

分类号: _____
UDC: _____

密级: _____
编号: _____

工学硕士学位论文

LDPC 码译码算法的 FPGA 设计与实现

硕士研究生 : 李加洪

指导教师 : 赵旦峰 教授

学位级别 : 工学硕士

学科、专业 : 通信与信息系统

所在单位 : 哈尔滨工程大学

论文提交日期: 2010 年 3 月

论文答辩日期: 2010 年 3 月

学位授予单位: 哈尔滨工程大学



Y1807875

Classified Index:

U.D.C:

A Dissertation for the Degree of M. Eng

Design and Implement of LDPC Decoding Algorithm Based on FPGA

Candidate: Li Jiahong

Supervisor: Professor Zhao Danfeng

Academic Degree Applied for: Master of Engineering

Specialty: Communication and Information Systems

Date of Submission: March, 2010

Date of Oral Examination: March, 2010

University: Harbin Engineering University

哈尔滨工程大学 学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。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。

作者（签字）：

日期：

李如洪
2010年3月17日

哈尔滨工程大学 学位论文授权使用声明

本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文（☒在授予学位后即可 ☐在授予学位12个月后 ☐解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）：

日期：

李如洪
2010年3月17日

导师（签字）：

2010年3月17日

摘 要

论文以 LDPC 码译码器的 FPGA 实现为研究对象, 主要研究校验矩阵的构造、码长、译码算法、译码迭代次数和译码数据量化方式等影响译码性能的各方面因素。首先对 LDPC 码的几种译码算法的纠错性能和运算复杂度进行对比分析。其次结合课题背景, 对影响译码性能的参数, 如码长、迭代次数和数据量化等, 通过计算机仿真分析, 确定译码器硬件实现的方案。

论文结合课题背景对三种译码器架构进行对比分析, 串行、并行和部分并行。在译码速度和资源利用之间达到平衡, 选择灵活性较强的部分并行构架。根据理论仿真的参数和部分并行构架, 给出了译码器的顶层电路和基本功能模块划分, 并用硬件描述语言编写代码, 通过 ISE 综合、布局布线和 Modelsim 上进行时序仿真验证, 基于 FPGA 实现 LDPC 码译码器。

最后利用 VC 搭建测试系统, 在计算机上模拟信源、信道和误码率统计, LDPC 编码、LDPC 译码、MSK 调制和 MSK 解调在 FPGA 上实现, 并给出系统的测试方案和测试结果。测试结果表明, 基于 FPGA 实现的 LDPC 码译码器有很好的性能, 与理论译码性能基本一致, 译码器设计正确。

关键词: LDPC 码; Normalized BP based 算法; 译码器; FPGA 实现

ABSTRACT

In order to implement the LDPC decoder based on FPGA, many factors influencing properties of decoder have been studied in detail, such as construction of check matrix, code length, decoding algorithm, number of iterations, data quantization, etc. Firstly, we compared the computing complexity and error-correcting capability of several algorithms of LDPC decoding. Secondly, combined with the topic background, the leading parameters on the performance of LDPC decoding, such as code length, number of iterations, data quantization, etc, are analyzed with computer simulation. Finally, we gave the scheme of the hardware implement of LDPC decoders.

Based on the topic background, we analyzed 3 types of decoder architecture, serial, parallel and partially parallel. The Partially-Parallel architecture of good flexible is chosen for the trade off between decoding speed and resource requirement. Base on parameters of simulation and Partially-Parallel architecture, top-level circuit and basic function partition of the LDPC decoder is given and designed by Verilog HDL language. Some verification has been executed in ISE and Modelsim, and the LDPC decoder has been implemented on FPGA.

Finally, the whole test system is realized by VC. Information resource, channel and bit error rate statistics are realized by computer, LDPC encoder, LDPC decoder, MSK modulation and MSK demodulation are implementation based on FPGA. This thesis provided the whole test scheme and result. The test results indicate that LDPC decoder implementation based on FPGA has good coding performance, basically same as the theoretical coding performance.

Key words: LDPC code; Normalized BP based algorithm; decoder; FPGA implementation

目 录

第 1 章 绪论	1
1.1 课题研究背景及意义	1
1.2 LDPC 码的研究现状	3
1.3 LDPC 码译码算法研究与发展	4
1.4 LDPC 码的硬件实现研究	6
1.5 论文内容安排	7
第 2 章 LDPC 译码算法和系统仿真分析	8
2.1 LDPC 码的基础知识	8
2.1.1 LDPC 码的定义	8
2.1.2 LDPC 码的二分图表示	9
2.1.3 LDPC 码校验矩阵的构造	11
2.2 LDPC 码的译码算法	12
2.2.1 基于比特翻转的硬判决译码	13
2.2.2 基于软判决的译码	14
2.2.3 概率 BP 译码算法	17
2.2.4 LLR BP 译码算法	20
2.2.5 UMP BP Based 译码算法	22
2.2.6 Normalized BP based 译码算法	23
2.3 系统仿真分析和方案确定	24
2.3.1 校验矩阵的确定	24
2.3.2 译码算法的确定	25
2.3.3 码长的确定	28
2.3.4 译码迭代次数的确定	29
2.3.5 数据量化方案的确定	30
2.3.6 系统方案确定	34
2.4 本章小结	35

第 3 章 LDPC 译码器的 FPGA 设计与测试	36
3.1 LDPC 码译码器的 FPGA 设计流程及开发平台	36
3.2 常用 LDPC 译码器的硬件结构	39
3.2.1 串行结构	39
3.2.2 全并行结构	41
3.2.3 部分并行结构	42
3.3 LDPC 译码器的 FPGA 设计	44
3.3.1 LDPC 译码器的整体结构	44
3.3.2 LDPC 译码器的时序设计	46
3.3.3 LDPC 译码器存储阵列	53
3.3.4 LDPC 译码器信息初始化	55
3.3.5 LDPC 译码器信息迭代更新	56
3.3.6 LDPC 译码器译码信息输出	59
3.4 LDPC 译码器的测试	59
3.5 本章小结	62
第 4 章 系统测试的设计与实现	63
4.1 测试通信系统模型	63
4.2 系统测试的设计	64
4.2.1 信源产生	65
4.2.2 加噪处理	65
4.2.3 量化处理	65
4.2.4 误码率统计	65
4.2.5 系统测试界面	66
4.3 测试系统的实现	67
4.4 本章小结	68
结论	69
参考文献	71
攻读硕士期间发表论文和取得的科研成果	77
致谢	78

第 1 章 绪论

随着现代数字通信系统的发展,通信业务的越来越多,对信息有效可靠的传输要求也越来越高。一方面为了确保通信系统的可靠性,采用优良的信道编码已经成为现代通信系统中不可或缺的技术,另一方面频带资源在通信系统非常宝贵,在带宽有限的情况下,要求信道编码不仅要有好的纠错性能,而且系统的传输码率要高。

1.1 课题研究背景及意义

1948年,美国贝尔实验室的香农(Claude E.Shannon)发表了题为《通信的数学理论》的论文^[1],成为了现代信息理论的奠基性论文,标志着现代信息论与编码的诞生。香农在论文中提出了信道编码定理:对于任意的通信信道都有确定的信道容量 C ,并且在传输速率 R 小于 C 时,存在一种编码方法,当码长足够长,采用最大似然译码,系统的错误率可以达到任意小。同时信道编码定理还在数学上证明了好信道编码存在的可能性,虽然没有给出具体信道编码实现的方法,但却从理论上给出了信道编码的理论极限,为人们指明了信道编码研究的方向和目标。

根据 Shannon 的信息理论,数字通信系统模型如图 1.1 所示。系统中信道引入的干扰导致接收信号的错误,引入信道编码就是在发送端添加冗余,从而使接收信号错误最小化。对于一个信道编码,假如信道编码输入 k 个信息单元,输出 n 个信息单元,则码率为 $R = k/n$ ($R < 1$)。从数字通信系统模型框图可以看出,信道编码的引入可以提高系统的可靠性,编码的目的就是产生差别最大的码字,使错误判决的可能性减小,但是以引入冗余为代价。

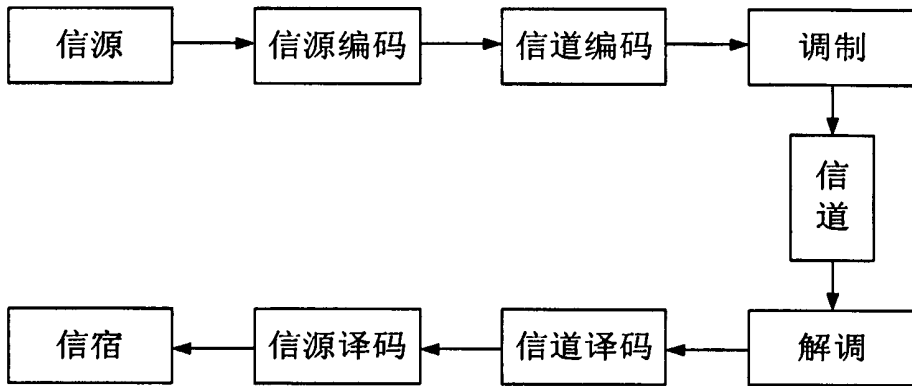


图 1.1 数字通信系统模型

虽然香农提出了一个随机选择的码可以寻找的好码,但是随机码采用最大似然译码的复杂度与码长呈指数关系,而在实际应用中,利用码长多项式复杂度来完成编译码的纠错码,因此尽管随机码的性能优越,但实用性不好。在 60 多年来,人们按照香农给出的基本条件来构造实用的好码,并且也取得很多成果。如:1950 年汉明 (Hamming) 发明了汉明码^[2],是一种能纠错一个错误的完备线性分组码;1962 年 Gallager 提出了低密度奇偶校验 (Low-density Parity-check) 码^[3],采用迭代译码方法进行译码,其纠错性能优异,但由于当时计算机水平的限制,没有引起人们的重视;1981 年 Tanner 提出了用于理解信道编码理论的 Tanner 图^[4],加快了信道编码发展的速度,现在已经广泛用于信道编码理论中;1993 年 C. Barrou、Glaveieux 和 Thitimajshima 提出 Turbo 码^[5],是一种信道编码理论界梦寐以求的可实用性非常好的码,其接近香农极限的性能标志着信道编码理论进入了一个崭新的阶段;1996 年,MacKay 和 Neal 重新发现 LDPC 码的优越性能^[6],人们还发现在二进制输入加性高斯白噪声 (BIAGWN) 信道下,当码率为 $1/2$,误码率为 10^{-6} ,码长为 10^7 时,仿真结果表明,最好的不规则 LDPC 码的门限值距香农限只有 0.0045dB^[7]。

目前国内外对 LDPC 码的探索研究已经广泛展开,并且应用潜力日益呈现,论文就是基于这种背景,在低信噪比信道的情况下,采用 LDPC 码的信道编码和 MSK 调制,在归一化信噪比 2dB 时,要求系统误码率低于

10^{-4} 。以此为设计的性能指标, 来实现整个通信系统。本文主要研究 LDPC 码译码的 FPGA 设计与实现。

1.2 LDPC 码的研究现状

自从 Mackay 和 Neal 重新发现 LDPC 码与 Turbo 码相当的性能, 而且在长码的情况下还超过了 Turbo 码后, LDPC 码成为新的研究热点, 得到大家的广泛关注。

目前, 对 LDPC 码的研究主要集中在以下几个方面:

第一: 考虑 LDPC 码在非 $GF(2)$ 上的构造, 即多元域上的编码, Mackay 和 Davey 等在此方向作了很多探索和尝试^[8], 并且取得了很好的成果。构造多元域上的好校验矩阵, 可以使性能有极大的提高。

第二: Gallager 提出的 LDPC 码, 其校验矩阵的列重和行重是固定的, 即规则的 LDPC 码; Luby、Mitzenmacher、Shokrollahi 和 Spielman 首先提出了构造不规则的二元 LDPC 码^[9], 即矩阵中的行重或列重不相同。研究表明, 相比最初的 Gallager 码, 非规则 LDPC 码利用和积算法(Sum-Product)迭代译码后, 在高斯白噪声信道中可以逼近信道容量。

第三: 将多元域与非规则进行优化组合, 以寻找性能更优的非 $GF(2)$ 的非规则 LDPC 码。Davey 已经找到了性能优于 Turbo 码的 LDPC 码。

第四: LDPC 码在工程中的应用。

由于 LDPC 码的优异性能以及在信息可靠传输中的良好应用前景(例如光通信、卫星通信、深空通信、第 4 代移动通信系统、高速与甚高速率数字用户线、光和磁记录系统等), 并且译码算法相对简单和硬件水平的提高, LDPC 码已经引起世界各国学术和 IT 业界的高度重视, 成为当今信道编码领域最受瞩目的研究热点。LDPC 码已经被 ADSL 标准^[10]和 DVB-S2 标准^{[11][12][13]}所采用。在 DVB-S2 标准中, LDPC 码从候选方案(串行和并行 Turbo 码(+RS)乘积码)中脱颖而出。尽管性能差别很小, 但 LDPC 码以最

低的复杂度提供了最好的性能, 并且短码还可以在性能损失 0.2dB 的情况下使复杂度减半, 这意味着在同等性能下, 采用 BP 算法译码的 LDPC 码的复杂度只有 Turbo 码的 1/4。LDPC 码在许多情况下将取代 Turbo 码的趋势已十分明显。研究 LDPC 码的学术意义、商业价值和 IT (特别是通信) 领域相关技术发展的推动作用巨大的。近几年国际上对 LDPC 码的理论研究已取得重要进展, 在工程应用和 VLSI (超大规模集成电路) 实现方面的研究正在全方位展开^{[14][15][16]}。

1.3 LDPC 码译码算法研究与发展

Gallager 不仅提出了规则 LDPC 码的构造方法, 而且提出了两种古典的译码方案, 一种基于硬判决译码, 另一种基于软判决译码。两种译码算法相比, 硬判决的译码运算量较小, 实现简单; 而软判决译码运算量大, 性能好, 应用迭代思想译码, 即 Gallager 的迭代译码算法, 通过证明迭代消息传递算法对于高性能低复杂度的编码方案是可行的。从完全硬判决到频谱软判决算法, 有很多迭代算法被应用于 LDPC 码的译码中, 并且每一种算法提供了不同的性能和复杂度的均衡。文献[17]提出的消息传递算法可以认为是比较好的折中算法, 消息传递算法也就是我们说的置信度传播 (Belief Propagation) 算法。

关于置信传播算法美国夏威夷大学的 Marc P.C. Fossorier 研究小组做了很多的研究, 给出了两种简化的 BP 译码算法, 在不需要信道先验信息情况下, 仅通过加法运算, 就可以快速迭代译码, 且软件与硬件的性能和复杂度有很好的平衡^[18]。将 BP 译码与可靠性译码相结合, 迭代中的 BP 算法传递的软信息值作为可靠性的度量, 来降低软判决译码的复杂度, 使 BP 次优译码最大限度的接近于最优译码^[19]。归一化后验概率算法的提出, 传递概率信息的准确性得到了进一步提高, 其中归一化因子可以理论推导得出, 也可以通过仿真确定, 与原 BP 译码算法相比, 算法的复杂度得到了极大的简化, 对接收值的归一化只需加法运算, 且高斯白噪声信道不影响译码性能。仿真结果表明, 在

高斯白噪声信道下,该算法性能与BP十分接近^{[20][21]}。通过对归一化后验概率算法的性能分析,并且采用密度进化估计信道参数的门限,性能逼近于BP算法^[22]。

另外北京邮电大学的吴湛击等人提出的差分译码算法^[23],迭代传递的消息是概率的差分值,消息更新都是加法域中进行,针对校验节点更新,可以选择若干个绝对值最小的差分值进行运算,进一步降低了复杂度。概率差分消息传递的译码算法与传统的BP译码算法相比,计算复杂度有很大的降低,且译码性能与收敛速度没有明显损失。

西安电子科技大学的贺玉成等提出量化的BP算法,用查表法来简化计算,译码性能接近于甚至超过连续译码算法^[24]。Yongyi Mao和A.H.Banihashmi提出基于概率的译码算法^{[25][26]},很大的提高了译码速度。通常的BP算法,在每一次消息迭代中运算中,校验节点和变量节点都要向相邻的节点传递信息即“灌溉”方案,对于不存在循环的码,“灌溉”的BP算法是一种最优的后验概率算法,然而对于中短码长即存在循环的码,这种BP算法不能达到最优效果。由Tanner图出发,根据最短循环的长度,得到每一个为节点的概率值,使位节点在迭代过程中有选择的进行消息传递,从而使位节点传递消息达到最优。仿真结果表明,在不增加复杂度的情况下,这种选择消息传递的BP性能优于传统的BP算法。A.Anastasopoulos将密度进化的和乘积算法(SPA)和最小和算法(min-sum)用于LDPC码和重复累积(RA)码的译码,最小和算法经过调整,性能接近于和乘积算法^[27]。

LDPC码的译码算法可以采用硬判决译码也可以采用软判决译码或者二者兼有。BF(Bit Filling)算法是一种低复杂的硬判决译码算法,其性能差于软判决译码算法,但实现简单。WBF(Weighted Bit Filling)及改进算法^{[28][29][30][31]}在纠错性能和译码复杂度进行了很好的均衡,它结合了BP算法和消息中一些软信息而改善了译码性能。如何利用LDPC码中其他信息改进翻转判决的处理策略,是进一步改善这类混合判决算法性能的一个研究方向。

不管LDPC码采用什么样的迭代译码算法,都具有三个优点:

1. 算法具有全并行结构, 译码速度快;
2. 译码算法的复杂性低^{[25][32]}, 运算量只随码长成线性增加;
3. 译码器设计灵活, 可以根据信道条件^{[33][34][35]}, 合理选择译码系统参数。

1.4 LDPC 码的硬件实现研究

目前工业界已经有 LDPC 编译码芯片问世。例如, 处在领先地位的 Flarion 公司已推出的基于 ASIC 的 Vector-LDPC 解决方案, 大约实用了 260 万门, 就可以支持高达 50000 的码长, 码率为 0.9, 最大迭代次数为 10, 译码器的吞吐量可以达到 10Gbps, 它的性能非常逼近香农限, 已经足以满足大多数通信业务的需求。并且 AHA 公司、Digital Fountain 公司也推出了各自的编译码解决方案。

Flarino 开发的 V-LDPC 的 flash-OFDM 移动无线集成芯片组可用于 IP 的移动宽带网。VOCAL Technologies.Ltd 也提出了一种 LDPC/Turbo 非对称解决方案用于 WLAN, 即上行链路采用 Turbo 码, 下行链路采用 LDPC 码, 研究表明采用此方案后对于 IEEE802.11 a/b/gWLAN 的移动终端的电池寿命可以延长 4 倍。Loeliger H A 等采用模拟 VLSI 实现和积译码算法的 LDPC 码译码器^[36], 避免数字系统中复杂的实数运算。Zhang T 和 Parhi K K 提出了一种编译码联合设计的 VLSI 实现的方案, 适合于并行译码^[37]。Levine B et al.给出了一种可以采用目前商用 FPGA 来实现的 LDPC 码验证性设计方案^[38]。Yeo E 等提出了用于磁记录的两类具有高吞吐率和低运算复杂度的 LDPC 码的译码方案, 并且给出了相应的串行译码结构^[39]。

DVB_S2 标准中, 采用了 LDPC 码与 BCH 码的级联作为信道编码方案, 并且 AHA 公司和意大利半导体公司已经拥有自主知识产权的 IP 核和编解码芯片以及 Xilinx 公司提供的 LDPC 码编码器的 IP 核都支持 DVB_S2 信道编码方案。还有我国的数字地面广播标准, 也采用具有自主知识产权的 QC_LDPC 码与 BCH 码的级联作为信道编码方案, 可以支持 0.4、0.6 和 0.8 三种码率。802.16e 标准采用了 LDPC 码为信道编码, 并提供了两种快速编码方案, 目前美国 Alberta 大学已经成功在 FPGA 上实现了 LDPC 码的编码

器设计及验证。

1.5 论文内容安排

本文结合 MSK 调制技术,以 LDPC 码译码器硬件设计与实现为研究目标,根据系统要求,对各种译码算法进行仿真分析,在译码性能和复杂度之间达到平衡,选出最佳的译码算法用于译码器的硬件实现。然后结合译码算法对通信系统进行深入的仿真分析与论证,确定影响译码性能的各个参数和数据量化方案。根据系统论证方案进行译码器顶层电路设计和各个功能模块划分,基于 FPGA 完成 LDPC 码译码器和下载测试。最后将信源、编码、调制、信道、解调、译码组成整个通信系统,通过 VC 测试界面完成系统测试。

第 1 章 主要介绍课题的研究背景、意义及目前研究状况。

第 2 章 主要介绍 LDPC 码的基本概念和原理,深入探讨 LDPC 的软判决译码算法,然后根据系统要求的对影响 LDPC 译码性能的各个参数进行详细的论证和仿真分析,最后通过仿真论证确定系统参数,为后面 LDPC 译码器的 FPGA 设计提供理论依据。

第 3 章 根据系统要求给出硬件译码器设计方案。首先,结合系统要求对现有的三种译码器结构进行分析对比。其次,详细介绍译码器的顶层设计和各功能模块的划分以及时序逻辑设计,基于 FPGA 实现 LDPC 码译码器。最后,给出 LDPC 码译码器的测试方案和最终的硬件下载和测试结果。

第 4 章 详细介绍了整个通信系统的搭建和测试工作,给出测试方案,并对实际测试结果与理论仿真进行对比分析。

第 2 章 LDPC 译码算法和系统仿真分析

LDPC 码是一种可以由非常稀疏校验矩阵定义的线性分组码, 具有优异的纠错性能和简单的迭代译码。本章首先简单介绍 LDPC 的基础知识, 然后详细介绍 LDPC 码的译码算法, 最后结合系统性能指标, 对系统参数进行仿真分析, 并确定满足系统要求各个参数。

2.1 LDPC 码的基础知识

2.1.1 LDPC 码的定义

LDPC 码是一种比较特殊的 (n, k) 线性分组码, 构成其核心部分的校验矩阵 H 是稀疏矩阵, 即校验矩阵 H 中包含大多数的“0”和极少数的“1”。由于校验矩阵具有稀疏性, 降低了 LDPC 码的译码复杂度, 便于硬件实现。在二元域中的规则 LDPC 码 (n, w_c, w_r) 表示码长为 n , 每列中包含 w_c 个“1”, 每行中包含 w_r 个“1”。

