

高效 LDPC 译码器的优化与 FPGA 实现

薛文 余海 王建新 束锋

(南京理工大学电子工程与光电技术学院, 南京, 210094)

摘要: 针对高效 LDPC 译码器设计过程中的参数选择问题, 提出了针对 Turbo 译码消息传播 (Turbo decoding message passing, TDMP) 译码算法的离散密度进化算法。利用这种离散密度进化算法对译码算法中的校正因子及量化精度进行了优化。与传统的通过数值仿真进行优化的方法相比, 本文算法效率大大提高, 且效果显著。测试结果表明, 优化的定点化译码器与纯浮点仿真相比性能只相差 0.1 dB 左右。在译码器实现结构设计中提出了一种基于分布式 RAM 的 P 消息循环存储结构, 与传统的基于寄存器和 Benes 网络的存储器结构相比, 资源消耗明显下降。在 Xilinx 公司的 FPGA 平台上进行了硬件实现与测试, 结果表明与同类译码器相比在资源消耗和吞吐率上均有一定优势, 是一种高效的 LDPC 硬件译码器。

关键词: 准循环 LDPC 码; Turbo 译码消息传播; 离散密度进化; 分层消息处理单元; P 消息循环存储器
中图分类号: TN911.22 **文献标志码:** A

Optimization and FPGA Implementation of Efficient LDPC Decoder

Xue Wen, Yu Hai, Wang Jianxin, Shu Feng

(School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China)

Abstract: A discrete density evolution algorithm for Turbo decoding message passing (TDMP) decoding algorithm is proposed to solve the problem of fixed parameter selection in efficient LDPC decoder design. By using the discrete density evolution algorithm, the modification factors and the quantization precision in the decoding algorithm are optimized. Compared with the traditional method, the efficiency is greatly improved and the effect is significant. Experimental results show that the performance of the optimized fixed-point decoder is only about 0.1 dB worse compared with the pure floating-point simulation. In the structure design of the decoder, a P -message circular memory structure based on distributed RAM is proposed. Compared with the traditional memory structure based on register and Benes network, the resource consumption is obviously decreased. The hardware implementation and test on FPGA platform of Xilinx company show that it has some advantages in terms of resource consumption and throughput compared with the same kind of decoder, and it is an efficient LDPC hardware decoder.

Key words: quasi-cyclic LDPC code; Turbo decoding message passing (TDMP); discrete density evolution; layer message processing unit (LMU); P -message cyclic memory

引 言

LDPC 码是一种接近香农极限的好码,广泛应用于卫星通信、深空通信及无线通信领域。欧洲第 2 代卫星电视标准^[1]就采用了 LDPC 码作为前向纠错码。国际空间数据系统咨询委员会^[2]也在其近地及深空通信的标准选择 LDPC 码作为纠错编码之一。LDPC 码是由 Gallager 于 1963 年提出的^[3],起初没有得到重视,直到 20 世纪 90 年代末期 Mackay 等“再发现”了 LDPC 码,并提出了和积算(Sum-product algorithm, SPA)^[4],使得 LDPC 码真正成为了一种实用的接近香农极限的好码。和积算法从纠错性能角度来说是一种最优的译码算法,但是考虑到硬件实现时复杂度还是较大。文献[5]中针对译码算法中的校验节点更新算法,提出了一种简化算法,用比较运算代替了 SPA 中的非线性运算,叫做最小和算法(Min-sum algorithm, MSA)。这种算法不仅降低了算法复杂度,同时还是一种不需要信道估计的算法。MSA 的纠错性能相比 SPA 有比较明显的下降。针对这个问题,文献[6,7]中给出了两种修正算法,分别叫做规范化最小和算法(Normalized min-sum algorithm, NMSA)和偏移最小和算法(Offset min-sum algorithm, OMSA)。这两种修正算法对简化过的校验节点更新算法进行了修正,使其计算结果更接近于 SPA 中的结果,从而在略微增加了计算复杂度的前提下大大改善了译码性能,使得修正后的 MSA 能够很接近 SPA 的性能水平。

除此之外,通过对 LDPC 码译码迭代方式的改进也可以影响译码性能。Mansour 等借鉴 Turbo 码的译码方式,提出了另一种迭代译码方式,叫做 Turbo 译码消息传播(Turbo decoding message passing, TDMP)^[8,9]方式,也被称为分层译码(Layer decoding)。研究表明,达到同样的译码性能,TDMP 所需的迭代次数要远小于两相消息传播(Two phase message passing, TPMP)。国内针对准循环 LDPC 码硬件实现工作也较多,文献[10]提出了一种基于 IR-UWB 系统的高速准循环 LDPC 编解码器,其设计思路值得参考。评价一个 LDPC 译码器的优劣也并没有一个简单的标准。正如在文献[11]中提到的,译码器的好坏要从资源消耗、吞吐率和纠错能力这 3 个方面来进行综合考量。

1 译码算法选择

本文的设计需求是设计一个高效的 FPGA 实现硬件译码器。目前适合高效实现的 LDPC 码主要是一些准循环的 Block-LDPC 码,本文选择了 802.16e(WiMAX)标准中定义的准循环 LDPC 码作为实现目标。

