

闻章, 刘绍凯. 支持 BIST 的 RS 编解码器在国产 FPGA 上的设计与实现[J]. 智能计算机与应用, 2025, 15(1): 171-177.
DOI: 10.20169/j.issn.2095-2163.24051801

支持 BIST 的 RS 编解码器在国产 FPGA 上的设计与实现

闻章, 刘绍凯

(中国电子科技集团公司第三十研究所, 成都 610041)

摘要: RS 编码是一类应用广泛的纠错编码, 由于有些国产 FPGA 平台未能提供该 IP 核的使用以及工程中在不同 FPGA 平台使用共享模块的需求, 本文设计了一种低复杂度、低资源使用的 RS 编码器和 RS 解码器, 提出了 FPGA 资源优化的方法, 使用工具软件 Matlab 和 Modelsim 进行了联合仿真, 验证了设计的 RS 编码器和 RS 解码器电路的正确性与合理性, 并在国产 FPGA 厂家安路科技的 EG4S20NG88 和 PH1A100GCG324 以及国外 FPGA 厂家 Xilinx 的 XC7K325TFFG900 上进行了实现, 并对实现后的资源使用情况进行了对比。同时通过增加 PRBS 模块实现了内建自测试功能, 极大地减少了模块在不同的 FPGA 平台之间移植的时间。

关键词: RS 编码器; RS 解码器; 内建自测试; PRBS; 国产 FPGA

中图分类号: TN911

文献标志码: A

文章编号: 2095-2163(2025)01-0171-07

BIST supported design and realization of reed-solomon encoder and decoder on domestic FPGA

WEN Zhang, LIU Shaokai

(No. 30 Institute of CETC, Chengdu 610041, China)

Abstract: Reed-Solomon code is one kind of error correction codes which has been widely used, since some domestic FPGA platforms do not provide the use of this IP core and the requirements of using shared module in different FPGA platforms in project, low-complexity and low resource usage Reed-Solomon encoder and RS decoder are designed in this paper, and the method of FPGA resource optimization is proposed, co-simulated with Matlab and Modelsim is used to verify the correctness and rationality of designed RS encoder and RS decoder circuit, and it's implemented on EG4S20NG88 and PH1A100GCG324 of Anlu Technology which is domestic FPGA manufacturer and XC7K325TFFG900 of Xilinx which is foreign FPGA manufacturer. Meantime, the PRBS module is added to implement Built-in Self-Test function which greatly reduces the time of porting the module among different FPGA platforms.

Key words: Reed-Solomon encoder; Reed-Solomon decoder; BIST; PRBS; domestic FPGA

0 引言

里德-所罗门 (RS) 码是由 Irving S. Reed 和 Gustave Solomon 在 1960 年提出的一种纠错编码, 之后成为了 BCH (Bose ray-Chaudhuri Hocquenghem) 码的一个最重要的子类^[1]。BCH 码纠错能力强、构造方便、编译码简单、有快速的译码算法, 具有严格的代数结构, 在编码的理论和实际中都起着重要的作用, 也是迄今为止研究最为详尽、分析最为透彻和成果最为丰富的码类^[2]。RS 码是线性分组码中一种典型的纠错码, 特别是在中、短码长的情况下可以

纠正随机错误和突发错误^[3]。RS 码出色的纠错性能被广泛应用于诸多领域: 将 RS 码应用于无线光通信系统中, 作为信道编码提升无线光通信系统的整体性能^[4]; 将 RS 码应用于海底光纤系统, 提出了一种能跟上光传输速率的并行 RS 解码器^[5]; 在跨用户数据去重技术中使用 RS 码的思想, 节省开销, 提高效率^[6]; 在运载火箭图像测量系统中使用 RS 码设计, 改善信道误码、提高信道增益、有效提高图像画质质量^[7]; 在战术数据链系统中从编码效率、冗余度、码长 3 个方面对 RS 码的抗干扰能力进行分析和研究^[8]; 在 400 Gb/s 以太网物理编解码子层

