第 33 卷第 11 期 通 信 学 报 Vol.33 No. 11
2012 年 11 月 Journal on Communications November 2012
doi:10.3969/j.issn.1000-436x.2012.11.021

基于 FPGA 的部分并行 QC-LDPC 译码器高效存储方法

袁瑞佳[1,2]，白宝明[1,2]

(1. 西安电子科技大学 综合业务网国家重点实验室，陕西 西安 710071；
2. 中电科技集团公司第 54 研究所 通信网信息传输与分发技术重点实验室，河北 石家庄 050002)

摘 要：针对部分并行结构的准循环低密度校验(QC-LDPC)码译码器，提出了一种将译码准码字存储在信道信息和外信息存储块中的高效存储方法，该方法不需要额外的存储块来存储译码准码字，能够减少译码器实验所需的存储资源数量，并且有效降低了译码电路的布线复杂度。在 Xilinx XC2V6 000-5ff1 152 FPGA 上的实验结果表明，提出的 QC-LDPC 码译码器设计方法能够在降低系统的 BRAM 资源需求量的同时有效地提高系统的运行频率和译码吞吐量。

关键词：LDPC 码；译码器；部分并行；高效存储；FPGA 实验
中图分类号：TN 911 **文献标识码：**A **文章编号：**1000-436X(2012)11-0165-06

Efficient storage method for FPGA-based partially parallel QC-LDPC decoder

YUAN Rui-jia[1,2], BAI Bao-ming[1,2]

(1.State Key Lab of Integrated Services Networks, Xidian University, Xi'an 710071, China; 2.Science and Technology on Information Transmission and Dissemination in Communication Networks Lab, CETC No.54 Research Institute, Shijiazhuang 050002, China)

Abstract: An efficient storage method of hard decisions sharing intrinsic and extrinsic memory banks for partially parallel QC-LDPC decoder was proposed. Extra memory banks for storing hard decisions were avoided in this method, which result in significantly reduced consumption of RAM resources and routed complexity. Implementation results based on a Xilinx XC2V6 000-5ff1 152 FPGA show that the proposed method improves the frequency and decodes throughput of the system, and significantly reduced the requirements for the number of BRAM.

Key words: LDPC code; decoder; partially parallel; efficient storage; FPGA implementation

1 引言

LDPC 码是一类校验矩阵非常稀疏的线性分组码，由 Gallager 于 1962 年提出[1]，后来 MacKay、Spielman 等证明在长分组情况下其性能比 Turbo 码更接近 Shannon 容量限[2,3]。由于 LDPC 码具有纠错性能优异、译码复杂度低和可实验并行译码等优点，目前已被数字卫星广播 (DVB-S2)、无线局域网(WLAN)和全球微波接入互操作性(WiMAX)等多个通信标准采纳。

收稿日期：2011-08-22；修回日期：2011-12-27
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（2012CB316100）；国家自然科学基金资助项目（60972046）；国家科技重大专项基金资助项目（2010ZX03003-003）；通信网信息传输与分发技术重点实验室开放课题基金资助项目（ITD-U1007）
Foundation Items: The National Basic Research Program of China (973 Program)(2012CB316100); The National Natural Science Foundation of China (60972046); The National S&T Major Project of China (2010ZX03003-003); Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory (ITD-U1007)

