Chapter 3

Linear Block Codes

Institute of Communications Engineering

Outlines

- Introduction to linear block codes
- Syndrome and error detection
- The minimum distance of a block code
- Error-detecting and error-correcting capabilities of a block code
- Standard array and syndrome decoding
- Probability of an undetected error for linear codes over a binary symmetric channel (BSC).
- Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

Introduction to Linear Block Codes

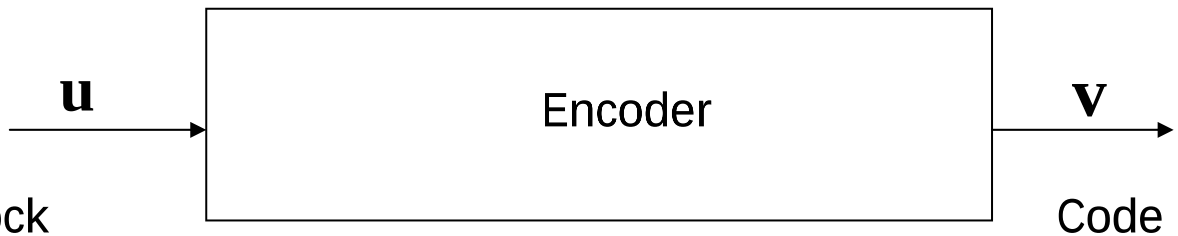
Institute of Communications Engineering

Introduction to Linear Block Codes

- We assume that the output of an information source is a sequence of binary digits "0" or "1"
- This binary information sequence is segmented into *message* block of fixed length, denoted by **u.**
  - Each message block consists of k information digits.
  - There are a total of 2^k distinct message.
- The encoder transforms each input message **u** into a binary n-tuple **v** with $n > k$
  - This n-tuple **v** is referred to as the *code word* (or *code vector*) of the message **u**.
  - There are distinct 2^k code words.

Introduction to Linear Block Codes

- This set of 2^k code words is called a *block* code.
- For a *block* code to be useful, there should be a one-to-one correspondence between a message **u** and its code word **v.**
- A desirable structure for a block code to possess is the linearity. With this structure, the encoding complexity will be greatly reduced.



Message block

(k info. digits)

(2^k distinct message)

Code word

(n-tuple , $n > k$)

(2^k distinct code word)

Introduction to Linear Block Codes

- **Definition 3.1.** A block code of length n and 2^k code word is called a *linear* (n, k) code iff its 2^k code words form a <u>k-dimensional subspace</u> of the vector space of all the n-tuple over the field GF(2).

- In fact, a binary block code is linear iff the module-2 sum of two code word is also a code word
  - **0** must be code word.
  - The block code given in Table 3.1 is a (7, 4) linear code.

Introduction to Linear Block Codes

TABLE 3.1 LINEAR BLOCK CODE WITH $k = 4$ AND $n = 7$

Messages	Code words
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 0 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

For large k, it is virtually impossible to build up the loop up table.

Introduction to Linear Block Codes

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space V_n of all the binary n-tuple, it is possible to find k linearly independent code word, $\mathbf{g}_0$, $\mathbf{g}_1$,…, $\mathbf{g}_{k-1}$ in C

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1} \quad (3.1)$$

where $u_i = 0$ or 1 for $0 \leq i < k$

Introduction to Linear Block Codes

- Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdot & \cdot & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdot & \cdot & g_{1,n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdot & \cdot & g_{k-1,n-1} \end{bmatrix} \tag{3.2}$$

where $\mathbf{g}_i = (g_{i0}, g_{i1}, \ldots, g_{i,n-1})$ for $0 \le i < k$

Introduction to Linear Block Codes

- If $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded, the corresponding code word can be given as follows:

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$$

$$= (u_0, u_1, \ldots, u_{k-1}) \bullet \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} \qquad (3.3)$$

$$= u_0 \mathbf{g}_0 + u_1 \mathbf{g}_1 + \cdots + u_{k-1} \mathbf{g}_{k-1}$$

Introduction to Linear Block Codes

- Because the rows of **G** generate the (n, k) linear code C, the matrix **G** is called a *generator matrix* for C
- Note that any k linearly independent code words of an (n, k) linear code can be used to form a generator matrix for the code
- It follows from (3.3) that an (n, k) linear code is completely specified by the k rows of a generator matrix **G**

Introduction to Linear Block Codes

Example 3.1

- the (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix :

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

$$\begin{aligned} \mathbf{v} &= 1 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1 + 0 \cdot \mathbf{g}_2 + 1 \cdot \mathbf{g}_3 \\ &= (1\,1\,0\,1\,0\,0\,0) + (0\,1\,1\,0\,1\,0\,0) + (1\,0\,1\,0\,0\,0\,1) \\ &= (0\,0\,0\,1\,1\,0\,1) \end{aligned}$$

Introduction to Linear Block Codes

- A desirable property for a linear block code is the *systematic structure* of the code words as shown in Fig. 3.1
  - where a code word is divided into two parts
    - The message part consists of k information digits
    - The redundant checking part consists of $n - k$ parity-check digits
- A linear block code with this structure is referred to as a *linear systematic block code*

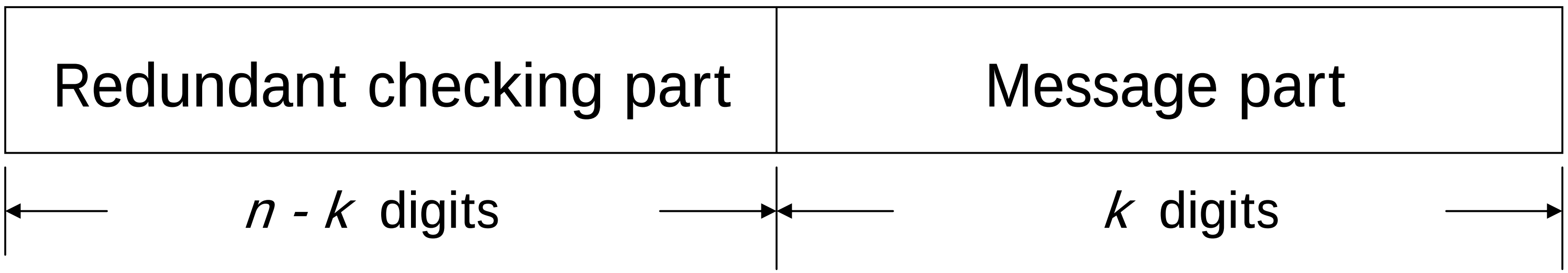


Fig. 3.1 Systematic format of a code word

Introduction to Linear Block Codes

- A linear systematic (n, k) code is completely specified by a $k \times n$ matrix G of the following form :

$$\begin{array}{c} \quad\quad\quad\quad \longleftarrow \text{P matrix} \longrightarrow \quad \longleftarrow k \times k \text{ identity matrix} \longrightarrow \end{array}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ . \\ . \\ . \\ \mathbf{g}_{k-1} \end{bmatrix} = \left[\begin{array}{cccccc|ccccccc} p_{00} & p_{01} & . & . & . & p_{0,n-k-1} & 1 & 0 & 0 & . & . & . & 0 \\ p_{10} & p_{11} & . & . & . & p_{1,n-k-1} & 0 & 1 & 0 & . & . & . & 0 \\ p_{20} & p_{21} & . & . & . & p_{2,n-k-1} & 0 & 0 & 1 & . & . & . & 0 \\ & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ p_{k-1,0} & p_{k-1,1} & . & . & . & p_{k-1,\,n-k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{3.4}$$

where $p_{ij} = 0$ or 1

Introduction to Linear Block Codes

- Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k\text{-}1})$ be the message to be encoded
- The corresponding code word is

$$v = (v_0, v_1, v_2, \ldots, v_{n-1})$$
$$= (u_0, u_1, \ldots, u_{k\text{-}1}) \cdot G \qquad (3.5)$$

- It follows from (3.4) & (3.5) that the components of **v** are

$$v_{n\text{-}k+i} = u_i \qquad \text{for } 0 \leq i < k \qquad (3.6a)$$

and

$$v_j = u_0\, p_{0j} + u_1\, p_{1j} + \cdots + u_{k\text{-}1}\, p_{k\text{-}1,\, j} \qquad \text{for } 0 \leq j < n\text{-}k \qquad (3.6b)$$

Introduction to Linear Block Codes

- Equation (3.6a) shows that the rightmost k digits of a code word **v** are identical to the information digits $u_0, u_1, \ldots u_{k-1}$ to be encoded

- Equation (3.6b) shown that the leftmost $n - k$ redundent digits are linear sums of the information digits

- The $n - k$ equations given by (3.6b) are called *parity-check equations* of the code

Example 3.2

- The matrix **G** given in example 3.1
- Let **u** = (u_0, u_1, u_2, u_3) be the message to be encoded
- Let **v** = $(v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word
- Solution :

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

Introduction to Linear Block Codes

- By matrix multiplication, we obtain the following digits of the code word **v**

$$
\begin{aligned}
v_6 &= u_3 \\
v_5 &= u_2 \\
v_4 &= u_1 \\
v_3 &= u_0 \\
v_2 &= u_1 + u_2 + u_3 \\
v_1 &= u_0 + u_1 + u_2 \\
v_0 &= u_0 + u_2 + u_3
\end{aligned}
$$

The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1)

Introduction to Linear Block Codes

- For any $k \times n$ matrix **G** with k linearly independent rows, there exists an $(n\text{-}k) \times n$ matrix **H** with $n\text{-}k$ linearly independent rows such that any vector in the row space of **G** is orthogonal to the rows of **H** and any vector that is orthogonal to the rows of **H** is in the row space of **G**.
- An n-tuple **v** is a code word in the code generated by **G** if and only if $\mathbf{v} \bullet \mathrm{H}^{\mathrm{T}} = 0$
- This matrix **H** is called a *parity-check matrix* of the code
- The 2^{n-k} linear combinations of the rows of matrix **H** form an $(n, n-k)$ linear code C_{d}
- This code is the null space of the (n, k) linear code C generated by matrix **G**
- $\mathrm{C_d}$ is called the *dual code* of C

Introduction to Linear Block Codes

- If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form :

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{n-k} & \mathbf{P}^T \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & . & . & . & 0 & p_{00} & p_{10} & . & . & . & p_{k\text{-}1,0} \\ 0 & 1 & 0 & . & . & . & 0 & p_{01} & p_{11} & . & . & . & p_{k\text{-}1,1} \\ 0 & 0 & 1 & . & . & . & 0 & p_{02} & p_{12} & . & . & . & p_{k\text{-}1,2} \\ . & & & & & & & & & & & & \\ . & & & & & & & & & & & & \\ . & & & & & & & & & & & & \\ 0 & 0 & 0 & . & . & . & 1 & p_{0,n\text{-}k\text{-}1} & p_{1,n\text{-}k\text{-}1} & . & . & . & p_{k\text{-}1,n\text{-}k\text{-}1} \end{bmatrix} \tag{3.7}$$

