

## 一种应用于不可分层 LDPC 码的并行分层译码算法

郭琨 黑勇 周玉梅 乔树山

(中国科学院微电子研究所专用集成电路与系统实验室 北京 100029)

**摘要:** 该文针对“不可分层”LDPC 码无法利用分层算法进行译码的问题, 提出了一种并行分层置信度传播 (Parallel-Layered Belief-Propagation, PLBP) 译码算法。与传统分层算法不同, 该算法在译码时并行进行各层更新, 串行进行层内各行更新。这种译码机制使得同一变量节点在各层内不同时进行更新, 从而实现各变量节点在一次迭代中分层递进更新的算法目标。仿真表明, 在不增加译码复杂度的情况下, 该文提出的 PLBP 算法与传统的洪水算法相比, 误码性能更优, 而且所需要的平均迭代次数降低了约 50%。此外, PLBP 算法采用了合并的节点更新运算, 最终使该算法达到的译码速度约为洪水算法的 4 倍。

**关键词:** LDPC 码; 不可分层; 并行分层译码; 中国移动多媒体广播

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2010)08-1956-05

DOI: 10.3724/SP.J.1146.2009.01160

## A Parallel Layered Decoding Algorithm for Non-layered LDPC Codes

Guo Kun Hei Yong Zhou Yu-mei Qiao Shu-shan

(ASIC and System Department, Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China)

**Abstract:** As layered algorithm can not be used in the decoding of the “non-layered” LDPC codes, a Parallel-Layered Belief-Propagation (PLBP) algorithm is proposed in this paper. The decoding of this algorithm is quite different from the original layered algorithm, for it uses parallel updating among all layers and serial updating within each layer. With such a decoding schedule, a variable node in different layers is updated at different time, which means the message is able to be updated layer by layer for each variable node. The simulation shows the PLBP algorithm not only achieves a better error performance, but also requires almost 50% less iterations than the Flooding algorithm with the same complexity. Additionally, a merged computation of the nodes is employed by the PLBP algorithm. As a result, the decoding speed of the PLBP algorithm can reach nearly 4 times as that of the original Flooding algorithm.

**Key words:** LDPC codes; Non-layered; Parallel-layered decoding; CMMB

### 1 引言

随着人们对多媒体、无线通信及宽带通信等需求的不断提高, 前向纠错码日益广泛地应用于各种通信系统中。LDPC(Low-Density Parity-Check)码是一种线性分组码, 由 Gallager 于 1962 年提出<sup>[1]</sup>。在 90 年代中期 MacKay<sup>[2]</sup>发现 LDPC 码是目前误码性能最接近香农极限的信道编码。LDPC 码的译码不仅因其固有的并行性, 而适合于高吞吐率的系统; 而且, 其在高信噪比条件下的误码平台较 Turbo 码更低, 因此许多通信系统都采用 LDPC 码作为其前向纠错码, 例如 IEEE 802.11n, 802.16e, DVB-S2 及中国多媒体广播标准(CMMB)等。

LDPC 码可由 Tanner 图<sup>[3]</sup>表示, 其中变量节点和校验节点分别代表码字比特和校验方程。每次迭代中表示码字比特为 0 或 1 的概率量——置信度,

将在上述两类节点之间传递并更新, 最终实现译码判决, 因而 LDPC 码的译码算法也称为“置信度传播”(BP)算法。其译码器结构可分为全并行<sup>[4]</sup>、部分并行<sup>[5]</sup>和串行<sup>[6]</sup>3 种。而在综合考虑吞吐率和存储器开销之后, 部分并行是应用最广泛的结构。LDPC 码的译码算法按节点的更新机制可分为: 传统的“洪水”(Flooding 或称为 TPMP)算法<sup>[7]</sup>和“分层”(Layered 或称为 TDMP)算法<sup>[8]</sup>。相比之下, 分层算法的逐层递进更新的译码方式不仅使算法的误码性能更优, 而且收敛速度更快。然而部分并行结构的分层算法要求 LDPC 码的校验矩阵必须满足在同一层内的列重不大于 1 的条件。因而称满足这一条件的 LDPC 码为“可分层”码, 否则为“不可分层”码<sup>[9]</sup>。本文针对这一类“不可分层”准循环 LDPC 码, 提出了一种并行分层置信度传播(PLBP)译码算法, 该算法在每次迭代中, 各层将并行进行更新, 而每层内部则按行串行更新。由于同一变量节点在

