

高吞吐量 QC-LDPC 码分层译码设计

徐 斌, 贺玉成

(华侨大学 厦门市移动多媒体通信重点实验室, 福建 厦门 361021)

摘 要: 针对低密度奇偶校验译码器吞吐量较低、存储资源消耗较多的问题, 提出一种 QC-LDPC 码分层译码算法。利用接收信道模块初始化似然比信息, 并结合存储校验信息和后验信息给出基于分层最小和的节点自更新译码算法, 根据后验信息符号位对译码器进行判决。仿真结果表明, 改进译码器资源消耗相对于传统译码器减少 20%, 当迭代次数为 10 时, 吞吐量可达 516.8 Mb/s。

关键词: 低存储量; 并行分层; 高吞吐量; 校验节点自更新算法; 译码器

中文引用格式: 徐斌, 贺玉成. 高吞吐量 QC-LDPC 码分层译码设计[J]. 计算机工程, 2019, 45(7): 121-125, 133.

英文引用格式: XU Bin, HE Yucheng. High throughput layered decoding design of QC-LDPC codes[J]. Computer Engineering, 2019, 45(7): 121-125, 133.

High Throughput Layered Decoding Design of QC-LDPC Codes

XU Bin, HE Yucheng

(Xiamen Key Laboratory of Mobile Multimedia Communications, Huaqiao University, Xiamen, Fujian 361021, China)

[Abstract] To address the low throughput and high storage resource consumption of the low density parity-check decoder, this paper proposes a layered decoding algorithm of QC-LDPC codes. The algorithm uses the receiving channel module to initialize the likelihood ratio information. Then the storage check information and posteriori information are combined to generate a node self-updating decoding algorithm based on the layered minimum sum. The decoder is judged based on the sign bit of posterior information. Simulation results show that the storage resources of the improved decoder are saved by 20% compared with the traditional decoder. When the number of iterations is 10, the throughput can reach 516.8 Mb/s.

[Key words] low storage; parallel layered; high throughput; check nodes automatic update algorithm; decoder

DOI: 10.19678/j.issn.1000-3428.0050741

0 概述

低密度奇偶校验码 (Low Density Parity-check Codes, LDPC) 是线性分组码, 其译码简单、性能接近容量限^[1]。文献[2-4]通过分层译码算法构造 LDPC 译码器, 使其无需存储变量节点信息, 降低了存储资源耗费, 且译码收敛速度比最小和 (Min-Sum, MS) 算法快。文献[5-6]提出一种双修正型最小和积译码算法, 该算法译码复杂度较低, 但吞吐量有一定的损失。文献[7-8]提出的交替译码框架, 能在同一时刻同时更新一帧校验消息和一帧变量消息, 两帧交替更新, 提高了吞吐量, 但需消耗较多的存储资源。本文在分层 MS 译码算法的基础上, 提出校验节点自更新译码算法, 并采用一种新的 QC-LDPC 码分层译码器结构, 以降低存储资源消耗和提高吞吐量。

1 分层译码算法

QC-LDPC 码具有准循环特性, 其校验矩阵可表示为分块矩阵^[9]。

$$\mathbf{H} = \begin{bmatrix} \mathbf{Q}(p_{11}) & \mathbf{Q}(p_{12}) & \cdots & \mathbf{Q}(p_{1t}) \\ \mathbf{Q}(p_{21}) & \mathbf{Q}(p_{22}) & \cdots & \mathbf{Q}(p_{2t}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}(p_{s1}) & \mathbf{Q}(p_{s2}) & \cdots & \mathbf{Q}(p_{st}) \end{bmatrix}_{sB \times tB} \quad (1)$$

其中, B 表示分块矩阵 $\mathbf{Q}(p_{ij})$ 的阶数, $0 \leq p_{ij} \leq B$, $i = 1, 2, \dots, s$, $j = 1, 2, \dots, t$ 。当 $p_{ij} = 0$ 时, $\mathbf{Q}(p_{ij})$ 表示一个 B 阶全零矩阵; 当 $1 \leq p_{ij} \leq B$ 时, $\mathbf{Q}(p_{ij})$ 表示 B 阶单位矩阵 \mathbf{I}_B 循环右移 $p_{i,j}$ 位后得到的矩阵, 特别地, $\mathbf{Q}(B) = \mathbf{I}_B$ 。因此, 每个子块列重最大为 1。 \mathbf{H} 是一个 $sB \times tB$ 矩阵, 列重最大为 s , 码长为 $n = tB$, 设计码率为 $R = (t - s)/t$ 。

