3 CÓDIGOS BCH BINÁRIOS

3.1. INTRODUÇÃO [1]

Os códigos BCH (Bose, Chaudhuri e Hocquenghen) são uma importante e extensa classe de códigos cíclicos com grande capacidade de correção de erros. Esses códigos são uma generalização dos códigos de Hamming para correção de múltiplos erros. Os códigos BCH binários foram descobertos por Hocquenghen em 1959 e por Bose e Chaudhuri, de forma independente, em 1960. A estrutura cíclica dos códigos BCH foi provada por Peterson em 1960. Em 1961, Gorenstein e Zierler generalizaram os BCH para códigos em p^m símbolos, onde p é um número primo. Entre os códigos BCH não binários a sub-classe mais importante é a dos códigos Reed-Solomon (RS), descobertos por Reed e Solomon em 1960, independentemente dos trabalhos de Bose, Chaudhuri e Hocquen.

O primeiro algoritmo de decodificação para os códigos BCH binários foi desenvolvido por Peterson em 1960. A partir daí o algoritmo de Peterson foi generalizado e refinado por Gorenstein e Zierler, Chien, Forney, Berlekamp, Massey, Burton e outros. Entre todos os algoritmos para a decodificação dos códigos BCH o algoritmo iterativo de Berlekamp e o algoritmo de busca de Chien são os mais eficientes.

Este texto apresenta os fundamentos dos códigos BCH, o processo de criação de um código, o processo de decodificação pelos algoritmos de Peterson e de Berlekamp, e algumas considerações sobre implementações.

3.2. CÓDIGOS BCH PRIMITIVOS BINÁRIOS

Para qualquer inteiro $m \geq 3$ e $t < 2^{m-1}$, existe um código binário BCH com os seguintes parâmetros:

Comprimento do bloco	$n = 2^m-1$
nº de dígitos de verificação de paridade	$n-k \leq mt$
Distância mínima	$d_{min} \geq 2^t + 1$

O polinômio gerador deste código é especificado em termos de suas raízes do campo de Galois $GF(2^m)$. O polinômio gerador é o polinômio de menor grau sobre $GF(2^m)$ que tem

$$\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t} \tag{3.1}$$

como suas raízes, [isto é, $\mathbf{g}(\alpha_i) = 0$ para $1 \leq i \leq 2t$]. Isso significa que $\mathbf{g}(X)$ tem $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ e seus conjugados como todas as suas raízes. Seja $\phi_i(X)$ o polinômio mínimo de α^i. Então, $\mathbf{g}(X)$ deve ser o mínimo múltiplo comum (MMC) de $\phi_1(X), \phi_2(X), \ldots, \phi_{2t}(X)$, isto é,

$$\mathbf{g}(X) = \text{MMC}\{\phi_1(X), \phi_2(X), \ldots, \phi_{2t}(X)\}. \tag{3.2}$$

Se i é um inteiro par, ele pode se representado como um produto da forma como se segue.

$$i = i' 2^l,$$

onde i' é um número ímpar, e $l \geq 1$. Então, $\alpha^i = \left(\alpha^i\right)^{2^l}$ é um conjugado de $\alpha^{i'}$ e, portanto, α^i e $\alpha^{i'}$ tem o mesmo polinômio mínimo; isto é,

$$\phi_i(X) = \phi_{i'}(X).$$

Consequentemente, toda potência par de α na sequência em (3.1) tem o mesmo polinômio mínimo que algumas potências impares precedentes na sequência. Como resultado, o polinômio gerador $\mathbf{g}(X)$ de um código BCH binário com comprimento $2^m - 1$ e capacidade de correção de t erros dado por (3.2) pode ser reduzido para

$$\mathbf{g}(X) = \text{MMC}\{\phi_1(X), \phi_3(X), \ldots, \phi_{2t-1}(X)\}. \tag{3.3}$$

Pelo fato do grau de cada polinômio mínimo ser m ou menor, o grau de $\mathbf{g}(X)$, que é igual a $n - k$, é no máximo igual a mt. Não há uma fórmula simples para a determinação de $n - k$, mas se t é pequeno, $n - k$ é exatamente igual a mt. A Tabela 3.1 apresenta os parâmetros para todos os códigos BCH de comprimento $2^m - 1$ com $m \leq 8$.

Tabela 3.1 - Características dos códigos BCH para valores de m até 8

n	k	t	n	k	t	n	k	t	n	k	t
7	4	1		120	1		247	1		115	22
	11	1		113	2		239	2		107	23
15	7	2		106	3		231	3		99	24
	5	3		99	4		223	4		91	25
	26	1		92	5		215	5		87	26
	21	2		85	6		207	6		79	27
31	16	3		78	7		199	7		71	29
	11	5		71	9		191	8	**255**	63	30
	6	7	**127**	64	10	**255**	187	9		55	31
	57	1		57	11		179	10		47	42
	51	2		50	13		171	11		45	43
	45	3		43	14		163	12		37	45
	39	4		36	14		155	13		29	47
	36	5		29	21		147	14		21	55
63	30	6		22	23		139	18		13	59
	24	7		15	27		131	19		9	63
	18	10		8	31		123	21			
	16	11									
	10	13									
	7	15									

De (3.3) pode-se observar que para um código BCH de comprimento $2^m - 1$, com capacidade de correção de um único erro, $\phi_{2t-1}(X) = \phi_1(X)$. Logo ele é gerado por

$$\mathbf{g}(X) = \phi_1(X).$$

Devido a α ser um elemento primitivo de $GF(2^m)$, $\phi_1(X)$ é um polinômio primitivo de grau m. Portanto, um código BCH de comprimento $2^m - 1$, com capacidade de correção de um único erro, é um código de Hamming.

EXEMPLO 3.1

Considere o GF(2^4) gerado por $p(X) = 1 + X + X^4$, apresentado na Tabela 3.2 e suas raízes conjugadas apresentadas na Tabela 3.3. Determine os polinômios geradores para códigos BCH com capacidades de correção de dois e de três erros.

Tabela 3.2 – GF(2^4) gerado por $p(X) = 1 + X + X^4$

Representação por potência	Representação polinomial	Representação vetorial
0	0	(0000)
$\alpha^0 = 1$	1	(1000)
α^1	α	(0100)
α^2	α^2	(0010)
α^3	α^3	(0001)
α^4	$1 + \alpha$	(1100)
α^5	$\alpha + \alpha^2$	(0110)
α^6	$\alpha^2 + \alpha^3$	(0011)
α^7	$1 + \alpha + \alpha^3$	(1101)
α^8	$1 + \alpha^2$	(1010)
α^9	$\alpha + \alpha^3$	(0101)
α^{10}	$1 + \alpha + \alpha^2$	(1110)
α^{11}	$\alpha + \alpha^2 + \alpha^3$	(0111)
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	(1111)
α^{13}	$1 + \alpha^2 + \alpha^3$	(1011)
α^{14}	$1 + \alpha^3$	(1001)

Tabela 3.3 – Raízes conjugadas de GF(2^4) gerado por $p(X) = 1 + X + X^4$

β	β^{2^l}	Raízes conjugadas
0	0	0
1	1	1
α	$\alpha^2, \alpha^4, \alpha^8$	$\alpha, \alpha^2, \alpha^4, \alpha^8$
α^3	$\alpha^6, \alpha^{12}, \alpha^{24} = \alpha^9$	$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$
α^5	α^{10}	α^5, α^{10}
α^7	$\alpha^{14}, \alpha^{28} = \alpha^{13}, \alpha^{56} = \alpha^{11}$	$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$

SOLUÇÃO

A partir de GF(2^4) pode-se criar códigos BCH com comprimento $n = 2^4 - 1 = 15$, ou seja, os códigos serão BCH (15, k).

Os polinômios mínimos correspondentes às raízes conjugadas são:

Raízes conjugadas	Polinômios mínimos
0	$\phi(X) = X$
1	$\phi_0(X) = X + 1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$\phi_1(X) = (X+\alpha)(X+\alpha^2)(X+\alpha^4)(X+\alpha^8)$ $= [X^2 + (\alpha+\alpha^2)X + \alpha^3][X^2 + (\alpha^4+\alpha^8)X + \alpha^{12}]$ $= [X^2 + \alpha^5 X + \alpha^3][X^2 + \alpha^5 X + \alpha^{12}]$ $= X^4 + (\alpha^5+\alpha^5)X^3 + (\alpha^{12}+\alpha^{10}+\alpha^3)X^2 + (\alpha^8+\alpha^{17})X + \alpha^{15}$ $= X^4 + X + 1$
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$\phi_3(X) = (X+\alpha^3)(X+\alpha^6)(X+\alpha^9)(X+\alpha^{12})$ $= [X^2 + (\alpha^3+\alpha^6)X + \alpha^9][X^2 + (\alpha^9+\alpha^{12})X + \alpha^{21}]$ $= [X^2 + \alpha^2 X + \alpha^9][X^2 + \alpha^8 X + \alpha^6]$ $= X^4 + (\alpha^2+\alpha^8)X^3 + (\alpha^9+\alpha^{10}+\alpha^6)X^2 + (\alpha^{17}+\alpha^8)X + \alpha^{15}$ $= X^4 + X^3 + X^2 + X + 1$
α^5, α^{10}	$\phi_5(X) = (X+\alpha^5)(X+\alpha^{10})$ $= X^2 + (\alpha^5+\alpha^{10})X + \alpha^{15}$ $= X^2 + X + 1$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$\phi_7(X) = (X+\alpha^7)(X+\alpha^{11})(X+\alpha^{13})(X+\alpha^{14})$ $= [X^2 + (\alpha^7+\alpha^{11})X + \alpha^{18}][X^2 + (\alpha^{13}+\alpha^{14})X + \alpha^{27}]$ $= [X^2 + \alpha^8 X + \alpha^{18}][X^2 + \alpha^2 X + \alpha^{12}]$ $= X^4 + (\alpha^2+\alpha^8)X^3 + (\alpha^{12}+\alpha^{10}+\alpha^{18})X^2 + (\alpha^{20}+\alpha^{20})X + \alpha^{15}$ $= X^4 + X^3 + X^2 + 1$

