13. BCH码

引言

- Hocquenghem在1959 年，Bose和Chaudhari 在1960年，分别独立提出BCH 码
- Peterson在1960年证明了BCH码的循环结构
- Gorenstein和Zierler在1961年将二进制BCH码推广到具有p^m个符号的码
- 在非二进制BCH码中，最重要的子类是RS码，由Reed和Solomon在1960年提出
- 二进制BCH码的第一种译码算法由Peterson于1960年提出，该算法被Gorenstein、Zierler、Chien、Forney、Berlekamp、Massey、Burton等许多研究人员改进，其中Berlekamp的迭代算法和Chien的搜索算法最为有效

目录

- 从扩展域的观点看循环汉明码
- BCH码的构造
- BCH码的译码
- BCH码应用实例
- Reed Solumn码

α^0	α^0	0 0 1
α^1	α^1	0 1 0
α^2	α^2	1 0 0
α^3	$\alpha+1$	0 1 1
α^4	$\alpha^2+\alpha$	1 1 0
α^5	$\alpha^2+\alpha+1$	1 1 1
α^6	α^2+1	1 0 1

从扩展域的观

- (7，4)循环Hamming
  有非零3维列矢量的一
- 也可以看作GF(2^3)域
  有序排列，这种排列若写作GF(2^3)本原元的升幂形式，有

(设本原多项式为$p(x)=x^3+x+1$）

$$H=\left(\alpha^{0},\alpha^{1},\cdots,\alpha^{5},\alpha^{6}\right)=\begin{bmatrix}1&0&0&1&0&1&1\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}$$

从扩展域的观点看循环HAMMING码

- (7，4)循环Hamming码的校验方程为

$$\mathbf{H}\mathbf{v}^T = 0 \qquad \Rightarrow \sum_{i=0}^{6} v_i \alpha^i = 0$$

$$v(x) = \sum_{i=0}^{6} v_i x^i \qquad \Rightarrow v(x=\alpha) = 0.$$

- (7，4) 循环Hamming 码的码多项式以本原元为根
- (7，4)循环Hamming码以本原多项式为生成多项式

从扩展域的观点看循环HAMMING码

- 从扩展域的观点看二进制循环Hamming码 $(n=2^m-1,\ k=2^m-m-1)$：
  - 生成多项式为$GF(2^m)$域的本原多项式
  - 校验矩阵为

$$\mathbf{H}=\begin{pmatrix}\alpha^0 & \alpha^1 & \alpha^2 & \cdots & \alpha^{2^m-2}\end{pmatrix}$$

从扩展域的观点看循环HAMMING码

- 从扩展域的观点看q进制Hamming code:

  $n = (q^m-1)/(q-1)$

  $k = n-m$

  - 设α是GF(q^m)的本原元, 有 $\alpha^{q^m-1}=1$
  - 令 $\beta = \alpha^{q-1}$, β 的阶为n，有 $n = (q^m-1)/(q-1)$
  - q进制Hamming Code的校验矩阵为:

    $$\mathbf{H} = \left(\beta^0, \beta^1, \cdots, \beta^{n-1}\right)$$

  - β在GF(q)上的最小多项式为生成多项式

目录

纠正T个错误的BCH码的构造

- **BCH码的校验矩阵H**

定理13.1：设α_0, α_1, …, α_{n-1}是GF(2^m)中所有非零元素的一种排列，$n=2^m-1$

t是大于零的正整数，且$t < 2^{m-1}-1$ ($2t+1<n$)

则能纠t个错的二进制 (n, k)线性码的校验矩阵为：

$$\mathbf{H}=\begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0^{\ 3} & \alpha_1^{\ 3} & \cdots & \alpha_{n-1}^{\ 3} \\ & & \cdots & \\ \alpha_0^{\ 2t-1} & \alpha_1^{\ 2t-1} & \cdots & \alpha_{n-1}^{\ 2t-1} \end{bmatrix}$$

