

适合行列合并译码算法的 LDPC 码的构造^①

贺一峰^② 段哲民^③

(西北工业大学电子信息学院 西安 710072)

摘要 针对低密度奇偶校验(LDPC)码译码器使用行列合并(RCJ)译码算法时存在的流水线冲突问题,提出了一种构造适合行列合并译码算法的 LDPC 码的方法。该构造方法在渐进添边(PEG)构造算法的基础上,加入行运算顺序的约束条件,使得利用该算法构造的 LDPC 码在行运算顺序上相邻两行内的非零块不同时共用一列,从而避免了硬件布局布线冲突,减少了流水线延时,提高了译码速率。仿真结果表明,用这种方法构造的 LDPC 码与全球微波互联接入(WiMAX)标准给出的 LDPC 码的译码性能相当,而且有效地减少了译码迭代次数,降低了硬件实现复杂度,提高了 LDPC 译码器吞吐率。

关键词 高速通信, LDPC 码, 行列合并译码算法, PEG 构造算法, 流水线冲突

0 引言

低密度奇偶校验(low density parity check, LDPC)码^[1]具有逼近香农极限的纠错性能和适合并行计算的简单译码算法^[2],因而已被许多通信标准采纳,如第二代数字视频广播(digital video broadcasting-second generation, DVB-S2)标准^[3]、全球微波互联接入(worldwide interoperability for microwave access, WiMAX)标准^[4]等。近年来已有不少研究人员研究了 LDPC 码的相关技术^[5,6],但随着通信速率的提升,如何提高 LDPC 码译码器的吞吐率,则成了 LDPC 码的研究难点^[7]。LDPC 码有多种译码算法,行列合并(row-column join, RCJ)译码算法^[8]是较典型的一种,但该算法存在流水线冲突问题,从而影响了译码速率的提升。现有的研究未能使流水线冲突从根本上得到解决。针对这种情况,本文从 LDPC 码构造入手进行了研究,提出了一种有效的 LDPC 码构造方法,用这种方法能构造出各种适合 RCJ 译码算法的准循环结构(quasi cyclic, QC) LDPC 码

(QC-LDPC 码),从而使 RCJ 译码器的译码性能达到全球微波互联接入(WiMAX)标准的性能。

1 相关研究

LDPC 码是线性分组码,一般用校验矩阵(即 H 矩阵)表示。LDPC 码也可以用二分图来表示,二分图分为校验节点和变量节点。校验节点代表 H 矩阵的行向量,用于实现校验约束;变量节点代表 H 矩阵中的列向量,即每个比特的码字。

LDPC 码的译码方法通常采用和积译码算法(sum product algorithm, SPA)^[9],但和积译码算法过于复杂,其采用的双曲函数并不利于硬件实现。在具体硬件实现时一般采用最小和译码算法^[10],把双曲函数改进为最小和译码算法,从而利于硬件实现。但是该译码算法收敛速度慢,计算量大,导致 LDPC 译码速率较低,并且实现该译码器需要占用大量硬件资源,包括逻辑门电路和存储资源。

针对以上 LDPC 译码器实现的缺点,He 等人提出了行列合并(RCJ)译码算法^[8]。该算法有如下优

① 国家自然科学基金(61071083)和973计划(2009CB320403)资助项目。

② 男,1983年生,硕士,高级工程师;研究方向:信道编译码理论和 LDPC 编译码器研究;E-mail: heyifeng1985@126.com

③ 通讯作者,E-mail: duanzhemin0715@126.com

(收稿日期:2015-03-25)

点:(1)减少了存储外信息上的资源消耗;(2)减少了译码器总体的硬件消耗;(3)提高了译码器吞吐率;(4)改善了 LDPC 码译码性能。由于行列合并译码算法具有快速译码性能,多将其作为 LDPC 码译码器的实现算法^[11-13]。但是行列合并译码算法的行运算器在运算完一行后才能运算下一行,而在运算每一行时必须等待上一行对数自然比 U_j 的更新结果才能开始运算。由于硬件的流水线设计和布局布线的困难, U_j 不可能及时更新,行与行运算之间存在延时,这就造成了流水线冲突问题,从而影响了译码速率的进一步提升。如果使用还没有更新的 U_j 信息,则会造成译码器性能下降。

为了解决行列合并译码器的流水线冲突问题,文献[11]在行与行运算之间加入了延时拍,这虽然解决了流水线冲突问题,但是带来了大量的延时,降低了译码器速率。Yun 等人^[12]简化了旋转矩阵的操作,降低了行与行之间的延时等待拍数,这在一定程度上减少了流水线冲突问题。有些研究重新调整了行之间运算的顺序或者行内块运算的顺序^[13],从而减少了流水线冲突问题。

