

同步部分并行结构的准循环 LDPC 码译码器

许恩杨 姜明 赵春明

(东南大学移动通信国家重点实验室 南京 210096)

摘要: 该文根据准循环 LDPC 码的结构特点, 提出了一种同步部分并行结构的译码器。在译码器中, 校验节点处理单元和变量节点处理单元同时并行工作, 使得迭代过程中新产生的软信息能够被提前使用, 加快迭代的收敛速度。同时, 采用差分演化的方法对各节点处理单元的起始位置进行优化, 进一步提高了译码器的性能。仿真结果表明, 该方案在译码性能和复杂度上都要优于现有其他方案, 适合高速译码器的实现。

关键词: 低密度奇偶校验(LDPC)码; 译码器; 同步部分并行结构

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2008)07-1630-05

A Synchro Partially Parallel Architecture for Quasi-Cyclic LDPC Codes

Xu En-yang Jiang Ming Zhao Chun-ming

(National Mobile Communication Research Lab., Southeast University, Nanjing 210096, China)

Abstract: Based on the structure of quasi-cyclic LDPC codes, a synchro partially parallel decoder is proposed in this paper. In the decoder, the check node process units and variable node process units work concurrently, where the new generated soft information is used in advance during the iteration process to accelerate the convergence speed. Furthermore, differential evolution is utilized to optimize the start positions of node process units in order to achieve better performance. Simulation results show that the proposed scheme outperforms others both in performance and complexity, and is very suitable for the implementation of high speed decoders.

Key words: Low-Density Parity-Check (LDPC) codes; Decoder; Synchro partially parallel architecture

1 引言

近年来, 低密度奇偶校验(Low-Density Parity-Check, LDPC)码^[1]作为一种可以接近信道容量上限的信道编码方式, 在编解码理论和实践中成为研究的热点。各种结构的 LDPC 码译码器被提出, 包括全并行结构、全串行结构、部分并行结构等^[2-5]。为了便于硬件实现, 实际系统中一般采用具有规则结构的码字。准循环 LDPC 码^[6]就是一类具有规则结构的 LDPC 码, 它可以达到甚至超过随机构造码字的性能, 同时编码可以通过移位寄存器实现^[7], 非常适合实际应用。

LDPC 码基于二分图的软判决译码一次迭代主要包括校验节点更新和变量节点更新两个步骤。在全并行译码器中, 所有校验节点或者变量节点同时进行更新, 译码速度非常快, 但是译码器中节点处理单元的数目和和各处理单元与存储单元之间连线的复杂度随着码长的增长快速增加, 因此只适合于中短码长和对译码速率要求极高的场合^[2]。全串行译码器中校验节点或变量节点的更新是单个依次进行的, 资源消耗比较少, 但由于其译码速率过低不适合高速应用。部

分并行译码器^[3-5]则是全并行和全串行译码器在硬件资源和译码速率之间的一个折中。在标准部分并行译码器中, 校验节点更新和变量节点更新是交替进行的, 硬件资源的使用效率较低。文献[4]提出了一种交叠部分并行方案, 相应的译码器每次迭代校验(变量)节点在所有变量(校验)节点的更新完成之前开始更新, 并通过优化每次迭代各节点处理单元起始处理位置来减少等待时间, 提高了硬件资源的使用效率和译码速率, 是现有译码方案中最好的一种。文献[5]中通过增加变量节点信息更新的频率加快迭代收敛速度, 虽然减少了迭代次数但由于校验节点更新时无法进行流水线处理, 译码速度提高不明显。

针对准循环 LDPC 码, 本文提出一种同步部分并行结构的译码器。在译码器中, 各个节点处理单元同时工作, 在迭代过程中提前使用新产生的软信息, 并通过对各节点处理单元起始位置的优化增加迭代过程中软信息提前使用量, 加快迭代译码的收敛速度, 提高译码器的性能。