Para correção de duplo erro:

Para correção de duplo erro o polinômio gerador é obtido fazendo:

$$\mathbf{g}(X) = \text{MMC}\{\phi_1(X), \phi_3(X)\}.$$

Como $\phi_1(X)$ e $\phi_3(X)$ são dois polinômio irredutíveis distintos,

$$\mathbf{g}(X) = \phi_1(X)\phi_3(X) = (1 + X + X^4)(1 + X + X^2 + X^3 + X^4)$$

$$\mathbf{g}(X) = 1 + X^4 + X^6 + X^7 + X^8. \qquad (3.4)$$

Logo $\mathbf{g}(X)$ gera um código BCH (15, 7) cíclico com $d_{min} \geq 5$. Uma vez que o polinômio gerador do código possui peso 5, então a distância mínima do código é exatamente 5.

Para correção de triplo erro:

Para correção de três erros o polinômio gerador é obtido fazendo:

$$\mathbf{g}(X) = \text{MMC}\{\phi_1(X), \phi_3(X), \phi_5(X)\} = (1 + X + X^4)(1 + X + X^2 + X^3 + X^4)(1 + X + X^2)$$

$$\mathbf{g}(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}. \tag{3.5}$$

Este é um código BCH (15, 5) cíclico com $d_{min} \geq 7$. Uma vez que o polinômio gerador do código possui peso 7, então a distância mínima do código é exatamente 7.

* * *

É importante observar que o peso do polinômio gerador é igual à distância mínima do código apenas para os códigos BCH primitivos. Existem códigos BCH não primitivos, que não são apresentados neste texto, para os quais essa afirmação não é válida. Uma abordagem mais aprofundada sobre a distância mínima dos códigos BCH pode ser encontrada em [1].

3.3. DECODIFICAÇÃO DOS CÓDIGOS BCH [1][2]

Admita que uma palavra código $\mathbf{c}(X) = v_0 + v_1 X + v_2 X^2 + ... + v_{n-1} X^{n-1}$ seja transmitida e os erros introduzidos pelo canal de comunicação tenham produzido a palavra recebida

$$\mathbf{r}(X) = r_0 + r_1 X + r_2 X^2 + ... + r_{n-1} X^{n-1}.$$

Se $\mathbf{e}(X)$ foi o padrão de erro introduzido pelo canal, então

$$\mathbf{r}(X) = \mathbf{c}(X) + \mathbf{e}(X). \tag{3.6}$$

Como nos códigos de blocos mais simples, o processo de decodificação se inicia por meio do cálculo da síndrome de erros para o vetor recebido $\mathbf{r}(X)$. Para os códigos BCH primitivos a divisão de $\mathbf{r}(X)$ pelo polinômio mínimo $\phi_i(X)$ de α_i, para $1 \leq i \leq 2t$, pode ser escrito como

$$\mathbf{r}(X) = \mathbf{a}_i(X)\boldsymbol{\phi}_i(X) + \mathbf{b}_i(X), \tag{3.7}$$

onde $\mathbf{b}_i(X)$ é o resto da divisão. Como $\boldsymbol{\phi}_i(\alpha^i) = 0$, então

$$S_i = \mathbf{b}_i(\alpha^i) = \mathbf{r}(\alpha^i) = r_0 + r_1\alpha^i + r_2\alpha^{2i} + ... + r_{n-1}\alpha^{(n-1)i}. \tag{3.8}$$

Assim, cada componente da síndrome S_i pode ser obtido diretamente pela determinação de $\mathbf{r}(X)$ com $X = \alpha^i$, de forma que

$$\mathbf{S} = (S_1, S_2, ... , S_{2t}), \tag{3.9}$$

ou seja, a síndrome de erros $\mathbf{S}$ para os códigos BCH é formada por $2t$ síndromes componentes S_i.

EXEMPLO 3.2

Considere o código BCH (15, 7) com capacidade de correção de duplo erro gerado por (3.4). Admita que o vetor recebido foi **r** = (100000001000000). Determine o conjunto de síndromes de erros para o vetor recebido.

SOLUÇÃO

O polinômio correspondente é

$$\mathbf{r}(X) = 1 + X^8.$$

A síndrome consiste de quatro componentes:

$$\mathbf{S} = (S_1, S_2, S_3, S_4).$$

Conforme apresentado no Exemplo 3.1, os polinômios mínimos para α, α^2 e α^4 são idênticos, e

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4.$$

O polinômio mínimo para α^3 é

$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4.$$

Dividindo $\mathbf{r}(X) = 1 + X^8$ por $\phi_1(X) = 1 + X + X^4$, obtém-se

$$\mathbf{b}_1(X) = X^2.$$

Dividindo $\mathbf{r}(X) = 1 + X^8$ por $\phi_3(X) = 1 + X + X^2 + X^3 + X^4$, obtém-se

$$\mathbf{b}_3(X) = 1 + X^3.$$

Usando $GF(2^4)$ dado na Tabela 3.2 e substituindo α, α^2 e α^4 em $\mathbf{b}_1(X)$, obtém-se

$$S_1 = \alpha^2, \qquad S_2 = \alpha^4, \qquad S_4 = \alpha^8.$$

Substituindo α^3 em $\mathbf{b}_3(X)$, obtém-se

$$S_3 = 1 + \alpha^9 = 1 + \alpha + \alpha^3 = \alpha^7.$$

Assim,

$$S = (\alpha^2, \alpha^4, \alpha^7, \alpha^8).$$

Conforme (3.8) as síndromes podem ser obtidas diretamente a partir de $\mathbf{r}(X)$ conforme mostrado a seguir

$$\begin{aligned} S_1 &= \mathbf{r}(\alpha^1) = 1 + \alpha^8 = 1 + 1 + \alpha^2 & S_1 &= \alpha^2 \\ S_2 &= \mathbf{r}(\alpha^2) = 1 + \alpha^{16} = 1 + \alpha^1 & S_2 &= \alpha^4 \\ S_3 &= \mathbf{r}(\alpha^3) = 1 + \alpha^{24} = 1 + \alpha^9 & S_3 &= \alpha^7 \\ S_4 &= \mathbf{r}(\alpha^4) = 1 + \alpha^{32} = 1 + \alpha^2 & S_4 &= \alpha^8 \end{aligned}$$

Assim,

$$S = (\alpha^2, \alpha^4, \alpha^7, \alpha^8).$$

* * *

Como $\alpha^1, \alpha^2, \ldots, \alpha^{2t}$ são raízes de cada polinômio código, então $\mathbf{c}(\alpha^i) = 0$ para $1 \le i \le 2t$. A partir de (3.6) e (3.8), obtém-se, para $1 \le i \le 2t$, a seguinte relação entre a síndrome e o padrão de erro,

$$S_i = \mathbf{e}(\alpha^i). \tag{3.10}$$

A equação (3.10) mostra que a síndrome **S** depende exclusivamente do padrão de erros **e**. Admita que o padrão de erro $\mathbf{e}(X)$ tem v erros nas localizações $X^{j1}, X^{j2}, \ldots, X^{jv}$; isto é

$$\mathbf{e}(X) = X^{j_1} + X^{j_2} + \ldots + X^{j_v} \tag{3.11}$$

onde $0 \le j_1 < j_2 < \ldots < j_v < n$. De (3.10) e (3.11) obtém-se o seguinte conjunto de equações:

$$\begin{aligned}
S_1 &= \alpha^{j_1} + \alpha^{j_2} + \ldots + \alpha^{j_v} \\
S_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \ldots + (\alpha^{j_v})^2 \\
S_3 &= (\alpha^{j_1})^3 + (\alpha^{j_2})^3 + \ldots + (\alpha^{j_v})^3 \\
&\vdots \\
S_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \ldots + (\alpha^{j_v})^{2t},
\end{aligned} \tag{3.12}$$

onde $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_v}$ são desconhecidos.

Qualquer método para a solução dessas equações é um algoritmo de decodificação para os códigos BCH.

Uma vez encontrados os valores para $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_v}$, as potências $j_1, j_2, \ldots, j_v$ indicam as localizações de erros em $\mathbf{e}(X)$, conforme mostrado em (3.11). Geralmente, o sistema de equações (3.12) têm muitas soluções possíveis. Cada solução produz um padrão de erro diferente. A solução correta é a que apresenta o menor número de erros entre aquelas em que o padrão de erro $\mathbf{e}(X)$ possui um número de erros igual a t ou menos ($v \le t$).

