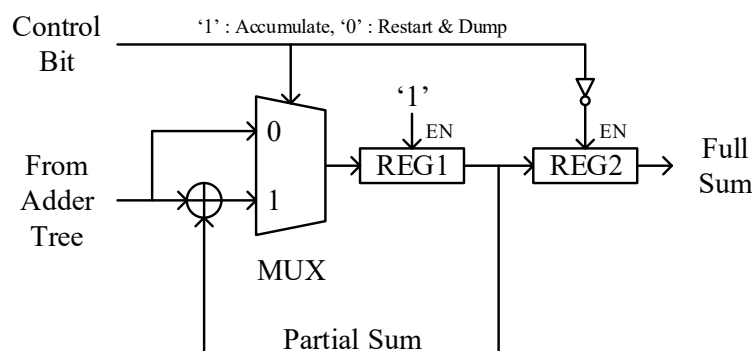


图 4-10 累加器硬件结构

Figure 4-10 Accumulator hardware structure

### (4) 非线性激活函数

非线性激活函数的硬件实现结构和非线性激活函数本身有关。对于经典的非线性激活函数  $\tanh$  而言，该函数是一个超越函数，在硬件上通常用查找表（Look-Up Table, LUT）实现，如图所示。非线性激活函数的自变量作为查找表的索引，查找表中相应的位置上存储着对应的应变变量，也即非线性激活函数的自变量-应变量的映射变成了查找表的索引-内容的映射。在 FPGA 中，查找表可以用 BRAM 实现，查找表的索引即 BRAM 的地址，查找表的深度或 BRAM 的存储容量与索引或地址的位宽有关，例如在实现时查找表的索引采用 8-bit 量化，则查找表的深度为  $2^8 = 256$ ；如果查找表的内容也用 8-bit 量化，则用于实现查找表所需的 BRAM 容量为 256B。

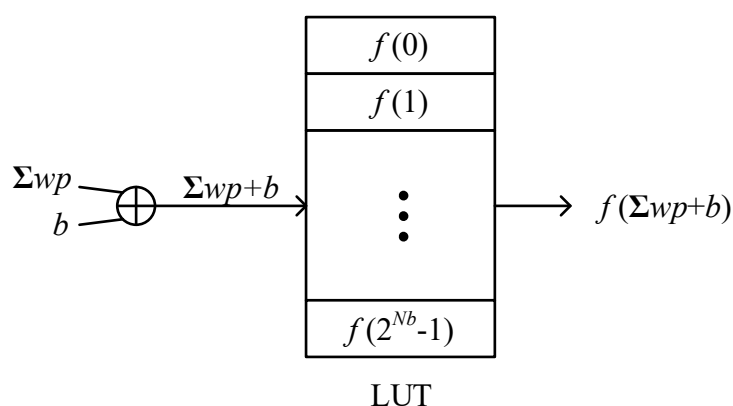


图 4-11 非线性激活函数的硬件查找表实现

Figure 4-11 Hardware lookup table implementation of non-linear activation function

但在本次设计中，AlexNet 采用如图 4-12 所示的 ReLU 作为非线性激活函数。ReLU 的实现较为简单，只需要将小于 0 的数置为零，大于零的数不变即可，则在硬件实现中可以采用非常简洁的结构实现，如图 4-13 所示。在进行非线性映射前，需要先将一张 fmap 中的每一个像素（在全连接层中则对应一个节点）加上一个偏置系数  $b$ ，不同的 fmap（节点）对应了不同的偏置系数  $b$ 。

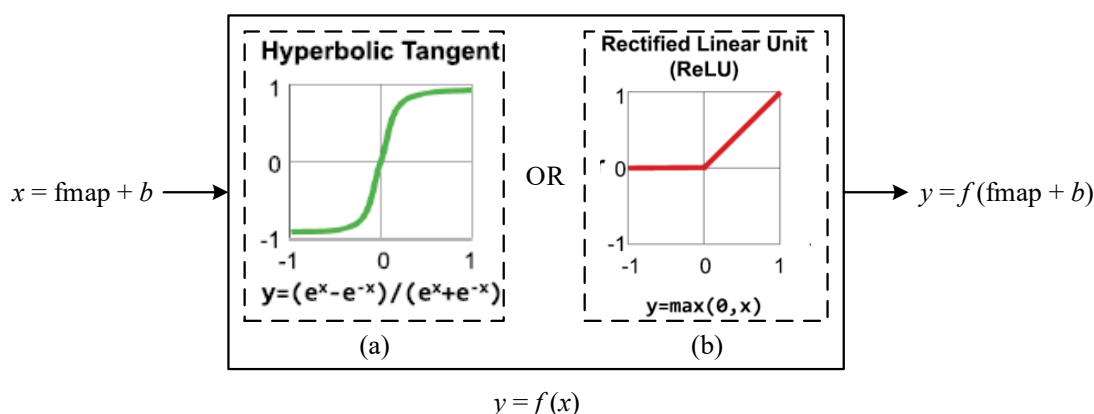


图 4-12 非线性激活函数：(a) tanh, (b) ReLU

Figure 4-12 Non-linear activation function: (a) tanh, (b) ReLU

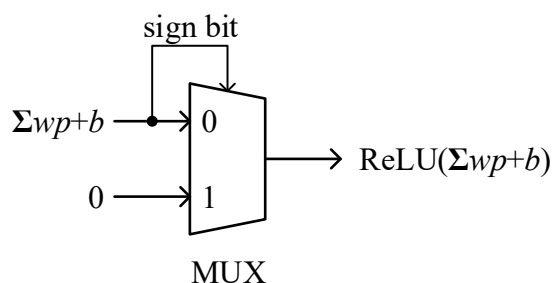


图 4-13 ReLU 的硬件实现

Figure 4-13 ReLU hardware implementation

### 4.2.2 池化层

池化模块由滑窗（Sliding Window）和池化函数（Pooling Function）两个子模块组成。

#### （1）滑窗

池化层中的滑窗与图 4-6 所示的二维卷积中的滑窗类似，区别主要在于窗口的尺寸与滑动步进不同。在本次设计的 AlexNet 中，滑窗大小均为  $2 \times 2$ ，步长均为 2。

#### （2）池化函数

对于本次设计中采用的最大值池化函数，由于滑窗大小均为  $2 \times 2$ ，所以只需要进行 4 个数的比较。传统硬件实施方案一般采用先比较后存储的方法，设计一个寄存器，将四个数据存储起来，然后每四个周期吐出一个数据，在这四个周期中取出输入的四个数中的最大值即可。硬件实施结构如图 4-14 所示。



图 4-14 传统池化层实施方案

Figure 4-14 Traditional pooling layer implementation

由于在本系统的设计中，每个通道每个周期只有一个输出结果，因此在输入了  $a_1$  和  $a_2$  两个像素点之后，就可以两两比较，得到一个最大值  $a_5$  暂存，然后再等后两个像素点  $a_3$  和  $a_4$  输出以后再两两比较，得到最大值  $a_6$ ，最后再  $a_5$  和  $a_6$  比较得到最大值输出池化结果。这样只需要比较两次，消耗两个时钟周期，缩短了系统时延。池化层的硬件如图 4-15 所示。

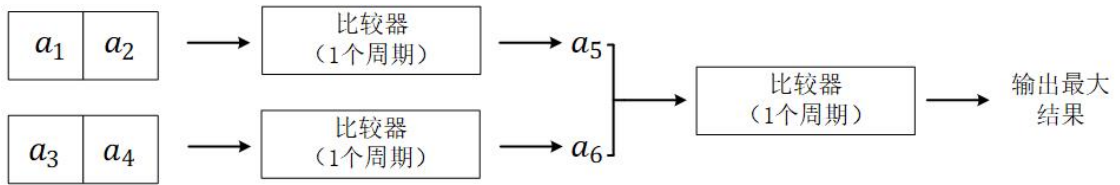


图 4-15 本系统提出的池化层实现方案

Figure 4-15 Proposed pooling layer implementation

### 4.2.3 全连接层

从 4.1 的分析中可以得知，全连接层的特点是参数密集，因此把全连接层 1 和全连接层 2 的参数存放在外部的 DDR 中。在表 4-2 中可以看出全连接层 1 的输入是 4096 个像素点，全连接层 1 的参数规模是  $4096 \times 4096$ ，输出是 4096 个像素点。因此，在本质上全连接层的运算可以看作是矩阵乘法。全连接层 1 的输入看作是向量  $v_{1 \times 4096}$ ，把全连接层 1 的参数看作是  $w_{4096 \times 4096}$ ，因此全连接层 1 的运算过程就是矩阵乘法  $v \times w$ 。因此整个过程可如图 4-16 所示。同理，全连接层 2 的运算过程可以看作是  $v_{1 \times 4096} \times w_{4096 \times 1024}$ ，全连接层 3 的运算过程可以看作是  $v_{1 \times 1024} \times w_{1024 \times 10}$ 。

