CHAPTER 8

Majority-Logic Decodable and Finite Geometry Codes

Majority-logic decoding is a simple and effective scheme for decoding certain classes of block codes, especially for decoding certain classes of cyclic codes. The first majority-logic decoding algorithm was devised in 1954 by Reed [1] for the class of RM codes presented in Chapter 4. Reed's algorithm was later extended and generalized by many coding theorists. The first unified formulation of majority-logic decoding was due to Massey [2].

Most majority-logic decodable codes found so far are cyclic codes. Important cyclic codes of this category are codes constructed based on finite geometries, namely, Euclidean and projective geometries. These codes are called *finite geometry codes* and they contain punctured RM codes in cyclic form as a subclass. A special subclass of finite geometry codes forms a special subclass of low-density parity-check codes that will be discussed in Chapter 17. Finite geometry codes were first investigated by Rudolph [3] in 1967. Rudolph's work was later extended and generalized by many coding researchers, from the late 1960s to the late 1970s.

In this chapter we first introduce majority-logic decoding based on orthogonal parity-check sums formed from the parity-check matrix or the dual space of a code. Then, we present several classes of cyclic majority-logic decodable codes.

8.1 ONE-STEP MAJORITY-LOGIC DECODING

Consider an (n, k) cyclic code C with parity-check matrix $\mathbb{H}$. The row space of $\mathbb{H}$ is an $(n, n-k)$ cyclic code, denoted by C_d, which is the dual code of C, or the null space of C. For any codeword $\mathbb{v}$ in C and any codeword $\mathbb{w}$ in C_d, the inner product of $\mathbb{v}$ and $\mathbb{w}$ is zero; that is,

$$\mathbb{w} \cdot \mathbb{v} = w_0 v_0 + w_1 v_1 + \cdots + w_{n-1} v_{n-1} = 0. \tag{8.1}$$

In fact, an n-tuple $\mathbb{v}$ is a codeword in C if and only if for any vector $\mathbb{w}$ in C_d, $\mathbb{w} \cdot \mathbb{v} = 0$. The equality of (8.1) is called a *parity-check equation.* Clearly, there are $2^{(n-k)}$ such parity-check equations.

Now, suppose that a codeword $\mathbb{v}$ in C is transmitted. Let $\mathbb{e} = (e_0, e_1, \cdots, e_{n-1})$ and $\mathbb{r} = (r_0, r_1, \cdots, r_{n-1})$ be the error vector and the received vector, respectively. Then,

$$\mathbb{r} = \mathbb{v} + \mathbb{e}. \tag{8.2}$$

For any vector $\mathbb{w}$ in the dual code C_d, we can form the following linear sum of the received digits:

$$A = \mathbb{w} \cdot \mathbb{r} = w_0 r_0 + w_1 r_1 + \cdots + w_{n-1} r_{n-1}, \tag{8.3}$$

which is called a *parity-check sum* or simply *check-sum.* If the received vector $\mathbb{r}$ is a codeword in C, this parity-check sum, A, *must be zero*; however, if $\mathbb{r}$ is not a

codeword in C, then A *may not* be zero. Combining (8.2) and (8.3) and using the fact that $\mathbf{w} \cdot \mathbf{v} = 0$, we obtain the following relationship between the check-sum A and error digits in $\mathbf{e}$:

$$A = w_0 e_0 + w_1 e_1 + \cdots + w_{n-1} e_{n-1}. \tag{8.4}$$

An error digit e_l is said to be *checked* by the check-sum A if the coefficient $w_l = 1$. In the following, we show that certain properly formed check-sums can be used for estimating the error digits in $\mathbf{e}$.

