CHAPTER 3

Linear Block Codes

We assume that the output of an information source is a sequence of the binary digits 0 and 1. In block coding, this binary information sequence is segmented into *message blocks* of fixed length; each message block, denoted by $\mathbf{u}$, consists of k information digits. There are a total of 2^k distinct messages. The encoder, *according to certain rules*, transforms each input message $\mathbf{u}$ into a binary n-tuple $\mathbf{v}$ with $n > k$. This binary n-tuple $\mathbf{v}$ is referred to as the *codeword* (or *code vector*) of the message $\mathbf{u}$. Therefore, corresponding to the 2^k possible messages, there are 2^k codewords. This set of 2^k codewords is called a *block code*. For a block code to be useful, the 2^k codewords must be distinct. Therefore, there should be a one-to-one correspondence between a message $\mathbf{u}$ and its codeword $\mathbf{v}$.

DEFINITION 3.1 A block code of length n and 2^k codewords is called a *linear* (n, k) code if and only if its 2^k codewords form a k-dimensional subspace of the vector space of all the n-tuples over the field $GF(2)$.

TABLE 3.1: Linear block code with $k = 4$ and $n = 7$.

Messages	Codewords
(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)

Because an (n, k) linear code C is a k-dimensional subspace of the vector space V_n of all the binary n-tuples, it is possible to find k linearly independent codewords, $\mathbb{g}_0, \mathbb{g}_1, \cdots, \mathbb{g}_{k-1}$, in C such that every codeword $\mathbb{v}$ in C is a linear combination of these k codewords; that is,

$$\mathbb{v} = u_0\mathbb{g}_0 + u_1\mathbb{g}_1 + \cdots + u_{k-1}\mathbb{g}_{k-1}, \tag{3.1}$$

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

$$
\begin{aligned}
\mathbb{v} &= \mathbb{u} \cdot G \\
&= (u_0, u_1, \cdots, u_{k-1}) \cdot \begin{bmatrix} \mathbb{g}_0 \\ \mathbb{g}_1 \\ \vdots \\ \mathbb{g}_{k-1} \end{bmatrix} \qquad (3.3) \\
&= u_0\mathbb{g}_0 + u_1\mathbb{g}_1 + \cdots + u_{k-1}\mathbb{g}_{k-1}.
\end{aligned}
$$

Clearly, the rows of $\mathbb{G}$ *generate* (or *span*) the (n, k) linear code C. For this reason, the matrix $\mathbb{G}$ is called a *generator* matrix for C. Note that any k linearly independent codewords 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 $\mathbb{G}$. Therefore, the encoder has only to store the k rows of $\mathbb{G}$ and to form a linear combination of these k rows based on the input message $\mathbb{u} = (u_0, u_1, \cdots, u_{k-1})$.

EXAMPLE 3.1

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

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

If $\mathbf{u} = (1\ 1\ 0\ 1)$ is the message to be encoded, its corresponding codeword, according to (3.3), will be

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

TABLE 3.1: Linear block code with $k = 4$ and $n = 7$.

Messages	Codewords
(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)

A desirable property for a linear block code to possess is the *systematic structure* of the codewords, as shown in Figure 3.1, in which a codeword is divided into two parts, the message part and the redundant checking part. The message part consists of k unaltered information (or message) digits, and the redundant checking part consists of $n - k$ *parity-check* digits, which are *linear sums* of the information digits. A linear block code with this structure is referred to as a *linear systematic block code*. The (7, 4) code given in Table 3.1 is a linear systematic block

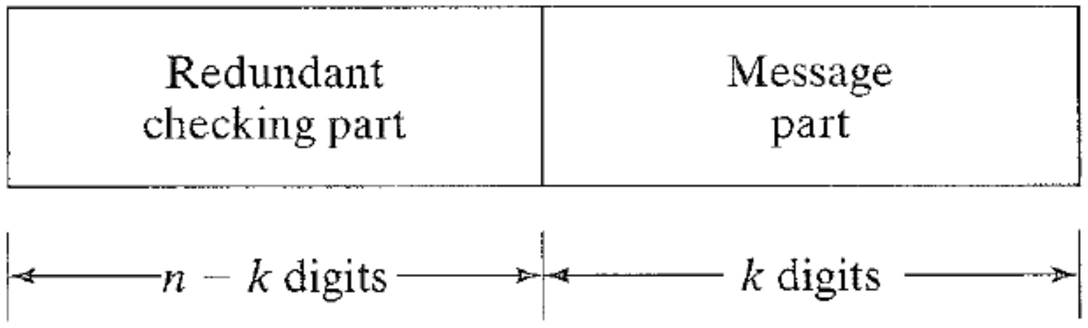

FIGURE 3.1: Systematic format of a codeword.

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

$$\mathbb{G} = \begin{bmatrix} \mathbb{g}_0 \\ \mathbb{g}_1 \\ \mathbb{g}_2 \\ \vdots \\ \mathbb{g}_{k-1} \end{bmatrix} = \Bigg[\overbrace{\begin{matrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ p_{20} & p_{21} & \cdots & p_{2,n-k-1} \\ & & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{matrix}}^{p \text{ matrix}} \;\vdots\; \overbrace{\begin{matrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & & \\ 0 & 0 & 0 & \cdots & 1 \end{matrix}}^{\substack{k \times k \\ \text{identity matrix}}} \Bigg]. \quad (3.4)$$

where $p_{ij} = 0$ or 1. Let $\mathbb{I}_k$ denote the $k \times k$ identity matrix. Then, $\mathbb{G} = [\mathbb{P}\ \mathbb{I}_k]$. Let $\mathbb{u} = (u_0, u_1, \cdots, u_{k-1})$ be the message to be encoded. The corresponding codeword is

$$\begin{aligned} \mathbb{v} &= (v_0, v_1, v_2, \cdots, v_{n-1}) \\ &= (u_0, u_1, \cdots, u_{k-1}) \cdot \mathbb{G}. \end{aligned} \quad (3.5)$$

It follows from (3.4) and (3.5) that the components of $\mathbf{v}$ are

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

and

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} \tag{3.6b}$$

for $0 \leq j < n - k$. Equation (3.6a) shows that the rightmost k digits of a codeword $\mathbf{v}$ are identical to the information digits $u_0, u_1, \cdots, u_{k-1}$ to be encoded, and (3.6b) shows that the leftmost $n - k$ redundant 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 $\mathbb{G}$ given in Example 3.1 is in systematic form. Let $\mathbf{u} = (u_0, u_1, u_2, u_3)$ be the message to be encoded, and let $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding codeword. Then,

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

By matrix multiplication, we obtain the following digits of the codeword $\mathbf{v}$:

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

The codeword corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1).

There is another useful matrix associated with every linear block code. As stated in Chapter 2, for any $k \times n$ matrix $\mathbb{G}$ with k linearly independent rows, there exists an $(n-k) \times n$ matrix $\mathbb{H}$ with $n-k$ linearly independent rows such that any vector in the row space of $\mathbb{G}$ is orthogonal to the rows of $\mathbb{H}$, and any vector that is orthogonal to the rows of $\mathbb{H}$ is in the row space of $\mathbb{G}$. Hence, we can describe the (n, k) linear code C generated by $\mathbb{G}$ in an alternative way as follows: *An n-tuple* $\mathbf{v}$ *is a codeword in the code C generated by* $\mathbb{G}$ *if and only if* $\mathbf{v} \cdot \mathbb{H}^{\mathrm{T}} = \mathbb{0}$. The code is said to be the null space of $\mathbb{H}$. This matrix $\mathbb{H}$ is called a *parity-check matrix* of the code. The 2^{n-k} linear combinations of the rows of matrix $\mathbb{H}$ 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 $\mathbb{G}$ (i.e., for any $\mathbf{v} \in C$ and any $\mathbf{w} \in C_d$, $\mathbf{v} \cdot \mathbf{w} = 0$). C_d is called the *dual code* of C. Therefore, a parity-check matrix for a linear code C is a generator matrix for its dual code C_d.