1.1 校验节点更新算法

对几种常用校验节点更新算法及其改进算法,SPA,MSA,NMSA 和 OMSA,利用密度进化原理进行了性能估计。图 1 是上述几种算法的估计性能比较图。实际通信中接收信号的信噪比是未知的,考虑估计误差,以上几种算法性能估计如图 2 所示。在不考虑信噪比估计误差时 SPA 性能最佳,MSA 最差,NMSA 与 OMSA 基本相当,考虑信噪比估计误差,SPA 性能恶化,而 MSA,NMSA 和 OMSA 的性能几乎不变。其原因主要是校验节点更新算法的差异性,MSA 中校验节点更新算法的表达式为

$$L(r_{ji}) = \prod_{i \in R(j)} \text{sgn}(L(q_{ij})) \cdot \min_{i \in R(j)} (|L(q_{ij})|) \quad (1)$$

其中主要运算为求最小值的运算,比较运算的结果对于输入数据的绝对大小并不敏感。进一步比较 NMSA 和 OMSA,发现 NMSA 更加适合于硬件实现。这两种算法对校验节点更新结果的修正方式为

$$L(r_{ji}) = \frac{1}{\alpha} L(r_{ji}) \quad (2)$$

$$L(r_{ji}) = \text{sgn}(L(r_{ji})) \cdot \max(|L(r_{ji})| - \beta, 0) \quad (3)$$

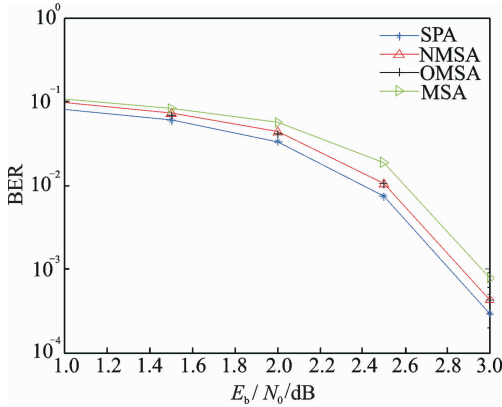


图 1 几种典型算法的性能估计及比较

Fig. 1 Performance estimation and comparison of typical algorithm

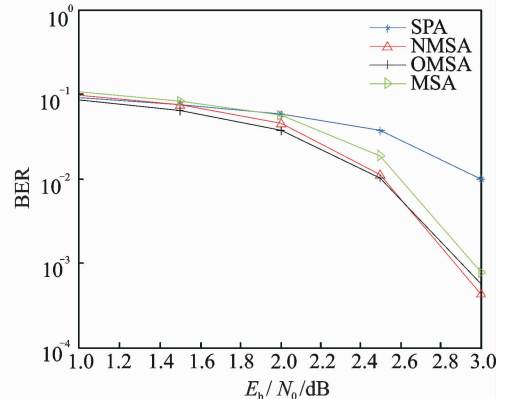


图 2 考虑噪声估计误差后的性能估计结果

Fig. 2 Performance estimation result considering noise estimation errors

可以看出,式(2)中的运算效果与校验节点更新值的绝对大小完全无关,但是式(3)中的偏移因子 β 如果设定好后,其被修正值一定要与之对应,如果被修正值的绝对大小发生改变,则修正效果将会改变。因此从这个意义上来说,NMSA虽然消耗的资源略多,但是NMSA显然更加实用。

1.2 迭代译码方式选择

最初 Mackay 提出的译码算法中迭代方式都采用了 TPMP 方式即迭代过程分为水平迭代和垂直迭代两个基本步骤,TDMP 方式的译码流程和 TPMP 方式有很大不同。这种译码迭代方式主要适用于准循环 LDPC 码。准循环 LDPC 码一般都是 Block-LDPC 码,其校验矩阵的是一个分块矩阵。

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \cdots & \mathbf{P}_{1,N_b} \\ \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \cdots & \mathbf{P}_{2,N_b} \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{P}_{M_s,1} & \mathbf{P}_{M_s,2} & \cdots & \mathbf{P}_{M_s,N_b} \end{bmatrix} \quad (4)$$

式中 $\mathbf{P}_{i,j}$ 是 $z \times z$ 的子矩阵,一部分为全零阵,另一部分为单位置换阵,单位置换阵是指单位阵进行循环移位所得到的矩阵,其特点是行重和列重均为1,这个特性很适用于 TDMP 译码。一般在表示校验矩阵时,常用特定的数字替代式(4)中的子矩阵,这样得到的矩阵称为母矩阵(Base matrix),可用来表示结构相同但码长不同的一类准循环 LDPC 码。母矩阵中每行所对应的校验矩阵称为一个块行,每列对应的校验矩阵称为一个块列。为了叙述方便先定义一些变量。定义 \mathbf{P}_i 为一个 n 维数组,其中 n 表示 LDPC 码的码长,用于存放变量节点的后验概率。定义 \mathbf{Q}_i 为一个 n 维数组,用于存放传递给校验节点的消息。定义 \mathbf{R}_{ij} 为一个 $M_b \times n$ 的数组,用于存放各层校验节点更新的结果。

(1) 初始化

$$\mathbf{P}_i = \ln \frac{\Pr\{x_i = 1 \mid y_i\}}{\Pr\{x_i = 0 \mid y_i\}} \quad (5)$$

