

基于 FPGA 的 LDPC 码编译码器联合设计

袁瑞佳 白宝明*

(西安电子科技大学综合业务网国家重点实验室 西安 710071)

(通信网信息传输与分发技术重点实验室 石家庄 050002)

摘 要: 该文通过对低密度校验(LDPC)码的编译码过程进行分析,提出了一种基于 FPGA 的 LDPC 码编译码器联合设计方法,该方法使编码器和译码器共用同一校验计算电路和复用相同的 RAM 存储块,有效减少了硬件资源的消耗量。该方法适合于采用校验矩阵进行编码和译码的情况,不仅适用于全并行的编译码器结构,同时也适用于目前广泛采用的部分并行结构,且能够使用和积、最小和等多种译码算法。采用该方法对两组不同的 LDPC 码进行部分并行结构的编译码器联合设计,在 Xilinx XC4VLX80 FPGA 上的实现结果表明,设计得到的编码器和译码器可并行工作,且仅占用略多于单个译码器的硬件资源,提出的设计方法能够在不降低吞吐量的同时有效减少系统对硬件资源的需求。

关键词: 数字通信系统; LDPC 码; 编码器; 译码器

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2012)01-0038-07

DOI: 10.3724/SP.J.1146.2011.00539

FPGA-based Joint Design of LDPC Encoder and Decoder

Yuan Rui-jia Bai Bao-ming

(State Key Lab. of Integrated Services Networks, Xidian University, Xi'an 710071, China)

(Science and Technology on Information Transmission and Dissemination in Communication Networks Lab, Shijiazhuang 050002, China)

Abstract: A joint design of FPGA-based encoder and decoder of LDPC codes is proposed. In this new design, the LDPC encoder and decoder share the same parity-check calculation circuit and the same RAM block, resulting in significantly reduced resource consumption in hardware implementations. The design is suitable for encoding and decoding realizations based on parity-check matrix. It can accommodate full-parallel architectures both for the encoder and decoder, or partial-parallel architectures that are widely adopted nowadays. Furthermore, various decoding algorithms such as the sum-product and the min-sum algorithms can be adopted in this design. The proposed joint design method is applied to design the encoder and decoder of two different groups of LDPC codes, both with a partial-parallel structure. The implementation based on an Xilinx XC4VLX80 FPGA shows that the designed encoder and decoder can work well in a parallel way, and only consumes slightly more hardware resources than that required by a single decoder. As a result, the proposed design can effectively reduce the hardware consumption without sacrificing the throughput.

Key words: Digital communication system; LDPC code; Encoder; Decoder

1 引言

在数字通信系统中,采用前向纠错编码(FEC)技术可大幅提高数据传输的可靠性。其中,LDPC 码凭借其优异的纠错性能、较低的译码复杂度和可并行实现的译码结构,已被数字卫星广播(DVB-S2)、无线局域网(WLAN)和全球微波互联接入

(WIMAX)等多个通信标准采纳。

如何在大规模集成电路上实现高速的 LDPC 编译码器一直是 LDPC 码应用研究的一个焦点。理论上,可根据二部图中的所有节点及边线实现全并行结构的 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现。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 等人^[1]提出了准循环结构的 LDPC 码,由于其校验矩

2010-06-02 收到,2011-09-07 改回

国家自然科学基金(60972046),新一代宽带无线移动通信网重大专项(2009ZX03003-011,2010ZX03003-003)和通信网信息传输与分发技术重点实验室开放课题(ITU-U1007)资助课题

