

## 面积优化的 RS(255,239) 高速译码器的设计与实现

龚政辉, 文 磊, 雷 菁

(国防科学技术大学 电子科学与工程学院, 湖南 长沙 410073)

**摘 要:** 针对基于改进型欧几里德(Modified Euclidean, ME)算法的 RS 码译码器所存在的不足, 提出一种面积优化的欧几里德算法的 FPGA 实现方案. 该方案充分利用改进型欧几里德模块的空闲资源, 采用复用的方法将原先的  $2t$  个 PE 模块减少为  $t$  个. 文章将该面积优化的欧几里德模块应用到 RS(255,239) 译码器的设计和实现中, 以达到减少芯片面积, 降低成本的目的. 经过仿真和测试, 基于此设计的高速并行 RS 译码器在正确实现译码功能的同时, 可以大幅减少硬件资源的占用率, 且其吞吐量达到 6.4Gbps.

**关键词:** RS 码; 面积优化; 译码; FPGA; 改进型欧几里德算法

**中图分类号:** TN911

**文献标识码:** A

**文章编号:** 1000-7180(2013)02-0021-04

## An Area—Efficient Implementation of High—Speed RS(255,239) Decoder

GONG Zheng-hui, WEN Lei, LEI Jing

(School of Electronic Science & Engineering, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Aiming at the drawback of Modified Euclidean (ME) algorithm based RS decoder, this paper presents an area—efficient implementation of ME block using FPGA. Making full use of the idle resource of ME block, this implementation scheme adopts multiple structure by reducing the number of PE block from  $2t$  to  $t$ . For reducing the hardware complexity and chip area, this paper applies this area—efficient ME block to the implementation of RS (255,239) decoder. Simulation and test result shows that the decoder based on this improvement can achieve the designed function and its throughput reaches 6.4Gbps while its hardware complexity is widely reduced.

**Key words:** Reed—Solomon codes; area—efficient; decoding; FPGA; Modified Euclidean algorithm

### 1 引言

RS(Reed—Solomon)码是由 Reed 和 Solomon 首先提出的一类多进制 BCH 码<sup>[1]</sup>. RS 码具有严格的代数结构, 构造方便, 编码简单等优点, 且为极大最小距离可分码(Maximum Distance Separable, MDS). 在给定码长  $n$  和信息位长度  $k$  时, RS 码可以纠正  $(n-k)/2$  个突发或随机错误, 是纠错能力最强的  $(n, k)$  码. RS 码在数字差错控制领域得到非常广泛的研究和应用<sup>[2-4]</sup>.

RS 码译码器的设计因求解关键方程的算法不同而分为两类, 即 BM(Berlekamp—Massey)算法以

及欧几里德(Euclidean)算法. 上述两种算法具有各自的优势, 且改进算法都在高速 FPGA 实现中得到广泛地使用. 文献[5]提出一种高速自适应的 Reed—Solomon 译码结构, 吞吐量达到 1.6Gbps; 文献[6]基于 RiBM 改进的无逆 BM 算法, 将吞吐量提高到 1.7Gbps. 文献[7]采用改进的欧几里德(Modified Euclidean, ME)算法设计了一种新的脉动结构, 在保证吞吐量达到 10Gbps 的同时, 减少了一定的资源. 文献[8]基于 BM 算法实现了吞吐量为 960Mbps 的译码器. 文献[9]提出一种动态可重构的方法使译码器吞吐量达到 992Mbps, 并一定程度降低了资源占用率.

收稿日期: 2012-05-25; 修回日期: 2012-06-26

基金项目: 中国空间技术研究 CAST 创新基金项目(CAST201016)

ME 算法的实质是通过求解两个多项式的最大公因式,从而获得错误位置和错误值多项式.该算法便于 FPGA 实现,因此一直是 RS 译码器中常用的算法.但是,该算法也有其不足之处,主要体现在其硬件资源占用太大,增加了实现的复杂度和成本.针对上述问题,本文提出一种面积优化的欧几里德实现方法,可以有效降低硬件复杂度和实现成本.

## 2 译码器原理和结构

假设接收到  $n$  重多项式为  $r(X) = c(X) + e(X)$ . RS 译码的基本思想为:从接收多项式  $r(X)$  找出错误的位置和错误值,即错误图样  $e(X)$ ;把接收码字  $r(X)$  减去  $e(X)$  就得到译码后正确的码字  $c(X) = r(X) - e(X)$ .经典的 RS 码欧几里德译码算法实现结构如图 1 所示.其中 FIFO(First in first out)为先入先出寄存器,用来存储接收多项式.