Introduction to Linear Block Codes

- Let h_j be the j_{th} row of $\mathbf{H}$

$$\mathbf{g}_i \cdot \mathbf{h}_j = p_{ij} + p_{ij} = 0$$

for $0 \le i < k$ and $0 \le j < n - k$

- This implies that $\mathbf{G} \bullet \mathbf{H}^{\mathrm{T}} = \mathbf{0}$

Introduction to Linear Block Codes

- Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded
- In systematic form the corresponding code word would be

$$\mathrm{v} = (v_0, v_1, \ldots, v_{n-k-1}, u_0, u_1, \ldots, u_{k-1})$$

- Using the fact that $\mathbf{v} \bullet \mathbf{H}^T = \mathbf{0}$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} = 0 \tag{3.8}$$

- Rearranging the equation of (3.8), we obtain the same parity-check equations of (3.6b)
- An (n, k) linear code is completely specified by its parity-check matrix

- **Example 3.3**
  - Consider the generator matrix of a (7,4) linear code given in example 3.1
  - The corresponding parity-check matrix is

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

Introduction to Linear Block Codes

- For any (n, k) linear block code $\boldsymbol{C}$, there exists a $k \times n$ matrix $\mathbf{G}$ whose row space given $\boldsymbol{C}$

- There exist an $(n - k) \times n$ matrix $\mathbf{H}$ such that an n-tuple $\mathbf{v}$ is a code word in $\boldsymbol{C}$ if and only if $\mathbf{v} \bullet \mathbf{H}^{\mathrm{T}} = 0$

- If $\mathbf{G}$ is of the form given by (3.4), then $\mathbf{H}$ may take form given by (3.7), and vice versa

Introduction to Linear Block Codes

- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily
- The encoding circuit is shown in Fig. 3.2
  - where
    - $\rightarrow\boxed{}\rightarrow$ denotes a shift-register stage (flip-flop)
    - $\rightarrow P_{ij} \rightarrow$ denotes a connection if $p_{ij} = 1$ and no connection if $p_{ij} = 0$
    - $\oplus$ denotes a modulo-2 adder
    - As soon as the entire message has entered the message register, the n-k parity-check digits are formed at the outputs of the n-k module-2 adders
  - The complexity of the encoding circuit is linear proportional to the block length
  - The encoding circuit for the (7,4) code given in Table 3.1 is shown in Fig 3.3

Introduction to Linear Block Codes

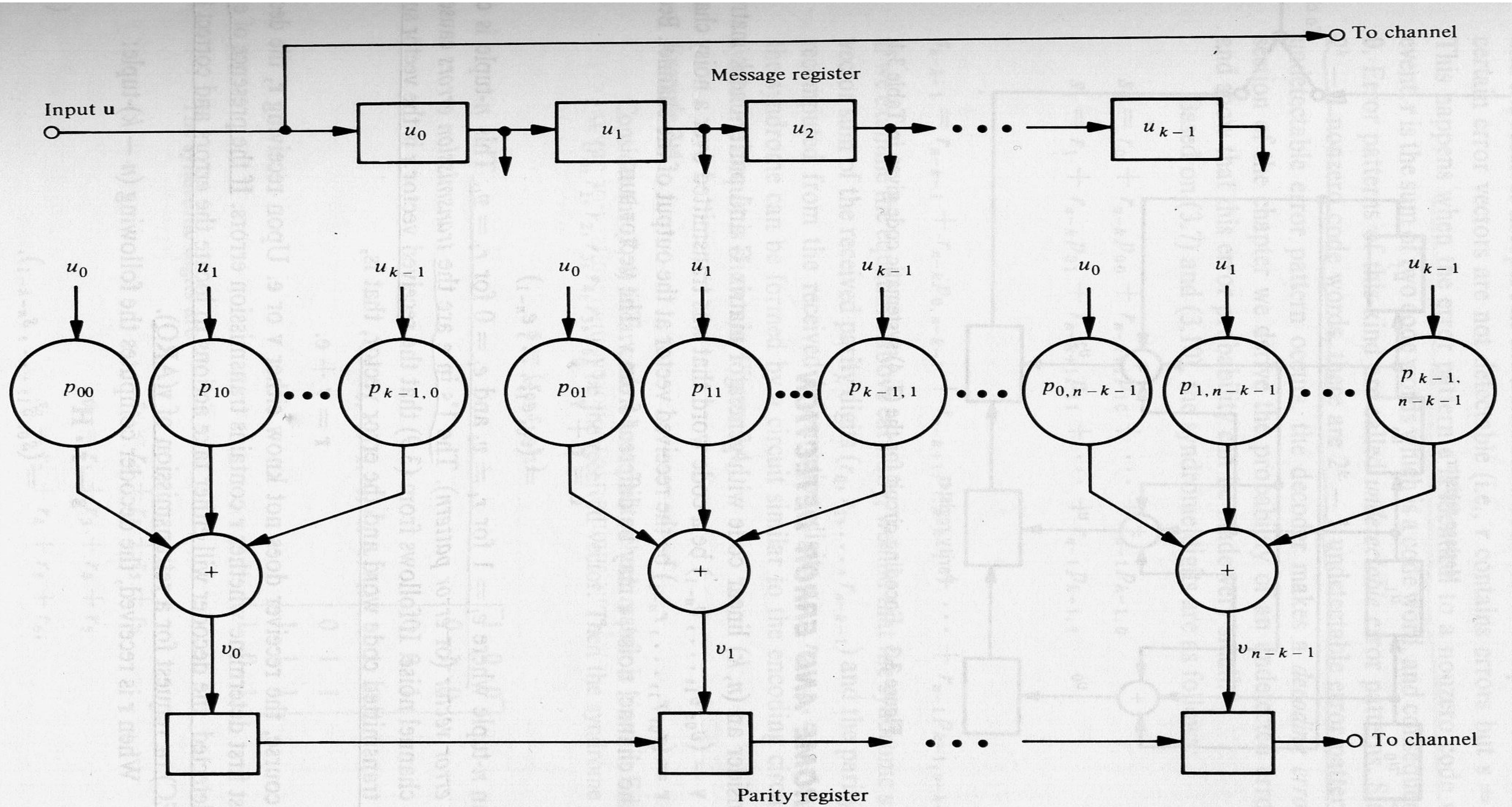


Figure 3.2 Encoding circuit for a linear systematic (n, k) code.

Introduction to Linear Block Codes

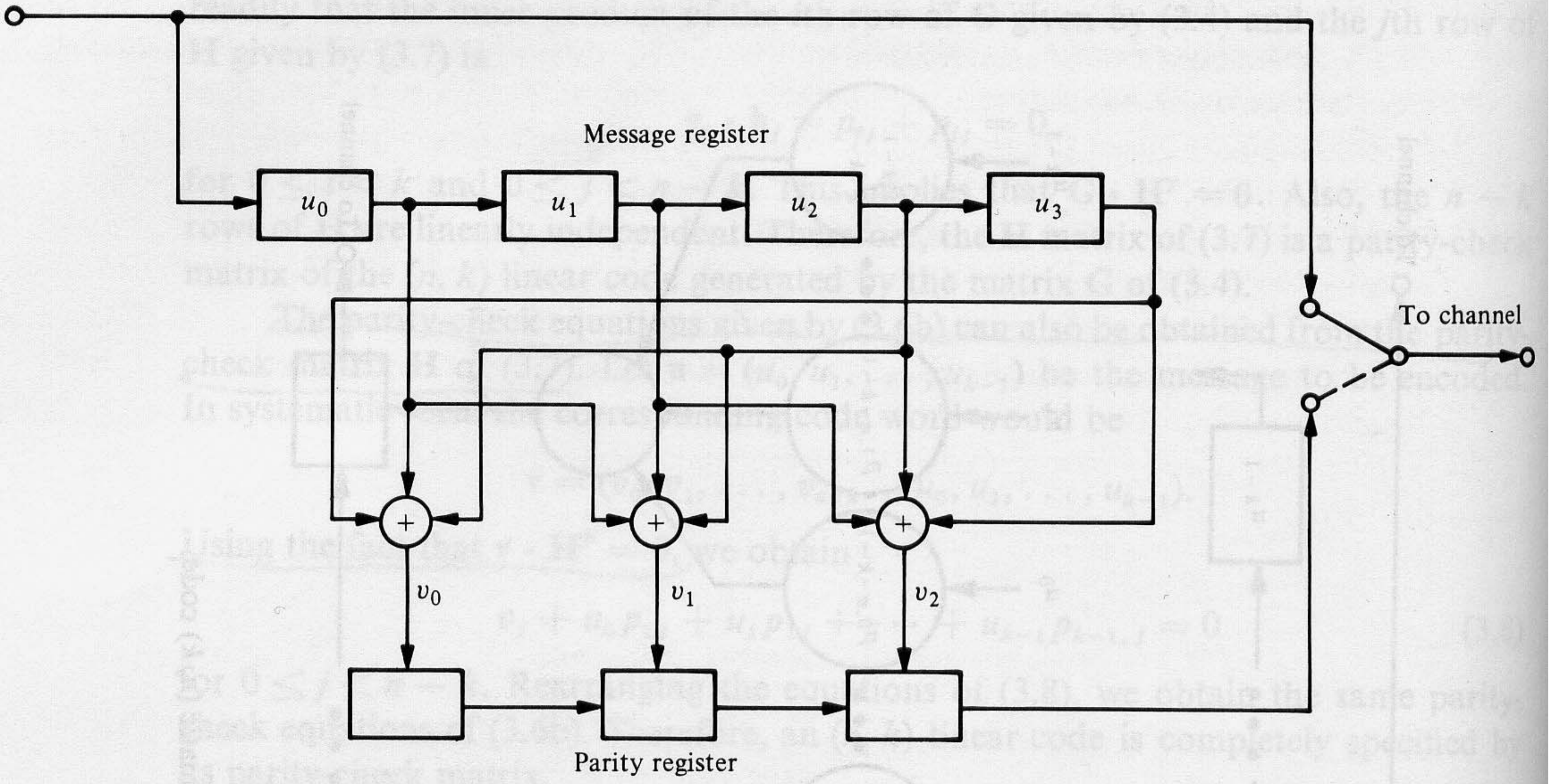
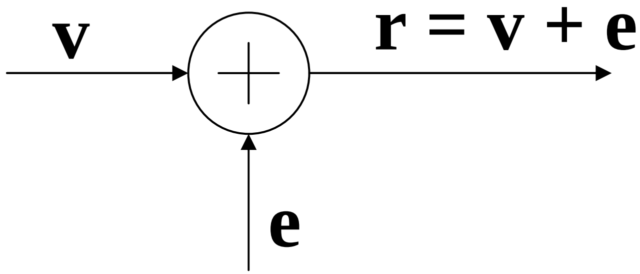


Figure 3.3 Encoding circuit for the (7, 4) systematic code given in Table 3.1.

