§ 9.4 循环码的定义和性质

9.4.1 循环码定义与码字的多项式表示

循环移位

$$\boldsymbol{v}=(v_0,v_1,\cdots,v_{n-1}) \longrightarrow \boldsymbol{v}^{(1)}=(v_{n-1},v_0,v_2,\cdots,v_{n-1})$$

定义9.4.1 一个（n, k）线性码C，若它的每个码字矢量的循环移位也是该码的码字，则我们称C为循环码。

[例9.4.1] 一个由4个码字构成的，最小重量为3的（6，2）循环码

$$\mathrm{C}=\{(000000),(010101),(101010),(111111)\}$$

用多项式表示码字矢量

$$\boldsymbol{v}=(v_0,v_1,\cdots,v_{n-1}) \longrightarrow v(x)=v_0+v_1x+v_2x^2+\cdots+v_{n-1}x^{n-1}$$

因为 $$x\cdot v(x)=v^{(1)}(x)+v_{n-1}(x^n+1)$$

$$v^{(1)}(x)\equiv x\cdot v(x) \mod(x^n+1)$$

9.4.2 循环码的性质

定理9.4.1 循环码C 中次数最低的非零码字多项式是唯一的。

[证明]令 $g(x)=g_0+g_1x+\cdots+g_{r-1}x^{r-1}+x^r$是码$C$ 中一个次数最低的非零码字多项式。若不是唯一的，则必然存在另一个次数为r 的码字多项式，$g'(x)=g'_0+g'_1x+\cdots+g'_{r-1}x^{r-1}+x^r\in C$

由于C 是线性的，所以

$$g(x)+g'(x)=(g_0+g'_0)+(g_1+g'_1)x+\cdots+(g_{r-1}+g'_{r-1})x^{r-1}\in C$$

这与假设 $g(x)$是次数最低非零码字多项式相矛盾。

定理9.4.2 令 $g(x)=g_0+g_1x+\cdots+g_{r-1}x^{r-1}+x^r$是（$n,k$）循环码$C$ 中最低次数非零码多项式，则常数项 $g_0=1$。

[证明]若 $g_0=0$ $\longrightarrow$

$$\begin{aligned} g(x)&=g_1x+g_2x+\cdots+g_{r-1}x^{r-1}+x^r \\ &=x(g_1+g_1x+\cdots+g_{r-1}x^{r-2}+x^{r-1}) \end{aligned}$$

所以 $g_1+g_2x+\cdots+g_{r-1}x^{r-2}+x^{r-1}\in C$ 与假设矛盾。

定理9.4.3 令 $g(x)=g_0+g_1x+\cdots+g_{r-1}x^{r-1}+x^r$是（$n,k$）循环码$C$ 中次数最低的非零码字多项式，则任何一个次数不大于n–1的二元多项式，当且仅当它是$g(x)$倍式时，才可成为一个码字多项式。

[证明] 充分性：

令 $v(x)$是次数不大于n–1的二元多项式，且是$g(x)$的倍式，

$$v(x)=(a_0+a_1x+\cdots+a_{n-r-1}x^{n-r-1})\cdot g(x)$$
$$=a_0g(x)+a_1xg(x)+\cdots+a_{n-r-1}x^{n-r-1}g(x)$$

上式中的被加项均是码字多项式，所以$v(x)$是也一个码字多项式；

必要性：

令$v(x)$ 是码C 中一个码字多项式，用$g(x)$除$v(x)$ 得到

$$v(x)=a(x)\cdot g(x)+b(x),\quad b(x)\text{次数小于}g(x)\text{次数；}$$

$$b(x)=v(x)+a(x)g(x),$$

由充分性，$a(x)g(x)$ 是一个码字多项式，故$b(x)$是次数小于r的码字多相式，于是导致矛盾。

总结上面3条定理得:

定理9.4.4 在一个二元（n, k）循环码中，存在唯一的次数为$n-k$的码字多项式$g(x)$，使得每个码字多项式都是$g(x)$的倍式，反之每个次数不大于$n-1$而且为$g(x)$倍式的多项式均对应于一个码字多项式。

所有次数不大于$n-1$，而且是$g(x)$倍式的多项式是由一切形如

$$u(x)=u_0+u_1x+\cdots+u_{n-r-1}x^{n-r-1}$$

多项式与$g(x)$相乘的结果，总共有2^{n-r}个。故2^{n-r}应该等于2^k，即$r=n-k$。

称$g(x)$为这个（n, k）循环码的生成多项式，$u(x)$为消息多项式。

[例9.4.2]由 $g(x)=1+x^2+x^4$ 生成的（6.2）循环码的码字

消息矢量	码字矢量	码字多项式
（u_0, u_1）	（$v_0, v_1, v_2, v_3, v_4, v_5$）	
（0，0）	（0 0 0 0 0 0）	$v_0(x)=0\cdot g(x)=0$
（1，0）	（1 0 1 0 1 0）	$v_1(x)=1\cdot g(x)=g(x)$
（0，1）	（0 1 0 1 0 1）	$v_2(x)=x\cdot g(x)=x+x^3+x^5$
（1，1）	（1 1 1 1 1 1）	$v_4(x)=(1+x)g(x)=1+x+x^2+x^3+x^4+x^5$

生成多项式必须满足一些条件:

定理 9.4.5 （n, k）循环码的生成多项式 $g(x)$ 是x^n+1的因子。

[证明]

$$
\begin{aligned}
x^k g(x) &= x^k + g_1 x^{k+1} + g_2 x^{k+2} + \cdots + g_{n-k-1} x^{n-1} + x^n \\
&= (x^n+1) + 1 + x^k + g_1 x^{k+1} + \cdots + g_{n-k-1} x^{n-1} \\
&= (x^n+1) + b(x)
\end{aligned}
$$

$b(x)$是$g(x)$ 连续向右移位k次后所得多项式，故$b(x)$ 是一个码字多项式 。

即
$$b(x) = u(x) \cdot g(x)$$

故
$$
\begin{aligned}
x^n + 1 &= x^k g(x) + u(x) g(x) \\
&= \{x^k + u(x)\} \cdot g(x)
\end{aligned}
$$

于是 $g(x)$ 是x^n+1的因式。

定理 9.4.6 若$g(x)$是$n-k$次多项式，而且是x^n+1的因式，则$g(x)$生成一个（n, k）循环码。

[证明] 令$g(x)$是x^n+1的一个次数为$n-k$的因式，则

$$v(x)=a_0g(x)+a_1xg(x)+\cdots+a_{k-1}x^{k-1}g(x)$$
$$=\left(a_0+a_1x+\cdots+a_{k-1}x^{k-1}\right)g(x)$$

是一个次数小于或等于（$n-1$）的多项式，且是$g(x)$的倍式。总共有2^k个这样多项式。这些多项式组成一个（n, k）线性分组码。

下面证明这个线性分组码是循环的：

若$v(x)$是该线性码的一个码字多项式，即是$g(x)$的一个倍式，则

$$xv(x)=v_0x+v_1x^2+\cdots+v_{n-1}x^n$$
$$=v_{n-1}\left(x^n+1\right)+v_{n-1}+v_0x+\cdots+v_{k-2}x^{n-1}$$
$$=v_{n-1}\left(x^n+1\right)+v^{(1)}(x)$$

所以$v^{(1)}(x)$也是$g(x)$的倍式。从而$v^{(1)}(x)$也是$g(x), xg(x), \cdots, x^{k-1}g(x)$的线性组合，所以$v^{(1)}(x)$也是一个码字多项式。

