摘 要

本论文探讨了 MIMO 检测的 FPGA 实现，以及围绕这个主题所进行的算法研究。

MIMO 技术被认为是新一代移动通信(beyond 3G/4G)的核心技术之一，它利用多个天线进行收发，能在不占用过多带宽的情况下提高系统的容量。MIMO 检测技术是 MIMO 技术体系中的一个热点。现有的检测算法很多，有的性能优良但复杂度太高，如 ML 检测算法，有的复杂度低但性能不好，如 ZF 检测算法。基于循环结构的 MMSE SIC 检测算法是一种两者兼顾的算法，但实现起来还是存在诸多困难。本文的重点就是讨论如何将这种检测算法在性能损失不大的情况下进行简化，使之适合硬件设计，并且将简化后的算法在 Xilinx 公司的 Virtex-II pro XC2VP70 芯片中实现。

第一章回顾了现代移动通信的发展，对下一代移动通信系统及关键技术进行了展望，介绍了课题来源与论文的安排。

第二章简要介绍了 MIMO 技术框架以及该技术在 B3G 下行链路中的应用，接下来重点介绍了 MMSE SIC 检测算法，通过复杂度分析来说明硬件实现的困难所在。然后提出自己的简化方案，即通过简要的推导证明在无循环的情况下 MMSE SIC 检测算法完全等同于 MMSE 检测算法，结合运算的特点提出了简化的对数似然比解调方案，从而省去了大量的非线性运算，简化后的方案在文中被称为 MMSE-LLR 检测方案。

在第三章中研究了矩阵求逆运算。由于基于 MMSE 的检测算法中会不可避免地遇到矩阵求逆运算，经过分析发现此类矩阵均为正定 Hermite 矩阵，根据此类矩阵的特点引入了变量循环重新编号法求逆算法，并且根据运算过程中数据的对称性提出了进一步的简化方案。

第四章首先介绍 FPGA 的基本原理与设计方法，然后详细描述了设计中将要采用的 Virtex-II pro 系列 FPGA 芯片的结构、特点和资源等。

第五章为 MIMO 检测的 FPGA 实现部分。在本章中先介绍了系统需求、基本框架，然后介绍了在系统中采用的器件分时复用的思想，接着对重点模块同时也

是难实现的模块的实现方案进行了详细的介绍，最后对硬件的性能进行了分析，以此来证明前面做的简化与设计都是行之有效的。

最后一章进行了概括性的总结与展望。

关键词：B3G，MIMO，MMSE，正定 Hermite 阵，FPGA

Abstract

In this thesis we have discussed the FPGA implementation of MIMO detection and research on related algorithms.

The MIMO technology is considered as the key technology in the next generation of wireless communications (beyond 3G/4G), which deploys multi antennas at both transmitter and receiver to enhance system capacity without using additional bandwidth. MIMO detection techniques are highlighted in MIMO researches. There are plenty of detection algorithms at present, some of which have good performance but also have high complexity, while some have low complexity but the performance are very good. The MMSE SIC algorithm which based on the iterative structure has a tradeoff between performance and complexity, while the implementation is very difficult. The main point of this thesis is to research how to reduce the complexity while not to lose much of the performance, which then is more suitable for hardware design when using Xilinx company's Virtex-II pro XC2VP70 chips.

In chapter one, we review the development of modern wireless communications and envisions the systems and key technologies of wireless communications of next generation in brief. The source and the arrangement of this thesis are provided.

In chapter two, we introduce the MIMO technology and its applications in the downlink of B3G system. We then discuss in detail the MMSE SIC detection algorithm. We show the difficulty in hardware implementation by complexity analysis, then based on the analysis we provide our scheme, which has low complexity and has been proved to be equivalent to the MMSE algorithm without iterations. We provide a scheme called the MMSE-LLR (Logarithm Likelihood Ratio) which is the simplified LLR demodulation scheme and omitted a lot of non-linear operations.

In MMSE based detection algorithms it's inevitable to do the matrix inversion operation, and we have found by analysis that such matrices are all positive definition Hermite matrices. So, in chapter three, by renumbers the looped variables we induce a new matrix inversion algorithm and based on the symmetry of the data we further

simplified the algorithm.

In chapter four, we introduced the principles and design methods of FPGA. Then we discuss in detail the structure, characteristics and resources of Virtex-II pro series FPGA chips which will be used in the implementations.

In chapter five, we provide the FPGA implementation of MIMO detections. We first introduce the system requirements and the basic framework and the principles of time division multiplexing. Then introduce in detail the implementation schemes of key modules, which are also the modules that are hard to be implemented. Finally, the analyses of hardware performances are provided to prove that the above simplifications and designs are really effective.

In the last chapter we conclude the whole thesis and some envisions are also provided.

Keywords: B3G, MIMO, MMSE, positive definition Hermite matrix, FPGA

图目录

表目录

缩略字表

3G	Third Generation	第三代移动通信系统
AMPS	Advanced Mobile Phone System	高级移动电话系统
ASIC	Application Specific Integrated Circuit	专用集成电路
B3G	beyond 3G	超三代(移动通信系统)
BER	Bit Error Rate	比特误码率
BLR	Bit Likelihood Ratio	比特似然比
BPSK	Binary Phase Shift Keying	二进制相移键控
CAI	Co-antenna Interference	天线共扰
CDMA	Code Division Multiple Access	码分多址
CLB	Configurable Logic Blocks	可配置逻辑模块
CP	Cyclic Prefix	循环前缀
CPLD	Complex Programmable Logic Device	复杂可编程逻辑器件
D-BLAST	Diagonal Bell Labs Layered Space-Time	对角贝尔实验室分层空时
DSP	Digital Signal Processor	数字处理器
EEPROM	Electrically Erasable and Programmable Read Only Memory	电可擦除只读存储器
EPROM	Erasable Programmable Read－Only Memory	紫外线可擦除只读存储器
FPGA	Field Programmable Gate Array	现场可编程门阵列
FPLMTS	Future Public Land Mobile Telephone System	未来地面移动电话系统
FuTURE	Future Technology for Universal Radio Environment	通用无线电环境新技术
GSM	Global System for Mobile Communications	全球移动通信系统
HDL	Hardware Description Language	硬件描述语言
ICI	Inter－Carrier Interference	载波间干扰
IMT2000	International Mobile Telecommunications 2000 (the 3G ITU standard)	国际移动通信 2000（ITU 的 3G 标准）

IP	Intellectual Property	知识产权
ISI	Inter Symbol Interference	符号间干扰
ITU	International Telecommunication Union	国际电信联盟
LDPC	Low-Density Parity-Check code	低密度极性校验码
LLR	Log Likelihood Ratio	对数似然比
MAC	Multiply Accumulator	乘加器
MIMO	Multiple-input Multiple-output	多入多出
ML	Maximum Likelihood	最大似然
MMSE	Minimum Mean Square Error	最小均方误差
OFDM	Orthogonal Frequency Division Multiplex	正交频分复用
PAL	Programmable Array Logic	可编程阵列逻辑
PROM	Programmable Read－Only Memory	可编程只读存储器
QAM	Quadrature Amplitude Modulation	正交幅度调制
QoS	Quality of Service	服务质量
SIC	Soft Interference Cancellation	软干扰抵消
SNR	Signal to Noise Ratio	信号噪声比
SOC	System On Chip	片上系统
SOPC	System On Programmable Chip	可编程片上系统
SRAM	Static Random Access Memory	静态随机存储器
TACS	Total access Communication system	全接入通信系统
TDD	Time Division Duplex	时分双工
TDMA	Time Division Multiplex Access	时分多址
V-BLAST	Vertical Bell Labs Layered Space-Time	垂直贝尔实验室分层空时
WARC	World Administrative Radio Conference	世界无线电管理委员会
ZF	Zero Forcing	迫零

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名：王录光　　日期：　　年　月　日

关于论文使用授权的说明

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

签名：王录光　　导师签名：

日期：　　年　月　日

第一章 引言

1.1 移动通信的发展概况

人类采用无线通信的历史可以追溯到遥远的古代。但直到十九世纪末，人们都是采用十分直观的方式实现简单的信息传输。古代战争中的烽火台、金鼓和旌旗都是直观无线通信的例子。1864 年，英国物理学家 J.C.Maxwell 创造性地总结人们已有的电磁学知识，预言了电磁波的存在。1897 年，M.G.马可尼完成的无线通信试验就是在固定站与一艘轮船之间进行的，当时的距离为 18 海里。在后来一个世纪多时间里，在飞速发展的计算机和半导体技术的推动下，无线移动通信的理论和技术不断取得进步。今天，无线移动通信已经发展到大规模商用并逐渐成为人们日常生活不可缺少的通信方式之一[1]。

第一代蜂窝移动通信系统出现于 20 世纪 80 年代早期，采用频分多址和模拟技术，包括模拟蜂窝和无绳电话系统。典型的系统有美国的 AMPS、英国的 TACS、前西德的 C-450 等。模拟系统的缺点主要有频谱利用率低、抗干扰能力差、系统保密性差等，但由于模拟技术十分成熟，因而在发展初期也得到了较为广泛的应用。模拟蜂窝技术由于不适合未来多媒体通信业务的需求，必将在日益激烈的市场竞争中被逐步淘汰。

从 20 世纪 80 年代中期开始，数字移动通信系统进入发展和成熟时期。蜂窝模拟网的容量已不能满足日益增长的移动用户的需求。80 年代中期，欧洲首先推出了全球移动通信系统(GSM)。随后美国和日本也相继指定了各自的数字移动通信体制。20 世纪 90 年代初，美国 Qualcomm 公司推出了窄带码分多址(CDMA)蜂窝移动通信系统，这是移动通信系统中具有重要意义的事件。从此，码分多址这种新的无线接入技术在移动通信领域占有了越来重要的地拉。除此之外，还有欧洲的 DCS1900，美国的 IS-54 等。这些目前正在广泛使用的数字移动通信系统是第二代移动通信系统。

早在部署第二代数字无线通信系统时，人们就已经展开了研制第三代无线通信系统(3G)的工作，要求在这些系统中添加宽带数据业务，以支持视频、互联网接入以及其他更高速率的业务。

1992 年，国际电信联盟(ITU)和世界无线电管理委员会(WARC)提出计划，准备向所有国家开放 2000MHz 频段，用于实施全球无线通信系统。这一计划开始称作未来地面移动电话系统(FPLMTS)，在 1995 年更名为国际移动电信标准 2000 (IMT-2000)。IMT-2000 的初衷是期望能够在世界范围内定义单一的、遍布各地的无线通信标准，提供通用的基本网络设备和手持终端，从而为全球所有用户提供通用的无线通信系统和设备。然而在 20 世纪 90 年代中期，由于多种数字无线技术都已经在商业领域内获得了巨大的成功，因此制定全球通用标准的想法也就变得不可行了。欧洲的 TDMA 和 GSM 技术以及北美的 IS-95CDMA 技术至今也没有达成共识，形成统一的第三代无线技术。这样，ITU 只好确定一系列标准，以供全球范围使用。

目前，人们已经把目光越来越多地投向三代以后(beyond 3G)的移动通信系统中，使其可以容纳庞大的用户数、改善现有通信品质，达到高速数据传输的要求。

1.2 B3G 的定义及其关键技术

下一代移动通信系统称为 B3G 或 4G。B3G 即 Beyond 3G、Beyond ITM-2000、超 3G、后 3G 等等；4G，即第四代移动通信系统。一些专家反对 4G 的叫法，他们认为这种叫法不够严谨，因为目前各国对第四代移动通信系统的理解相差甚远，另外在3G还没大规模商用就开始4G说法对运营公司不利，这种观点得到了 ITU-R 的支持；而另外一些专家则认为 B3G 的概念包括的范围太广，他们极力支持 4G 这种称法。在本文里这两个概念是相同的，不做任何区分。

与 3G 相比，B3G 移动通信系统的特点主要有：

✧ 高速率的数据传输

B3G 的数据传输速率从 3G 的 2Mb/s 发展到 100Mb/s，移动台的速率从步行到车行甚至更快都能实现高质量的通信。虽然传输速率提高的同时也提高了系统的带宽，但 B3G 能够提高频带的利用率。

✧ 灵活多样的通信业务

通信发展的最初是以话音业务为主，但随着社会文明的发展，各种新的通信业务的不断出现，多种多样的数据业务将取代话音业务的主导地位，就对现有的

通信系统提出了新的要求。B3G 满足人们各种通信要求，在各种通信媒体之间建立无缝漫游，实现完全综合的通信网络。

✧ 全 IP 的网络

IP 与各种无线接入方式是兼容的，因此在设计核心网时有很大的灵活性。B3G 网络实现 IP 地址的个人化，现有的 IPv4 的地址空间是有限的，而 IPv6 将较好的解决这一问题。

为了实现高速率的数据传输与灵活多样的通信业务，B3G 移动通信系统采用了一系列新技术。B3G 移动通信系统的关键技术有：

✧ MIMO 技术

MIMO(多输入多输出)技术是指利用多发射、多接收天线进行空间分集的技术，能够有效的将通信链路分解成为许多并行的子信道，从而大大提高容量。信息论已经证明，当不同的接收天线和不同的发射天线之间互不相关时，MIMO 系统能够很好地提高系统的抗衰落和噪声性能，从而获得巨大的容量。在功率带宽受限的无线信道中，MIMO 技术是实现高数据速率、提高系统容量、提高传输质量的空间分集技术。

✧ OFDM 技术

OFDM(正交频分复用)是一种无线环境下的高速传输技术，其主要思想就是在频域内将给定信道分成许多正交子信道，在每个子信道上使用一个子载波进行调制，各子载波并行传输。尽管总的信道是非平坦的，即具有频率选择性，但是每个子信道是相对平坦的，在每个子信道上进行的是窄带传输，信号带宽小于信道的相应带宽。OFDM 技术的优点是可以消除或减小信号波形间的干扰，提高了频谱利用率。

✧ 调制与编码技术

B3G 移动通信系统采用新的调制技术，如多载波正交频分复用调制技术以及单载波自适应均衡技术等调制方式，以保证频谱利用率和延长用户终端电池的寿命。B3G 移动通信系统采用更高级的信道编码方案(如 Turbo 码、级连码和 LDPC 等)、自动重发请求(ARQ)技术和分集接收技术等，从而在低 Eb/N0 条件下保证系统足够的性能。

✧ 软件无线电技术

软件无线电是将标准化、模块化的硬件功能单元经过一个通用硬件平台，利用软件加载方式来实现各种类型的无线电通信系统的一种具有开放式结构的新技术。软件无线电的核心思想是在尽可能靠近天线的地方使用宽带 A/D 和 D/A 变换器，并尽可能多地用软件来定义无线功能，各种功能和信号处理都尽可能用软件实现。其软件系统包括各类无线信令规则与处理软件、信号流变换软件、信源编码软件、信道纠错编码软件、调制解调算法软件等。软件无线电使得系统具有灵活性和适应性，能够适应不同的网络和空中接口。软件无线电技术能支持采用不同空中接口的多模式手机和基站，能实现各种应用的可变 QoS。

✧ 基于 IP 的核心网

B3G 移动通信系统的核心网是一个基于全 IP 的网络，同已有的移动网络相比具有根本性的优点，即：可以实现不同网络间的无缝互联。核心网独立于各种具体的无线接入方案，能提供端到端的 IP 业务，能同已有的核心网和 PSTN 兼容。核心网具有开放的结构，能允许各种空中接口接入核心网；同时核心网能把业务、控制和传输等分开。采用 IP 后，所采用的无线接入方式和协议与核心网络(CN)协议、链路层是分离独立的。IP 与多种无线接入协议相兼容，因此在设计核心网络时具有很大的灵活性，不需要考虑无线接入究竟采用何种方式和协议。

1.3 本论文的研究内容和主要贡献

1.3.1 课题来源和研究任务

根据“十五”“863”通信技术主题战略发展报告，我国移动通信研究开发的主要目标是面向未来 10 年无线通信领域的发展趋势与需求,研究 Beyond3G/4G 新一代蜂窝通信空中接口技术，建立相关关键技术验证系统，支持面向未来的无线通信新业务，对形成新一代无线与移动通信知识产权和体制标准做出较大贡献，为国家培养一支规模性的具有国际竞争力的超前研究队伍，形成我国无线通信方面人才队伍的合理布局，使我国移动通信技术研究与国际先进水平同步发展，为实现我国未来无线通信产业的跨越式发展创造条件[4]。

围绕上述目标，启动一个名为 FuTURE 的未来移动通信研究计划。本课题就

✧ 软件无线电技术

软件无线电是将标准化、模块化的硬件功能单元经过一个通用硬件平台，利用软件加载方式来实现各种类型的无线电通信系统的一种具有开放式结构的新技术。软件无线电的核心思想是在尽可能靠近天线的地方使用宽带 A/D 和 D/A 变换器，并尽可能多地用软件来定义无线功能，各种功能和信号处理都尽可能用软件实现。其软件系统包括各类无线信令规则与处理软件、信号流变换软件、信源编码软件、信道纠错编码软件、调制解调算法软件等。软件无线电使得系统具有灵活性和适应性，能够适应不同的网络和空中接口。软件无线电技术能支持采用不同空中接口的多模式手机和基站，能实现各种应用的可变 QoS。

✧ 基于 IP 的核心网

B3G 移动通信系统的核心网是一个基于全 IP 的网络，同已有的移动网络相比具有根本性的优点，即：可以实现不同网络间的无缝互联。核心网独立于各种具体的无线接入方案，能提供端到端的 IP 业务，能同已有的核心网和 PSTN 兼容。核心网具有开放的结构，能允许各种空中接口接入核心网；同时核心网能把业务、控制和传输等分开。采用 IP 后，所采用的无线接入方式和协议与核心网络(CN)协议、链路层是分离独立的。IP 与多种无线接入协议相兼容，因此在设计核心网络时具有很大的灵活性，不需要考虑无线接入究竟采用何种方式和协议。

1.3 本论文的研究内容和主要贡献

1.3.1 课题来源和研究任务

根据“十五”“863”通信技术主题战略发展报告，我国移动通信研究开发的主要目标是面向未来 10 年无线通信领域的发展趋势与需求,研究 Beyond3G/4G 新一代蜂窝通信空中接口技术，建立相关关键技术验证系统，支持面向未来的无线通信新业务，对形成新一代无线与移动通信知识产权和体制标准做出较大贡献，为国家培养一支规模性的具有国际竞争力的超前研究队伍，形成我国无线通信方面人才队伍的合理布局，使我国移动通信技术研究与国际先进水平同步发展，为实现我国未来无线通信产业的跨越式发展创造条件[4]。

围绕上述目标，启动一个名为 FuTURE 的未来移动通信研究计划。本课题就

围绕上述目标，启动一个名为 FuTURE 的未来移动通信研究计划。本课题就

✧ 软件无线电技术

软件无线电是将标准化、模块化的硬件功能单元经过一个通用硬件平台，利用软件加载方式来实现各种类型的无线电通信系统的一种具有开放式结构的新技术。软件无线电的核心思想是在尽可能靠近天线的地方使用宽带 A/D 和 D/A 变换器，并尽可能多地用软件来定义无线功能，各种功能和信号处理都尽可能用软件实现。其软件系统包括各类无线信令规则与处理软件、信号流变换软件、信源编码软件、信道纠错编码软件、调制解调算法软件等。软件无线电使得系统具有灵活性和适应性，能够适应不同的网络和空中接口。软件无线电技术能支持采用不同空中接口的多模式手机和基站，能实现各种应用的可变 QoS。

✧ 基于 IP 的核心网

B3G 移动通信系统的核心网是一个基于全 IP 的网络，同已有的移动网络相比具有根本性的优点，即：可以实现不同网络间的无缝互联。核心网独立于各种具体的无线接入方案，能提供端到端的 IP 业务，能同已有的核心网和 PSTN 兼容。核心网具有开放的结构，能允许各种空中接口接入核心网；同时核心网能把业务、控制和传输等分开。采用 IP 后，所采用的无线接入方式和协议与核心网络(CN)协议、链路层是分离独立的。IP 与多种无线接入协议相兼容，因此在设计核心网络时具有很大的灵活性，不需要考虑无线接入究竟采用何种方式和协议。

1.3 本论文的研究内容和主要贡献

1.3.1 课题来源和研究任务

根据“十五”“863”通信技术主题战略发展报告，我国移动通信研究开发的主要目标是面向未来 10 年无线通信领域的发展趋势与需求,研究 Beyond3G/4G 新一代蜂窝通信空中接口技术，建立相关关键技术验证系统，支持面向未来的无线通信新业务，对形成新一代无线与移动通信知识产权和体制标准做出较大贡献，为国家培养一支规模性的具有国际竞争力的超前研究队伍，形成我国无线通信方面人才队伍的合理布局，使我国移动通信技术研究与国际先进水平同步发展，为实现我国未来无线通信产业的跨越式发展创造条件[4]。

围绕上述目标，启动一个名为 FuTURE 的未来移动通信研究计划。本课题就

是来源于此项目的第二阶段，项目编号：2003AA12331006。在本阶段中电子科技大学负责TDD方式下行链路的FPGA设计。B3G-TDD下行链路采用MIMO+OFDM的基本框架，采用TDMA+OFDMA的多址方式。

本论文的主要任务是研究MIMO检测技术以及基于FPGA的设计与实现。

1.3.2 本论文的主要贡献

本论文在两方面做出了贡献。在算法方面，这里对算法进行的一切研究都是围绕下一步的FPGA实现展开。对硬件设计而言，一个好的算法不仅要性能优异，而且还要保证复杂度在可实现的范围之内。本文中所提到的MMSE SIC检测算法就其性能而言是一种非常好的算法，但其复杂度也非常大，如何能简化到可以用有限的FPGA资源实现出来是本文算法研究部分的重点。这部分的主要亮点有3个：

- ✧ 分析了在无循环的情况下MMSE SIC检测算法实际上等同于MMSE检测算法；
- ✧ 分析了MMSE滤波因子的特点，引入并简化了变量循环重新编号法求逆矩阵算法，使其FPGA设计的难度相对降低了很多；
- ✧ 分析了MIMO检测结果输出比特似然比(BLR)算法的硬件实现难度，对采用结果输出为对数似然比(LLR)的检测算法进行了简化，降低了实现难度。

在FPGA设计方面，在这一部分除了介绍常规的设计技巧以外也有三个亮点：

- ✧ 根据整个下行链路的系统需求巧妙地安排时钟频率与器件的处理方式，达到最大限度的资源复用；
- ✧ 在设计MMSE滤波器所涉及的矩阵求逆模块的设计上，结合在MIMO检测内部4个时钟处理一个数据的时钟要求，以及变量循环重新编号法对4阶矩阵需循环4次的特点，巧妙地采用5级时钟循环处理方式，既达到了输入数据与循环数据不发生冲突，又达到了资源最大限度的合理利用；
- ✧ 幅度自动调整算法的研究与设计，用很少的资源实现了数据幅度的自动调整，拓宽了MIMO检测所能处理的数据范围。