各层内的更新时间各不相同, 每个变量节点的更新都可以通过分层串行运算完成。PLBP 算法在不增加译码复杂度的前提下, 将分层的节点更新机制扩展至“不可分层”准循环 LDPC 码, 实现了在提高系统误码性能的同时降低平均迭代次数的目标。

本文第 2 节对 LDPC 码的两种译码算法进行了简要地概述。在第 3 节中将以 CM-MB 系统中的 LDPC 码为例, 详细介绍“并行分层置信度传播”译码算法, 而仿真结果将在第 4 节中给出, 最后对文章进行了总结。

## 2 LDPC 码的译码算法

洪水算法是最传统的 LDPC 译码算法, 其特点是: 在一次迭代中, 先进行所有校验节点的更新, 再进行所有变量节点的更新。因而在一次迭代中, 变量节点将综合其相邻的所有校验节点的反馈信息作唯一一次更新。洪水算法的主要缺点是平均迭代次数较大, 从而带来译码延时较大等不足。重叠 (Overlapped) 算法<sup>[10]</sup>作为洪水算法的改进, 虽然通过时间复用减少了每次迭代所需的时间, 但由于其未触及节点更新机制的改变, 因此误码性能和算法收敛速度较传统的洪水算法并没有提升。

分层算法与洪水算法不同的是将校验矩阵按行或列分成若干“层”。在一次迭代中, 先并行更新第一层中的所有校验节点及与之相邻变量节点, 然后逐层进行更新, 每一层的更新都在前一层的基础上进行, 并以最后一层的输出作为该次迭代的输出。由于变量节点在一次迭代中分层多次被更新, 因而分层算法的误码性能更优, 同时算法收敛速度约为洪水算法的 2 倍。但为保证变量节点分层串行更新, 校验矩阵中同一层中列重必须不大于 1。为打破这一约束, 文献[9]中提出了两种方案, 它们都是对分层算法的近似, 但这种近似随着同一层内列重大于 1 的列数的增大, 性能也随之恶化。而 PTSP 算法<sup>[11]</sup>通过对分层算法的改进使系统的吞吐率获得了提高, 但付出的代价是误码性能和算法收敛速度的降低。

## 3 并行分层置信度传播算法

### 3.1 不可分层的准循环 LDPC 码

不可分层 LDPC 码是一类特殊的准循环码: 在校验基础矩阵的同一位置上存在多个元素, 即在生成校验矩阵时, 该位置将扩展为多个互相重叠的循环单位矩阵(该子矩阵的列重大于 1)。应用于 DVB-S2, CM-MB 及 IEEE 802.15.3c 等系统中的 LDPC 码都属于这类“不可分层”准循环 LDPC 码。如上所述, 分层算法收敛速度快的关键是在一次迭代中

变量节点分层多次更新。但由于传统的分层算法在译码时并行更新一层内的所有节点, 如果层内列重大于 1, 变量节点则仅能通过层内多个校验节点进行唯一一次更新, 因而带来性能的损失。针对上述的不可分层的准循环 LDPC 码, 本文提出了一种并行分层置信度传播译码算法。该算法在不增加运算复杂度的前提下, 无损地继承了传统分层算法在误码性能和收敛速度方面的优点, 是传统分层算法应用于不可分层 LDPC 码的改进形式。