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:

$$\mathbb{H} = [\mathbb{I}_{n-k}\mathbb{P}^T] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}. \tag{3.7}$$

where $\mathbb{P}^T$ is the transpose of the matrix $\mathbb{P}$. Let $\mathbb{h}_j$ be the jth row of $\mathbb{H}$. We can readily check that the inner product of the ith row of $\mathbb{G}$ given by (3.4) and the jth row of $\mathbb{H}$ given by (3.7) is

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

for $0 \leq i < k$ and $0 \leq j < n - k$. This implies that $\mathbb{G} \cdot \mathbb{H}^T = \mathbb{0}$. Also, the $n - k$ rows of $\mathbb{H}$ are linearly independent. Therefore, the $\mathbb{H}$ matrix of (3.7) is a parity-check matrix of the (n, k) linear code generated by the matrix $\mathbb{G}$ of (3.4).

The parity-check equations given by (3.6b) also can be obtained from the parity-check matrix $\mathbb{H}$ of (3.7). Let $\mathbb{u} = (u_0, u_1, \cdots, u_{k-1})$ be the message to be encoded. In systematic form the corresponding codeword will be

$$\mathbb{v} = (v_0, v_1, \cdots, v_{n-k-1}, u_0, u_1, \cdots, u_{k-1}).$$

Using the fact that $\mathbb{v} \cdot \mathbb{H}^T = \mathbb{0}$, we obtain

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

for $0 \leq j < n - k$. Rearranging the equations of (3.8), we obtain the same parity-check equations of (3.6b). Therefore, an (n, k) linear code is completely specified by its parity-check matrix.

EXAMPLE 3.3

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

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

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

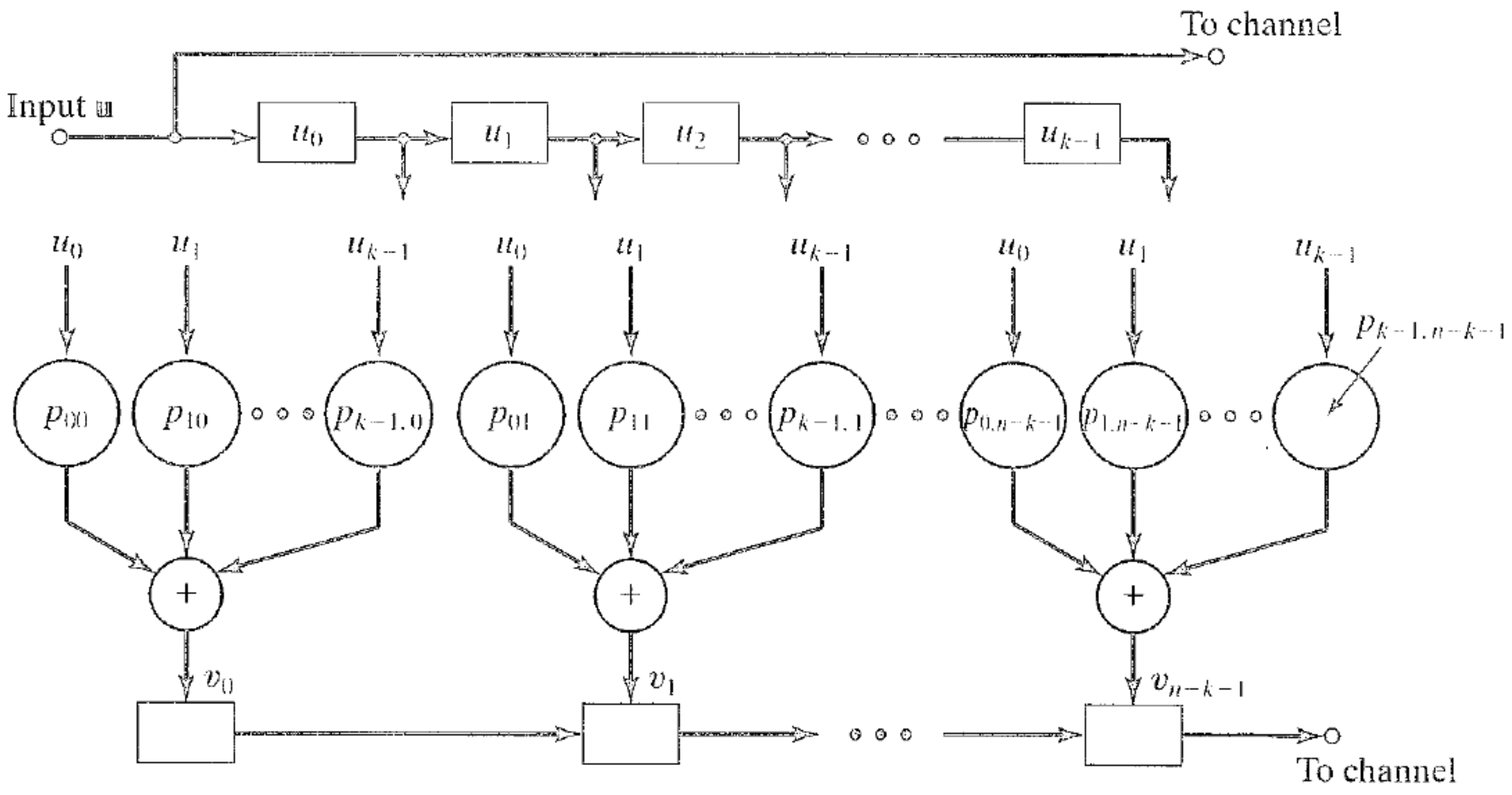

FIGURE 3.2: Encoding circuit for a linear systematic (n, k) code.

