Journal of Computer Applications
计算机应用,2012,32(3):867-869,873
ISSN 1001-9081
CODEN JYIIDU
2012-03-01
http://www.joca.cn

文章编号:1001-9081(2012)03-0867-03
doi:10.3724/SP.J.1087.2012.00867

并行高效 BCH 译码器设计及 FPGA 实现

张湘贤*,杨 涛,魏东梅,向 玲
(西南科技大学 信息工程学院,四川 绵阳 621010)
(*通信作者电子邮箱 source03@163.com)

摘 要:针对并行 BCH 译码器的特点,采用异或门实现有限域上常系数乘法,从而降低硬件复杂度。先计算部分错误位置多项式,再根据仿射多项式和格雷码理论,进行逻辑运算得到剩余的错误位置多项式,从而减少了系统所占用的资源。在现场可编程门阵列(FPGA)开发软件 ISE10.1 上进行了时序仿真,验证了该算法时间和空间的高效性。

关键词:现场可编程门阵列;BCH 译码器;仿射多项式;格雷码
中图分类号:TN919 **文献标志码**:A

Design and FPGA implementation of parallel high-efficiency BCH decoder

ZHANG Xiang-xian*, YANG Tao, WEI Dong-mei, XIANG Ling
(*School of Information Engineering, Southwest University of Science and Technology, Mianyang Sichuan* 621010, *China*)

Abstract: According to the characteristics of parallel BCH decoder, the multiplication of constant coefficient in finite field was realized by using XOR gates to reduce hardware complexity. The part of the error location polynomial was calculated, and then the remaining error location polynomial could be obtained using the theory of affine polynomial and Gray code. The proposed algorithm reduces the system resources occupied. Through timing simulation on Field Programmable Gate Array (FPGA)'s development software ISE10.1, the high-efficiency of the algorithm on time and space has got verified.

Key words: Field Programmable Gate Array (FPGA); BCH decoder; affine polynomial; Gray code

0 引言

BCH(Bose, Chaudhuri and Hocquenghem)码是一类重要的循环码,能纠正多个随机错误,具有构造方便、编码简单、纠错能力强等特点,在编码理论中具有重要地位。采用硬件实现译码过程,要求译码方法综合考虑到运算速度和占用的资源面积。

目前已有的采用并行方式实现 BCH 译码器存在占用资源多等不足。金婕等[1]提出的并行 BCH 译码器结构要依靠时钟进行同步且部分运算需要多个周期。孙怡等[2]用并行方式实现时要用到多个 FLEX10K 芯片。

本文介绍一种改进的并行 BCH 译码器设计方法,在现场可编程门阵列(Field Programmable Gate Array, FPGA)中采用少量异或门实现有限域常系数乘法运算,硬件复杂度低,占用资源少,结合仿射多项式与格雷码,使用扩展的并行钱(Chien)搜索方法,分两步计算错误位置多项式,能大大减少系统所占用的资源。

1 BCH 码译码的一般方法

基于有限域 $GF(2^m)$ 的 BCH(n,k,t),码长 $n=2^m-1$,信息位长 $k=n-mt$,纠错能力为 t。BCH 码译码一般分为以下几个步骤:1) 由接收到的 $\boldsymbol{r}(x)$ 计算出伴随式 $\boldsymbol{S}$;2) 由伴随式 $\boldsymbol{S}$ 得到错误位置多项式 $\boldsymbol{\sigma}(x)$;3) 求解错误位置多项式 $\boldsymbol{\sigma}(x)$ 确定错误位置;4) 根据错误位置进行纠错[3]。

设接收数据为:

$$\boldsymbol{r}(x)=\boldsymbol{r}_{n-1}x^{n-1}+\boldsymbol{r}_{n-2}x^{n-2}+\cdots+\boldsymbol{r}_1x+\boldsymbol{r}_0=\boldsymbol{v}(x)+\boldsymbol{e}(x) \tag{1}$$

其中:$\boldsymbol{v}(x)$ 和 $\boldsymbol{e}(x)$ 分别为码多项式和错误图样多项式。接收矢量的伴随式可由接收数据与校验矩阵相乘得到:

$$\boldsymbol{S}=\boldsymbol{r}\cdot\boldsymbol{H}^{\mathrm{T}}=(S_1,S_2,\cdots,S_{2t}) \tag{2}$$

