文章编号 1004-924X(2000)05-0410-06

弱对偶基下 RS 码译码方法的研究

曾晓洋, 郝志航, 魏仲慧
(中国科学院长春光学精密机械与物理研究所, 吉林 长春 130021)

摘要: 讨论了高速 RS 码译码器的设计问题。研究了有限域元素在弱对偶基(WDB)下的表示, 基于弱对偶基下的最优弱对偶基的计算方法, 给出了有限域比特并行乘法器的设计; 采用了一种可以避免求逆运算的修正 BM 迭代算法, 并且利用这样的迭代算法和基于弱对偶基的比特并行乘法器构成了广泛应用的 RS 码的译码器。对译码器定量分析的结果表明: 弱对偶基下比特并行乘法器设计复杂度降低, 便于 VLSI 实现; 修正 BM 迭代算法使得简单的硬件实现成为可能, 且有利于 On-The-Fly 纠错。译码器的数据吞吐率可达较高值, 有利于高速应用场合。

关 键 词: 弱对偶基; 比特并行; 译码器; RS 码; BM 算法
中图分类号: TN919.3 文献标识码: A

1 引 言

由于 RS 码具有同时纠突发错误和随机错误的能力, 且纠突发错误更有效, 因而广泛地应用于数据通信和数据存贮系统的差错控制中, 以作为提高数据传输和数据存贮可靠性的重要手段。令 α 为有限域 $GF(2^m)$ 的本原元, 则符号取自 $GF(2^m)$ 长度为 $n = 2^m - 1$ 能纠正 t 个错误的 RS 码由下面的多项式生成:

$$g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{2t-1} x^{2t-1} + x^{2t} \quad (1)$$

其中 $g_i \in GF(2^m)$, $0 \le i < 2t$。对于多项式 $r(x) = \sum_{i=0}^{n-1} r_i x^i$, 其中 $r_i \in GF(2^m)$, $0 \le i \le n-1$, 当且仅当 $r(x)$ 是由(1)式给出的生成多项式 $g(x)$ 的倍数时, $r(x)$ 为一码多项式。我们易知每个码多项式都有根 $\alpha, \alpha^2, \cdots, \alpha^{2t}$, 即 $r(\alpha^i) = 0$, $1 \le i \le 2t$。由(1)生成的 RS 码有 $2t$ 个校验符号, 其最小距离正好为 $2t+1$, 故它能纠正小于或等于 t 的任何形式的差错组合。

文献[1]中作者在讨论有限域元素的对偶基表示的基础上, 给出了对偶基下有限域元素的乘法运算方法, 分析了比特串行乘法器的原理和实现框图, 继而设计了 RS 码的译码器。本文中, 我们使用弱对偶基(weakly dual basis 简写为 WDB)的概念, 找到了一种最优的弱对偶基, 并且基于该最优弱对偶基给出了一种新的比特并行有限域元素乘法器, 利用这种乘法器和改进了的 BM 迭代算法完成了 RS 码译码器的设计。

2 弱对偶基(WDB)下有限域元素乘法

2.1 有限域元素的弱对偶基(WDB)表示

记有限域为 $GF(2^m)$, α 是有限域的本原元。有限域的元素可由基底的线性组合来表示。基 $\{\alpha^i\}$ $(0 \le i \le m-1)$ 被称为多项式基。弱对偶基的概念是通过迹(trace)的扩充来定义的: 即在常规的迹函数中插入一个非零的有限域元素来定义。

定义: 设 $\{\alpha\}$ 和 $\{\beta_i\}$ 是有限域 $GF(2^m)$ 的两个基, $\gamma \in GF(2^m)$。则当且仅当满足条件 $Tr(\gamma\alpha^i\beta_j) = \delta_{ij}$, $(i, j = 0, 1, 2, \cdots, m-1)$ 时, 则 $\{\alpha\}$ 和 $\{\beta_i\}$ 互为弱对偶基(WDB)或称 $\{\beta_i\}$ 是 $\{\alpha\}$ 的弱对偶基。其中 $Tr(\cdot)$ 是从域 $GF(2^m)$ 到域 $GF(2)$ 的迹函数。当 $i = j$ 时, $\delta_{ij} = 1$; $i \ne j$ 时, $\delta_{ij} = 0$。