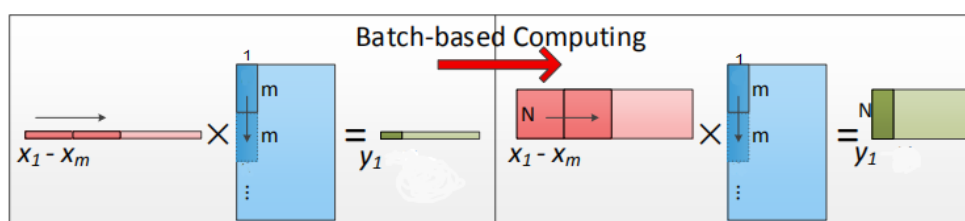


图 4-16 全连接层矩阵乘法

Figure 4-16 Fully connected layer matrix multiplication

如 4-16 左图所示，本项目中的全连接层的运算是通过复用乘法器来完成的，拿全连接层 1 举例，图中的  $m$  等于 128，这也是使用的乘法器的数量，每一次运算，输入缓存 RAM 会将对应的 128 个输入像素点输出到乘法器阵列模块中用于计算，然后会在下一个周期将下一组的 128 个输入像素点传输到乘法器阵列中去完成运算，以此类推，最后将多组数据的计算结果累加得到输出结果。但是由于全连接层有着带宽的限制，全连接层数据输入和处理的速度要大于 DDR3 参数读取的速度，因此就需要引入批处理的方式来帮助降低全连接层运算所需要的带宽。

批处理方法的具体操作步骤见 4-16 右图，将每一个卷积层 5 的输出定义为一帧，一帧有 4096 个像素点，全连接层每一次运算都需要存够一定帧数的数据之后才会开始计算在本项目中，批处理参数被设置为 8，也即是说，每存 8 帧数据才会开始全连接层的运算。这样就可以使所需的带宽降低为原先的八分之一

本文设计的系统把参数存储在片外的 DDR3 上，这就带来了 DDR3 指令消耗和读取速度匹配的问题。因为全连接层 1 和全连接层 2 的参数都存储在都一个 DDR3 上，而 DDR3 在读取的过程中如果有大量的换行换地址的操作将会严重增加带宽的开销从而使 DDR3 的读取速度变慢，不能满足设计要求。于是，本文设计中将全连接层 1 和全连接层 2 的参数存储在一次读取出来的 512 个 bit 中，而由于全连接层 1 的参数尺寸是  $4096 \times 4096$ ，全连接层 2 的参数尺寸是  $4096 \times 1024$ ，两者有 4:1 的关系，因此取 512bit 的前 480 个 bit，将其划分为 4:1 的两个空间，分别存放全连接层 1 和全连接层 2 的参数。

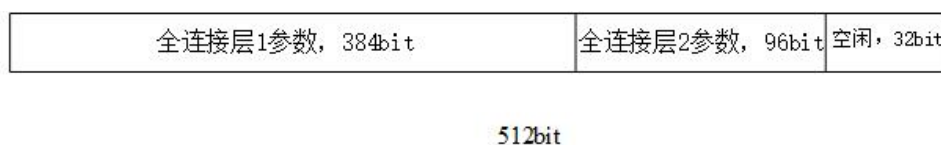


图 4-17 全连接层参数分布

Figure 4-17 Fully connected layer parameter distribution

除此之外,为了能让全连接层 1 和全连接层 2 协同处理,以达到同时读取 DDR3 中参数的目的,需要有对应的控制信号。在本文提出方案中,采用状态机来完成这一控制。全连接层 1 和全连接层 2 的状态机流程图如图 4-18 所示。首先当存满 8 帧数据之后,开始批处理,对全连接层 1 的数据进行运算,此时发出 DDR3 的读操作,读出参数,但是此时全连接层 2 的运算还没有开始,因此先将全连接层 2 的参数暂存在 FPGA 的片上 BRAM 中。接着进入全连接层 1 和全连接层 2 的共同运算的过程,此时仍然发送 DDR3 的读取指令。然后判断是否读取到了 DDR3 存储的最后一个地址。当没有读取到最后一个地址的时候,证明参数没有读取完,因此将继续全连接层 1 和全连接层 2 的运算;当已经读取到最后一个地址的时候,证明已经到了最后一个 512bit,接下来将只有全连接层 2 进行运算。最后全连接层 2 运算结束之后回到批处理的状态。

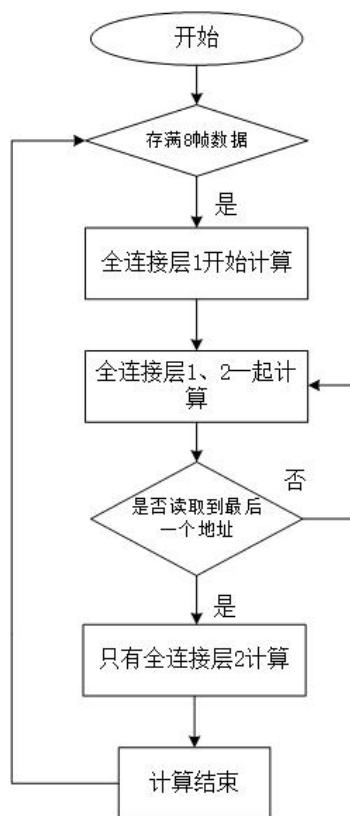


图 4-18 全连接层状态机

Figure 4-18 State machine of fully connected layer

全连接层的硬件电路结构图如图 4-19 所示。

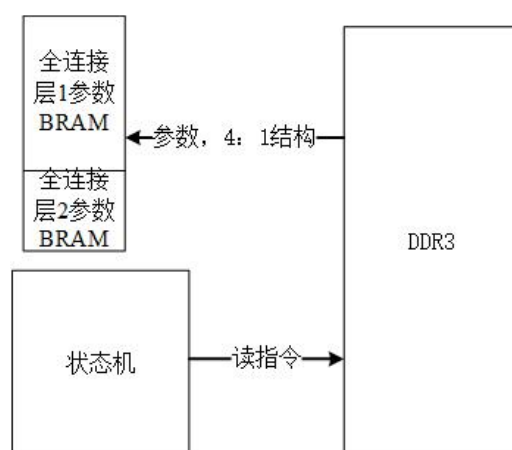


图 4-19 全连接层硬件结构

Figure 4-19 Fully connected layer hardware structure

### 4.3 定点量化策略

在软件平台上，为了保证计算精度，整个网络参数均采用双精度浮点数表示。本小节针对得到的参数进行数据量化，采用定点数表示这些参数，以保证网络占用的存储空间最小。

本文采用的量化实现方案分为如下三个部分：首先分析网络参数的最大值和最小值，得到取值范围，从而确定定点数据采用的位宽，保证其不会溢出；其次，计算选择不同精度时得到的量化数据与原来浮点数据相比的误差，选择误差相对较小的精度作为备用精度选项。以上两步中选用的量化位宽和精度应该满足式(4-1)：

$$2^{W-P-1} > \max(|D_{max}| \times 2^P, |D_{min}| \times 2^P) \quad (4-1)$$

其中： $P$ 为量化精度； $W$ 为量化位宽； $D$ 表示量化前的浮点数据。最后，将上一步筛选得到的量化数据替换原来的浮点数据，在软件平台上测试，选取相比于原来网络精度下降最小的一组数据，作为我们最终硬件实现所采用的定点数据。先确定量化精度范围再替换参数重新测试的方案，在保证量化精度的前提下，降低了对测试集进行重新测试的次数，减少了确定量化参数所需的时间。数据量化使软件平台中网络参数从 64 位浮点数转化为 8 位定点数，使存储网络所需的存储空间下降为原来的 $\frac{1}{8}$ ，而且从浮点运算转换为定点运算降低了硬件实现所需的计算资源和功耗开销。

在本文的设计中，没有对每一层的计算结果采取统一的量化策略，而是针对每一层的参数特点选取不同的量化位宽，采用动态量化。表 4-4 给出了每一层计算

结果量化选取的 $P$ 和 $W$ 。

表 4-4 每一层参数量化指标  
Table 4-4 Parameters of each layer

网络层	$P$ (量化精度)	$W$ (量化位宽)
卷积层 1	4	8
卷积层 2	1	8
卷积层 3	3	11
卷积层 4	7	15
卷积层 5	9	17
全连接层 1	12	20
全连接层 2	15	23
全连接层 3	17	25