另外有 $S_2=S_1^2,S_4=S_2^2,S_6=S_3^2,\cdots$

令错误位置多项式为:

$$\boldsymbol{\sigma}(x)=(1+\boldsymbol{\beta}_1x)(1+\boldsymbol{\beta}_2x)\cdots(1+\boldsymbol{\beta}_tx)=\boldsymbol{\sigma}_0+\boldsymbol{\sigma}_1x+\boldsymbol{\sigma}_2x^2+\cdots+\boldsymbol{\sigma}_tx^t \tag{3}$$

伴随式与错误位置多项式的关系可用如下矩阵表示:

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_t \\ S_2 & S_3 & \cdots & S_{t+1} \\ \vdots & \vdots & & \vdots \\ S_t & S_{t+1} & \cdots & S_{2t-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{\sigma}_t \\ \boldsymbol{\sigma}_{t-1} \\ \vdots \\ \boldsymbol{\sigma}_1 \end{bmatrix}=\begin{bmatrix} S_{t+1} \\ S_{t+2} \\ \vdots \\ S_{2t} \end{bmatrix} \tag{4}$$

由式(4)有:

$$\boldsymbol{\sigma}_1=S_1$$
$$\boldsymbol{\sigma}_2=S_2+S_3/S_1$$
$$\boldsymbol{\sigma}_3=(S_3^2+S_1^6+S_3S_1^3+S_1S_5)/(S_3+S_1^3)$$
$$\cdots$$

采用钱搜索方法求解 $\boldsymbol{\sigma}(x)=0$,得到 $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_t$,即求得错误位置。根据错误位置把输入数据的对应位取反即完成纠错。

2 BCH 码译码的并行实现方法

以 BCH(15,7)为例说明具体实现方法,此种格式时最多

收稿日期:2011-07-12;**修回日期**:2011-11-22。
基金项目:国防科工局核能开发科研项目(20111108-01);四川省科技计划项目(2010GZ0199)。
作者简介:张湘贤(1985-),男,湖南长沙人,硕士研究生,主要研究方向:FPGA 重配置设计; 杨涛(1972-),男,四川三台人,教授,博士,主要研究方向:机电系统仿真与控制; 魏东梅(1974-),女,四川绵阳人,讲师,硕士,主要研究方向:信息安全; 向玲(1987-),女,重庆人,硕士研究生,主要研究方向:阵列信号处理。

可以纠正两位错($t = 2$),可将此方法扩展到更长数据位和纠正更多位错误的应用之中。

2.1 计算伴随式 S_1 和 S_3

当 $t = 2$,式(2)可写成:

$$\boldsymbol{H} = \begin{bmatrix} 1 & \boldsymbol{a} & \boldsymbol{a}^2 & \cdots & \boldsymbol{a}^{14} \\ 1 & \boldsymbol{a}^3 & \boldsymbol{a}^6 & \cdots & \boldsymbol{a}^{12} \end{bmatrix} \tag{5}$$

可以把式(5)分为两部分,$\boldsymbol{H}_1$ 表示第一行,$\boldsymbol{H}_3$ 表示第二行。$GF(2^4)$ 中所有域元素在表1中列出,如果用 $GF(2^4)$ 上对应的二进制四重数组表示 $\boldsymbol{H}_1$ 和 $\boldsymbol{H}_3$,则两个矩阵都是4×15的矩阵。根据式(2)则有:

$$\boldsymbol{S}_1 = \boldsymbol{r} \cdot \boldsymbol{H}_1^{\mathrm{T}}$$

$$\boldsymbol{S}_3 = \boldsymbol{r} \cdot \boldsymbol{H}_3^{\mathrm{T}}$$

结合各域元素的表示式,只要用少量异或门就可实现计算伴随式 $\boldsymbol{S}$[4],图1所示为用异或门实现计算 $\boldsymbol{S}_3$。与文献[5]提出的伴随式计算电路要用到一个 $p \times 2tm$ 维乘法器和多个 $m \times m$ 维乘法器,文献[6]完成伴随式计算需要 n 个周期相比,采用此方法实现计算伴随式在速度和资源使用方面更有优势。