我们仅对多项式基 $\{\alpha^i\}$ $(0 \le i \le m-1)$ 的弱对偶基感兴趣。考虑多项式基 $\{\alpha^i\}$ 及其弱对偶基 $\{\beta_i\}$, 对于 $A \in GF(2^m)$, $A = \sum_{i=0}^{m-1} a_i \alpha^i = \sum_{i=0}^{m-1} a_i^* \beta_i$, 其中 a_i 和 a_i^* 分别是 A 相对于多项式基及其弱对

收稿日期: 2000-05-29; 修订日期: 2000-07-05
基金项目: "863"高技术项目(863-2-7-4-6)

偶基的坐标。这样, 对于 $0 \leq j \leq m-1$, 我们有:

$$Tr(\gamma\alpha^j A) = Tr\left(\gamma\alpha^j \sum_{i=0}^{m-1} a_i^* \beta\right) = \sum_{i=0}^{m-1} a_i^* Tr(\gamma\alpha^j \beta) = a_j^* \quad (2)$$

2.2 有限域元素乘法

设 $\{a^i\}$ 和 $\{\beta_i\}$ 如上所定义, 有限域 $GF(2^m)$ 上两元素 A 和 B 分别由 $A = \sum_{i=0}^{m-1} a_i \alpha^i$ 及 $B = \sum_{i=0}^{m-1} b_i^* \beta$ 给出。考虑元素 A 和 B 的乘积, 在弱对偶基上由 $AB = C = \sum_{j=0}^{m-1} c_j^* \beta_j$ 给出。由(1)我们有: $c_j^* = Tr(\gamma\alpha^j AB) = Tr\left(\gamma\alpha^j B \sum_{i=0}^{m-1} a_i \alpha^i\right) = \sum_{i=0}^{m-1} Tr(\gamma\alpha^{i+j} B) a_i$。我们定义:

$$c_{j,i}^* = Tr(\gamma\alpha^{i+j} B) = \begin{cases} b_{i+j}^* & 0 \leq i+j \leq m-1 \\ Tr(\gamma\alpha^{i+j} B) & m \leq i+j \leq 2m-2 \end{cases} \quad (3)$$

明显地, 如果我们算出 $Tr(\gamma\alpha^{m+l} B)$, $l = 0, 1, 2, \cdots, m-2$, 则 c_j^* 通过(3)式来计算:

$$c_j^* = c_{j,0}^* a_0 + c_{j,1}^* a_1 + \cdots + c_{j,m-1}^* a_{m-1}, \quad j = 0, 1, 2, \cdots, m-1 \quad (4)$$

令 $f(x)$ 为有限域 $GF(2^m)$ 的 m 阶本原多项式: $f(x) = x^m + \sum_{i=0}^{m-1} f_i x^i$; α 为 $f(x)$ 的一个根; $Tr(\gamma\alpha^{m+l} B) = b_{m+l}^*$。则我们可得到求 b_{m+l}^*, $l = 0, 1, 2, \cdots, m-2$ 的方程如下:

$$b_{m+l}^* = Tr\left(\gamma B \alpha^l \sum_{i=0}^{m-1} f_i \alpha^i\right) = \sum_{i=0}^{m-1} f_i Tr(\gamma B \alpha^{l+i}) = \sum_{i=0}^{m-1} f_i b_{i+l}^* \quad (l = 0, 1, \cdots, m-2) \quad (5)$$

综上我们可以将有限域两元素 $A = \sum_{i=0}^{m-1} a_i \alpha^i$ 及 $B = \sum_{i=0}^{m-1} b_i^* \beta_i$ 的乘积 $C = \sum_{i=0}^{m-1} c_j^* \beta_j$ 表述成下面的形式:

$$\begin{bmatrix} b_0^* & b_1^* & \cdots & b_{m-1}^* \\ b_1^* & b_2^* & \cdots & b_m^* \\ \cdots & \cdots & \cdots & \cdots \\ b_{m-1}^* & b_m^* & \cdots & b_{2m-2}^* \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \cdots \\ a_{m-1} \end{bmatrix} = \begin{bmatrix} c_0^* \\ c_1^* \\ \cdots \\ c_{m-1}^* \end{bmatrix} \quad (6)$$

