CHAPTER 4

Important Linear Block Codes

HAMMING CODES

A CLASS OF SINGLE-ERROR-CORRECTING AND DOUBLE-ERROR-DETECTING CODES

THE (24, 12) GOLAY CODE

PRODUCT CODES

INTERLEAVED CODES

HAMMING CODES

For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters:

Code length:	$n = 2^m - 1$,
Number of information symbols:	$k = 2^m - m - 1$,
Number of parity-check symbols:	$n - k = m$,
Error-correcting capability:	$t = 1$ ($d_{min} = 3$).

The parity-check matrix $\mathbb{H}$ of this code consists of all the nonzero m-tuples as its columns. In systematic form, the columns of $\mathbb{H}$ are arranged in the following form:

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

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

For example, let $m = 3$. The parity-check matrix of a Hamming code of length 7 can be put in the form

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

which is the parity-check matrix of the (7, 4) linear code given in Table 3.1 (see Example 3.3). Hence, the code given in Table 3.1 is a Hamming code. The columns of $\mathbb{Q}$ may be arranged in any order without affecting the distance property and weight distribution of the code. In systematic form, the generator matrix of the code is

$$\mathbb{G} = \begin{bmatrix} \mathbb{Q}^T & \mathbb{I}_{2^m-m-1} \end{bmatrix},$$

where $\mathbb{Q}^T$ is the transpose of $\mathbb{Q}$, and $\mathbb{I}_{2^m-m-1}$ is a $(2^m - m - 1) \times (2^m - m - 1)$ identity matrix.

Because the columns of $\mathbb{H}$ are nonzero and distinct, no two columns add to zero. It follows from Corollary 3.2.1 that the minimum distance of a Hamming code is at least 3. Since $\mathbb{H}$ consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say $\mathbb{h}_i$ and $\mathbb{h}_j$, must also be a column in $\mathbb{H}$, say $\mathbb{h}_l$. Thus,

$$\mathbb{h}_i + \mathbb{h}_j + \mathbb{h}_l = 0.$$

It follows from Corollary 3.2.2 that the minimum distance of a Hamming code is exactly 3. Hence, the code is capable of correcting all the error patterns with a single error or 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 (Theorem 3.5). The number of $(2^m - 1)$-tuples of weight 1 is $2^m - 1$. Because $n - k = m$, the code has 2^m cosets. Thus, the zero vector $\mathbf{0}$ and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array. Thus, a Hamming code corrects only error patterns of single error and no others. This is a very interesting structure. 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 leaders. Thus, Hamming codes form a class of single-error-correcting perfect codes [2]. Perfect codes are rare. Besides the Hamming codes, the only other nontrivial binary perfect code is the (23, 12) Golay code (see Section 5.9) [3–5].

Hamming codes can easily be decoded with the table-lookup scheme described in Section 3.5. The decoder for a Hamming code of length $2^m - 1$ can be implemented in the same manner as that for the (7, 4) Hamming code given in Example 3.9.

Points:

We may delete any l columns from the parity-check matrix $\mathbb{H}$ of a Hamming code. This deletion results in an $m \times (2^m - l - 1)$ matrix $\mathbb{H}'$. Using $\mathbb{H}'$ 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:	$n - k = m$
Error-correcting capability:	$d_{min} \geq 3$.

If we delete columns from $\mathbb{H}$ properly, we may obtain a shortened Hamming code with a minimum distance of 4. For example, if we delete from the submatrix $\mathbb{Q}$ all the columns of even weight, we obtain an $m \times 2^{m-1}$ matrix,

$$\mathbb{H}' = \begin{bmatrix} \mathbb{I}_m & \mathbb{Q}' \end{bmatrix},$$

where $\mathbb{Q}'$ consists of $2^{m-1} - m$ columns of odd weight. Because all the columns of $\mathbb{H}'$ have odd weight, no three columns add to zero; however, for a column $\mathbb{h}_i$ of weight 3 in $\mathbb{Q}'$, there exist three columns $\mathbb{h}_j$, $\mathbb{h}_l$, and $\mathbb{h}_s$ in $\mathbb{I}_m$ such that $\mathbb{h}_i + \mathbb{h}_j + \mathbb{h}_l + \mathbb{h}_s = \mathbb{0}$. Thus, the shortened Hamming code with $\mathbb{H}'$ as a parity-check matrix has a minimum distance of exactly 4.

Points:

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. 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; however, when double errors occur, the syndrome is also nonzero, but it contains an even number of 1's. Based on these facts, decoding can be accomplished as follows:

1. If the syndrome s is zero, we assume that no error occurred.
2. If s is nonzero and it contains an 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.
3. If s is nonzero and it contains an even number of 1's, an uncorrectable error pattern has been detected.

