

高码率 LDPC 码译码器的优化设计与实现

张靖琳 刘荣科 赵岭

(北京航空航天大学电子信息工程学院 北京 100083)

摘要: 本文以 CCSDS 推荐的 7/8 码率 LDPC 码为例, 提出了一种适于高码率 LDPC 码译码器的硬件结构优化方法。高码率的 LDPC 码通常也伴随着行重与列重的比例较高的问题。本方法是在拆分校验矩阵的基础上, 优化常用的部分并行译码结构, 降低了高码率 LDPC 码译码时存在的校验节点运算单元(CNU)与变量节点运算单元(VNU)之间的复杂度不平衡, 并由此提高了译码器的时钟性能。实验证明, 本文方案提供的结构与常用的部分并行译码结构相比, 节省硬件资源为 41%; 采用与本文方案相同的硬件资源而未经矩阵拆分的部分并行译码方案的码速率为本文方案的 75%。

关键词: LDPC 码; 译码器; 优化设计

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2009)01-0083-04

Optimized Decoder Design and Implement for High Rate LDPC Codes

Zhang Jing-lin Liu Rong-ke Zhao Ling

(School of Elec. and Infor. Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: This paper brings up with a hardware structure optimized method which is suitable for high code rate LDPC decoder, for example, CCSDS recommended LDPC (Low-Density Parity-Check code) code with code rate of 7/8. LDPC code with high code rate is usually concomitant with problem that row weight is far larger than the column weight. This optimized method is based on check matrix splitting, it also optimizes the common components of parallel decoder structure, and reduces complexity imbalance between Check Node processing Units (CNU) and Variable Node processing Units (VNUs) existed in high code rate LDPC decoder. Thus, clock performance of the decoder is improved. Experiment has proved, compared with usual partial parallel decode structure, structure provided by this paper saves 41% hardware resources, and the code rate of partial parallel decode structure which adopts the same amount of hardware resources is just 75% than code rate of the structure in this paper.

Key words: LDPC codes; Decoder; Optimized design

1 引言

21 世纪是太空的时代, 是航天技术、空间技术飞速发展的世纪, 作为深空探测的基础和中心问题之一, 深空通信的能力将最终决定深空探测技术的发展。众所周知, 改进信道编译码技术、数据压缩技术可以更经济有效地改善深空通信系统的性能, 因此, 设计和优化信源压缩、信道编码和调制已经成了深空通信中对抗路径损失的一个重大突破口。

在 1996 年被重新发现后^[1,2], 经过各方面的深入研究, LDPC 码的优越性能和实用性开始被人们深刻认识, LDPC 码也逐渐开始出现在各种通信系统的应用中。空间数据系统咨询委员会(CCSDS)继 1999 年提出了空间通信应用中 Turbo 码标准后, 于 2006 年 8 月提出了用于近地通信的 LDPC 码的建议^[3]。由此可见, 选用 LDPC 码作为信道编码, 将成为未来的空间通信中的新趋势。

在 LDPC 码的实现中, 译码算法和译码器设计是决定码字性能和应用前景的重要因素之一。目前, BP 算法^[1]作为 LDPC 码的一种经典译码算法被人们广泛使用。在应用中, 人们需要根据具体码字和系统需求, 对 BP 算法做出相应的改进或调整, 使译码器在复杂度适当的条件下, 向最优译码性能靠近^[4]。

应用 FPGA 对上述的 CCSDS 推荐的 LDPC 码译码器进行硬件实现时, 若采用传统的部分并行结构^[4,5]完成译码器设计, 将存在以下问题: (1)高码率一般也将带来较高的行重与列重的比例, 使 CNU 模块与 VNU 模块规模出现严重不平衡, 这也将引起译码器时钟性能的降低; (2)传统的部分并行结构的 BP 算法中, CNU 与 VNU 的一次运算分别占用一次迭代周期的 1/2, 硬件时间利用率仅为 50%, 存在较大的硬件资源的浪费。

本文提出了一种基于拆分校验矩阵的改进的部分并行结构的 LDPC 码译码器设计方法, 以 CCSDS 推荐的 7/8 码率 LDPC 码为例, 经实验, 验证了该方法的有效性。