*通信作者:白宝明 bmbai@mail.xidian.edu.cn

阵的准循环特性，译码器可以采用部分并行结构进行实现，从而使得基于长码的高吞吐量LDPC码译码器得以实用化。现阶段部分并行结构QC-LDPC译码器实现的研究工作已全面展开，文献[2-8]给出了几种有效的基于FPGA的译码器实现方法。在编译码器的实现上，如果直接采用生成矩阵编码，那么算法复杂度将随着码长的增加以平方速度递增，2001年Richardson等人^[9]提出了一种简化的编码算法，但其所需的存储和运算量依然过于庞大。此后，学者们提出了一些结构化的编码方法，其中一种是设计生成矩阵也具有准循环性质的LDPC码，利用生成矩阵进行编码，文献[10]给出了该方法的编码电路；另一种方法是采用特殊结构的准循环校验矩阵进行编码，例如IEEE 802.16e (WiMAX), IEEE 802.11n(WLAN)等标准采用了一种双对角结构的准循环校验矩阵进行编码，针对该结构，文献[11]提出了一种快速递归编码算法及该算法的并行硬件实现架构。但现有的LDPC码编译码器设计方法都只针对编码器或译码器单独设计，并各自优化其吞吐量和FPGA资源，而实际的应用系统多数属于全双工通信系统，系统需要编译码器两者同时工作，采用分离式设计将使得编译码器和译码器的硬件资源无法得到有效复用，导致FPGA资源的利用率不高。

本文针对IEEE 802.16e, IEEE 802.11n等标准建议的LDPC码，提出了一种编码器与译码器联合设计方法，该方法使得编码器能够利用译码器在垂直运算时闲置的校验计算电路进行编码。提出的实现结构中，待编码数据、待译码数据和译码码字共用相同的存储块，编译码器的各处理单元共享同一组地址信息，有效减少了编译码系统对硬件资源的需求。

2 基于H矩阵的LDPC编码

IEEE 802.16e, IEEE 802.11n等标准采用的准循环双对角结构LDPC码能够利用校验矩阵实现快速编译码，在工程领域得到了广泛应用。这类LDPC码的校验矩阵H由多个大小相等的子矩阵构成，每个子矩阵为零方阵或单位阵向右循环移位的置换阵。若以-1表示 $z \times z$ 的全零矩阵，非负整数 a 表示 $z \times z$ 单位阵循环右移 a 位后的置换阵，那么一个大小为 $m \times n$ 的准循环校验矩阵H可由一个 $m_b \times n_b$ 的基矩阵 H_b 唯一表示， H_b 中的每个元素对应H中的一个子矩阵，其中码长 $n = z \times n_b$ ，校验位个数 $m = z \times m_b$ ，信息位个数 $k = z \times k_b$ 。将基矩阵 H_b 分解为两部分 $H_b = [(\mathbf{H}_{b_1})_{m_b \times k_b} |$

$(\mathbf{H}_{b_2})_{m_b \times m_b}]$ ， \mathbf{H}_{b_1} 对应系统位部分， \mathbf{H}_{b_2} 对应校验位部分， \mathbf{H}_{b_2} 满足如下条件： \mathbf{H}_{b_2} 的第1列 \mathbf{h}_b 是 m_b 维的列向量，其中只含3个不等于-1的元素，且有 $h_b(0) = h_b(m_b-1) \neq -1$ ，这里将第3个非-1元素的行号记为 x 。 \mathbf{H}_{b_2} 中除 \mathbf{h}_b 以外的部分 \mathbf{H}'_{b_2} 是个双对角阵，用 i 和 j 表示其行号和列号，当 $i = j$ 和 $i = j + 1$ 时矩阵中对应位置的元素为0，其它元素均为-1，如式(1)所示。

$$\mathbf{H}_{b_2} = [\mathbf{h}_b | \mathbf{H}'_{b_2}] = \begin{bmatrix} h_b(0) & 0 & & & \\ -1 & 0 & 0 & -1 & \\ \vdots & & 0 & \ddots & \\ h_b(x) & & \ddots & 0 & \\ \vdots & & & -1 & 0 & 0 \\ h_b(m_b-1) & & & & & 0 \end{bmatrix} \quad (1)$$