The weight distribution of a Hamming code of length $n = 2^m - 1$ is known [2]. The number of code vectors of weight i, A_i, is simply the coefficient of z^i in the expansion of the following polynomial:

$$A(z) = \frac{1}{n+1}\{(1+z)^n + n(1-z)(1-z^2)^{(n-1)/2}\}. \tag{4.1}$$

This polynomial is the weight enumerator for the Hamming codes.

EXAMPLE 4.1

Let $m = 3$. Then, $n = 2^3 - 1 = 7$, and the weight enumerator for the $(7, 4)$ Hamming code is

$$A(z) = \tfrac{1}{8}\{(1+z)^7 + 7(1-z)(1-z^2)^3\} = 1 + 7z^3 + 7z^4 + z^7.$$

Hence, the weight distribution for the $(7, 4)$ Hamming code is $A_0 = 1$, $A_3 = A_4 = 7$, and $A_7 = 1$.

The dual code of a $(2^m - 1, 2^m - m - 1)$ Hamming code is a $(2^m - 1, m)$ linear code. This code has a very simple weight distribution; it consists of the all-zero codeword and $2^m - 1$ codewords of weight 2^{m-1}. Thus, its weight enumerator is

$$B(z) = 1 + (2^m - 1)z^{2^{m-1}}. \tag{4.2}$$

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 (4.1) or from (3.36) and (4.2). Computing $P_u(E)$ from (3.36) and (4.2) is easier. Combining (3.36) and (4.2), we obtain

$$P_u(E) = 2^{-m}\{1 + (2^m - 1)(1 - 2p)^{2^{m-1}}\} - (1 - p)^{2^m - 1}. \tag{4.3}$$

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., $P_u(E) \leq 2^{-m}$) [6, 7], as can be shown by using the expression of (4.3) (see Problem 4.3). Therefore, Hamming codes are good error-detection codes.

A CLASS OF SINGLE-ERROR-CORRECTING AND DOUBLE-ERROR-DETECTING CODES

Single-error-correcting and double-error-detecting (SEC–DED) codes have been widely used for error control in communication and computer memory systems. In this section we present a class of SEC–DED codes that are suitable and commonly used for improving computer memory reliability. This class of codes was constructed by Hsiao [8]. The most important feature of this class of codes is their fast encoding and error detection in the decoding process, which are the most critical on-line processes in memory operations.

SEC–DED codes that have the features described can be constructed by shortening Hamming codes presented in the previous section. Construction begins with a Hamming code of length $2^m - 1$ and minimum distance 3. Let H be its parity-check matrix. Delete columns from H such that the new parity-check matrix $\mathbf{H}_0$ satisfies the following requirements:

1. Every column should have an odd number of 1's.
2. The total number of 1's in the $\mathbf{H}_0$ matrix should be a minimum.
3. The number of 1's in each row of $\mathbf{H}_0$ should be made equal, or as close as possible, to the average number (i.e., the total number of 1's in $\mathbf{H}_0$ divided by the number of rows).

The first requirement guarantees the code generated by H_0 has a minimum distance of at least 4. Therefore, it can be used for single-error correction and double-error detection. The second and third requirements yields minimum logic levels in forming parity or syndrome bits, and less hardware in implementation of the code. Hsiao [8] provided an algorithm to construct H_0 and found some optimal SEC–DED codes. Several parity-check matrices in systematic form for message (or data) lengths of 16, 32, and 64 are given in Figure 4.1. The parameters of a list of Hsiao's codes are given in Table 4.1.

TABLE 4.1: Parameters of a list of Hsiao's codes.

n	k	$n-k$	Total number of 1's in H	Average number of 1's per row
12	8	4	16	4
14	9	5	32	6.4
15	10	5	35	7
16	11	5	40	8
22	16	6	54	9
26	20	6	66	11
30	24	6	86	14.3
39	32	7	103	14.7
43	36	7	117	16.7
47	40	7	157	22.4
55	48	7	177	25.3
72	64	8	216	27
80	72	8	256	32
88	80	8	296	37
96	88	8	336	42
104	96	8	376	47
112	104	8	416	52
120	112	8	456	57
128	120	8	512	64
130	121	9	446	49.6
137	128	9	481	53.5

In computer applications, these codes are encoded and decoded in parallel manner. In encoding, the message bits enter the encoding circuit in parallel, and the parity-check bits are formed simultaneously. In decoding, the received bits enter the decoding circuit in parallel, the syndrome bits are formed simultaneously, and the received bits are corrected in parallel. Single-error correction is accomplished by the table-lookup decoding described in Example 3.9. Double-error detection is accomplished by examining the number of 1's in the syndrome vector s. If the syndrome s contains an even number of 1's, then either a double-error pattern or a multiple-even-error pattern has occurred.