Suppose that there exist J vectors in the dual code C_d,

$$\begin{aligned}
\mathbf{w}_1 &= (w_{10}, w_{11}, \cdots, w_{1,n-1}),\\
\mathbf{w}_2 &= (w_{20}, w_{21}, \cdots, w_{2,n-1}),\\
&\vdots\\
\mathbf{w}_J &= (w_{J0}, w_{J1}, \cdots, w_{J,n-1}),
\end{aligned}$$

that have the following properties:

1. The $(n-1)$th component of each vector is a 1; that is,

$$w_{1,n-1} = w_{2,n-1} = \cdots = w_{J,n-1} = 1.$$

2. For $i \neq n-1$, there is *at most* one vector whose ith component is a 1; for example, if $w_{1,i} = 1$, then $w_{2,i} = w_{3,i} = \cdots = w_{J,i} = 0$.

These J vectors are said to be *orthogonal* on the $(n-1)$th digit position. We call them *orthogonal vectors*. Now, we form J parity-check sums from these J orthogonal vectors:

$$\begin{aligned}
A_1 &= \mathbf{w}_1 \cdot \mathbf{r} = w_{10} r_0 + w_{11} r_1 + \cdots + w_{1,n-1} r_{n-1}\\
A_2 &= \mathbf{w}_2 \cdot \mathbf{r} = w_{20} r_0 + w_{21} r_1 + \cdots + w_{2,n-1} r_{n-1}\\
&\vdots\\
A_J &= \mathbf{w}_J \cdot \mathbf{r} = w_{J0} r_0 + w_{J1} r_1 + \cdots + w_{J,n-1} r_{n-1}.
\end{aligned} \tag{8.5}$$

Because $w_{1,n-1} = w_{2,n-1} = \cdots = w_{J,n-1} = 1$, these J check-sums are related to the error digits in the following manner:

$$\begin{aligned}
A_1 &= w_{10} e_0 + w_{11} e_1 + \cdots + w_{1,n-2} e_{n-2} + e_{n-1}\\
A_2 &= w_{20} e_0 + w_{21} e_1 + \cdots + w_{2,n-2} e_{n-2} + e_{n-1}\\
&\vdots\\
A_J &= w_{J,0} e_0 + w_{J,1} e_1 + \cdots + w_{J,n-2} e_{n-2} + e_{n-1}.
\end{aligned} \tag{8.6}$$

We see that the error digit e_{n-1} is checked by all the preceding check-sums. Because of the second property of the orthogonal vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_J$ any error digit

other than e_{n-1} is checked by at most one check-sum. *These J check-sums are said to be orthogonal on the error digit e_{n-1}.* Since $w_{i,j} = 0$, or 1, each of the foregoing check-sums orthogonal on e_{n-1} is of the form

$$A_j = e_{n-1} + \sum_{i \neq n-1} e_i.$$

If all the error digits in the sum of A_j are zero for $i \neq n-1$, *the value of e_{n-1} is equal to A_j* (i.e., $e_{n-1} = A_j$). Based on this fact, the parity-check sums orthogonal on e_{n-1} can be used to estimate e_{n-1} or to decode the received digit r_{n-1}.

Suppose that there are $\lfloor J/2 \rfloor$ or fewer errors in the error vector $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1})$ (i.e., $\lfloor J/2 \rfloor$ or fewer components of $\mathbf{e}$ are 1). If $e_{n-1} = 1$, the other nonzero error digits can distribute among at most $\lfloor J/2 \rfloor - 1$ check-sums orthogonal on e_{n-1}. Hence, at least $J - \lfloor J/2 \rfloor + 1$, or *more than half* of the check-sums orthogonal on e_{n-1}, are equal to $e_{n-1} = 1$; however, if $e_{n-1} = 0$, the nonzero error digits can distribute among at most $\lfloor J/2 \rfloor$ check-sums. Hence, at least $J - \lfloor J/2 \rfloor$ or *at least half* of the check sums orthogonal on e_{n-1} are equal to $e_{n-1} = 0$. Thus, the value of e_{n-1} is equal to the value assumed by a *clear majority* of the parity-check sums orthogonal on e_{n-1}; if no value is assumed by a clear majority of the parity-check sums (i.e., there is a *tie*), the error digit e_{n-1} is zero. Based on the preceding facts, an algorithm for decoding e_{n-1} can be formulated as follows:

> *The error digit e_{n-1} is decoded as 1 if a clear majority of the parity-check sums orthogonal on e_{n-1} is 1; otherwise, e_{n-1} is decoded as 0.*

Correct decoding of e_{n-1} is guaranteed if there are $\lfloor J/2 \rfloor$ or fewer errors in the error vector $\mathbf{e}$. If it is possible to form J parity-check sums orthogonal on e_{n-1}, it is possible to form J parity-check sums orthogonal on any error digit because of the cyclic symmetry of the code. The decoding of other error digits is identical to the decoding of e_{n-1}. The decoding algorithm just described is called *one-step majority-logic decoding* [2]. If J is the maximum number of parity-check sums orthogonal on e_{n-1} (or any error digit) that can be formed, then, by one-step majority-logic decoding, any error pattern of $\lfloor J/2 \rfloor$ or fewer errors can be corrected. The parameter $t_{ML} = \lfloor J/2 \rfloor$ is called the *majority-logic error-correcting capability* of the code. Let d_{min} be the minimum distance of the code. Clearly, the one-step majority-logic decoding is effective for this code only if $t_{ML} = \lfloor J/2 \rfloor$ is equal to or close to the error-correcting capability $t = \lfloor (d_{min} - 1)/2 \rfloor$ of the code; in other words, J should be equal to or close to $d_{min} - 1$.

> DEFINITION 8.1 A cyclic code with minimum distance $d_{\min}$ is said to be *completely orthogonalizable* in one step if and only if it is possible to form $J = d_{\min} - 1$ parity-check sums orthogonal on an error digit.

At this point, a clarifying example will be helpful.

EXAMPLE 8.1

Consider a (15, 7) cyclic code generated by the polynomial

$$g(X) = 1 + X^4 + X^6 + X^7 + X^8.$$

The parity-check matrix of this code (in systematic form) is found as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \mathbf{h}_5 \\ \mathbf{h}_6 \\ \mathbf{h}_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Consider the following linear combinations of the rows of $\mathbf{H}$:

	Digit positions:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\mathbf{w}_1 =$	$\mathbf{h}_3 =$	(0	0	0	1	0	0	0	0	0	0	0	1	1	0	1),
$\mathbf{w}_2 =$	$\mathbf{h}_1 + \mathbf{h}_5 =$	(0	1	0	0	0	1	0	0	0	0	0	0	0	1	1),
$\mathbf{w}_3 = \mathbf{h}_0 + \mathbf{h}_2 + \mathbf{h}_6 =$		(1	0	1	0	0	0	1	0	0	0	0	0	0	0	1),
$\mathbf{w}_4 =$	$\mathbf{h}_7 =$	(0	0	0	0	0	0	0	1	1	0	1	0	0	0	1).

We see that all four vectors have a 1 at digit position 14 (or X^{14}), and at any other digit position, no more than one vector has a 1. Therefore, these four vectors are orthogonal on the digit position 14. Let $\mathbf{r}$ be the received vector. The four parity-check sums formed from these orthogonal vectors are related to the error digits as follows:

$$\begin{aligned}
A_1 = \mathbf{w}_1 \cdot \mathbf{r} &= e_3 + e_{11} + e_{12} + e_{14} \\
A_2 = \mathbf{w}_2 \cdot \mathbf{r} &= e_1 + e_5 + e_{13} + e_{14} \\
A_3 = \mathbf{w}_3 \cdot \mathbf{r} &= e_0 + e_2 + e_6 + e_{14} \\
A_4 = \mathbf{w}_4 \cdot \mathbf{r} &= e_7 + e_8 + e_{10} + e_{14}.
\end{aligned}$$