本文第2节在简单介绍准循环 LDPC 码及标准部分并行译码算法的基础上给出同步部分并行译码算法, 并讨论节点处理单元起始位置的优化方法; 第3节将具体说明译码器的硬件结构; 在第4节将给出仿真结果并与其他方案进行比较和分析; 最后是结束语。

2006-12-04 收到, 2007-05-21 改回

国家自然科学基金重大项目(60496311)和国家 863 计划项目(2006AA01Z263)资助课题

2 准循环 LDPC 码的同步部分并行译码算法

2.1 准循环 LDPC 码和标准部分并行译码算法

准循环 LDPC 码是一类结构规则的 LDPC 码, 其校验矩阵由一个个小方阵组成, 一般情况下每个小方阵是置换单位阵或全零阵。在本文中用 (N, M, K, Z) 表示一个长度 $N = L \times Z$ 的准循环 LDPC 码, 信息比特长度为 K , 校验矩阵有 $M = J \times Z$ 行, 具体如下:

$$H = \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,L-1} \\ P_{1,0} & P_{1,1} & \cdots & P_{1,L-1} \\ \vdots & \vdots & \ddots & \vdots \\ P_{J-1,0} & P_{J-1,1} & \cdots & P_{J-1,L-1} \end{bmatrix}$$

其中 $P_{i,j}$ 为大小 $Z \times Z$ 的置换单位阵或全零阵。

这里先简单介绍标准部分并行算法。定义 $B(m) = \{n : H_{nm} = 1\}$ 为与校验节点 c_m 相连的变量节点集合, $A(n) = \{m : H_{nm} = 1\}$ 为与变量节点 v_n 相连的校验节点集合, 用 $B(m) \setminus n$ 表示和 c_m 相连的除 v_n 以外的变量节点集合, 用 $A(n) \setminus m$ 表示和 v_n 相连的除 c_m 以外的校验节点集合。设输入译码器的初始似然比序列为 y_n , 译码器使用 J 个校验节点处理单元和 L 个变量节点处理单元进行并行译码, 分别对应于校验矩阵的 J 个行块, L 个列块。定义 R_{nm}^k , Q_{nm}^k 为第 k 次迭代中校验节点 c_m 向变量节点 v_n , 变量节点 v_n 向校验节点 c_m 传递的似然比信息, Q_n^k 为第 k 次迭代得到的变量节点 v_n 的似然比信息。标准部分并行迭代译码算法的步骤如下:

(1) 初始化: 对所有 $n, m \in A(n)$, $Q_{nm}^0 = y_n$ 。

(2) 迭代译码

在第 k 次迭代的前 Z 个节拍, J 个校验节点处理单元同时进行信息更新。在第 $z(0 \leq z \leq Z-1)$ 个节拍, 第 $j(0 \leq j \leq J-1)$ 个校验节点处理单元更新校验节点 c_m , $m = z + jZ$ 的信息: 对所有 $n \in B(m)$,

$$R_{nm}^k = 2 \tanh^{-1} \left\{ \prod_{n' \in B(m) \setminus n} \tanh[Q_{n'm}^k / 2] \right\}$$

在第 k 次迭代的后 Z 个节拍, L 个变量节点处理单元同时进行信息更新。在第 $z(0 \leq z \leq Z-1)$ 个节拍, 第 $l(0 \leq l \leq L-1)$ 个变量节点处理单元更新变量节点 v_n , $n = z + lZ$ 的信息: $Q_n^k = y_n + \sum_{m \in A\{n\}} R_{nm}^k$, 对所有 $m \in A\{n\}$, $Q_{nm}^k = Q_n^k - R_{nm}^k$ 。

(3) 根据 Q_n^k 判决得到 d_n 。如果 d_n 满足所有校验式, 译码结束, 将 d_n 作为译码输出; 否则转到(2)继续迭代, 直到达到最大迭代次数或者某次迭代的结果满足所有校验式为止。

2.2 同步部分并行译码算法

在标准部分并行算法中, 校验节点和变量节点信息的更新是交替进行的, 一次迭代需要 $2Z$ 个节拍。在同一个节拍内, 校验节点处理单元和变量节点处理单元只有一种在工作, 硬件的使用效率不高。交叠部分并行算法则提高了硬件的使用效率, 相应的译码器中校验节点和变量节点信息的更