Syndrome and Error Detection

Syndrome and Error Detection

- Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n\text{-}1})$ be a code word that was transmitted over a noisy channel
- Let $\mathbf{r} = (r_0, r_1, \ldots, r_{n\text{-}1})$ be the received vector at the output of the channel



- $\mathbf{e} = \mathbf{r} + \mathbf{v} = (e_0, e_1, \ldots, e_{n\text{-}1})$ is an n-tuple
  - $e_i = 1$ for $r_i \neq v_i$
  - $e_i = 0$ for $r_i = v_i$
  - The n-tuple $\mathbf{e}$ is called the *error vector* (or *error pattern*)

Syndrome and Error Detection

- Upon receiving **r**, the decoder must first determine whether **r** contains transmission errors
- If the presence of errors is detected, the decoder will take actions to locate the errors
  - Correct errors (FEC)
  - Request for a retransmission of **v** (ARQ)
- When **r** is received, the decoder computes the following $(n - k)$-tuple :

$$\mathbf{s} = \mathbf{r} \bullet \mathbf{H}^{T} = (s_0, s_1, \ldots, s_{n-k-1}) \tag{3.10}$$

which is called the *syndrome* of **r**

Syndrome and Error Detection

- $\mathbf{s} = \mathbf{0}$ if and only if $\mathbf{r}$ is a code word and receiver accepts $\mathbf{r}$ as the transmitted code word
- $\mathbf{s} \neq \mathbf{0}$ if and only if $\mathbf{r}$ is not a code word and the presence of errors has been detected
- When the error pattern $\mathbf{e}$ is identical to a nonzero code word (i.e., $\mathbf{r}$ contain errors but $\mathbf{s} = \mathbf{r} \bullet \mathbf{H}^T = \mathbf{0}$), error patterns of this kind are called *undetectable* error patterns
  - Since there are $2^k - 1$ nonzero code words, there are $2^k - 1$ undetectable error patterns

Syndrome and Error Detection

- Based on (3.7) and (3.10), the syndrome digits are as follows :

$$s_0 = r_0 + r_{n-k}\, p_{00} + r_{n-k+1}\, p_{10} + \cdots + r_{n-1}\, p_{k-1,0}$$

$$s_1 = r_1 + r_{n-k}\, p_{01} + r_{n-k+1}\, p_{11} + \cdots + r_{n-1}\, p_{k-1,1}$$

$$\vdots \qquad (3.11)$$

$$s_{n-k-1} = r_{n-k-1} + r_{n-k}\, p_{0,n-k-1} + r_{n-k+1}\, p_{1,n-k-1} + \cdots + r_{n-1}\, p_{k-1,n-k-1}$$

- The syndrome **s** is the vector sum of the received parity digits $(r_0,r_1,\ldots,r_{n-k-1})$ and the parity-check digits recomputed from the received information digits $(r_{n-k},r_{n-k+1},\ldots,r_{n-1})$.
- A general syndrome circuit is shown in Fig. 3.4

Syndrome and Error Detection

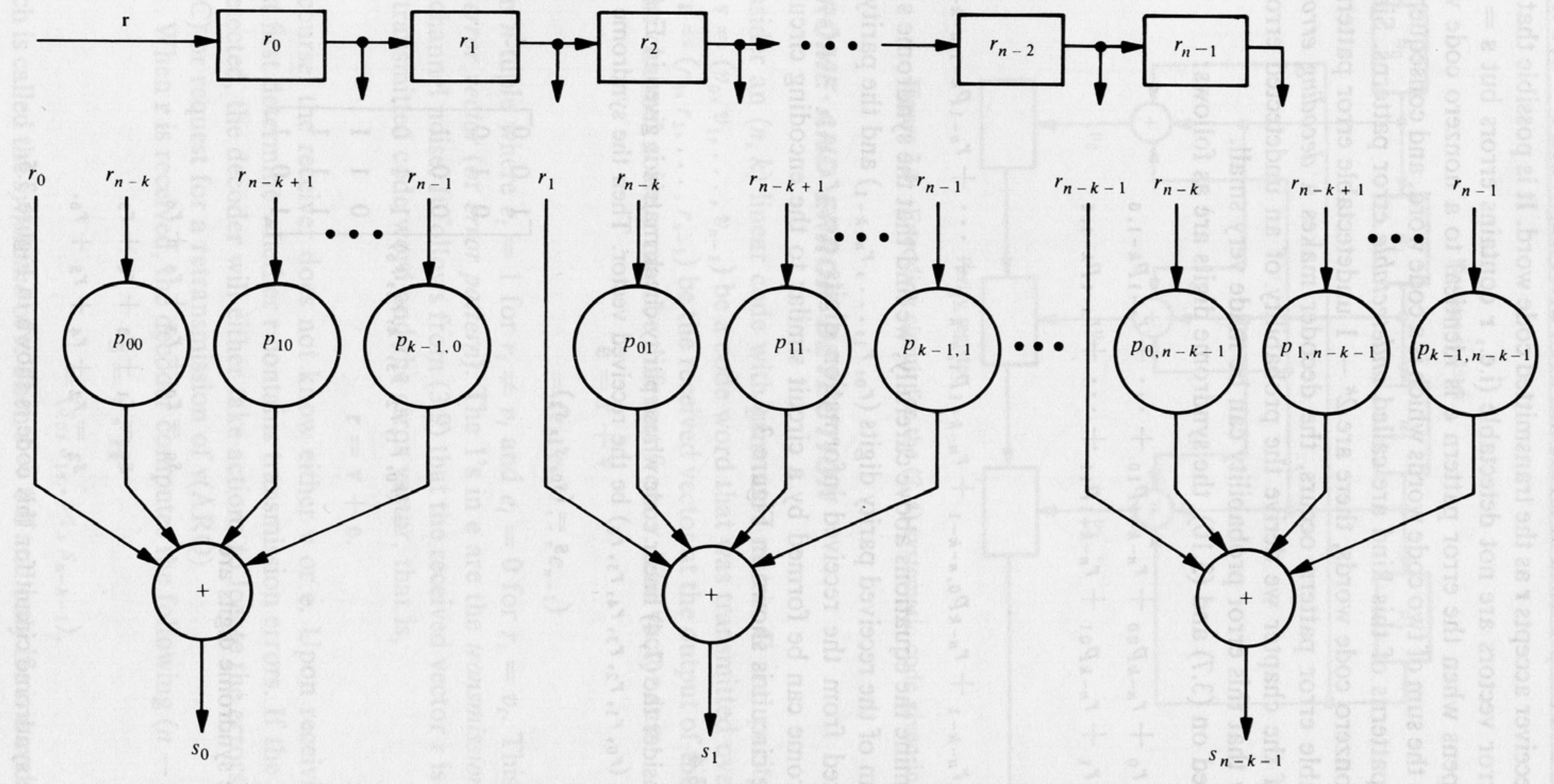


Figure 3.4 Syndrome circuit for a linear systematic (n, k) code.

Syndrome and Error Detection

- **Example 3.4**
  - The parity-check matrix is given in example 3.3
  - Let $\mathbf{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7)$ be the received vector
  - The syndrome is given by

$$\mathbf{s} = (s_0, s_1, s_2) = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Syndrome and Error Detection

- **Example 3.4**
  - The syndrome digits are

$$s_0 = r_0 + r_3 + r_5 + r_6$$
$$s_1 = r_1 + r_3 + r_4 + r_5$$
$$s_2 = r_2 + r_4 + r_5 + r_6$$

  - The syndrome circuit for this code is shown below

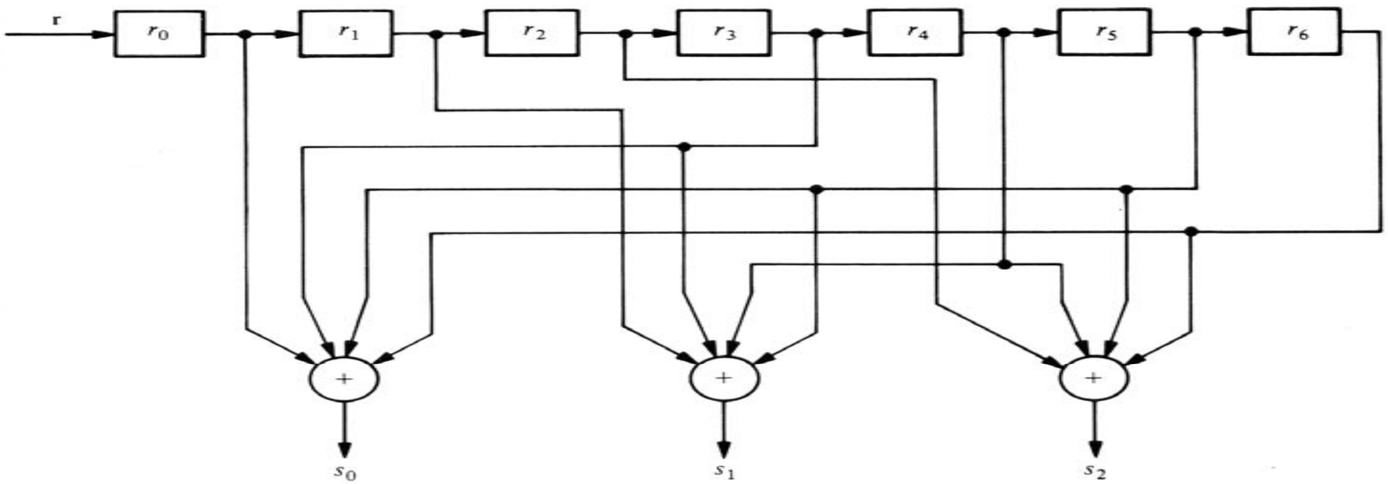


Figure 3.5 Syndrome circuit for the (7, 4) code given in Table 3.1.