Consider the parity-check matrix of the (72, 64) SEC–DED code given in Figure 4.1(c). Each row contains 27 ones. If three-input X-OR gates are used to form syndrome bits, each syndrome bit is formed by a three-level X-OR tree with 13 gates. The eight X-OR trees for generating the eight syndrome bits are identical. These provide uniform and minimum delay in the error-correction process.

$$\mathbf{H}_0 = \begin{bmatrix} 1000001001100100111100 \\ 0100000011111010001010 \\ 0010001110111001100000 \\ 0001001110000111010001 \\ 0000100001001111000111 \\ 0000010100010000111111 \end{bmatrix}$$

(a)

$$\mathbf{H}_0 = \begin{bmatrix} 10000001000101010000010000011110001101 1 \\ 01000000001000000011111011100010110000 1 \\ 00100000001011011110000100100101010011 0 \\ 00010001111111100000000110100100010001 00 \\ 00001000110110011111111000010000000010 00 \\ 00000100010000100100100111111111001000 0 \\ 00000011100000101001000010000001111111 1 \end{bmatrix}$$

(b)

$$\mathbf{H}_0 = \left[\begin{array}{cccccccc|c} 01000000 & 01100100 & 11111111 & 00000111 & 00111000 & 11001000 & 00001000 & 00001001 & 10010010 \\ 00100000 & 10010010 & 01100100 & 11111111 & 00000111 & 00111000 & 11001000 & 00001000 & 00001001 \\ 00010000 & 00001001 & 10010010 & 01100100 & 11111111 & 00000111 & 00111000 & 11001000 & 00001000 \\ 00001000 & 00001000 & 00001001 & 10010010 & 01100100 & 11111111 & 00000111 & 00111000 & 00001000 \\ 00000100 & 11001000 & 00001000 & 00001001 & 10010010 & 01100100 & 11111111 & 00000111 & 11001000 \\ 00000010 & 00111000 & 11001000 & 00001000 & 00001001 & 10010010 & 01100100 & 11111111 & 00000111 \\ 00000001 & 00000111 & 00111000 & 11001000 & 00001000 & 00001001 & 10010010 & 01100100 & 11111111 \end{array}\right]$$

(c)

$$\mathbf{H}_0 = \left[\begin{array}{cccccccc|c} 10000000 & 11111111 & 00001111 & 00001111 & 00001100 & 01101000 & 10001000 & 10001000 & 10000000 \\ 01000000 & 11110000 & 11111111 & 00000000 & 11110011 & 01100100 & 01000100 & 01000100 & 01000000 \\ 00100000 & 00110000 & 11110000 & 11111111 & 00001111 & 00000010 & 00100010 & 00100010 & 00100110 \\ 00010000 & 11001111 & 00000000 & 11110000 & 11111111 & 00000001 & 00010001 & 00010001 & 00010110 \\ 00001000 & 01101000 & 10001000 & 10001000 & 10000000 & 11111111 & 00001111 & 00000000 & 11110011 \\ 00000100 & 01100100 & 01000100 & 01000100 & 01000000 & 11110000 & 11111111 & 00001111 & 00001100 \\ 00000010 & 00000010 & 00100010 & 00100010 & 00100110 & 11001111 & 00000000 & 11111111 & 00001111 \\ 00000001 & 00000001 & 00010001 & 00010001 & 00010110 & 00110000 & 11110000 & 11110000 & 11111111 \end{array}\right]$$

(d)

FIGURE 4.1: (a) Parity-check matrix of an optimal (22, 16) SEC–DED code; (b) parity-check matrix of an optimal (39, 32) SEC–DED code; (c) parity-check matrix of an optimal (72, 64) SEC–DED code; (d) parity-check matrix of another optimal (72, 64) SEC–DED code.

THE (24, 12) GOLAY CODE

Besides the Hamming codes, the only other nontrivial binary perfect code is the (23, 12) Golay code constructed by M. J. E. Golay in 1949 [3]. This code has a minimum distance of 7 and is capable of correcting any combination of three or fewer random errors in a block of 23 digits. The code has abundant and beautiful algebraic structure, and it has become a subject of study by many coding theorists and mathematicians; many research papers have been published on its structure and decoding. The Golay code is the most extensively studied single code. In addition to having beautiful structure, this code has been used in many real communication systems for error control. This code in its cyclic form will be studied in Chapter 5.