可以看出,行列合并译码器的流水线冲突问题是影响其译码速率的关键问题,但现有研究只关注行列合并译码器的设计,这样只能减轻流水线冲突问题,而且还有可能带来额外的计算复杂度或者需要更多的存储空间,并不能从根本上解决流水线冲突问题。本文提出的适合行列合并译码算法的准循环结构低密度奇偶校验(QC-LDPC)码的构造方法,从根本上解决了行列合并译码器的流水线冲突问题。QC-LDPC 码的 H 矩阵由许多子矩阵构成,每个子矩阵是一个循环移位矩阵,因此,QC-LDPC 码的 H 矩阵可用基矩阵来表示,基矩阵中每个元素代表一个循环移位矩阵,这种准循环结构有利于在应用中实现 LDPC 译码器,并且保证 LDPC 的性能^[14,15]。采用本文提出的构造方法构造的 LDPC 码在行运算顺序上相邻两行中的非零块不在同一列,确保在实现译码器时不会造成流水线冲突,从而降低了行与行之间的延时,提高了译码速率。另外,采用本文提出的构造方法构造的 LDPC 码的性能并不会降低。

2 行列合并译码算法

行列合并译码算法与传统的和积译码算法相比,不需要任何列运算器,上一行计算更新的结果 U_j 可以直接用于下一行的运算,从而提高了 LDPC 码收敛速度。在译码时,其行运算器基于置信度传播算法,根据对数自然比 U_j 和外信息 v_{ij} 计算出更新的对数自然比 U'_j 和外信息 v'_{ij} 。算法的具体步骤如下:

步骤 1: 初始化

$$U_j = \log(p_j^0/p_j^1), v_{ij} = 0 \quad (1)$$

步骤 2:

$$u_{ij} = U_j - v_{ij} \quad (2)$$

步骤 3:

$$v'_{ij} = \prod_{k \in N(i) \setminus j} \text{sign}(u_{ik}) \times \phi\left[\sum_{k \in N(i) \setminus j} \phi(|u_{ik}|)\right] \quad (3)$$

步骤 4:

$$U'_j = u_{ij} + v'_{ij} \quad (4)$$

其中, $\phi(x) = \log((e^x + 1)/(e^x - 1))$, p_j^0 为接收到的第 j 个比特为 0 的概率, p_j^1 为接收到的第 j 个比特为 1 的概率,且 $1 \leq i \leq m$; $1 \leq j \leq n$, m 为校验节点个数, n 为变量节点个数,即 LDPC 码码字长度。每行运算时,使用上一行计算得到的对数自然比 U_j 和本行的外信息 v_{ij} 之差 u_{ij} ,计算得到新的本行外信息,再把新的外信息加入 u_{ij} ,得到更新的对数自然比 U'_j 。把更新的对数自然比 U'_j 和外信息 v'_{ij} 存储在寄存器中便于下次计算时用。对数自然比 U'_j 可以立即用于下一行的计算中,这样就提高了 LDPC 码的译码收敛速率,从而提高了译码器速率。在 m 行运算之后,每次迭代都会根据 U'_j 做出译码硬判决,从而判断是否迭代停止。

在行列合并译码算法步骤 3 中,运算较为复杂,并不利于译码器实现。因此,根据最小和译码算法^[10],步骤 3 中更新的外信息 v'_{ij} 的绝对值可以从 u_{ij} 的最小值减去一个偏移量 β 得到下式:

$$|v'_{ij}| = \max(\min_{k \in N(i) \setminus j} |u_{ik}| - \beta, 0) \quad (5)$$

其符号可以写成

$$\operatorname{sgn}(v'_{ij}) = \prod_{k \in N(i) \setminus j} \operatorname{sgn}(u_{ik}) \quad (6)$$

在第 5 节的仿真和实现中,本文均采用式(5)和式(6)代替步骤 3,进行具体的仿真和实现。

3 流水线冲突问题

图 1 给出了一个 QC-LDPC 码 H 矩阵的示意图,其子矩阵为 $p \times p$ 的循环移位矩阵,其中 p 为循环移位矩阵的大小。图 1 只是为了说明行列合并译码算法流水线冲突问题,因此可能存在长为 4 的环,在本文的构造算法中去除了长为 4 的环长。