对于 $H_{m \times n}$ 校验矩阵, 每一行对应一个校验方程, 每一列对应码字的一位。每一行中“1”的个数称为行重, 每一列中“1”的个数称为列重。由校验矩阵可以计算出码率 R_c ,

$$R_c = (n - m) / n \quad (2-1)$$

其中 n 为码长, m 为校验位长, $n - m$ 表示信息位长。码率也可以用列重和行重表示, 如 2-2 式所示。

$$R_c = 1 - w_c / w_r \quad (2-2)$$

对于一个校验矩阵 H , 如果行重或列重不为常数, 所构成的 LDPC 码称为不规则 LDPC 码。

由于 LDPC 码是一类特殊的线性分组码, 因此它也具有线性分组码的特性。将输入信息序列通过生成矩阵 G 映射为码字 c , 则所有的码字满足校验方程, 也就是 $c * H^T = 0$ 。如式 (2-3) 所示的校验矩阵 H , 表示为 (12,3,6) 规则 LDPC 码, 码长 n 为 12, 列重 w_c 为 3, 行重 w_r 为 6。式 (2-4) 为校验矩阵所对应的校验方程, 其中码字 $c = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}) \in C$, 满足 $c * H^T = 0$ 。

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-3)$$

$$\begin{cases} c_1 + c_2 + c_3 + c_7 + c_9 + c_{10} = 0 \\ c_2 + c_5 + c_6 + c_7 + c_{11} + c_{12} = 0 \\ c_1 + c_4 + c_6 + c_7 + c_8 + c_9 = 0 \\ c_2 + c_3 + c_4 + c_{10} + c_{11} + c_{12} = 0 \\ c_5 + c_6 + c_8 + c_9 + c_{10} + c_{11} = 0 \\ c_1 + c_3 + c_4 + c_5 + c_8 + c_{12} = 0 \end{cases} \quad (2-4)$$

2.1.2 LDPC 码的二分图表示

定义 2.1: 二部图是一个节点由无向分支连接的图, 且其节点可以分为两类, 分支只能连接处于不同类中的节点^[40]。

由于 Michael Tanner 最先研究用二部图来表示 LDPC 码, 所以将之称为 Tanner 图。在 Tanner 图中, 校验矩阵分为两类节点: 变量节点和校验节点, 其中变量节点个数为 n , 校验节点个数为 m 。只有当校验矩阵 H 为“1”的位置, 才存在校验节点与变量节点间的分支连接。对于规则的 LDPC 码, 一

共有 $w_r * m = w_c * n$ 个分支, 其中每个校验节点对应于 w_r 条分支连接, 每个变量节点对应于 w_c 条分支连接。如图 2.1 为式(2-3)校验矩阵所对应的 Tanner 图, v_i 为变量节点, f_i 为校验节点, v_i 与 f_i 之间的连线就是边。与每一个校验节点连接的所有变量节点构成一个校验方程。

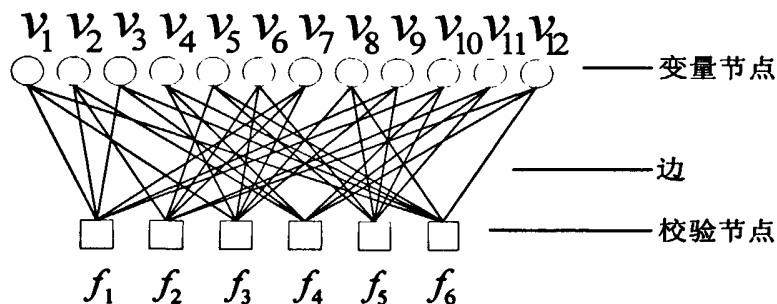


图 2.1 (12, 3, 6) 规则 LDPC 码的 Tanner 图

定义 2.2: 连接一个节点分支数目称为节点的阶^[40]。如图 2.1 中变量节点的阶是 3, 校验节点的阶是 6。

定义 2.3: 环和周长: Tanner 图中一个长为 l 的环是一个能够回到起点的包含 l 个分支的闭合路径。图的最小环长称为图的周长^[40]。

在 Tanner 图中可能最小的环长为 4, 对应于校验矩阵 H 中一个子矩阵, 如图 2.2 所示。

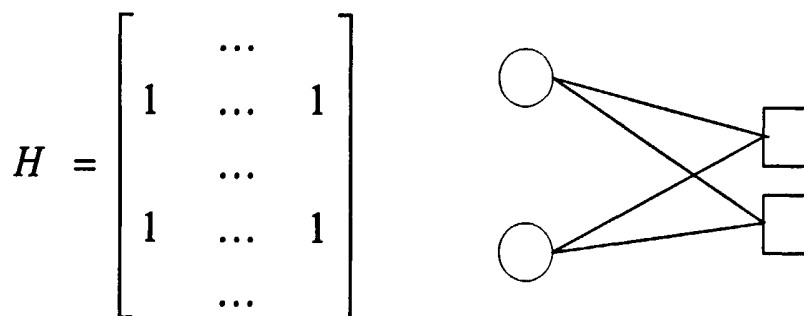


图 2.2 环长为 4 的 Tanner 图

由于 Tanner 图中的存在短环导致了 LDPC 码的译码不充分, 节点之间外部信息的传递独立性减小, 抗干扰能力下降, 从而没有达到最优的译码

性能。Tanner 图中的环决定了节点向外传递的消息再次回到本节点所需要的最小迭代次数，并且随着环的增大，LDPC 码的译码性能也随之改善。因此如何构造性能优异、编译码简单的校验矩阵已经成为了人们研究的热点问题。

2.1.3 LDPC 码校验矩阵的构造

LDPC 码的一个吸引人之处，就是它构造方式的简单。构造 LDPC 码关键是构造它的稀疏校验矩阵，其稀疏校验矩阵的特征直接影响到 LDPC 码编译码器的复杂度和性能。

对于稀疏校验矩阵的构造主要遵循两个原则^[41]：

- 1.避免短环的出现。短环导致外部消息在变量节点与校验节点之间来回的传递，更新效果不好。

- 2.避免变量节点连接的校验方程过于集中。变量节点连接的校验方程过于集中，容易导致 LDPC 码错误地板的发生。

现阶段校验矩阵的构造方式主要分为两大类：随机校验矩阵和结构化校验矩阵。随机校验矩阵纠错性能好，但由于其随机性，使得编译码复杂度高，甚至难于实现。结构化校验矩阵具有确定的结构，可以实现线性时间的编码和简单译码，硬件复杂度相对于随机校验矩阵得到了极大简化，并且可以采用相应的措施使其纠错性能接近于随机校验矩阵。目前研究比较热门的结构化校验矩阵是 QC_LDPC 码的构造。

QC_LDPC 码是一种特殊结构化的 LDPC 码，其校验矩阵是由一系列大小一致的循环子矩阵构成^{[42][43]}。一个循环矩阵都应具有以下特点：

- 1.是一个方阵；

- 2.矩阵中的每一行都是前一行右循环移动 1 位的结果，其中矩阵中第一行是最后一行移位结果；

- 3.矩阵中的每一列都是前一列下循环移动 1 位的结果，其中矩阵中第一

列是最后一列移位结果。

如式(2-5)所示的是重量为 1 的 4×4 循环矩阵。对于单位矩阵，也可以看成是循环系数为 0 的循环矩阵。

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (2-5)$$

QC_LDPC 码的校验矩阵就可以表示为：

$$H = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdot & \cdot & \cdot & A_{0,L-1} \\ A_{1,0} & A_{1,1} & \cdot & \cdot & \cdot & A_{1,L-1} \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ A_{M-1,0} & A_{M-1,1} & \cdot & \cdot & \cdot & A_{M-1,L-1} \end{bmatrix} \quad (2-6)$$

其中 $A_{i,j}$ 是由单位矩阵、循环矩阵和全零阵，其大小都是 $p \times p$ ，整个 QC_LDPC 码的校验矩阵 H 由 $M \times L$ 个 $A_{i,j}$ 矩阵构成。其构造的校验矩阵有三个优点：

1. 降低了校验矩阵对非零元素位置信息的存储要求；
2. 实现了 LDPC 码的线性复杂度编码；
3. 降低了 LDPC 码译码器的复杂度。

2.2 LDPC 码的译码算法

Gallager 于 1963 年给出了 LDPC 码的两类译码方案：基于硬判决的译码和基于软判决的译码。基于硬判决的 LDPC 码译码的运算量小，复杂度低，但是没有充分发挥 LDPC 码的性能优势。然而基于软判决的译码采用的是一种概率译码，通过反复的迭代运算，使得 LDPC 码的性能逼近于香农极限。

采用 Tanner 图来描述的 LDPC 码使得译码很方便。在讨论 LDPC 码译码算法之前先给出外部信息传递的消息以及与节点相邻节点的集合：

1. $q_{v,f}$ 表示从变量节点 v 向校验节点 f 传递的消息；
2. $h_{f,v}$ 表示从校验节点 f 向变量节点 v 传递的消息；
3. N_v 表示与变量节点 v 相邻节点的集合；
4. N_f 表示与校验节点 f 相邻节点的集合。

在硬判决译码中 $q_{v,f}$ 和 $h_{f,v}$ 的取值范围为 $\{0, 1\}$ ，而对于软判决译码 $q_{v,f}$ 和 $h_{f,v}$ 取值范围为 $(-\infty, +\infty)$ 。

2.2.1 基于比特翻转的硬判决译码

比特翻转硬判决译码的译码原理是利用接收序列中某些比特翻转与相对应的校验失败数目的变化进行译码，即翻转最少的接收序列来满足校验方程。

下面给出比特翻转的硬判决流程：

1. 计算校验和，即 $\hat{c} * H^T = 0$ ，如果 $\hat{c} * H^T = 0$ ，则译码结束；
2. 统计码字中第 i 比特不满足校验方程的个数，记作 f_i ，
 $i = 0, 1, \dots, n-1$ ；
3. 定义集合 Ω ，集合中只包含最大的 f_i ；
4. 翻转集合 Ω 中的比特；
5. 重复步骤 1 至 4，直到 $\hat{c} * H^T = 0$ 或达到最大的迭代次数。

比特翻转算法只进行逻辑运算，因此实现简单、译码速度快并且复杂度低，但是译码性能较差，而且硬判决比特翻转译码算法还存在两个缺点：

1. 如果同时出现几个比特的错误，在进行翻转不能满足校验方程，使迭代进入一个死循环；
2. 如果错误的比特数正好满足校验方程，则出现伪译码，从而导致错误译码。

2.2.2 基于软判决的译码

软判决译码方案采用最大似然概率译码,被认为是二进制信道中最佳译码方案,并且现代信道编码性能优越主要原因之一就是采用了最大似然概率的迭代译码。

设发送信息序列为 $\{x_i\}$,噪声干扰为 $\{n_i\}$,接收信息序列为 $\{y_i\}$,其中噪声 $\{n_i\}$ 为高斯白噪声,则 y_i 可以表示为 $y_i = x_i + n_i$,其中 i 为码字的一个比特。

下面为迭代译码的一般流程^[44]:

- 1.初始化,即将接收的软信息 $\{y_i\}$ 转换为概率信息;
- 2.从变量节点向校验节点传递消息;
- 3.从校验节点向变量节点传递消息;
- 4.判决信息,估计码字 \hat{x} 和 \hat{c} 。如果 $\hat{c} * H^T = 0$ 或达到最大的迭代次数,则停止译码,输出译码结果;否则返回步骤 1。

迭代译码中消息的传递可以用 Tanner 因子图表示信息的传递,如图 2.3 所示和 2.4 所示。

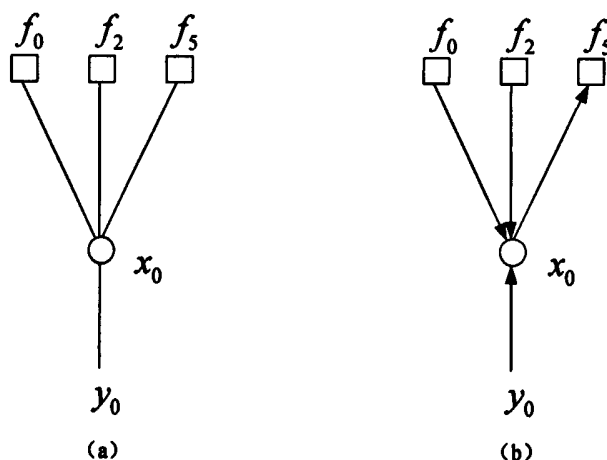


图 2.3 校验矩阵列传递信息的 Tanner 因子图

在消息传递迭代译码的每一次迭代运算都必须经过两次消息的传递,即变量节点到校验节点的消息传递和校验节点到变量节点的消息传递。图

2.3 是式(2-3)中校验矩阵 H 第一列的 Tanner 因子图, 变量节点 x_0 向与其连接的校验节点传递外部信息, 外部信息从接收信息 y_0 和其它校验节点获得。如 $x_0 \rightarrow f_3$ 传递的外部信息是从 y_0 获得的信道接收的概率信息和校验节点 f_0, f_2 的上半次迭代得到的概率信息。 $x_i \rightarrow f_j$ 信息传递是所有的变量节点与校验节点的信息对进行的。译码算法上半次迭代中消息传递如图 2.4 所示, 进行消息的反向传递 $f_j \rightarrow x_i$, 即由校验节点传向变量节点。图 2.4(a)

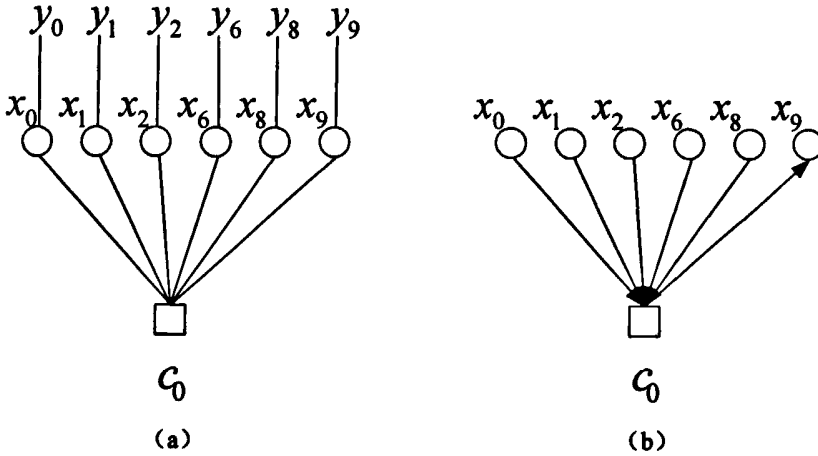


图 2.4 校验矩阵行传递消息的 Tanner 因子图

是式(2-3)中校验矩阵 H 第一行的 Tanner 因子图, 校验节点 f_0 向与其连接的变量节点传递外部信息, 图 2.4(b)为校验节点 f_0 向变量节点 x_9 传递信息。当 $x_i \rightarrow f_j$ 和 $f_j \rightarrow x_i$ 结束, 表明完成一次迭代。只有当迭代次数达到最大或者 $\hat{c} * H^T = 0$ 时, 才结束译码, 输出译码结果。

对于最大似然概率译码关键的一步就是计算与接收序列软信息相对应的概率信息以及在迭代过程中消息概率计算。为了便于进一步讨论, 下面先给出将要用到的符号:

$P_i = \Pr(c_i = 1 | y_i)$: 码字中 $c_i = 1$ 的后验概率信息;

$P_{kj} = \Pr(c_{kj} = 1 | y_{kj})$: 包含 c_i 的第 j 个校验方程中的第 k 个比特 $c_{kj} = 1$ 的后验概率信息;

$N_x(i) = \{j : h_{ji} = 1\}$: 表示与第 i 变量节点相邻的校验节点的集合, 即对

应于校验矩阵第 i 列中 “1” 的集合;

$N_f(j) = \{i: h_{ji} = 1\}$: 表示与第 j 校验节点相邻的变量节点的集合, 即对应于校验矩阵第 j 行中 “1” 的集合;

$N_x(i) \setminus j$: 表示除了 j 外与变量节点 i 相邻的所有校验节点的集合;

$N_f(j) \setminus i$: 表示除了 i 外与校验节点 j 相邻的所有变量节点的集合;

S_i : 表示所有满足码字 c 中包含码元 c_i 的 ω_c 校验方程的事件。

引理 2.1(Gallager): 设有 m 个独立二元比特的序列 $a = (a_1, a_2, \dots, a_m)$, 其中 $P_r(a_k = 1) = P_k$, 则 a 含有偶数个 “1” 的概率为^[40]

$$\Pr(\sum_{k=1}^m a_k = 0) = \frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2P_k) \quad (2-7)$$

对应的 a 含有奇数个 “1” 的概率为 $\frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2P_k)$ 。

[证明] (采用归纳法)

当 $m = 2$ 时,

$$\Pr(\text{even}) = \Pr(a_1 \oplus a_2 = 0) = P_1 P_2 + (1 - P_1)(1 - P_2) = \frac{1}{2} + \frac{1}{2} (1 - 2P_1)(1 - 2P_2)$$

假设式(2-7)对 $m = L - 1$ 成立, 对 $Z_L = a_1 + a_2 + \dots + a_L$, 则有

$$\begin{aligned} \Pr(Z_L = 0) &= \Pr(Z_{L-1} \oplus a_L = 0) \\ &= \frac{1}{2} + \frac{1}{2} [1 - 2\Pr(Z_{L-1} = 1)](1 - 2P_L) \\ &= \frac{1}{2} + \frac{1}{2} [1 - 1 + \prod_{k=1}^{L-1} (1 - 2P_k)](1 - 2P_L) \\ &= \frac{1}{2} + \frac{1}{2} \prod_{k=1}^L (1 - 2P_k) \end{aligned}$$

定理 2.1(Gallager): 设信道为无记忆信道, 给定信道观测矢量 y 和事件 S_i 后 c_i 的似然比^[40]

$$\frac{\Pr(c_i = 0 | y, S_i)}{\Pr(c_i = 1 | y, S_i)} = \frac{1 - P_i}{P_i} \prod_{j \in N_x(i)} \left[\frac{1 + \prod_{l' \in N_j(j) \setminus \{i\}} (1 - 2P_{l'j})}{1 - \prod_{l' \in N_j(j) \setminus \{i\}} (1 - 2P_{l'j})} \right] \quad (2-8)$$

[证明] 由贝叶斯准则, 有

$$\frac{\Pr(c_i = 0 | y, S_i)}{\Pr(c_i = 1 | y, S_i)} = \frac{1 - P_i}{P_i} * \frac{\Pr(S_i | c_i = 0, y)}{\Pr(S_i | c_i = 1, y)} \quad (2-9)$$

设定 $c_i = 1$, 如果包括 c_i 的校验方程中其他 $\omega_c - 1$ 比特包含奇数个“1”, 就能满足 c_i 的校验。由引理 2.1 可知, 在第 j 个校验方程中其余 $\omega_c - 1$ 比特有奇数个“1”的概率为

$$\frac{1}{2} - \frac{1}{2} \prod_{l' \in N_j(j) \setminus \{i\}} (1 - 2P_{l'j}) \quad (2-10)$$

假设校验方程中的码字比特满足独立性, 则 ω_c 个校验联合概率是每个校验概率之积, 有

$$\Pr(S_i | c_i = 1, y) = \prod_{j \in N_x(i)} \left[\frac{1}{2} - \frac{1}{2} \prod_{l' \in N_j(j) \setminus \{i\}} (1 - 2P_{l'j}) \right] \quad (2-11)$$

同理有

$$\Pr(S_i | c_i = 0, y) = \prod_{j \in N_x(i)} \left[\frac{1}{2} + \frac{1}{2} \prod_{l' \in N_j(j) \setminus \{i\}} (1 - 2P_{l'j}) \right] \quad (2-12)$$

将式(2-11)和式(2-12)代入式(2-9), 就得到定理 2.1。

2.2.3 概率 BP 译码算法

在 20 世纪 90 年代人们将人工智能中的信度传播(BP, belief-propagation)算法引用到 LDPC 码中, 并成为了 LDPC 码的现代译码方案, 即软判决译码。BP 译码算法是建立在 Tanner 图上的 LDPC 码的软判决译码, 其每一次

迭代包括两步：校验节点信息更新和变量节点信息更新^[40]。根据信息的表现形式，BP 译码可以分为概率 BP 译码算法和 LLR BP 译码算法。概率 BP 译码算法的信息都是以概率形式表示的，可以用于任何进制的 LDPC 译码，而 LLR BP 译码算法的信息是以对数似然比表示的，主要针对二进制的 LDPC 译码。

为了方便概率 BP 译码算法的讨论，先定义将要用的符号：

$c_{kj} \in GF(2)$ ：表示第 j 个含有 c_i 的校验方程的第 k 位；

$x_{kj} = (-1)^{c_{kj}}$ ：与 c_{kj} 相应码字映射的传输序列 x_{ki} ；

$y_{kj} = x_{kj} + n_k$ ， n_k 是高斯白噪声 $N(0, \sigma^2)$ ， y_{ki} 为接收序列信息；

$h_{ji}(b)$ ($b=0,1$)：表示校验节点 f_j 传递给变量节点 v_i 的外部传递信息，即在给定信息位而且其他信息位满足独立概率分布条件下，校验方程 j 所满足的概率信息；

$q_{ij}(b)$ ：表示变量节点 v_i 传递给校验节点 f_j 的外部传递信息，即在获得除 v_i 自身以外其他所有校验节点以及信道的外部传递信息后，判断 $c_i = b$ 的概率；

由引理 2.1，可以得到校验节点的更新规则：

$$\begin{cases} h_{ji}(1) = \frac{1}{2} - \frac{1}{2} \prod_{i' \in N_f(j) \setminus (i)} (1 - 2P_{i'j}) \\ h_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in N_f(j) \setminus (i)} (1 - 2P_{i'j}) \end{cases} \quad (2-13)$$

由定理 2.1，可以得到变量节点的更新规则：

$$\begin{cases} q_{ij}(0) = (1 - P_i) \prod_{j' \in N_x(i) \setminus (j)} h_{i'j'}(0) \\ q_{ij}(1) = P_i \prod_{j' \in N_x(i) \setminus (j)} h_{i'j'}(1) \end{cases} \quad (2-14)$$

