

一种低开销双二元 turbo 译码器设计

曹 旭, 郝学飞, 胡国荣

(中国科学院 微电子研究所, 北京 100029)

摘 要: 设计一种低开销双二元 turbo 译码器, 提出了一种能够适应滑动窗算法的交织器结构, 通过与传统方案中的交织器联合使用, 大大降低了交织与解交织过程所需要的存储单元. 同时将取模归一化(modulo normalization)技术运用到双二元 turbo 译码器加比选(ACS)模块的设计上, 缩短了关键路径的延时, 提高了时钟频率和吞吐量. 采用 FPGA 对译码器进行了验证, 提出的译码器和传统的译码器相比, 存储资源节省 12%, 和使用存储器存储交织/解交织地址的译码器相比, 存储资源节省 97%.

关键词: turbo 码; 交织器; 双二元; 译码器

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2011)05-0036-04

Design of a Double Binary Turbo Decoder with Low Cost

CAO Xu, HAO Xue-fei, HU Guo-rong

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

Abstract: A low cost double binary turbo decoder was presented in this paper, with a new architecture of interleaver used. Through the combination with the interleaver used in traditional scheme, this interleaver which is applied to the sliding window algorithm can reduce memory bank needed for interleaving and deinterleaving. Modulo normalization was also used to design the add-compare-select (ACS) module of the double binary turbo decoder for high clock rate and throughput. The verification, based on FPGA, indicates the proposed decoder can reduce the memory size by 12%, comparing with the traditional one, and 97%, comparing with the decoder uses ram to store interleaving/deinterleaving address.

Key words: turbo code; interleaver; double binary; decoder

1 引言

双二进制卷积 turbo 码(DB-CTC), 是 IEEE802.16e(WiMax)标准中多载波 OFDM 系统物理层采用的前向纠错码方案之一, 与传统的单二进制卷积 Turbo 码(SB-CTC)相比, 具有编码效率高、纠错性能好以及译码时延小等优点. 然而 DB-CTC 的这种优越的性能是以译码过程的复杂计算来换取的, 与 SB-CTC 的译码相比, DB-CTC 译码需要消耗更多的硬件资源, 因而设计出低开销的硬件译码器是 DB-CTC 应用的一个关键问题.

CTC 码的编码器中, 交织器扮演着重要的角色, 它可以使 CTC 码具有近似伪随机长码的性能,

译码器的交织器需要与编码器中的交织器完全相同. 在 CTC 译码的过程中, 需要对分量码的输入输出信息进行交织/解交织, 这一过程通常是由相应的存储器(RAM)配合合适的地址产生器来完成的. 文中提出了一种新型的交织器结构, 通过和传统的交织器联合使用, 能够产生合适的读写 RAM 地址, 从而在使用一块 RAM 的情况下完成交织/解交织的过程. 在设计加比选(ACS)模块时, 使用了取模归一化处理(modulo normalization)技术, 降低了译码器关键路径的延时, 提高了时钟频率和吞吐量.

2 双二元 turbo 码的译码算法

有效地 DB-CTC 译码算法为二元 MAP 算法,

文中采用了二元 ML-MAP 算法^[1],简化了译码器的设计,同时能够满足误比特率的要求.下面简要介绍 DB ML-MAP 算法的过程:

$$\gamma_k^z(s_{k-1}, s_k) = \Lambda_{apr,k}^z(u_k = z) + \frac{L_c}{2} \cdot (y_k^{s_1} x_k^{s_1} + y_k^{s_2} x_k^{s_2} + \sum_{i=1}^m y_k^{s_i} x_k^{s_i}) \quad (1)$$

$$\alpha_k(s_k) = \max_{s_{k-1}} (\gamma_k^z(s_{k-1}, s_k) + \alpha_{k-1}(s_{k-1})) \quad (2)$$

$$\beta_k(s_k) = \max_{s_{k-1}} (\gamma_{k+1}^z(s_k, s_{k+1}) + \alpha_{k-1}(s_{k-1})) \quad (3)$$

$$\Lambda_{apo,k}^z(u_k) = \max_{s_{k-1}, s_k, u_k=z} (\alpha_{k-1}(s_{k-1}) + \gamma_k^z(s_{k-1}, s_k) + \beta_k(s_k)) - \max_{s_{k-1}, s_k, u_k=00} (\alpha_{k-1}(s_{k-1}) + \gamma_k^{00}(s_{k-1}, s_k) + \beta_k(s_k)) \quad (4)$$