Para grandes valores de t, a solução direta de (3.12) é difícil e ineficaz. A seguir é apresentado um procedimento eficaz para a determinação de α^{j_l} para $l = 1, 2, \ldots, v$ dos componentes S_i's da síndrome.

- **ALGORITMO DE PETERSON [1] [2] [3]**

Por conveniência, seja

$$\beta_l = \alpha^{j_l} \tag{3.13}$$

para $1 \le l \le v$. Esses elementos são chamados de *números de localização de erros*. Assim o sistema em (3.12) pode ser reescrito da seguinte forma:

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \ldots + \beta_v \\ S_2 &= \beta_1^{\,2} + \beta_2^{\,2} + \ldots + \beta_v^{\,2} \\ S_3 &= \beta_1^{\,3} + \beta_2^{\,3} + \ldots + \beta_v^{\,3} \\ &\vdots \\ S_{2t} &= \beta_1^{\,2t} + \beta_2^{\,2t} + \ldots + \beta_v^{\,2t}. \end{aligned} \tag{3.14}$$

Essas 2t equações são funções simétricas em β_1, β_2, ... , β_v, que são conhecidas como *funções simétricas de soma de potências*. Agora, considere o seguinte polinômio:

$$\begin{aligned} &\sigma(X) \triangleq (1+\beta_1 X)(1+\beta_2 X)\ldots(1+\beta_v X) = \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \ldots + \sigma_v X^v \\ &\sigma(X) = 1 + \sigma_1 X + \sigma_2 X^2 + \ldots + \sigma_v X^v. \end{aligned} \tag{3.15}$$

As raízes de $\sigma(X)$ são β_1^{-1}, β_2^{-1}, ... , β_v^{-1}, que são o inverso dos números localizadores de erros. Por essa razão, $\sigma(X)$ é chamado de polinômio localizador de erros. Nota-se que $\sigma(X)$ é um polinômio não conhecido cujos coeficientes devem ser determinados. Os coeficientes de $\sigma(X)$ e o número localizador de erros são relacionados pelas seguintes equações:

$$\begin{aligned} \sigma_0 &= 1 \\ \sigma_1 &= \beta_1 + \beta_2 + \ldots + \beta_v \\ \sigma_2 &= \beta_1\beta_2 + \beta_2\beta_3 + \ldots + \beta_{v-1}\beta_v \\ &\vdots \\ \sigma_v &= \beta_1\beta_2 \ldots \beta_v. \end{aligned} \tag{3.16}$$

Os σ_i's são conhecidos como *funções simétricas elementares de* β_l's. De (3.14) e (3.16) pode-se verificar que os σ_i's estão relacionados com os componentes de síndrome S_j's. De fato, eles estão relacionados com os componentes de síndrome pelas seguintes *identidades de Newton*:

$$\begin{aligned} &S_1 + \sigma_1 = 0 \\ &S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\ &S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0 \\ &\quad\vdots \\ &S_v + \sigma_1 S_{v-1} + \ldots + \sigma_{v-1} S_1 + v\sigma_v = 0 \\ &S_{v+1} + \sigma_1 S_v + \ldots + \sigma_{v-1} S_2 + \sigma_v S_1 = 0 \end{aligned} \tag{3.17}$$

Para o caso binário, uma vez que 1 + 1 = 2 = 0, tem-se

$$i\sigma_i = \begin{cases} \sigma_i & \text{para } i \text{ ímpar} \\ 0 & \text{para } i \text{ par} \end{cases}$$

que resulta em

$$\begin{aligned} &S_1 + \sigma_1 = 0 \\ &S_3 + \sigma_1 S_2 + \sigma_2 S_1 + \sigma_3 = 0 \\ &S_5 + \sigma_1 S_4 + \sigma_2 S_3 + \sigma_3 S_2 + \sigma_4 S_1 + \sigma_5 = 0 \\ &\quad \vdots \end{aligned} \tag{3.18}$$

Desta forma o sistema fica restrito a um número t de equações. Uma vez que é possível determinar as funções simétricas elementares σ_1, σ_2, ... , σ_v do sistema de equações (3.18), os números localizadores de erros β_1, β_2, ... , β_v podem ser encontrados pela determinação das raízes do polinômio localizador de erros $\sigma(X)$. Novamente, o sistema de equações (3.18) pode ter muitas soluções; entretanto, a solução será aquela que resultar em um $\sigma(X)$ de grau mínimo. Este $\sigma(X)$ produzirá um padrão de erro $\mathbf{e}(X)$ com um número mínimo de erros. Se $v \leq t$, este $\sigma(X)$ dará o padrão de erro $\mathbf{e}(X)$ verdadeiro.

A seguir será apresentado um algoritmo para a determinação do polinômio $\sigma(X)$ de grau mínimo que satisfaz as primeiras t equações de (3.18), apesar de podermos determinar até $2t$ síndromes. Isso se deve ao fato de que para os códigos binários apenas as síndromes ímpares serão usadas no processo de decodificação.

O algoritmo de Peterson pode ser resumido nos seguintes passos:

1. Calcule a síndrome $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$ a partir do polinômio recebido $\mathbf{r}(X)$.
2. Determine os polinômios localizadores de erros $\sigma(X)$ a partir dos componentes.
3. Determine os números localizadores de erros β_1, β_2, ... , β_v por meio das raízes de $\sigma(X)$ e corrija os erros em $\mathbf{r}(X)$.

Os passos 1 e 3 são muito simples enquanto que o passo 2 é a parte mais complicada da decodificação dos códigos BCH, onde a complexidade de decodificação aumenta com o aumento da capacidade de correção de erros dos códigos.

Por exemplo, na decodificação de um código BCH binário, com capacidade de correção de apenas um erro, haverá somente um valor de síndrome S_1 e a primeira linha de (3.18) determina que

$$\sigma_1 = S_1$$

Para $t = 1$, o polinômio localizador de erros, obtido a partir de (3.15) resume-se a

$$\sigma(X) = 1 + S_1 X.$$

Para códigos com capacidade de correção de dois erros, $\sigma_3 = 0$. Além disso, pode-se demonstrar facilmente que para códigos binários a seguinte igualdade é verdadeira [2]:

$$S_{2i} = S_i^2 \quad \text{para qualquer } i. \tag{3.19}$$

Assim, duas síndromes (S_1 e S_3) devem ser calculadas e por meio das duas primeiras equações de (3.18) obtém-se

$$S_1 + \sigma_1 = 0 \quad \Rightarrow \quad \sigma_1 = S_1 \tag{3.20}$$

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 = 0 \quad \Rightarrow \quad S_3 + S_1 S_1^2 + \sigma_2 S_1 = 0 \quad \Rightarrow \quad \sigma_2 = \frac{S_3 + S_1^3}{S_1} \tag{3.21}$$

Usando esta técnica ou qualquer outra técnica padrão para a solução do sistema de equações lineares de (3.18) pode-se determinar os valores dos σ_i's para qualquer capacidade de correção de erros. A Tabela 3.4 apresenta os coeficientes do polinômio localizador de erros em função dos componentes da síndrome, para códigos com capacidade de correção $1 \le t \le 5$.

Tabela 3.4 – Coeficientes do polinômio localizador de erros em função dos componentes da síndrome para $1 \le t \le 5$.

t	σ_l (S_i)
1	$\sigma_1 = S_1$
2	$\sigma_1 = S_1$ $\sigma_2 = \frac{S_3 + S_1^3}{S_1}$
3	$\sigma_1 = S_1$ $\sigma_2 = \frac{S_1^2 S_3 + S_5}{S_1^3 + S_3}$ $\sigma_3 = \left(S_1^3 + S_3\right) + S_1 \sigma_2$
4	$\sigma_1 = S_1$ $\sigma_2 = \frac{S_1\left(S_7 + S_1^7\right) + S_3\left(S_1^5 + S_5\right)}{S_3\left(S_1^3 + S_3\right) + S_1\left(S_1^5 + S_5\right)}$ $\sigma_3 = \left(S_1^3 + S_3\right) + S_1 \sigma_2$ $\sigma_4 = \frac{\left(S_5 + S_1^2 S_3\right) + \left(S_1^3 + S_3\right)\sigma_2}{S_1}$
5	$\sigma_1 = S_1$ $\sigma_2 = \frac{\left(S_1^3 + S_3\right)\left[\left(S_1^9 + S_9\right) + S_1^4\left(S_5 + S_3 S_1^2\right) + S_3^2\left(S_1^3 + S_3\right)\right] + \left(S_1^5 + S_5\right)\left(S_7 + S_1^7\right) + S_1\left(S_3^2 + S_1 S_5\right)}{\left(S_1^3 + S_3\right)\left[\left(S_7 + S_1^7\right) + S_1 S_3\left(S_1^3 + S_3\right)\right] + \left(S_5 + S_1^2 S_3\right)\left(S_1^5 + S_5\right)}$ $\sigma_3 = \left(S_1^3 + S_3\right) + S_1 \sigma_2$ $\sigma_4 = \frac{\left(S_1^9 + S_9\right) + S_3^2\left(S_1^3 + S_3\right) + S_1^4\left(S_5 + S_3 S_1^2\right) + \left[\left(S_7 + S_1^7\right) + S_1 S_3\left(S_1^3 + S_3\right)\right]\sigma_2}{\left(S_1^5 + S_5\right)}$ $\sigma_5 = \left(S_5 + S_3 S_1^2\right) + S_1 \sigma_4 + \left(S_1^3 + S_3\right)\sigma_2$

Exemplo 3.3

Admita que o vetor código todo zero do código BCH (15, 5) gerado por (3.5) tenha sido corrompido por ruído resultando no vetor recebido **r** = 000101000000100. Decodifique o vetor recebido.