在大规模集成电路上实验高速的LDPC码编译码器一直是LDPC码应用研究的一个焦点。理论上，可以根据二部图中的所有节点及边线实验全并行结构的LDPC码译码器，但随着码长的增长，过高的布线复杂度及庞大的资源需求量将导致全并行结构的译码器难以实验。采用串行结构虽然可以减少硬件资源的消耗量，但译码器所需的存储空间会随着码长的增长而迅速增加，并且其较低的译码吞吐量通常不能满足实际应用的需求。2001年Kou等提出了准循环结构的LDPC(QC-LDPC)码[4]。通过合理的构造方法，QC-LDPC码能够获得与随机构造LDPC码相比拟的性能[5]，由于其校验矩阵的准循环特性，QC-LDPC码能够采用部分并行结构设计出基于长码的高吞吐量译码器，实验硬件资源需求量和译码吞吐量的有效折中[6,7]。然而，由于部分并行结构译码器自身具有资源复用和并行处理的双重特性，与全并行和串行2种结构的译码器相比，其对存储块资源的需求量要大得多。一方面，由于每个节点处理单元分时处理不同的节点，部分并行结构译码器必须对非当前处理节点的数据作缓存处理，而全并行结构译码器由于处理单元和数据间的一对一关系，则完全免于这部分开销。另一方面，由于各节点处理单元的并行处理关系，部分并行结构译码器需要同时读写多路数据，因此不像串行结构译码器那样只需单一的大容量存储块，而同时需要大量不同的存储块。现阶段部分并行结构QC-LDPC译码器实验的研究工作已全面展开，文献[8~11]给出了几种有效的实验方法，但这些实验中BRAM资源的开销仍然较大，其中，文献[8]设计的译码器所占用的BRAM资源数量远小于文献[9~11]。文献[8~11]中给出的译码方法均需要单独的存储块缓存译码准码字，一般地，译码准码字所使用的存储块个数占译码器总存储块个数的比例都比较大。

本文提出了一种将译码准码字存储在信道信息和外信息存储块中的译码存储结构，该结构能够在不增加逻辑资源使用量的同时节省用于存储译码准码字的BRAM资源。

2 LDPC码的译码

令X表示一个长为N的LDPC码，其校验矩阵为$\boldsymbol{H}=[h_{mn}]_{M\times N}$。下面以归一化最小和算法为例，介绍LDPC码的译码过程，对于其他LDPC码译码算法其迭代过程类似，只有水平运算和垂直运算的计算处理有所变化。

LDPC码的迭代译码是根据信道接收的对数似然比（LLR）信息$\{f_n\}$对译码码字$\boldsymbol{c}=\{c_n\}$进行迭代修正的一个过程，每次迭代包含垂直运算和水平运算2部分，垂直运算过程处理变量节点的更新，水平运算过程处理校验节点的更新。以符号v_n表示$\boldsymbol{H}$矩阵中第n列对应的变量节点，q_m表示第m行对应的校验节点，定义$M(n)=\{m:h_{mn}=1\}$为与v_n相连的校验节点集合，$N(m)=\{n:h_{mn}=1\}$为与q_m相连的变量节点集合，$M(n)\backslash m$表示除m外所有与v_n相连的校验节点集合，$N(m)\backslash n$表示除n外所有与q_m相连的变量节点集合。符号Q_{mn}^{iter}、R_{mn}^{iter}和Q_n^{iter}分别表示第$iter$次迭代中v_n向q_m传递的似然比信息、q_m向v_n传递的似然比信息和v_n取值的似然比信息，a为译码归一化修正因子。

归一化最小和算法的译码步骤如下。

1) 初始化：对所有m和n，将R_{mn}^{iter}的初始值置为0，迭代次数$iter$设为1次。

2) 垂直运算：对所有n及$m\in M(n)$，计算$Q_{mn}^{iter}=f_n+\sum_{m'\in M(n)\backslash m}R_{m'n}^{iter-1}$和$Q_n^{iter}=f_n+\sum_{m\in M(n)}R_{m'n}^{iter-1}$。

3) 水平运算：对所有m及$n\in N(m)$，计算$R_{mn}^{iter}=\left(\prod_{n'\in N(m)\backslash n}\operatorname{sign}(Q_{mn'}^{iter})\right)\cdot a\cdot\min_{n'\in N(m)\backslash n}\left|Q_{mn'}^{iter}\right|$。

4) 译码判决：判决Q_n^{iter}得到c_n，检查c_n是否满足所有校验。若校验方程组满足或已达到最大迭代次数，将c_n作为译码结果输出，否则将$iter$累加1后转至步骤2)继续迭代。

