适用于 CCSDS 标准的 RS(255,223)码编码器设计 *

包 涛, 许家栋, 张会生
(西北工业大学 电子信息学院, 西安 710072)

摘 要: 研究了在 CCSDS 标准下 RS 编码器的时域编码方法。分析了 RS 码的编码原理,基本单元电路设计,包括有限域加法器和乘法器,并着重阐述了自然基下常系数并行乘法器的实现方法。在此基础上,选用系数对称的生成多项式,在 Quartus Ⅱ 5.0 编译环境下设计了 RS(255,223)对称结构的编码器,节约了硬件资源,给出了仿真结果图,经检验输出结果正确。采用此方法设计的 RS(255,223)编码器具有控制单元简单、模块结构规则,易于 FPGA 实现,可用于高速场合等特点。

关键词: 编码; 现场可编程门阵列; RS 码编码器; 并行乘法器
中图分类号: TN911.22 **文献标志码**: A **文章编号**: 1001-3695(2010)04-1430-04
doi:10.3969/j.issn.1001-3695.2010.04.062

Design for RS (255,223) encoder based on CCSDS criteria

BAO Tao, XU Jia-dong, ZHANG Hui-sheng
(*School of Electronic & Information, Northwestern Polytechnical University, Xi' an* 710072, *China*)

Abstract: This paper researched a kind of Reed-Solomon (RS) encoder under CCSDS standard, which used time domain coding method. It analysed the principle of Reed-Solomon(RS) encode, the design of basic unit circuit, including Galois field adder and multiplier, and addressed the application method of parallel multiplier with constant coefficients under the natural base. This paper designed a RS (255,223) encoder with symmetric coefficients of generator polynomial under the Quartus Ⅱ 5.0 environment. In addition, figured out the simulation results, and results of the output were correct by the test. The RS (255,223) encoder designed by this method not only effectively simplifies the control unit and has the regular modular structure, but also is easy for FPGA implementation and can be used in high-speed occasions.

Key words: encoding; field programmable gate arrays; RS (255, 223) encoder; parallel multiplier

现代通信系统中,纠错码技术是实现可靠通信的基本方法。由于 RS 码强有力的纠错功能,已经被 NASA(国家航空航天局)、ESA(欧洲航天局)、CCSDS(空间数据系统咨询委员会)等空间组织用于空间信道纠错。RS(255,223)码被 CCSDS 选为常规分包遥测信道纠错编码和高级在轨系统前向和反向链路的纠错编码,是实现 CCSDS 标准低差错率信道纠错编码的关键部件。只要每个码子(255 个符号)中出现的错误不超过 16 个符号,它都能给予纠正[1]。

有限域具有其他代数系统所不具备的特征(如无进位、固定字长和无舍入误差等),因此在纠错编码、数据加密、开关理论和数字信号处理等领域得到广泛应用,在 RS(255,223)码的所有运算中就是以有限域 $GF(2^8)$ 为基础。有限域的两个主要运算是模加和模乘。通常,后者需要更多的计算时间和更复杂的电路。因此,用低复杂度的逻辑电路来实现快速乘法运算显得非常重要。

本课题以我国有高速数据传输要求的复杂航天器以及邻近航天器间数据传输系统为应用背景,对适用于 CCSDS 标准的 RS(255,223)码编码器进行了深入研究,并在 Altera 公司 Stratix 系列 EP1S80B956C6 封装形式为 BGA956 的芯片上,设计实现了 RS(255,223)码的硬件编码器。