纠正T个错误的BCH码的构造

BCH码的校验矩阵H

定理证明： 将二进制(n, k)线性码记作

$$\mathbf{v}=(v_0,v_1,\cdots,v_{n-1})\in C$$

由 $\mathbf{Hv}^T=0$

得 $$\sum_{i=0}^{n-1} v_i \alpha_i^{\,j}=0,\quad j=1,3,\cdots,2t-1$$

将上述校验方程及其等价方程联立有

$$\sum_{i=0}^{n-1} v_i \alpha_i^{\,j}=0,\quad j=1,2,\cdots,2t$$

2019/3/25

BCH码的校验矩阵H

定理证明（续）：相应的校验矩阵是

$$
\mathbf{H'}=\begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0{}^2 & \alpha_1{}^2 & \cdots & \alpha_{n-1}{}^2 \\ & & \cdots & \\ \alpha_0{}^{2t} & \alpha_1{}^{2t} & \cdots & \alpha_{n-1}{}^{2t} \end{bmatrix}
$$

从**H**’中任选2t 列组成的2t×2t 维行列式**B**

纠正T个错误的BCH码的构造

BCH码的校验矩阵H

定理证明（续）：根据Vandermonde行列式定理

可得

$$\mathbf{B}=\begin{vmatrix}\beta_1 & \beta_2 & \cdots & \beta_{2t}\\ \beta_1^{\ 2} & \beta_2^{\ 2} & \cdots & \beta_{2t}^{\ 2}\\ & & \cdots & \\ \beta_1^{\ 2t} & \beta_2^{\ 2t} & \cdots & \beta_{2t}^{\ 2t}\end{vmatrix}=\beta_1\cdots\beta_{2t}\prod_{i<j}(\beta_j-\beta_i)\neq 0$$