[例9.4.3] 多项式 x^7+1 可分解成：$x^7+1=(1+x)(1+x+x^3)(1+x^2+x^3)$

由 $g(x)=1+x+x^3$ 生成的（7, 4）循环码

消息矢量	码字矢量	码字多项式
（0000）	（0000000）	$v_0(x)=0\cdot g(x)$
（1000）	（1101000）	$v_1(x)=1\cdot g(x)$
（0100）	（0110100）	$v_2(x)=x\cdot g(x)=x+x^2+x^4$
（1100）	（1011100）	$v_3(x)=(1+x)g(x)=1+x^2+x^3+x^4$
（0010）	（0011010）	$v_4(x)=x^2g(x)=x^2+x^3+x^5$
（1010）	（1110010）	$v_5(x)=(1+x^2)g(x)=1+x+x^2+x^5$
（0110）	（0101110）	$v_6(x)=(x+x^2)g(x)=1+x^3+x^4+x^5$
（1110）	（1000110）	$v_7(x)=(1+x+x^2)g(x)=x+x^4+x^5$
（0001）	（0001101）	$v_8(x)=x^3g(x)=x^3+x^4+x^6$
（1001）	（1100101）	$v_9(x)=(1+x^3)g(x)=1+x+x^4+x^6$
（0101）	（0111001）	$v_{10}(x)=(x+x^3)g(x)=x+x^2+x^3+x^6$
（1101）	（1010001）	$v_{11}(x)=(1+x+x^3)g(x)=1+x^2+x^6$
（0011）	（0010111）	$v_{12}(x)=(x^2+x^3)g(x)=x^2+x^4+x^5+x^6$
（1011）	（1111111）	$v_{13}(x)=(1+x^2+x^3)g(x)=1+x+x^2+x^3+x^4+x^5+x^6$
（0111）	（0100011）	$v_{14}(x)=(x+x^2+x^3)g(x)=x+x^5+x^6$
（1111）	（1001011）	$v_{15}(x)=(1+x+x^2+x^3)g(x)=1+x^3+x^5+x^6$

§ 9.5 系统循环码的编码及译码

9.5.1 系统循环码的编码

在（n, k）系统循环码中，k 位消息位集中在码字矢量的右侧（最高位）。

构成系统循环码的步骤如下：

1、用x^{n-k} 乘以消息多项式$m(x)=m_0+m_1x+\cdots+m_{k-1}x^{k-1}$；

2、用生成多项式 $g(x)$ 除 $x^{n-k}m(x)$，得到余式$b(x)$（校验位多项式）；

3、构成码字多项式$c(x)=x^{n-k}m(x)+b(x)$ ；

[例9.5.1] 考虑由$g(x)=1+x+x^3$生成的（7.4）循环码，消息多项式是

$m(x)=1+x^2+x^3$，求相应的系统码字多项式。

[解] 1、 $x^{n-k}\cdot m(x)=x^3\cdot m(x)=x^3+x^5+x^6$

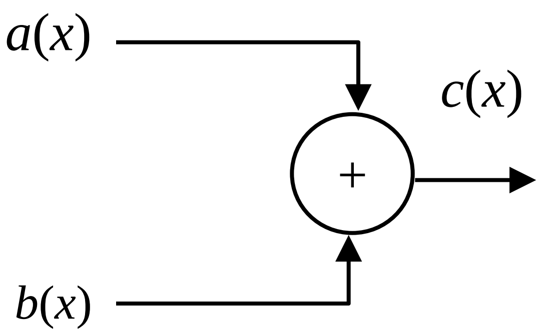
2、 $x^{n-k}\cdot m(x)=g(x)\cdot(1+x+x^2+x^3)+1$

3、 $c(x)=x^3\cdot m(x)+b(x)$

$$=1+x^4+x^5+x^6 \longleftrightarrow \boldsymbol{c}=(1001011)$$

9.5.2 多项式运算的电路实现

一、多项式相加

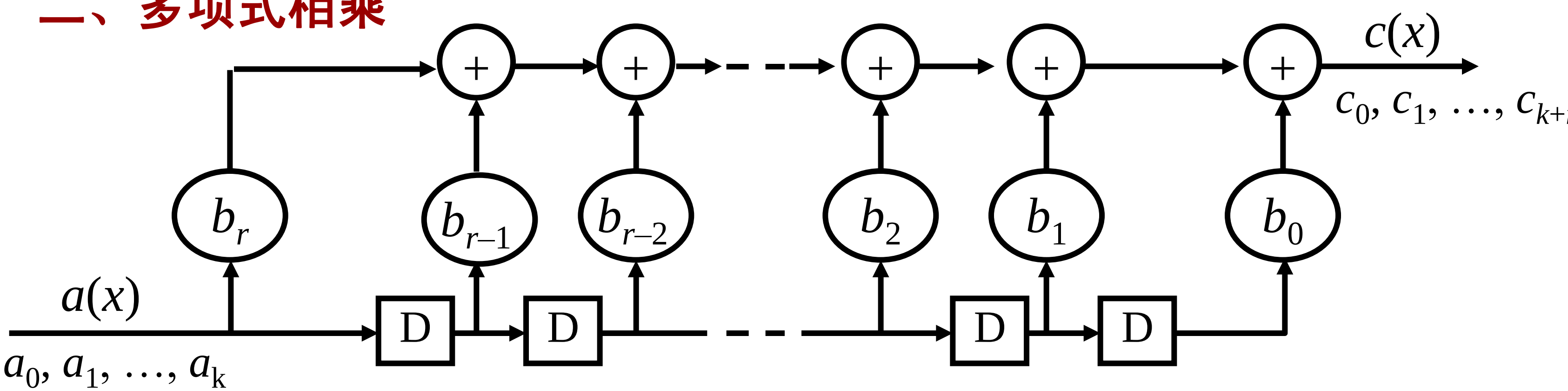


$$c(x) = a(x) + b(x)$$

$$= (a_0 + b_0) + (a_1 + b_1)x + \cdots + (a_{n-1} + b_{n-1})x^{n-1}$$

多项式 $a(x)$, $b(x)$ 的系数依次从高到低位输入到模2加法器，和式的系数从高到低位依次输出。

二、多项式相乘

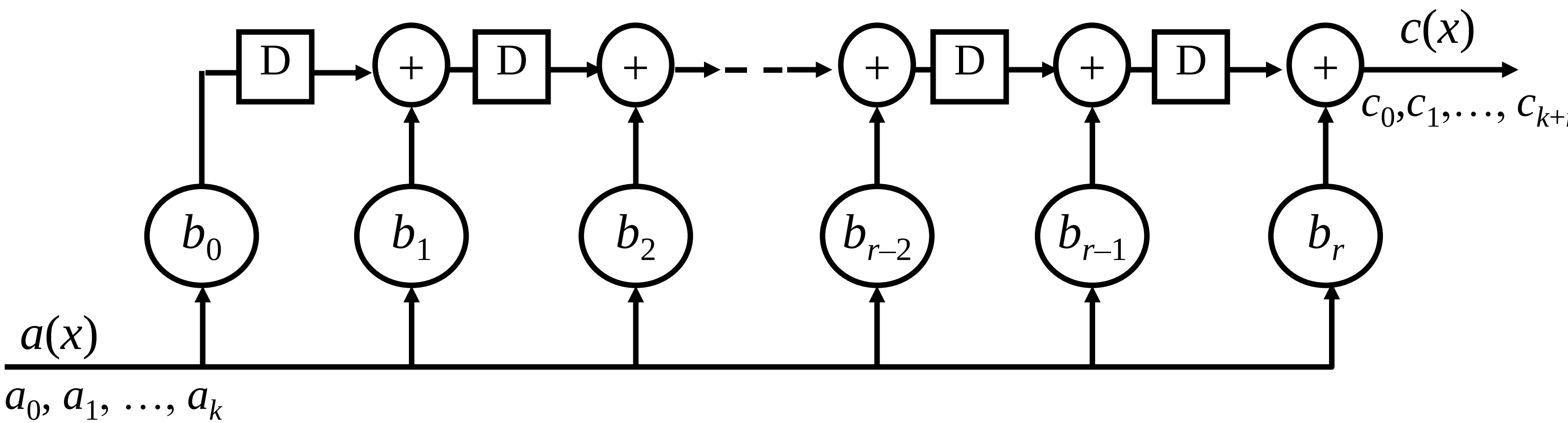


$$a(x) = a_0 + a_1 x + \cdots + a_k x^k \qquad b(x) = b_0 + b_1 x + \cdots + b_r x^r$$

$$c(x) = a(x) \cdot b(x) = a_k b_r x^{k+r} + (a_k b_{r-1} + a_{k-1} b_r) x^{k+r-1} +$$