根据高斯白噪声信道，得 $x_i = b$ 的后验概率：

$$P_r(x_i = b | y_i) = \frac{1}{1 + e^{-2y_i x_i / \sigma^2}} \quad (2-15)$$

其中 $b \in \{+1, -1\}$ 。

下面为概率 BP 译码算法流程：

1. 初始化

$$\begin{cases} q^0_{ij}(0) = 1 - P_i = \Pr(x_i = +1 | y_i) = \frac{1}{1 + \exp(-2y_i / \sigma^2)} \\ q^0_{ij}(1) = P_i = \Pr(x_i = -1 | y_i) = \frac{1}{1 + \exp(2y_i / \sigma^2)} \end{cases} \quad (2-16)$$

2. 校验节点更新，即校验矩阵中行变换处理

$$\begin{cases} h'_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in N_f(j) \setminus \{i\}} (1 - 2q^{i'-1}_{i'j}(1)) \\ h'_{ji}(1) = 1 - h'_{ji}(0) \end{cases} \quad (2-17)$$

3. 变量节点更新，即校验节点中列变换处理

$$\begin{cases} q^l_{ij}(0) = (1 - P_i) \prod_{j' \in N_s(i) \setminus \{j\}} h'_{j'i}(0) \\ q^l_{ij}(1) = P_i \prod_{j' \in N_s(i) \setminus \{j\}} h'_{j'i}(1) \end{cases} \quad (2-18)$$

归一化，有

$$\begin{cases} q^l_{ij}(0) = \frac{q^l_{ij}(0)}{q^l_{ij}(0) + q^l_{ij}(1)} \\ q^l_{ij}(1) = 1 - q^l_{ij}(0) \end{cases} \quad (2-19)$$

4. 译码判决

对所有 i 的计算：

$$\begin{cases} Q'_i(0) = K_i (1 - P_i) \prod_{j \in N_s(i)} h'_{ji}(0) \\ Q'_i(1) = K_i P_i \prod_{j \in N_s(i)} h'_{ji}(1) \end{cases} \quad (2-20)$$

其中常数 K_i 的选择应使 $Q'_i(0) + Q'_i(1) = 1$ 。

对所有变量节点判决：

$$\hat{c}_i = \begin{cases} 1 & , Q'_i(1) > Q'_i(0) \\ 0 & , \text{其他} \end{cases} \quad (2-21)$$

5.译码停止

如果 $\hat{c} * H^T = 0$ 或者达到最大迭代次数，则译码结束，输出译码结果；否则返回步骤 2 继续迭代。

2.2.4 LLR BP 译码算法

概率 BP 译码算法译码性能是 LDPC 码中译码最好的译码算法，但是算法中有指数运算和很多乘法运算，并且当码长很大时，译码计算会不稳定，影响收敛性。为了简化算法中乘法运算量，用对数似然比 BP 译码算法来代替概率 BP 译码算法。

为了方便讨论，先定义对数似然比(LLR)：

信道初始化消息：

$$L(c_i) = \ln \frac{\Pr(x_i = +1 | y_i)}{\Pr(x_i = -1 | y_i)} \quad (2-22)$$

变量节点向校验节点传递的消息：

$$L(q_{ij}) = \ln \frac{q_{ij}(0)}{q_{ij}(1)} \quad (2-23)$$

校验节点向变量节点传递的消息：

$$L(h_{ji}) = \ln \frac{h_{ji}(0)}{h_{ji}(1)} \quad (2-24)$$

变量节点收集到的所有消息：

$$L(Q_i) = \ln \frac{Q_i(0)}{Q_i(1)} \quad (2-25)$$

又知：

$$\tanh\left[\frac{1}{2}\ln\left(\frac{P_0}{P_1}\right)\right] = P_0 - P_1 = 1 - 2P_1 \quad (2-26)$$

$$1 - 2h_{ji}(1) = \prod_{i' \in N_f(j) \setminus \{i\}} (1 - 2q_{i'j}(1)) \quad (2-27)$$

由式(2-26)和式(2-27)可得:

$$\tanh\left[\frac{1}{2}L(q_{ji})\right] = \prod_{i' \in N_f(j) \setminus \{i\}} \tanh\left[\frac{1}{2}L(q_{i'j})\right] \quad (2-28)$$

根据上述定义, 下面给出 LLR BP 译码算法的流程:

1. 初始化(主要针对高斯白噪声信道)

$$L(q_{ij}) = L(c_{ij}) = 2y_i / \sigma^2 \quad (2-29)$$

2. 校验节点更新

$$L(h_{ji}) = 2 \tanh^{-1} \left[\prod_{i' \in N_f(j) \setminus \{i\}} \tanh\left(\frac{1}{2}L^{k-1}(q_{i'j})\right) \right] \quad (2-30)$$

可以将式(2-30)进一步简化, 根据 Gallager 的定义, 将 $L(q_{ij})$ 表示为 $L(q_{ij}) = \alpha_{ij}\beta_{ij}$, 其中 $\alpha_{ij} = \text{sign}(L(q_{ij}))$, $\beta_{ij} = |L(q_{ij})|$

则有

$$\tanh\left[\frac{1}{2}L(q_{ji})\right] = \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot \prod_{i' \in N_f(j) \setminus \{i\}} \tanh\left(\frac{1}{2}\beta_{i'j}\right) \quad (2-31)$$

式(2-30)可以表示为

$$\begin{aligned} L(r_{ji}) &= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot 2 \tanh^{-1} \left(\prod_{i' \in N_f(j) \setminus \{i\}} \tanh\left(\frac{1}{2}\beta_{i'j}\right) \right) \\ &= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot 2 \tanh^{-1} \ln^{-1} \left[\ln \left(\prod_{i' \in N_f(j) \setminus \{i\}} \tanh\left(\frac{1}{2}\beta_{i'j}\right) \right) \right] \\ &= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot 2 \tanh^{-1} \ln^{-1} \sum_{i' \in N_f(j) \setminus \{i\}} \ln \left(\tanh\left(\frac{1}{2}\beta_{i'j}\right) \right) \end{aligned}$$

$$= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot f\left(\sum_{i' \in N_f(j) \setminus \{i\}} f(\beta_{i'j})\right) \quad (2-32)$$

其中

$$f(x) = -\ln[\tanh(x/2)] = \ln\left(\frac{e^x + 1}{e^x - 1}\right) \quad (2-33)$$

根据性质 $f^{-1} = f$ ，有

$$f[f(x)] = \ln \frac{e^{f(x)} + 1}{e^{f(x)} - 1} = x \quad (2-34)$$

可以根据函数 $f(x)$ 性质，用查表法实现校验节点更新。

3. 变量节点更新

$$\begin{aligned} L(q_{ij}) &= \ln\left(\frac{1-P_i}{P_i}\right) + \sum_{j' \in N_x(i) \setminus \{j\}} L(h_{j'i}) \\ &= L(c_i) + \sum_{j' \in N_f(i) \setminus \{j\}} L(h_{ji'}) \end{aligned} \quad (2-35)$$

4. 译码判决

$$L(Q_i) = L(c_i) + \sum_{j \in N_x(i)} L(h_{ji}) \quad (2-36)$$

对所有的变量节点有

$$\hat{c}_i = \begin{cases} 1 & , L(Q_i) < 0 \\ 0 & , \text{其他} \end{cases} \quad (2-37)$$

5. 译码停止

如果 $\hat{c} * H^T = 0$ 或者达到最大迭代次数，则译码结束，输出译码结果；否则返回步骤 2 继续迭代。

2.2.5 UMP BP Based 译码算法

为了进一步简化译码算法的复杂度，将 LLR BP 译码算法中的校验节点更新表示为：

$$\begin{aligned}
 L(h_{ji}) &= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot f\left(\sum_{i' \in N_f(j) \setminus \{i\}} f(\beta_{i'j})\right) \\
 &= \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot \min_{i' \in N_f(j) \setminus \{i\}} (\beta_{i'j})
 \end{aligned} \tag{2-38}$$

对于高斯白噪声信道，初始化的 LLR 消息因子 $\frac{2}{\sigma^2}$ 对迭代没有影响，因此可以忽略。下面给出 UMP BP Based 译码算法的译码流程：

1. 初始化

$$L(q_{ij}) = L(c_{ij}) = y_i \tag{2-39}$$

2. 校验节点更新

$$L(h_{ji}) = \prod_{i' \in N_f(j) \setminus \{i\}} \alpha_{i'j} \cdot \min_{i' \in N_f(j) \setminus \{i\}} (\beta_{i'j}) \tag{2-40}$$

3. 变量节点更新

$$L(q_{ij}) = L(c_i) + \sum_{j' \in N_f(i) \setminus \{j\}} L(h_{ji'}) \tag{2-41}$$

4. 译码判决

$$L(Q_i) = L(c_i) + \sum_{j \in N_x(i)} L(h_{ji}) \tag{2-42}$$

对所有的变量节点有

$$\hat{c}_i = \begin{cases} 1 & , L(Q_i) < 0 \\ 0 & , \text{其他} \end{cases} \tag{2-43}$$

5. 译码停止

如果 $\hat{c} * H^T = 0$ 或者达到最大迭代次数，则译码结束，输出译码结果；否则返回步骤 2 继续迭代。

2.2.6 Normalized BP based 译码算法

虽然 UMP BP Based 译码算法中只有加法和比较运算，使硬件复杂度上

得到了极大的降低，但同时给译码性能带来了的较大损失。因此，为了在硬件复杂度与译码性能之间取一个折中方案，引入了归一化因子 μ 来校正 UMP BP Based 译码算法中校验节点更新引起的误差。即将式(2-40)变为：

$$L(h_{ji}) = \prod_{i \in N_f(j) \setminus \{i\}} \alpha_{i,j} \cdot \mu \cdot \min_{i \in N_f(j) \setminus \{i\}} (\beta_{ij}) \quad (2-44)$$

其中 $\mu \in (0,1)$ 。可以通过求第一次迭代校验消息和第二次迭代校验消息的期望比值和密度进化来估计归一化因子 μ 。

2.3 系统仿真分析和方案确定

结合通信系统的性能指标要求：采用 MSK 调制，信道编码码率为 1/2，归一化信噪比在 SNR=2dB 时，系统误码率要达到 10^{-4} 以下。以此为目标，采用性能优异的 LDPC 码作为信道编码，分析 LDPC 码译码器对系统性能影响的各个因素，通过仿真论证，确定满足系统性能要求的系统参数。

LDPC 码对系统性能影响的因素主要有：校验矩阵 H ，码长，码率，译码算法，迭代次数，数据量化方式等。然后对多种预选方案进行仿真，并分析仿真结果，确定满足系统性能指标的各个参数，其中本文采用的信道编码码率固定，在此不予考虑。

2.3.1 校验矩阵的确定

目前 LDPC 码的校验矩阵构造方式主要有两类：随机构造和结构化构造。随机构造由于校验矩阵随机性强，纠错性能好，但是编译码复杂度高，硬件实现困难，因此应用场合较少。然而结构化构造由于校验矩阵结构性很强，比较利于硬件实现，并且可以设计环长较大的码，纠错性能可以与随机构造的 LDPC 码纠错性能相媲美。

本文采用的 LDPC 码校验矩阵是一种具有矩阵结构性很强的下三角结构非规则准循环矩阵，其译码器的硬件实现比较灵活。QC_LDPC 码的校验

矩阵的参数主要有矩阵维数、循环子矩阵 $p \times p$ 的大小、节点度数分布等。校验矩阵的维数主要由码长和码率来决定，本文码率固定为 $1/2$ ，因此校验矩阵维数由码长决定，并且校验矩阵的维数是 p 的整数倍。 p 的值决定了译码器吞吐量，循环子矩阵的大小决定了译码器设计的复杂度，对于相同码长情况下，如果 p 值很大，则校验矩阵 H 包含循环子矩阵的个数较少，因此译码器设计较简单，同时译码性能有所损失；然而 p 值小，则校验矩阵 H 包含循环子矩阵的个数多，因此译码器设计复杂，译码性能较好。根据系统要求，本文选择 $p = 256$ 就能满足系统指标。对于节点度数分布可以通过密度进化得到，本文不做重点研究。

2.3.2 译码算法的确定

通过分析 LDPC 码的两大类译码方法，虽然基于硬件判决的译码算法在硬件实现上具有很大的优势，但由于其性能满足不了本系统要求，因此，本系统只能选择基于软判决的译码算法，虽然硬件实现复杂度大，但性能比较优越。前文讨论了 4 种软判决译码算法，其中概率 BP 译码算法具有最优的译码性能，在信噪比较低的情况下，仍具有优异的纠错性能，并且随着信噪比的增大，误码率下降速度快；随着迭代次数的增加，误码率曲线收敛快。LLR BP 译码算法是概率 BP 译码算法在对数域上的等效简化算法，大量的乘法运算变为加法运算，运算量得到了极大的简化，并且减少了译码时间和硬件实现复杂度。UMP BP Based 译码算法(最小和算法)是 LLR BP 译码算法进一步简化，整个译码过程中只存在加法运算和比较运算，译码硬件实现复杂度得到了进一步的简化，但是其性能较差，并且随着信噪比的增大，误码率下降比较慢；随着迭代次数的增加，误码率曲线的收敛速度也较慢。Normalized BP based 译码算法是对 LLR BP 译码算法和 UMP BP Based 译码算法的一种折中，其译码性能与 LLR BP 译码算法相媲美，而译码复杂度与 UMP BP Based 译码算法相当。如图 2.5 所示为 4 中软判决译码

算法的误码率曲线。

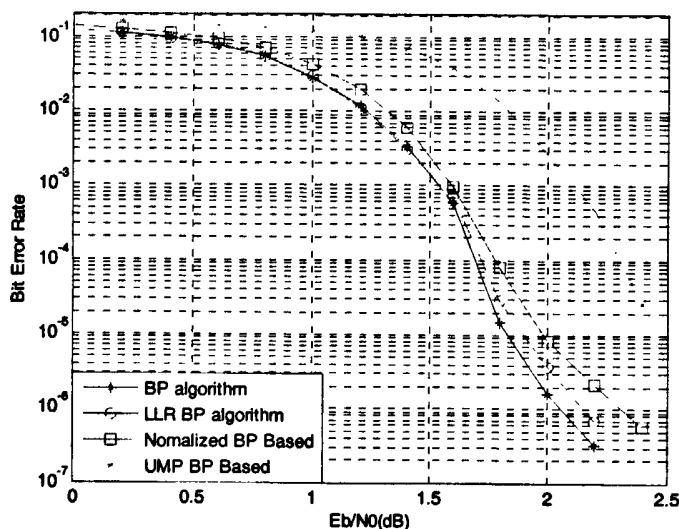


图 2.5 码长 2048 和码率 1/2 的 4 种软判决译码算法的误码率

对于硬件实现 LDPC 码译码器，概率 BP 译码算法，运算量大，硬件实现复杂度高，一般不考虑采用，除非在码长比较短且对误码率要求比较高的场合。对于 UMP BP Based 译码算法，虽然硬件实现简单，但是由于译码性能比较差，因此，只有在对性能要求不高的情况下采用。LLR BP 译码算法的译码性能略微低于概率 BP 译码算法，硬件复杂度且大量减少，因此比较适合于对性能要求较高的场合。Normalized BP based 译码算法与 LLR BP 译码算法相比较，从译码性能上来说，Normalized BP based 译码算法稍差一些，但是从总体的译码性能上看，已经具有相当好的译码性能；从计算复杂度来说，Normalized BP based 译码算法与 UMP BP Based 译码算法的运算量相当，整个译码过程只涉及到加法和比较运算，唯一不同就是 Normalized BP based 译码算法引入了归一化因子 μ ，并且不会带来额外的运算量。Normalized BP based 译码算法与 LLR BP 译码算法的算法主要区别就是 Normalized BP based 译码算法在校验节点更新用 $\mu \cdot \min()$ 函数代替了 $f()$ 函数，从而使算法得以简化，但也是导致了性能上有略微的下降。Normalized BP based 译码算法与 LLR BP 译码算法的硬件实现主要区别是

LLR BP 译码算法的校验节点更新的函数 $f()$ 采用查表法, 增加了存储单元, 而 Normalized BP based 译码算法只需采用移位就能实现校验节点的更新。综合考虑, 本文采用 Normalized BP based 译码算法来进行 LDPC 码译码器的 FPGA 实现设计。

对于 Normalized BP based 译码算法采用不同的归一化因子 μ 译码性能有很大的差异。归一化因子 μ 是一个小于 1 的整数, 可以用第一次迭代校验节点和第二次迭代校验节点输出的消息的期望进行近似推导, 如式(2-45),

$$\mu = \frac{E(|L_2|)}{E(|L_1|)} \quad (2-45)$$

其中 $E(|L_1|)$ 和 $E(|L_2|)$ 分别为第一次迭代校验节点和第二次迭代校验节点输出的期望。归一化因子 μ 的取值与码率、列重和信噪比有关, 为了简化运算, 一般在所有的迭代过程中采用同一值; 但如果想进一步改善译码性能, 可以根据迭代次数和信噪比来取不同的值。图 2.6 是在码长为 2048, 码率 1/2, 最大迭代次数 25, 信噪比在 2dB 下, 不同的归一化因子条件下的译码性能。

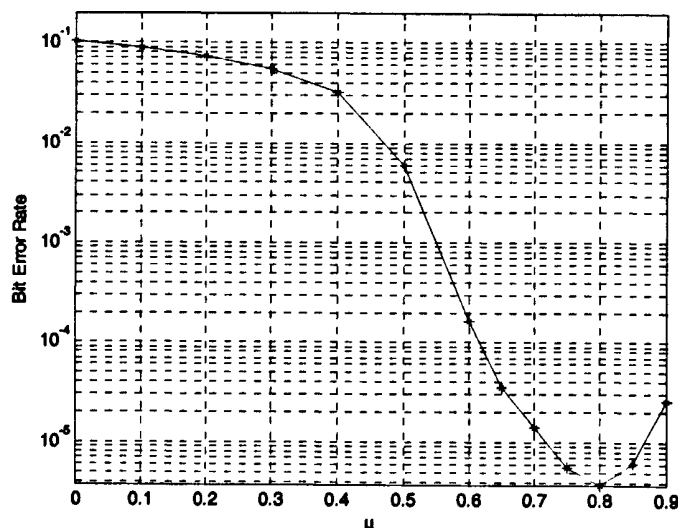


图 2.7 不同的归一化因子对误码率影响

从图 2.6 中可以看出, 当归一化因子在 0.75 到 0.85 之间译码性能较好,

其中在 0.8 时译码性能最佳,但考虑到硬件实现,本设计选择归一化因子 0.75 的 Normalized BP based 译码算法来实现 LDPC 码硬件译码器。

2.3.3 码长的确定

在相同译码算法和迭代次数的条件下,码长越长,LDPC 码的译码性能就越好。但是码长越长,虽然得到更好的译码性能,却增加了 LDPC 码译码器的硬件资源、硬件设计复杂度和译码时间,所以,在实际工程应用中,综合考虑码长所带来的编码增益与硬件资源损耗和译码延时的平衡,需进行折中选择。根据本系统性能要求,选取三种码长进行仿真:1024,1536,2048 作为备选方案,如图 2.7 所示为三种码长在 Normalized BP based 译码算法下的误码率曲线。

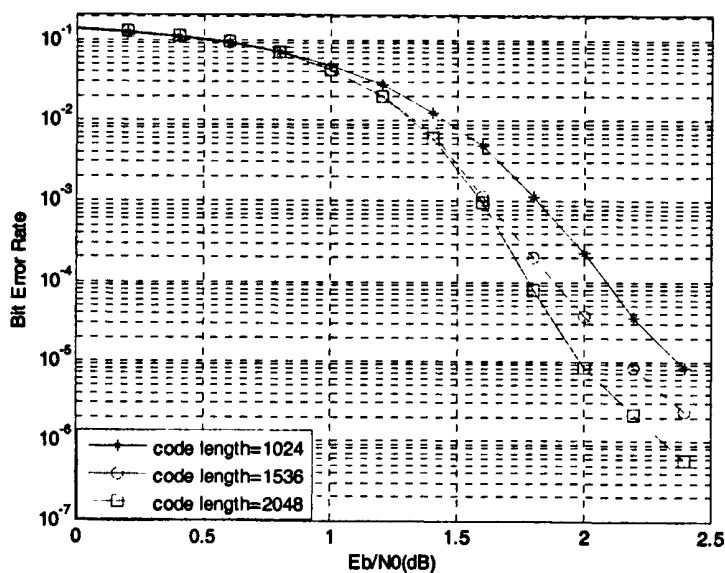


图 2.7 码长 1024、1536 和 2048 的误码率曲线

从图 2.7 可以看出,在归一化信噪比 2dB 下,1536 码长和 2048 码长的误码率在 10^{-4} 以下,满足系统要求。但是考虑到硬件实现,系统性能可能有所损失,因此,本文采用码长 2048 的 LDPC 码来实现硬件译码器。

2.3.4 译码迭代次数的确定

LDPC 码译码的译码性能优越的原因只之一就是采用的消息迭代译码。选择不同的迭代次数对 LDPC 码译码器的性能影响很大,即在相同信噪比下,迭代次数大时,译码算法的纠错性能好;迭代次数小时,纠错性能差。但当迭代次数达到一定值时,译码的纠错性能改善就不会很明显。并且当迭代次数增大时,译码运算时间越长,增加了译码延时,影响了 LDPC 码的译码器数据处理速度。因此,选择合适的迭代次数对于 LDPC 码的译码器非常重要。为了找到一个合适的译码迭代次数,对 Normalized BP based 译码算法进行仿真。如图 2.8 所示为采用非规则的准循环校验矩阵,码长 2048,码率 1/2,调制为 MSK,在高斯白噪声信道下,对应 4 种迭代次数的误码率曲线。