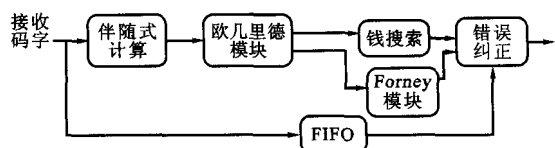


图 1 RS 码译码器实现结构图

伴随式模块的功能是由接收多项式  $r(X)$  计算出  $2t$  个包含错误图样信息的伴随式  $s(X) = s_0 + s_1 X + \dots + s_{2t-1} X^{2t-1}$ . 钱搜索和 Fornoy 模块的功能是由错误位置  $\sigma(X)$  和错误值  $\omega(X)$  多项式求解得到错误图样,以进行最后的纠错.上述两个模块的原理和实现电路在文献[2]中已有详尽叙述.至此,我们所关心的是如何利用  $s(X)$  得到  $\sigma(X)$  和  $\omega(X)$ .事实上,可以通过求解关键方程  $s(X)\sigma(X) = \omega(X) \bmod X^{2t}$  [10] 来获得所需要的错误位置多项式  $\sigma(X)$  和错误值多项式  $\omega(X)$ . 求解关键方程的算法有许多,其中 ME 算法更适合于硬件实现.该算法初始化如下:

$$R_0(X) = X^{2t}, \quad Q_0(X) = S(X)$$

$$L_0(X) = 0, \quad U_0 = 1$$

第  $i$  次迭代时

$$R_i(X) = [\sigma_{i-1} b_{i-1} R_{i-1}(X) + \bar{\sigma}_{i-1} a_{i-1} Q_{i-1}(X)] - X^{\lfloor l_{i-1} \rfloor} [\sigma_{i-1} a_{i-1} Q_{i-1}(X) + \bar{\sigma}_{i-1} a_{i-1} R_{i-1}(X)] \quad (1)$$

$$Q_i(X) = \sigma_{i-1} Q_{i-1}(X) + \bar{\sigma}_{i-1} R_{i-1}(X) \quad (2)$$

$$L_i(X) = [\sigma_{i-1} b_{i-1} L_{i-1}(X) + \bar{\sigma}_{i-1} a_{i-1} U_{i-1}(X)] - X^{\lfloor l_{i-1} \rfloor} [\sigma_{i-1} a_{i-1} U_{i-1}(X) + \bar{\sigma}_{i-1} a_{i-1} L_{i-1}(X)] \quad (3)$$

$$U_i(X) = \sigma_{i-1} U_{i-1}(X) + \bar{\sigma}_{i-1} L_{i-1}(X) \quad (4)$$

其中  $a_{i-1}, b_{i-1}$  为多项式  $R_{i-1}(X), Q_{i-1}(X)$  的首项系数.同时:

$$l_{i-1} = \deg(R_{i-1}(X)) - \deg(Q_{i-1}(X)) \quad (5)$$

$$\sigma_{i-1} = \begin{cases} 1, & \text{if } l_{i-1} \geq 0 \\ 0, & \text{if } l_{i-1} < 0 \end{cases} \quad (6)$$

算法迭代停止的条件是  $\deg(R_i(X)) < t$ , 其中  $\deg()$  代表多项式的阶数.当停止条件满足时,错误值多项式  $\omega(X) = R_i(X)$  而错误位置多项式  $\sigma(X) = L_i(X)$ .如图 2 所示,即为 ME 模块,该模块一共由  $2t$  个基本运算模块(Processing Element)组成.每个 PE 模块[11]相当于一次迭代过程,由阶数比较(Degree Computation)和多项式计算(Polynomial Arithmetic)模块组成.