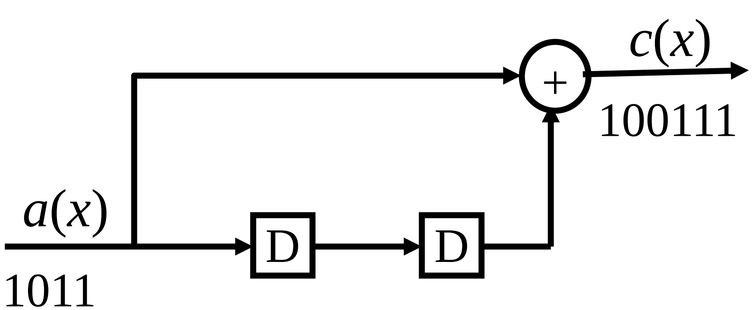
$$(a_k b_{r-2} + a_{k-1} b_{r-1} + a_{k-2} a_r) x^{k+r-2} + \cdots + (a_1 b_0 + a_0 b_1) x + a_0 b_0$$

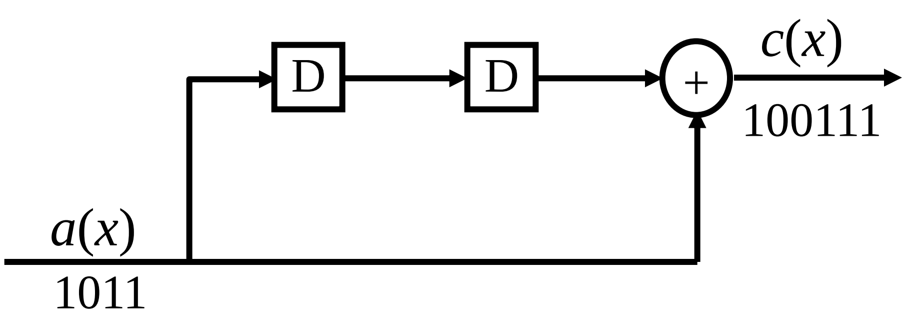
多项式乘法的另一种实现电路



[例9.5.2] $a(x) = 1 + x^2 + x^3 \qquad b(x) = 1 + x^2$

$$c(x) = a(x) \cdot b(x) = 1 + x^3 + x^4 + x^5$$





三、多项式除法电路

被除式

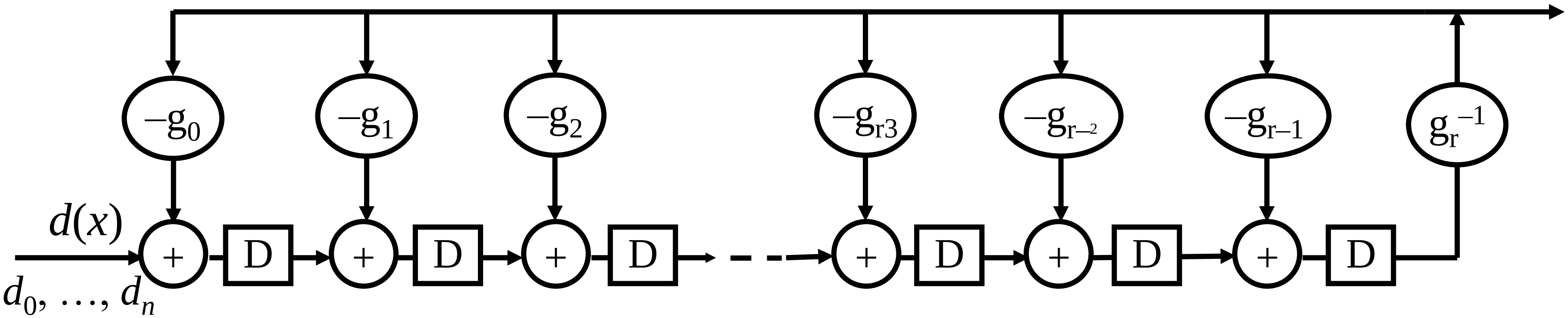
除式

$$d(x)=d_0+\cdots+d_n x^n \qquad g(x)=g_0+g_1x+\cdots+g_r x^r$$

商式

余式

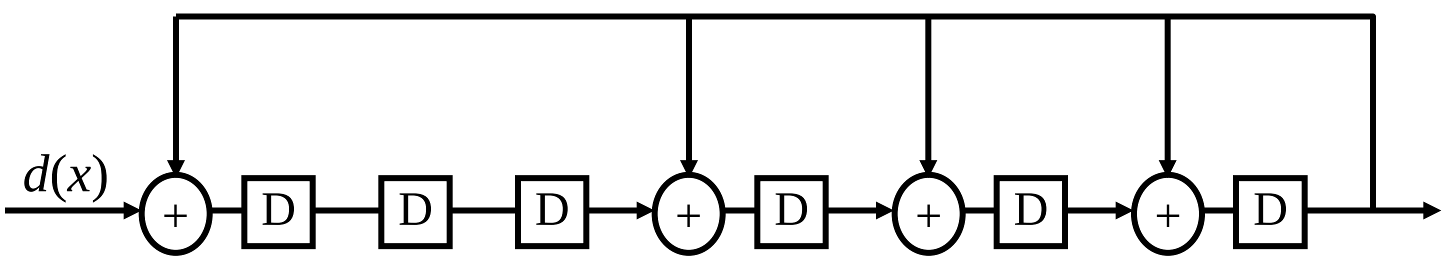
$$d(x)=q(x)g(x)+r(x) \qquad 0\le \deg(r(x))<r$$