#### 4.4 系统性能分析

输入 MNIST 数据集作为测试数据。FPGA 的处理主频设置为 250MHz，每隔 300 个时钟周期输入一个像素点，输入一张  $28 \times 28$  大小的测试图需要  $28 \times 28 \times 300$  共 235200 个时钟周期，共 940800ns。由于系统的吞吐率达到百分之百，因此系统推断一张图所需要花费的时间为 940800ns，频率为 0.00106MHz，系统的处理速度达到 1060FPS，即 1s 可以处理 1060 张图片。

系统的测试方法为：将系统与上位机相连，使用 PCIe 接口给 FPGA 发送数据，在 Vivado 端通过抓取数据查看 FPGA 内部的推断的数据结果，同时将推断数据通过千兆网口传输到上位机，图 4-1 给出了使用 Vivado Debugger 抓取的 FPGA 推断数据结果，其中 tx\_sent\_data 是发送的推断结果，tx\_sent\_start 表示发送使能。在 PC 端上位机中，使用 Matlab 将推断结果与标签对比，然后将数据可视化界面看推断结果，记录推断的正确率。图 4-1 给出了 FPGA 的推断结果的发送。图 4-2 为 Matlab 将 FPGA 推断结果和标签对比的过程。



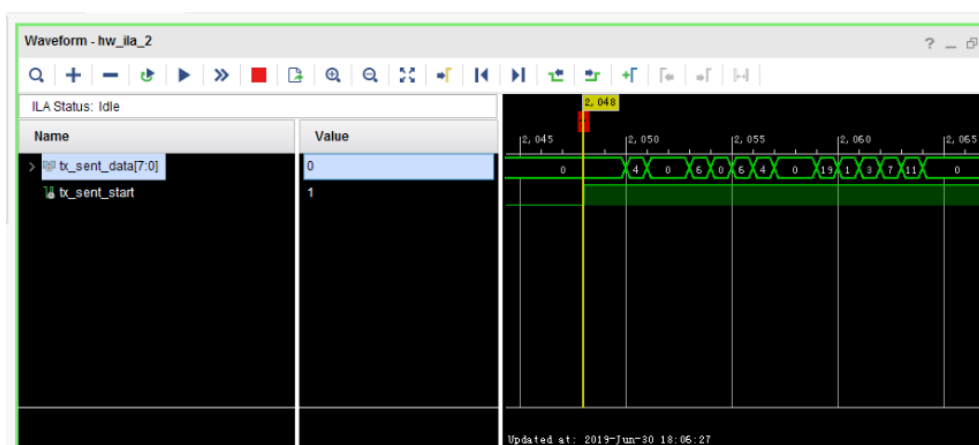


图 4-1 Vivado Debugger 抓取 FPGA 推断结果

Figure 4-1 FPGA inference results captured by Vivado Debugger

推断结果为: 7

7

图 4-2 Matlab 推断结果对比

Figure 4-2 Matlab inference result comparison

输入多张图测试，记录推断结果的正确识别率，统计结果如表 4-5 所示，识别率曲线如图 4-3 所示。从 4-3 中可以看出，结果表明测试结果正确的识别率在 94% 以上，该方案具有良好的推断能力，当输入数据为 20000 张图像时，推断的正确率最大。

表 4-5 识别率  
Table 4-5 Accuracy

输入图像数	识别率
1000	89%
5000	94.26%
10000	95.17%
20000	95.4%
30000	95.31%
40000	95.21%
50000	94.96%
60000	94.80%

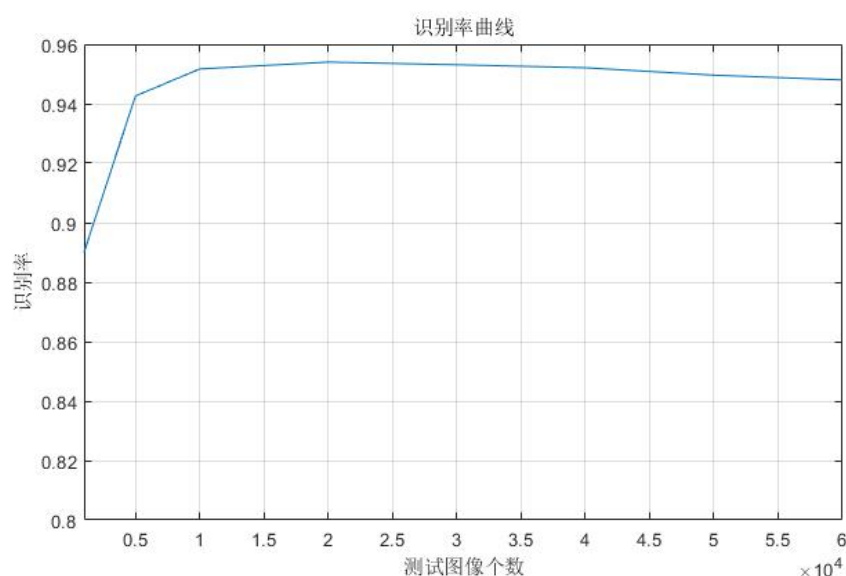


图 4-3 识别率曲线

Figure 4-3 Accuracy curve

本节使用 Xilinx Vivado 软件进行了整个工程的综合编译，由综合报告可知，该系统占用了 220932 个 LUT，285786 个触发器，1262 个 DSP 乘法器以及 822 个 BRAM。其中 FPGA 内部存储资源以及 DSP 资源使用较多，这是因为在本文提出的方案中，实现过程中存在着大量的数据缓冲，以及参数存储也用到了大量的片上存储资源，并且卷积层和全连接层所用到的 8 点的定点乘法运算均采用 FPGA 内部的 DSP 来完成。FPGA 资源消耗以及占用比如图 4-4 所示。

Resource	Utilization	Available	Utilization %
LUT	90167	203800	44.24
LUTRAM	11363	64000	17.75
FF	175686	407600	43.10
BRAM	402	445	90.34
DSP	752	840	89.52
IO	19	500	3.80
GT	8	16	50.00
MMCM	3	10	30.00

(a) FPGA1 资源占用

(a) FPGA1 resource utilization

Resource	Utilization	Available	Utilization %
LUT	22636	203800	11.11
LUTRAM	3181	64000	4.97
FF	34453	407600	8.45
BRAM	88	445	19.78
DSP	128	840	15.24
IO	120	500	24.00
GT	8	16	50.00
MMCM	3	10	30.00
PLL	1	10	10.00

(b) FPGA2 资源占用

(b) FPGA2 resource utilization

图 4-4 FPGA 资源占用

Figure 4-4 FPGA resource utilization

接下来将 FPGA 的系统实现与 CPU、GPU 的通用硬件平台作对比。在计算速度和功耗分析中，采用 FPGA 的实现方案与 CPU 和 GPU 的实现方案具有明显的效能优势。针对  $28 \times 28$  尺寸的 MNIST 手写数据集的计算过程中，在 12.595W 的开发板全功耗下可以获得 1060FPS 的处理速度。两片 FPGA 的功耗如图 4-6 所示。

CPU 在相同网络结构下的推断 256 张图片的速度如图 4-5 所示, 可以得出 CPU 的推断速度为 64FPS, 因此基于 FPGA 的神经网络推断速度已经高于 CPU。同理, 可以得到 GPU 平台的推断速度, 为 1736FPS。各硬件平台的具体的性能参数如表 4-5, 效能比较如图 4-7 所示。从图 4-7 可以看出, FPGA 在功耗效能方面明显优于 CPU 和 GPU, 仅为 CPU 的 9.39%, 为 GPU 的 11.53%。

```

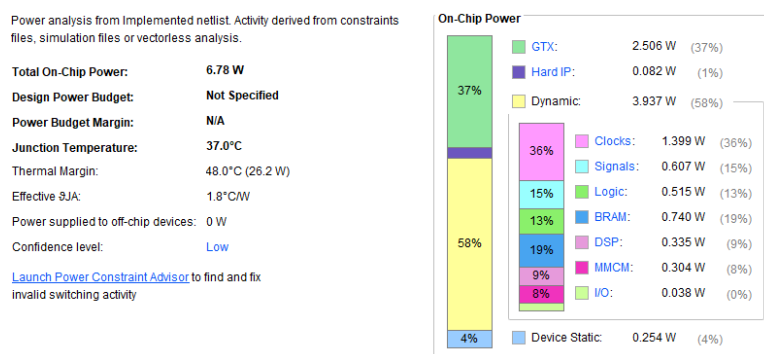
/Users/gongyu/PycharmProjects/test_5/AlexNet.py:148:
.process_time instead
start = time.clock()
Testing Accuracy: 0.9375
/Users/gongyu/PycharmProjects/test_5/AlexNet.py:150:
.process_time instead
推断所用时间为: 3.9776580000000195
end = time.clock()

Process finished with exit code 0