SOLUÇÃO

O polinômio recebido obtido a partir do vetor recebido **r** é

$$\mathbf{r}(X) = X^3 + X^5 + X^{12}$$

De onde se obtém as síndromes

$$S_1 = \mathbf{r}(\alpha^1) = \alpha^3 + \alpha^5 + \alpha^{12} = \alpha^3 + \alpha + \alpha^2 + 1 + \alpha + \alpha^2 + \alpha^3 = 1 \qquad S_1 = 1$$
$$S_3 = \mathbf{r}(\alpha^3) = \alpha^9 + \alpha^{15} + \alpha^{36} = \alpha + \alpha^3 + 1 + \alpha^2 + \alpha^3 = \alpha^{10} \qquad S_3 = \alpha^{10}$$
$$S_5 = \mathbf{r}(\alpha^5) = \alpha^{15} + \alpha^{25} + \alpha^{60} = 1 + 1 + \alpha + \alpha^2 + 1 = \alpha^{10} \qquad S_5 = \alpha^{10}$$

Da Tabela 3.4 obtém-se para $t = 3$, com o auxílio da Tabela 3.2, os seguintes resultados para σ_i:

$$\sigma_1 = S_1 = 1$$
$$\sigma_2 = \frac{S_1^2 S_3 + S_5}{S_1^3 + S_3} = \frac{1^2 \alpha^{10} + \alpha^{10}}{1^2 + \alpha^{10}} = 0$$
$$\sigma_3 = \left(S_1^3 + S_3\right) + S_1\sigma_2 = 1^2 + \alpha^{10} = \alpha^5$$

Logo, o polinômio localizador de erros (3.15) torna-se

$$\sigma(X) = 1 + X + \alpha^5 X^3.$$

As raízes de $\sigma(X)$ podem ser encontradas fazendo

$$\sigma(\alpha^0) = 1 + 1 + \alpha^5 1^3 = \alpha^5$$
$$\sigma(\alpha^1) = 1 + \alpha^1 + \alpha^5(\alpha^1)^3 = 1 + \alpha^1 + \alpha^8 = \alpha^5$$
$$\sigma(\alpha^2) = 1 + \alpha^2 + \alpha^5(\alpha^2)^3 = 1 + \alpha^2 + \alpha^{11} = \alpha^2$$
$$\sigma(\alpha^3) = 1 + \alpha^3 + \alpha^5(\alpha^3)^3 = 1 + \alpha^3 + \alpha^{14} = 0 \quad \Rightarrow \quad \alpha^3 \text{ é raiz}$$
$$\vdots$$
$$\sigma(\alpha^{10}) = 1 + \alpha^{10} + \alpha^5(\alpha^{10})^3 = 1 + 1 + \alpha + \alpha^2 + \alpha + \alpha^2 = 0 \quad \Rightarrow \quad \alpha^{10} \text{ é raiz}$$
$$\vdots$$
$$\sigma(\alpha^{12}) = 1 + \alpha^{12} + \alpha^5(\alpha^{12})^3 = 1 + 1 + \alpha + \alpha^2 + \alpha^3 + \alpha + \alpha^2 + \alpha^3 = 0 \quad \Rightarrow \quad \alpha^{12} \text{ é raiz}$$
$$\vdots$$
$$\sigma(\alpha^{14}) = 1 + \alpha^{14} + \alpha^5(\alpha^{14})^3 = 1 + 1 + \alpha^3 + \alpha^2 = \alpha^6$$

As posições de erros são o inverso das raízes do polinômio localizador de erros, logo:

$$\beta_1 = \frac{1}{\alpha^3} = \alpha^{-3} = \alpha^{12} \quad \Rightarrow \quad \text{erro na posição } X^{12}$$

$$\beta_2 = \frac{1}{\alpha^{10}} = \alpha^{-10} = \alpha^{5} \quad \Rightarrow \quad \text{erro na posição } X^{5}$$

$$\beta_3 = \frac{1}{\alpha^{12}} = \alpha^{-12} = \alpha^{3} \quad \Rightarrow \quad \text{erro na posição } X^{3}$$

Logo o polinômio localizador de erros (3.11) torna-se:

$$\mathbf{e}(X) = X^3 + X^5 + X^{12}$$

Uma vez obtido o polinômio localizador de erros, basta somar o vetor erro com o vetor recebido para obter-se o vetor código que, provavelmente, foi o vetor transmitido, ou seja:

$$\mathbf{e}(X) = X^3 + X^5 + X^{12} \qquad \Rightarrow \qquad \mathbf{e} = 000101000000100$$

$$\mathbf{c'} = \mathbf{r} + \mathbf{e} = 000101000000100 + 000101000000100$$

$$\mathbf{c'} = 000000000000000.$$

* * *

- **ALGORITMO ITERATIVO DE BERLEKAMP SIMPLIFICADO PARA O CASO BINÁRIO [2][4]**

Para a correção de mais de seis erros em um vetor recebido correspondente a uma palavra-código BCH transmitida o método de Peterson para a solução dos coeficientes de $\sigma(X)$, obtido a partir dos valores das síndromes, tornam-se trabalhoso e ineficiente. Isso se deve ao fato de que o número de operações no campo finito, para a solução dos coeficientes de $\sigma(X)$ cresce aproximadamente com o quadrado do número de erros a ser corrigido. Por esse motivo, a utilização do algoritmo iterativo desenvolvido por Berlekamp, para a solução das identidades de Newton, é mais vantajosa porque requer um número de operações que cresce linearmente com o número de erros a ser corrigido. O algoritmo de Berlekamp será apresentado a seguir sem o seu desenvolvimento matemático assim como as correspondentes suas provas. Maiores detalhes sobre este algoritmo pode ser encontrado em [4]

O algoritmo de Berlekamp consiste da busca iterativa dos coeficientes σ do polinômio localizador de erros $\sigma(X)$, onde o número de iterações para a decodificação dos t erros de uma palavra binária de um código BCH, é igual a t. O número de cada iteração é representado por μ. Assim, o polinômio localizador de erros para a iteração μ é denotado por

$$\sigma^{(\mu)}(X) = 1 + \sigma_1^{(\mu)} X + \sigma_2^{(\mu)} X^2 + \ldots + \sigma_t^{(\mu)} X^t. \tag{3.22}$$

O polinômio localizador de erros para a iteração $\mu + 1$ é determinado por

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + \Delta^{(\mu)} X T^{(\mu)}(X), \tag{3.23}$$

onde $\Delta^{(\mu)}$ é definida como sendo a *discrepância* encontada quando uma versão provisória de $\sigma^{(\mu)}(X)$, construída para uma linha de (3.17), não satisfaz a próxima linha, enquanto $T^{(\mu)}(X)$ é uma *correção polinomial*.

A *discrepância* $\Delta^{(\mu)}$ é definida como

$$\text{o coeficiente de } X^{(2\mu+1)} \text{ no produto } [1+S(X)]\sigma^{(\mu)}(X) \tag{3.24}$$

Enquanto o valor da correção polinomial para a próxima linha é determinado de acordo com as seguintes condições

$$T^{(\mu+1)}(X) = \begin{cases} X^2 T^{(\mu)}(X) & \text{se } \Delta^{(\mu)} = 0 \quad \underline{\textbf{ou}} \quad \text{se o grau de } \sigma^{(\mu)}(X) > \mu \\ \dfrac{X\,\sigma^{(\mu)}(X)}{\Delta^{(\mu)}} & \text{se } \Delta^{(\mu)} \neq 0 \quad \underline{\textbf{e}} \quad \text{o grau de } \sigma^{(\mu)}(X) \leq \mu \end{cases} \tag{3.25}$$

O algoritmo é inicializado na iteração μ = 0 sendo que o polinômio localizador de erros e a *correção polinomial* para este passo é feito, respectivamente,

$$\begin{aligned} \sigma^{(0)}(X) &= 1 \\ T^{(0)}(X) &= 1 \\ \Delta^{(0)} &= S_1 \end{aligned} \tag{3.26}$$

Após essas definições o algoritmo pode ser resumido nos seguintes passos:

1) Comece a construção de uma tabela com quatro colunas que deverão abrigar, respectivamente, os valores μ, $\sigma^{(\mu)}(X)$, $T^{(\mu)}(X)$ e $\Delta^{(\mu)}$, admitindo para a primeira linha, correspondente à iteração μ = 0, os valores apresentados em (3.26), ou seja:

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	$\sigma^{(0)}(X)=1$	$T^{(0)}(X)=1$	$\Delta^{(0)} = S_1$
1			
2			
⋮	⋮	⋮	⋮
t			

2) Determine os valores de $\sigma^{(\mu)}(X)$ e $T^{(\mu)}(X)$ para a próxima linha, ou seja, obtenha os valores $\sigma^{(\mu+1)}(X)$ e $T^{(\mu+1)}(X)$ por meio de (3.23) e (3.25), respectivamente. Complete a linha calculando o valor de $\Delta^{(\mu)}$ usando (3.24). Note que o valor de $\sigma^{(\mu)}(X)$ a ser usado em (3.24) é o valor presente nesta mesma linha.