作者简介: 闻章 (1990—), 男, 硕士, 主要研究方向: 信息安全与保密通信; 刘绍凯 (1986—), 男, 硕士, 主要研究方向: 信息安全与保密通信。

Email: 278728960@qq.com。

收稿日期: 2024-05-18

哈尔滨工业大学主办 ◆ 科技创新与应用

(PCS)中实现了高速RS码电路,以满足高速以太网的需求^[9];在工业系统中使用RS码,改善了单向传输中高误码率和数据包丢失的情况,并实现了多个RS编译码单元并行处理的管理控制^[10];在电力线通信中使用RS码,保证数据的稳定可靠^[11];将RS码应用于引信装订系统中,解决接收端出现码元传输错误,不能纠错的问题^[12];使用RS码,解决了电力通信系统的监测信息因长距离传输导致出现误码的问题^[13];在软件无线电通信系统上验证了RS码的纠错性能^[14]。

近几年,美国先后对中兴、华为的制裁,唤醒了国民对自主研发的认识,大力发展国产芯片迫在眉睫。国产FPGA(现场可编程门阵列)厂家也勇于直面挑战、牢牢抓紧了机遇,新产品如雨后春笋般涌现出来。但是不同的国产FPGA使用的EDA(电子设计自动化)工具不尽相同,对常用IP的支撑程度也各有区别,给常用模块如RS编解码器在不同的FPGA平台上的使用带来了困难。

本文设计了一种低复杂度、低资源使用的RS编码器和RS解码器,同时实现了内建自测试(Build-in Self-Test)功能,很大程度上简化了该RS编解码器在不同FPGA平台间移植的调试、验证工作。在国产FPGA平台上进行了验证,所使用的FPGA平台是安路的EG4S20NG88和PH1A100GCG324,采用自主开发的EDA软件Tang Dynasty,目前只有一些常用IP核,没有提供RS编解码的IP核。本文设计的IP核可以对上述FPGA平台进行有效支持,同时由于其低复杂度和低资源使用的特性,也能很好地扩展,应用于其他国产厂家以及国外厂家的FPGA平台。

1 RS编解码器的设计

在伽罗华域 $GF(q)$ 中令 α 为 $GF(q)$ 中的本原元,其生成多项式以 $\alpha, \alpha^2, \dots, \alpha^{2^t}$ 为其全部的根,则

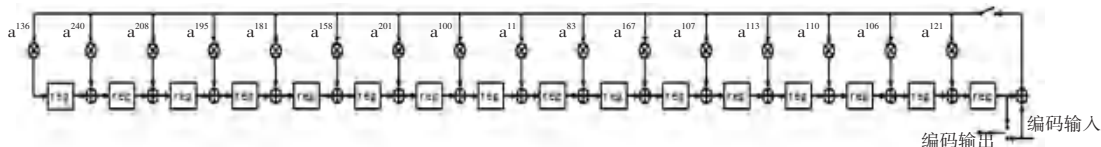


图1 RS编码器电路结构图

Fig. 1 Circuit structure of RS encoder

1.2 RS解码器的设计

为了减少硬件实现的复杂度,节省硬件资源,在RS解码器设计中也使用了乘法器因子矩阵的计算

生成多项式:

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{2^t}) \quad (1)$$

其最小距离 $d_{\min} = 2t + 1$,该RS码能够纠正的错误小于等于最大纠错数 t 。

本文采用RS(36,20)码进行RS编解码器电路的设计与实现。伽罗华域为 $GF(2^8)$,本原多项式:

$$f(x) = x^8 + x^4 + x^3 + x^2 + 1 \quad (2)$$

使用Matlab辅助计算,易得出生成多项式:

$$g(x) = x^{16} - 118x^{15} + 52x^{14} - 103x^{13} + 31x^{12} - 104x^{11} + 126x^{10} - 187x^9 + 232x^8 - 17x^7 + 56x^6 - 183x^5 + 49x^4 - 100x^3 + 81x^2 - 44x + 79 \quad (3)$$

