Chapter 3
Linear Block Codes
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
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.

```
Encoder

u -> v
```

Message block
- $(k \text{ info. digits})$
- $(2^k \text{ distinct message})$

Code word
- $(n\text{-tuple }, n > k)$
- $(2^k \text{ distinct code word})$
Definition 3.1. A block code of length $n$ and $2^k$ code words is called a linear $(n, k)$ code iff its $2^k$ code words form a $k$-dimensional subspace of the vector space of all the $n$-tuple over the field $\text{GF}(2)$.

In fact, a binary block code is linear iff the module-2 sum of two code words is also a code word

- $0$ must be code word.
- The block code given in Table 3.1 is a $(7, 4)$ linear code.
### Table 3.1 Linear Block Code with $k = 4$ and $n = 7$

<table>
<thead>
<tr>
<th>Messages</th>
<th>Code words</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0 0 0 0)</td>
<td>(0 0 0 0 0 0 0)</td>
</tr>
<tr>
<td>(1 0 0 0)</td>
<td>(1 1 0 1 0 0 0)</td>
</tr>
<tr>
<td>(0 1 0 0)</td>
<td>(0 1 1 0 1 0 0)</td>
</tr>
<tr>
<td>(1 1 0 0)</td>
<td>(1 0 1 1 1 0 0)</td>
</tr>
<tr>
<td>(0 0 1 0)</td>
<td>(1 1 1 0 0 1 0)</td>
</tr>
<tr>
<td>(1 0 1 0)</td>
<td>(0 0 1 1 0 1 0)</td>
</tr>
<tr>
<td>(0 1 1 0)</td>
<td>(1 0 0 0 1 1 0)</td>
</tr>
<tr>
<td>(1 1 1 0)</td>
<td>(0 1 0 1 1 1 0)</td>
</tr>
<tr>
<td>(0 0 0 1)</td>
<td>(1 0 1 0 0 0 1)</td>
</tr>
<tr>
<td>(1 0 0 1)</td>
<td>(0 1 1 1 0 0 1)</td>
</tr>
<tr>
<td>(0 1 0 1)</td>
<td>(1 1 0 0 1 0 1)</td>
</tr>
<tr>
<td>(1 1 0 1)</td>
<td>(0 0 0 1 1 0 1)</td>
</tr>
<tr>
<td>(0 0 1 1)</td>
<td>(0 1 0 0 0 1 1)</td>
</tr>
<tr>
<td>(1 0 1 1)</td>
<td>(0 0 1 0 1 1 1)</td>
</tr>
<tr>
<td>(0 1 1 1)</td>
<td>(1 0 1 0 1 1 1)</td>
</tr>
</tbody>
</table>

For large $k$, it is virtually impossible to build up the loop up table.
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, \(g_0, g_1, \ldots, g_{k-1}\) in \(C\)

\[
v = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}
\]  \hspace{1cm} (3.1)

where \(u_i = 0\) or \(1\) for \(0 \leq i < k\)
Let us arrange these \( k \) linearly independent code words as the rows of a \( k \times n \) matrix as follows:

\[
G = \begin{bmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{k-1}
\end{bmatrix} = \begin{bmatrix}
g_{0,0} & g_{0,1} & g_{0,2} & \cdots & g_{0,n-1} \\
g_{1,0} & g_{1,1} & g_{1,2} & \cdots & g_{1,n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1}
\end{bmatrix}
\]  

(3.2)

where \( g_i = (g_{i,0}, g_{i,1}, \ldots, g_{i,n-1}) \) for \( 0 \leq i < k \)
If \( \mathbf{u} = (u_0, u_1, \cdots, u_{k-1}) \) is the message to be encoded, the corresponding code word can be given as follows:

\[
\mathbf{v} = \mathbf{u} \cdot \mathbf{G}
\]

\[
= (u_0, u_1, \ldots, u_{k-1}) \cdot \begin{bmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{k-1}
\end{bmatrix}
\]

\[
= u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}
\]
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 $G$. 


Example 3.1

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

\[ G = \begin{bmatrix}
  g_0 \\
  g_1 \\
  g_2 \\
  g_3 \\
\end{bmatrix} = \begin{bmatrix}
  1 & 1 & 0 & 1 & 0 & 0 & 0 \\
  0 & 1 & 1 & 0 & 1 & 0 & 0 \\
  1 & 1 & 1 & 0 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \]

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

\[ v = 1 \cdot g_0 + 1 \cdot g_1 + 0 \cdot g_2 + 1 \cdot g_3 \]

\[ = (1101000) + (0110100) + (1010001) \]

\[ = (0001101) \]
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**

![Systematic format of a code word](image-url)
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:

\[
G = \begin{bmatrix}
g_0 \\
g_1 \\
g_2 \\
\vdots \\
g_{k-1}
g_k
\end{bmatrix} = \begin{bmatrix}
p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\
p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\
p_{20} & p_{21} & \cdots & p_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

(3.4)

where \(p_{ij} = 0\) or \(1\)
Introduction to Linear Block Codes

Let \( u = (u_0, u_1, \ldots, u_{k-1}) \) be the message to be encoded.