1.3.3 本文的结构安排

第一章引言，回顾移动通信系统的发展历史，介绍本文所研究的B3G系统的定义及技术要求，对本文研究工作的课题来源、论文内容进行了概述。

第二章MIMO技术与检测算法，首先介绍了MIMO的基本知识，接下来介绍了MIMO技术在本项目(B3G下行无线链路设计)中的应用。然后是本文的重点部分，先介绍完整的基于循环结构的MMSE SIC检测算法，然后又对其进行了大量的简化，并且对简化方案进行了性能分析，以保证简化工作行之有效。出于文章的连贯性考虑，将此部分的矩阵求逆算法研究工作放在下一章中来讲述。

第三章正定Hermite矩阵求逆算法研究，本章的中心任务是解决MMSE检测算法中最影响复杂度的一个器件(MMSE滤波器)的设计难题。在本章里首先介绍了几种最常见的求逆算法，然后介绍了一种比较优异的全选主元高斯-约当消元法，这种方案针对一般矩阵是一种好算法，但对于MMSE检测算法中出现的矩阵并不算最优，因此接下来的重点是介绍针对正定Hermite阵提出的变量循环重新编号法，以及本人对此算法做的一些简化。

第四章开发环境与工具，首先介绍了FPGA的基本原理，结构，设计方法与流程等FPGA开发的基本常识，然后对本文硬件实现选用的器件——Xilinx公司Virtex-II Pro系列芯片XC2VP70做了一个详细的介绍。

第五章MIMO检测的FPGA设计，首先介绍了系统的整体方案，整体安排，然后分模块进行介绍，最后介绍了MIMO检测部分的仿真与性能分析，通过最后的性能曲线和综合资源报告验证本文所进行的研究和设计的合理性。

第二章 MIMO 技术研究

2.1 MIMO 技术介绍

2.1.1 MIMO 基本原理

MIMO 技术本质上是空间分集与空间复用的结合，分集可以保证传输的可靠性，复用则可以提高传输速率。在传统的无线通信理论中，多径效应会引起无线信号的衰落，因而被视为一种不利因素，但如果在发送端与接收端同时采用多天线系统，只要各天线单元间距足够大，无线信道的多径分量足够丰富，那么天线接收到的多径信号的衰落就趋于独立，MIMO 技术就利用了这一特性。对于一个具有 N_T 副发射天线和 N_R 副接收天线的 MIMO 系统，其发射信号经过空时编码后形成 N_T 个信息子流，以相同的频率经不同的天线同时发射出去，经过多径信道的传播，这些并行子流以不同的路径到达接收机，并由不同的天线接收，接收机利用空时解码对各接收信号进行处理，并恢复出原始数据流。由于这些子流同时发送，占用同一频带，因而并未增加带宽。若各发射接收天线间的通道响应相互独立，则可以创造多个并行空间信道。所以说，MIMO 系统可以充分利用无线信道的多径传播，获取复用增益与分集增益，从而显著提高无线链路的容量与质量。

2.1.2 MIMO 信道模型

$N_R \times N_T$ 的 MIMO 系统信道模型如图 2-1 所示，

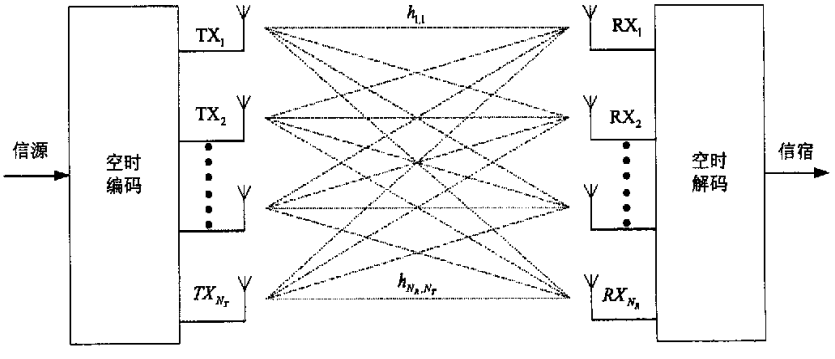

图2-1 MIMO 信道模型

N_R、N_T分别表示接收天线数和发射天线数目，其数学模型[4]为：

$$\mathbf{y}=\mathbf{Hs}+\mathbf{n} \tag{2.1}$$

式中，$\mathbf{y}=\left[y_1,y_2,...,y_{N_R}\right]^T$表示接收到的信号矢量，$\mathbf{H}$是$N_R\times N_T$信道矩阵，$\mathbf{s}=\left[s_1,s_2,...,s_{N_T}\right]^T$为发送信号矢量，$\mathbf{n}=\left[n_1,n_2,...,n_{N_R}\right]^T$代表零均值高斯白噪声矢量。

为了在后续章节中描述方便，可以将(2.1)式改写为如下的形式：

$$\mathbf{y}=\mathbf{h}_i s_i+\sum_{j=1,j\neq i}^{N_T}\mathbf{h}_j s_j+\mathbf{n} \tag{2.2}$$

其中，$\mathbf{h}_i$和$\mathbf{h}_j$分别表示信道矩阵$\mathbf{H}$的第i列和第j列，s_i和s_j分别表示第i和第j根天线上的发送符号，并且$S_i,S_j\in\mathbf{A}=\{a_1,...,a_M\}$，$\mathbf{A}$是调制星座点的集合(可以是M-PSK或M-QAM调制方式)。

2.1.3 MIMO系统的容量

系统容量是通信系统的重要指标之一，$N_R\times N_T$的MIMO系统，对于N_T发N_R收的MIMO系统，假定信道为独立的Rayleigh衰落，则系统的容量[6]可以表示为：

$$C=log_2\,det\left[\mathbf{I}_{N_R}+\frac{\rho}{N_T}\mathrm{HH}^H\right]\ (\mathrm{bps/Hz}) \tag{2.3}$$

其中，ρ是接收端平均信噪比。

当N_R、N_T很大时，则信道容量C近似为：

$$C\approx min\{N_R,N_T\}log_2(\rho/2) \tag{2.4}$$

可以看出，MIMO系统的信道容量随着天线数量的增大而线性增大。也就是说MIMO技术可以成倍地提高无线信道容量，在不增加带宽和天线发送功率的情况下，频谱利用率可以成倍地提高。

2.1.4 国内外研究状况

MIMO 技术最早是由 Marconi 于 1908 年提出，上世纪 70 年代有人提出将其用于通信系统，但奠基工作是 90 年代由 AT&T Bell 实验室的学者完成的。1995 年 I.E.Talatar 推导了衰落情况下的 MIMO 系统容量；1996 年 G.J.Foshini 给出了一种多入多出处理算法——对角贝尔实验室分层空时(D-BLAST)算法；1998 年 V.Tarokh 等讨论了用于多入多出的空时编码；1998 年 Wolniansky 等人采用垂直－贝尔实验室分层空时(V-BLAST)[6]算法建立了一个 MIMO 实验系统，在室内试验中达到了 20 bit/s/Hz 以上的频谱利用率；2002 年 10 月，朗讯公司贝尔实验室研制出世界上第一颗 BLAST 芯片，这一芯片支持最高 4×4 的天线布局，可处理的最高数据速率达到 19.2Mbps。2003 年 8 月，Airgo Networks 推出了 AGN100 Wi-Fi 芯片组，称其是世界上第一款集成了 MIMO 技术的批量上市产品。除此以外，也有很多其他的高校、企业和科研院所都正在积极进行 MIMO 技术的深入研究，进行了大量的工作，研究内容主要包括空时编码算法、信道建模、多天线设计与布局、互耦分析、信道相关性评估等。

2.2 MIMO 技术在 B3G 系统中的应用

在国家“863” FuTURE 计划第二阶段中，电子科技大学负责 TDD 方式下行链路设计[7]。根据系统需求分析，B3G-TDD 下行链路采用 MIMO+OFDM 的基本框架，采用 TDMA＋OFDMA 的多址方式。

考虑 B3G 系统的业务需求大都为宽带高速多媒体数据流，而且下行的数据量可能远大于上行业务量，因此本方案利用了 TDD 的特点设计了灵活的帧结构，上下行链路可进行灵活的时隙配置。而且终端的移动速度范围很大，从很慢的步行移动速度到高速车载环境，本方案都满足业务的 QoS 需求。

2.2.1 B3G 系统配置

TDD 模式的 Beyond3G 试验系统总体结构及组成见图 2-2，包括 3 个移动台、业务演示终端及 4 天线的天线阵/射频前端、2 个 AP(AP1、AP2)及相应的一个或两个天线阵、1 个控制域。

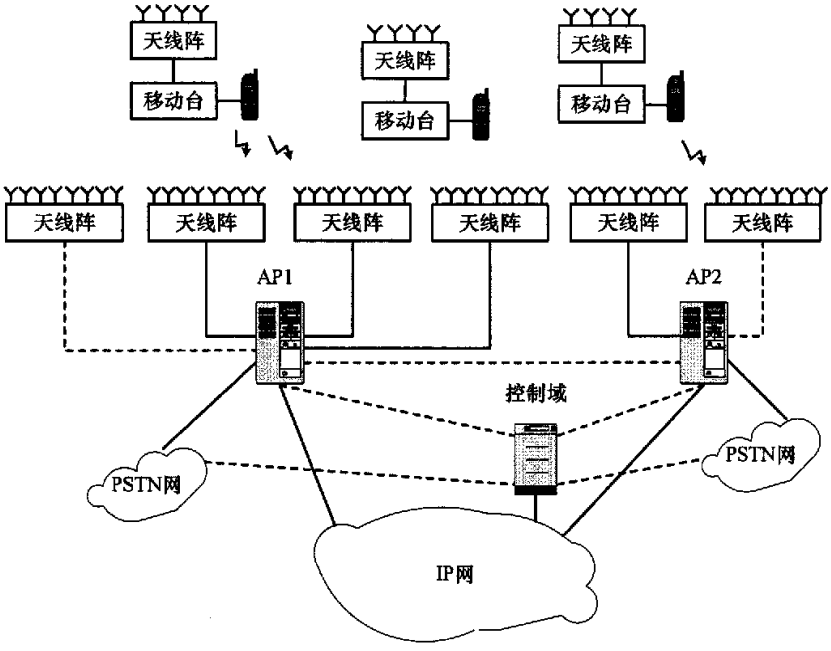

图2-2 B3G 实验系统

2.2.2 B3G 系统的基本参数

根据文献[7]，B3G-TDD 系统的基本参数如表 2-1 所示。

表2-1 B3G 系统参数

载频 fc	3.5GHz
系统带宽 B	20MHz
子载波数 N	1024
有效子载波数 Nu	884
循环扩展 CP	216(10.8us)
符号周期 TS	51.2+10.8=62.0us
调制方式	BPSK，16QAM
车速 V	5-250km/hour
信道模型	COST207
发送接收天线数	4(基站)×4(移动台)

2.2.3 B3G 系统的基本框架

这里的 B3G 系统采用 MIMO+OFDM 技术，因此其基带系统结构如图 2-3 所示。发射机发出的信源，经过信道编码交织后，串并转换，然后进行调制，空时处理，就开始多天线的 OFDM 调制，然后组帧后经过多天线发射出去。在接收机部分，经过同步处理，解帧后，进行 OFDM 解调，然后进行信道估计，对数据进行恢复，接着是 MIMO 检测处理，Turbo 接收处理，最后得到信宿。

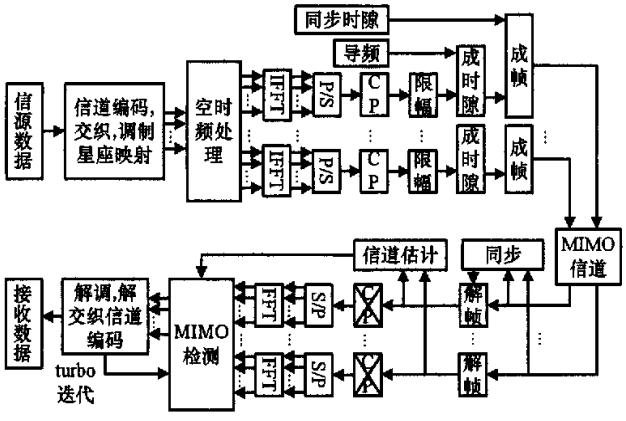

图2-3 B3G 下行链路基本框架

2.3 MIMO 检测算法研究

2.3.1 MMSE SIC 检测算法

2.3.1.1 算法分解

MIMO 检测的中心任务是做 QAM 软解调，给 LDPC 译码提供星座点的似然信息，基于 MMSE SIC 算法的 MIMO 检测过程看成有如下几个环节构成，见图 2-4

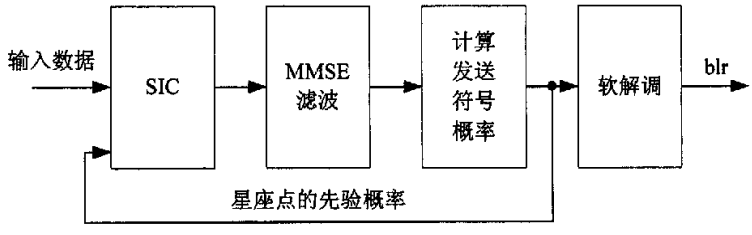

图2-4 MMSE SIC 检测算法处理框图

2.3.1.2 SIC 处理

在检测第 i 根天线上发送信号的时候，从接收信号中减去对其它天线上信号的干扰(Co-antenna Interference, CAI)的软估计值，得到如下表达式[8]：

$$\begin{aligned}\mathbf{y}_i &= \mathbf{y} - \sum_{j=1, j\neq i}^{N_T} E\{s_j\}\mathbf{h}_j \\ &= \mathbf{h}_i s_i + \sum_{j=1, j\neq i}^{N_T} \mathbf{h}_j \left(s_j - E\{s_j\}\right) + \mathbf{n}\end{aligned} \tag{2.5}$$

式中的 $E\{s_j\}$ 为发送符号的均值，有下面公式求得

$$E\left\{s_j\right\} = \sum_{a_n \in A} a_n p\left\{s_j = a_n\right\} \tag{2.6}$$

其中，$p\left\{s_j = a_n\right\}$ 是星座图上点 a_n 的先验概率，由 MIMO 内部循环提供或者由系统大循环(即迭代)提供。

2.3.1.3 MMSE 滤波

为了进一步抑止残余的 CAI，对 $\mathbf{y}_i$ 进行 MMSE 滤波，得到第 i 根发送天线上符号 s_i 的估计值：

$$\hat{s}_i = \mathbf{w}_i^H \mathbf{y}_i \tag{2.7}$$

$\mathbf{w}_i$ 为 MMSE 滤波器权值矢量。$\mathbf{w}_i$ 由下式计算得到：

$$\mathbf{w}_i = P_{S_i}\left[\mathbf{H}\mathbf{R}_i\mathbf{H}^H + \sigma_n^2 \mathbf{I}_{N_R}\right]^{-1}\mathbf{h}_i \tag{2.8}$$

其中，矩阵 $\mathbf{R}_i$ 是对角矩阵，其定义如下：

$$\mathbf{R}_i = \mathrm{diag}\left[\mathrm{var}\left\{s_1\right\}, \ldots, \mathrm{var}\left\{s_{i-1}\right\}, P_{S_i}, \mathrm{var}\left\{s_{i+1}\right\}, \ldots, \mathrm{var}\left\{s_{N_T}\right\}\right] \tag{2.9}$$

$\mathbf{I}_{N_R}$ 是 $N_R \times N_R$ 的单位阵。σ_n^2 为噪声的方差。需要注意的是，对于 OFDM 系统，高速情况下(如 250 *km/hour* 的时候)需考虑加上 ICI(Inter-Carrier Interference)的影响。根据文献[10]的结论，在仿真中，ICI 由下式给出

$$\delta_{ICI} = (1.64/N_T)\times(v/3.6\times 11.666667\times 0.000062)^2 \tag{2.10}$$

式中 v 是移动台的移动速度，单位为 $km/hour$ 。

公式(2.8)与公式(2.9)中的 P_{S_i} 、 $\mathrm{var}\{s_i\}$ 分别表示第 i 根发射天线的功率和方差，定义如下：

$$P_{S_i} = \sum_{a_n \in A} |a_n|^2 \, p\{s_i = a_n\} \tag{2.11}$$

$$\mathrm{var}\{s_i\} = \sum_{a_n \in A} |a_n|^2 \, p\{s_i = a_n\} - |E\{s_i\}|^2 \tag{2.12}$$

2.3.1.4 计算发送符号概率

经过 SIC 和 MMSE 滤波后，数据残留的干扰和噪声可以很好的进行高斯近似[9]。因此，可以将 $\hat{s}_i$ 表达为对等的高斯信道形式

$$\hat{s}_i = \mu_i s_i + \eta_i \tag{2.13}$$

其中

$$\mu_i = \mathbf{w}_i^H \mathbf{h}_i \tag{2.14}$$

而 $\eta_i \sim N_c\left(0, \sigma_{\eta_i}^2\right)$，且

$$\begin{aligned} \sigma_{\eta_i}^2 &= \mathbf{w}_i^H \left[\mathbf{H}\mathbf{R}_i\mathbf{H}^H + \sigma_n^2 \mathbf{I}_{N_R}\right]\mathbf{w}_i - \mu_i^2 P_{S_i} \\ &= \left(\mu_i - \mu_i^2\right) P_{S_i} \end{aligned} \tag{2.15}$$

于是，可以得到发送符号 s_i 对应于调制符号集 $\mathbf{A}$ 中每个待选符号的符号概率

$$p\{\hat{s}_i | s_i = a_n\} \approx \frac{1}{\pi\sigma_{\eta_i}^2} \exp\left[\frac{-\left|\mathbf{w}_i^H(\mathbf{y}_i - \mathbf{h}_i a_n)\right|^2}{\sigma_{\eta_i}^2}\right] \tag{2.16}$$

其中， $a_n \in A, n = 1, \cdots, M$ 。

2.3.1.5 软解调

为简洁起见，不妨设 $p\left\{\hat{s}_i \middle| s_i = a_n\right\}$ 为 $\eta_{i,n}$。发送符号 s_i 的第 k 个比特 c_k 对应的比特似然比由下式求出：

$$blr_{s_i}(k) = \frac{p\left\{c_k = 1 \middle| s_i\right\}}{p\left\{c_k = 0 \middle| s_i\right\}} = \frac{\sum_{n=1, c_k=1}^{M} \eta_{i,n}}{\sum_{n=1, c_k=0}^{M} \eta_{i,n}} \tag{2.17}$$

该比特似然比将作为下级 LDPC 译码器的输入。

2.3.1.6 检测流程

基于循环的 MMSE SIC 检测流程见图 2-5：

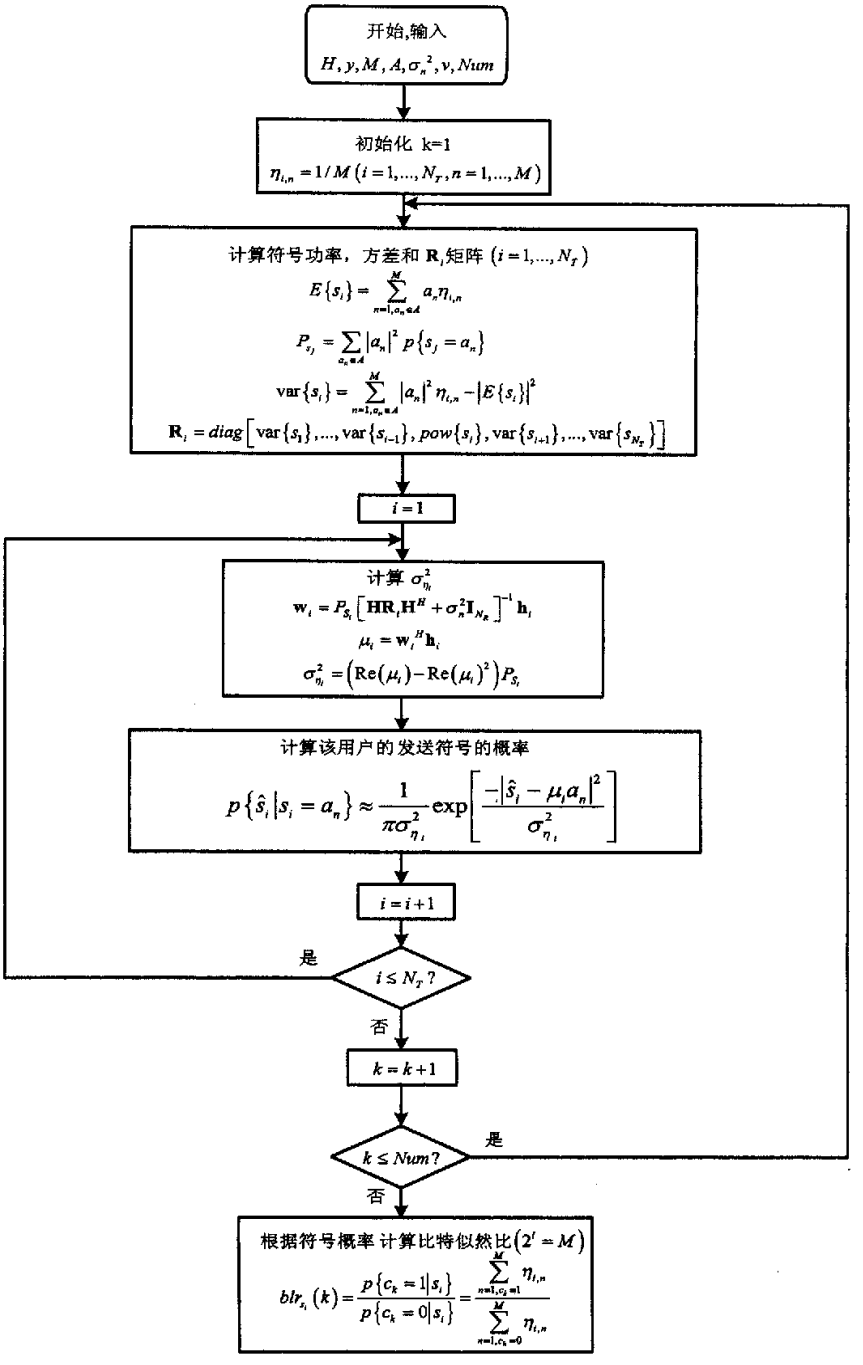

图2-5 MMSE SIC 检测算法流程图

2.3.1.7 复杂度分析

MMSE 算法的复杂度主要由计算 MMSE 系数 $\mathbf{w}_i$，均值 μ_i，和方差 $\sigma_{\eta_i}^2$ 组成，具体复杂度由表 2-2 给出，N 为发射天线数目，M 为调制阶数，定义每个发射矢量为 $s=\left[s_1,\cdots,s_N\right]^T$，检测一个发射信号矢量所需的复杂度：