以上的结构特性，使得该类LDPC码能够以循环递归的方式进行快速编码^[19]。若将一个码字 \mathbf{c} 分成输入的信息序列 \mathbf{s} 和编码产生的校验序列 \mathbf{p} 两部分，将 \mathbf{s} 和 \mathbf{p} 再分解为 k_b 和 m_b 组 z 维子向量，那么码字可表示为 $\mathbf{c} = [\mathbf{s} | \mathbf{p}] = [\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{k_b-1}, \mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m_b-1}]$ ，用符号 $L_{i,j}$ 表示矩阵 \mathbf{H}_{b_1} 中第 i 行第 j 列元素对应的 $z \times z$ 子矩阵，令 $h_b(0) = h_b(m_b-1) = l$ ， $h_b(x) = l_x$ ，且令

$$\mathbf{b}_i = \sum_{j=0}^{k_b-1} L_{i,j} \mathbf{s}_j, \quad 0 \leq i \leq m_b - 1 \quad (2)$$

那么由 $\mathbf{c} \times \mathbf{H}^T = \mathbf{0}$ ，可以得到

$$\mathbf{b}_0 + \mathbf{p}_0^{(l)} + \mathbf{p}_1 = \mathbf{0} \quad (3)$$

$$\mathbf{b}_i + \mathbf{p}_i + \mathbf{p}_{i+1} = \mathbf{0}, \quad 1 \leq i \leq m_b - 2, i \neq x \quad (4)$$

$$\mathbf{b}_x + \mathbf{p}_0^{(l_x)} + \mathbf{p}_x + \mathbf{p}_{x+1} = \mathbf{0} \quad (5)$$

$$\mathbf{b}_{m_b-1} + \mathbf{p}_0^{(l)} + \mathbf{p}_{m_b-1} = \mathbf{0} \quad (6)$$

其中 $\mathbf{p}_i^{(l)}$ 表示 \mathbf{p}_i 向右循环移动 l 次后的向量。

将式(3)-式(6)对应于不同行的 m_b 个方程相加可得

$$\mathbf{p}_0^{(l_x)} = \sum_{i=0}^{m_b-1} \mathbf{b}_i \quad (7)$$

求出 \mathbf{p}_0 后，可依次利用式(3)到式(6)递归求解得到其它各 \mathbf{p}_i 值

$$\mathbf{p}_1 = \mathbf{b}_0 + \mathbf{p}_0^{(l)} \quad (8)$$

$$\mathbf{p}_2 = \mathbf{b}_1 + \mathbf{p}_1 \quad (9)$$

⋮

$$\mathbf{p}_{m_b-1} = \mathbf{b}_{m_b-1} + \mathbf{p}_0^{(l)} \quad (10)$$

将求解得到的各校验向量 \mathbf{p}_i 与信息序列相结合即可得到系统形式的码字。由式(2)-式(10)的编码过程可以看到，式(2)中求解每个 \mathbf{b}_i 向量需要将多

个矩阵与向量相乘的结果相加,且共有 m_b 个 \mathbf{b}_i 向量需要求解,即便由于矩阵的结构特性使得矩阵乘法可以使用向量的循环移位代替,其计算量仍然较大。而其它式子均为2~3个向量的加和,其计算量实际较少,因此整个编码器的计算复杂度主要集中于式(2)。通过对式(2)分析可知, \mathbf{b}_i 实际由 \mathbf{H} 矩阵每行对应位为1的各信息位相加得到,显然,如果将码字序列的校验位部分全部置零后利用 \mathbf{H} 矩阵求解其伴随式,那么得到的伴随式与 \mathbf{b}_i 等效。因此可以利用系统中现存且闲置的校验计算电路,通过把校验位置零后计算伴随式的方法将其复用到对信息位的编码上。本文的编译码器资源复用方案即利用了这种校验计算电路编码的思想进行设计。

3 LDPC 码的译码过程分析

下面以归一化最小和算法为例,分析LDPC码的译码过程及常见的译码器实现方式,结果表明:常规LDPC码译码器实现中包含一校验计算电路,且由于译码算法本身的交替迭代性质导致该校验计算电路在译码过程中有一半的时间处于空闲状态。