根据每个子块的列重量分布, 可校验矩阵 \mathbf{H} 按

基金项目: 福建省自然科学基金(2014J05076, 2014J01243, 2015J01262)。

作者简介: 徐 斌 (1993—), 男, 硕士研究生, 主研方向为信道编码; 贺玉成, 教授、博士。

收稿日期: 2018-03-12 修回日期: 2018-04-24 E-mail: 1215170615@qq.com

上述分块形式划分为 s 层, 每层对应 B 个校验节点 $c_{i,h}$ ($i=1,2,\dots,s, h=1,2,\dots,B$)。令 $\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$ 表示一个码字, $\mathbf{x}_j=(x_{j,1}, x_{j,2}, \dots, x_{j,B})$ 表示由 B 个变量节点 x_{jl} ($j=1,2,\dots,t, l=1,2,\dots,B$) 构成的子码。根据分层方式, 可推得各个节点的相邻校验节点集为:

$$c_{i,h} = \{x_{j,l} : j=1,2,\dots,t; p_{ij} > 0; \\ l = (p_{ij} + h - 1) \bmod B + 1\} \quad (2)$$

第 i 层对应 B 个校验方程如下:

$$\mathbf{Q}(p_{i1})\mathbf{x}_1^T + \mathbf{Q}(p_{i2})\mathbf{x}_2^T + \dots + \mathbf{Q}(p_{it})\mathbf{x}_t^T = \mathbf{0}_{B \times 1} \quad (3)$$

其中, 码字 \mathbf{x} 的每个变量节点每层只出现在一个校验方程中。

分层最小和算法^[10]描述如下: 基于第一层子方程组, 更新校验消息, 然后把校验消息代入到变量节点中更新变量消息。当完成第一层的变量消息更新后, 代入下一个分层校验消息和更新变量消息。每一层消息更新过程使用上一层发送的变量信息和初始校验信息。当所有分层都完成消息更新后, 一轮迭代译码更新完成。

分层最小和算法的特点是利用后验信息和校验信息来表示变量信息, 使变量信息无需储存, 降低了资源的消耗。但是分层最小和算法仍需要存储大量的校验信息以及后验信息, 且后验信息值的范围随着迭代次数的增加会慢慢变大, 需要花费更多比特去储存才能得到合理的性能。为减少存储资源的消耗, 提出校验节点自更新方法, 该方法基于分层最小和算法, 将后验信息融合到校验信息中, 避免了后验信息的储存, 从而减少译码器资源的开销。

在第 $k \geq 1$ 次迭代中, 令 $q_{i,j}^k$ 表示第 i 层的变量节点 \mathbf{x}_j 接收的更新校验信息, $\gamma_{i,j}^k$ 是变量节点 \mathbf{x}_j 从第 i 层发往第 $i+1$ 层的更新变量信息, γ^k 是第 k 次迭代更新的后验概率信息, i 指层序号, n_i 指与第 i 层校验节点相连的变量节点, $n_{i'} \in N(i) \setminus n_i$ 是指除 n_i 外, 与第 i 层校验节点相邻的变量节点的集合, λ_i 指信道信息初始值, α 表示校正因子, 一般取 $\alpha = 0.75$, 便于硬件实现。校验节点自更新译码算法的具体步骤如下:

步骤 1 LDPC 码初始化。初始化结果为:

$$\gamma^0 = \lambda_i \quad (4)$$

$$r_i^0 = 0 \quad (5)$$

步骤 2 LDPC 码迭代过程 (第 k 次迭代, 第 i 层更新) 包括分层更新和译码判决, 具体分析如下:

1) 分层更新计算公式如下:

$$q_{i,j}^k = \gamma^0 + \sum_{c_{i'} \in m(v_j), i' > i} r_{i'}^{k-1} + \sum_{c_{i'} \in m(v_j), i' < i} r_{i'}^k \quad (6)$$

$$r_{i,j}^k = \alpha \prod_{v_j \in N(c_i) \setminus v_j} \text{sgn}(q_{i,j}^k) \times \min_{v_j \in N(c_i) \setminus v_j} (|q_{i,j}^k|) \quad (7)$$

$$\gamma^k = \gamma^0 + \sum_{c_i \in m(v_j)} r_i^k \quad (8)$$

2) 译码判决。设译码码字 $\mathbf{c} = \{c_1, c_2, \dots, c_t\}$, 若 $\gamma^k < 0$, 则 $c_m = 1$; 否则 $c_m = 0$ 。