表1 $GF(2^4)$ 的域元素表示式

幂多项式	二进制表示值	幂多项式	二进制表示值
1	0001	$\boldsymbol{a}^8$	0101
$\boldsymbol{a}$	0010	$\boldsymbol{a}^9$	1010
$\boldsymbol{a}^2$	0100	$\boldsymbol{a}^{10}$	0111
$\boldsymbol{a}^3$	1000	$\boldsymbol{a}^{11}$	1110
$\boldsymbol{a}^4$	0011	$\boldsymbol{a}^{12}$	1111
$\boldsymbol{a}^5$	0110	$\boldsymbol{a}^{13}$	1101
$\boldsymbol{a}^6$	1100	$\boldsymbol{a}^{14}$	1001
$\boldsymbol{a}^7$	1011	$\boldsymbol{a}^{15}$	0001

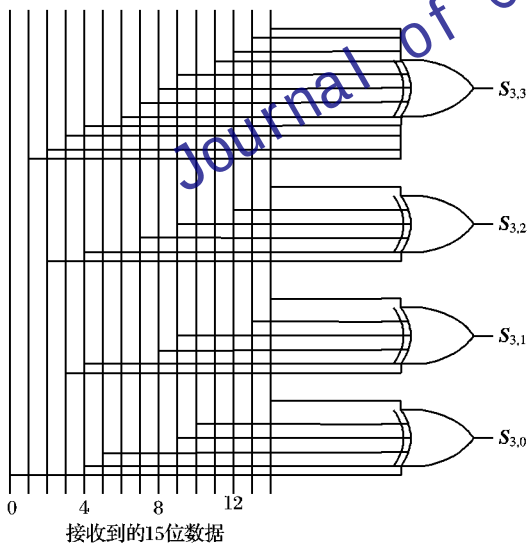

图1 用异或门实现计算 $\boldsymbol{S}_3$

2.2 计算错误多项式 $\sigma(x)$

根据 $\boldsymbol{S}_1$、$\boldsymbol{S}_2$ 和 $\boldsymbol{S}_3$,得到错误多项式:

$$\boldsymbol{\sigma}(x) = (\boldsymbol{S}_2 + \boldsymbol{S}_3/\boldsymbol{S}_1)\boldsymbol{x}^2 + \boldsymbol{S}_1\boldsymbol{x} + 1 \tag{6}$$

可以将上式变形为:

$$\boldsymbol{\sigma}(x) = (\boldsymbol{S}_3 + \boldsymbol{S}_1^3)\boldsymbol{x}^2 + \boldsymbol{S}_1^2\boldsymbol{x} + \boldsymbol{S}_1 \tag{7}$$

将逻辑代数化简,用一些简单的逻辑门电路就可以由 $\boldsymbol{S}_1$ 得到 $\boldsymbol{S}_1^3$ 和 $\boldsymbol{S}_1^2$。采用式(7)在计算 $\boldsymbol{\sigma}(\boldsymbol{x})$ 时避免了除法运算,可以提高运算速度,减少逻辑资源。

2.3 有限域常系数乘法计算

在求解 $\boldsymbol{\sigma}(x)$ 时,一般要用有限域乘法器(Finite Field Multiplier, FFM)计算 $\boldsymbol{\sigma}_j\boldsymbol{a}^i$。根据有限域 $GF(2^m)$ 上运算的特性,$\boldsymbol{\sigma}_j\boldsymbol{a}^i$ 可以写成如下的矩阵实现:

$$\boldsymbol{\sigma}_j\boldsymbol{a}^i = [\boldsymbol{\sigma}_{j,3} \quad \boldsymbol{\sigma}_{j,2} \quad \boldsymbol{\sigma}_{j,1} \quad \boldsymbol{\sigma}_{j,0}] \begin{bmatrix} \boldsymbol{a}_3^i & \boldsymbol{a}_2^i & \boldsymbol{a}_1^i & \boldsymbol{a}_0^i \\ \boldsymbol{a}_3^{i+1} & \boldsymbol{a}_2^{i+1} & \boldsymbol{a}_1^{i+1} & \boldsymbol{a}_0^{i+1} \\ \boldsymbol{a}_3^{i+2} & \boldsymbol{a}_2^{i+2} & \boldsymbol{a}_1^{i+2} & \boldsymbol{a}_0^{i+2} \\ \boldsymbol{a}_3^{i+3} & \boldsymbol{a}_2^{i+3} & \boldsymbol{a}_1^{i+3} & \boldsymbol{a}_0^{i+3} \end{bmatrix} \tag{8}$$