$$\Lambda_{er,k}^z(u_k) = \Lambda_{apo,k}^z(u_k) - \Lambda_{apr,k}^z(u_k) - \Lambda_{in,k}^z(u_k) \quad (5)$$

$$\begin{cases} \Lambda_{in,k}^{00}(u_k) = 0 \\ \Lambda_{in,k}^{01}(u_k) = L_c \cdot y_k^{s_1} \\ \Lambda_{in,k}^{10}(u_k) = L_c \cdot y_k^{s_2} \\ \Lambda_{in,k}^{11}(u_k) = L_c \cdot (y_k^{s_1} + y_k^{s_2}) \end{cases} \quad (6)$$

公式中符号的含义分别为: γ_k^z 分支度量, α_k 前向递归路径度量, β_k 后向递归路径度量, $\Lambda_{apr,k}^z$ 先验对数似然比, $\Lambda_{apo,k}^z$ 后验对数似然比, $\Lambda_{er,k}^z$ 外部后验值, $\Lambda_{in,k}^z$ 内部先验值, u_k 时刻 k 的双二进制比特 (u_k^a, u_k^b), $x_k^{s_1}, x_k^{s_2}, x_k^{s_i}$ 代表发送的码字调制方式采用 BPSK 时 $\in \{-1, 1\}$, $y_k^{s_1}, y_k^{s_2}, y_k^{s_i}$ 是软接收码字, m 为效验比特的个数, $z \in \{00, 01, 10, 11\}$, s_k 为状态索引号.

最后根据计算出的后验对数似然比做出硬判决:

$$z = \arg_z (\max (\Lambda_{apo,k}^{00}, \Lambda_{apo,k}^{01}, \Lambda_{apo,k}^{10}, \Lambda_{apo,k}^{11})) \quad (7)$$

3 译码器的硬件结构

采用 ML-MAP 算法的译码器的整体框图见图 1, 图中有 3 个状态度量迭代单元, 它们并行工作于 3 个连续的窗口, 分别计算这 3 个窗口的前向路径度量 (α_k)、后向路径度量 (β_k) 和后向路径度量的初始值 (β_0). 为了确保一个有效的 β_0 , 文中选取窗口的大小 L 等于 24. 和文献[2]类似, 来自于信道的输入信息和外部后验值都经过一个起缓冲作用的存储单元, 用来克服存储器访问的瓶颈. 不同于文献[2]的是, 文中将计算出的分支度量 γ_k^z 存放到 4 个单口的存储器 (RAM) 中, 而不是将计算分支度量所需要的参数存放到 RAM 中, 这样在计算状态度量时就可以直接从 RAM 中读出数据, 送到加比选 (ACS) 单元中, 少去了部分组合逻辑运算, 可以减少关键路径的延时. 对一个窗口而言, 它的 α_k 首先计算出来, 存放在一个后进先出 (lifo) 的缓冲单元中. 当 β_k 计算出来时, 读出存放在 lifo 中的可用的 α_k , 和 γ_k^z 一起计算出后验对数似然比 ($\Lambda_{apo,k}^z$). 最后计算出外部后验值 ($\Lambda_{er,k}^z$) 并且通过 2 个 lifo 调整 $\Lambda_{er,k}^z$ 的顺序, 将其存放在 RAM 中, 供下次迭代使用, 这样就完成了一次完整的迭代过程.