```

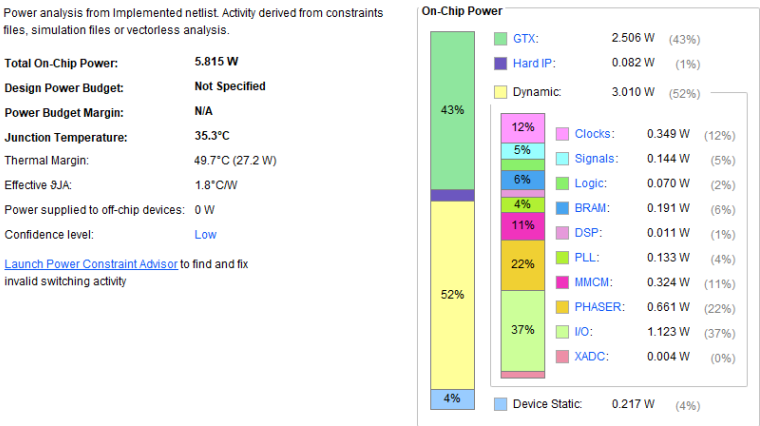
图 4-5 CPU 推断 256 张图消耗的时间

Figure 4-5 Time consumption by the CPU to infer 256 images



(a) FPGA 1 功耗

(a) FPGA 1 power consumption



(b) FPGA 2 功耗  
(b) FPGA2 Power consumption  
图 4-6 2 片 FPGA 功耗

表 4-5 不同硬件平台性能比较  
Table 4-5 Performance comparison of different hardware platforms

	硬件平台	FPS	功耗	效能(功耗/图数)
GPU/TensorFlow	GTX1070	1736	180W	0.103
CPU/TensorFlow	Intel I7-6700K	64	81W	1.2656
FPGA	KC705	1060	12.595W	0.01188

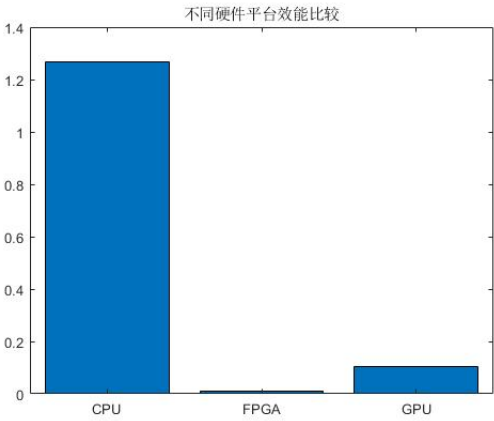


图 4-7 各硬件平台效能比较图  
Figure 4-5 Comparison of the performance of each hardware platform

4.5 本章小结

本章提出了一种适用于 5G 大带宽大数据量智能通信系统的基于 FPGA 的神

神经网络硬件加速系统，相比其他硬件平台，本文提出的系统架构有更低的计算复杂度和计算延迟，同时具有低功耗和易移植性。本章的主要内容包括：

(1) 基于 FPGA 的神经网络硬件加速系统架构设计。分析了神经网络的结构，提出了神经网络的并行计算策略，充分利用 FPGA 中并行化和流水线的特点，合理分配乘法器数目，达到神经网络的硬件加速。

(2) 给出了该系统中硬件加速的关键技术。针对神经网络的卷积层、池化层和全连接层，提出优化策略和实施方案，采用共用卷积核、同步参数等策略实现了神经网络的硬件加速。

(3) 给出了神经网络硬件加速的定点量化策略。根据不同层的结构特点做了不同的量化策略，在保证推断精度的同时减少了计算复杂度和系统的延迟。

(4) 计算机仿真结果表明，本章提出的基于 FPGA 的硬件加速方案的在 MNIST 数据集上最高能达到 95.4% 的准确率；在延迟方面，FPGA 的计算速度可达 1060FPS，远远高于 CPU 硬件平台，略低于 GPU 的硬件平台；在功耗方面，FPGA 的推断每张图的功耗仅为 CPU 的 9.39%，为 GPU 的 11.53%，具有低功耗的特点。

综上所述，本章提出的基于 FPGA 的神经网络硬件加速方案具有高准确率、低延迟、低功耗的特点，且 FPGA 大量购入平均价格低、便携性强的特点，更适应于神经网络的边缘计算加速，适用于 5G 大带宽大数据量的特点，对 5G 智能通信系统的实现和部署有重要的意义。

## 第五章 多通道宽带数字信号处理硬件平台设计与实现

针对 5G 智能宽带通信系统对数字信号处理和算法实现的需要，本章设计并实现了一套适用于大带宽 5G 智能通信的数字信号处理硬件平台，详细描述了该硬件平台的系统总体和模块化单板硬件的设计，最后分析总结了该硬件平台所能达到的性能。

本章 5.1 节分析了大带宽 5G 智能通信系统对数字信号处理硬件平台的功能与性能需求，并制定了相应的系统设计目标；根据系统设计目标，5.2 和 5.3 节分别给出了系统总体和模块的硬件设计；5.4 节分析总结了该系统的性能；5.5 节对本章的主要内容进行了总结。

### 5.1 系统设计目标

在 5G 智能宽带通信系统中，数字信号处理是该系统的核心，而合适的硬件平台是数字信号处理的实现的基础。在 5G 智能通信系统中，大带宽的特点对数字信号处理的平台有了新的要求。表 5-1 给出了 5G 智能通信系统对数字信号处理硬件平台的功能和性能需求，以及本章针对这些功能和需求采取的对应措施。

表 5-1 数字信号处理硬件平台设计需求分析  
Table 5-1 Digital signal processing hardware platform design requirements analysis

功能需求	性能需求	技术保障措施
信号接收	大带宽	采用高速 ADC 器件
	多通道	集成多个 ADC 通道
	可扩展	模块化硬件设计
信号处理	高速	采用高性能 FPGA 作为数字信号处理器
	并行	
	实时	
	可编程	
基带传输	高速	采用高速收发器
	稳健	均衡技术，屏蔽技术
数据存储	高速	采用大容量高速 DDR3-SODIMM

	大容量	
	轻便	

根据上文对 5G 智能通信系统中的数字信号处理硬件平台的设计需求，表 5-2 给出了分析后根据需求制定出的相应硬件设计目标。

表 5-2 数字信号处理硬件平台设计目标  
Table 5-2 Digital signal processing hardware platform design goals

功能	性能	设计目标
信号接收 (ADC)	通道数	≥8，可扩展
	信号频率范围	550~950MHz
	转换率	≥1Gsps
	有效量化位数	≥7bits
信号处理	处理能力	500Msps 并行数据流实时处理
	可编程能力	可编程
基带传输	以太网传输速率	1000Mbps
	QSFP+光收发器传输速率	6.25Gbps
	PCIe 接口传输速率	32Gbps
数据存储	存储容量	2G
	运行速率	1600Mb/s

5.2 系统硬件设计

为了满足表 5-2 给出的硬件平台设计目标，本文的硬件系统设计方案采用一种模块化的设计方法：每一块单板硬件模块都是同构的，具有相同的结构和功能，都有相同的信号发射/采集、信号处理、基带传输和数据存储的功能；当实际应用规模需求较大时，可以将多个单板硬件模块通过以太网或 QSFP+光口进行并联或级联以扩展系统硬件的发射/接收通道数、信号处理能力以及基带传输能力；所有的单板硬件模块均配备标准的 PCIe 2.0 接口，可以与上位主机相连，以实现系统控制、模块间互联以及即插即用式的系统扩展。硬件平台的系统架构如图 5-1 所示。