1 RS(255,223)编码原理

根据 CCSDS 对于 AOS 信道编码中 RS 编码的建议[1],RS(255,223)码的各项参数如下:

a) $m=8$,为 RS(255,223)码一个符号数表示信息位数;

b) $t=16$,为 RS(255,223)码一个码字内所能纠正的最大错误符号数,m 和 t 为独立参数;

c) $n=2^m-1=255$,为 RS(255,223)码一个码字所包含的符号数;

d) $2t$ 是 RS(255,223)码一个码字检验位的符号个数;

e) $GF(2^8)$ 域上的生成多项式为

$$P(x)=x^8+x^7+x^2+x+1 \tag{1}$$

f) 码生成多项式为

$$G(x)=\prod_{j=b}^{b+2t-1}(x-\alpha^i)=\sum_{i=0}^{2t-1}g_i x_i \tag{2}$$

当 $b=2^{m-1}-t$ 时,生成多项式的系数具有对称性,即 $g_i=g_{2t-i}$,且 $g_0=g_{2t}=1$。对于 RS(255,223)码,$b=112$。

一个 RS(n, k)编码器的实现通常都包括一个 t 级的反馈移位寄存器(FSR)。典型的 RS 编码器结构如图 1 所示[2],其复杂性取决于所用的乘法器。在设计有限域乘法器时需要考虑两个因素:一是基的选择,用于表示有限域上的元素;另一个

收稿日期: 2009-08-19; **修回日期**: 2009-09-30 **基金项目**: 国家"863"计划资助项目(2005AA741072)

作者简介:包涛(1983-),女(蒙古族),内蒙古巴彦淖尔人,博士研究生,主要研究方向为无线通信、信息与编码技术等(baotao322@gmail.com);许家栋(1948-),男,博导,主要研究方向为微波技术、天线与电磁散射理论等;张会生(1955-),男,博导,主要研究方向为卫星通信、移动通信、无线通信等.

是乘法器结构的选择。

2　*RS* 编码器的基本单元电路

在 RS 编码算法中,需要用到大量的多项式求值运算,而这些多项式的求解可以分别使用乘加运算来实现。因此本文首先研究 $GF(2^8)$上有限域加法器和乘法器的逻辑电路设计,然后再讨论 RS 编码器的实现问题。

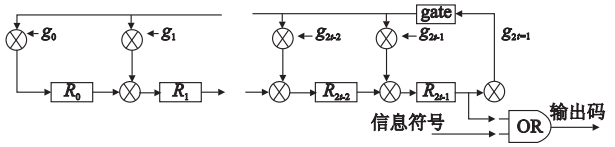

图1 典型的RS编码器

2.1　有限域元素的基表示

$GF(2^m)$域中含有 2^m个元素,任何一个元素 a 可以表示成 m 个线性独立的域元素之和

$$a=a_0\gamma^0+a_1\gamma^1+\cdots+a_{m-1}\gamma^{m-1};\ 0\leqslant i\leqslant m-1,a_i\in GF(2^m) \quad (3)$$

其中:$\{\gamma^0,\gamma^1,\cdots,\gamma^{m-1}\}$称为 $GF(2^m)$在 $GF(2)$上的基;$a_0,a_1,\cdots,a_{m-1}$称为元素 a 在此基下的坐标。

若 $f(x)$是 $GF(2)$域上的 m 阶本原首一多项式,$\alpha\in GF(2^m)$,且 $f(\alpha)=0$,即 α 是 $GF(2^m)$域的本原域元素,则$\{1,\alpha,\cdots,\alpha^{m-1}\}$被称为标准基、多项式基或自然基。

2.2　有限域加法器

RS 编/译码算术运算都是在伽罗华域 $GF(p^m)$上进行的,由于没有进位,伽罗华域中的加法运算比普通加法运算更容易实现。在做加法运算时,只需简单地将它们的矢量表示进行异或,因此仅用寄存器和异或门就可完成。

将 $GF(2^m)$域元素用自然基表示为

$$a=a_0+a_1\alpha+\cdots+a_{m-1}\alpha^{m-1} \quad (4)$$

$$b=b_0+b_1\alpha+\cdots+b_{m-1}\alpha^{m-1} \quad (5)$$

$$c=a+b=(a_0+b_0)+(a_1+b_1)\alpha+\cdots+(a_{m-1}+b_{m-1})\alpha^{m-1} \quad (6)$$