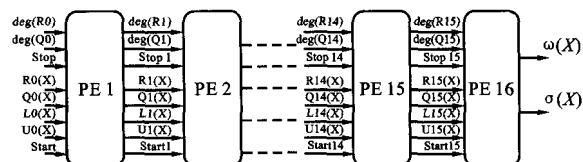


图 2 改进的欧几里德模块

## 3 面积优化的欧几里德模块

上节所述译码器的时序如图 3 所示.可以看到当输入接收码字时,伴随式计算模块、钱搜索和 Fornoy 模块是连续工作的,而 ME 模块则存在大量的空闲时间.针对 ME 模块利用率不高的问题,文献[7]中提出一种流水线结构的译码器,更好的利用 ME 模块,提高译码器的吞吐量.但对于本文要实现的 RS(255,239)码,文献[7]所提出的译码器在资源占用方面仍具有进一步优化的空间.

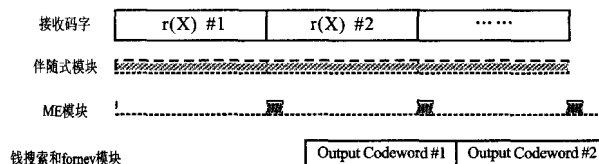


图 3 经典 RS 码译码时序图

ME 算法停止迭代的条件是  $\deg(R_i(X)) < t$ , 而初始条件为  $R_0(X) = X^{2t}$ . 因此为了达到停止条件,改进欧几里德算法需要  $2t$  个 PE 模块.但如前所述,在图 1 所示的译码过程中,  $2t$  个 PE 模块会长期处于空闲状态;文献[7]通过采用流水线结构提高了 ME 模块的利用率.在此,提出面积优化的欧几里德模块以进一步提高资源的利用率和减少占用的资源.

对于 RS(255,239)译码器,其面积优化的欧几里德模块只使用  $t = 8$  个 PE 模块.显然,只使用  $t$  个 PE 是无法完成迭代计算的,但可以将经过  $t$  次迭代

之后的多项式再次送入到复用的欧几里德模块的入口进行迭代. 考虑到每个 PE 模块的延时为 5 个时钟周期, 总时延为 40, 而计算出的伴随式长度为 16, 远远小于其码长 255, 因此复用 PE 模块是可行的. 首先, 16 个伴随式符号经过  $t$  个 PE 模块需要 40 个时钟周期, 这时下一组伴随式还没有计算完成, 因此可以将其再次输入到该  $t$  个 PE 模块, 相当于经过了  $2t$  个 PE 模块, 从而得到错误位置和错误值多项式. 如此, 不仅完成了原来  $2t$  个 PE 模块的功能, 且减少了  $t$  个 PE 模块, 节省了一半的资源. 由于欧几里德是译码器中资源占用最多的模块, 面积优化的欧几里德模块可以有效地提高 ME 模块的利用率和减少资源的使用. 面积优化的欧几里德模块结构如图 4 所示. 由于复用了 PE 模块所以须加入控制模块以完成伴随式的输入和错误位置、错误值多项式的输出.

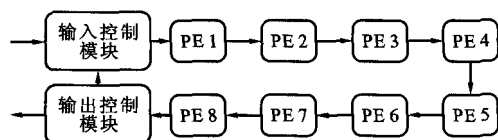


图4 面积优化的欧几里德模块

为提高译码器的吞吐量, 本译码器采用文献[7]中介绍的四路并行流水线结构, 并使用面积优化的欧几里德模块, 其结构如图 5 所示.

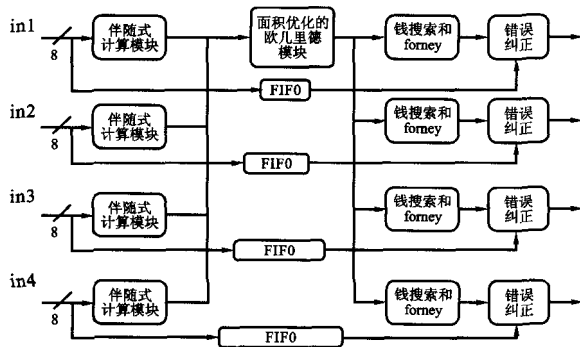
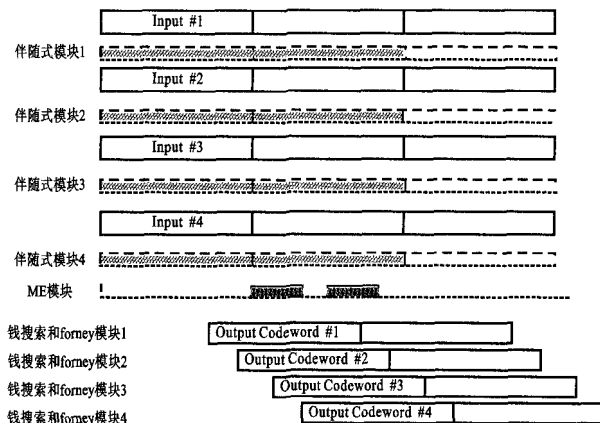