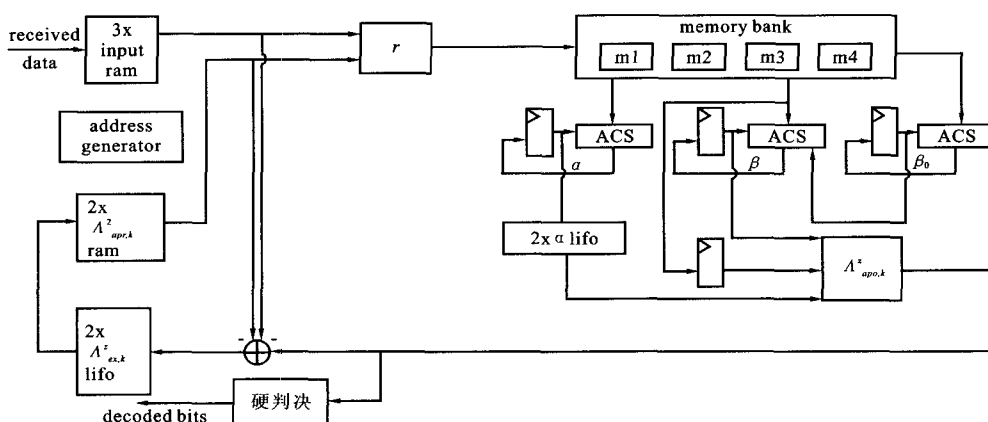


图1 Turbo 译码器的整体结构

3.1 memory bank 的工作时序

文中设计的 memory bank 的工作时序见图 2, 和文献[3]相比, 这种流水方式更加紧促, 并且计算 β_0 时对 memory bank 的读取地址也更加规律, 减少了控制逻辑. 由于在 $L \sim 2L$ 期间计算出的 β_0 为最后

一个窗口的后向路径度量所需要的初始值, 因此只需要一组寄存器将其保存, 直到计算到达最后一个窗口. (注: 在图 2 中 W 为写存储器操作, β_0 为读存储器操作用于计算 β_0 , β 为读存储器操作用于计算 β_k , α 为读存储器操作用于计算 α_k).

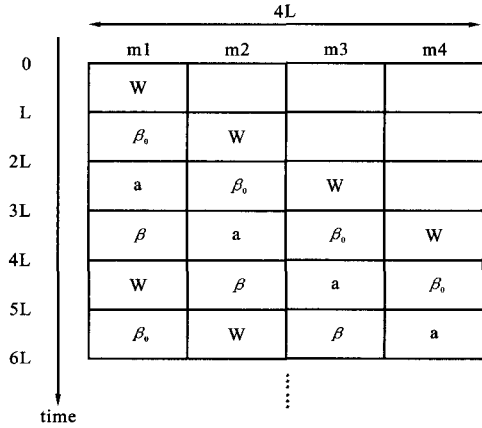


图2 存储器组的工作时序

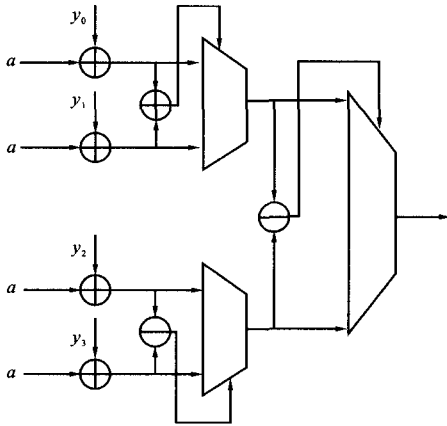


图3 加比选模块的结构

3.2 ACS 的硬件结构

文献[2]将取模归一化(modulo normalization)度量信息用于 SB-CTC 译码器,设计出具有最小路径延时的 ACS 模块,文中在文献[2]的基础上,将这种方法用于 DB-CTC 译码器,设计出来的 ACS 如图 3 所示.

和动态归一化处理与静态归一化处理^[4]相比,这种方法减少了 ACS 组合逻辑电路的延时,提高了时钟频率.由译码算法可知 α_k 和 β_k 的更新都需要用到前一时刻的值,从而形成了一个反馈环,这样就不能用流水线的方式来提高时钟频率,因此这种方法对于高吞吐量和高时钟频率的设计来讲是最佳选择.

3.3 交织器的硬件结构

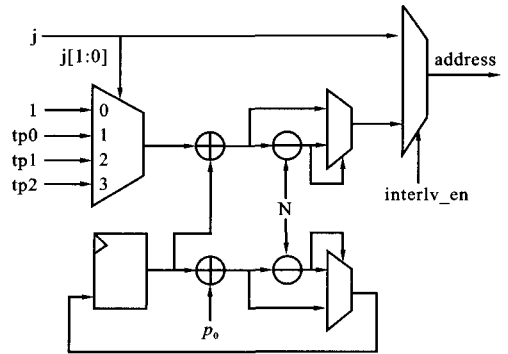
IEEE802.16e 中交织地址的产生需要 5 个参数, p_0, p_1, p_2, p_3 和 N , 这些参数根据码率和调制方式的不同而不同,具体参见文献[5]. 交织地址的产生规律可以参见表 1, 其中 j 代表交织之后的地址, $p(j)$ 代表交织之前的地址. 表中最主要的两个操作为模除和乘法, 而模除和乘法的实现都需要消耗大

的硬件资源,增加译码器的面积和关键路径的延时.