该RS码的码长 $n = 36$,消息长度 $k = 20$,最大可纠正 $(n - k)/2 = 8$ 个符号错误。

1.1 RS编码器的设计

RS(36,20)编码是将20个字节的输入数据进行移位后模 $g(x)$ 的运算,最后编码后的输出数据为36字节带有16字节的校验信息。

RS(36,20)编码器电路结构图如图1所示,主要由线性反馈移位寄存器(LFSR)、有限域乘法器、有限域加法器组成。

采用乘法器因子矩阵的计算方式,即通过公式推导,将16个常数有限域乘法器化简为简单的异或运算,避免使用通用有限域乘法器,节省了乘法器资源,降低了硬件复杂度,便于在各种FPGA平台间进行移植。例如设输入为8 bit的 x 、输出为8 bit的 y ,输出前的最后一级乘法可以简化为:

$$y[0] = x[2] \oplus x[3] \oplus x[4] \oplus x[7]$$

$$y[1] = x[0] \oplus x[3] \oplus x[4] \oplus x[5]$$

$$y[2] = x[0] \oplus x[1] \oplus x[2] \oplus x[3] \oplus x[5] \oplus x[6] \oplus x[7]$$

$$y[3] = x[1] \oplus x[6]$$

$$y[4] = x[0] \oplus x[3] \oplus x[4]$$

$$y[5] = x[0] \oplus x[1] \oplus x[4] \oplus x[5]$$

$$y[6] = x[0] \oplus x[1] \oplus x[2] \oplus x[5] \oplus x[6]$$

$$y[7] = x[1] \oplus x[2] \oplus x[3] \oplus x[6] \oplus x[7]$$

方式对硬件实现进行了优化。常见的解码器的结构框图如图2所示。

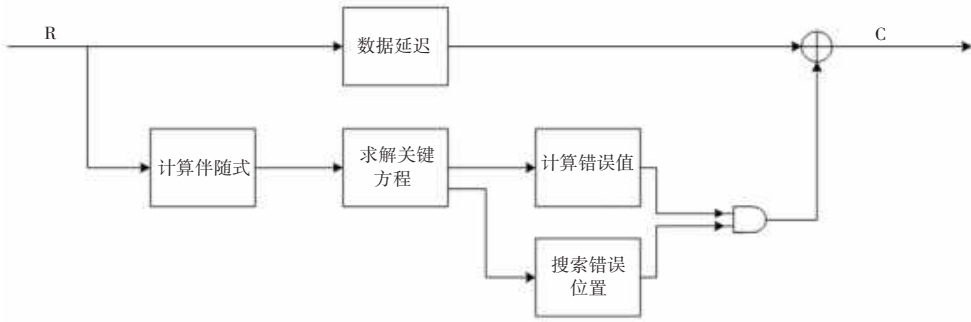


图 2 RS 解码器结构框图

Fig. 2 Structure diagram of RS decoder

解码的过程:

(1) 根据接收到的码字 $R(x)$ 求出伴随式 S_1, S_2, \dots, S_{2t} , 公式如下:

$$S_i = \sum_{j=0}^{n-1} r_j \alpha^{ij} \quad (4)$$

(2) 由伴随式 S_1, S_2, \dots, S_{2t} 求解关键方程:

$$S(x)\sigma(x) = \omega(x) \bmod x^{2t} \quad (5)$$

其中, $\sigma(x)$ 为错误位置多项式, (x) 为错误估值多项式。

通常求解关键方程需要使用 BM (Berlekamp-Massey) 迭代算法或者 Euclid 迭代算法以及上述两种算法的改进、优化算法。Berlekamp 从代数学的角度出发, 提出了 RS 码的迭代译码算法, 避免了 Peterson 算法中繁琐的矩阵运算, 线性反馈移位寄存器综合也有与之相同的结论, 后人统称为 BM 算法^[15]。BM 迭代算法的迭代过程是基于自回归滤波器原理来求解最短反馈连接多项式的过程。Euclid 迭代算法的实质是通过求解两个多项式的最大公因式来获取错误位置和错误值多项式^[16]。Euclid 迭代算法需进行除法运算、需记录大量中间值, 解码速度比 BM 算法慢。BM 算法的缺点是需要使用求逆运算, 运算效率低且电路实现复杂。不含求逆运算的改进 BM 算法也被称为 iBM 算法, 在此基础上进行优化, 采用脉动阵列实现的改进算法被称为 riBM 算法^[17]。本设计使用改进的 BM 算法, 相比传统 BM 算法使用更少的乘法器, 其求解关键方程的步骤如下:

初始化:

$$\lambda_0(0) = b_0(0) = 1$$

$$\lambda_i(0) = b_i(0) = 0, (i = 0, 1, \dots, t)$$

$$k(0) = 0, \gamma(0) = 1$$

输入值为:

$$s_i, (i = 0, 1, \dots, 2t - 1)$$

$$\hat{\delta}_i(0) = \hat{\theta}_i(0) = s_i, (i = 0, 1, \dots, 2t - 1)$$

r 从 0 到 $2t - 1$ 循环依次执行下列步骤:

$$\lambda_i(r + 1) = \gamma(r)\lambda_i(r) - \hat{\delta}_0(r)b_{i-1}(r), (i = 0, 1, \dots, t)$$

$$\hat{\delta}_i(r + 1) = \gamma(r)\hat{\delta}_{i+1}(r) - \hat{\delta}_0(r)\hat{\theta}_i(r), (i = 0, 1, \dots, 2t - 1)$$

如果 $\hat{\delta}_0(r) \neq 0$ 且 $k(r) \geq 0$ 则

$$b_i(r + 1) = \lambda_i(r), (i = 0, 1, \dots, t)$$

$$\hat{\theta}_i(r + 1) = \hat{\delta}_{i+1}(r), (i = 0, 1, \dots, 2t - 1)$$

$$\gamma(r + 1) = \hat{\delta}_0(r)$$

$$k(r + 1) = -k(r) - 1$$

否则

$$b_i(r + 1) = b_{i-1}(r), (i = 0, 1, \dots, t)$$

$$\hat{\theta}_i(r + 1) = \hat{\theta}_i(r), (i = 0, 1, \dots, 2t - 1)$$

$$\gamma(r + 1) = \gamma(r)$$

$$k(r + 1) = k(r) + 1$$

最后输出结果:

$$\lambda_i(2t), (i = 0, 1, \dots, t)$$

$$\omega_i^{(h)}(2t) = \hat{\delta}_i(2t), (i = 0, 1, \dots, t - 1)$$

(3) Chien^[18] 在 1964 年提出了一种搜索算法用于求出错误位置, 被称为钱氏搜索法, 在工程上得到了广泛运用; Forney^[19] 在 1966 年提出了一种用于错误位置多项式 $\sigma(x)$ 和错误估值多项式 $\omega(x)$ 来求解错误值的算法, 被称为 Forney 算法, 大量应用于 RS 解码过程中的最后一步。在求解出错误值后, 将接收数据延迟后与对应位置的错误值进行运算求得纠正后的数据输出:

如果 $\sigma(\alpha^i) = 0$ 则

$$C_{255-i} = R_{255-i} + \omega(\alpha^i) / \sigma'(\alpha^i) \quad (6)$$

其中, $i = 1, 2, \dots, 255$; R 为接收到的码字; C 为纠错后的码字。

2 具有 BIST 功能的 RS 编解码器实现与功能验证

2.1 RS 编解码器的实现

为了便于测试实现的模块功能是否正常、方便在不同的 FPGA 平台移植时能够快速验证模块功能是否正确,引入了伪随机序列(PRBS)产生模块,用于构建一个具有内建自测试功能(BIST)的 RS 编解码器。线性反馈移位寄存器(LFSR)是构造密钥流生成器的最重要部件之一,序列硬件实现简单、密钥

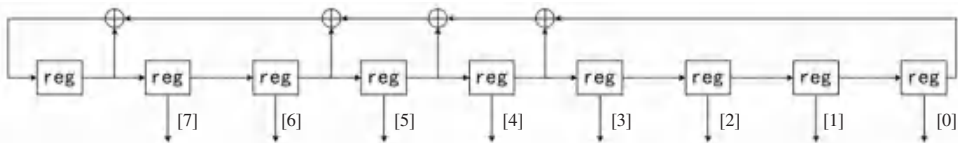


图3 PRBS产生模块结构图

Fig. 3 Structure diagram of PRBS generation module

生成速率高、便于进行密钥安全强度分析^[20]。本文采用LFSR序列来实现伪随机序列产生模块。

本设计中使用的反馈多项式为 $f(x) = x^8 + x^4 + x^3 + x^2 + 1$ 的m序列发生器来产生伪随机序列,使用简单的移位寄存器和加法器即能实现,其结构图如图3所示。

具有BIST功能的RS编解码器结构如图4所示,通过增加噪声掺杂模块、比较判决模块、两个开关和两个PRBS模块实现RS编解码器的内建自测试功能。

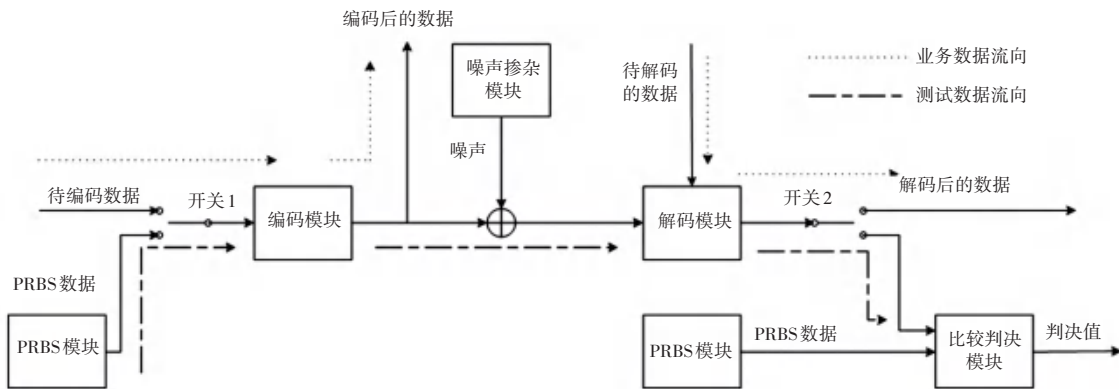


图4 具有BIST功能的RS编解码器结构图

Fig. 4 Structure diagram of RS codec with BIST function

(1) RS编解码器的业务功能启用时,待编码数据通过开关1选择后进入编码模块,输出编码后的数据;待解码数据进入解码模块后通过开关2选择后输出解码后的数据。

(2) RS编解码器的内建自测试功能启用时,PRBS数据通过开关1选择进入编码模块,对输出的编码后的数据进行噪声掺杂并输入解码模块;解码模块输出的纠正错误后的数据通过开关2选择进入比较判决模块,同时PRBS模块输出的PRBS数据进入比较判决模块,将纠正错误后的数据中前20字节信息位与PRBS模块产生的伪随机序列数据进行比较,如果有误则输出高电平,否则输出低电平。

2.2 资源比较

在国产FPGA平台安路EG4和PH1A100上实现上述RS编解码器,同时在XILINX XC7K325TFFG900平台上对本文设计的IP核进行了移植,用XILINX提

供的IP核进行了实现,实现的编码器和解码器的资源使用情况见表1。

表1 RS编解码器实现的资源使用情况

Table 1 Resource utilization of RS codec implementation

资源类型	EG4	PH1A100	325T	325T(XILINX IP核)
LUT	3 588	1 308	2 064	8 837
FF	1 336	904	1 275	9 135
BRAM	1	1	0.5	1

本文的RS编解码器设计在国产FPGA平台安路EG4和PH1A100以及XILINX的7K325T上实现所使用的资源较少,分别仅占用总逻辑资源的18.31%、1.88%和0.67%。在XILINX的325T上本文的设计实现的RS编解码器也要比XILINX提供的IP核资源少很多,所以即使所选用的FPGA平台提供了RS编解码器的IP核,在考虑资源优化的情况下仍可采用本文的设计。

使用 Modelsim 对本文设计的 RS 编解码模块进行行为级仿真,展示该模块的工作流程并验证功能是否符合预期。

2.3 RS 编解码器业务功能模式的验证

使用业务功能模式时 RS 编码模块的输入和输出码字如图 5 所示,使能(enable)信号高电平表示 RS 编码模块使能,输入数据信号 enc_in 信号为从 ROM 中读取的待编码数据,信息位指示(data)信号高电平表示此时的 enc_in 数据为 RS 编码码字的信息位。RS 编码码字的信息位为“1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20”,输出的 RS 编码后码字为“1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、40、26、9、227、115、23、232、156、179、251、44、142、200、12、50、33”,其中 16 个字节数据“40、26、9、227、115、23、232、156、179、251、44、142、200、12、50、33”为编码加入的校验位。使用 Matlab 程序对以上数据进行验证,结果正确。

对上述 RS 编码模块的输出码字进行噪声掺杂,可以随机对第 1、6、7、12、19、27、32、36 个位置的码字分别加入噪声 8'd11、8'd22、8'd33、8'd44、

8'd55、8'd66、8'd77、8'd88,RS 解码模块的输入即为“10、2、3、4、5、16、38、8、9、10、11、32、13、14、15、16、17、18、36、20、40、26、9、227、115、23、170、156、179、251、44、195、200、12、50、121”,RS 解码模块的输入码字如图 6 所示。

RS 解码模块的输出码字如图 7 所示,RS 解码模块对于输入“10、2、3、4、5、16、38、8、9、10、11、32、13、14、15、16、17、18、36、20、40、26、9、227、115、23、170、156、179、251、44、195、200、12、50、121”,错误值 error 信号为计算出的错误值 8'd11、8'd22、8'd33、8'd44、8'd55、8'd66、8'd77、8'd88,错误位置为码字第 1、6、7、12、19、27、32、36 个位置,计算出的错误值与输入码字有限域相加得到纠正错误后的 RS 解码模块输出码字“1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、40、26、9、227、115、23、232、156、179、251、44、142、200、12、50、33”。此时解码模块的输出码字与 RS 编码模块的输出码字即未掺杂噪声的码字一致,说明解码模块检测到并纠正了 8 个错误,满足了 RS 码(36,20)最大纠正 8 个错误的设计需求。



图 5 RS 编码模块的输入和输出码字

Fig. 5 Input and output codewords of RS encoding module



图 6 RS 解码模块的输入码字

Fig. 6 Input codeword of RS decoding module



图 7 RS 解码模块的输出码字

Fig. 7 Output codeword of RS decoding module

2.4 RS 编解码器测试功能模式的验证

RS 编码模块的输入和输出码字如图 8 所示,随机序列输入信号 enc_in_switch 为 PRBS 模块产生的随机序列,作为 RS 编码的信息位“81、40、A0、50、A8、D4、6A、35、1A、8D、46、A3、51、28、94、4A、A5、52、29、14”,输出的 RS 编码后码字为“81、40、A0、50、A8、D4、6A、35、1A、8D、46、A3、51、28、94、4A、