The corresponding code word is

\[
v = (v_0, v_1, v_2, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \tag{3.5}
\]

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

\[
v_{n-k+i} = u_i \quad \text{for } 0 \leq i < k \tag{3.6a}
\]

and

\[
v_j = u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} \quad \text{for } 0 \leq j < n-k \tag{3.6b}
\]
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 \( n - k \) redundant digits are linear sums of the information digits.

The \( n - k \) equations given by (3.6b) are called \textit{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:

$$v = u \cdot G = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$
By matrix multiplication, we obtain the following digits of the code word \( v \):

\[
\begin{align*}
    v_6 &= u_3 \\
    v_5 &= u_2 \\
    v_4 &= u_1 \\
    v_3 &= u_0 \\
    v_2 &= u_1 + u_2 + u_3 \\
    v_1 &= u_0 + u_1 + u_2 \\
    v_0 &= u_0 + u_2 + u_3
\end{align*}
\]

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 $(n-k) \times n$ matrix $H$ with $n-k$ 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 $v \cdot H^T = 0$

- 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 $G$

- $C_d$ is called the *dual code* of $C$
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:

\[
H = \begin{bmatrix}
I_{n-k} & P^T \\
1 & 0 & 0 & . & . & . & 0 & p_{00} & p_{10} & . & . & . & p_{k-1,0} \\
0 & 1 & 0 & . & . & . & 0 & p_{01} & p_{11} & . & . & . & p_{k-1,1} \\
0 & 0 & 1 & . & . & . & 0 & p_{02} & p_{12} & . & . & . & p_{k-1,2} \\
. & & & & & & & & & & & & . \\
. & & & & & & & & & & & & . \\
0 & 0 & 0 & . & . & . & 1 & p_{0,n-k-1} & p_{1,n-k-1} & . & . & . & p_{k-1,n-k-1}
\end{bmatrix}
\]

(3.7)
Let \( h_j \) be the \( j_{th} \) row of \( H \)

\[
g_i \cdot h_j = p_{ij} + p_{ij} = 0
\]

for \( 0 \leq i < k \) and \( 0 \leq j < n - k \)

This implies that \( G \cdot H^T = 0 \)
Let \( u = (u_0, u_1, \ldots, u_{k-1}) \) be the message to be encoded.

In systematic form the corresponding code word would be

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

Using the fact that \( v \cdot H^T = 0 \), we obtain

\[
v_j + u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} = 0
\]

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

\[ H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 \\
\end{bmatrix} \]
Summaries

For any \((n, k)\) linear block code \(C\), there exists a \(k \times n\) matrix \(G\) whose row space gives \(C\).

There exist an \((n - k) \times n\) matrix \(H\) such that an \(n\)-tuple \(v\) is a code word in \(C\) if and only if \(v \cdot H^T = 0\).

If \(G\) is of the form given by (3.4), then \(H\) 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
- \(\square\) denotes a shift-register stage (flip-flop)
- \(P_{ij}\) 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.
**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 \( v = (v_0, v_1, \ldots, v_{n-1}) \) be a code word that was transmitted over a noisy channel.

Let \( r = (r_0, r_1, \ldots, r_{n-1}) \) be the received vector at the output of the channel.

\[
\mathbf{e} = \mathbf{r} + \mathbf{v} = (e_0, e_1, \ldots, e_{n-1}) \quad \text{is an } n \text{-tuple}
\]

- \( e_i = 1 \) 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).
After 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:

\[
s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1})
\]  

which is called the *syndrome* of \( r \).
 Syndrome and Error Detection

- $s = 0$ if and only if $r$ is a code word and receiver accepts $r$ as the transmitted code word
- $s \neq 0$ if and only if $r$ is not a code word and the presence of errors has been detected
- When the error pattern $e$ is identical to a nonzero code word (i.e., $r$ contain errors but $s = r \cdot H^T = 0$), 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
Based on (3.7) and (3.10), the syndrome digits are as follows:

\[
\begin{align*}
    s_0 &= r_0 + r_{n-k} p_{00} + r_{n-k+1} p_{10} + \cdots + r_{n-1} p_{k-1,0} \\
    s_1 &= r_1 + r_{n-k} p_{01} + r_{n-k+1} p_{11} + \cdots + r_{n-1} p_{k-1,1} \\
    \quad \vdots \\
    s_{n-k-1} &= r_{n-k-1} + r_{n-k} p_{0,n-k-1} + r_{n-k+1} p_{1,n-k-1} + \cdots + r_{n-1} p_{k-1,n-k-1}
\end{align*}
\]