2 基于 Log-BP 算法的译码器设计

2.1 CCSDS 推荐的 7/8 码率 LDPC 码

CCSDS 于 2006 年 8 月在以 Orange Book(Experimental Specification)^[3]的形式发布了一项 NASA/GSFC 对于近地任务中的信道编码技术的研究成果。该文件提供了一种适用近地应用的高数据率和高可靠性的准循环 LDPC(QC-LDPC)码^[6, 7]——(7154, 8176)码, 用于定义该 7/8 码率的码字的校验矩阵, 即 \mathbf{H} 矩阵, 如下所示。

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,16} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,16} \end{bmatrix}$$

\mathbf{H} 矩阵由 2 行 \times 16 列共 32 个子矩阵构成, 每个子矩阵均为 511 行 \times 511 列的循环行列式的方阵, 由两个 I 矩阵的循环行列式叠加组成, 所以 \mathbf{H} 矩阵的行重为 $2 \times 16 = 32$, 列重为 $2 \times 2 = 4$, 每行每列的重量恒定, 比例为 $32:4=8:1$, 该 LDPC 码为规则码。

2.2 传统的部分并行结构的 Log-BP 算法的译码器实现

Log-BP(对数似然比域内的 BP)算法是 BP 算法在对数似然域内的一种简化算法, 它可以将 BP 算法所需的乘法运算转化为加法运算, 使其在不降低性能的基础上更适于硬件实现。

式(1)~式(7)表述了 log-BP 算法的迭代流程, 译码器的输入为来自信道的接收软信息, 输出为译码硬判决结果。其中, $p_n^0 = p(x_n = 0)$ 和 $p_n^1 = p(x_n = 1) = 1 - p_n^0$ 为先验概率, m 表示校验节点, j 表示变量节点。令 $M(j)$ 表示与变量节点 j 相连的校验节点的集合, $N(m)$ 表示与校验节点 m 相连的变量节点的集合。具体步骤为

初始化:

$L(q_{mj})$ 初始化为信道输入 LLR, 即

$$L(q_{mj}) = L(c) = \ln(p_n^0 / p_n^1) = -2r_j / \sigma^2, \forall m \in M(j), \forall j \quad (1)$$

迭代过程:

(1)水平/行过程(Check Node Processing, CNU): 对所有 m, n , 计算

$$A_{mj} = \sum_{\substack{n \in N(m) \\ n \neq j}} \Psi(L(q_{mn})) \quad (2)$$

$$S_{mj} = \prod_{\substack{n \in N(m) \\ n \neq j}} \text{sgn}(L(q_{mn})) \quad (3)$$

$$R_{mj} = -s_{mj} \Psi(A_{mj}) \quad (4)$$

其中

$$\Psi(x) = \ln(\tanh(|x|/2)) = \ln((1 - e^{-|x|}) / (1 + e^{-|x|}))$$

(2)垂直/列过程(Variable Node Processing, VNU): 对所有 m, n , 计算

$$L(q_j) = \sum_{m \in M(j)} R_{mj} + \frac{-2r_j}{\sigma^2} \quad (5)$$

$$L(q_{mj}) = L(q_j) - R_{mj} \quad (6)$$

判决步骤:

(a)对 $L(q_j)$ 判决:

$$x_n = \begin{cases} 1, & L(q_j) \geq 0 \\ 0, & L(q_j) < 0 \end{cases}, \quad n = 1, 2, \dots, N \quad (7)$$

(b)若 $\mathbf{H} \cdot \mathbf{x} = 0$, 译码正确; 否则继续迭代直到最大迭代次数。

根据以上的算法, 采用传统的部分并行结构设计译码器, 按图 1 的 \mathbf{H} 矩阵中子矩阵的个数, 将行并行度设为 2, 列并行度设为 16, 则译码器需要 2 个 CNU 运算单元并行实现式(2)~式(4)的水平过程运算, 16 个 VNU 运算单元并行实现式(5)和式(6)的垂直过程运算。每一个 CNU 或 VNU, 对 \mathbf{H} 的每一个子矩阵内部的 511 行或列而言, 是串行运算的, 并且 CNU 与 VNU 运算之间采用交叠结构^[8]。因此, 完成一次迭代需要 $2 \times 511 = 1022$ 个时钟周期。

若设时钟频率为 clk , 总迭代次数为 n , 则译码速率 $= (\text{clk} \times 8176) / (2 \times 511 \times n) = \text{clk} \times 8 / n (\text{bps})$ 。