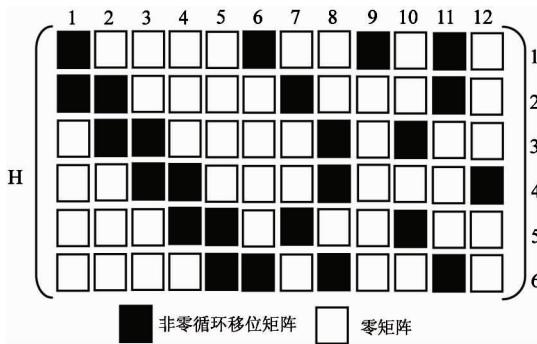


图 1 QC-LDPC 码 H 矩阵

在行列合并译码中,由于硬件的流水线设计和布局布线的难度,上一行对数自然比 U_j 不能及时更新,使得下一行运算开始时需要延时等待,这就造成了流水线冲突问题,从而影响了译码速率的提升。如图 1 中 LDPC 码 H 矩阵,第 1 行和第 2 行存在第 1 列的两个非零循环移位矩阵和第 11 列的两个非零循环移位矩阵共用一列,所以这两行运算之间需要插入延时拍,等待第 1 行算完,每一列的对数自然比计算完成后第二行再进行计算;如果 LDPC 码校验矩阵中两行内的非零元素不在同一列,这两行就可以连续运算,不存在延时。例如第 1 行和第 3 行没有非零循环移位矩阵共用同一列,这两行可以连续运算,不需要等待上一行运算完,因此第 1 行和第 3 行之间不需要添加延时拍,可以连续运算。因此,延时拍的存在极大地影响了 LDPC 译码器的速率,所以需要找到一种方法彻底地解决延时拍的问题,使译码器实现时不需要添加任何延时拍。

4 适合行列合并译码算法的 LDPC 码的构造

本文给出了一种适合行列合并译码算法的 LDPC 码的构造方法。该方法在渐进添边 (progressive edge-growth, PEG) 构造算法^[16] 的基础上加入了行运算顺序的约束,使得构造的 LDPC 码在行运算顺序上相邻两行内的非零块不同时共用一列,行与行运算之间不需要加入延时拍,从而解决了行列合并译码器行与行运算之间的延时问题,从根本上解决了行列合并译码器的流水线冲突问题,提高了译码速率。

4.1 循环移位矩阵位置的确定

确定每一个循环移位矩阵的位置也就是确定基矩阵每一个非负元素的位置,等价于为基于子矩阵的二分图中的每一条边线确定端点。由于二分图的周长是影响 LDPC 码性能的关键因素之一,扩大基于子矩阵的二分图的周长,基于比特的二分图周长也相应得到扩大。而基于子矩阵的二分图周长与每条边线权重无关,具体步骤如下:

(1) 初始化

(a) 向基于子矩阵的二分图中填充 m 个没有任何连接的校验节点,它们的当前维度为 0,记作 $\{d_e^1, d_e^2, \dots, d_e^m\}$;

(b) 根据维度分布,将 n 个变量节点按照维度从低到高排列为 $\{v_1, v_2, \dots, v_n\}$,它们的维度为 $\{d_v^1, d_v^2, \dots, d_v^n\}$;

(c) 设定行运算顺序。

(2) 构造:For $i = 1$ to n

(a) 取出变量节点 v_i 添加到基于子矩阵的二分图中,当前维度 $d_i = 0$,已用的校验节点集合为空;

(b) 以 v_i 为根节点,按照 PEG 算法,对当前二分图做树状展开,如图 2 所示;

(c) 统计距离根节点最远的校验节点集合 $\{c'_1, c'_2, \dots\}$,它们的维度为 $\{d'_{e1}, d'_{e2}, \dots\}$;

(d) 从 $\{c'_1, c'_2, \dots\}$ 中挑选出与已用过的校验节点集合 $\{c_1^{\text{used}}, c_2^{\text{used}}, \dots\}$ 在行运算顺序上不相邻的校验节点集合 $\{c''_1, c''_2, \dots\}$,它们的维度为 $\{d''_{e1}, d''_{e2}, \dots\}$;