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

\[ H = [I_{n-k} \quad P^T] \quad s = r \cdot H^T \]
Syndrome and Error Detection

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

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

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

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

however,

\[
\mathbf{v} \cdot \mathbf{H}^T = 0
\]

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

\[
\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T
\] (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 $e \cdot H^T$ yield the following linear relationship between the syndrome digits and the error digits:

\[
\begin{align*}
  s_0 &= e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \cdots + e_{n-1}p_{k-1,0} \\
  s_1 &= e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \cdots + e_{n-1}p_{k-1,1} \\
  &\quad \vdots \\
  s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{0,n-k-1} + \cdots + e_{n-1}p_{k-1,n-k-1}
\end{align*}
\]

(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 patterns 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.
Example 3.5

We consider the (7,4) code whose parity-check matrix is given in example 3.3

Let \( v = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1) \) be the transmitted code word

Let \( r = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) \) be the received vector

The receiver computes the syndrome

\[ s = r \cdot H^T = (1 \ 1 \ 1) \]

The receiver attempts to determine the true error vector \( e = (e_0, e_1, e_2, e_3, e_4, e_5, e_6) \), 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
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**, \( e = (0 \ 0 \ 0 \ 0 \ 1 \ 0) \) is the most probable error vector that satisfies the equation above.

Taking \( e = (0 \ 0 \ 0 \ 0 \ 1 \ 0) \) as the true error vector, the receiver decodes the received vector \( r = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) \) into the following code word:

\[
\mathbf{v}^* = \mathbf{r} + \mathbf{e} = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) + (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0) \\
= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1)
\]

where \( \mathbf{v}^* \) is the actual transmitted code word.
The Minimum Distance of a Block Code
The Minimum Distance of a Block Code

Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary $n$-tuple, the **Hamming weight** (or simply weight) of $v$, denote by $w(v)$, is defined as the number of nonzero components of $v$

- For example, the Hamming weight of $v = (1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0)$ is 3

Let $v$ and $w$ be two $n$-tuple, the **Hamming distance** between $v$ and $w$, denoted $d(v,w)$, is defined as the number of places where they differ

- For example, the Hamming distance between $v = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1)$ and $w = (0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1)$ is 3
The Minimum Distance of a Block Code

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

\[ d(v, w) + d(w, x) \geq d(v, x) \]  (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, \( v \) and \( w \), is equal to the Hamming weight of the sum of \( v \) and \( w \), that is

\[ d(v, w) = w(v + w) \]  (3.15)

- For example, the Hamming distance between \( v = (1 0 0 1 0 1 1) \) and \( w = (1 1 1 0 0 1 0) \) is 4 and the weight of \( v + w = (0 1 1 1 0 0 1) \) is also 4
Given, a block code $C$, the minimum distance of $C$, denoted $d_{\text{min}}$, is defined as

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

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 $C$

$$d_{\text{min}} = \min\{w(v + w) : v, w \in C, v \neq w\}$$
$$= \min\{w(x) : x \in C, x \neq 0\} \quad (3.17)$$
$$\equiv w_{\text{min}}$$
The Minimum Distance of a Block Code

- The parameter \( w_{\text{min}} \equiv \{ w(x) : x \in C, x \neq 0 \} \) is called the \textit{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 \( H \).
  - For each code vector of Hamming weight \( l \), there exist \( l \) columns of \( H \) such that the vector sum of these \( l \) columns is equal to the zero vector.
  - Conversely, if there exist \( l \) columns of \( H \) whose vector sum is the zeros vector, there exists a code vector of Hamming weight \( l \) in \( C \).
Proof

Let the parity-check matrix be

\[ H = [h_0, h_1, \ldots, h_{n-1}] \]

where \( h_i \) represents the \( i \)th column of \( H \).

Let \( \mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \) be a code vector of weight \( l \) and \( \mathbf{v} \) has \( l \) nonzero components.

Let \( v_{i1}, v_{i2}, \ldots, v_{il} \) be the \( l \) nonzero components of \( \mathbf{v} \), where \( 0 \leq i_1 < i_2 < \cdots < i_l \leq n-1 \), then \( v_{i1} = v_{i2} = \cdots = v_{il} = 1 \).