复杂度方面, 上述结构的译码器需要的主要硬件资源如表 1 所示。

表 1 常用的部分并行译码结构所需的主要硬件资源

32 输入 CNU	4 输入 VNU	迭代变量存储器	查找表(LUT)
2 个	16 个	64 个	128 个

表 1 中, 迭代变量存储器用于存储迭代运算的结果, 64 对应 \mathbf{H} 矩阵的子矩阵个数, 每个存储器大小为 $512 \times m$, m 为量化比特数; 查找表 LUT 用于实现式(2)和式(4)的双曲正切函数。

CNU 为具有 32 个输入的复杂运算单元, 而 VNU 为具有 4 个输入简单运算单元, 它们的具体运算结构, 分别根据式(2)~式(4)及式(5), 式(6)设计。译码时, 由于 CNU 模块与 VNU 模块的复杂度严重不平衡, 需要安排 CNU 所插入的流水线级数 VNU 的, 或牺牲 VNU 的时钟性能维持与 CNU 相同的运算时间, 这影响整个译码器的迭代运算速度或时钟性能, 降低译码效率。

3 译码器的优化设计与实现

为了解决上述 CNU 与 VNU 的严重不平衡的问题, 以第 2.1 节中的 7/8 码率 LDPC 码为例, 首先, 将校验矩阵 \mathbf{H} 进行如下所示的拆分。

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,8} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,8} \end{bmatrix}, \quad \mathbf{H}_1 = \begin{bmatrix} \mathbf{A}_{1,9} & \mathbf{A}_{1,10} & \cdots & \mathbf{A}_{1,16} \\ \mathbf{A}_{2,9} & \mathbf{A}_{2,10} & \cdots & \mathbf{A}_{2,16} \end{bmatrix}$$

\mathbf{H} 矩阵被左右拆分为两个新的矩阵 \mathbf{H}_0 和 \mathbf{H}_1 , 它们的行重均为 \mathbf{H} 矩阵的一半 $32/2=16$, 列重不变仍为 4。相应地, 与第 2.2 节所描述的译码迭代流程相比, 拆分后, 需要改变迭代的水平运算过程, 如下所示: 拆分后迭代的水平/行过程(Check Node Processing, CNU), 分成 3 步:

(a)对 \mathbf{H}_0 进行 CNU 运算, 计算出用于 \mathbf{H}_1 的 CNU 运算

的相关信息 A_{mH_0} 和 S_{mH_0} 。对 H_0 所有 m, n ，计算

$$A_{mH_0} = \sum_{n \in N(m)} \Psi(L(q_{mn})) \quad (8)$$

$$S_{mH_0} = \prod_{n \in N(m)} \text{sgn}(L(q_{mn})) \quad (9)$$

(b)对 H_1 进行 CNU 运算,一方面计算出原 H 矩阵 H_1 部分的 CNU 信息,另一方面计算出用于 H_0 的 CNU 运算的相关信息 A_{mH_1} 和 S_{mH_1} , 对 H_1 所有 m, n , 计算

$$A_{mH_1} = \sum_{n \in N(m)} \Psi(L(q_{mn})) \quad (10)$$

$$A_{mj} = \sum_{\substack{n \in N(m) \\ n \neq j}} \Psi(L(q_{mn})) + A_{mH_0} \quad (11)$$

$$S_{mH_1} = \prod_{n \in N(m)} \text{sgn}(L(q_{mn})) \quad (12)$$

$$S_{mj} = \prod_{\substack{n \in N(m) \\ n \neq j}} \text{sgn}(L(q_{mn})) \times S_{mH_0} \quad (13)$$

$$R_{mj} = s_{mj} \Psi(A_{mj}) \quad (14)$$

(c)对 H_0 进行 CNU 运算,计算出原 H 矩阵的 H_0 部分的 CNU 信息。

$$A_{mj} = \sum_{\substack{n \in N(m) \\ n \neq j}} \Psi(L(q_{mn})) + A_{mH_1} \quad (15)$$

$$S_{mj} = \prod_{\substack{n \in N(m) \\ n \neq j}} \text{sgn}(L(q_{mn})) \times S_{mH_1} \quad (16)$$

$$R_{mj} = s_{mj} \Psi(A_{mj}) \quad (17)$$

其中

$$\Psi(x) = \ln(\tanh(|x/2|)) = \ln((1 - e^{-|x|}) / (1 + e^{-|x|}))$$

对照第 2.2 节说明,仍然采用部分并行译码结构, CNU 的个数和结构将根据上文中的拆分后的新矩阵重新设计。

如图 1 所示, M 对应迭代变量存储器,下标中的数字分别与 H 矩阵的各子矩阵的编号对应,字母 a, b 分别与组成各子矩阵的两个循环行列式对应。 N_1 和 N_2 用于存储拆分矩阵后 CNU 计算中的用于求和的中间变量。

