# Hamming Code

- H(n,k): k information bit length, n overall code length
- n=2^m-1, k=2^m-m-1:
- H(7,4), rate (4/7);  H(15,11), rate (11/15); H(31,26), rate (26/31)
- H(7,4): Distance d=3, correction ability 1, detection ability 2.
- Remember that it is good to have larger distance and rate.
- Larger n means larger delay, but usually better code

# Hamming Code Example

- H(7,4)
- Generator matrix G: first 4-by-4 identical matrix

$$\mathbf{G} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

- Message information vector p

$$\mathbf{p} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

- Transmission vector x
- Received vector r

and error vector e $\mathbf{r} = \mathbf{x} + \mathbf{e}_i$

$$\mathbf{Gp} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \mathbf{x}$$

- Parity check matrix H

$$\mathbf{H} := \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

# Error Correction

- If there is no error, syndrome vector z=zeros

$$\mathbf{Hr} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{z}$$

- If there is one error at location 2

$$\mathbf{Hr} = \mathbf{H}(\mathbf{x} + \mathbf{e}_i) = \mathbf{Hx} + \mathbf{He}_i = \mathbf{0} + \mathbf{He}_i = \mathbf{He}_i$$

$$\mathbf{r} = \mathbf{x} + \mathbf{e}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

- New syndrome vector z is

$$\mathbf{Hr} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \mathbf{z}$$

which corresponds to the second column of **H**. Thus, an error has been detected in position 2, and can be corrected

# Important Hamming Codes

- Hamming (7,4,3) -code. It has 16 codewords of length 7. It can be used to send $2^7 = 128$ messages and can be used to correct 1 error.
- Golay (23,12,7) -code. It has 4 096 codewords. It can be used to transmit 8 3888 608 messages and can correct 3 errors.
- Quadratic residue (47,24,11) -code. It has 16 777 216 codewords and can be used to transmit 140 737 488 355 238 messages and correct 5 errors.

# Reed–Muller code

Reed-Muller codes form a family of codes defined recursively with interesting properties and easy decoding.

If $D_1$ is a binary $[n,k_1,d_1]$ -code and $D_2$ is a binary $[n,k_2,d_2]$ -code, a binary code $C$ of length $2n$ is defined as follows $C = \{ \mid u \mid u + v \mid,$ where $u \in D_1$, $v \in D_2\}$.

Lemma $C$ is $[2n,k_1 + k_2, \min\{2d_1,d_2\}]$ -code and if $G_i$ is a generator matrix for $D_i$, $i = 1, 2$, then $\begin{pmatrix} G_1 & G_1 \\ 0 & G_2 \end{pmatrix}$ is a generator matrix for $C$.

Reed-Muller codes $R(r,m)$, with $0 \le r \le m$ are binary codes of length $n = 2^m$. $R(m,m)$ is the whole set of words of length $n$, $R(0,m)$ is the repetition code.

If $0 < r < m$, then $R(r + 1,m + 1)$ is obtained from codes $R(r + 1,m)$ and $R(r,m)$ by the above construction.

# Cyclic code

- Cyclic codes are of interest and importance because
  - They posses rich algebraic structure that can be utilized in a variety of ways.
  - They have extremely concise specifications.
  - They can be efficiently implemented using simple *shift register*
  - Many practically important codes are cyclic
- In practice, cyclic codes are often used for error detection (Cyclic redundancy check, CRC)
  - Used for packet networks
  - When an error is detected by the receiver, it requests retransmission
  - ARQ

# BASIC DEFINITION of Cyclic Code

Definition A code $C$ is cyclic if

(i) $C$ is a linear code;

(ii) any cyclic shift of a codeword is also a codeword, i.e. whenever $a_0, \ldots a_{n-1} \in C$, then also $a_{n-1}\, a_0 \ldots a_{n-2} \in C$.

Example

(i) Code $C$ = {000, 101, 011, 110} is cyclic.

(ii)

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

# *FREQUENCY of CYCLIC CODES*

Comparing with linear codes, the cyclic codes are quite scarce. For, example there are 11 811 linear (7,3) linear binary codes, but only two of them are cyclic.

