cyclic codes

- subclass of linear block codes with simple shift register implementation
- n-bit code vector $c = (c_0 \; c_1 \; \ldots \; c_{n-1})$

  cyclic shift by 1 position

$$c^{(1)} = (c_{n-1} \; c_0 \; \ldots \; c_{n-2})$$
$$c^{(2)} = (c_{n-2} \; c_{n-1} \; c_0 \; \ldots \; c_{n-3})$$
$$\vdots$$
$$c^{(i)} = (c_{n-i}, \; \ldots \; c_{n-i-1})$$

- All cyclic shifts are codewords in a cyclic code.
- $c(X) \triangleq c_0 + c_1 X + c_2 X^2 + \ldots + c_{n-1} X^{n-1}$

  code polynomial $X \rightarrow$ real variable

  sum of 2 polynomials $\Rightarrow \oplus$ addition of the corresponding (binary (0,1)) coefficients
- $X\,c(X) = c_0 X + c_1 X^2 + \ldots + c_{n-1} X^n$

$$c^{(1)}(X) = c_{n-1} + c_0 X + \ldots + c_{n-2} X^{n-1}$$

sum the two polynomials (note : $(c_0 \oplus c_0) X = 0$)

$$X\,c(X) + c^{(1)}(X) = c_{n-1} + c_{n-1} X^n$$

$$\Rightarrow c^{(1)}(X) = X\,c(X) + c_{n-1}(X^n + 1)$$

similarly

$$\underline{c^{(i)}(X)} = X^i c(X) + \underbrace{q(X)}_{(c_{n-i} + \ldots + c_{n-1} X^{i-1})} (X^n + 1)$$

$$c^{(i)}(X) = X^i c(X) \bmod (X^n + 1)$$

cyclic codes should satisfy above constraint

- $c^{(i)}(X)$ is the remainder and
  $q(X)$ is the quotient after division by (X^n+1)
- Code generation $\Rightarrow$ generator polynomial

  $g(X)$ of a cylic code (n,k), $r=n-k$ is a factor of X^n+1

$$\boxed{g(X) = 1 + g_1 X + \cdots + g_{r-1} X^{r-1} + X^r}$$

- $c_m(X) = a_m(X)\, g(X) \qquad m = 1, 2, \ldots, 2^k$

$$c^{(1)}(X) = X\, c(X) + c_{n-1}\,(X^n+1)$$

Since $g(X)$ divides X^n+1 and $c(X)$, it also divides $c^{(1)}(X)$, i.e. $c^{(1)}(X)$ can be represented by $c^{(1)}(X) = a_1(X)\, g(X)$

$\therefore$ For a systematic cyclic code

$$(\underbrace{b_0\ b_1\ \ldots\ b_{r-1}}_{r=n-k \text{ parity bits}}\ \underbrace{m_0\ m_1\ \cdots\ m_{k-1}}_{k \text{ message bits}})$$