The (23, 12) Golay code can be extended by adding an overall parity-check bit to each codeword. This extension results in a (24, 12) code with a minimum distance of 8. This code is capable of correcting all error patterns of three or fewer errors and detecting all error patterns of four errors. It is not a perfect code anymore; however, it has many interesting structural properties and has been widely used for error control in many communication systems, especially in the U.S. space program. It served as the primary *Voyager* error-control system, providing clear color pictures of Jupiter and Saturn between 1979 and 1981.

In this section we study the (24, 12) Golay code and its decoding. A generator matrix in systematic form for this code is as follows [12, 23, 24]:

$$\mathbb{G} = [\ \mathbb{P}\ \ \mathbb{I}_{12}\ ],$$

where $\mathbb{I}_{12}$ is the identity matrix of dimension 12 and

$$\mathbb{P} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,1 \\ 0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,1\,1 \\ 0\,0\,1\,1\,1\,0\,1\,1\,0\,1\,0\,1 \\ 0\,1\,1\,1\,0\,1\,1\,0\,1\,0\,0\,1 \\ 1\,1\,1\,0\,1\,1\,0\,1\,0\,0\,0\,1 \\ 1\,1\,0\,1\,1\,0\,1\,0\,0\,0\,1\,1 \\ 1\,0\,1\,1\,0\,1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,1\,0\,1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,1\,0\,0\,0\,1\,1\,1\,0\,1 \\ 1\,0\,1\,0\,0\,0\,1\,1\,1\,0\,1\,1 \\ 0\,1\,0\,0\,0\,1\,1\,1\,0\,1\,1\,1 \\ 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0 \end{bmatrix}. \tag{4.75}$$

The $\mathbb{P}$ matrix has the following properties: (1) it is symmetrical with respect to its diagonal; (2) the ith column is the transpose of the ith row; (3) $\mathbb{P} \cdot \mathbb{P}^T = \mathbb{I}_{12}$, where $\mathbb{P}^T$ is the transpose of $\mathbb{P}$; and (4) the submatrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times (or cyclically shifting the first column upward 11 times). It follows from the second property that

$$\mathbb{P}^T = \mathbb{P}.$$

Consequently, the parity-check matrix in systematic form for the (24, 12) extended Golay code is given by

$$\begin{aligned}\mathbb{H} &= \begin{bmatrix} \mathbb{I}_{12} & \mathbb{P}^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I}_{12} & \mathbb{P} \end{bmatrix}.\end{aligned} \tag{4.76}$$

It can be proved that the code is self-dual [see Problem 4.18].

A simple decoding algorithm for the (24, 12) Golay code can be devised using the properties of the $\mathbb{P}$ matrix [23]. For $0 \leq i \leq 11$, let $\mathbb{p}_i$ denote the ith row of $\mathbb{P}$ and $\mathbb{u}^{(i)}$ the 12-tuple in which only the ith component is nonzero. For example, $\mathbb{u}^{(5)} = (0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0)$. We readily see that

$$\mathbb{p}_i = \mathbb{u}^{(i)} \cdot \mathbb{P}. \tag{4.77}$$

Let $\mathbb{e} = (\mathbb{x}, \mathbb{y})$ be an error vector, where $\mathbb{x}$ and $\mathbb{y}$ are binary 12-tuples. Suppose a codeword $\mathbb{v}$ is transmitted, and a correctable error pattern $\mathbb{e} = (\mathbb{x}, \mathbb{y})$ occurs. Then,

the received vector is $\mathbb{r} = \mathbb{v} + \mathbb{e}$. The syndrome of $\mathbb{r}$ is

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

It follows from (4.76) and $\mathbb{P}^T = \mathbb{P}$ that

$$\begin{aligned} \mathbb{s} &= (\mathbb{x}, \mathbb{y}) \cdot \begin{bmatrix} \mathbb{I}_{12} \\ \mathbb{P} \end{bmatrix} \\ &= \mathbb{x} \cdot \mathbb{I}_{12} + \mathbb{y} \cdot \mathbb{P} \\ &= \mathbb{x} + \mathbb{y} \cdot \mathbb{P}. \end{aligned} \tag{4.78}$$

Using the property $\mathbb{P} \cdot \mathbb{P}^T = \mathbb{I}_{12}$, we can express $\mathbb{y}$ in terms of $\mathbb{x}$, $\mathbb{s}$, and $\mathbb{P}$ as follows:

$$\mathbb{y} = (\mathbb{s} + \mathbb{x}) \cdot \mathbb{P}. \tag{4.79}$$

In the following, we first show that a correctable error pattern for the (24, 12) Golay code can be expressed in terms of $\mathbb{P}$, $\mathbb{p}_i$, $\mathbb{u}^{(i)}$, and $\mathbb{s}$. We then present a decoding algorithm for the code.