表2-2 MMSE SIC 算法复杂度

	复加	复乘	实加	实乘
$\mathbf{w}_i$	$2N^2-N$	$2N^3+N^2$		
μ_i	$N-1$	N		
$\sigma_{\eta_i}^2$			1	2
$p(\hat{s}_i \mid s_i)$	$2M$	$3M$		3
求逆		N^3		
BLR			$4M-8$	4
总计	$2N^3+2NM-N$	$3N^4+N^3+N^2+3MN$	$N+4M-8$	$5N+4$

其中，求逆运算用 N^3 次复乘来衡量，前 5 项需要对发射向量的每个分量进行计算，总的复杂度需要乘以 N。

2.3.2 MMSE-LLR 检测算法

2.3.2.1 迭代的等效处理

若采用4×4的 MIMO 信道，调制方式为 16-QAM，在无循环情况下做完一次完整的 MIMO 检测，需要的乘法运算次数就达到了 4184 次，而现在一流的 FPGA 芯片上所提供的乘法器核仅几百个(参看表 4-1)，并且 MMSE SIC 检测算法中还存在大量的非线性运算(如 exp 函数)，这些运算对硬件资源的损耗也是大的惊人。因此，不对算法进行简化几乎就无法用 FPGA 芯片来实现 MIMO 检测。下面将讨论基于4×4的 MIMO 信道，调制方式为 16-QAM 的 MIMO 检测的简化算法。

在调查过各种 FPGA 芯片资源和性能情况后，容易就能得出这样的结论：在

同一块 FPGA 内部，无论如何都完成不了常规意义下的 MIMO 检测内部的循环或者系统的大循环，所有的循环只能做等效处理。现在以系统大循环(迭代)为例来解释一下等效处理的基本思路，如图 2-6 所示。其中，循环次数为 4，需要进入存储器的是信道信息 **H** 与信号 y，在做完一次完整地 MIMO 检测运算过程中，输入各个 MIMO 检测模块的信道信息 **H** 与信号 y 是不变的，唯一改变的是座图上点先验概率 $p\{s_j = a_n\}$。在 MIMO 检测 1 种星座点的先验概率 $p\{s_j = a_n\}$ 被量化为 $1/M$ (采用 M-QAM 调制方式)。现阶段开发的就是最前端的那个 MIMO 检测模块，其他的有待后续开发。

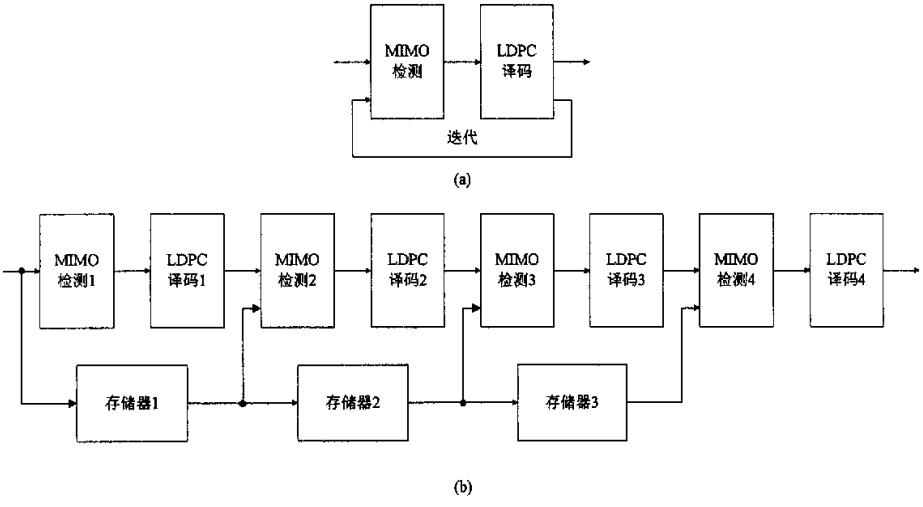

图2-6 迭代等效原理图

在星座点先验概率为 $1/M$ 且天线进行了功率归一化处理后，发送符号的均值、功率以及方差分别为：

$$E\{s_j\} = 0 \tag{2.18}$$

$$\mathrm{var}\{s_j\} = P_{s_j} = \frac{1}{4} \tag{2.19}$$

2.3.2.2 SIC 处理的简化

按照上面的描述，软干扰抵消也就变成了

$$\mathbf{y}_i = \mathbf{y} \tag{2.20}$$

于是，整个软干扰抵消被简化掉了。

2.3.2.3 MMSE 滤波的简化

由表 2-2 可以看出，整个 MMSE SIC 算法实现的难点就在 MMSE 滤波器的设计上。如果能将这一部分作大规模简化，就可以大大降低 MIMO 检测的实现难度。

采用上述约定后的对角矩阵 $\mathbf{R}_i$ 可以简化为

$$\begin{aligned}\mathbf{R}_i &= \mathrm{diag}\left[\mathrm{var}\{s_1\},...,\mathrm{var}\{s_{i-1}\},P_{S_i},\mathrm{var}\{s_{i+1}\},...,\mathrm{var}\{s_{N_T}\}\right] \\ &= \frac{1}{4}\mathbf{I}_4\end{aligned} \tag{2.21}$$

滤波因子 $\mathbf{w}_i$ 便简化为

$$\begin{aligned}\mathbf{w}_i &= P_{S_i}\left[\mathbf{H}\mathbf{R}_i\mathbf{H}^H+\sigma_n^2\mathbf{I}_{N_R}\right]^{-1}\mathbf{h}_i \\ &= \frac{1}{4}\left[\frac{1}{4}\mathbf{H}\mathbf{H}^H+\sigma_n^2\mathbf{I}_4\right]^{-1}\mathbf{h}_i \\ &= \left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]^{-1}\mathbf{h}_i\end{aligned} \tag{2.22}$$

值得注意的是，这一步的简化具有非常重要的意义。简化前，对于每一个天线都要运算一次 $\left[\mathbf{H}\mathbf{R}_i\mathbf{H}^H+\sigma_n^2\mathbf{I}_{N_R}\right]^{-1}$，而简化后的滤波因子中 $\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]^{-1}$ 对每根天线而言是固定不变的，只求一次即可。把以前 $12N^4$ 次乘法运算压缩为 $8N^3$ 次乘法运算，对于 4×4 MIMO 信道，这一步仅需做 512 次乘法。

则第 i 根发送天线上符号 s_i 的估计值为：

$$\hat{s}_i = \mathbf{w}_i^H\mathbf{y} \tag{2.23}$$

然而，仅做这些简化还是不够的，还应看出矩阵 $\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4$ 是一个正定 Hermite 矩阵(证明见附录 A)，根据此矩阵的对称性和正定性使这一步在性能毫不损失的情况下将乘法次数缩减为 260 次。正定 Hermite 矩阵求逆算法请看下章。

2.3.2.4 LLR 解调方案

求发送符号 s_i 对应于调制符号集 A 中每个待选符号概率的实现难点是：非线性运算太多。要做 64 次 exp 函数的运算，在 FPGA 器件中非线性运算一般用查表法

来实现。按数据位宽为 18bit 来算，将需要 64 个深度为 2^18 的存储器，损耗的存储资源太多。

软解调求比特似然比(BLR)的实现难点是，输出的动态范围太大，通常情况公式中的分子(比特为 1 的概率)与分母(比特为 0 的概率)一个接近 0 另一个接近 1，这样的比值常常在10^{-34}～10^{34}之间波动，这样的动态范围很难用定点数来表达。

本设计的解决方案是改输出比特似然比(BLR)为输出对数似然比[11](LLR)，发送符号s_i的第k个比特c_k对应的对数似然比由下式求出(证明见附录 B)：

$$LLR_{s_i}(k)=\frac{\min_{c_k=0}\left(\left|\hat{s}_i-\mu_i a_n\right|^2\right)-\min_{c_k=1}\left(\left|\hat{s}_i-\mu_i a_n\right|^2\right)}{\sigma_{\eta_i}^2} \tag{2.24}$$

式中，μ_i、$\sigma_{\eta_i}^2$的求法与 MMSE SIC 算法相同。

简化后的 MIMO 检测算法可以称为 MMSE-LLR 检测算法。其检测过程可以看成由如图 2-7 所示的两部分构成：

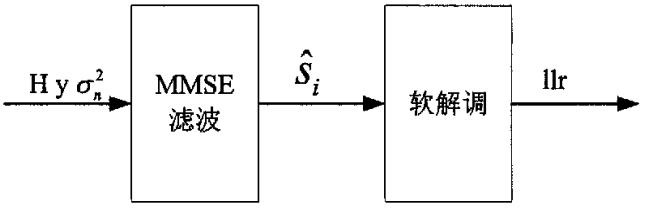

图2-7 MMSE-LLR 算法框图

2.3.2.5 MMSE-LLR 检测流程

MMSE-LLR 检测流程图如下：

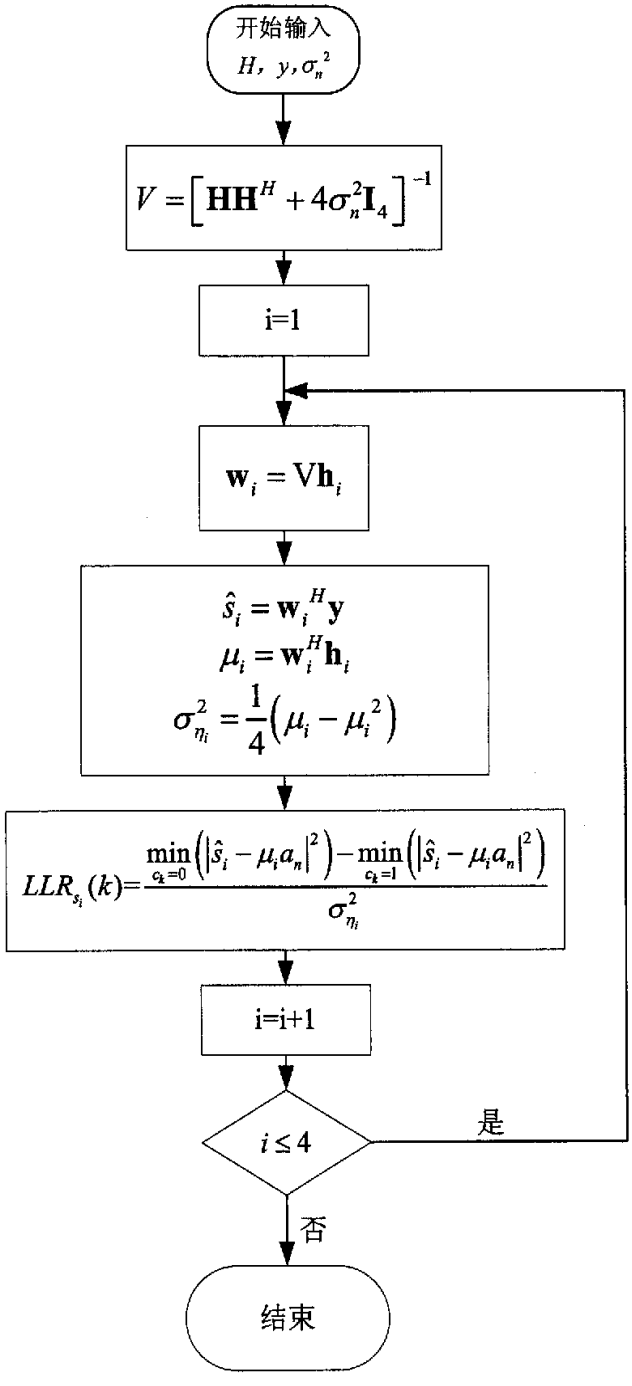

图2-8 MMSE-LLR 检测算法流程图

2.3.2.6 MMSE-LLR 检测算法复杂度

简化后的复杂度如下表：

表2-3 MMSE-LLR 算法复杂度

	复加	复乘	实加	实乘
HH^H	$\frac{N(N-1)(N+1)}{2}$	$\frac{N^2(N+1)}{2}$		
$\mathbf{HH}^H+4\sigma_n^2\mathbf{I}_4$			N	
求逆	$\frac{N^2(N-1)}{2}$	$\frac{N^2(N-1)}{2}$		$2N(N-1)$
$\mathbf{w}_i$	N^3-N^2	N^3		
μ_i	N^2-N	N^2		
$\hat{s}_i$	N^2-N	N^2		
$\sigma_{\eta_i}^2$			N	$2N$
$\lvert\hat{s}_i-\mu_i a_n\rvert^2$	NM		NM	$2NM+2N$
LLR			$N\log_2 M$	$N\log_2 M$
总计	$N^3+\frac{1}{2}N^2$ $+\frac{3}{2}N+NM$	$2N^3+2N^2$	$NM+N$ $+N\log_2 M$	N^2+2NM+ $2N+N\log_2 M$

2.3.2.7 简化前后性能对比

简化前后的性能对比情况，如下图所示：

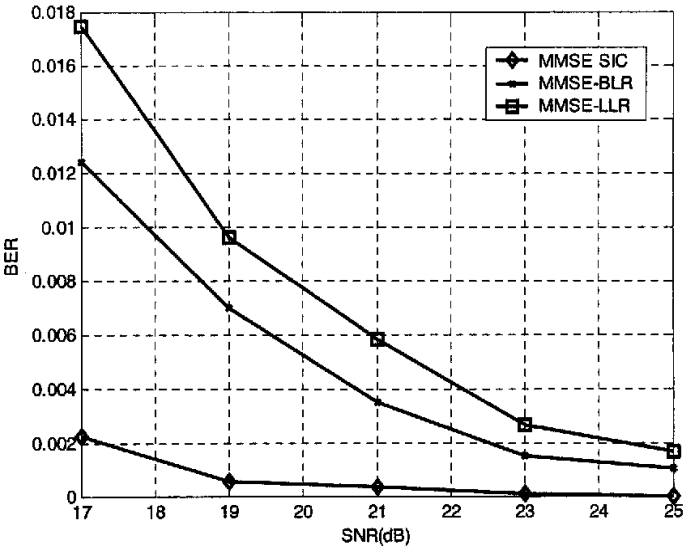

图2-9 三种检测方案的性能对比曲线

从图 2-9 可以看出循环与否对性能影响很大，但是改输出 BLR 为输出 LLR 对性能的影响并不大。因此，建议在后续的工作中加入循环，输出为 LLR。

2.4 小结

本章首先介绍了 MIMO 技术的基本原理，然后简述了 MIMO 技术在 B3G 中的应用。接下来重点介绍完整的基于循环结构的 MMSE SIC 检测算法，通过复杂度分析来说明硬件实现的困难所在。通过简化，即通过简要的推导来表明在无循环的情况下 MMSE SIC 检测算法完全等同于 MMSE 检测算法，结合运算的特点提出了简化的对数似然比解调方案，节省了巨大的资源开销。最后对简化方案进行了性能分析，以保证简化工作行之有效。

第三章 正定 Hermite 矩阵求逆算法研究

3.1 引言

由复杂度分析一节可以看出，无论是采用 MMSE SIC 检测算法，还是采用 MMSE 检测算法，矩阵求逆在整个 MIMO 检测中占的资源比重很大。如果没有一个好的求逆算法，MIMO 检测实现起来将非常困难。本章首先介绍一下常用的几种求逆算法，然后再介绍全选主元高斯-约旦法，这是一种比较流行的矩阵求逆算法，常出现在软件程序员所写的代码之中，简要分析一下这种方案的优劣，以及为什么在本次设计中并没有选用的原因。接下来将重点介绍一种根据 MMSE 滤波器的特点，所引入的一种名为变量循环重新编号法的针对正定 Hermite 矩阵行之有效的求逆算法，这种算法也是本人调查过的求逆算法中最优的一种(只能实现对正定实对称阵和正定 Hermite 阵的求逆)，最后介绍本人在使用变量循环重新编号法过程中所发现的一些规律，对编号法又提出了一些行之有效的简化。

3.2 常用的矩阵求逆算法[13]

3.2.1 定义法

此法用于阶数较低的矩阵。当矩阵的行列式$|A|\neq 0$时，矩阵存在逆$AA^{-1}=I$。根据$AA^{-1}=I$列方程组就可求出方阵的逆。

3.2.2 伴随阵法

设矩阵$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$，当行列式$|A|\neq 0$时，$A^{-1}$存在，并且

$$A^{-1}=\frac{1}{|A|}A^{*}=\frac{1}{|A|}\times\begin{bmatrix}A_{11} & A_{21} & \cdots & A_{n1}\\ A_{12} & A_{22} & \cdots & A_{n2}\\ \cdots & \cdots & \cdots & \cdots\\ A_{1n} & A_{2n} & \cdots & A_{nn}\end{bmatrix} \tag{3.1}$$

其中矩阵 A^{*} 称为 A 的伴随矩阵，A_{ij} 为矩阵 A 中元素 a_{ij} 的代数余子式。用伴随矩阵求逆，当阶数很高时，计算量很大，对于二阶、三阶矩阵比较适用。

3.2.3 初等变换法

设 n 阶矩阵 A，作 n×2n 矩阵 $(A|E_n)$，然后对此矩阵只施以初等变换，把子块 A 变为 E_n，同时也将子块 E_n 变为 A^{-1}。即：

$$(A|E_n)\rightarrow(E_n|A^{-1}) \tag{3.2}$$

同时也可以作 2n×n 矩阵($\frac{A}{E_n}$)，然后对此矩阵只施以列变换，即：

$$\left(\frac{A}{E_n}\right)\rightarrow\left(\frac{E_n}{A^{-1}}\right) \tag{3.3}$$

3.2.4 Gauss-Jordan 法

由定义 $AA^{-1}=I$，设 $Y=AX$（X，Y 均为 n 维向量），则 $X=A^{-1}Y$。若将 $Y=AX$ 改写成 $X=BY$，则 $A^{-1}=B$。

3.3 全选主元(Gauss-Jordan)法

对于一个 n×n 阶的矩阵，高斯一约旦全选主元法求逆的步骤如下[12]：

首先，对于 k 从 0 到 n-1 作如下几步：

1. 从第 k 行、第 k 列开始的右下角子阵中选取绝值最大的元素，并记住此元素所在的行号和列号，在通过行交换和列交换将它交换到主元素位置上。这一步称为全选主元。
2. m(k,k)=1/m(k,k)

3. m(k,j)=m(k,j)×m(k,k)　　j=0,1,...,n-1;j≠k

4. m(i,j)=m(i,j)-m(i,k)×m(k,j)　　i, j=0,1,...,n-1; i,j≠k

5. m(i,k)=-m(i,k)×m(k,k)　　i=0,1,...,n-1; i≠k

6. 根据在全选主元过程中所记录的行、列交换的信息进行恢复，恢复的原则如下：在全选主元过程中，先交换的行(列)后进行恢复；原来的行(列) 交换用列(行)交换来恢复。

和常规的矩阵求逆算法相比，全选主元法求逆占用存储空间较小，运算量还是较小的，因而运算速度高，计算结果精度高。但是这种算法有排序操作，时间开销不固定，也不算非常适合硬件设计的需要。

3.4 变量循环重新编号法

3.4.1 算法

对于正定实对称阵或正定 Hermite 阵，存在一种名为变量循环重新编号法[12]的矩阵求逆方法，具体计算方法如下：

设 $a_{i,j}$ 为 A 的第 i 行第 j 列上的元素，则按下面一组公式进行运算得到另一个矩阵 A_1，

$$\begin{cases} a'_{mn}=1/a_{11} & \\ a'_{m,j-1}=-a_{1j}/a_{11} & (j=2,3,\cdots,n) \\ a'_{i-1,n}=a_{i1}/a_{11} & (j=2,3,\cdots,n) \\ a'_{i-1,j-1}=a_{ij}-a_{i1}\cdot a_{1j}/a_{11} & (j=2,3,\cdots,n) \end{cases} \tag{3.4}$$

$a'_{i,j}$ 为 A_1 中第 i 行第 j 列上的元素，然后将 A_1 中的元素(在这一次的循环中 $a_{i,j}$ 表示 A_1 中的元素，$a'_{i,j}$ 表示 A_2 中的元素)再次带入上述公式，得到的矩阵为 A_2，以后依次循环，直至求出 A_n 为止，A_n 便是 A 的逆矩阵。

3.4.2 推导过程

设 A 为 $n\times n$ 对称正定矩阵，则关系式

$$y = Ax \tag{3.5}$$

确定了 R_n 上的一个映象，如果能求出逆关系

$$x = By \tag{3.6}$$

则得到 A 的逆矩阵 $B = A^{-1}$。现将(3.5)详细写出如下：

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \cdots \qquad \cdots \qquad \cdots \\ y_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{cases} \tag{3.7}$$

由于 A 正定，必有 $a_{11} > 0$，故可以从(3.7)中第 1 个等式解出 x_1(称为交换 x_1 和 y_1 的位置)，再将 x_1 代入其他等式，得到新的关系式：

$$\begin{cases} x_1 = a'_{11}y_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n \\ y_2 = a'_{21}y_1 + a'_{22}x_2 + \cdots + a'_{2n}x_n \\ \cdots \qquad \cdots \qquad \cdots \\ y_n = a'_{n1}y_1 + a'_{n2}x_2 + \cdots + a'_{nn}x_n \end{cases} \tag{3.8}$$

这里，新系数的计算公式如下：

$$\begin{cases} a'_{11} = 1/a_{11} \\ a'_{1j} = -a_{1j}/a_{11} \quad (j = 2,3,\cdots,n) \\ a'_{i1} = a_{i1}/a_{11} \quad (i = 2,3,\cdots,n) \\ a'_{ij} = a_{ij} - a_{1i}a_{1j}/a_{11} \quad (i,j = 2,3,\cdots,n) \end{cases} \tag{3.9}$$

用同样的方法交换 x_2 和 y_2 的位置，如此继续，最后可得 $x = By$，B 就是所求的 A 的逆 A^{-1}。

但是，若按这种方式来运算，显得非常繁杂，容易混淆变量，为此，采用“变量循环重新编号法”来实施，使每一步的交换都在 x_2 和 y_2 的位置上进行。于是，类似于(3.9)的计算公式适用于每一个计算步骤，其计算公式如下：

$$\begin{cases} a'_{nn}=1/a_{11} & \\ a'_{n,j-1}=-a_{1j}/a_{11} & (j=2,3,\cdots,n) \\ a'_{i-1,n}=a_{i1}/a_{11} & (i=2,3,\cdots,n) \\ a'_{i-1,j-1}=a_{ij}-a_{i1}a_{1j}/a_{11} & (i,j=2,3,\cdots,n) \end{cases} \tag{3.10}$$

事实上，对(3.8)稍作改写可得

$$\begin{cases} y_2=a'_{22}x_2+a'_{23}x_3+\cdots+a'_{2n}x_n+a'_{21}y_1 \\ y_3=a'_{32}x_2+a'_{33}x_3+\cdots+a'_{3n}x_n+a'_{31}y_1 \\ \cdots \quad \cdots \quad \cdots \\ y_n=a'_{n2}x_2+a'_{n3}x_3+\cdots+a'_{nn}x_n+a'_{n3}y_1 \\ x_1=a'_{12}x_2+a'_{13}x_3+\cdots+a'_{1n}x_n+a'_{11}y_1 \end{cases} \tag{3.11}$$