3 高效存储译码架构

在现有的一些译码器设计方法中，对于一个校验矩阵由mn个大小为$L\times L$的子矩阵构成的规则(j,k)QC-LDPC码，如果设计的部分并行结构译码器的复用指数为L，处理数据的量化比特数为Q，那么译码器总共需要n个$j+1$输入的变量节点处理单元$\mathrm{VNU}_1,\mathrm{VNU}_2,\cdots,\mathrm{VNU}_n$，$m$个$k$输入的校验节点处理单元$\mathrm{CNU}_1,\mathrm{CNU}_2,\cdots,\mathrm{CNU}_m$以及$m$个$k$输入的校验方程计算单元$\mathrm{PCU}_1,\mathrm{PCU}_2,\cdots,\mathrm{PCU}_m$。在数据的存储上还需要$n$个存储块$F_t(0\le t\le k-1)$用于存储信道信息和$mn$个存储块$M_{s,t}(0\le s\le m-1,0\le t\le n-1)$用于存储$\mathrm{VNU}_t$和$\mathrm{CNU}_s$处理过程中

交换的外信息，其中，每个存储块包含 L 个存储单元，存储单元的位宽均为 Q。在现有的译码器结构中[8~11]，还需要 $\lceil j/2 \rceil k$ 到 jk 个存储块用于存储译码准码字，存储块的地址均采用模 L 的计数器产生。由于垂直运算处理完成后可同时产生水平运算和校验方程来计算所需的外信息和译码准码字，且水平运算和校验方程计算没有数据处理交集，因此 CNU 和 PCU 可并行工作，本文针对两者的数据流向作存储资源复用，提出了能够节省译码准码字存储块的译码器结构，该结构与采用的具体译码算法无关。图 1 给出了一个基于规则(j, k) QC-LDPC 码的译码器结构示例，译码器的数据处理过程如下。

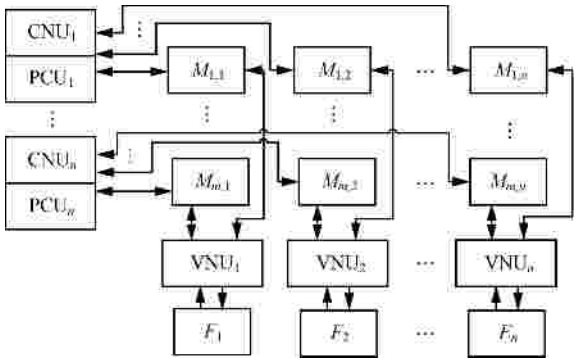

图 1　(j, k)规则 QC-LDPC 的高效存储译码器架构

1) 初始化。初始化迭代次数 $iter$ =1，最大迭代次数为 MAX_ITER。将接收到的一帧信道信息存储到存储块 F_t 中，并将所有存储块 $M_{s,t}$ 的内容初始化为 0。其中，F_t 中的数据按列顺序存储，$M_{s,t}$ 中的数据按行顺序存储。如果系统采用先水平运算后垂直运算的迭代顺序，$M_{s,t}$ 中译码迭代外信息部分的初始值应设置为信道接收信息。

2) 变量节点更新。n 个变量节点处理单元 VNU_1, VNU_2,…,VNU_n 同时从 $M_{s,t}$ 中读取上一次迭代校验节点更新的外信息，进行变量节点的更新。在每个时钟周期，第 t 列对应的变量节点处理单元 VNU_t 从存储块 $M_{1,t}$,$M_{2,t}$,…,$M_{j,t}$ 和 F_t 中各读取 j 路外信息和一路信道信息。处理后产生 j+1 路输出，其中一路是所有输入数据的符号位异或和，即 1bit 的译码准码字，其他 j 路分别是外信息的总和减去不同的 j 路外信息的结果。在数据存储的设计上，译码器将 1bit 的译码准码字作为高位分别与 j 个 Q bit 的外信息合并后，以上一次迭代的信息读出地址 L- $x_{s,t}$, … ,L- 1,0, … ,L- $x_{s,t}$- 1 回写到存储块 $M_{1,t}$, $M_{2,t}$,…,$M_{j,t}$ 中，$x_{s,t}$ 为第 t 列的第 j 个存储块的矩阵偏移量。采用这样的存储方式使得译码准码字在 $M_{s,t}$ 中被重复存储 m 份，但因此也使得译码器能够同时从不同的偏移量开始读取 m 份译码准码字，因而使 PCU 在计算校验和的时候能够采用与校验节点处理单元 CNU 相同的并行度进行处理。同样地，信道信息的存储也是将 1bit 的译码准码字并入 Qbit 的信道信息中，并以读取 F_t 的地址顺序 0,…,L- 1 写入 F_t，F_t 中的译码准码字主要用于译码结果的缓存输出。每次译码迭代总共需要 L 个时钟周期来完成 nL 列的变量节点更新。