式中$\beta_1, \ldots, \beta_{2t}$是从$\alpha_0, \ldots, \alpha_{n-1}$中选出的2t个元素

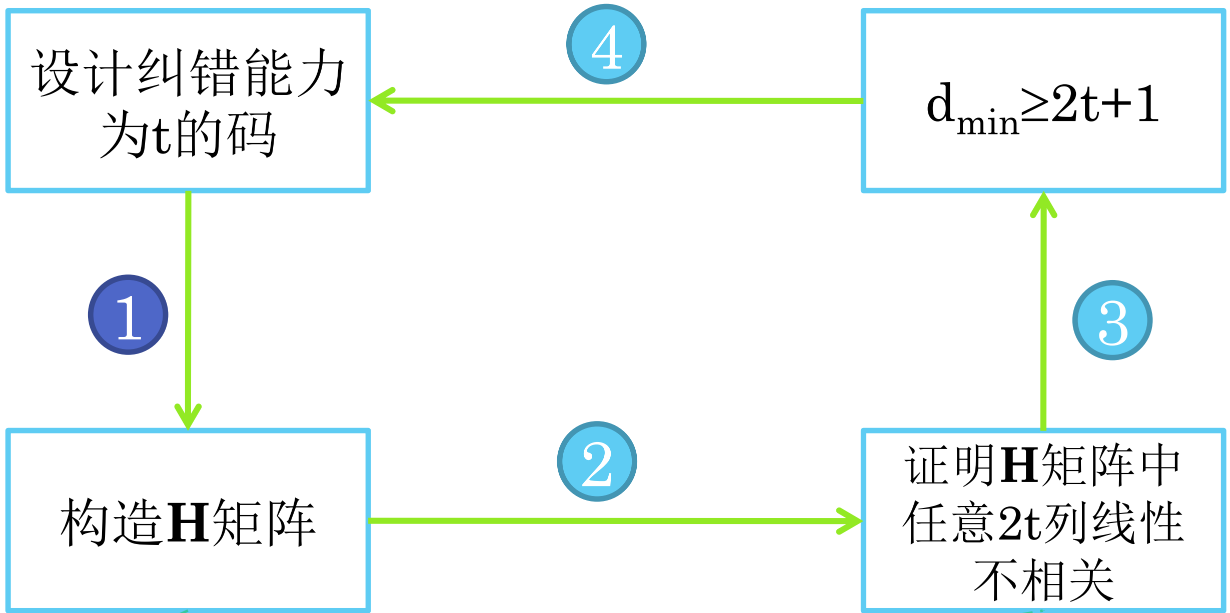
→ 显然**H**中任意2t列互不相关

→ 该线性码的码间最小距离大于2t

→ 它可以纠正 t个错

纠正T个错误的BCH码的构造

- 小结



$$\mathbf{H}=\begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_{n-1}\\ \alpha_0^{\ 3} & \alpha_1^{\ 3} & \cdots & \alpha_{n-1}^{\ \ 3}\\ & & \cdots & \\ \alpha_0^{\ 2t-1} & \alpha_1^{\ 2t-1} & \cdots & \alpha_{n-1}^{\ \ 2t-1}\end{bmatrix}_{t\times n} \Rightarrow \mathbf{H}'=\begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_{n-1}\\ \alpha_0^{\ 2} & \alpha_1^{\ 2} & \cdots & \alpha_{n-1}^{\ \ 2}\\ & & \cdots & \\ \alpha_0^{\ 2t} & \alpha_1^{\ 2t} & \cdots & \alpha_{n-1}^{\ \ 2t}\end{bmatrix}_{2t\times n} \Rightarrow \mathbf{B}=\begin{vmatrix}\beta_1 & \beta_2 & \cdots & \beta_{2t}\\ \beta_1^{\ 2} & \beta_2^{\ 2} & \cdots & \beta_{2t}^{\ 2}\\ & & \cdots & \\ \beta_1^{\ 2t} & \beta_2^{\ 2t} & \cdots & \beta_{2t}^{\ 2t}\end{vmatrix}_{2t\times 2t} = \beta_1\cdots\beta_{2t}\prod_{i<j}(\beta_j-\beta_i)\neq 0$$

2019/3/25

纠正T个错误的BCH码的构造

BCH码的生成多项式

本原（或狭义）BCH码：

- 假如选择 $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$的排序正好是本原元α的升幂排列 $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{n-1}$，则构成本原BCH码，其中$n=2^m-1$
- 此时校验方程为：

$$\sum_{i=0}^{n-1} v_i (\alpha^j)^i = 0 \quad j = 1,3,\cdots,2t-1$$

对比$\sum_{i=0}^{n-1} v_i \alpha_i^{\ j} = 0, \quad j = 1,3,\cdots,2t-1$

纠正T个错误的BCH码的构造

BCH码的生成多项式

本原（或狭义）BCH码（续）：

- 码多项式以 $\alpha^1, \alpha^3, \ldots, \alpha^{2t-1}$为根
- 码多项式v(x) 以

  $LCM\{m_1(x),\ m_3(x), \ldots m_{2t-1}(x)\}$为生成多项式

  其中LCM{ }为最小公倍式

  $m_1(x), m_3(x), \ldots m_{2t-1}(x)$分别是 $\alpha^1, \alpha^3, \ldots, \alpha^{2t-1}$ 的最小多项式

纠正T个错误的BCH码的构造

- **BCH码的生成多项式**

t =7?

例13.1:设计 $n = 2^4 - 1 = 15$, $t = 3$的本原BCH码，

设α是GF(2^4)的本原元，本原多项式x^4+x+1.

$g(x) = \mathrm{LCM}\{m_1(x),\ m_3(x),\ m_5(x)\}$

$$m_1(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^4)(x+\alpha^8) = x^4 + x + 1$$

$$m_3(x) = (x+\alpha^3)(x+\alpha^6)(x+\alpha^{12})(x+\alpha^9) = x^4 + x^3 + x^2 + x + 1$$

$$m_5(x) = (x+\alpha^5)(x+\alpha^{10}) = x^2 + x + 1$$

$$g(x) = m_1(x)\cdot m_3(x)\cdot m_5(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

$$\deg[g(x)] = 10$$

$$k = 15 - \deg[g(x)] = 5$$

纠正T个错误的BCH码的构造

BCH码的生成多项式

码长$n \neq 2^m-1$（$n \mid 2^m-1$）的BCH码

- 令β是GF(2^m)中一个n阶元素，n是2^m-1的因子
- 令g(x)是以β，β^2，…，β^{2t}作为根的最低次二元多项式
- 令$m_1(x)$, $m_2(x)$, …$m_{2t}(x)$分别是 β^1, β^2, …, β^{2t}的最小多项式