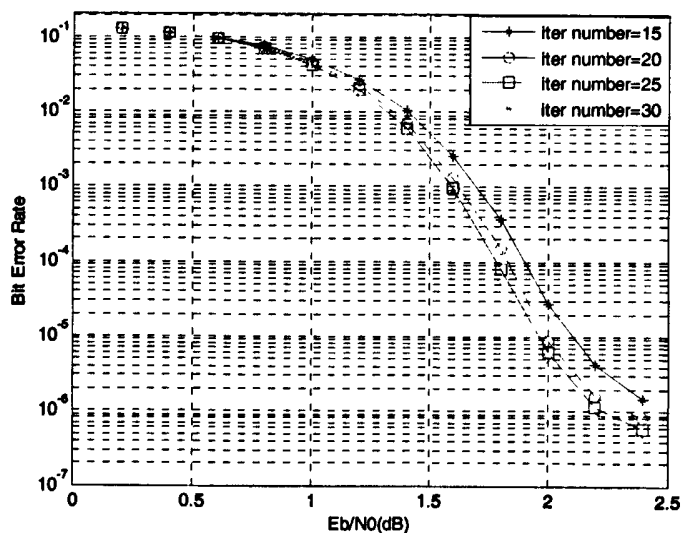


图 2.8 码长 2048 和码率 1/2 的 4 种迭代次数的误码率

从图中可以看出,随着迭代次数的增大,译码性能也随之改善,但当迭代次数增大到一定值以后,译码性能改善不是十分明显。综合考虑到译码器的硬件实现和译码性能以及译码时间,本文采用译码算的迭代次数为 25 次用于译码器的硬件实现。

2.3.5 数据量化方案的确定

量化是从模拟系统到数字系统必备的环节，量化的好坏直接影响了数字系统的性能，因此，选择合适的量化方案对数字系统非常重要。量化就是利用预先规定的有限个电平来表示模拟抽样值的过程^[45]。

量化可以分为均匀量化和非均匀量化。均匀量化就是把输入信号的取值按等距离分割的量化，量化距离(即量化间隔)取决与输入信号的变化范围和量化电平。均匀量化的主要缺点就是不管抽样值大小如何，量化噪声(由于量化误差而引起的信号失真)的均方根值都固定不变。非均匀量化是根据信号的不同区间来确定量化间隔，即小信号的区间，量化间隔也小；反之，量化间隔大。非均匀量化与均匀量化相比，有两个突出的优点：第一、当输入信号具有非均匀分布的概率密度时，输出可以得到较高的平均信号量化噪声功率比；第二、量化噪声功率的均方根值基本与信号抽样值成比例。虽然非均匀量化改善了量化信号，但是以牺牲硬件资源为代价，因此，为了简化硬件资源以及结合系统要求，本系统采用均匀量化。

对于 LDPC 码的量化译码设计，量化主要包括两部分：

- 1.接收信息的量化；
- 2.消息迭代过程中中间变量的量化。

2.3.5.1 接收信息量化方案的确定

由于接收信息的数据范围与消息迭代中的中间变量的数据范围不一致，因此，必须选择合适的量化级数才能满足译码要求。图 2.9 所示的是本系统的通信系统模型，信息序列 $\{u_k\}$ 经过 LDPC 码编码器得到码字序列 $\{c_k\}$ ，然后经过 MSK 调制得到发送序列 $\{x_k\}$ ，再经过高斯白噪声信道 $N(0, \sigma^2)$ ，接收序列 $\{r_k\}$ 经过 MSK 解调得 $\{y_k\}$ ， $\{y_k\}$ 经过量化得到量化接收序列 $\{y_k^{[q]}\}$ ，将量化接收序列 $\{y_k^{[q]}\}$ 送入 LDPC 码译码器，最后输出译码信

息序列 $\{\hat{u}_k\}$ 。系统模型可以表示为

$$y_k = x_k + n_k \quad (2-46)$$

其中 n_k 为噪声抽样值。

对于高斯白噪声信道, LDPC 码的码率为 R , 将每符号传输能量归一化, 则加性高斯白噪声的均方根可以表示为:

$$\sigma = \left(2R \times 10^{\frac{(E_b/N_0)_{dB}}{10}} \right)^{\frac{1}{2}} \quad (2-47)$$

其中 E_b/N_0 为归一化信噪比。

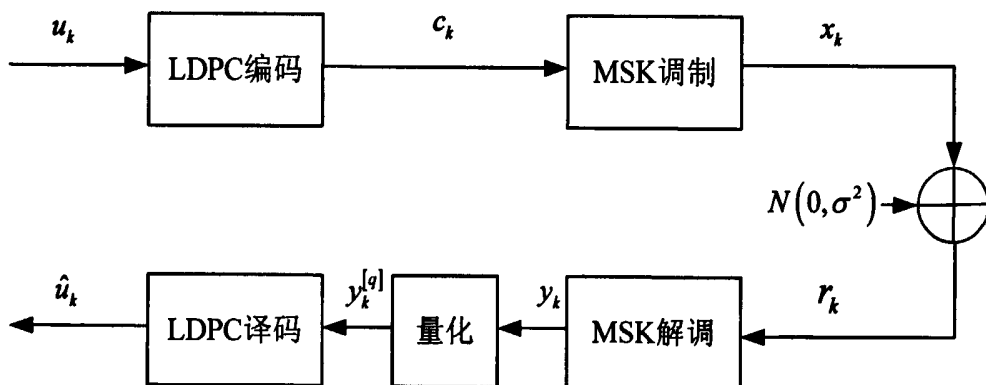


图 2.9 系统模型

对于本系统, 接收信息满足 $y=0$ 的对称分布, 并且是混合高斯分布, 因此可以根据正态分布的“ 3σ 原则”来确定量化范围, 记为 $D=[-d, d], (d>0)$ 。可以求出在量化范围 D 的概率

$$p = 1 - \phi\left(\frac{d-1}{\sigma}\right) - \phi\left(\frac{d+1}{\sigma}\right) \quad (2-48)$$

其中 $\phi(y) = \int_{-\infty}^y \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ 。

图 2.10 为码率 $R=1/2$, 信噪比 0 到 2dB 之间时的接收信息的概率密度曲线, 根据“ 3σ 原则”可以确定量化范围 $D=[-4, 4]$ 。在量化范围确定后,

可以通过改变量化级数来分析量化对译码性能的影响。

图 2.11 所示为码长 2048，码率 1/2 的非规则准循环 LDPC 码接收值采用 4 比特、5 比特、6 比特和 7 比特量化在 Normalized BP based 译码算法下的译码性能曲线。从仿真结果分析，译码器对接收值在 5 比特量化下就能达到系统指标，即信噪比在 2dB 下，译码器误比特率小于 10^{-4} 。

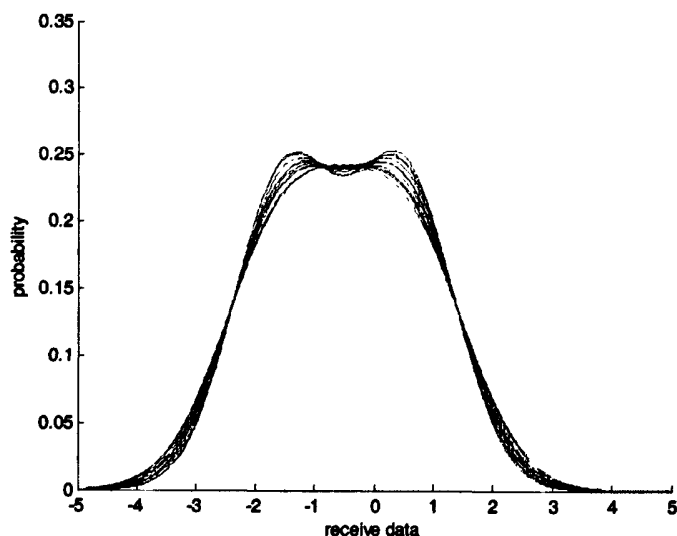


图 2.10 接收信号的概率密度曲线

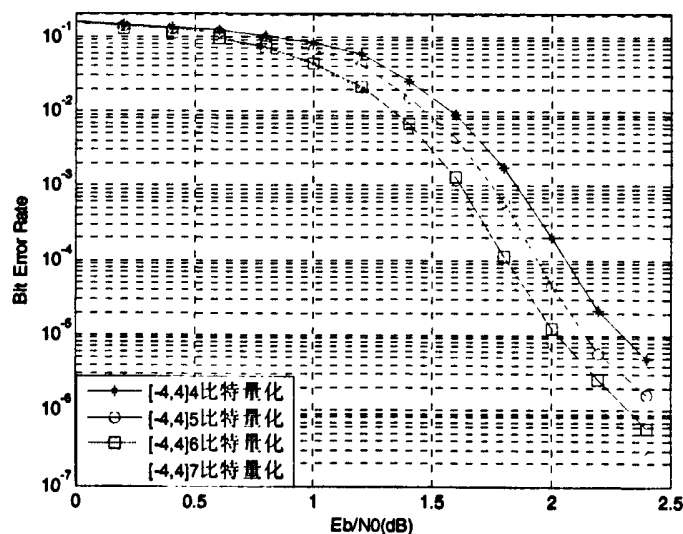


图 2.11 接收信号量化译码性能曲线

2.3.5.2 中间变量信息量化方案的确定

对于 Normalized BP based 译码算法有两个译码中间变量需要量化, 即 $L(q_{ij})$ 和 $L(h_{ji})$, 它们的取值范围都为 $(-\infty, \infty)$ 。中间变量的量化与接收信息量化方法类似, 首先确定量化范围, 然后再确定量化级数。

为了确定量化区间, 首先需要对中间变量的动态范围进行统计, 如图 2.12 和图 2.13 所示为码长 2048, 码率 1/2 的非规则准循环 LDPC 码迭代次数 1-25 次的校验节点信息和变量节点信息的统计特性。

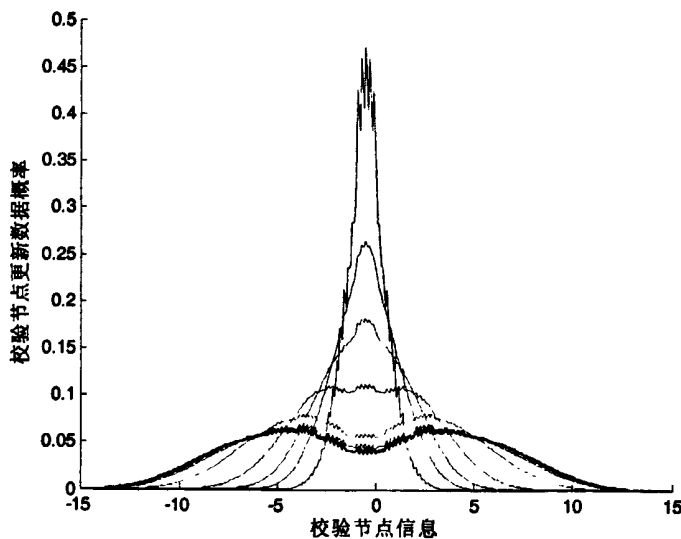


图 2.12 校验节点信息统计特性

由图 2.12 和图 2.13 可知, 校验节点信息和变量节点信息的取值范围随着迭代次数增加而变化。尽管中间变量信息动态范围在 $(-\infty, \infty)$ 之间, 但是从图中看出校验节点信息值 99% 的落在 $[-12, 12]$ 之间、变量节点信息值 99% 的落在 $[-30, 30]$ 之间, 因此校验节点信息选择量化范围为 $[-12, 12]$ 、变量节点信息选择量化范围为 $[-30, 30]$ 就足以满足译码器性能要求。

图 2.14 为码长 2048, 码率 1/2 的非规则准循环 LDPC 码中间变量采用 10 比特量化和未量化在 Normalized BP based 译码算法下的译码性能曲线。从仿真结果分析, 中间变量值采用 10 比特量化引入的量化误差很小, 对译

码性能影响不大，能达到系统指标。

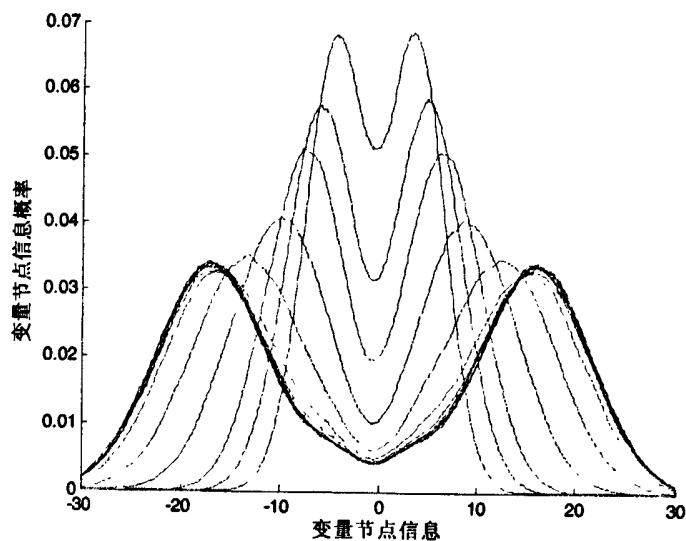


图 2.13 变量节点信息统计特性

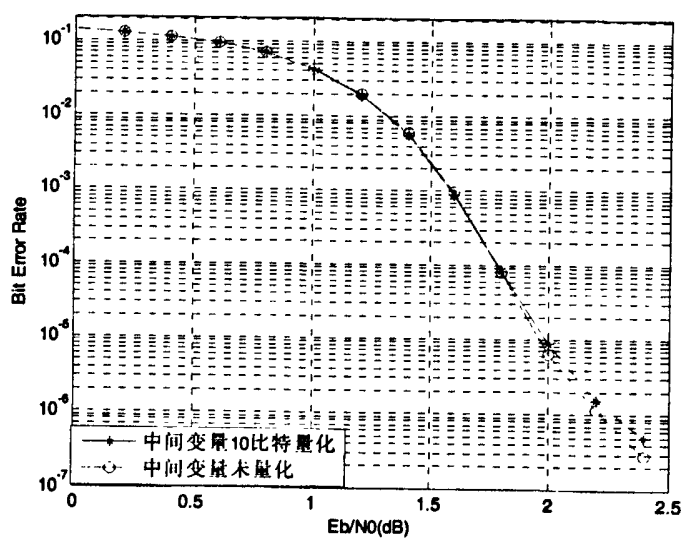


图 2.14 中间变量 10 比特量化与未量化误比特曲线

2.3.6 系统方案确定

根据对系统参数的仿真分析，本文采用码长 2048、码率 1/2、非规则准

循环矩阵、归一化因子 $\mu = 0.75$ 的 Normalized BP based 译码算法、迭代次数为 25 次、接收值和中间变量均用 10 比特量化来设计 LDPC 码的硬件译码器，其中接收值和译码中间变量采用相同量化比特是为了方便译码器的硬件实现。

2.4 本章小结

本章首先简单介绍了 LDPC 码的基本原理，然后详细阐述了 LDPC 码的译码算法，并分析各种译码算法的译码性能和算法复杂度。根据系统要求，通过仿真和分析 LDPC 码的校验矩阵、码长、译码算法和量化方案等影响 LDPC 码译码性能的各个参数，并最终确定了硬件实现的系统参数。

第3章 LDPC 译码器的 FPGA 设计与测试

根据第二章的系统仿真论证, LDPC 码译码器采用 2048 码长的非规则准循环矩阵、10bit 的信息量化和译码迭代 25 次的 Normalized BP based 译码算法实现, 就能满足系统要求。本章将重点介绍 LDPC 码译码器的 FPGA 的设计与测试。

3.1 LDPC 码译码器的 FPGA 设计流程及开发平台

LDPC 码译码器的设计是采用 XILINX 公司的 Virtex-2 Pro 系列中 FPGA 产品中的 XC2OVP30 芯片, 使用 ISE 和 Modlesim 开发工具, 通过 Verilog HDL 硬件描述语言来具体实现。

XILINX 的 Virtex 系列是 FPGA 产品引领现场可编程门阵列行业, 主要用于面向电信基础设施、汽车工业、高端消费电子等应用^[46]。Virtex-2 Pro 系列在 Virtex-2 的基础上增强了嵌入式处理, 内嵌了 PowerPC405 内核, 还包括了先进的主动互联(Active Interconnect)技术, 以解决高性能系统所面临的挑战。此外, 它还增加了高速串行收发器, 提供了千兆以太网的解决方案。其主要特征是: 采用 $0.13\ \mu\text{m}$ 工艺; 核电压为 1.5V, 工作时钟可以达到 420MHz; 支持 20 多种 I/O 接口标注; 增加了 2 个采用高性能 RISC 技术、频率高达 400MHz 的 PowerPC 处理器; 增加多个 3.125Gb/s 速率的 Rocket 串行收发器; 内嵌了多个 18×18 硬核乘法器, 提高了 DSP 处理能力; 具有完全的系统时钟管理功能, 多达 12 个 DCM 模块。XC2OVP30 的主要技术特征如表 3.1 所示。

ISE 是 XILINX 公司推出的 FPGA/CPLD 集成开发环境, 包括了逻辑设计所需的各种功能和简便易用的内置式工具和向导, 使得 I/O 分配、功耗分

表 3.1 XC2OVP30 主要技术特征

型号	Slice 数目	分布式 RAM 容量	块 RAM 容量	PowerPC	专用 乘法 器数 目	DCM 数目	Rocket I/O	最大 可用 I/O 数 目
XC2OVP30	13696	428Kb	2448Kb	2	136	8	8	644

析、时序驱动设计收敛、HDL 仿真等关键步骤变得容易而直观^[46]。ISE 主要功能包括设计输入、综合、仿真、实现、和下载，涵盖了 FPGA 开发的全过程。从功能上讲，可以在不借助任何的第三方 EDA 软件就可以完成 FPGA 开发。

Mentor 公司推出的 Modelsim 仿真软件，具有速度快、精度高并且便于操作，同时可以进行代码分析，可以统计出不同代码消耗资源的情况。由于其主要用于编译和仿真，不能单独编译器件和下载，因此，必须与 ISE 等软件联合使用。

目前采用 FPGA 设计的系统，不再是单一的功能的实现，而是设计多功能系统需要庞大的工作量和很强的工作难度，必须按照一种层次化、结构化的设计方法才能完成系统设计。一般 FPGA 的开发流程如图 3.1 所示，其包括电路功能设计、设计输入、功能仿真、综合、综合后仿真、实现与布局布线、时序仿真与验证、板级仿真与验证以及芯片编程与调试等主要步骤^[46]。

电路功能设计：在系统设计之前，首先要进行的是方案论证、系统设计和 FPGA 芯片选择等准备工作。一般都采用自顶向下的设计方法，把系统分成若干个基本单元，然后把每个基本单元划分为下一层次的基本单元，直到可以直接使用 EDA 元件库为止。

设计输入：将所设计的系统或电路以开发软件要求的某种形式表示出

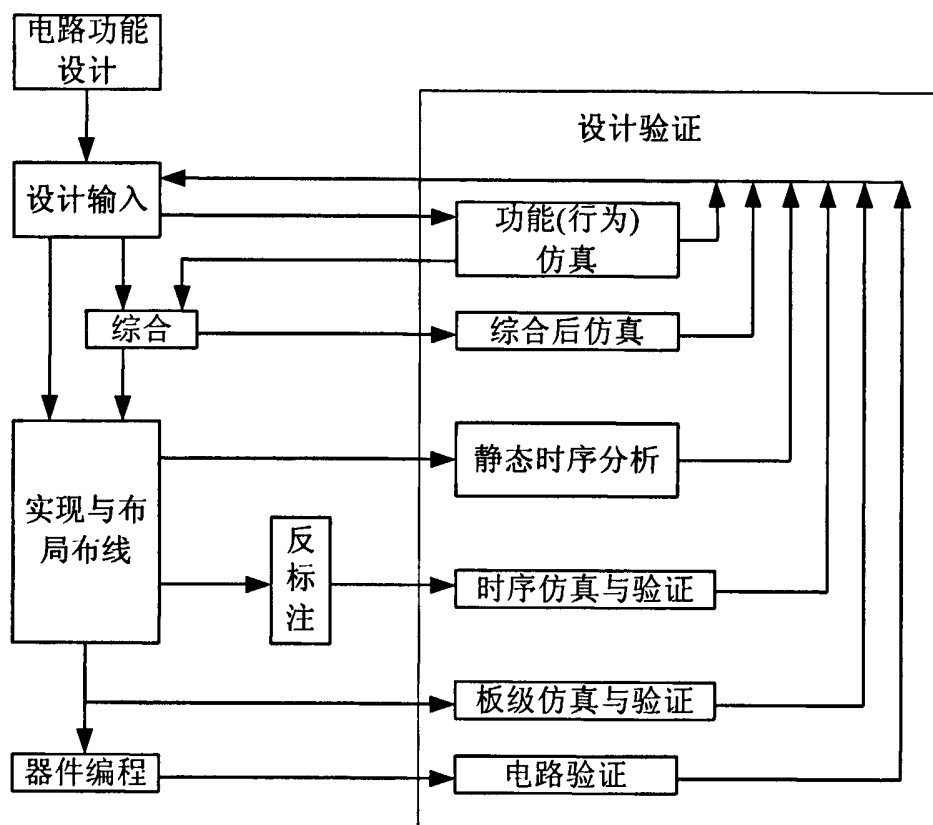


图 3.1 FPGA 开发流程图

来，并输入给 EDA 工具的过程。常用的方法有硬件描述语言(HDL)和原理图输入方式等。本系统采用 Verilog HDL 硬件描述语言实现设计，其优点是便于模块划分和移植，具有强大的逻辑描述和仿真功能，而且输入效率高和语法简练。

功能仿真：在编译之前对设计的电路进行逻辑功能验证。

综合：将较高级抽象层次的描述转化成较低层次的描述。综合优化是将设计输入编译成由与门、或门、非门、RAM、触发器等基本逻辑单元组成的逻辑连接网表，而并非真实的门级电路。

综合后仿真：检查综合结果是否和原来设计一致。

实现与布局布线：实现是将综合生成的逻辑网表配置到具体的 FPGA

芯片上, 布局布线是其中最重要的过程。布局将逻辑网表中的硬件原语与底层单元合理地配置到芯片内部的固有硬件结构上, 并且在速度与面积两者之间合理选择。布线根据布局的拓扑结构, 利用芯片内部的各种连线资源, 合理、正确地连接各个元件。

时序仿真与验证: 将布局布线的延时信息反标注到设计网表中来检测有无时序违规(即不满足时序约束条件或器件固有的时序规则)现象。在布局布线后的时序仿真, 对系统和各个模块进行时序仿真, 时序关系分析, 系统性能估计, 检查和消除竞争冒险非常重要。

