

基于 FPGA 的 Turbo 码译码实验平台设计

贾墨林 聂 伟*

(北京化工大学 计算机系统与通信实验中心, 北京 100029)

摘 要: 实现一种基于 FPGA 的 Turbo 译码器实验平台, 包括译码数学模型、FPGA 实现方案、测试结果及分析。系统采用一个通用硬件平台和模块化的软件设计, 可对译码算法和迭代次数进行可视化设置。通过仿真与实际测试, 结果正确且工作稳定可靠。

关键词: Turbo 译码; 实验教学平台; FPGA

中图分类号: TP391.6

引 言

Turbo 码以其优越的纠错性能, 成为向前纠错编码领域的一个重大突破。在低信噪比下其表现出接近香浓极限的优异性能, 使其在通信领域中具有广阔的前景。目前, Turbo 码已被多个国际标准所采纳, 并被广泛应用于卫星通信、数字广播电视(DVB)、无线接入以及第三代移动通信等多个领域^[1-2]。很多高校都将 Turbo 码原理作为通信原理等课程的内容, 但译码算法复杂, 是一个教学难点。通信技术的快速发展要求通信专业的学生和科技人员不但要掌握扎实的理论知识, 还应具有较强的工程概念和动手能力。因此, 本文设计并实现了一种用于实验教学的 Turbo 译码实验平台。

FPGA 具有实时性好、逻辑运算快、用户可重复编程和外设配置简单等优点。现在已经有更多门数、更高速度、功能更强的 FPGA, 其内嵌多种 IP 核。这都使采用 FPGA 实现 Turbo 码这种复杂运算的高速译码器作为实验教学平台成为最佳的选择。

本文在研究 Turbo 码迭代译码方法的基础上实现了 SOVA 和 Max-Log-Map 两种子译码算法。可以通过自带的控制器对子译码器译码算法、迭代次数进行可视化设置, 实现译码。文中给出译码系统的硬件结构、两种译码算法分析及实现、算法仿真、实

际测试, 并对测试结果进行了分析。

1 系统硬件设计

根据软件仿真结果, 译码系统需要 9889 个逻辑单元。系统硬件结构设计参考文献[3]。系统主要由译码器模块和控制器模块组成。其中控制器模块完成对译码器模块的管理; 译码器模块根据用户的要求完成相应的译码方式。控制器模块与译码器模块的接口采用串行同步方式进行通信。输入控制指令可通过键盘和 LCD(或上位机)进行可视化操作。

与文献[3]硬件设计所不同的是译码过程无需数模转换。编码信号经过预处理电路送 FPGA, 译码后送入数字终端, 显示译码结果。为方便观察译码过程, 由 FPGA 的扩展 I/O 接口引出多个测试点。可根据需要, 通过软件锁定管脚, 设置观测点。

2 译码器软件设计

2.1 译码器整体结构设计

Turbo 码获得优异性能的根本原因之一是采用了迭代译码, 通过分量译码器之间软信息交换来提高译码性能。系统先要对接收的信号进行信道可靠性因子加权, 然后根据加权后的信号计算分支转移, 从而进行译码。Turbo 译码器主要由 2 个交替工作的软输入软输出(SISO)译码器 DEC1、DEC2 串行级联以及相应的交织器和解交织器组成。译码时, 译码器 DEC1 对分量码 RSC1 进行最佳译码后产生关于信息序列的似然信息, 然后将信息经过交织送给译码模块 DEC2。DEC2 将此信息作为先验信息, 对分量码 RSC2 进行最佳译码后产生关于交织后的信息序列的似然信息, 然后将外信息经过解交织返还

收稿日期: 2009-10-11

基金项目: 北京化工大学实验教学专项建设资金

第一作者: 男, 1984 年生, 硕士生

* 通讯联系人

E-mail: niewei@mail.buct.edu.cn

给 DEC1, 进行下一次译码。就这样, 迭代译码继续进行下去, 直到获得所需要的性能为止。随着迭代次数的增加, 两个分量译码器得到的外部信息值对译码性能提高的作用越来越小, 在达到一定的迭代次数后, 译码性能不再提高。一般, 迭代 5 ~ 10 次即可得到很好的结果^[4]。这时根据分量译码器 2 的输出对数似然比经过解交织后再进行硬判决即得到译码输出。

