

Wimax 系统双二进制 Turbo 码的译码算法及 VLSI 实现

李江鹏, 刘海洋, 陈 杰

(中国科学院微电子研究所 通信与多媒体 SoC 实验室, 北京 100029)

摘 要: 设计了一款应用于 World Interoperability for Microwave Access (Wimax) 系统的双二进制 Turbo 码译码器。该译码器对传统 Max-log-MAP 译码算法进行改进, 在增加很少计算复杂度下有效地补偿了传统算法中 max 计算带来的误差。此外, 提出了一种低复杂度, 适用于 Wimax 系统中所有 Turbo 码码长的通用交织器结构。仿真结果表明, 改进的 Max-log-MAP 译码算法在误码率 10^{-4} 下相对于传统算法获得了约 0.35~0.4 dB 的译码增益; 提出的通用交织器结构明显降低了译码器的计算复杂度和面积, 提高了系统的吞吐率。该译码器可以在 200 MHz 工作频率下得到 20.91 Mb/s 的吞吐率, 完全满足 Wimax 系统数据率的要求。

关键词: Max-log-MAP 算法; 通用交织器; 双二进制 Turbo 码; Wimax

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2011)03-0043-05

Algorithm and VLSI Implementation for Double-Binary Turbo Codes in Wimax

LI Jiang-peng, LIU Hai-yang, CHEN Jie

(Department of Communication and Multimedia SoC, Institute of Microelectronic
of China Academy of Science, Beijing 100029, China)

Abstract: This paper designs a double-binary Turbo decoder applied in Wimax system. The decoder efficiently compensates the error resulted from the max calculation in the Max-log-MAP algorithm by amending the conventional algorithm with only small increase in complexity. In addition, a new generalized interleaver structure which satisfies all codeword lengths in Wimax is proposed. Simulation results show that, the amending algorithm can offer approximately 0.35 to 0.4 dB gain compared with the conventional algorithm at bit error rate (BER) of 10^{-4} . The proposed interleaver structure significantly reduces the computational complexity and areas. A double-binary turbo decoder is designed for Wimax can achieve the throughput of 20.91 Mb/s at the operation frequency of 200 MHz, which is sufficient for the data rate of the system.

Key words: Max-log-MAP algorithm; low-complexity interleaver; double-binary turbo codes; Wimax

1 引言

法国学者 C. Berrou 在 1993 年首次提出 Turbo 码。仿真结果表明, 在 10^{-5} 误码率下 Turbo 码与香农极限仅有 0.5 dB 差距^[1]。双二进制 Turbo 码在性能上比传统二进制 Turbo 码有更多的优势, 目前已被 Wimax、DVB-RCS 以及第三代移动通信系统采用^[2]。

最大后验概率 (MAP) 算法是 Turbo 码的最优译码算法, 但是由于复杂度过高而难以实现。绝大多数 Turbo 码译码采用次优 Max-log-MAP 算法。在该算法中, max 计算代替了 MAP 算法中的指数运算, 从而降低了计算复杂度, 但是会造成译码性能损失^[3-4]。为了弥补这一损失, 文献[4-5]中提出了一些经典的改进算法。但是, 这些改进算法都需要额外的乘法运算, 从而导致了译码复杂度和译码器面积的

增加。

在 Turbo 译码器实现方面,交织器对译码器的面积和速度起了重要的作用^[6]。Wimax 系统中支持 12 种码长的双二进制 Turbo 码,对于这 12 种码长,交织器深度互不相同,从而导致设计复杂度和译码器面积大大增加。

本文提出了一种改进 Max-log-MAP 译码算法的双二进制 Turbo 译码器的 VLSI 结构。该译码器对传统 Max-log-MAP 译码算法进行改进,在很少增加计算复杂度下有效地提高了译码性能。此外,本文设计了一款适用于 Wimax 系统中所有码长的双二进制 Turbo 码的低复杂度交织器结构,有效地降低了译码器的复杂度和面积。