LDPC码的迭代译码是通过信道接收信息 f_n 对译码码字 c_n 进行修正的一个过程,每次迭代包含垂直运算和水平运算两部分,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 \mathbf{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 $M(n)=\{m: \mathbf{H}_{mn}=1\}$ 为与 v_n 相连的校验节点集合, $N(m)=\{n: \mathbf{H}_{mn}=1\}$ 为与 q_m 相连的变量节点集合, $M(n)/m$ 表示除 m 外所有与 v_n 相连的校验节点集合, $N(m)/n$ 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, α 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0次;

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

(3)水平运算:对所有 m 及 $n \in N(m)$ 计算 $R_{mn}^{\text{iter}} = \left(\prod_{n' \in N(m) \setminus n} \text{sign}(Q_{mn'}^{\text{iter}}) \right) \times \alpha \times \min_{n' \in N(m) \setminus n} |Q_{mn'}^{\text{iter}}|$;

(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。

现有的LDPC码译码器实现通常被划分为3个主要工作模块^[5-12],其中变量节点处理单元(VNU)完成垂直运算,校验节点处理单元(CNU)完成水平运算,校验方程计算单元(PCU)完成译码判决。从上面的译码过程可知,垂直运算和水平运算不能同时进行,每次迭代中垂直运算需要以水平运算产生的 $R_{mn}^{\text{iter}-1}$ 作为输入,水平运算反过来又需要垂直运算的结果 Q_{mn}^{iter} ,因此在同一时间里VNU与CNU必有一方空闲。另外,PCU的校验处理必需安排在VNU对 Q_n^{iter} 信息更新完毕之后,但其与CNU没有数据处理交集,因此PCU与CNU通常被设计为并行工作。对于部分并行结构的译码器,各模块的并行度一般按照校验矩阵的分块来划分。对于子矩阵大小为 $z \times z$ 的校验阵,每轮迭代的前半部分各个VNU对其对应分块的 z 列变量节点顺序进行更新,若VNU数据更新处理的流水线长度为 L_v ,那么VNU模块在一次迭代中的工作总时长为 $z + L_v$ 个系统时钟周期。相似地,每轮迭代的后半部分各个CNU对其对应分块的 z 行校验节点逐行更新,若CNU数据更新处理的流水线长度为 L_c ,那么CNU模块在一次迭代中的工作总时长为 $z + L_c$ 个系统时钟周期。而PCU则在CNU处理的同时顺序将 z 行的校验和进行加和,另外还需要 L_p 个时钟周期检查伴随式的各分量是否全部为0,因此CNU模块在一次迭代中的工作总时长为 $z + L_p$ 个系统时钟周期。对于大部分校验阵, z 比 L_v , L_c 和 L_p 要大得多,而且 L_v , L_c 和 L_p 通常大小相近,或可以被设计为完全相等。比如为了运算方便VNU和CNU通常需要用不同的数据形式(补码和原码)表示,而在实际设计中,可以选择将原码和补码间的转换安排在VNU或CNU部分来达到两者流水线长度的平衡。译码器各模块的工作时序如图1所示。

由图1的工作时序可以看到,负责译码判决的PCU只有在每次迭代的后半部分处于工作状态,而PCU模块主要由校验计算电路组成,根据第1节的讨论,我们可以利用该校验计算电路在垂直运算的空闲时间对信息位进行编码,从而提高整个编译码系统的资源利用率,达到减少硬件资源消耗量的目的。以上讨论基于归一化最小和算法,但对其