### 3.2 并行分层置信度传播(PLBP)译码算法

本文所提出 PLBP 算法与传统的分层译码算法都将准循环校验矩阵的一个行块作为一层。而不同的是: PLBP 算法在译码时, 所有层并行进行更新, 而在每层内部各行的更新又是串行进行的。由于同一变量节点在各层内的更新时间各不相同, 使得更新信息可以在层间传递, 从而实现在一次迭代中各个变量节点都分层递进更新的算法目标。由于各层内部的更新是串行进行的, 所以即使对于上述的“不可分层”LDPC 码而言 PLBP 算法也适用。而在节点更新算法方面, PLBP 算法合并了两类节点的更新运算<sup>[9]</sup>, 并以归一化最小和算法<sup>[12]</sup>作为校验节点的更新算法。

**3.2.1 节点更新算法** 设  $X=[x_1, x_2, \dots, x_N]$  和  $Y=[y_1, y_2, \dots, y_N]$  分别表示发射端和接收端的相应 LDPC 码, 其校验矩阵为  $M \times N$  的准循环矩阵。在第  $k$  次迭代中,  $r_{mn}(k)$  和  $q_{nm}(k)$  分别表示从校验节点  $m$  到变量节点  $n$  的信息和从变量节点  $n$  到校验节点  $m$  的信息,  $A_n(k)$  表示变量节点  $n$  的软判决输出, 而  $N(m)$  和  $M(n)$  分别表示在 Tanner 图中与校验节点  $m$  相连的所有变量节点的集合及与变量节点  $n$  相连的所有校验节点的集合。

设变量节点的本征置信度为  $\lambda_n, n=1, 2, \dots, N$ 。

$$\lambda_n = \ln \left( \frac{p(x_n | y_n = 0)}{p(x_n | y_n = 1)} \right) \quad (1)$$

每次迭代由校验节点更新和变量节点更新组成。当校验节点更新时, 对  $m=1, 2, \dots, M, \forall n \in N(m)$ 。  $r_{mn}(k)$  可根据式(2)和式(3)计算得到。

$$\text{sgn}(r_{mn}(k)) = \prod_{n' \in N(m) \setminus n} \text{sgn}(q_{n'm}(k-1)) \quad (2)$$

$$|r_{mn}(k)| = \prod_{n' \in N(m) \setminus n} \Psi \left( \sum_{n' \in N(m) \setminus n} \Psi(|q_{n'm}(k-1)|) \right) \quad (3)$$

其中  $\Psi(x) = \ln(e^x + 1/e^x - 1)$ 。当变量节点更新时, 对  $n=1, 2, \dots, N, \forall m \in M(n)$ ,  $q_{nm}(k)$  和  $A_n(k)$  可分别由式(4), 式(5)计算。

$$q_{nm}(k) = \lambda_n + \sum_{m' \in M(n) \setminus m} r_{m'n}(k) \quad (4)$$

$$A_n(k) = \lambda_n + \sum_{m \in M(n)} r_{mn}(k) \quad (5)$$

在 PLBP 算法中，一次迭代被分割为若干次子迭代，每次子迭代中只进行一层校验节点的更新，因而两种节点的更新可以合并。将式(4)和式(5)代入式(2)和式(3)，从而得到  $r_{mn}(k)$  和第  $p$  次子迭代中  $A_n^{(p)}(k)$  的更新公式(6)-式(8)。

$$\text{sgn}(r_{mn}(k)) = \prod_{n' \in N(m) \setminus n} \text{sgn}(A_{n'}^{(p)}(k-1) - r_{mn'}(k-1)) \quad (6)$$

$$|r_{mn}(k)| = \prod_{n' \in N(m) \setminus n} \Psi \left( \sum_{n' \in N(m) \setminus n} \Psi \left( |A_{n'}^{(p)}(k-1) - r_{mn'}(k-1)| \right) \right) \quad (7)$$

$$A_{n'}^{(p)}(k) = A_{n'}^{(p)}(k-1) - r_{mn'}(k-1) + r_{mn'}(k) \quad (8)$$

$$|r_{mn}(k)| = \min_{n' \in N(m) \setminus n} \alpha \times \left( |A_{n'}^{(p)}(k-1) - r_{mn'}(k-1)| \right) \quad (9)$$