2 改进的 Max-log-MAP 算法

双二进制 Turbo 译码是由两个软输入软输出(SISO)译码器输出的外部信息,通过交织模块和解交织模块串联起来的迭代过程。其中交织模块包括输入数据交织器,外部信息交织器;解交织模块为外部信息解交织器。对于 Max-log-MAP 算法,双二进制 Turbo 译码中第 k 个符号的三个后验似然比(PLLR)可表示为

$$L_i^r(\hat{u}_k) = \max_{S_{k+1}, S_k, i} (\gamma_k^i(S_{k-1}, S_k) + \alpha_{k-1}(S_{k-1}) + \beta_k(S_k)) - \max_{S_{k-1}, S_k, 0} (\gamma_k^i(S_{k-1}, S_k) + \alpha_{k-1}(S_{k-1}) + \beta_k(S_k)) \quad (1)$$

式中, $i \in \{01, 10, 11\}$, S_k 为时刻 k 的译码器状态, $\gamma_k^i(S_{k-1}, S_k)$ 表示在输入 $u_k = i$ 时状态 S_{k-1} 到 S_k 的分支度量, $\alpha_{k-1}(S_{k-1})$ 表示状态 S_{k-1} 的前向度量, $\beta_k(S_k)$ 表示状态 S_k 的后向度量。若 x_k 表示 k 时刻的编码符号, S_1 和 S_2 表示双二进制输入符号的系统位, P_1 和 P_2 表示双二进制输入符号的校验位, 则

$$\alpha_k(S_k) = \max_{S_{k-1}, S_k} [\gamma_k(S_{k-1}, S_k) + \alpha_{k-1}(S_{k-1})] \quad (2)$$

$$\beta_{k-1}(S_{k-1}) = \max_{S_{k-1}, S_k} [\gamma_k(S_{k-1}, S_k) + \beta_k(S_k)] \quad (3)$$

$$\gamma_k(S_{k-1}, S_k) = (x_k^{S_1} y_k^{S_1} + x_k^{S_2} y_k^{S_2} + x_k^{P_1} y_k^{P_1} + x_k^{P_2} y_k^{P_2})/2 + L_e^i \quad (4)$$

在式(4)中, L_e^i 表示从上一个 SISO 译码器得到的外部信息。此外部信息作为上一次迭代的后验似然比的近似值,也是本次迭代中的先验似然比的近似值。

在 Max-log-MAP 算法中,由于将 $\max^*$ 运算采用 $\max$ 近似,导致性能上明显下降。文献[4-5]为弥补其性能损失提出了一些改进算法,这些改进算法

均是基于如下关系式^[3]:

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

为避免非线性运算项 $\ln(1 + e^{-|x-y|})$,通常是对外部信息乘以一个小于 1 的常数系数^[5]。为了兼顾性能和硬件实现,此系数常取为 0.7。这样,在每次迭代中需要三次额外的乘法运算。

本文提出一种对传统 Max-log-MAP 算法迭代间外部信息平滑滤波的修改方法来弥补由于 $\max$ 计算而带来的误差。通过前面的分析可知,由于 $\max$ 运算导致式(1)~(3)的计算产生误差,从而使得外部信息产生误差。而外部信息作为下一次迭代的先验信息,从而使得这种误差在迭代中传递。因此,可以通过检测外部信息的最大值并对其进行抑制误差的传递。若信息序列可以被正确译码,外部信息最大值的位置经过最初的几次迭代后应固定。

假设 (i) th 表示第 i 次迭代过程,在迭代中外部信息的最大值可如下表示:

$$\text{the } (i-1) \text{ th: } \max(L_{1,i-1}(\hat{u}_k), L_{2,i-1}(\hat{u}_k), L_{3,i-1}(\hat{u}_k)) = L_{r,i-1}(\hat{u}_k) \quad (6)$$

$$\text{the } (i) \text{ th: } \max(L_{1,i}(\hat{u}_k), L_{2,i}(\hat{u}_k), L_{3,i}(\hat{u}_k)) = L_{m,i}(\hat{u}_k) \quad (7)$$

$$\text{the } (i+1) \text{ th: } \max(L_{1,i+1}(\hat{u}_k), L_{2,i+1}(\hat{u}_k), L_{3,i+1}(\hat{u}_k)) = L_{n,i+1}(\hat{u}_k) \quad (8)$$