显然，若对(3.11)式中变量按如下规则作一次重新编号：

$$\begin{array}{cccccccc} x_1 & x_2 & \cdots & x_k & \cdots & x_{n-1} & x_n \\ \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\ x_n & x_1 & \cdots & x_{k-1} & \cdots & x_{n-2} & x_{n-1} \\ y_1 & y_2 & \cdots & y_k & \cdots & y_{n-1} & y_n \\ \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\ y_n & y_1 & \cdots & y_{k-1} & \cdots & y_{n-2} & y_{n-1} \end{array} \tag{3.12}$$

则得到对于每一步均适用的变换公式(3.10)，其中变量的顺序经 n 次变换后恢复原状。

3.5 变量循环重新编号法的简化

在使用变量循环重新编号法求逆矩阵的过程中，会得到这样的结论：在第 $k(1\le k\le n)$次循环后，主对角线上的元素 $a_{i,i}(1\le i\le n)$永远为正实数，关于对角线对称的两个复数($a_{i,j}$与$a_{j,i}$)要么互为共轭复数，要么互为负共轭复数(即一个为另一个共轭复数的相反数)，他们的对应关系可以用图 3-1 来表示

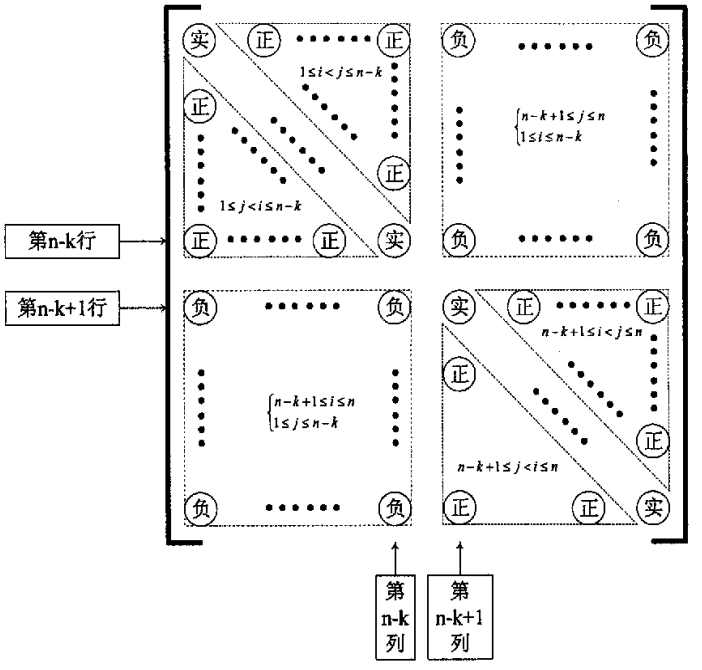

图3-1 第 k 次循环后的数据对称关系示意图

图 3-1 说明：

(实)：表示所在位置上的元素为一个正实数；

(正)：表示所在位置上的元素和关于主对角线对称位置上的元素为一对共轭对称复数；

(负)：表示所在位置上的元素和关于主对角线对称位置上的元素为一对负共轭对称复数。

即，如果按如图 3-1 所示的方式对矩阵进行分块为

$$\begin{matrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} n\text{-}k\text{行} \\ k\text{行} \end{matrix} \\ n-k\text{列} \quad k\text{列} \end{matrix} \tag{3.13}$$

则可以证明(证明见附录 C)：

1) 对于任何 k，A_{11} 和 A_{22} 都是正定 Hermite 阵；

2) $A_{21}=-A_{12}^{H}$。

显然，变量循环重新编号法求正定 Hermite 矩阵的逆矩阵算法中，存在大量的冗余运算。因此，可以利用对称性来对这种算法进行简化处理。具体简化方案有四种，分别表述如下：

✧ 简化方法一，在循环运算过程中利用对称关系求主对角线以下的元素

在第 k($1 \le k \le n$)次循环时，对角线及其以上的元素 $a'_{i,j}(i \le j)$仍按上述公式计算，即

$$\begin{cases} a'_{n,n} = 1/a_{1,1} & \\ a'_{i,n} = a_{i+1,1}/a_{1,1} & 1 \le i \le \text{n-1} \\ a'_{i,j} = a_{i+1,j+1} - a_{i+1,1}a_{1,j+1}/a_{1,1} & 1 \le i \le j \le \text{n-1} \end{cases} \tag{3.14}$$

主对角线以下的元素 $a'_{i,j}(i > j)$靠对称性算出，即

$$a'_{i,j} = \begin{cases} \overline{a'_{j,i}} & 1 \le j < i \le n-k \text{或} n-k+1 \le j < i \le n \\ -\overline{a'_{j,i}} & n-k+1 \le i \le n \text{且} 1 \le j \le n-k \end{cases} \tag{3.15}$$

✧ 简化方法二，在循环运算过程中利用对称关系求主对角线以上的元素

在第 k($1 \le k \le n$)次循环时，对角线及其以下的元素 $a'_{i,j}(i \ge j)$仍按上述公式计算，即

$$\begin{cases} a'_{n,n} = 1/a_{1,1} & \\ a'_{n,j} = -a_{1,j+1}/a_{1,1} & 1 \le j \le \text{n-1} \\ a'_{i,j} = a_{i+1,j+1} - a_{i+1,1}a_{1,j+1}/a_{1,1} & 1 \le j \le i \le \text{n-1} \end{cases} \tag{3.16}$$

主对角线以上的元素 $a'_{i,j}(i < j)$靠对称性算出，即

$$a'_{i,j} = \begin{cases} \overline{a'_{j,i}} & 1 \le i < j \le n-k \text{或} n-k+1 \le i < j \le n \\ -\overline{a'_{j,i}} & n-k+1 \le j \le n \text{且} 1 \le i \le n-k \end{cases} \tag{3.17}$$

✧ 简化方法三，利用上三角(包括主对角线)中的元素进行循环运算。循环完毕后，根据对称性求下三角(不包括对主角线)中的元素。

在第 k($1 \le k \le n$)次循环时上三角形内的数据 $a'_{i,j}(i \le j)$按下述公式运算，

$$
\begin{aligned}
&a'_{n,n} = 1/a_{1,1} \\
&a'_{i,n} = \begin{cases} \overline{a_{1,i+1}}/a_{1,1} & 1 \le i \le \text{n-k}; \\ -\overline{a_{1,i+1}}/a_{1,1} & \text{n-k+1} \le i \le \text{n-1}; \end{cases} \\
&a'_{i,j} = \begin{cases} a_{i+1,j+1} - \overline{a_{1,i+1}} a_{1,j+1}/a_{1,1} & 1 \le i \le \text{n-k}; \\ a_{i+1,j+1} + \overline{a_{1,i+1}} a_{1,j+1}/a_{1,1} & \text{n-k+1} \le i < j \le \text{n}; \end{cases} \\
&a'_{i,i} = \begin{cases} a_{i+1,i+1} - |a_{1,i+1}|^2/a_{1,1} & 1 \le i \le n-k; \\ a_{i+1,i+1} + |a_{1,j+1}|^2/a_{1,1} & n-k+1 \le i \le n; \end{cases}
\end{aligned} \tag{3.18}
$$

循环 n 次后根据对称性求下三角形内的数据 $a'_{i,j}(i>j)$，即

$$a'_{i,j} = \overline{a'_{j,i}}。$$

✧ 简化方法四，利用下三角(包括主对角线)中的元素进行循环运算。循环完毕后，根据对称性求上三角(不包括对主角线)中的元素。

在第 k($1 \le k \le n$)次循环时下三角形内的数据 $a'_{i,j}(i \ge j)$按下述公式运算

$$
\begin{aligned}
&a'_{n,n} = 1/a_{1,1} \\
&a'_{n,j} = \begin{cases} -\overline{a_{j+1,1}}/a_{1,1} & 1 \le j \le \text{n-k}; \\ \overline{a_{j+1,1}}/a_{1,1} & \text{n-k+1} \le j \le \text{n-1}; \end{cases} \\
&a'_{i,j} = \begin{cases} a_{i+1,j+1} - a_{i+1,1}\overline{a_{j+1,1}}/a_{1,1} & 1 \le j \le \text{n-k}; \\ a_{i+1,j+1} + a_{i+1,1}\overline{a_{j+1,1}}/a_{1,1} & \text{n-k+1} \le j < i \le \text{n}; \end{cases} \\
&a'_{j,j} = \begin{cases} a_{j+1,j+1} - |a_{j+1,1}|^2/a_{1,1} & 1 \le j \le n-k; \\ a_{j+1,j+1} + |a_{j+1,1}|^2/a_{1,1} & n-k+1 \le j \le n; \end{cases}
\end{aligned} \tag{3.19}
$$

循环 n 次后根据对称性求上三角形内的数据 $a'_{i,j}(i<j)$，即

$$a'_{i,j} = \overline{a'_{j,i}}$$

第四章 硬件平台介绍

4.1 FPGA 简介

在数字化、信息化的时代，数字集成电路应用得非常广泛。随着微电子技术与工艺的发展，数字集成电路从电子管、晶体管、中小规模集成电路(VLSIC)逐步发展到今天的专用集成电路(ASIC)。ASIC 的出现降低了产品的生产成本，提高了系统的可靠性，减少了产品的物理尺寸，推动了社会的数字化进程。但是 ASIC 因其设计周期长，改版投资大，灵活性差等缺陷制约着它的应用范围。硬件工程师希望有一种更灵活的设计方法，根据需要，在实验室就能设计、更改大规模数字逻辑，研制自己的 ASIC 并马上投入使用。这就是可编程逻辑器件提出的基本思想。

可编程逻辑器件随着微电子制造工艺的发展取得了长足的进步。从早期的只能存储少量数据，完成简单逻辑功能的可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)，发展到能完成中大规模的数字逻辑功能的可编程阵列逻辑(PAL)和通用阵列逻辑，今天已经发展成为可以完成超大规模的复杂组合逻辑与时序逻辑的现场可编程逻辑器件(FPGA)和复杂可编程逻辑器件(CPLD)。随着工艺技术的发展与市场的需求，超大规模、高速、低功耗的新型 FPGA/CPLD 不断推陈出新。新一代的 FPGA 甚至集成了中央处理器(CPU)或数字处理器(DSP)内核，在一片 FPGA 上进行软硬件协同设计，为实现片上可编程系统(SOPC, System On Programmable Chip)提供了强大的硬件支持。

4.1.1 FPGA 的基本原理

简化的 FPGA 结构有四部分组成：输入/输出模块、二维逻辑阵列模块、连线资源和内嵌存储器，如图 4-1 所示。输入/输出模块是芯片与外界的接口，完成不同电器特性下的输入输出功能要求；二维逻辑阵列模块是可编程逻辑的主体，可以根据设计灵活地改变连接与配置，完成不同的逻辑功能；连线资源连接所有的二维逻辑阵列模块和输入/输出模块，连线的长度和工艺决定着信号在连接线上的驱动能力和传输速度；内嵌存储器结构可以在芯片内部存储数据。

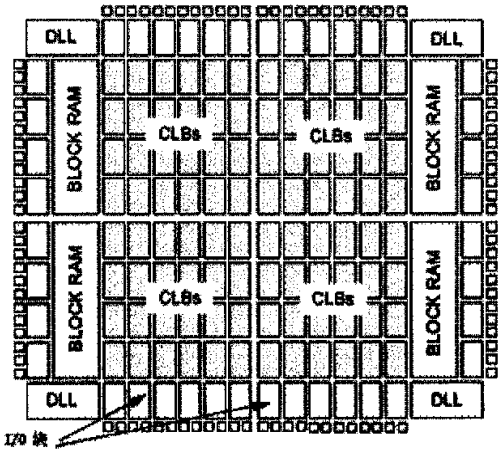

图4-1 简化的 FPGA 结构

CLB(Configurable Logic Blocks)可配置逻辑模块，是 FPGA 中的最小逻辑单元。单一 CLB 结构如下：

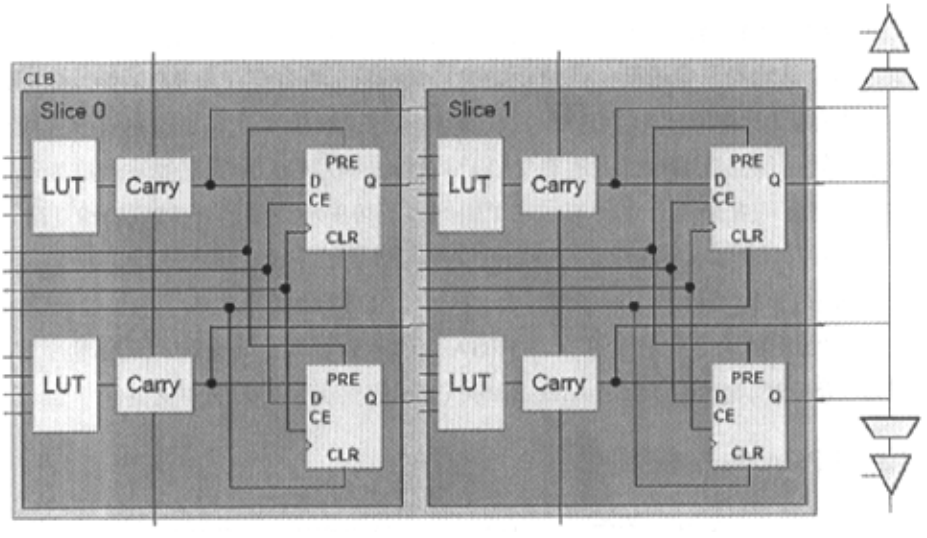

图4-2 单一 CLB 结构

由图 4-2 可以看出，CLB 的功能是通过查找表(Look-Up-Table, LTU)来实现的。LTU 通常是由静态存储器(SRAM)构成函数发生器，由它来控制执行 FPGA 应用函数的逻辑。SRAM 的地址起输入作用，SRAM 的输出为逻辑函数的值，由此输出状态控制传输门或多路开关信号的通断，实现与其他功能块的可编程连接。

4.1.2 FPGA 的特点

FPGA 既继承了 ASIC 的大规模、高集成度、高可靠性的优点，又克服了普通 ASIC 设计周期长、投资大、灵活性差的缺点，逐步成为复杂数字硬件电路设计中的理想首选。当代 FPGA 有以下特点：

- ✧ 规模越来越大。单一芯片内部可以容纳上百万个晶体管，FPGA 芯片的规模也越来越大。单一逻辑门数已逾百万，芯片的规模越大所能实现的功能就越强，同时也更适于实现片上系统(SOC，System On Chip)。
- ✧ 开发过程投资小。FPGA 芯片在出厂前都做过百分之百的测试，而且 FPGA 设计灵活，发现错误时可直接更改设计，减少了投片风险，节约了许多潜在的花费。
- ✧ FPGA 一般可以反复的编程、擦除。在不改变外围电路的情况下，设计不同片内逻辑界能实现不同的电路功能。
- ✧ 保密性能好。在某些场合下，根据要求选用防止反向技术的 FPGA，能很好地保护系统的安全性和设计者的知识产权。
- ✧ FPGA 开发工具智能化，功能强大。

新型 FPGA 内嵌 CPU 或 DSP 内核，支持软硬件协同设计，可以做为片上可编程系统(SOPC)的硬件平台。

4.2 Xilinx Virtex-II Pro 系列芯片介绍[16]

4.2.1 综述

Virtex-II Pro 系列 FPGA 芯片是 Xilinx 公司在 2002 年推出的功能强大的 FPGA 芯片，采用 0.13mm，1.5V 工艺技术，最高速率达 400MHz 以上，多达 444 个 18X18 嵌入式乘法器，并集成了嵌入式 IBM PowerPC 405 处理器硬核，拥有传输速率高达 3.125 Gbps RocketI/O 串行收发器和单端和差分电子 I/O，具备强大的连接功能。

Virtex-II Pro 系列 FPGA 芯片的许多创新特性使可编程逻辑的设计工程师可以在 FPGA 平台上创建完备的系统，满足对先进应用速度、性能和高级特性的需求。

它将加速专用 ASIC 产品向 FPGA 的转移，特别是在光纤网络系统，前兆路由器，无线蜂窝系统，视频广播系统得到应用。

4.2.2 资源列表

Virtex-II Pro 系列的产品比较多，支持的不同用户范围很广，基本型号的资源情况参看表 4-1。

表4-1 Virtex-II Pro 系列 FPGA 资源列表

Device[1]	RocketIO Transceiver Blocks	PowerPC Processor Blocks	Logic Cells[2]	CLB (1 = 4 slices = max 128 bits)		18 X 18 Bit Multiplier Blocks	Block SelectRAM+		DCMs	Maximum User I/O Pads
				Slices	Max Distr RAM (Kb)		18 Kb Blocks	Max Block RAM (Kb)		
XC2VP2	4	0	3,168	1,408	44	12	12	216	4	204
XC2VP4	4	1	6,768	3,008	94	28	28	504	4	348
XC2VP7	8	1	11,088	4,928	154	44	44	792	4	396
XC2VP20	8	2	20,880	9,280	290	88	88	1,584	8	564
XC2VPX20	8[4]	1	22,032	9,792	306	88	88	1,584	8	552
XC2VP30	8	2	30,816	13,696	428	136	136	2,448	8	644
XC2VP40	0[3], 8, or 12	2	43,632	19,392	606	192	192	3,456	8	804
XC2VP50	0[3] or 16	2	53,136	23,616	738	232	232	4,176	8	852
XC2VP70	16 or 20	2	74,448	33,088	1,034	328	328	5,904	8	996
XC2VPX70	20[4]	2	74,448	33,088	1,034	308	308	5,544	8	992
XC2VP100	0[3] or 20	2	99,216	44,096	1,378	444	444	7,992	12	1,164

4.2.3 芯片结构

Virtex-II Pro 系列 FPGA 芯片的基本结构如图 4-3 所示

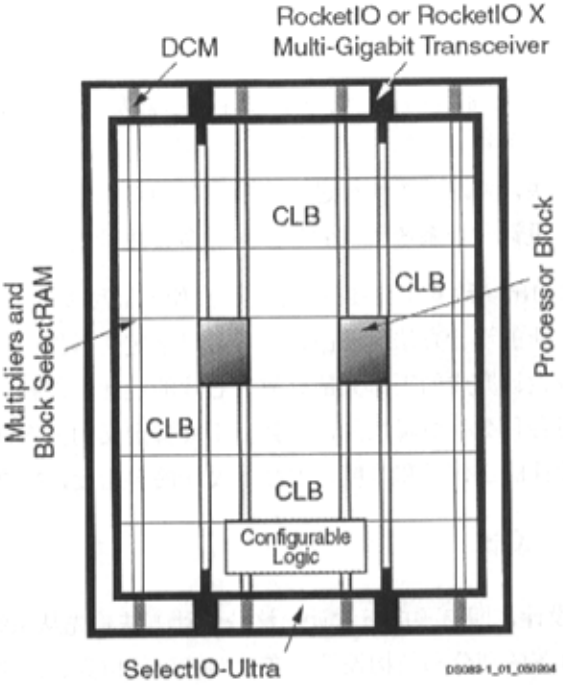

图4-3 Virtex-II Pro 系列的结构示意图

4.2.4 CLB 结构

Virtex-II Pro 系列的单一 CLB 结构如图 4-4 所示

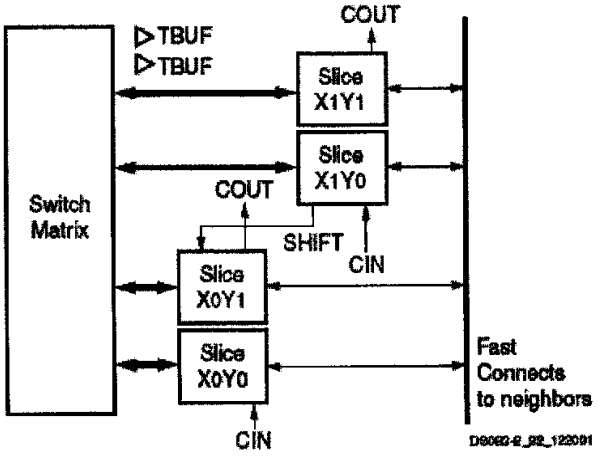

图4-4 Virtex-II Pro 系列的单一 CLB 结构

4.3 FPGA 的设计方法与流程

近 20 年来，电子系统的设计方法和设计技术都发生了深刻的变化。在以前，数字系统大多是采用搭积木的方式设计的，即由一些固定功能的器件加上一定的外围电路构成模块，由这些模块再进一步形成各种功能电路。在设计时，几乎没有灵活性可言，设计一个系统所需的芯片种类多且数量大。

FPGA 技术的出现改变了传统的设计思路，使人们可以通过设计芯片来实现各种不同的功能，新的设计方法能够由设计者自己来定义器件内部的逻辑和管脚，将原来由电路板设计完成的工作大部分放在芯片的设计中进行。这样不仅可以通过芯片设计实现各种数字逻辑功能，而且由于管脚定义的灵活性，大大减轻了原理图和印制板设计的工作量和难度，增减了设计的自由度，提高了效率。

4.3.1 Top-down 设计

Top-down 设计，即自顶向下设计。这种设计方法首先从系统设计入手，在顶层进行功能方框图的划分和结构设计。在功能级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述，然后用综合工具将设计转化为具体门电路网表，其对应的物理实现就是 FPGA 电路。由于设计的主要仿真和调试过程是在高层次完成的，这不仅有利于早期发现结构设计上的错误，避免设计工作的浪费，而且也减少了逻辑功能仿真的工作量，提高了设计的一次成功率。

4.3.2 IP 复用技术

当电子系统的设计越来越向高层发展的时候，基于 IP 复用(IP reuse)的设计技术越来越显示出其优越性。IP(Intellectual Property)，其原来的含义是指知识产权、著作权等，在 IC 设计领域可将其理解为实现某种功能的设计，IP 核(IP 模块)则是指完成某些功能的设计模块。

设计者在设计一个系统时，可以自行设计各个功能模块，也可以用 IP 模块来构建。作为设计者来说，想要在短时间内开发出新产品，一个比较好的方法就是使用 IP 核来完成设计。基于 IP 复用的设计技术给设计这带来了诸多好处，如节省时间、缩短开发周期、避免重复劳动等等。

4.3.3 FPGA 中的流水线设计技术

流水设计的要领是把在一个周期内执行的逻辑操作分成几步较小的操作，并在多个较高速的时钟周期内完成。

在图 4-5 中，图(a)的数据通路中的逻辑被分为三个小部分。如果它的 Tpd 为 X，则该电路的最高时钟频率为 1/X。而在图(b)中，假设在理想情况。

在计算中并没有包括电路中寄存器的时钟到输出的时延以及信号的建立时间，因此实际的延迟应比 X/3 稍大。在忽略它们的情况下，可以看到流水线技术可以用来提高系统的数据流量，也就是在单位时间内所处理的数据量。但是，采用这种方法的代价是输出信号将相对于输入滞后 3 个时钟周期。