A5、52、29、14、D4、DB、C3、6D、B7、2F、09、80、D0、30、CA、DC、2B、A8、3D、62”,其中 16 个字节数据“D4、DB、C3、6D、B7、2F、09、80、D0、30、CA、DC、2B、A8、3D、62”为编码加入的校验位。使用 Matlab 程序对以上数据进行验证,结果正确。

对上述 RS 编码模块的输出码字进行噪声掺杂,可以随机对 8 个位置的码字分别加入噪声

8'h21, 8'h2C, 8'h37, 8'h42, 8'h4D, 8'h58, 8'h0B, 8'h16, 此时掺杂的码字位置为第 4、8、12、15、20、25、31、34 位, RS 解码模块的输入即为“81、40、A0、71、A8、D4、6A、19、1A、8D、46、94、51、28、D6、4A、A5、52、29、59、D4、DB、C3、6D、EF、2F、09、80、D0、30、C1、DC、2B、BE、3D、62”, RS 解码模块的输入码字如图 9 所示。

RS 解码模块的输出码字如图 10 所示, RS 解码模块对于输入“81、40、A0、71、A8、D4、6A、19、1A、8D、46、94、51、28、D6、4A、A5、52、29、59、D4、DB、C3、6D、EF、2F、09、80、D0、30、C1、DC、2B、BE、3D、62”, error 信号为计算出的错误值 8'h21, 8'h2C, 8'h37, 8'h42, 8'h4D, 8'h58, 8'h0B, 8'h16, 错误位置为码字第 4、8、12、15、20、25、31、34 个位置, 计算出的错误值与输入码字有限域相加得到纠正错误后的 RS 解码模块输出码字“81、40、A0、50、A8、D4、

6A、35、1A、8D、46、A3、51、28、94、4A、A5、52、29、14、D4、DB、C3、6D、B7、2F、09、80、D0、30、CA、DC、2B、A8、3D、62”。此时解码模块的输出码字与上述 RS 编码模块的输出码字即未掺杂噪声的码字一致, 即解码模块检测到并纠正了 8 个错误, 满足了 RS 码 (36, 20) 最大纠正 8 个错误的设计需求。

RS 解码模块的输出与 PRBS 模块的输出码字如图 11 所示, 纠正错误后的 RS 解码模块输出码字与 PRBS 模块生成的伪随机序列一致, PRBS 比较结果 prbs_compare 信号保持低电平。

对 BIST 功能进行多次连续测试, RS 解码模块的输出码字中的信息位均与 PRBS 模块的输出码字一致, prbs_compare 信号一直保持低电平, 说明此时 RS 编解码器工作正常, BIST 功能的多次连续测试如图 12 所示。



图 8 RS 编码模块的输入和输出码字

Fig. 8 Input and output codewords of RS encoding module



图 9 RS 解码模块的输入码字

Fig. 9 Input codeword of RS decoding module



图 10 RS 解码模块的输出码字

Fig. 10 Output codeword of RS decoding module



图 11 RS 解码模块的输出与 PRBS 模块的输出码字

Fig. 11 Output codewords of the RS decoding module and the PRBS module



图 12 BIST 功能的多次连续测试

Fig. 12 Multiple consecutive tests of BIST function

3 结束语

随着国产 FPGA 芯片的百花齐放, 使用国产化 FPGA 的需求也越来越多。不同的国产 FPGA 厂家的 EDA 工具或有不同、对常用 IP 的支持各有区别, 本文设计了一种低复杂度、低资源使用的 RS 编码器和 RS 解码器, 同时实现了内建自测试 (Build-in Self-Test) 功能, 采用乘法器因子矩阵的计算方式以及改进的 BM 算法, 降低了硬件实现复杂度和资源使用率, 满足了不同的 FPGA 平台使用 RS 编解码器的需求, 降低了移植的难度和验证的工作量。最后通过 Matlab 和 Modelsim 的仿真验证, 展示了 RS 编解码器的工作流程, 并验证了所设计的 RS 编解码器的功能符合设计需求。