Since \( \mathbf{v} \) is code vector, we must have

\[ 0 = \mathbf{v} \cdot H^T \]

\[ = v_0 h_0 + v_1 h_1 + \cdots + v_{n-1} h_{n-1} \]

\[ = v_{i1} h_{i1} + v_{i2} h_{i2} + \cdots + v_{il} h_{il} \]

\[ = h_{i1} + h_{i2} + \cdots + h_{il} \]
Proof

- Suppose that $h_{i_1}, h_{i_2}, \ldots, h_{i_l}$ are $l$ columns of $H$ such that
  \[ h_{i_1} + h_{i_2} + \cdots + h_{i_l} = 0 \] (3.18)

- Let $x = (x_1, x_2, \ldots, x_{n-1})$ whose nonzero components are $x_{i_1}, x_{i_2}, x_{i_l}$
  \[
  x \cdot H^T = x_0 h_0 + x_1 h_1 + \cdots + x_{n-1} h_{n-1}
  = x_{i_1} h_{i_1} + x_{i_2} h_{i_2} + \cdots + x_{i_l} h_{i_l}
  = h_{i_1} + h_{i_2} + \cdots + h_{i_l}
  \]

- It following from (3.18) that $x \cdot H^T = 0$, $x$ is code vector of weight $l$ in $C$
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
If the minimum distance of a block code $C$ is $d_{\text{min}}$, any two distinct code vector of $C$ differ in at least $d_{\text{min}}$ places.

A block code with minimum distance $d_{\text{min}}$ is capable of detecting all the error pattern of $d_{\text{min}} - 1$ or fewer errors.

However, it cannot detect all the error pattern of $d_{\text{min}}$ errors because there exists at least one pair of code vectors that differ in $d_{\text{min}}$ places and there is an error pattern of $d_{\text{min}}$ errors that will carry one into the other.

The random-error-detecting capability of a block code with minimum distance $d_{\text{min}}$ is $d_{\text{min}} - 1$. 
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 \(v\) into another code word \(w\), thus \(w\) 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 \(r\) will not be a code word and the syndrome will not be zero.
These $2^n - 2^k$ error patterns are detectable error patterns. Let $A_i$ 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 $C$

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

(3.19)

where $p$ is the transition probability of the BSC.

If the minimum distance of $C$ is $d_{\text{min}}$, then $A_1$ to $A_{d_{\text{min}} - 1}$ are zero.
Assume that a block code \( C \) with minimum distance \( d_{\text{min}} \) is used for random-error correction. The minimum distance \( d_{\text{min}} \) is either odd or even. Let \( t \) be a positive integer such that:

\[
2t + 1 \leq d_{\text{min}} \leq 2t + 2
\]

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 \( C \).

\[
d(\mathbf{v}, \mathbf{r}) + d(\mathbf{w}, \mathbf{r}) \geq d(\mathbf{v}, \mathbf{w})
\]

\- Suppose that an error pattern of \( t' \) errors occurs during the transmission of \( \mathbf{v} \). We have \( d(\mathbf{v}, \mathbf{r}) = t' \).
Since \( v \) and \( w \) are code vectors in \( C \), we have

\[
d(v, w) \geq d_{\text{min}} \geq 2t + 1.
\]

\[
d(w, r) \geq d(v, w) - d(v, r)
\]

\[
\geq d_{\text{min}} - t'
\]

\[
\geq 2t + 1 - t' \geq t + 1
\]

\[
> t \quad (\text{if } t \geq t') \Rightarrow d(w, r) > t'
\]

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 \( P(r|v) \) is greater than the conditional probability \( P(r|w) \) for \( w \neq v \). Q.E.D.
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 \( v \) and \( w \) be two code vectors in \( C \) such that \( d(v, w) = d_{\text{min}} \).
- Let \( e_1 \) and \( e_2 \) be two error patterns that satisfy the following conditions:
  - \( e_1 + e_2 = v + w \)
  - \( e_1 \) and \( e_2 \) do not have nonzero components in common places.
- We have \( w(e_1) + w(e_2) = w(v + w) = d(v, w) = d_{\text{min}}. \) (3.23)
Suppose that $v$ is transmitted and is corrupted by the error pattern $e_1$, then the received vector is

$$r = v + e_1$$

The Hamming distance between $v$ and $r$ is

$$d(v, r) = w(v + r) = w(e_1). \quad (3.24)$$

The Hamming distance between $w$ and $r$ is

$$d(w, r) = w(w + r) = w(w + v + e_1) = w(e_2) \quad (3.25)$$

Now, suppose that the error pattern $e_1$ contains more than $t$ errors [i.e. $w(e_1) \geq t + 1$].

