

一种改进的 Turbo 码译码算法

仵 林 博, 朱 怡 安, 田 杭 沛

(西北工业大学 航空微电子中心, 陕西 西安 710072)

摘 要: 针对已有 Turbo 码译码算法延迟长, 存储空间需求大的问题, 在对算法的计算单元、存储操作进行优化的基础上提出了一种新的状态度量归一化处理方法及基于分块的滑动窗算法, 并构建了算法的寄存器传输级模型. 该算法将分块并行技术和滑动窗算法有机的结合在一起, 能够有效降低运算中的时延及存储资源需求. 仿真结果表明, 该算法在保证性能的前提下, 具有较好的可实现性.

关键词: Turbo 码; 分块并行; 滑动窗算法

中图分类号: TN919.3

文献标识码: A

文章编号: 1000-7180(2010)12-0146-03

An Improved Turbo decoding Algorithm

WU Lin-bo, ZHU Yi-an, TIAN Hang-pe

(Aviation Microelectronics Center, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: To solve the problem of long time delay and large storage space of the existing turbo decoding, a new state metric re-scaling method and a sliding window algorithm based on sub-blocks are proposed, which is based on the optimization of calculation unit and storage operations, and also we build the register-transmission model for the algorithm. The algorithm combines parallel processing technology and sliding window algorithm organically, effectively reduces the decoding delay and the occupied memory size. The simulation results show that the algorithm has preferable realization without performance degradation.

Key words: Turbo code; parallel processing based on sub-blocks; sliding window algorithm

1 引言

Turbo 码是近些年来通讯系统纠错编码^[1]领域的重大突破, 在编码块足够长时, 具有接近 Shannon 极限的优异性能. 作为第三代移动通信系统中高速数据业务的信道编码方案, Turbo 码^[2]已在卫星通信, 3G 移动通信以及多媒体通信等领域得到了广泛应用. 然而, 由于 Turbo 码译码算法复杂度大, 译码延迟大, 计算过程中需要大量的存储资源. 文中通过对 Turbo 码译码过程中计算单元和存储操作进行优化, 提出了一种新的状态度量归一化处理方法和基于分块的滑动窗算法, 不仅降低了译码时延, 而且有效的降低了对存储资源的需求.

2 Turbo 码编/译码器结构与算法

Turbo 码编码器结构如图 1 所示, 它由两个完全相同的分量码编码器和一个交织器并行级联而成, 两个分量译码器为 8 状态(13, 15)RSC 码, 其中 u_k 表示编码输入的信息比特, x_{1p} , x_{2p} 分别表示对应两个分量编码器的编码输出.

Turbo 码译码器基本结构如图 2 所示. 它主要由两个软输入软输出(SISO)的分量码译码器 DEC1、DEC2 通过交织器和解交织器串行级联而成, 其中交织器与编码器中所使用的交织器相同. 每次迭代一个分量译码器输出的对数似然比经过提取出外信息给另一个分量译码器作为先验信息, 形成迭代译码. 随着分量译码器的先验信息不断更新, 它

输出的对数似然比逐步收敛. 经过若干次迭代译码后,对 DEC2 的输出对数似然比进行硬判决就可得到译码输出.