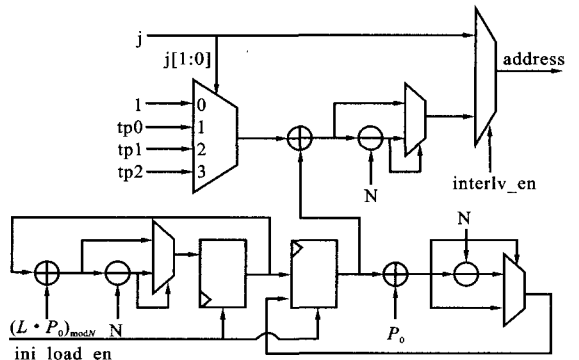
表 1 交织地址产生的方式

for $j = 0$ to $N-1$
case ($j \bmod 4$)
case0: $p(j) = (P_0 \cdot j + 1)_{\bmod N}$
case1: $p(j) = (P_0 \cdot j + 1/N/2 + P_{11})_{\bmod N}$
case2: $p(j) = (P_0 \cdot j + 1 + P_2)_{\bmod N}$
case3: $p(j) = (P_0 \cdot j + 1 + N/2 + P_3)_{\bmod N}$

由译码算法可知,在计算 $\Lambda_{er,k}^z$ 时要用到先验对数似然比 $\Lambda_{apr,k}^z$ 和内部先验值 $\Lambda_{in,k}^z$, 而从译码器的结构来看对 $\Lambda_{apr,k}^z$ 和 $\Lambda_{in,k}^z$ 的读取,都需要交织器提供相应的交织/解交织地址. 传统的交织器通常用累加器和减法器代替乘法器和除法器的使用,用以节省硬件资源和关键路径的延时. 图 4 为文中所采用的 2 个交织器,其中(a)是对文献[6]的修正,(b)为文中所设计的交织器.



(a) 交织地址为正向递增的交织器



(b) 交织地址由初始值开始递减的交织器

图 4 交织器的结构

从图 4(a)中可以看到,使用了一个累加器在每一个时钟周期用来累加 p_0 . IEEE802.16e 的 17 种模式的 CTC 编解码器中,所有的 p_0 都小于 N , 所以 $(p_0 \cdot (j-1))_{\bmod N} + p_0 \leq 2N-1$, 这样所有取模运算都可以简化为一个减法器和一个 2 选 1 的选择器. 图中的 tp_0, tp_1, tp_2 是根据参数 p_1, p_2, p_3 和 N

计算出来的介于 $0 \sim N-1$ 之间的常数, $lp + (p_0 \cdot (j-1))_{\text{mod } N} + p_0 \leq 2N-1$, 同样可以用一个减法器和一个 2 选 1 的选择器来代替模除。

图 4(a) 所示交织器能够正常工作的前提是交织地址必须从 $0 \sim N-1$ 递增。然而 β_k 的计算是一个后向迭代的过程, 从而导致 $\Lambda_{\alpha, k}^z$ 的计算并不是按照地址从 $0 \sim N-1$ 的顺序计算的, 而是按照 $L \sim 0$ 、 $2L \sim L$ 、 $3L \sim 2L, \dots$ (L 为滑动窗的大小) 的顺序计算的, 这就导致交织地址的产生方式不是地址的递增或者单纯递减, 而是以 L 为周期递减的, 并且每次递减的起始位置都不同。文中提出了一种新型的交织器结构, 以适应上述的读取顺序, 从而避免使用乘法器和除法器, 具体结构如图 4(b) 所示。