Trivial cyclic codes. For any field $F$ and any integer $n >= 3$ there are always the following cyclic codes of length $n$ over $F$:

- No-information code - code consisting of just one all-zero codeword.
- Repetition code - code consisting of code-words $(a, a, \ldots, a)$ for $a \in F$.
- Single-parity-check code - code consisting of all code-words with parity 0.
- No-parity code - code consisting of all code-words of length $n$

For some cases, for example for $n = 19$ and $F = GF(2)$, the above four trivial cyclic codes are the only cyclic codes.

# *EXAMPLE of a CYCLIC CODE*

The code with the generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

has code-words

$$c_1 = 1011100 \qquad c_2 = 0101110 \qquad c_3 = 0010111$$

$$c_1 + c_2 = 1110010 \qquad c_1 + c_3 = 1001011 \qquad c_2 + c_3 = 0111001$$

$$c_1 + c_2 + c_3 = 1100101$$

and it is cyclic because the right shifts have the following impacts

$$c_1 \to c_2, \qquad c_2 \to c_3, \qquad c_3 \to c_1 + c_3$$

$$c_1 + c_2 \to c_2 + c_3, \qquad c_1 + c_3 \to c_1 + c_2 + c_3, \qquad c_2 + c_3 \to c_1$$

$$c_1 + c_2 + c_3 \to c_1 + c_2$$

# *POLYNOMIALS over GF(q)*

A codeword of a cyclic code is usually denoted

$$a_0 a_1 \ldots a_{n-1}$$

and to each such a codeword the polynomial

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

is associated.

$F_q[x]$ denotes the set of all polynomials over $GF(q)$.

*deg* (f($x$)) = the largest $m$ such that $x^m$ has a non-zero coefficient in $f(x)$.

Multiplication of polynomials If f($x$), g($x$) $\in F_q[x]$, then

$$deg\,(f(x)\,g(x)) = deg\,(f(x)) + deg\,(g(x)).$$

Division of polynomials For every pair of polynomials a($x$), b($x$) $\neq 0$ in $F_q[x]$ there exists a unique pair of polynomials q($x$), r($x$) in $F_q[x]$ such that

$$a(x) = q(x)b(x) + r(x),\ \deg\,(r(x)) < deg\,(b(x)).$$

Example Divide $x^3 + x + 1$ by $x^2 + x + 1$ in $F_2[x]$.

Definition Let f($x$) be a fixed polynomial in $F_q[x]$. Two polynomials g($x$), h($x$) are said to be congruent modulo f($x$), notation

$$g(x) \equiv h(x)\ (\mathrm{mod}\ f(x)),$$

if g($x$) - h($x$) is divisible by f($x$).

# EXAMPLE

The task is to determine all ternary codes of length 4 and generators for them.

Factorization of $x^4 - 1$ over $GF(3)$ has the form

$$x^4 - 1 = (x - 1)(x^3 + x^2 + x + 1) = (x - 1)(x + 1)(x^2 + 1)$$

Therefore there are $2^3 = 8$ divisors of $x^4 - 1$ and each generates a cyclic code.

| Generator polynomial | Generator matrix |
|---|---|
| $1$ | $I_4$ |
| $x$ | $\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$ |
| $x + 1$ | $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ |
| $x^2 + 1$ | $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ |
| $(x - 1)(x + 1) = x^2 - 1$ | $\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$ |
| $(x - 1)(x^2 + 1) = x^3 - x^2 + x - 1$ | $[\,-1\ 1\ -1\ 1\,]$ |
| $(x + 1)(x^2 + 1)$ | $[\,1\ 1\ 1\ 1\,]$ |
| $x^4 - 1 = 0$ | $[\,0\ 0\ 0\ 0\,]$ |

# Cyclic Code Encoder

Encoding using a cyclic code can be done by a multiplication of two polynomials - a message polynomial and the generating polynomial for the cyclic code.

Let $C$ be an $(n,k)$-code over an field $F$ with the generator polynomial

$g(x) = g_0 + g_1 x + \ldots + g_{r-1} x^{r-1}$ of degree $r = n - k$.