We see that e_{14} is checked by all four check-sums, and no other error digit is checked by more than one check-sum. If $e_{14} = 1$, and if there is one or no error occurring among the other 14 digit positions, then at least three (majority) of the four sums, A_1, A_2, A_3, and A_4, are equal to $e_{14} = 1$. If $e_{14} = 0$ and if there are two or fewer errors occurring among the other 14 digit positions, then at least two of the four check-sums are equal to $e_{14} = 0$. Hence, if there are two or fewer errors in $\mathbf{e}$, the one-step majority-logic decoding always results in correct decoding of e_{14}. Because the code is cyclic, four parity-check sums orthogonal on any error digit can be formed. It can be checked that four is the maximum number of parity-check sums orthogonal on any error digit that can be formed. Thus, by one-step majority-logic decoding, the code is capable of correcting any error pattern with two or fewer errors. It can be shown that there exists at least one error pattern with three errors that cannot be corrected. Consider an error pattern $\mathbf{e}$ with three errors, e_0, e_3, and e_8 (i.e., $e_0 = e_3 = e_8 = 1$). From the four parity-check sums orthogonal on e_{14}, we have $A_1 = 1$, $A_2 = 0$, $A_3 = 1$, and $A_4 = 1$. Because the majority of the four sums is 1, according to the decoding rule, e_{14} is decoded as 1. This results in an incorrect decoding. The code given in this example is actually a BCH code with a minimum distance of exactly 5. Therefore, it is completely orthogonalizable.

Given an (n, k) cyclic code C for which J parity-check sums orthogonal on an error digit can be formed, the one-step majority-logic decoding of the code can easily be implemented. First, from the null space C_d of the code, we determine a set of J vectors $\mathbb{w}_1, \mathbb{w}_2, \cdots, \mathbb{w}_J$ that are orthogonal on the highest-order digit position, X^{n-1}. Then, J parity-check sums $A_1, A_2, \cdots, A_J$ orthogonal on the error digit e_{n-1} are formed from these J orthogonal vectors and the received vector $\mathbb{r}$. From (8.5), we see that the vector $\mathbb{w}_j$ tells what received digits should be summed up to from the check-sum A_j. The J check-sums can be formed by using J multi-input modulo-2 adders. Once these J check-sums are formed, they are used as inputs to a J-input majority-logic gate. The output of a majority-logic gate is 1 if and only if more than half its inputs are 1; otherwise, the output is 0. The output is the estimated value of e_{n-1}. A general one-step majority-logic decoder is shown in Figure 8.1. This decoder is called the type-II one-step majority-logic decoder [2]. The error correction procedure is as follows:

Step 1. With gate 1 turned on and gate 2 turned off, the received vector $\mathbb{r}$ is read into the buffer register.

Step 2. The J parity-check sums orthogonal on e_{n-1} are formed by summing the appropriate received digits.

Step 3. The J orthogonal check sums are fed into a majority-logic gate. The first received digit r_{n-1} is read out of the buffer and is corrected by the output of the majority-logic gate.

Step 4. At the end of step 3, the buffer register has been shifted one place to the right with gate 2 on. Now, the second received digit is in the rightmost stage of the buffer register and is corrected in exactly the