总之，流水技术在提高系统处理速度的同时也造成了输出滞后，并且还需要额外的寄存器资源，但由于大多数 FPGA 器中的每个单元都有寄存器，因由使于使用流水设计技术。在整个课题研究中，流水技术得到了大量的使用。

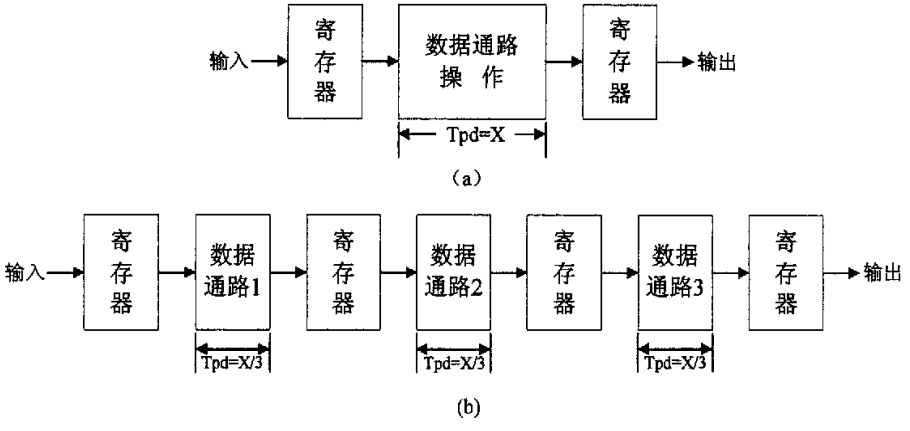

图4-5 流水线设计示意图

4.3.4 FPGA 的设计流程

一个完整的 FPGA 设计流程包括电路设计与输入、功能仿真、综合、综合后仿真、实现、布线后仿真和下板调试等主要步骤，如图 4-6 所示。

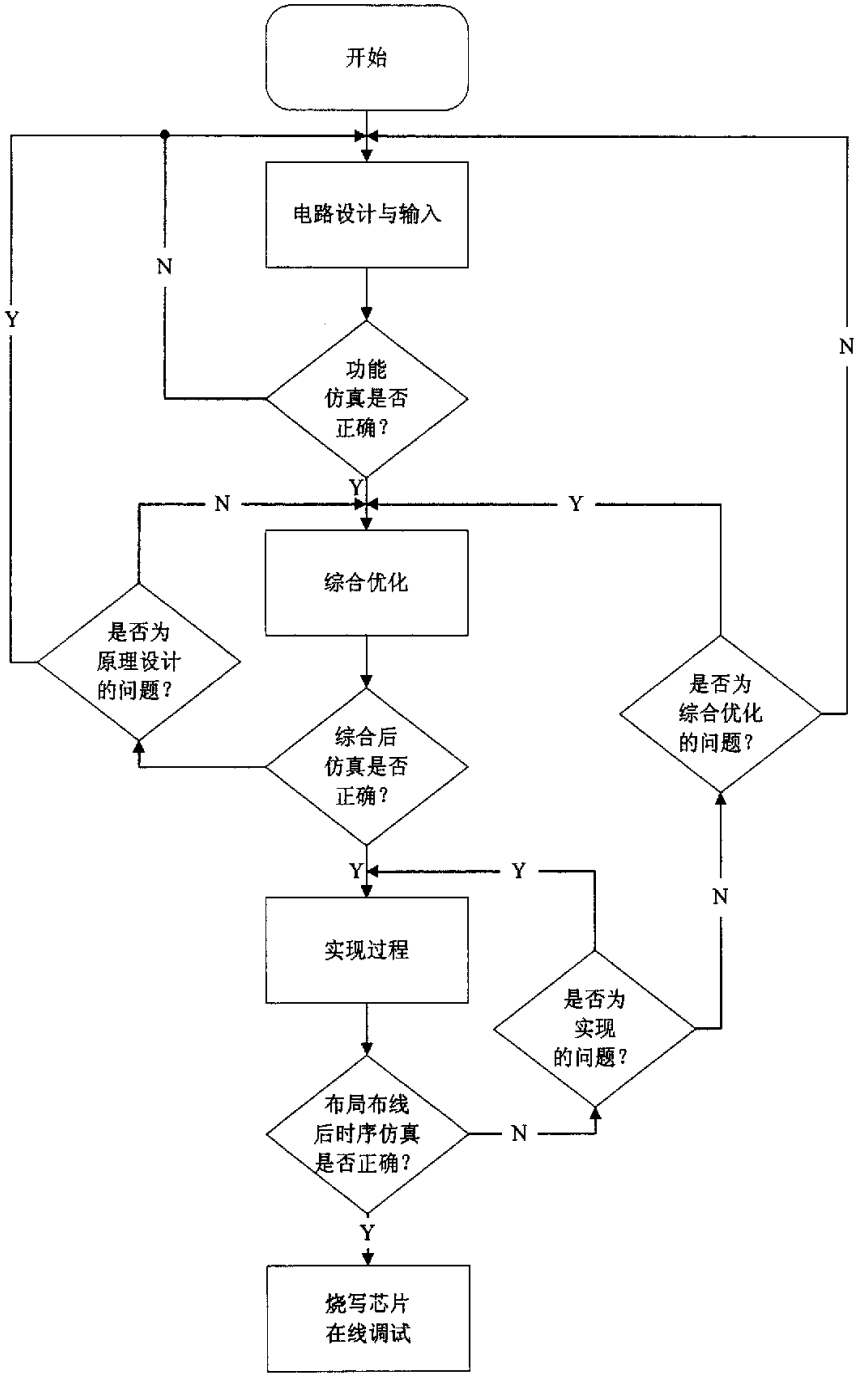

图4-6 完整的 FPGA 设计流程

4.4 Verilog HDL 简介

Verilog HDL 是在使用最广泛的 C 语言的基础上发展起来的一种硬件描述语言。1983 年，GDA(Gateway Design Automation)公司的 Phil Moorby 首创 Verilog HDL，最初只设计了一个仿真与验证工具，之后又陆续开发了相关的故障模拟与时序分析工具。1985 年，Moorby 推出它的第三个商用仿真器 Verilog-XL，获得了巨大的成功，从而使得 Verilog HDL 迅速得到推广应用。1989 年，Cadence 公司收购了 GDA 公司，使得 Verilog HDL 成为了该公司的独家专利。1990 年，Cadence 公司公开发表了 Verilog HDL，并成立 OVI(Open Verilog Internation)组织以促进 Verilog HDL 的发展。1995 年，Verilog HDL 成为 IEEE 标准，即 IEEE Standard 1364-1995。

Verilog HDL 的最大特点就是易学易用、语法较自由，描述功能强，从高层的系统描述到底层的版图设计，都能很好的支持。由于 Verilog HDL 的优越性，目前已得到了广泛的使用，在 ASIC 和 FPGA 设计领域更是处于主流地位。

4.5 小结

本章首先介绍了 FPGA 的基本知识，然后详细介绍了本论文设计中所采用的 Xilinx Virtex-II pro 系列芯片。最后简述了 FPGA 设计方法与流程，以及开发 FPGA 的硬件描述语言,为后续实现工作做必要准备。

第五章 MIMO 检测的 FPGA 实现

5.1 引言

本章是论文的重点，在本章里将详细讲述 MIMO 检测的 FPGA 实现。本设计采用的硬件描述语言为 Verilog HDL 语言，采用的工具为 Xilinx 公司的 ISE 集成开发工具、Model Tech 公司的仿真工具 ModelSim 等。在设计中所采用的算法为简化的 MMSE-LLR 检测算法；采用的定点数格式为 1 个符号位 4 个整数位 11 个小数数位，这是前期定点仿真的结果，兼顾了性能和实现复杂度。

本章所重点讲述的矩阵求逆、幅度自动调整等模块是整个 MIMO 检测实现的难点。从 2.3 节中可以看出，整个 MIMO 检测中除矩阵求逆运算以外的其他运算多为矩阵和矩阵相乘、矩阵和向量相乘、向量和向量相乘等大规模乘法运算，本文挑选了一个有代表性的矩阵乘法运算模块来做讲解。其他运算模块大多可以采用设计好的底层模块，用类似搭积木的方式来设计。

5.2 系统需求分析

由表 2-1 可以看出，在峰值时期，系统将在 $62\mu s$ 的时间内，从链路前端沿单根天线传来 884 个有效子载波(即 16-QAM 符号)，也就是说单天线的理论符号传输率为 14.26MBaud/S，实际传输速率应大于或等于此传输速率。因此，本链路中 MIMO 检测模块与上级模块之间数据传输速率定为 20M 组/秒，这里的组是指一个完整的信号矢量 y(每根天线发过来一个 QAM 符号)、与此对应的一组信道信息 H 以及天线上的噪声功率 σ_n^2。本地晶振产生 40MHz 时钟，可以经 DCM 倍频或分频后使用。为了充分发挥 Xilinx Vertex-II pro vp70 芯片的性能，在设计中将 MIMO 检测部分的时钟频率定为 80MHz，接收到的数据经过 80MHz 的时钟采样和其他处理后供 MIMO 检测模块使用。采样处理前后的标志信号和数据波形如图 5-1 所示。

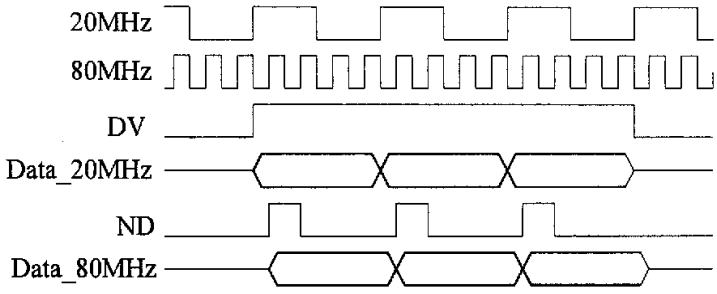

图5-1 输入端信号处理

图 5-1 说明：

- ✧ 20MHz 时钟和 80MHz 时钟是由本地晶振产生的 40MHz 时钟经同一个 DCM 分频和倍频后得来；
- ✧ Data_20MHz 指的是从上级模块传递到 MIMO 检测片子上的有效数据，包括信道信息 H、信号 y、噪声方差 σ_n^2，总共 41 路信号并行输入；
- ✧ DV 为这些数据的有效标志信号，当 DV 为高电平时，表示数据线上的数据有效，否则为无效；
- ✧ Data_80MHz 是 Data_20MHz 信号经 80MHz 的时钟采样后而得到的，因此，这些信号一定是按 4 个 80MHz 时钟周期为变化周期来进行变化(即，在连续的 4 个时钟周期内一定保持不变)；
- ✧ ND(new data)信号为新数据有效信号，当 ND 为高电平时，表示该时钟及其接下来的 3 个时钟里的数据为有效的新数据，ND 一定是出现在 Data_80MHz 有效的第 1 个时钟周期内，且宽度仅为 1 个 80MHz 的时钟周期。

MIMO 的下级模块为 LDPC 译码模块，MIMO 检测模块算出的对数似然比(LLR)按如图 5-2 所示的方式排列好，用 Rocket I/O 传输到 LDPC 译码模块。

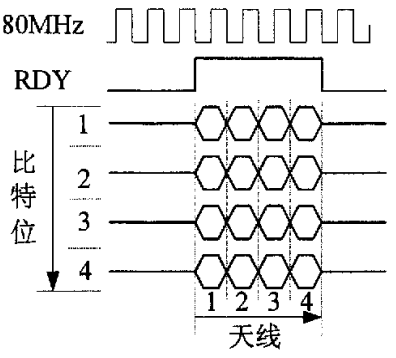

图5-2 输出信号格式

5.3 系统总体方案设计

由上一节的分析可以看出，在MIMO检测模块内部实际的数据传输速率20MHZ，而处理时钟为80MHZ，也就是说处理1组数据有4个时钟周期可用。如果按常规意义上的流水设计将使得每一个子器件在4个时钟周期内计算的同一个数据，这样将造成巨大的浪费。本节的一个重要任务就是讨论如何复用器件，提高器件的利用率。

由图2-8可以看出，整个MIMO检测的运算主要分两类：

1) 4天线数据混合运算，即4根天线数据到齐了才能进行运算。如，$\left[\mathbf{H}\mathbf{H}^{H}+4\sigma_{n}^{2}\mathbf{I}_{4}\right]^{-1}$；
2) 4天线数据独立运算，即在本步骤中4根天线上的数据独立，不需要数据交互。

对于四天线独立运算模块，可以采用如图5-3所示的方法来处理，先把并行的四根天线上的数据转化为四天线串行(并\串变换参看5.6节)，然后依次进入运算模块，做到四天线分时复用的目的。这样，将全并行时的四份完全相同的器件变成了一份，资源直接节省为原来的四分之一。

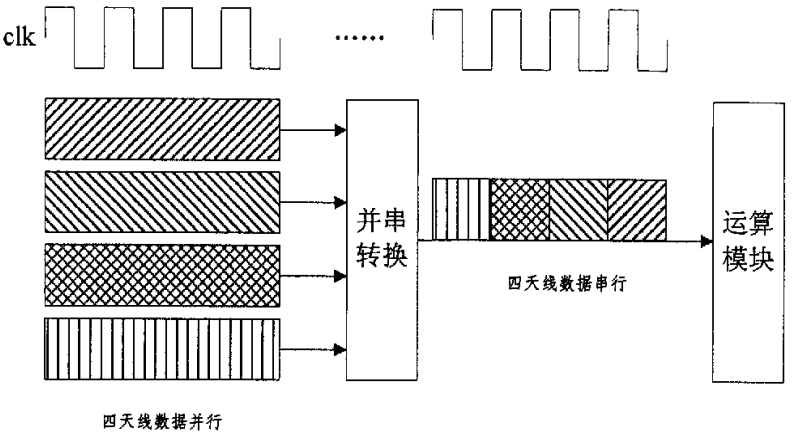

图5-3 四天线分时复用原理图

图 5-3 说明：

不同底纹的方块表示不同天线上的数据，并行时每个数据维持 4 个时钟周期，串行时每个数据仅维持一个时钟周期。

5.4 系统总体模块划分

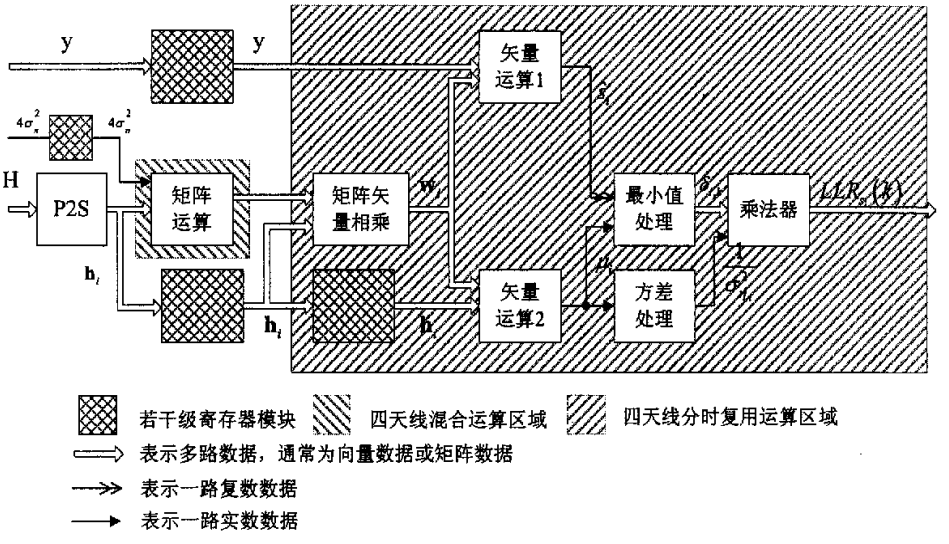

图5-4 MIMO 模块划分框图

图 5-4 各模块的主要作用：

1) 寄存器模块，主要的作用是将该路输入信号延迟几个时钟，以便和其他信号对齐。每个运算模块的各路输入都应同时到达。如果不能满足要求，就借助于寄存器模块来实现对齐操作，在各个模块内部也存在此类模块。

2) P2S 模块，并\串转换模块，详细描述见 5.6 节。

3) 矩阵运算模块的主要作用是实现$[\mathrm{HH}^H+4\sigma_n^2]^{-1}$，实际上这一步的实现要分为三个步骤，分别是(1)$\mathrm{HH}^H$；(2)$\mathrm{HH}^H+4\sigma_n^2$；(3) $[\mathrm{HH}^H+4\sigma_n^2]^{-1}$。其中第一步和第三步是实现的难点，将在后面的章节中作详细介绍。

4) 矩阵矢量相乘模块的主要作用是求滤波因子$\mathbf{w}_i$，即公式(2.22)

5) 矢量运算 1 模块的主要作用是用滤波因子$\mathbf{w}_i$对接收信号 y 进行滤波，即公式$\hat{s}_i=\mathbf{w}_i^H\mathbf{y}$

6) 矢量运算 2 实现的功能是$\mu_i=\mathbf{w}_i^H\mathbf{h}_i$，此模块表面看来与矢量运算 1 很相似，实际则不同，不同之处在于矢量运算 1 的输出结果$\hat{s}_i$为复数，而矢量运算 2 的输出μ_i理论上是一个正实数。

7) 最小值处理模块实现的功能是：
$$\delta_{i,k}=\min_{c_k=0}\left(\left|\hat{s}_i-\mu_i a_n\right|^2\right)-\min_{c_k=1}\left(\left|\hat{s}_i-\mu_i a_n\right|^2\right)$$

8) 方差处理模块实现的功能是：$\sigma_{\eta_i}^2=\frac{1}{4}\left(\mu_i-\mu_i^2\right)$，查表求$\frac{1}{\sigma_{\eta_i}^2}$

9) 乘法器模块实现的功能是：$\mathrm{LLR}_{S_i}(k)=\delta_{i,k}\times\frac{1}{\sigma_{\eta_i}^2}$。

5.5 常用底层模块的设计

5.5.1 定点数运算规律

所谓的定点数就是小数点的位置在数中固定不变，这种机器数称为定点数。例如，在本设计中的大部分数据采用的格式(如图 5-5 所示)，就是定点数格式。与此相对应的是浮点数，即小数点的位置在数中是浮动的。

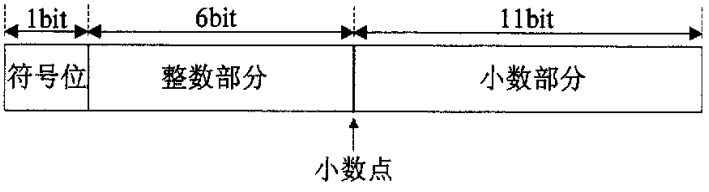

图5-5 MIMO 检测输入定点数格式

在基于 FPGA 设计的硬件系统中，数据一般都采用补码形式表示，按补码的运算规律进行运算。在运算过程中，当数的绝对值超过所用二进制位数所允许表示的最大值时，就会发生溢出，如果不进行溢出处理就会造成运算错误，如用 8 位二进制数表示一个定点数，其补码的范围为-128~+127。

FPGA 芯片中数据的运算规则为[17]：

- ✧ 参与运算的数据为定点数，小数点的位置要靠设计者自己把握。
- ✧ 参与运算的数据都用补码表示，按补码运算规律进行运算。
- ✧ 符号位参与运算。
- ✧ 当数据在运算过程中产生溢出时，应该进行溢出处理，否则将会出现错误。

5.5.2 加减法器件的设计

补码的加法运算

$$[x+y]_{补}=[x]_{补}+[y]_{补} \tag{5.1}$$

补码的减法运算

$$[x-y]_{补}=[x]_{补}+[-y]_{补} \tag{5.2}$$

不管 y 的真值为正还是为负，已知 $[y]_{补}$ 求 $[-y]_{补}$ 的方法是：将 $[y]_{补}$ 连同符号位一起变反，末位加“1”。

采用 IP core 设计的加减法器件则可以避开上述繁琐的判断，例如 Xilinx 公司就存在一种名为 Adder Subtracter 的 IP core，这种 IP core 有加减标志位(ADD)，当 ADD 为高电平时做加法运算，低电平时做减法运算。为了避免溢出，输出的结果一定要比输入多 1 位。例如输入的数据 A、B 都是 18 bit 二进制补码数据，那么加减后的结果 Q 为 19 bit 二进制补码数据，具体格式见图 5-6。可以认为输出数据的整数位比输入数据的整数位增加了一位，结果为常规意义的二进制补码数据；也可以认为输出数据的整数位和小数位分别与输入数据相同，符号位被扩展。如果要将输出数据规格化为输入数据格式，那么就要进行溢出处理，具体的处理过程见表 5-1。

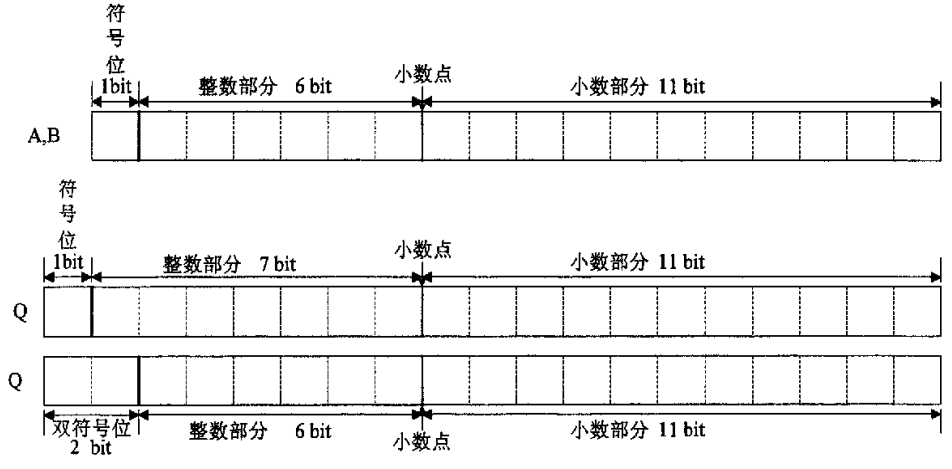

图5-6 加减法输入输出数据格式

表5-1 双符号位的溢出判断表

双符号位	溢出判断	处理方式
00	结果为正，无溢出	截取低 18 位
01	结果为正，发生上溢	上限幅
10	结果为负，发生下溢	下限幅
11	结果为负，无溢出	截取低 18 位

5.5.3 乘法器的设计

两个定点数相乘，结果位宽为输入数据的两倍，即整数位(包括符号位)、小数

位分别为输入数据的两倍。如图 5-7 所示，

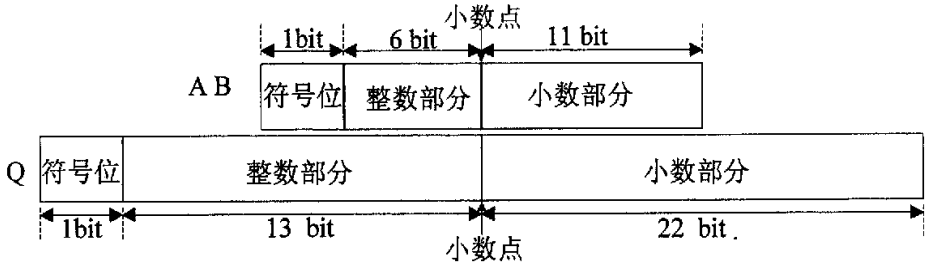

图5-7 乘法器输入输出数据格式

如果要规格化为输入数据格式，则需对整数位进行限幅处理，小数位进行舍入处理。舍入的原则是：使本次舍入所造成的误差以及按此舍入规则产生的累计误差都比较小，下面介绍常用的 3 种舍入处理方案。