并不需要专门的乘法器计算式(8)。$\boldsymbol{\sigma}_j\boldsymbol{a}$,$\boldsymbol{\sigma}_j\boldsymbol{a}^2$,… 都可以参照伴随式的计算方法采用异或门实现,对于不同的数据长度结构每一个运算最多用到 m 个 m 输入端的异或门。图2给出了按钱搜索方法利用式(8)计算 $\boldsymbol{a}^5$ 的算式:$\boldsymbol{S}_1^2$ 与 $\boldsymbol{a}^5$ 相乘,$(\boldsymbol{S}_1^3 + \boldsymbol{S}_3)$ 与 $(\boldsymbol{a}^5)^2$ 相乘,采用一个三输入的异或门实现对3个数值求和。若求和结果为零,则 $\boldsymbol{a}^5$ 是位置多项式的一个解,结果为非零时,则不是位置多项式的解[7]340。

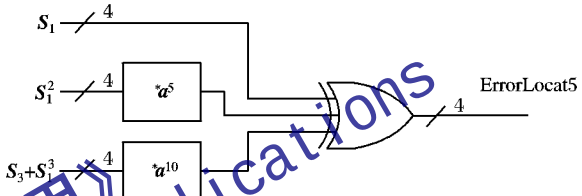

图2 采用钱搜索计算 $\boldsymbol{a}^5$ 结构

2.4 格雷码与 BCH 码相结合

定义1 若

$$\boldsymbol{L}(y) = \sum_i \boldsymbol{L}_i y^{2^i};\ \boldsymbol{L}_i \in GF(2^m) \tag{9}$$

则称表达式 $\boldsymbol{L}(y)$ 是 $GF(2^m)$ 域上的一个 p 表达式,同时也是一个线性表达式[7]341,[8]。

若 $GF(2^m)$ 域上存在表达式 $\boldsymbol{A}(\boldsymbol{y})$ 满足

$$\boldsymbol{A}(\boldsymbol{y}) = \boldsymbol{\varepsilon} + \boldsymbol{L}(\boldsymbol{y});\ \boldsymbol{\varepsilon} \in GF(2^m) \tag{10}$$

称 $\boldsymbol{A}(\boldsymbol{y})$ 为仿射表达式。文献[9]中说明了 p 表达式和仿射表达式的性质。

如果 $GF(2^m)$ 域上的元素按格雷码顺序排列,则各码字之间的汉明距离为1。对于 $GF(2^m)$ 域上元素已经按顺序排列的格雷码,有:

$$\boldsymbol{A}(\boldsymbol{x}_i) = \boldsymbol{\varepsilon} + \boldsymbol{L}(\boldsymbol{x}_i)$$

$$\boldsymbol{A}(\boldsymbol{x}_{i-1}) = \boldsymbol{\varepsilon} + \boldsymbol{L}(\boldsymbol{x}_{i-1})$$

两式相加,得

$$\boldsymbol{A}(\boldsymbol{x}_i) = \boldsymbol{A}(\boldsymbol{x}_{i-1}) + \boldsymbol{L}(\boldsymbol{a}^{\boldsymbol{\delta}(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})}) \tag{11}$$

其中:$\boldsymbol{\delta}(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})$ 表示第 i 个元素 $\boldsymbol{x}_i$ 与第 $(i-1)$ 个元素 $\boldsymbol{x}_{i-1}$ 不相同的那一位。式(11)表明如果已经计算得到 $\boldsymbol{x}_{i-1}$,则可以由 $\boldsymbol{x}_{i-1}$ 仿射得到 $\boldsymbol{x}_i$ 的值。