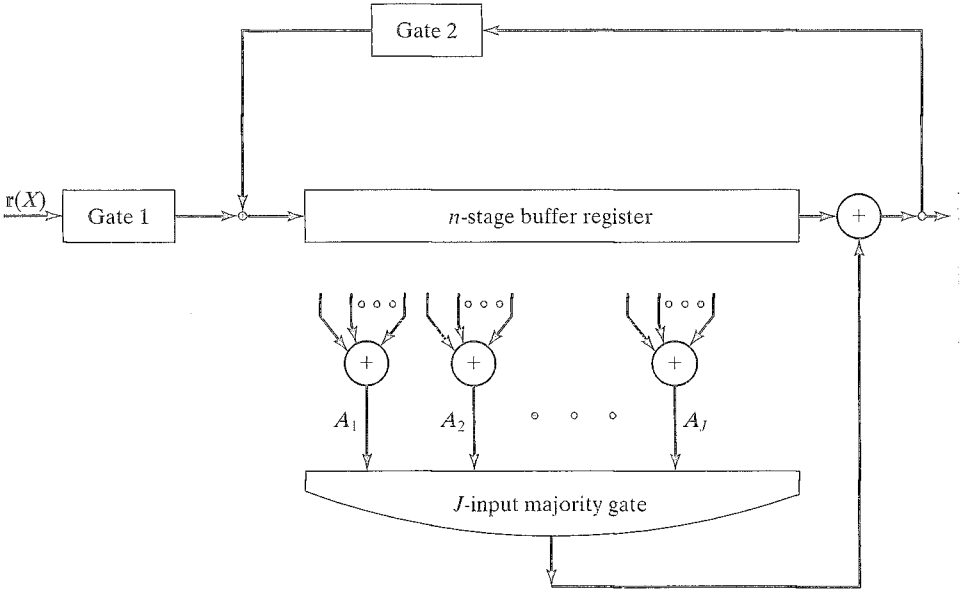

FIGURE 8.1: General type-II one-step majority-logic decoder.

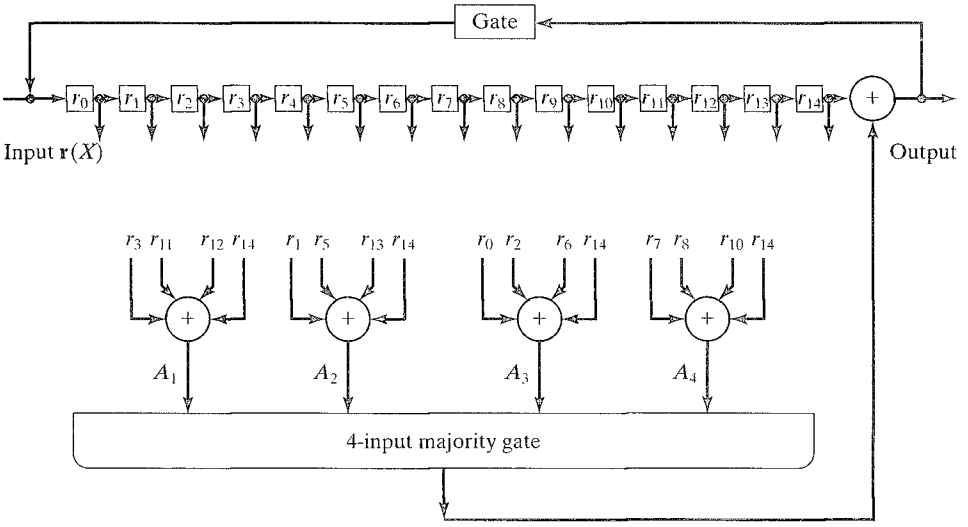

FIGURE 8.2: Type-II one-step majority-logic decoder for the (15, 7) BCH code.

same manner as was the first received digit. The decoder repeats step 2 and 3.

Step 5. The received vector is decoded digit by digit in the same manner until a total of n shifts.

If the received vector $\mathbf{r}$ contains $\lfloor J/2 \rfloor$ or fewer errors, the buffer register should contain the transmitted code vector, and the inputs to the majority-logic gate should all be zero at the completion of the decoding operation. If not all the inputs to the majority gate are zero, an *uncorrectable error pattern* has been detected.

The type-II one-step majority-logic decoder for the (15, 7) BCH code considered in Example 8.1 is shown in Figure 8.2.