3) 校验节点更新和校验方程的计算。所有的 CNU_1, CNU_2,…,CNU_m 和 PCU_1, PCU_2,…, PCU_m 是并行工作的。每个时钟周期，CNU_s 和 PCU_s($0 \le s \le m-1$)从存储块 $M_{s,1}$,$M_{s,2}$,…,$M_{s,k}$ 中各读取一路数据，将读取得到的 k 路$(Q+1)$ bit 数据拆分为 Q bit 的外信息和 1bit 的译码准码字，其中外信息送入 CNU_s 处理，译码准码字送入 PCU_s 处理。将 CNU_s 处理后输出的 k 个 Q bit 外信息扩展位宽后以读写顺序 0,…,L- 1 回写到 $M_{s,1}$,$M_{s,2}$,…,$M_{s,k}$ 中，由于读写地址一致，所有的 k 个存储块共享同样的地址产生单元。m 个 PCU_s 输出 m 个不同的伴随式分量，如果全部伴随式分量均为 0，则表示校验方程已全部满足，译码器停止迭代并输出译码结果。否则，将迭代次数加 1 后转到步骤 2)继续进行迭代，直到达到最大迭代次数 MAX_ITER。每次译码迭代总共需要 L 个时钟周期来完成所有 mL 行校验节点的更新。

4　存储资源消耗量分析

在提出的高效存储译码架构中，校验节点更新与校验方程计算是同时进行的，变量节点更新完毕后，将迭代后的外信息和译码准码字分别作为校验节点处理单元和校验方程计算单元的输入。译码器总共需要 $n+mn$ 个存储块，其中，mn 个存储块 $M_{s,t}$ $(0 \le s \le m-1,\ 0 \le t \le n-1)$用于存储外信息，$n$ 个存储块 F_t $(0 \le t \le n-1)$用于存储信道信息，译码准码字共享用于存储信道信息和外信息的存储块，每个存储块 $M_{s,t}$ 和 F_t 的大小均为 L $(Q+1)$ bit。采用该存储结构的另一个优点在于读取 $M_{s,t}$ 时 PCU 与 CNU 可以共享同一组地址信息，使得 PCU 不需要额外的地址产生单元，从而能够减少逻辑资源的消耗量。在现有的一些译码器设计中[8~11]，文献[8]的实验方法使用的存储块数量最少，共需要

$n+mn+\lceil m/2\rceil n$ 块 BRAM 资源，而本文提出的译码结构只需 $n+mn$ 块。将节省 BRAM 资源的百分比记为 r，那么

$$r=\frac{\lceil m/2\rceil n}{n+mn+\lceil m/2\rceil n}=\frac{1}{(1+m)/\lceil m/2\rceil+1} \quad (1)$$