纠正T个错误的BCH码的构造

BCH码的生成多项式

码长$n \neq 2^m-1$ ($n \mid 2^m-1$)的BCH码（续）

- $g(x)=LCM\{m_1(x),\ m_2(x), \ldots m_{2t}(x)\}$
- 因为$\beta^n=1$，$\beta, \beta^2, \ldots, \beta^{2t}$是$x^n+1$的根，可知g(x)是$x^n+1$的因式
- 用g(x)生成的循环码是码长为n的纠t个错误的BCH码

目录

- 从扩展域的观点看循环汉明码
- BCH码的构造
- BCH码的译码
- BCH 码应用实例
- Reed Solumn码

BCH码的译码

- BCH码可以纠正t个错误, 它的码多项式v(x)以$\alpha^1, \alpha^2, \ldots, \alpha^{2t}$为根
- 设接收多项式r(x) = v(x) + e(x)

  其中e(x)为错误图案多项式

  定义 $$e(x)=\sum_{i=0}^{n-1} e_i x^i = \sum_{l=1}^{\gamma} Y_l x^{i_l}$$

  式中 γ 为错误的个数

  i_l(l = 1, 2, …, γ)表示错误位置

  $Y_l(\neq 0)$为错误的数值

BCH码的译码

- 定义接收多项式r(x)的伴随多项式

$$s(x)=\sum_{j=1}^{2t}s_j x^{j-1}$$

其中 $s_j=r(x=\alpha^j)$

$$=e(x=\alpha^j)=\sum_{l=1}^{\gamma}Y_l X_l^j$$

$X_l=\alpha^{i_l}$ 与错误位置有关

- 译码就是由伴随多项式s(x)解出与错误位置有关的X_l和错误的数值Y_l

- 定义**错误位置多项式**

$$\sigma(x)=\prod_{l=1}^{\gamma}(1-X_l x)=\sigma_0+\sigma_1 x+\cdots+\sigma_\gamma x^\gamma$$

它以 X_l^{-1}为根，即

$$0=\sigma_0+\sigma_1 X_l^{-1}+\cdots+\sigma_\gamma X_l^{-\gamma}$$

方程式两端各项乘以 $Y_l X_l^{j+\gamma}$ ，则有

$$0=Y_l(\sigma_0 X_l^{j+\gamma}+\sigma_1 X_l^{j+\gamma-1}+\cdots+\sigma_\gamma X_l^{j})$$

- 上式对于$l = 1, 2, \ldots, \gamma$ 均成立, 相加求和, 可以得到

$$0=\sum_{l=1}^{\gamma}\sigma_0 Y_l X_l^{\,j+\gamma}+\sum_{l=1}^{\gamma}\sigma_1 Y_l X_l^{\,j+\gamma-1}+\cdots+\sum_{l=1}^{\gamma}\sigma_\gamma Y_l X_l^{\,j}$$

- 注意到 $\sigma_0=1$ 和s_j的定义,

$$s_j=e(x=\alpha^j)=\sum_{l=1}^{\gamma}Y_l X_l^{\,j}$$

有 $0=s_{\gamma+j}+\sigma_1 s_{\gamma+j-1}+\cdots+\sigma_\gamma s_j$

其中 $j = 1, 2, \ldots, \gamma$.