板级仿真与验证: 主要用于高速电路设计中, 对高速系统的信号完整性、电磁干扰等特征进行分析。

芯片编程与调试: 设计的最后一步, 将编程数据文件下载到 FPGA 芯片中, 进行调试分析。

3.2 常用 LDPC 译码器的硬件结构

对于 LDPC 码译码器的 FPGA 的设计主要考虑两个方面的问题: 译码器的译码速率和硬件资源的消耗。其实速率和资源损耗是一对矛盾, 即设计高速率译码器要消耗大量的硬件资源。根据不同的应用背景, 对速率和资源消耗有不同的要求, 即如果设计高速率译码器, 必须以大量硬件资源换取; 然而对低速率译码, 可以选择合理的设计来减少资源损耗; 对于中速率译码, 可以是前两者的方案折中。针对不同的情况, LDPC 译码器设计有三种不同硬件结构: 串行结构、全并行结构和部分并行结构^{[47][48][49]}。下面将对这三种结构做简单的介绍。

3.2.1 串行结构

对译码速率要求不高的场合, 其硬件设计思想是在满足系统对速率的最低要求下, 应尽量节省硬件资源的消耗, 而 LDPC 码的译码器的串行结构就是

基于这种思想而产生的。由LDPC码的校验矩阵的Tanner图表示，可以知道其译码过程主要是在校验节点与变量节点之间进行信息传递来完成。校验节点决定了校验方程在某比特取0或者1、其他比特取先验概率情况下，校验方程成立的概率；变量节点是在该点先验概率和其他校验节点传递的信息来共同决定某信息比特取1或者0的概率。如图3.2所示为译码器串行结构，其中 $L(c)$ 为初始化信息， $L(q)$ 为变量节点向校验节点传递的信息， $L(h)$ 为校验节点向变量节点传递的信息， $L(e)$ 为译码结果，CNU(Check Node processing Unit)为校验节点处理单元，VNU(Variable Node processing Unit)为变量节点处理单元。

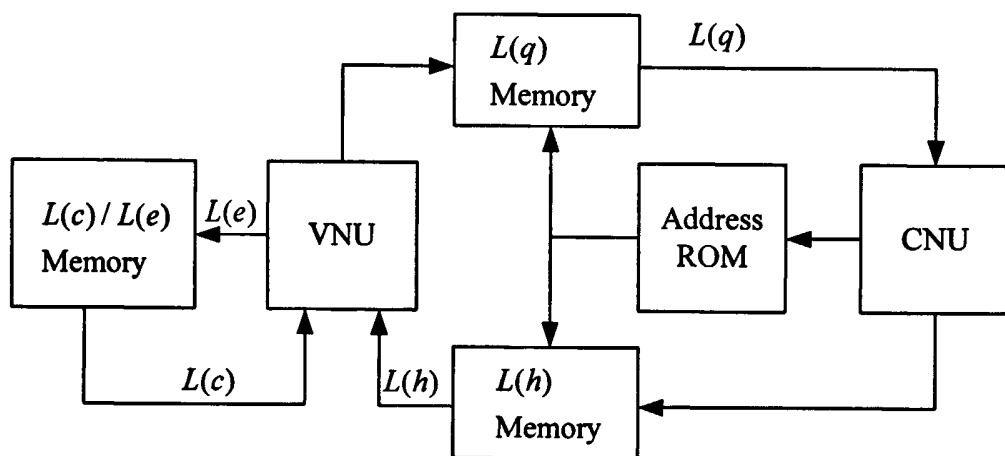


图 3.2 LDPC 码的译码器串行结构

在串行译码过程中，CNU 每次需要 w_r 个 $L(q)$ ，通过计算得到 w_r 个 $L(h)$ 存到 $L(h)$ Memory 中，经过 M 次完成校验节点更新；VNU 每次需要 w_c 个 $L(h)$ 和对应比特的 $L(c)$ ，通过计算得到 w_c 个 $L(q)$ 存到 $L(q)$ Memory 中，经过 N 次完成变量节点更新，其中 M 和 N 分别代表校验矩阵的行数和列数。CNU 的读写地址必须满足校验矩阵的 Tanner 图的特殊结构，根据 Tanner 图预先计算，然后存入 Address ROM 中，对于 VNU 的读写地址简化为从 1 到 N ，可以用一般计数器实现。

串行译码结构主要有两个优点：

1. 占用资源少；

2.设计灵活；即在校验矩阵的维数、行重和列重相同的情况下，只需改变 ROM 地址就可以满足不同的 LDPC 码。

串行译码结构的缺点是译码速率不能太高，一般在 10Mbps 以下。影响串行译码结构译码速率的主要两个因素是译码器时钟和译码迭代次数，因此可以合理的调整译码器的时钟和迭代次数来提高译码速率。但是也存在的缺点：

- 1.提高译码时钟，给译码器时序设计带来了困难；
- 2.减少译码迭代次数，会使译码性能下降。

综合各方面因素，串行译码结构只适用以低速译码，而不适用于高速译码。

3.2.2 全并行结构

LDPC 码译码器的全并行结构是将其校验矩阵的 Tanner 图直接映射为硬件结构。如图 3.3 所示为 (12,3,6) 规则 LDPC 码译码器的全并行结构。

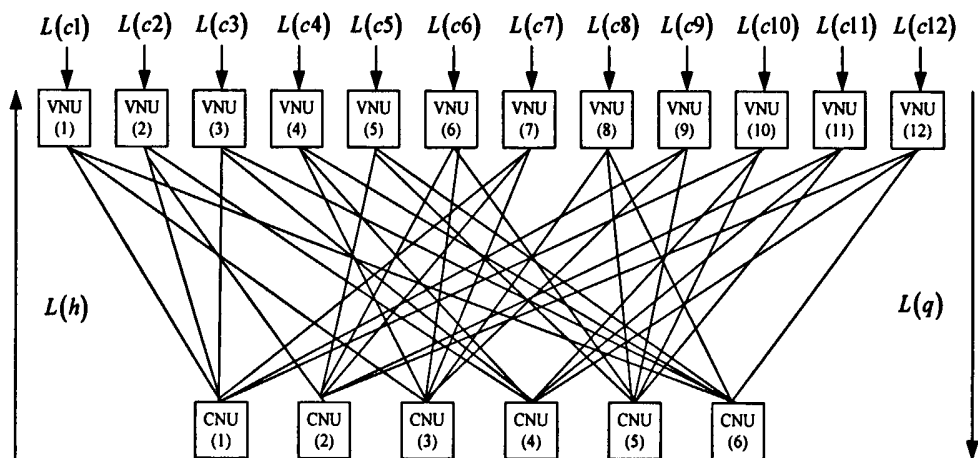


图 3.3 LDPC 码的全并行译码结构

LDPC 码译码器采用全并行结构的最大优点是可实现高速率译码，文献[50]中采用全并行译码结构以时钟频率为 64MHz 实现了码长 1024、码率

1/2、译码速率达到 1Gbit/s。

全并行译码结构缺点是：

1.消耗资源大。例如对 $M \times N$ 的校验矩阵需要 M 个 CNU 和 N 个 VNU，当校验矩阵维数较大时，将会损耗大量的资源。

2.时序控制复杂和信号同步困难。根据两类节点之间连线 (*message_edge*) 数目与 Tanner 图中边 (*graph_wire*) 的个数、 $L(q)$ 和 $L(h)$ 的比特位宽 ($L(q)_width$ 和 $L(h)_width$) 的关系，可以得：

$$message_wire = graph_edge \times (L(q)_wire + L(h)_wire) \quad (3-1)$$

当校验矩阵的维数较大时，相应的 CNU 和 VNU 的数量增加，导致 CNU 与 VNU 的连线急剧增加，要满足 CNU 和 VNU 之间数据信号传输的同步就非常困难，因此必须增加复杂的时序控制电路来实现同步，但将会消耗更多的硬件资源。

3.设计不灵活。由于全并行译码结构完全从校验矩阵的 Tanner 图映射过来，因此，一旦校验矩阵确定，译码器也就确定。

综合各方面因素，只有在对译码速率要求较高的情况下，才采用全并行译码结构。

3.2.3 部分并行结构

部分并行译码结构是对全并行译码结构和串行译码结构的一种改进方案，即避免了全并行译码结构的资源消耗大、硬件实现难度大和串行译码结构的译码速率低的缺点。如图 3.4 为部分并行译码结构，图中有 K 个 CNU 和 L 个 VNU 运算单元。通过复用 CNU 和 VNU 来提高译码速率，与串行译码结构相比，可以提高 $L \times K$ 倍。从本质上讲，串行译码结构和并行译码结构是部分并行译码结构的特例，即 $K=1$ 和 $L=1$ 对应于串行译码结构， $K=M$ 和 $L=N$ 对应于并行译码结构。部分并行译码结构可以根据系统要求，灵活的选择 K 和 L 的值来设计译码器。如图 3.4 所示 $L(q)$ 和 $L(h)$ 的

Memory, 为了进一步降低硬件资源的消耗, 可以将 $L(q)$ 和 $L(h)$ 共用一个 Memory, 实现存储资源减半。文献[51]采用 18 个 CNU 和 36 个 VNU 的部分并行译码结构实现了码长 9216、码率 1/2, 译码速率达到 27Mbps 的 LDPC 码译码器。

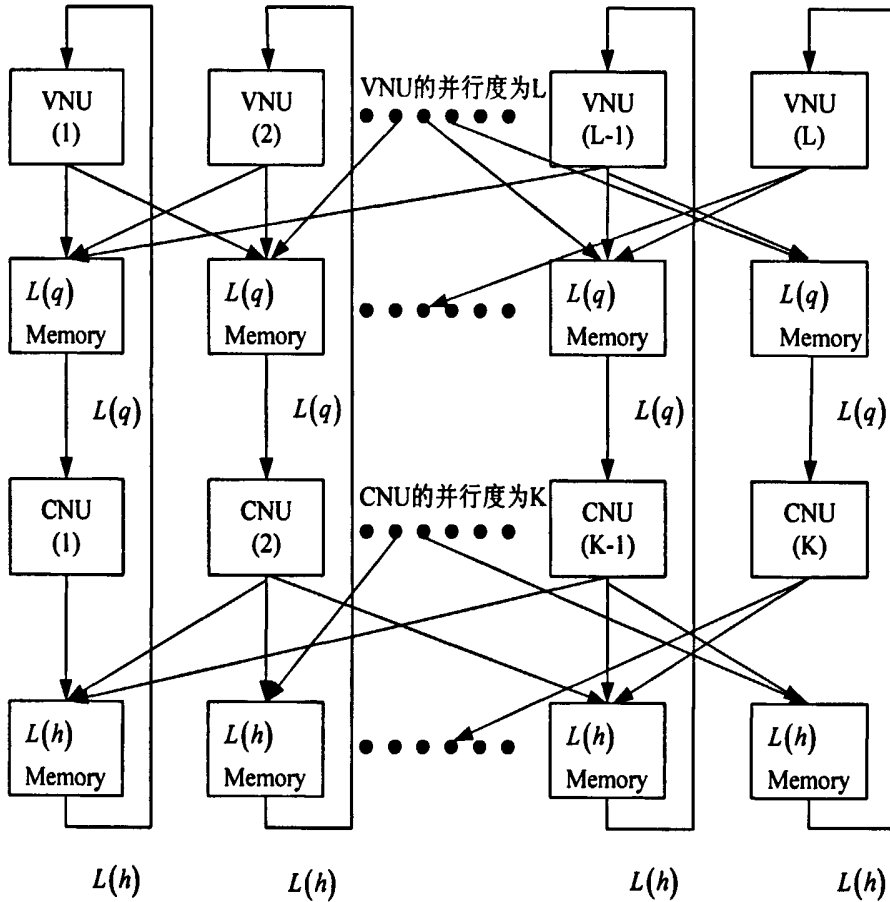


图 3.4 部分并行译码结构

部分并行译码结构的优点是：

1. 可以实现较高译码器速率；
2. 硬件资源较少；
3. 设计灵活。

综合考虑, 本系统采用部分并行译码结构来设计 LDPC 译码器。

3.3 LDPC 译码器的 FPGA 设计

由第二章可知, LDPC 码译码器采用具有下三角结构的非规则准循环的校验矩阵、码长为 2048、码率为 1/2、归一化因子 $\mu = 0.75$ 的 Normalized BP based 译码算法、迭代次数为 25 和 10bit 数据量化的部分并行译码结构的 FPGA 来实现。接下来详细介绍本系统 LDPC 码译码器的 FPGA 设计。

3.3.1 LDPC 译码器的整体结构

LDPC 码的译码器设计是根据数据流设计的, 即首先接收量化比特信息, 其次是接收信息进行校验节点更新, 然后将校验节点更新信息与接收信息同时参与到变量节点更新, 接着再将变量节点更新信息返回到校验节点更新, 这样反复在校验节点与变量节点之间进行信息迭代更新, 达到最大迭代次数, 才停止译码, 输出译码信息。如图 3.5 所示为 LDPC 码译码器的数据流程图。

LDPC 码译码器主要由接收信息缓冲模块、串并转换模块、时序控制模块、初始信息存储阵列、译码中间信息存储阵列、校验节点信息处理模块、变量节点信息处理模块、译码信息缓存模块、并串转换模块组成, 如图 3.6 所示为译码器整体结构图。译码器的译码步骤如下:

- 1.接收比特信息速率转换。由于接收信息速率与译码器中信息处理速率不一致, 在本设计中, 译码器信息处理时钟远高于接收信息时钟, 因此需要将接收信息进行速率转换, 由接收比特信息缓冲模块实现。

- 2.用串并转换来实现初始信息的分块存储。根据非规则循环校验矩阵的结构特点来划分部分并行译码器的译码结构, 即将信息分块处理实现部分并行译码。

- 3.从译码中间信息存储阵列中取出数据进行校验节点信息处理, 其中第一次校验节点信息处理, 是来自初始信息; 其后的校验节点信息处理是来

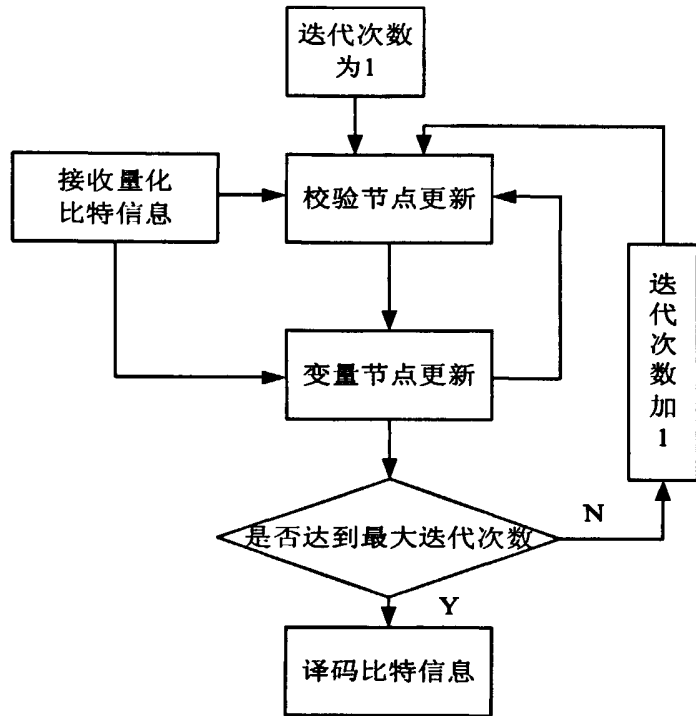


图 3.5 LDPC 码译码器的数据流程图

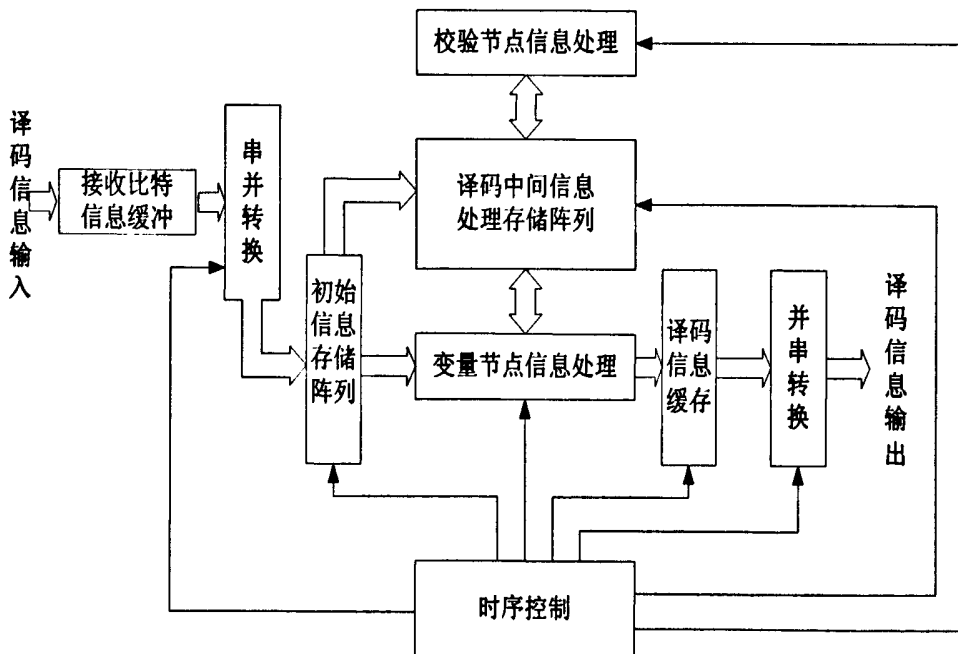


图 3.6 译码器整体结构图

自上次变量节点更新的信息。

4.从译码中间信息存储阵列和初始信息存储阵列中取出数据进行变量节点信息处理，其中变量节点信息处理包括变量节点的更新和译码判决。变量节点的更新信息返回译码中间信息存储阵列，提供下一次校验节点更新使用；而译码判决信息，虽然每次都进行判决，但不输出译码信息，到达最大迭代次数，输出译码信息，停止译码。

5.并串转换输出译码信息。通过译码判决后的信息被缓存在译码信息缓存模块中，而并串转换模块实现信息复接，复接后的信息为码字信息。

可以看出整个译码器的数据流由时序控制模块严格控制，因此译码器设计成功与否主要取决于时序的设计。

3.3.2 LDPC 译码器的时序设计

LDPC 译码器的时序控制模块控制着译码器内各模块的执行顺序，即什么时候开始译码，什么时候开始变量节点更新，什么时候开始校验节点更新，什么时候译码结束等，使译码器内部执行清晰明了。时序控制是整个译码器的核心模块，决定着译码器的设计成功与否，因此，在设计译码器其他模块之前，必须设计好整个译码器的执行时序电路。设计一个好的时序电路会有效改善整个设计电路和节约设计时间。

由于 LDPC 码的译码算法中信息之间进行迭代更新，并且每次迭代都有大量的信息需要处理，因此译码器的控制时序比较复杂。根据译码器的数据处理特点，可以将译码器的时序控制分为译码器主控制、初始化控制、校验节点更新控制、变量节点更新控制，如图 3.7 所示为译码器时序控制结构框图。从图中可以看出，译码器主控制作为译码器的核心控制时序，控制着初始化控制、校验节点更新控制和变量节点更新控制，并且可以根据通信系统的实际需要设置译码器的迭代次数，对通信环境不好和对误码率要求比较高的情况，设置大的迭代次数；反之，设置相对小的迭代次数，

从而达到可配置的迭代次数的译码器。

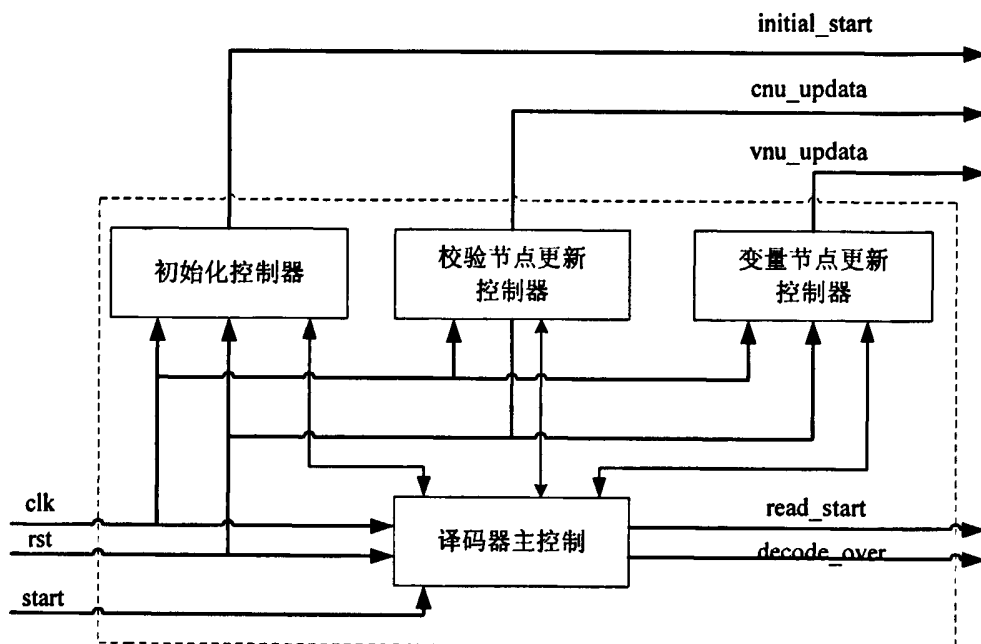


图 3.7 译码器时序控制框图

译码器主控制主要是负责译码器启动、复位、初始化启动、校验节点更新启动、变量节点更新启动、译码结果输出和译码结束，译码器主控制的状态转换如图 3.8 所示。

译码器主控制由 **start** 信号启动，即标志着整个译码器被启动。当译码器主控制被启动后，首先进入写状态，即将接收缓冲比特信息从缓冲模块读入到译码器内部。在从缓冲模块到译码器内部数据传递过程中，经过串并转换模块，数据以校验矩阵的结构分块存入到初始信息存储阵列，数据写入完成后，向主控制返回结束标志。接着主控制进入初始化状态，此状态主要完成的任务是将初始信息存储阵列中的初始信息转入到译码中间信息存储阵列，准备进行译码信息处理。在初始化结束后，输出结束标志，通知主控制，准备进入下一步。当一切准备就绪之后，译码器主控制开始启动数据迭代更新，数据迭代更新包括两个部分即校验节点数据更新和变量节点数据更新。根据译码算法的数据流程，首先进行检验节点数据更新，

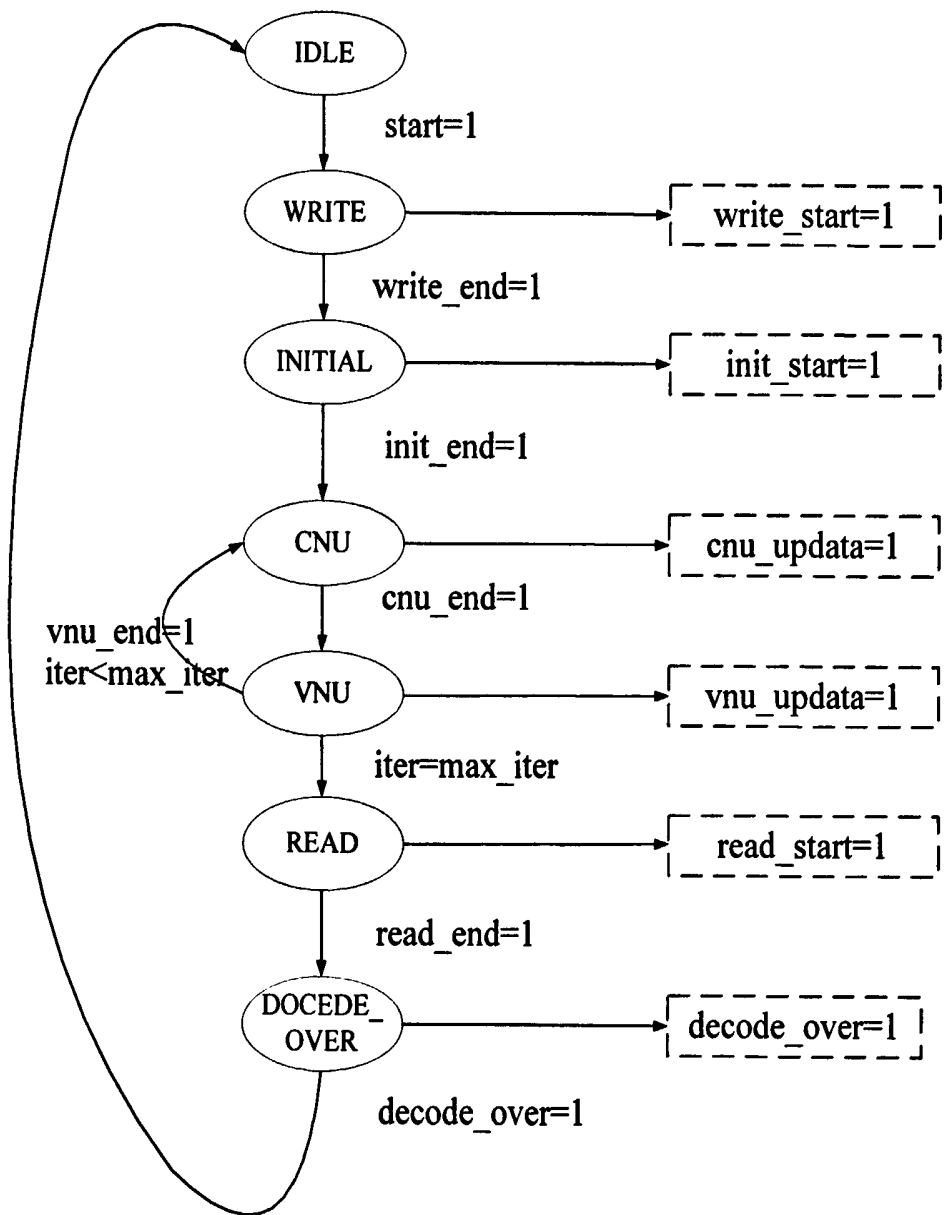


图 3.8 译码器主控制状态图

再进行变量节点数据更新，然后再到校验节点数据更新，就这样交替更新处理，即迭代，当达到最大迭代次数之后，译码器结束数据处理，并将译码信息存入译码信息缓存模块。主控制在译码信息存入译码信息缓存模块后，启动读状态，将译码信息输出。当主控接收到读结束返回标志，表明

译码器译码功能完成，准备结束，输出译码结束标志，译码器进入空闲状态，准备下一次译码开始。译码器主控制时序仿真如图 3.9 所示。

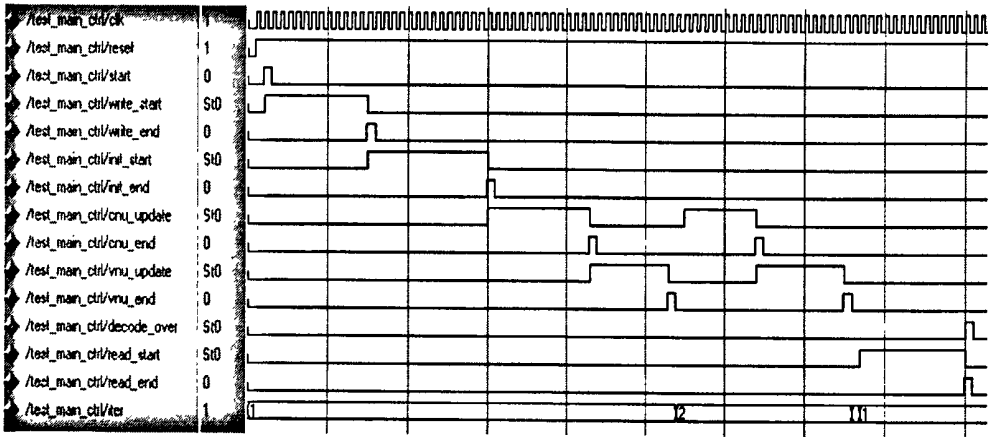


图 3.9 译码器主控制时序仿真

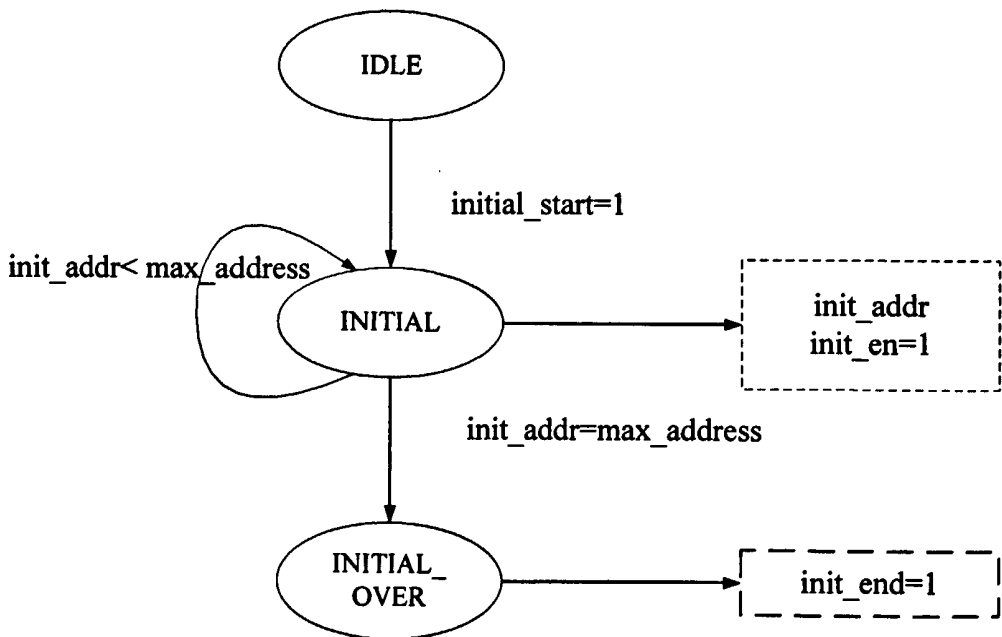


图 3.10 初始化控制状态图

初始化控制是为初始化提供初始化地址和初始化使能。初始化控制的状态图如图 3.10 所示。当译码器完成接收信息从接收信息缓冲模块到初始

化信息存储阵列之后,由译码主控制启动初始化控制。当初始化被启动后,开始输出初始化地址和初始化使能,控制初始信息存储阵列的信息导入译码中间信息处理阵列,在数据全部导入后,初始化结束,输出初始化结束标志。如图 3.11 所示初始化控制时序仿真。

由于译码器中校验节点更新和变量节点更新与译码器采用的校验矩阵密切相关,校验矩阵非零元素的分布影响着数据更新的复杂度。尽管译码器所采用的校验矩阵结构性很强,但是非零元素的比较稀疏,为了更好的进行校验节点更新和变量节点更新,必须有严格校验节点更新控制和变量节点更新控制来进行时序控制。

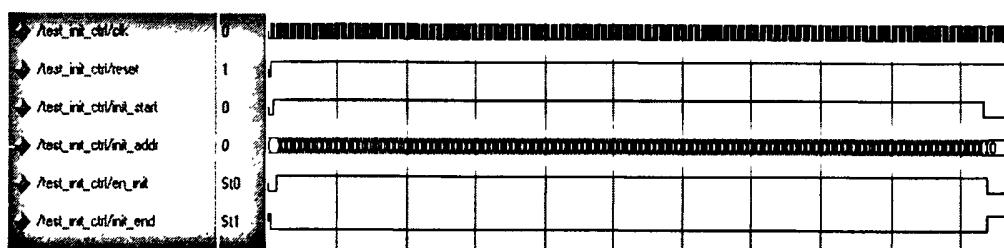


图 3.11 初始化控制时序仿真

校验节点更新控制如图 3.12 所示。当校验节点更新控制被主控制启动后,直接进入校验节点更新地址状态,地址状态为校验节点信息处理提供地址使能,用于启动地址计数地址器(根据校验矩阵结构设计进行校验节点更新的地址器)读取译码信息处理阵列的数据进行校验节点更新。由于校验节点更新是基于校验方程的,即每进行一次校验节点更新,校验方程中某节点的信息更新由其他节点的信息决定。每一次校验节点更新针对于每一个校验方程,每个校验方程的变量节点数对应于校验矩阵中某行的行重,由于译码器采用部分并行译码器结构,因此每个校验方程中的节点更新是同时完成的,相当于译码器的速度提高了 w_r 倍。数据同时进行更新,对时序要求严格,因此为了方便时序电路的设计,将校验方程节点信息存到不同的中间信息存储 RAM 中。读使能启动后,校验节点更新数据同时从 RAM 中取出;然后送入更新运算单元;运算结束后再与存入 RAM,在数据更新

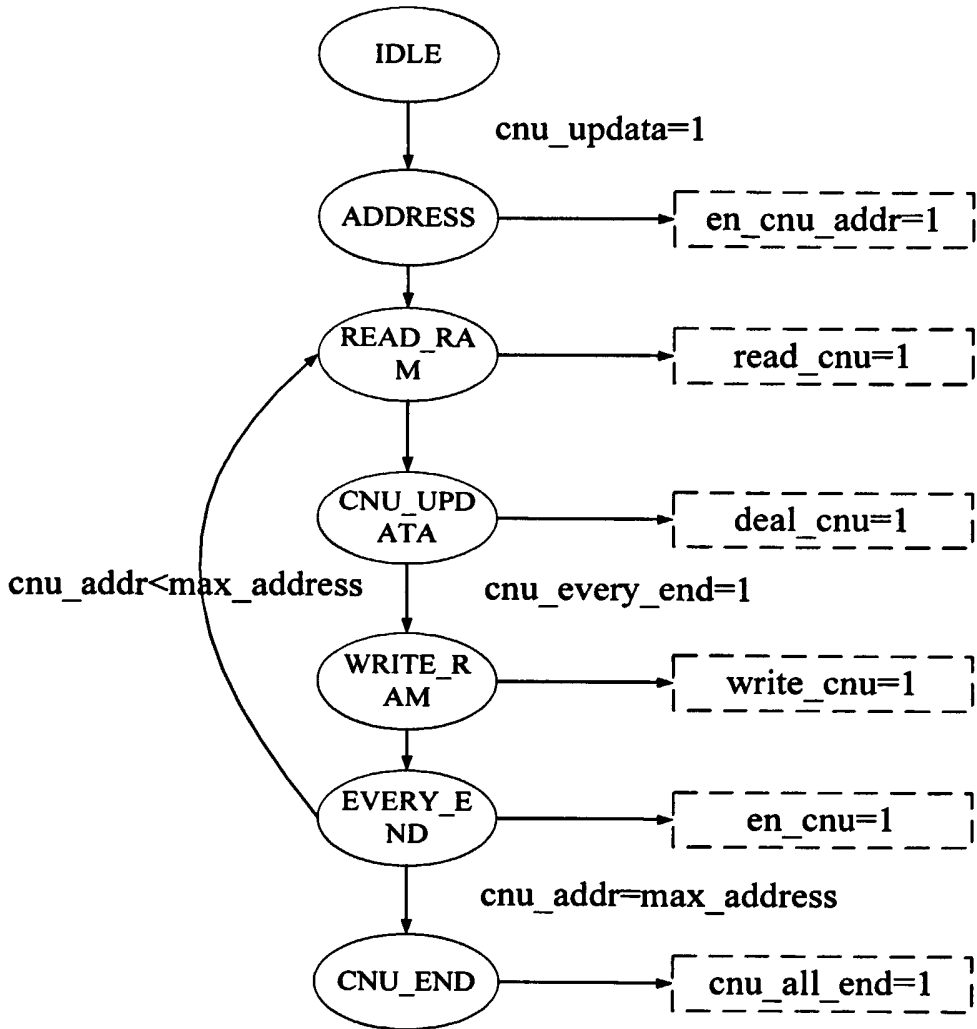


图 3.12 校验节点信息更新控制状态图

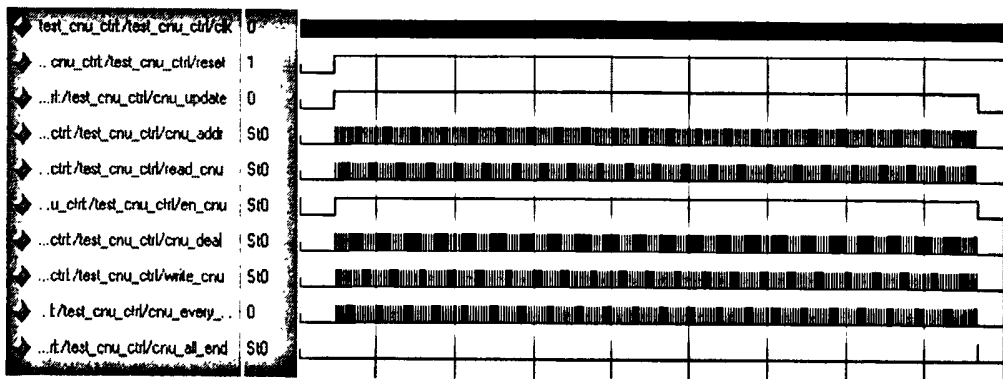


图 3.13 校验节点更新控制时序仿真图

过程中,保持 RAM 地址不变,每次译码迭代更新运算次数由校验矩阵中单位循环子矩阵的 p 决定。当 RAM 地址到 p 最大值,译码器结束 1 次迭代校验更新,输出更新结束标志,校验节点更新控制时序仿真如图 3.13 所示。

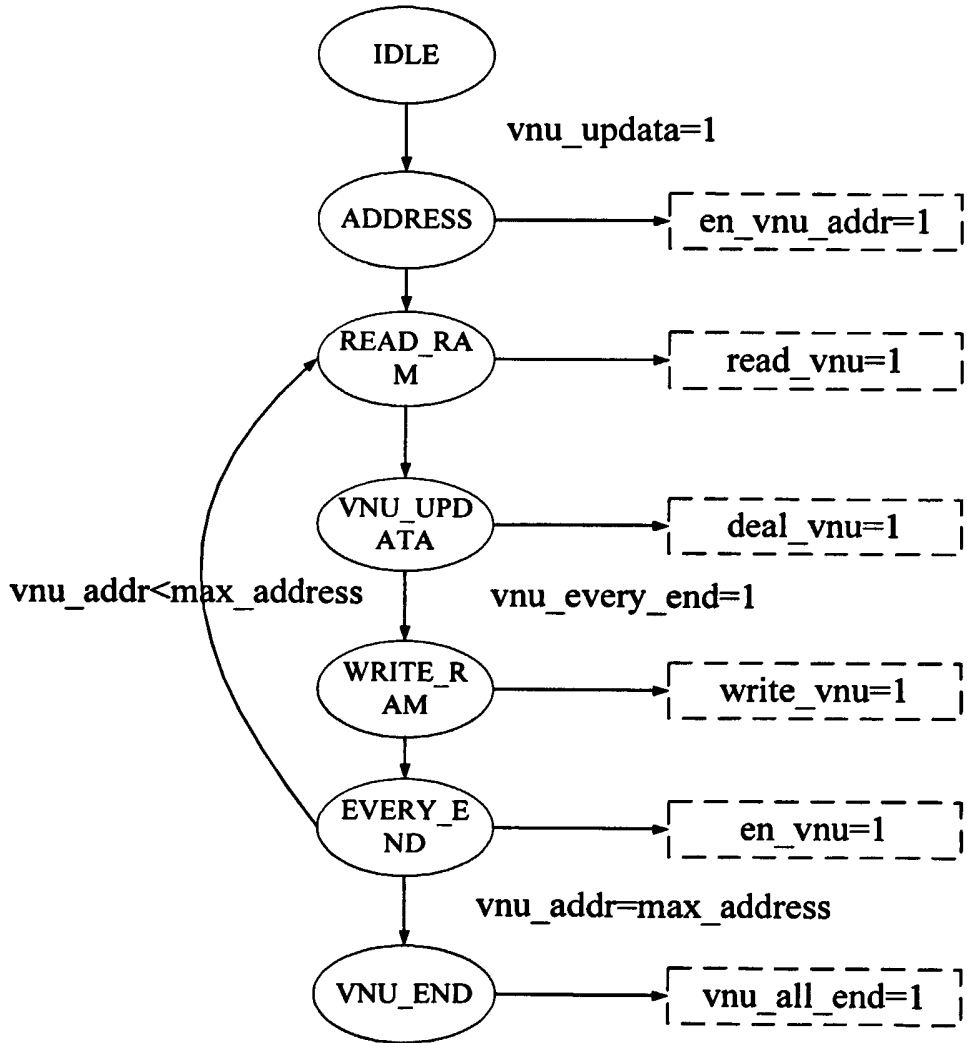


图 3.14 变量节点更新控制状态图

变量节点更新控制如图 3.14 所示。每一次译码校验节点更新结束之后,译码器进入变量节点更新。变量节点更新控制流程基本与校验节点更新控制器一致,译码器先将变量节点信息从中间信息存储阵 RAM 中读出,然后进入更新处理单元,保持 RAM 地址不变将更新数据存回 RAM,变量节点

更新控制时序仿真如图 3.15 所示。

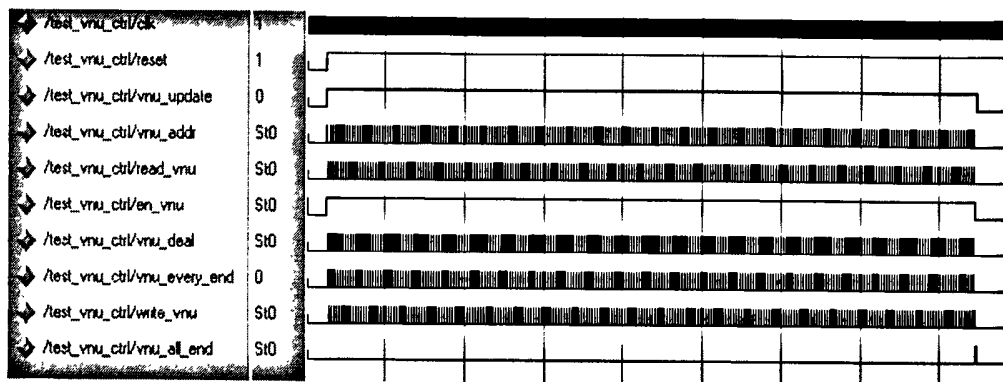


图 3.15 变量节点更新控制时序仿真

3.3.3 LDPC 译码器存储阵列

LDPC 码译码器的存储阵列包括三部分：初始信息存储阵列、译码中间信息存储阵列和译码信息缓存。

初始信息存储负责存储译码器接收初始信息，对于校验矩阵的循环子矩阵维数为 $p \times p$ 和码长为 N 的 LDPC 码，将接收初始信息等分成 N/p 段，分别存入到 N/p 块初始信息存储阵列中，初始信息存储如图 3.16 所示。由于每次变量节点更新都用到初始信息，因此每一帧译码初始信息在译码迭代更新过程中保持不变。



图 3.16 初始信息存储阵列

根据第 2 章可知，译码器所采用校验矩阵由相同大小的单位阵、单位循环阵和全零阵组成。考虑到译码器硬件资源的损耗，因此译码中间信息存储阵列只存储校验矩阵中非零元素的位置信息。根据校验矩阵的结构特点，将译码信息分块存储，存储器的数量取决于单位阵和单位循环阵的数

量, 存储器的深度取决于子矩阵的 p , 译码中间信息存储阵列如图 3.17 所示。

RAM_11	RAM_12
RAM_21	RAM_22
\vdots	...	\ddots	$RAM_{(M-1)L}$
RAM_M1	RAM_M2	...	RAM_ML

图 3.17 译码器中间信息存储阵列

译码器中间信息存储阵列需要存储变量节点更新信息和校验节点更新信息, 如果两种信息分别存储, 存储器的数量消耗很大。为了减少存储资源的消耗, 将变量节点更新信息和校验节点更新信息的存储器复用。根据译码数据更新流程, 即先校验节点更新再变量节点更新, 将译码中间信息交替存储, 采用双口 RAM 组成存储阵列。对于每个 RAM 有两个地址输入端口: 一个写端口和一个读端口, 其中写端口由初始化地址、校验节点更新地址和变量节点更新地址组成, 在时序控制下用三输入数据选择器选择合适的地址, 将初始信息、校验节点更新信息和变量节点更新信息写入 RAM; 读端口由校验节点更新地址和变量节点更新地址组成, 根据时序要求选择相应地址从 RAM 中读出数据进行校验节点更新和变量节点更新。