结合式(7)有:

$$\boldsymbol{\sigma}(\boldsymbol{a}^2) = \boldsymbol{S}_1 + \boldsymbol{S}_1^2\boldsymbol{a}^2 + (\boldsymbol{S}_3 + \boldsymbol{S}_1^3)\boldsymbol{a}^4$$

根据 $\boldsymbol{\sigma}(\boldsymbol{a}^2)$ 就可以得到其他变量的值。求 $\boldsymbol{\sigma}(\boldsymbol{a}^5)$ 和 $\boldsymbol{\sigma}(\boldsymbol{a}^{11})$ 的过程如下:

$GF(2^4)$ 域上各元素只有四位二进制值,结合 $\boldsymbol{a}^0$,$\boldsymbol{a}^1$,$\boldsymbol{a}^2$,$\boldsymbol{a}^3$ 具有独热码(one hot code)的特性,定义如下表达式:

$$\begin{cases} \boldsymbol{L}(\boldsymbol{a}^0) = \boldsymbol{S}_1^2 + (\boldsymbol{S}_3 + \boldsymbol{S}_1^3) \\ \boldsymbol{L}(\boldsymbol{a}^1) = \boldsymbol{S}_1^2\boldsymbol{a}^1 + (\boldsymbol{S}_3 + \boldsymbol{S}_1^3)\boldsymbol{a}^2 \\ \boldsymbol{L}(\boldsymbol{a}^2) = \boldsymbol{S}_1^2\boldsymbol{a}^2 + (\boldsymbol{S}_3 + \boldsymbol{S}_1^3)\boldsymbol{a}^4 \\ \boldsymbol{L}(\boldsymbol{a}^3) = \boldsymbol{S}_1^2\boldsymbol{a}^3 + (\boldsymbol{S}_3 + \boldsymbol{S}_1^3)\boldsymbol{a}^6 \end{cases} \tag{12}$$

表 1 中 $a^2=(0100)$,$a^5=(0110)$,$a^{11}=(1110)$,a^5 与 a^2 是第 1 位不同,根据式(11)有:

$$\sigma(a^5)=\sigma(a^2)+L(a^1)=$$
$$S_1+S_1^2a^2+(S_3+S_1^3)a^4+S_1^2a^1+(S_3+S_1^3)a^2=$$
$$S_1+S_1^2(a^2+a^1)+(S_3+S_1^3)(a^4+a^2)=$$
$$S_1+S_1^2a^5+(S_3+S_1^3)a^{10}$$

同理 a^{11} 与 a^2 是第 3 位和第 1 位不同,有

$$\sigma(a^{11})=\sigma(a^2)+L(a^3)+L(a^1)=$$
$$S_1+S_1^2a^2+(S_3+S_1^3)a^4+S_1+S_1^2a^3+(S_3+S_1^3)a^6+$$
$$S_1+S_1^2a^1+(S_3+S_1^3)a^2=$$
$$S_1+S_1^2(a^2+a^3+a^1)+(S_3+S_1^3)(a^4+a^6+a^2)=$$
$$S_1+S_1^2a^{11}+(S_3+S_1^3)a^7$$

上述是以 a^2 为比较基准,同样也可以以 $GF(2^4)$ 域上其他元素作为比较基准。在选择比较基准时应使表达式的项尽可能少。

按照上述的理论推导,分两步求解 $\sigma(x)$,可以大大减少硬件资源。在图 3 中,第 1 步按照 2.3 节所述方法计算 $\sigma(a^0)$,$\sigma(a^1)$,$\sigma(a^2)$,$\sigma(a^3)$,$\sigma(a^{14})$ 和 $L(a^0)$,$L(a^1)$,$L(a^2)$,$L(a^3)$,$L(a^{14})$,第 2 步根据第 1 步计算的 5 个值分别求出剩下的 10 个值。