Since $2t + 1 \leq d_{\text{min}} \leq 2t + 2$, it follows from (3.23) that

$$w(e_2) = d_{\text{min}} - w(e_1) \leq (2t + 2) - (t + 1) = t + 1$$
Error-Detecting and Error-Correcting Capabilities of a Block Code

- Combining (3.24) and (3.25) and using the fact that \( w(e_1) \geq t + 1 \) and \( w(e_2) \leq t + 1 \), we have
  \[
  d(v, r) \geq d(w, r)
  \]
- This inequality says 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 $d_{\text{min}}$ guarantees correcting all the error patterns of $t = \lfloor (d_{\text{min}} - 1)/2 \rfloor$ or fewer errors, where $\lfloor (d_{\text{min}} - 1)/2 \rfloor$ denotes the largest integer no greater than $(d_{\text{min}} - 1)/2$

- The parameter $t = \lfloor (d_{\text{min}} - 1)/2 \rfloor$ 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:

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

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

- The minimum distance $d_{\text{min}}$ of the code is at least $\lambda + l + 1$. 

59
Standard Array and Syndrome Decoding
Let $v_1, v_2, \ldots, v_{2^k}$ 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 $v_i$ 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 $v_i$

If the received vector $r$ is found in the subset $D_i$, $r$ is decoded into $v_i$

Correct decoding is made if and only if the received vector $r$ is in the subset $D_i$ that corresponds to the actual code vector transmitted
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 $v_1 = (0, 0, \ldots, 0)$ as the first (leftmost) element.
- From the remaining $2^n - 2^k$ $n$-tuple, an $n$-tuple $e_2$ is chosen and is placed under the zero vector $v_1$.
- Now, we form a second row by adding $e_2$ to each code vector $v_i$ in the first row and placing the sum $e_2 + v_i$ under $v_i$.
- An unused $n$-tuple $e_3$ is chosen from the remaining $n$-tuples and is placed under $e_2$.
- Then a third row is formed by adding $e_3$ to each code vector $v_i$ in the first row and placing $e_3 + v_i$ under $v_i$.
- We continue this process until all the $n$-tuples are used.
Standard Array and Syndrome Decoding

\[
\begin{array}{cccccc}
0 & v_2 & \cdots & v_i & \cdots & v_{2k} \\
e_2 & e_2 + v_2 & \cdots & e_2 + v_i & \cdots & e_2 + v_{2k} \\
e_3 & e_3 + v_2 & \cdots & e_3 + v_i & \cdots & e_3 + v_{2k} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
e_l & e_l + v_2 & \cdots & e_l + v_i & \cdots & e_l + v_{2k} \\
e_{2n-k} & e_{2n-k} + v_2 & \cdots & e_{2n-k} + v_i & \cdots & e_{2n-k} + v_{2k} \\
\end{array}
\]

Figure 3.6 Standard array for an \((n, k)\) linear code.
Then we have an array of rows and columns as shown in Fig 3.6