系统中 2 个子译码器采用了 2 种算法以供选择。控制模块主要根据来自单片机的指令控制子译码器选择 SOVA 算法还是 Max-Log-Map 算法 (简称 ML-Map) 以及迭代次数。译码器软件设计, 如图 1。

L_r 是关于 u_k 的对数似然比

$$L_r(u_k) = \ln \frac{P(u_k = 1 | y)}{P(u_k = -1 | y)}$$

每一个 SISO 计算得到对数似然比减去信道输出软值和先验信息, 就得到其外信息。

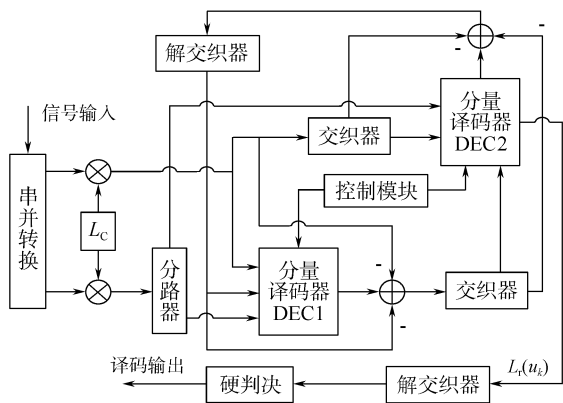


图 1 Turbo 译码器 FPGA 软件结构

Fig.1 Software configuration of FPGA in the Turbo decoder

2.2 子译码器软件设计

2.2.1 ML-Map 译码算法分析及设计

Map 算法需要大量的乘法运算和指数运算及大量的存储, 因此不适合于实际运用。Log-Map 算法是把 Map 算法中的乘法运算转换为对数域中的加法运算, 并且不需要对数运算, 使得 Map 实际运用成为可能^[5]。在实现的时候。可以将原来在对数域内的加法运算转换为取两个数的较大者加一个修正项的运算

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|})$$

由于修正函数是 x 和 y 的差的绝对值的函数, 所以可以用一维查找表实现, 就是 ML-Map 译码算法。然而, 这种算法相对于 Map 算法有 0.5 dB 的损失,

但是运算复杂度大大降低^[6]。针对 MAP 算法中用到的似然概率比 Λ_k 以及计算过程中的 3 个中间度量: 分支度量 γ_k , 前向路径度量 α_k 后向路径度量 β_k 分别取它们的对数

$$L_r = \log \Lambda_k, \bar{\gamma}_k = \log \gamma_k, \bar{\alpha}_k = \log \alpha_k, \bar{\beta}_k = \log \beta_k$$

分支度量为

$$\bar{\gamma}_k(S_{k-1}, S_k) = \bar{\Lambda}_{in,k}(u_k) x_s^{i1} + L_c y_k^1 x_s^{i1} + L_c y_k^2 x_s^{i2}$$

$\bar{\Lambda}_{in,k}$ 为先验概率信息, x_s 表示发送前的码字, y_k 表示接收到的码字, 注意到 x_s^i 是系统码, 在编码器中不进行处理, 它的值仍等于 u_k , 1 和 2 上角标分别表示信息比特和校验比特, L_c 为信道可靠因子。

$$\bar{\alpha}_k(S_k) = \max_{S_{k-1}}^* (\bar{\gamma}_k(S_{k-1}, S_k) + \bar{\alpha}_{k-1}(S_{k-1}))$$

$$\bar{\beta}_k(S_k) = \max_{S_{k+1}}^* (\bar{\gamma}_{k+1}(S_k, S_{k+1}) + \bar{\beta}_{k+1}(S_{k+1}))$$