PLBP 算法采用归一化最小和算法对  $\Psi(x)$  函数

进行近似。  $r_{mn}(k)$  最终以式(9)进行运算。通过节点更新运算的合并，变量节点的更新被植入校验节点更新之中。每次迭代运算由双节点顺序更新变换为单节点更新，从而使迭代运算的速度提高了一倍。

**3.2.2 PLBP 算法流程** 下面以 CMMB 系统中的 LDPC 码<sup>[13]</sup>为例，详细说明并行分层置信度传播算法的译码过程。图 1 是经过一定变换之后的 CMMB 系统中码率为 3/4 的校验基础矩阵。其中每个矩阵元素代表  $256 \times 256$  的子矩阵，图中的数字为循环位移因子，而其中圈出的矩阵元素表示该子矩阵是由两个单位矩阵经循环位移重叠而成的。因此该矩阵无法在部分并行的结构上实现传统的分层译码。在上述的 CMMB 系统中 3/4 码率的校验矩阵中第 7, 10, 14 和 33 列块违反了“可分层”译码准则，本文以第 33 列块为例分析 PLBP 算法的译码过程。图 2 表示了第 33 列块在一次迭代中的译码过程，其中将第 4 行块分解为两个循环单位矩阵——第 4a 层(记为 L4a)和第 4b 层(记为 L4b)。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
1	0			0			208							0			0		0		0			0			0			0			0		0		
2	0			0		232				0			145			180		244			0			0			0			255			194				
3	0			255			0			0			250		248		0		0		0		219		253	189	219				186						
4	0	0		0			166			0			252		0		0		0		254	253		248			199		0		218		138		255		
5	0			128						198		254	251	179		153				249			0			0			236			0		0	0		
6	0			0						244		142		235		101	254			228				248			217			0		216			112		
7		0				192		251	237			142		245		252			229		189		119		233	104								0			
8		0				224		253		218				0		240	17			252		226					173	248						29			
9		0																																			