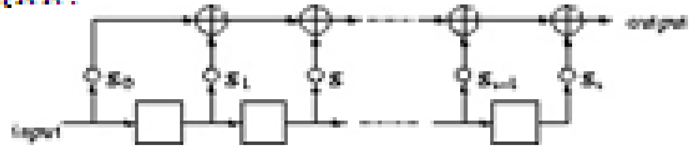
If a message vector $m$ is represented by a polynomial $m(x)$ of degree $k$ and $m$ is encoded by

$$m \Rightarrow c = mG_1,$$

then the following relation between $m(x)$ and $c(x)$ holds

$$c(x) = m(x)g(x).$$

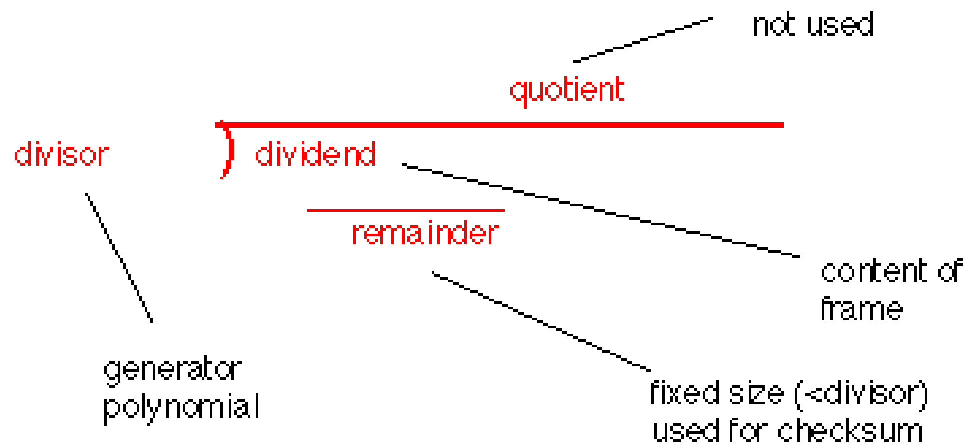
Such an encoding can be realized by the shift register shown in Figure below, where input is the $k$-bit message to be encoded followed by $n - k$ 0' and the output will be the encoded message



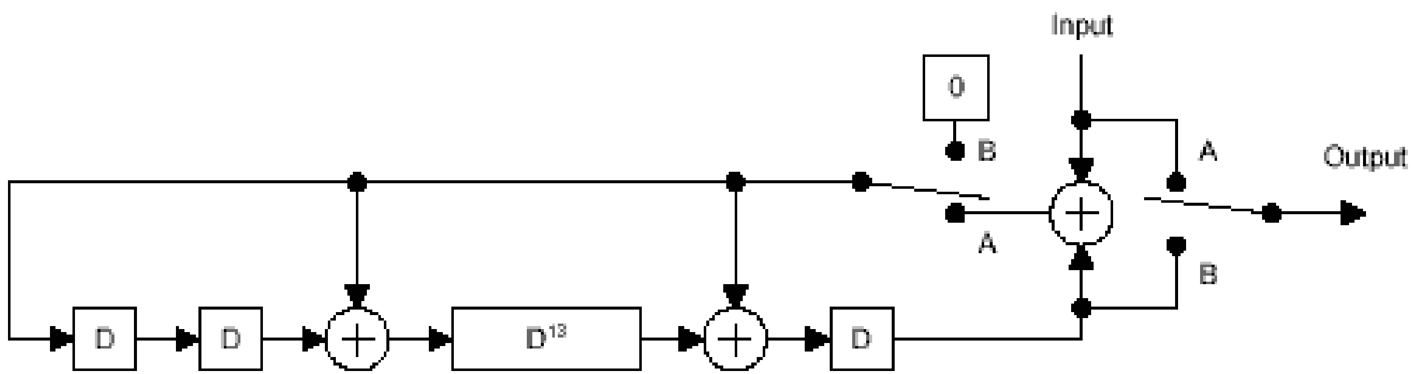
Shift-register encodings of cyclic codes. Small circles represent multiplication by the corresponding constant, ⊕ nodes represent modular addition, squares are delay elements