式中, r, m, n 分别代表三个外部信息对应最大值的位置。正确译码时,经过几次迭代后 r, m, n 应相等,即最大值的位置稳定。反之, r, m, n 的值不相同,即最大值的位置不断变化。假设译码过程中,某位置对应的三个外部信息在第 i 次迭代时最大值位置为 m , 第 $i+1$ 次迭代时最大值位置为 n , 则在第 $i+1$ 次迭代后对外部信息做如下处理:

$$\hat{L}_{i+1}^e(\hat{u}_k) = \begin{cases} L_{m,i+1}^e(\hat{u}_k) & m = n \\ (L_{m,i+1}^e(\hat{u}_k) + L_{n,i}^e(\hat{u}_k))/2 & m \neq n \end{cases} \quad (9)$$

式(9)的操作是将本次迭代的输出外部信息(即下一次迭代的先验信息)的值与上一次的值进行平滑。对于正确译码,这种趋势会在下一次迭代中得到巩固。反之,对于错误译码,会使得最大值幅度降低,从而有效地抑制了这种错误的传递。在译码器实现中,仅在 $m \neq n$ 时用一次比较操作和一次加法操作即可完成整个操作,避免了传统补偿算法中乘法运算。在硬件实现中,移位操作可以在加法器输出时直接将最后一位舍去来完成,不占用任何额外资源。由于对外部信息的自适应平滑滤波操作在本次更新外部信息前操作,这样当前外部信息存储器中

存储的恰好是上一次迭代的外部信息,故不需要额外的资源来存储上一次外部信息值。

3 Wimax 中双二进制 Turbo 译码器的结构设计及实现

Turbo 译码中,后向度量的计算在前项度量和分支度量后进行。这种串行计算导致译码器的吞吐率降低,并且需要大量的存储单元。滑窗技术解决了这个问题^[4]。在滑窗结构中,一个长的码字被分为几部分相同长度的短码字(窗口),Max-log-MAP 算法分别应用在这几部分码字中完成译码。

3.1 通用交织器设计

在滑窗结构中,窗长是一个重要的参数。如果窗长过长,会导致吞吐率降低和大量存储单元的需求;若窗长过短会导致译码性能的降低。对于 Turbo 码,通常将窗长取为 32^[4]。但在 Wimax 系统中,并不是所有的 Turbo 码长均是 32 的倍数。故在本文中采用变长窗长的结构,其窗长值的选取依据是应尽量在 32 附近以保证译码性能与电路性能的折中。

本文提出的通用交织器结构即是按照表 1 中窗长参数设计的。由于译码算法在每个窗口内单独进行,故交织器的深度与窗口长度对应。交织规则如图 1 所示。

表 1 Wimax 系统不同码长码字对应的滑窗长度 bit

码长	滑窗长度	码长	滑窗长度
24	24	120	30
36	36	144	36
48	48	180	36
72	36	192	32
96	32	216	36
108	36	240	40

由此可知,对于不同的交织深度需要进行的取余需要用除法运算来实现。由于滑窗长度最长为 48,故若使用除法器则需要 6 位除法器。然而除法器会消耗巨大的电路资源并导致数据吞吐率下降^[7]。为解决此问题,本文提出一种避免除法运算的通用 Wimax 系统 Turbo 码所有码长的低复杂度交织器结构。从图 1 的交织规则中可知,输入交织器数据的地址是顺序的,因此可以对上一次的取余值加 P_0 得到本次的取余值。此时,只需要求得初始的第一个输入数据地址的取余值。根据方程:

```

Step 1:
for i=0...N-1
    if(mod2=1) (Ai,Bi)=(Bi,Ai)
    else      (Ai,Bi)=(Ai,Bi)
Step2:
for i=0...N-1
    switch i mod 4:
    case 0: P(i)=(P0*i+1) mod N
    case 1: P(i)=(P0*i+1+N/2+P1) mod N
    case 2: P(i)=P0*i+1+N/2+P2 mod N
    case 3: P(i)=(P0*i+1+N/2+P3) mod N

```

图 1 Wimax 系统中 Turbo 码的交织规则

$$(a+b) \bmod N = (a_{\bmod N} + b_{\bmod N}) \bmod N \quad (10)$$

可将图 1 中 step2 表示为

$$((P_0 * j)_{\bmod N} + \text{init}_{\bmod N})_{\bmod N} = (Q_1 + Q_2)_{\bmod N} \quad (11)$$