(2) 子迭代

第 p 次迭代中,第 m 次子迭代中的第 s 个节点更新的表达式为

$$\mathbf{Q}_{(1;N_b)}(s) = \mathbf{P}_{(1;N_b)}(s) - \mathbf{R}_{p,(1;N_b)}(s) \quad (6)$$

$$\mathbf{R}_{p,(1;N_b)}(s) = f(\mathbf{Q}_{(1;N_b)}(s)) \quad (7)$$

$$P_{(1:N_b)}(s) = Q_{(1:N_b)}(s) + R_{(1:N_b)}(s) \quad (8)$$

其中函数 f 表示校验节点更新的计算公式,可以选择各种不同的算法。对于每个子迭代要完成 \approx 个节点的更新,每次迭代要完成 M_b 次子迭代。

(3) 判决与停止迭代条件

每完成一次迭代可以对后验概率 P_i 进行判决,判决结果若满足校验矩阵则译码成功,否则返回步骤(2),直到达到最大迭代次数退出。

利用离散密度进化的思想,给出了准循环 LDPC 码的 TDMP 离散密度进化算法。图 3 是各变量节点 P 消息的概率质量函数随迭代次数增加的进化过程。同样的迭代次数下,TDMP 译码的密度进化过程明显更快,其相应的误比特率也较 TPMP 更低,所以实现高速译码器,应该选择 TDMP 方式。

2 译码算法相关参数的优化

目前大多数译码器采用定点实现,一般经定点化后性能会有不同程度的下降。译码算法进行定点化时需要考虑输入的初始化信息的量化精度和迭代译码过程中消息在进行运算时的量化精度。

利用 TDMP 的离散密度进化算法先对输入初始消息的量化精度进行优化。分别采用不同的输入量化精度,并采用离散密度进化算法对译码性能进行估计。图 4 是输入消息不同量化精度下利用密度进化得到的误比特率性能估计结果。量化精度为 5 bit 时最差,而量化精度为 6 bit 和 7 bit 时在较低信噪比下性能相当,但是高信噪比时量化精度高的性能反而更差。从图 4 中可以看出, P 消息值随迭代次数增加而增加,因此会出现限幅效应,相比之下,输入消息量化为 6 bit 时初始值较小因此限幅效应影响较小。将输入消息量化为 7 bit 时,中间量化位宽调整为 8 bit,再次使用离散密度进化算法进行性能估计,结果如图 5 所示。此时输入位宽为 7 bit 的性能略好于输入为 6 bit,但是差距很小。不妨将输入消息的量化精度暂定为 6 bit。改变迭代中间过程中消息的位宽,估计性能,结果如图 6 所示。当位宽为 6 bit 时性能较差,位宽采用 7 bit 和 8 bit 时性能几乎一致。综上所述,输入消息的量化位宽选择设定为 6 bit,而迭代中间过程消息的量化位宽选择为 7 bit 是比较合理的设定。利用离散密度进化对硬件实现时的具体参数进行优化设计,取代原有的靠仿真确定具体实现参数的方法,大大提高了效率,且最终实现结果也证明这种优化方法达到了较好的效果。

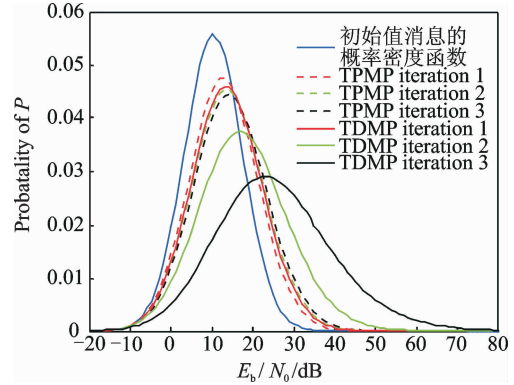


图 3 TDMP 与 TPMP 变量节点后验概率的概率质量函数的进化

Fig. 3 Evolution of variable node posterior probability mass function for TDMP and TPMP

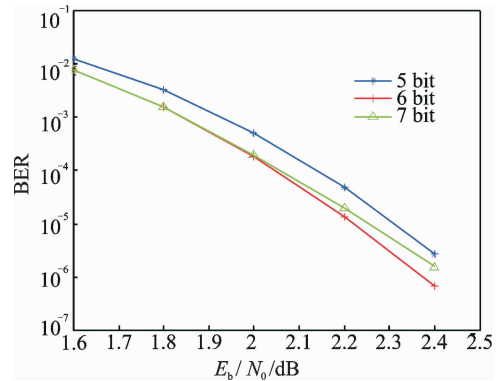


图 4 不同输入量化精度下的误比特率性能估计

Fig. 4 Bit error rate performance estimation with different input quantization precision

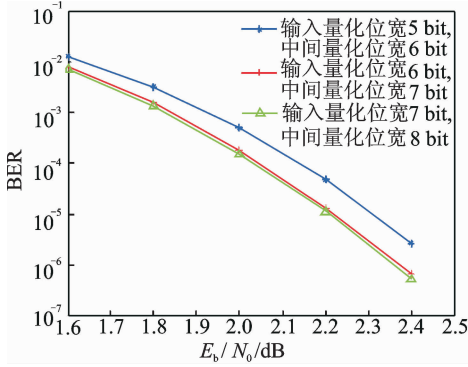


图 5 不同输入量化位宽和中间量化位宽下的性能比较

Fig. 5 Performance comparison of different input quantization bit width and intermediate quantization bit width

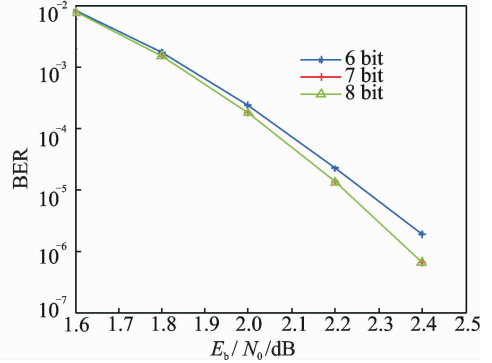


图 6 不同中间消息位宽下的性能比较

Fig. 6 Performance comparison of different intermediate message widths

3 译码器的 FPGA 实现

高效译码器的 FPGA 实现过程中有几个关键点:高效的迭代译码结构、设计结构更加优化的译码器功能模块、合理的存储结构以及合理的流水线结构。其中节点更新单元及存储结构的设计是难点,本文采用了基于 TS 结构的高效节点更新单元,并提出了一种适合码字结构的 P 消息存储单元结构,较好解决了这两个难点问题。

3.1 译码器总体框架

1.2 节中给出了 TDMP 的译码具体流程,按照这个实现流程,译码器的实现框图如图 7 所示。译码器可以分成以下几个主要部分:

(1) 分层消息处理单元:

分层消息处理单元 (Layer message processing, LMU) 是主要的运算模块,主要完成变量节点后验概率值 (P 消息) 和校验节点消息 (R 消息) 的更新运算。

(2) P 消息循环存储器

这是本文提出的一种重要的实现结构,其功能是高效存储上一次子迭代得到的 P 消息,并且能高效地读出提供给下一次子迭代,相当于一个交织器。

(3) R 消息存储器

该模块用于存储每一层更新的 R 消息。由于每次子迭代时都需要利用上一次迭代中该层对应的 R 消息,因

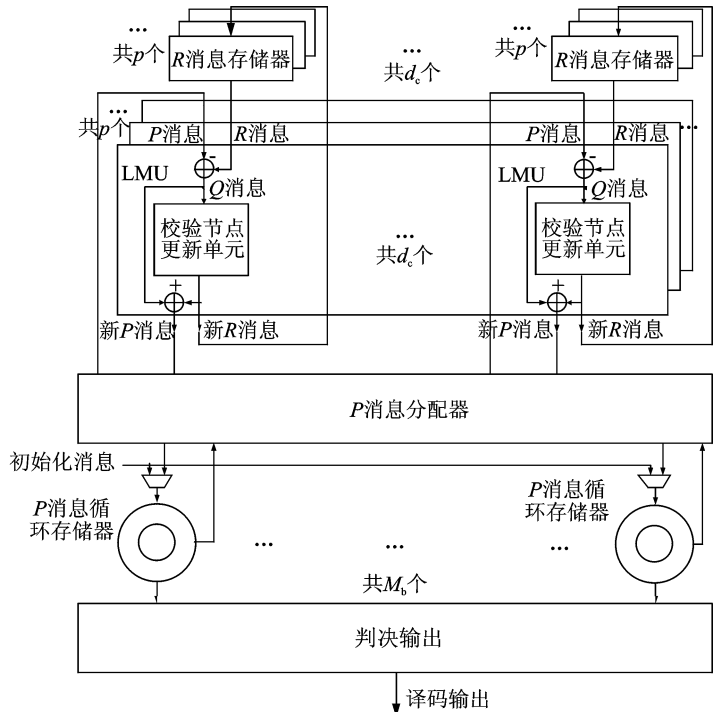


图 7 LDPC 译码器实现框图

Fig. 7 LDPC decoder implementation block diagram

此在每次迭代中每一层的 R 消息都必须保留以备下次迭代时调用。

(4) P 消息分配器

由于校验矩阵中每层中都存在一些全零阵,因此每次进行分层消息处理时只是将部分 P 消息存储器(非零子矩阵所对应的 P 存储器)中的消息传递给分层消息处理单元,所以需要分配 P 消息。

(5) 判决输出模块

对 P 消息进行判决,并通过校验决定是否停止迭代。

3.2 分层消息处理单元

分层消息处理单元在实现时先完成式(1)中的运算,然后再根据式(2)进行校正得到最终计算结果。式(1)中运算的硬件实现过程为

$$L(r_{ji}) = \begin{cases} \prod_{i \in R(j)} \text{sgn}(L(q_{ij})) \cdot \text{Min}(|L(q_{ij})|) & i \neq \text{Minpos} \\ \prod_{i \in R(j)} \text{sgn}(L(q_{ij})) \cdot \text{Min2}(|L(q_{ij})|) & i = \text{Minpos} \end{cases} \quad (9)$$

式中:Minpos 表示最小值所对应的消息标号,Min2 表示求序列中次小值的函数。文献[12]中对求解一个序列的最小值和次小值的问题进行了详尽论述,并给出了一种称为树形结构(Tree structure, TS)的高效实现结构,如图 8 所示。

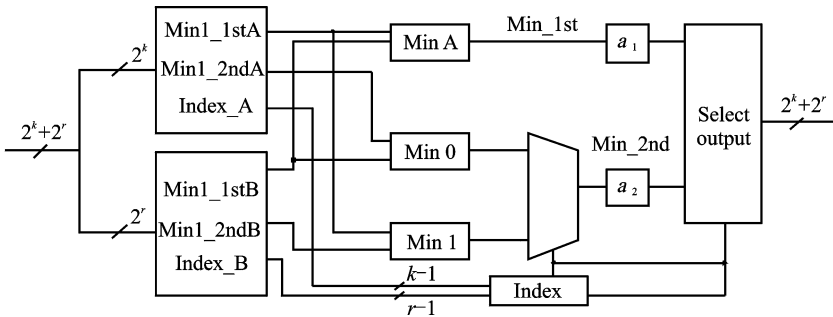


图 8 校验节点更新单元的实现结构

Fig. 8 Implementation structure of check node renew unit

3.3 P 消息循环存储器

该模块是译码器中的核心模块。该循环存储器将每个变量节点对应的 P 消息存于其中。根据校验矩阵的结构分为 N_b 块,每块的深度为 z 。译码时先将所有的初始化消息写入存储器中,由于准循环码的每个非零矩阵都是单位置换阵,因此每个存储器都是循环存储结构。每次子迭代选择该层中非零子矩阵所对应的存储块,从这些存储块中按照该子矩阵的偏移量确定起始位置,并以并行度 p 从每个存储块中一次同时读出 p 个消息,读出的消息经过分层消息处理单元进行更新后以并行度 p 写回到循环缓冲区中的原来存储位置。这个循环存储器关键就是要实现从任意一个起始地址开始一次同时读出或者写入 p 个消息,其结构如图 9 所示。