其中:

$$b_{m+l}^* = \sum_{i=0}^{m-1} f_i b_{i+l}^*, \quad l = 0, 1, \cdots, m-2 \quad (7)$$

2.3 最优弱对偶基

考虑到在定义弱对偶基时的变量 γ, 则对于任意的多项式基 $\{\alpha^i\}$ $(0 \leq i \leq m-1)$, 其弱对偶基不是一个而是有 $(2^m - 1)$ 个。这样我们就可以选择其中最优的一个, 而最优的原则是尽量使从弱对偶基到多项式基的变换最方便。

对于有限域 $GF(2^8)$, 令其本原多项式为:

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

α 为有限域 $GF(2^8)$ 的本原元素且满足 $f(\alpha) = 0$。则由文献[6]可得到: 当 $\gamma = \alpha^{250}$ 时, 多项式基 $\{1, \alpha, \cdots, \alpha^7\}$ 的最优弱对偶基为 $\{\alpha^{252}, \alpha^{251}, \alpha^{45}, \alpha^{98}, \alpha^1, 1, \alpha^{254}, \alpha^{253}\}$。

3　弱对偶基下比特并行乘法器

在一些应用中, 我们更多地采用比特并行乘法结构而不是比特串行结构来达到所需的性能。例如在 RS 译码器中由于电路复杂, 使用基于弱对偶基的比特并行结构的乘法器将更有效。为达此目的, 现在我们考虑这类乘法器的设计。

令 $A, B, C \in GF(2^m)$, 且 $C = A \cdot B$。有限域元素 A 由多项式基 $(1, \alpha, \alpha^2, \cdots, \alpha^{m-1})$ 表示, B 由相对于多项式基的最优弱对偶基 $\{\beta\}$ $(i = 0, 1, \cdots, m-1)$ 表示, 即: $A = \sum_{i=0}^{m-1} a_i \alpha^i$, $B = \sum_{i=0}^{m-1} b_i^* \beta_i$。则 $C = A \cdot B = \sum_{j=0}^{m-1} c_j^* \beta_j = \sum_{j=0}^{m-1} \left(\sum_{i=0}^{m-1} Tr(\gamma\alpha^{i+j} B) \beta_j\right)$。则由(6)式有:

$$\begin{aligned} c_0^* &= b_0^* a_0 + b_1^* a_1 + \cdots + b_{m-1}^* a_{m-1} \\ c_1^* &= b_1^* a_0 + b_2^* a_1 + \cdots + b_m^* a_{m-1} \\ &\cdots \quad \cdots \quad \cdots \quad \cdots \\ c_{m-1}^* &= b_{m-1}^* a_0 + b_m^* a_1 + \cdots + b_{2m-2}^* a_{m-1} \end{aligned}$$

其中 b_{m+l}^* $(l \geq 0)$ 由(7)式给定。从以上方程可知 m 个乘积位由具有下面形式的方程生成:

$$h(a, b) = b_k^* a_0 + b_{k+1}^* a_1 + \cdots + b_{k+m-1}^* a_{m-1} \quad (9)$$

在这些方程中, 所不同的只是 k 值。

这样基于弱对偶基的有限域 $GF(2^m)$ 上的比特并行乘法器可由 $GF(2)$ 上 m 个乘积模块(类型 A)和生成 b_k^* $(k = m, m+1, \cdots, 2m-2)$ 的模块(类型 B)来构造。例如对有限域 $GF(2^8)$, 考虑其本原多项式 $f(x) = x^8 + x^4 + x^3 + x^2 + 1$。由(9)式可知需要 8 个类型 A 的模块来实现方程 $h(a, b) = b_k^* a_0 + b_{k+1}^* a_1 + \cdots, b_{k+7}^* a_7$, 实现它的电路如图 1。由(7)和 $f(x) = x^8 + x^4 + x^3 + x^2 + 1$ 可有: $b_8^* = b_0^* + b_2^* + b_3^* + b_4^*$, $b_9^* = b_1^* + b_3^* + b_4^* + b_5^*$, $b_{10}^* = b_2^* + b_4^* + b_5^* + b_6^*$, $b_{11}^* = b_3^* + b_5^* + b_6^* + b_7^*$, $b_{12}^* = b_4^* + b_6^* + b_7^* + b_8^*$, $b_{13}^* = b_5^* + b_7^* + b_8^* + b_9^*$, $b_{14}^* = b_6^* + b_8^* + b_9^* + b_{10}^*$。实现它们可用类型 B 模块, 电路如图 2。