根据 Wimax 协议^[8-9],可知上式中 init 的值的范围是 $(0, N)$, N 为码块的长度。这样 Q_2 的值就是 init 的值。init 的求值可通过加法器来完成。 Q_1 的实现需要考虑交织和解交织的差别。在交织器的实现中,图 1 中 i 的第一个值为 0,故 Q_1 的初值为 0,而 0 对任何值取余均为 0。这样可以通过每次对上一步 Q_1 值加 P_0 的操作来得到后续的 Q_1 值。在解交织运算中由于 i 的初值为窗口长度减 1 而使 Q_1 的实现复杂。如果得到第一个 Q_1 的值,那么后续的 Q_1 值可以通过对 $Q_1 - P_0$ 操作得到。 Q_1 的初值通过 $\frac{P_0 * (k * N - 1)}{N}$ 得到。其中 $k * N - 1$ 表示每个窗口中第一个进入解交织操作数据的地址。 k 的范围是 $(0, 1)$ 。如果 $k(P_0 - 1)$ 为整数,可得结果为 $N * k - P_0$ 。这样在实现中可以将不同的码块长度以及不同的 P_0 所对应的 Q_1 初值 $N * k - P_0$ 值存入一个查找表(LUT)。根据表 1 中不同码块长度对应窗长的计算,除 192 和 108 外的其余码块长度均满足 $k(P_0 - 1)$ 为整数。对于 192 和 108 这两种码长,该结果为 $N(1-k) - P_0$,可将此值存入上面的 LUT 中。这样,在解交织器中 Q_1 值得以求解。

由于 Q_1 和 Q_2 均为对 N 取余后的结果,其范围皆为 $(0, N)$, 所以 $Q_1 + Q_2$ 范围是 $(0, 2N)$ 。这样 $\text{mod}(Q_1 + Q_2)N$ 可以通过 $Q_1 + Q_2$ 与 N 比较得到结果。交织的第一步操作可以在第二步完成后根据其地址最低位(LSB)来完成。交织器和解交织器的电路结构如图 2 所示。

图 2(a)中的 Init LUT 存储不同码长的 init0, init1, init2, init3 的值。此交织器与解交织器避免了除法器的使用。除法器会消耗非常大的电路资源,

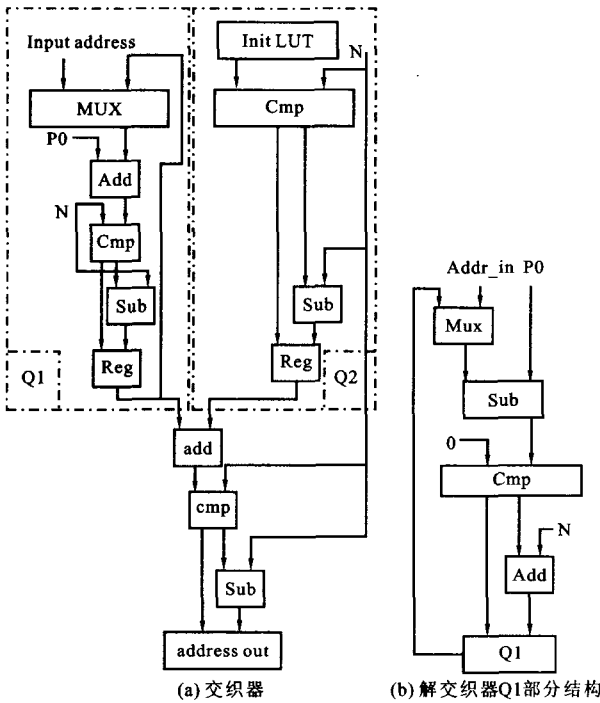


图 2 Turbo 译码中交织器与解交织器电路结构

Wimax 系统中需要的 6 位除法器会有上万门^[7], 并且一次除法运算在关键路径较短时需要很多时钟. 本文提出的交织器结构每次计算仅需 2 个时钟, 并且采用流水结构, 故在速度和面积上都对电路进行了优化.

3.2 双二进制 Turbo 译码器的实现

相比于文献[4-5]中的改进算法, 本文算法仅需一个加法器和一次移位操作, 在计算复杂度和电路资源消耗上均有优势. 译码器的仿真环境如表 2 所示, (q, f) 表示 q 比特量化, 其中 f 比特为小数.