# Cyclic Code Decoder

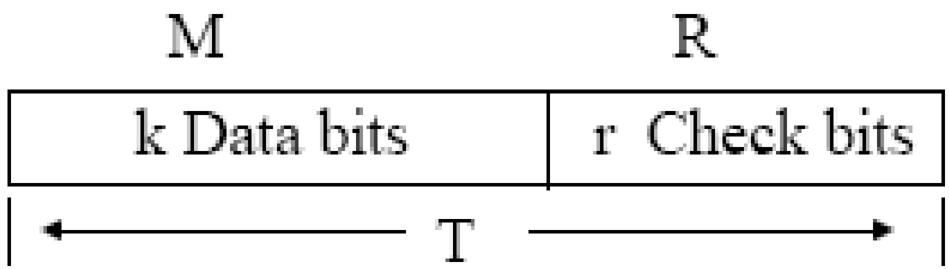
- Divider




- Similar structure as multiplier for encoder



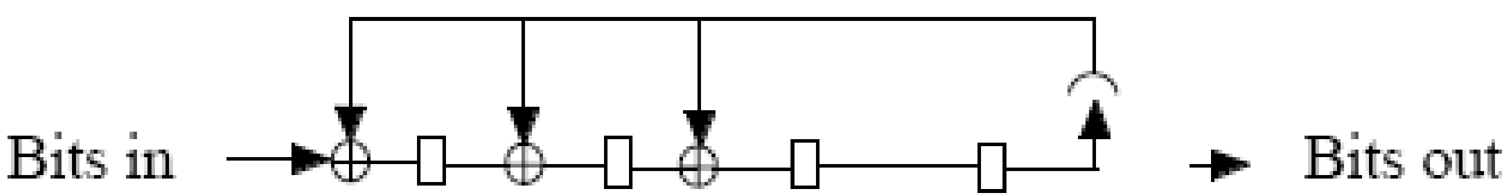
# Cyclic Redundancy Checks (CRC)




M = info bits
R = check bits
T = codeword

$$T = M\,2^r + R$$

A CRC is implemented using a feedback shift register



# **Example** of CRC

$r = 3$, $G = 1001$

$M = 110101 \Rightarrow M2^r = 110101000$

```
          110011
       ___________
1001  |  110101000
         1001
         ----
         01000
          1001
          ----
          0001100
             1001
             ----
             01010
              1001
              ----
               011 = R (3 bits)
```

Modulo 2 Division

# Checking for errors

- Let T' be the received sequence
- Divide T' by G
  - If remainder = 0 assume no errors
  - If remainder is non zero errors must have occurred

Example:
Send T = 110101011
Receive T' = 110101011
(no errors)

No way of knowing how many errors occurred or which bits are In error

```
1001 | 110101011
       1001
       ----
       01000
        1001
        ----
        0001101
           1001
           ----
           01001
            1001
            ----
             000 => No errors
```