由于 m 为正整数，因此 $\lceil m/2\rceil$ 的取值与 m 的奇偶性有关，当 m 为奇数时，$\lceil m/2\rceil=(m+1)/2$；当 m 为偶数时，$\lceil m/2\rceil=m/2$，因此有以下关系

$$m/2\le\lceil m/2\rceil\le(m+1)/2 \quad (2)$$

由式(1)和式(2)可得

$$m/(3m+2)\le r\le 1/3 \quad (3)$$

又因为 m 为正整数，即有以下关系

$$1\le m\le+\infty \quad (4)$$

由式(4)可得到

$$1/5\le m/(3m+2)\le 1/3 \quad (5)$$

联合式(3)和式(5)的结果，即可得到 r 的取值范围为

$$1/5\le r\le 1/3 \quad (6)$$

由上面的分析可知，本文的设计方法较文献[8]的设计能够节省 20%~33.3%的 BRAM 资源，而且当行分块数 m 为奇数时，节省的资源百分比 r 恒为 33.3%；当 m 为偶数时，r 在 20%和 33.3%之间取值，且 m 越大时 r 也越大。

5 主要模块的硬件实验

针对本文提出的高效存储译码架构，笔者在实际的 FPGA 硬件平台上进行了译码器实验验证。为了提高译码器的处理速度，扩大系统的吞吐量，译码器的各模块均采用了多级流水线技术，下面列举主要功能模块的设计。

5.1 变量节点处理单元（VNU）

在 LDPC 码的迭代译码过程中，变量节点处理单元以加减运算为主，数据以补码形式表示有利于简化操作，而校验节点处理单元则要求比较、求绝对值和乘法运算，更适合于用原码表示。RAM_M 中的外信息采用原码形式存储，因此在变量节点更新时需要进行原码和补码间的转换。本文实验的 VNU 流水线结构如图 2 所示，其中，i 代表 VNU 单元的外信息输入路数，即对应变量节点的连线数。为了降低 VNU 的关键路径延迟，垂直更新过程被设计为 5 级流水线，图中的虚线表示流水线的划分。in_1,…, in_i 表示 i 路不同的外信息输入，f_n 表示信道接收信息，c 表示译码码字。加法器阵列的主要功能是将所有外信息和信道接收信息加和，减法器阵列则将加和结果分别减去各路输入，求得更新后的外信息值。由于中间运算的数据位宽大于存储位宽，为了避免数据精度的丢失，截位模块需要对中间数据进行截位操作。在外信息输入路数较多的情况下，加法器阵列的流水线操作还可以继续细化，从而进一步地减少译码电路的关键路径延迟。

5.2 校验节点处理单元（CNU）

CNU 的实验可分为上下 2 个处理部分，上部分主要完成对外信息绝对值的比较和选择运算，下部分完成对输入数据符号的异或操作，CNU 总体被划分为 5 级流水线，其结构如图 3 所示。第 1 级流水线用于分离输入数据的符号位和绝对值；第 2 级流水线求出各输入外信息绝对值的最小值和次小值以及对各符号位进行加和；第 3 级流水线将求得的最小、次小值乘以 a，并从符号位总和中减去各输入外信息的符号，求出迭代后的外信息符号；第 4 级流水线根据各外信息是否等于最小值，从修正后的最小、次小值中选择 j 路绝对值；第 5 级流水线将数值大小和符号位合并得到更新后的外信息。

