

10G EPON 中面积优化的 RS(255,223)解码器设计

兰 天,刘辉华,杨改改

(电子科技大学 电子科学技术研究院,四川 成都 611731)

摘 要: 提出了一种面积优化的 RS(reed—solomon)解码器的设计方法. 其运用一种改进的 ME(Modified Euclidean)算法求解关键方程模块,其它模块采用迭代结构. 该方法减少了解码器中伽罗法域乘法器的使用,缩减了硬件规模. 基于 TSMC 90nm 标准单元库的实现结果显示该文设计的解码器规模约为 24000 门,与同类设计相比规模最大可缩减 36%.

关键词: 里德所罗门码;ME 算法;RS(255,223);10G EPON

中图分类号: TN403

文献标识码: A

文章编号: 1000—7180(2013)07—0163—04

An Area-Efficient Design of RS(255,223) Decoder for 10G EPON

LAN Tian, LIU Hui-hua, YANG Gai-gai

(Institute of Electronic Science and Technology, UESTC, Chengdu 611731, China)

Abstract: This paper presents a new area-efficient implementation for Reed-solomon(RS) decoder. By using modified euclidean arithmetic to implement the equation solving circuits and the folding architecture for other modules, this method has decreased GF multipliers and simplify the hardware structure. Based on the TSMC 90nm standard cell library, the proposed RS decoder consists of about 24000 gates, which is about 36% smaller than the same kind of conventional ones.

Key words: Reed—Solomon; ME; RS(255,223); 10G EPON

1 引言

在万兆以太网(10Gigabit Ethernet Passive Optical Network, 10G EPON)系统中,RS 码凭借其突发错误和随机错误的良好纠错能力而向该系统提供了强有力的前向纠错能力. 同时 RS 码在移动通信、磁盘存储等系统中都得到了应用.

文中通过对 802.3av 协议^[1]的分析,得出 10G EPON 信息传输过程中每 255 个字节数据绝大多数情况下至多出现 2 比特错误. 分析结论与文献[2]的观点一致. 利用传统的 RS 解码算法设计解码器,硬件资源消耗较多,并不适用于 10G EPON 系统. 因此,文中提出一种改进的 ME 算法及迭代复用的方法设计 RS 解码器,使得电路规模大大降低.

2 RS 解码器算法介绍

RS 解码的基本思想就是:从接收多项式 $R(x)$ 找出错误位置和错误值,即求得错误多项式 $E(x)$,从 $R(x)$ 中减去 $E(x)$ 就得到解码后的正确码字 $C(x) = R(x) - E(x)$.

RS 解码算法分为时域解码,变换域解码以及软解码. 变换域解码和软解码若用硬件来实现过于复杂,耗费的硬件资源过大,适合于软件实现. 时域解码算法具有快速,耗费资源少,控制电路小等优点. 因而选用时域编码.

时域解码算法通常的步骤为:(1)由接收码字计算 RS 码的伴随式 $s(x)$;(2)由伴随式求解错误位置多项式 $\delta(x)$ 和错误值多项式 $w(x)$,也称为解关键方程.(3)利用钱搜索算法^[3],由错误位置多项式求

出各个错误位置,错误位置多项式的根的倒数即错误位置。(4)利用福尼算法^[3],由错误值多项式求出各个错误位置对应的错误值。(5)利用错误位置和错误值进行纠错。

下面主要介绍步骤(2)使用的改进的 ME 算法。其它涉及到的算法参见文献[3]。

定义关键方程为

$$s(x) * \delta(x) = w(x) * \text{mod}(x^{2t}) \quad (1)$$

其中, $s(x)$ 表示伴随式, $\delta(x)$ 表示错误位置多项式, $w(x)$ 表示错误值多项式, $\text{mod}(x^{2t})$ 表示对 x^{2t} 求余。

(1) 假设 RS 码的纠错能力为 t ,由伴随式计算模块得到的结果为 $s(x)$,令 $R_0(x) = x^{2t}$, $Q_0(x) = s(x)$, $L_0(x) = 0$, $U_0(x) = 1$ 。

(2) 设 $\deg(\cdot)$ 表示多项式的阶数,在第 i 个迭代周期内,若 $R_{i-1}(x)$ 和 $Q_{i-1}(x)$ 的最高位系数分别为 a_{i-1} 和 b_{i-1} 。检查 $\deg(R_{i-1}(x))$ 是否小于 $2t$,小于 $2t$ 则计算结束。

(3) 判断 $R(x)$ 和 $Q(x)$ 的阶数。

当 $\deg(R_{i-1}(x))$ 大于 $\deg(Q_{i-1}(x))$ 时,有:

$$R_i(x) = b_{i-1} * R_{i-1}(x) - x * a_{i-1} * Q_{i-1}(x) \quad (2)$$

$$L_i(x) = b_{i-1} * L_{i-1}(x) - x * a_{i-1} * U_{i-1}(x) \quad (3)$$

当 $\deg(R_{i-1}(x))$ 等于 $\deg(Q_{i-1}(x))$ 时,有:

$$R_i(x) = b_{i-1} * R_{i-1}(x) - a_{i-1} * Q_{i-1}(x) \quad (4)$$

$$L_i(x) = b_{i-1} * L_{i-1}(x) - a_{i-1} * U_{i-1}(x) \quad (5)$$

当 $\deg(R_{i-1}(x))$ 小于 $\deg(Q_{i-1}(x))$ 时,有:

$$R_i(x) = x * b_{i-1} * R_{i-1}(x) - a_{i-1} * Q_{i-1}(x) \quad (6)$$

$$L_i(x) = x * b_{i-1} * L_{i-1}(x) - a_{i-1} * U_{i-1}(x) \quad (7)$$

(4) 如果 $\deg(R_{i-1}(x)) < t$,算法结束,并得到错误位置 $\delta(x) = L_i(x)$,错误值 $w(x) = R_i(x)$ 。

改进的 ME 算法与传统的 ME 算法相比,通过减少 $R(x)$ 和 $Q(x)$ 的迭代计算,使得纠错能力稍有减弱,但大大减少了伽罗法域乘法器,从而减小了解码器的面积。因此,本文采用改进的 ME 算法实现 RS(255, 223) 解码器,符合 802.3av 协议,适用于 10G EPON 系统。

3 RS 解码器算法验证

在验证 RS 解码算法时,采用编解码联合验证。具体做法是,把 RS 编码器的输出信息加上不同的

错误值,作为 RS 解码器的输入,验证 RS 解码器能否发现错误及纠错。

RS 编码,就是根据输入信息位得到校验位。详细的编码算法参见文献[3]。对于 RS(255, 223),输入 223 位信息位,输出 223 信息位和 32 位校验位。其中,RS(255, 223) 本原多项式和生成多项式分别为 $p(x) = x^8 + x^4 + x^3 + x^2 + 1$, $g(x) = (x-1) * (x-a) * (x-a^2) \cdots * (x-a^{31})$ 。

首先,由 matlab 库函数 $g(x) = \text{rs_genpoly}(255, 223)$ 计算生成多项式。然后,由 matlab 库函数 $\text{code} = \text{rs_enc}(msg, n, k, g(x))$ 进行 RS 编码。其中, msg 是输入的 223 位信息, n 为输出信息的个数, k 为校验位的个数。对于 RS(255, 223), $n = 255$, $k = 223$, $m = 8$ 。给定一组输入信息 $msg = gf([223 \ 222 \cdots 3 \ 2 \ 1], 8)$,得到 32 位校验位 $[253, 9, 60, \dots, 170, 26]$ 。

若把 RS 编码输出作为 RS 解码的输入,即没有错误发生,则 RS 解码器不进行解码,直接输出信息位 $[223 \ 222 \ 221 \cdots 3 \ 2 \ 1]$ 。因此,为了验证算法的纠错能力,需要加入错误值。在 RS 编码得到后 16 位校验位加入 16 个错误,错误值为 $[16, 15, 14, \dots, 3, 2, 1]$ 。

最后,得到 RS 解码器的 255 位输入数据 $[223, 222, \dots, 2, 1, 253, 9, 60, 220, 91, 202, 48, 83, 201, 181, 81, 123, 117, 63, 98, 219, 4, 6, 73, 0, 235, 191, 99, 103, 122, 111, 149, 23, 51, 146, 168, 24]$ 。下面依次介绍各步骤的实现结果。

3.1 伴随式计算

伴随式计算是 RS 解码的第 1 步,涉及的算法参见文献[3],本计算单元得到 32 项多项式系数从高到低为 $[223, 115, \dots, 84, 196]$ 。得出的伴随式用于解关键方程。

3.2 解关键方程

伴随式计算的输出作为解关键方程的输入,利用第 2 节提到的改进的 ME 算法得到错误位置多项式系数 $\delta_0, \delta_1, \dots, \delta_{15}, \delta_{16}$ 分别为 $[30, 200, \dots, 159, 200]$ 。错误值多项式系数 $w_0, w_1, \dots, w_{14}, w_{15}$ 分别为 $[97, 133, \dots, 88, 28]$ 。

3.3 计算错误位置

解关键方程得出错误位置多项式系数后,利用钱搜索算法^[3],依次带入 $a^i (i = 1, 2, \dots, 254)$,得出错误位置 $0, 1, \dots, 15$ 。

3.4 计算错误值

解关键方程得出错误值多项式系数后,利用福

尼算法^[3],依次带入 $a^i (i=1,2,\dots,254)$,得出错误值 $1,2,\dots,16$.

3.5 纠正错误值

纠正错误值就是把输入数据加上找出的对应位置上的错误值.得到纠错输出为 26,170,145,55,18,147,104,114,110,105,180,231,13,71,9,20.

4 RS 解码器硬件设计

RS 解码器的硬件设计参考文献[4-6]并修改,得到 RS 解码器的整体结构框图,如图 1 所示.

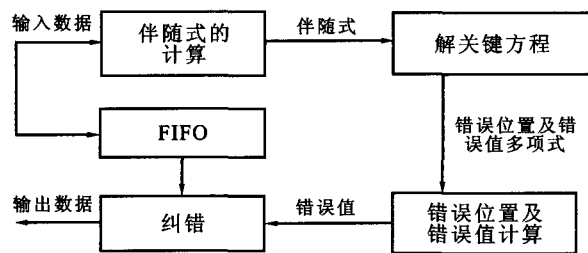


图 1 RS 解码器结构框图

4.1 伴随式计算

定义伴随式为

$$s(x) = \sum_{i=1}^{2t} s_i x^i \quad (8)$$

第 i 个伴随式 s_i 的公式为:

$$s_i = \{ \dots [(r_{n-1} * a^i + r_{n-2}) * a^i + r_{n-3}] * a^i + \dots + r_1 \} * a^i + r_0 \quad (9)$$

其中, $r_{n-1} \dots r_0$ 表示输入的 n 位二进制数, $a^i (i=1,2,\dots,32)$ 对应 32 个特征值,通过查找伽罗法域乘法表得到: $[2,4,\dots,157]$. 伴随式的计算采用迭代的方式进行,迭代单元结构如图 2 所示.

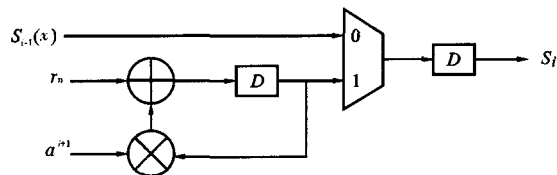


图 2 伴随式计算模块迭代单元示意图

4.2 解关键方程

解关键方程模块是解码器的核心模块,设计的好坏直接决定解码器的性能.硬件设计应用第 2 节介绍的改进的 ME 算法,输入伴随式模块计算出的 32 个伴随式因子,得到 17 个错误位置多项式系数 $\delta_0, \delta_1, \delta_2, \delta_3, \dots, \delta_{16}$ 和 16 个错误值多项式系数 $w_0, w_1, w_2, \dots, w_{15}$.

4.3 计算错误位置

定义错误位置多项式为

$$\delta(x) = \delta_0 + \delta_1 x + \delta_2 x^2 + \dots + \delta_{16} x^{16} \quad (10)$$

其中, $\delta_0, \delta_1, \delta_2, \dots, \delta_{16}$ 由解关键方程模块得出.得到错误位置多项式的各项系数后,然后把 $a^i (i=1,2,\dots,254)$ 代入 $\delta(x)$ 并化简,得到 $\delta(a^i) = \delta_0 + \delta_1 a^i + \delta_2 (a^i)^2 + \dots + \delta_{16} (a^i)^{16} = \delta_0 + \delta_1 a^i + \delta_2 (a^2)^i + \dots + \delta_{16} (a^{16})^i$. 错误位置计算电路结构如图 3.

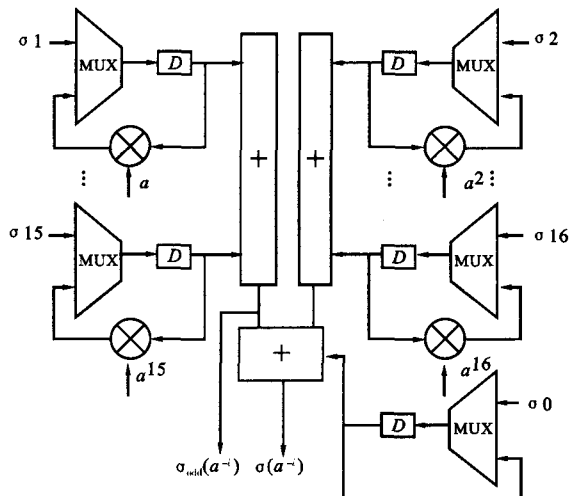


图 3 错误位置计算电路结构示意图

4.4 计算错误值

类似的,定义错误值多项式为

$$w(x) = w_0 + w_1 x + w_2 x^2 + \dots + w_{15} x^{15} \quad (11)$$

其中, $w_0, w_1, w_2, \dots, w_{15}$ 由解关键方程模块得出.得到错误位置多项式的各项系数后,然后把 $a^i (i=1,2,\dots,254)$ 代入 $w(x)$ 并化简,得到 $w(a^i) = w_0 + w_1 a^i + w_2 (a^i)^2 + \dots + w_{15} (a^i)^{15} = w_0 + w_1 a^i + w_2 (a^2)^i + \dots + w_{15} (a^{15})^i$. 错误值计算电路结构如图 4.

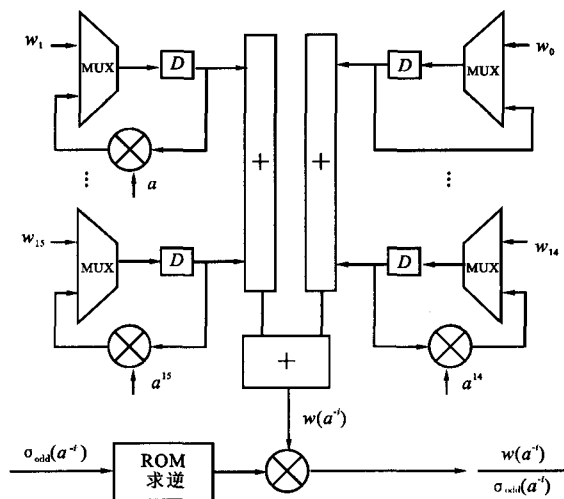


图 4 错误值计算电路结构示意图

4.5 纠正错误值

硬件实现时,把输入数据寄存入 RAM,然后加上相应地址上的错误值,完成纠错.

计算公式为

$$data_out = data_in + error \quad (12)$$

其中, $data_in$ 为输入数据, $error$ 为错误值。

5 RS 解码器实现结果及性能比较

应用第 4 节介绍的电路结构, 本文设计并实现了 RS(255, 223) 解码器。整个解码器模型由 Verilog 语言描述, 用 Synopsys EDA 工具进行综合和仿真验证。仿真结果如图 5 所示, 仿真结果显示为十六进制, 换算成十进制后与 3.5 节计算结果完全一致。



图 5 RS 解码结果

基于 TSMC 90nm 标准单元库, 利用 Synopsys DC 综合出的解码器电路规模约为 24000 门。表 1 对文献[7]及本文设计的 RS(255, 223) 解码器进行了规模比较。

表 1 解码器规模比较

RS decoder	Throughout (Mbytes/sec)	Size (gates)
RS(255, 223)	—	52000
改进的 RS(255, 223)	符合 10G EPON 标准	24000

6 结束语

本文提出了一种面积优化的 RS 解码器的实现方法。该方法运用改进的 ME 算法实现解关键方程模块, 其它模块采用迭代结构实现, 提高了解码器主要部件的复用率, 精简了电路结构, 缩减了电路面积。在解码速度和纠错能力相当的情况下, 本文设计

的 RS(255, 223) 与同类解码器相比可降低约 36% 的电路规模。

参考文献:

- [1] IEEE. LAN/MAN standards committee of the IEEE computer society[S]. New York, IEEE 802.3av-2009.
- [2] Dusan Suvakovic. Compact highly-utilized reed solomon decoder architectures for optical access networks [C]// IEEE Globecom 2011(C. U)SA: Houston, 2011: 1-6.
- [3] 袁强, 胡钢. GPON 中 FEC 编解码器的研究与实现 [D]. 成都: 电子科技大学, 2009.
- [4] 胡庆生, 王志功, 张军, 等. 2.5G b/s reed-solomon 解码器的 VLSI 优化实现[J]. 电路与系统学报, 2005, 10(2): 57-65.
- [5] Lee Hanho. High-speed VLSI decoder [J]. IEEE Trans on Very Large Integration (VLSI) Systems, 2003, 11(2): 288-294.
- [6] Lee H. Modified euclidean algorithm block for high-speed reed-solomon decoder [J]. IEEE Electronics Letters, 2001, 37(14): 903-904.
- [7] Jensen K. Colored petrinets basic concepts, a nalysis methods and practical use[M]. Monographs in Theoretical Computer Science (Volume 1: Basic Concepts). Berlin/Heidelberg/New York: Springer-Verlag, 1997.

作者简介:

兰天男, (1987—), 硕士研究生。研究方向为 10G 以太网无源光网络、通信信道纠错技术。

刘辉华男, (1976—), 博士。研究方向为高速串并并串接口设计、数模混合 IC 设计。

杨改改女, (1985—), 硕士研究生。研究方向为通信技术、数字 IC 设计。