The parity-check sums orthogonal on an error digit also can be formed from the syndrome digits. Let

$$\mathbb{H} = \begin{bmatrix} \mathbb{h}_0 \\ \mathbb{h}_1 \\ \mathbb{h}_2 \\ \vdots \\ \mathbb{h}_{n-k-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{00} & p_{01} & & p_{0,k-1} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{10} & p_{11} & & p_{1,k-1} \\ 0 & 0 & 1 & 0 & \cdots & 0 & p_{20} & p_{21} & & p_{2,k-1} \\ \vdots & & & & & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{n-k-1,0} & p_{n-k-1,1} & \cdots & p_{n-k-1,k-1} \end{bmatrix}$$

be the parity-check matrix for an (n, k) cyclic code C in systematic form. Because the orthogonal vectors $\mathbb{w}_1, \mathbb{w}_2, \cdots, \mathbb{w}_J$ are vectors in the row space of $\mathbb{H}$, they are linear combinations of rows of $\mathbb{H}$. Let

$$\begin{aligned} \mathbb{w}_j &= (w_{j0}, w_{j1}, \cdots, w_{j,n-1}) \\ &= a_{j0}\mathbb{h}_0 + a_{j1}\mathbb{h}_1 + \cdots + a_{j,n-k-1}\mathbb{h}_{n-k-1}. \end{aligned}$$

Because of the systematic structure of $\mathbb{H}$, we see that

$$w_{j0} = a_{j0}, \quad w_{j1} = a_{j1}, \cdots, w_{j,n-k-1} = a_{j,n-k-1}. \tag{8.7}$$

Let $\mathbb{r} = (r_0, r_1, \cdots, r_{n-1})$ be the received vector. Then, the syndrome of $\mathbb{r}$ is

$$\mathbb{s} = (s_0, s_1, \cdots, s_{n-k-1}) = \mathbb{r} \cdot \mathbb{H}^T,$$

where the ith syndrome digit is

$$s_i = \mathbb{r} \cdot \mathbb{h}_i \tag{8.8}$$

for $0 \leq i < n - k$. Now, consider the parity-check sum

$$\begin{aligned} A_j &= \mathbb{w}_j \cdot \mathbb{r} \\ &= (a_{j0}\mathbb{h}_0 + a_{j1}\mathbb{h}_1 + \cdots + a_{j,n-k-1}\mathbb{h}_{n-k-1}) \cdot \mathbb{r} \\ &= a_{j0}\mathbb{r} \cdot \mathbb{h}_0 + a_{j1}\mathbb{r} \cdot \mathbb{h}_1 + \cdots + a_{j,n-k-1}\mathbb{r} \cdot \mathbb{h}_{n-k-1}. \end{aligned} \tag{8.9}$$

From (8.7), (8.8), and (8.9), we obtain

$$A_j = w_{j0}s_0 + w_{j1}s_1 + \cdots + w_{j,n-k-1}s_{n-k-1}. \tag{8.10}$$

Thus, the check-sum A_j is simply a *linear sum of the syndrome digits whose coefficients are the first $n - k$ digits of the orthogonal vector* $\mathbb{w}_j$. Based on (8.10), we obtain a different implementation of the one-step majority-logic decoding, as shown in Figure 8.3 (the received vector can be shifted into the syndrome register from the *right* end). This decoder is called the type-I one-step majority-logic decoder [2]. The error correction procedure is as follows:

Step 1. The syndrome is computed as usual by shifting the received polynomial $\mathbb{r}(X)$ into the syndrome register.

Step 2. The J parity-check sums orthogonal on e_{n-1} are formed by taking proper sums of the syndrome digits. These J check-sums are fed into a J-input majority-logic gate.

Step 3. The first received digit is read out of the buffer register and is corrected by the output of the majority gate. At the same time the syndrome register is also shifted once (with gate 2 on), and the effect e_{n-1} on the syndrome is removed (with gate 3 on). The new contents in the syndrome register form the syndrome of the altered received vector cyclically shifted one place to the right.

Step 4. The new syndrome formed in step 3 is used to decode the next received digit r_{n-2}. The decoder repeats steps 2 and 3. The received digit r_{n-2} is corrected in exactly the same manner as the first received digit r_{n-1} was corrected.