For any correctable error pattern with weight $w(\mathbb{e}) \leq 3$, we have the following four possibilities:

(1) $w(\mathbb{x}) \leq 3$ and $w(\mathbb{y}) = 0$,
(2) $w(\mathbb{x}) \leq 2$ and $w(\mathbb{y}) = 1$,
(3) $w(\mathbb{x}) \leq 1$ and $w(\mathbb{y}) = 2$,
(4) $w(\mathbb{x}) = 0$ and $w(\mathbb{y}) = 3$.

Theses four possibilities define four different types of correctable error patterns. For $0 \leq j \leq 3$, let $\mathbf{e}^{(j)} = (\mathbf{x}, \mathbf{y})$, for which $w(\mathbf{y}) = j$, and $w(\mathbf{x}) \leq 3 - j$. Suppose $\mathbf{e} = \mathbf{e}^{(0)}$. It follows from (4.78) that $\mathbf{s} = \mathbf{x}$ and $w(\mathbf{s}) = w(\mathbf{x}) \leq 3$. In this case,

$$\mathbf{e} = (\mathbf{s}, \mathbb{0}),$$

where $\mathbb{0}$ is the all-zero 12-tuple. Suppose $\mathbf{e} = \mathbf{e}^{(1)}$ and $\mathbf{y} = \mathbf{u}^{(i)}$. Then, it follows from (4.78) that

$$\mathbf{s} = \mathbf{x} + \mathbf{u}^{(i)} \cdot \mathbb{P} = \mathbf{x} + \mathbf{p}_i.$$

Hence, $\mathbf{x} = \mathbf{s} + \mathbf{p}_i$, and $w(\mathbf{s} + \mathbf{p}_i) = w(\mathbf{x}) \leq 2$. In this case,

$$\mathbf{e} = (\mathbf{s} + \mathbf{p}_i,\ \mathbf{u}^{(i)}).$$

Suppose $\mathbf{e} = \mathbf{e}^{(2)}$ or $\mathbf{e}^{(3)}$, and $w(\mathbf{x}) = 0$. It follows from (4.79) that

$$\mathbf{y} = \mathbf{s} \cdot \mathbb{P},$$

and $w(\mathbf{s} \cdot \mathbb{P}) = w(\mathbf{y}) = 2$ or 3. For this case, we can express $\mathbf{e}$ as follows:

$$\mathbf{e} = (\mathbb{0},\ \mathbf{s} \cdot \mathbb{P}).$$

Now, suppose $\mathbf{e} = \mathbf{e}^{(2)}$, and $w(\mathbf{x}) = 1$. If the nonzero component of $\mathbf{x}$ is at the ith position, then $\mathbf{x} = \mathbf{u}^{(i)}$. It follows from (4.79) that

$$\begin{aligned}\mathbf{y} &= (\mathbf{s} + \mathbf{u}^{(i)}) \cdot \mathbb{P} \\ &= \mathbf{s} \cdot \mathbb{P} + \mathbf{u}^{(i)} \cdot \mathbb{P} \\ &= \mathbf{s} \cdot \mathbb{P} + \mathbf{p}_i\end{aligned}$$

and $w(\mathbf{s} \cdot \mathbb{P} + \mathbf{p}_i) = w(\mathbf{y}) = 2$. Consequently, we can express $\mathbf{e}$ as follows:

$$\mathbf{e} = (\mathbf{u}^{(i)},\ \mathbf{s} \cdot \mathbb{P} + \mathbf{p}_i).$$

A decoding algorithm can be devised for the (24, 12) Golay code based on the preceding analysis and expressions of correctable error patterns. The decoding consists of the following steps:

Step 1. Compute the syndrome $\mathbf{s}$ of the received sequence $\mathbf{r}$.

Step 2. If $w(\mathbf{s}) \leq 3$, then set $\mathbf{e} = (\mathbf{s}, \mathbf{0})$ and go to step 8.

Step 3. If $w(\mathbf{s} + \mathbf{p}_i) \leq 2$ for some row $\mathbf{p}_i$ in $\mathbf{P}$, then set $\mathbf{e} = (\mathbf{s} + \mathbf{p}_i, \mathbf{u}^{(i)})$ and go to step 8.

Step 4. Compute $\mathbf{s} \cdot \mathbf{P}$.

Step 5. If $w(\mathbf{s} \cdot \mathbf{P}) = 2$ or 3, then set $\mathbf{e} = (\mathbf{0}, \mathbf{s} \cdot \mathbf{P})$ and go to step 8.

Step 6. If $w(\mathbf{s} \cdot \mathbf{P} + \mathbf{p}_i) = 2$ for some row $\mathbf{p}_i$ in $\mathbf{P}$, then set $\mathbf{e} = (\mathbf{u}^{(i)}, \mathbf{s} \cdot \mathbf{P} + \mathbf{p}_i)$ and go to step 8.