步骤 3 校验结果。若译码序列满足 $\mathbf{c} \times \mathbf{H}^T = \mathbf{0}$ 或者译码次数已经达到最大迭代次数, 则停止译码; 否则重新执行步骤 2。

2 高吞吐量译码架构设计

译码器的最大吞吐量表达式为:

$$T = \frac{f \times N \times R}{C \times I_{\max}} \quad (9)$$

其中, f 是工作频率, 单位是 Hz, N 是码长, 单位是 bit, R 是指码率, C 是译码器每次迭代所花费的时钟周期数, 单位是 s, I_{\max} 是最大迭代次数, T 是吞吐量, 单位是 $\text{bit} \cdot \text{s}^{-1} \cdot \text{Hz}^{-1}$ 。由于长码相对于短码性能更加优越, 因此在设计过程中, 一般采用长码。为提高吞吐量, 采用并行更新的方式更新数据, 但会增加输入输出的延迟, 导致译码过程不同步。在这段时间内, 译码模块处于等待状态无法更新, 使得吞吐量变低。式 (9) 中分母实际应该为 $(C \times I_{\max} + D)$, 因为 D 为额外时钟延迟, 所以达不到最大吞吐量。本文提出一种新的译码器结构, 通过改进输入、译码、输出 3 个模块, 可最大化减少 D 值, 实现 3 个模块的连续工作, 使译码器可达到最高吞吐量。

根据文献[11], 针对 1/2 码率的 (2 304, 1 152) QC-LDPC 码, 如图 1 所示, 2 304 为码长, 1 152 为信息位长度, 每个子矩阵大小为 96×96 , 每个数字代表单位矩阵偏移量, 空白的地方表示零矩阵。实现连续译码的过程是: 在输入过程中, 1 个时钟周期输入 2 位信息, 传输 1 帧信息需要 1 152 个时钟周期, 传输 2 帧信息需要 2 304 个周期。在输出过程中, 1 帧信息总长度为 1 152 位, 1 个时钟周期输出 1 位信息, 输出 2 帧信息则需要 2 304 个时钟周期。在译码过程中, 2 帧信息并行更新, 每次迭代更新要 96 个时钟周期, 将 I_{\max} 和每次迭代更新所消耗的时钟数相乘, 结果为消耗的时钟 m 。当 I_{\max} 取校验矩阵行数的值时, 时钟 m 值就是信息位长度 1 152, 即译码消耗了 1 152 个时钟周期。输入和输出采用的是双倍频率时钟周期, 其耗费 2 304 个双倍频率时钟周期, 译码采用单倍频率时钟周期, 其耗费 1 152 个单倍频率时钟周期, 2 304 个双倍频率时钟周期在时间上等同于 1 152 个单倍频率时钟周期, 即输入、输出、译码这 3 个过程所花费一样的时钟周期数, 因此可实现连续译码。

译码器设计采用校验节点自更新算法,提出高吞吐率的 QC-LDPC 码译码器框架,该框架支持部分并行分层译码,如图 5 所示。本文方法由接收信道信息、信息存储器、校验信息处理、硬判决以及迭代译码控制等主要部分组成。迭代译码控制模块是译码器的核心,各个子模块之间通过不同的时序进行控制,每个存储器的读写地址、使能信号由地址生成器提供,且初始化信息、外信息也是由不同的时序控制。