图5 四路并行流水线译码器

四路伴随式计算模块会同时计算并得到四组伴随式, 为了避免伴随式之间数据的相互干扰, 四组伴随式必须串行的依次经过欧几里德模块. 考虑到面积优化的欧几里德模块只有 40 个存储单元, 将伴随式依次输入的方案是不可行的. 由此, 将四路伴随式分为 2 组, 组内两路伴随式首尾相接. 先将第一组两路伴随式送入面积优化的欧几里德模块, 共 32 个符号. 待第一组伴随式重新进入面积优化的欧几里德模块之后, 即 72 个时钟之后, 第二组伴随式接着输入进面积优化的欧几里德模块, 其时序如图 6 所示.

图6 采用面积优化欧几里德模块  
RS(255,239)译码器时序图

#### 4 仿真和结果分析

选用 Xilinx 公司的 Virtex-5 芯片来实现 RS (255, 239) 码译码器, 综合出的时钟频率达 250MHz. 本文实现了三种译码器: 第一种, 一路输入输出, 使用原始欧几里德模块; 第二种, 一路输入输出, 使用本文提出的面积优化的欧几里德模块; 第三种, 四路并行流水线结构译码器, 亦使用面积优化的欧几里德模块. 三种译码器的主要资源见表 1.

表1 三种译码器资源对比

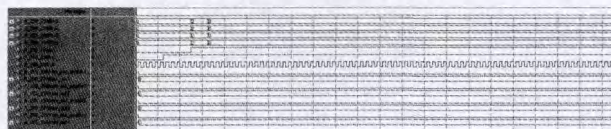
|              | 第一种  | 第二种  | 第三种  |
|--------------|------|------|------|
| 时钟频率(MHz)    | 200  | 200  | 200  |
| 使用的寄存器数量     | 4722 | 2955 | 4945 |
| 使用的 LUT 数量   | 4701 | 3091 | 5792 |
| 吞吐量(Gbits/s) | 1.6  | 1.6  | 6.4  |

由于使用面积优化的欧几里德模块, 因此第二种译码器在具有相同的吞吐量时, 减少了 34% 资源的占用. 在使用了并行流水线结构和复用结构的欧几里德模块的条件下, 第三种译码器在资源增加 17% 的情况下, 其吞吐量增加了四倍. 因此, 本文提出的面积优化的欧几里德确实有效降低了电路复杂度, 减少了芯片面积, 降低了成本. 经计算分析, 对于 RS(255, 239) 码, 一个面积优化的欧几里德模块可以配合 7 路并行流水线输入输出, 在 200MHz 时钟频率下, 吞吐量可以达到 11.2Gb/s. 进一步, 如果希望继续提高吞吐量, 可以单独提高复用结构欧几里德模块的时钟频率. 因为该模块复用结构简单, 通过合理布局布线很容易获得更高的工作频率.

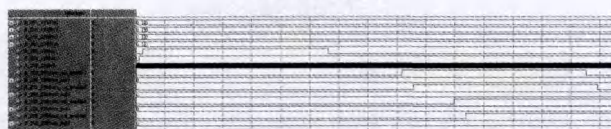
##### 4.1 译码器功能验证

图 7 显示了四路并行流水线结构译码器的仿真图. 由图 7(a) 所示, 译码器有四路并行输入, 当 en 信号为高电平时输入有效. 仿真中, 输入为加入随机或突发错误且数量不大于 8 的全零码字. 本例中有

3 个错误符号,位于码字开头,且错误值为 2. 由图 7 (b)可以看到经过译码器的纠错,四路输出为全零码字,由各路的输出有效信号高电平标记. 可以看到四路输出的延迟是不一致的,究其原因在于四路伴随式依次串行通过复用结构欧几里德模块,所以导致了输出的不同步. 为了克服这种现象,可以通过加入 FIFO 来调整各路的延迟,以使它们同时输出.