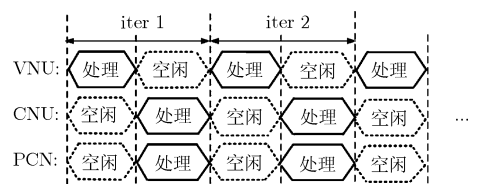


图1 LDPC 编译码器各模块的工作时序图

4.2 编译码器的码长码率兼容设计

在编译码器采用完全相同的校验矩阵进行编译码的情况下, 本文提出的复合结构能够将整个 PCU 的校验电路资源复用到编码器的码字计算上, 从而节省编码器所需的大部分硬件资源。典型的应用有中继通信和对等的点对点通信等, 尤其在卫星中继通信中, 由于低轨道中继卫星需要通过译码再生对高轨道卫星的数据进行转发, 同时还需要将自身观测得到的数据回传到地面站, 其要求的编码吞吐量比译码吞吐量高, 是该编译码器联合设计方法应用的一个理想场景。但在更多的实际应用中, 由于信道条件不对称、设备的尺寸及功耗的约束不对等以及上下行业务需求不同等原因, 通信系统的收发两端很难采用完全一致的编译码体制。下面讨论提出的复合结构在编译码矩阵的码长和码率上的兼容设计。

当编译码采用不同的校验矩阵时, 需要根据两矩阵的差异性对复合结构进行修正, 需要考虑的矩阵差异主要包括 4 个方面: (1) 矩阵对应分块的偏移量不相同; (2) 对应各分块的行重列重不同及分布不一致; (3) 矩阵的分块数不相等; (4) 子矩阵的大小不同。假设不同的两校验矩阵 H_1 和 H_2 , 其中两矩阵的公共部分标记为 A , 矩阵 H_1 中独立于 H_2 的部分标记为 B , H_2 中独立于 H_1 的部分标记为 C , 如图 3 所示。

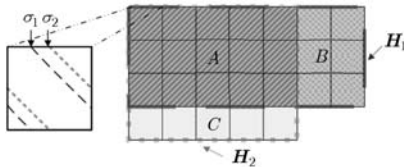


图3 编译码采用不同校验矩阵示意图

首先, 由于矩阵的偏移量不同并且各分块间的偏移量差值不可能完全一致, 这将导致对应分块的编码和译码信息无法共用相同的 RAM 和地址发生器。针对这一问题, 可以对初始化编码信息的存储地址作一定的偏移来达到数据访问的一致性。对于两矩阵相同的情况, 在数据初始化阶段, RAM_M 中对应各矩阵分块的分块 RAM 均从 0 地址开始顺序存入各编码信息, 而在译码迭代阶段, 编译码器则以公共偏移量作为初始地址顺序读写子矩阵中的信息进行计算。而对于两矩阵不相同的情况, 若编译码校验阵在同一位置的两个子矩阵的向右循环移位次数分别为 σ_1 和 σ_2 , 那么在数据初始化阶段, 该分块的 RAM 可以将编码矩阵和译码矩阵的地址差值作为初始地址对编码信息进行存储, 具体操作时