在n次移位后，商多项式 $q(x)$出现在输出端，寄存器中保存的是余式$r(x)$。

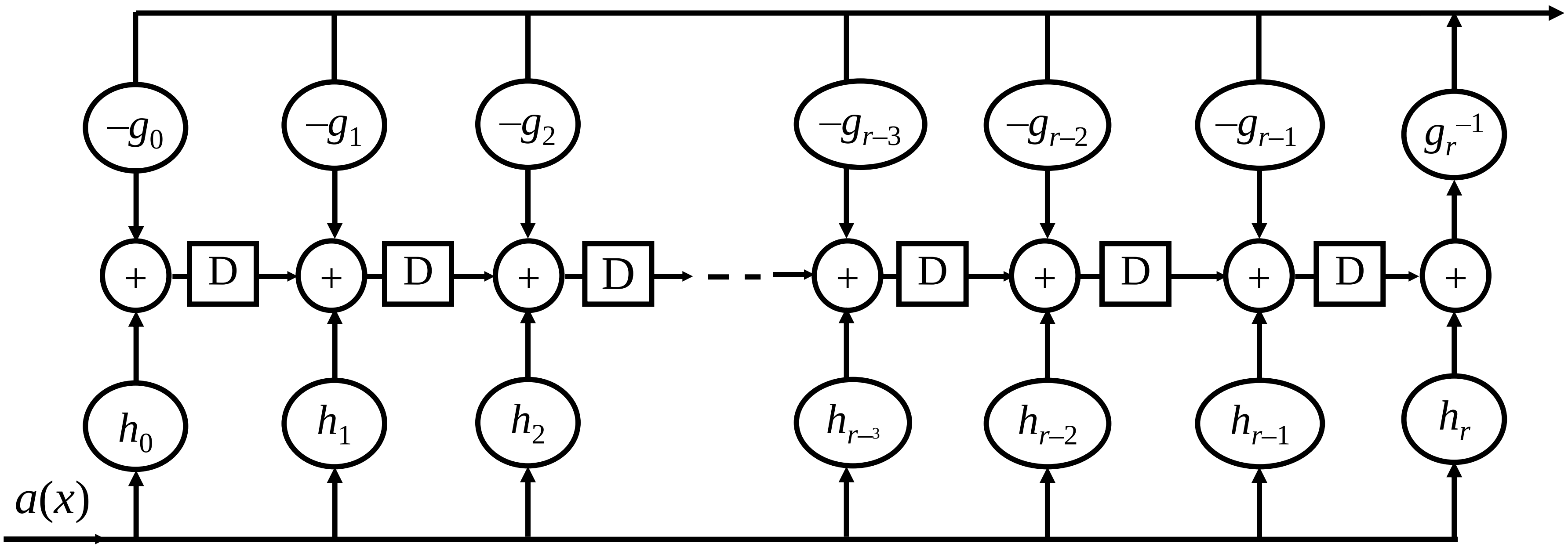
[例9.5.3] $g(x)=x^6+x^5+x^4+x^3+1$

$$d(x)=x^{13}+x^{11}+x^{10}+x^7+x^4+x^3+x+1$$

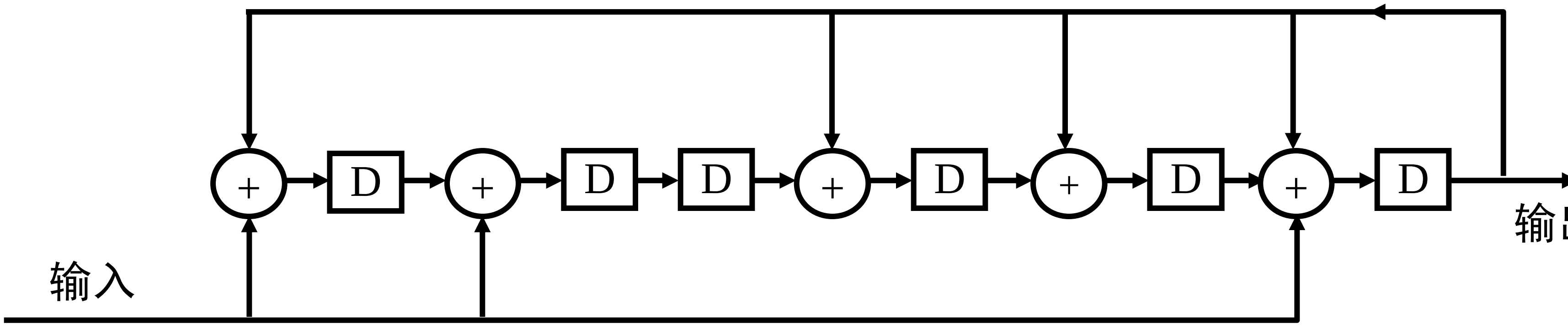


四、乘一个多项式后，再除一个多项式的电路

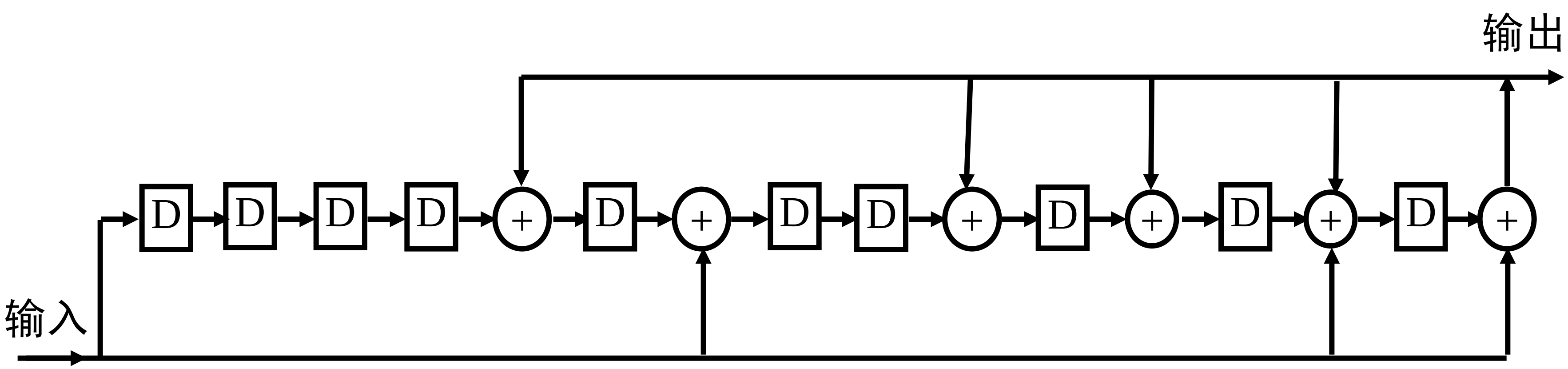
$$\left.\begin{aligned} a(x) &= a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0 \\ h(x) &= h_r x^r + h_{r-1} x^{r-1} + \cdots + h_1 x + h_0 \\ g(x) &= g_r x^r + g_{r-1} x^{r-1} + \cdots + g_1 x + g_0 \end{aligned}\right\} \quad a(x) \cdot h(x) = q(x) \cdot g(x) + r(x)$$