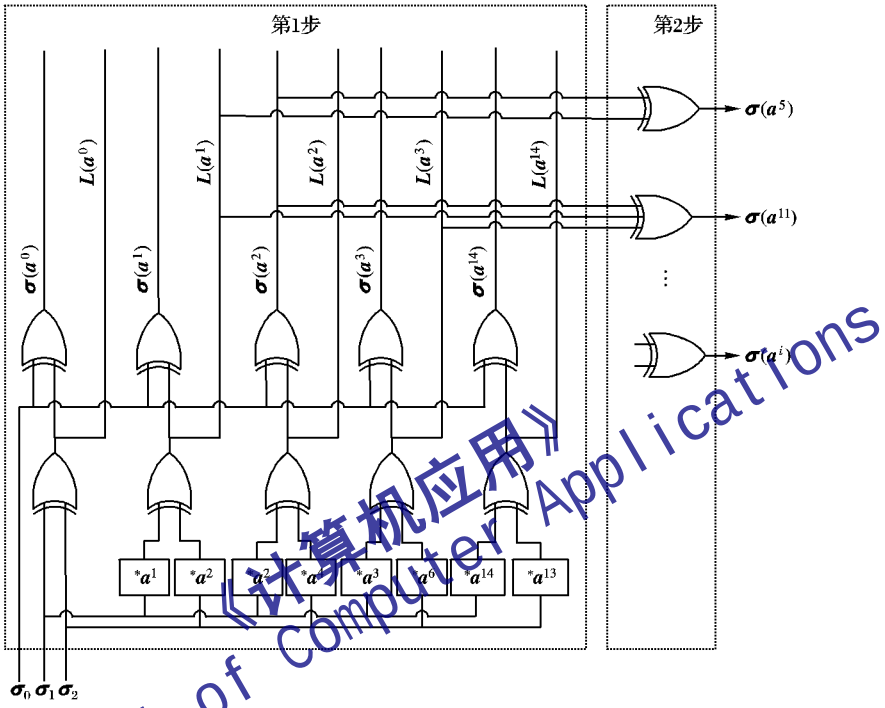

图 3　扩展的并行钱搜索方法

3　基于 FPGA 的综合实现

采用 Xilinx ISE 10.1 软件编程,选中 XC3S400 作为目标器件,进行引脚和时序约束,经过综合后运行时序仿真(Post-Route Simulation),图 4 是 BCH(15,7)结构的时序仿真波形图,图 5 是 BCH(31,21)结构的时序仿真波形图。图中 Received 为接收到的 n 位的数据,$S11:S_1$,$S12:S_1^2$,$S13:S_1^3$,$S3:S_3$,$S313:S_3+S_1^3$,ErrorNum 为接收数据中错误位数,Decoded 为译码输出结果。从仿真结果中可看出:对于BCH(15,7)加入输入信号后经过15ns左右就得到稳定的输出信号,对于