1) 0 舍 1 入法

类似于十进制中的“四舍五入法”。在舍入前，先判断即将舍去比特位中的最高位，如果是“0”直接将这些位舍去，如果是“1”则将这些位舍去后，在新数据的最低位加“1”。0 舍 1 入法的舍入函数曲线见图 5-8，横轴表示实际大小，纵轴表示舍入后的结果。

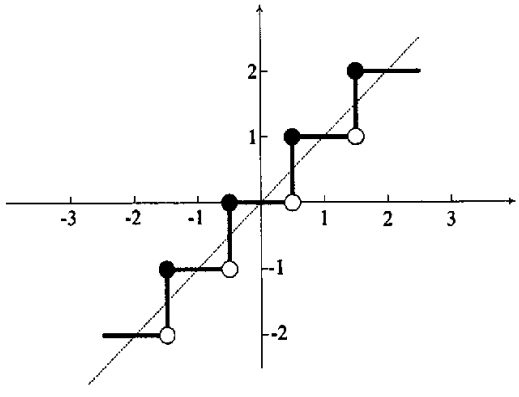

图5-8 0 设 1 入法函数曲线

2) 截断法

截断法最容易实现，无需判断要截去的那些位的实际大小，直接截断即可。

此类方法在 FPGA 中最容易实现，截断法的舍入函数曲线如图 5-9 所示。

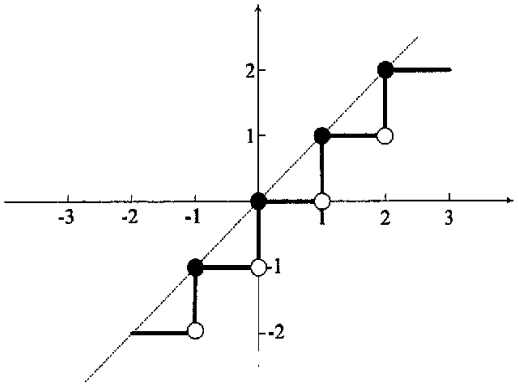

图5-9 截断法函数曲线

3) 末位置 1 法

这种方法比较简单，没有进位、借位运算，在逻辑上容易实现。具体的处理方法是：将要舍去的位直接舍去，将新数据的最低位恒置“1”。末位置一法的函数曲线如图 5-10 所示。

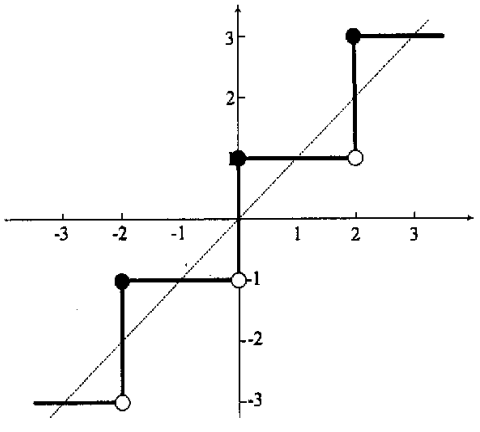

图5-10 末尾置 1 法函数曲线

有上述函数曲线可以看出，0 舍 1 入法每次所造成的误差最小且累计误差也最小，但由于存在判断与进位运算，故它是这三种方案中最耗资源的一种舍入方案，三种舍入方案的优劣性比较参看表 5-2。

表5-2 三种舍入方案优劣比较

舍入方法	单次误差	累计误差	FPGA 实现难度
0 舍 1 入	最小	较小	难
截断法	较小	较大	易
末位置 1 法	较大	较小	易

5.5.4 乘加器的设计

乘加器的主要作用是实现下述运算

$$Q=\sum_{0}^{count-1}A(n)\times B(n) \tag{5.3}$$

其电路结构如图 5-11 所示

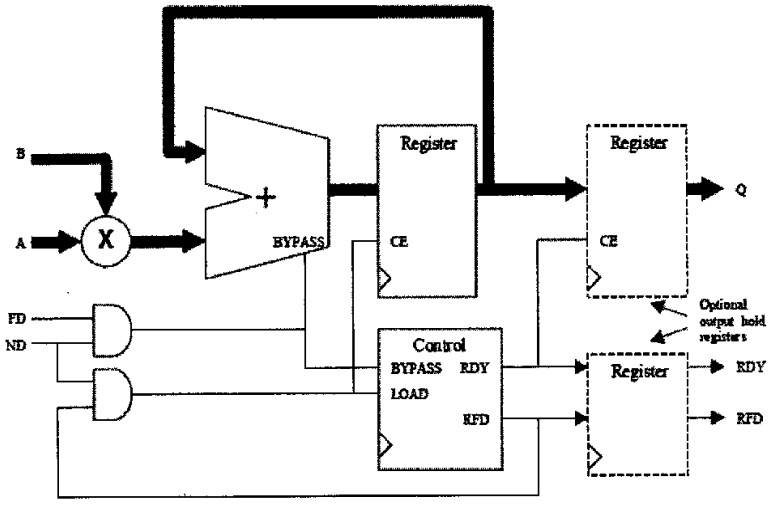

图5-11 乘加器电路图

图 5-12 是 count=4 的乘加器的输入输出波形示意图

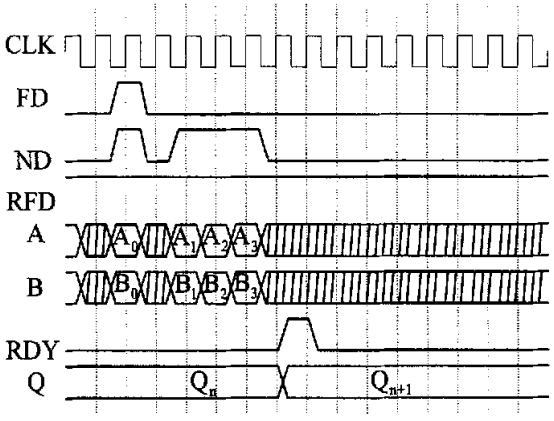

图5-12 乘加器的输入输出波形

值得注意的是如果输入数据 A、B 的位宽为 p 那么输出 Q 的位宽应该为 2p+2(count=4)，采用与乘法器相同的原理对输出结果进行限幅与舍入操作。

5.5.5 用查表法实现非线性运算

在 FPGA 设计中，非线性运算一般通过查表法来实现，如对数函数、指数函数、三角函数等等。查表法的运算规则是：将自变量作为地址，存储器输出作为函数值，通过读取存储器的方式来实现函数运算。本设计中的求倒运算是通过查表法来实现的。当地址位宽较大时，可以采用插值法来减小表的大小，依此来节省存储资源。

在 MIMO 检测模块中用到的其他底层器件还很多，譬如比较器、累加器等等，它们的实现也很简单，大部分都有现成的 IP core 可以利用，在此不一一赘述。

5.6 P2S 模块的设计与实现

P2S 模块的主要作用是实现 **H** 矩阵中各行元素的从并到串的转换。这里的串行不是指单一数据中高低位的串行，而是指矩阵中各行数据按时间串行排列。转换前矩阵中各行数据并行排列，总共占用 4×36 条物理线路，串行后为 36 条物理线路。并\串转换的用途有两个：

1) 满足矩阵乘法 $\mathbf{HH}^H$ 的需要，做矩阵乘法时用到的乘加器(MAC)要求输入数据串行输入；

2) 满足后级天线间的分时复用需要。

常规的并\串转换电路如图 5-13 所示。

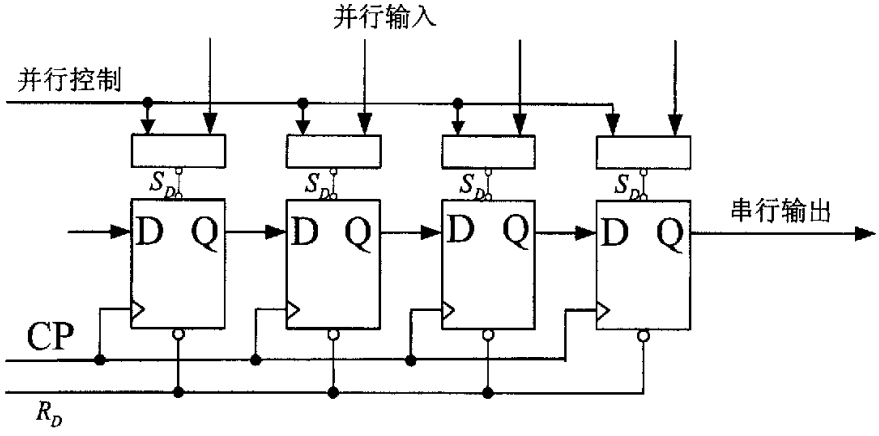

图5-13 并\串转换电路

尽管这里的并\串转换概念与常规的不同，但是也可以采用类似的方法来设计。用 ND 信号作为并\串转换的指示信号，当 ND 为高电平时，四路数据并行写入 4 个串联的寄存器中，当 ND 为低电平时，存进去的数据在寄存器中串行流动，从而达到了并\串转换的目的。

行为仿真后的波形如下

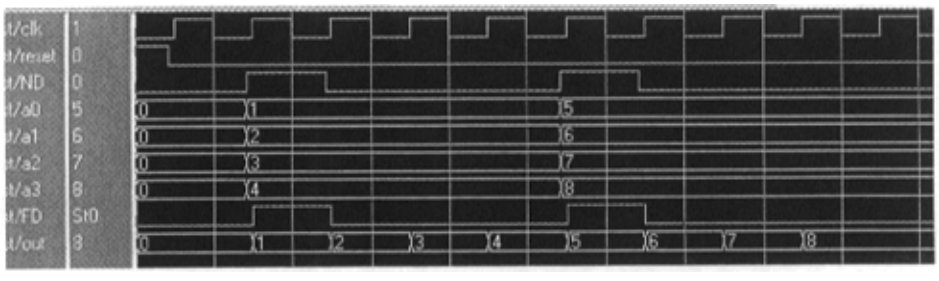

图5-14 并\串转换波形图

5.7 矩阵乘法模块的设计与实现

此模块的主要功能是实现运算 $\mathbf{HH}^H$，矩阵乘法的定义非常简单，但运算量却很大，如果不进行很好的优化处理，将造成资源浪费。

设 $\mathrm{A}=\mathbf{H}\mathbf{H}^{H}$，$\mathrm{a}_{ij}\in\mathrm{A}$，则 $\mathrm{A}^{H}=(\mathbf{H}\mathbf{H}^{H})^{H}=\mathbf{H}\mathbf{H}^{H}=\mathrm{A}$，说明 A 为 Hermite 矩阵，A 中的元素满足如下规律，$\mathrm{a}_{ij}=\overline{\mathrm{a}_{ji}}$，即 A 矩阵上三角中元素与下三角中的元素互为共轭对称复数，只需求出一边(包括主对角线)的元素即可，而另一边的元素可以靠对称性得来。

以上三角为例来做处理，则对于对角线中的元素 $a_{ii}\,(1\le i\le 4)$

$$\begin{aligned} a_{ii} &= \sum_{k=1}^{4}\left|h_{ik}\right|^{2} \\ &= \sum_{k=1}^{4}\mathrm{Re}(h_{ik})\times\mathrm{Re}(h_{ik})+\sum_{k=1}^{4}\mathrm{Im}(h_{ik})\times\mathrm{Im}(h_{ik}) \end{aligned} \tag{5.4}$$

对于对角线上方的元素 $a_{ij}\,(1\le i<j\le 4)$

$$\begin{aligned} a_{ij} &= \sum_{k=1}^{4}h_{ik}\overline{h_{jk}} \\ &= \left(\sum_{k=1}^{4}\mathrm{Re}(h_{ik})\times\mathrm{Re}(h_{jk})+\sum_{k=1}^{4}\mathrm{Im}(h_{ik})\times\mathrm{Im}(h_{jk})\right)+i\times\left(\sum_{k=1}^{4}\mathrm{Im}(h_{ik})\times\mathrm{Re}(h_{jk})-\sum_{k=1}^{4}\mathrm{Re}(h_{ik})\times\mathrm{Im}(h_{j}\right. \end{aligned} \tag{5.5}$$

即

$$\begin{aligned} \mathrm{Re}(a_{ij}) &= \left(\sum_{k=1}^{4}\mathrm{Re}(h_{ik})\times\mathrm{Re}(h_{jk})+\sum_{k=1}^{4}\mathrm{Im}(h_{ik})\times\mathrm{Im}(h_{jk})\right) \\ \mathrm{Im}(a_{ij}) &= \left(\sum_{k=1}^{4}\mathrm{Im}(h_{ik})\times\mathrm{Re}(h_{jk})-\sum_{k=1}^{4}\mathrm{Re}(h_{ik})\times\mathrm{Im}(h_{jk})\right) \end{aligned} \tag{5.6}$$

如果采用全并行流水线结构来设计，则这个矩阵乘法的实现需要 128 个乘法器。由于时钟频率为实际数据速率的 4 倍，故也可以采用循环长度为 4 的乘加器来实现。对于对角线上的元素 $a_{ii}\,(1\le i\le 4)$，需两个 MAC；对于非对角线上的元素需 4 个 MAC，MAC 内部的核心器件则为 1 个乘法器和一个累加 1 器，用这种方法来实现实际上使用的乘法器的个数仅为 32 个，为原来的四分之一，从而达到了节省资源的目的。

5.8 矩阵求逆模块的设计与实现

5.8.1 算法

本模块的主要功能是实现运算$\left[\mathbf{HH}^H+4\sigma_n^2\mathbf{I}_4\right]^{-1}$，在本设计中采用的算法为变量循环重新编号法简化方案三。对于4×4MIMO 信道，需要求逆的矩阵为 4 阶正定 Hermite 矩阵，求逆公式如下：

$$a'_{4,4}=1/a_{11} \tag{5.7}$$

$$a'_{33}=a_{44}\pm|a_{14}|^2\times(1/a_{11}) \tag{5.8}$$

$$a'_{22}=a_{33}\pm|a_{13}|^2\times(1/a_{11}) \tag{5.9}$$

$$a'_{11}=a_{22}\pm|a_{12}|^2\times(1/a_{11}) \tag{5.10}$$

$$a'_{34}=\pm\overline{a_{14}}\times(1/a_{11}) \tag{5.11}$$

$$a'_{24}=\pm\overline{a_{13}}\times(1/a_{11}) \tag{5.12}$$

$$a'_{14}=\pm\overline{a_{12}}\times(1/a_{11}) \tag{5.13}$$

$$a'_{23}=a_{34}\pm\overline{a_{13}}a_{14}\times(1/a_{11}) \tag{5.14}$$

$$a'_{13}=a_{24}\pm\overline{a_{12}}a_{14}\times(1/a_{11}) \tag{5.15}$$

$$a'_{12}=a_{23}\pm\overline{a_{12}}a_{13}\times(1/a_{11}) \tag{5.16}$$

在循环过程中，上述公式中的正负号(加减号)规律在表 5-3 中注出。

表5-3 各公式在循环过程中加减(正负) 号的选取

公式	(5.8)	(5.9)	(5.10)	(5.11)	(5.12)	(5.13)	(5.14)	(5.15)	(5.16)
第一次循环	-	-	-	+	+	+	-	-	-
第二次循环	+	-	-	-	+	+	-	-	-
第三次循环	+	+	-	-	-	+	+	-	-
第四次循环	+	+	+	-	-	-	+	+	+

5.8.2 循环结构设计

变量循环重新编号法在运算过程中需循环的次数与矩阵阶数相同，对于4×4MIMO信道，在求逆过程中应该进行4次循环。硬件设计可行的方案有如下几种：

1) 串行结构。将循环展开，用串行结构来实现，如图5-15所示

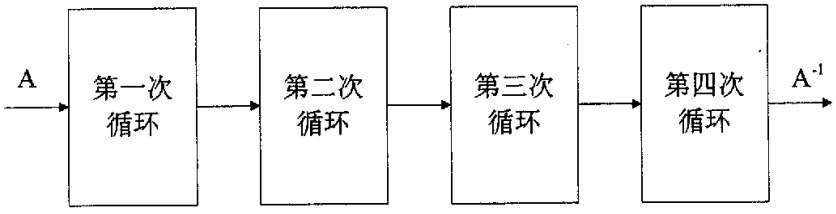

图5-15 矩阵求逆的串行处理框图

这种结构是可行的，但是由于需要4份几乎一样的运算模块，资源利用率很低。

2) 1级时钟周期循环方案。采用类似累加器模块的循环结构，即1个时钟周期完成一次循环。由于每个矩阵都有4个时钟周期可用，故在下一个矩阵到达之前已经完成了对当前矩阵的求逆运算。

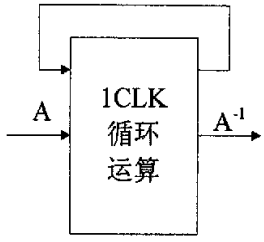

图5-16 1 级时钟完成一次循环方案

这种方案表面看来可行，实际则不能实现，因为在运算过程中存在下面这类运算 $a'_{i,j}=a_{i+1,j+1}-a_{i+1,1}a_{1,j+1}/a_{1,1}, i,j=1,2,\cdots,n-1$，太多的组合逻辑使得器件的工作时钟根本达不到 80MHz。

3) 5 级时钟循环方案。本文中的设计方案：采用 5 级时钟来完成一次循环，在模块内部有四级基本运算单元和一级输入输出控制单元，基本构架如图 5-17 所示

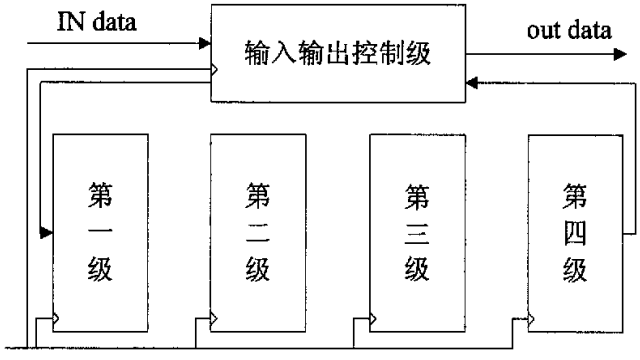

图5-17 五级时钟完成一次循环方案框图

做完一次循环需 5 个时钟周期，数据输入后延迟 20 个时钟周期出结果，示意图参看图 5-18。

各级运算所做的工作如表 5-4，第四级中的加减号与正负号由状态机来控制，状态机的基本描述见 5.8.3 节。第五级即输入输出控制，详细描述见 5.8.4 节。

表5-4 各级运算单元工作分配表

I	II	III	IV	V
查表求出 $a'_{4,4}=1/a_{11}$	寄存前级结果	寄存前级结果	寄存前级结果	输入输出控制级
寄存 a_{44} ，求出 $\lvert a_{14}\rvert^2$		$\lvert a_{14}\rvert^2*(1/a_{11})$	$a'_{33}=a_{44}\pm\lvert a_{14}\rvert^2*(1/a_{11})$	
寄存 a_{33} ，求出 $\lvert a_{13}\rvert^2$		$\lvert a_{13}\rvert^2*(1/a_{11})$	$a'_{22}=a_{33}\pm\lvert a_{13}\rvert^2*(1/a_{11})$	
寄存 a_{22} ，求出 $\lvert a_{12}\rvert^2$		$\lvert a_{12}\rvert^2*(1/a_{11})$	$a'_{11}=a_{22}\pm\lvert a_{12}\rvert^2*(1/a_{11})$	
求出 $\overline{a_{14}}$	寄存前级结果	$\overline{a_{14}}*(1/a_{11})$	$a'_{34}=\pm\overline{a_{14}}*(1/a_{11})$	
求出 $\overline{a_{13}}$	寄存前级结果	$\overline{a_{13}}*(1/a_{11})$	$a'_{24}=\pm\overline{a_{13}}*(1/a_{11})$	
求出 $\overline{a_{12}}$	寄存前级结果	$\overline{a_{12}}*(1/a_{11})$	$a'_{14}=\pm\overline{a_{12}}*(1/a_{11})$	
寄存 a_{23} ，求出 $\overline{a_{13}}a_{14}$		$\overline{a_{13}}a_{14}*(1/a_{11})$	$a'_{23}=a_{34}\pm\overline{a_{13}}a_{14}*(1/a_{11})$	
寄存 a_{24} ，求出 $\overline{a_{12}}a_{14}$		$\overline{a_{12}}a_{14}*(1/a_{11})$	$a'_{13}=a_{24}\pm\overline{a_{12}}a_{14}*(1/a_{11})$	
寄存 a_{23} ，求出 $\overline{a_{12}}a_{13}$		$\overline{a_{12}}a_{13}*(1/a_{11})$	$a'_{12}=a_{23}\pm\overline{a_{12}}a_{13}*(1/a_{11})$	