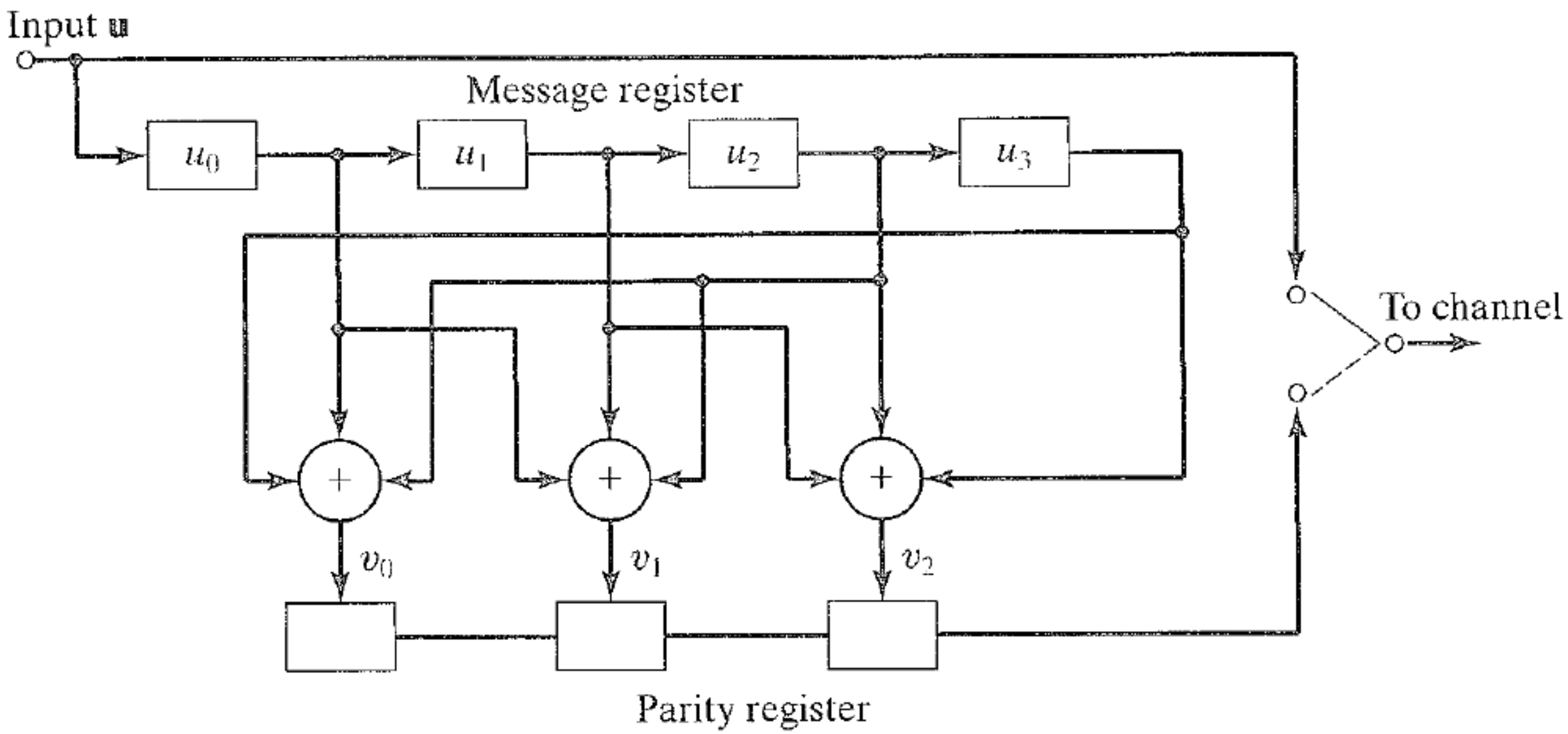

FIGURE 3.3: Encoding circuit for the (7, 4) systematic code given in Table 3.1.

SYNDROME AND ERROR DETECTION

Let $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ be a codeword that was transmitted over a noisy channel. Let $\mathbf{r} = (r_0, r_1, \cdots, r_{n-1})$ be the received vector at the output of the channel. Because of the channel noise, $\mathbf{r}$ may be different from $\mathbf{v}$. The vector sum

$$\begin{aligned} \mathbf{e} &= \mathbf{r} + \mathbf{v} \\ &= (e_0, e_1, \cdots, e_{n-1}) \end{aligned} \tag{3.9}$$

is an n-tuple, where $e_i = 1$ for $r_i \neq v_i$, and $e_i = 0$ for $r_i = v_i$. This n-tuple is called the *error vector* (or *error pattern*), which simply displays the positions where the received vector $\mathbf{r}$ differ from the transmitted codeword $\mathbf{v}$. The 1's in $\mathbf{e}$ are the *transmission errors* caused by the channel noise. It follows from (3.9) that the received vector $\mathbf{r}$ is the vector sum of the transmitted codeword and the error vector; that is,

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

When $\mathbb{r}$ is received, the decoder computes the following $(n-k)$-tuple:

$$\begin{aligned} \mathbb{s} &= \mathbb{r} \cdot \mathbb{H}^T \\ &= (s_0, s_1, \cdots, s_{n-k-1}). \end{aligned} \tag{3.10}$$

which is called the *syndrome* of $\mathbb{r}$. Then, $\mathbb{s} = \mathbb{0}$ if and only if $\mathbb{r}$ is a codeword, and $\mathbb{s} \neq \mathbb{0}$ if and only if $\mathbb{r}$ is not a codeword. Therefore, when $\mathbb{s} \neq \mathbb{0}$, we know that $\mathbb{r}$ is not a codeword and the presence of errors has been detected. When $\mathbb{s} = \mathbb{0}$, $\mathbb{r}$ is a codeword, and the receiver accepts $\mathbb{r}$ as the transmitted codeword. It is possible that the errors in certain error vectors are not detectable (i.e., $\mathbb{r}$ contains errors but $\mathbb{s} = \mathbb{r} \cdot \mathbb{H}^T = \mathbb{0}$).

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

$$\mathbb{s} = \mathbb{e} \cdot \mathbb{H}^T.$$

This happens when the error pattern $\mathbb{e}$ is identical to a nonzero codeword. In this event, $\mathbb{r}$ is the sum of two codewords, which is a codeword, and consequently $\mathbb{r} \cdot \mathbb{H}^T = \mathbb{0}$. Error patterns of this kind are called *undetectable* error patterns. Because there are $2^k - 1$ nonzero codewords, there are $2^k - 1$ undetectable error patterns. When an undetectable error pattern occurs, the decoder makes a *decoding error*. In a later section of this chapter we derive the probability of an undetected error for a BSC and show that this error probability can be made very small.

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

$$\begin{aligned}
s_0 &= r_0 + r_{n-k}p_{00} + r_{n-k+1}p_{10} + \cdots + r_{n-1}p_{k-1,0} \\
s_1 &= r_1 + r_{n-k}p_{01} + r_{n-k+1}p_{11} + \cdots + r_{n-1}p_{k-1,1} \\
&\vdots \\
s_{n-k-1} &= r_{n-k-1} + r_{n-k}p_{0,n-k-1} + r_{n-k+1}p_{1,n-k-1} + \cdots + r_{n-1}p_{k-1,n-k-1}.
\end{aligned} \tag{3.11}$$

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

EXAMPLE 3.4

Consider the (7, 4) linear code whose parity-check matrix is given in Example 3.3. Let $\mathbf{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector. Then, the syndrome is given by

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

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 in Figure 3.5.