$$m(X) = m_0 + m_1 X + \cdots + m_{k-1} X^{k-1}$$

$$b(X) = b_0 + b_1 X + \cdots + b_{r-1} X^{r-1}$$

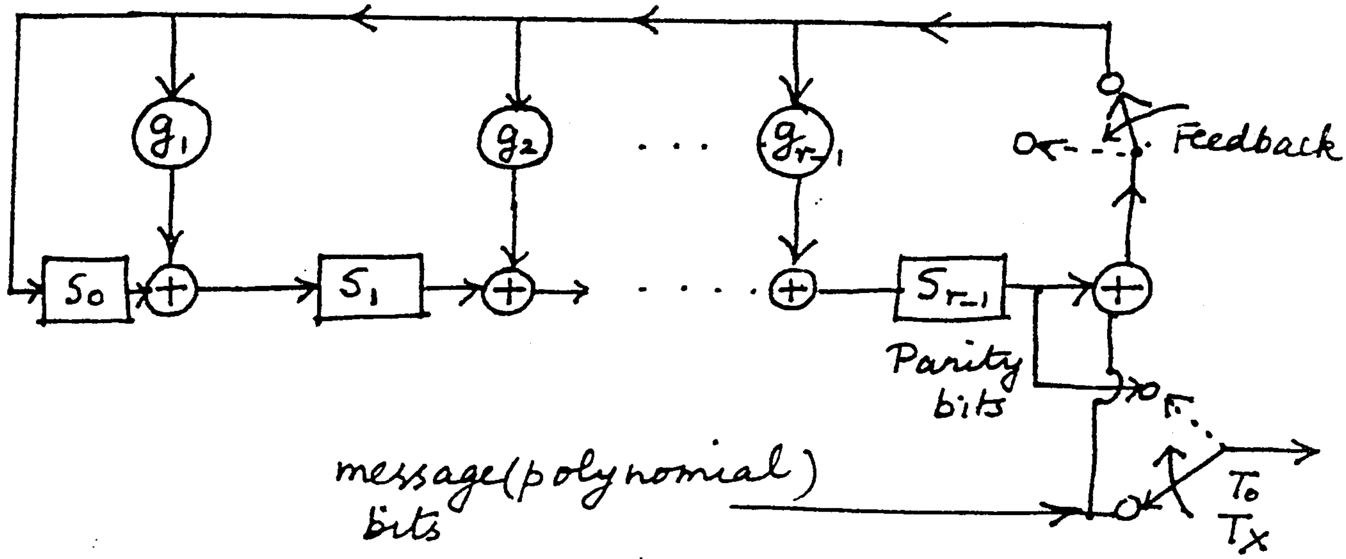
Want $\quad c(X) = b(X) + X^{n-k} m(X)$

$$= a_m(X)\, g(X)$$

$$\Rightarrow \quad \frac{X^r m(X)}{g(X)} = a_m(x) + \frac{b(X)}{g(X)}$$

$\Rightarrow$ $b(X)$ equals the remainder left over after dividing $X^r m(X)$ by $g(X)$

· Shift register encoder



i) Feedback switch closed, output switch in message-bit position, register in all-zero state

ii) k- message bits shifted into register and simultaneously delivered to transmitter

iii) After k shift cycles, register contains r parity bits

iv) Feedback switch is now opened output switch moved to deliver the check bits

• syndrome $s(X) = \text{rem} \dfrac{r(X)}{g(X)}$

where $\bar{r}(X)$ is a received word

If $r(X)$ is a valid code, rem $= 0$

Ex: (7, 4) cyclic code $r = 7-4 = 3$

$$g(x) = 1 + x + x^3 = 1 + x + 0 + x^3$$

$$m(x) = 0 + 0 + x^2 + x^3 \qquad \begin{pmatrix} 0 & 0 & 1 & 1 \\ m_0 & m_1 & m_2 & m_3 \end{pmatrix}$$