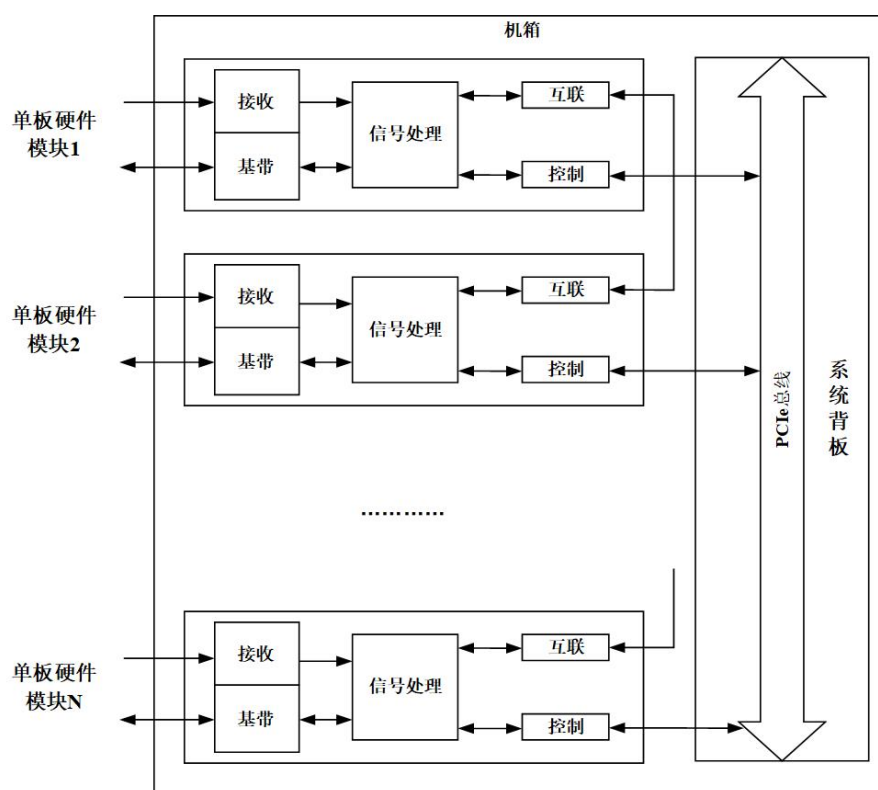


图 5-1 数字信号处理硬件平台系统架构

Figure 5-1 Digital signal processing hardware platform system architecture

### 5.3 模块化单板硬件设计

#### 5.3.1 单板硬件设计

本文给出的单板硬件模块化架构如图 5-2 所示：单个单板硬件系统集成了 8 个宽带信号接收（ADC）通道；一片高速数 FPGA 负责实现数字信号处理算法；接口配有标准的高速 SFP 光口收发器和千兆高速以太网口；单板配有标准 PCIe 2.0 x8 接口与上位主机的 PCIe 总线相连接，实现主机对单板系统的控制和两者之间的通信信号传输；PLL 模块为系统的同步时钟模块，为单板系统提供时钟，同时用来支持单板系统的扩展。



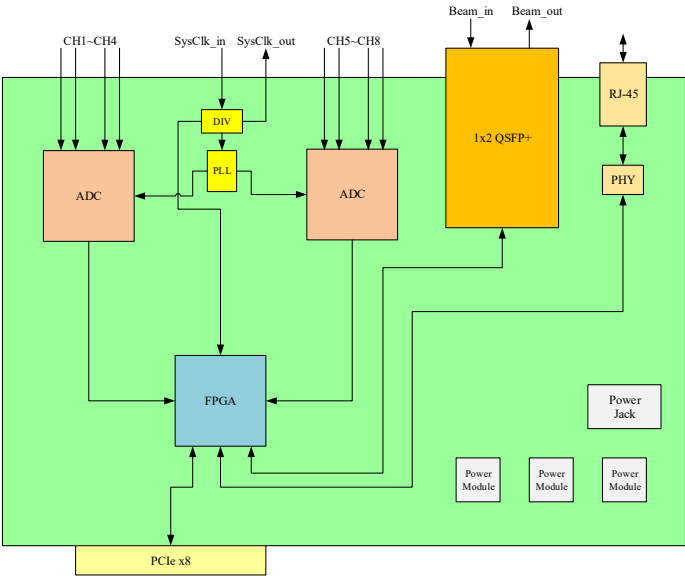


图 5-2 模块化单板硬件架构

Figure 5-2 Modular circuit board architecture

表 5-3 模块化单板硬件功能与核心器件选型

Table 5-3 Functions and corresponding key components of the modular circuit board

功能区	完成功能	主要器件
信号收集功能区	原始数据采集	EV8AQ160 ETC1-1-13TR
FPGA 综合处理&控制功能区	信号处理 外围器件控制	XC7K325T-2FFG900I
DDR3 存储功能区	数据存储	SO-DIMM DDR3 内存条
PCIe 接口功能区	主机控制 数据传输	PCIe edge connector
千兆网口功能区	数据传输	88E1111 HFJ11-1G01E
SFP 光口功能区	数据传输	1x2 QSFP+光模块
时钟链路功能区	提供时钟信号	25MHz 有源晶振 SiT9102 NBXDBA019 LTC6951 LTC6954

表 5-3 总结了模块化单板硬件设计中各部分功能区、他们对应实现的功能和

所需要的核心器件。下面分别对其中的信号收集/发送，FPGA 综合处理和控制和基带信号传输的设计做介绍。

5.3.2 接收功能设计

该功能区的核心器件 ADC 芯片选用 E2V 公司的 EV8AQ160，该款芯片单片集成 4 个通道，信号采样部分使用 2 片 ADC 同时对 8 路输入信号进行数据采集。EV8AQ160 的重要性能如表 5-4 所示<sup>[31]</sup>。为了尽可能提高信号接收的灵敏度，需要对模拟前端进行设计，采用合适的匹配阻抗减少对输入功率的反射。因此，模拟前端的设计是信号接收功能的硬件设计难点。图 5-3 给出了模拟前端电路设计。

表 5-4 EV8AQ160 基本性能参数  
Table 5-4 EV8AQ160 typical performance parameters

参数	典型值	
输入信号	共模输入电压	1.8 V (typ)
	差分摆幅	250 mV (typ)
转换结果输出	差分摆幅	320 mV (typ)
	共模输出电压	1.28V (typ)
采样性能 (@620MHz)	采样率	200-1250MSPS
	量化位数	8 Bit
	有效位宽	7.3 Bit
	SNR	45 dB
	SFDR	56 dBc

为了保证 2 片 ADC 芯片的同步性，该部分模块做出以下设计：FPGA 接收采样数据时采用 50MHz 基准时钟输入，通过 FPGA 内部锁相环倍频产生一个 500MHz 时钟作接收 ADC 数据之用。2 片 ADC 输出的时钟仅作备份使用。

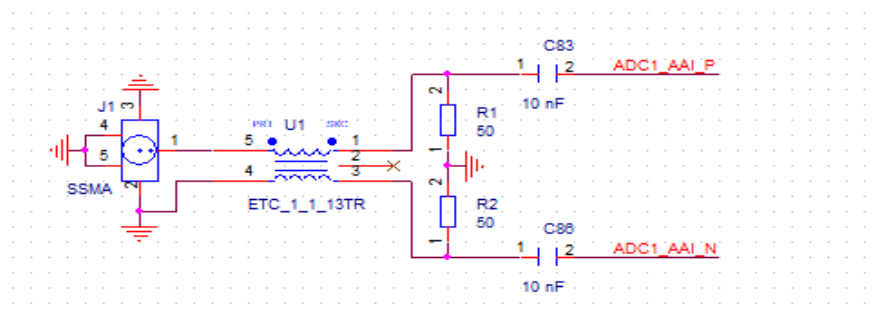


图 5-3 EV8AQ160 模拟前端电路设计  
Figure 5-3 EV8AQ160 analog front-end circuits design

### 5.3.3 数字信号处理功能设计

在 5G 智能通信系统中，由于带宽的增大，数据流量剧增，相对应的数字信号处理器要具备高速和并行的处理能力；同时 5G 系统中的低延迟特点要求数字信号处理器具有实时处理的特点；另外，由于智能通信系统中的神经网络的广泛应用，大量的参数需要更迭，这要求数字信号处理器要有灵活性和可编程性以支持系统信号处理算法的升级和迭代。传统通用的处理器具有优秀的可编程性，但不具备实时性的特点；专用处理器具有高速和实时的处理能力，但灵活性和可编程性差；FPGA 作为一种介于通用处理器和专用处理器之间的折中，不仅具有高速、实时的处理性能，同时又可以灵活可编程，满足 5G 智能通信系统的数字信号处理的需求。本文选用了 Xilinx 公司的高性能 FPGA XC7K325T-2FFG900I 作为系统的数字信号处理器。该器件采用 28nm 制程，支持最高 550MHz 的时钟频率，并且具有丰富的逻辑、算术、存储和 I/O 资源，如表 5-所示<sup>[32]</sup>。

表 5-5 XC7K325T-2FFG900I 资源概览  
Table 5-5 XC7K325T-2FFG900I resources overview

资源	数量
可编程逻辑块	50950
25*18 乘法器	840
块存储器	16020Kb
以太网 MAC	2
GTP 收发器	16
可编程 I/O	500

图 5-5 为 5G 智能通信系统的数字信号处理算法在本文设计单板硬件上的实现过程。FPGA 所具备的并行化和流水线结构对含有海量数据的大带宽数字信号处理非常有利，同时 FPGA 可以实现通信系统中的多通道架构，便于扩展。此外，FPGA 具备丰富的接口，具备多种电气特性，可以与其他器件十分方便的连接。