3) Repita o passo 2 até que o todos os coeficiente σ_i tenham sido encontrados.

EXEMPLO 3.4

Com os dados do Exemplo 3.3 encontre o polinômio localizador de erros $\sigma(X)$ utilizando o algoritmo de Berlekamp simplificado para o caso binário.

SOLUÇÃO

Do Exemplo 3.3 tem-se que $S_1 = 1$; $S_3 = \alpha^{10}$ e $S_5 = \alpha^{10}$ e de (3.19) obtém-se os demais valores de síndromes:

$$S_2 = S_1^2 = 1$$
$$S_4 = S_2^2 = 1$$
$$S_6 = S_3^2 = \left(\alpha^{10}\right)^2 = \alpha^5$$

Assim, pode-se escrever que $S(X) = X + X^2 + \alpha^{10}X^3 + X^4 + \alpha^{10}X^5 + \alpha^5 X^6$

1º Passo: $\mu = 0$

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	$\sigma^{(0)}(X) = 1$	$T^{(0)}(X) = 1$	$\Delta^{(0)} = S_1$

2º Passo: $\mu = \mu + 1 = 1$

De (3.23) obtém-se:

$$\sigma^{(\mu+1)}(X) = \sigma^{(1)}(X) = \sigma^{(0)}(X) + \Delta^{(0)} X T^{(0)}(X) = 1 + S_1 \cdot X \cdot 1$$
$$\sigma^{(1)}(X) = 1 + X$$

De (3.25) obtém-se:

$$T^{(\mu+1)}(X) = T^{(1)}(X) = \frac{X\,\sigma^{(0)}(X)}{\Delta^{(0)}} = \frac{X}{1}$$
$$T^{(1)}(X) = X$$

Conforme (3.24), para valor atualizado $\mu = 1$, deve-se obter o coeficiente de $X^{(2\mu+1)} = X^3$, no produto:

$$[1 + S(X)]\sigma^{(1)}(X) = (1 + X + X^2 + \alpha^{10}X^3 + X^4 + \alpha^{10}X^5 + \alpha^5 X^6)(1 + X).$$

Resolvendo apenas para o termo de terceiro grau, obtém-se:

$$\alpha^{10}X^3 + X \cdot X^2 = (1 + \alpha^{10})X^3 = \alpha^5 X^3$$

Assim o coeficiente procurado é α^5. Logo, $\Delta^{(1)} = \alpha^5$.

Até a iteração $\mu = 1$ a tabela do algoritmo de Berlekamp passa a ser

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	S_1
1	$1+X$	X	α^5

3º Passo: $\mu = \mu + 1 = 2$

De (3.23) obtém-se:

$$\sigma^{(\mu+1)}(X) = \sigma^{(2)}(X) = \sigma^{(1)}(X) + \Delta^{(1)} X T^{(1)}(X) = (1+X) + \alpha^5 X \cdot X$$
$$\sigma^{(2)}(X) = 1 + X + \alpha^5 X^2$$

De (3.25) obtém-se:

$$T^{(\mu+1)}(X) = T^{(2)}(X) = \frac{X\,\sigma^{(1)}(X)}{\Delta^{(1)}} = \frac{X(1+X)}{\alpha^5} = \alpha^{-5}(X + X^2)$$
$$T^{(2)}(X) = \alpha^{10} X + \alpha^{10} X^2$$

Conforme (3.24), para valor atualizado $\mu = 2$, deve-se obter o coeficiente de $X^{(2\mu+1)} = X^5$, no produto $[1+S(X)]\sigma^{(\mu)}(X)$.

Resolvendo apenas para o termo de quinto grau, obtém-se:

$$(1 + X + X^2 + \alpha^{10} X^3 + X^4 + \alpha^{10} X^5 + \alpha^5 X^6)(1 + X + \alpha^5 X^2)$$
$$\alpha^{10} X^5 + X \cdot X^4 + \alpha^5 X^2 \cdot \alpha^{10} X^3 = \alpha^{10} X^5 + X^5 + X^5 = \alpha^{10} X^5$$

Assim o coeficiente procurado é α^{10}. Logo, $\Delta^{(2)} = \alpha^{10}$.

Até a iteração $\mu = 2$ a tabela do algoritmo de Berlekamp passa a ser

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	S_1
1	$1+X$	X	α^5
2	$1+X+\alpha^5X^2$	$\alpha^{10}X+\alpha^{10}X^2$	α^{10}

4º Passo: $\mu = \mu + 1 = 3$

De (3.23) obtém-se:

$$\sigma^{(\mu+1)}(X) = \sigma^{(3)}(X) = \sigma^{(2)}(X) + \Delta^{(2)} X T^{(2)}(X) = 1 + X + \alpha^5 X^2 + \alpha^{10} X(\alpha^{10} X + \alpha^{10} X^2)$$
$$\sigma^{(3)}(X) = 1 + X + \alpha^5 X^2 + \alpha^5 X^2 + \alpha^5 X^3$$
$$\sigma^{(3)}(X) = 1 + X + \alpha^5 X^3$$

Até a iteração $\mu = 3$ a tabela do algoritmo de Berlekamp passa a ser

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	S_1
1	$1+X$	X	α^5
2	$1+X+\alpha^5X^2$	$\alpha^{10}X+\alpha^{10}X^2$	α^{10}
3	$1+X+\alpha^5X^3$	-	-

Assim, $\sigma(X) = \sigma^{(3)}(X) = 1 + X + \alpha^5 X^3$ que é idêntico ao encontrado no Exemplo 3.3.

* * *

Berlekamp mostrou que se o número de erros no bloco recebido não for maior do que *t*, o algoritmo identificará corretamente a localização de todos os erros. Se houver mais do que *t* erros, uma variedade de eventos podem ocorrer. É possível (ainda que raramente) que o algoritmo indique corretamente as posições de erros por meio de um polinômio $\sigma(X)$ com grau maior do que *t*. Muito mais frequentemente, entretanto, o algoritmo irá convergir para um polinômio $\sigma(X)$ com grau *t* ou menor, indicando uma aparente solução correta (embora de fato incorreta).

Quando o número de erros do bloco recebido é menor do que *t*, não são necessárias as *t* iterações para se obter o polinômio localizador de erros. Note que se a discrepância for zero, i.e., $\Delta^{(\mu)} = 0$, então (3.23) torna-se

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$$

e a execução do algoritmo pode ser interrompida. Veja Exemplo 3.5.

EXEMPLO 3.5

Admita que o vetor código todo zero do código BCH (15, 5) gerado por (3.5) tenha sido corrompido por ruído resultando no vetor recebido **r** = 000100000000100. Encontre o polinômio de erro para o vetor recebido usando o algoritmo de Berlekamp.

SOLUÇÃO

O polinômio recebido obtido a partir do vetor recebido **r** é

$$\mathbf{r}(X) = X^3 + X^{12}$$

De onde se obtém as síndromes

$$S_1 = \mathbf{r}(\alpha^1) = \alpha^3 + \alpha^{12} = \alpha^3 + 1 + \alpha + \alpha^2 + \alpha^3 = \alpha^{10} \qquad S_1 = \alpha^{10}$$
$$S_3 = \mathbf{r}(\alpha^3) = \alpha^9 + \alpha^{36} = \alpha + \alpha^3 + \alpha^2 + \alpha^3 = \alpha^5 \qquad S_3 = \alpha^5$$
$$S_5 = \mathbf{r}(\alpha^5) = \alpha^{15} + \alpha^{60} = 0 \qquad S_5 = 0$$

De (3.19) obtém-se os demais valores de síndromes:

$$S_2 = S_1^2 = \left(\alpha^{10}\right)^2 = \alpha^5$$
$$S_4 = S_2^2 = \left(\alpha^5\right)^2 = \alpha^{10}$$
$$S_6 = S_3^2 = \left(\alpha^5\right)^2 = \alpha^{10}$$

Consequentemente, $S(X) = \alpha^{10}X + \alpha^5 X^2 + \alpha^5 X^3 + \alpha^{10} X^4 + \alpha^{10} X^6$.

1º Passo: $\mu = 0$

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	α^{10}

2º Passo: $\mu = \mu + 1 = 1$

De (3.23) obtém-se:

$$\sigma^{(\mu+1)}(X) = \sigma^{(1)}(X) = \sigma^{(0)}(X) + \Delta^{(0)} X T^{(0)}(X) = 1 + \alpha^{10} X \cdot 1$$
$$\sigma^{(1)}(X) = 1 + \alpha^{10} X$$

De (3.25) obtém-se:

$$T^{(\mu+1)}(X) = T^{(1)}(X) = \frac{X\,\sigma^{(0)}(X)}{\Delta^{(0)}} = \frac{X}{\alpha^{10}} = \alpha^{-10} X$$
$$T^{(1)}(X) = \alpha^5 X$$

Conforme (3.24), para $\mu = 1$, deve-se obter o coeficiente de $X^{(2\mu+1)} = X^3$, no produto:

$$[1 + S(X)]\sigma^{(1)}(X) = (1 + \alpha^{10}X + \alpha^5 X^2 + \alpha^5 X^3 + \alpha^{10} X^4 + \alpha^{10} X^6)(1 + \alpha^{10} X).$$

Resolvendo apenas para o termo de terceiro grau, obtém-se:

$$\alpha^5 X^3 + \alpha^{10} X \cdot \alpha^5 X^2 = \alpha^5 X^3 + X^3 = \alpha^{10} X^3$$

Assim o coeficiente procurado é 0. Logo, $\Delta^{(1)} = \alpha^{10}$.