乘以多项式 x^5+x+1，再除以多项式 $x^6+x^5+x^4+x^3+1$ 的电路



乘以多项式 $x^{10}+x^9+x^5+1$，再除以 $x^6+x^5+x^4+x^3+1$ 的电路



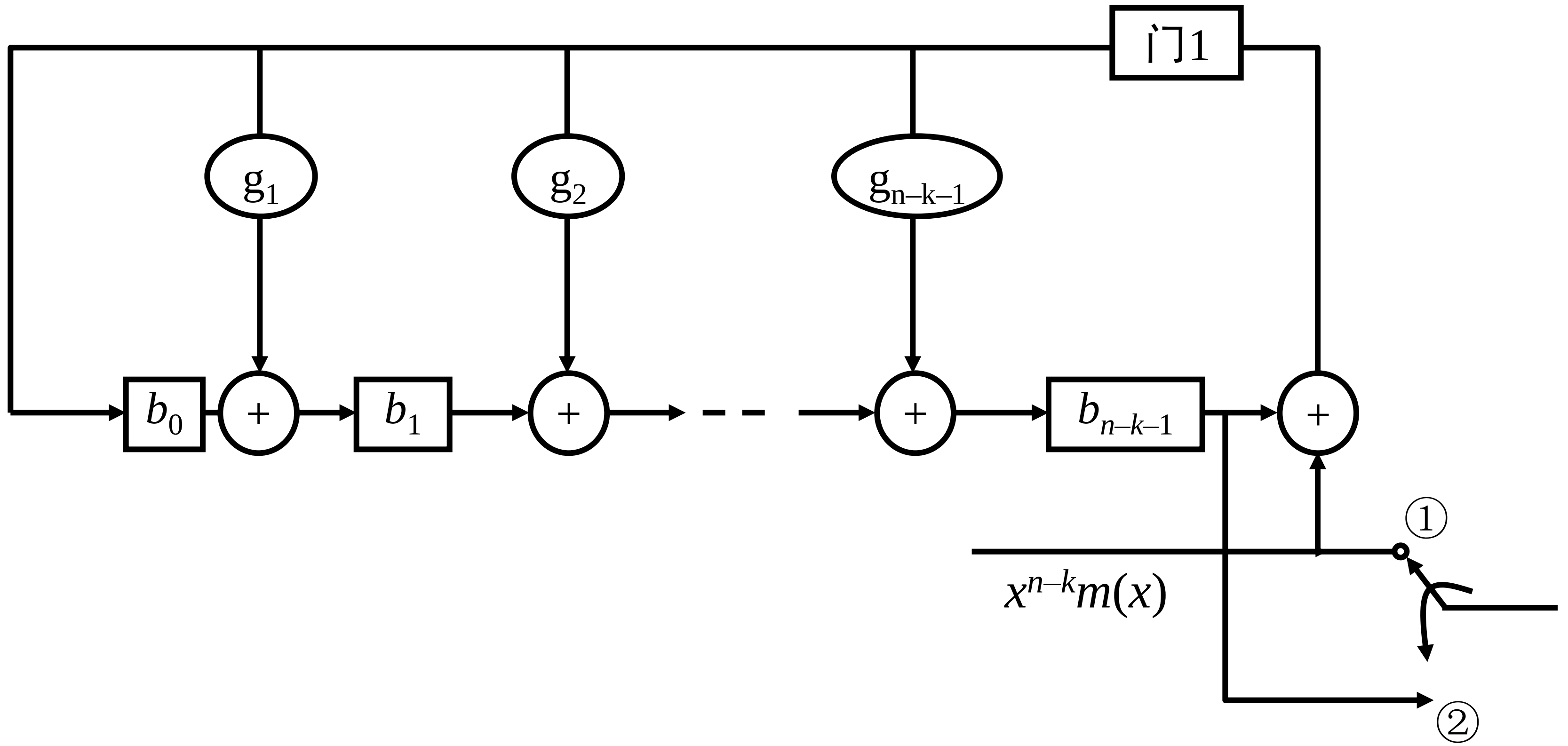
9.5.3 循环码编码的电路实现

一个（n, k）系统循环码的编码过程由三步组成：

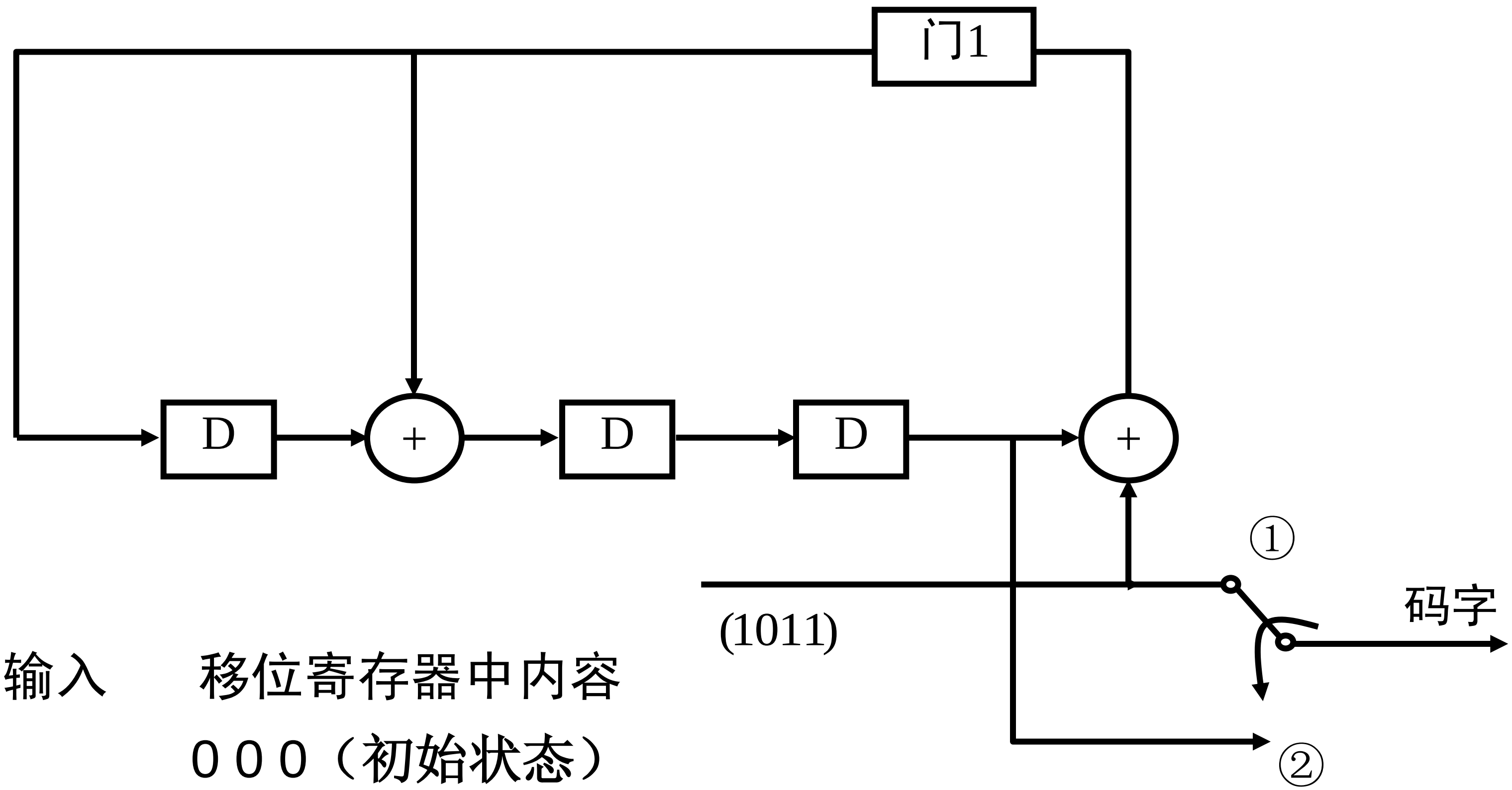
1、用x^{n-k} 乘以消息多项式$m(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$；

2、用生成多项式 $g(x)$ 除 $x^{n-k} m(x)$，得到余式$b(x)$（校验位多项式）；

3、构成码字多项式$c(x) = x^{n-k} m(x) + b(x)$ ；



[例9.5.4] 考虑由$g(x)=1+x+x^3$生成的（7.4）系统循环码。



输入	移位寄存器中内容
	0 0 0（初始状态）
1	1 1 0（第一次移位）
1	1 0 1（第二次移位）
0	1 0 0（第三次移位）
1	1 0 0（第四次移位）

输出完整码字为（1001011）

9.5.4 循环码的译码及其实现

伴随式的计算

由于循环码的循环结构，使得伴随式有如下性质：

定理9.5.1 令 $s(x)$ 是接收多项式 $r(x)=r_0+r_1x+\cdots+r_{n-1}x^{n-1}$ 的伴随式，则用生成多项式 $g(x)$ 除 $xs(x)$ 所得的余式 $s^{(1)}(x)$是 $r(x)$ 向右循环位移一位后 $r^{(1)}(x)$ 的伴随式