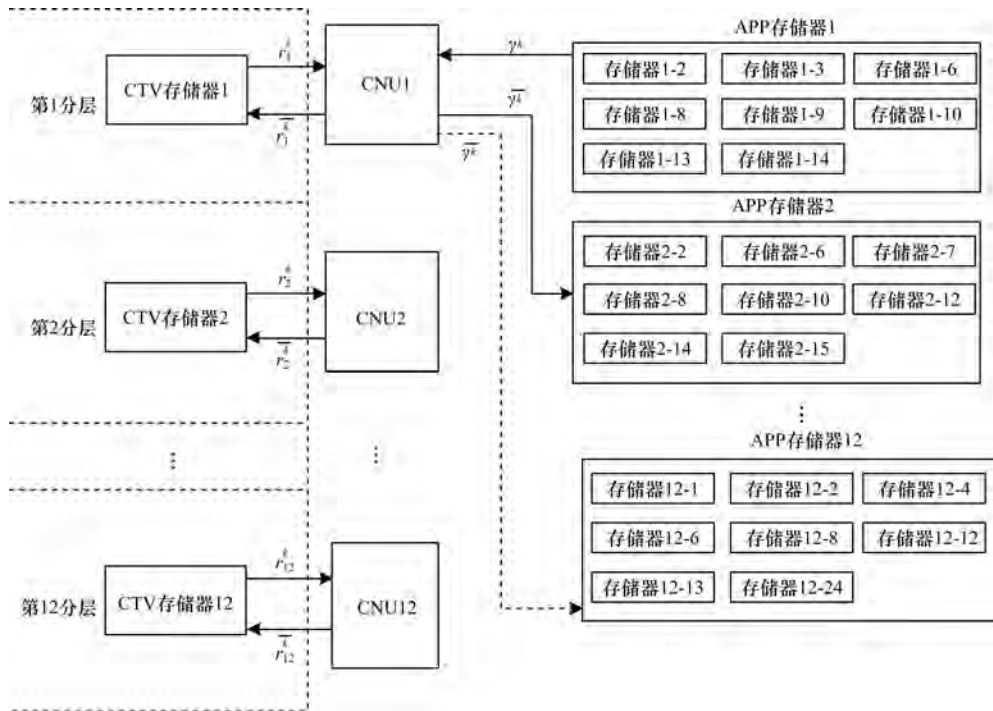


图5 高吞吐量的 QC-LDPC 码译码器框架

3.2 接收信道信息模块

接收信道信息模块是对信道似然比信息进行初始化,初始化为接收到量化过后的译码信息。在接收信道信息模块中,对信道似然比信息存储器的写入顺序操作与子矩阵偏移量大小有关系,令偏移量为16,似然比信息存储器对应的初始化操作如图6所示,0地址存入的值是偏移量对应第16个已更新校验信息的数据,1地址存入的值是对应第17个已更新校验信息的数据,2地址存入的值对应第18个已更新校验信息的数据。地址按顺序存入相应的数据,不一样的偏移量对应不一样的接收信道信息模块,共96个不一样的偏移量,对应96个接收信道信息模块。



图6 接收信道信息模块初始化过程

3.3 信息存储模块

信息存储模块包括校验信息和后验信息这2个存储模块,校验矩阵 H 被分成12个分层,且每个分层有一个专门的校验节点单元(Check Node Unit, CNU),后验信息存储模块存储后验概率信息,校验信息存储模块存储校验节点信息。每层的非零子矩阵对应一

个后验信息存储模块和一个校验信息存储模块。在分层译码过程中,每一个后验信息存储模块向CNU输出一个后验信息,并从其他分层CNU中获取一个新的后验信息。同时,每个校验信息存储模块向CNU输出校验信息,并且从同一层CNU得到新的校验信息。本文采用单端口同时进行读写操作。因为校验矩阵 H 中有96个非零子矩阵,所以后验信息存储模块和校验信息存储模块包含了96个单端口存储单元。

3.4 校验信息处理单元模块

校验信息处理单元模块是校验节点自更新算法中的核心单元,对于校验信息的更新来说,每一层采用一个CNU去执行一系列的功能,主要包括加法、比较等,CNU成为最长的延迟路径。在新的译码算法中,每个CNU需要对比8个信息,找出其中的最小值、次小值及各自的地址。因此,对比装置成了CNU中最关键的元素。2输入的比较器包含一个加法器和一个复用器,3输入的比较器包含3个加法器、5个复用器和一些基础逻辑门。一个8输入的比较器是由多个2输入和3输入比较器构成。为减少关键路径延迟和提高工作频率,CNU被分割成3级流水寄存器。CNU的3级流水线结构如图7所示。由式(6)可知,在第1级中,第一个加法器组将初始信道信息、前一次迭代得到的校验信息和本次迭代得到的校验信息相加得到VTC信息 $q_1 \sim q_8$ 。由式(7)可知,在第2级中,将第1级得到的VTC信息进行符号分离、比较选取最小值等操作,重新合并端口的最小值与符号,得到新的校验信息 $r_1 \sim r_8$ 。由式(8)可知,在第3级中,加法器将初始信道信息与第2级得到的校验信息 r_i 相加,得到新的后验概率信息 APP_{new} 。