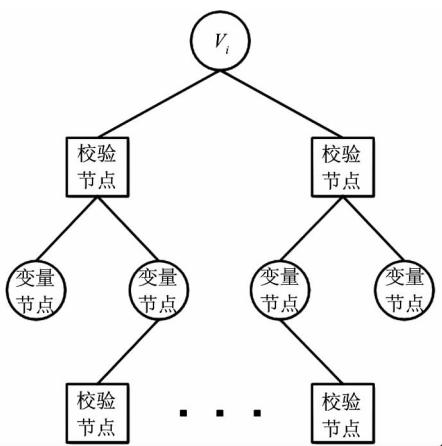


图 2 LDPC 码树状展开图

(e) 从 $\{d''_{c1}, d''_{c2}, \dots\}$ 中挑选出最小的维度 d_c^j , 找到与之对应的校验节点 c_j , 其中 $1 \leq j \leq m, m$ 为校验节点总个数。如果同时存在多个校验节点, 其维度相同且最小, 就从中随机选择一个;

(f) 把 c_j 添加到以 v_i 为根节点的二分图树状展开中, 并且把 c_j 添加到已用的校验节点集合 $\{c_{used}^1, c_{used}^2, \dots\}$ 中;

(g) $d_i = d_i + 1$, 判断 d_i 是否等于 d_v^i , 如果是, 则返回步骤(2)(a), 否则返回步骤(2)(b)。

上述确定每一个循环移位矩阵位置的过程中, 根据预先设定的行运算顺序, 在挑选校验节点时加入约束。例如设定行运算顺序为 $\{1, 2, 3, 4, 5, \dots\}$, 该变量节点树状展开图中已用的校验节点集合为 $\{c_1, c_3\}$, 那么 $\{c_2, c_4\}$ 校验节点就不能再连接到该变量节点了。这样保证了构造的 LDPC 码在行运算顺序上相邻两行内的非零块不同时共用一列。

4.2 循环移位偏移量的确定

构造 QC-LDPC 码的另外一个关键步骤是确定每一个循环移位矩阵的偏移量, 即基于子矩阵的二分图中每一条边线的权重。当基于子矩阵的二分图中的环路确定以后, 边线权重的选择最终决定了基于比特的二分图中环的长度。又由于每新增一条边线, 都会导致二分图中新增多个环路, 而周长仅与其中长度最短的环路相关, 因此本文在选择边线权重时只考虑该边线在基于子矩阵的二分图中参与的最短环路。

设边线参与的最短环路共有 t 个, 边线权重值

为 x , 则可以得到 t 个环路的路径累计值集合:

$$\{s_i = (\sum_{k=0}^{2l-2} (-1)^k p_{i_k j_k} - x) \bmod p \quad (7)$$

特别地, 当最短环路长度为 4 时, 就必须计算出不能选择的边线权重值集合:

$$X = \{x_i \mid (\sum_{k=0}^{2l-2} (-1)^k p_{i_k j_k} - x_i) \bmod p = 0, 0 \leq x_i < p, 1 \leq i \leq t\} \quad (8)$$

在此基础上, 本文给出两种确定边线权重的准则: 环路最大化原则和随机原则。随机原则是一种挑选权重的低复杂度准则。该准则只保证 QC-LDPC 码不出现长度为 4 的环, 即在选取权重时, 只保证不选择出现在式(8)中集合 X 的所有数值, 对于其余的任意小于 p 的非负数值进行等概率随机挑选。随机原则的最大优点就是复杂度很低, 因为每一个循环移位矩阵位置的选择是在保证基于子矩阵的二分图环路最大的基础上进行的, 所以随机挑选的权重并不会导致二分图中出现大量的短环。最大化环路原则以最大化局部环路为最终目的。当选择的边线权重 x 使得环路权重累计值 s 与子矩阵大小 p 互质时, 得到的环路长度是最长的, 因此在采用最大化环路原则确定权重时, 必须针对每一个环路计算环路累计值 s_i , 得出满足互质条件的权重候选集合 Y_i , 最终的权重候选集合 $Y = \cap Y_i$, 如果 Y 为空集, 则按照随机原则的方法, 从所有不会构成 4 元环的权重中随机选择一个; 如果不为空, 则在 Y 中随机挑选一个数值作为权重。不难发现, 当 p 为素数时, 随机原则与最大化环路原则是等价的。