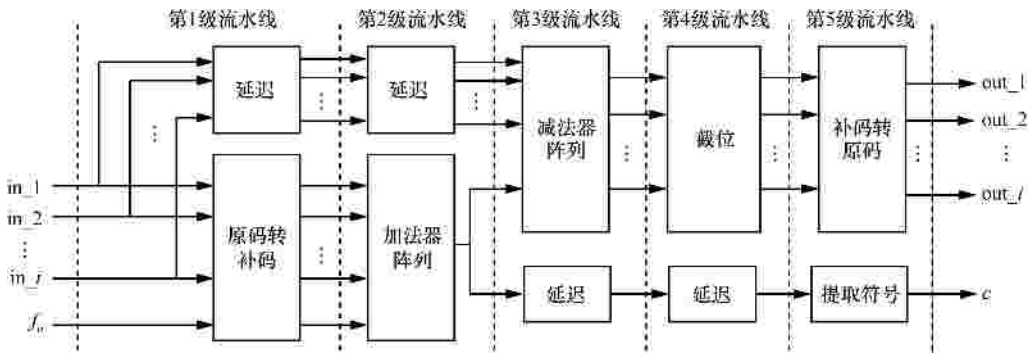

图 2 VNU 的流水线结构

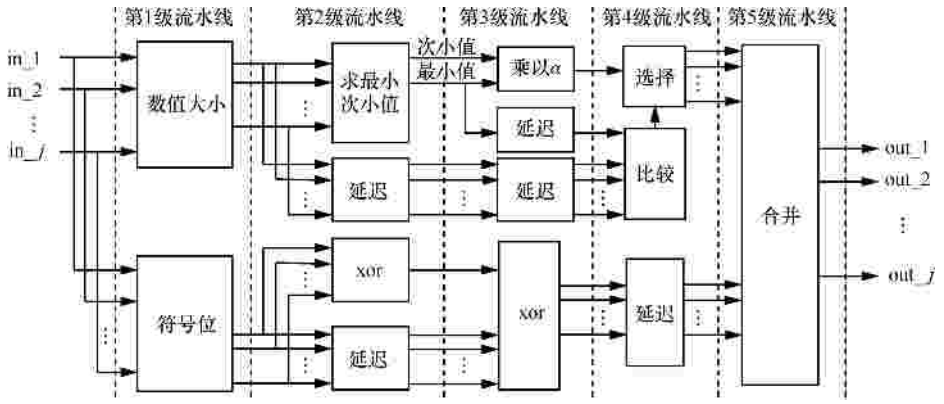

图 3　CNU 的流水线结构

表 1　采用提出的高效存储方法设计的译码器与文献[8]的实验结果对比

存储方法	slice	F/F	LUT	Block RAM	关键路径时延/ns	吞吐量/(Mbit·s^{-1})
文献[8]的方法	1 477	1 012	2 648	36	6.802	58.8
本文提出方法	1 485	1 682	2 050	24	4.739	84.4

当 j 值较大时，为了提高处理速度，第 2 级流水线中的求最小、次小值操作还可以进一步细化流水线。当修正因子 a 选择为 0.75 时（对多数 LDPC 码，a 选择 0.75 有较好的修正效果），第 3 级流水线中的乘法电路可以采用减法及移位操作来代替，从而简化计算电路。

6　实验结果

采用本文提出的高效存储设计方法对文献[8]中码长为 1 536 的规则(3, 6) QC-LDPC 码进行部分并行结构的译码器实验，芯片使用 Xilinx 公司的 XC2V6 000-5ff1 152 FPGA，在 ISE 10.1 平台上进行逻辑综合和布局布线，在 ModelSim 6.2 上进行功能仿真和时序仿真，具体的硬件资源占用情况和吞吐量如表 1 所示。

表 1 将本文的实验结果和文献[8]设计的译码器进行了比较。为了比较的公平性，设计的译码器采用了与文献[8]相同的硬件平台 Xilinx Vertex II xc2v6 000-5ff1 152 FPGA，2 种实验采用的码字、译码算法和流水线级数也相同，分别是码长为 1 536 的规则(3, 6) QC-LDPC 码、8bit 量化的归一化最小和译码算法和 5 级流水线电路。该 FPGA 芯片的总资源为 33 792 slice 和 144 块 18kbit 的 BRAM。表 1 分别列出了 2 种实验所消耗的 slice、F/F（触发器）、LUT（查找表）和 BRAM 资源、译码器的关键路径时延和信息吞吐量。从表 1 可以看到，采用提出的高效存储方法设计得到的译码器占用的 slice 资源数量为 1 485，略多于文献[8]译码器所消耗的 1 477 slice。Vertex II 系列的 FPGA 芯片中每单位 slice 资源包含 2 个 F/F 单元和 2 个 LUT 单元。由实验数据可见，文献[8]设计的译码器电路使用的 LUT 资源较多，而 F/F 资源较少，分别占用了所有使用 slice 资源中 89.7%的 LUT 和 34.4%的 F/F。对于硬件电路的布局布线，某一种资源的占用比例过高将使得电路难以获得较小的关键路径时延，从而限制电路的工作主频。而采用本文提出方法设计的译码器中，F/F 和 LUT 2 种资源的占用比例较为均衡，分别占用了所使用 slice 资源中 69%的 LUT 和 56.7%的 F/F，均衡的资源分布使电路更容易获得较短的关键路径时延，实验结果的对比也验证了这一点。从表 1 中的数据可见，本文设计的译码器具有更小的关键路径时延，因而能够实验更高的译码吞吐量。在 BRAM 的占用数量上，采用本文提出方法设计的译码器一共需要 24 块 BRAM，其中，包含 mn=18 块信道信息存储块和 n=6 块外信息存储块，与文献[8]占用的 36 块 BRAM 相比，可节省 33.3%的存储块资源。