考虑到地址的循环移位性质, 还需要对分块大小进行取模运算, 即信息存储首地址应为 $(\sigma_1 - \sigma_2) \cdot \text{mod } z$, 这样在译码迭代时就可以将译码矩阵的地址偏移量 σ_2 作为公共地址来读取编译码信息。由于 RAM 共用, 改变编码信息的写入顺序显然需要同时更改初始译码外信息的存储顺序, 但是由于 RAM_M 存储块中的各外信息部分的初始值均为 0, 因此信息位存储的初始顺序的改变实际并不扰乱译码初始信息的读写顺序; 其次, 两校验阵在行重和列重上的差异也会产生一些问题, 在垂直运算阶段, 由于 VNU 和 ENC 同时工作, 相同列分块的列重不同和分布不一致将使得编译码数据的访问不一致, 造成 RAM 资源共用困难。而在水平运算阶段, 由于 CNU 和 PCU 使用的同为译码矩阵, 因此不存在矩阵差异对数据读写带来的影响。另外, 在垂直和水平运算间需要复用校验计算电路, 而行重不相同则会引起校验计算电路不通用。在具体的电路实现中, 由于存储的编译码数据是经过拆分和组合后读出或写入的, 因此并非必须相同位置的矩阵分块才能共享 RAM 存储, 实际上只要在垂直运算阶段两矩阵在同一列分块的列重相等即可组合共用相同的 RAM, 而两矩阵对应列分块的列重不同则可以通过列分块的交换来缓解, 这是由于校验阵的列交换不改变码本空间, 对于最后无法通过列分块解决的列重不相等则需要放弃 RAM 共用。对于行重不同引起的校验计算电路不通用, 可以在校验计算电路中实现较大的行重, 在处理行重较小的校验矩阵时作屏蔽处理来解决问题; 还有, 矩阵的分块数不相等会对资源复用带来较大的影响, 显然只有两个矩阵的交集部分(区域 A)能够采用资源复用设计, 其余各矩阵的独立部分(区域 B 和区域 C)则需要分开设计, 需要注意的是, 考虑编码矩阵的区域范围时只需考虑系统位部分, 因为编码时校验位部分本身就需要通过置零的方式进行屏蔽, 而对于编码矩阵中与译码矩阵不重合的部分在校验计算电路可以直接舍去校验位的加和; 最后, 对于子矩阵大小不同的情况, 虽然也可以通过资源复用来达到节约资源的目的, 但从图 1 的时序分析可以知道, 子矩阵的大小不同将引起各模块计算时间的不相等, 造成时隙等待, 从而增加控制上的复杂度, 并且会造成吞吐量的降低。而从码构造上看, 矩阵的分块大小并不会对码性能构成大的约束, 因此子矩阵大小差异的问题通过构造分块大小相等的编译码矩阵来解决显然更加合适。

5 硬件实现结果

采用提出的编译码器联合设计方法对两组不同

的LDPC码编译码器进行部分并行结构的实现,第1组编译码均采用码长为2304 bit,码率为5/6的LDPC码;第2组编译码采用不同的校验矩阵,两矩阵的分块大小相同,其中译码采用16800 bit的1/2码率LDPC码,编码采用8400 bit的3/4码率LDPC码。芯片使用Xilinx公司的XC4VLX80 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。两个译码器均采用4 bit均匀量化,译码算法采用归一化最小和算法,归一化修正因子 α 选择为0.75,最大迭代次数均为10次。图4是在AWGN信道,BPSK调制条件下,没有量化的50次和积迭代译码和量化后的归一化最小和译码的性能对比。

对于两组编译码器实现,表1分别列出了单个编码器、单个译码器和联合设计的编译码器三者的资源使用情况及系统的信息吞吐量。除了采用提出的编译码复合结构以外,联合设计的编译码器和单个编译码器实现使用的设计优化技术均相同。对于码率为5/6的第1组实现,独立设计和联合设计两种情况下编译码器均满足5 ns的布局布线约束,工作时钟频率为200 MHz。从表1的数据可见,联合设计的编译码器消耗的系统资源仅略多于单个译码器实现,其中逻辑资源多0.1%,BRAM资源多2块,而在编译码器分开设计的情况下,单独设计一个编码器需要消耗6.6%的芯片逻辑资源及14块BRAM资源,采用复合结构设计可以节省约30%的逻辑资源和10%的BRAM资源。对于编译码矩阵不同的第2组实现,独立设计时编译码器的工作时钟频率分别可达222 MHz和200 MHz,联合设计时系统工作主频稍有降低,为196 MHz。联合设计的编译码器消耗的系统逻辑资源比单个译码器多0.3%,BRAM资源多2块,与分开设计的情况相比可节省约25%的逻辑资源和12%的BRAM资源。