Até a iteração $\mu = 1$ a tabela do algoritmo de Berlekamp passa a ser

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	α^{10}
1	$1 + \alpha^{10} X$	$\alpha^5 X$	α^{10}

3º Passo: $\mu = \mu + 1 = 2$

Como $\Delta^{(1)} = \alpha^{10}$, de (3.23) obtém-se:

$$\sigma^{(\mu+1)}(X) = \sigma^{(2)}(X) = \sigma^{(1)}(X) + \Delta^{(1)} X T^{(1)}(X) = 1 + \alpha^{10} X + \alpha^{10} X \cdot \alpha^{5} X$$
$$\sigma^{(2)}(X) = 1 + \alpha^{10} X + X^2$$

De (3.25) obtém-se:

$$T^{(\mu+1)}(X) = T^{(2)}(X) = \frac{X\,\sigma^{(1)}(X)}{\Delta^{(1)}} = \frac{X(1+\alpha^{10}X)}{\alpha^{10}} = \alpha^{-10} X(1+\alpha^{10}X) = \alpha^{5} X(1+\alpha^{10}X)$$
$$T^{(2)}(X) = \alpha^{5} X + X = \alpha^{10} X$$

Conforme (3.24), para $\mu = 2$, deve-se obter o coeficiente de $X^{(2\mu+1)} = X^5$, no produto:

$$[1+S(X)]\sigma^{(2)}(X) = (1+\alpha^{10}X+\alpha^{5}X^2+\alpha^{5}X^3+\alpha^{10}X^4+\alpha^{10}X^6)(1+\alpha^{10}X+X^2)\,.$$

Resolvendo apenas para o termo de quinto grau, obtém-se:

$$\alpha^{10} X \cdot \alpha^{10} X^4 + \alpha^{5} X^3 \cdot X^2 = \alpha^{5} X^5 + \alpha^{5} X^5 = 0$$

Assim o coeficiente procurado é 0. Logo, $\Delta^{(2)} = 0$.

Resultando, para $\mu = 2$, em

μ	$\sigma^{(\mu)}(X)$	$T^{(\mu)}(X)$	$\Delta^{(\mu)}$
0	1	1	α^{10}
1	$1+\alpha^{10}X$	$\alpha^{5}X$	α^{10}
2	$1+\alpha^{10}X+X^2$	$\alpha^{10}X$	0

Logo, o polinômio localizador de erros (3.15) torna-se

$$\sigma(X) = 1 + \alpha^{10} X + X^2.$$

Pode-se verificar facilmente que apenas α^3 e α^{12} são raízes de $\sigma(X)$. Como as posições de erros são o inverso das raízes do polinômio localizador de erros, então:

$$\beta_1 = \frac{1}{\alpha^3} = \alpha^{-3} = \alpha^{12} \quad \Rightarrow \quad \text{erro na posição } X^{12}$$
$$\beta_2 = \frac{1}{\alpha^{12}} = \alpha^{-12} = \alpha^{3} \quad \Rightarrow \quad \text{erro na posição } X^{3}$$

Logo o polinômio localizador de erros (3.11) torna-se:

$$\mathbf{e}(X) = X^3 + X^{12}$$

* * *

3.4. CONSIDERAÇÕES SOBRE IMPLEMENTAÇÕES [1][2]

A implementação de um codificador para os códigos BCH primitivos binários não apresenta grande dificuldade devido ao fato destes códigos serem cíclicos. Assim sendo, um codificador para códigos BCH primitivos binários é idêntico a um codificador com registradores de deslocamento para códigos cíclicos.

Já a decodificação é um pouco mais complexa. Cada passo da decodificação de um código BCH pode ser implementada tanto por *software* como por hardware. Algumas considerações sobre essas formas de implementação são apresentadas a seguir.

- **ETAPA 1: CÁLCULO DA SÍNDROME**

Esta primeira etapa da decodificação não apresenta maiores dificuldades para a sua implementação em *hardware*. De fato, o circuito para o cálculo das síndromes é semelhante àquele utilizado para os códigos cíclicos. A diferença é que ao invés um circuito para o cálculo da síndrome, são necessários 2t circuitos para o cálculo do conjunto de síndromes dos códigos BCH. Isso pode ser feito considerando que o resto da divisão de $\mathbf{r}(X)$ por $\phi_i(X)$ tem a forma

$$\mathbf{b}_i(X) = b_0 + b_1 X + b_2 X^2 + \ldots + b_{2t\text{-}1} X^{2t\text{-}1},$$

e que

$$S_i(\alpha^i) = \mathbf{b}_i(\alpha^i) = b_0 + b_1 \alpha^i + b_2 \alpha^{i\,2} + \ldots + b_{2t\text{-}1}\, \alpha^{i(2t\text{-}1)}$$

O Exemplo 3.6 apresenta a solução para a determinação das síndromes do código BCH (15, 7) do Exemplo 3.2.

EXEMPLO 3.6

Desenhe um circuito para a determinação das síndromes para o BCH (15, 7), com capacidade de correção de até dois erros (t = 2) do Exemplo 3.2.

SOLUÇÃO

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4$$
e
$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4.$$

$$\begin{aligned}
S_1 = \mathbf{b}_1(\alpha^1) &= b_0 + b_1\alpha + b_2\alpha^2 + \mathrm{b}_3\alpha^3 \\
S_2 = \mathbf{b}_2(\alpha^2) &= b_0 + b_1\alpha^2 + b_2\alpha^4 + b_3\alpha^6 = b_0 + b_1\alpha^2 + b_2(1+\alpha) + b_3(\alpha^2+\alpha^3) \\
&= b_0 + b_1\alpha^2 + b_2 + b_2\alpha + b_3\alpha^2 + b_3\alpha^3 = (b_0+b_2) + b_2\alpha + (b_1+b_3)\alpha^2 + b_3\alpha^3 \\
S_3 = \mathbf{b}_3(\alpha^3) &= b_0 + b_1\alpha^3 + b_2\alpha^6 + b_3\alpha^9 = b_0 + b_1\alpha^3 + b_2(\alpha^2+\alpha^3) + b_3(\alpha+\alpha^3) \\
&= b_0 + b_1\alpha^3 + b_2\alpha^2 + b_2\alpha^3 + b_3\alpha + b_3\alpha^3 = b_0 + b_3\alpha + b_2\alpha^2 + (b_1+b_2+b_3)\alpha^3 \\
S_4 = \mathbf{b}_4(\alpha^4) &= b_0 + b_1\alpha^4 + b_2\alpha^8 + b_3\alpha^{12} = b_0 + b_1(1+\alpha) + b_2(1+\alpha^2) + b_3(1+\alpha+\alpha^2+\alpha^3) \\
&= b_0 + b_1 + b_1\alpha + b_2 + b_2\alpha^2 + b_3 + b_3\alpha + b_3\alpha^2 + b_3\alpha^3 \\
&= b_0 + b_1 + b_2 + b_3 + (b_1+b_3)\alpha + (b_2+b_3)\alpha^2 + b_3\alpha^3
\end{aligned}$$

Que resulta no circuito da Figura 3.1.