4.3 总算法流程

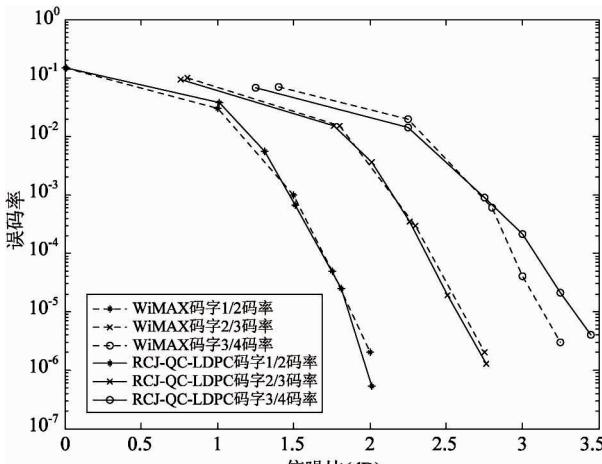
在基于子矩阵的二分图边线连接关系完全确定之后再确定每条边线的权重, 面临的将是一个环路长度的全局优化问题, 复杂度太高。根据 PEG 算法, 本文采用优化局部环路长度, 在确定每一个循环移位矩阵位置之后, 立即确定它的循环移位偏移量, 即选择每一个循环移位偏移量时仅考虑它对当前二分图环路的影响。总算法流程如下:

(1) 初始化

(a) 利用外互信息传递 (extrinsic mutual information transfer, EXIT) 图^[17] 等方法确定变量节点的维度分布;

(b) 初始化基于子矩阵的二分图;

- (c) 设定行运算顺序。
- (2) 构造:向基于子矩阵的二分图中逐个添加变量节点,挑选校验节点建立连接,并确定边线权重
- 向二分图中添加新的变量节点;
 - 以当前变量节点为根节点,将二分图展开为树状图,如图 2;
 - 挑选距离根节点最远的校验节点集合,并从中挑选出与已用的校验节点集合在行运算顺序上不相邻的校验节点;
 - 从中选择当前维度最低的校验节点作为目的节点,并把这个校验节点加到已用的校验节点集合中。如果同时存在多个校验节点,其维度相同且最小,就从中随机选择一个;
 - 遍历根节点到目的节点的所有路径,根据 $s = \sum_{k=0}^{l-1} (-1)^k p_{i_k j_l}$ 计算累积权重 s ;
 - 根据式(8)求出根节点到目的节点、边线不能选择的权重集合以及候选权重集合;
 - 从候选权重集合中挑选出边线的权重;
 - 判断如果根节点的维度已经满足初始维度分布,则向二分图中添加下一个变量节点,否则返回步骤(2)(b);



(a) 2016 bit 码长

图 3 RCJ-QC-LDPC 码与 WiMAX 标准中 LDPC 码字误码率性能比较曲线图

本研究根据提出的构造适合 RCJ 译码算法的 LDPC 码的方法构造了码长为 2016bit, 码率分别为 1/2、2/3 和 3/4 的 RCJ-QC-LDPC 码(图 3(a)); 以

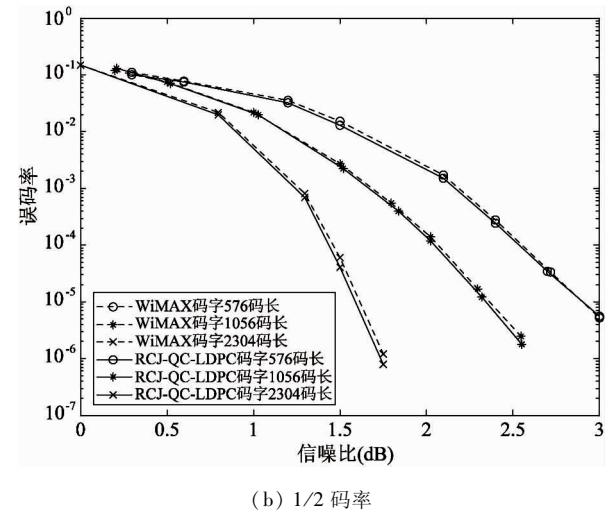
- 判断如果所有 n 个变量节点都已经被添加, 基矩阵构造完成, 否则返回步骤(2)(a);
- 根据构造得到的基矩阵, 在校验矩阵相应位置填充相应偏移量的单位矩阵的循环移位矩阵, 构造结束。

5 仿真与实现

5.1 仿真分析

根据本文提出的适合行列合并(JRC)译码算法的 LDPC 码的构造方法, 本文构造了与 WiMAX 标准^[4]中相同码长码率的 QC-LDPC 码, 称为行列合并(CRJ)准循环低密度奇偶校验码(RCJ-QC-LDPC)。本文与 WiMAX 标准中的码字进行了性能仿真比较, 仿真信道采用二进制加性高斯白噪声(binary input additive white gaussian noise, BIAWGN)信道, 本文构造的 RCJ-QC-LDPC 码采用行列合并译码算法, 采用式(5)和(6)的最小和译码方式进行行运算, 最大迭代次数 15 次。