新交叠进行, 每次迭代校验(变量)节点在所有变量(校验)节点的更新完成之前就开始更新。在相等的迭代次数下, 其性能和标准部分并行算法完全相同, 只是需要的时间更短, 译码速度更快。但在译码的部分节拍中, 译码器中校验节点处理单元和变量节点处理单元还是只有一种在工作, 硬件使用效率没有达到 100%。为了减少等待时间, 还需要调整每次迭代节点处理单元的起始位置, 增加了控制的复杂度。

本文给出一种并行度和标准部分并行方案相同的同步部分并行译码算法, 其硬件使用效率达到了 100%, 并且相对交叠部分并行算法控制更加简单。在同步部分并行译码算法中, 每次迭代分为 Z 个节拍, 校验节点处理单元和变量节点处理单元的起始位置向量分别为 $\mathbf{S} = [s_0, s_1, \dots, s_{J-1}]$, $\mathbf{T} = [t_0, t_1, \dots, t_{L-1}]$ 。定义 $R_{nm}^{k,z}$, $Q_{nm}^{k,z}$ 为第 k 次迭代的第 z 个节拍校验节点 c_m 向变量节点 v_n , 变量节点 v_n 向校验节点 c_m 传递的似然比信息, 整个算法的步骤如下:

(1) 初始化: 对所有 $m, n \in B(m)$, $R_{nm}^{0,Z-1} = 0$; 对所有 $n, m \in A(n)$, $Q_{nm}^{0,Z-1} = y_n$ 。

(2) 迭代译码: 在第 k 次迭代的各个节拍, J 个校验节点处理单元和 L 个变量节点处理单元同时工作。具体来说, 在第 $z(0 \leq z \leq Z-1)$ 个节拍:

第 $j(0 \leq j \leq J-1)$ 个校验节点处理单元更新校验节点 c_m , $m = jZ + (s_j + z) \bmod Z$ 的信息: 对所有 $n \in B(m)$, $R_{nm}^{k,z} = \beta \prod_{n' \in B(m) \setminus n} \operatorname{sgn}(Q_{n'm}^{k,z-1}) \min_{n' \in B(m) \setminus n} |Q_{n'm}^{k,z-1}|$, 其中 $Q_{n'm}^{k-1} = Q_{n'm}^{k-1,Z-1}$, β 为修正因子, 取值和具体码字有关, 一般取最优值附近便于硬件实现的数值;

第 $l(0 \leq l \leq L-1)$ 个变量节点处理单元更新变量节点 v_n , $n = lZ + (t_l + z) \bmod Z$ 的信息: $Q_n^k = y_n + \sum_{m \in A\{n\}} R_{nm}^{k,z-1}$, 对所有 $m \in A\{n\}$, $Q_{nm}^{k,z} = Q_n^k - R_{nm}^{k,z-1}$, 其中 $R_{nm}^{k-1} = R_{nm}^{k-1,Z-1}$ 。

(3) 根据 Q_n^k 判决得到 d_n : 如果 d_n 满足所有校验式, 译码结束, 将 d_n 作为译码输出; 否则转到(2)继续迭代, 直到达到最大迭代次数或者某次迭代的结果满足所有校验式为止。

图 1 给出了标准部分并行、交叠部分并行、同步部分并行 3 种方案的时序图。从图中可以看出, 3 种方案完成 k 次迭代需要的节拍数分别为 $2kZ$, $2kZ/\eta$, kZ , 即在相等的时间内三者可以分别进行 k , ηk , $2k$ 次迭代, 其中 η 是交叠部分并行方案相对标准部分并行方案的速度增益因子, $1 < \eta < 2$ 。