Syndrome and Error Detection

- Since $\mathbf{r}$ is the vector sum of $\mathbf{v}$ and $\mathbf{e}$, it follows from (3.10) that

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^{\mathrm{T}} = (\mathbf{v} + \mathbf{e}) \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{v} \cdot \mathbf{H}^{\mathrm{T}} + \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}}$$

- however,

$$\mathbf{v} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{0}$$

- consequently, we obtain the following relation between the syndrome and the error pattern :

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}} \qquad (3.12)$$

Syndrome and Error Detection

- If the parity-check matrix **H** is expressed in the systematic form as given by (3.7), multiplying out $\mathbf{e} \cdot \mathbf{H}^T$ yield the following linear relationship between the syndrome digits and the error digits :

$$
\begin{aligned}
s_0 &= e_0 + e_{n-k}\, p_{00} + e_{n-k+1}\, p_{10} + \cdots + e_{n-1}\, p_{k-1,0} \\
s_1 &= e_1 + e_{n-k}\, p_{01} + e_{n-k+1}\, p_{11} + \cdots + e_{n-1}\, p_{k-1,1} \\
&\;\;. \\
&\;\;. \\
s_{n-k-1} &= e_{n-k-1} + e_{n-k}\, p_{0,n-k-1} + \cdots + e_{n-1}\, p_{k-1,n-k-1}
\end{aligned}
\qquad (3.13)
$$

Syndrome and Error Detection

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction
- Because the $n - k$ linear equations of (3.13) do not have a unique solution but have 2^k solutions
- There are 2^k error pattern that result in the same syndrome, and the true error pattern **e** is one of them
- The decoder has to determine the true error vector from a set of 2^k candidates
- To minimize the probability of a decoding error, the most *probable* error pattern that satisfies the equations of (3.13) is chosen as the true error vector

Syndrome and Error Detection

- **Example 3.5**
  - We consider the (7,4) code whose parity-check matrix is given in example 3.3
  - Let $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ be the transmitted code word
  - Let $\mathbf{r} = (1\ 0\ 0\ 1\ 0\ 0\ 1)$ be the received vector
  - The receiver computes the syndrome

$$\mathbf{s} = \mathbf{r} \bullet \mathbf{H}^T = (1\ 1\ 1)$$

  - The receiver attempts to determine the true error vector $\mathbf{e} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$, which yields the syndrome above

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

  - There are $2^4 = 16$ error patterns that satisfy the equations above

Syndrome and Error Detection

- **Example 3.5**
  - The error vector **e** = (0 0 0 0 0 1 0) has the smallest number of nonzero components
  - If the channel is a **BSC**, **e** = (0 0 0 0 0 1 0) is the most probable error vector that satisfies the equation above
  - Taking **e** = (0 0 0 0 0 1 0) as the true error vector, the receiver decodes the received vector **r** = (1 0 0 1 0 0 1) into the following code word

$$\begin{aligned}\mathbf{v}^* = \mathbf{r} + \mathbf{e} &= (1\ 0\ 0\ 1\ 0\ 0\ 1) + (0\ 0\ 0\ 0\ 0\ 1\ 0) \\ &= (1\ 0\ 0\ 1\ 0\ 1\ 1)\end{aligned}$$

  - where **v*** is the actual transmitted code word

The Minimum Distance of a Block Code

The Minimum Distance of a Block Code

- Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the *Hamming weight* (or simply weight) of $\mathbf{v}$, denote by $w(\mathbf{v})$, is defined as the number of nonzero components of $\mathbf{v}$
  - For example, the Hamming weight of $\mathbf{v} = (1\ 0\ 0\ 0\ 1\ 1\ 0)$ is 4
- Let $\mathbf{v}$ and $\mathbf{w}$ be two n-tuple, the *Hamming distance* between $\mathbf{v}$ and $\mathbf{w}$, denoted $d(\mathbf{v},\mathbf{w})$, is defined as the number of places where they differ
  - For example, the Hamming distance between $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ and $\mathbf{w} = (0\ 1\ 0\ 0\ 0\ 1\ 1)$ is 3

The Minimum Distance of a Block Code

- The Hamming distance is a metric function that satisfied the triangle inequality

$$d(\mathbf{v}, \mathbf{w}) + d(\mathbf{w}, \mathbf{x}) \geq d(\mathbf{v}, \mathbf{x}) \qquad (3.14)$$

  - the proof of this inequality is left as a problem
- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, $\mathbf{v}$ and $\mathbf{w}$, is equal to the Hamming weight of the sum of $\mathbf{v}$ and $\mathbf{w}$, that is

$$d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} + \mathbf{w}) \qquad (3.15)$$

  - For example, the Hamming distance between $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ and $\mathbf{w} = (1\ 1\ 1\ 0\ 0\ 1\ 0)$ is 4 and the weight of $\mathbf{v} + \mathbf{w} = (0\ 1\ 1\ 1\ 0\ 0\ 1)$ is also 4

The Minimum Distance of a Block Code

- Given, a block code C, the minimum distance of C, denoted $d_{\min}$, is defined as

$$d_{\min} = \min\{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \qquad (3.16)$$

- If C is a linear block, the sum of two vectors is also a code vector
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C

$$\begin{aligned} d_{\min} &= \min\{w(\mathbf{v} + \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\} \\ &= \min\{w(\mathbf{x}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\} \qquad (3.17) \\ &\equiv w_{\min} \end{aligned}$$

The Minimum Distance of a Block Code

- The parameter $w_{\min} \equiv \{w(\mathbf{x}): \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}$ is called the *minimum weight* of the linear code C
- **Theorem 3.1** The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words
- **Theorem 3.2** Let C be an (n, k) linear code with parity-check matrix $\mathbf{H}$.
  - For each code vector of Hamming weight l, there exist l columns of $\mathbf{H}$ such that the vector sum of these l columns is equal to the zero vector
  - Conversely, if there exist l columns of $\mathbf{H}$ whose vector sum is the zeros vector, there exists a code vector of Hamming weight l in C.

The Minimum Distance of a Block Code

- Proof
  - Let the parity-check matrix be
$$H = [\mathbf{h}_o, \mathbf{h}_1, \ldots , \mathbf{h}_{n\text{-}1}]$$
  - where $\mathbf{h}_i$ represents the ith column of **H**
  - Let $\mathbf{v} = (v_0, v_1, \ldots , v_{n\text{-}1})$ be a code vector of weight l and **v** has l nonzero components
  - Let $v_{i1}, v_{i2}, \ldots, v_{il}$ be the l nonzero components of **v**, where $0 \le i_1 < i_2 < \cdots < i_l \le n\text{-}1$, then $v_{i1} = v_{i2} = \cdots = v_{il} = 1$
  - since **v** is code vector, we must have

$$\begin{aligned}\mathbf{0} &= \mathbf{v} \bullet \mathbf{H}^{\mathrm{T}} \\ &= v_o\mathbf{h}_0 + v_1\mathbf{h}_1 + \cdots + v_{n\text{-}1}\mathbf{h}_{n\text{-}1} \\ &= v_{i1}\mathbf{h}_{i1} + v_{i2}\mathbf{h}_{i2} + \cdots + v_{il}\mathbf{h}_{il} \\ &= \mathbf{h}_{i1} + \mathbf{h}_{i2} + \cdots + \mathbf{h}_{il}\end{aligned}$$

The Minimum Distance of a Block Code

- Proof
  - Suppose that $\mathbf{h}_{i1}$, $\mathbf{h}_{i2}$, … , $\mathbf{h}_{il}$ are l columns of H such that

$$\mathbf{h}_{i1} + \mathbf{h}_{i2} + \cdots + \mathbf{h}_{il} = \mathbf{0} \tag{3.18}$$

  - Let $\mathbf{x} = (x_1, x_2, \ldots, x_{n\text{-}1})$ whose nonzero components are x_{i1}, x_{i2}, x_{il}

$$\begin{aligned}\mathbf{x} \bullet \mathbf{H}^{\mathrm{T}} &= x_0\mathbf{h}_0 + x_1\mathbf{h}_1 + \cdots + x_{n\text{-}1}\mathbf{h}_{n\text{-}1} \\ &= x_{i1}\mathbf{h}_{i1} + x_{i2}\mathbf{h}_{i2} + \cdots + x_{il}\mathbf{h}_{il} \\ &= \mathbf{h}_{i1} + \mathbf{h}_{i2} + \cdots + \mathbf{h}_{il}\end{aligned}$$

  - It following from (3.18) that $\mathbf{x} \bullet \mathbf{H}^{\mathrm{T}} = \mathbf{0}$, $\mathbf{x}$ is code vector of weight l in C

The Minimum Distance of a Block Code

- Let C be a linear block code with parity-check matrix **H**
- **Corollary 3.2.1** If no d-1 or fewer columns of **H** add to **0**, the code has minimum weight at least d
- **Corollary 3.2.2** The minimum weight of C is equal to the smallest number of columns of **H** that sum to **0**

Error-Detecting and Error-Correcting Capabilities of a Block Code

Error-Detecting and Error-Correcting Capabilities of a Block Code

- If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places
- A block code with minimum distance d_{min} is capable of detecting all the error pattern of $d_{min} - 1$ or fewer errors
- However, it cannot detect <u>all</u> the error pattern of d_{min} errors because there exists at least one pair of code vectors that differ in d_{min} places and there is an error pattern of d_{min} errors that will carry one into the other
- The random-error-detecting capability of a block code with minimum distance d_{min} is $d_{min} - 1$

Error-Detecting and Error-Correcting Capabilities of a Block Code

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n
- Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words
- If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word $\mathbf{v}$ into another code word $\mathbf{w}$, thus $\mathbf{w}$ will be received and its syndrome is zero
- There are $2^k - 1$ *undetectable* error patterns
- If an error pattern is not identical to a nonzero code word, the received vector $\mathbf{r}$ will not be a code word and the syndrome will not be zero

Error-Detecting and Error-Correcting Capabilities of a Block Code

- These $2^n - 2^k$ error patterns are *detectable* error patterns
- Let A_l be the number of code vectors of weight i in C, the numbers A_0, A_1,..., A_n are called the *weight distribution* of C
- Let $P_u(E)$ denote the probability of an undetected error
- Since an undetected error occurs only when the error pattern is identical to a nonzero code vector of C

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$$

- where p is the transition probability of the **BSC**
- If the minimum distance of C is d_{min}, then $\mathbf{A}_1$ to $\mathbf{A}_{d\min - 1}$ are zero

Error-Detecting and Error-Correcting Capabilities of a Block Code

- Assume that a block code C with minimum distance $d_{\min}$ is used for random-error correction. The minimum $d_{\min}$ distance is either odd or even. Let t be a positive integer such that:

$$2t+1 \le d_{\min} \le 2t+2$$

- Fact 1: The code C is capable of correcting all the error patterns of t or fewer errors.
- Proof:
  - Let $\mathbf{v}$ and $\mathbf{r}$ be the transmitted code vector and the received vector, respectively. Let $\mathbf{w}$ be any other code vector in C.

$$d(\mathbf{v},\mathbf{r}) + d(\mathbf{w},\mathbf{r}) \ge d(\mathbf{v},\mathbf{w})$$

  - Suppose that an error pattern of t' errors occurs during the transmission of $\mathbf{v}$. We have $d(\mathbf{v},\mathbf{r})=t'$.

Error-Detecting and Error-Correcting Capabilities of a Block Code

- Since **v** and **w** are code vectors in C, we have

$$d(\mathbf{v},\mathbf{w}) \geq d_{\min} \geq 2t+1.$$

$$\begin{aligned} d(\mathbf{w},\mathbf{r}) &\geq d(\mathbf{v},\mathbf{w}) - d(\mathbf{v},\mathbf{r}) \\ &\geq d_{\min} - t' \\ &\geq 2t+1-t' \\ &\geq t+1 > t \quad (\text{if } t \geq t') \end{aligned}$$

- The inequality above says that if an error pattern of t or fewer errors occurs, the received vector **r** is closer (in Hamming distance) to the transmitted code vector **v** than to any other code vector **w** in C.
- For a BSC, this means that the conditional probability $P(\mathbf{r}|\mathbf{v})$ is greater than the conditional probability $P(\mathbf{r}|\mathbf{w})$ for $\mathbf{w} \neq \mathbf{v}$. Q.E.D.