[证明] 由于 $$xr(x)=r_{n-1}(x^n+1)+r^{(1)}(x)$$

故 $$r^{(1)}(x)=r_{n-1}(x^n+1)+xr(x)$$

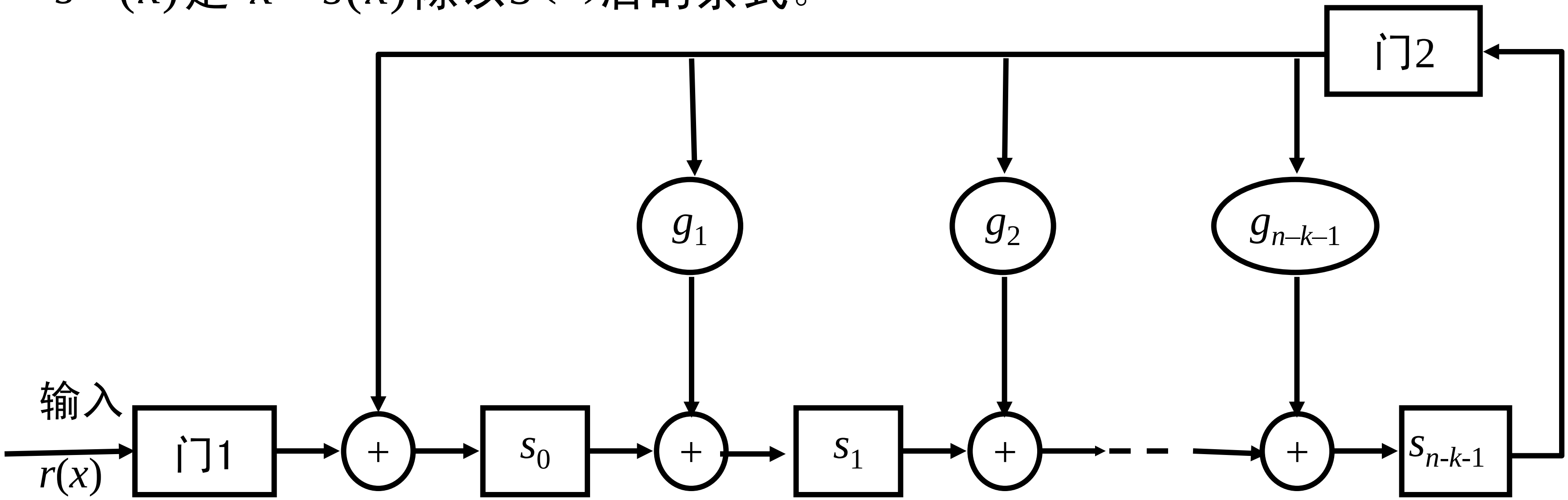
若$r(x)$写成 $$r(x)=q(x)g(x)+s(x)$$

利用 $$x^n+1=h(x)\cdot g(x)$$

得到 $$r^{(1)}(x)=[r_{n-1}h(x)+xq(x)]g(x)+xs(x)$$

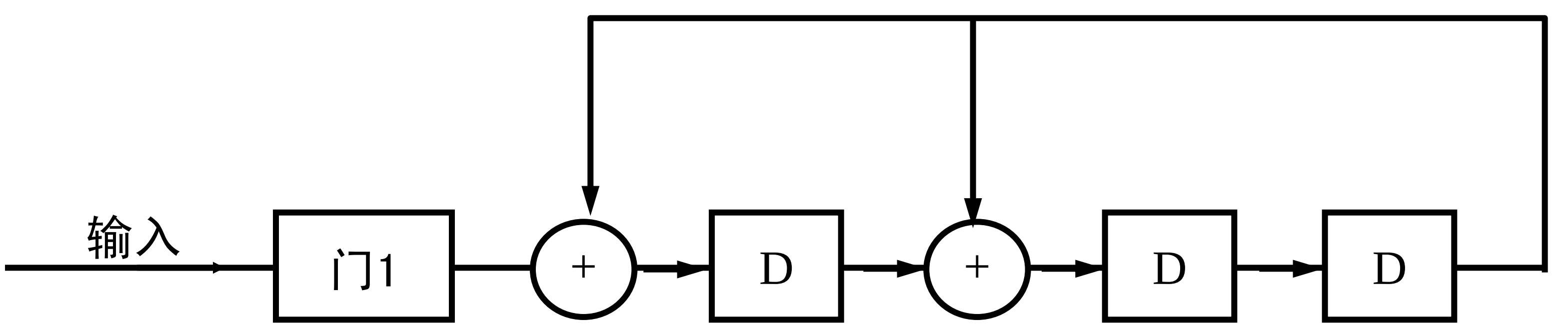
于是 $r^{(1)}(x)$ 的伴随式是$xs(x)$ 除以$g(x)$的余式，我们记之为$s^{(1)}(x)$ 。

类似的，把 $r(x)$ 连续循环移位 i 次，所得的多项式 $r^{(i)}(x)$ 的伴随多项式 $s^{(i)}(x)$ 是 $x^i \cdot s(x)$ 除以 $g(x)$ 后的余式。