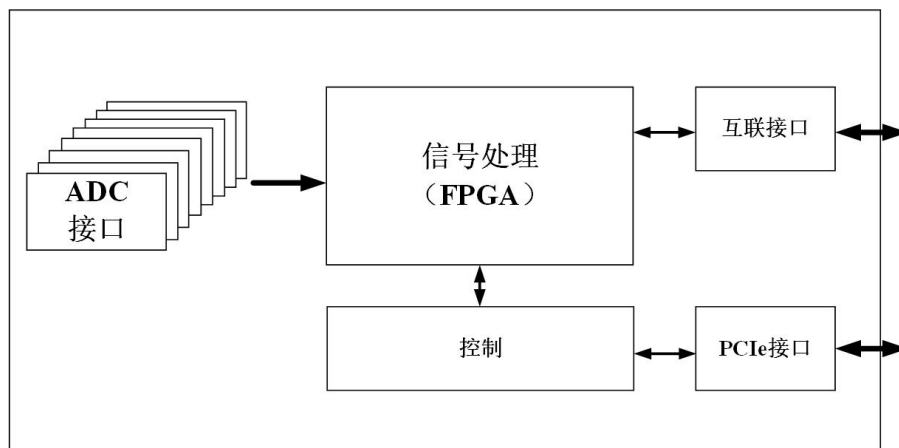


图 5-3 数字信号处理算法的 FPGA 实现框图

Figure 5-3 FPGA implementation of digital signal processing algorithms

### 5.3.4 基带信号传输功能设计

#### (1) 1000Mbps 以太网传输

千兆网口功能区实现千兆以太网接口的物理层协议，核心器件 PHY 芯片为 MARVELL 公司的 Alaska 系列吉比特以太网收发器 88E1111，该器件应用广泛，支持多种类型的 MAC 接口。MAC 接口采用 RGMII 接口；MDI 接口采用 1000BASE-T 端口<sup>[33]</sup>。图 5-6 显示了千兆网口功能区的连接框图。

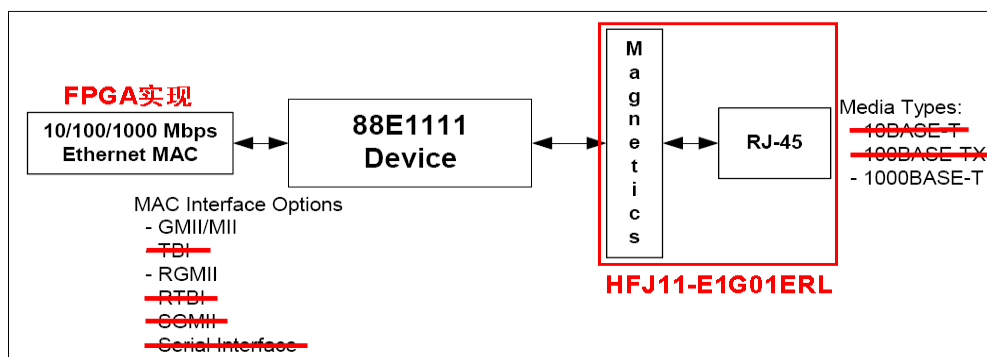


图 5-4 88E1111 (PHY) 与 FPGA (MAC) 之间的 RGMII 接口

Figure 5-4 RGMII interface between 88E1111 (PHY) and FPGA (MAC)

## (2) 6.25Gbps 光传输

光口收发模块硬件核心是 2 片光口收发器 QSFP+模块, 该核心器件是武汉电信器件公司 (WTD) 制造的高速光收发模块 RTX320-550, 完成收发信号的光电/电光转换, 其发射波长为 1310nm, 接收波长范围为 1250-1620nm, 收发数据率达到 6.25Gbps, 收发距离 2Km<sup>[34]</sup>。本方案中, 其收发端直接连接在 FPGA 的 GTX 端口<sup>[35]</sup>, 控制端直接连接到 FPGA 的普通 IO<sup>[36]</sup>。其接口电路设计如图 5-5 所示。

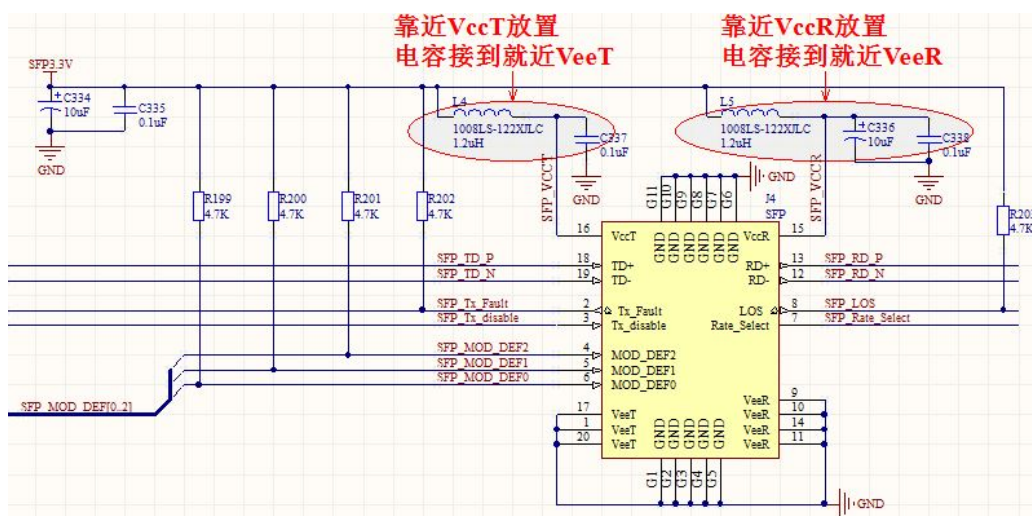


图 5-5 QSFP+光收发器接口电路设计

Figure 5-5 QSFP+ optical transceiver interface circuits design

## (3) PCIe 传输

相比 VME 和 VPX 接口, VPX 是 VME 的技术升级版<sup>[37][38]</sup>, 但是 VPX 接口使用需要自己在处理板的基础上设计一片背板来使用 VPX 接口<sup>[38]</sup>, 不如 PCI-E 接口即插即用使用方便<sup>[39]</sup>, 故此本文的系统设计采用 PCIe 接口。PCIe 附加卡采用金手指设计, 主要在于其机械设计。为了简化本次单板硬件的电源设计, 本文设计中不采用 PCIe 供电, 同时电源部分的设计也可以因此简化。PCIe 接口长度和连接见图 5-8 所示。

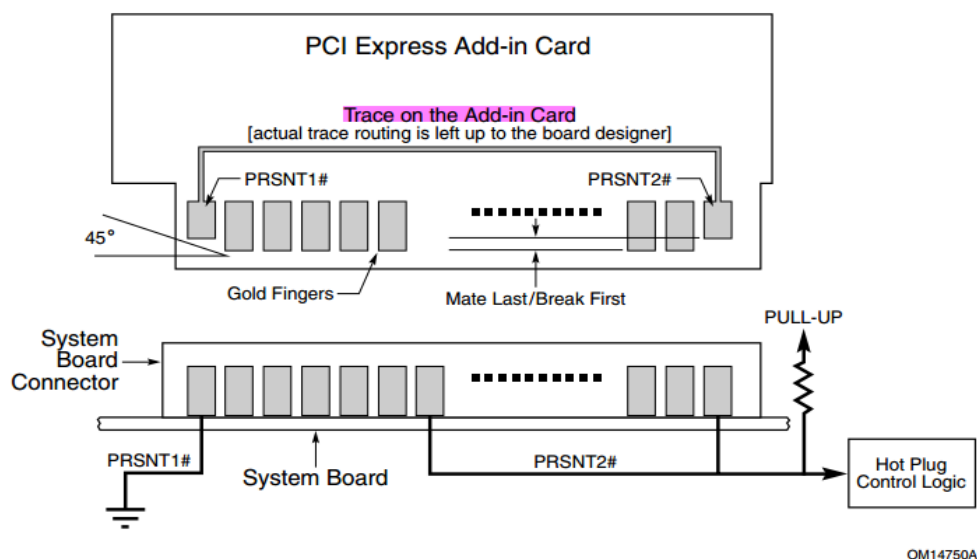


图 5-6 PCIe 接口长度和连接

Figure 5-6 Diagram of OSERDES/ISERDES transmission

## 5.4 系统性能测试及分析

根据上述设计得到电路板结构为 12 层，外形尺寸为 233.35mm\*160mm，符合 6U 欧卡标准<sup>[40]</sup>。表 5-6 给出了硬件平台的设计目标与实现结果的对照，图 5-10 为 ADC 的性能测试结果。对该单板硬件性能测试所得到的性能水平如下：