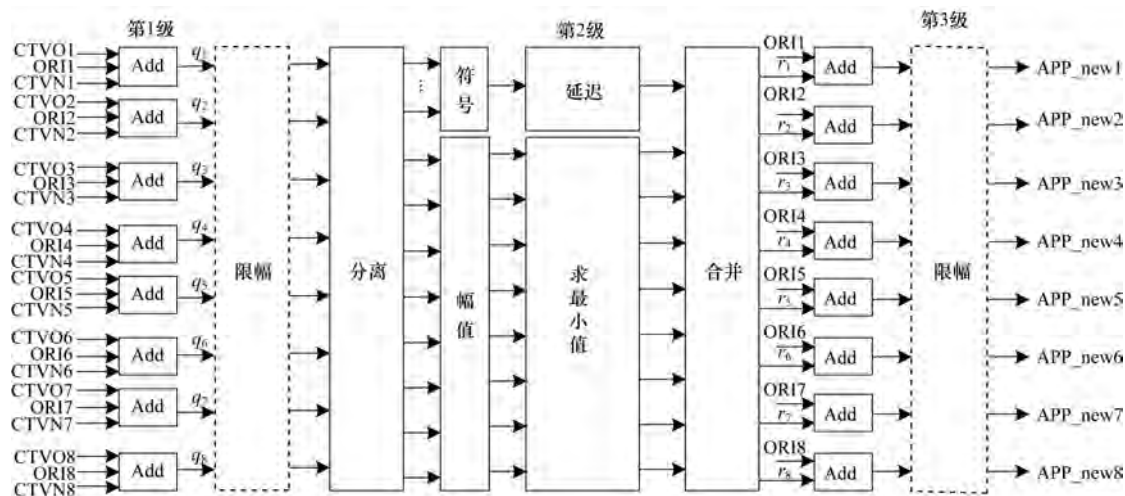


图 7 CNU 模块 3 级流水线结构框图

3.5 硬判决单元

LDPC 译码器输出比特位由后验信息的符号位决定。在传统分层译码中^[13], 变量节点一般更新到最底层才开始进行硬判决。而部分并行分层译码却使用一种不同的方法来进行更新, 会使每一个分层在每次迭代过程中产生一个后验概率信息, 并且不同的分层之间同时工作。以修正偏移量后的校验矩阵的第 1 列为例, 第 4 层 CNU 从第 13 行开始更新, 第 9 层 CNU 从第 1 行开始更新, 第 12 层 CNU 从第 81 行开始更新, 根据修正偏移量后的校验矩阵, 分别对应第 74 行、第 13 行、第 28 行。第 1 列后验概率信息传递的路径是从第 4 层到第 12 层、第 12 层到第 9 层、第 9 层到第 4 层。在每次迭代结束之后, 第 13 列最后的后验概率信息储存在第 12 层, 第 28 列最后的后验概率信息储存在第 4 层, 第 74 列最后的后验信息储存在第 9 层。因此, 第 1 列的硬判决比特储存在 3 个不同分层的存储器中。

4 译码器性能

在译码器硬件实现中, 文献[14-16]以及本文提出的译码器均采用 altera 公司 StratixIV 系列的 EP4SGX530, 由 Quartus II 10.0 综合后, 本文提出译码器的最高工作频率可达 308.2 MHz。根据式(9)译码器吞吐量的计算, 当最大迭代次数为 10 时, 译码器的吞吐量为 516.8 Mb/s。本文提出的译码器和其他译码器之间性能比较如表 1 所示。

表 1 译码器性能比较

性能指标	文献[14] 算法	文献[15] 算法	文献[16] 算法	本文算法
码长	12 288	2 304	8 176	8 192
码率	1/2	1/2	7/8	1/2
工作频率/MHz	152.5	167.2	201.7	308.2
吞吐量/(Mb · s ⁻¹)	211.6	268.3	288.9	516.8
迭代次数	10	8	8	10
资源消耗(LUTs)	49 423	18 149	26 259	8 125