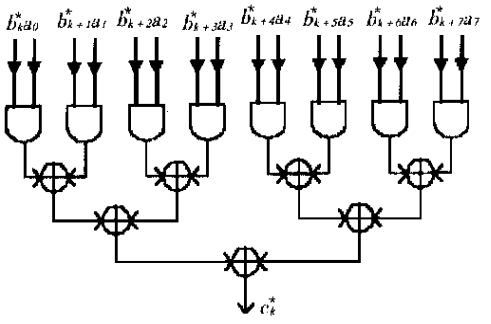

Fig. 1　Type A module for $GF(2^8)$

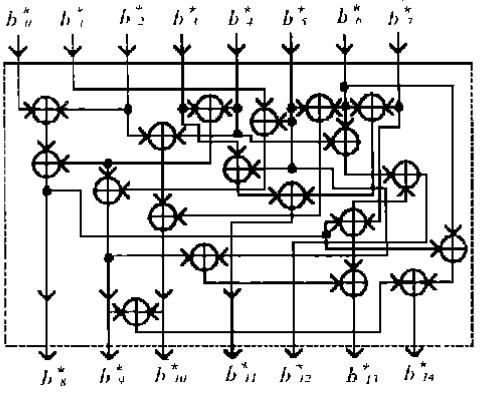

Fig. 2　Type B module for $GF(2^8)$

由类型 A 模块和类型 B 模块组合成一个完整的有限域 $GF(2^8)$ 上的比特并行乘法器如图 3。在图 3 中 $R_i = b_i^*$, $S_i = a_i (i = 0, 1, \cdots, 7)$, 一旦赋予这些值, 乘积项 c_i^* $(i = 0, 1, \cdots, 7)$ 在输出方便立即可得。

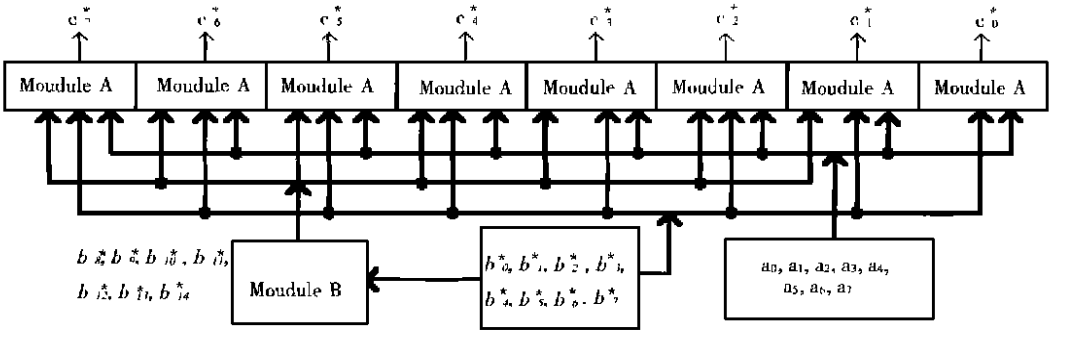

Fig. 3　Bit-parallel multiplier based on WDB

关于基于弱对偶基的 $GF(2^m)$ 比特并行乘法器的复杂度我们有①: 一个基于弱对偶基的 $GF(2^m)$ 比特并行乘法器一共需要 m^2 个与门和 $(m-1)(H(ff)-2+m)$ 个异或门。其中 $H(ff)$ 是本原多项式 $f(x)$ 汉明重量。这样整个比特并行乘法器的时延为 $T_A + T_X(t \times [\log_2(H(ff)-1)] + [\log_2 m])$。其中 T_A 表示一个 2 输入与门延迟, T_X 表示一个 2 输入异或门的延迟。