Step 7. If the syndrome does not correspond to a correctable error pattern, stop the decoding process, or request a retransmission. (This represents a decoding failure.)

Step 8. Set the decoded codeword $\mathbf{v}^* = \mathbf{r} + \mathbf{e}$ and stop.

EXAMPLE 4.7

Suppose the (24, 12) Golay code is used for error control. Let $\mathbf{r} = (1\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1)$ be the received sequence. To decode $\mathbf{r}$, we first compute the syndrome $\mathbf{s}$ of $\mathbf{r}$:

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,0\,0).$$

Because $w(\mathbf{s}) > 3$, we go to decoding step 3. We find that

$$\begin{aligned}\mathbf{s} + \mathbf{p}_{11} &= (1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,0\,0) + (1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0) \\ &= (0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0),\end{aligned}$$

and $w(\mathbf{s} + \mathbf{p}_{11}) = 2$. So we set

$$\begin{aligned}\mathbf{e} &= (\mathbf{s} + \mathbf{p}_{11},\ \mathbf{u}^{(11)}) \\ &= (0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1)\end{aligned}$$

and decode $\mathbf{r}$ into

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

PRODUCT CODES

Let C_1 be an (n_1, k_1) linear code, and let C_2 be an (n_2, k_2) linear code. Then, an $(n_1 n_2, k_1 k_2)$ linear code can be formed such that each codeword is a rectangular array of n_1 columns and n_2 rows in which every row is a codeword in C_1, and every column is a codeword in C_2, as shown in Figure 4.3. This two-dimensional code is called the *direct product* (or simply the *product*) of C_1, and C_2 [25]. The $k_1 k_2$ digits in the upper right corner of the array are information symbols. The digits in the upper left corner of this array are computed from the parity-check rules for C_1 on rows, and the digits in the lower right corner are computed from the parity-check rules for C_2 on columns. Now, should we compute the check digits in the lower left corner by using the parity-check rules for C_2 on columns or the parity-check rules for C_1 on rows? It turns out that either way yields the same $(n_1 - k_1) \times (n_2 - k_2)$ check digits (see Problem 4.21), and it is possible to have all row codewords in C_1 and all column codewords in C_2 simultaneously.

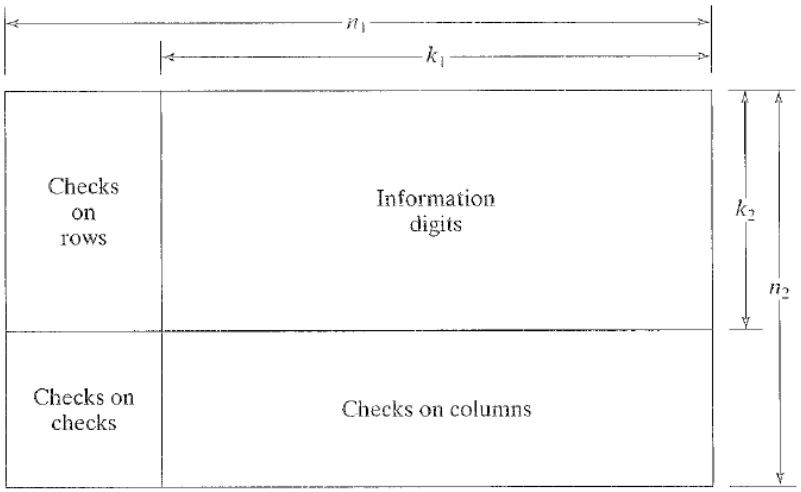

FIGURE 4.3: Code array for the product code $C_1 \times C_2$.

The product code $C_1 \times C_2$ is encoded in two steps. A message of $k_1 k_2$ information symbols is first arranged as shown in the upper right corner of Figure 4.3. At the first step of encoding, each row of the information array is encoded into a codeword in C_1. This row encoding results in an array of k_2 rows and n_1 columns, as shown in the upper part of Figure 4.3. At the second step of encoding each of the n_1 columns of the array formed at the first encoding step is encoded into a codeword in C_2. This results in a code array of n_2 rows and n_1 columns, as shown in Figure 4.3. This code array also can be formed by first performing the column encoding and then the row encoding. Transmission can be carried out either column by column or row by row.

If code C_1 has minimum weight d_1 and code C_2 has minimum weight d_2, the minimum weight of the product code is exactly $d_1 d_2$. A minimum-weight codeword in the product code is formed by (1) choosing a minimum-weight codeword in C_1 and a minimum-weight codeword in C_2 and (2) forming an array in which all columns corresponding to zeros in the codeword from C_1 are zeros, and all columns corresponding to ones in the codeword from C_1 are the minimum-weight codeword chosen from C_2.