(a) 译码器输入



(b) 译码器输出

图 7 译码器仿真结果

此外,本文还利用 Matlab 将随机数据编码,并加入不大于 8 个错误符号的随机和突发错误,最后将待译码数据存入 FPGA 的 ROM 中. 译码器从 ROM 中读取数据并译码,然后将译码结果与原始数据比较. 结果显示,译码器确实具有设计的纠错能力.

#### 4.2 译码器性能对比

将本文设计的译码器与文献[5]和文献[9]中的译码器进行比较,结果如表 2. 相对与文献[5]和文献[9],本文采用面积优化的 ME 算法求解关键方程将资源减少了 46%和 6%,节约了资源. 同时,提高了吞吐量资源占用比,在较少硬件消耗的同时,增加了译码器吞吐量,使其更加适合高速应用.

表 2 不同译码器的比较

|          | RS 码型     | 译码性能<br>(Gbps)         | 资源消耗<br>(LEs) | 算法      | 吞吐量资源比<br>Mbps/LEs |
|----------|-----------|------------------------|---------------|---------|--------------------|
| 文献[5]    | (255 239) | 1.6Gbps<br>时钟 200MHz   | 5753          | BM      | 0.27               |
| 文献[9]    | (255 239) | 0.992Gbps<br>时钟 124MHz | 3295          | ME      | 0.3                |
| 本文第二种译码器 | (255 239) | 1.6Gbps<br>时钟 200MHz   | 3091          | 面积优化 ME | 0.51               |
| 本文第三种译码器 | (255 239) | 6.4Gbps<br>时钟 200MHz   | 5792          | 面积优化 ME | 1.1                |

#### 5 结束语

本文提出一种针对改进欧几里德算法的复用结构实现方法. 该方法优化了面积,提高了译码器主要模块的使用率,并且实现简单. 基于此模块实现的 RS(255,239)码译码器,在译码速率和纠错能力相

当的情况下,面积优化的并行译码器可以降低 34% 的电路规模. 同时,通过该方法实现的 RS(255,239) 四路并行译码器较传统译码结构只增加 17% 的资源,而其译码速率可以达到 6.4Gbps. 进一步,经过分析可知该译码器最大可以支持七路并行译码,最大吞吐量为 11.2Gbps.

#### 参考文献:

- [1] Reed I S, Solomon G. Polynomial codes over certain fields[J]. Soc. Ind. Appl. Math., 1960(8): 300—304.
- [2] 孙江辉,张定云. RS 码在空空导弹遥测系统中的应用[J]. 现代电子技术,2011, 34(2):14—16.
- [3] 马健,王卫民. 一种基于 ME 算法的 RS 译码器 VLSI 高速实现方法[J]. 电子科技,2011, 24(4):17—19.
- [4] 张天瑜. 欧几里德算法的 RS 译码研究和 FPGA 仿真[J]. 武汉理工大学学报,2010,32(2):84—89.
- [5] 丘昕,张浩,元中端,等. 一种高速自适应 Reed—Solomon 译码结构及其 VLSI 优化实现[J]. 电子与信息学报,2009,31(2):484—488.
- [6] 张拯宁,李集林,战勇杰,等. CCSDS—RS(255,223)码高速译码器的硬件实现研究[J]. 中国空间科学技术,2009, 29(5):75—83.
- [7] 郭志勇,杨一波,李广军. 应用于 10G EPON 系统的 RS(255,223)码译码器设计[J]. 电子技术应用,2010, 36(9):58—60.
- [8] Jiabin You, Shaochuan Wu. Design and realization of reed—solomon codes based on FPGA technique[C]// International Conference on Mechatronic Science, Electric Engineering and Computer. China: Wuhan, 2011:2086—2089.
- [9] 谭思炜,潘红兵. 一种动态可重构 Reed—Solomon 译码器的设计[J]. 计算机测量与控制,2011,19(3):673—678.
- [10] Shu Lin, Daniel J, Costello, Jr. 差错控制编码[M]. 北京:机械工业出版社,2007.
- [11] Lee H. High—speed VLSI architecture for parallel reed—solomon decoder [J], IEEE. Trans, VLSI, 2003,11(2):320—323.

#### 作者简介:

龚政辉 男,(1988—),硕士研究生. 研究方向为信道编码理论.

文磊 男,(1981—),博士,讲师. 研究方向为信道编码理论.

雷菁 女,(1968—),博士,教授,硕士生导师. 研究方向为通信信号处理与通信网络