图 1 CMMB 系统中 3/4 码率的 LDPC 码校验矩阵

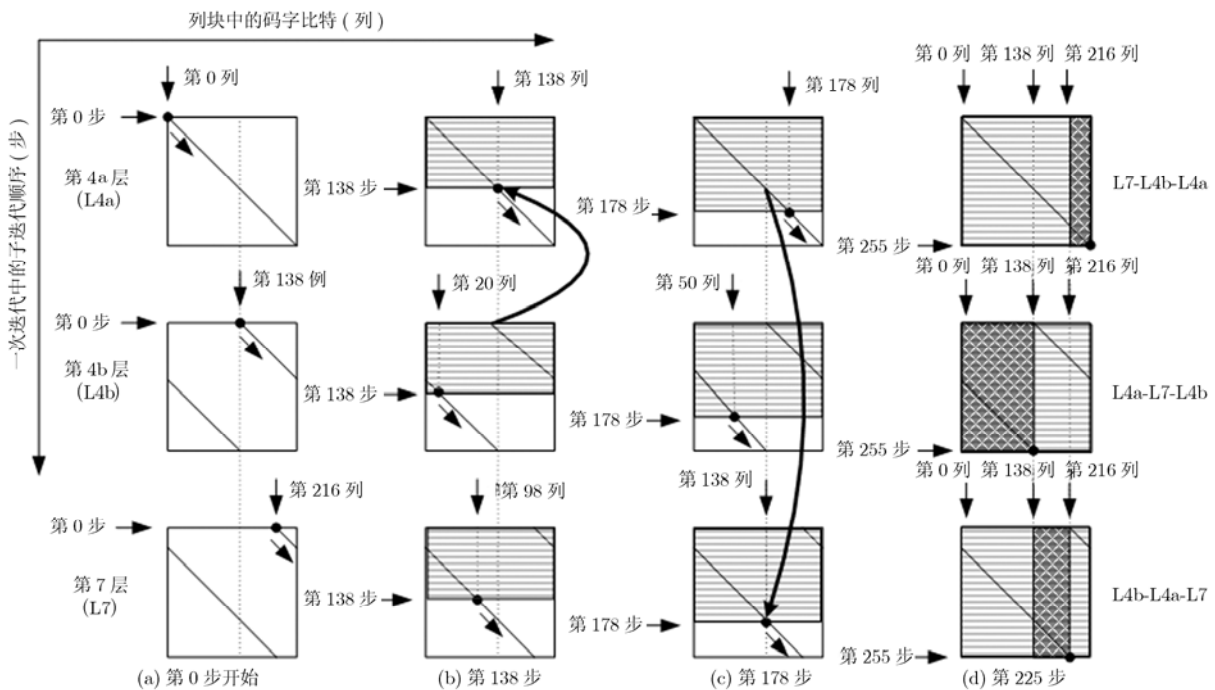


图 2 一次迭代中 PLBP 算法的译码信息传播示意图

下面具体分析 PLBP 算法的译码过程：

(1)初始化 由式(1)计算所有变量节点的本置信度  $\lambda_n$ ,  $n = 1, 2, \dots, N$ .  $\forall m \in M(n)$ ,  $q_{nm}(0) = \lambda_n$ ,  $r_{nm}(0) = 0$ 。

(2)迭代运算 在 PLBP 译码算法中, 将校验矩阵中的每一行认为是一个子层, 因而一次迭代将被分解成 256 次子迭代, 记为第 0 步至第 255 步。设每次迭代都从第 0 行开始, 顺序执行至第 255 行终止。

在第  $k$  次迭代译码中：

(a)从第 0 步开始, 在第 4a 层和第 4b 层和第 7 层中分别以第 0 列、第 138 列和第 216 列参与运算, 通过式(6), 式(8), 式(9)更新  $r_0(k)$ ,  $r_{138}(k)$ ,  $r_{216}(k)$  及  $A_0^{(0)}(k)$ ,  $A_{138}^{(0)}(k)$ ,  $A_{216}^{(0)}(k)$ , 如图 2(a)所示。随后, 这 3 层分别按行顺序进行节点更新。

(b)当进行至第 138 步时, 第 4a 层将执行第 138 列的更新, 而由于第 4b 层早在第 0 步就已经更新了该列的变量节点信息。因而由第 4b 层更新的信息  $A_{138}^{(0)}(k)$  就可以参与第 4a 层中的变量节点信息  $A_{138}^{(138)}(k)$  的计算中去了, 此时的第 138 列完成了两次变量节点更新, 如图 2(b)所示。

(c)而当迭代进行至 178 步时, 第 7 层将更新第 138 列, 此时该列已经利用第 4a 和 4b 层完成了两次更新, 因此第 7 层又可以使用最新的信息  $A_{138}^{(138)}(k)$  进行运算。至此第 138 列的信息已经通过分层算法依次完成了 3 次更新, 并以最后第 7 层的运算结果  $A_{138}^{(178)}(k)$  作为本次迭代软判决输出, 如图 2(c)所示。

(d)随着子迭代的不断进行, 当完成了第 255 步时, 类似于第 138 列的信息更新过程将发生在第 33 列块以及整个矩阵的所有列上, 因而对于每个变量节点而言, 都在一次迭代中实现了分层依次更新。但不同的是对于每列其利用各层进行更新的顺序可能不同, 如图 2(d)所示, 第 216~255 列的按层更新顺序是 L7—L4b—L4a, 第 0~137 列的更新顺序是 L4a—L7—L4b, 而第 138~215 列的更新顺序是 L4b—L4a—L7。