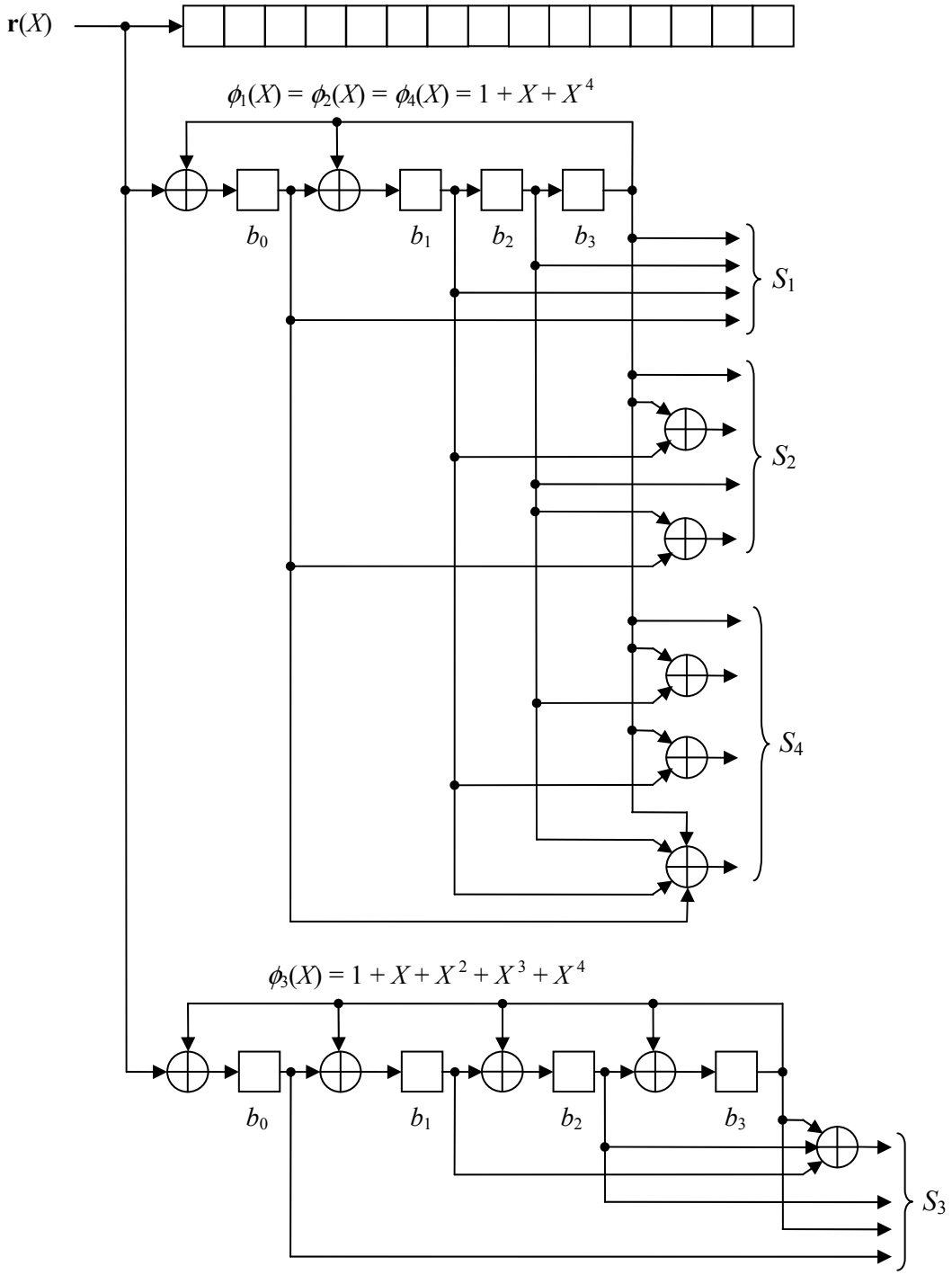

Figura 3.1 - Circuito para a determinação das síndromes para o BCH (15, 7) do Exemplo 3.2.

* * *

A vantagem da implementação em *hardware*, conforme a apresentada no Exemplo 3.6, é a velocidade. Entretanto, implementações em *software* apresentam a vantagem de serem mais baratas.

- **ETAPA 2: DETERMINAÇÃO DO POLINÔMIO LOCALIZADOR DE ERROS**

Nesta etapa da decodificação o custo, a velocidade e a complexidade dependem do algoritmo usado e do tipo de implementação, ou seja, *software* ou *hardware*. Geralmente, a implementação pura em *hardware* torna o processo de determinação do polinômio localizador de erros mais rápido e a velocidade depende de quanto processamento está sendo feito em paralelo. Entretanto, mais uma vez, a implementação em *hardware* tende a ser mais cara do que a implementação por *software*.

Independentemente do algoritmo usado para a implementação desta etapa de decodificação em *hardware* puro, basicamente sua implementação é feita por circuitos que realizam as operações de adição e multiplicação no Campo de Galois. O Capítulo 6 da Referência [1] apresenta algumas alternativas para a realização dessas operações.

- **ETAPA 3: LOCALIZAÇÃO DOS ERROS E CORREÇÕES**

Este passo pode ser implementado em *hardware* usando o circuito de busca de Chien, apresentado na Figura 3.2. Para entender como o circuito funciona, seja o polinômio localizador de erros $\sigma(X)$ dividido por X^t, conforme apresentado a seguir.

$$\frac{\sigma(X)}{X^t} = 1 + \sigma_1 X^{-1} + \sigma_2 X^{-2} + \cdots + \sigma_t X^{-t}$$

O valor de X que satisfaz $\sigma(X) = 0$ satisfaz, consequentemente, a equação

$$\sigma_1 X^{-1} + \sigma_2 X^{-2} + \cdots + \sigma_t X^{-t} = 1$$

Admitindo como convenção que a transmissão ocorra com os bits de ordem mais alta em primeiro lugar, é conveniente aplicar o teste da raiz para o localizador α^{n-1} primeiramente. Note que a substituição de termo X^{-i} em α^{n-1} resulta em α^{-in+i}, o que equivale a α^i considerando o comprimento total do código BCH, uma vez que $\alpha^n = 1$ e portanto $\alpha^{-in} = 1$. Consequentemente, o teste de α^{n-1} como uma possível raiz de $\sigma(X)$ é o mesmo teste para

$$\sigma_1 \alpha^1 + \sigma_2 \alpha^2 + \cdots + \sigma_t \alpha^t \overset{?}{=} 1$$

e, em geral, o teste para α^{n-j} como um localizador de erro é equivalente à verificação se α^j satisfaz ou não

$$\sum_{i=1}^{t} \sigma_i \alpha^{il} = 1 \quad \text{para } j = 0, 1, \ldots, n-1$$

Conforme mostrado na Figura 3.2, o circuito de busca de Chien realiza essa sequência de testes da seguinte forma:

1. Os registradores de $\sigma(X)$ são carregados com seus respectivos valores σ_i;
2. Cada valor σ_i é multiplicado por seu correspondente α^i e os resultados são recarregados nos registradores $\sigma(X)$ que são somados e o resultado comparado com a unidade;
3. O passo 2 é repetido n vezes de modo que o valor atualizado a cada uma das n multiplicações seja sempre comparado com a unidade;
4. A cada repetição da operação descrita no passo 2 a palavra recebida armazenada no registrador de deslocamento é deslocada de um bit de forma que o bit na posição mais à direita do registrador de deslocamento é somado com 1 se estiver errado ou com zero se estiver correto.

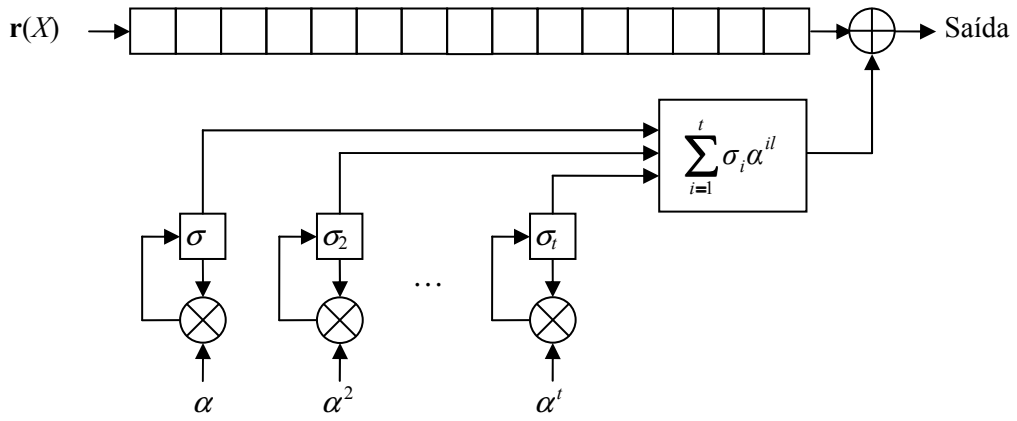

Figura 3.2 - Circuito de busca de Chien

Se o número de erros na palavra recebida for igual a t, então o circuito de busca de Chien localizará e corrigirá corretamente os erros. Entretanto, quando o circuito de busca de Chien encontrar menos que l raízes, quando dado os coeficientes de um polinômio localizador de erros de grau l, isto significa que o polinômio não tem todas as suas raízes para a localização de erros. Assim sendo, o polinômio não é um legítimo polinômio localizador de l-erros.

Este tipo de evento é, de fato, a indicação mais comum de um padrão de erros detectado e incorrigível, i.e., um padrão contendo mais do que t erros que não é usado para estimar, na decodificação, uma palavra código errada. No entanto, em certos casos, padrões com mais do que t erros escapam da detecção do circuito de busca de Chien e por esta razão uma verificação adicional é necessária.

Quando os coeficientes do polinômio localizador de erros encontrados na segunda etapa de decodificação indicam a presença de $l < t$ erros, e o circuito de busca de Chien realiza l correções de bit, a palavra resultante deve ser testada por meio das equações de síndrome para validação da palavra código. Uma ou falha no teste das síndromes indica um padrão de erros detectável mas não corrigível.

Para um estudo mais aprofundado da segunda e da terceira etapas de decodificação, recomenda-se uma leitura cuidadosa do Capítulo 6 da Referência [1].

3.5. EXERCÍCIOS

1. Determine o polinômio gerador do código BCH com capacidade de correção de cinco erros ($t = 5$) a partir do $GF(2^5)$ gerado por $p(X) = 1 + X^2 + X^5$.

