# Introduction to Channel Coding

## I. Error control coding (overview)

- Shannon showed that reliable communications can be achieved by proper coding of information to be transmitted provided that the rate of information transmission is below the channel capacity.
- Coding is achieved by adding properly designed redundancy to each message before its transmission. The added redundancy is used for error control. *The redundancy may appear in the form of extra symbols (or bits), or in the form of channel signal-set expansion or in the form of combination of both*.
- Coding may be designed and performed separately from modulation, or designed in conjunction with modulation as a single entity. In the former case, redundancy appears in the form of extra symbols, normally called parity-check symbols.
- Coding achieved by adding extra redundant digits is known as ***conventional coding***, in which error control (or coding gain) is achieved at the expense of bandwidth expansion or data rate reduction. Therefore, conventional coding is suitable for error control in power limited channels, such as deep space channel.
- In the case that coding is designed in conjunction with modulation, redundancy comes from channel signal-set expansion. This combination of coding and modulation is usually known as ***coded modulation***, which allows us to achieve error control (or coding gain) without compromising bandwidth efficiency. We refer this technique as the bandwidth efficient coding.
- <u>**Historical notes:**</u>

  Hamming codes (1950)
  Reed-Muller codes (1954)
  BCH codes (by Bose, Ray-Chaudhuri and Hocquenghem, 1959) } BM 算法（1968）
  Reed-Solomon codes (1960) }
  Low-density parity-check codes (by Gallager in 1962, rediscovered in 90's)
  Convolutional codes (by Elias, 1955)
  Viterbi algorithm (1967)
  Concatenated codes (by Forney, 1966)
  Trellis-coded modulation (by Ungerboeck, 1982)
  Turbo codes (by Berrou , 1993)
  Space-time codes (by Vahid Tarokh,1998)
- Applications:
  Deep space, satellite, mobile communications, voice modem, data networks, etc.
- Two simple examples:

Repetition codes: $\left.\begin{array}{l} 0 \rightarrow 000000 \\ 1 \rightarrow 111111 \end{array}\right\}$ $(n, 1)$ code

Single parity-check codes:

$$\left.\begin{array}{cccccc} 0 & 0 & 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & 1 & \vdots & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 1 & 1 & \vdots & 0 \end{array}\right\} (n, n\text{-}1) \text{ code}$$

- **References:**


[1] Shu Lin and D. J. Costello, Jr. *Error Control Coding: Fundamentals and Applications*. 2nd ed. Prentice-Hall, 2004.

[2] 王新梅，肖国镇．*纠错码－原理与方法*．西安：西安电子科技大学出版社，1991.

[3] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Inform. Theory*, vol.44, no.6, pp.2531-2560, Oct. 1998.

[4] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proceedings of The IEEE*, vol.95, no.6, pp.1150-1177, June 2007.


## II. Block coding

- In block coding, information sequence is divided into messages of $k$ information bits (or symbols) each. Each message is mapped into a structured sequence of $n$ bits (with $n>k$), called a codeword.

$$\underbrace{(u_0, u_1, \cdots, u_{k-1})}_{\text{message}} \leftrightarrow \underbrace{(c_0, c_1, \cdots, c_{n-1})}_{\text{codeword}}$$

- The mapping operation is called encoding. Each encoding operation is independent of past encodings. The collection of all codewords is called an ***(n, k) block code,*** where $n$ and $k$ are the length and dimension of the code, respectively.
- In the process of encoding, n-k redundant bits are added to each message for protection against transmission errors.
- For example, consider a (5,2) binary code of size $M=2^k=4$:

$$\begin{array}{l} 00 \leftrightarrow 10101 = \mathbf{c}_1 \\ 01 \leftrightarrow 10010 = \mathbf{c}_2 \\ 10 \leftrightarrow 10010 = \mathbf{c}_3 \\ 11 \leftrightarrow 11110 = \mathbf{c}_4 \end{array} \qquad \mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$$

- An important class of block codes is the class of linear block codes. A block code is said to be linear if the vector sum of two codewords is also a codeword:

$$\mathbf{c}_i = (c_{i,0}, c_{i,1}, \cdots c_{i,n-1}) \in \mathcal{C}, \quad \mathbf{c}_j = (c_{j,0}, c_{j,1}, \cdots c_{j,n-1}) \in \mathcal{C}$$

$$\mathbf{c}_i \oplus \mathbf{c}_j = (c_{i,0} \oplus c_{j,0}, c_{i,1} \oplus c_{j,1}, \cdots, c_{i,n-1} \oplus c_{j,n-1}) \in \mathcal{C}$$

More general, a linear code is a subspace of $\mathrm{GF}(q)^n$. （矢量加、标量乘运算封闭）

- *Linear block codes* are normally put in systematic form:

$$(c_0, c_1, \cdots, c_{n-1}) = (\underbrace{c_0, c_1, \cdots, c_{n-k-1}}_{parity-check\ \ part}, \underbrace{u_0, u_1, \cdots, u_{k-1}}_{message\ \ part})$$

- Each parity-check bit is a linear sum of message bits, i,e,

$$c_j = \sum_{i=0}^{k-1} p_{ij} u_i, \qquad j = 0, 1, \cdots, n-k-1.$$

where $p_{ij}$=0 or 1. The $n-k$ equations which gives the parity-check bits are called the ***parity-check equations***. They specify the encoding rule.

- For an (n, k) block code, the ratios

$$R = \frac{k}{n} \qquad \text{and} \qquad \eta = \frac{n-k}{n}$$

are called ***code rate*** and redundancy, respectively.

- An example for block code:

Let $n$=7 and $k$=4. Consider the (7, 4) linear systematic block code

Message: $(u_0, u_1, u_2, u_3)$

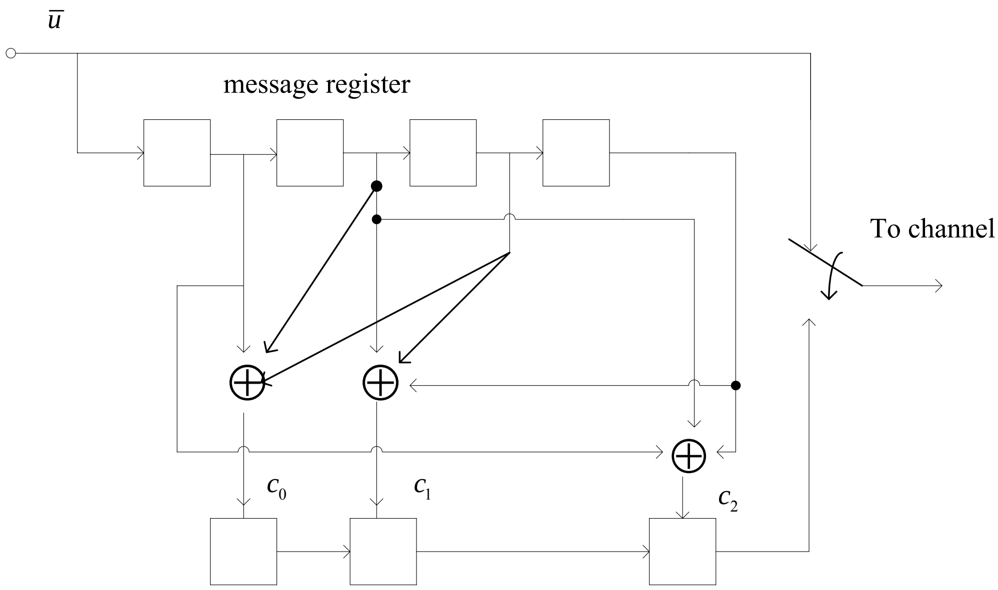
Codeword: $(c_0, c_1, c_2, c_3, c_4, c_5, c_6) = (c_0, c_1, c_2, u_0, u_1, u_2, u_3)$

Here,
$$c_0 = u_0 + u_1 + u_2$$
$$c_1 = u_1 + u_2 + u_3$$
$$c_2 = u_0 + u_1 + u_3$$

In matrix form:

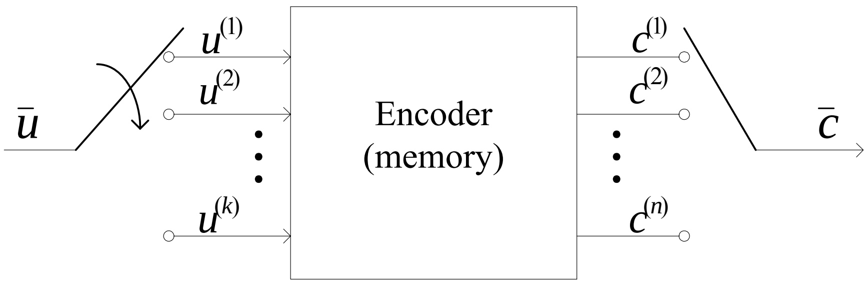
$$\mathbf{c} = (u_0, u_1, u_2, u_3)\begin{bmatrix} 1 & 0 & 1 & \vdots & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & \vdots & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & \vdots & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & \vdots & 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{u} \cdot \mathbf{G}$$

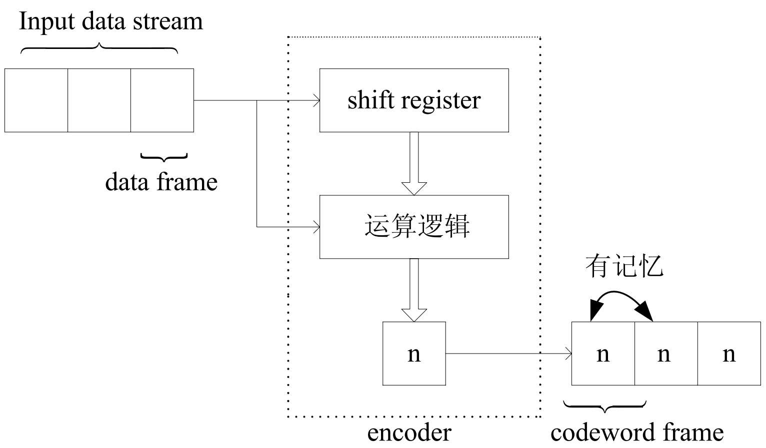
Encoder circuit:




## III. Convolution Coding

- During each unit of time, the input to convolutional is also a $k$-bit message block and the corresponding is also an $n$-bit coded with $k<n$. (为避免混淆，可改为用$k_0$, $n_0$表示)
- Each coded $n$-bit output block depends not only on the corresponding $k$-bit input message block at the same time unit but also on the $m$ previous message blocks.



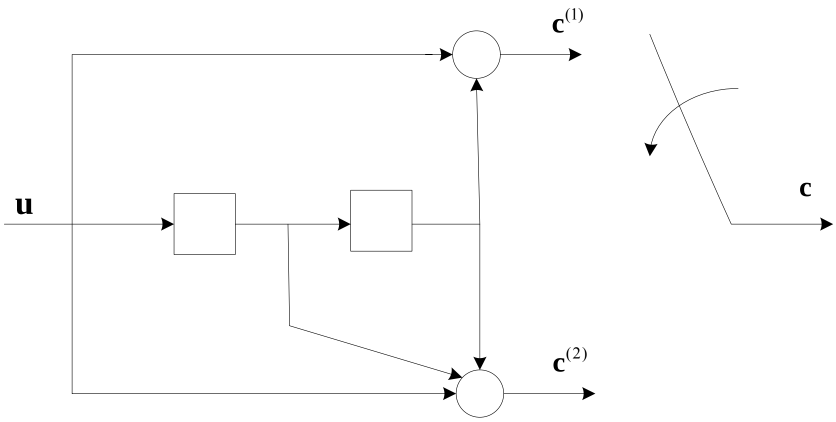




- The code rate is defined as $R = k/n$.
  The parameter $m$ is called the memory order of the code.

- In the process of coding, the information sequence **u** is divided into data frames of length k. These subsequences of length k are applied to the *k*-input terminals of the encoder, producing coded sequence of length *n*.
- An example:

  Let *n*=2, *k*=1 and *m*=2. Consider a rate-1/2 (2,1,2) convolutional code which is specified by the following two generator sequences:

$$\mathbf{g}^{(1)} = (101), \quad \mathbf{g}^{(2)} = (111)$$




Note: $\mathbf{g}^{(1)}$, $\mathbf{g}^{(2)}$ 可看作编码器的两个冲激响应，由 $\mathbf{u} = \delta = (100...)$ 得到。冲激响应至多持续 $m+1$ 个时间单位，且可写为：

$$\mathbf{g}^{(1)} = \left(g_0^{(1)}, g_1^{(1)}, \ldots, g_m^{(1)}\right), \mathbf{g}^{(2)} = \left(g_0^{(2)}, g_1^{(2)}, \ldots, g_m^{(2)}\right)$$

- Let $\mathbf{u} = (u_0, u_1, \cdots)$ be the input message sequence. Then the two output sequences are

$$\left.\begin{array}{l} \mathbf{c}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)} \\ \mathbf{c}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)} \end{array}\right\} \text{编码方程} \qquad \text{（与冲激响应的卷积运算）}$$