(3)硬判决输出 由于各列按层更新的顺序不同, 而分层算法是以最后一层的更新信息作为本次迭代的输出信息, 因此 PLBP 的译码算法其译码输出并不是集中在某一层中, 而是分散在各个层中。如图 2(d)中阴影部分所示, 第 0~137 列、138~215 列和第 216~255 列的在一次迭代中的输出信息分别位于第 4b 层、第 7 层和第 4a 层中。根据式(10)做出硬判决  $\hat{x}(k)$ , 当  $H \cdot [\hat{x}_1(k), \hat{x}_2(k), \dots, \hat{x}_N(k)]^T = 0$ , 或达到最大迭代次数, 则停止迭代开始译码输出。从上述算法分析可以看出, 本文提出的并行

分层译码算法的前提是各层在并行进行迭代运算时, 作为起点的列必须不同。只有在这种情况下, 变量节点在各层中的更新时间才是各不相同的。正因为这种不同的存在, 才使得信息在各层之间传递成为可能。因此为了保证这一前提, 可使迭代从循环位移单位矩阵中除第 1 行之外的另一行开始。

$$\hat{x}(k) = \begin{cases} 0, & A_n(k) \geq 0 \\ 1, & A_n(k) < 0 \end{cases} \quad (10)$$

本文提出的 PLBP 译码算法克服了“不可分层” LDPC 码无法在部分并行的结构上实现分层译码的缺点, 并不增加任何译码运算量的前提下, 充分发挥分层算法在误码性能和算法收敛速度方面的优势。

#### 4 仿真结果及分析

为验证本文所提出 PLBP 算法的性能, 本文将分别基于洪水算法和 PLBP 算法对 CMMB 系统中的码率为 3/4 的 LDPC 码进行比较仿真试验。仿真中采用 AWGN 信道, 调制方式为 BPSK, 输入的软判决信息采用 6 bit 量化位宽, 算法的最大迭代次数分别为 30, 20, 10 和 5 次。仿真中统计的输出比特数大于  $1 \times 10^8$ 。

图 3(a)码率为 3/4 的 LDPC 码的误比特率特性, 可以看出在相同的信噪比和相同的最大迭代次数的条件下, 本文提出的 PLBP 算法与传统的洪水算法相比, 具有更优的误码性能。其中当 PLBP 算法的最大迭代数为 20 次时误码性能甚至比最大迭代数为 30 次的洪水算法更好, 这将为系统吞吐率的提高赢得很大的空间。而算法达到收敛时所需的平均迭代次数由图 3(b)表示, PLBP 算法完成正确译码所需的平均迭代次数比传统的洪水算法要少得多, 在一些信噪比条件下仅约为洪水算法的一半。因而, PLBP 算法将大幅提高译码速度。同时, 由于 PLBP 算法中将洪水算法中的两步节点运算合并为一步(如式(6), 式(7), 式(9)所示), 所以, 每次迭代所需时间将为洪水算法的一半。综上, PLBP 算法可达到的译码速度约为传统洪水算法的 4 倍。

#### 5 结束语

本文提出了一种 LDPC 码的并行分层置信度传播译码算法, 该算法将分层译码的节点更新机制应用于那些“不可分层”的准循环 LDPC 码。译码中节点更新在各层之间并行进行, 而在各层内部串行进行。由于变量节点在各层之间的更新存在时间差, 因此信息可以在层间传递, 而使变量节点的分层更新成为可能。仿真表明, 这种并行分层置信度传播算法较传统的洪水算法具有更优的误码性能, 而所