在 AOS 标准中,RS(255,223)中的域元素为 $GF(2^8)$中的元素,因此当两数相加时,只需进行异或操作即可简单实现。

该部分生成的逻辑框图如图2 所示(采用 Quartus5.0 软件生成)。其中:a、b 是输入信号,8 位数据宽;clk 是时钟输入信号;c 为输出信号,8 位数据宽,时钟上升沿触发。

2.3　有限域乘法器

伽罗华域中的乘法运算是 RS 编/译码器中最主要的算术运算,其所占用资源和速度在很大程度上决定了 RS 编/译码器芯片的占用资源和性能,因此它是实现 RS 编/译码器的中心和关键。从实现结构上来说,有限域乘法器可以分为比特串行乘法器和比特并行乘法器。串行乘法器是按比特进行运算,将两个元素分别以自然基表示

$$a=a_0+a_1\alpha+\cdots+a_{m-1}\alpha^{m-1} \quad (7)$$

$$b=b_0+b_1\alpha+\cdots+b_{m-1}\alpha^{m-1} \quad (8)$$

按比特相乘

$$a\times b=b_0a_0+b_0a_1\alpha+\cdots+b_0a_{m-1}\alpha^{m-1}+b_1a_0\alpha+b_1a_1\alpha^2+\cdots+b_1a_{m-1}\alpha^m+\cdots+b_{m-1}a_0\alpha^{m-1}+\cdots+b_{m-1}a_{m-1}\alpha^{2m-2} \quad (9)$$

根据域元素的运算法则可知,若本原多项式为 $P(x)=\sum_{j=0}^{m}P_j\alpha^j$($P_m=1$),则

$$a^{m+k}=a^k\sum_{j=0}^{m-1}P_j\alpha^j \quad (10)$$

依照式(10)对(9)中的高阶项进行降阶,就可得到最后的结果,串行乘法器的结构框图如图 3 所示。其中:R_i、C_i是寄存器,$P_0\sim P_{m-1}$是 $P(x)$的系数。

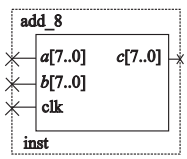

图2 伽罗华域的加法逻辑框图

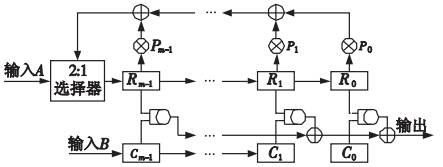

图3 比特串行乘法器的原理框图

串行乘法器的硬件实现比较简单,但是由于运算按比特进行,高速实现的难度较大,不易达到较高的吞吐率。在高速应用时一般考虑采用比特并行乘法器。比特并行乘法器的原理结构如图 4 所示。

所谓比特并行是指通过电路连接,直接实现输出结果的多位运算,最后结果并行输出,这样可以显著提高处理速度。

图 4 中以每符号 4 bit 为例,$a\times b=c_0+c_1\alpha+c_2\alpha^2+c_3\alpha^3$,$b_0\sim b_3$经过 Block1 得到 $b_4\sim b_6$,其他的四个功能模块 A、B、C、D 分别计算 $c_0\sim c_3$。若本原多项式 $P(x)=x^4+x=1$,则

$$b_4=b_0+b_3,b_5=b_2+b_3,b_6=b_1+b_2+b_3$$

$$c_0=a_0b_0+a_1b_1+a_2b_2+a_3b_3,\cdots,c_3=a_0b_3+a_1b_2+a_2b_1+a_3b_4 \quad (11)$$