(1) 信号接收：单模块集成 8 个 1Gsps 转换率的 ADC 通道，并且支持模块间并联以实现通道扩展，在目标频率范围内（550-950MHz）有效量化位数达到 7.3bits 以上。

(2) 数字信号处理：支持对 500Msps 并行数据流的高速实时处理，并且可重复编程，具备灵活性。图 5-9 是系统对本文提出 DBF 算法处理和对 AlexNet 网络边缘计算加速的实时数据抓取。在两个算法中，系统主频都在 500M，能正确得到结果。

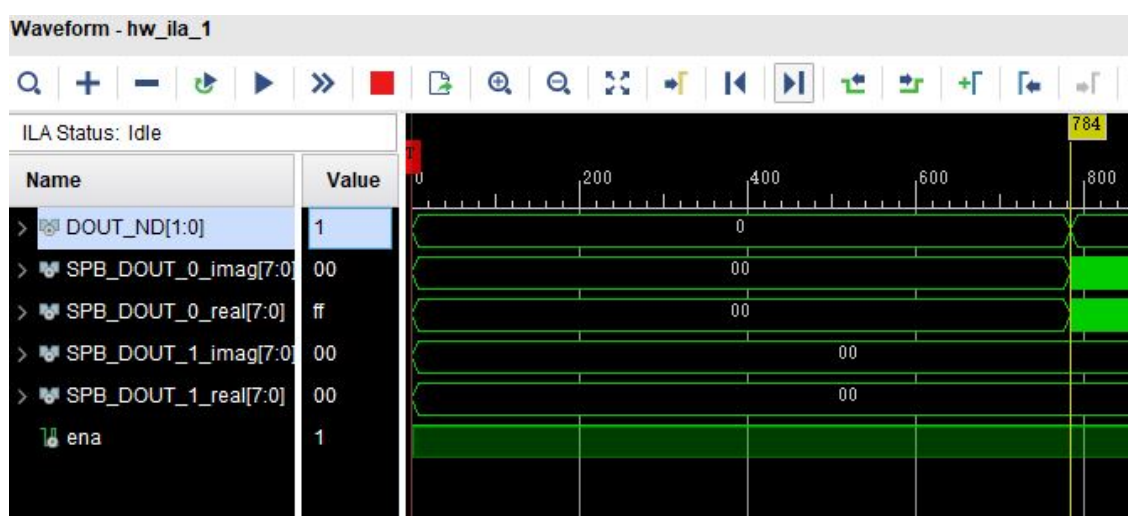


图 5-9 数字信号处理 500MHz 性能检测

Figure 5-9 Digital signal processing 500MHz performance test

(3) 基带传输：千兆以太网传输接口、QSFP+光传输接口和 PCIe 传输接口分别具备 1000Mbps、6.25Gbps 和 32Gbps 的传输能力。

表 5-6 设计目标与实现结果对照

Table 5-6 Design goals and implementation results comparison

功能	性能	设计目标	实现结果
信号接收 (ADC)	通道数	$\geq 8$ , 可扩展	8, 可扩展
	信号频率范围	550~950MHz	$\geq 550\sim 950\text{MHz}$
	转换率	$\geq 1000\text{Msps}$	1000Msps
	有效量化位数	$\geq 7\text{bits}$	7.3~7.5bits
信号处理	处理能力	并行 500MHz	支持
	可编程能力	可编程	支持
基带传输	以太网传输速率	1000Mbps	1000Mbps
	QSFP 光收发器传输速率	6.25Gbps	6.25Gbps
	PCIe 接口传输速率	32Gbps	32Gbps

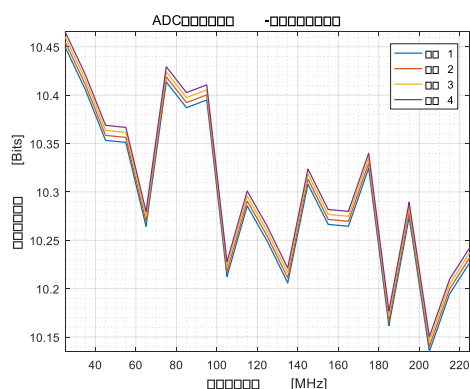


图 5-10 ADC 性能测试结果

Figure 5-10 Performance test results of the ADC

## 5.5 本章小结

本章首先分析了 5G 智能通信系统对数字信号处理硬件平台的功能和性能需求，然后根据这些需求制定了相应的设计目标，最后根据这些设计目标给出了系统的总体设计和对应的模块化单板硬件设计，最后对单板硬件的实物做了性能分析。本章的内容包括：

(1) 通过对 5G 智能通信系统的分析，得出了该系统的硬件平台需要具有信号接收、处理、传输和可扩展的功能，同时该硬件平台应能够处理多通道、大数据、大带宽、实时的高速数字信号。

(2) 首先给出了系统的设计方案，然后给出了模块化的单板硬件设计方案，每个单板硬件具有同一化的特点，并且给出了单板硬件的功能和接口。

(3) 给出了同一化的模块单板硬件的设计。该单板硬件配备有 8 个 1000Msps 采样率的 ADC 通道，高速数字信号处理的核心器件是具备丰富的逻辑和算术运算资源和强大可编程性的 FPGA，并依靠 6.25Gbps 的 QSFP+光收发器、千兆以太网口以及 32Gbps 的 PCIe 作为高速数据的基带传输接口。

(4) 通过对单板硬件的系统分析得出，本章设计实现针对 5G 智能通信系统的数字信号处理硬件平台具有 1000Mbps 的数据接收能力、500MHz 的高速数字信号实时处理能力以及高达 32Gbps 的高速数据基带传输能力，并且可以通过对单板硬件的并联或者级联进行系统扩展。

综上所述，本章设计并实现的数字信号处理硬件平台很好地满足了 5G 智能通信系统的需求。



## 第六章 总结与展望

### 6.1 本文总结

针对 5G 智能通信系统中多波束合成复杂度高系统计算延迟高、智能系统中神经网络边缘计算功耗高延时高以及需要合适的硬件平台来支持数字信号处理算法实现的问题, 本文提出了一种适用于 5G 智能通信系统宽带大数据吞吐的多波束数字波束合成方案, 同时还提出了一种适用于智能物联网系统的基于 FPGA 的神经网络边缘计算硬件加速方案, 最后设计并实现了一套适用于 5G 智能通信系统的数字信号处理硬件平台。本文提出的技术方案不仅实现了大带宽大数据吞吐的低延迟、低计算复杂度的多波束数字波束合成与低延迟、低功耗的神经网络边缘计算硬件加速, 而且具有低成本、灵活度强的特点, 工程可实现性强。

本文的主要研究成果包括以下几点:

(1) 设计了一种适用于大带宽大数据吞吐的兼顾低延迟和低复杂度的多波束数字波束合成技术方案。该技术方案基于数字正交化和数字信道化两个处理过程, 降低了宽带信号的数据处理速度, 同时, 利用了“先抽取后滤波”的工程实践思想, 降低了算法的计算复杂度和系统延迟; 合成多波束在空间的覆盖方向上的覆盖范围大于  $120^\circ$ , 波束旁瓣抑制达到 20dB。与分数延迟滤波器算法相比, 本文所设计的多波束合成技术在性能指标相当的条件下, 在复杂度和系统延迟上达到了 50% 以上的优化, 且随着合成波束数目的增加和通道数的增加, 这种优势更加明显。

(2) 实现了一种低延迟、低功耗的智能物联网神经网络边缘计算硬件加速方案。该方案充分利用了 FPGA 的多通道并行运算和流水化结构, 对传统神经网络的推断运算过程做了改进, 大大降低了系统时延, 同时在分别在卷积层、池化层和全连接层的硬件实现做了优化, 并且采用了可靠的定点量化方案在减少计算量的同时保证了推断精度和推断准确率。该方案与 CPU 硬件平台相比具有明显的速度优势, 和 GPU 硬件平台相比速度略低, 但大大降低了功耗, 适用于边缘计算。此外该系统具有灵活性和便携性, 易于工程实现。

(3) 设计了一套适用于 5G 智能通信系统的多通道宽带数字信号处理硬件平台。该硬件平台的设计采用了模块化设计, 分别实现了系统设计和单板硬件设计。该平台支持多通道扩展以及数字信号处理和基带传输的扩展。单板集成了 8 个