- 将上述结果写成下面的方程组

$$\begin{bmatrix} -s_{\gamma+1} \\ -s_{\gamma+2} \\ \vdots \\ -s_{2\gamma} \end{bmatrix} = \begin{bmatrix} s_{\gamma} & s_{\gamma-1} & \cdots & s_1 \\ s_{\gamma+1} & s_{\gamma} & \cdots & s_2 \\ \vdots & \vdots & \vdots & \vdots \\ s_{2\gamma-1} & s_{2\gamma-2} & \cdots & s_{\gamma} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{\gamma} \end{bmatrix}$$

  - 当错误个数为 γ时，**[s]** 满秩，**[s]**$^{-1}$ 存在, 可求出 $\sigma_1, \sigma_2, \ldots, \sigma_{\gamma}$., 从而确定错误位置多项式$\sigma(x)$
  - 当错误个数 $< \gamma$ 时, **[s]**不满秩, 需减秩重新求解
  - 若方程组无解, 说明错误个数大于 γ

Massey迭代法

BCH码的译码

- 求解方程σ(x)=0的根, 确定错误位置X_l.
- 再求解错误数值方程组确定Y_l , 方程组如下

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_\gamma \end{bmatrix} = \begin{bmatrix} X_1 & X_2 & \cdots & X_\gamma \\ X_1^{\ 2} & X_2^{\ 2} & \cdots & X_\gamma^{\ 2} \\ \vdots & \vdots & \vdots & \vdots \\ X_1^{\ \gamma} & X_2^{\ \gamma} & \cdots & X_\gamma^{\ \gamma} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\gamma \end{bmatrix}$$

- 最后由Y_l , X_l得到错误多项式e(x)
- 译码输出为

$$\hat{v}(x) = r(x) - e(x)$$

Forney算法

- **BCH码译码小结：**
  - 利用接收多项式r(x)，计算伴随多项式s(x)
  - 由伴随多项式系数，求错误位置多项式σ(x)的系数（同时确定错误位数！）
  - 通过求解σ(x)=0来确定错误位置
  - 进一步计算错误数值
  - 确定错误图案多项式e(x)
  - 译码输出 $\hat{v}(x)=r(x)-e(x)$

BCH码的译码

$r(x)$ → 计算件随多项式s(x) → 确定错误图案多项式e(x) → $\hat{v}(x)=r(x)-e(x)$ ⑥

① $s(x)=\sum_{j=1}^{2t} s_j x^{j-1}$

$s_j=r(x=\alpha^j)=\sum_{l=1}^{\gamma} Y_l X_l^j$

⑤ $e(x)=\sum_{i=0}^{n-1} e_i x^i=\sum_{l=1}^{\gamma} Y_l x^{i_l}$

② 求错误位置多项式σ(x)系数 → γ 为错误的位数

③ 求解方程σ(x)=0 → i_l 表示错误位置

④ 求解方程组：$s_j=\sum_{l=1}^{\gamma} Y_l X_l^j$ → Y_l 为错误的数值

- 例13.2: (15, 5)二元BCH码以α, α^3, α^5为根. 若接收多项式$r(x)=x^9+x^8+x^6+x^4+x^3+1$ 试进行译码.

解:首先求伴随式系数

$$s_1=\alpha^{14} \qquad s_2=\alpha^{13} \qquad s_3=\alpha^{14}$$

$$s_4=\alpha^{11} \qquad s_5=1 \qquad s_6=\alpha^{13}$$

于是有

$$\begin{bmatrix} s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2 \\ s_5 & s_4 & s_3 \end{bmatrix}\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix}=\begin{bmatrix} s_4 \\ s_5 \\ s_6 \end{bmatrix}$$

BCH码的译码

0	0	0 0 0 0
α^0	1	0 0 0 1
α^1	α	0 0 1 0
α^2	α^2	0 1 0 0
α^3	α^3	1 0 0 0
α^4	$\alpha+1$	0 0 1 1
α^5	$\alpha^2+\alpha$	0 1 1 0
α^6	$\alpha^3+\alpha^2$	1 1 0 0
α^7	$\alpha^3+\alpha+1$	1 0 1 1
α^8	α^2+1	0 1 0 1
α^9	$\alpha^3+\alpha$	1 0 1 0
α^{10}	$\alpha^2+\alpha+1$	0 1 1 1
α^{11}	$\alpha^3+\alpha^2+\alpha$	1 1 1 0
α^{12}	$\alpha^3+\alpha^2+\alpha+1$	1 1 1 1
α^{13}	$\alpha^3+\alpha^2+1$	1 1 0 1
α^{14}	α^3+1	1 0 0 1