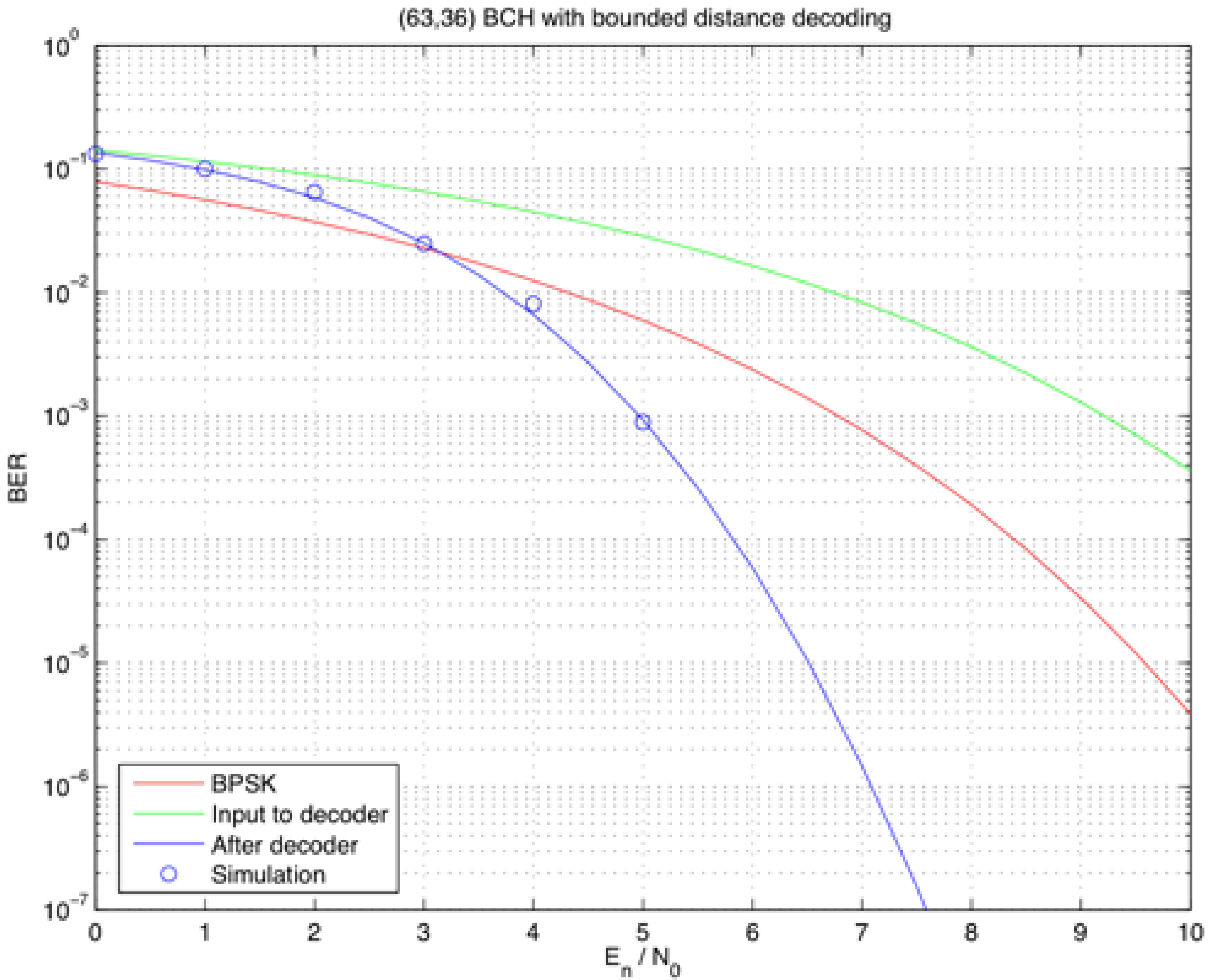
# Capability of CRC

- An error E(X) is undetectable if it is divisible by G(x). The following can be detected.
  - All single-bit errors if G(x) has more than one nonzero term
  - All double-bit errors if G(x) has a factor with three terms
  - Any odd number of errors, if P(x) contain a factor x+1
  - Any burst with length less or equal to n-k
  - A fraction of error burst of length n-k+1; the fraction is 1-2^(-(-n-k-1)).
  - A fraction of error burst of length greater than n-k+1; the fraction is 1-2^(-(n-k)).
- Powerful error detection; more computation complexity compared to Internet checksum

# BCH Code

- Bose, Ray-Chaudhuri, Hocquenghem
  - Multiple error correcting ability
  - Ease of encoding and decoding
- Most powerful cyclic code
  - For any positive integer m and t<2^(m-1), there exists a t-error correcting (n,k) code with n=2^m-1 and n-k<=mt.
- Industry standards
  - (511, 493) BCH code in ITU-T. Rec. H.261 “video codec for audiovisual service at kbit/s” a video coding a standard used for video conferencing and video phone.
  - (40, 32) BCH code in ATM (Asynchronous Transfer Mode)