4　RS 码的译码

4.1　RS 码的译码程序

令 $R(x) = \sum_{i=0}^{n-1} r_i x^i$ 为接收字多项式, $C(x) = \sum_{i=0}^{n-1} c_i x^i$ 为发送码多项式, $E(x) = \sum_{i=0}^{n-1} e_i x^i$ 为差错图样多项式。则它们之间存在下面的关系: $C(x) = R(x) - E(x)$。设伴随式多项式为 $S(x)$, 差错定位多项式为 $\sigma(x)$。RS 码的基本译码过程如下:

(1) 计算伴随式, 以形成伴随式多项式。根据接收字多项式 $R(x)$, 得到 RS 码的伴随式 $S_j = R(\alpha^j) = \sum_{i=0}^{n-1} r_i \alpha^{ji}$ $(1 \le j \le 2t)$, 继而可形成伴随式多项式 $S(x) = \sum_{i=1}^{2t} S_i x^i$。

(2) 计算差错定位多项式。根据上面求得的伴随式多项式 $S(x)$, 得到求差错定位多项式的关键方程 $S(x)\sigma(x) \equiv \omega(x) \pmod{x^{2t+1}}$, 其中差错定位多项式 $\sigma(x) = \sum_{i=0}^{t} \sigma_i x^i$, 多项式 $\omega(x)$ 为差错值多项式。通过关键方程, 我们利用修正的 BM 迭代算法求得 $\sigma(x)$ 和 $\omega(x)$。

(3) 求差错位置。求差错位置一般使用 *Chien* 氏搜索法。即依次将 $1, \alpha, \cdots, \alpha^{n-1}$ 代入差错定位多项式 $\sigma(x)$ 中, 由于 $\alpha^n = 1, \alpha^{-l} = \alpha^{n-l}$, 故若 α^l 是

① 曾晓洋, 郝志航, 魏仲慧. 基于弱对偶基的 $GF(2^m)$ 上比特并行乘法器. 已投《微电子学》。

$\sigma(x)$ 的根, 则 α^{n-l} 是差错位置数。

(4) 计算差错值。根据 Forney 算法可以得到差错值为: $e_{n-l}=\frac{\omega(\alpha^l)}{\alpha^l\sigma'(\alpha^l)}$。

4.2 伴随式的计算

伴随式的计算可以通过使用 $2t$ 个 m 比特的并行乘法器来获得。接收字多项式 $R(x)$ 并行进入乘法器计算便开始, 将乘积项相加就可形成 S_j。其中 $r_i(i=0,1,\cdots,n-1)$ 是以弱对偶基表示的, 固定元素 α^j 以多项式基表示, 则获得的伴随式元素 S_j 也是以弱对偶基表示的。计算中的乘法器为弱对偶基下的比特并行乘法器。

4.3 修正的 BM 算法

通过伴随式求差错定位多项式和差错值多项式时, 利用 Euclid 算法或连分式方法的复杂度要高于使用 BM 迭代算法。故我们采用 BM 迭代算法, 但是原始的 BM 迭代算法存在求逆运算, 而求逆运算是一种即复杂又费时间的运算。文献[6]中作者提出了一种不需求逆的 BM 迭代算法, 新方法使得简单的硬件实现成为可能, 且有利于 On-The-Fly 纠错。

设 $S_1, S_2, \cdots, S_{2t}$ 为给定的值, 则下面的迭代算法用来计算 $\sigma^{(2t)}(x)$ 和 $\omega^{2t}(x)$:

初始条件: $\sigma^{(0)}(x)=\omega^{(0)}(x)=1, \lambda^{(0)}(x)=\beta^{(0)}(x)=1, l^{(0)}=0$; 当 $k\le 0$ 时, $\gamma^{(k)}=1$。

$$d^{(k)}=\sum_{j=0}^{n-1}\sigma_j^{(k-1)}S_{k-j};$$

$$\delta^{(k)}=\begin{cases}1; & d^{(k)}\neq 0,\ and\ 2l^{(k-1)}\le k-1\\ 0; & d^{(k)}=0,or\ 2l^{(k-1)}>k-1\end{cases}$$

$$l^{(k)}=\delta^{(k)}(k-l^{(k-1)})+(1-\delta^{(k)})l_{k-1};$$

$$\begin{bmatrix}\sigma^{(k)}(x)\\ \lambda^{(k)}(x)\end{bmatrix}=\begin{bmatrix}\gamma^{k-1} & -d^{(k)}x\\ \delta^{(k)} & (1-\delta^{(k)})x\end{bmatrix}\cdot\begin{bmatrix}\sigma^{(k-1)}(x)\\ \lambda^{(k-1)}(x)\end{bmatrix};$$

$$\begin{bmatrix}\omega^{(k)}(x)\\ \beta^{(k)}(x)\end{bmatrix}=\begin{bmatrix}\gamma^{k-1} & -d^{(k)}x\\ \delta^{(k)} & (1-\delta^{(k)}x\end{bmatrix}\cdot\begin{bmatrix}\omega^{(k-1)}(x)\\ \beta^{(k-1)}(x)\end{bmatrix};$$

$$\gamma^{(k)}=\delta^{(k)}d^{(k)}+(1-\delta^{(k)})\gamma^{k-1)}。$$

其中 $k=1,2,\cdots,2t$。算法终止时有差错定位多项式 $\sigma(x)=\sigma^{(2t)}(x)$, $\omega(x)=\omega^{(2t)}(x)$。

计算差错定位多项式 $\sigma(x)$ 的电路图如图 4, 差错值多项式 $\omega(x)$ 的计算与差错定位多项式相类似, 顾不重复讨论。对于 $\sigma(x)$, 输入序列为 $S_1, \cdots, S_{2t}$。多项式 $\sigma(x)$ 的最高阶数是 t, 这样需要 $(t+1)$ 个寄存器来存储用于计算 d 的伴随式。类似地, 寄存器 σ 用来寄存 $\sigma(x)$ 的系数 $\sigma_0, \sigma_1, \cdots, \sigma_t$, 寄存器 λ 用来寄存 $\lambda(x)$ 的系数 $\lambda_0, \lambda_1, \cdots, \lambda_t$。而寄存器 γ 和 l 用来寄存 γ 和 l 的值。判断单元用来产生控制开关的信号, 计数器 k 保存时间函数 k 的更新值。输出是寄存器 σ 中的值序列。图 4 所示结构的操作过程如下: 初始状态, 寄存器 T 的内容为 $(T_0, T_1, \cdots, T_t)=(S_0, 0, 0, \cdots, 0)$; 同时设寄存器 $\sigma=(\sigma_0, \sigma_1, \cdots, \sigma_t)=(1, 0, 0, \cdots, 0)$, $\lambda=(\lambda_0, \lambda_1, \cdots, \lambda_t)=(1, 0, 0, \cdots, 0)$, δ, l, k 分别为 1, 0, 0。经过计算的延迟后, 新的更新值存储在寄存器 σ, λ, γ, l 中, 新的伴随式移进 T_0, 同时 T_i 的内容右移进 $T_{i+1}(i=0, 1, \cdots, t-1)$。然后计数器 k 加 1。这样的过程迭代进行, 直至计数器 k 的内容为 $2t$ 时, 操作序列终止。

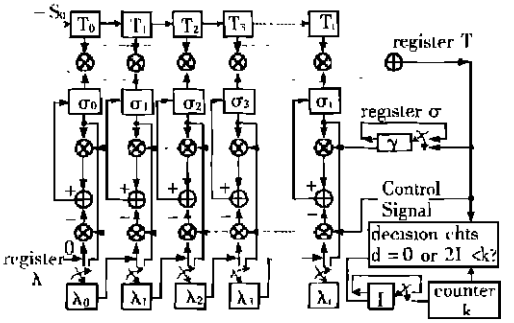

Fig.4　Implementation of error locator computation

4.4 Chien 氏搜索与差错值计算

RS 码译码程序的最后一步是通过差错定位多项式 $\sigma(x)$ 确定差错位置, 通过差错值多项式 $\omega(x)$ 计算差错值。我们使用图 5 所示的电路实现 Chien 氏搜索过程和差错值的计算。