It is not easy to characterize the correctable error patterns for the product code; this depends on how the correction is done. One method involves using a two-step decoding. The decoding is first performed on rows and then on columns. In this case a pattern will be correctable if and only if the uncorrectable patterns on rows after row correction leave correctable patterns on the columns. It generally improves the correction to decode by rows, then columns, then columns and rows again. This method, of course, increases the decoding delay. This type of decoding is called *iterative decoding* [25].

The product code is capable of correcting any combination of $[(d_1 d_2 - 1)/2]$ errors, but the method described will not achieve this. For example, consider the product code of two Hamming single-error-correcting codes. The minimum distance of each is 3, so the minimum distance of the product is 9. A pattern of four errors at the corners of a rectangle gives two errors in each of the two rows and two columns and is therefore not correctable by simple correction on rows and columns. Nevertheless, simple correction on rows and columns, although nonoptimum, can be very effective. The complexity of the two-step decoding is roughly the sum of the complexities of the two component code decodings.

Comment:

EXAMPLE 4.8

Consider the product of the $(5, 4)$ SPC code C_1 with itself. The product code $C_1 \times C_1$ is a $(25, 16)$ linear block code C with a minimum distance of 4. Suppose the message $\mathbf{u} = (1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1)$ is to be encoded. This message is read into a storage buffer and arranged into a 4×4 information array, as shown in Figure 4.4. The first four information symbols form the first row of the information array, the second four information symbols form the second row, and so on. At the first step of encoding, a single (even) parity-check symbol is added to each row of the information array. This results in a 4×5 array, as shown in Figure 4.4. In the second step of encoding a single (even) parity-check symbol is added to each of the five columns of the array, as shown in Figure 4.4. Suppose this code array is transmitted column by column. At the received end, the received sequence is rearranged into a 5×5 code array column by column, called the *received array*. Suppose a single error occurs at the intersection of a row and a column. The erroneous row and column containing this single error are indicated by parity-check failures, then the error is corrected by complementing the received symbol (i.e., 0 to 1, and 1 to 0) at the intersection. All the single-error patterns can be corrected in this manner. Checking the row and column parity failures cannot correct any double-error pattern, but it can detect all the double-error patterns. When a double-error pattern occurs, there

1	1	0	1	1
1	0	0	0	1
0	0	1	0	1
1	1	1	0	1
1	0	0	1	0

1	1	0	1	1
1	0	0	0	1
0	0	1	0	1
1	1	1	0	1
1	0	0	1	0

are three possible distributions of the two errors: (1) they are in the same row; (2) they are in the same column; or (3) they are in different rows and different columns. In the first case, there are two column parity failures but no row parity failure. Hence, errors are detected but they cannot be located. In the second case, there are two row parity failures but no column parity failure. Again, errors are detected but cannot be located. In the third case, there are two row parity failures and two column parity failures, so there are four intersecting locations. The two errors are situated at two opposite diagonal positions, but we cannot determine the positions.

In constructing a two-dimensional product code, if we do not form the $(n_1 - k_1) \times (n_2 - k_2)$ checks on checks in the lower left corner of Figure 4.3, we obtain an incomplete code array, as shown in Figure 4.5. This incomplete product of two codes results in a $(k_1 n_2 + k_2 n_1 - k_1 k_2, k_1 k_2)$ linear block code with a minimum distance of $d_1 + d_2 - 1$ (see Problem 4.22). The code has a higher rate but smaller minimum distance than the complete product code.