Error-Detecting and Error-Correcting Capabilities of a Block Code

- Fact 2: The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
  - Let $\mathbf{v}$ and $\mathbf{w}$ be two code vectors in C such that $d(\mathbf{v},\mathbf{w})=d_{\min}$.
  - Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be two error patterns that satisfy the following conditions:
    - $\mathbf{e}_1+\mathbf{e}_2=\mathbf{v}+\mathbf{w}$
    - $\mathbf{e}_1$ and $\mathbf{e}_2$ do not have nonzero components in common places.
  - We have $w(\mathbf{e}_1) + w(\mathbf{e}_2) = w(\mathbf{v}+\mathbf{w}) = d(\mathbf{v},\mathbf{w}) = d_{\min}$. (3.23)

Error-Detecting and Error-Correcting Capabilities of a Block Code

- Suppose that $\mathbf{v}$ is transmitted and is corrupted by the error pattern $\mathbf{e}_1$, then the received vector is

$$\mathbf{r} = \mathbf{v} + \mathbf{e}_1$$

- The Hamming distance between $\mathbf{v}$ and $\mathbf{r}$ is

$$d(\mathbf{v}, \mathbf{r}) = w(\mathbf{v} + \mathbf{r}) = w(\mathbf{e}_1). \quad (3.24)$$

- The Hamming distance between $\mathbf{w}$ and $\mathbf{r}$ is

$$d(\mathbf{w}, \mathbf{r}) = w(\mathbf{w} + \mathbf{r}) = w(\mathbf{w} + \mathbf{v} + \mathbf{e}_1) = w(\mathbf{e}_2) \quad (3.25)$$

- Now, suppose that the error pattern $\mathbf{e}_1$ contains more than t errors [i.e. $w(\mathbf{e}_1) \geq t+1$].
- Since $2t + 1 \leq d_{min} \leq 2t + 2$, it follows from (3.23) that

$$w(\mathbf{e}_2) = d_{min} - w(\mathbf{e}_1) \leq (2t + 2) - (t+1) = t + 1$$

Error-Detecting and Error-Correcting Capabilities of a Block Code

- Combining (3.24) and (3.25) and using the fact that $w(\mathbf{e}_1) \geq t+1$ and $w(\mathbf{e}_2) \leq t + 1$, we have

$$d(\mathbf{v}, \mathbf{r}) \geq d(\mathbf{w}, \mathbf{r})$$

- This inequality say that there exists an error pattern of l ($l > t$) errors which results in a received vector that is closer to an incorrect code vector than to the transmitted code vector.
- Based on the maximum likelihood decoding scheme, an incorrect decoding would be committed. Q.E.D.

Error-Detecting and Error-Correcting Capabilities of a Block Code

- A block code with minimum distance $d_{\min}$ guarantees correcting all the error patterns of $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or fewer errors, where $\lfloor (d_{\min} - 1)/2 \rfloor$ denotes the largest integer no greater than $(d_{\min} - 1)/2$
- The parameter $t = \lfloor (d_{\min} - 1)/2 \rfloor$ is called the *random-error correcting capability* of the code
- The code is referred to as a *t-error-correcting code*
- A block code with random-error-correcting capability t is usually capable of correcting many error patterns of $t + 1$ or more errors
- For a t-error-correcting (n, k) linear code, it is capable of correcting a total 2^{n-k} error patterns (shown in next section).

Error-Detecting and Error-Correcting Capabilities of a Block Code

- If a t-error-correcting block code is used strictly for error correction on a BSC with transition probability p, the probability that the decoder commits an erroneous decoding is upper bounded by:

$$P(E) \le \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

- In practice, a code is often used for correcting λ or fewer errors and simultaneously detecting l ($l > \lambda$) or fewer errors. That is, when λ or fewer errors occur, the code is capable of correcting them; when more than λ but fewer than l+1 errors occur, the code is capable of detecting their presence without making a decoding error.
- The minimum distance $d_{\min}$ of the code is at least $\lambda + l + 1$.

Standard Array and Syndrome Decoding

Standard Array and Syndrome Decoding

- Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{2^k}$ be the code vector of C
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \ldots, D_{2^k}$ such that the code vector $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$
- Each subset D_i is one-to-one correspondence to a code vector $\mathbf{v}_i$
- If the received vector $\mathbf{r}$ is found in the subset D_i, $\mathbf{r}$ is decoded into $\mathbf{v}_i$
- Correct decoding is made if and only if the received vector $\mathbf{r}$ is in the subset D_i that corresponds to the actual code vector transmitted

Standard Array and Syndrome Decoding

- A method to partition the 2^n possible received vectors into 2^k disjoint subsets such that each subset contains one and only one code vector is described here
  - First, the 2^k code vectors of C are placed in a row with the all-zero code vector $\mathbf{v}_1 = (0, 0, \ldots, 0)$ as the first (leftmost) element
  - From the remaining $2^n - 2^k$ n-tuple, an n-tuple $\mathbf{e}_2$ is chosen and is placed under the zero vector $\mathbf{v}_1$
  - Now, we form a second row by adding $\mathbf{e}_2$ to each code vector $\mathbf{v}_i$ in the first row and placing the sum $\mathbf{e}_2 + \mathbf{v}_i$ under $\mathbf{v}_i$
  - An unused n-tuple $\mathbf{e}_3$ is chosen from the remaining n-tuples and is placed under $\mathbf{e}_2$.
  - Then a third row is formed by adding $\mathbf{e}_3$ to each code vector $\mathbf{v}_i$ in the first row and placing $\mathbf{e}_3 + \mathbf{v}_i$ under $\mathbf{v}_i$.
  - we continue this process until all the n-tuples are used.

Standard Array and Syndrome Decoding

$$
\begin{array}{llllll}
\mathbf{0} & \mathbf{v}_2 & \cdots & \mathbf{v}_i & \cdots & \mathbf{v}_{2^k} \\
\mathbf{e}_2 & \mathbf{e}_2 + \mathbf{v}_2 & \cdots & \mathbf{e}_2 + \mathbf{v}_i & \cdots & \mathbf{e}_2 + \mathbf{v}_{2^k} \\
\mathbf{e}_3 & \mathbf{e}_3 + \mathbf{v}_2 & \cdots & \mathbf{e}_3 + \mathbf{v}_i & \cdots & \mathbf{e}_3 + \mathbf{v}_{2^k} \\
\vdots & & & & & \vdots \\
\mathbf{e}_l & \mathbf{e}_l + \mathbf{v}_2 & \cdots & \mathbf{e}_l + \mathbf{v}_i & \cdots & \mathbf{e}_l + \mathbf{v}_{2^k} \\
\vdots & & & & & \vdots \\
\mathbf{e}_{2^{n-k}} & \mathbf{e}_{2^{n-k}} + \mathbf{v}_2 & \cdots & \mathbf{e}_{2^{n-k}} + \mathbf{v}_i & \cdots & \mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}
\end{array}
$$

Figure 3.6 Standard array for an (n, k) linear code.

Standard Array and Syndrome Decoding

- Then we have an array of rows and columns as shown in Fig 3.6
- This array is called a *standard array* of the given linear code C
- **Theorem 3.3** No two n-tuples in the same row of a standard array are identical. Every n-tuple appears in one and only one row
- Proof
  - The first part of the theorem follows from the fact that all the code vectors of C are distinct
  - Suppose that two n-tuples in the lth rows are identical, say $\mathbf{e}_l + \mathbf{v}_i = \mathbf{e}_l + \mathbf{v}_j$ with $i \neq j$
  - This means that $\mathbf{v}_i = \mathbf{v}_j$, which is impossible, therefore no two n-tuples in the same row are identical

Standard Array and Syndrome Decoding

- Proof
  - It follows from the construction rule of the standard array that every n-tuple appears at least once
  - Suppose that an n-tuple appears in both lth row and the mth row with $l < m$
  - Then this n-tuple must be equal to $\mathbf{e}_l + \mathbf{v}_i$ for some i and equal to $\mathbf{e}_m + \mathbf{v}_j$ for some j
  - As a result, $\mathbf{e}_l + \mathbf{v}_i = \mathbf{e}_m + \mathbf{v}_j$
  - From this equality we obtain $\mathbf{e}_m = \mathbf{e}_l + (\mathbf{v}_i + \mathbf{v}_j)$
  - Since $\mathbf{v}_i$ and $\mathbf{v}_j$ are code vectors in C, $\mathbf{v}_i + \mathbf{v}_j$ is also a code vector in C, say $\mathbf{v}_s$
  - This implies that the n-tuple $\mathbf{e}_m$ is in the lth row of the array, which contradicts the construction rule of the array that $\mathbf{e}_m$, the first element of the mth row, should be unused in any previous row
  - No n-tuple can appear in more than one row of the array

Standard Array and Syndrome Decoding

- From Theorem 3.3 we see that there are $2^n/2^k = 2^{n-k}$ disjoint rows in the standard array, and each row consists of 2^k distinct elements
- The 2^{n-k} rows are called the *cosets* of the code C
- The first n-tuple $\mathbf{e}_j$ of each coset is called a *coset leader*
- Any element in a coset can be used as its coset leader

Standard Array and Syndrome Decoding

- **Example 3.6** consider the (6, 3) linear code generated by the following matrix :

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

  - The standard array of this code is shown in Fig. 3.7

Standard Array and Syndrome Decoding

Coset leader							
000000	011100	101010	110001	110110	101101	011011	000111
100000	111100	001010	010001	010110	001101	111011	100111
010000	001100	111010	100001	100110	111101	001011	010111
001000	010100	100010	111001	111110	100101	010011	001111
000100	011000	101110	110101	110010	101001	011111	000011
000010	011110	101000	110011	110100	101111	011001	000101
000001	011101	101011	110000	110111	101100	011010	000110
100100	111000	001110	010101	010010	001001	111111	100011

Figure 3.7 Standard array for the (6, 3) code.