- 例解(续):

  计算方程系数矩阵的行列式: $|\mathbf{s}| = 0$

  可知实际出错个数 $\gamma < 3$, 有$\sigma_3 = 0$

  降阶 $\begin{bmatrix} s_2 & s_1 \\ s_3 & s_2 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} s_3 \\ s_4 \end{bmatrix}$

  计算方程组系数的行列式: $|\mathbf{s}| \neq 0$

  求得 $\sigma_1 = \alpha^{14} \qquad \sigma_2 = \alpha^6$

- 例解(续):

可知错误位置多项式 $\sigma(x)=1+\alpha^{14}x+\alpha^{6}x^{2}$

它的根为 $X_1^{-1}=\alpha^4$ $X_2^{-1}=\alpha^5$

相应错误位置: $X_1=\alpha^{11}$ $X_2=\alpha^{10}$

对于二元BCH码，相应的错误数值是1，

因此错误图案是：$e(x)=x^{11}+x^{10}$

译码输出 $\hat{v}(x)=r(x)-e(x)$

$$=x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+x^{4}+x^{3}+1$$

目录

- 从扩展域的观点看循环汉明码
- BCH码的构造
- BCH码的译码
- BCH 码应用实例
- Reed Solumn码

BCH码应用实例

- **（508，472）BCH截短码**

  原始码是参数为（511，475）的系统BCH码，

  纠错能力: t=4

  生成多项式为：

$$g_1(x)=x^{36}+x^{35}+x^{34}+x^{31}+x^{30}+x^{25}+x^{23}+x^{21}+x^{20}+x^{19}+x^{16}+x^{15}+x^{11}+x^{8}+x^{7}+x^{5}+1$$

  生成多项式的八进制表示形式为：

  $（1630256304641）_8$

255	87	26	11013676341474323643523163430717204620672254527331172131 7
255	79	27	6670003563765750002027034420736617462101532671176654134235 5
255	71	29	24024710520644321515554172112331163205444250362557643221706035
255	63	30	107544750551635443253152173577070036661117264552676136567025433 01
255	55	31	7315425203501100133015275306032054325414326755010557044426035473617
255	47	42	253354201706264656303304137740623317512333414544604500506602455254317 3
255	45	43	1520205605523416113110134637642370156367002447076237303320215702505154 1
255	37	45	51363302550670074141774472454375304207357061743234323476443547374030440 03
255	29	47	30257155366730714655270640123613771153422423242011741140602546574104035650 37
255	21	55	1256215257060332656001773153607612103227341405653074542521153121614466513473725
255	13	59	46417320050525645444265737142500660043306774454765614031746772135702613446050054 7
255	9	63	157260252174724632010310432553551346141623672120440745451127661155477055616775160 57
511	502	1	1021
511	493	2	1112711
511	484	3	1530225571
511	475	4	1630256304641
511	466	5	1112724662161763
511	457	6	1142677410335765707
511	448	7	1034122337164372224005
511	439	8	1561350064670543777423345
511	430	9	1727400306127620173461431627
511	421	10	1317711625267264610360644707513
511	412	11	1337530164410305712316173767147101
511	403	12	1573436303657311762726657724651203651
511	394	13	1102510344130333354270407474305341234033
511	385	14	1775546025777712372455452107300530331444031

BCH码应用实例

- **（504，432）BCH截短码**

  原始码是参数为（511，439）的系统BCH码，

  纠错能力: t=8

  生成多项式为：

$$g_2(x)=x^{72}+x^{71}+x^{69}+x^{68}+x^{67}+x^{63}+x^{61}+x^{60}+x^{59}+x^{57}+x^{50}+x^{49}+x^{47}+x^{44}+x^{43}+x^{41}$$
$$+x^{40}+x^{39}+x^{35}+x^{33}+x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{24}+x^{23}+x^{22}+x^{21}+x^{20}+x^{19}+x^{18}+x^{17}$$
$$+x^{13}+x^{10}+x^{9}+x^{7}+x^{6}+x^{5}+x^{2}+1$$