This array is called a \textit{standard array} of the given linear code $C$

\textbf{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

\textbf{Proof}

\begin{itemize}
  \item The first part of the theorem follows from the fact that all the code vectors of $C$ are distinct
  \item Suppose that two $n$-tuples in the $l$th rows are identical, say $e_i + v_i = e_l + v_j$ with $i \neq j$
  \item This means that $v_i = v_j$, which is impossible, therefore no two $n$-tuples in the same row are identical
\end{itemize}
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 \( l \)th row and the \( m \)th row with \( l < m \).
- Then this \( n \)-tuple must be equal to \( e_l + v_i \) for some \( i \) and equal to \( e_m + v_j \) for some \( j \).
- As a result, \( e_l + v_i = e_m + v_j \).
- From this equality we obtain \( e_m = e_l + (v_i + v_j) \).
- Since \( v_i \) and \( v_j \) are code vectors in \( C \), \( v_i + v_j \) is also a code vector in \( C \), say \( v_s \).
- This implies that the \( n \)-tuple \( e_m \) is in the \( l \)th row of the array, which contradicts the construction rule of the array that \( e_m \), the first element of the \( m \)th 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 $e_j$ of each coset is called a *coset leader*.
- Any element in a coset can be used as its coset leader.
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
<table>
<thead>
<tr>
<th>Coset leader</th>
<th>011100</th>
<th>101010</th>
<th>110001</th>
<th>110110</th>
<th>101101</th>
<th>011011</th>
<th>000111</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100000</td>
<td>111100</td>
<td>001010</td>
<td>010001</td>
<td>010110</td>
<td>001101</td>
<td>111011</td>
<td>100111</td>
</tr>
<tr>
<td>010000</td>
<td>001100</td>
<td>111010</td>
<td>100001</td>
<td>100110</td>
<td>111101</td>
<td>001011</td>
<td>010111</td>
</tr>
<tr>
<td>001000</td>
<td>010100</td>
<td>100010</td>
<td>111001</td>
<td>111110</td>
<td>100101</td>
<td>010011</td>
<td>001111</td>
</tr>
<tr>
<td>000100</td>
<td>011000</td>
<td>101110</td>
<td>110101</td>
<td>110010</td>
<td>101001</td>
<td>011111</td>
<td>000011</td>
</tr>
<tr>
<td>000010</td>
<td>011110</td>
<td>101000</td>
<td>110011</td>
<td>110100</td>
<td>101111</td>
<td>011001</td>
<td>000101</td>
</tr>
<tr>
<td>000001</td>
<td>011101</td>
<td>101011</td>
<td>110000</td>
<td>110111</td>
<td>101100</td>
<td>011010</td>
<td>000110</td>
</tr>
<tr>
<td>100100</td>
<td>111000</td>
<td>001110</td>
<td>010101</td>
<td>010010</td>
<td>001001</td>
<td>111111</td>
<td>100011</td>
</tr>
</tbody>
</table>

**Figure 3.7** Standard array for the (6, 3) code.
A standard array of an \((n, k)\) linear code \(C\) consists of \(2^k\) disjoint columns.

Let \(D_j\) denote the \(j\)th column of the standard array, then

\[
D_j = \{v_j, e_2 + v_j, e_3 + v_j, \ldots, e_{2n-k} + v_j\} \quad (3.27)
\]

\(v_j\) is a code vector of \(C\) and \(e_2, e_3, \ldots, e_{2n-k}\) 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 \(v_j\) is transmitted over a noisy channel, from (3.27) we see that the received vector \(r\) 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.
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 $\alpha_i$ 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

$$P(E) = 1 - \sum_{i=0}^{n} \alpha_i p^i (1 - p)^{n-i}$$
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 $P(E) \approx 1.37 \times 10^{-3}$
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 \(d_{\text{min}}\), all the \(n\)-tuples of weight of \(t = \left\lfloor (d_{\text{min}} - 1) / 2 \right\rfloor\) 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.
Proof of the (1)

Since the minimum distance of $C$ is $d_{\text{min}}$, the minimum weight of $C$ is also $d_{\text{min}}$

Let $x$ and $y$ be two $n$-tuples of weight $t$ or less

The weight of $x + y$ is

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

Suppose that $x$ and $y$ are in the same coset, then $x + y$ must be a nonzero code vector in $C$

This is impossible because the weight of $x+y$ 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
Proof of the (2)

- Let \( \mathbf{v} \) be a minimum weight code vector of \( C \) (i.e., \( w(\mathbf{v}) = d_{\text{min}} \))
- Let \( \mathbf{x} \) and \( \mathbf{y} \) be two \( n \)-tuples which satisfy the following two conditions:
  - \( \mathbf{x} + \mathbf{y} = \mathbf{v} \)
  - \( \mathbf{x} \) and \( \mathbf{y} \) do not have nonzero components in common places
- It follows from the definition that \( \mathbf{x} \) and \( \mathbf{y} \) must be in the same coset (because \( \mathbf{y} = \mathbf{x} + \mathbf{v} \)) and
  \[
  w(\mathbf{x}) + w(\mathbf{y}) = w(\mathbf{v}) = d_{\text{min}}
  \]
- Suppose we choose \( \mathbf{y} \) such that \( w(\mathbf{y}) = t + 1 \)
- Since \( 2t + 1 \leq d_{\text{min}} \leq 2t + 2 \), we have \( w(\mathbf{x}) = t \) or \( t+1 \).
- If \( \mathbf{x} \) is used as a coset leader, then \( \mathbf{y} \) cannot be a coset leader.
Theorem 3.5 reconfirms the fact that an \((n, k)\) linear code with minimum distance \(d_{\text{min}}\) is capable of correcting all the error pattern of \(\left\lfloor (d_{\text{min}} - 1)/2 \right\rfloor\) 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 \(e_i\).
- A vector in this coset is the sum of \(e_i\) and some code vector \(v_i\) in \(C\).
- The syndrome of this vector is
  \[ (e_i + v_i)H^T = e_iH^T + v_iH^T = e_iH^T \]
Proof