由表 1 可以看出, 与文献[14-16]译码器相比, 本文译码器吞吐量较大, 且消耗的存储资源较小。因此, 本文译码器同时具有高效、复杂度低、吞吐量大、收敛速度快等性能, 有效解决了吞吐量与资源消耗的矛盾。

5 结束语

基于分层 MS 译码算法, 本文提出一种高吞吐量、低存储量译码器, 实现了(2 304, 1 152) LDPC 码。通过修正偏移量提高译码器的吞吐量。同时, 将后验信息和变量信息融合到校验信息中, 从而减少译码器的存储资源消耗。仿真结果显示, 该译码器吞吐量较大, 且收敛速度较快。

参考文献

- [1] GALLAGER R G. Low-density parity-check codes[J]. IRE Transactions on Information Theory, 1962, 8(1): 21-28.
- [2] 郭琨, 黑勇, 周玉梅, 等. 一种应用于不可分层 LDPC 码的并行分层译码算法[J]. 电子与信息学报, 2010, 32(8): 1965-1960.
- [3] KOU Y, LIN S, FOSSORIER M P C. Low-density parity-check codes based on finite geometries: a rediscovery and new results[J]. IEEE Transactions on Information Theory, 2001, 47(7): 2711-2736.
- [4] MANSOUR M M. A turbo-decoding message-passing algorithm for sparse parity-check matrix codes[J]. IEEE Transactions on Signal Processing, 2006, 54(11): 4376-4392.
- [5] DARABIHA A, CARNSON A C, KSCHISCHANG F R. Block-interleaved LDPC decoders with reduced interconnect complexity[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2008, 55(1): 74-78.
- [6] WANG Fei, ZHANG Peng, LIU Changyin. FPGA implementation of a novel multi-rate QC-LDPC encoder for DTMB standard[J]. Applied Mechanics and Materials, 2013, 397-400: 2024-2027.
- [7] FENG Wei, WANG Yanmin, LIN Dengsheng, et al. When mmWave communications meet network densification: a scalable interference coordination perspective[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(7): 1459-1471.

(下转第 133 页)

(上接第 125 页)

- [8] QIU Lipeng, CHEN Xiaopeng, ZHOU Lin, et al. FPGA implementation of reed-solomon codes based on visible light wireless communication system [C] // Proceedings of International Symposium on Advances in Electrical, Electronics and Computer Engineering. Paris, France: Atlantis Press, 2016; 234-237.
- [9] SUN Yang, CAVALLARO J R. VISL architecture for layered decoding of QC-LDPC codes with high circulant weight [J]. IEEE Transactions on Very Large Scale Integration Systems, 2013, 21 (10): 1960-1964.
- [10] HAILES P, XU Lei, MAUNDER R G, et al. A survey of FPGA-based LDPC decoders [J]. IEEE Communications Survey and Tutorials, 2016, 18 (2): 1098-1122.
- [11] 云飞龙, 杜锋, 朱宏鹏, 等. 一种高吞吐量 QC-LDPC 码译码器的 FPGA 实现 [C] // 第七届中国卫星导航学术年会论文集. 北京: [出版者不详], 2016; 16-20.
- [12] 云飞龙, 朱宏鹏, 吕晶, 等. 一种基于奇偶并行译码架构的高吞吐量译码器设计 [J]. 通信技术, 2016, 49 (3): 264-269.
- [13] CHUNG S Y, RICHARDSON T J. Analysis of sum product decoding of low-density parity-check codes using a Gaussian approximation [J]. IEEE Transactions on Information Theory, 2001, 47 (2): 657-670.
- [14] HARIRI A A A, MONTEIRO F, SIÉLER L, et al. Configurable and high-throughput architectures for quasicyclic low-density parity-check codes [C] // Proceedings of International Conference on Electronics, Circuits and Systems. Washington D. C., USA: IEEE Press, 2014; 790-793.
- [15] MAHDI A, PALIOURAS V. A low complexity-high throughput QC-LDPC encoder [J]. IEEE Transactions on Signal Processing, 2014, 62 (10): 2696-2708.
- [16] DING Hong, YANG Shuai, LUO Wu, et al. Design and implementation for high speed LDPC decoder with layered decoding [C] // Proceedings of International Conference on Communications and Mobile Computing. Washington D. C., USA: IEEE Computer Society, 2009: 156-160.

编辑 赵 辉