电路的上半部分为差错值计算电路。由差错值计算公式 $e_{n-l}=\omega(\alpha^l)/\alpha^l\sigma'(\alpha^l)$ 可知 $\sigma'(x)$ 是 $\sigma(x)$ 的微分形式, 可以得到 $\sigma'(x)=\sigma_1+\sigma_3x^2+\cdots+\sigma_{2i+1}x^{2i}+\cdots$。寄存器 Z 的初始值为 $Z_1, Z_2, \cdots, Z_t(Z_{v+1}=Z_{v+2}=\cdots=Z_t=0$, 对于 $v\le t)$。r_{n-1} 正要从缓冲存储器中读出之前, t 个乘法器由移位脉冲控制进行乘法运算, 且将下列值存在寄存器中: $Z_1\alpha, Z_2\alpha^2, \cdots, Z_t\alpha^t$。累加器 C 的输出 $Z(\alpha)=1+Z_1\alpha+Z_2\alpha^2+\cdots+Z_t\alpha^t$, 若 B 的值为 -1, 则 $Z(\alpha)$ 与控制门的输出相乘得到差错多项式 $E(x)$。

电路的下半部分为 Chien 氏搜索电路, 过程如下: (1) t 个寄存器寄存 $\sigma_1, \sigma_2, \cdots, \sigma_t$, 当存贮器系统通道输出数据的实际差错个数 $v<t$ 时, 则有: $\sigma_{v+1}=\sigma_{v+2}=\cdots=\sigma_t=0$。(2) r_{n-1} 正要从缓冲存储器中读出之前, t 个乘法器由移位脉冲

控制进行乘法运算, 且将下列值存在寄存器中: $\sigma_1\alpha$, $\sigma_2\alpha^2$, …, $\sigma_t\alpha^t$, 并将这些值分奇偶送入累加器 1 和 2 中分别累加, 然后将累加器 1 和 2 的结果送入加法器 B 中, 若值为 - 1 则将控制门打开, 把差错值与缓冲器输出的 r_{n-1} 相减得到 c_{n-1}, 完成对 r_{n-1} 的纠错。(3) r_{n-1} 译码完成后, 再进行一次相乘, 此时 $\sigma_1(\alpha^2)$, $\sigma_2(\alpha^2)^2$, …, $\sigma_t(\alpha^t)^2$ 存在寄存器 σ 中, 并进行相加运算和检验, 对 r_{n-2} 进行纠错。(4) 其余码元同(2) 一样进行纠错。