1Gsps 采样率的 ADC 通道；以具有强大运算能力、丰富逻辑资源和灵活可编程的 FPGA 作为数字信号处理的运算核心；在基带数据传输方面，配备了 6.25Gbps QSFP+光收发器、千兆以太网和 PCIe 等高速数据传输接口。该硬件平台具备多路 500MHz 宽带信号的接收和实时处理能力以及高达 32Gbps 的基带传输能力，满足了 5G 智能通信系统中对大宽带、大数据的接收、处理、传输和扩展的需求。

## 6.2 未来工作展望

本文提出的低复杂度的多波束合成技术方案和基于 FPGA 的神经网络边缘计算硬件加速方案以及设计并实现的数字信号处理硬件平台很好地解决了 5G 智能通信系统中的多波束合成、边缘计算和数字信号处理算法实现的问题，能够有效地提升无线通信系统和智能物联网的性能，下面是对未来进一步研究工作的展望：

（1）考虑数字波束合成和模拟波束合成相结合，进一步在降低系统计算复杂度和系统时延的同时，增加系统的实时性和工程上的可实施性。

（2）本文提出的基于 FPGA 的神经网络边缘计算硬件加速方案虽然能够满足边缘计算低功耗和低延迟的要求，但是开发周期相比 GPU 长；同时，基于浮点数和 GPU 的神经网络结构复杂，不利于 FPGA 定点实现。因此可以进一步研究 FPGA 和 CPU、GPU 的协同平台对边缘计算的加速，同时对网络做压缩和优化。

（3）未来的通信设备将继续朝着低功耗、小型化的趋势不断发展，后续可以开展基于高集成度 SoC 的数字信号处理硬件平台的设计与实现的研究，在保证功能和性能的同时优化硬件的功耗和体积。

## 参 考 文 献

- [1] Lee J , Tejedor E , Ranta-Aho K , et al. Spectrum for 5G: Global Status, Challenges, and Enabling Technologies[J]. IEEE Communications Magazine, 2018, 56(3):12-18.
- [2] 王胡成, 徐晖, 程志密, et al. 5G 网络技术研究现状和发展趋势[J]. 电信科学, 2015, 31(9):149-155.
- [3] Yong F . Neural network and artificial intelligent[C]// International Conference on Advanced Computer Theory & Engineering. IEEE, 2010.
- [4] O"Leary, Daniel E . Artificial Intelligence and Big Data[J]. IEEE Intelligent Systems, 2013, 28(2):96-99.
- [5] Li R , Zhao Z , Zhou X , et al. Intelligent 5G: When Cellular Networks Meet Artificial Intelligence[J]. IEEE Wireless Communications, 2017:2-10.
- [6] Molisch A F , Ratnam V V , Han S , et al. Hybrid Beamforming for Massive MIMO: A Survey[J]. IEEE Communications Magazine, 2017, 55(9):134-141.
- [7] Large-Scale Antenna Systems with Hybrid Analog and Digital Beamforming for Millimeter Wave 5G
- [8] Sohrabi F , Yu W . Hybrid Digital and Analog Beamforming Design for Large-Scale Antenna Arrays[J]. IEEE Journal of Selected Topics in Signal Processing, 2016:1-1.
- [9] Hu Y C, Patel M, Sabella D, et al. Mobile edge computing—A key technology towards 5G[J]. ETSI white paper, 2015, 11(11): 1-16.
- [10] Shi W, Dustdar S. The promise of edge computing[J]. Computer, 2016, 49(5): 78-81.
- [11] Tran T X, Hajisami A, Pandey P, et al. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges[J]. IEEE Communications Magazine, 2017, 55(4): 54-61.
- [12] Perahia E, Stacey R. Next generation wireless LANs: 802.11 n and 802.11 ac[M]. Cambridge university press, 2013.
- [13] Dahlman E, Parkvall S, Skold J. 4G: LTE/LTE-advanced for mobile broadband[M]. Academic press, 2013.
- [14] 陈光辉. 分数延迟 FIR 数字滤波器设计及其 FPGA 实现[D]. 湘潭: 湘潭大学, 2010.

- [15] 黄翔东, 徐婧文, 张博, 等. 低复杂度的可变分数时延滤波器设计[J]. 电子与信息学报, 2018, 40(4): 985-991.
- [16] Skouby K E, Lynggaard P. Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services[C]//2014 International Conference on Contemporary Computing and Informatics (IC3I). IEEE, 2014: 874-878.
- [17] 孙大为, 张广艳, 郑纬民. 大数据流式计算:关键技术及系统实例[J]. 软件学报, 2014(04):153-176.
- [18] Reznik A, Arora R, Cannon M, et al. Developing software for multi-access edge computing[J]. ETSI, Sophia Antipolis, France, White Paper, 2017, 20.
- [19] Klas, Guenter. Edge Computing and the Role of Cellular Networks[J]. Computer, 2017, 50(10):40-49.
- [20] Cass S. Taking AI to the edge: Google's TPU now comes in a maker-friendly package[J]. IEEE Spectrum, 2019, 56(5): 16-17.
- [21] Zhang C, Li P, Sun G, et al. Optimizing fpga-based accelerator design for deep convolutional neural networks[C]//Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2015: 161-170.
- [22] 杨森. 基于 FPGA 的人脸检测识别加速平台[D]. 吉林大学, 2018.
- [23] 周乐柱, 李斗, 郭文嘉. 卫星通信多波束天线综述[D]. , 2001.
- [24] 朱子平, 吕继荣, 洪一. 数字波束形成在雷达中的应用与分析[J]. 中国电子科学研究院学报, 2006, 1(3).
- [25] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012.
- [26] 英特尔软件学院教材编写组. 处理器架构[M]. 上海交通大学出版社, 2011.
- [27] 陈钢. 众核 GPU 体系结构相关技术研究[D]. 复旦大学, 2011.
- [28] 邹德财, 吴海涛, 李云. XILINX 的 FPGA 芯片架构剖析[J]. 航空计算技术, 2007(02):84-87.
- [29] Amara A , Frédéric Amiel, Ea T . FPGA vs. ASIC for low power applications[J]. Microelectronics Journal, 2006, 37(8):669-677.
- [30] 张雪松, 郭陈江, 侯榜焕. 一种改进的 Dolph-Chebyshev 方法在阵列天线方向图综合中的应用[J]. 航空计算技术, 2009(02):50-52+55.
- [31] Teledyne e2v. EV8AQ160datasheet. Rev. B. 2019.
- [32] Xilinx. DS180. 7 Series FPGAs Data Sheet: Overview. V2.6. 2018.
- [33] Marvell Semiconductor. MV-S105540-00. 88E1111 product brief. Rev. A. 2013.

- [34] WTD. 40G QSFP+ optical transceiver with DDM function. 2017.
- [35] Xilinx. UG476. 7 Series FPGAs GTX/GTH Transceivers. V1.12.1. 2018
- [36] Xilinx. UG471. 7 Series FPGAs SelectIO Resources. V1.10. 2018.
- [37] VME Bus Technology. VME, VME64 and VME64x General and technical information. 2005
- [38] 王学宝. 基于 VPX 标准的 PMC/XMC 载板设计[J]. 计算机测量与控制, 2010(08):148-151.
- [39] PCI Express Base Specification Revision 3.0. 2010
- [40] PICMG. PICMG 2.0. CompactPCI base specification. Rev. 3.0. 1999.

## 致 谢

首先,我要感谢我的导师宫新保老师。在攻读硕士学位的两年半时间里,宫老师给我提供了一个良好的科研环境,为我创造了许多机会,激励我不断地提升自己的能力。宫老师除了在科研工作上给予了我很多启发和指导外,在为人处世方面也教给了我很多道理,他深邃的思想、开阔的眼界、严谨的态度和踏实的作风对我产生了极大的影响,将使我终身受益。

我要特别感谢实验室的宫新保老师,宫老师在课题的研究中给予了我许多重要的指导和帮助,在论文的撰写上也给我提出了不少关键的意见和建议。

此外,我还要感谢这几年来在实验室与我一起学习成长的师兄弟黄森洪、杨高雄和宋宜轩等人。在与他们的合作与交流中我获得了许多灵感,拓宽了视野,增长了知识。

最后,衷心感谢在百忙之中抽出宝贵时间来参与论文评审和答辩的各位尊敬的专家和老师。

## 攻读硕士学位期间已发表或录用的论文

- [1] 第一作者. 低延迟、大吞吐率的 5G DBF 技术研究与实现 [J].电视技术, 2020