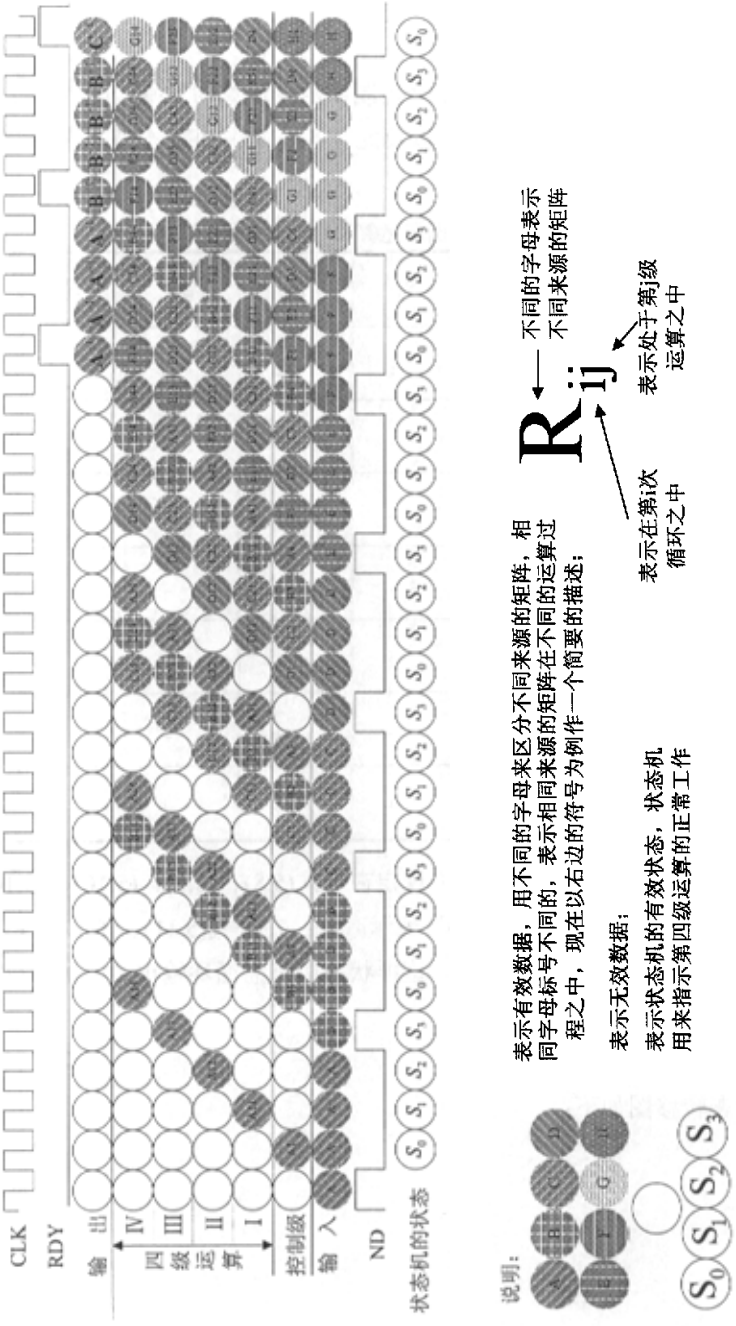

图5-18 5级时钟循环方案求逆矩阵内部循环示意图

5.8.3 状态机的设计

在这里，状态机的主要作用是用输出来控制第四级运算的加减号(正负号)。状态机的状态、第四级运算所处的循环状态以及此时各公式中加减号(正负号)的对应关系见表 5-5，参看图 5-18。

表5-5 状态机输出与第四级运算符号的对应关系

状态机状态	S_0	S_1	S_2	S_3
第四级运算所处的循环状态	1	2	3	4
公式(5.8)中的加减号	-	+	+	+
公式(5.9)(5.14)中的加减号	-	-	+	+
公式(5.10)(5.15)(5.16)中的加减号	-	-	-	+
公式(5.11)中的正负号	+	-	-	-
公式(5.12)中的正负号	+	+	-	-
公式(5.13)中的正负号	+	+	+	-

为了减少状态机的状态，在本设计中规定在公式(5.8)、(5.9)、(5.10)、(5.14)、(5.15)、(5.16)中，由“1”表示加法，“0”表示减法；在公式(5.11)、(5.12)、(5.13)里由“1”表示负号，由“0”表示正号。这样状态机只需循环输出，000，100，110，111 四个状态即可。

状态机的状态转移图如下：

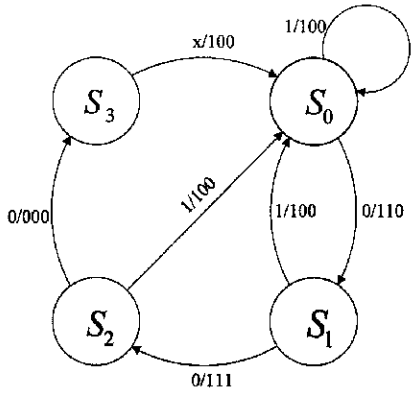

图5-19 状态转移图

5.8.4 输入输出控制单元的设计与实现

这里说的输入信号不单单是指外界输入到矩阵求逆模块的信号，也包括做完k($1 \le k \le 3$)次循环回来的信号。从图 5-18 中可以看出，只要满足时钟频率为数据流实际率的 4 倍的 4 阶正定 Hermite 阵，均可用这种 5 级时钟循环结构来实现求逆，不会造成新输入数据与循环回来数据之间的冲突。

从图 5-18 中可以看出，输出与循环的判断主要是通过每个矩阵的下角标来实现的，在实际设计中能不能采用类似的思想呢？答案是肯定的。在这里定义两个变量来充当图中下角标的功能，它们分别为 num 以及把它延迟 4 级时钟后得到的 reg_num。当 ND 到达时，令 num 为 1，新数据输入到运算模块中，标志着这组数据开始第一次循环运算。4 个时钟后，reg_num 为 1，标志着第一次循环运算完成。当 reg_num=4 时，表示 4 次循环全部完成，运算后的数据输出到矩阵求逆模块外，其他状态参看表 5-6。

表5-6 输入输出控制

ND	0			1	
reg_num	0	4	1、2、3	0	4
num	0	0	reg_num+1	1	
RDY	0	1	0	0	1
输出	锁存	输出新数据	锁存	锁存	输出新数据
运算输入端	0		输入循环结果	输出新数据	
状态说明	未工作	输入停止	循环过程中	起步	4 次循环完毕，结果输出，新数据输入。

5.8.5 仿真波形

图 5-20、图 5-21 是 Model sim 仿真出来的波形(它们是相连的)。为了容易辨认，这里选取的矩阵全是对角阵，主要是为了方便检查时序，并不是为了验证其功能。对于正定对角阵而言，每次循环完毕后的矩阵，及其最终输出的矩阵均为对角阵，故没有显示非对角线上的元素。用 b 来命名的元素表示第一级运算的输入端。

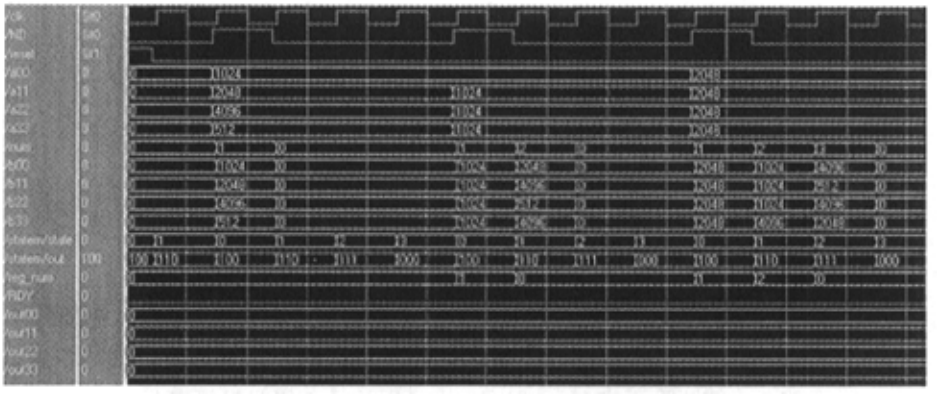

图5-20 矩阵求逆模块仿真波形图-1

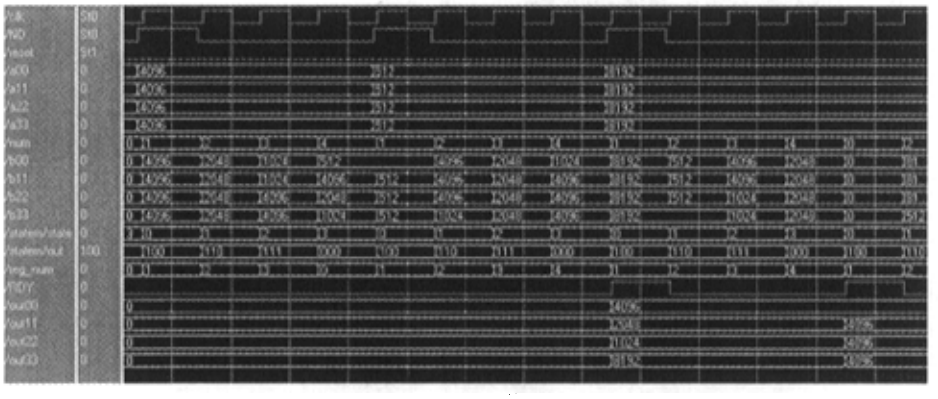

图5-21 矩阵求逆模块仿真波形图-2

从波形上可以看出状态机及其输入输出控制端与设计思想完全一致。这里的输入输出结果均为定点数，小数位数为 11 位。当输入的矩阵为

$$A=\begin{bmatrix} 1/2 & & & \\ & 1 & & \\ & & 2 & \\ & & & 1/4 \end{bmatrix}$$ 时，输出矩阵为 $$A^{-1}=\begin{bmatrix} 2 & & & \\ & 1 & & \\ & & 1/2 & \\ & & & 4 \end{bmatrix}$$

5.9 幅度自动调整模块的设计

MIMO 检测是一个定点器件，定点的运算能力受限于位宽，当输入数据幅度很大时，在运算过程中将遭到限幅，从而造成器件性能下降。当输入数据幅度很小时，将造成$\mathbf{HH}^H+4\sigma_n^2$很小，而求逆的结果却很大，依然造成了限幅，性能有了很大的损失。因此，如何对输入数据幅度进行自动处理是本节的主要任务。

5.9.1 算法原理

本论文的设计思路是在 MIMO 检测模块前加一个幅度自动调整模块，当数据幅度大小不符合要求时，对其进行幅度调整，使调整后的数据能在 MIMO 检测器中进行很好的处理。幅度自动调整模块与 MIMO 检测模块的连接关系见图 5-22。

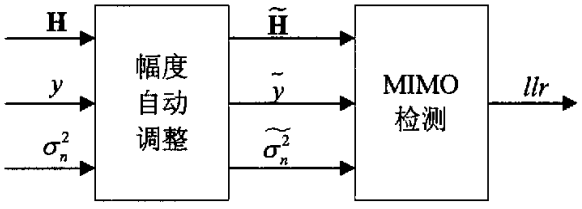

图5-22 幅度自动调整模块与 MIMO 检测模块的连接关系

在 MMSE-LLR 检测算法中，如果将输入信号按公式(5.17)规律变换，不影响检测结果。

$$\begin{cases} \mathrm{H}\xrightarrow{\lambda}\lambda\mathrm{H} \\ \mathrm{y}\xrightarrow{\lambda}\lambda\mathrm{y} \\ \sigma_{\mathrm{n}}^2\xrightarrow{\lambda^2}\lambda^2\sigma_{\mathrm{n}}^2 \end{cases} \qquad \lambda>0 \tag{5.17}$$

证明：设

$$\begin{cases} \mathrm{H}'=\lambda\mathrm{H} \\ \mathrm{y}'=\lambda\mathrm{y} \\ (\sigma_{\mathrm{n}}^2)'=\lambda^2\sigma_{\mathrm{n}}^2 \end{cases} \qquad \lambda>0 \tag{5.18}$$

则，

$$\begin{aligned}\mathbf{w}'_i &= \left[\mathbf{H}'(\mathbf{H}^H)'+4(\sigma_n^2)'\mathbf{I}_4\right]^{-1}\mathbf{h}'_i = \left[\lambda^2\mathbf{H}\mathbf{H}^H+4\lambda^2\sigma_n^2\mathbf{I}_4\right]^{-1}\lambda\mathbf{h}_i \\ &= \frac{1}{\lambda^2}\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]^{-1}\lambda\mathbf{h}_i = \frac{1}{\lambda}\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]^{-1}\mathbf{h}_i \\ &= \frac{1}{\lambda}\mathbf{w}_i\end{aligned} \tag{5.19}$$

$$\hat{s}'_i = \mathbf{w}'^{H}_i\mathbf{y}'_i = \frac{1}{\lambda}\mathbf{w}_i^H\lambda\mathbf{y}_i = \hat{s}_i \tag{5.20}$$

$$\mu'_i = \mathbf{w}'^{H}_i\mathbf{h}'_i = \frac{1}{\lambda}\mathbf{w}_i^H\lambda\mathbf{h}_i = \mu_i \tag{5.21}$$

$$(\sigma_{\eta_i}^2)' = \frac{1}{4}\left(\mu'_i - \mu'^{2}_i\right) = \sigma_{\eta_i}^2 \tag{5.22}$$

$$\begin{aligned}LLR'_{s_i}(k) &= \frac{\min\limits_{c_k=0}\left(\left|\hat{s}'_i-\mu'_i a_n\right|^2\right)-\min\limits_{c_k=1}\left(\left|\hat{s}'_i-\mu'_i a_n\right|^2\right)}{(\sigma_{\eta_i}^2)'} \\ &= LLR_{s_i}(k)\end{aligned} \tag{5.23}$$

经浮点验证，让λ按下述规律变化检测出的结果与变换前完全相同，

$$\lambda(k) = a\sin(\omega k+\varphi)+\mathrm{b}>0 \tag{5.24}$$

故可以基于这样一种调整思想，当输入数据幅度很大时，找到一个合适的$\lambda(0<\lambda<1)$，用λ做衰减因子，将数据调整到可以处理的范围之内；当输入数据幅度很小时，找一个合适的$\lambda(1<\lambda)$，用λ做放大因子，将数据调整到适当的范围之内，使得求逆之后的数据尽可能不溢出。

5.9.2 基于信道信息 H 的精确调整

调整方法如图 5-23 所示：

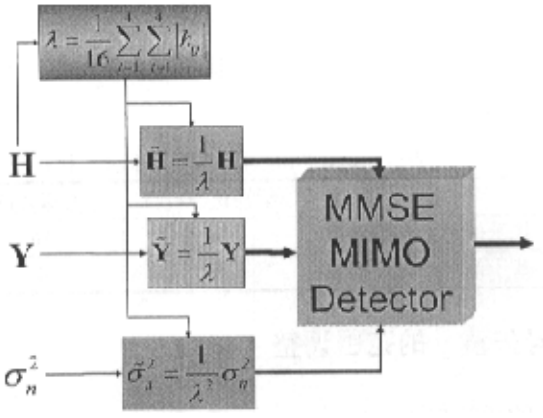

图5-23 基于信道信息调整示意图

这种调整方法的核心思想就是将信道矩阵 H 中元素的平均幅度调整为 1。

5.9.3 基于接收信号 Y 的精确调整

调整方法如图 5-24 所示

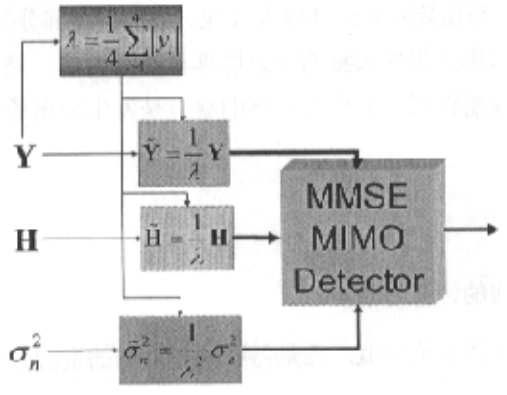

图5-24 基于接收信号调整示意图

这种调整方法的核心思想就是将信号矢量 y 中元素的平均幅度调整为 1。

5.9.4 两种方案的比较

在 MIMO 检测模块中最容易发生溢出的是$\left[\mathbf{HH}^H + 4\sigma_n^2\mathbf{I}_4\right]^{-1}$运算，采用基于信道信息的精确调整方案最有利于控制这一步的溢出。但这种方案与基于信号的精确调整方案相比，缺点是运算量太大。它们的复杂度对比见表 5-7。由信道传输

模(见 2.1.2 节)可以看出 $\mathbf{H}$、y 的幅度存在线形关系，因此，后一种方案对 MIMO 检测内的运算也能起到有效的溢出抑制。但还应该看出后一种方案的运算量依然很大，因此，提出了下面的调整方案。

表5-7 两种精确调整方案复杂度对比

	乘法运算(次)	开方运算(次)
基于信道的幅度调整	74	16
基于信号的幅度调整	50	4

5.9.5 基于信号矢量 y 的范围调整

调整因子 λ 的求法如下，

$$\lambda=\frac{1}{8}\left(\sum_{i=1}^{4}\left|\mathrm{Re}(y_i)\right|+\sum_{i=1}^{4}\left|\mathrm{Im}(y_i)\right|\right) \tag{5.25}$$

然后根据调整因子 λ 的大小来对所有输入信号进行移位操作(相当于乘上 2^n)，使得调整后 y 中的 8 个数据(实虚部分开)绝对值的平均幅度在一个特定范围，经过仿真验证，将这个范围定在0.5～1较为合理。采用实虚部分开，求绝对值平均的优点是省去了复数求模过程中大量的平方运算与开方运算。采用本方案所仅需 7 个加法器和一些寄存器资源，但这种方案的缺点是发生溢出的概率比上面两种方案稍稍有些增加。

5.9.6 硬件设计

5.9.6.1 绝对值平均的计算方法

对 8 个数据分别求绝对值，还是需要消耗大量的资源。在实际设计中没有采用加法器，而是采用加减法器，加减器标志位用输入数据的符号位做同或运算得来。实际的处理就变成了同符号数相加，异符号数相减，结果为两个数据的绝对值之和或绝对值之和的相反数。这样求出来的结果也就变成了

$$\pm\left(\sum_{i=1}^{4}\left|\mathrm{Re}(y_i)\right|+\sum_{i=1}^{4}\left|\mathrm{Im}(y_i)\right|\right) \tag{5.26}$$

乘1/8运算是一个移位操作，实际上这个移位操做并不需要真正进行，数据的大小

在 FPGA 芯片的内部并没有改变，只是按运算后的格式理解就可以了(见图 5-25)。

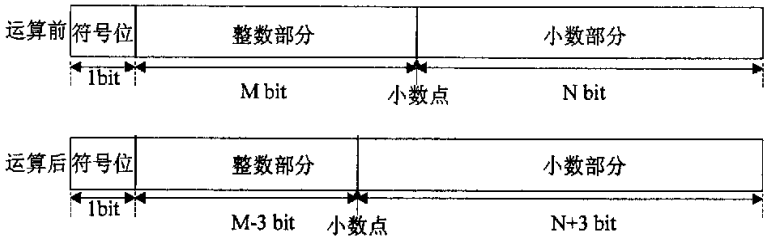

图5-25 移位运算前后数据格式

运算完毕后实际的到的 λ 为

$$\lambda = \pm\left(\sum_{i=1}^{4}\left|\mathrm{Re}(y_i)\right| + \sum_{i=1}^{4}\left|\mathrm{Im}(y_i)\right|\right) \tag{5.27}$$

5.9.6.2 调整的实现方法

假设算出来的 λ 的定点数格式为一个符号位，4 个整数位，11 个小数为，那么移位操作按表 5-8 进行。这里说的调整，指的是对 H 和 y 的调整，σ_n^2 的调整幅度为它们平方，所移位数为它们的两倍。

表5-8 移位控制列表

说明		整数部分				小数部分				调整因子	调整
Bit 位	18	17	16	15	14	13	12	11	10		
Bit 位的值	0	1	x							1/16	左 4
		0	1	x						1/8	左 3
			0	1	x					1/4	左 2
				0	1	x				1/2	左 1
					0	1	x			1	NO
						0	1	x		2	右 1
							0	1	x	4	右 2
	1	0	x							1/16	左 4
		1	0	x						1/8	左 3
			1	0	x					1/4	左 2
				1	0	x				1/2	左 1
					1	0	x			1	NO
						1	0	x		2	右 1
							1	0	x	4	右 2

5.10 设计验证与性能分析

5.10.1 设计验证

在 MIMO 检测设计过程中需要对每一个子模块都进行功能仿真，以便早期发现设计中潜在的问题，及时修改源程序，使设计的正确性得到确认。在设计完成后，还要对系统进行功能仿真，此次仿真的主要目的是测试 MIMO 检测性能的优劣。其仿真过程是这样的，测试数据全是从 COSSAP 链路截取的浮点数据，转换为定点数据后作为 FPGA 的测试向量。然后用仿真软件 ModelSim 对 MIMO 检测模块进行行为仿真，将仿真的结果转化为浮点数后重新输入到 COSSAP 链路统计误码率。图 5-26 是硬件仿真结果与 COSSAP 链路的性能对比曲线。由于仿真的速度通常很慢，功能仿真所耗时间与实际运行时的处理时间经常是在 1000:1 以上，并且逻辑越复杂，仿真速度越慢。故本曲线中的每个信噪比取的数据为 100000 组，这样的数据不足以进行交织，故去掉了交织的环节。加交织后的性能比无交织的性能会有很大的提升。

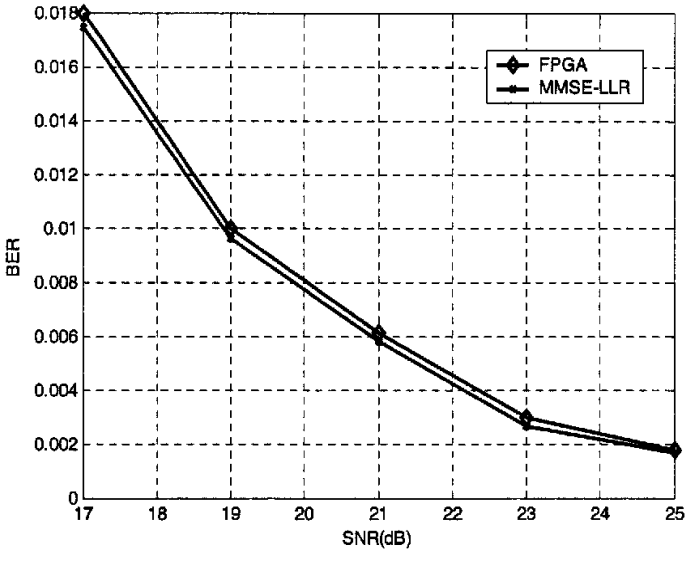

图5-26 FPGA 仿真结果与浮点仿真结果对比曲线

从图 5-26 可以看出，设计还是相当成功的，FPGA 硬件仿真的结果与浮点仿真结果的相差不大。

行为仿真完毕后需对系统进行布线后仿真，以确保设计功能与 FPGA 实际运行情况一致。由于布局布线的时延信息反标到设计网表中，所以布线后仿真最准确，能较好地反映芯片的实际情况。MIMO 检测的布线后仿真的结果与行为仿真完全一致。在完成各种仿真后便可进行下载调试了。

5.10.2 性能分析

5.10.2.1 MIMO 检测实物图

本设计采用 1 片 Xilinx 的 Virtex-II Pro 系列产品中的 XC2VP70，引脚封装采用 1704，速度采用-5。在同一块 PCB 板上还用另外两块相同的芯片，一块用来进行信道估计处理，一块拿来备用(在实际调试过程中用来存储信号矢量 y)，这些芯片在板内的排列方式如图 5-27 所示。