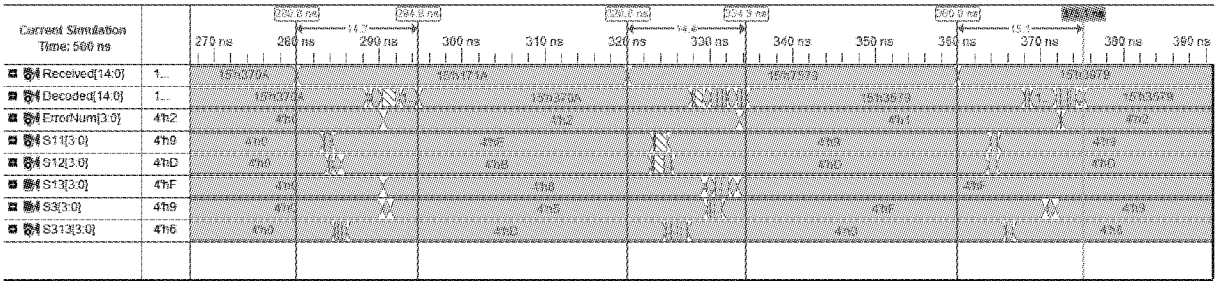

图 4　BCH(15,7)时序仿真波形图

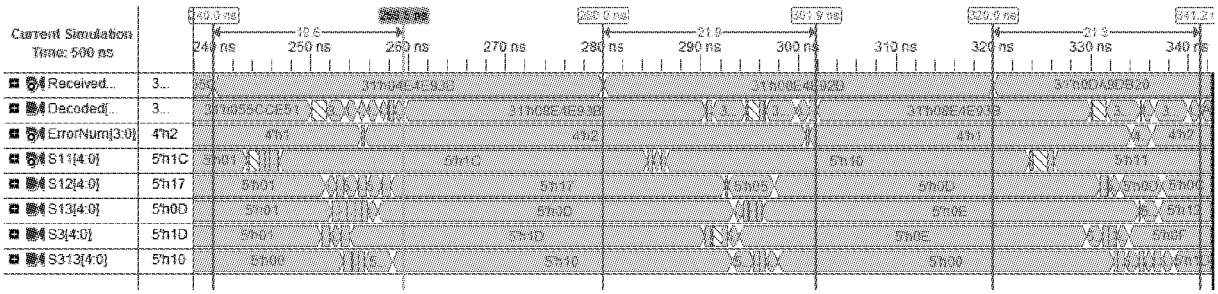

图 5　BCH(31,21)时序仿真波形图

(下转第 873 页)

2) 本文假设数据处理服务时间服从指数分布,如果服务时间服从一般分布,则可建立 $M/G/k/\infty/\infty$ 排队系统模型。

3) 本文假设数据存在有效期限制,建立了有限等待的排队网络模型;如果数据有效期不存在,则可建立无限等待的排队网络模型。

4) 本文假设系统不存在缓冲区限制,即队长无限。如果队长有限,则建立 $M/M/k/N/\infty$ 排队网络模型。

6　结语

本文针对空间信息数据处理系统效能评估展开研究,首先分析了系统效能评估的核心指标体系,基于排队网络理论,建立了效能评估模型-两级多服务台排队网络服务模型,提出了模型求解方法,结合算例验证了方法的正确性和有效性。由于方法相对其他方法具有简单、省时、费用低的特点,因此具有很强的实用价值。

参考文献:

[1] 安雪滢,赵勇,杨乐平,等. 基于模糊理论的卫星系统效能评估仿真研究[J]. 系统仿真学报, 2006, 18(8):2334 -2337.

[2] 冯书兴. 卫星系统综合效能分析研究[J]. 控制与决策, 2003, 11(6):740 -743.

[3] 陈浩光,秦大国,李云芝. 军用卫星系统效能评估的基本原则与方法研究[J]. 装备指挥技术学院学报, 2001, 12(2): 27 -30.

[4] FEUERSTEIN P, ERIC P H, MATTHEW L G, *et al*. Generalized representation of space-based platform for various orbit type [C]// Advanced Simulation Technologies Conference — Military, Government, and Aerospace Simulation Symposium (ASTC-MGA). Crystal City, Virginia: The Society for Modeling and Simulation International, 2004.

[5] 刘华翔,黄俊,朱荣昌. 综合航空武器平台作战效能评估综述[J]. 系统工程学报, 2003, 18(1): 55 -61.

[6] 郭玉华. 多类型对地观测卫星联合任务规划关键技术研究[D]. 长沙: 国防科学技术大学, 2009.

[7] 丁洪波,张士峰,胡正东. 支持作战的天基信息系统效能分析的排队论模型[J]. 火力与指挥控制, 2008, 33(4):79 -82.

[8] 刘亮亮,狄东记,李华. 空间信息支援下装甲部队侦察系统作战效能评估[J]. 兵工自动化, 2010, 29(2): 23 -27.

[9] ANDREW L L H, HANLY S V, MUKHTAR R G. Active queue management for fair resource allocation in wireless networks [J]. IEEE Transactions on Mobile Computing, 2008, 7(2): 220 -247.

[10] DOVROLIS C, STILIADIS D, RAMANATHAN P. Proportional differentiated services delay differentiation and packet scheduling [J]. IEEE/ACM Transactions on Networking, 2002, 10(1): 12 -26.

[11] 张正,刘景泰,王鸿鹏. 基于排队网络的网络服务器性能分析与优化[J]. 计算机应用, 2010, 30(12): 3148 -3152.