对于伽罗华域乘法运算的研究工作已做了很多,其中最适用于 RS 编码器的主要有以下几种[3]:

a)查表法。这种方法将有限域的每个元素用本原元的指数表示,并在 ROM 中存储每个元素的对数及反对数,有限域中任意两个元素的乘积可通过查表获得。这种乘法器随着位数的增加占用的芯片资源将迅速增加。

b)线性反馈移位寄存器法。有限域中的任意两个元素的乘积可由线性反馈移位寄存器电路获得。这种方法实现的乘法器电路简单、面积小,但速度慢。

c)Messey-Omura 法。这种乘法器采用有限域中的正交基表示乘数和被乘数,乘积项的每一个分量可用同一个函数表示。该乘法器对有限域元素的平方运算及指数运算非常有效。这种方法实现的乘法器占用芯片资源较少,但需要将有限域中的元素用特殊基表示,这与有限域中的本原多项式选择有关。

d)Berlekamp 位串法。这种乘法器因为乘数用对偶基表示,被乘数用标准基表示,乘积项用对偶基表示,所以乘法器只需移位和异或运算。这种乘法器在目前所有已实现的乘法器中占用的芯片资源最少,但需要基与基之间的转换。

在本课题编码器的结构设计中,考虑到 RS 编码原理简单,并且生成多项式 $g(x)$系数固定的特点,本文将编码器中的有限域乘法器设计成一个乘数固定的乘法器,称为常系数乘法器。这种乘法器的乘数与被乘数均用自然基表示,不需要基与基之间的转换,大大减少了所需占用的资源,应用广泛、结构简单,易于扩展到高阶有限域中。

设输入 $A=a_7\alpha^7+a_6\alpha^6+\cdots+a_2\alpha^2+a_1\alpha+a_0$。当有限域乘法器的一个乘数为常数时,如 $g_1=69$(生成多项式的一次项系数),用自然基表示为 $g_1=\alpha^6+\alpha^2+1$。

令输出 $B=b_7\alpha^7+b_6\alpha^6+\cdots+b_1\alpha+b_0$,推出

$$B=(A\times g_1)\bmod P(x)=(a_7\alpha^7+\cdots+a_1\alpha+a_0)\cdot(\alpha^6+\alpha^2+1)=b_7\alpha^7+b_6\alpha^6+\cdots+b_3\alpha^3+b_2\alpha^2+b_1\alpha+b_0 \quad (12)$$

A 和 g_1的乘积 B 可分两步得到:

a)将 $A(\alpha)$和 $g_1(\alpha)$两个多项式按常规方法相乘,得到一

个次数不大于 14 的多项式 $C(x)$。

b)将 $C(x)$ 对 $GF(2^8)$ 的本原多项式 $P(x)=x^7+x^2+x+1$ 求模,得到次数不大于 7 的多项式,即得 A 与 g_1 的乘积 B。

式(13)所示为表示系数 B 的一组异或方程,其中的加号表示异或运算。也就是说,乘以 g_1 的模块只需要由异或门组合来实现,这大大减少了硬件资源的消耗,并且提高了其运行速度。

$$b_7=a_4+a_3+a_2+a_1,b_6=a_4+a_0$$
$$b_5=a_7+a_3,b_4=a_7+a_6+a_2$$
$$b_3=a_6+a_5+a_1,b_2=a_7+a_5+a_4+a_0$$
$$b_1=a_6+a_2+a_1,b_0=a_5+a_4+a_3+a_2+a_0 \quad (13)$$

按照同样的计算方法可以得到其他 15 个有限域乘法器。在使用 VHDL 进行设计时,可以将每一个乘法器封装成一个子函数,然后在编码器模块中进行调用。这样设计出的编码器程序结构简单,易于理解,不容易出错。

该部分生成的逻辑框图如图 5 所示,码组 A 由 din[7..0] 输入,码组 B 由 dout[7..0]输出。

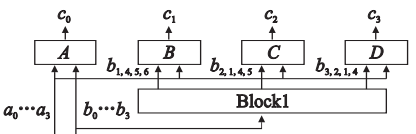

图4 比特并行乘法器的原理框图

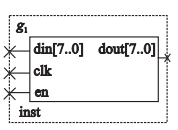

图5 伽罗华域的乘法逻辑框图(乘以g_1的模块)