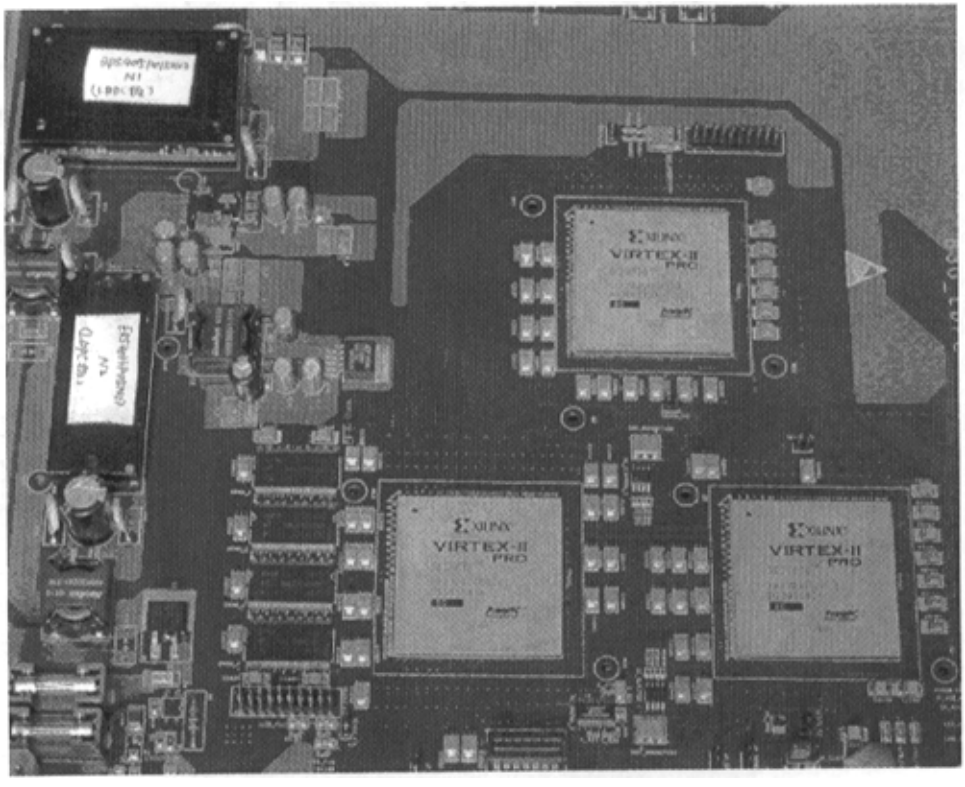

图5-27 MIMO 检测电路板

5.10.2.2 资源占用

MIMO 检测模块实际占用的芯片为一片，资源占用情况如表 5-9 所示

表5-9 MIMO 检测的资源占用情况

名称	使用数量	总数量	占用百分比
逻辑片 Slice	17203	33088	51%
Slice 寄存器	25400	66176	38%
4 输入查找表(LUT)	15293	66176	23%
输入输出端口 IOB	748	966	75%
Block RAM	130	328	31%
18×18 乘法器	184	328	56%
全局时钟 GCLK	1	16	6%

5.10.2.3 运行速度

B3G TDD 下行链路中 MIMO 检测板的全局时钟为 40MHz，数据的实际速率为 20MHz。MIMO 检测设计的时钟频率为 80MHz，即 4 个时钟周期处理一组数据。MIMO 检测部分的综合报告显示，最小周期为 8.976ns，即可达到的最高工作频率为 111.414 MHz，满足系统设计需求。

5.11 小结

本章中首先介绍了系统需求、基本框架、总体模块划分，然后介绍了在系统中采用的器件分时复用的思想，接着对矩阵求逆模块和幅度自动调整模块的实现方案进行了详细的介绍。最后对硬件的性能进行了分析，以此来证明前面做的简化与设计都是行之有效的。

第六章 总结

本论文在MIMO检测算法方面的贡献有：

- ✧ 证明了在无循环的情况下MMSE SIC检测算法实际上等同于MMSE检测算法；
- ✧ 分析了MMSE滤波因子的特点，引入并简化了变量循环重新编号矩阵求逆算法，使其FPGA设计的难度大大降低；
- ✧ 如果MIMO检测输出的结果为比特似然比(BLR)，那么将存在大量的非线性运算。在本文中改输出结果为对数似然比(LLR)，采用对数域的近似处理，避免的非线性运算，节省了大量的资源，降低了实现难度。

在FPGA实现方面的贡献有：

- ✧ 根据整个下行链路的系统需求巧妙地安排时钟频率与器件的处理方式，达到了最大可能的资源复用；
- ✧ 在MMSE滤波器所涉及的矩阵求逆模块的设计上，结合在MIMO检测内部4个时钟处理一组数据的时钟要求，以及变量循环重新编号法对4阶矩阵需循环4次的特点，巧妙地采用5级时钟循环处理方式，既达到了输入数据与循环数据不发生冲突，又达到了资源最大限度的合理利用；
- ✧ 幅度自动调整算法的研究与设计，用很少的资源实现了数据幅度的自动调整，拓宽了MIMO检测所能处理的数据范围。

下一步的工作：

从2.3.2.7节性能曲线对比(见图2-9)可以看出，基于循环的MMSE SIC检测算法的性能的确比无循环的MMSE-LLR检测算法好很多。因此，下一步的工作是开发等效循环结构(见2.3.2.1迭代的等效处理一节)中的另外MIMO检测模块，在新的模块中由于星座点不是等概分布，因此要加入软干扰抵消环节。如何降低新MIMO检测模块的复杂度，如何用尽可能少的FPGA资源完成高性能的MIMO检测处理也是下一步的任务。

致谢

首先，我衷心感谢我的导师李少谦教授在近三年的时间里对我学业上孜孜不倦的教诲和生活上无微不至的关怀。在我攻读硕士学位期间，李老师不仅为我创造了良好的学习和工作环境，而且在学术上悉心指导，思想上耐心启迪，李老师渊博的学识、豁达的风度、严谨的治学，以及对科学前沿的敏锐洞察力，都给我留下了深刻的印象。这一切都切实的影响着我生活、学习、工作的各个方面，将使我在自己今后的很长一段人生历程里受益匪浅。真的很感谢！

感谢我的直接指导老师—刘皓老师，在两年多的时间里，给予了我无数的帮助和指导，指导我克服一个又一个的科研困难。我还要感谢指导过我的胡剑浩教授和王军博士以及教研室其他指导帮助过我的老师。

感谢我的师兄黄海洋、赵萧宇同学，感谢他们在项目初期做的大量基础性工作。

感谢项目组中一直与我并肩作战的武文杰、张斌、欧思斯同学，感谢这个团队内所有的同学，这段时间大家互相帮助，互相鼓励，一起克服困难，取得成功，令我终身难忘。

感谢我的师弟马俊、杨苗，正是和他们一道才完成了 MIMO 检测的 FPGA 设计工作。

感谢我的好友彭小勇、谢丹、徐晓军、郑翔、郑恒峰、刘学、余涛等，在论文撰写阶段给我提供了大量的帮助，在此表示由衷的感谢。

最后，我要感谢我的家人。感谢他们一直以来对我的关爱和支持。

参考文献

[1] 佟学俭, 罗涛. OFDM 移动通信技术原理与应用. 人民邮电出版社，2003.6

[2] 何琳琳, 杨大成. 移动通信. 4G 移动通信系统的主要特点和关键技术, 2004.10

[3] 刘伟, 丁志杰. 4G 移动通信系统的研究进展与关键技术. 中国数据通信, 2004.2

[4] 尤小虎. 我国未来移动通信研究发展展望. 通讯世界, 2002.12

[5] 文雪, 赵宏志, 王军. MIMO 检测算法说明 Version 0.0.3,电子科技大学通信抗干扰技术国家级重点实验室, 2004.10

[6] Golden G D, Foschini G J, Valenzuela R A, etal. Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture, Electronics Letters, 1999;35(1)

[7] 北京邮电大学, 电子科技大学, 华中科技大学, 上海交通大学. B3G－TDD 方案概要设计和需求分析, Ver 8.0, 2004.1.18

[8] Melanie Witzke, Stephan Baro, Frank Schreckenbach, Joachim Hagenauer. Iterative Detection of MIMO Signals with Linear Detectors, Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on , Volume: 1, Pages:289 – 293, 3-6 Nov. 2002

[9] Takumi ITO, Xiaodong WANG, Yoshikazu KAKURA, Performance Comparison of MF and MMSE Combined Iterative Interference Canceller and V-BLAST techniques in MIMO/OFDM Systems, VTC-2003 Fall, vol. 1, pp.488-492, Oct. 2003

[10] Ye (Geoffery) Li and Leonard J.Cimini, Jr. Bounds on Interchannel Interference of OFDM in Time-Varying Impairments, IEEE Transactions On Communications, Vol. 49, No. 3, pp. 401-404, March 2001

[11] 赵潇宇等. 64QAM 信号的软解调算法及其定点实现. 中国西部青年通信学术会议论文集. P293~P297, UESTC Chengdu, 2004.12

[12] 徐士良. 常用算法程序集. 清华大学出版社. 2004.11

[13] 龚莹. 矩阵求逆算法及其在 TMS320VC33 上的实现. 机床与液压, 2004.9

[14] 王诚等. FPGA/CPLD 设计工具—Xilinx ISE 5.x 使用详解. 人民邮电出版社,2003.6

[15] 王金明. Verilog HDL 程序设计教程. 人民邮电出版社, 2004

[16] XILINX, Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet, DS083 (v4.5) October 10, 2005

[17] 佟远祯等. 计算机组成原理. 电子工业出版社, 1996

附录 A

矩阵$\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]$是正定 Hermite 矩阵，证明如下

对称性：

$$\begin{aligned}\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]^H&=\left[\mathbf{H}\mathbf{H}^H\right]^H+\left[4\sigma_n^2\mathbf{I}_4\right]^H\\&=\left[\mathbf{H}^H\right]^H\left[\mathbf{H}\right]^H+4\sigma_n^2\left[\mathbf{I}_4\right]^H\\&=\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\end{aligned}$$

正定性：

设$\alpha\neq 0$，则

$$\begin{aligned}\alpha^H\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]\alpha&=\alpha^H\left[\mathbf{H}\mathbf{H}^H\right]\alpha+\alpha^H\left[4\sigma_n^2\mathbf{I}_4\right]\alpha\\&=\left[\mathbf{H}^H\alpha\right]^H\left[\mathbf{H}^H\alpha\right]+4\sigma_n^2\alpha^H\alpha\\&=\left\|\mathbf{H}^H\alpha\right\|^2+4\sigma_n^2\left\|\alpha\right\|^2\\&>0\end{aligned}$$

故，$\left[\mathbf{H}\mathbf{H}^H+4\sigma_n^2\mathbf{I}_4\right]$是正定 Hermite 矩阵

附录 B

MMSE SIC 检测算法中公式(5.28)的推导过程

$$\because \quad \ln\left(\sum_{k}\eta_{i,n}\right)\approx\max_{k}\left(\ln\eta_{i,n}\right)$$

$$\therefore \quad LLR_{s_i}(l)=\ln\left(\mathrm{BLR}_{s_i}(l)\right)$$

$$=\ln\left(\frac{\sum\limits_{n=0,c_l=1}^{15}\eta_{i,n}}{\sum\limits_{n=0,c_l=0}^{15}\eta_{i,n}}\right)$$

$$=\ln\left(\sum_{n=0,\,c_l=1}^{15}\eta_{i,n}\right)-\ln\left(\sum_{n=0,\,c_l=0}^{15}\eta_{i,n}\right)$$

$$\approx\max_{c_l=1}\left[\ln\left(\eta_{i,n}\right)\right]-\max_{c_l=0}\left[\ln\left(\eta_{i,n}\right)\right]$$

$$=\max_{c_l=1}\left[\ln\left(\frac{\exp\left\{\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_n)\right|^2\right\}/\sigma_{\eta_i}^2\right\}}{\sum\limits_{j=0}^{15}\exp\left\{\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_j)\right|^2\right\}/\sigma_{\eta_i}^2\right\}}\right)\right]$$

$$-\max_{c_l=0}\left[\ln\left(\frac{\exp\left\{\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_n)\right|^2\right\}/\sigma_{\eta_i}^2\right\}}{\sum\limits_{j=0}^{15}\exp\left\{\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_j)\right|^2\right\}/\sigma_{\eta_i}^2\right\}}\right)\right]$$

$$=\max_{c_l=1}\left[\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_n)\right|^2\right\}/\sigma_{\eta_i}^2\right]$$

$$-\max_{c_l=0}\left[\left\{\min\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_m)\right|^2,a_m\in\mathbf{A}\right]-\left|\mathbf{w}_i^{\mathrm{H}}\cdot(\mathbf{y}_i-\mathbf{h}_i a_n)\right|^2\right\}/\sigma_{\eta_i}^2\right]$$

$$=\min_{c_l=0}\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{y}_i-\mu_i a_n\right|^2/\sigma_{\eta_i}^2\right]-\min_{c_l=1}\left[\left|\mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{y}_i-\mu_i a_n\right|^2/\sigma_{\eta_i}^2\right]$$

$$=\frac{\min\limits_{c_l=0}\left(\left|\mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{y}_i-\mu_i a_n\right|^2\right)-\min\limits_{c_l=1}\left(\left|\mathbf{w}_i^{\mathrm{H}}\cdot\mathbf{y}_i-\mu_i a_n\right|^2\right)}{\sigma_{\eta_i}^2}$$

注：c_l 为发送符号 s_i 的第 l 个比特位

附录 C

在使用变量循环重新编号法求逆矩阵的过程中，会得到这样的结论：在第k($1 \le k \le n$)次循环后，主对角线上的元素$a_{i,i}(1 \le i \le n)$永远为正实数，关于对角线对称的两个复数($a_{i,j}$与$a_{j,i}$)要么互为共轭复数，要么互为负共轭复数(即一个为另一个共轭复数的相反数)，如果对矩阵进行分块为

$$\begin{matrix}\begin{bmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix} & \begin{matrix}n\text{-}k\text{行}\\ k\text{行}\end{matrix}\\ n-k\text{列}\quad k\text{列} & \end{matrix}$$

则可以证明：

1) 对于任何 k，$\mathbf{A}_{11}$ 和 $\mathbf{A}_{22}$ 都是正定 Hermite 阵；

2) $\mathbf{A}_{21} = -\mathbf{A}_{12}^{H}$。

证明过程如下：

为了是证明过程中的变量不至于混淆，我们采用给变量加上表的方式来区分变量，如$a^{(k)}$表示第k次循环运算完成后的结果。

结论(1)的证明

对 A 矩阵、第 k 次循环后的矩阵 $\mathbf{A}^{(k)}$，以及 $\mathbf{A}^{-1}$ 分别按如下方式进行分块

$$\begin{matrix}\begin{bmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix} & \begin{matrix}k\text{行}\\ n\text{-}k\text{行}\end{matrix}\\ k\text{列}\quad n-k\text{列} & \end{matrix},\quad \begin{matrix}\begin{bmatrix}\mathbf{A}_{11}^{(k)} & \mathbf{A}_{12}^{(k)}\\ \mathbf{A}_{21}^{(k)} & \mathbf{A}_{22}^{(k)}\end{bmatrix} & \begin{matrix}n\text{-}k\text{行}\\ k\text{行}\end{matrix}\\ n-k\text{列}\quad k\text{列} & \end{matrix},\quad \begin{matrix}\begin{bmatrix}\mathbf{A}_{11}^{'} & \mathbf{A}_{12}^{'}\\ \mathbf{A}_{21}^{'} & \mathbf{A}_{22}^{'}\end{bmatrix} & \begin{matrix}k\text{行}\\ n\text{-}k\text{行}\end{matrix}\\ k\text{列}\quad n-k\text{列} & \end{matrix}$$

由 3.4.2 节的推导过程可以看出，A 、$\mathbf{A}^{(k)}$、$\mathbf{A}^{-1}$对应的线性方程分别为

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \cdots \qquad\quad \cdots \qquad\quad \cdots \\ y_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{cases} \tag{1}$$

$$\begin{cases} y_{k+1} = a_{1,1}^{(k)} x_{k+1} + \cdots + a_{1,n-k}^{(k)} x_n + a_{1,n-k+1}^{(k)} y_1 + \cdots + a_{1,n}^{(k)} y_k \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ y_n = a_{n-k,1}^{(k)} x_{k+1} + \cdots + a_{n-k,n-k}^{(k)} x_n + a_{n-k,n-k+1}^{(k)} y_1 + \cdots + a_{n-k,n}^{(k)} y_k \\ x_1 = a_{n-k+1,1}^{(k)} x_{k+1} + \cdots + a_{n-k+1,n-k}^{(k)} x_n + a_{n-k+1,n-k+1}^{(k)} y_1 + \cdots + a_{n-k+1,n}^{(k)} y_k \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_k = a_{n,1}^{(k)} x_{k+1} + \cdots + a_{n,n-k}^{(k)} x_n + a_{n,n-k+1}^{(k)} y_1 + \cdots + a_{n,n}^{(k)} y_k \end{cases} \tag{2}$$

$$\begin{cases} x_1 = a_{11}^{'} y_1 + a_{12}^{'} y_2 + \cdots + a_{1n}^{'} y_n \\ x_2 = a_{21}^{'} y_1 + a_{22}^{'} y_2 + \cdots + a_{2n}^{'} y_n \\ \cdots \quad \cdots \quad \cdots \\ x_n = a_{n1}^{'} y_1 + a_{n2}^{'} y_2 + \cdots + a_{nn}^{'} y_n \end{cases} \tag{3}$$

如果令 $x_{k+1} = x_{k+2} = \cdots = x_n = 0$，则可由方程组(1)得

$\overline{y_k} = \mathbf{A}_{11} \overline{x_k}$

有方程组(2)得

$\overline{x_k} = \mathbf{A}_{22}^{(k)} \overline{y_k}$

故 $\mathbf{A}_{22}^{(k)} = \mathbf{A}_{11}^{-1}$

在这里，$\overline{x_k} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_k \end{pmatrix}$，$\overline{y_k} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_k \end{pmatrix}$

如果令 $y_1 = y_2 = \cdots = y_k = 0$，则可由方程组(2)得

$\overline{y_{n-k}} = \mathbf{A}_{11}^{(k)} \overline{x_{n-k}}$

由方程组(3)得

$\overline{x_{n-k}} = \mathbf{A}_{22}^{'} \overline{y_{n-k}}$

故 $\mathbf{A}_{11}^{(k)}=\left(\mathbf{A}_{22}^{'}\right)^{-1}$

在这里 $\overline{x_{n-k}}=\begin{pmatrix}x_{n-k+1}\\x_{n-k+2}\\\cdots\\x_n\end{pmatrix}$，$\overline{y_{n-k}}=\begin{pmatrix}y_{n-k+1}\\y_{n-k+2}\\\cdots\\y_n\end{pmatrix}$

因此，跟据正定 Hermite 的性质可知 $\mathbf{A}_{11}^{(k)}$ 和 $\mathbf{A}_{22}^{(k)}$ 都是正定 Hermite 阵。

结论(2)的证明

结论(2)可以利用数学归纳法来证明

1> 验证第一次循环后结论成立

第一次循环后的分块如下

$$\begin{bmatrix}\mathbf{A}_{11}^{(1)} & \mathbf{A}_{12}^{(1)}\\ \mathbf{A}_{21}^{(1)} & \mathbf{A}_{22}^{(1)}\end{bmatrix}\begin{matrix}\text{n-1行}\\ \text{1行}\end{matrix}$$
$$n\text{-1列}\quad 1\text{列}$$

需要验证的是 $\mathbf{A}_{12}^{(1)}=-\overline{\mathbf{A}_{21}^{(1)}}$，$\mathbf{A}_{12}^{(1)}$ 中的元素为 $a_{i-1,n}^{(1)}(j=2,3,\cdots,n)$

由公式可知 $a_{i-1,n}^{(1)}=a_{i1}/a_{11}=\overline{a_{1i}}/\overline{a_{11}}=-\overline{a_{n,i-1}^{(1)}}$，故 $\mathbf{A}_{12}^{(1)}=-\overline{\mathbf{A}_{21}^{(1)}}$

2> 假设第 $k\,(1\le k<n)$ 次循环后结论成立

即对于如图所示的分块，

$$\begin{bmatrix}\mathbf{A}_{11}^{(k)} & \mathbf{A}_{12}^{(k)}\\ \mathbf{A}_{21}^{(k)} & \mathbf{A}_{22}^{(k)}\end{bmatrix}\begin{matrix}n\text{-}k\text{行}\\ k\text{行}\end{matrix}$$
$$n\text{-}k\text{列}\quad k\text{列}$$

$\mathbf{A}_{12}^{(k)}=-\overline{\mathbf{A}_{21}^{(k)}}$ 成立

3> 推导第 $k+1$ 次循环后结论成立

第 $k+1$ 次后的分块模型为

$$\begin{bmatrix}\mathbf{A}_{11}^{(k+1)} & \mathbf{A}_{12}^{(k+1)}\\ \mathbf{A}_{21}^{(k+1)} & \mathbf{A}_{22}^{(k+1)}\end{bmatrix}\begin{matrix}n\text{-}k\text{-1行}\\ k+1\text{行}\end{matrix}$$
$$n\text{-}k\text{-1列}\quad k+1\text{列}$$

在此将 $\mathbf{A}_{12}^{(k+1)}$ 划分为两个分区来分别证明

第一分区的元素为，$a_{i-1,n}^{(k+1)}(2\le i\le n-k)$，那么，对于这一分区的元素

$$a_{i-1,n}^{(k+1)}=a_{i1}^{(k)}/a_{11}^{(k)}=\overline{a_{1i}^{(k)}}\Big/\overline{a_{11}^{(k)}}=-\overline{a_{n,i-1}^{(k+1)}}$$

第二分区的元素为，$a_{i-1,j-1}^{(k+1)}(2\le i\le n-k,\ n-k+1\le j\le n)$，那么，对于这一分区的元素

$$\begin{aligned}a_{i-1,j-1}^{(k+1)}&=a_{ij}^{(k)}-a_{i1}^{(k)}a_{1j}^{(k)}/a_{11}^{(k)}\\&=-\overline{a_{ji}^{(k)}}-\left(\overline{a_{1i}^{(k)}}\right)\left(-\overline{a_{j1}^{(k)}}\right)\Big/\left(\overline{a_{11}^{(k)}}\right)\\&=-\overline{a_{j-1,i-1}^{(k+1)}}\end{aligned}$$

故结论(2)成立

攻读硕士学位期间的研究成果

参加的科研项目有：

国家重点 863 项目(Beyond 3G)："新一代蜂窝移动通信系统无线传输链路技术研究(第三期)"。负责下行链路中 MIMO 检测部分的研究与基于 FPGA 的硬件实现工作，已经基本完成。此项技术的性能将在 2006 年 6 月进行全方面的测试。

发表论文：

[1] 郑翔，王灵光，刘皓，李少谦. 基于对数域的高速 LOG-FFT 的实现及改进. 2004 中国西部青年通信学术会议. 2004 年 11 月

[2] 王灵光，刘皓，王军，李少谦. MIMO 检测算法研究与实现. 2005 中国西部青年通信学术会议. 2005 年 11 月

申请专利：

[1] 一种 MIMO 通信系统中最小均方误差滤波算法，申请号：200510021655

个人简历

王灵光，男，生于1979年11月15日。1999年9月至2003年7月，就读于电子科技大学通信与信息工程学院通信工程专业，获工学学士学位。2003年9月至2006年4月，电子科技大学攻读通信与信息系统专业硕士学位，在近几年内主要从事新一代移动通信中MIMO检测技术的研究与基于FPGA实现。

在攻读硕士学位期间，获得校研究生二等奖学金。