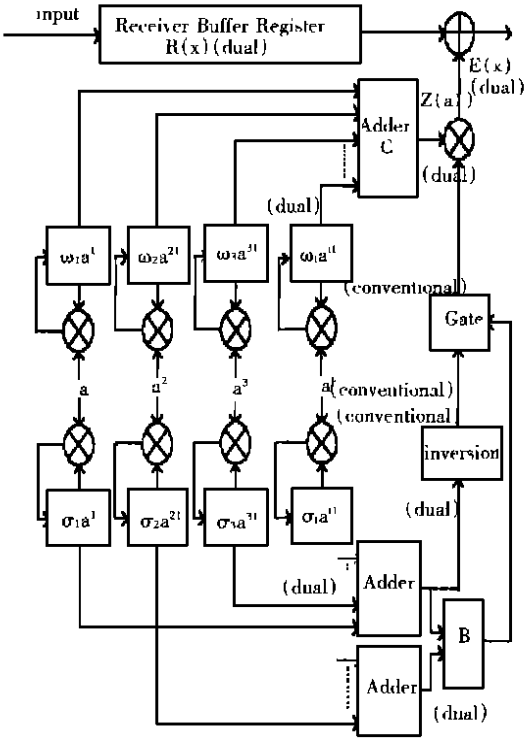

Fig. 5　Chien searching and error-value computing circuit

5　结　　论

基于弱对偶基的概念, 在选择最优弱对偶基的基础上, 我们设计了有限域上的比特并行乘法器。有限域上的比特并行乘法器有利于有限域上元素乘法的快速实现, 并且能加速译码器的数据吞吐率, 适用于高速应用场合。为避免原始 BM 迭代算法中即复杂又费时间的求逆运算, 使用了一种不需求逆的 BM 迭代算法, 新方法使得简单的硬件实现成为可能, 且有利于 On-The-Fly 纠错。一个完整的 RS 码译码器如图 6 所示, 它一共需要 $2n$ 个符号周期来完成一码。

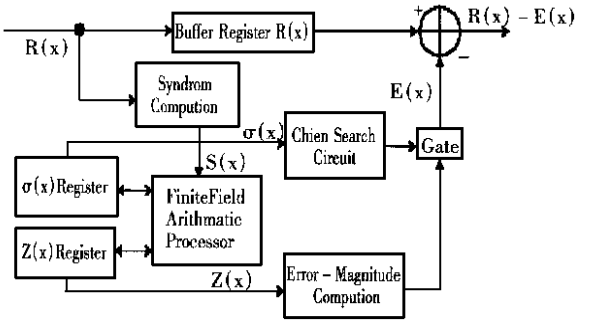

Fig. 6　A complete RS decoder architecture

参考文献:

[1] Lin Shu, Kasami Tao. Encoding and decoding of Reed-solomon codes in dual basis[J]. ACTA Electronics SINICA, 1986, 14(4): 6- 20.

[2] Fenn S T J, Benaissa M, Taylor David. GF(2^m) multiplication and division over the dual basis[J]. IEEE Trans. On Computer, 1996, 45(3): 319- 327.

[3] Morii M, Kasahara M, Whiting D L. Efficient bit-serial multiplication and the discrete-time Wiener-Hopf equation over finite field[J]. IEEE Trans. On Information Theory, 1989, 35(6): 1177- 1183.

[4] Wu H P, Hasan M A. Low complexity bit-parallel multipliers for a class of finite field[J]. IEEE Trans. On Computer, 1998, 47(8): 883- 887.

[5] Patel Arvid M. On-the-fly decoder for multiple byte errors[J]. IBM J. Develop., 1986, 30(3): 259- 269.

[6] 曾晓洋, 魏仲慧, 郝志航. 适用于 On-The-Fly 纠错的译码算法[A]. 第 11 届信息通信理论研讨会[C]. 中国·江西, 2000.

[7] 王新梅, 肖国镇. 纠错码——原理与方法[M]. 西安: 西安电子科技大学出版社, 1991.

Decoding method for RS codes in weakly-dual-basis

ZENG Xiao-yang, HAO Zhi-hang, WEI Zhong-hui

(*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun* 130021, *China*)

Abstract: The design problem of the high-speed RS decoder is discussed. The presentation of the finite-field elements in WDB is studied. And based on the computing method for the optimum WDB, the design for the bit-parallel multiplier of finite-field is presented. By selecting the bit-parallel multiplier based on WDB and the modified BM iterative algorithm that can avoid inversion, the widely used RS decoder is constructed. The analysis results indicates: the complexity of the bit-parallel multiplier is low and is suited for VLSI implementation; the modified BM iterative algorithm makes the simple hardware implementation possible and is advantageous to On-The-Fly error correcting. The throughout of the decoder can reach a high value and it is suited for the high speed application.

Key words: weakly-dual-basis; bit-parallel; decoder; RS codes; BM algorithm

作者简介: 曾晓洋(1972-), 男, 湖南衡阳人。中国科学院长春光机所博士研究生, 攻读光学工程博士学位, 研究方向为空间电子学。作为技术骨干参与多项国防重点工程和 "863- 2"重点课题的研究工作, 从事 EDA(电子设计自动化)及应用电子技术工作。感兴趣的研究领域有: 通信与存储系统检纠错码的应用研究, 数字图像与信号处理技术的研究等, 已发表科研论文多篇。E-mail: zengxiaoyang@yeah.net