- At the *l*th time unit, the input is a single bit $u_l$. The corresponding output is a block of two bits, $(c_l^{(1)}, c_l^{(2)})$, which is given by

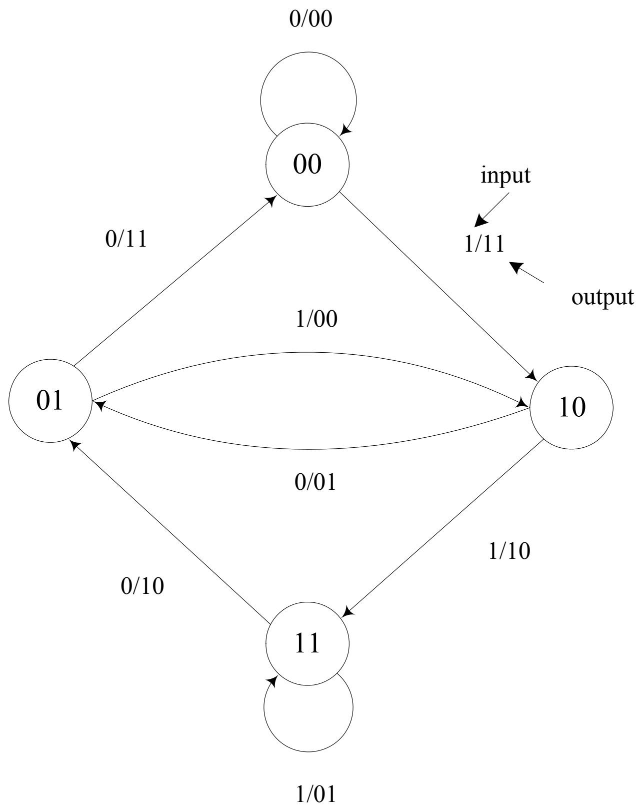
$$c_l^{(j)} = \sum_{i=1}^{m} u_{l-i} g_i^{(j)} = u_l g_0^{(j)} + u_{l-1} g_1^{(j)} + \cdots + u_{l-m} g_m^{(j)}$$

$$\Rightarrow \begin{cases} c_l^{(1)} = u_l \qquad + u_{l-2} \\ c_l^{2} = u_l + \underbrace{u_{l-1} + u_{l-2}}_{memory} \end{cases}$$

- The output codeword is given by $\mathbf{c} = \left(c_0^{(1)} c_0^{(2)}, c_1^{(1)} c_1^{(2)}, \cdots\right)$.

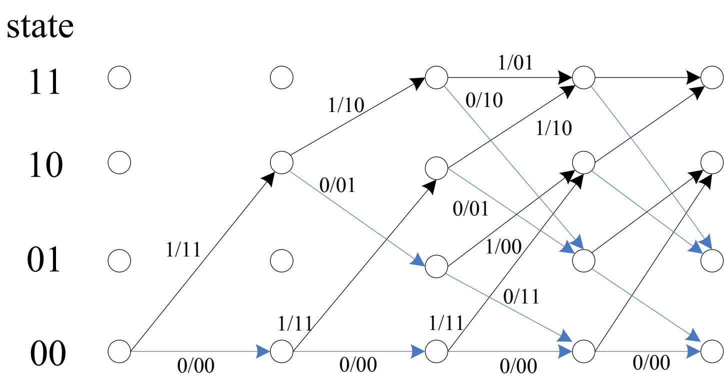
For $\mathbf{u} = (1011100\cdots), \mathbf{c} = (11, 01, 00, 10, 01, 10, 11, \cdots)$

- <u>State Diagram:</u> Since the encoder is a linear sequential circuit, its behavior can be described by a state diagram. The encoder state at time $l$ is represented by the message bits stored in the memory units.
- The encoder of the (2, 1, 2) convolutional code given in the example has 4 possible states, and its state diagram is shown in the figure below .




label = input/output

- **<u>Trellis diagram</u>**: The state diagram can be expanded in time to display the state transition of a convolutional encoder in time. This expansion in time results in a trellis diagram.




- The encoding of a message sequence **u** is equivalent to tracing a path through the trellis.
- The trellis structure is very useful in decoding a convolutional code.

# IV. Conventional Coding

## 1. Types of codes

$\left.\begin{cases}\text{block codes - linear codes, cyclic codes}\\ \text{convolutional codes}\end{cases}\right\}$ classfication based on structure

$\begin{cases}\text{random-error-correcting codes}\\ \text{burst-error-correcting codes}\end{cases}$

$\begin{cases}\text{Binary codes}\\ \text{Nonbinary codes}\end{cases}$

$\begin{cases}\text{error-correction codes}\\ \text{error-detection codes}\end{cases}$

## 2. Error correcting capacity/ability

- The error correcting capacity of a code $\mathcal{C}$ depends on its distance structure.
- The ***Hamming distance*** between two codewords, **x** and **y**, in a code, denoted by $d_{\mathrm{H}}(\mathbf{x},\mathbf{y})$, is defined as the number of places in which they differ.

$$d_{\mathrm{H}}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} d_H(x_i, y_i), \quad d_{\mathrm{H}}(x_i, y_i) = \begin{cases} 1, & \text{if } x_i \neq y_i \\ 0, & \text{if } x_i = y_i \end{cases}$$

$$\text{or } d_{\mathrm{H}}(\mathbf{x},\mathbf{y}) = |\{i : x_i \neq y_i\}|$$

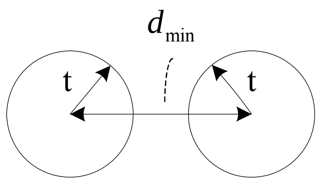
For example, $d_H(010,111) = 2$, $\quad d_H(30102, 21103) = 3$

- Hamming distance satisfies the axioms for a distance metric:
  1) $d_H(\mathbf{x},\mathbf{y}) \geq 0,$ with equality iff $\mathbf{x} = \mathbf{y}$
  2) $d_H(\mathbf{x},\mathbf{y}) = d_H(\mathbf{y},\mathbf{x})$ (对称性)
  3) $d_H(\mathbf{x},\mathbf{y}) \leq d_H(\mathbf{x},\mathbf{z}) + d_H(\mathbf{z},\mathbf{y})$
- ***The minimum Hamming distance*** of a code $\mathcal{C}$ is defined as

$$d_{\min} \triangleq \min\left\{d_H(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\right\}$$

- For a convolutional code, this minimum Hamming distance is usually called the minimum free distance, denoted by $d_{free}$.
- An (*n*, *k*) block code with minimum Hamming distance $d_{\min}$ is capable of correcting

$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$ or fewer random errors over a block of $n$ digits (using minimum distance decoding rule). This parameter $t$ is called the error correcting capacity of the code.



decoding sphere $d_{\min} \geq 2t + 1$

- Error detection ability: $e = d_{\min} - 1$
- Erasure correction ability: $\rho = d_{\min} - 1$
- Up to $\rho$ erasures and $t$ random errors can be corrected if $d_{\min} \geq 2t + \rho + 1$.
- The minimum Hamming distance of a linear block code depends on the choice of parity-check equations and the number of parity bits, $n - k$.

**3. Important codes**

1) Algebraic block codes

- Hamming codes

- BCH codes: A large class of powerful multiple random error-correcting codes, rich in algebraic structure, algebraic decoding algorithms available.

- Golay (23, 12) code: A perfect triple-error-correcting code, widely used and generated by

$$g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$$

- Reed-Muller codes

- Reed-Solomon codes: nonbinary, correcting symbol errors or burst errors ,most widely used for error control in data communications and data storages.

2) Convolutional codes: (2, 1, 6) code generated by

$$\mathbf{g}^{(1)} = (1101101), \quad \mathbf{g}^{(2)} = (1001111)$$

This code has $d_{\text{free}} = 10$.

3) Codes (defined) on graphs:

$$\left.\begin{array}{l}\text{Low-density parity-check codes} \\ \text{Turbo codes}\end{array}\right\} \text{capacity-approching codes}$$

**4. Types of error control schemes**

- Forward-error-correction (FEC): An error-correction code is used.

- Automatic-repeat-request (ARQ): An error-detection code is used.

If the presence of error is detected in a received word, a retransmission is requested. The request signal is sent to the transmitter through a feedback channel. Retransmission continues until no errors being detected.

- Hybrid ARQ: A proper combination of FEC and ARQ.

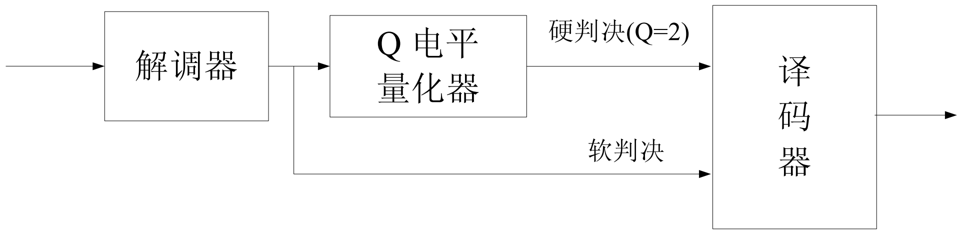
**5. Decoding**

- Based on the received sequence, the encoding rules and the noise characteristics of the channel, the receiver makes a decision which message was actually transmitted. This decision making operation is called decoding.
- Hard-decision

When binary coding is used, the modulator has only binary inputs ($M$=2). If binary demodulator output quantization is used ($Q$=2), the decoder has only binary inputs. In this case, the demodulator is said to make hard decision. Decoding based on hard decisions made by the demodulator is called *hard-decision decoding*.

- Soft-decision

If the output of demodulator consists of more than two quantization levels (Q>2) or is left unquantized, the demodulator is said to make soft decisions. Decoding based on this is called ***soft-decision decoding***.




- Hard-decision decoding is much easier to implement than soft-decision decoding. However, soft-decision decoding offers significant performance improvement over hard-decision decoding. See figure 2.

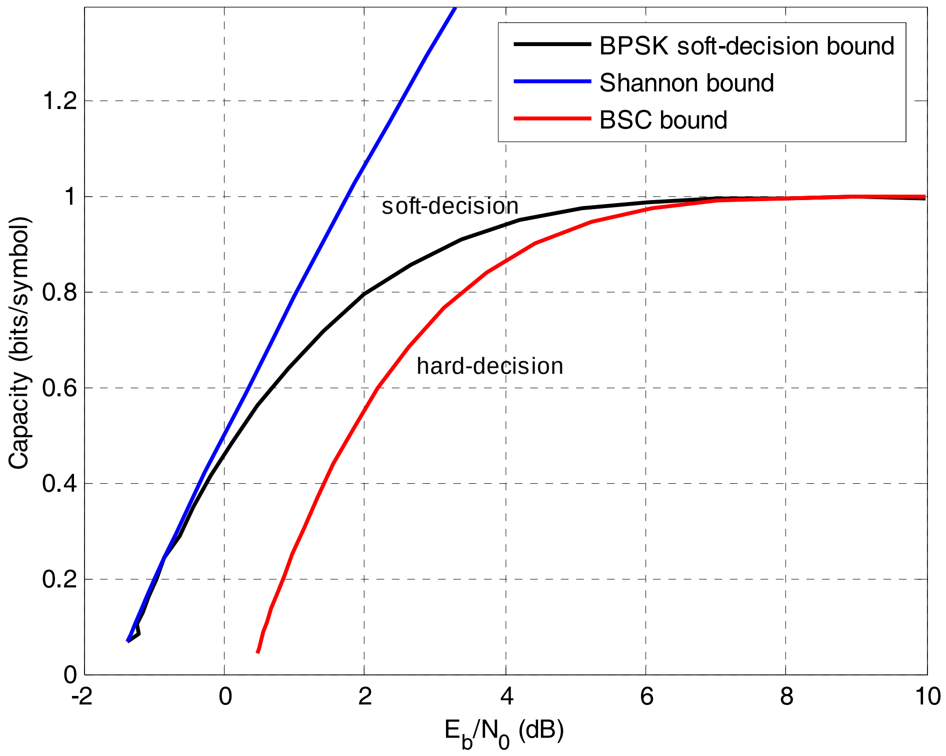



Figure 2 软判决与硬判决译码的信道容量

- Optimal decoding

  Given that $\mathbf{y}$ is received, the conditional error probability of decoding is defined as

$$P(E|\mathbf{y}) \triangleq P(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{y})$$

  Then the error probability of

$$P(E) = \sum_{\mathbf{y}} P(E|\mathbf{y})P(\mathbf{y})$$

  A decoding rule that minimizes $P(E)$ is referred to as an optimal decoding rule.

  Since minimize $P(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{y})$ is equivalent to maximize $P(\hat{\mathbf{c}} = \mathbf{c}|\mathbf{y})$, we have

  ***MAP rule***: $\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{c}|\mathbf{y})$

- Maximum-likelihood decoding (MLD):

  Note that $P(\mathbf{c}|\mathbf{y}) = \dfrac{P(\mathbf{c})P(\mathbf{y}|\mathbf{c})}{P(\mathbf{y})}$, we have

  ***ML rule***: $\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{y}\,|\,\mathbf{c})$ (Suppose all the messages are equally likely)

## 6. MLD for a BSC

In coding for a BSC, every codeword and every received word are binary sequences.

- Suppose some codeword is transmitted and the received word is $\mathbf{y} = (y_1, y_2, \ldots, y_n)$.

For a codeword $\mathbf{c}_i$, the conditional probability $P(\mathbf{y}\,|\,\mathbf{c}_i)$ is

$$P(\mathbf{y}\,|\,\mathbf{c}_i) = p^{d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i)}(1-p)^{n-d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i)}$$

For $p$<1/2, $P(\mathbf{y}\,|\,\mathbf{c}_i)$ is a monotonially decreasing function of $d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i)$. Then

$$P(\mathbf{y}\,|\,\mathbf{c}_i) > P(\mathbf{y}\,|\,\mathbf{c}_j) \text{ iff } d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i) < d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_j)$$

- MLD:

1) Compute $d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i)$ for all $\mathbf{c}_i \in \mathcal{C}$.

2) $\mathbf{c}_i$ is taken as the transmitted codeword if $d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_i) < d_{\mathrm{H}}(\mathbf{y},\mathbf{c}_j)$ for $\forall j \neq i$.

3) Decoding $\mathbf{c}_i$ into message $\mathbf{u}_i$.

This is called the ***minimum distance (nearest neighbor) decoding***.

**7. Performance measure and coding gain**

- Block-error probability: It is the probability that a decoded word is in error.
- Bit-error probability: It is the probability that a decoded bit is in error.
- The usual figure of merit for a communication system is the ratio of energy per information bit to noise power spectral density, $E_b / N_0$, that is required to achieve a given error probability.
- Coding gain of a coded communication system over an uncoded system with the same modulation is defined the reduction, expressed in dB, in the required $E_b/N_0$ to achieve a target error probability.

$$\text{Coding gain} = \left[\frac{E_b}{N_0}\right]_{uncoded} - \left[\frac{E_b}{N_0}\right]_{coded} \quad \text{(in dB)}$$

- **Shannon limit**: A theoretical limit on the minimum SNR required for coded system with code rate $R_c$ to achieve error-free information transmission. See figures 3 and 4.

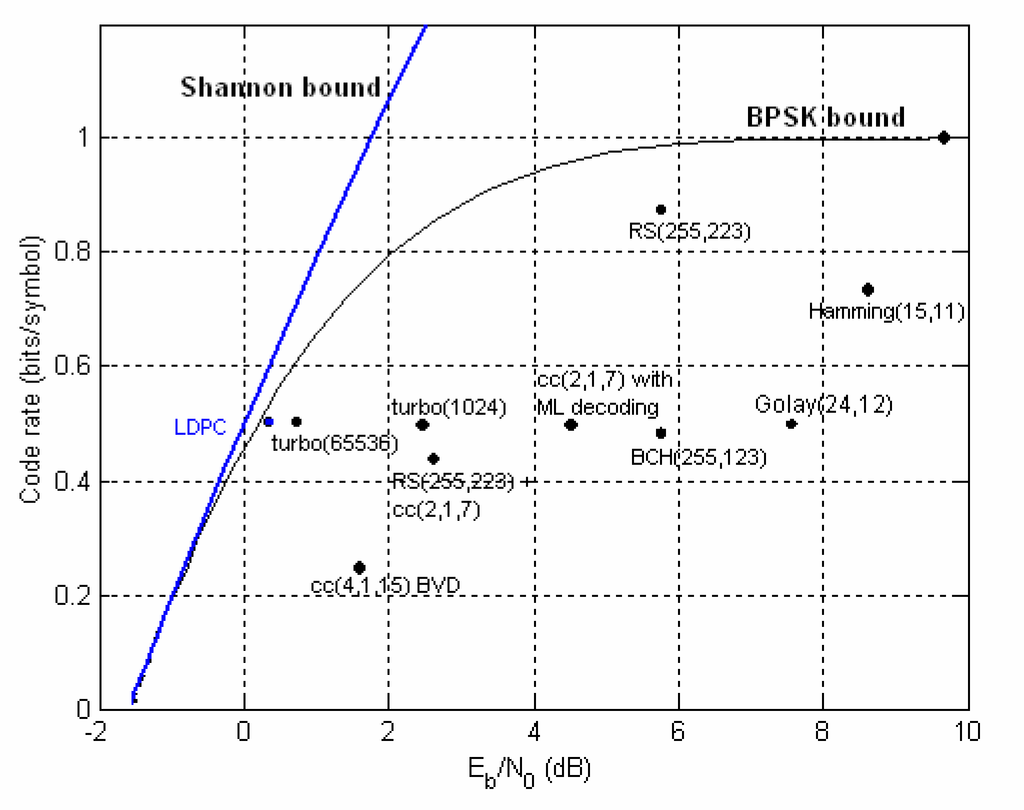



Figure 3 Milestones in the drive towards channel capacity achieved over the past 50 years.

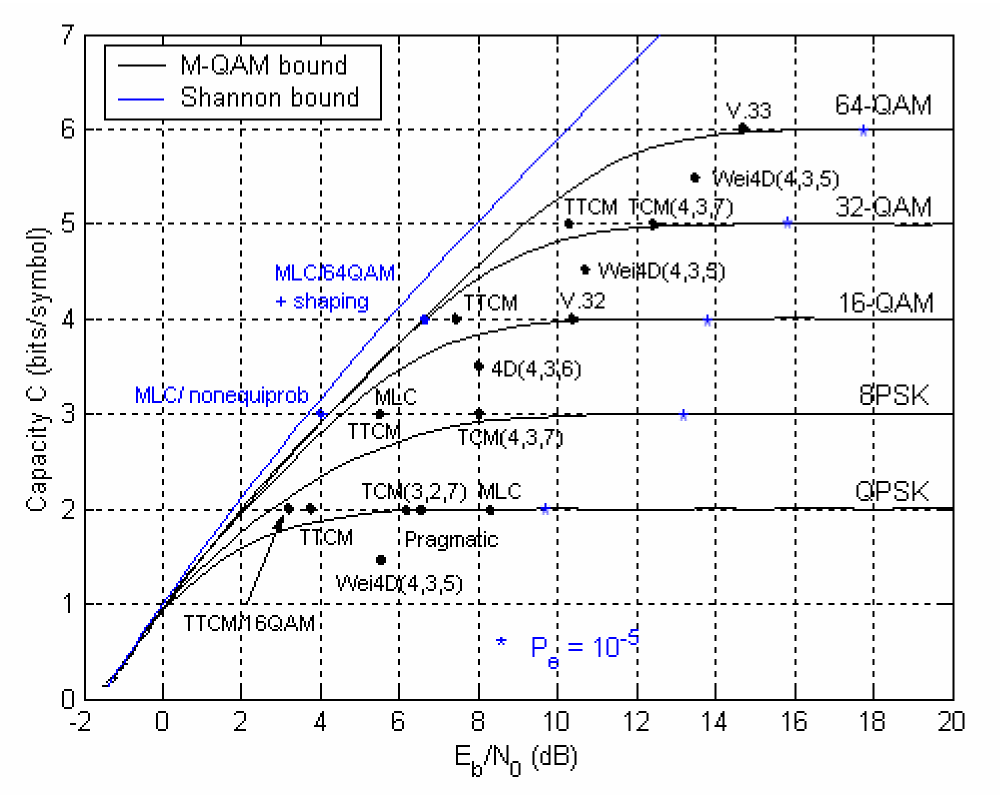



Figure 4 Performance of various coded modulation schemes.

## V. Finite Fields (分组码的代数结构)

### *1. Binary arithmetic and field*

- Consider the binary set {0,1}, Define two binary operations, called addition '+' and multiplication '·', on {0,1} as follows

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| · | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

- These two operations are commonly called module-2 addition and multiplication respectively. They can be implemented with an XOR and an AND gate, respectively.
- The set {0, 1} together with module-2 addition and multiplication is called a binary field, denoted by GF(2) or $\mathbb{F}_2$ .

### *2. Vector space over GF(2)*

- A binary $n$-tuple is an ordered sequence, $(a_1,a_2,\ldots,a_n)$, with $a_i \in \text{GF}(2)$ .

  - There are $2^n$ distinct binary $n$-tuples.
  - Define an addition operation for any two binary $n$-tuples as follows:

$$(a_1, a_2, ..., a_n) + (b_1, b_2, ..., b_n) = (a_1 + b_1, a_2 + b_2, ..., a_n + b_n)$$

  where $a_i+b_i$ is carried out in module-2 addition.

  - Define a scalar multiplication between and element $c$ in GF(2) and a binary $n$-tuple $(a_1,a_2,\ldots,a_n)$ as follows:

$$c \cdot (a_1, a_2, ..., a_n) = (c \cdot a_1, c \cdot a_2, ..., c \cdot a_n)$$

  where $c \cdot a_i$ is carried out in module-2 multiplication.

- Let $V^n$ denote the set of all $2^n$ binary $n$-tuples. The set $V^n$ together with the addition defined for any two binary $n$-tuples in $V^n$ and the scalar multiplication is called a vector space over GF(2). The elements in $V^n$ are called vectors.
  - Note that $V^n$ contains the all-zero $n$-tuple (0,0,…,0) and

$$(a_1, a_2, ..., a_n) + (a_1, a_2, ..., a_n) = (0, 0, ..., 0)$$

- Example: Let $n$=4. Then

$$V^4 = \begin{Bmatrix} (0000), (0001), (0010), (0011), (0100), (0101), (0110), (0111) \\ (1000), (1001), (1010), (1011), (1100), (1101), (1110), (1110) \end{Bmatrix}$$

- A subset $\mathcal{S}$ of $V^n$ is called a subspace of $V^n$ if

  1) the all-zero vector is in $\mathcal{S}$;

2) the sum of two vectors in $\mathcal{S}$ is also a vector in $\mathcal{S}$;

For example: $\mathcal{S}$={(0000),(0101),(1010),(1111)} forms a subspace of $V^4$.

- A linear combination of $k$ vectors, $\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_k$ in $V^n$ is a vector of the form

$$\mathbf{V} = c_1\mathbf{V}_1 + c_2\mathbf{V}_2 + \cdots + c_k\mathbf{V}_k$$

where $c_i \in \mathrm{GF}(2)$ and is called the coefficient of $\mathbf{V}_i$.

- The subspace formed by the $2^k$ linear combinations of $k$ linearly independent vectors $\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_k$ in $V^n$ is called a $k$-dimensional subspace of $V^n$.

- A binary polynomial is a polynomial with coefficients from the binary field.
For example, $1+x^2$, $1+x+x^3$.
  - A binary polynomial $p(x)$ of degree $m$ is said to be irreducible if it is not divisible by any binary polynomial of degree less than $m$ and greater than zero. For example, $1+x+x^2$, $1+x+x^3$.
  - An irreducible polynomial $p(x)$ of degree $m$ is said to be ***primitive*** if the smallest positive integer $n$ for which $p(x)$ divides $x^n+1$ is $n=2^m-1$. For example, $p(x)=1+x+x^4$. (it divides $x^{15}+1$)
  - For any positive integer $m$, there exists a primitive polynomial of degree $m$. (可查表)

**4. Galois fields**

- Groups: A group is an algebraic structure (G, *) consisting of a set G and an operation * satisfying the following axioms:
  1) Closure: For any $a, b\in G$, the element $a*b$ is in G;
  2) Associative law: For any $a, b, c\in G$, $a*(b*c)=(a*b)*c$;
  3) Identity element: There is an element $e\in G$ for which $e*a=a*e=a$ for all $a\in G$;
  4) Inverse: For every $a\in G$, there exists a unique element $a^{-1}\in G$, such that $a*a^{-1} = a^{-1}*a = e$.
- A group is called a commutative group or **Abelian group** if $a*b = b*a$ for all $a, b\in G$.
Examples: - 整数，有理数，实数，with addition;
  - Integers with module-m addition.

- Fields: A field $\mathbb{F}$ is a set that has two operations defined on it : Addition and multiplication, such that the following axioms are satisfied:
  1) The set is an Abelian group under addition; (单位元称为‘0’)
  2) The set is closed under multiplication, and the set $\{a\in\mathbb{F}, a\neq 0\}$ forms an Abelian group (whose identity is called ‘1’) under the multiplication (*);
  3) Distributive law: For all $a, b, c\in\mathbb{F}$, $(a+b)*c=(a*c)+(b*c)$.

我们经常用‘0’表示加法运算下的单位元，‘-a’表示a的加法逆。经常用‘1’表示乘法

运算下的单位元，'$a^{-1}$'表示a的乘法逆。这样，减法a-b means a+(-b), 除法a/b means $b^{-1}a$.

- A field with $q$ elements, if it exists, is called a ***finite field***, or a *Galois field*, and is denoted by GF($q$).

  For example, $GF(2)$——the smallest field

- Let $\mathbb{F}$ be a field. A subset of $\mathbb{F}$ is called a subfield if it is a field under the inherited addition and multiplication. 原来的域$\mathbb{F}$称为 an extension field of the subfield.

- In any field, if $ab$=$ac$ and a≠0, then b=c.
- For any positive integer $m$≥1,there exists a Galois field of $2^m$ elements, denoted by $GF(2^m)$.

- The construction of $GF(2^m)$ is very much the same as the construction of the complex-number field from the real-number field.

We begin with a primitive polynomial $p(x)$ of degree $m$ with coefficients from the binary field $GF(2)$.

- Let $\alpha$ be the root of $p(x)$; i.e., $p(\alpha)$=0. Then the field elements can be represented by $\{0,1,\alpha,\alpha^2,...,\alpha^{q-2}\}$, where $q$=$2^m$.

$$0=\alpha^{-\infty},\ 1=\alpha^0,\ \alpha^1,\ \alpha^2,\ \ldots,\ \alpha^{q-2},$$

$1$=$\alpha^{q-1}$ (since $\alpha$ is a root of $p(x)$ and $p(x)\,|\,x^{2^m-1}+1$, $\alpha$ must be a root of $x^{2^m-1}+1$)

– For example: Construct $GF(4)$ from $GF(2)$ using $p(x)=x^2+x+1$.

| Polynomial notation | Binary notation | Integer | Exponential |
|---|---|---|---|
| 0 | 00 | 0 | 0 |
| 1 | 01 | 1 | 1 |
| $x$ | 10 | 2 | $\alpha$ |
| $x$+1 | 11 | 3 | $\alpha^2$<br>$\alpha^3=1=\alpha^0$ |

- The element $\alpha$ whose powers generate all the nonzero elements of $GF(2^m)$ is called ***a primitive element*** of $GF(2^m)$.

## VI. Binary Linear Block codes

- **An (*n, k*) linear block code over $GF(2)$ is simply a *k*-dim subspace of the vector**

**space $V^n$ of all the binary *n*-tuples**.

- In any linear code, the all-zero word, as the vector-space origion, is always a codeword (∵ if **c** is a codeword, then (-**c**) is also a codeword, so dose **c**+(-**c**)).
- The Hamming weight $w(\mathbf{c})$ of a vector **c** is the number of nonzero components of **c**. Obviously, $w(\mathbf{c}) = d_{\mathrm{H}}(\mathbf{c}, \mathbf{0})$.
- The minimum Hamming weight of a code $\mathcal{C}$ is the smallest Hamming weight of any nonzero codeword of $\mathcal{C}$.

$$w_{\min} = \min_{\mathbf{c}\in\mathcal{C}, \mathbf{c}\neq 0} w_H(\mathbf{c})$$

For a linear code, $d_H(\mathbf{c}_1, \mathbf{c}_2) = d_H(\mathbf{0}, \mathbf{c}_2 - \mathbf{c}_1) = d_H(\mathbf{0}, \mathbf{c}) = w(\mathbf{c})$

$$d_{\min} = \min\left\{ d_H\left(\mathbf{0}, \mathbf{c}_i - \mathbf{c}_j\right) \middle| \mathbf{c}_i, \mathbf{c}_j \in \mathcal{C}, i \neq j \right\} = \min_{\mathbf{c}\neq 0} w(\mathbf{c}) = w_{\min}$$

### *1. Generator matrix*

- A generator matrix for a linear block code $\mathcal{C}$ of length *n* and dimension *k* is any *k*×*n* matrix **G** whose rows form a basis for $\mathcal{C}$.

Every codeword is a linear combination of the rows of **G**.

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} \cdots & g_{0,n-1} \\ g_{10} & g_{11} \cdots & g_{1,n-1} \\ \vdots & & \\ g_{k-1,0} & g_{k-1,1} \cdots & g_{k-1,n-1} \end{bmatrix}_{k\times n}$$

- Encoding procedure: $\mathbf{c} = \mathbf{u}\cdot\mathbf{G} = [u_0, u_1, \cdots u_{k-1}] \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \sum_{l=0}^{k-1} u_l \mathbf{g}_l$

Example: For a (6, 3) linear block code,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} 011100 \\ 101010 \\ 110001 \end{bmatrix}$$

The codeword for the message $\mathbf{u} = (1\,0\,1)$ is

$$\mathbf{c} = \mathbf{u}\cdot\mathbf{G} = 1\cdot(011100) + 0\cdot(101010) + 1\cdot(110001) = (101101)$$

- An (*n*, *k*) linear systematic code is completely specified by an *k*×*n* generator matrix of the following form:

$$\mathbf{G}=\begin{bmatrix}\mathbf{g}_0\\ \mathbf{g}_1\\ \vdots\\ \mathbf{g}_{k-1}\end{bmatrix}=\Bigg[\underbrace{\begin{matrix}p_{00} & p_{01}\cdots p_{0,n-k-1}\\ p_{10} & p_{11}\cdots p_{1,n-k-1}\\ \vdots & \\ p_{k-1,0} & p_{k-1,1}\cdots p_{k-1,n-k-1}\end{matrix}}_{P\ matrix\ with\ p_{ij}=0 or 1}\ \vdots\ \underbrace{\begin{matrix}1\,0\cdots0\\ 0\,1\cdots0\\ \vdots\\ 0\,0\cdots1\end{matrix}}_{k\times k\ identity\ matrix}\Bigg]=\left[\mathbf{P}\,\vdots\,\mathbf{I}_k\right]$$

For example, the (6, 3) code above is a systematic code:

$$c_5=u_2$$
$$c_4=u_1$$
$$c_3=u_0$$

$$\left.\begin{aligned}c_2&=u_0+u_1\\ c_1&=u_0+\quad u_2\\ c_0&=\quad u_1+u_2\end{aligned}\right\}\text{ parity-check equations}$$

## 2. *Parity-check matrix*

- An ($n$, $k$) linear code can also be specified by an $(n\text{-}k)\times n$ matrix **H**.