图 3 比较了 RCJ-QC-LDPC 码与 WiMAX 标准 LDPC 码的误码率性能。



(b) 1/2 码率

及码率为 1/2, 码长分别为 576bit、1056bit 和 2304bit 的 RCJ-QC-LDPC 码(图 3(b)), 并分别将其与 WiMAX 标准^[4]中相同码长、码率的 LDPC 码进行了

误比特性能的仿真比较。其中, WiMAX 标准中的 LDPC 码采用和积译码算法, 最大迭代次数 15 次。仿真结果表明, 本文构造的 QC-LDPC 码与 WiMAX 标准中相应参数的 LDPC 码性能相当, 尤其在低码率 $1/2$ 和 $2/3$ 码率时, 对于任何码长(576、1056、2016 以及 2304bit)均无性能损失, 但 2016 码长高码率 $3/4$ 码率时性能在 10^{-4} 以下开始出现性能损失, 此时本文构造的 RCJ-QC-LDPC 码与 WiMAX 标准中的 LDPC 码具有 0.2dB 的性能差距。

本文研究了迭代次数对本研究构造的 RCJ-QC-LDPC 码的性能影响(见图 4)。本研究构造了码长为 2016bit, 码率分别为 $3/4$ 和 $5/6$ 的 LDPC 码, 采用行列合并译码算法, 最大迭代次数分别为 30 次、15 次和 4 次。从仿真结果可以看出, 降低迭代次数, 对 RCJ-QC-LDPC 码性能影响不大, 尤其对高码率 $5/6$ 码率的 RCJ-QC-LDPC 码, 迭代 4 次的仿真结果与迭代 30 次、15 次的仿真结果误码性能相同, 对 $3/4$ 码率 RCJ-QC-LDPC 码, 在误码率为 10^{-6} 时, 迭代 15 次与迭代 30 次误码性能相差 0.2dB 。这样的仿真结果表明, 本文构造的 RCJ-QC-LDPC 码在译码时不需要采用多次迭代完成译码, 只需要少量的迭代就可以快速收敛完成译码, 从而提高了译码速率。尤其是高码率 $5/6$ 码率的 RCJ-QC-LDPC 码, 减少迭代次数并不会影响译码性能, 可以用于实现 LDPC 码高速译码器, 具体实现方案将在 5.2 节中给出。

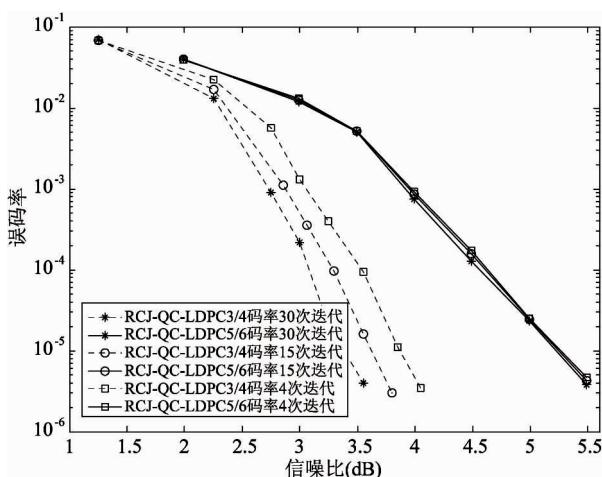


图 4 RCJ-QC-LDPC 码在不同迭代次数时误码率性能曲线图

5.2 实现

本文在 Xilinx 芯片 XC5VFX130T 上采用行列合并译码算法实现了本文构造的 RCJ-QC-LDPC 码, 其原理框图见图 5。首先信道信息存储器存储来自信道的信道信息, 之后在译码过程中, 信道信息存储器负责存储更新的对数自然比 U'_j 。外信息存储器存储每行运算过程中的外信息 v_{ij} , 以便用于下次行运算。行运算器应用式(5)和(6)的最小和译码方式进行行运算。减法器用于实现行列合并译码步骤(2), 计算对数自然比 U_j 和本行的外信息 v_{ij} 之差 u_{ij} 。加法器用于实现行列合并译码步骤(4), 计算本行新的外信息 v'_{ij} 和 u_{ij} 之和, 得到的结果 U'_j 更新到信道信息存储器中, 用于下一行运算中。移位器用于调整信道信息 U_j 的排列顺序, 便于与外信息 v_{ij} 相减。行列合并译码器信息吞吐率可以按下式进行计算:

$$\text{译码器信息吞吐率} = \frac{\text{译码器时钟速率} \times \text{码长} \times \text{码率}}{\text{译码所需时钟拍数}} \quad (9)$$