2.3 节点处理单元起始位置的优化

同步部分并行译码算法同一次迭代输入各个节点处理单元的似然比信息中, 有的是上一次迭代产生的, 有的是本次迭代前几个节拍得到的, 即本次迭代中产生的软信息被提前使用。软信息的提前使用可以加快迭代的收敛速度, 提高译码的性能。由于每个节点处理单元与校验矩阵的 Z 行或 Z 列对应, 各个节点处理单元起始位置不同, 迭代过程中软信

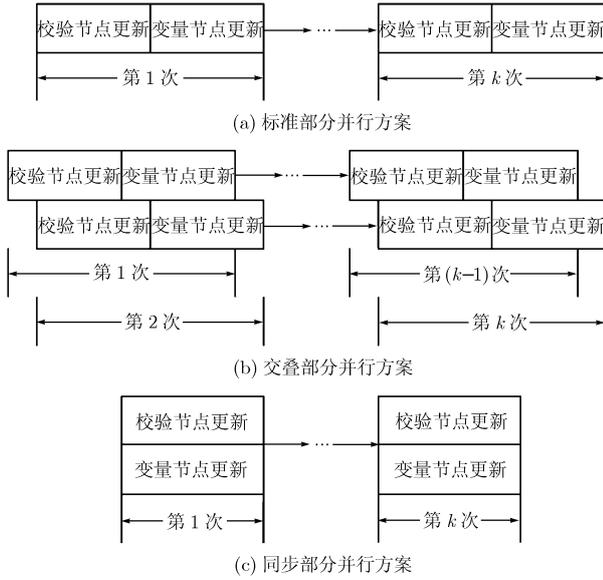


图1 各种部分并行译码方案时序比较

息的提前使用情况就不一样，最终的性能也有差别，因此需要对起始位置向量 \mathbf{S}, \mathbf{T} 进行优化。

仿真过程中发现迭代中软信息提前使用量越大，迭代收敛得越快，相同迭代次数下性能越好。因此在优化起始位置向量时，把目标函数 F 定义为一次迭代过程输入各节点处理单元的似然比信息中被本次迭代更新过的数目。当起始位置向量 \mathbf{S}, \mathbf{T} 确定后，可以采用类似同步部分并行译码的方法来计算 F ，具体步骤如下：

(1)初始化

对所有 $m, n \in B(m)$, $R_{mn} = R'_{mn} = 0$ ；对所有 $n, m \in A(n)$, $Q_{nm} = Q'_{nm} = 0$ ； $F = 0$ 。

(2)for $z = 0$ to $Z-1$ do

对所有的 $m, n \in B(m)$, $R'_{mn} = R_{mn}$ ；对所有 $n, m \in A(n)$, $Q'_{nm} = Q_{nm}$ ；

for $j = 0$ to $J-1$ do

$m = jZ + (s_j + z) \bmod Z$ ，对所有 $n \in B(m)$, $R_{mn} = 1$ ， $F = F + \sum_{n \in B(m)} Q'_{nm}$ ；

for $l = 0$ to $L-1$ do

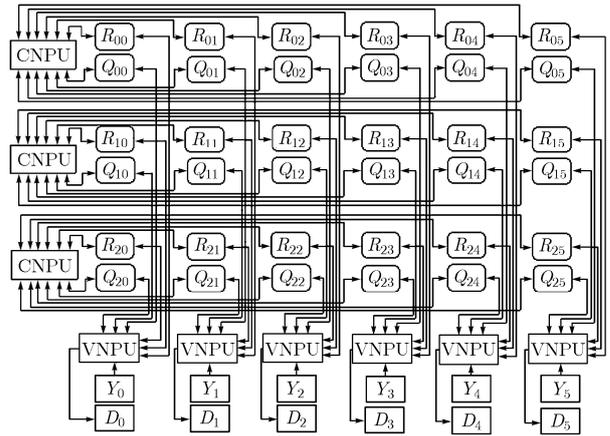
$n = lZ + (t_l + z) \bmod Z$ ，对所有 $m \in A(n)$, $Q_{nm} = 1$ ， $F = F + \sum_{m \in A(n)} R'_{mn}$ 。