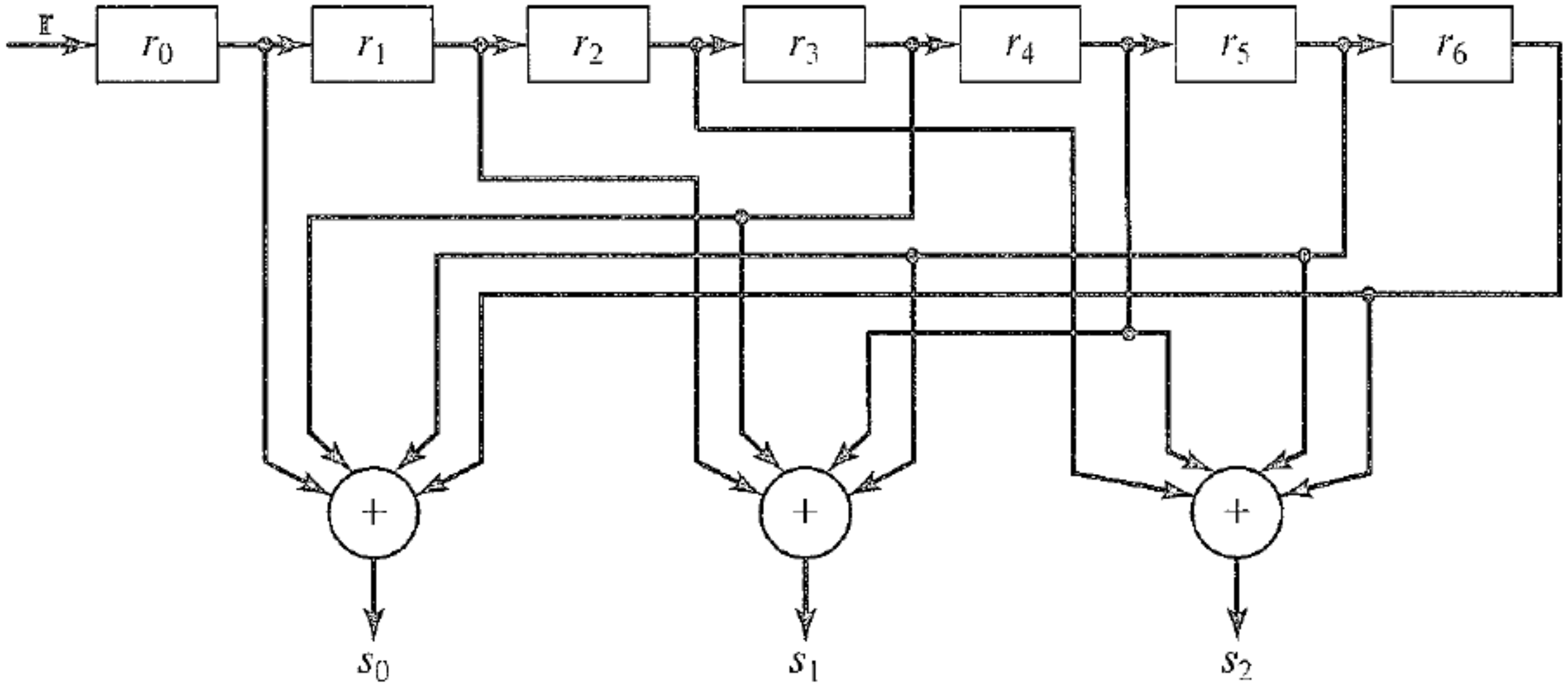

FIGURE 3.5: Syndrome circuit for the (7, 4) code given in Table 3.1.

At this point, one would think that any error correction scheme would be a method of solving the $n-k$ linear equations of (3.13) for the error digits. Once the error pattern $\mathbb{e}$ was found, the vector $\mathbb{r}+\mathbb{e}$ would be taken as the actual transmitted codeword. Unfortunately, determining the true error vector $\mathbb{e}$ is not a simple matter. This is because the $n-k$ linear equations of (3.13) do not have a unique solution but have 2^k solutions (this will be proved in Theorem 3.6). In other words, there are 2^k error patterns that result in the same syndrome, and the true error pattern $\mathbb{e}$ is just one of them. Therefore, 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. If the channel is a BSC, the most probable error pattern is the one that has the smallest number of nonzero digits.

EXAMPLE 3.5

Again, 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 codeword and $\mathbf{r} = (1\,0\,0\,1\,0\,0\,1)$ be the received vector. On receiving $\mathbf{r}$, the receiver computes the syndrome:

$$\mathbf{s} = \mathbf{r} \cdot \mathbb{H}^T = (1\,1\,1).$$

Next, 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 given syndrome. It follows from (3.12) or (3.13) that the error digits are related to the syndrome digits by the following linear equations:

$$1 = e_0 + e_3 + e_5 + e_6$$

$$1 = e_1 + e_3 + e_4 + e_5$$

$$1 = e_2 + e_4 + e_5 + e_6.$$

There are $2^4 = 16$ error patterns that satisfy the preceding equations, namely,

$$\begin{array}{ll}
(0\,0\,0\,0\,0\,1\,0), & (1\,0\,1\,0\,0\,1\,1), \\
(1\,1\,0\,1\,0\,1\,0), & (0\,1\,1\,1\,0\,1\,1), \\
(0\,1\,1\,0\,1\,1\,0), & (1\,1\,0\,0\,1\,1\,1), \\
(1\,0\,1\,1\,1\,1\,0), & (0\,0\,0\,1\,1\,1\,1), \\
(1\,1\,1\,0\,0\,0\,0), & (0\,1\,0\,0\,0\,0\,1), \\
(0\,0\,1\,1\,0\,0\,0), & (1\,0\,0\,1\,0\,0\,1), \\
(1\,0\,0\,0\,1\,0\,0), & (0\,0\,1\,0\,1\,0\,1), \\
(0\,1\,0\,1\,1\,0\,0), & (1\,1\,1\,1\,1\,0\,1).
\end{array}$$

The error vector $\mathbf{e} = (0\,0\,0\,0\,0\,1\,0)$ has the smallest number of nonzero components. If the channel is a BSC, $\mathbf{e} = (0\,0\,0\,0\,0\,1\,0)$ is the most probable error vector that satisfies the preceding equations. Taking $\mathbf{e} = (0\,0\,0\,0\,0\,1\,0)$ as the true error vector, the

receiver decodes the received vector $\mathbf{r} = (1\,0\,0\,1\,0\,0\,1)$ into the following codeword:

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

THE MINIMUM DISTANCE OF A BLOCK CODE

The *Hamming weight* (or simply *weight*) of $\mathbf{v}$, denoted by $w(\mathbf{v})$

The *Hamming* *distance* (or simply *distance*) between $\mathbf{v}$ and $\mathbf{w}$, denoted $d(\mathbf{v}, \mathbf{w})$.

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

The Hamming distance is a metric function that satisfies the *triangle inequality*. Let $\mathbf{v}$, $\mathbf{w}$, and $\mathbf{x}$ be three n-tuples. Then,

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

Given a block code C, one can compute the Hamming distance between any two distinct codewords. The *minimum distance* of C, denoted by d_{min}, is defined as

$$d_{min} \triangleq \min\{d(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in C, \mathbf{v} \neq \mathbf{w}\}. \tag{3.16}$$

THEOREM 3.1 The minimum distance of a linear block code is equal to the minimum weight of its nonzero codewords and vice versa.

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

THEOREM 3.2 Let C be an (n, k) linear code with parity-check matrix $\mathbb{H}$. For each codeword of Hamming weight l, there exist l columns of $\mathbb{H}$ such that the vector sum of these l columns is equal to the zero vector. Conversely, if there exist l columns of $\mathbb{H}$ whose vector sum is the zero vector, there exists a codeword of Hamming weight l in C.

Corollary 3.2.1 Let C be a linear block code with parity-check matrix $\mathbb{H}$. If no $d-1$ or fewer columns of $\mathbb{H}$ add to $\mathbb{0}$, the code has minimum weight at least d.

Corollary 3.2.2 Let C be a linear code with parity-check matrix $\mathbb{H}$. The minimum weight (or the minimum distance) of C is equal to the smallest number of columns of $\mathbb{H}$ that sum to $\mathbb{0}$.

Consider the (7, 4) linear code given in Table 3.1. The parity-check matrix of this code is

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

ERROR-DETECTING AND ERROR-CORRECTING CAPABILITIES OF A BLOCK CODE

Let C be an (n, k) linear code. Let A_i be the number of codewords of weight i in C. The numbers $A_0, A_1, \cdots, A_n$ are called the *weight distribution of C*. If C is used only for error detection on a BSC, the probability that the decoder will fail to detect the presence of errors can be computed from the weight distribution of C. Let $P_u(E)$ denote the probability of an undetected error. Because an undetected error occurs only when the error pattern is identical to a nonzero codeword of C,

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}, \tag{3.19}$$

where p is the transition probability of the BSC. If the minimum distance of C is d_{min}, then A_1 to $A_{d_{min}-1}$ are zero.

Consider the $(7, 4)$ code given in Table 3.1. The weight distribution of this code is $A_0 = 1$, $A_1 = A_2 = 0$, $A_3 = 7$, $A_4 = 7$, $A_5 = A_6 = 0$, and $A_7 = 1$. The probability of an undetected error is

$$P_u(E) = 7p^3(1-p)^4 + 7p^4(1-p)^3 + p^7.$$

If $p = 10^{-2}$, this probability is approximately 7×10^{-6}. In other words, if 1 million codewords are transmitted over a BSC with $p = 10^{-2}$, on average seven erroneous codewords pass through the decoder without being detected.