在本文提出的编译码器复合结构中,编码器的绝大部分资源由译码器共享得到,因而减少了编码器对硬件资源的额外需求,从而显著提高了FPGA资源的利用率。

该复合结构主要从工作时序上对编码器和译码器的各组成部分作资源复用,除了要求RAM的位宽较一般的译码器位宽宽以外,其它主要的译码模块的设计与现有的设计技术不冲突,多数的其它设计优化方法都可以应用到本结构中,该设计仅利用了译码工作时的空闲时序进行额外计算。从设计原理上看,对各模块增加资源复用设置只需要在原有电路上增加一些选通开关,对译码电路的布线复杂度影响较少,因此对整个编译码系统的工作频率及吞吐量影响也较少,具体的实现结果也表明了这一点。该资源复用方案适用于一般的编译码系统,能够达到节省系统资源而基本不降低编译码吞吐量的目的,其资源复用程度只与编码和译码采用的校验矩阵的相似度有关,而与系统具体使用的LDPC码的码长和码率等参数没有关系。从两组编译码器的实现结果可以看到,对于第1组高码率短帧长LDPC码和第2组码率较低的LDPC长码,提出的联合设计方法都能够在节省资源的同时实现与独立设计的编译码器相当的吞吐量,而第1组实现由于使用完全相同的校验矩阵进行编译码,其资源节约百分比明显要高于编译码矩阵不同的第2组实现。需要说明的是,编译码器所能够实现的吞吐量大小只与采用的LDPC码结构和具体模块的设计有关,而该资源复用结构仅提供一种不降低吞吐量的资源复用设计,正如给出的两个实现例子中,第1组实现的吞吐量明显要高于第2组,但并不表明该方法更适合于高码率短码的编译码器设计,也不表明该方法能够使高码率译码器的吞吐量做得更高,这是由于编译码系统本身选用的参数不同而导致吞吐量

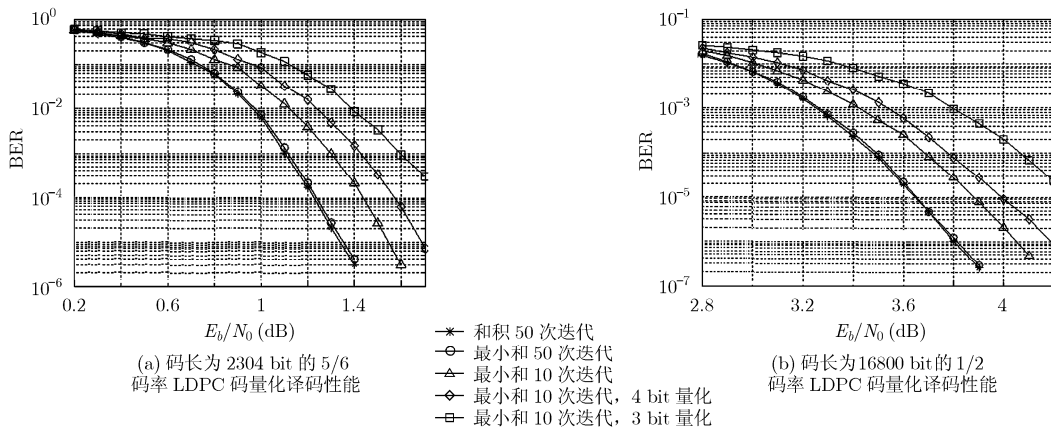


图4 两个LDPC码的量化译码性能

表1 联合设计的编译码器与单个编译码器的资源使用情况及吞吐量对比

码长(bit)与码率	结构	Slice资源	BRAM资源	主频(MHz)	吞吐量
码长: 2304 码率: 5/6	单个编码器	2366(6.6%)	14	200	1.66 Gbps
	单个译码器	5363(15.0%)	105	200	166 Mbps
	编译码器联合设计	5412(15.1%)	107	200	编码: 1.66 Gbps 译码: 166 Mbps
编码码长: 8400 编码码率: 3/4	单个编码器	2036(5.7%)	16	222	835 Mbps
	单个译码器	5978(16.6%)	101	200	100 Mbps
译码码长: 16800 译码码率: 1/2	编译码器联合设计	6047(16.9%)	103	196	编码: 735 Mbps 译码: 98 Mbps