计算接收多项式 $r(x)$ 以及 $r^{(i)}(x)$ 的伴随式电路

[例9.5.5] 由 $g(x)=1+x+x^3$ 生成的（7, 4）循环码的伴随式计算电路

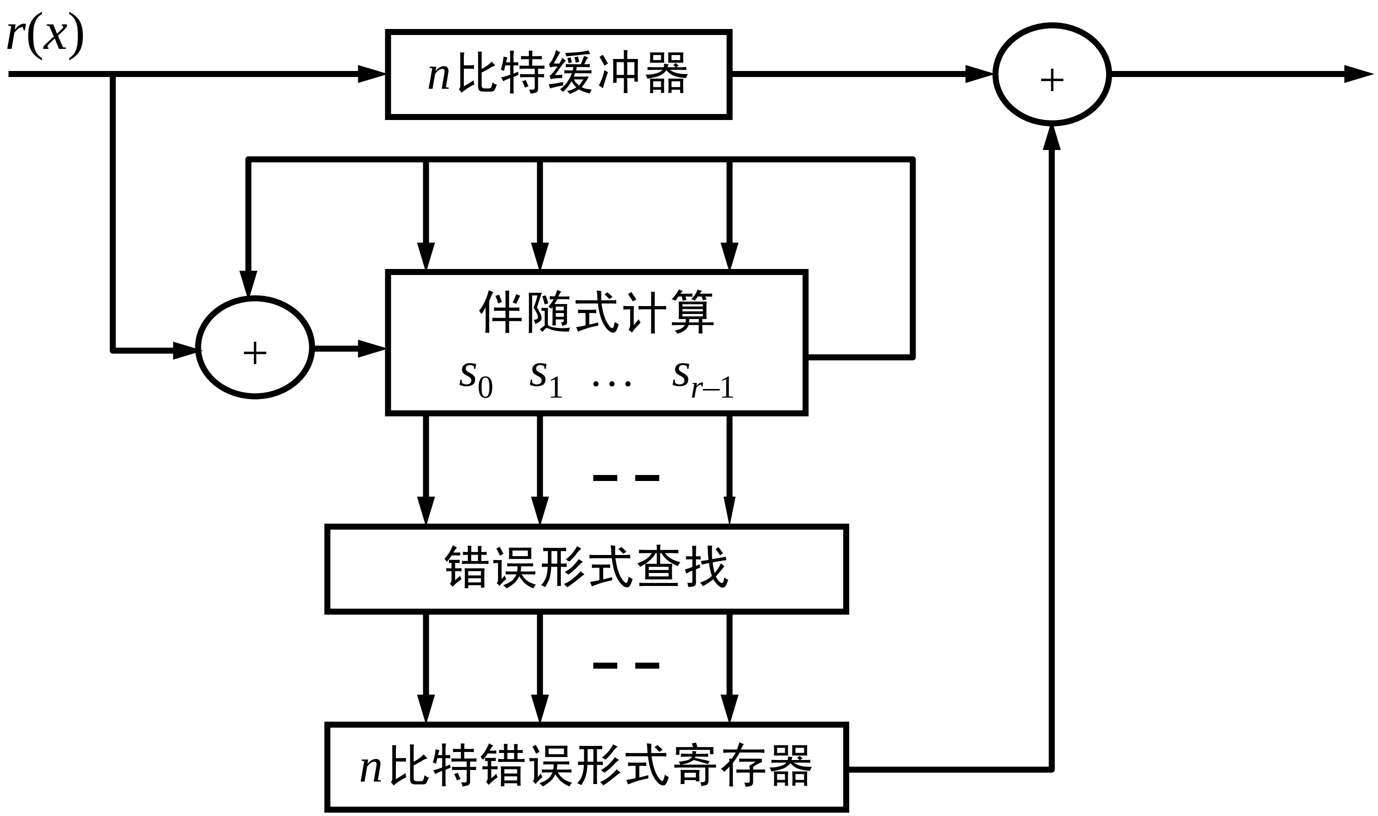


循环码的通用译码算法

1、计算接收多项式 $r(x)$ 对应的伴随式 $s(x)$ ；

2、根据伴随式 $s(x)$，查表寻找对应的错误多项式（陪集首项）；

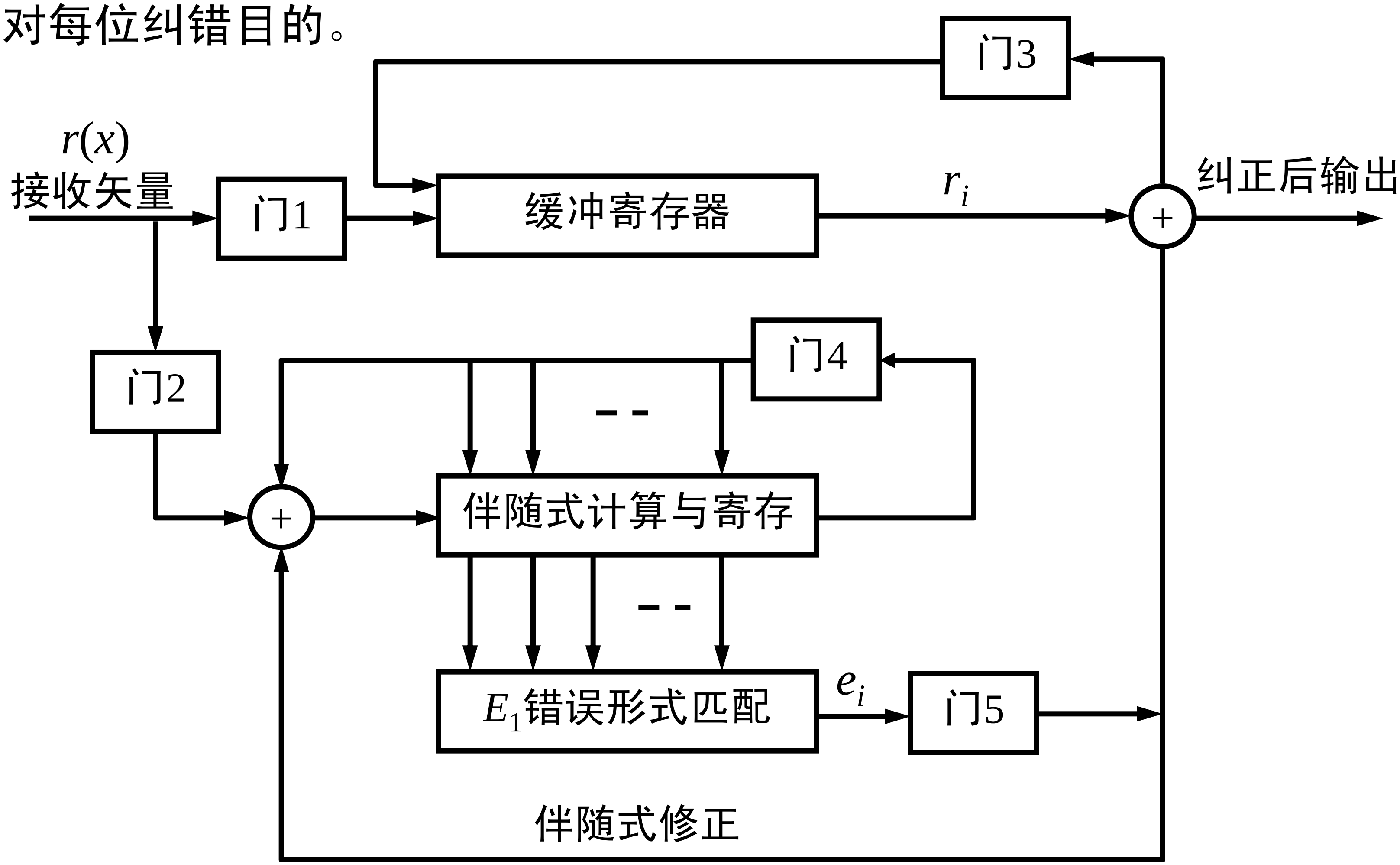
3、把接收多项式和错误多项式相加就纠正了相应的错误；



梅吉特（Meggitt）译码器

错误形式分为二大类：$E_1 = \{e(x) \mid e_{n-1} = 1\}$　　$E_0 = \{e(x) \mid e_{n-1} = 0\}$

通过对接收到矢量$r(x)$逐次循环移位，每次检测、纠正首位错误，达到对每位纠错目的。



1、缓冲寄存器和伴随式寄存器清零，门1，门2，门4接通，门3，门5断开，接收矢量逐位移入到伴随式计算与寄存电路，同时输入到缓冲寄存器。当全部输入后，这时伴随寄存器中寄存的内容为的伴随式。

2、门1、门2断开，门3、门4、门5接通置 i=0，检查伴随式 $s(x)$对应的错误形式是否属E_1，若是则E_1错误形式匹配电路输出“1”，否则输出“0”。

3、置 i=i+1，缓存器输出最高位缓存内容，与E_1错误形式匹配电路输出e_{n-1}相加，纠正该位接收符号的错误。同时把e_{n-1}反馈到伴随式计算与寄存电路的输入，以消除该位错误对于伴随式的影响。缓存器和伴随寄存器同时作一次循环位移，得到新的字$\tilde{r}^{(i)}(x)$和它对应的伴随式 $\tilde{s}^{(i)}(x)$。

4、利用新的伴随式$\tilde{s}^{(i)}(x)$来检查是否与E_1错误形式相匹配，若是则E_1错误匹配电路输出“1”，否则输出“0”。

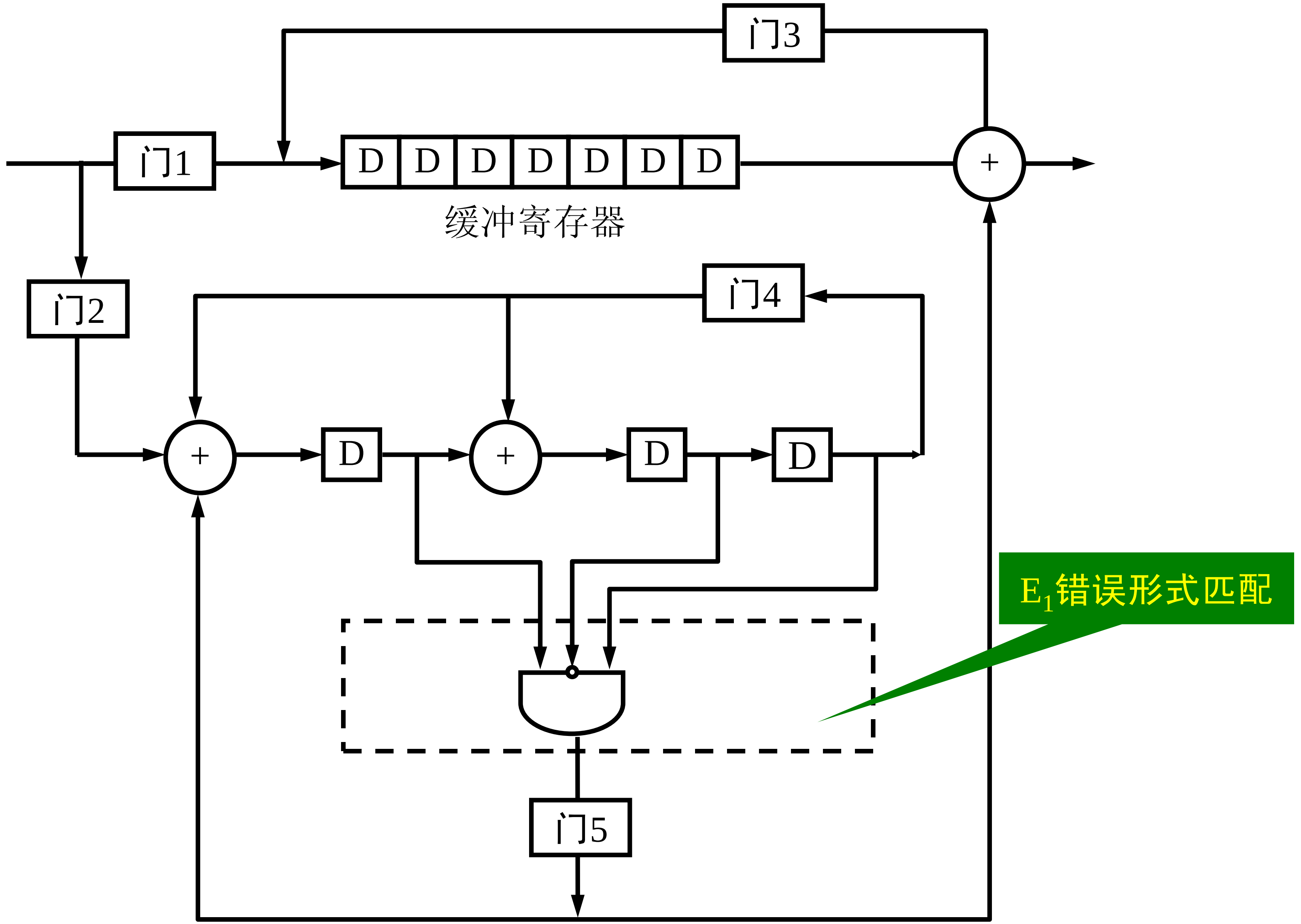
5、若 i=n 则译码结束，不然重复第3步。

如果译码终止后伴随寄存器中内容为全零，则表示成功地纠正了错误，不然表示出现了一个不可纠正的错误。

[例9.5.6] 由 $g(x)=1+x+x^3$ 生成的（7, 4）循环码，这个码的最小Hamming距离是3，可纠正所有7种一位错误。7种一位错误形式和它们对应的伴随式示于下表