# BCH Performance

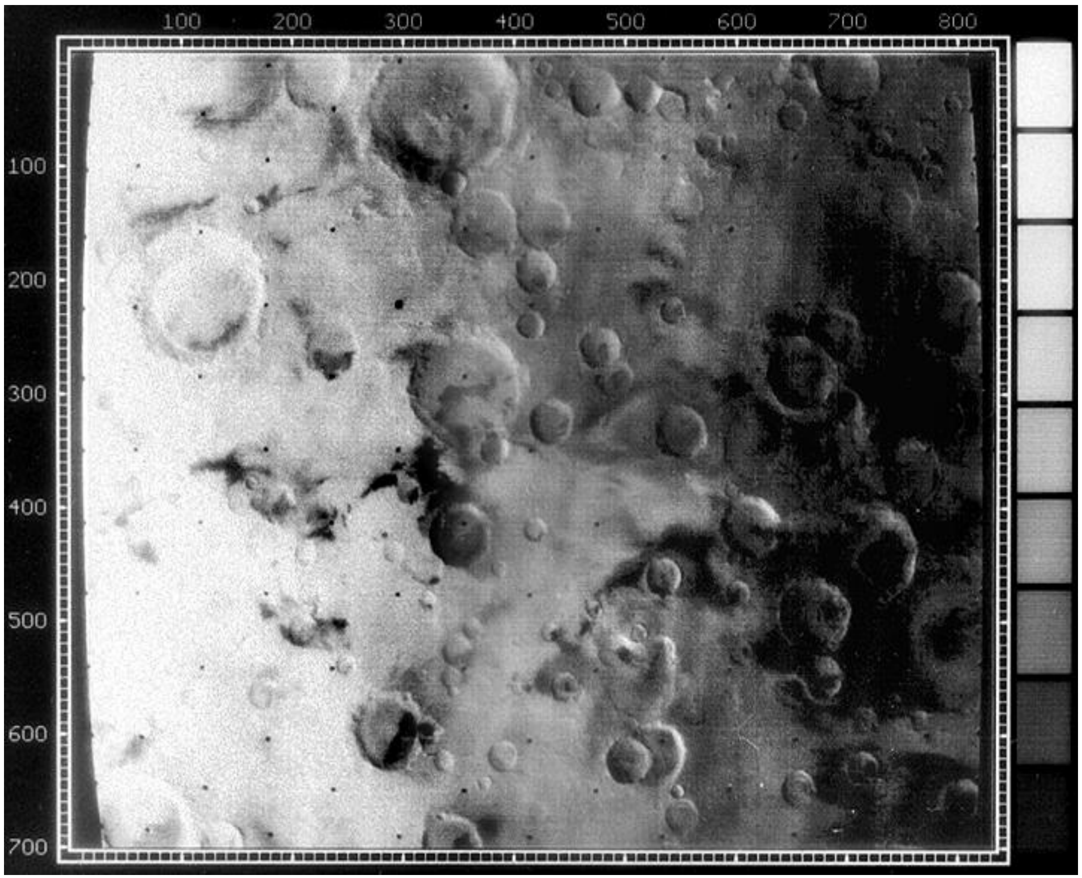


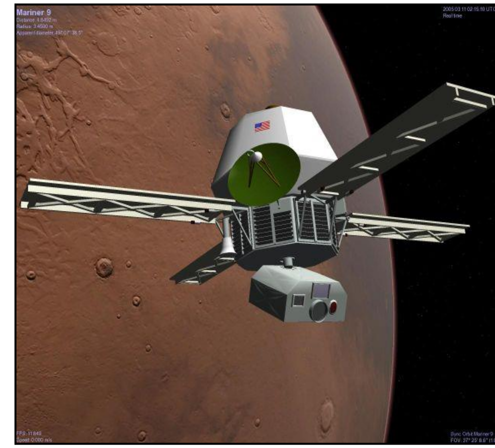
# Reed-Solomon Codes

- An important subclass of non-binary BCH
- Wide range of applications
  - Storage devices (tape, CD, DVD…)
  - Wireless or mobile communication
  - Satellite communication
  - Digital television/Digital Video Broadcast(DVB)
  - High-speed modems (ADSL, xDSL…)

# *1971: Mariner 9*

- Mariner 9 used a [32,6,16] *Reed-Muller* code to transmit its grey images of Mars.



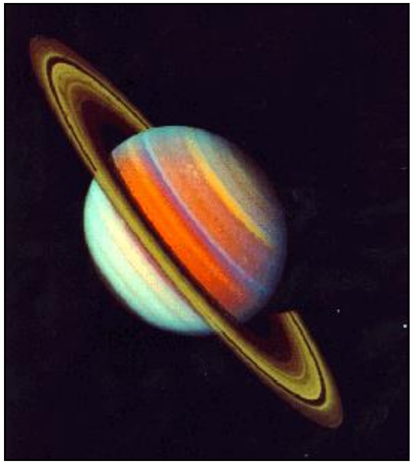

camera rate:

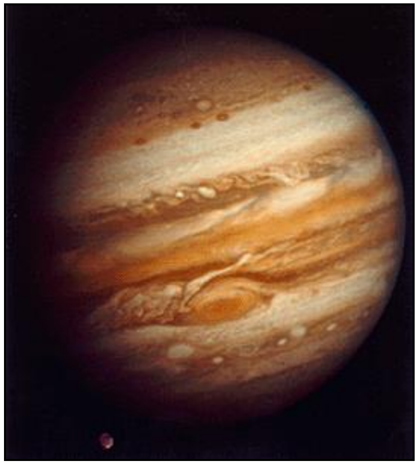
100,000 bits/second

transmission speed:
16,000 bits/second

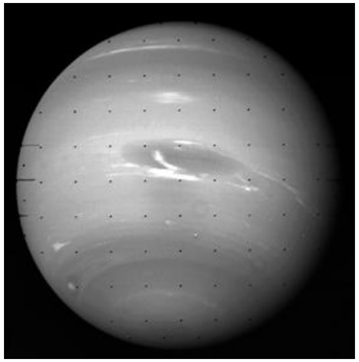
# *1979+: Voyagers I & II*

- Voyagers I & II used a [24,12,8] *Golay* code to send its color images of Jupiter and Saturn.



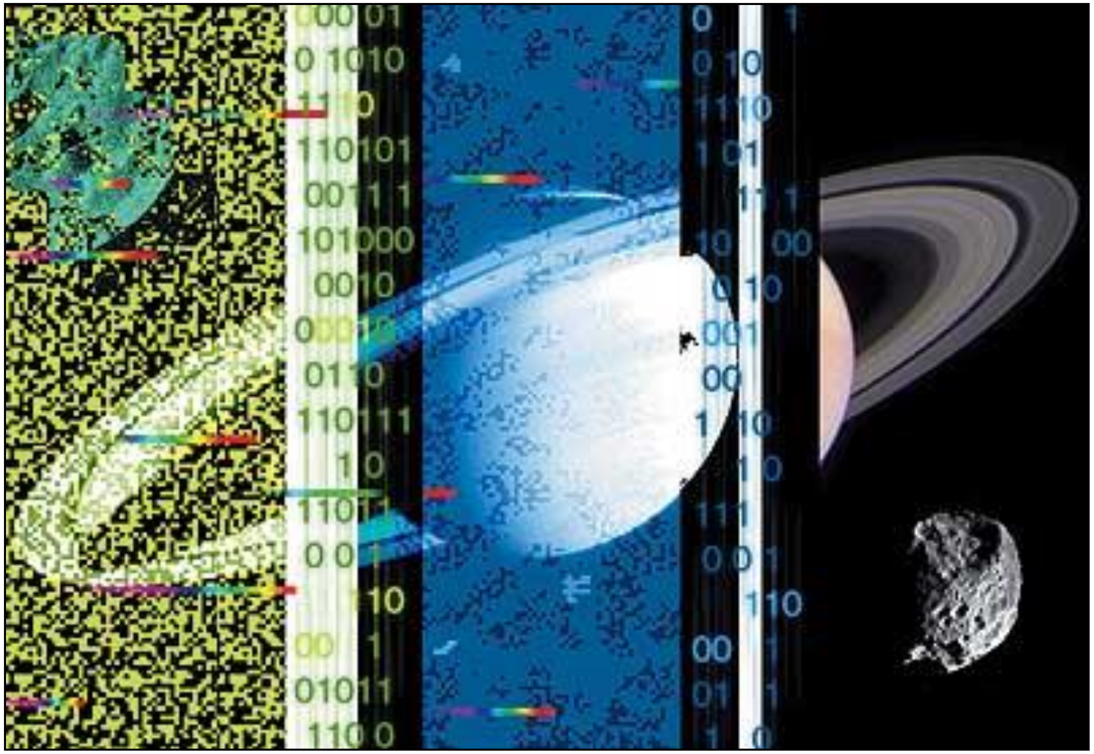


- Voyager 2 traveled further to Uranus and Neptune. Because of the higher error rate it switched to the more robust *Reed-Solomon* code.



# *Modern Codes*

- More recently *Turbo codes* were invented, which are used in 3G cell phones, (future) satellites, and in the Cassini-Huygens space probe [1997–].



- Other modern codes: Fountain, Raptor, LT, online codes…
- Next, next class

# Error Correcting Codes

**imperfectness** of a given code as the difference between the code's required Eb/No to attain a given word error probability (Pw), and the minimum possible Eb/No required to attain the same Pw, as implied by the sphere-packing bound for codes with the same block size *k* and code rate *r*.