- Let $e_j$ and $e_l$ be the coset leaders of the $j$th and $l$th cosets respectively, where $j < l$.
- Suppose that the syndromes of these two cosets are equal.
- Then,

$$e_j H^T = e_l H^T$$

$$(e_j + e_l)H^T = 0$$

- This implies that $e_j + e_l$ is a code vector in $C$, say $v_j$.
- Thus, $e_j + e_l = v_j$ and $e_l = e_j + v_j$.
- This implies that $e_l$ is in the $j$th 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.
The decoding of a received vector consists of three steps:

- **Step 1.** Compute the syndrome of \( r, r \cdot H^T \)
- **Step 2.** Locate the coset leader \( e_l \) whose syndrome is equal to \( r \cdot H^T \), then \( e_l \) is assumed to be the error pattern caused by the channel
- **Step 3.** Decode the received vector \( r \) into the code vector \( v \)
  
i.e., \( v = r + e_l \)

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

<table>
<thead>
<tr>
<th>Syndrome</th>
<th>Coset leaders</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 0 0)</td>
<td>(1 0 0 0 0 0 0 0)</td>
</tr>
<tr>
<td>(0 1 0)</td>
<td>(0 1 0 0 0 0 0 0)</td>
</tr>
<tr>
<td>(0 0 1)</td>
<td>(0 0 1 0 0 0 0 0)</td>
</tr>
<tr>
<td>(1 1 0)</td>
<td>(0 0 0 1 0 0 0 0)</td>
</tr>
<tr>
<td>(0 1 1)</td>
<td>(0 0 0 0 1 0 0 0)</td>
</tr>
<tr>
<td>(1 1 1)</td>
<td>(0 0 0 0 0 1 0 0)</td>
</tr>
<tr>
<td>(1 0 1)</td>
<td>(0 0 0 0 0 0 0 1)</td>
</tr>
</tbody>
</table>
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 \( \mathbf{r} \)

\[
\mathbf{s} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}
\]
From Table 3.2 we find that \((0 1 1)\) is the syndrome of the coset leader \(e = (0 0 0 0 1 0 0)\), then \(r\) is decoded into

\[
\mathbf{v}^* = \mathbf{r} + \mathbf{e}
\]

\[
= (1 0 0 1 1 1 1) + (0 0 0 0 1 0 0)
\]

\[
= (1 0 0 1 0 1 1)
\]

which is the actual code vector transmitted.

The decoding is correct since the error pattern caused by the channel is a coset leader.
Suppose that $v = (0 0 0 0 0 0 0)$ is transmitted and $r = (1 0 0 0 1 0 0)$ is received.

We see that two errors have occurred during the transmission of $v$.

The error pattern is not correctable and will cause a decoding error.

When $r$ is received, the receiver computes the syndrome $s = r \cdot H^T = (1 1 1)$.

From the decoding table we find that the coset leader $e = (0 0 0 0 0 1 0)$ corresponds to the syndrome $s = (1 1 1)$.
r is decoded into the code vector

\[ v^* = r + e = (1 0 0 0 1 0 0) + (0 0 0 0 0 1 0) = (1 0 0 0 1 1 0) \]

Since \( v^* \) 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.
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:

\[
\begin{align*}
    e_0 &= f_0(s_0, s_1, \ldots, s_{n-k-1}) \\
    e_1 &= f_1(s_0, s_1, \ldots, s_{n-k-1}) \\
    & \quad \vdots \\
    e_{n-1} &= f_{n-1}(s_0, s_1, \ldots, s_{n-k-1})
\end{align*}
\]

where \(s_0, s_1, \ldots, s_{n-k-1}\) are the syndrome digits

where \(e_0, e_1, \ldots, e_{n-1}\) 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
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

\[
\begin{align*}
e_0 &= s_0 \land s'_1 \land s'_2 \\
e_1 &= s_0' \land s_1 \land s'_2 \\
e_2 &= s_0' \land s'_1 \land s_2 \\
e_3 &= s_0 \land s_1 \land s_2' \\
e_4 &= s_0' \land s_1 \land s_2 \\
e_5 &= s_0 \land s_1 \land s_2 \\
e_6 &= s_0' \land s_1' \land s_2 \\
\end{align*}
\]

where \( \land \) denotes the logic-AND operation

where \( s' \) denotes the logic-COMPLEMENT of \( s \)


**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**

<table>
<thead>
<tr>
<th>Syndromes</th>
<th>Correctable error patterns (coset leaders)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
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
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:

\[
A(z) = A_0 + A_1 z + \cdots + A_n z^n
\]

\[
B(z) = B_0 + B_1 z + \cdots + B_n z^n
\] (3.31)