经过迭代译码所得的信息比特 u_k 的对数似然比值 (LLR) 可表示为

$$L_R(u_k) = \max_{S_{k-1}^{u_k}=+1}^* (\bar{\alpha}_k(S_k) + \bar{\gamma}_{k+1}(S_k, S_{k+1}) + \bar{\beta}_{k+1}(S_{k+1})) - \max_{S_{k-1}^{u_k}=-1}^* (\bar{\alpha}_k(S_k) + \bar{\gamma}_{k+1}(S_k, S_{k+1}) + \bar{\beta}_{k+1}(S_{k+1}))$$

由于该算法是迭代算法, 下一次译码将利用上一次译码器的软输出提供的附加信息进行译码, 这个外征信息为

$$\bar{\Lambda}_{e,k}(u_k) = L_r(u_k) - \bar{\Lambda}_{in,k}(u_k) - L_c y_k^1$$

按算法流程划分功能模块, 软件实现结构如图 2。

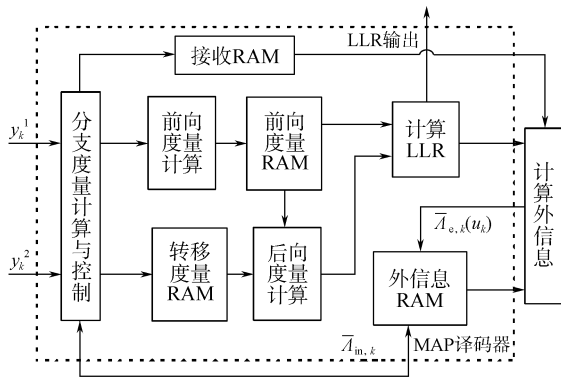


图 2 Max-Log-Map 子译码算法 FPGA 软件结构

Fig.2 Software configuration of FPGA in the Max-Log-Map

2.2.1 SOVA 译码算法分析

SOVA 算法的译码过程就是在接收序列 R 的控制下, 在码的格图上走编码器走过的路径。对于某一状态 S^d , SOVA 算法选择一条幸存路径, 这是通过计算最小路径度量 (或最大相关度量) 而得到的, 此时 d 取为 1。同时, 该状态还对应着一条待选路径,

即竞争路径,此时 d 取为 2。根据贝叶斯准则,后验概率可以表示为

$$P(S^d|Y) = P(Y|S^d) \frac{P(S^d)}{P(Y)}$$

若接收序列是固定的,且与路径度量无关,因此可以省略。故最大化的结果可以表示为

$$\max_k P(Y|S^d) P(S^d)$$

整理后可以得到 SOVA 的度量计算公式为

$$M_k^d = M_{k-1}^d + \sum_{j=1}^N x_{k,j}^d L_c \gamma_{k,j} + x_k^d \bar{A}_{in,k}(u_k)$$

x_k^d 表示该分支路径的码字输出序列。指定节点处的幸存路径的可靠性度量,即在 k 时刻某个结点的软信息为

$$\Delta_k = |M_k^1 - M_k^2|/2$$

最优路径和竞争路径的度量差值越小意味这次选取的可靠性越低。正确选择幸存路径的概率为

$$P_c = \frac{\exp(M_k^1/2)}{\exp(M_k^1/2) + \exp(M_k^2/2)}$$

式中,若 $M_k^1 - M_k^2 \leq 0$ 。代表传输不可靠度。于是根据 BR-SOVA^[7]算法进行回溯更新。删除低似然路径,保留必要的信息,以给每个输出比特提供一个可信度。更新后对数似然比 L_R 为

$$L_R(u_k) = \ln \frac{P_c}{1 - P_c}$$

可得输出软信息

$$\bar{A}_{in,k}(u_k) = u_k L_r(u_k)$$

SOVA 算法是基于序列的译码算法,算法中主要就是加法运算,然后通过比较进行选择。由于其延迟与计算量都较低,比较适用于工程实现中。按算法流程划分功能模块,软件实现结构如图 3。