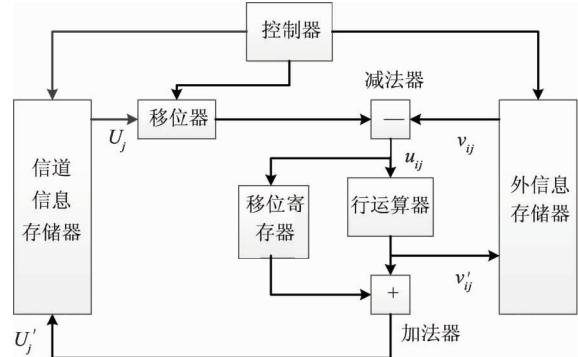


图 5 行列合并译码器实现框图

译码所需时钟拍数为每次迭代所需时钟拍和迭代次数的乘积。表 1 比较了本文构造的 RCJ-QC-LDPC 码和 WiMAX 中 LDPC 码应用本文设计的行列合并译码器的信息吞吐率。码长均为 2016bit, 码率分别为 $3/4$ 和 $5/6$ 。本文构造的 RCJ-QC-LDPC 码子矩阵大小和 WiMAX 中的 LDPC 码大小相同, 同为 84×84 。应用行列合并译码算法时, WiMAX 中 LDPC 码在行与行运算之间要插入 8 拍延时拍, 而本文构造的 RCJ-QC-LDPC 码由于其构造时就考虑了行列合并译码算法的特性, 因此行与行运算之间不需

要插入延时拍。8 拍延时拍包括 1 拍用于写入信道信息存储器,2 拍用于读取信道信息存储器,2 拍用于移位信道信息,3 拍用于行运算。码长为 2016 比特,码率为 $3/4$,子矩阵大小为 84×84 的 LDPC 码型基矩阵为 6 行,因此 WiMAX 中的 LDPC 码每次迭代所需时钟拍比本文构造的 RCJ-QC-LDPC 码多 48 拍。码长为 2016bit,码率为 $5/6$,子矩阵大小为 84×84 的 LDPC 码型基矩阵为 4 行,因此 WiMAX 中的 LDPC 码每次迭代所需时钟拍比本文构造的 RCJ-QC-LDPC 码多 32 拍。依据图 4 的仿真结果,码率为 $3/4$ 的 LDPC 码型迭代次数均采用 15 次,码率为 $5/6$ 的 LDPC 码型迭代次数均采用 4 次,用于实现高速 LDPC 码译码器。译码器时钟速率为 200MHz,本文构造的 RCJ-QC-LDPC 码和 WiMAX 中的 LDPC 码的译码器吞吐率结果见表 1。

表 1 LDPC 码译码器吞吐率比较

码型	码率	迭代次数	每次迭代所需时钟拍数	译码器信息吞吐率
WiMAX LDPC 码	$3/4$	15	156	129Mbps
WiMAX LDPC 码	$5/6$	4	147	571Mbps
RCJ-QC-LDPC 码	$3/4$	15	108	187Mbps
RCJ-QC-LDPC 码	$5/6$	4	115	730Mbps

如表 1 所示,对于 $3/4$ 码率的 LDPC 码,采用 15 次迭代,本文构造的 RCJ-QC-LDPC 码译码器可以比 WiMAX 中的 LDPC 码译码器信息吞吐率提高 45%。对于 $5/6$ 码率的 LDPC 码,采用 4 次迭代,本文构造的 RCJ-QC-LDPC 码译码器可以比 WiMAX 中的 LDPC 码译码器信息吞吐率提高 28%,达到了 730Mbps,从而实现了对 LDPC 码进行高速译码。

6 结 论

本文针对行列合并译码算法的流水线冲突问题给出了一种有效的 LDPC 码构造方法。该构造方法使得构造的 LDPC 码校验矩阵,在行运算顺序上,相邻两行内的非零块不同时共用一列,解决了行列合并译码算法行与行运算之间的流水线冲突问题。利

用本文的 LDPC 码构造方法,可以构造各种各样的性能优异的 RCJ-QC-LDPC 码。仿真结果表明,本文构造的 RCJ-QC-LDPC 码性能与 WiMAX 标准中 LDPC 码性能相当,并且降低了译码迭代次数,对于 RCJ-QC-LDPC 码性能影响不大。本文针对 LDPC 码译码器信息吞吐率与 WiMAX 标准中 LDPC 码译码器进行了比较,结果表明,本文构造的 RCJ-QC-LDPC 码译码器比 WiMAX 标准中 LDPC 码译码器吞吐率提高 28% ~ 45%,可以实现 LDPC 码高速译码器,从而更好地适应现代高速通信的需求。