3 RS(255,223)码的编码器设计与实现

3.1 RS 码的编码器结构

就 RS 码的编译码技术而言,存在时域和频域两种算法。其中时域编译码算法出现较早,由于比较成熟而被广泛采用;频域编译码算法虽然出现较晚,但由于可借助快速傅里叶正反变换(FFT/IFFT)提高编译码速度,具有较大的发展潜力[4,5]。

对于 RS 编码,时域和频域算法都比较简单,时域编码只需要一次多项式除法,频域编码只需要一次 IFFT 变换。不同的是,用时域编码编出的码字是系统码,而频域编码编出的码字是非系统码。

对于线性分组码来说,在其他指标相同的情况下,非系统码和系统码的纠错能力相同,但在译码时系统码可直接读出信息元,非系统码还要增加一个信息元的转换电路。考虑到系统码解码的方便性,本文采用时域系统码的编码方案(除法器结构)对输入信息进行编码。

由第 2 章中对符合 CCSDS 标准的 RS 码参数的介绍可知,$GF(2^8)$ 域上 RS(255,223)码的生成多项式为

$$G(x)=\sum_{i=0}^{32}g_ix^i=\prod_{j=128-t}^{127+t}(x-\alpha^j) \quad (14)$$

当 $t=16$ 时,相乘得到 32 次多项式,如式(15)所示,可以看出,其系数具有对称性。

$$g(x)=x^{32}+\alpha^{249}x^{31}+\alpha^{59}x^{30}+\alpha^{66}x^{29}+\alpha^{4}x^{28}+\alpha^{43}x^{27}+\alpha^{126}x^{26}+\alpha^{251}x^{25}+\alpha^{97}x^{24}+\alpha^{30}x^{23}+\alpha^{3}x^{22}+\alpha^{213}x^{21}+\alpha^{50}x^{20}+\alpha^{66}x^{19}+\alpha^{170}x^{18}+\alpha^{5}x^{17}+\alpha^{24}x^{16}+\alpha^{5}x^{15}+\alpha^{170}x^{14}+\alpha^{66}x^{13}+\alpha^{50}x^{12}+\alpha^{213}x^{11}+\alpha^{3}x^{10}+\alpha^{30}x^{9}+\alpha^{97}x^{8}+\alpha^{251}x^{7}+\alpha^{126}x^{6}+\alpha^{43}x^{5}+\alpha^{4}x^{4}+\alpha^{66}x^{3}+\alpha^{59}x^{2}+\alpha^{249}x+1 \quad (15)$$

有限域 $GF(2^8)$ 中的每一个元素都用它的一组自然基底 $\{1,\alpha,\alpha^2,\alpha^3,\alpha,^4\alpha^5,\alpha^6,\alpha^7\}$ 来表示。设 α 是本原多项式 $P(x)$ 的根,则有 $\alpha^8=\alpha^7+\alpha^2+\alpha+1$。按此方法可以求得有限域 $GF(2^8)$ 中的所有元素,表 1 列出了 $P(x)$ 下 $GF(2^8)$ 域中的部分元素,由此可以计算出生成多项式的各项系数,如式(16)所示。

表 1 $GF(2^8)$ 域元素表

幂	多项式表示	十进制值	二进制值
a	x	2	0000_0010
a^2	x^2	4	0000_0100
a^3	x^3	8	0000_1000
a^4	x^4	16	0001_0000
a^5	x^5	32	0010_0000
a^6	x^6	64	0100_0000
a^7	x^7	128	1000_0000
a^8	x^7+x^2+x+1	135	1000_0111
a^9	x^7+x^3+1	137	1000_1001
a^{10}	$x^7+x^4+x^2+1$	149	1001_0101
a^{11}	$x^7+x^5+x^3+x^2+1$	173	1010_1101
a^{12}	$x^7+x^6+x^4+x^3+x^2+1$	221	1101_1101
a^{13}	$x^5+x^4+x^3+x^2+1$	61	0011_1101
a^{14}	$x^6+x^5+x^4+x^3+x$	122	0111_1010