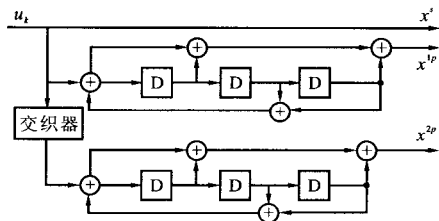


图1 Turbo 码编码器结构

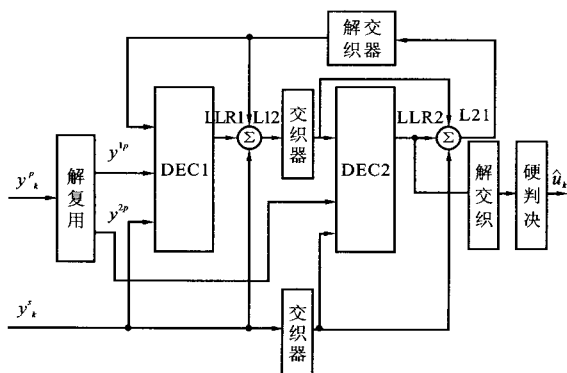


图2 Turbo 码译码器结构

Turbo 码译码器采用软输入软输出的译码算法进行译码,根据译码方式的不同可分为:逐序列译码的维比特译码算法(SOVA)和逐比特的最大后验概率译码(MAP),其中 SOVA 算法运算复杂度最低,性能较低,便于硬件实现. MAP 算法性能最好,但运算复杂度高,且需要较大的存储空间. 因此,我们通过对数域进行计算,从而降低译码复杂度,即采用 Log-Map 算法. Log-Map 算法和 Map 算法在原理上是一致的,设接收到的码字为 y ,对应的发送码字为 u_k ,定义对数后验概率为:

$$L(\hat{u}_k) = \ln \frac{P(u_k = 1 | y)}{P(u_k = 0 | y)}$$

$$= \max_{u_k=1}^{(s',s)} (A_{k-1}(s') + M_k(s',s) + B_k(s)) - \max_{u_k=0}^{(s',s)} (A_{k-1}(s') + M_k(s',s) + B_k(s)) \quad (1)$$

根据判决准则:当 $L(\hat{u}_k) > 0$ 时,硬判决 $\hat{u}_k = 1$, 否则, $\hat{u}_k = 0$. 式(1)中, s' 是接收到 y 前的移位寄存器的状态, s 是接收到 y 后移位寄存器的状态, $x_{k,v}$ 为校验位, $y_{k,l}$ 为接收到的信息位, $y_{k,v}$ 为接收到的校验位, L_c 为信道可信度, $A_k(s)$ 、 $B_k(s)$ 、 $M_k(s',s)$ 分别代表对数域的前向度量、后向度量、分支度量,其表达式分别表示为:

$$A_k(s) = \max * (M_{u_k=1}(s',s) + A_{k-1}(s')),$$

$$M_{u_k=0}(s',s) + A_{k-1}(s')) \quad (2)$$

$$B_{k-1}(s') = \max * (M_{u_k=1}(s',s) + B_k(s), M_{u_k=0}(s',s) + B_k(s)) \quad (3)$$

$$M_k(s',s) = u_k L_c y_{k,l}/2 + x_{k,v} L_c y_{k,v}/2 + u_k L(u_k)/2 \quad (4)$$

其中

$$\max * (x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|}) \quad (5)$$

在译码过程中,从一个状态节点到达另一个状态节点都对应于一个 $A_k(s)$, 一个 $B_k(s)$ 和两个 $M(s',s)$,按照如下的计算步骤完成译码输出:

(1) 先根据接收到的码序列计算每个码字对应的分支度量.

(2) 根据计算得到的分支度量值和路径度量值按照式(2)进行前向迭代,计算前向累积度量.

(3) 根据分支度量值和对应的路径度量值按照式(3)进行后向迭代,计算后向累积度量.

(4) 根据式(1)计算 $L(\hat{u}_k)$ 和进行硬判决输出.

3 Turbo 码译码算法改进与优化

Turbo 码译码是一个复杂的过程,有两个关键的操作部分,一个是前后向度量值和对数似然比的计算,主要是加比选运算单元,另一个是前后向度量值的存储,这两项操作消耗占到了系统资源的 80%^[3],因此可以通过减少对各种数据的存储和读取次数来降低 Turbo 码译码器^[4]的时延和功耗.

3.1 改进的译码算法

经典的 Log-MAP 算法计算步骤是先进行前向递推,然后再进行后向递推,最后再计算出每比特的对数似然比,这样的运算步骤带来较大的延时. 我们通过分析发现前向度量和后向度量可并行运算,也就是先进行后向递推,然后进行前向度量和对数似然比的计算同步进行. 这样不仅提高了计算速度,减少了约 1/3 的译码时延,而且还省略对前向累计度量的存储,节约了存储资源. 图 3 是经典的译码算法和改进的译码算法在子译码中的译码过程(一个周期为 T)对应的时序图,图中下半部分是改进的译码时序^[5-6].