If a block code C with minimum distance d_{min} is used for random-error correction, one would like to know how many errors the code is able to correct. The minimum distance d_{min} is either odd or even. Let t be a positive integer such that

$$2t + 1 \leq d_{min} \leq 2t + 2. \tag{3.20}$$

Next, we show that the code C is capable of correcting all the error patterns of t or fewer errors. Let $\mathbb{v}$ and $\mathbb{r}$ be the transmitted codeword and the received vector, respectively. Let $\mathbb{w}$ be any other codeword in C. The Hamming distances among $\mathbb{v}$, $\mathbb{r}$, and $\mathbb{w}$ satisfy the triangle inequality:

$$d(\mathbb{v}, \mathbb{r}) + d(\mathbb{w}, \mathbb{r}) \geq d(\mathbb{v}, \mathbb{w}). \tag{3.21}$$

Suppose that an error pattern of t' errors occurs during the transmission of $\mathbb{v}$. Then, the received vector $\mathbb{r}$ differs from $\mathbb{v}$ in t' places, and therefore $d(\mathbb{v}, \mathbb{r}) = t'$. Because $\mathbb{v}$ and $\mathbb{w}$ are codewords in C, we have

$$d(\mathbb{v}, \mathbb{w}) \geq d_{min} \geq 2t + 1. \tag{3.22}$$

Combining (3.21) and (3.22) and using the fact that $d(\mathbb{v}, \mathbb{r}) = t'$, we obtain the following inequality:

$$d(\mathbb{w}, \mathbb{r}) \geq 2t + 1 - t'.$$

If $t' \leq t$,

$$d(\mathbb{w}, \mathbb{r}) > t.$$

The preceding inequality says that if an error pattern of t or fewer errors occurs, the received vector $\mathbb{r}$ is closer (in Hamming distance) to the transmitted codeword $\mathbb{v}$ than to any other codeword $\mathbb{w}$ in C. For a BSC, this means that the conditional probability $P(\mathbb{r}|\mathbb{v})$ is greater than the conditional probability $P(\mathbb{r}|\mathbb{w})$ for $\mathbb{w} \neq \mathbb{v}$. Based on the maximum likelihood decoding scheme, $\mathbb{r}$ is decoded into $\mathbb{v}$, which is the actual transmitted codeword. The result is a correct decoding, and thus errors are corrected.

In contrast, the code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case in which an error pattern of l errors results in a received vector that is closer to an incorrect codeword than to the transmitted codeword. To show this, let $\mathbf{v}$ and $\mathbf{w}$ be two codewords 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:

i. $\mathbf{e}_1 + \mathbf{e}_2 = \mathbf{v} + \mathbf{w}$.
ii. $\mathbf{e}_1$ and $\mathbf{e}_2$ do not have nonzero components in common places.

Obviously, we have

$$w(\mathbf{e}_1) + w(\mathbf{e}_2) = w(\mathbf{v} + \mathbf{w}) = d(\mathbf{v}, \mathbf{w}) = d_{min}. \tag{3.23}$$

Now, 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

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

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

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

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

$$w(\mathbf{e}_2) \leq t + 1.$$

Combining (3.24) and (3.25) and using the fact that $w(\mathbf{e}_1) > t$ and $w(\mathbf{e}_2) \leq t + 1$, we obtain the following inequality:

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

This inequality says that there exists an error pattern of $l(l > t)$ errors that results in a received vector that is closer to an incorrect codeword than to the transmitted codeword. Based on the maximum likelihood decoding scheme, an incorrect decoding would be performed.

In summary, a block code with minimum distance $d_{\min}$ *guarantees* correction of 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. The (7, 4) code given in Table 3.1 has minimum distance 3 and thus $t = 1$. The code is capable of correcting any error pattern of single error over a block of seven digits.

A block code with random-error-correcting capability t is usually capable of correcting many error patterns of $t+1$ or more errors. A t-error-correcting (n, k) linear code is capable of correcting a total of 2^{n-k} error patterns, including those with t or fewer errors (this will be seen in the next section). 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

$$P(E) \leq \sum_{i=t+1}^{n} \binom{n}{i} p^{i}(1-p)^{n-i}. \tag{3.26}$$

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. For this purpose, the minimum distance d_{min} of the code is at least $\lambda + l + 1$ (left as a problem). Thus, a block code with $d_{min} = 10$ is capable of correcting three or fewer errors and simultaneously detecting six or fewer errors.

STANDARD ARRAY AND SYNDROME DECODING

In this section we present a scheme for decoding linear block codes. Let C be an (n, k) linear code. Let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the codewords of C. No matter which codeword is transmitted over a noisy channel, the received vector $\mathbf{r}$ may be any of the 2^n n-tuples over $GF(2)$. Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \cdots, D_{2^k}$ such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$. Thus, each subset D_i is one-to-one correspondence to a codeword $\mathbf{v}_i$. If the received vector $\mathbf{r}$ is found in the subset D_i, $\mathbf{r}$ is decoded into $\mathbf{v}_i$. Decoding is correct if and only if the received vector $\mathbf{r}$ is in the subset D_i that corresponds to the codeword 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 codeword is described here. The partition is based on the linear structure of the code. First, we place the 2^k codewords of C in a row with the all-zero codeword $\mathbf{v}_1 = (0, 0, \cdots, 0)$ as the first (leftmost) element. From the remaining $2^n - 2^k$ n-tuples, we choose an n-tuple $\mathbf{e}_2$ and place it under the zero vector $\mathbf{v}_1$. Now, we form a second row by adding $\mathbf{e}_2$ to each codeword $\mathbf{v}_i$ in the first row and placing the sum $\mathbf{e}_2 + \mathbf{v}_i$ under $\mathbf{v}_i$. Having completed the second row, we choose an unused n-tuple $\mathbf{e}_3$ from the remaining n-tuples and place it under $\mathbf{v}_1$. Then, we form a third row by adding $\mathbf{e}_3$ to each codeword $\mathbf{v}_i$ in the first row and placing $\mathbf{e}_3 + \mathbf{v}_i$ under $\mathbf{v}_i$. We continue this process until we have used all the n-tuples. Then, we have an array of rows and columns, as shown in Figure 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.

From Theorem 3.3 we see that there are $2^n/2^k = 2^{n-k}$ disjoint rows in the standard array, and that each row consists of 2^k distinct elements. The 2^{n-k} rows are called the *cosets* of the code C, and the first n-tuple $\mathbf{e}_j$ of each coset is called a *coset leader* (or coset representative). The coset concept for a subgroup was presented in Section 2.1. Any element in a coset can be used as its coset leader. This does not change the elements of the coset; it simply permutes them.

EXAMPLE 3.6

Consider the (6, 3) linear code generated by the following matrix:

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

The standard array of this code is shown in Figure 3.7.

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 a (6, 3) code.

THEOREM 3.4 Every (n, k) linear block code is capable of correcting 2^{n-k} error patterns.