Standard Array and Syndrome Decoding

- A standard array of an (n, k) linear code C consists of 2^k disjoint columns
- Let D_j denote the jth column of the standard array, then

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2+\mathbf{v}_j, \mathbf{e}_3+\mathbf{v}_j, \ldots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\} \quad (3.27)$$

  - $\mathbf{v}_j$ is a code vector of C and $\mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_{2^{n-k}}$ are the coset leaders
- The 2^k disjoint columns $D_1, D_2, \ldots, D_{2^k}$ can be used for decoding the code C.
- Suppose that the code vector $\mathbf{v}_j$ is transmitted over a noisy channel, from (3.27) we see that the received vector $\mathbf{r}$ is in D_j if the error pattern caused by the channel is a coset leader
- If the error pattern caused by the channel is not a coset leader, an erroneous decoding will result

Standard Array and Syndrome Decoding

- The decoding is correct if and only if the error pattern caused by the channel is a coset leader
- The 2^{n-k} coset leaders (including the zero vector **0**) are called the *correctable error patterns*
- **Theorem 3.4** Every (n, k) linear block code is capable of correcting 2^{n-k} error pattern
- To minimize the probability of a decoding error, the error patterns that are most likely to occur for a given channel should be chosen as the coset leaders
- When a standard array is formed, each coset leader should be chosen to be a vector of *least weight* from the remaining available vectors

Standard Array and Syndrome Decoding

- Each coset leader has minimum weight in its coset
- The decoding based on the standard array is the minimum distance decoding (i.e. the maximum likelihood decoding)
- Let α_i denote the number of coset leaders of weight *i*, the numbers $\alpha_0, \alpha_1, \ldots, \alpha_n$ are called the *weight distribution* of the coset leaders
- Since a decoding error occurs if and only if the error pattern is not a coset leader, the error probability for a BSC with transition probability *p* is

$$P(\mathrm{E}) = 1 - \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-i}$$

- **Example 3.7**
  - The standard array for this code is shown in Fig. 3.7
  - The weight distribution of the coset leader is $\alpha_0 = 1$, $\alpha_1 = 6$, $\alpha_2 = 1$ and $\alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 0$
  - Thus,

$$P(E) = 1 - (1-p)^6 - 6p(1-p)^5 - p^2(1-p)^4$$

  - For $p = 10^{-2}$, we have $P(E) \approx 1.37 \times 10^{-3}$

Standard Array and Syndrome Decoding

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns, it is capable of correcting only 2^{n-k} error patterns
- The probability of a decoding error is much higher than the probability of an undetected error
- **Theorem 3.5**
  - (1) For an (n, k) linear code C with minimum distance d_{min}, all the n-tuples of weight of $t = \lfloor (d_{min} - 1)/2 \rfloor$ or less can be used as coset leaders of a standard array of C.
  - (2) If all the n-tuple of weight t or less are used as coset leader, there is at least one n-tuple of weight $t + 1$ that cannot be used as a coset leader

Standard Array and Syndrome Decoding

- Proof of the (1)
  - Since the minimum distance of C is $d_{\min}$, the minimum weight of C is also $d_{\min}$
  - Let $\mathbf{x}$ and $\mathbf{y}$ be two n-tuples of weight t or less
  - The weight of $\mathbf{x} + \mathbf{y}$ is

$$w(\mathbf{x} + \mathbf{y}) \le w(\mathbf{x}) + w(\mathbf{y}) \le 2t < d_{\min} \quad (2t+1 \le d_{\min} \le 2t+2)$$

  - Suppose that $\mathbf{x}$ and $\mathbf{y}$ are in the same coset, then $\mathbf{x} + \mathbf{y}$ must be a nonzero code vector in C
  - This is impossible because the weight of $\mathbf{x}+\mathbf{y}$ is less than the minimum weight of C.
  - No two n-tuple of weight t or less can be in the same coset of C
  - All the n-tuples of weight t or less can be used as coset leaders

Standard Array and Syndrome Decoding

- Proof of the (2)
  - Let **v** be a minimum weight code vector of C (i.e., $w(\mathbf{v}) = d_{min}$)
  - Let **x** and **y** be two n-tuples which satisfy the following two conditions:
    - $\mathbf{x} + \mathbf{y} = \mathbf{v}$
    - **x** and **y** do not have nonzero components in common places
  - It follows from the definition that **x** and **y** must be in the same coset and

$$w(\mathbf{x}) + w(\mathbf{y}) = w(\mathbf{v}) = d_{min}$$

  - Suppose we choose **y** such that $w(\mathbf{y}) = t + 1$
  - Since $2t + 1 \leq d_{min} \leq 2t + 2$, we have $w(\mathbf{x})=t$ or $t+1$.
  - If **x** is used as a coset leader, then **y** cannot be a coset leader.

Standard Array and Syndrome Decoding

- Theorem 3.5 reconfirms the fact that an (n, k) linear code with minimum distance $d_{\min}$ is capable of correcting all the error pattern of $\lfloor (d_{\min} - 1)/2 \rfloor$ or fewer errors
- But it is not capable of correcting all the error patterns of weight $t + 1$
- **Theorem 3.6** All the 2^k n-tuples of a coset have the same syndrome. The syndrome for different cosets are different
- Proof
  - Consider the coset whose coset leader is $\mathbf{e}_l$
  - A vector in this coset is the sum of $\mathbf{e}_l$ and some code vector $\mathbf{v}_i$ in C
  - The syndrome of this vector is

$$(\mathbf{e}_l + \mathbf{v}_i)\mathbf{H}^{\mathrm{T}} = \mathbf{e}_l\mathbf{H}^{\mathrm{T}} + \mathbf{v}_i\mathbf{H}^{\mathrm{T}} = \mathbf{e}_l\mathbf{H}^{\mathrm{T}}$$

Standard Array and Syndrome Decoding

- Proof
  - Let $\mathbf{e}_j$ and $\mathbf{e}_l$ be the coset leaders of the jth and lth cosets respectively, where $j < l$
  - Suppose that the syndromes of these two cosets are equal
  - Then,

$$\mathbf{e}_j\mathbf{H}^{\mathrm{T}} = \mathbf{e}_l\mathbf{H}^{\mathrm{T}}$$

$$(\mathbf{e}_j + \mathbf{e}_l)\mathbf{H}^{\mathrm{T}} = \mathbf{0}$$

  - This implies that $\mathbf{e}_j + \mathbf{e}_l$ is a code vector in C, say $\mathbf{v}_j$
  - Thus, $\mathbf{e}_j + \mathbf{e}_l = \mathbf{v}_i$ and $\mathbf{e}_l = \mathbf{e}_j + \mathbf{v}_i$
  - This implies that $\mathbf{e}_l$ is in the jth coset, which contradicts the construction rule of a standard array that a coset leader should be previously unused

Standard Array and Syndrome Decoding

- The syndrome of an n-tuple is an $(n–k)$-tuple and there are 2^{n-k} distinct $(n–k)$-tuples
- From theorem 3.6 that there is a one-to-one correspondence between a coset and an $(n–k)$-tuple syndrome
- Using this one-to-one correspondence relationship, we can form a decoding table, which is much simpler to use than a standard array
- The table consists of 2^{n-k} coset leaders (the correctable error pattern) and their corresponding syndromes
- This table is either stored or wired in the receiver

Standard Array and Syndrome Decoding

- The decoding of a received vector consists of three steps:
  - Step 1. Compute the syndrome of $\mathbf{r}$, $\mathbf{r} \bullet \mathbf{H}^T$
  - Step 2. Locate the coset leader $\mathbf{e}_l$ whose syndrome is equal to $\mathbf{r} \bullet \mathbf{H}^T$, then $\mathbf{e}_l$ is assumed to be the error pattern caused by the channel
  - Step 3. Decode the received vector $\mathbf{r}$ into the code vector $\mathbf{v}$ i.e., $\mathbf{v} = \mathbf{r} + \mathbf{e}_l$
- The decoding scheme described above is called the *syndrome decoding* or *table-lookup decoding*

- **Example 3.8** Consider the (7, 4) linear code given in Table 3.1, the parity-check matrix is given in example 3.3
  - The code has $2^3 = 8$ cosets
  - There are eight correctable error patterns (including the all-zero vector)
  - Since the minimum distance of the code is 3, it is capable of correcting all the error patterns of weight 1 or 0
  - All the 7-tuples of weight 1 or 0 can be used as coset leaders
  - The number of correctable error pattern guaranteed by the minimum distance is equal to the total number of correctable error patterns

Standard Array and Syndrome Decoding

- The correctable error patterns and their corresponding syndromes are given in Table 3.2

TABLE 3.2 DECODING TABLE FOR THE (7, 4) LINEAR CODE GIVEN IN TABLE 3.1

Syndrome	Coset leaders
(1 0 0)	(1 0 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0 0)
(1 1 0)	(0 0 0 1 0 0 0)
(0 1 1)	(0 0 0 0 1 0 0)
(1 1 1)	(0 0 0 0 0 1 0)
(1 0 1)	(0 0 0 0 0 0 1)

Standard Array and Syndrome Decoding

- Suppose that the code vector $\mathbf{v} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ is transmitted and $\mathbf{r} = (1\ 0\ 0\ 1\ 1\ 1\ 1)$ is received
- For decoding $\mathbf{r}$, we compute the syndrome of $\mathbf{r}$

$$\mathbf{s} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$$

WiTS Lab Inst. of Comm. Engr.

Standard Array and Syndrome Decoding

- From Table 3.2 we find that (0 1 1) is the syndrome of the coset leader $\mathbf{e}$ = (0 0 0 0 1 0 0), then $\mathbf{r}$ is decoded into

$$\begin{aligned}\mathbf{v}^* &= \mathbf{r} + \mathbf{e} \\ &= (1\ 0\ 0\ 1\ 1\ 1\ 1) + (0\ 0\ 0\ 0\ 1\ 0\ 0) \\ &= (1\ 0\ 0\ 1\ 0\ 1\ 1)\end{aligned}$$

  - which is the actual code vector transmitted
  - The decoding is correct since the error pattern caused by the channel is a coset leader

Standard Array and Syndrome Decoding

- Suppose that $\mathbf{v} = (0\ 0\ 0\ 0\ 0\ 0\ 0)$ is transmitted and $\mathbf{r} = (1\ 0\ 0\ 0\ 1\ 0\ 0)$ is received
- We see that two errors have occurred during the transmission of $\mathbf{v}$
- The error pattern is not correctable and will cause a decoding error
- When $\mathbf{r}$ is received, the receiver computes the syndrome

$$\mathbf{s} = \mathbf{r} \bullet \mathbf{H}^T = (1\ 1\ 1)$$

- From the decoding table we find that the coset leader $\mathbf{e} = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ corresponds to the syndrome $\mathbf{s} = (1\ 1\ 1)$

Standard Array and Syndrome Decoding

- **r** is decoded into the code vector

$$\begin{aligned}\mathbf{v}^* &= r + e \\ &= (1\ 0\ 0\ 0\ 1\ 0\ 0) + (0\ 0\ 0\ 0\ 0\ 1\ 0) \\ &= (1\ 0\ 0\ 0\ 1\ 1\ 0)\end{aligned}$$

- Since $\mathbf{v}^*$ is not the actual code vector transmitted, a decoding error is committed
- Using Table 3.2, the code is capable of correcting any single error over a block of seven digits
- When two or more errors occur, a decoding error will be committed

Standard Array and Syndrome Decoding

- The table-lookup decoding of an (n, k) linear code may be implemented as follows
- The decoding table is regarded as the truth table of n switch functions :

$$e_0 = f_0(s_0, s_1, \ldots, s_{n-k-1})$$

$$e_1 = f_1(s_0, s_1, \ldots, s_{n-k-1})$$

$$\vdots$$

$$e_{n-1} = f_{n-1}(s_0, s_1, \ldots, s_{n-k-1})$$

  - where $s_0, s_1, \ldots, s_{n\text{-}k\text{-}1}$ are the syndrome digits
  - where $e_0, e_1, \ldots, e_{n\text{-}1}$ are the estimated error digits

- The general decoder for an (n, k) linear code based on the table-lookup scheme is shown in Fig. 3.8