对数据进行 RS 编码的实质就是求其校验码,而校验码即是 $m(x)x^{n-k}$ 被 $G(x)$ 除以后的余式,在硬件实现上为图 6 所示的一组线性反馈移位寄存器(LFSR)。

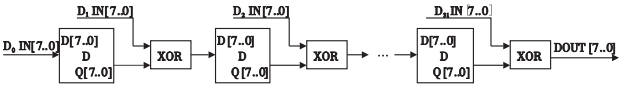

图6 移位寄存器

$$g(x)=x^{32}+69x^{31}+251x^{30}+239x^{29}+38x^{28}+105x^{27}+20x^{26}+238x^{25}+218x^{24}+114x^{23}+188x^{22}+229x^{21}+112x^{20}+139x^{19}+85x^{18}+64x^{17}+213x^{16}+64x^{15}+85x^{14}+139x^{13}+112x^{12}+229x^{11}+188x^{10}+114x^{9}+218x^{8}+238x^{7}+20x^{6}+105x^{5}+38x^{4}+239x^{3}+251x^{2}+69x+1 \quad (16)$$

即:$g_{31}=g_1=69$,$g_{30}=g_2=251$,$g_{29}=g_3=239$,$g_{28}=g_4=39$,$g_{27}=g_5=105$,$g_{26}=g_6=20$,$g_{25}=g_7=238$,$g_{24}=g_8=218$,$g_{23}=g_9=114$,$g_{22}=g_{10}=188$,$g_{21}=g_{11}=229$,$g_{20}=g_{12}=112$,$g_{19}=g_{13}=139$,$g_{18}=g_{14}=85$,$g_{17}=g_{15}=64$,$g_{16}=213$,$g_0=1$。

从图 7 可以看出,RS 编码器硬件结构中有些单元如乘法器、加法器和移位寄存器是重复的,这样可以把它独立出来做成底层模块。图中的 $g_0,g_1,\cdots,g_{31}$ 模块即为 2.3 节介绍的常系数乘法器,有限域的加法器在 2.2 节已经作过详细介绍,而移位寄存器如图 6 所示。

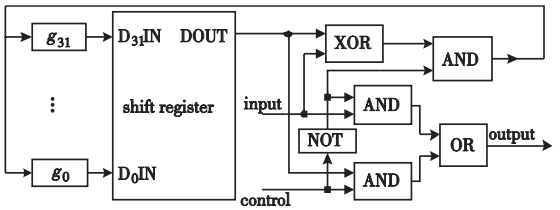

图7 RS编码器硬件结构

RS 编码电路的工作原理是:

a)2t 级移位寄存器初始状态全部清零,此时控制信号输出为“0”,送入信息组 $m(x)$ 的系数,并进行移位,高次位系数首先进入电路,它一方面直接经或门输出,另一方面乘以 $x^{(n-k)}$ 次后进入除法电路,从而完成了 $x^{n-k}\cdot m(x)$ 的作用。

b)k 次移位后 $m(x)$ 全部送入电路,完成了除法作用,此时移位器内保留的是余式 $r(x)$ 的系数。在二进制的情况下此系数就是校验码元。

c)此时控制信号输出为“1”,经过 $n-k$ 次移位后,寄存器中的校验码元全部输出,与最初的 k 位信息组成一个长为 n 的码字。

再置控制信号输出为“0”,继续上面的步骤,即可对后面

的数据进行 RS 编码。

3.2　RS 编码器的 MATLAB 仿真

在对 RS 编码原理研究的基础上,本文提出了一种 RS(255,223)编码器的软件解决方案。此方案在日后的算法验证和本文下一节的硬件设计调试中起着至关重要的作用。