A standard array of an (n, k) linear code C consists of 2^k disjoint columns. Each column consists of 2^{n-k} n-tuples, with the topmost one as a codeword in C. Let D_j denote the jth column of the standard array. Then,

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}, \tag{3.27}$$

where $\mathbf{v}_j$ is a codeword of C, and $\mathbf{e}_2, \mathbf{e}_3, \cdots, \mathbf{e}_{2^{n-k}}$ are the coset leaders. The 2^k disjoint columns $D_1, D_2, \cdots, D_{2^k}$ can be used for decoding the code C as described earlier in this section. Suppose that the codeword $\mathbf{v}_j$ 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. In this event, the received vector $\mathbf{r}$ will be decoded correctly into the transmitted codeword $\mathbf{v}_j$; however, if the error pattern caused by the channel is not a coset leader, an erroneous decoding will result. This can be seen as follows. The error pattern $\mathbf{x}$ caused by the channel must be in some coset and under some nonzero codeword, say in the lth coset and under the codeword $\mathbf{v}_i \neq \mathbf{0}$. Then, $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$, and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + (\mathbf{v}_i + \mathbf{v}_j) = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector $\mathbf{r}$ is thus in D_s and is decoded into $\mathbf{v}_s$, which is not the transmitted codeword. This results in an erroneous decoding. Therefore, the decoding is correct if and only if the error pattern caused by the channel is a coset leader. For this reason, the 2^{n-k} coset leaders (including the zero vector $\mathbf{0}$) are called the *correctable error patterns*. Summarizing the preceding results, we have the following theorem:

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. For a BSC, an error pattern of smaller weight is more probable than an error pattern of larger weight. Therefore, when a standard array is formed, each coset leader should be chosen to be a vector of *least weight* from the remaining available vectors. If coset leaders are chosen in this manner, each coset leader has minimum weight in its coset. As a result, the decoding based on the standard array is the minimum distance decoding (i.e., the maximum likelihood decoding). To see this, let $\mathbf{r}$ be the received vector. Suppose that $\mathbf{r}$ is found in the ith column D_i and lth coset of the standard array. Then, $\mathbf{r}$ is decoded into the codeword $\mathbf{v}_i$. Because $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, the distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l). \tag{3.28}$$

Now, consider the distance between $\mathbf{r}$ and any other codeword, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j).$$

Because $\mathbf{v}_i$ and $\mathbf{v}_j$ are two different codewords, their vector sum, $\mathbf{v}_i + \mathbf{v}_j$, is a nonzero codeword, say $\mathbf{v}_s$. Thus,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s). \tag{3.29}$$

Because $\mathbf{e}_l$ and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows from (3.28) and (3.29) that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j).$$

This says that the received vector is decoded into a closest codeword. Hence, if each coset leader is chosen to have minimum weight in its coset, the decoding based on the standard array is the minimum distance decoding, or MLD.

Let α_i denote the number of coset leaders of weight i. The numbers $\alpha_0, \alpha_1, \cdots, \alpha_n$ are called the *weight distribution* of the coset leaders. Knowing these numbers, we can compute the probability of a decoding error. Because 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

$$P(E) = 1 - \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-i}. \tag{3.30}$$

EXAMPLE 3.7

Consider the (6, 3) code given in Example 3.6. The standard array for this code is shown in Figure 3.7. The weight distribution of the coset leaders 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 $P(E) \approx 1.37 \times 10^{-3}$.

THEOREM 3.5 For an (n, k) linear code C with minimum distance $d_{\min}$, all the n-tuples of weight $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or less can be used as coset leaders of a standard array of C. If all the n-tuples of weight t or less are used as coset leaders, there is at least one n-tuple of weight $t + 1$ that cannot be used as a coset leader.

Theorem 3.5 reconfirms the fact that an (n, k) linear code with minimum distance $d_{\min}$ is capable of correcting all the error patterns 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 syndromes for different cosets are different.

We recall that the syndrome of an n-tuple is an $(n-k)$-tuple, and there are 2^{n-k} distinct $(n-k)$-tuples. It follows from Theorem 3.6 that there is a one-to-one correspondence between a coset and an $(n-k)$-tuple syndrome; or there is a one-to-one correspondence between a coset leader (a correctable error pattern) and a 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 patterns) and their corresponding syndromes. This table is either stored or wired in the receiver. The decoding of a received vector consists of three steps:

1. Compute the syndrome of $\mathbf{r}$, $\mathbf{r} \cdot \mathbf{H}^T$.
2. Locate the coset leader $\mathbf{e}_l$ whose syndrome is equal to $\mathbf{r} \cdot \mathbf{H}^T$. Then $\mathbf{e}_l$ is assumed to be the error pattern caused by the channel.
3. Decode the received vector $\mathbf{r}$ into the codeword $\mathbf{v}^* = \mathbf{r} + \mathbf{e}_l$.

The described decoding scheme is called the *syndrome decoding* or *table-lookup decoding*. In principle, table-lookup decoding can be applied to any (n, k) linear code. It results in minimum decoding delay and minimum error probability;

however, for large $n-k$, the implementation of this decoding scheme becomes impractical, and either a large storage or a complicated logic circuitry is needed. Several practical decoding schemes that are variations of table-lookup decoding are discussed in subsequent chapters. Each of these decoding schemes requires additional properties of a code other than the linear structure.

EXAMPLE 3.8

Consider the (7, 4) linear code given in Table 3.1. The parity-check matrix, as given in Example 3.3, is

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

The code has $2^3 = 8$ cosets, and therefore there are eight correctable error patterns (including the all-zero vector). Because the minimum distance of the code is 3, it is capable of correcting all the error patterns of weight 1 or 0. Hence, all the 7-tuples of weight 1 or 0 can be used as coset leaders. There are $\binom{7}{0} + \binom{7}{1} = 8$ such vectors. We see that for the (7, 4) linear code considered in this example, the number of correctable error patterns 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 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)

Suppose that the codeword $\mathbb{v} = (1 0 0 1 0 1 1)$ is transmitted, and $\mathbb{r} = (1 0 0 1 1 1 1)$ is received. For decoding $\mathbb{r}$, we compute the syndrome of $\mathbb{r}$:

$$\mathbb{s} = (1\ 0\ 0\ 1\ 1\ 1\ 1)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = (0\ 1\ 1).$$

From Table 3.2 we find that (0 1 1) is the syndrome of the coset leader $\mathbf{e} = (0000100)$. Thus, (0000100) is assumed to be the error pattern caused by the channel, and $\mathbf{r}$ is decoded into

$$\begin{aligned}\mathbf{v}^* &= \mathbf{r} + \mathbf{e}\\ &= (1001111) + (0000100)\\ &= (1001011),\end{aligned}$$

which is the codeword transmitted. The decoding is correct, since the error pattern caused by the channel is a coset leader.

Now, suppose that $\mathbf{v} = (0000000)$ is transmitted, and $\mathbf{r} = (1000100)$ 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:

$$\mathbf{s} = \mathbf{r} \cdot \mathbb{H}^T = (111).$$

From the decoding table we find that the coset leader $\mathbf{e} = (0000010)$ corresponds to the syndrome $\mathbf{s} = (111)$. As a result, $\mathbf{r}$ is decoded into the codeword

$$\begin{aligned}\mathbf{v}^* &= \mathbf{r} + \mathbf{e}\\ &= (1000100) + (0000010)\\ &= (1000110).\end{aligned}$$

Because $\mathbf{v}^*$ is not the codeword transmitted, a decoding error is committed.

Using Table 3.2, we see that 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 switching functions:

$$\begin{aligned}
e_0 &= f_0(s_0, s_1, \cdots, s_{n-k-1}), \\
e_1 &= f_1(s_0, s_1, \cdots, s_{n-k-1}), \\
&\vdots \\
e_{n-1} &= f_{n-1}(s_0, s_1, \cdots, s_{n-k-1}),
\end{aligned}$$

where $s_0, s_1, \cdots, s_{n-k-1}$ are the syndrome digits, which are regarded as switching variables, and $e_0, e_1, \cdots, e_{n-1}$ are the estimated error digits. When these n switching functions are derived and simplified, a combinational logic circuit with the $n - k$ syndrome digits as inputs and the estimated error digits as outputs can be realized. The implementation of the syndrome circuit was discussed in Section 3.2. The general decoder for an (n, k) linear code based on the table-lookup scheme is shown