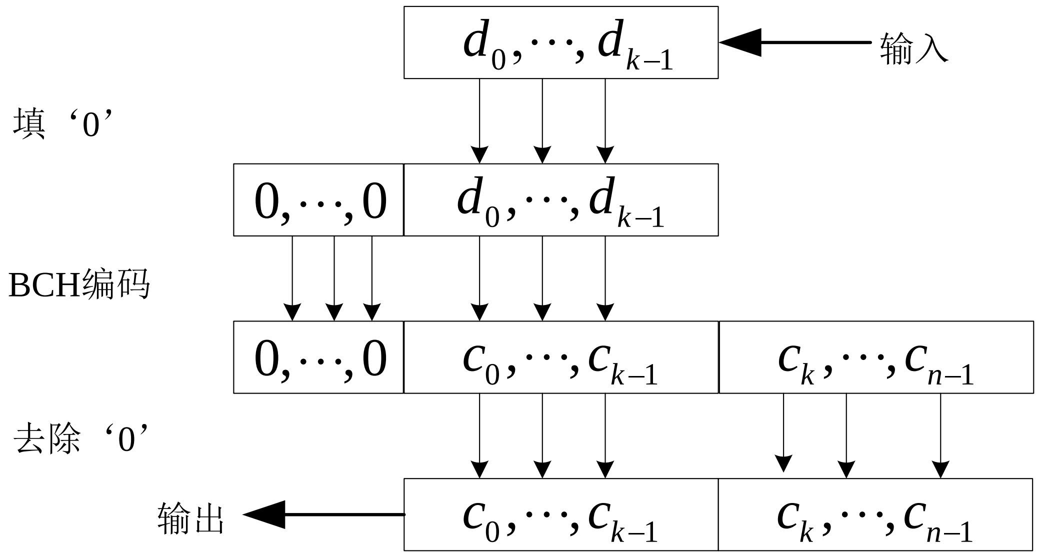
  生成多项式的八进制表示形式为：

  （1561350064670543777423345）$_8$

BCH码应用实例

截短码编码过程

- 数据比特流按照 $d_0, d_1, \cdots, d_{k-1}$ 的顺序进入编码器，编码器在 d_0 前填入 k_s-k个'0',
- 进行编码，得到长度为 n_s比特的原始BCH系统码码字，
- 再去除之前所填的k_s-k 个‘0’，得到截短后的n比特编码序列 $c_0, c_1 \cdots, c_{n-1}$ ，并按此顺序输出.



目录

- 从扩展域的观点看循环汉明码
- BCH码的构造
- BCH码的译码
- BCH码应用实例
- Reed Solumn码

Reed Solumn 码

- Reed Solumn 码是广义BCH码，
  它以GF(2^m)为码元.

  设 $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$

  $v_i \in GF(2^m)$， $i = 0, 1, \ldots, n-1.$

  $n = 2^m - 1$

  v 是RS码的充要条件是：

$$\sum_{i=0}^{n-1} v_i \alpha^{ij} = 0 \qquad j = 1, 2, \cdots, 2t$$

REED SOLUMN 码

- **RS码的主要参数:**

  码元取自$GF(2^m)$

  码长 $n = 2^m - 1$ 个符号

  纠错能力 t 个符号

  校验位 $2t$ 个符号

  信息位 $n-2t$ 个符号

  生成多项式 $g(x)=(x+\alpha)(x+\alpha^2)\ldots(x+\alpha^{2t})$

REED SOLUMN 码

- **特点：**纠正连续突发错误
- **应用：**
  - 卫星通信
  - 数字视频应用
  - 磁盘压缩技术

例：(1) (255, 223)RS码 + 卷积码

美国NASA 八十年代末，深空探测器

(2) (240, k)RS码+LDPC

CMMB标准(Mobile Multimedia broadcast)

作业

- 习题13.1,13.3, 13.4