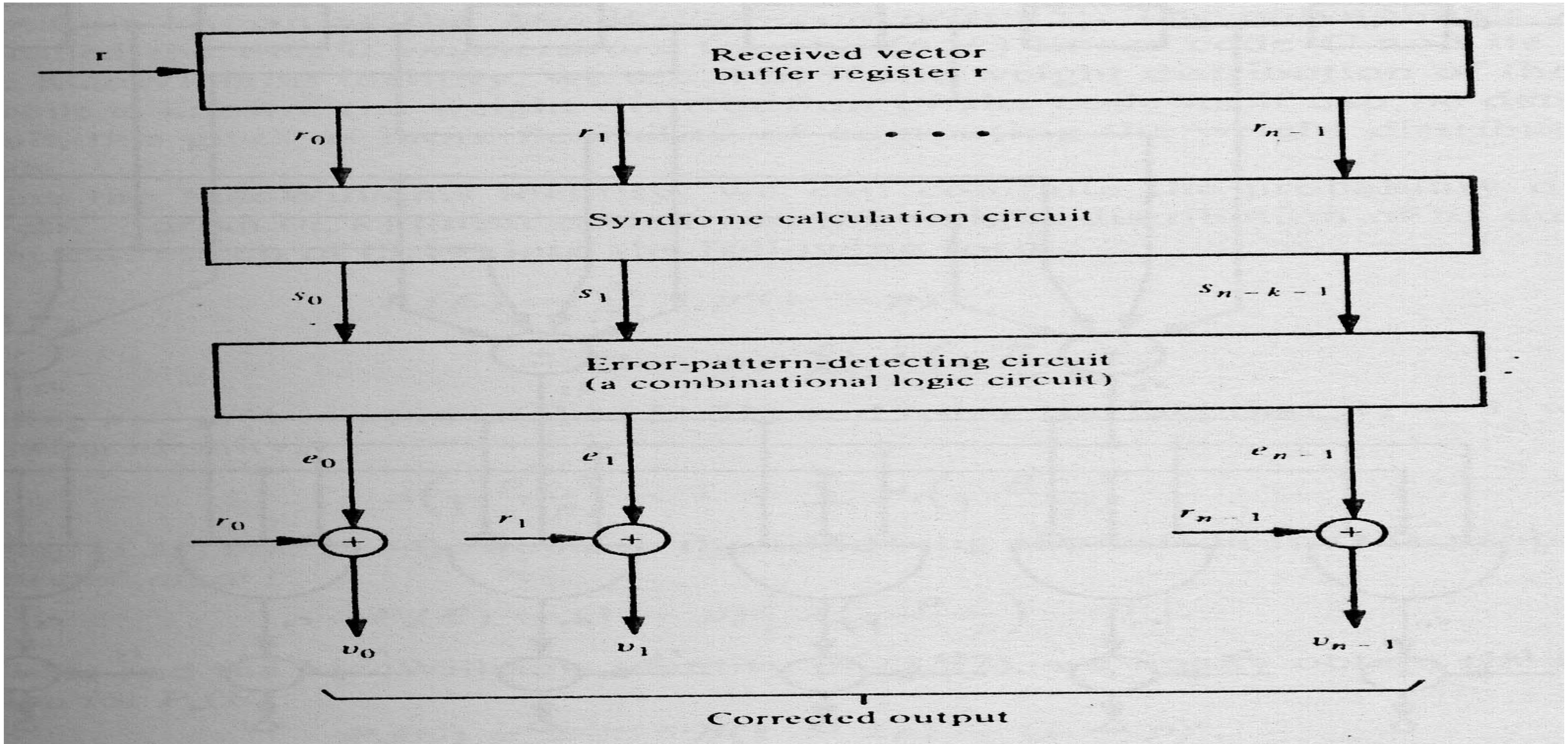


Figure 3.8 General decoder for a linear block code.

Standard Array and Syndrome Decoding

- **Example 3.9** Consider the (7, 4) code given in Table 3.1
  - The syndrome circuit for this code is shown in Fig. 3.5
  - The decoding table is given by Table 3.2
  - From this table we form the truth table (Table 3.3)
  - The switching expression for the seven error digits are

$$e_0 = s_0 \wedge s_1' \wedge s_2' \qquad e_1 = s_0' \wedge s_1 \wedge s_2'$$

$$e_2 = s_0' \wedge s_1' \wedge s_2 \qquad e_3 = s_0 \wedge s_1 \wedge s_2'$$

$$e_4 = s_0' \wedge s_1 \wedge s_2 \qquad e_5 = s_0 \wedge s_1 \wedge s_2$$

$$e_6 = s_0 \wedge s_1' \wedge s_2$$

  - where $\wedge$ denotes the logic-AND operation
  - where s' denotes the logic-COMPLENENT of s

Standard Array and Syndrome Decoding

TABLE 3.3 TRUTH TABLE FOR THE ERROR DIGITS OF THE CORRECTABLE ERROR PATTERNS OF THE (7, 4) LINEAR CODE GIVEN IN TABLE 3.1

Syndromes			Correctable error patterns (coset leaders)						
s_0	s_1	s_2	e_0	e_1	e_2	e_3	e_4	e_5	e_6
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1

- The complete circuit of the decoder is shown in Fig. 3.9

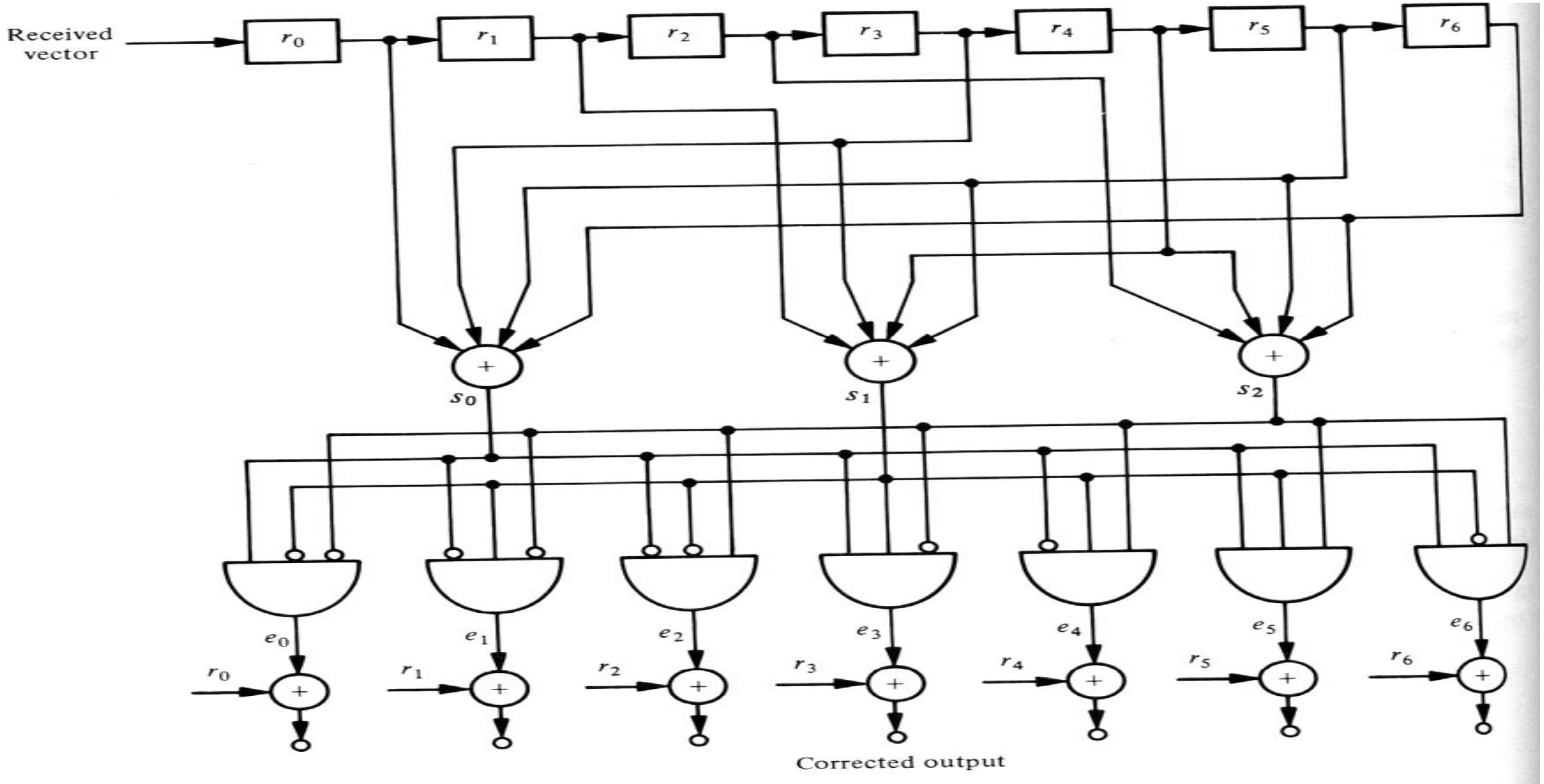


Figure 3.9 Decoding circuit for the (7, 4) code given in Table 3.1.

Probability of An Undetected Error for Linear Codes Over a BSC

Institute of Communications Engineering

Probability of An Undetected Error for Linear Codes Over a BSC

- Let $\{A_0, A_1, \ldots, A_n\}$ be the weight distribution of an (n, k) linear code C
- Let $\{B_0, B_1, \ldots, B_n\}$ be the weight distribution of its dual code C_d
- Now we represent these two weight distribution in polynomial form as follows :

$$A(z) = A_0 + A_1 z + \cdots + A_n z^n$$
$$B(z) = B_0 + B_1 z + \cdots + B_n z^n \qquad (3.31)$$

- Then $A(z)$ and $B(z)$ are related by the following identity :

$$A(z) = 2^{-(n-k)} (1+z)^n B(1-z/1+z) \qquad (3.32)$$

- This identity is known as the *MacWilliams identity*

Probability of An Undetected Error for Linear Codes Over a BSC

- The polynomials $A(z)$ and $B(z)$ are called the *weight enumerators* for the (n, k) linear code C and its dual C_d
- Using the MacWilliams identity, we can compute the probability of an undetected error for an (n, k) linear code from the weight distribution of its dual.
- From equation 3.19:

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$$

$$= (1-p)^n \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i \qquad (3.33)$$

Probability of An Undetected Error for Linear Codes Over a BSC

- Substituting $z = p/(1-p)$ in $A(z)$ of (3.31) and using the fact that $A_0 = 1$, we obtain

$$A\left(\frac{p}{1-p}\right)-1=\sum_{i=1}^{n}A_i\left(\frac{p}{1-p}\right)^i \qquad (3.34)$$

- Combining (3.33) and (3.34), we have the following expression for the probability of an undetected error

$$\boxed{P_u(E)=(1-p)^n[A\left(\frac{p}{1-p}\right)-1]} \qquad (3.35)$$

Probability of An Undetected Error for Linear Codes Over a BSC

- From (3.35) and the MacWilliams identity of (3.32), we finally obtain the following expression for $P_u(\mathrm{E})$:

$$\boxed{P_u(E) = 2^{-(n-k)} B(1-2p) - (1-p)^n} \tag{3.36}$$

where

$$B(1-2p) = \sum_{i=0}^{n} B_i (1-2p)^i$$

- Hence, there are two ways for computing the probability of an undetected error for a linear code; often one is easier than the other.
- If n-k is smaller than k, it is much easier to compute $P_u(E)$ from (3.36); otherwise, it is easier to use (3.35).