文献[13]采用了图 10 所示结构来实现同样的功能。这种结构依靠 Benes 网络来实现数据交织,即实现从任意起始位置同时读出 p 个消息,而存储消息则是采用寄存器存储。采用这种结构在码长较短时还是比较高效的,但是随着码长的增加其资源消耗会越来越大。采用 RAM 存储消息,虽然由于实现结构的问题,不适合使用大容量的块状随机存储器(Block RAM, BRAM),但是如果使用分布式 RAM

实现存储器,相比采用触发器会大大节省资源消耗。Benes 网络的实现复杂度也相当高,即使像文献[13]中那样对 Benes 网络进行了简化,当规模较大时其资源消耗依然很可观。因此本文提出的这种结构更加高效,更适合于 FPGA 实现,且在码长增加时,所消耗的资源基本与码长呈线性关系。

3.4 R 消息循环存储器

由于 R 消息更新时只对非零矩阵进行,对于每个层来说,全零矩阵对应的 R 消息永远是零,因此在存储时,每一层只需要存储非零子矩阵对应的 R 消息即可。 R 消息存储器的具体结构如图 11 所示。考虑到译码器实现的并行度 p ,要求 R 消息存储器每次能够同时并行输出 p 个消息,当 R 消息更新计算完成后也能够同时并行输入 p 个消息,因此可以将存储器的位宽设置为 w_p 位,其中 w 表示 R 消息的量化精度。

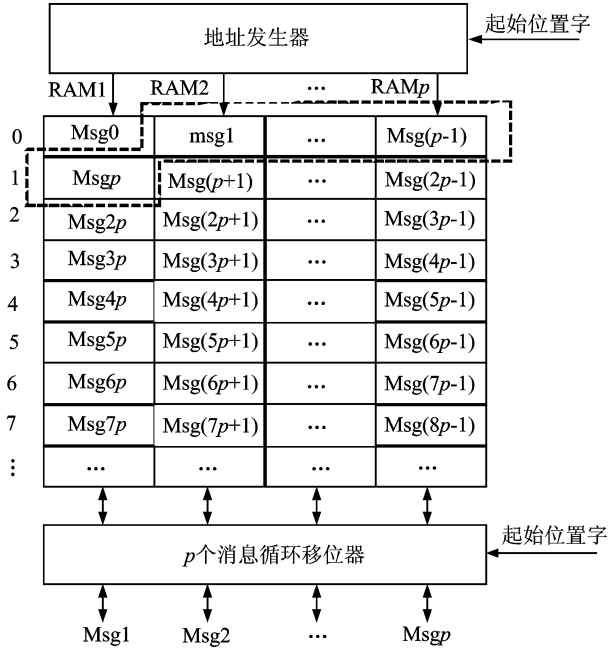


图 9 P 消息循环存储器结构图

Fig. 9 P message cycle memory structure

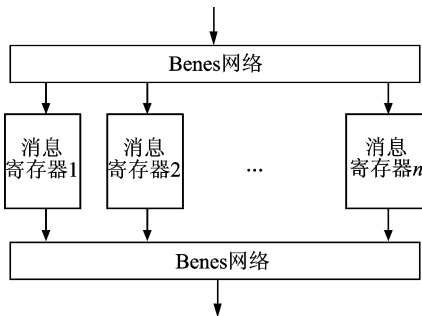


图 10 基于 Benes 网络的 P 消息存储器

Fig. 10 P message memory based on Benes network

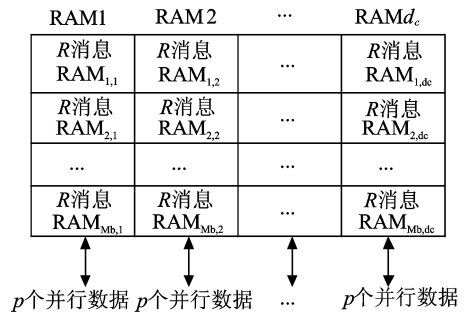


图 11 R 消息存储器结构

Fig. 11 R message memory structure

3.5 校验模块

常规的校验方法是将判决结果代入校验矩阵中计算校验值。文献[13]提出了一种更为实用的方法,即对比本次迭代的译码结果和上次迭代的译码结果,如果发现两次相同则认为译码已经完成,否则继续迭代直到达到最大迭代次数。这种方法是基于译码算法的收敛性,即当译码正确后再次进行迭代,其结果也应该是正确的,这样电路实现的复杂性就降低了。

3.6 译码流程控制

译码整个流程可以用图 12 所示有限状态机进行控制,说明如下:

(1) 消息初始化

译码器一开始处在空闲状态,当检测到有效数据到来时进入初始化变量节点的状态,译码器需要等

待所有节点初始化完成后才可以开始译码。

(2) 分层消息处理

分层消息处理过程非常复杂,必须采用流水线结构才能工作在较高的速率下。流水线分配如下:读出原有的 P 消息并计算 Q 消息的过程分配一级流水线;中间进行节点更新运算过程分配三级流水线,将更新的 P 消息和 R 消息写入存储器也分配一级流水线。每一次子迭代其流水线结构设计如图 13 所示。由于采用了流水线结构,因此完成一次子迭代共需 $5 + \frac{z}{p} - 1$ 个时钟,而完成一次迭代,需要进行 M_b 次子迭代。进一步研究迭代过程,发现由于采用了流水线结构,在每次子迭代开始后 $\frac{z}{p}$ 个时钟后,本层消息已经全部输入完毕,此时分层消息处理单元已经空闲,可以接收下一层的输入信息。但是由于采用了流水线结构还需要等待 4 个时钟,最后输入的消息才能够处理完毕。事实上,如果相邻两层之间的非零子矩阵并不在同一个块列中,则可立即进行下一层译码操作,但是如果相邻两层存在位于同一块列的非零子矩阵,则有可能造成流水线冲突。解决冲突的办法是将下一层中消息处理的顺序进行调整。如果通过调整层内操作顺序可以解决流水线冲突,则还可以进一步将存储器读写步骤进行旁路^[13]。如图 14 所示。根据以上分析,在采用流水线结构后完成一次子迭代需要 $5 + \frac{z}{p} - 1$ 个时钟,而如果采用读写旁路技术则时钟数进一步减少到 z/p 个时钟。

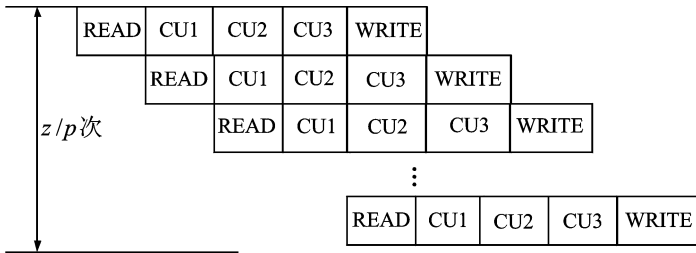


图 13 子迭代过程的流水线结构

Fig. 13 Pipeline structure of sub-iterative process

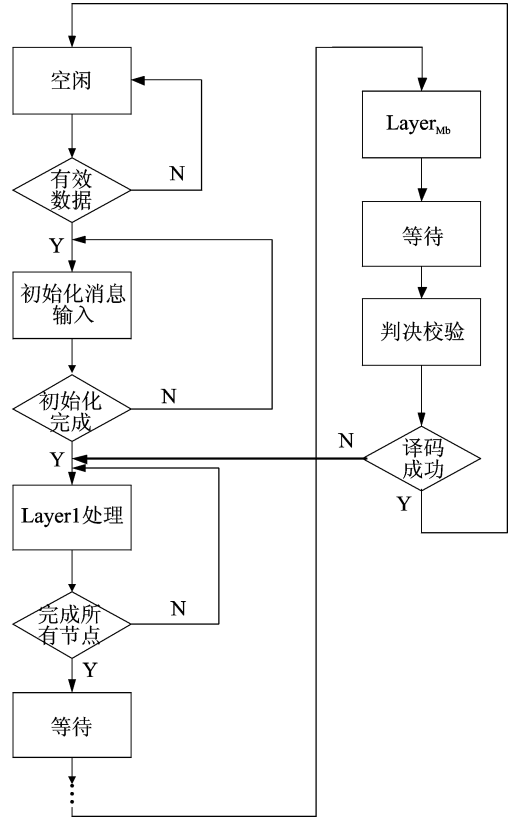


图 12 译码器的有限状态机

Fig. 12 Finite state machines for decoder

如果只考虑主要迭代过程所消耗的时钟个数,则可以大概估计出译码器的吞吐率为

$$\text{Rate}_{th} = (k / ((z/p) M_b)) f_{clk} \tag{10}$$

式中 k 表示码字中信息位的长度, f_{clk} 表示 FPGA 工作时钟频率。

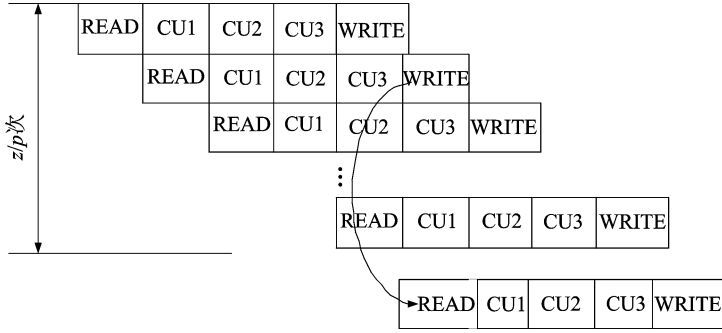


图 14 读写旁路时序图

Fig. 14 Read and write bypass timing diagram

3.7 FPGA 实现结果与比较

根据设计思路,在 Xilinx 公司的 XC6VLX130T FPGA 平台上对各个主要模块进行了实现,资源消耗见表 1,基保 FF 表示触发器,Slice 为 FPGA 的一种组成单元,LUT 表示查找表。

表 1 各主要模块资源消耗

Tab. 1 Main module resource consumption

模块	FF	Slice	LUT	BRAM/bit
10 : 2 比较器	16	66	162	0
LUM	220	205	518	0
PRAM	9	113	295	0

图 15 是分层消息处理单元在 FPGA 平台上利用 Chipscope 软件抓取的输入输出数据的实际数。利用分布式 RAM 和读写控制逻辑在 FPGA 上实现了 P 消息循环存储器,存储容量为 64 Byte。可以在循环存储器中任意起始位置开始同时读写连续 8 个消息。其中每个消息位宽为 7 bit。单个 P 消息循环存储器在 FPGA 平台上。