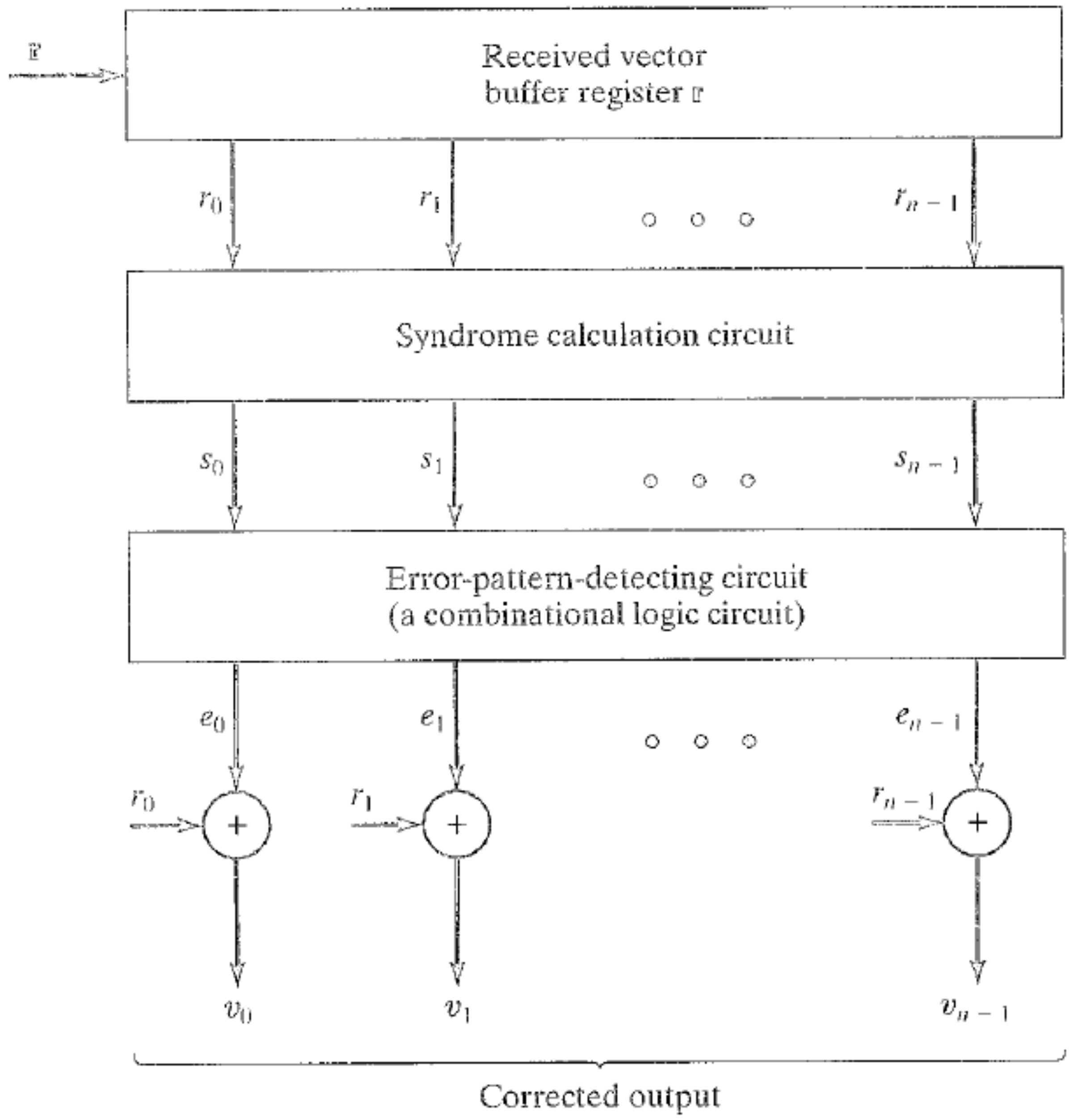
r
Received vector buffer register r
r_0
r_1
r_{n-1}
Syndrome calculation circuit
s_0
s_1
s_{n-1}
Error-pattern-detecting circuit (a combinational logic circuit)
e_0
e_1
e_{n-1}
r_0
r_1
r_{n-1}
v_0
v_1
v_{n-1}
Corrected output

FIGURE 3.8: General decoder for a linear block code.

EXAMPLE 3.9

Again, we consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Figure 3.5. The decoding table is given by Table 3.2. From this table we form the truth table (Table 3.3). The switching expressions for the seven error digits are

$$e_0 = s_0 \Lambda s_1' \Lambda s_2', \qquad e_1 = s_0' \Lambda s_1 \Lambda s_2',$$

$$e_2 = s_0' \Lambda s_1' \Lambda s_2, \qquad e_3 = s_0 \Lambda s_1 \Lambda s_2',$$

$$e_4 = s_0' \Lambda s_1 \Lambda s_2, \qquad e_5 = s_0 \Lambda s_1 \Lambda s_2,$$

$$e_6 = s_0 \Lambda s_1' \Lambda s_2,$$

where Λ denotes the logic-AND operation and s' denotes the logic-COMPLEMENT of s. These seven switching expressions can be realized by seven 3-input AND gates. The complete circuit of the decoder is shown in Figure 3.9.

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

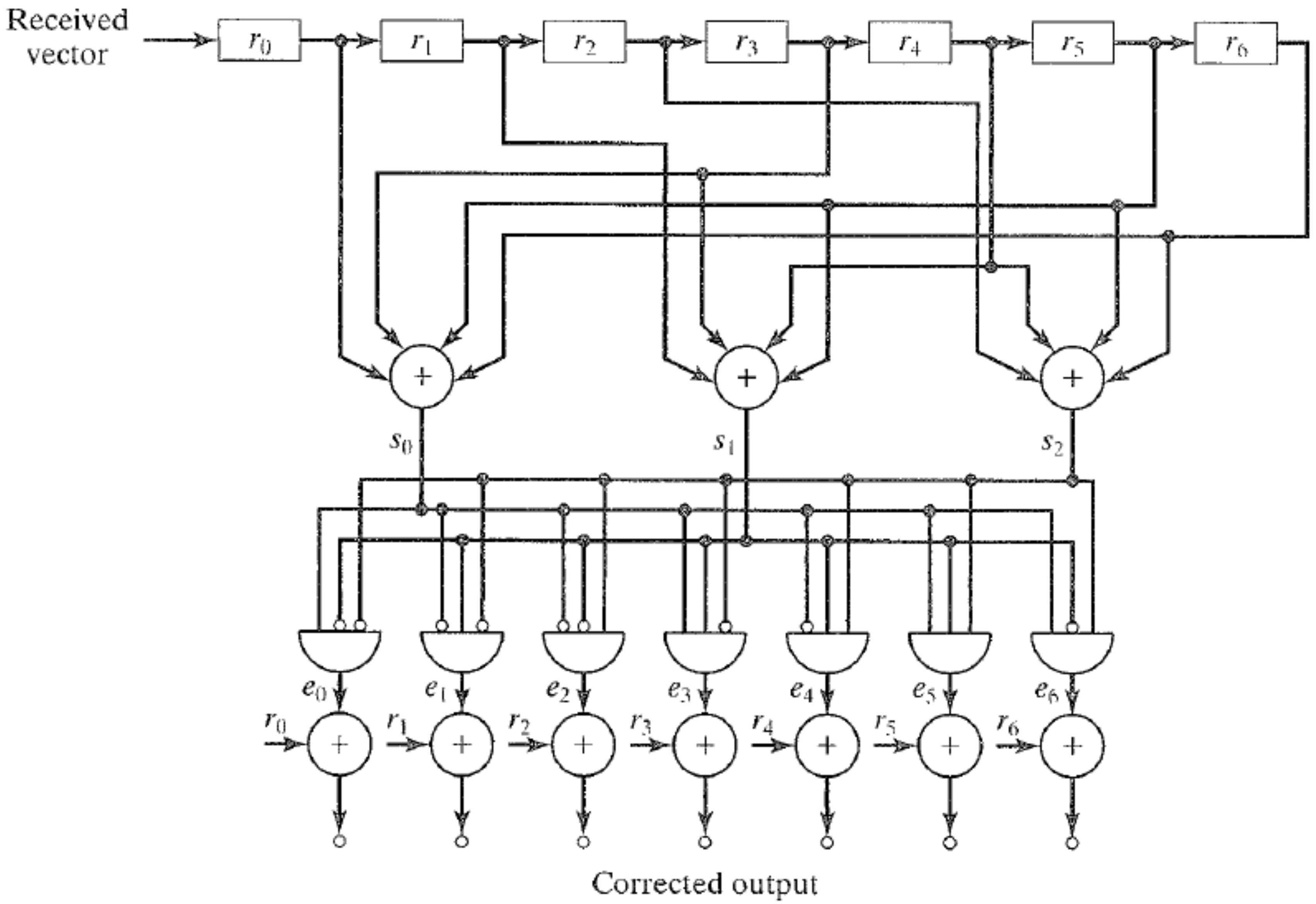

FIGURE 3.9: Decoding circuit for the (7, 4) code given in Table 3.1.