由于需要优化的变量数目较多，这里采用差分进化的方法^[8]。差分进化是一种鲁棒性很高的优化算法，同时具有爬山算法和遗传算法的特性，并且易于并行实现，适用于多变量优化问题。针对具体的码字，结合上述目标函数的计算方法和差分进化算法，可以得到一组使迭代过程中软信息提前使用量最大的起始位置向量 $\mathbf{S}^*, \mathbf{T}^*$ ，供迭代译码使用。

3 译码器的硬件结构

在第2节描述译码算法的基础上，这一节讨论译码器的

硬件结构。图2给出了一个规则的准循环LDPC码对应的译码器结构图，其中 $J = 3, L = 6$ ，存储器阵列 R_{ab}, Q_{ab} ($0 \leq a < J, 0 \leq b < L$) 分别存储校验节点输出给变量节点、变量节点输出给校验节点的似然比信息，大小都是 Z ，按行的顺序存放。 $Y_0 \sim Y_{L-1}$ 用于存储初始似然比信息， $D_0 \sim D_{L-1}$ 用于存储根据 Q_n^k 硬判得到的结果。对非规则码，译码器的结构类似。



CNPU (Check Node Process Unit), 校验节点处理单元
VNPU (Variable Node Process Unit), 变量节点处理单元
图2 规则(3,6)准循环LDPC码的同步部分并行结构译码器

图3和图4分别给出了校验节点处理单元、变量节点处理单元的示意图。在每次迭代的第 z 个节拍，第 j 个校验节点处理单元处理校验矩阵第 j 行块中第 $(s_j + z) \bmod Z$ 行对应的校验节点，从相应的存储器中读出似然比信息，得到各似然比信息符号的乘积和它们绝对值的最小值、次小值，作

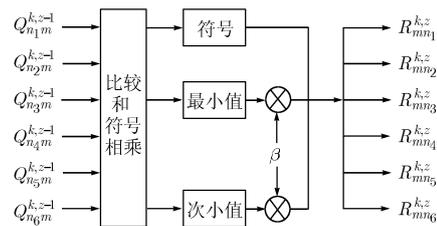


图3 校验节点处理单元

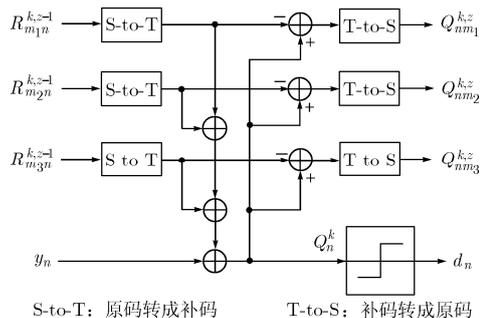


图4 变量节点处理单元

修正后乘上符号输出到相应的存储器中;第 l 个变量节点处理单元处理校验矩阵第 l 列块中第 $(t_l+z)\bmod Z$ 列对应的变量节点,先把从存储器中读出的似然比信息和初始的似然比信息求和得到 Q_n^k ,从 Q_n^k 中减去输入的各个似然比信息,就得到输出给校验节点的信息,对 Q_n^k 硬判决可以得到 d_n 。为了便于计算,变量节点处理单元内部数据用2的补码形式表示,在输入和输出端分别增加了原码转补码、补码转原码单元。

4 仿真结果和分析

在前面两节对同步部分并行译码算法和相应译码器硬件结构讨论的基础上,这一节结合具体的码字进行仿真,验证算法的性能。为了便于后面叙述,本文把3种方案分别缩写如下:标准部分并行(STPP),交叠部分并行(OLPP),同步部分并行(SCPP)。由于在迭代次数相等的情况下OLPP方案的性能与STPP方案完全相同,在比较3种方案性能时只仿真了STPP方案,OLPP方案的性能可以直接由相同迭代次数的STPP方案得到。