3 软件仿真、实际测试及分析

3.1 译码器软件仿真

仿真利用 QuartusII7.2 完成。设定系统工作时钟为 20MHz,迭代次数为 3,子分量译码器采用 SOVA 算法,Turbo 译码器的仿真结果如图 4 所示。

其中,sgm2 为控制加性高斯白噪声信道信噪比的系数,iterin 为预设迭代次数,p_out 为按帧分组的原始伪随机序列,hardout 为 Turbo 译码器输出的硬判决输出。采用 ML-Map 算法译码结果相同,但由于计算复杂延时较长。

3.2 译码器实际测试及分析

下面对译码器不同译码算法及迭代次数进行实

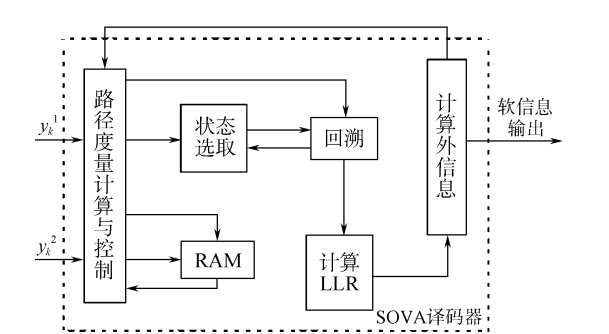


图 3 SOVA 子译码算法 FPGA 软件结构
Fig. 3 Software configuration of FPGA in SOVA

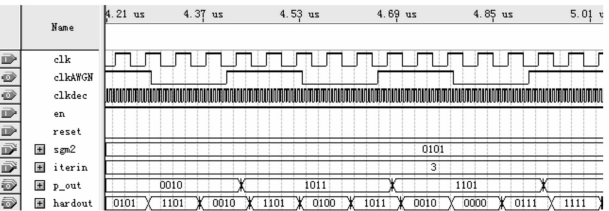


图 4 译码器仿真结果
Fig. 4 Simulation waveform of the Turbo decoder

际测试。设定系统工作时钟为 20 MHz。利用已有的 Turbo 编码平台,采用加性白噪声模块进行加噪(由 FPGA 实现,并且噪声功率可调)。对两种译码算法在不同迭代次数情况下利用 SynC 公司误码仪(VEEX-TX50E)进行测试,测试结果如下表 1,给出不同和迭代次数信噪比下 2 种译码算法误码范围

		表 1 误码率测试对比	
		Table 1 Comparison of bit error rates	
信噪比	迭代次数	误码率	
		SOVA	ML-Map
1.0 dB	4	$9.37 \times 10^{-2} \sim 9.52 \times 10^{-2}$	$4.23 \times 10^{-3} \sim 4.40 \times 10^{-3}$
	5	$8.99 \times 10^{-2} \sim 9.05 \times 10^{-2}$	$8.00 \times 10^{-4} \sim 8.12 \times 10^{-4}$
	6	$8.90 \times 10^{-2} \sim 8.98 \times 10^{-2}$	$1.11 \times 10^{-4} \sim 1.24 \times 10^{-4}$
2.0 dB	4	$2.09 \times 10^{-4} \sim 2.18 \times 10^{-4}$	$1.21 \times 10^{-6} \sim 1.22 \times 10^{-6}$
	5	$2.57 \times 10^{-5} \sim 2.69 \times 10^{-5}$	0.00
	6	$2.73 \times 10^{-6} \sim 2.87 \times 10^{-6}$	0.00

ML-Map 算法和 SOVA 算法都接收信道传来的软判决信息和信息比特的先验信息作为译码输入,译码输出不仅可以给出判决,而且也可以给出的后验概率 LR 值,可以统称它们为 SISO 算法。两类算法的主要区别是对编码格图中判决路径的选取方式不同,从而带来的性能不同。SOVA 算法比 ML-Map 算法实现要简单,ML-Map 的译码性能比 SOVA 要

好。这是因为,在译码过程中前者所保留的软信息比后者要多,总是从集合中选取路径量度最大的一条路径;而后者总是找出两条路径中具有最大似然值的一条,删除另一条。一定信噪比条件下随着迭代次数增加误码率快速下降。测试结果与理论值基本相符。