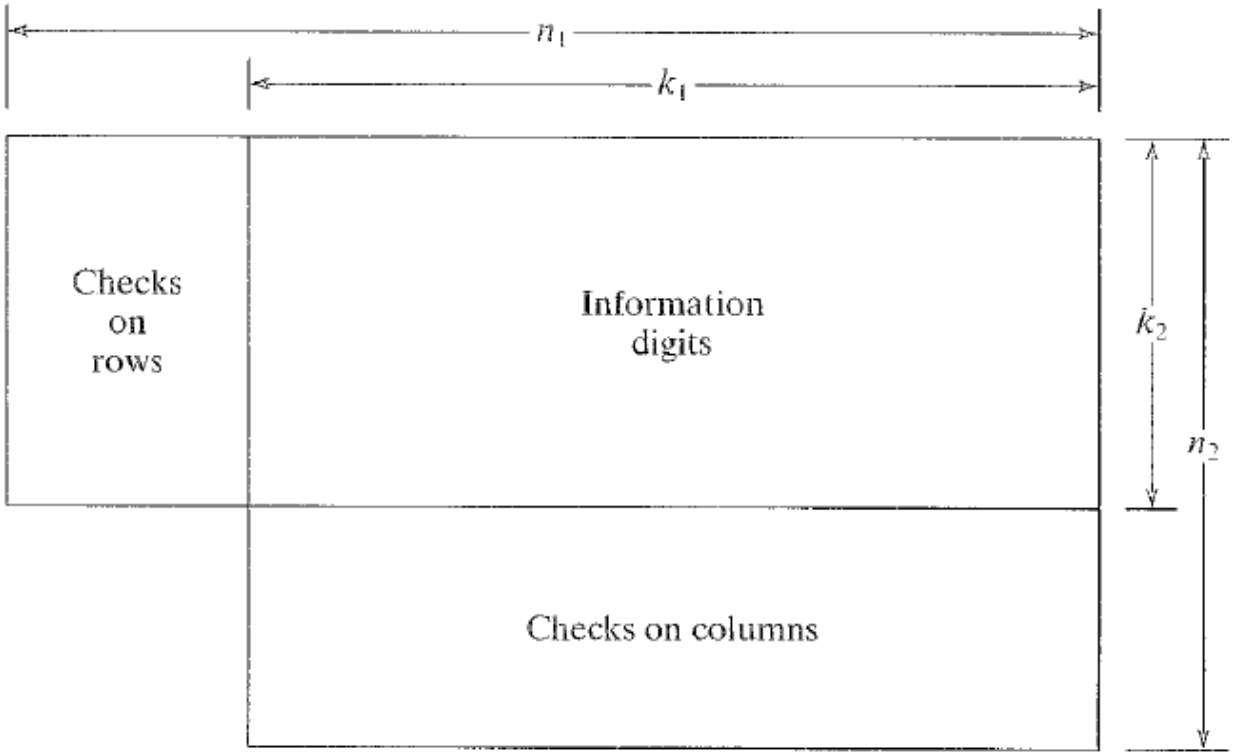

FIGURE 4.5: Incomplete product of two codes.

INTERLEAVED CODES

Given an (n, k) linear block code C, it is possible to construct a $(\lambda n, \lambda k)$ linear block code (i.e., a code λ times as long with λ times as many information symbols) by interleaving, that is, simply by arranging λ codewords in C into λ rows of a rectangular array and then transmitting the array column by column, as shown in Figure 4.6. The resulting code, denoted by C^{λ}, is called an *interleaved code*. The parameter λ is referred to as the *interleaving depth* (or *degree*). If the minimum distance of the base code C is d_{min}, the minimum distance of the interleaved code is also d_{min}.

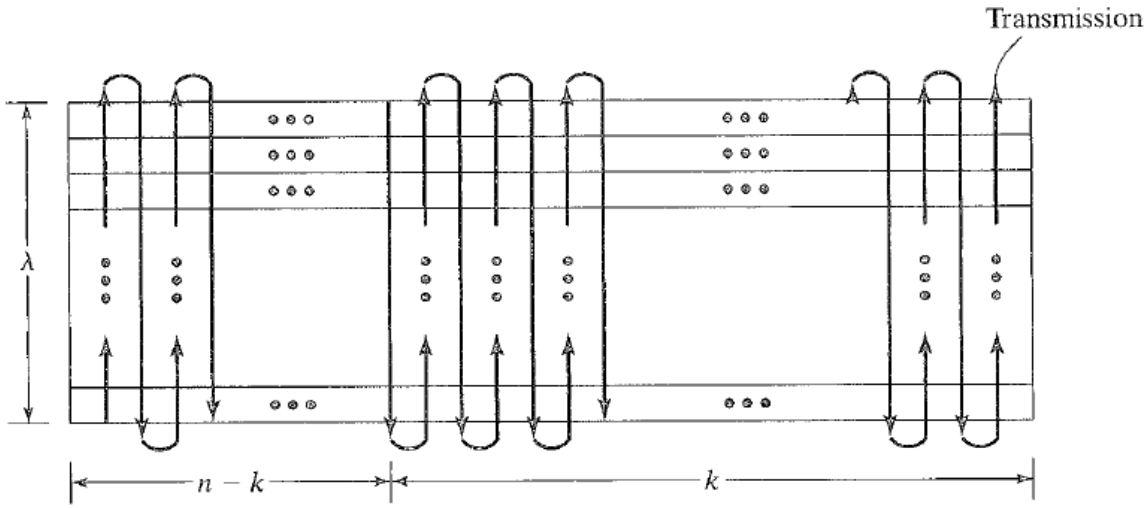

FIGURE 4.6: Transmission of an interleaved code.

Comment: