8.6. Bound of length of the codeword

- Bound of length of N-symbol codeword used to determine the minimum codeword length for detection/correction
- Bound of length of N-symbol codeword for detection:
  - When transmitting a N-symbol codeword through channel with t-error, the number of errors would be:

    N_{1E}= $\sum_{i=1}^{t} C_N^i (r-1)^i$
    - The channel with t-errors, it means the codeword may has from 1 to t error-positions
    - When the codeword has i error-positions, number of received error combinations will be C_N^i
      - $C_N^i = \frac{N!}{(N-i!)i!}$
    - Each error-position has (r-1) ways of errors
  - To detect the error, it needs to have enough number of “don’t care” combination B_N

    $B_N \geq N_{1E}$ → $\boldsymbol{r^N}$**-** $\boldsymbol{r^L}$ **≥** $\boldsymbol{\sum_{i=1}^{t} C_N^i (r-1)^i}$ **(8.3)**
    - (8.3) is bound of length of N-symbol codeword of detection code
    - If r=2: → r^N- r^L ≥ $\sum_{i=1}^{t} C_N^i$
    - If r=2, t=1 → r^N- r^L ≥ C_N^1 → N ≥ L +1 → only need to add one symbol to the binary message to detect 1-error

8.6. Bound of length of the codeword (cont.)

- Bound of length of N-symbol codeword for correction:
  - With correction code, received combination must be separated → error combinations are separated → number of the error combinations:

    $N_E = r^L \times N_{1E}$

    Where r^L is number of codewords
  - To correct the error, it needs to have enough number of "don't care" combination B_N

    $B_N \geq N_E \rightarrow r^N - r^L \geq r^L \sum_{i=1}^{t} C_N^i (r-1)^i$

    $\rightarrow r^{N-L} - 1 \geq \sum_{i=1}^{t} C_N^i (r-1)^i$

    $r^{N-L} \geq \sum_{i=0}^{t} C_N^i (r-1)^i$

  logarithm with base r:

  N-L $\boldsymbol{\geq log_r(\sum_{i=0}^{t} C_N^i (r-1)^i)}$ **(8.4)**

    - (8.4) is bound of length of N-symbol codeword of correction code
    - If r = 2 → N-L $\geq log_r(\sum_{i=0}^{t} C_N^i)$
    - If r =2, t =1 → N-L $\geq log_2(C_N^0 + C_N^1) = log_2$ (1 +N)
      - E.g: L=4 then N ≥ 7

8.7. Detection/correction code construction

- Detection code construction:
  - Given L,t,r
  - Step 1: use (8.3) to calculate the length of codeword. Choose Nmin
  - Step 2: Choose N-symbol combination of 0 as first codeword. Continue find (r^L-1) N-symbol combinations as codewords so that minimum distance of code d satisfies (8.1)

- Correction code construction:
  - Given L,t,r
  - Step 1: use (8.4) to calculate the length of codeword. Choose Nmin
  - Step 2: Choose N-symbol combination of 0 as first codeword. Continue find (r^L-1) N-symbol combinations as codewords so that minimum distance of code d satisfies (8.2)

8.8. Parity code

- Binary code may detect 1-error
- Apply (8.3), the length of parity codeword N is length of message L plus 1
- To assure that d(K_N) $\geq$ 2, the added symbol must be:
  - If message has an even number of positions whose value is 1, added symbol =0
  - If message has an odd number of positions whose value is 1, added symbol =1

  → All codewords has even number of positions whose value is 1 (even codeword)
- To verify a binary combination is even or not,

  $\mathrm{P} = \ XOR_{j=1}^{L} m_{ij}$ where $m_{ij}\ is\ j^{th}$ symbol in message m_i
  - If P = 0 : even, P=1 : odd
  - P called parity bit (PB)

8.8. Parity code (cont.)

- Encoding algorithm:
  - Calculate P of message
  - Codeword is message m_i plus P
- Decoding algorithm:
  - Calculate the syndrome S (sign to detect error, S ≤ 0: no error, S > 0 : error)
    - S = $XOR_{j=1}^{L} b_j$ where $b_j\ is\ j^{th}$ symbol of received word b
      - S = 0: No error
      - S = 1: Error

8.8. Parity code (cont.)

- Example:
  - Set of message {00,01,10,11}. L = 2
    - 00, 11: even message → P = 0
    - 10,01: odd message→ P = 1
  - Code (set of codewords) will be

    000,110,101,011
  - If received word 010 then s = 1 →error

8.9. Hamming code

- Linear binary block code proposed by R. Hamming
- Can correct 1-error
- Have largest length:
  - According to (8.4) N-L $\geq log_r(\sum_{i=0}^{t} C_N^i (r-1)^i)$
    - r = 2, t = 1 → N-L $\geq log_2$ (1 +N) → 2^{N-L} ≥ 1 + N → N $\leq 2^{R_N}$ - 1
      - Nmax = 2^{R_N} - 1
- Hamming code uses linear space to represent code
  - Code that uses linear space called linear code

8.9. Hamming code (cont.)

- Linear space
  - A vector space over a field F is a set V together with two operations that satisfy the eight axioms listed below.
    - The first operation, called vector addition or simply addition +
      - $u, v \in V \rightarrow w = u + v \in V$
    - The second operation, called scalar multiplication .
      - $u \in F, v \in V \rightarrow w = u . v \in V$

8.9. Hamming code (cont.)

- Linear space
  - Axioms:
    - Associativity of addition $u + (v + w) = (u + v) + w$
    - Commutativity of addition $u + v = v + u$
    - Identity element of addition There exists an element $0 \in V$, called the zero vector, such that $v + 0 = v$ for all $v \in V$.
    - Inverse elements of addition For every $v \in V$, there exists an element $-v \in V$, called the additive inverse of v, such that $v + (-v) = 0$.
    - Compatibility of scalar multiplication with field multiplication $a(bv) = (ab)v$
    - Identity element of scalar multiplication $1v = v$, where 1 denotes the multiplicative identity in F.
    - Distributivity of scalar multiplication with respect to vector addition $a(u + v) = au + av$
    - Distributivity of scalar multiplication with respect to field addition $(a + b)v = av + bv$

8.9. Hamming code (cont.)

- Linear space
  - If the element of *V* is N-dimension vector then *V* is called N-dimension vector space
    - a ∈ V then a = a_1, a_2,.., a_N
    - a_i has discrete values from 0 to r-1 → discrete space with base *r*
  - Generator matrix
    - Set of *N* independent elements of *V* called set of base elements
      - Base elements are denoted by g_1, g_2,.., g_N
    - Set of base elements can generate all elements of *V*
    - Arrange each N-dimension element in one row → N x N matrix whose rows are independent.
      - This matrix is called generator matrix (G)
    - a ∈ *V* if and only if a = C .G → a = $\sum_{i=1}^{N} c_i g_i$ → a = a_1, a_2,.., a_N
      - *C* is coefficient vector
      - In discrete space with base r: value of c_i is 0/1/…/r-1
      - *C* has r^N values
      - a = C .G can generate all N-dimension elements of space
    - If G is unit matrix
      - *G* is in canonical form

8.9. Hamming code (cont.)

- Linear space
  - L-dimension subspace (L < N) is a subspace of N-dimension space.
    - Each element of L are N-dimension elements
    - Has maximum L independent elements
      - Can be considered as set of base elements of subspace
    - Generator matrix has L rows, N columns ($G_{L,N}$)
    - One element a ∈ L-dimension subspace if and only if a = C$G_{L,N}$ while C= c_1, c_2,.., c_L
    - Number elements of subspace is r^L
    - $G_{L,N}$ is in canonical form when its first (L x L) submatrix if unit matrix
      - Code generated by $G_{L,N}$ is called systematic code
      - L first symbols are carrying information symbols, remaining symbols are checked symbols
  - N-L dimension subspace that is orthogonal with L-dimension subspace :
    - its elements are orthogonal with L-dimension subspace
    - Called orthogonal space
    - Generator matrix has (N-L) row, N columns ($H_{N-L,N}$)
      - $G_{L,N}(H_{N-L,N})^T$ = 0
      - a ∈ $G_{L,N}$ if and only if a$(H_{N-L,N})^T$ = 0
      - $H_{N-L,N}$ is called "check parity matrix"
    - $H_{N-L,N}$ is in canonical form when its first ((N-L) x (N-L)) submatrix is unit matrix