4 结束语

本文实现了一种基于 FPGA 的 Turbo 译码系统实验平台,通过该平台学生能清楚地观察 Turbo 译码器各环节的作用及数据流程,更好地掌握两种算法的区别及迭代次数对译码的影响。硬件电路、软件设计均采用开放接口和模块化结构,非常方便扩展和升级,且方便学生进一步研究和改进;通过仿真和实际测试,译码器完全满足设计要求且工作稳定可靠。

另外,因为在硬件电路上扩展了多种接口,所以也可以作为其他通信技术平台,因此非常适合本科生综合课程设计、毕业设计等使用,促进教学实验从传统的验证型向研究型发展,培养创新型人才。

参考文献:

- [1] Nikopour H, Khandani A K, Jamali S K. Turbo-coded OFDM transmission over a nonlinear channel[J]. IEEE Transactions on Vehicular Technology, 2005, 54(4): 1361 - 1367.
- [2] 张青春,王坤,石波涌,等. 基于 FPGA 的 Turbo 译码器设计[J]. 电子测量技术, 2008, 31(2): 113 - 115. Zhang Q C, Wang K, Shi B Y, et al. Design of the turbo decoder with FPGA[J]. Electronic Measurement Technology, 2008, 31(2): 113 - 115. (in Chinese)
- [3] 贾墨林,聂伟. 基于 FPGA 的多体制中频数字解调器实现[J]. 北京化工大学学报: 自然科学版, 2010, 37(1): 130 - 134.
- [4] Jia M L, Nie W. The realization of IF digital multi-demodulator based on FPGA[J]. Journal of Beijing University of Chemical Technology: Natural Science, 2010, 37(1): 130 - 134. (in Chinese)
- [5] 李祥明,乐光新,尹长川. Turbo 码译码器“及早判决”门限的确定及输出信噪比停止迭代准则[J]. 北京邮电大学学报, 2000, 23(1): 46 - 50. Li X M, Yue G X, Yin C C. A threshold for the ‘Early Detection’ method and an output SNR stop criterion for turbo codes[J]. Journal of Beijing University of Posts and Telecommunications, 2000, 23(1): 46 - 50. (in Chinese)
- [6] 朱磊,陈红. 3G 系统中 Turbo 译码改进及 DSP 实现[J]. 现代电子技术, 2008, 31(5): 54 - 60. Zhu L, Chen H. DSP implementation of turbo decoder for 3G system[J]. Modern Electronic Technique, 2008, 31(5): 54 - 60. (in Chinese)
- [7] 陈俊霖,朱光喜. 下一代移动通信系统高速并 Turbo 译码研究与 FPGA 实现[J]. 电子技术应用, 2006, 32(9): 126 - 129. Chen J L, Zhu G X. Research and realization of FPGA of high-speed parallel turbo decoding in the next generation mobile communication[J]. Application of Electronic Technique, 2006, 32(9): 126 - 129. (in Chinese)
- [8] 刘星成,朱帆. 基于修正路径量度的 Turbo 码 SOVA 译码算法研究[J]. 通信学报, 2008, 29(4): 124 - 129. Liu X C, Zhu Z. Study on the SOVA decoding algorithm for Turbo codes based on modified path-metric[J]. Journal on Communications, 2008, 29(4): 124 - 129. (in Chinese)

Design of a Turbo decoder experimental platform based on field-programmable gate array

JIA Molin NIE Wei

(Computer System and Communication Laboratory, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: This paper presents a design of a Turbo decoder experimental platform based on a field-programmable gate array (FPGA), including mathematical models of the decoder, realization of the program based on the FPGA, and analysis of test results. The system is based on a general hardware platform and modular software design and provides visualization of the settings of the decoding algorithms and iterations. Simulations and actual tests show that the decoder meets the design requirements of stability and reliability.

Key words: Turbo decoder; experimental teaching platform; FPGA

<http://www.journal.buct.edu.cn>