错误形式 $e(x)$	伴随式 $s(x)$	伴随式矢量 (s_0, s_1, s_2)
$e_6(x)=x^6$	$s(x)=1+x^2$	1 0 1
$e_5(x)=x^5$	$s(x)=1+x+x^2$	1 1 1
$e_4(x)=x^4$	$s(x)=x+x^2$	0 1 1
$e_3(x)=x^3$	$s(x)=1+x$	1 1 0
$e_2(x)=x^2$	$s(x)=x^2$	0 0 1
$e_1(x)=x$	$s(x)=x$	0 1 0
$e_0(x)=1$	$s(x)=1$	1 0 0

$$E_1=\{e_6(x)\} \qquad E_0=\{e_0(x), e_1(x), e_2(x), e_3(x), e_4(x), e_5(x)\}$$



门3
门1
D D D D D D D
+
缓冲存储器
门2
门4
+
D
+
D
D
E1错误形式匹配
门5

§ 9.6 几个重要的循环码

9.6.1 Hamming循环码

由GF(2)上m次本原多项式生成的长度为 $2^m-1(m\geq 3)$ 的循环码是 $(2^m-1, 2^m-1-m)$ Hamming码。

对所有 $i=0,1,2,\cdots,2^m-2-m$，用生成多项式 $g(x)$除 x^{m+i}，可得

$$x^{m+i}=a_i(x)g(x)+b_i(x)$$

$$b_i(x)=b_{i,0}+b_{i,1}+b_{i,2}x^2+\cdots+b_{i,m-1}x^{m-1}$$

$$c_i(x)=b_{i,0}+b_{i,1}x+\cdots+b_{i,m-1}x^{m-1}+x^{m+i}$$

这(2^m-1-m)码字线性独立，故这些码字构成生成矩阵。

$$G=\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & \cdots & b_{0,m-1} & 1 & 0 & \cdots & \cdots & 0 \\ b_{1,0} & b_{1,1} & b_{1,2} & \cdots & b_{1,m-1} & 0 & 1 & 0 & \cdots & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \cdots & b_{2,m-1} & 0 & 0 & 1 & \cdots & 0 \\ & \cdots & & \cdots & & & & & \cdots & \\ b_{2^m-2-m,0} & b_{2^m-1-m,1} & b_{2^m-2-m,2} & \cdots & b_{2^m-1-m,m-1} & 0 & 0 & & \cdots & 1 \end{bmatrix}$$

$$H=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & b_{0,0} & b_{1,0} & \cdots & b_{2^m-1-m,0} \\ 0 & 1 & 0 & \cdots & 0 & b_{0,1} & b_{1,1} & \cdots & b_{2^m-2-m,1} \\ & & & \cdots & & & & \cdots & \\ 0 & 0 & 0 & \cdots & 1 & b_{0,m-1} & b_{1,m-1} & \cdots & b_{2^m-2-m,m-1} \end{bmatrix}$$

可以证明H中无全零列矢量，无二列矢量相同，故可纠正全部一位错误。

生成多项式的表示：常用八进制数字表示生成多项式$g(x)$；

八进制 13 ➡ 二进制 0 0 1 0 1 1 ➡ $g(x)=x^3+x+1$

八进制 23 ➡ 二进制 0 1 0 0 1 1 ➡ $g(x)=x^4+x+1$

八进制 211 ➡ 二进制 0 1 0 0 0 1 0 0 1 ➡ $g(x)=x^7+x^3+1$

9.6.2 BCH码

对于任何正整数 m 和 t（$t < 2^{m-1}$），存在具有如下参数的BCH码：

1、码长 $n = 2^m - 1$

2、校验位数目 $n - k \le mt$

3、最小距离 $d \ge 2t + 1$

BCH码的生成多项式的构成：

α 是$GF(2m)$的本原元，考虑 α 的如下幂序列：

$$\alpha,\ \alpha^2, \alpha^3, \alpha^4, \cdots, \alpha^{2t}$$

令 $m_i(t)$ 是 α^i 的最小多项式，则满足所列参数要求的BCH码生成多项式，

$$g(x) = LCM[m_1(x), m_2(x), m_3(x), \cdots, m_{2t}(x)]$$

利用共轭元具有相同最小多项式的特点，则生成多项式可以写成，

$$g(x) = LCM[m_1(x), m_3(x), \cdots, m_{2t-1}(x)]$$

[例9.6.1] 在$GF[2^4]$上构造长度为$2^4-1=15$，分别能纠一位和两位错误的BCH码。

长度为15的能纠一位错误的BCH码的生成多项式以α^1和α^2为根，故

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

$g(x)$的八进制表示为“23”；生成(15，11)Hamming码。

长度为15，能纠正二位错误的BCH码的生成多项式以$\alpha^1,\alpha^2,\alpha^3,\alpha^4$为根，

$$g(x)=(x+\alpha^1)(x+\alpha^2)(x+\alpha^4)(x+\alpha^8)(x+\alpha^3)(x+\alpha^6)(x+\alpha^{12})(x+\alpha^9)$$

$$=(x^4+x+1)(x^4+x^3+x^2+x+1)=x^8+x^7+x^6+x^4+1$$

$g(x)$的八进制表示为“723”；生成(15，7)BCH码，能纠正任意二位错误。

9.6.3 Reed-Solomon (R-S)码

RS码是一类非二进制的BCH码，具有很强的纠错能力。RS码的码元符号取自有限域$GF(q)$，它的生成多项式的根也是$GF(q)$中的本原元，所以它的符号域和根域相同。能纠正 t 个错误的R-S码具有如下参数：

码长： $n = q - 1$

校验位数目： $n - k = 2t$

最小距离： $d = 2t + 1$

R-S码的最小距离为校验位数目加1，达到了Singleton限界。

当 $q = 2^m$，RS码的码元符号取自 $GF(2^m)$ ，码字长度为 $n = 2^m - 1$。一个能纠正 t 位符号错误的RS码的生成多项式是

$$g(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^3)\cdots(x+\alpha^{2t})$$

其中α 为$GF(2^m)$ 的本原元。

[注意] $GF(2^m)$ 中元素可用长度为m的二元矢量表示，长度为 $2^m - 1$ 的码字用二进制符号表示长度为$m(2^m - 1)$，能纠正 $m \cdot t$ 个二进制符号错误。

[例9.6.3] 一个符号取自GF（2^3），长度为7，能纠正2个八进制错误的RS码的生成多项式为：

$$g(x)=(x+\alpha)(x+\alpha^2)(x+\alpha^3)(x+\alpha^4)=\alpha^3+\alpha x+x^2+\alpha^3x^3+x^4$$

其中α为本原多项式 x^3+x+1 的根。信息位长度为3，监督位长度为4。

对于消息多项式 $m_i(x)=x^i$，$i=0,1,2$，系统码字为

$$c_0(x)=\alpha^3+\alpha x+x^2+\alpha^3x^3+x^4$$

$$c_1(x)=\alpha^6+\alpha^6x+x^2+\alpha^2x^3+x^5$$

$$c_2(x)=\alpha^5+\alpha^4x+x^2+\alpha^4x^3+x^6$$

生成矩阵：

$$G=\begin{pmatrix}\alpha^3 & \alpha & 1 & \alpha^3 & 1 & 0 & 0\\ \alpha^6 & \alpha^6 & 1 & \alpha^2 & 0 & 1 & 0\\ \alpha^5 & \alpha^4 & 1 & \alpha^4 & 0 & 0 & 1\end{pmatrix}$$

校验矩阵：

$$H=\begin{pmatrix}1 & 0 & 0 & 0 & \alpha^3 & \alpha^6 & \alpha^5\\ 0 & 1 & 0 & 0 & \alpha & \alpha^6 & \alpha^4\\ 0 & 0 & 1 & 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & \alpha^3 & \alpha^2 & \alpha^4\end{pmatrix}$$

用二元矢量来表示$GF(2^3)$的元素，则（7,3）RS码字长度为21比特，信息位长度为9比特。例如：

$$\boldsymbol{m}=(\alpha^6,\alpha^5,\alpha^2) \quad \Longleftrightarrow \quad \boldsymbol{m}=(101,111,001)$$

$$\boldsymbol{c}=(\alpha^6,\alpha^5,\alpha^2)\cdot G=(\alpha^{10},\alpha^8,\alpha^4,0,\alpha^6,\alpha^5,\alpha^2)$$

$$=(\alpha^3,\alpha^2,\alpha^4,0,\alpha^6,\alpha^5,\alpha^2)$$

$$\boldsymbol{c}=(110,001,011,000,101,111,001)$$

对于BCH码和RS码的译码，已经有一些很有效的代数算法，纠错时要确定错误位置，以及求出相应的错误值。对于二进制BCH码只要确定错误位置就行。