7　结束语

QC-LDPC 码凭借其特殊的结构特性，能够采用部分并行结构实验基于长码的高吞吐量译码器。然而，由于部分并行结构的 QC-LDPC 码译码器本身具有资源复用和并行处理的双重特性，其对存储块资源的需求量要比全串行结构和全并

行结构的译码器大得多。现有的译码器设计方法中均需要数量较多的存储块存储译码过程中由判决产生的译码准码字，本文提出了一种使译码准码字与信道信息和外信息共享存储资源的高效存储设计方法，该方法能够有效减少 QC-LDPC 码译码器实验对 BRAM 资源数量的需求，并且能够简化译码电路的布线复杂度。基于 Xilinx Vertex II xc2v6 000-5ff1 152 FPGA 的实验结果表明，本文提出的存储方法与文献[8]的设计方案相比可以节省约 33%的 BRAM 资源，其中，文献[8]的存储设计方案明显优于其他现有的设计方案，实验结果还表明采用该设计方法得到的 QC-LDPC 码译码器的布线复杂度比文献[8]的方案更简单，能够实验更高的系统时钟频率，因而能够达到更大的译码吞吐量。

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] MACKAY D J C, NEAL R M. Near Shannon limit performance of low density parity check codes[J]. Electronics Letters, 1996, 32(18): 1645- 1646.

[3] SPIELMAN D A. Linear-time encodable and decodable error-correcting codes[J]. IEEE Transactions on Information Theory, 1996, 42(11): 1723-1731.

[4] KOU Y, LIN S, FOSSORIE R. 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.

[5] FOSSORIER M P C. Quasi-cyclic low-density parity-check codes from circulant permutation matrices[J]. IEEE Transactions on Information Theory, 2004, 50(8):1788-1793.

[6] WANG W J, WU X G, ZHU X X. A 223Mbit/s FPGA implementation of (10240, 5120) irregular structured low density parity check decoder[A]. Vehicular Technology Conference[C]. Calgary, Canada, 2008. 767-771.

[7] HONG Q, WANG J, LEI W. A resource-efficient decoder architecture for LDPC codes[A]. International Conference on Electrical and Control Engineering (ICECE)[C]. Dhaka, Bangladesh, 2010.244-248.

[8] DAI Y M, YAN Z Y, CHEN N. Optimal overlapped information passing decoding of quasi-cyclic LDPC codes[J]. IEEE Transactions on Very Large Scale Integration System, 2008, 16(5):565-578.

[9] MIN H K, TAE D P, CHUL S K. An FPGA design of low power LDPC decoder for high-speed wireless LAN[A]. Communication Technology (ICCT)[C]. Nanjing, China, 2010.1460-1463.

[10] ZHANG L M, GUI L, XU Y Y. Configurable multi-rate decoder architecture for QC-LDPC codes based broadband broadcasting system[J]. IEEE Transactions on Broadcasting, 2008, 54(2):226-235.

[11] CHEN X, KANG J, LIN S. Memory system optimization for FPGA based implementation of quasi-cyclic LDPC codes decoders[J]. IEEE Transactions on Circuits and Systems, 2011, 58(1):98-111.

作者简介：

袁瑞佳 1982-

白宝明 1966-