先看规则码的性能,首先根据文献[6]中的方法构造了码率为0.413的规则码(155,93,64,31)。图5给出了AWGN信道STPP和SCPP两种方案不同迭代次数下的误帧率(FER)曲线,其中SCPP方案校验节点处理单元的修正因子 β 取0.8,最大迭代次数设为40。为了比较初始位置向量对性能的影响,这里选取了两个不同的初始位置向量W1和W2: $S_{155}^{W1} = [16,14,27]$, $T_{155}^{W1} = [1,2,3,10,1]$; $S_{155}^{W2} = [22,18,4]$, $T_{155}^{W2} = [24,25,14,16,18]$,相应的目标函数值分别为 $F_{155}^{W1} = 148$, $F_{155}^{W2} = 465$ 。从图5两个初始位置向量对应的FER曲线可以看出,SCPP-W2的性能要优于SCPP-W1,在FER为 10^{-4} 处,增益达到0.1dB,可见初始位置向量对SCPP方案性能的影响。按照文献[4]中的方法,我们得到该码字OLPP方案的速度增益 $\eta_{155} = 1.94$,可以算出在SCPP方案40次迭代时间内OLPP方案可以进行39次迭代。从图5可以看出,SCPP-W2方案40次迭代的性能和STPP方案100次迭代的接近,在FER为 10^{-4} 处,与STPP方案20次迭代、40次迭代相比分别有0.5dB,0.25dB的增益,即在相同的迭代时间内,SCPP方案的性能要优于STPP方案和OLPP方案。

再看非规则码的性能。这里选取IEEE802.16e标准^[9]中

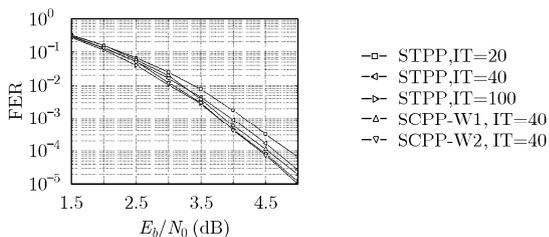


图5 (155,93,64,31)码字的FER曲线

1/2, 5/6 两种码率的码字(1056,528,528,44), (2304, 384, 1920,96)进行了仿真,得到了AWGN信道STPP方案和SCPP方案不同迭代次数下的FER曲线。其中SCPP方案中校验节点处理单元修正因子 β 的取值均为0.75,经过差分演化得到的起始位置向量如下: $S_{1056}^* = [20,5,15,25,18,6,0,12,31,16,39,32]$, $T_{1056}^* = [9,18,0,25,26,17,13,3,11,37,5,0,18,39,26,39,33,33,5,3,20,26,2,15]$; $S_{2304}^* = [48,11,36,81]$, $T_{2304}^* = [50,20,34,46,5,53,95,89,66,59,37,2,20,80,24,16,12,78,61,78,12,24,68,3]$ 。图6是(1056,528,528,44)码字的FER曲线,可以看出当FER低于 10^{-3} 后,SCPP方案40次迭代的性能介于STPP方案40次和60次迭代之间;在FER为 10^{-4} 处,SCPP方案与相同迭代次数的STPP方案相比有0.05dB的增益。图7中(2304,384,1920,96)码字SCPP方案的性能增益更加明显,当FER低于 10^{-3} 后,SCPP方案40次迭代的性能不仅好于相同迭代次数STPP方案的性能,同时超过了STPP方案60次的性能,在FER为 10^{-4} 处,与STPP方案60次迭代的性能相比有0.1dB的增益。这里也按照文献[4]中的方法计算了两个码字OLPP方案相对STPP方案的速度增益,分别为 $\eta_{1056} = 1.81$, $\eta_{2304} = 1.74$ 。在SCPP方案40次迭代时间内,OLPP方案可以对这两个码字分别进行36,35次迭代。图6和图7中SCPP方案40次迭代的性能都已经超过了STPP方案40次迭代的性能,与STPP方案20次迭代相比至少有0.5dB的增益。因此在相等的迭代时间内,SCPP方案优于STPP和OLPP两种方案。