PROBABILITY OF AN UNDETECTED ERROR FOR LINEAR CODES OVER A BSC

If an (n, k) linear code is used only for error detection over a BSC, the probability of an undetected error $P_u(E)$ can be computed from (3.19) if the weight distribution of the code is known. There exists an interesting relationship between the weight distribution of a linear code and the weight distribution of its dual code. This relationship often makes the computation of $P_u(E)$ much easier. Let $\{A_0, A_1, \cdots, A_n\}$ be the weight distribution of an (n, k) linear code C, and let $\{B_0, B_1, \cdots, B_n\}$ be the weight distribution of its dual code C_d. Now, we represent these two weight distributions in polynomial form as follows:

$$\begin{aligned} A(z) &= A_0 + A_1 z + \cdots + A_n z^n, \\ B(z) &= B_0 + B_1 z + \cdots + B_n z^n. \end{aligned} \tag{3.31}$$

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

$$A(z) = 2^{-(n-k)}(1+z)^n B\left(\frac{1-z}{1+z}\right). \tag{3.32}$$

This identity is known as the *MacWilliams identity* [13]. The polynomials $A(z)$ and $B(z)$ are called the *weight enumerators* for the (n, k) linear code C and its dual C_d. From the MacWilliams identity, we see that if the weight distribution of the dual of a linear code is known, the weight distribution of the code itself can be determined. As a result, this gives us more flexibility in computing the weight distribution of a linear code.

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. First, we put the expression of (3.19) into the following form:

$$\begin{aligned} P_u(E) &= \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \\ &= (1-p)^n \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i . \end{aligned} \tag{3.33}$$

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

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

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

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

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

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

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).

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,

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

(see Example 3.3). Taking the linear combinations of the rows of $\mathbb{H}$, we obtain the following eight vectors in the dual code:

$$\begin{array}{ll} (0\,0\,0\,0\,0\,0\,0), & (1\,1\,0\,0\,1\,0\,1), \\ (1\,0\,0\,1\,0\,1\,1), & (1\,0\,1\,1\,1\,0\,0), \\ (0\,1\,0\,1\,1\,1\,0), & (0\,1\,1\,1\,0\,0\,1), \\ (0\,0\,1\,0\,1\,1\,1), & (1\,1\,1\,0\,0\,1\,0). \end{array}$$

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.

Theoretically, we can compute the weight distribution of an (n, k) linear code by examining its 2^k codewords or by examining the 2^{n-k} codewords of its dual and then applying the MacWilliams identity; however, 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 codes are still unknown. Consequently, it is very difficult, if not impossible, to compute their probability of an undetected error.

Although it is difficult to compute the probability of an undetected error for a specific (n, k) linear code for large n and k, 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.

probability of an undetected error for an (n, k) linear systematic code:

$$\begin{aligned} \mathbb{P}_u(\mathbb{E}) &\le 2^{-(n-k)} \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\ &= 2^{-(n-k)}[1-(1-p)^n]. \end{aligned} \tag{3.42}$$

Because $[1-(1-p)^n] \le 1$, it is clear that $\mathbb{P}_u(\mathbb{E}) \le 2^{-(n-k)}$.

The preceding result says that there exist (n, k) linear codes with the probability of an undetected error, $P_u(E)$, upper bounded by $2^{-(n-k)}$. In other words, there exist (n, k) linear codes with $P_u(E)$ *decreasing exponentially with the number of parity-check digits*, $n-k$. Even for moderate $n-k$, these codes have a very small probability of an undetected error. For example, let $n-k=30$. There exist (n, k) linear codes for which $P_u(E)$ is upper bounded by $2^{-30} \approx 10^{-9}$. Many classes of linear codes have been constructed for the past five decades; however, only a few small classes of linear codes have been proved to have a $P_u(E)$ that satisfies the upper bound $2^{-(n-k)}$. It is still not known whether the other known linear codes satisfy this upper bound.

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 $\mathbb{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded. The single parity-check digit is given by

$$p = u_0 + u_1 + \cdots + u_{k-1} \tag{3.43}$$

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

$$\mathbb{v} = (p, u_0, u_1, \ldots, u_{k-1}).$$

From (3.43), we readily see that $p = 1$ if the weight of message $\mathbb{u}$ is odd, and $p = 0$ if the weight of message $\mathbb{u}$ is even. Therefore, 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

$$\mathbb{G} = \left[\begin{array}{c|c} 1 & 1\,0\,0\,0 \cdots 0 \\ 1 & 0\,1\,0\,0 \cdots 0 \\ 1 & 0\,0\,1\,0 \cdots 0 \\ \vdots & \vdots \\ 1 & 0\,0\,0\,0 \cdots 1 \end{array}\right] = \left[\begin{array}{c|c} 1 & \\ 1 & \\ 1 & \mathbb{I}_k \\ \vdots & \\ 1 & \end{array}\right]. \tag{3.44}$$

From (3.44) we find that the parity-check matrix of the code is

$$\mathbb{H} = [1\ 1\ \cdots\ 1]. \tag{3.45}$$

Because all the codewords have even weights, a SPC code is also called an even-parity-check code. SPC codes are often used for simple error detection. Any error pattern with an odd number of errors will change a codeword into a received vector of odd weight that is not a codeword. Hence, the syndrome of the received vector is not equal to zero. Consequently, 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 $(0\ 0\ \cdots\ 0)$ and the all-one codeword $(1\ 1\ \cdots\ 1)$. This code is obtained by simply repeating a single message bit n times. The generator matrix of the code is

$$\mathbb{G} = [1\ 1\ \cdots\ 1]. \tag{3.46}$$

From (3.44) through (3.46), we readily 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 $n/2$. Therefore, its rate R is equal to $\frac{1}{2}$. Let $\mathbb{G}$ be a generator matrix of a self-dual code C. Then, $\mathbb{G}$ is also a generator matrix of its dual code C_d and hence is a parity-check matrix of C. Consequently,

$$\mathbb{G} \cdot \mathbb{G}^T = \mathbb{0} \tag{3.47}$$

Suppose $\mathbb{G}$ is in systematic form, $\mathbb{G} = [\mathbb{P}\ \mathbb{I}_{n/2}]$. From (3.47), we can easily see that

$$\mathbb{P} \cdot \mathbb{P}^T = \mathbb{I}_{n/2}. \tag{3.48}$$

Conversely, if a rate-$\frac{1}{2}$ $(n, n/2)$ linear block code C satisfies the condition of (3.47) [or (3.48)], then it is a self-dual code (the proof is left as a problem).

EXAMPLE 3.11

Consider the (8, 4) linear block code generated by the matrix

$$\mathbb{G} = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0\,1 \end{bmatrix}.$$

The code has a rate $R = \frac{1}{2}$. It is easy to check that $\mathbb{G} \cdot \mathbb{G}^T = \mathbb{0}$. Therefore, it is a self-dual code.