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

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

This identity is known as the MacWilliams identity
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:

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

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

(3.33)
Substituting \( z = \frac{p}{1 - p} \) in \( A(z) \) of (3.31) and using the fact that \( A_0 = 1 \), we obtain

\[
A\left(\frac{p}{1 - p}\right)-1 = \sum_{i=1}^{n} A_i \left(\frac{p}{1 - p}\right)^i
\]  

(3.34)

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

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

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

$$P_u(E) = 2^{-(n-k)} B(1 - 2p) - (1 - p)^n$$  \hspace{1cm} (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).
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

\[
H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

Taking the linear combinations of the row of \(H\), we obtain the following eight vectors in the dual code

\[
(0 \ 0 \ 0 \ 0 \ 0 \ 0), \quad (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1),
\]
\[
(1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1), \quad (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0),
\]
\[
(0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0), \quad (0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1),
\]
\[
(0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1), \quad (1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0)
\]
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.
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.
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

\[
P_u(E) \leq 2^{-(n-k)} \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i}
\]

\[= 2^{-(n-k)} \left[ 1 - (1-p)^n \right] \]  \hspace{1cm} (3.42)

Since \([1 - (1-p)^n] \leq 1\), it is clear that \(P_u(E) \leq 2^{-(n-k)}\).

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**: \( t = 1 (d_{\text{min}} = 3) \)
- **The parity-check matrix** \( \mathbf{H} \) of this code consists of all the nonzero \( m \)-tuple as its columns \((2^m-1)\).
Hamming Codes

In systematic form, the columns of $H$ are arranged in the following form:

$$H = [I_m \ Q]$$

where $I_m$ is an $m \times m$ identity matrix

The submatrix $Q$ consists of $2^m - m - 1$ columns which are the $m$-tuples of weight 2 or more.

The columns of $Q$ 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
  \[ G = [Q^T \ I_{2^m-m-1}] \]
- where \( Q^T \) is the transpose of \( Q \) and \( I_{2^m-m-1} \) 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 \( h_i \) and \( h_j \), must also be a column in \( H \), say \( h_l \)
  \[ h_i + h_j + h_l = 0 \]

- 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 $(2^m - 1)$-tuples 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 0 and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array.
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 5.9).

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: $n = 2^m - l - 1$
  - Number of information symbols: $k = 2^m - m - l - 1$
  - Number of parity-check symbols: $n - k = m$
  - Minimum distance: $d_{\text{min}} \geq 3$
- 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 $m \times 2^{m-1}$ matrix.

$$H' = [I_m \quad Q']$$

- $Q'$ 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 $h_i$ of weight 3 in $Q'$, there exists three columns $h_j$, $h_l$, and $h_s$ in $I_m$ such that $h_i + h_j + h_l + h_s = 0$.
- 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 ($e \times H^T$ corresponds to a column in $H$’)
- 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
  \[ P_u(E) = 2^{-m} \left\{ 1 + (2^m - 1)(1 - 2p)^{2^{m-1}} \right\} - (1 - p)^{2^{m-1}} \]
- The probability $P_u(E)$ for Hamming codes does satisfy the upper bound $2^{-(n-k)} = 2^{-m}$ for $p \leq \frac{1}{2}$ [i.e., $P_u(E) \leq 2^{-m}$]
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 \( u = (u_0, u_1, \ldots, u_{k-1}) \) 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

\[
v = (p, u_0, u_1, \ldots, u_{k-1})
\]

\( p = 1(0) \) if the weight of message \( u \) 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.
The generator of the code in systematic form is given by

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

The parity-check matrix of the code is

\[
H = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
\end{bmatrix}
\]

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

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

From the generator matrices, we see that the $(n,1)$ repetition code and the $(n,n-1)$ 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 $n/2$. $\Rightarrow$ Rate=$1/2$. 
Let $G$ be a generator matrix of a self-dual code $C$. Then, $G$ is also a generator matrix of its dual code $C_d$ and hence is a parity-check matrix of $C$. Consequently,

$$G \cdot G^T = 0$$

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

$$P \cdot P^T = I_{n/2}$$

Conversely, if a rate $\frac{1}{2}$ ($n,n/2$) linear block code $C$ satisfies the condition of $G \cdot G^T = 0$ or $P \cdot P^T = I_{n/2}$, then it is a self-dual code.

Example: the (8,4) linear block code generated by the following matrix has a rate $R = \frac{1}{2}$ and is a self-dual code since $G \cdot G^T = 0$.

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