在图 2 中,各 VNU 的结构不变,总的个数减少至 8 个,对应拆分后的 H_0 与 H_1 矩阵的子矩阵列数;各 CNU 根据式(8)-式(17)调整 CNU 的结构,加入中间变量的输入和输出部分,即将各 CNU 的各部分减半,但总的个数不变,仍为两个。CNU 的具体结构,如图 2 所示。

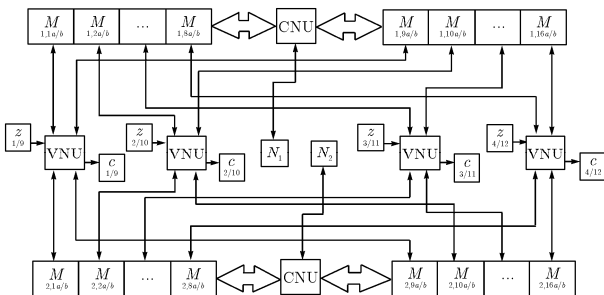


图 1 译码器整体结构图

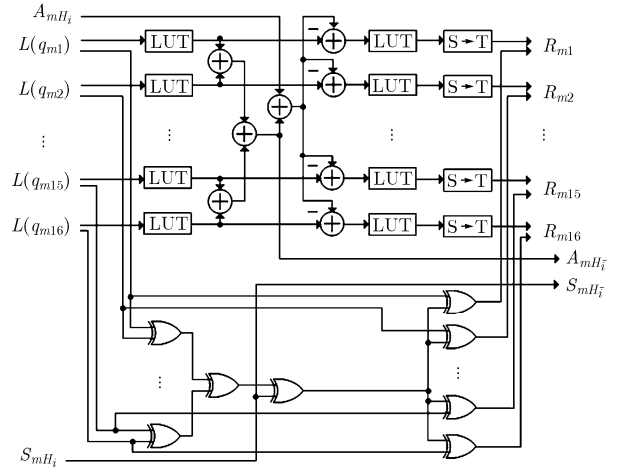


图 2 拆分矩阵后做出相应改变的 CNU 结构

在拆分后,CNU 与 VNU 可以在同时间内对不同的 H_0 或 H_1 进行运算,完成更新后,交换运算对象。具体的运算顺序如图 3。

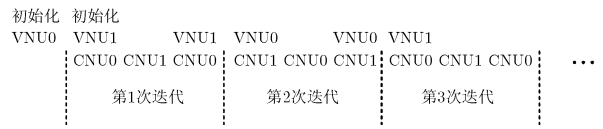


图 3 拆分矩阵后做出相应改变的迭代顺序图

采用了上述的改进方案,一次译码迭代的周期由 $511 \times 2 = 1022$ 个时钟变为了 $511 \times 3 = 1533$ 个时钟,译码速率 $= (\text{clk} \times 8176) / (3 \times 511 \times n) = \text{clk} \times 8/n \times 2/3$ (bps),为第 2.2 节的 $2/3$ 。

在复杂度方面,译码器需要的硬件资源如表 2 所示。

表 2 改进的部分并行译码结构所需的主要硬件资源

17(16+1)输入 CNU	4 输入 VNU	迭代变量存储 器	查找表 (LUT)
2 个	8 个	64+2 个	64 个

将表 1 与表 2 对比,发现改进后的译码器运算结构减小为原来的一半,存储器规模略有增加。

改进后,CNU 模块的硬件时间利用率由 50% 上升至 100%,VNU 模块则由 50% 上升至 67%。这两个模块的复杂度差别也得到了降低,译码器迭代的时钟性能高于第 2.2 节。时钟频率的提高,可以从另一个侧面解决码率降低为原来的 $2/3$ 的问题。

4 译码器实现结果及分析

采用 Verilog HDL 语言,分别以传统的部分并行结构和拆分 H 矩阵后进行的部分并行结构,对本文中的 CCSDS 推荐的 $7/8$ 码率 LDPC 码进行描述后,使用 Altera 公司 Stratix

系列的 EP1S801508C7 FPGA 进行了综合, 综合工具为 Altera 公司提供的 QuartusII5.1, 表 3 对比了两种方案的综合结果。

表 3 两种方案的综合结构比较

译码器采用结构	逻辑资源/LEs	存储器资源/bit
常用的部分并行结构	13,728 (17%)	344,064(4%)
拆分后改进的部分并行结构	8,093 (10%)	327,680(4%)