  Let $\mathbf{c}=(c_0\ c_1\cdots c_{n-1})$ be an $n$-tuple. Then **c** is a codeword iff

$$\mathbf{c}\cdot\mathbf{H}^T=\mathbf{0}=\underbrace{(0\,0\cdots0)}_{n-k个}$$

  The matrix **H** is called a parity-check matrix. By definition,

$$\mathbf{G}\mathbf{H}^T=\mathbf{0}$$

- For an ($n$, $k$) linear systematic code with generator matrix $\mathbf{G}=[\mathbf{P}\quad\mathbf{I}_k]$, the parity-check matrix is

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}_0\\ \mathbf{h}_1\\ \vdots\\ \mathbf{h}_{n-k}\end{bmatrix}=\left[\mathbf{I}_{n-k}\,\vdots-\mathbf{P}^T\right]$$

- Example: For (7, 4) Hamming code

$$\mathbf{G}=\begin{bmatrix}1101000\\0110100\\1110010\\ \underbrace{101}_{\mathrm{P}}0001\end{bmatrix}\Rightarrow\mathbf{H}_{3\times7}=\begin{bmatrix}100\vdots1011\\010\vdots1110\\001\vdots\underbrace{0111}_{\mathbf{P}^T}\end{bmatrix}\Rightarrow\begin{cases}c_0=u_0+u_2+u_3\\c_1=u_0+u_1+u_2\\c_2=u_1+u_2+u_3\end{cases}$$

  Thus, block code $\mathcal{C}=\left\{\mathbf{c}\in GF(q)^n\,\middle|\,\mathbf{c}\mathbf{H}^T=\mathbf{0}\right\}$.

***3. Syndrome decoding***

- Error vector (or error pattern): Let **c** be the transmitted codeword, and **r** be the received word. Then the difference between **r** and **c** gives the pattern of errors: $\mathbf{e} = \mathbf{r} - \mathbf{c}$ (for binary codes, $\mathbf{e} = \mathbf{r} \oplus \mathbf{c}$ )

  $e_j$=1 indicates that the *j*-th position of **r** has an error.
- Obviously, $\mathbf{r} = \mathbf{c} + \mathbf{e}$.
- There are in total $2^n$ possible error patterns. Among them, only $2^{n-k}$ patterns are correctable by an (*n*, *k*) linear code.
- To test whether a received vector **r** contains errors, we compute the following (*n*-*k*)-tuple:

$$\mathbf{s} = \left(s_0, s_1, \cdots, s_{n-k-1}\right) \triangleq \mathbf{r} \cdot \mathbf{H}^T$$

$$= (\mathbf{c} + \mathbf{e}) \cdot \mathbf{H}^T$$

$$= \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$$

问题：$\mathbf{s} \Rightarrow \mathbf{e} = ?$

If $\mathbf{s} \neq \mathbf{0} \Rightarrow \mathbf{e} \neq \mathbf{0}$

If $\mathbf{s} = \mathbf{0} \Rightarrow$ 无错，$\mathbf{e} = \mathbf{0}$； 或错误不可检：$\mathbf{e} \in \mathcal{C}$.

- The (*n*-*k*)-tuple, **s** is called the syndrome of **r**.
  Any method solving these n-k equations is a decoding method.
- 最小距离译码就是找重量最轻的 **e** such that $\mathbf{e}\mathbf{H}^T = \mathbf{r}\mathbf{H}^T = \mathbf{s}$
- Syndrome decoding consists of these steps:
  1) Calculate syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T$ of received *n*-tuple.
  2) Find 最可能的错误图样 **e** with $\mathbf{e}\mathbf{H}^T = \mathbf{s}$ ---> 非线性运算
  3) 估计发送码字 $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.
  4) Determine message $\hat{\mathbf{u}}$ from the encoding equation $\hat{\mathbf{c}} = \hat{\mathbf{u}}\mathbf{G}$.
- <u>Example</u>: (7, 4) Hamming code. Suppose **c** = (1001011) is transmitted and **r**=(1001001) is received. Then $\mathbf{s} = (s_0, s_1, s_2) = \mathbf{r}\mathbf{H}^T = (111)$

Let $\mathbf{e} = (e_0, e_1, ..., e_6)$ be the error pattern. Since $\mathbf{s} = \mathbf{e}\mathbf{H}^T$, we have the following 3 equations :

$$1=e_0+e_3+e_5+e_6$$
$$1=e_1+e_3+e_4+e_5$$
$$1=e_2+e_4+e_5+e_6$$

There are 16 possible solutions, 其中 **e**=(0000010)是重量最小，是最可能发生的错误图样，故 $\hat{\mathbf{c}} = \mathbf{r} \oplus \mathbf{e}$ =(10010010)⊕(0000010)=(1001011).

- **Standard array**

| | | | | |
|---|---|---|---|---|
| $\mathbf{c}_1=\mathbf{0}$ | $\mathbf{c}_2$ | $\mathbf{c}_3$ | … | $\mathbf{c}_M$ |
| $\mathbf{e}_2$ | $\mathbf{e}_2+\mathbf{c}_2$ | $\mathbf{e}_2+\mathbf{c}_3$ | … | $\mathbf{e}_2+\mathbf{c}_M$ |
| $\mathbf{e}_3$ | $\mathbf{e}_3+\mathbf{c}_2$ | $\mathbf{e}_3+\mathbf{c}_3$ | … | $\mathbf{e}_3+\mathbf{c}_M$ |
| … | … | … | … | … |
| $\mathbf{e}_{2^r}$ | $\mathbf{e}_{2^r}+\mathbf{c}_2$ | $\mathbf{e}_{2^r}+\mathbf{c}_3$ | … | $\mathbf{e}_{2^r}+\mathbf{c}_M$ |

$M=2^k$, $r = n-k$

- Each row is called a ***coset***.

### *4. Hamming codes*

- First class of codes devised for error correction.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters:

code length: $n = 2^m - 1$

dimension: $k = 2^m - m - 1$

Number of parity-check symbols: $n$-$k$=$m$
Error correcting capability: $t$=1
Minimum distance: $d_{\min}$=3

### *5. Hamming bound (sphere packing)*

For a ($n$, $k$) linear block code over GF($q$) with error correction ability $t$,

$$n-k \geq \log_q V_q(n,t) = \log_q \left( \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \right)$$

其中$V_q(n,t)=\sum_{i=0}^{t}\binom{n}{i}(q-1)^i$是半径为 t 的 Hamming 球的 volume.

**证明：**在 GF($q$)上的 $n$ 维空间中，总共有$q^n$个 n 维向量，有$M=q^k$个码字（Hamming 球），因此有

$$q^k \cdot V_q(n,t) \leq q^n$$

从而

$$\frac{q^n}{q^k} \geq V_q(n,t)$$

$$n-k \geq \log_q V_q(n,t)$$

- Note: Hamming codes are one of few perfect codes – achieving Hamming bound.