图 4(b) 在图 4(a) 的基础上增加了一个初始地址产生逻辑, 该逻辑由一个累加器和一个减法器, 以及一个 2 选 1 的选择器构成。累加器以 L 为周期, 每次增加 $(L \cdot p_0)_{\text{mod } N}$, 累加的结果再由相应的逻辑取模, 就得到了正确的初始地址。ini_load_en 是一个以 L 为周期的外部输入信号, 用于周期性的给寄存器中置入初始值。图 4(a) 中的累加器由一个减法器代替, 因为交织地址从初始值开始递减。

在具体实现译码器时, 联合使用图 4(a) 和图 4(b) 中的两种交织器, 可以用一块存储器完成交织/解交织过程, 而传统方案中单独使用图 4(a) 中的交织器需要两块存储器来完成交织/解交织, 因而这一过程节省了 50% 的存储空间。以对 $\Lambda_{\alpha, k}^z$ 的读写为例, 具体过程为, 在奇数次迭代的时候, interlv_en 有效, 读取 $\Lambda_{\alpha, k}^z$ 时使用图 4(b) 中的交织器, 实现交织的作用。而计算出来的 $\Lambda_{\alpha, k}^z$, 作为下次迭代过程的 $\Lambda_{\alpha, k}^z$ 存储的时候, 使用图 4(a) 中的交织地址产生器产生写地址, 实现解交织的作用。这样使用一块存储器就可以同时完成交织和解交织的操作。偶数次迭代的时候, interlv_en 无效, 产生的交织地址即为计数器的值。

4 综合结果与比较

采用文中所提出的设计方法, 用 verilog 语言进行 RTL 级描述, 开发环境为 quartus II, 使用 Altera 公司的 stratix II 系列芯片 EP2S90F1020C4 实现。表 2 为文中所设计的译码器, 和仅使用传统的交织器结构所设计的译码器, 以及文献[7]中用存储器存放交织/解交织地址的方式设计的译码器的性能和资源的比较。

表 2 turbo 译码器的资源和性能对比

	文中	仅使用传统交织器	文献[7]
Turbo 码的类型	DB-CTC	DB-CTC	DB-CTC
译码算法	SW	SW	SW
滑动窗大小	24	24	32
器件类型	EP2S90F1020C4	EP2S90F1020C4	EP2S90F1020C4
包长/B	60	60	60
最高时钟频率/MHz	106	106	110
存储单元/bit	43236	49050	1592064
逻辑资源	4399	4437	19148
寄存器	1715	1740	7377

5 结束语

文中设计的低开销双二元 turbo 译码器, 满足 IEEE802.16e 标准的要求, 提出的交织器结构, 能够适应滑动窗算法中交织/解交织地址的产生顺序, 和传统方案中的交织器联合使用, 能够有效减少交织/解交织过程中存储器的使用, 降低了译码器硬件资源的消耗。结果表明文中所设计的译码器和传统方案中仅使用一个交织器的译码器相比, 节省了 12% 的存储空间, 而相比用存储器存放交织/解交织地址完成交织/解交织过程的译码器, 节省了 97% 的存储空间。

参考文献:

- [1] Lin ChengHung, Chen ChunYu, Wu AnYeu, et al. Low-power memory-reduced traceback MAP decoding for double-binary convolutional turbo decoder[J]. IEEE transaction on circuits and systems, 2009, 56(5): 1005-1016.
- [2] Benkeser C, Burg A, Cupaiuolo T, et al. Design and Optimization of an HSDPA turbo decoder ASIC[J]. IEEE Journal of solid state circuits, 2009, 44(1): 98-106.
- [3] Wu ChienMing, Shieh MingDer. VLSI architectural design tradeoffs for sliding-window log-MAP decoders[J]. IEEE transaction on very large scale integration systems, 2005, 13(4): 439-447.
- [4] Liu Haisheng, Diguett JeanPhilippe, Jego Christophe, et al. Energy efficient turbo decoder with reduced state metric quantization[J]. IEEE workshop on Signal Processing Systems, 2007: 237-242.
- [5] LAN/MAN standards committee. part 16: air interface for fixed broadband wireless access system[S]. New York: IEEE, 2004.