表 2 仿真环境

SISO 算法	改进 Max-log-MAP 算法
量化方式	输入符号(8,4)
	前向/后向/状态度量(13,4)
	外部信息(13,4)
	后延概率信息(14,4)

基于本文提出的算法和通用交织器结构, 双二进制 Turbo 译码器的 VLSI 实现结构如图 3 所示.

综合考虑数据吞吐率与电路资源, 本文在译码器结构上采用单个 SISO 电路单元串行计算. 其中两个 SISO_ctrl 模块为一次迭代两次 SISO 计算的配置单元, 外部信息的交织模块和解交织模块即在这两个配置单元中. Data_in_ Interleaver 为输入数据的交织模块. Para 模块存储不同码长对应的参数. Judge 模块为最终的译码判决单元. 为提高数据

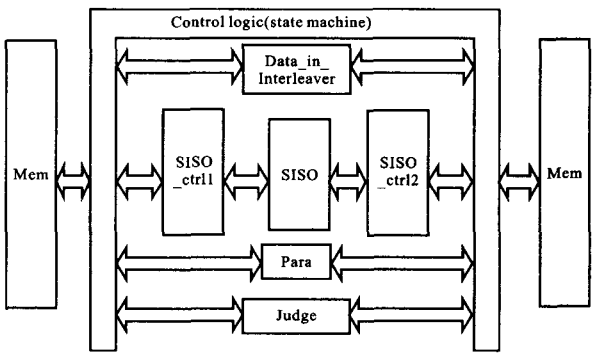


图 3 双二进制 Turbo 译码器结构

吞吐率, 本文将前向度量/分支度量(SISO_fsc)计算和后向度量/外部信息(SISO_bec)计算分开, 通过两块双口 SRAM 对数据的缓存从而保证 SISO_fsc 与 SISO_bec 两模块间的流水结构, 如图 4 所示. 为避免溢出, 在每次计算前、后向度量后均对其做相对状态零值的归一化操作.

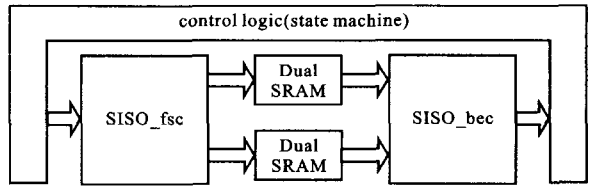


图 4 SISO 模块的流水结构

4 仿真与综合结果

为评估设计的译码器的性能, 本节对提出的译码器与文献[5]中提出的改进算法的译码器以及传统算法译码器的误比特率(BER)进行了仿真. 仿真条件为: 1/2 码率, 480 比特码块长度, AWGN 信道. 仿真结果如图 5 所示. 可以看出, 在误比特率为 10^{-4} 时, 本文译码器相对于传统算法译码器获得约 0.35~0.4 dB 的增益, 相对于文献[5]中提出的改进算法译码器有约 0.1 dB 的性能损失. 但本文译码器的改进算法仅需要一次额外的求和及移位运算, 避免了文献[5]中的乘法计算, 降低了译码复杂度, 节省了电路面积. 译码器各部分的电路面积如表 3 所示.

表 3 译码器电路面积

模块名称	面积/ μm^2	电路门数
Input_int	17 832.8	1 789
SISO_ctrl1	28 963.3	2 905
SISO_ctrl2	79 590.2	7 983
SISO	582 200.2	58 395
Para	1 560.1	156
Judge	23 737.5	2 380
总计	733 884.1	73 608