Probability of An Undetected Error for Linear Codes Over a BSC

- **Example 3.10** consider the (7, 4) linear code given in Table 3.1
  - The dual of this code is generated by its parity-check matrix

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

  - Taking the linear combinations of the row of **H**, we obtain the following eight vectors in the dual code

(0 0 0 0 0 0 0), (1 1 0 0 1 0 1),
(1 0 0 1 0 1 1), (1 0 1 1 1 0 0),
(0 1 0 1 1 1 0), (0 1 1 1 0 0 1),
(0 0 1 0 1 1 1), (1 1 1 0 0 1 0)

Probability of An Undetected Error for Linear Codes Over a BSC

- **Example 3.10**
  - Thus, the weight enumerator for the dual code is

$$B(z) = 1 + 7z^4$$

  - Using (3.36), we obtain the probability of an undetected error for the (7, 4) linear code given in Table 3.1

$$P_u(E) = 2^{-3}[1 + 7(1 - 2p)^4] - (1 - p)^7$$

  - This probability was also computed in Section 3.4 using the weight distribution of the code itself

Probability of An Undetected Error for Linear Codes Over a BSC

- For large n, k, and $n - k$, the computation becomes practically impossible
- Except for some short linear codes and a few small classes of linear codes, the weight distributions for many known linear code are still unknown
- Consequently, it is very difficult to compute their probability of an undetected error

Probability of An Undetected Error for Linear Codes Over a BSC

- It is quite easy to derive an upper bound on the average probability of an undetected error for the ensemble of all (n, k) linear systematic codes

$$P_u(E) \le 2^{-(n-k)} \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$= 2^{-(n-k)}[1-(1-p)^n] \qquad (3.42)$$

  - Since $[1 - (1 - p)^n] \le 1$, it is clear that $P_u(E) \le 2^{-(n-k)}$.
- There exist (n,k) linear codes with probability of an undetected error, $P_u(E)$, upper bounded by $2^{-(n-k)}$.
- Only a few small classes of linear codes have been proved to have $P_u(E)$ satisfying the upper bound $2^{-(n-k)}$.

Hamming Codes

Hamming Codes

- These codes and their variations have been widely used for error control in digital communication and data storage systems
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
  - Code length: $n = 2^m - 1$
  - Number of information symbols: $k = 2^m - m - 1$
  - Number of parity-check symbols: $n - k = m$
  - Error-correcting capability : $t = 1 (\, d_{\min} = 3)$
  - The parity-check matrix **H** of this code consists of all the nonzero m-tuple as its columns (2^m-1).

Hamming Codes

- In systematic form, the columns of $\mathbf{H}$ are arranged in the following form :

$$\mathbf{H} = [\mathbf{I}_m \quad \mathbf{Q}]$$

  - where $\mathbf{I}_m$ is an $m \times m$ identity matrix
  - The submatrix $\mathbf{Q}$ consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more

- The columns of $\mathbf{Q}$ may be arranged in any order without affecting the distance property and weight distribution of the code

Hamming Codes

- In systematic form, the generator matrix of the code is

$$\mathbf{G} = [\mathbf{Q}^T \;\; \mathbf{I}_{2^m-m-1}]$$

  - where $\mathbf{Q}^T$ is the transpose of $\mathbf{Q}$ and $\mathbf{I}_{2^m-m-1}$ is an $(2^m - m - 1) \times (2^m - m - 1)$ identity matrix
- Since the columns of **H** are nonzero and distinct, no two columns add to zero
- Since **H** consists of all the nonzero *m*-tuples as its columns, the vector sum of any two columns, say $\mathbf{h}_i$ and $\mathbf{h}_j$, must also be a column in **H**, say $\mathbf{h}_l$

$$\mathbf{h}_i + \mathbf{h}_j + \mathbf{h}_l = \mathbf{0}$$

- The minimum distance of a Hamming code is exactly 3

Hamming Codes

- The code is capable of correcting all the error patterns with a single error or of detecting all the error patterns of two or fewer errors

- If we form the standard array for the Hamming code of length $2^m - 1$
  - All the (2^m-1)-tuple of weight 1 can be used as coset leaders
  - The number of (2^m-1)-tuples of weight 1 is $2^m - 1$
  - Since $n - k = m$, the code has 2^m cosets
  - The zero vector $\mathbf{0}$ and the (2^m-1)-tuples of weight 1 form all the coset leaders of the standard array

Hamming Codes

- A t-error-correcting code is called a *perfect code* if its standard array has all the error patterns of t or fewer errors and no others as coset leader
- Besides the Hamming codes, the only other nontrivial binary perfect code is the (23, 12) Golay code (section 5.3)
- Decoding of Hamming codes can be accomplished easily with the table-lookup scheme

Hamming Codes

- We may delete any l columns from the parity-check matrix **H** of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix **H'**
- Using **H'** as a parity-check matrix, we obtain a *shortened* Hamming code with the following parameters :
  - Code length: $n = 2m - l - 1$
  - Number of information symbols: $k = 2^m - m - l - 1$
  - Number of parity-check symbols: $n - k = m$
  - Minimum distance : $d_{\min} \geq 3$
  - If we delete columns from **H** properly, we may obtain a shortened Hamming code with minimum distance 4

Hamming Codes

- For example, if we delete from the submatrix **Q** all the columns of even weight, we obtain an m x 2^{m-1} matrix.

$$\mathbf{H'}=[\mathbf{I}_m \ \ \mathbf{Q'}]$$

  - **Q**' consists of $2^{m-1}-m$ columns of odd weight.
  - Since all columns of **H**' have odd weight, no three columns add to zero.
  - However, for a column $\mathbf{h}_i$ of weight 3 in **Q**', there exists three columns $\mathbf{h}_j$, $\mathbf{h}_l$, and $\mathbf{h}_s$ in $\mathbf{I}_m$ such that $\mathbf{h}_i+\mathbf{h}_j+\mathbf{h}_l+\mathbf{h}_s=0$.
  - Thus, the shortened Hamming code with H' as a parity-check matrix has minimum distance exactly 4.
  - The distance 4 shortened Hamming code can be used for correcting all error patterns of single error and simultaneously detecting all error patterns of double errors

Hamming Codes

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1's ($\mathbf{e}$ x $\mathbf{H}'^{T}$ corresponds to a column in $\mathbf{H}'$)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1's

- Decoding can be accomplished in the following manner :
  - If the syndrome **s** is zero, we assume that no error occurred
  - If **s** is nonzero and it contains odd number of 1's, we assume that a single error occurred. The error pattern of a single error that corresponds to **s** is added to the received vector for error correction
  - If **s** is nonzero and it contains even number of 1's, an uncorrectable error pattern has been detected

Hamming Codes

- The dual code of a $(2^m-1, 2^m-m-1)$ Hamming code is a $(2^m-1, m)$ linear code
- If a Hamming code is used for error detection over a **BSC**, its probability of an undetected error, $P_u(E)$, can be computed either from (3.35) and (3.43) or from (3.36) and (3.44)
- Computing $P_u(E)$ from (3.36) and (3.44) is easier
- Combining (3.36) and (3.44), we obtain

$$P_u(E) = 2^{-m}\{1+(2^m-1)(1-2p)^{2^{m-1}}\} - (1-p)^{2^m-1}$$

- The probability $P_u(E)$ for Hamming codes does satisfy the upper bound $2^{-(n-k)} = 2^{-m}$ for $p \leq ½$ [i.e., $P_u(E) \leq 2^{-m}$]

Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

Institute of Communications Engineering

National Sun Yat-sen University

Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

- A *single-parity-check* (SPC) code is a linear block code with a single parity-check digit.
  - Let $\mathbf{u}=(u_0,u_1,\ldots,u_{k-1})$ be the message to be encoded. The single parity-check digit is given by

$$p= u_0+u_1+\ldots+u_{k-1}$$

    which is simply the modulo-2 sum of all the message digits.
  - Adding this parity-check digit to each k-digit message results in a $(k+1,k)$ linear block code. Each codeword is of the form

$$\mathbf{v}=(p,\mathrm{u}_0,\mathrm{u}_1,\ldots,u_{k-1})$$

  - p=1(0) if the weight of message $\mathbf{u}$ is odd(even).
  - All the codewords of a SPC code have even weights, and the minimum weight (or minimum distance) of the code is 2.

Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

- The generator of the code in systematic form is given by

$$\mathbf{G} = \begin{bmatrix} 1 & \vdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & \vdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 1 & \vdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & & & \vdots & & \\ 1 & \vdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 & \vdots & \\ 1 & \vdots & \\ 1 & \vdots & \mathbf{I}_k \\ \vdots & \vdots & \\ 1 & \vdots & \end{bmatrix}$$

- The parity-check matrix of the code is

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

- A SPC code is also called an even-parity-check code.
- All the error patterns of odd weight are detectable.

Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

- A repetition code of length n is an (n,1) linear block code that consists of only two codewords, the all-zero codeword (00…0) and the all-one codeword (11…1).
  - This code is obtained by simply repeating a single message bit n times. The generator matrix of the code is

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

- From the generator matrixes, we see that the $(n,1)$ repetition code and the $(n,n\text{-}1)$ SPC code are dual codes to each other.
- A linear block code C that is equal to its dual code C_d is called a *self-dual code*.
- For a self-dual code, the code length n must be even, and the dimension k of the code must be equal to $n/2$. => Rate=1/2.

Single-Parity-Check Codes, Repetition Codes, and Self-Dual Codes

- Let $\mathbf{G}$ be a generator matrix of a self-dual code C. Then, $\mathbf{G}$ is also a generator matrix of its dual code C_d and hence is a parity-check matrix of C. Consequently,

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

- Suppose $\mathbf{G}$ is in systematic form, $\mathbf{G}=[\mathbf{P}\ \mathbf{I}_{n/2}]$. We can see that

$$\mathbf{P}\cdot\mathbf{P}^T=\mathbf{I}_{n/2}$$

- Conversely, if a rate ½ $(n,n/2)$ linear block code C satisfies the condition of $\mathbf{G}\cdot\mathbf{G}^T=\mathbf{0}$ or $\mathbf{P}\cdot\mathbf{P}^T=\mathbf{I}_{n/2}$, then it is a self-dual code.
- Example: the (8,4) linear block code generated by the following matrix has a rate $R=$ ½ and is a self-dual code since $\mathbf{G}\cdot\mathbf{G}^T=\mathbf{0}$.

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