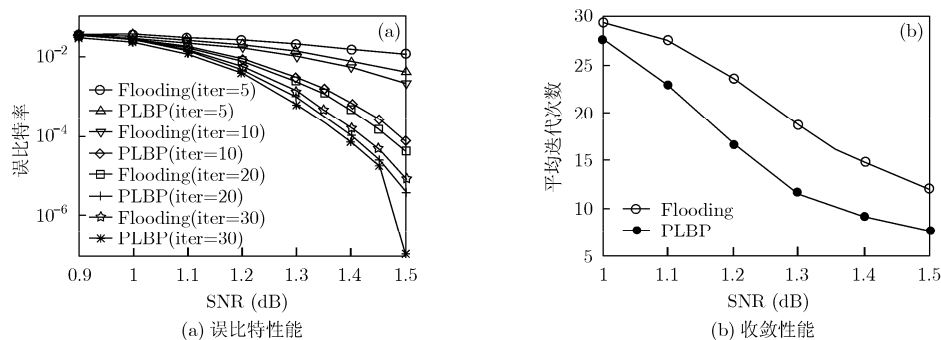


图 3 PLBP 算法与洪水算法的比较仿真

需的平均迭代次数约是洪水算法的一半。同时,该算法通过节点运算的合并使其译码速度进一步提高,最终在没有增加任何运算复杂度的情况下,并行分层置信度传播的译码速度可达洪水算法的近 4 倍。

### 参考文献

- [1] Gallager R G. Low-density parity-check codes. *IRE Transactions on Information Theory*, 1962, 8(1): 21-28.
- [2] MacKay D J C. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 1999, 45(3): 399-431.
- [3] Tanner R M. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 1981, IT-27(5): 399-431.
- [4] Tehrani S S and Mannor S, *et al.* Fully parallel stochastic LDPC decoders. *IEEE Transactions on Signal Processing*, 2008, 56(11): 5692-5703.
- [5] Park Chang-soo, Kim Seong-woon, and Hwang Sun-young, *et al.* Design of a low-area, high-throughput LDPC decoder using shared memory banks for DVB-S2. *IEEE Transactions on Consumer Electronics*, 2009, 55(2): 850-854.
- [6] Cocco M, Dielissen J, and Heijligers M, *et al.* A scalable architecture for LDPC decoding[C]. Proceedings IEEE Conference on Design Automation and Test in Europe, Paris, France, Feb. 16-20, 2004, 3: 88-93.
- [7] Mansour M M and Shanbhag N R. High-throughput LDPC decoders. *IEEE Transactions on Very Large Scale Integration Systems*, 2003, 11(6): 976-996.
- [8] Mansour M M and Shanbhag N R. A 640-Mb/s 2048-bit programmable LDPC decoder chip. *IEEE Journal of Solid-State Circuits*, 2006, 41(3): 634-698.
- [9] Rovini M, Rossi F, and Cio P, *et al.* Layered decoding of non-layered LDPC codes[C]. The 9th Euromicro Conference on Digital System Design, Dubrovnik, Croatia, Aug. 30-Sep.1, 2006: 537-544.
- [10] Shih Xin-yu, Zhan Cheng-zhou, and Lin Cheng-hung, *et al.* An 8.29mm<sup>2</sup> 52mW multi-mode LDPC decoder design for mobile WiMax system in 0.13 $\mu$ m CMOS process. *IEEE Journal of Solid-state Circuits*, 2008 43(3): 672-683.
- [11] Dai Yong-mei, Chen Ning, and Yan Zhi-yuan. Memory efficient decoder architectures for quasi-cyclic LDPC codes. *IEEE Transactions on Circuits and Systems*, 2008, 55(9): 2898-2911.
- [12] Chen J, Dholakia A, and Eleftheriou E, *et al.* Reduced-complexity decoding of LDPC codes. *IEEE Transactions on Communication*, 2005, 53(8): 1288-1299.
- [13] 国家广播电影电视总局广播科学研究院,北京泰美科技有限公司. GY/T 220.1-2006《移动多媒体广播 第 1 部分: 广播信道帧结构、信道编码和调制》[S]. 北京: 国家广播电影电视总局规划院, 2006.

郭 琨: 女, 1982 年生, 博士生, 研究方向为无线通信系统中的差错控制编码技术。

黑 勇: 男, 1974 年生, 博士, 研究员, 博士生导师, 研究方向为无线通信系统中的编码调制技术、低功耗处理器技术和专用集成电路技术。

周玉梅: 女, 1962 年生, 研究员, 博士生导师, 研究方向为深亚微米 VLSI 物理设计, 低功耗设计及数模混合集成电路技术。

乔树山: 男, 1981 年生, 博士, 研究方向为无线通信系统中的调制编码技术。