3.2 加比选运算单元设计

Turbo 译码器的主要运算单元是加比选单元(Add-Compare-Select Unit),一个 ACS 单元由两个加法器、一个比较器、一个减法器和一个 LUT 修正单元组成,其电路图如图 4 所示. 其中 LUT 修正单元是根据式(5)补偿函数进行运算, Metric Re-Scal-

高 DSP 的数据处理速率,因此要求一条存取指令能够对多个数据进行操作。例如,数据采集处理系统采集的是字符型(8B)数据流,而进行数据交换的双端口存储器是短整型(16B)数据,利用数据打包可以一次存储和读取两个字符型(8B)数据,使访问存储器的时间减少了一半,提高了系统的实时性,这种数据处理方式对大容量数据传输的效果改善更显著。

5 结束语

文中以 TI 公司的 TMS320C67XX 芯片作为开发平台,采用双端口存储器作为数据传输通道实现双 DSP 单元的紧耦合互联,实现了一种高速数据采集处理系统,适合于大容量数据通信的双处理器系统。系统已经推广应用于某型飞机的综合训练实时评估中。实践证明,系统对多路总线数据的实时采集精度高、速度快、运行稳定可靠。

参考文献:

- [1] 张英武,郭天雷,袁国顺. 高可靠微处理器的设计[J]. 微电子学与计算机, 2009, 26(1): 59-62.

- [2] 司栋森,李增智,牛胜芬,等. 基于双 DSP 的高速数据采集控制器的设计[J]. 计算机工程, 2006, 32(22): 227-228.
- [3] 贺占庄. 一种多处理器并行计算机系统的设计[J]. 微电子学与计算机, 2006, 23(2): 198-200.
- [4] 丁双喜,张盛兵,贾宝峰. 双通道流水线 FLASH 存储系统的设计[J]. 微电子学与计算机, 2008, 25(8): 224-228.
- [5] 张洁. Blackfin DSP 在高速数据采集中的应用[J]. 电子测量技术, 2007, 30(2): 133-134.
- [6] 张雄伟,曹铁通. DSP 芯片的原理与开发应[M]. 2 版. 北京:电子工业出版社, 2000.

作者简介:

向建军 男, (1975-), 博士. 研究方向为并行计算与实时系统集成、雷达信号处理.

许蕴山 男, (1968-), 硕士生导师. 研究方向为雷达信号处理与检测.

夏海宝 男, (1970-), 硕士生导师. 研究方向为雷达信号处理与检测.

(上接第 148 页)

在功能验证无误的情况下,文中采用 Verilog HDL 语言构建了寄存器传输级模型,并采用 Synopsys 公司 DC 综合工具完成了电路的前端设计. 综合采用 SMIC 公司 0.18 μm 工艺库,时钟频率达到 132.13MHz,芯片面积约为 455 303 μm^2 ,满足目前 3G 通讯中对译码速率的要求。

5 结束语

文中通过对 Turbo 码译码算法的改进,可将译码时延缩短 1/3,译码性能没有损失,能有效地节省存储资源. 基于分块的滑动窗算法能最大限度的降低译码时延和译码功耗,并降低对存储器资源的需求,在对译码时延,译码性能要求高的通讯系统中具有较高的实用价值。

参考文献:

- [1] Peter Sweeney. 差错控制编码[M]. 俞越,张丹,译. 北京:清华大学出版社, 2004.
- [2] 刘东华. Turbo 码原理与应用技术[M]. 北京:电子工业出版社, 2004.

- [3] 冯芒,阎鸿森. 一种低功耗的 Turbo 码译码算法[J]. 西安交通大学学报, 2004, 38(10): 1081-1085.
- [4] 姚如贵,王永生,徐娟. 分组相关快衰落信道下自适应 Turbo 码译码算法研究[J]. 微电子学与计算机, 2005, 22(6): 94-97.
- [5] Dimitris Ampeliotis, PKostas Berberidis. Low complexity turbo equalization for high data rate wireless communications[J]. EURASIP Journal on Wireless Communications and Networking, 2006(2): 1-6.
- [6] Xavier Jaspas, Luc Vandendorpe. Turbo techniques for joint source-channel decoding of multimedia content [C]// Proceedings of the 2nd international conference on Mobile multimedia communications. Alghero, 2006 (7): 1-6.

作者简介:

仵林博 男, (1983-), 硕士研究生. 研究方向为信息编码和 IC 设计.

朱怡安 男, 博士生导师. 研究方向为计算机网络、并行计算.

田杭沛 男, 博士研究生. 研究方向为 DSP 体系结构.