如图 6、图 7 所示,在样本数据较小的情况下,文中方法不如 EPM 算法精度高;而在样本数据较大的情况下,文中方法具有较高的优势。

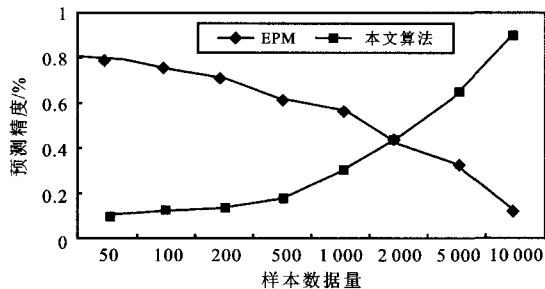


图 6 样本数据量对预测精度的影响

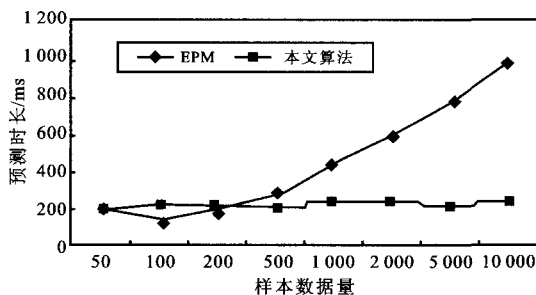


图 7 样本数据量对预测时长的影响

5 结束语

实验结果表明,该算法能够高效地记录并描述数据流的变化态势,并对数据流在下一时刻的到来数据进行预测.它适用于类似金融、气象等具有连续性,且前后变化幅度可控的情况,及要求预测效率较高,但预测时长较短的情况;特别在样本数据量较大的情况下,文中方法具有较高的优势.相较其他同类算法而言,该算法的精度和预测成功率都较高.但该算法仍存在局限性和不足,如对于变化幅度较大且不连续的数据流预测效率较低;不适用于需要预测出未来较长时间内情况变化的数据;另外,有向图构造时,采用的是经典数据流聚类方法,该方法本身仍存在不足,可进一步改进以提高基于有向图构造的

数据流预测算法效率。

参考文献:

- [1] Domingos P, Hulten G. Mining high-speed data streams[J]. Knowledge Discovery and Data Mining, 2000;71-80.
- [2] Dong Guozhu, Han Jiawei, Laks V S, et al. Online mining of changes from data streams: research problems and preliminary results[C]// Workshop on Management and Processing of Data Streams. In cooperation with ACM SIGMOD/PODS and FCRC. San Diego, CA, USA: ACM, 2003.
- [3] 刘耀宗,王湛. 数据流的预测与分类研究[J]. 计算机科学, 2007, 34(11): 170-173.
- [4] 刘海峰,刘守生,姚泽清,等. 一种基于改 K-means 聚类的文本特征选择模型[J]. 微电子学与计算机, 2009 (6): 29-31.
- [5] 周黔,吴铁军. 一种动态数据流的实时趋势分析算法[J]. 控制与决策, 2008(10): 1182-1185.
- [6] Yu Meng, Margaret H Dunham. Efficient mining of emerging events in a dynamic spatiotemporal environment[J]. Lecture Notes in Computer Science, 2006 (3918): 750-754.
- [7] Aggarwal J, Han J, Yu P S, et al. A framework for clustering evolving data streams[C]// Proceedings of the 29th International Conference on Very Large Data Bases (VLDB) Berlin, Germany: VLDB, 2003: 81-92.
- [8] 王树广. 分布式数据流上的连续异常检测[J]. 微电子学与计算机, 2008, 50(9): 158-160.

作者简介:

由欣女, (1983-), 硕士研究生. 研究方向为计算机软件与理论数据库与知识库.

印桂生男, (1964-), 教授, 博士生导师. 研究方向为数据库、虚拟现实、信息安全.

(上接第 39 页)

- [6] Yen ShaWei, Hu MingChih, Chen ChihLung, et al. A 0.92mm² 23.4mW fully-compliant CTC decoder for WiMAX 802.16e application [C]// Custom Integrated Circuits Conference, CICC '09. San Jose, CA: IEEE, 2009: 191-194.
- [7] 吕闻. 基于 FPGA 的高效 CTC 译码器的设计和实现[D]. 西安: 西北工业大学, 2007.

作者简介:

曹旭 (1985-), 硕士研究生. 研究方向为专用集成电路设计与系统应用.

郝学飞 (1976-), 副研究员. 研究方向为无线通信算法, 无线通信芯片设计.

胡国荣 (1966-), 研究员. 研究方向为多媒体通信, 信道信源芯片设计.