RS 编码过程可分为如下几步:

a)计算 RS(255,223)码的生成多项式系数。调用 MATLAB 子函数 GENPOLY = RSGENPOLY(N,K,PRIM_POLY,b)。

参数:N = 255 为一个码字所包含的符号数;K = 223 为信息位数;PRIM_POLY = $(110000111)_2$ = 391 为本原多项式 $P(x)$的系数。如式(16)所示,当 t = 16 时,生成多项式系数具有对称性,因而取 $b = 128 - t = 112$。输出矩阵 GENPOLY[1,32]为生成多项式系数。

经计算得生成多项式各项系数:

$$g_0=1,g_1=69,g_2=251,g_3=239,g_4=38,g_5=105,g_6=20,g_7=238,g_8=218,g_9=114,g_{10}=188,g_{11}=229,g_{12}=112,g_{13}=139,g_{14}=85,g_{15}=64,g_{16}=213,g_{17}=64,g_{18}=85,g_{19}=139,g_{20}=112,g_{21}=229,g_{22}=188,g_{23}=114,g_{24}=218,g_{25}=238,g_{26}=20,g_{27}=105,g_{28}=38,g_{29}=239,g_{30}=251,g_{31}=69$$

b)求校验码。调用 MATLAB 子函数 CODE = RSENC(MSG,N,K,GENPOLY)。

参数:取待编码信息 MSG = [1,2,3,…,222,223];N = 255;K = 223;GENPOLY[1,32]为生成多项式系数矩阵。输出矩阵 CODE[1,255]为编码后的码字,等于 223 个信息位加 32 个校验位。

运算得到码字 CODE[1,255] = [1,2,3,…,222,223,184,32,247,171,36,60,227,188,154,55,147,106,94,94,203,163,227,48,127,207,53,23,106,196,188,77,106,51,167,148,14,65]。

c)常系数乘法器。调用函数 RESULT = CODE_MUX(A,B)。

参数:$A = a_7\alpha^7 + a_6\alpha^6 + \cdots + a_2\alpha^2 + a_1\alpha + a_0$代表任意一个信息位;$B$ 代表生成多项式系数,如 $B = g_{30} = 251 = (11111011)_2$,输出多项式如式(17)(18)所示。

$$\begin{aligned}\mathrm{RESULT} &= (a_7\alpha^7 + a_6\alpha^6 + \cdots + a_1\alpha + a_0)\cdot(\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \\ &\alpha^3 + a + 1) = b_7\alpha^7 + b_6\alpha^6 + \cdots + b_1\alpha^1 + b_0 \qquad (17)\end{aligned}$$

其中

$$\begin{aligned}&b_7 = a_0 + a_2 + a_4 + a_5 + a_6 + a_7, b_6 = a_0 + a_1 + a_2 + a_3\\ &b_5 = a_0 + a_1 + a_2 + a_7, b_4 = a_0 + a_1 + a_6\\ &b_3 = a_0 + a_5, b_2 = a_4 + a_7\\ &b_1 = a_0 + a_2 + a_3 + a_4 + a_5, b_0 = a_0 + a_1 + a_3 + a_5 + a_6 + a_7 \qquad (18)\end{aligned}$$

通过此函数可以得到每移入一位信息,32 个常系数乘法器各次的输出。此函数的重要性在 RS 编码器的硬件设计和调试中显得尤为明显,它可以减少大量冗余的工作而专门用来验证每次时钟到来时的输出结果是否正确。