2. Admita que uma palavra código do Exercício 1 foi transmitida e o polinômio recebido tenha sido $\mathbf{r}(X) = X^{23} + X^{21} + X^{16} + X^{13} + X^{10} + X^7 + X^5 + X^4 + X^3 + 1$. Decodifique o polinômio recebido pelo algoritmo de Peterson e também pelo algoritmo de Berlekamp.

3.6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] LIN, S.; COSTELO JR, D. J. *Error Control Coding: Fundamentals and Applications*. Upper Saddle River, New Jersey: Prentice Hall, 2ª ed. 2004.

[2] MICHELSON, A. M.; LEVESQUE, A. H. *Error-control Techniques for Digital Communication.* New York: John Wiley & Sons, 1984.

[3] PETERSON, W. W.; WELDON, E. J. *Error-Correcting Codes.* Cambridge, Massachusetts: MIT Press, 1972.

[4] BERLEKAMP, E. R. *Algebric Coding Theory*. New York: McGraw-Hill, 1968.

ANEXO 3.1

POLINÔMIOS MÍNIMOS DE ELEMENTOS DE GF(2^m)

Os polinômios mínimos estão representados por seus expoentes entre parênteses e o número apresentado à esquerda dos expoentes é o número do polinômio mínimo. Por exemplo, para m = 4 a notação

3 (0, 2, 3) significa $\phi_3 = 1 + X^2 + X^3$.

Como, α^i e todos os seus conjugados tem o mesmo polinômio mínimo, então para m = 4 o polinômio acima é o polinômio mínimo de

$$\alpha, (\alpha^3)^{2^1} = \alpha^6, (\alpha^3)^{2^2} = \alpha^{12}, (\alpha^3)^{2^3} = \alpha^{24} = \alpha^9.$$

TABELA DOS POLINÔMIOS MÍNIMOS DE ELEMENTOS DE GF(2^M) PARA $1 < M < 8$

***m* = 2**			
1 (0, 1, 2)	-	-	-
***m* = 3**			
1 (0, 1, 3)	3 (0, 2, 3)	-	-
***m* = 4**			
1 (0, 1, 4)	3 (0, 1, 2, 3, 4)	5 (0, 1, 2)	7 (0, 3, 4)
***m* = 5**			
1 (0, 2, 5)	3 (0, 2, 3, 4)	5 (0, 1, 2, 4, 5)	7 (0, 1, 2, 3, 5)
11 (0, 1, 3, 4, 5)	13 (0, 3, 5)	-	-
***m* = 6**			
1 (0, 1, 6)	3 (0, 1, 2, 4, 6)	5 (0, 1, 2, 5, 6)	7 (0, 3, 6)
9 (0, 2, 3)	11 (0, 2, 3, 5, 6)	13 (0, 1, 3, 4, 6)	15 (0, 2, 4, 5, 6)
21 (0, 1, 2)	23 (0, 1, 4, 5, 6)	27 (0, 1, 3)	31 (0, 5, 6)
***m* = 7**			
1 (0, 3, 7)	3 (0, 1, 2, 3, 7)	5 (0, 2, 3, 4, 7)	7 (0, 1, 2, 4, 5, 6, 7)
9 (0, 1, 2, 3, 4, 5, 7)	11 (0, 2, 4, 6, 7)	13 (0, 1, 7)	15 (0, 1, 2, 3, 5, 6, 7)
19 (0, 1, 2, 6, 7)	21 (0, 2, 5, 6, 7)	23 (0, 6, 7)	27 (0, 1, 4, 6, 7)
29 (0, 1, 3, 5, 7)	31 (0, 4, 5, 6, 7)	43 (0, 1, 2, 5, 7)	47 (0, 3, 4, 5, 7)
55 (0, 2, 3, 4, 5, 6, 7)	63 (0, 4, 7)	-	-
***m* = 8**			
1 (0, 2, 3, 4, 8)	3 (0, 1, 2, 4, 5, 6, 8)	5 (0, 1, 4, 5, 6, 7, 8)	7 (0, 3, 5, 6, 8)
9 (0, 2, 3, 4, 5, 7, 8)	11 (0, 1, 2, 5, 6, 7, 8)	13 (0, 1, 3, 5, 8)	15 (0, 1, 2, 4, 6, 7, 8)
17 (0, 1, 4)	19 (0, 2, 5, 6, 8)	21 (0, 1, 3, 7, 8)	23 (0, 1, 5, 6, 8)
25 (0, 1, 3, 4, 8)	27 (0, 1, 2, 3, 4, 5, 8)	29 (0, 2, 3, 7, 8)	31 (0, 2, 3, 5, 8)
37 (0, 1, 2, 3, 4, 6, 8)	39 (0, 3, 4, 5, 6, 7, 8)	43 (0, 3, 5, 7, 8)	45 (0, 3, 4, 5, 8)
47 (0, 3, 5, 7, 8)	51 (0, 1, 2, 3, 4)	53 (0, 1, 2, 7, 8)	55 (0, 4, 5, 7, 8)
59 (0, 2, 3, 6, 8)	61 (0, 1, 2, 3, 6, 7, 8)	63 (0, 2, 3, 4, 6, 7, 8)	85 (0, 1, 2)
87 (0, 1, 5, 7, 8)	91 (0, 2, 4, 5, 6, 7, 8)	95 (0, 1, 2, 3, 4, 7, 8)	111 (0, 1, 3, 4, 5, 6, 8)
119 (0, 3, 4)	127 (0, 4, 5, 6, 8)	-	-

ANEXO 3.2

POLINÔMIOS GERADORES DE CÓDIGOS BCH PRIMITIVOS BINÁRIOS DE COMPRIMENTO ATÉ $2^8 - 1$

Os polinômios apresentados neste anexo estão representados na forma octal. Cada dígito na representação está codificado como se segue:

$0 \leftrightarrow 000$ $1 \leftrightarrow 001$ $2 \leftrightarrow 010$ $3 \leftrightarrow 011$ $4 \leftrightarrow 100$ $5 \leftrightarrow 101$ $6 \leftrightarrow 110$ $7 \leftrightarrow 111$

onde os dígitos binários são os coeficientes do polinômio, com o termo de ordem mais alta mais à esquerda. Por exemplo, para o código BCH (15, 5), obtém-se, da tabela a seguir, a representação octal 2467, que na forma binária fica

010 100 110 111

Resultando em

$$g(X) = X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1.$$

TABELA DOS POLINÔMIOS GERADORES DE CÓDIGOS BCH PRIMITIVOS BINÁRIOS DE COMPRIMENTO ATÉ $2^8 - 1$

n	*k*	*t*	Polinômio Gerador
7	4	1	13
15	11	1	23
	7	2	721
	5	3	2467
31	26	1	45
	21	2	3551
	16	3	107657
	11	5	5423325
	6	7	313365047
63	57	1	103
	51	2	12471
	45	3	1701317
	39	4	166623567
	36	5	1033500423
	30	6	157464165547
127	120	1	211
	113	2	41567
	106	3	11554743
	99	4	3447023271
	92	5	624730022327
	85	6	130704476322273
	78	7	26230002166130115
	71	9	6255010713253127753
	64	10	1206534025570773100045

Continuação.

n	*k*	*t*	Polinômio Gerador
127	57	11	3352652525057050535177 21
	50	13	5444651252331401242150142 1
	43	14	1772177221365122752122057434 3
	36	15	3146074666522075044764574721735
	29	21	4031144613676706036675301411761 55
	22	23	1233760704047225224354544626637647043
	15	27	2205704244560455477052301376221760435 3
	8	31	7047264052751030651476224271567733130217
255	247	1	435
	239	2	267543
	231	3	156720665
	223	4	75626641375
	215	5	23157564726421
	207	6	16176560567636227
	199	7	7633031270420722341
	191	8	2663470176115333714567
	187	9	52755313540001322236351
	179	10	22624710717340432416300455
	171	11	15416214212342356077061630637
	163	12	75004155100756015515747245146 01
	155	13	37575130054076650157225064646776 33
	147	14	16421301735371655253041653054410117 11
	139	15	46140173206017556157072273024745356744 5
	131	18	21571333147151015126125027744214202416 54
	123	19	12061405224206600371721032651614122627250626 7
	115	21	60526665572100247263636404600276352556313472737
	107	22	22205772322066256312417300235347420176574750154441
	99	23	10656667253473174222741416201574332252411076432303431
	91	25	67502650303274441727236317247325110755507627207243445 61
	87	26	11013676341474323643523163430717204620672254527331172131 7
	79	27	66700035637657500020270344207366174621015326711766541342355
	71	29	24024710520644321515554172112332163205444250362557643221706 035
	63	30	10754475055163544325315217357707003666111726455267613656702 543301
	55	31	73154252035011101330152753060320543254143267550105570444260 35473617
	47	42	25335420170626465630330413774062331751233341454460450050660 24552543173

Continuação.

n	*k*	*t*	Polinômio Gerador
255	45	43	1520205605523416113110134637642370156367002447076237303320215702505154 1
	37	45	5136330255067007414177447245437530420735706174323432347644354737403044003
	29	47	3025715536673071465527064012361377115342242324201174114060254657410403565037
	21	55	1256215257060332656001773153607612103227341405653074542521153121614466513473725
	13	59	4641732005052564544426573714250066004330677445476561403174677213570261344605005 47
	9	63	1572602521747246320103104325535513461416236721204407454511276611554770556167751605 7