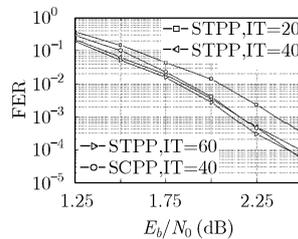


图6 (1056,528,528,44)码字的FER曲线

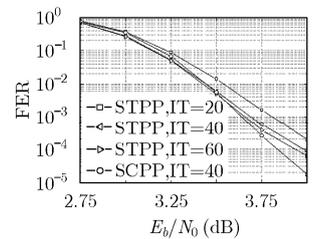


图7 (2304,384,1920,96)码字的FER曲线

从以上仿真结果可以看出,对规则码和非规则码,SCPP方案与STPP方案和OLPP方案相比,在性能上都有优势。同时,SCPP方案中对校验节点处理时采用的是修正的最小和算法,其复杂度要低于STPP和OLPP方案中校验节点的处理方法。当然,STPP方案和OLPP方案中校验节点处理时也可以采用修正的最小和算法,但是性能会受到劣化。此外,在OLPP方案中为了减少相邻两次迭代之间校验节点的等待时间,还需要调整每次迭代节点处理单元的起始位置,增加了控制的复杂度。因此,同步部分并行方案在性能和复杂度上都有一定的优势,适合于高速译码器的实现。

5 结束语

本文根据准循环LDPC码的结构特点,提出了一种同步

部分并行结构的译码器。在译码器中, 校验节点处理单元和变量节点处理单元同时并行工作, 迭代过程中提前使用新产生的软信息, 加快迭代的收敛速度。同时采用了差分演化的方法对各节点处理单元的起始位置进行优化, 进一步提高了译码器的性能。仿真结果表明, 该方案在译码性能和复杂度上都要优于现有的其他方案, 非常适合高速译码器的实现。此外, 这种同步部分并行处理的思想也可以应用到多元域 LDPC 码处理及其他迭代检测译码系统中, 提高系统性能。

参考文献

- [1] Gallager R G. Low-density parity-check codes. *IRE Trans. on Inform. Theory*, 1962, 1(8): 21-28.
- [2] Blanksby A and Howland C. A 690-mw 1-Gbps 1024-b Rate-1/2 low-density parity-check code decoder. *IEEE Journal of Solid State Circuits*, 2002, 37(3): 404-412.
- [3] Zhang T and Parhi K K. A 54 Mbps (3, 6)-regular FPGA LDPC decoder. *IEEE Proc. of SIPS*, San Diego, CA, USA, Oct. 2002: 127-132.
- [4] Chen Y and Parhi K K. Overlapped message passing for quasi-cyclic low density parity check codes. *IEEE Trans. on Circuits and Systems*, 2004, 51(6): 1106-1113.
- [5] Shimizu K, Ishikawa T, and Togawa N, *et al.* Partially-parallel LDPC decoder achieving high-efficiency message-passing schedule. *IEICE Trans. on Fundamentals of Electronics*, 2006, E89-A(4): 969-978.
- [6] Sridhara D, Fuja T, and Tanner R M. Low density parity check codes from permutation matrices. *Proceedings of 2001 Conference on Information Sciences and Systems*, The Johns Hopkins University, March 2001: 21-23.
- [7] Li Z, Chen L, and Zeng L, *et al.* Efficient encoding of quasi-cyclic low-density parity-check codes. *IEEE Trans. on Com*, 2006, 54(1): 71-81.
- [8] Price K and Storn R. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimize*, 1997, 19(11): 341-359.
- [9] IEEE P802.16e/D12. IEEE standard for local and metropolitan area networks-part 16: Air interface for fixed and mobile broadband wireless access systems (amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands). IEEE, 2006.
- [10] 赵春明, 许恩杨, 姜明等. 双涡轮结构低密度奇偶校验码解码器. 中国发明专利, 申请号 200610096535.9, 2006.9.

许恩杨: 男, 1982年生, 硕士, 研究方向为信道编解码等.

姜明: 男, 1976年生, 讲师, 研究方向为信道编解码等.

赵春明: 男, 1959年生, 教授, 博士生导师, 主要研究方向为信道编解码、无线通信系统等.