大小的差异,第1组实现的编码并行度本身要高于第2组,且计算信息吞吐量时,高码率LDPC码的信息位比例要较低码率情况时高,因此在相同的帧处理速度下,高码率码可实现的信息吞吐量也高。综上所述,与分离式设计相比,本文提出的联合设计方法能够在不降低系统吞吐量的前提下减少硬件资源的需求量,特别适用于资源受限的硬件平台。

6 结论

本文分析了准循环双对角结构LDPC码的编译码过程,利用译码器在垂直运算时闲置的校验计算电路,通过把校验位置零后计算伴随式的方法将其复用到编码器的码字计算中,提出了一种利用校验矩阵编译码的编译码器联合设计方法。该方法使得编译码器共享相同的RAM存储块及地址产生器,有效减少了实现所需的存储资源和逻辑资源。在Xilinx XC4VLX80 FPGA上的实现结果表明,联合设计的编译码器能够进行高速编译码,且仅占用略多于单个译码器的硬件资源,提出的设计方法能够在不降低系统吞吐量的同时减少对硬件资源的需求量。

参考文献

- [1] Kou Y, Lin S, Fossorier, *et al.* 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.
- [2] 乔华, 管武, 董明科, 等. LDPC码高速译码器的设计与实现 [J]. *北京大学学报(自然科学版)*, 2008, 44(3): 347-352.
Qiao Hua, Guan Wu, Dong Ming-ke, *et al.* Design and implementation of LDPC decoder with high throughput[J]. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 2008, 44(3): 347-352.
- [3] Wang Zhong-feng and Cui Zhi-qiang. Low-complexity high-speed decoder design for quasi-cyclic LDPC codes [J]. *IEEE Transactions on Very Large Scale Integration System*, 2007, 15(1): 104-114.
- [4] Kim Min-hyuk, Park Tae-Doo, Kim Chul-Seong, *et al.* An FPGA design of low power LDPC decoder for high-speed wireless LAN [C]. *Communication Technology(ICCT)*, Nanjing: IEEE, 2010: 1460-1463.
- [5] Zhang Luo-ming, Gui Lin, Xu You-yun, *et al.* Configurable multi-rate decoder architecture for QC-LDPC codes based broadband broadcasting system [J]. *IEEE Transactions on Broadcasting*, 2008, 54(2): 226-235.
- [6] Chen Xiao-heng, Kang Jing-yu, Lin Shu, *et al.* Memory system optimization for FPGA based implementation of quasi-cyclic LDPC codes decoders [J]. *IEEE Transactions on Circuits and System*, 2011, 58(1): 98-111.
- [7] Wang Wen-jun, Wu Xiao-guang, Zhu Xiao-xuan, *et al.* A 223 Mbps FPGA implementation of (10240, 5120) irregular structured low density parity check decoder[C]. *Vehicular Technology Conference*, Calgary: IEEE, 2008: 767-771.
- [8] Dai Yong-mei, Yan Zhi-yuan, and Chen Ning. Optimal overlapped message passing decoding of quasi-cyclic LDPC Codes [J]. *IEEE Transactions on Very Large Scale Integration System*, 2008, 16(5): 565-578.
- [9] Richardson T J and Urbanke R L. Efficient encoding of low-density parity-check codes [J]. *IEEE Transactions on Information Theory*, 2001, 47(2): 638-656.
- [10] Li Zong-wang, Chen Lei, Lin Shu, *et al.* Efficient encoding of quasi-cyclic low-density parity-check codes[J]. *IEEE Transactions on Communications*, 2006, 54(1): 71-81.
- [11] Perez J M and Fernandez V. Low-cost encoding of IEEE 802.11n [J]. *Electronics Letters*, 2008, 44(4): 1411-1412.

袁瑞佳: 男, 1982年生, 博士生, 研究方向为信道编码与调制。

白宝明: 男, 1966年生, 教授, 博士生导师, 研究方向为信息与编码理论、编码调制技术和无线通信、量子通信与编码。