对于译码器中间信息存储阵列最为关键是校验节点更新地址器和变量节点更新地址器的设计, 地址器设计合理将会节省大量硬件资源。根据 LDPC 码校验矩阵的结构特性, 校验节点更新地址器和变量节点更新地址器都采用 p 进制的计数器实现, 其中校验节点更新地址器的初值为循环移位子矩阵的移位系数, 变量节点更新地址器的初值为 0。校验节点更新地址器和变量节点更新地址器如图 3.18 和 3.19 所示, 其中 load 11、load 12 等为校验节点更新地址器初值, 即循环子矩阵的移位系数。

译码信息缓存负责存储译码器判决信息。与初始信息存储一样, N/p 段译码信息, 存储信息次序与初始存储信息一一对应, 存储器深度都是 p 。

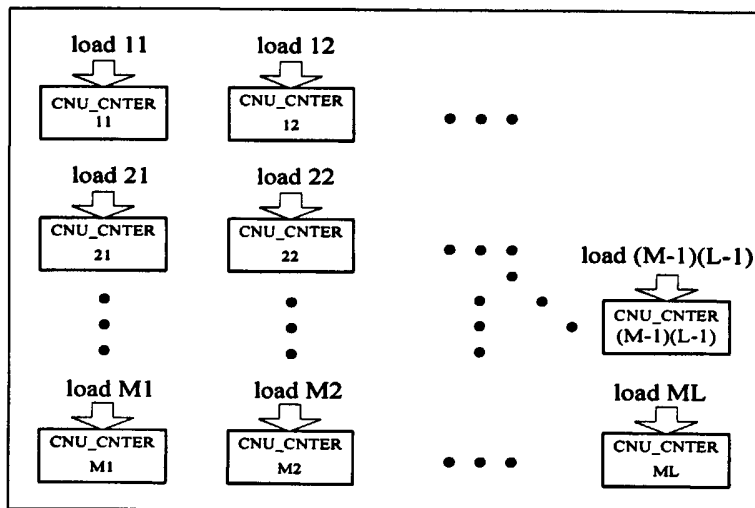


图 3.18 校验节点更新地址器

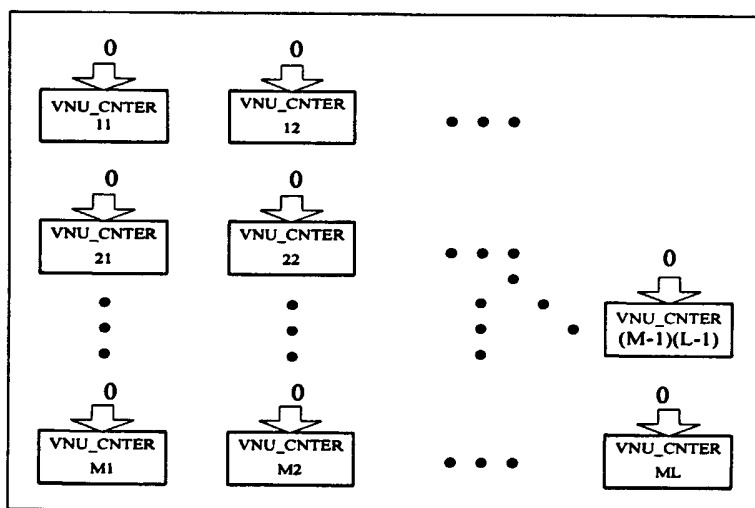


图 3.19 变量节点更新地址器

3.3.4 LDPC 译码器信息初始化

由 Normalized BP based 译码算法可知，译码在第一次校验节点更新必

须使用接收的初始信息,因此,设计译码器必须在译码器进行信息更新之前,将接收初始信息导入到译码器中间信息存储阵列,这一过程就是译码器的信息初始化。硬件实现的初始化就是将初始信息存储阵列 IRAM_X 信息存入到译码器中间变量存储阵列 RAM_X1、RAM_X2……RAM_XM 中。

3.3.5 LDPC 译码器信息迭代更新

LDPC 译码器的信息迭代更新包括校验节点更新和变量节点更新。设 LDPC 码的校验矩阵有 $M \times L$ 个 $p \times p$ 的子矩阵构成,即校验矩阵中的每一行包括 L 个子矩阵,每一列包括 M 个子矩阵。由上文可知,本系统采用部分并行译码结构,因此,译码器设计采用 M 个校验节点更新单元(CNU)和 L 个变量节点更新单元(VNU)。

译码器的校验节点更新和变量节点更新是根据 Normalized BP based 译码算法而设计的,校验节点更新单元结构和变量节点更新单元结构如图 3.20 和图 3.22 所示。

图 3.20 可以看到校验节点更新首先将输入的数据进行分离(即绝对值和符号分离),然后单独对绝对值和符号进行处理,最后将处理好的绝对值部分和处理完的符号部分合并成完整的数据输出。绝对值部分先做最小值运算,用比较器实现;然后将得到的最小值与归一化因子 μ 相乘,用移位实现。符号部分直接进行异或。在校验节点更新的整个过程都要保持 RAM 地址不变,这样保证更新的数据存储到原位置。图 3.21 为校验节点更新单元经过 Modelsim 的时序仿真。

图 3.21 中所示的 data_in1、data_in2、data_in3、data_in4、data_in5 为输入校验更新信息, data_out1、data_out2、data_out3、data_out4、data_out5 为校验更新结果信息, cnu_end1 为更新结束标志位,当为高电平时,表示更新结束。从校验节点更新单元的时序仿真结果看,校验节点更新单元设计正确。

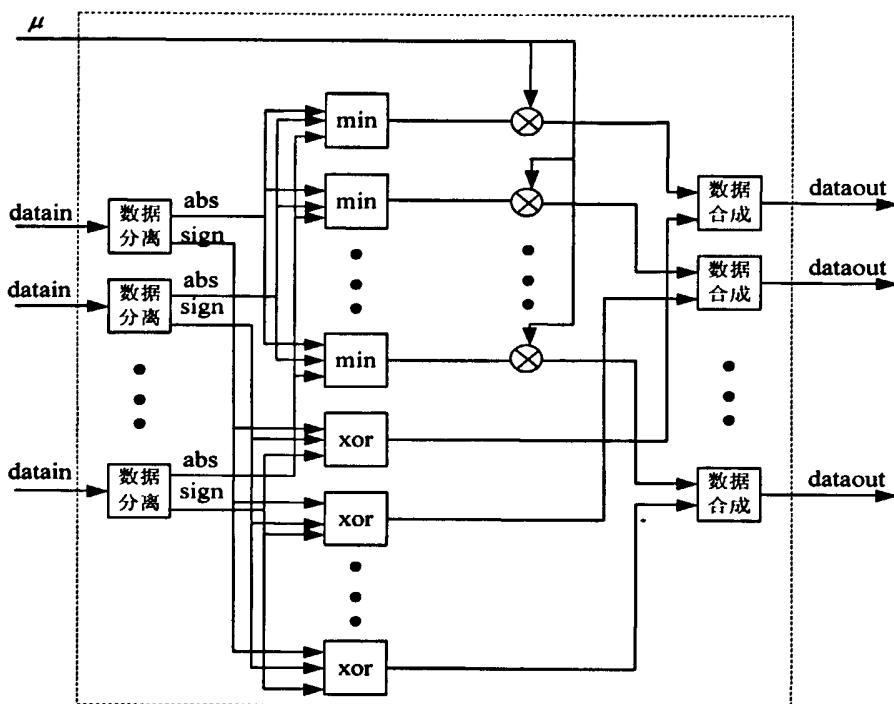


图 3.20 校验节点更新单元结构

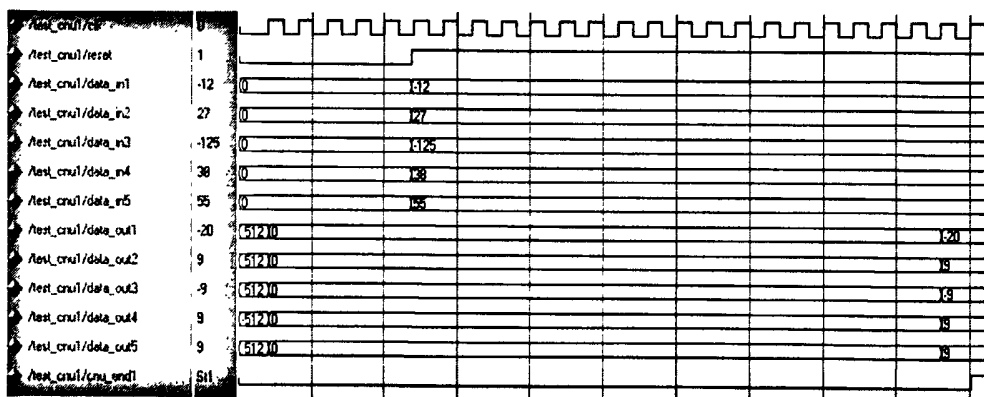


图 3.21 校验节点更新单元时序仿真

图 3.22 可以看到变量节点更新单元包括两部分：一是变量节点更新；二是译码判决。变量节点更新首先对输入数据求和，然后用所得到的和减去相应的输入数据得到更新的变量信息数据输出。将变量更新时输入数据所求的和做判决(即取数据的符号位)，输出的 1bit 的数据就是译码信息，存

入到译码信息缓存中。同样在变量节点更新的整个过程都要保持 RAM 地址不变，保证更新的数据存储到原位置。变量节点更新单元经过 Modelsim 的时序仿真如图 3.23 所示。在每次迭代更新都进行译码判决，但是只有达到最大迭代次数后，才输出最后的译码信息。

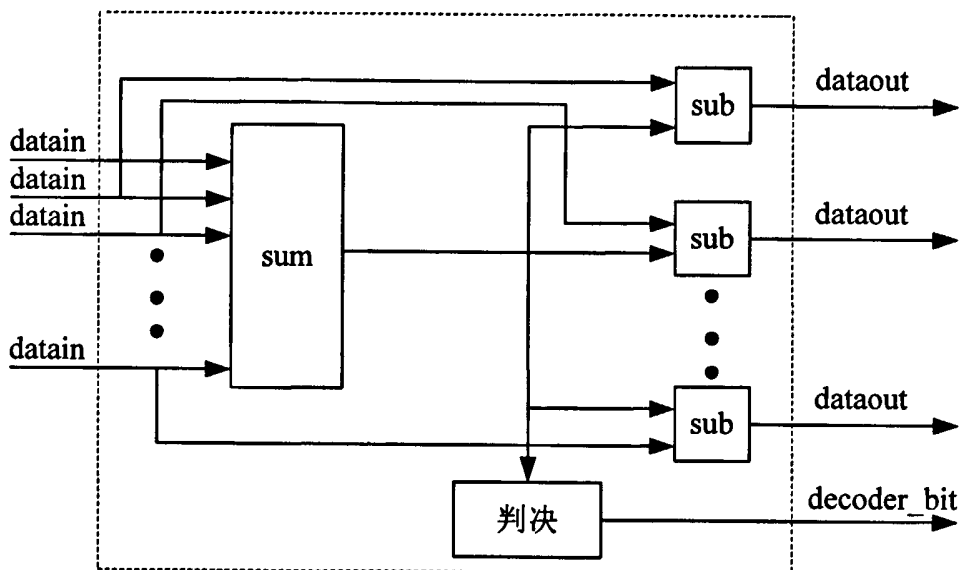


图 3.22 变量节点更新单元结构

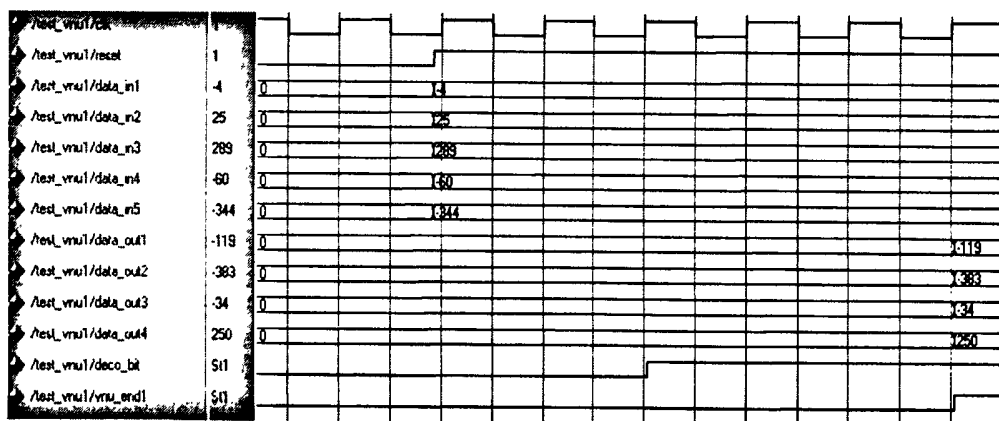


图 3.23 变量节点更新单元时序仿真

图 3.23 中所示的 data_in1、data_in2、data_in3、data_in4、data_in5 为输入变量更新信息，其中 data_in1 为输入的初始信息，data_out1、data_out2、

data_out3、data_out4 为变量更新结果信息，deco_bit 为判决的译码信息，vnu_end1 变量节点更新结束标志，高电平有效。从变量节点更新单元的时序仿真结果看，变量节点更新单元设计正确。

3.3.6 LDPC 译码器译码信息输出

译码器在信息更新达到最大迭代次数后，由译码器主控启动读使能将译码信息从译码信息缓存中输出，当所有译码信息读完时输出译码结束标志。图 3.24 所示为译码器译码信息输出仿真时序。将译码信息与编码前的信源信息进行对比，结果完全一致，表明译码器设计正确。

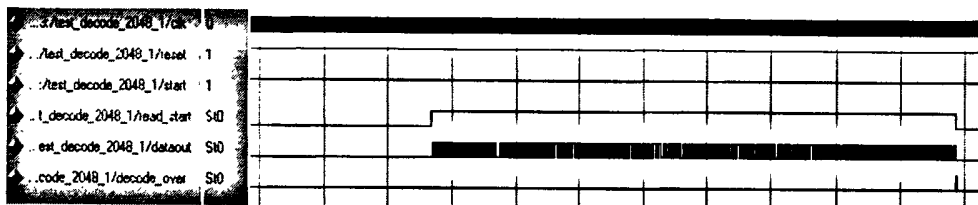


图 3.24 译码信息输出仿真时序

3.4 LDPC 译码器的测试

通过上文的对各功能模块的仿真实验，确保了各功能模块的正确，然后将各模块组合成完整的 LDPC 译码器。本设计用 XILINX 公司的 V2PRO 系列的 XC2VP30 的 FPGA 实现译码器，经 ISE 开发软件进行综合和布局布线，综合报告如下所示：

Device utilization summary:

Selected Device: 2vp30ff896-6

Number of Slices:	6732	out of	13696	49%
Number of Slice Flip Flops:	7346	out of	27392	26%

Number of 4 input LUTs:	12496	out of	27392	45%
Number used as logic:	12231			
Number used as Shift registers:	9			
Number used as RAMs:	256			
Number of IOs:	15			
Number of bonded IOBs:	14	out of	556	2%
IOB Flip Flops:	10			
Number of BRAMs:	38	out of	136	27%
Number of GCLKs:	4	out of	16	25%
Number of DCMs:	1	out of	8	12%

Timing Summary:

Speed Grade: -6

Minimum period: 3.717ns (Maximum Frequency: 269.041MHz)

Minimum input arrival time before clock: 5.116ns

Maximum output required time after clock: 3.615ns

Maximum combinational path delay: No path found

从综合报告可以看出,译码器系统能达到的最大时钟为 269.041MHz。为了确保译码器系统的设计功能的正确性和可靠性,在 ISE 的布局布线后,利用 Modelsim 6.0 进行后仿真,仿真波形如图 3.25 所示。

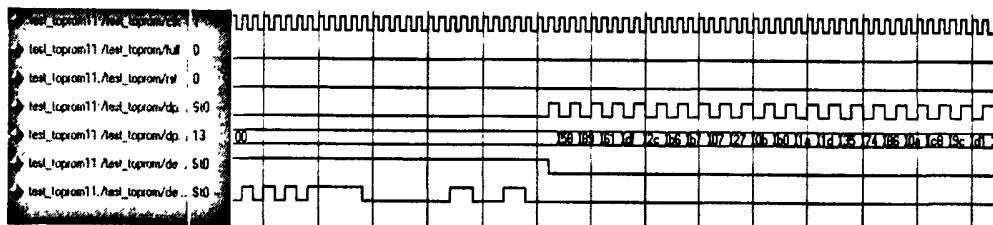


图 3.25 译码器结果局部放大图

从仿真波形可知,按第二章设计要求实现的硬件译码器,译码结果正确。为了进一步验证译码器的正确性,采用计算机产生信源、编码、调制、加噪、解调,然后通过串口将量化信息传给 FPGA 译码,译码结束后,再将译码结果通过串口返回计算机,与信源对比分析,测试系统如图 3.26 所示和译码器硬件系统测试如图 3.27。

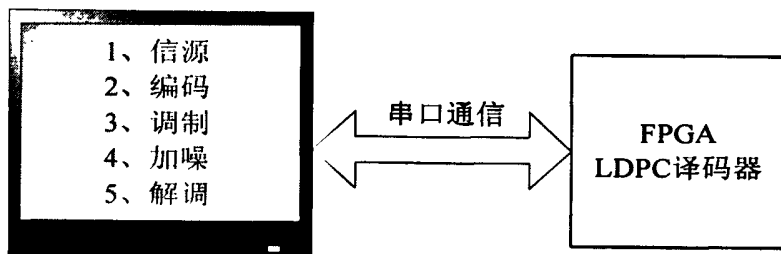


图 3.26 译码器测试系统

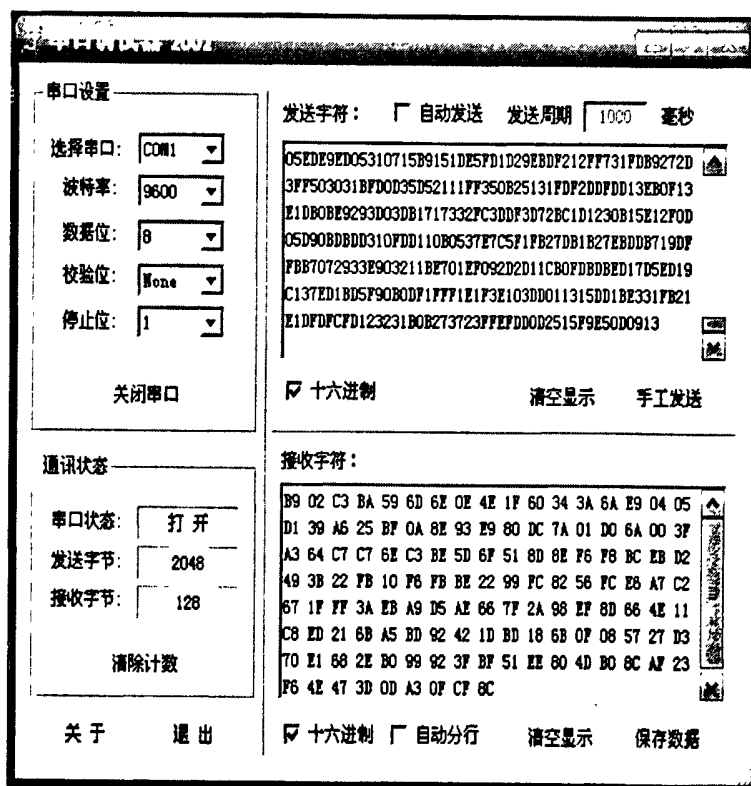


图 3.27 译码器硬件系统测试

通过多次系统测试,译码器在信噪比 2dB 下,译码结果完全正确,表明本文所设计的译码器正确,在下一章将会结合整个通信系统对译码器进一步验证。

3.5 本章小结

本章详细介绍了基于 FPGA 的 LDPC 码译码器及其子模块的设计与译码算法的实现过程。详细分析了译码器实现的整体结构和各功能模块的设计以及顶层控制时序的设计,并对各功能模块和译码器整体进行了时序仿真验证。最后用计算机模拟通信系统的其他模块,对 FPGA 实现的译码器进行系统测试,测试结果论证了译码器的设计正确。

第 4 章 系统测试的设计与实现

前文介绍了 LDPC 译码器的设计和测试，下面详细介绍整个通信系统的搭建和系统测试。

4.1 测试通信系统模型

根据本论文背景建立一个通信系统，包括信道编码（采用 LDPC 码），调制（采用 MSK 调制），信道（高斯白噪声信道），解调，译码等环节，其系统框图如图 4.1 所示。

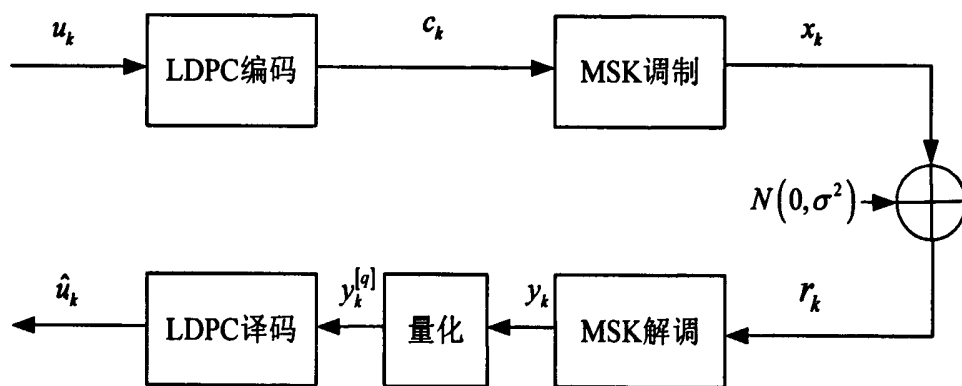


图 4.1 系统模型

首先产生信息序列 $\{u_k\}$ 经过 LDPC 码编码器得到码字序列 $\{c_k\}$ ，然后经过 MSK 调制得到发送序列 $\{x_k\}$ ，再经过加性高斯白噪声（AWGN）信道 $N(0, \sigma^2)$ ，接收序列 $\{r_k\}$ 经过 MSK 解调得 $\{y_k\}$ ， $\{y_k\}$ 经过量化得到量化接收序列 $\{y_k^{[q]}\}$ ，将量化接收序列 $\{y_k^{[q]}\}$ 送入 LDPC 码译码器，最后输出译码信息序列 $\{\hat{u}_k\}$ 。