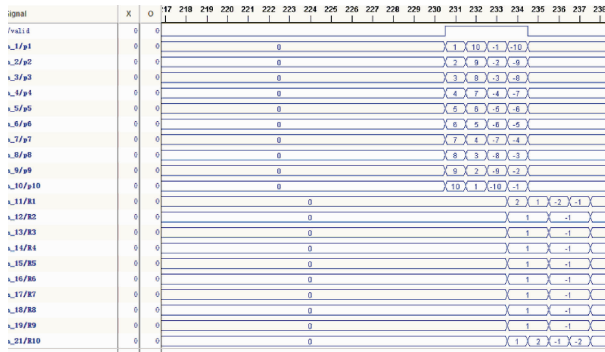


图 15 分层消息处理单元在 FPGA 实现平台上验证波形

Fig. 15 Verify waveform on FPGA platform of layer message processing unit

实现的资源消耗见表 1。图 16 是 P 消息循环存储器在 FPGA 平台上利用 Chipscope 软件抓取的实际数据流。由于并行度 $p=8$, 消息位宽为 7 bit, 因此每次需要同时写入或读出分层消息处理模块的 R 消息位宽为 56 bit。分层消息处理模块 R 消息需要每层都要保存, RAM 的深度为 $z/p \times M_b$, 考虑到整个译码器实现中其他部分并未用到 BRAM, 因此这样采用 BRAM 实现 R 存储器可以节省其他资源的消耗。表 2 是整个译码器的总资源消耗, 以及参考文献中译码器的资源消耗, 以及参考文献中译码器的资源消耗。选择几个 FPGA 实现的译码器, 校验矩阵都是 WiMAX 协议定义的类型。比较结果表明, 本文中实现的译码器消耗的触发器资源明显更少, 而吞吐率在参考文献中的几个译码器中较高。综合后最大工作时钟可以达到 98.7 MHz, 根据计算在这个时钟速率下对 2/3 码率可以达到 157 Mbps 的吞吐率, 对 3/4 码率译码可以达到 236 Mbps 的吞吐率; 与文献[13, 14]相比具有较明显优势; 与文献[15]相比, 由于该译码器适应 Wimax 的所有码率, 如果与同码率情况相比本文的译码器在吞吐率上也具有一定优势。将设计的硬件译码器下载到 FPGA 平台上进行测试, 将编码后的信号经过加噪量化后逐帧送入译码器进行译码, 并测试其误比特率性能, 其实测性能曲线如图 17 所示。结果表明 LDPC 硬件译码器的性能与浮点数值仿真的结果相差不大, 在中低信噪比下译码性能只比浮点仿真时差 0.1 dB 左右。该译码器应用于某实际数字通信系统中, 性能表现良好, 与未编码的情况进行比较, 其编码增益在误码率为 10^{-5} 时大于 6 dB。

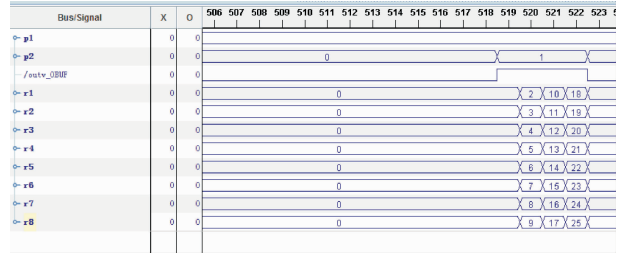


图 16 P 消息循环存储器在 FPGA 平台上的验证波形
Fig. 16 Verify waveform on FPGA platform of P message cycle memory

表 2 是整个译码器的总资源消耗, 以及参考文献中译码器的资源消耗。选择几个 FPGA 实现的译码器, 校验矩阵都是 WiMAX 协议定义的类型。比较结果表明, 本文中实现的译码器消耗的触发器资源明显更少, 而吞吐率在参考文献中的几个译码器中较高。综合后最大工作时钟可以达到 98.7 MHz, 根据计算在这个时钟速率下对 2/3 码率可以达到 157 Mbps 的吞吐率, 对 3/4 码率译码可以达到 236 Mbps 的吞吐率; 与文献[13, 14]相比具有较明显优势; 与文献[15]相比, 由于该译码器适应 Wimax 的所有码率, 如果与同码率情况相比本文的译码器在吞吐率上也具有一定优势。将设计的硬件译码器下载到 FPGA 平台上进行测试, 将编码后的信号经过加噪量化后逐帧送入译码器进行译码, 并测试其误比特率性能, 其实测性能曲线如图 17 所示。结果表明 LDPC 硬件译码器的性能与浮点数值仿真的结果相差不大, 在中低信噪比下译码性能只比浮点仿真时差 0.1 dB 左右。该译码器应用于某实际数字通信系统中, 性能表现良好, 与未编码的情况进行比较, 其编码增益在误码率为 10^{-5} 时大于 6 dB。

表 2 译码器总资源消耗比较

Tab. 2 Comparison of resource consumption in decoders

方法	Edges	FF	Slice	LUT	BRAM	吞吐率/MHz
本文	5 120	2 568	5 812	14 988	80 460/8	157
文献[12]	1 800	9 574		12 840	19 968	32
文献[14]	3 460		5 200			38
文献[15]	7 680	6 330		11 028	60 288/100	61~278