表 3 中, 第 2, 3 栏中的百分数为实际耗用资源在用于综合的 FPGA 中所占资源的比例。表中译码器所耗用的逻辑资源, 都是在取量化比特数为 7bit 时的所得。

在拆分矩阵后, 根据新矩阵而设计的译码器结构, 在耗用的逻辑资源上比元译码器结构减少 41%, 并且在存储器资源的占用上也略有降低, 这是由于, 虽然对应拆分后的矩阵, 译码器中需要加入两个用于存储中间变量的存储器 N_1 和 N_2 , 但迭代运算中 LUT 的减半使得存储器资源得到了实质性的降低。

本文中的改进结构是在拆分矩阵后, 与传统的译码器结构相比, 虽然降低了耗用资源, 但在同样的时钟频率下, 也将码速率降为原来的 $2/3$ 。从另一角度看, 若采用与改进结构相同的资源, 按传统的部分并行结构构造译码器, 硬件资源规模将仅可以支持 1 个 CNU 模块和 8 个 VNU 模块, 那么译码速率 $= (\text{clk} \times 8176) / (4 \times 511 \times n) = \text{clk} \times 8/n \times 1/2(\text{bps})$, 显然低于本文中提供的译码器结构的译码速率, 因此, 在拆分矩阵的基础上改进的译码器结构, 是一种在硬件资源和译码速率之间的优良的折衷。

实际中可以将 H 矩阵拆分任意多的部分, 并相应地调整原式(1)-式(6)实行部分并行译码, 如图 3, 若 H 矩阵被拆分为 m 个部分, 则译码速率将降低为图 1 所示方案的译码速率的 $2/(m+1)$ 。但当拆分过多时, 虽然可以在一定程度上减少耗用资源的规模, 但译码速率也将大大降低, 因此本文仅推荐将 H 矩阵拆分为 2 或 3 个部分。

5 结束语

本文提出了一种基于拆分校验矩阵的 LDPC 码译码器结构, 并对该译码器进行了 FPGA 仿真、综合实验, 验证了其

功能。以 CCSDS 建议的 7/8 码率 LDPC 码为例, 该结构尤其适用于高码率的准循环 LDPC 码译码器的硬件实现, 降低耗用逻辑资源的规模达 41%。在平衡译码速率与硬件资源的优化方法中, 本文无疑推荐了一种性能较高的译码器结构, 但如何能够在维持硬件资源消耗水平的情况下, 进一步提高译码速率, 或在维持原有译码速率的情况下, 降低硬件资源消耗水平, 仍有待进一步研究。

参考文献

- [1] MacKay D J C and Neal R M. Near shannon limit performance of low density parity check codes. *Electro. Lett.*, 1996, 32(18): 1645-1646.
- [2] Gallager R. Low density parity check codes. *IRE Trans. on Inform. Theory*, 1962, IT-8(1): 21-28.
- [3] Consultative Committee for Space Data Systems. Low density parity check codes for use in near-earth and deep space applications. CCSDS 131.1-O-1, Orange Book, <http://public.ccsds.org/publications/archive/131x1o1s.pdf>, 2006. August.
- [4] Zhang T. Efficient VLSI architectures for error-correction coding, [Ph.D. dissertation], University of Minnesota, 2002.
- [5] Zhong H, Zhang T, and Haratsch E F. High-rate quasi-cyclic LDPC codes for magnetic recording channel with low error floor, Proc. IEEE ISCAS. Kos Island, Greece, 2006.5: 3546-3549.
- [6] Tanner R M, Sridhara D, Sridharan A, Fuja T E, and Costello D J. LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. on Info. Theory*, 2004, 50(12): 2966-2984.
- [7] Fossorier M P C. quasi-cyclic low density parity check codes from circulant permutation matrices. *IEEE Trans. on Info. Theory*, 2004, 50(8): 1788-1793.
- [8] Dai Y M and Yan Z Y. Optimal overlapped message passing decoding for quasi-cyclic low-density parity-check codes. Proc. Globecom, San Jose, California, USA, 2005, Vol. 4: 2395-2399.

张靖琳: 女, 1983 年生, 硕士, 研究方向为信道编码。

刘荣科: 男, 1973 年生, 博士, 副教授, 研究方向为扩频通信、信源、信道编码等。

赵 岭: 男, 1980 年生, 博士, 研究方向为信道编码。