

自适应MET-LDPC编码方案及FPGA实现

何元权¹, 谢东福²

(1. 重庆邮电大学 移动通信重点实验室, 重庆 400065; 2. 厦门大学 宽带无线通信实验室, 福建 厦门 361005)

【摘要】 介绍自适应多边类型低密度奇偶校验码(MET-LDPC)多码率编码方案,在不改变编码器生成矩阵的基础上采用规则删余的方法构造多码率编码并硬件实现。在编码器硬件实现中,采用寄存器组比特交织方案,减少资源占用,提高了处理能力。在占用较低硬件资源的情况下,编码器吞吐量为100 Mbit/s,满足高速编码的要求。

【关键词】 多边类型低密度奇偶校验码;多码率编码;交织;现场可编程门阵列

【中图分类号】 TN919.81

【文献标识码】 A

Encoding Scheme Design and FPGA Implementation of Adaptive MET-LDPC

HE Yuanquan¹, XIE Dongfu²

(1. Key Lab of Mobile Communication in Chongqing, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

2. Key Lab of Broadband Wireless Communication, Xiamen University, Fujian Xiamen 361005, China)

【Abstract】 The adaptive encoding scheme of MET-LDPC is proposed that can encode information bits in any code rate without amending generated matrix by rule puncturing. Meanwhile, in the process of encoder's implementation, a new bit interleaved scheme adopted, which can reduce resources occupation and improve the work efficiency. The encoder's throughput is 100 Mbit/s with little hardware device and meets the demand of high speed encoding application.

【Key words】 MET-LDPC; multi-rate encoding; interleave; FPGA

0 引言

多边类型低密奇偶校验码(MET-LDPC)码是在2004年由Tom Richardson首次提出^[1],因为码型结构包容近年来涌现出的众多新码型,如RA码、IRA码和CT码等,在各种短帧和码率下都有很好的纠错性能,错误地板很低,同时兼有编、译码实现复杂度低的特点。

在无线传输系统中,通信系统为了保证服务质量,提高吞吐量,需要多码率编码来进行传输信息比特的纠错编码。现有的基于LDPC多码率编码方式研究中,常见的编码码率调整方式是设计几种不同码率的LDPC编码方案^[2-4],尽管保证了多码率的LDPC在接收端均有良好的译码性能,但各自对应的生成矩阵不一样,在硬件实现中增加了发送方和接收方的硬件资源开销。当编码方案越多,需要同时存储的生成矩阵越多,这对于资源的占用是非常巨大的。为此,在LDPC的编码实现中,结合硬件资源占用,提出了新的自适应编码方式——删余^[5-7]。编码首先以低码率进行编码,当需要使用较高码率进行编码时,则对编码产生的校验比特进行删余,传输未被删除的比特,使得码率动态发生变化。

笔者在删余的基础上,设计了多边类型LDPC多码率编码方案,并硬件实现,与传统方案相比具有如下优点:1) 编码方式简单,仅需要删余产生的校验比特来构

造不同码率;2) 资源占用少,各种码率均采用唯一编码矩阵进行编码,在FPGA实现中,仅需要存储这一编码方案;3) 在不占用RAM资源的情况下,采用一种新的交织实现方法,节省了系统资源。

1 MET-LDPC 码

MET-LDPC码是近年来提出的一种更为广义的LDPC码^[1],具有优秀的性能,极低的错误地板。MET-LDPC码的形式包括很多结构。近年来涌现的众多类LDPC码,如不规则LDPC码、规则/不规则重复积累码(RA/IRA码)、级联数码(CT码)、准循环LDPC码(QC-LDPC码)等,都可以认为是MET-LDPC码的特例。

多边类型的LDPC码变量点和校验点表示为

$$v(r, x) = \sum v_{b,d} r^b x^d, \mu(x) = \sum \mu_d x^d \quad (1)$$

式中: $d = d_1, d_2, \dots, d_n$,表示多边类型的度,其中 n 为边类型的数量; $x = x_1, x_2, \dots, x_n$,表示变量点; $x^d = \prod_{i=1}^n x_i^{d_i}$; $b = b_1, b_2, \dots, b_t$,表示接收的度; $r = r_1, r_2, \dots, r_t$,表示与变量点相应的接收分配的度,其中 t 为不同接收分配类型的数量; $r^b = \prod_{i=1}^t x_i^{b_i}$ 。假设多边类型的LDPC码的码长为 N ,因此,变量点类型的数量为 $v_{(b,d)}N$,同理校验点类型的数量为 $\mu_d N$ 。

基金项目:国家自然科学基金项目(60972053)

2 易于硬件实现的 MET-LDPC 多码率编码方案

在文献[7]中,提到了一种采用随机删余来构造自适应编码的方案,该方案采用了简化的分组和排序法(GS算法)对 MET-LDPC 进行自适应编码,尽管取得良好的性能,但是在实现过程中却面临诸多难题:1) 根据删余算法的规则,编码处的删余在译码处就会在相应的删余点补零进行译码恢复过程。由于是随机删余,在硬件实现中难以实现,会占用大量的资源。2) 随机删余在每一次的删余过程中都需要进行计算,提高了复杂度。

因此,提出一种易于硬件实现的间隔删余算法,并通过实现分析了间隔删余算法与 GS 算法的资源消耗。利用多边类型 LDPC 码良好的码型结构进行删余来构造自适应多边类型 LDPC 编码方案。以码长为 640 bit,码率为 1/2 的 MET-LDPC 码作为母码来进行自适应编码。

为了更好地介绍易于硬件实现的间隔删余算法,先介绍删余比特的划分(见图 1)。

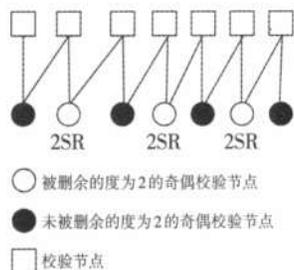


图 1 删余点的划分

1 步可恢复节点(1SR): 如果 1 个删余变量点至少与 1 个幸存校验节点相连接,则称这个删余变量点为 1 步可恢复节点。

k 步可恢复节点(kSR): 如果 1 个删余变量点与 1 个校验节点相连,同时与这个校验节点相连的其余变量点中有 mSR 节点($0 < m \leq k-1$)和至少 1 个 $(k-1)SR$ 节点,则称这个删余变量点为 kSR 。

在译码过程中,为能更快更准地译码,尽量先删余度为 1 的校验点,因为度为 1 的校验点与 1 个校验节点相连,同时这个校验节点又与 3 个信息比特相连,因此在迭代过程中,1 步就可以更新度为 1 的校验点的软信息。而度为 2 的校验点连接 2 个校验节点,在迭代过程中,根据校验方程,能够使信息比特更新来自于度为 2 的校验点的软信息,在删余度为 2 的校验点时,应间隔删余,这样在迭代过程中,删余的校验点就可以得到来自前向和后向的似然信息,使得删余的校验点的似然信息能够尽快更新,提高了信息恢复的准确度。

具体删余流程如下:

1) 计算码率自适应方案中需被删余的比特数 N_{punc}

$$N_{punc} = N - \frac{K}{R_1} \quad (2)$$

2) 根据 N_{punc} 来确定对 N_{one_punc} 的删余个数,如果 $N_{punc} \geq N_{one_length}$, 则 $N_{one_punc} = N_{one_length}$, 否则 $N_{one_punc} = N_{punc}$ 。

3) 由 N_{one_punc} 与 N_{one_length} 的关系来决定 N_{two_punc} , 如果 $N_{punc} \geq N_{one_length}$, 则 $N_{two_punc} = N_{punc} - N_{one_length}$ 。

4) 根据 N_{one_punc} 对度为 1 的校验比特进行连续删余,因此度为 1 的校验比特中删余的比特位置为 $1, 2, 3, \dots, N_{one_punc}$ 。

5) 根据 N_{two_punc} 对度为 2 的校验节点进行间隔删余,且第 1 个比特不删余,即 2 个删余比特之间有 1 个不删余比特,因此在 IRA 中删余的比特位置为 $2, 4, 6, \dots, 2 \times N_{two_punc}$ 。

上述流程中: K 为编码器输入的信息比特个数; N 为当编码码率是 0.5 时,编码器输出的比特个数; R_1 为所要求达到的码率; N_{punc} 为所需要删余的比特数; N_{one_punc} 为所要删余的度为 1 的校验比特个数; N_{one_length} 为度为 1 的校验比特总个数; N_{two_punc} 为所要删余的度为 2 的校验比特个数。

为了更好地对比两种算法,采用 WET-LDPC 编码码率为 0.5, 输入信息比特个数为 640 来进行仿真,删余性能比较如图 2 所示。

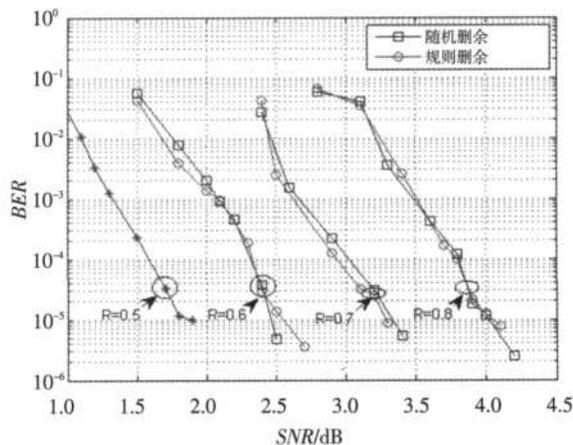


图 2 随机删余与间隔删余性能比较

由图 2 可见,采用本方法的规则删余和随机删余在译码性能上不相上下。同时,多码率 MET-LDPC 与其他 LDPC 相比,尤其是在低信噪比的情况下,性能有所提升(见图 3)。

为了更好地比较两种删余算法的资源消耗,选用了 Xilinx 公司的 Virtex4 xc4vlx15 芯片,仿真环境为 model-sim SE6.5E。表 1 给出了两种删余算法的资源对比情况,可见间隔删余算法简单,且硬件实现时资源占用较少。

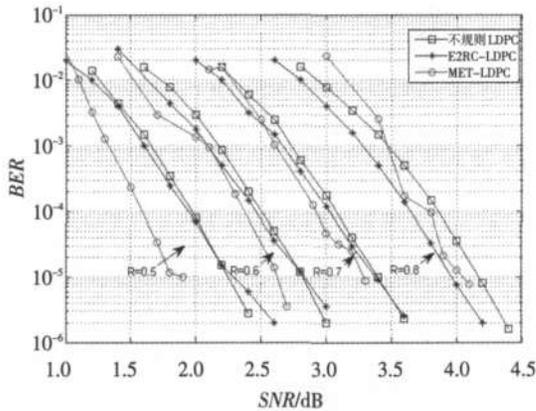


图3 MET-LDPC与其他LDPC性能比较

表1 删余模块资源对比

资源名称	Slice个数	Flip Flop个数	LUT个数
GS删余	56	27	99
间隔删余	13	15	26

3 多码率MET-LDPC编码器硬件实现

基于本文编码码率为0.5的MET-LDPC编码因子图,结合便于硬件实现的删余算法,得到自适应多码率MET-LDPC编码器硬件实现框图,如图4所示。

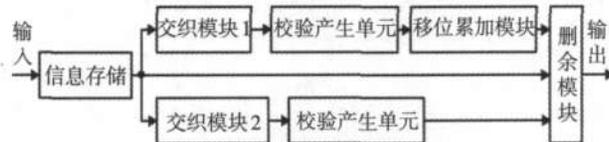


图4 编码器硬件实现框图

根据多边类型LDPC码特点,编码器的关键在于交织和移位累加。重点介绍所采用的易于硬件实现的交织方法,相对于传统交织器的实现,具有交织易操作、可变性强的特性,当交织需要更改时,只需要更改相应参数,而传统的交织更改比较复杂。

传统交织^[8-9]将交织方案的地址提取出来,写入到硬件电路中地址存储单元,当有数据输入时,相应的地址也伴随输入,从而达到交织的目的(见图5)。

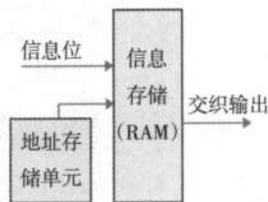


图5 传统交织实现方案

本文实现的是采用一种矩阵变化的交织方案,信息数据流输入到一个存储矩阵,矩阵进行相应的变化,然后再输出,达到交织的目的(见图6)。



图6 本文的交织实现方案

为了更好地区分交织区别,信息长度为25的交织实例如

$$\delta = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 0 & 4 & 1 \end{bmatrix} \quad (3)$$

$$x_0 = 0, x_1 = 3, x_2 = 4, x_3 = 2, x_4 = 1 \quad (4)$$

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 & 24 \end{bmatrix} \Rightarrow$$

$$B = \begin{bmatrix} 3 & 2 & 0 & 4 & 1 \\ 8 & 7 & 5 & 9 & 6 \\ 13 & 12 & 10 & 14 & 11 \\ 18 & 17 & 15 & 19 & 16 \\ 23 & 22 & 20 & 24 & 25 \end{bmatrix} \Rightarrow \quad (5)$$

$$C = \begin{bmatrix} 3 & 12 & 5 & 19 & 21 \\ 8 & 17 & 10 & 24 & 1 \\ 13 & 22 & 15 & 4 & 6 \\ 18 & 2 & 20 & 9 & 11 \\ 23 & 7 & 0 & 14 & 16 \end{bmatrix}$$

$$(address(0), address(1), \dots, address(24)) = (3, 12, 5, \dots, 16) \quad (6)$$

式(3)中,矩阵 δ 第1行表示输入的列数,第2行表示列数的调整,即第3列到第0列,第2列到第1列,第0列到第2列,第4列到第3列,第1列到第4列,对应式(5)中A到B过程。式(4)表示列变换后的循环操作,即第0列循环移位0次,第1列循环移位3次,第2列循环移位4次,第3列循环移位2次,第4列循环移位1次,对应式(5)中B到C的过程。如果按照传统的交织实现方案,需要将 $(address(0), address(1), \dots, address(24))$ 地址写入到地址存储单元,当有数据写入的时候,地址存储单元同步输出存储地址,信息存储便将输入的信息存入到输入的地址存储单元中去,这样会占用较大的存储空间。同时,为保证交织正确,要求输入的信息位与当前的地址存储单元的输出相同步,因此实现复杂度要求较高。而新型的交织实现方法只需要输入相应的列变化和列循环方式,便可以起到交织的作用,简单易更改。

4 仿真与分析

本文自适应MET-LDPC编码器是在Xilinx公司的XC4VSX35 FPGA芯片下实现,设计中使用流水线、乒乓操作等技巧提高系统工作的频率。综合后的硬件资源消耗如表2所示。

表2 MET-LDPC编码器资源占用表

资源类型	资源消耗	资源利用率/%
flip flop个数	11 019	57
LUT个数	10 543	54
10B个数	44	20

验证时,分别以0.6,0.7,0.8码率对多边类型LDPC进行编码,输出码流格式为32位为例,得到时序仿真结果如图7所示。在Testbench对一次时序仿真的输出码

流进行保存,并和 Matlab 中编码的结果比较,多边类型 LDPC 编码器的输出与 Matlab 计算所得到的结果是完全一致的。同理,可以验证在其他码率的情况下,自适应多边类型 LDPC 编码器的编码结果也一样。同时,仿真结果表明,该编码器的吞吐量能达到 100 Mbit/s。

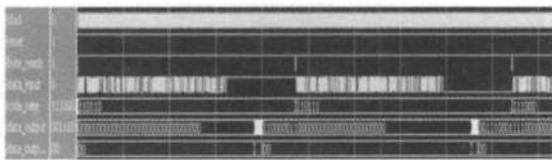


图7 0.6,0.7,0.8 码率编码输出图(截图)

5 小结

现代通信系统为了提高吞吐量以及利用率,通常需要在不同的信噪比情况下对信息比特采用不同的编码方案进行编码,因而自适应编码器的实现尤为重要。笔者应用一种特殊的易于硬件实现的交织方法和移位累加模块实现了自适应 MET-LDPC 的编码器,通过不同的删余方法可以得到不同的码率。综合结果表明:在占用资源适中的情况下,编码器可以达到较高的数据吞吐量,同时也能实现不同码率下的编码,为 MET-LDPC 的实际应用奠定了基础。

参考文献:

[1] RICHARDSON T J, URBANKE R L. Multi-edge type LDPC codes[C]//

(上接第 74 页)

```
TY_ATTRIBUTES) NULL, PAGE_READWRITE, 0.4096, "ShareRAM");  
if(Addr=(LPVOID)MapViewOfFile(hMFile, FILE_MAP_ALL_ACCESS, 0, 0, 0) == NULL) ExitProcess(0);  
UnmapViewOfFile(Addr);  
CloseHandle(hMFile);
```

构造的网络双重防御系统运行稳定、性能可靠,达到了预期目的。

参考文献:

[1] HWANG K, GANGADHAR M. Micro-firewalls for dynamic network security with distributed intrusion detection[C]//Proc. NCA 2001.[S. l.]: IEEE Press, 2001: 68-79.
[2] 薛静锋, 宁宇鹏. 入侵检测技术[M]. 北京: 机械工业出版社, 2004.
[3] WU Junqi, HU Zhengbing. Study of intrusion detection systems (IDSs) in network security[C]//Proc. WiCOM '08. [S. l.]: IEEE Press, 2008: 1-4.
[4] MARTINOVIC I, GOLLAN N, SCHMITT J B. Firewalling wireless sensor networks: security by wireless[C]//Proc. LCN2008. [S. l.]: IEEE Press, 2008: 770-777.

Proc. Workshop honoring professor Bob McEliece on his 60th birthday. Pasadena, California: [s.n.], 2002: 24-25.
[2] 李胜利, 汪洋溢. DVB-S2 及其相关技术[J]. 无线电通信技术, 2009(2): 9-11.
[3] 施玉海, 余方毅. LDPC 在 DVB-S2 中的应用[J]. 电视技术, 2006, 30(5): 91-94.
[4] YANG M, RYAN W E, LI Y. Design of efficiently encodable moderate-length high-rate irregular LDPC codes[J]. IEEE Trans. Comm., 2004, 52(4): 564-571.
[5] KIM J, RAMAMOORTHY A, MCLAUGHLIN S W. The design of efficiently-encodable rate-compatible LDPC codes[J]. IEEE Trans. Comm., 2006, 57: 365-375.
[6] DAMMER U, NAROSKA E, SCHMERMBECK S, et al. A data puncturing IR-scheme for type-II hybrid ARQ protocols using LDPC codes[C]//Proc. Global Telecommunications Conference, 2004. [S. l.]: IEEE Press, 2004: 3012-3016.
[7] YOU Ying, XIAO Min, WANG Lin. The rate-compatible multi-edge type LDPC codes with short block length[C]//Proc. 5th International Conference on Wireless Communications, Networking and Mobile Computing. [S. l.]: IEEE Press, 2009: 1-4.
[8] 武学彬, 薛永林. 用于 DVB-S2 的 LDPC 编码器设计[J]. 电视技术, 2007, 31(S1): 114-116.
[9] 张文俊, 王琳, 徐哲鑫. 自适应码率 QC-LDPC 码编码器的 FPGA 实现[J]. 重庆邮电大学学报: 自然科学版, 2008(5): 534-537.

责任编辑: 许盈

收稿日期: 2010-12-27

[5] LIU A X, GOUDA M G. Diverse firewall design[J]. IEEE Trans. Parallel and Distributed Systems, 2008, 19(9): 1237-1251.
[6] HAMED H, EL-ATAWY A, AL-SHAER E. On dynamic optimization of packet matching in high-speed firewalls[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(10): 1817-1830.
[7] STAVRIDOU V, DUTERTRE B, RIEMENSCHNEIDER R A, et al. Intrusion tolerant software architectures [C]//Proc. DISCEX '01. [S. l.]: IEEE Press, 2001: 230-241.
[8] SALAH K, QAHTAN A. Boosting throughput of Snort NIDS under Linux[C]//Proc. Innovations in Information Technology, 2008. [S. l.]: IEEE Press, 2008: 643-647.
[9] LEITA C, MERMOUD K, DACIER M. ScriptGen: an automated script generation tool for Honeyd[C]//Proc. Computer Security Applications Conference, 21st Annual. [S. l.]: IEEE Press, 2005: 214.

作者简介:

任高明(1986-), 硕士生, 主要从事信息安全研究;
杨全(1982-), 讲师, 主要从事信息安全研究;
乔向东(1970-), 硕士生导师, 主要从事信息安全研究。

责任编辑: 许盈

收稿日期: 2010-07-07