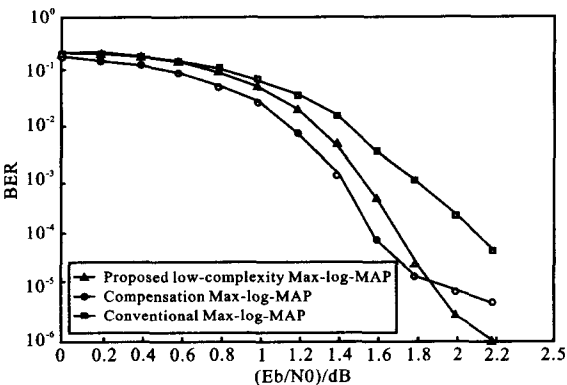


图 5 码长为 480 比特的 Turbo 码误比特率性能曲线

表 5 给出了 SISO 模块各部分的资源. 本文提出的交织器电路面积如表 6 所示. 可以看出, 本文的设计相对于传统的采用除法器的设计得到很大改善. Turbo 译码器整体电路规模约 73 608 门(不含存储器), 关键路径延时为 4. 87 ns. 单口 SRAM 的存储器资源为 24 064 bit, 双口 SRAM 的存储器资源为 19 968 bit. 对于 1/2 码率下最长码长(480 bit), 进行 8 次迭代所需 4 591 的时钟周期, 故 Wimax 系统的 Turbo 译码器可以达到 20. 91 Mb/s 的吞吐率.

表 5 SISO 内部各模块面积

模块名称	面积/资源	电路门数
SISO_fsc	310 539. 1 μm^2	31 147 gates
SISO_bec	271 670. 2 μm^2	27 248 gates
Dual-SRAM	19 968 bits	

表 6 本文设计交织器的电路面积

模块名称	面积/ μm^2	电路门数
Interleaver	12 121. 4	1 216
Leinterleaver	12 121. 4	1 216
De_Leinterleaver	14 373. 4	1 442

5 结束语

文中设计了一款应用与 Wimax 系统的 Turbo 码译码器. 在该译码器中, 对传统 Max-log-MAP 算法进行了改进, 通过对连续两次迭代中的外信息进行平滑, 在增加很少运算复杂度的条件下提高了译码性能. 在误比特率为 10^{-4} 时, 本文的修改算法相对于传统算法获得约 0. 35~0. 4 dB 的增益. 同时, 设计了一款适用于 Wimax 系统所有码长的 Turbo 码译码的通用交织器. 由于其通用性以及避免除法运算从而降低了译码复杂度, 节省了电路面积并提

高了译码器的吞吐率. 综合结果表明, 文中设计的双二进制 Turbo 译码器在工作频率为 200 MHz 下可以达到 20. 91 Mb/s 的吞吐率, 完全满足 Wimax 系统数据率的要求.

参考文献:

[1] Berrou C, Glavieux A, Thitimajshima P. Near shannon limit error correcting coding and decoding: Turbo codes [C]// IEEE International Conference on Communications. Geneva: IEEE, 1993:1064-1070.

[2] 韦耿, 刘文予, 冯斌. Rayleigh 信道下 Turbo 码误比特率模型. 微电子学与计算机, 2008, 25(3): 18-20.

[3] Viterbi A J. An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes[J]. Selected Areas on Communications, 1998, 16 (2): 260-264.

[4] Zhan C, Arslan T, Erdogan A T, et al. An efficient decoder scheme for double binary circular turbo codes [C]// Proceedings 2006 IEEE International Conference on. Acoustics, Speech and Signal Processing. Goulouse: IEEE, 2006: 229-232.

[5] Vogt J, Finger A. Improving the max-log-MAP turbo decoder[J]. Electronics Letters, 2000, 36(23): 1937-1939.

[6] 方永权, 杜兴民, 陶祁. Turbo 码快速交织器的设计及其 CPLD 实现[J]. 微电子学与计算机, 2007, 22(8): 69-75.

[7] Harris D L, Berman S F, Horowitz M A. SRT division architectures and implementations[C]// The 13th IEEE Symposium on Computer Arithmetic. Pacifid Grove: IEEE, 1997: 18-25.

[8] IEEE standard for local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems[S]. Piscatauay, NJ: IEEE Computer Society and IEEE Microwave Theory and Techniques Society, 2007.

[9] Kim J H, Pard I C. Double-binary circular turbo decoding based on border metric encoding[J]. IEEE Transactions on circuits and systems, 2008, 55(1): 79-83.

作者简介:

李江鹏 男, (1984-), 硕士研究生. 研究方向为通信系统中物理层关键算法设计与 VLSI 实现.

刘海洋 男, (1980-), 博士. 研究方向为通信基带算法与 SoC 设计.

陈杰 男, (1963-), 研究员. 研究方向为通信与多媒体信号处理 SOC.