8.9. Hamming code (cont.)

- Linear code:
  - One codeword of linear code is mapped to one element of L-dimension subspace
  - Other elements of N-dimension space which don't belong to L-dimension subspace is "don't care combination"
  - With linear code: if *a* is codeword then *a* is generated by *a = CG*

    or ***a* satisfies $\boldsymbol{aH^T}$=0**
  - To simplify $G_{L,N}$ is denoted by G, $H_{N-L,N}$ is denoted by H
  - To encode: calculate a = CG (C is message, G is generator matrix) or **calculate *a* from $\boldsymbol{aH^T}$=0 and message C is the given parameter of $\boldsymbol{aH^T}$=0**
  - To decode: when receive b, calculate syndrome S = bH^T
    - S = 0: no error
    - S > 0 : error
    - Since b = a + e where e = {e_1, e_2,…, e_N} is "error combination", S = (a+e)H^T= aH^T+ eH^T = eH^T

      → e can be calculated using S

8.9. Hamming code (cont.)

- Hamming code:
  - To build Hamming code or to decode a codeword of Hamming code, Hamming uses only "check parity matrix" H
  - Hamming proposes: each column of check parity matrix is a (N-L) binary number
    - The value of binary number = order number of column
  - Hamming code is binary code that can correct 1-error
  - Length of Hamming code N = 2^{R_N} - 1
  - To build: Solve aH^T=0 to determine codeword a
    - If a=$a_1 a_2 \dots a_N$ is codeword needed to be built then aH^T=0
    - aH^T=0 is matrix equation which generates system of (N-L) first-order equations
      - $a_i h_i^T$ = 0 when h_i is the i^{th} row of matrix H
      - Systems of equations can only determine (N-L) a_i, other L symbol a_i of a will be given parameters
        - Given parameters are L-symbol message
        - a_i are given parameters
          - Its position corresponds with column order of matrix H
            - The column has only one symbols its value = 1 to solve easier the equations

8.9. Hamming code (cont.)

- Hamming code:
  - To decode:
    - Let b is received combination, need to calculate syndrome S = $\mathrm{b}H^T$
    - If S = 0 → no error
    - If S ≠ 0 → S = eH^T = H_i^T where H_i^T is i^{th} row of H^T

      = H_i where H_i is i^{th} column of matrix H
    - H_i is the (N-L)-dimension binary combination that has value *i*

    → Syndrome is the (N-L)-dimension binary combination that has value *i*

    → Syndrome indicates wrong position

8.9. Hamming code (cont.)

- Example
  - L = 4, t = 1, r =2
  - Let message m = $\{m_1, m_2, m_3, m_4\}$
  - N is calculated by N = 2^{R_N} - 1 → N = 7
  - Check matrix (check parity matrix):
    - H= $\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$
  - Position 1,2,4 of matrix H has only one position that has value = 1

→a = (x,y, m_1,z, m_2, m_3, m_4) vị trí thêm vào tương ứng

→then aH^T = {z+ $m_2 + m_3 + m_4$, y+ $m_1 + m_3 + m_4$,x+ $m_1 + m_2 + m_4$} = {0,0,0}

8.9. Hamming code (cont.)

- x = $m_1 + m_2 + m_4$
- y = $m_1 + m_3 + m_4$
- z= $m_1 + m_2 + m_3$

→a = $\{m_1 + m_2 + m_4, m_1 + m_3 + m_4, m_1, m_1 + m_2 + m_3, m_2, m_3, m_4\}$

- If input message is 0000 → codeword 0000000
- If input message is 0100 → codeword 1001100
- If input message is 1111 →codeword 1111111

8.9. Hamming code (cont.)

- To decode: calculate syndrome S = $bH^T = eH^T = H_i$ = (N-L) dimension binary combination has value *i*

→ Detect and correct error

- If codeword is 1100110 and the error is in third bit, giving 11*1*0110
  - Syndrome is $\boxed{1110110\ H^T}$ = 011 (indicate bit error is the third bit)

only correct 1 position