3.3　RS 编码器的 FPGA 实现

编码器的实现方法有乘法器结构、基于多项式的除法器结构、Systolic 阵列结构等,但考虑到除法器结构的编码器只用到了常系数乘法器,而常系数乘法器的硬件实现非常简单,占用的硬件资源也较少。因此,本文设计的是基于多项式除法器结构的 RS(255,223)编码器。此设计是在 Quartus Ⅱ5.0 环境下,使用 VHDL 语言描述整个编码器模型,并以 Altera 公司生产的 EP1S80B956C6 FPGA 芯片为硬件平台进行实现。本设计在 Quartus Ⅱ5.0 软件下的综合结果如表 2 所示。

表 2　RS(255,223)编码模型的综合结果

综合结果	Total logic element	Total pins	最高时钟频率
RS(255,223)	399	290	190 MHz

利用 ModelSim SE 5.8 软件对设计进行仿真,结果如图 8、9 所示。

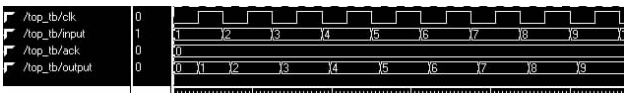

图8　RS(255,223)编码器仿真初期信息元输入输出波形图

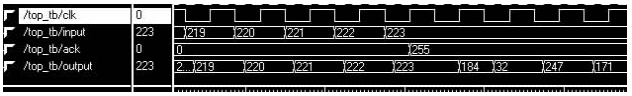

图9　RS 255 223 编码器仿真末期校验元输出波形图

图 8 显示了仿真初期的输出波形,控制信号 ack 置 0,输入数据 input 是 1 ~ 223 的 223 个符号,可以看到信息码组 output 也同样按位从 1 到 223 依次输出。而图 9 清楚地显示了校验元的输出情况,控制信号 ack 置全 1,即 $(11111111)_2$ = $(255)_{10}$,当信息码组输入完第 223 个符号时,则停止输入。此时输出信号 output 紧接着第 223 个符号,依次输出校验元 184、32、247、171 等。可以验证输出的校验元与 3.2 节中 MATLAB 环境下计算出的数据完全吻合。

4　结束语

本文详尽研究和分析了纠错码相关的数学知识和具体的 RS 编码理论。在此基础上,按照 CCSDS 标准下的 RS 码参数,深入研究了适用于 AOS 系统的 RS 码编码器。考虑到系统码解码的方便性,选用时域编码方法。本文中对编码器的基本单元电路进行了详细分析,包括有限域加法器和乘法器。最后,在软硬件环境下分别设计出了能够应用于 AOS 系统的 RS(255,223)码的高速编码器,并对其实现过程及为节约硬件资源而作的各项优化进行了详细介绍。与文献[2]相比,本文的编码器需要的有限域乘法器从 31 个减少到 16 个,大大地减少了芯片面积,同时本文的编码器能够获得较快的速度,可应用于深空卫星通信的级连系统中。

将所设计的 RS(255,223)编码器进行 Quartus Ⅱ综合和优化,采用 Altera 公司 Stratix 系列的 EP1S80B956C6 芯片,得到编码器的性能参数如下:工作时钟频率 30 MHz,数据吞吐率为 240 Mbps,占用逻辑单元数为 401 个,完成 1 个码字的编码需要 255 个时钟周期。

参考文献:

[1] SWEENEY P. 差错控制编码[M]. 俞越,张丹,译. 北京:清华大学出版社, 2004:158-170.

[2] CCSDS 700.0-G-3, Advanced orbiting systems, networks and data links: summary of concept, rationale and performance[S]. 1992.

[3] 张思周,陈爱平,宋永淳,等. AOS 数据模拟器及处理器的设计与实现[J]. 遥测遥控, 2005, 26(2): 44-49.

[4] LEE H. A high-speed low-complexity Reed-Solomon decoder for optical communications [J]. IEEE Trans on Comput, 2005, 52(8): 461-465.

[5] BAEK J H, SUNWOO M H. New degree computationless modified Euclid algorithm and architecture for Reed-Solomon decoder[J]. IEEE Trans on Very Large Scale Integration, 2006, 14(8):915-920.