参 考 文 献

- [1] Gallager R G. Low density parity check codes: [Ph. D dissertation]. Cambridge: MIT Press, 1963. 30-35
- [2] MacKay D J C, Neal R M. Near Shannon limit performance of low-density parity-check codes. *Electronic Letter*, 1997, 33(6): 1645-1646
- [3] Marchand C, Boutillon E. Before convergence early stopping criterion for inner LDPC code in DVB standards. *Electronics Letters*, 2015, 51(1): 114-116
- [4] Andrade J, Falcao G, Silva V. Flexible design of wide-pipeline-based WiMAX QC-LDPC decoder architectures on FPGAs using high-level synthesis. *Electronics Letters*, 2014, 50(11): 839-840
- [5] 姜小波, 叶德盛, 吴文涛等. 低功耗异步 LDPC 解码器运算通路设计. *电子学报*, 2013, 41(4): 685-689
- [6] Yue Z, Lau F C M. Implementation of decoders for LDPC block codes and LDPC convolutional codes based on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 2014, 25(3): 663-672
- [7] Cushon K, Hemati S, Leroux C, et al. High-throughput energy-efficient LDPC decoders using differential binary message passing. *IEEE Transactions on Signal Processing*, 2014, 63(3): 619-631
- [8] He Z Y, Sébastien R, Paul F. FPGA implementation of LDPC decoders based on joint row-column decoding algorithm. In: Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, USA, 2007. 1653-1656
- [9] You S D, Shun J H. Influences of inaccurate estimation of noise variance in sum-product algorithm for DVB-T2 receiver. In: Proceedings of the 2014 IEEE International

- Conference on Consumer Electronics (ICCE), Las Vegas, USA, 2014. 27-28
- [10] Angarita F, Valls J, Almenar V, et al. Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor. *IEEE Transactions on Circuits and Systems*, 2014, 61(7) : 2150-2158
- [11] Xin M Z, Ying T. High-speed multi-block-row layered decoding for quasi-cyclic LDPC codes. In: Proceedings of the 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Atlanta, USA, 2014. 11-14
- [12] Yun C, Qi C Z, Di W, et al. An efficient multirate LDPC-CC decoder with a layered decoding algorithm for the IEEE 1901 standard. *IEEE Transactions on Circuits and Systems II*, 2014, 61(12) : 992-996
- [13] Haroun A, Nour C A, Arzel M, et al. Low-complexity layered BP-based detection and decoding for a NB-LDPC coded MIMO system. In: Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 2014. 5107-5112
- [14] Li J, Liu K, Lin S, et al. A matrix-theoretic approach to the construction of non-binary quasi-cyclic LDPC codes. *IEEE Transactions on Communications*, 2015, 63 (4) : 1057-1068
- [15] 黎海涛, 杨磊磊, 刘飞等. 多元 LDPC 译码器的设计与实现. 高技术通讯, 2013, 23(12) : 1299-1307
- [16] Xue Q J, Xiang G X, Moon H L. Efficient progressive edge-growth algorithm based on Chinese remainder theorem. *IEEE Transactions on Communications*, 2014, 62 (2) : 442-451
- [17] Narasimhan T L, Chockalingam A. EXIT chart based design of irregular LDPC codes for large-MIMO systems. *IEEE Communications Letters*, 2013, 17(1) : 115-118

Construction of LDPC codes suitable for the row-column joint decoding algorithm

He Yifeng, Duan Zhemin

(Department of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072)

Abstract

To deal with the pipeline collision problem of a low density parity check (LDPC) codes' decoder using the row-column joint (RCJ) decoding algorithm, a method for construction of the LDPC codes suitable for the RCJ decoding algorithm is proposed. The proposed construction method adds a constraint condition in the order of line operation on the basis of the progressive edge-growth (PEG) construction algorithm, to guarantee that the non-zero matrix in two adjacent lines does not share the same column in the order of line operation to avoid the pipeline collision, reduce the pipeline delay, and increase the decoding rate. The simulation results show that the performance of the LDPC codes constructed by using the proposed method is about the same as that of the LDPC codes proposed by the communication standard of WiMAX. Meanwhile, the proposed construction method can effectively decrease the number of decoding iteration, reduce the complexity of decoder implementation, and greatly improve the throughput of LDPC decoders.

Key words: high-speed communication, LDPC codes, row-column joint decoding algorithm, PEG construction algorithm, pipeline collision