参考文献

- [1] LEE H. A high-speed low-complexity Reed-Solomon decoder for optical communications [J]. IEEE Transactions on Circuits And Systems-II: Express Briefs, 2005, 52(8): 461-465.
- [2] 刘文国. 基于 FPGA 的 RS(255,223) 编解码器的高速并行实现 [D]. 成都: 电子科技大学, 2009.
- [3] HANG Pingping, LI Jinming. Low power design and ASIC research of RS code [J]. Journal of Measurement Science and Instrumentation, 2023, 14(2): 156-163.
- [4] 张翔. 无线光通信中的编码与调制技术研究及应用 [D]. 北京: 北京邮电大学, 2015.
- [5] LEE H. High-speed VLSI architecture for parallel Reed-Solomon decoder [J]. IEEE Transactions on Very Large Scale Integration Systems, 2003, 11(2): 288-294.
- [6] 刘小梅, 唐鑫, 杨舒婷, 等. 基于 Reed-Solomon 编码的抗边信道攻击云数据安全去重方法 [J]. 信息安全学报, 2022, 7(6): 81-93.
- [7] 李芳培, 荣为君, 袁承宗, 等. 基于 RS 码的某运载火箭图像测量系统设计与实现 [J]. 计算机测量与控制, 2018, 26(11): 89-92.
- [8] 朱安周, 靳志广, 段涛, 等. RS 码在战术数据链系统中的性能分析 [J]. 信息通信, 2018, 8, 188: 43-46.
- [9] 韦春雷, 吴新春, 黄孝兵. 用于 400Gb/s 以太网 PCS 的递推 RS 编码电路 [J]. 光通信技术, 2024, 48(1): 71-73.
- [10] 严紫薇, 陈少华, 张宝朋. 采用 RS 编码的改进高速数据传输平台实现 [J]. 单片机与嵌入式系统应用, 2021, 21(5): 79-83.
- [11] 曾鹏, 张志宇, 邓建晖. 基于 PLC 的 RS 编解码器设计与实现 [J]. 智能计算机与应用, 2019, 9(2): 50-53.
- [12] 张欣伟, 王鹏. 基于 FPGA 的引信感应装定 RS 编码器设计 [J]. 国外电子测量技术, 2019, 38(1): 112-115.
- [13] 汪智杰, 周治柱. 基于国产 FPGA 的 RS 编解码器设计与应用 [J]. 电子设计工程, 2022, 30(22): 140-144.
- [14] 程小枫, 柳松, 顾苏, 等. 基于软件无线的 RS 纠错码的实现 [J]. 航天电子对抗, 2018, 34(1): 35-39.
- [15] 曾晓洋. 高性能 Reed-Solomon 码译码方法及其相关技术的研究 [D]. 长春: 中国科学院长春光学精密机械与物理研究所, 2001.
- [16] 龚政辉. RS 码高速译码实现及其软判决译码算法的研究 [D]. 长沙: 国防科学技术大学, 2012.
- [17] SARWATE D V, SHANBHAG N R. High-speed architectures for Reed-Solomon decoders [J]. IEEE Transactions on Very Large Scale Integration Systems, 2001, 9(5): 641-655.
- [18] CHIEN R T. Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes [J]. IEEE Transactions on Information Theory, 1964, 10(4): 357-363.
- [19] FORNEY G. Generalized minimum distance decoding [J]. IEEE Transactions on Information Theory, 1966, 12(2): 125-131.
- [20] 马将. 伪随机序列的构造及其随机性研究 [D]. 秦皇岛: 燕山大学, 2023.