4.2 系统测试的设计

系统测试搭建，首先用计算机产生信源，通过串口发给硬件编码器，编码器从串口 1 接收数据后，进行编码，并把编码结果直接传给硬件 MSK 调制，通过采样 MSK 信号，将采样值通过串口 1 再传回计算机进行加噪，加噪后再通过串口 2 传给硬件 MSK 解调，解调不直接进行硬判决，而是将软信息进行量化后送给硬件译码器，译码器译码结束后，通过串口 2 再返回译码结果给计算机，将译码信息与信源进行对比统计误码率，系统示意图如图 4.2 所示。

在实现编码器、译码器、调制解调后，将设计测试系统。测试系统除了编译码器和调制解调还包括信源产生、加噪、量化、误码率统计，都用 VC 实现。

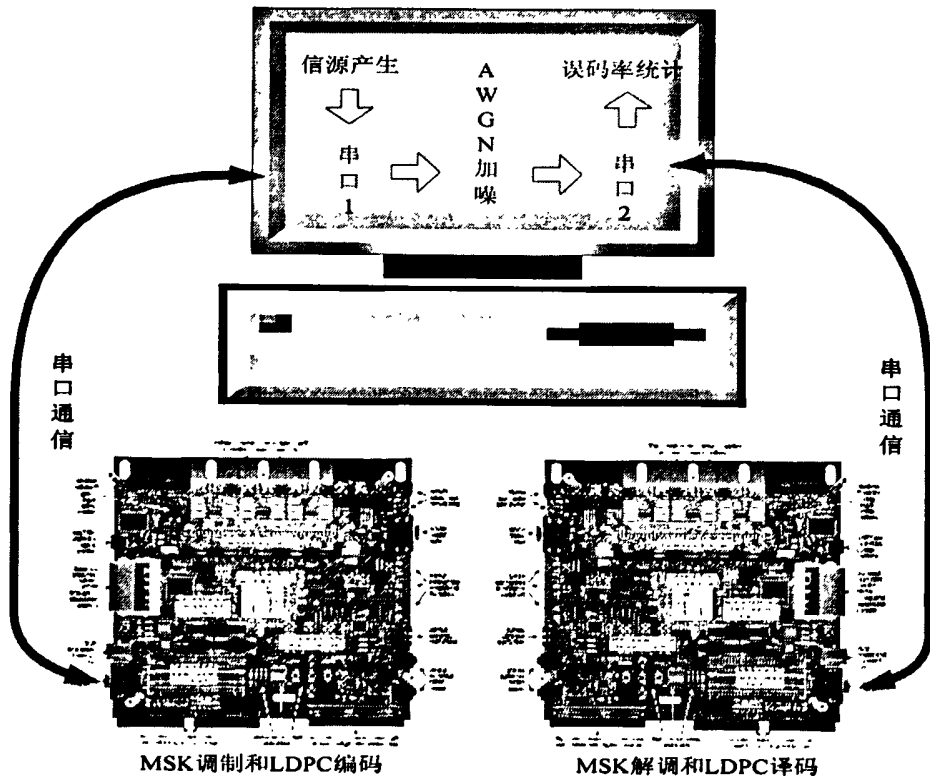


图 4.2 硬件调试系统测试示意图

4.2.1 信源产生

采用 VC 中库函数提供的两个函数 `srand` 和 `rand` 来产生随机数, 将这两函数以编码器所需数据长度来定时产生随机数作为信源, 并将信源转换为十六进制后, 通过串口发送的硬件编码器。

4.2.2 加噪处理

经过编码和调制后, 数据通过串口传回计算机, 在 VC 中进行加噪。由于信道为高斯白噪声信道 $N(0, \sigma^2)$, 因此需要知道信噪比和信号平均功率来计算噪声功率 σ^2 , 其中信噪比可以预先设置、信号平均功率由调制后的信息数据直接求得。

首先通过 VC 产生高斯分布信号 $N(0,1)$, 然后根据信号平均功率和信噪比求出噪声幅度, 最后求出相应信噪比的高斯白噪声 $N(0, \sigma^2)$, 将噪声与信号叠加实现加噪。

4.2.3 量化处理

由于信号是基于浮点数加噪, 然而从串口发送到解调和译码器的数据信息要求进行量化, 采用 10bit 量化。将量化的信息转换为整型数据, 并以补码的形式从串口发送给解调器。

4.2.4 误码率统计

译码器从串口 2 送入的译码信息后, 与信源逐比特进行比较, 统计错误的比特数, 将错误比特总数除以测试数据计算出误码率。

4.2.5 系统测试界面

系统测试界面主要包括信源发送、信道输入、信道输出、信宿接收、测试帧数、误码率等，测试界面如图 4.3 所示。

哈尔滨工程大学

误码率测试系统

测试帧数 已发帧数

☒ 十六进制显示

信源发送
6c2b5ec896fdc3350e0cb7291d7c9072bf14ae62527de551c77e7c...

信道输入
80 B1 DB F6 FF F6 DB B1 80 4F 25 0A 01 0A 25 4F 80 4F 25 C...
6 DB B1 80 4F 25 0A 01 0A 25 4F 80 B1 DB F6 FF F6 DB B1 E...
0A 01 0A 25 4F 80 4F 25 0A 01 0A 25 4F 80 B1 DB F6 FF F6 I...

信道输出
0715ff3929172117e3f51115273d0d1b0905d1fdd7cfe5f1ff29070 ^
e9cf0901efebd1e901f1f5e3f511cfcb0d17f7e9e9bfff1e3f10305f1fc
0bfd3135fb33234517fb05d3c1e5f9fd11f71b13313713f1f0723ff
e3ebdde3cd01f9ebef5f70b09ef212307eddbdfe5f7f3dbcdfe103

信宿接收
6C 2B 5B C8 96 FD C3 35 0B 0C B7 29 1D 7C 90 72 BF 14 AB ^

接收帧 误码率

误帧率 丢帧率

图 4.3 系统测试界面

系统测试步骤如下：

1. 计算机产生信源显示在信源发送窗口，然后从串口 1 发送给硬件编码

器，编码结束后调制；

2.将已调信息采样再从串口 1 返回计算机，显示到信道输入窗口；

3.对已调信息进行加噪和量化，并显示在信道输出窗口，然后从串口 2 发送给硬件解调，解调结束后译码；

4.译码信息从串口 2 返回并显示在信宿接收窗口，然后译码信息与信源对比，统计误码率。

4.3 测试系统的实现

在完成所有硬件设计和测试系统搭建之后，下一步就是进行系统测试。为了更准确的测试系统的实际性能，对系统采用多个信噪比点进行了测试，测试结果如表 4.1 所示。

表 4.1 实际测试结果

E_b/N_0	0dB	0.5dB	1dB	1.5dB	2.0dB
测试的帧数	200	300	500	500	1000
BER	0.1431	0.1084	0.0539	3.8887e-3	7.8125e-6

将实际系统测试结果与理论仿真结果进行对比，如图 4.4 所示。从图中可以看出两者基本吻合，虽然实际测试的性能与理论仿真曲线有一定的偏差，但经多次测试验证，理论和实际的性能都比较接近，表明译码器设计正确。

从图 4.4 可以看出，在码率为 1/2，码长为 2048，调制为 MSK 调制，译码算法采用 Normalized BP based，迭代 25 次译码的 QC_LDPC 码的信道编码对应的通信系统方案在归一化信噪比 2dB 时，系统误码率为 7.8125×10^{-6} ，低于 10^{-4} ，能满足系统的性能指标。

由于本文所用的测试通信系统是在计算机中完成，信道为计算机模拟信道，且采用基带调制解调，与实际的通信系统有些偏差。虽然系统在归一化信噪比 2dB 时达到了系统要求，但考虑实际信道与模拟信道的区别以

及基带调制与带载波调制的区别,测试系统性能与实际系统有些偏差。因此,为了更进一步的检测系统性能,应该采用带载波的调制解调和实际的信道中测量。

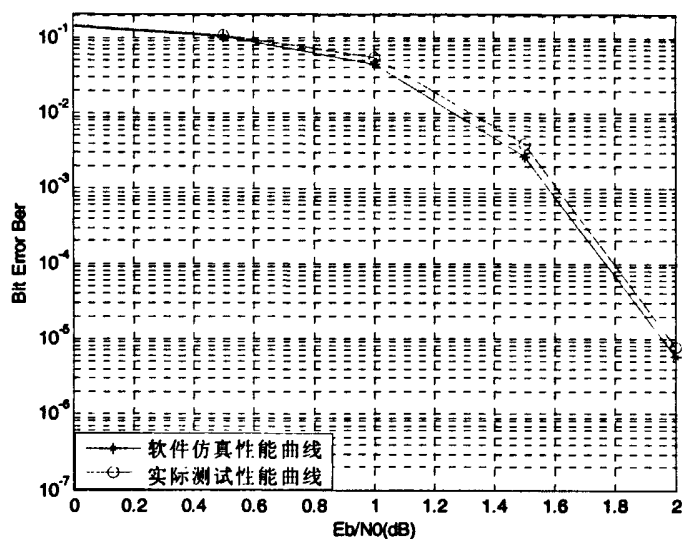


图 4.4 整个通信系统性能仿真图

4.4 本章小结

本章详细介绍了整个测试通信系统模型以及测试系统设计和实现,对测试系统进行了分析,完成各个测试模块的设计和整个系统测试的实现。最后,将实际的系统的测试结果与理论仿真进行对比,分析本文 FPGA 实现的 LDPC 码译码器和整个通信系统的不足,并为下一步完善 LDPC 码和通信系统的研究指明方向。

结 论

本文以 LDPC 码作为信道编码的通信系统为研究对象。在低信噪比信道下, 信道编码采用 1/2 码率的 LDPC 码, 结合 MSK 调制, 在归一化信噪比 2dB 时, 要求系统误码率低于 10^{-4} 。以此为设计的性能指标, 在 FPGA 上设计和实现 LDPC 码硬件译码器, 并且用 VC 设计测试系统, 并实现系统测试。本文主要完成了以下几方面的工作:

1.介绍了 LDPC 码的基础知识及其译码算法。对各种译码算法的纠错性能和运算复杂度以及硬件实现复杂度进行对比分析, 并确定合适的译码算法用于硬件实现。

2.分析影响系统性能的各方面因素。通过计算机对各个参数进行仿真分析, 并确定用于 LDPC 码硬件译码器实现的各参数。

3.分析影响 LDPC 码硬件译码器设计的因素。结合课题背景对现有的译码器设计结构对比分析, 考虑到译码速度和硬件资源消耗以及译码器的设计灵活性, 选择部分并行结构来实现硬件译码器。

4.LDPC 码硬件译码器的设计与测试。结合译码算法和部分并行结构对译码器进行电路设计, 即顶层模块设计和基本功能模块划分, 然后用硬件描述语言编写, 经 ISE 的综合、布局和布线, 并在 Modelsim 上进行时序仿真验证, 最后将程序代码下载到 FPGA, 用计算机串口进行测试, 验证了译码器设计正确。

5.利用 VC 搭建测试系统, 并完成系统测试。结合课题背景, 用 VC 实现信源产生、加噪和误码率统计, 与硬件实现的编译码和调制解调组合成测试系统, 并在不同信噪比下完成系统测试。

本文设计的译码器由于时间仓促, 因此还有许多需要完善的地方, 下一步的研究重点集中在以下几个方面:

1.改善译码器的译码性能。本文采用 Normalized BP based 译码算法中的归一化因子 μ 在整个译码过程为定值, 但对整个译码过程归一化因子 μ 与迭代次数相关, 因此可以根据不同迭代次数选择不同的归一化因子 μ 改善译码器的译码性能。

2.设计高码率译码器以及多码率译码器。研究高码率 LDPC 码的校验矩阵构造, 选择结构化比较强的 QC_LDPC 校验矩阵, 设计适用于多码率的复合译码结构。

3.研究多元域下非规则 LDPC 码译码器的硬件实现。在不损失译码性能情况, 简化多元域的 LDPC 码的译码算法, 降低译码器的硬件实现复杂度。

4.选择性能更优的译码算法实现硬件译码器。采用高速 DSP 来实现具有大量指数、对数和乘法运算的概率 BP 和 LLR BP 的译码算法, 不仅可以提高译码性能, 同时在相同性能下, 可以减小码长来降低硬件资源消耗。

参考文献

- [1] C.E.Shannon. A mathematical theory of communication. Bell Systematic Technical Journal. Vol. 27. July/Oct, 1948. 379-423P.
- [2] R W Hamming. Error detecting and error correcting codes. Bell System Tech J, 1950.
- [3] R G Gallager. Low Density Parity Check Codes. IEEE Transactions on Information Theory, 1962, 8(3):208-220P.
- [4] R M Tanner. A recursive approach to low complexity codes. IEEE Trans Inform Theory, 1981 IT-27.
- [5] C.Berrou, A.Glavieux, P.Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1).Proc. ICC'93. Geneva, 1993:1064-1070P.
- [6] Mackay D J C, Neal R.M. Near Shannon Limit Performance of Low Density Parity Check Codes. Electronics Letters, 1996:1645-1652P.
- [7] S-Y Chung, G D Forney, T Richardson, R Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. IEEE Commun. Letters, 2001.
- [8] M.C.Davey, D.J.C.Mackay. Low Density Parity Check Codes over GF(q) [J].IEEE Commun.Lett., 1998, 2(6):165-167.
- [9] M.G. Luby, M.Mitzenmacher, M.A.Shokrollahi, et al. Improved Low-Density Parity-Check Codes Using Irregular Graphs. IEEE Transactions on Information Theory, 2001, 47(2):585-598.
- [10] "Low-density parity-check codes for DSL transmission", Temporary Document BI-095, Study Group 15/4, Goa, India, 23-27 Oct 2000.

- [11] Second generation framing structure, channel coding and modulation systems for Broadcasting, Iterative Service, News Gathering and other broadband and satellite applications, Draft ETSI EN 302 307 European Standard. <http://www.ebu.ch>, 2004.
- [12] Alberto Morello, Vittoria Mignone. DVB-S2 ready to lift off [EB/OL]. <http://www.ebu.ch>, 2004.
- [13] Alberto Morello. The new generation DVB system by SATELLITE [EB/OL]. <http://www.ebu.ch>, 2004.
- [14] Bane Vasic, B.D.Ivan, Raymond K.Kostuk. Low Density Parity Check Codes and Iterative Decoding for Long-Haul Optical Communication Systems. *IEEE Journal Light-wave Technology*, 2003, 21(2):438-446
- [15] Hongxin Song, J.R.Cruz. Reduced-Complexity Decoding of Q-ary LDPC Codes for Magnetic Recording. *IEEE Transactions on Magnetic*, 2003, 39(2):1081-1087
- [16] B.D.Ivan, Bane Vasic. Projective Geometry LDPC Codes for Ultra long-haul WDM High-Speed Transmission. *IEEE Photonics Technology Letters*, 2003, 15(5):784-786.
- [17] T.Richardson, R. Urbanke. The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding. *IEEE Transactions on Information Theory*, 2001, 47(2):599-618.
- [18] Marc P.C.Fossorier, Miodrag Mihaljevic, Hideki Imai. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Transactions on Communications*, Vol.47, No.5, May 1999:673-680.
- [19] Marc P.C.Fossorier. Iterative Reliability-Based Decoding of Low-Density Parity Check Codes. *IEEE Journal on Selected Areas in Communication*, Vol 19, No.5, May.2001:908-917.

- [20] Jinghu Chen, Marc P.C.Fossorier. Decoding Low-Density Parity Check Codes with Normalized APP-Based Algorithm. GLOBECOM 2001-IEEE Global Telecommunications Conference, No. 1, Nov. 2001:1026:1030.
- [21] Jinghu Chen, Marc P.C.Fossorier. Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes. IEEE Transaction on Communications, Vol.50, No.3, Mar. 2002:406-414.
- [22] Jinghu Chen, Marc P.C.Fossorier. Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes. IEEE Communications Letters, Vol.6, No.5, May. 2002:208-210.
- [23] 吴湛击, 利璐颖, 欧阳子月, 王文博. 新的高效 LDPC 码的译码方法. 北京邮电大学学报. 2007 年 10 月, 第 30 卷第 6 期, 105-107.
- [24] Yucheng He, et al. Fast Decoding of LDPC using Quantization. IEEE Electronics Letter, 14th.Vol.48, No.4, Feb.2002.
- [25] Yongyi Mao, Amir H.Banihashemi. Decoding Low-Density Parity-Check Codes with Probabilistic Scheduling. IEEE Communications Letters, Vol.5, No.10, October. 2001.
- [26] Yongyi Mao, Amir H. Banihashemi. A New Schedule off Decoding Low-Density Party-Check Codes. GLOBECOM 2001-IEEE Global Telecommunications Conference, No.1, Nov. 2001:1007-1010.
- [27] Acholleas Anastasopoulos. A Comparison between the Sum-Product and the Min-Sum Iterative Detection Algorithms Based on Density Evolution. GLOBECOM 2001-IEEE Global Telecommunications Conference, No.1, Nov. 2001:1021-1025.
- [28] Y Kou, S Lin, and M Fossorier. Low Density Parity Check Codes based on Finite Geometries: A Rediscovery and New Results. IEEE Trans. On Inform. Theory, Vol.47, No.5, pp. 2711-2736, Nov. 1992.
- [29] J Zhang and M P C Fossorier. A Modified Weighted Bit-Flipping

- Decoding of Low-Density Parity-Check Codes. IEEE Communications Letters, Vol.8, No.3, pp. 165-167, Mar. 2004.
- [30] Z Liu and D A Pados. Low Complexity Decoding of Finite Geometry LDPC Codes. IEEE Trans. on Communications, Vol.53, No.3, pp. 415-421, Mar. 2005.
- [31] M Shan, C M Zhao and M Jiang, Improved Weighted Bit-Flipping Algorithm for Decoding LDPC Codes. IEE Proc. Commun, Vol.152, No.6, pp.919-922, Dec. 2005.
- [32] S. Hirst, B. Honry. Decoding of Generalized Low-Density Parity-Check Codes Using Weighted Bit-Flip Voting. IEE Proceedings Communications, 2002, 149(1):1-5.
- [33] 孙韶辉, 贺玉成, 王新梅. 低密度校验码在瑞利衰落信道中的性能分析. 计算机学报, 2002, 25(10):1077-1082.
- [34] Matthew C. Davey, David J.C. Mackay. Reliable Communication over Channels with Insertions, Deletions, and Substitutions. IEEE Transactions on Information Theory, 2001, 47(2):687-698
- [35] Hans-Andrea Loeliger, Felis Lustenberger, Markus Helfenstein, et al. Probability Propagation and Decoding in Analog VLSI. IEEE Transactions on Information Theory, 2001, 47(2):837-843.
- [36] Loeliger H A et al. Probability Propagation and Decoding in Analog VLSI, IEEE Trans. Information Theory, 2001, 47(2):837-843.
- [37] Zhang T and Parhi K K. VLSI Implementation-Oriented (3, K)-Regular Low-Density Parity-Check Codes, from Internet, { tzhang, parhi }@ece.umn.edu.
- [38] Levine B et al. Implementation of Near Shannon Limit Error-Correcting Codes Using Reconfigurable Hardware, from Internet, blevince@ce.cmu.edu.

- [39] Yeo E et al. High Throughput Low-Density Parity-Check Decoder Architectures, Proc. IEEE ICCT'2001, 2001:3019-3024.
- [40] 王育民, 李晖, 梁传甲. 信息论与编码理论. 北京: 高等教育出版社, 2005:258-298 页.
- [41] 王琳, 徐立凯等. 高效信道编译码技术及其应用. 北京: 人民邮电出版社, 2007:125-147 页.
- [42] Zongwang Li, Lei Chen, Lingqi Zeng, Shu Lin and Wai H.Fong. Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes. IEEE Trans. Commun, Janu.2006, 54(1):71-81.
- [43] 李睿. QC-LDPC 部分并行译码器的设计与实现. 电子科技大学硕士学位论文. 2007:9-10 页.
- [44] 袁东风, 张海刚等. LDPC 码理论与应用. 北京: 人民邮电出版社, 2008:72-106 页.
- [45] 樊昌信, 张甫翊, 徐炳祥, 吴成柯. 通信原理. 北京: 国防工业出版社, 2001:195-198.
- [46] 田耘, 徐文波. Xilinx FPGA 开发实用教程. 北京: 清华大学出版社, 2008:9-19.
- [47] 单鸣. LDPC 码解码技术研究及实现. 东南大学硕士学位论文. 2004:45-48.
- [48] 刘腾宇. LDPC 码译码器的 FPGA 实现研究. 哈尔滨工程大学. 2007:44-48.
- [49] T. Zhang. Efficient VLSI Architectures for Error-Correcting Coding. Ph.D. dissertation, University of Minnesota July, 2002: 51-77P.
- [50] BLANKSBY A J, HOWLAND C J. A 690-mW 1-Gb/s 1024-bit, rate-1/2, low-density parity-check code decoder. IEEE Journal of Solid state Circuits, 2002, 37(3): 404-412P.
- [51] ZHANG Tong, PARHI K K. A 54 Mbps (3, 6)-regular FPGA LDPC

decoder. Proceedings of IEEE Workshop on Signal Processing Systems (SIPS 02), Oct 16-18, 2002, San Diego, CA, USA. Piscataway, NJ, USA: IEEE, 2002:127-132P.

攻读硕士期间发表论文和取得的科研成果

1. 李加洪, 赵旦峰, 薛瑞, 陈广. 一种可变码长 QC_LDPC 译码结构设计与实现. 黑龙江科技信息. 2010 年第 5 期

致 谢

在本文完成之际，谨向所有指导、帮助和关心过我的老师、同学、朋友和亲人表示衷心的感谢！

首先要衷心的感谢我的导师赵旦峰教授，感谢赵老师这两年多来对我的悉心教导与帮助。赵老师为我创造了良好的学习环境，在学术上悉心指导，思想上耐心启迪，生活上关心备至。赵老师严谨求学的治学态度，对事业不倦的追求，以及和蔼可亲平易近人的品格给我留下了深刻的印象，是我永远学习的楷模。

同时，我要感谢实验室的师兄薛睿和各位同学，在论文写作过程中给予我的帮助和在这两年多的研究生生活中给我带来的快乐！

感谢哈尔滨工程大学的各位老师，感谢你们教育我如何做人做事，谢谢你们的教育和栽培。

最后，我要感谢父母和家人对我的支持和鼓励，无论在我人生的任何阶段，他们都给予了我无私的关怀和理解，使我顺利完成了学业。