$$x^r m(x) = x^3 m(x) = 0 + 0 + 0 + 0 + 0 + x^5 + x^6$$

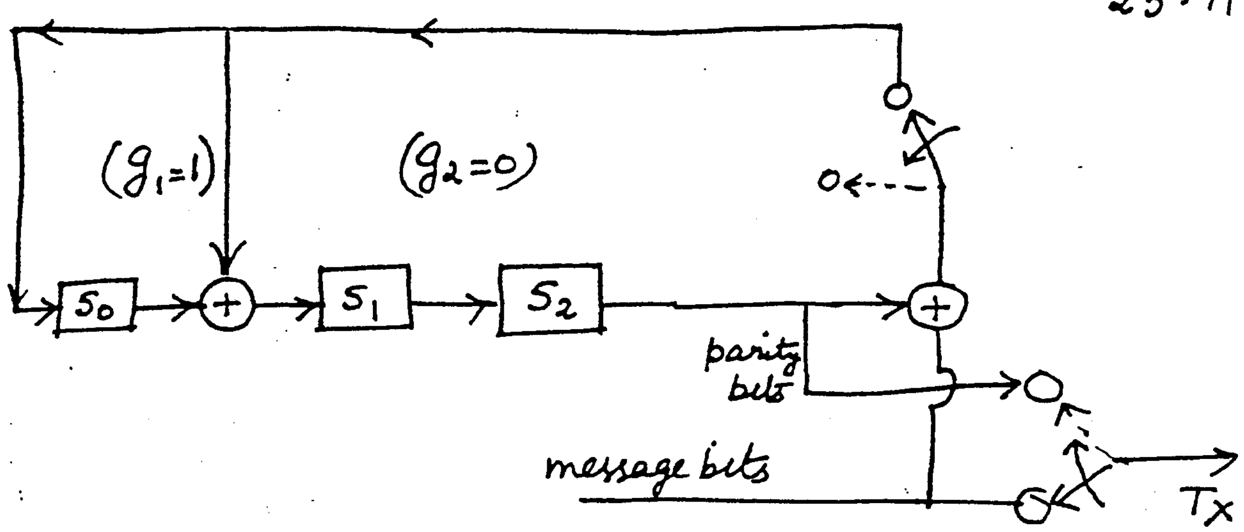
$$\begin{array}{r|l}
 & x^3 + x^2 + x + 0 \\
x^3 + 0 + x + 1 & x^6 + x^5 + 0 + 0 + 0 + 0 + 0 \\
 & \underline{x^6 + 0 + x^4 + x^3} \\
 & \quad x^5 + x^4 + x^3 + 0 \\
 & \quad \underline{x^5 + 0 + x^3 + x^2} \\
 & \qquad x^4 + 0 + x^2 + 0 \\
 & \qquad \underline{x^4 + 0 + x^2 + x} \\
 & \qquad\quad 0 + 0 + x + 0 \\
 & \qquad\quad \underline{0 + 0 + 0 + 0} \\
 & b(x) = \underline{\qquad 0 + x + 0}
\end{array}$$

$$c(x) = b(x) + x^r m(x)$$

$$= 0 + x + 0 + 0 + 0 + x^5 + x^6$$

$$= x + x^5 + x^6$$

$$= (\underbrace{0\ 1\ 0}_{\text{parity bits}} \ ; \ \underbrace{0\ 0\ 1\ 1}_{\text{message bits}})$$



a)

Input bit	Register bits before shift			Register bits after shift		
m	S_2	S_1	S_0	$S_2' = S_1$	$S_1' = S_0 \oplus S_2 \oplus m$	$S_0' = S_2 \oplus m$
1	0	0	0	0	1	1
1	0	1	1	1	0	1
0	1	0	1	0	0	1
0	0	0	1	0	1	0

a) shift register encoder for (7,4) code

b) For $m = (0\ 0\ 1\ 1)$

After 4 cycles, register holds $b = (0\ 1\ 0)$ in agreement with manual division

Reed-Solomon (RS) code

- nonbinary ⇒ encodes m-bit symbols
  e.g. $m = 8 \Rightarrow$ byte oriented
- (n, k) RS code : $r = n - k$
  k information symbols r parity symbols

no. of symbols $n = 2^m - 1$
can correct 't' RS symbols

$$t = r/2$$

- e.g. $m = 8$ $n = 2^8 - 1 = 255$
  If $t = 16$ $r = 2t = 32$ $k = n - r$
  $= 223$

code rate $R_c = \frac{k}{n} = \frac{223}{255} \approx 7/8$

Total no. of bits in codeword $= 255 \times 8$
$= 2040$ bits

- can correct a burst of $16 \times 8 = 128$
  consecutive bit errors

Ex: compact disc (CD) digital audio
sampling rate $= 44.1$ kHz
16 bits / sample ⇒ 700 kb/s
⇒ upto 70 minutes of material can be stored on a single disc (10^{10} bits)
uses cross-interleave RS codes
- Two RS codes $(28, 24)$ $(32, 28)$
  inner   outer