[12] 张英,赵莉茹,谷新亮. 基于排队网络的多业务网络资源分析方法[J]. 计算机应用, 2009, 29(9): 2425 -2431.

[13] 谢广军,刘军,王刚,等. 基于多类顾客排队网络的 Exp-RAID 系统性能评价模型[J]. 计算机研究与发展, 2008, 45(2): 207 -211.

[14] 文江平. 卫星军事应用技术[M]. 北京: 国防工业出版社, 2007: 12 -15.

[15] HAN S M, BEAK S W, CHO K R, *et al*. Satellite mission scheduling using gGenetic algorithm [C]// International Conference on Instrumentation, Control, Information Technology and System Integration. Piscataway, NJ: IEEE Press, 2008: 1226 -1230.

[16] 宋斌,牛存良,熊友奇. 防空作战模拟与效能评估[M]. 北京: 军事科学出版社, 2005: 76 -78.

(上接第 869 页)

BCH(31,21)译码过程为 20 ns 左右。文献[10]采用串行与并行相结合的方式实现 BCH(15,11)译码,纠正一位错,在 XC3S400 上实现时间为 73 ns。

根据第 2 章的分析可知,译码过程的所有运算都可以由逻辑门用组合逻辑形式实现,表 2 列出了两种数据长度情况下 FPGA 的资源使用情况。

表 2　两种译码器使用资源

逻辑资源类别	BCH(15,7)译码器使用量	BCH(31,21)译码器使用量
4 输入查找表(LUT:Look-Up Table)	143	333
基本逻辑单元(slice)	77	170
输入输出单元	34	66

4　结语

本设计中完全采用并行结构,数据是并行输入并行输出,不需要时钟进行同步,运算延迟时间仅为少量级数逻辑门的传输时间,运算速度快,适用于工作频率比较高的场合。

结合有限域变换的特性,对有限域的乘法做变换,采用异或门实现乘法运算,减少硬件复杂度。在钱搜索过程中,结合仿射多项式与格雷码,使用并行扩展的钱搜索方法,根据第一步先计算的一部分错误位置多项式,再使用少量异或门得到剩下的大部分错误位置多项式,分两步计算可以减少系统占用资源。仿真结果表明,此种译码方法具有运算速度快、占用资源少等优点。

参考文献:

[1] 金婕,于敦山. 高速并行 BCH 译码器的 VLSI 设计[J]. 北京大学学报: 自然科学版, 2009, 45(2):233 -237.

[2] 孙怡,田上力. BCH 码译码器的 FPGA 实现[J]. 电路与系统, 2000, 5(4):98 -100.

[3] 沈连丰. 信息论与编码[M]. 北京: 科学出版社, 2004.

[4] 赵华,殷奎喜. (15,7) BCH 编译码器的 VHDL 设计[J]. 现代电子技术, 2004(20):100 -101.

[5] 张亮,王志功,胡庆生. 并行 BCH 伴随式计算电路的优化[J]. 信号处理, 2010, 26(3):458 -461.

[6] ZHANG BOTAO, LIU DONGPEI, WANG SHIXIAN, *et al*. Design and implementation of area-efficient DVB-S2 BCH decoder [C]// Proceedings of the Second International Conference on Computer Engineering and Technology. Piscataway, NJ: IEEE Press, 2010: 179-184.

[7] KRISTIAN H, WAHYONO H. Ultra - fast - scalable BCH decoder with efficient-extended fast Chien search [C]// Proceedings of the third IEEE International Conference on Computer Science and Information Technology. Piscataway, NJ: IEEE Press, 2010: 338 -343.

[8] LIN T C, TRUONG T K, CHEN P D. A fast algorithm for the syndrome calculation in algebraic decoding of Reed-Solomon codes [J]. IEEE Transactions on Communications, 2007, 55(12): 2240 -2244.

[9] FEDORENKO S V, TRIFONOV P V. Find roots of polynomials over finite field[J]. IEEE Transactions on Communications, 2002, 50 (11): 1709 -1711.

[10] GHEORGHE S, CONSTANTIN A. An implementation of BCH codes in a FPGA [C]// 2010 International Conference on Applied Electronics. Piscataway, NJ: IEEE Press, 2010: 1 -4.