4 结束语

本文首先采用离散密度进化原理对译码器设计过程中的算法选择及实现参数进行了优化, 相比直接采用数值仿真进行优化的方法, 极大地提高了效率, 且经过优化的译码器与纯浮点仿真相比译码性能仅仅相差 0.1 dB, 由于采用了经过优化的实现参数, 使译码器在进行硬件实现时有更高的实现效率。在具体实现中, 本文提出了一种基于分布式 RAM 和读写控制的循环 P 消息存储器, 相比传统的采用 Benes 网络和寄存器的实现结构大大减小了逻辑资源的消耗, 由于采用 Xilinx 公司的 FPGA 特有的分布式 RAM 使得 RAM 的使用效率大大提高。针对流水线结构进行了优化, 解决了流水线冲突问题, 并通过读写旁路技术进一步缩短了迭代时间, 降低了功耗。最终达到了 157 MHz 的较高吞吐率及较好的译码性

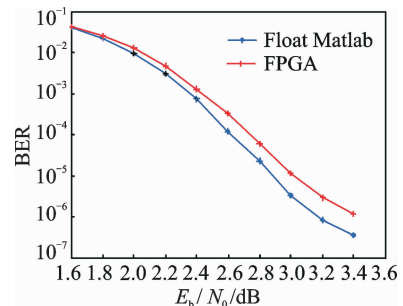


图 17 实测 FPGA 误比特率与浮点仿真误比特率比较
Fig. 17 Comparison of bit error rate between floating-point simulation and FPGA decoder

能。测试性能表明与几种典型的同类型译码器相比,本文提出的硬件译码器结构和实现方案在消耗较少硬件资源的前提下可以获得更高的吞吐率和更接近浮点仿真的性能,较好地解决了译码器硬件实现过程中在性能资源和吞吐率之间的矛盾,是一种高效的 LDPC 码硬件译码器。

参考文献:

- [1] European Telecommunications Standards Institute (ETSI). Digital video broadcasting (DVB) EN 302307-1[S]. France: ETSI, 2014.
- [2] LAN/MAN Standards Committee. IEEE draft amendment standard for local and metropolitan area networks-Part 16: Air interface for fixed and mobile broadband wireless access systems-advanced air interface IEEE P862. 16 m/DT[S]. New York: IEEE, 2010.
- [3] Gallager R G. Low density parity check codes[D]. Cambridge: MIT Press, 1963.
- [4] Mackay D J C. Good error correcting codes based on very sparse matrices[J]. IEEE Transaction on Information Theory, 1999, 45(2): 399-431.
- [5] Fossorier M, Mihaljevic M, Imai H. Reduce complexity iterative decoding of low density parity check codes based on belief propagation[J]. IEEE Transaction on Communications, 1999, 47: 673-680.
- [6] Chen J, Fossorier M C. Near optimum universal belief propagation based decoding of low-density parity check codes [J]. IEEE Transactions on Communication, 2002, 50: 406, 414.
- [7] Chen J, Dholakia A, Eleftheriou E. Reduced complexity decoding of LDPC codes[J]. IEEE Transactions on Communications, 2005, 53(8): 1288, 1299.
- [8] Mansour M M, Shanbhag N R. Turbo decoder architectures for low-density parity-check codes[C]//Proc IEEE GLOBE-COM. Taipei, China: [s. n.], 2002: 1383-1388.
- [9] Mansour M M, Shanbhag N R. Architecture-aware low-density parity-check codes[C]//Circuits Proceedings of the 2003 International Symposium on. Bangkok, Thailand: IEEE, 2003: 57-60.
- [10] 曾辉, 黄鲁, 杨灿美. 基于 IR-UWB 系统的高速准循环 LDPC 编解码器设计[J]. 数据采集与处理, 2015, 30(3): 599-605.
Zeng Hui, Huang Lu, Yang Canmei. High-rate quasi-cyclic LDPC design for IR-UWB system[J]. Journal of Data Acquisition and Processing, 2015, 30(3): 599-605.
- [11] Hailes P, Xu L, Maunder R G. A survey of FPGA-based LDPC decoders[J]. IEEE Communications Surveys & Tutorials, 2016, 18(2): 1098-1122.
- [12] 徐勤. 多媒体传感网络中 LDPC 译码电路的研究与设计[D]. 南京: 东南大学, 2012.
Xu Meng. Research and design of LDPC decoder in wireless multi media sensor networks[D]. Nanjing: Southeast University, 2012.
- [13] Wey C L, Shieh M D, Lin S Y. Algorithms of finding the first two minimum values and their hardware implementation[J]. IEEE Transaction on Circuits and System I, 2008, 55(11): 35-37.
- [14] Wang Y L, Ueng Y L, Peng C L, et al. Processing-task arrangement for a low complexity full-mode WiMAX LDPC codec [J]. IEEE Transactions on Circuits and Systems I, 2011, 58(2): 415-428.
- [15] Gunnam K K, Choi G S, Yeary M B. VLSI architectures for layered decoding for irregular LDPC codes of wiMax[C]//IEEE International Conference on Communications. [S. l.]: IEEE, 2007: 4542-4547.

作者简介:



薛文(1979-),男,讲师,研究方向:信道编解码,软件无线电及其硬件实现, E-mail: xuewen@njjust.edu.cn。



余海(1979-),男,博士研究生,研究方向:无线通信与移动网络。



王建新(1963-),男,教授,博士生导师,研究方向:通信信号处理,雷达信号处理和软件无线电技术。



束锋(1973-),男,研究员,博士生导师,研究方向:无线通信与移动网络、雷达信号处理和无线定位。