Step 5. The decoder decodes the received vector $\mathbb{r}$ digit by digit in the same manner until a total of n shifts of the buffer and the syndrome registers.

At the completion of the decoding operation, the syndrome register should contain only zeros if the decoder output is a codeword. If the syndrome register

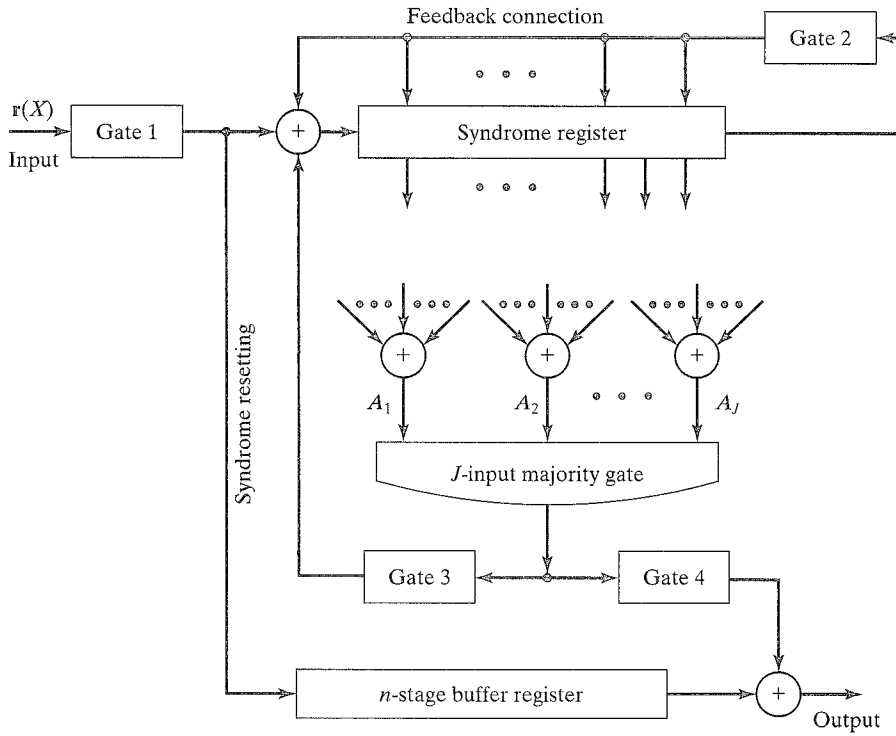

FIGURE 8.3: General type-I one-step majority-logic decoder.

does not contain all zeros at the end of the decoding, an uncorrectable error pattern has been detected. If we are interested in decoding only the received message digits but not the received parity digits, the buffer register needs store only the k received message digits, and it consists of only k stages. In this case, both type-I and type-II decoders require roughly the same amount of complexity.

EXAMPLE 8.2

Consider the (15, 7) BCH code given in Example 8.1. From the vectors $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$, and $\mathbf{w}_4$ that are orthogonal on the digit position 14, we find that the parity-check sums orthogonal on e_{14} are equal to the following sums of syndrome digits:

$$A_1 = s_3, \quad A_2 = s_1 + s_5, \quad A_3 = s_0 + s_2 + s_6, \quad A_4 = s_7.$$

Based on these sums we construct the type-I one-step majority-logic decoder for the (15, 7) BCH code as shown in Figure 8.4. Suppose that the all-zero codeword $(0, 0, \cdots, 0)$ is transmitted, and $\mathbf{r}(X) = X^{13} + X^{14}$ is received. Clearly, there are two errors at locations X^{13} and X^{14}. After the entire received polynomial has entered the syndrome register, the syndrome register contains (0 0 1 1 1 0 0 1). The four parity-check sums orthogonal on e_{14} are

$$A_1 = 1, \quad A_2 = 0, \quad A_3 = 1, \quad A_4 = 1.$$

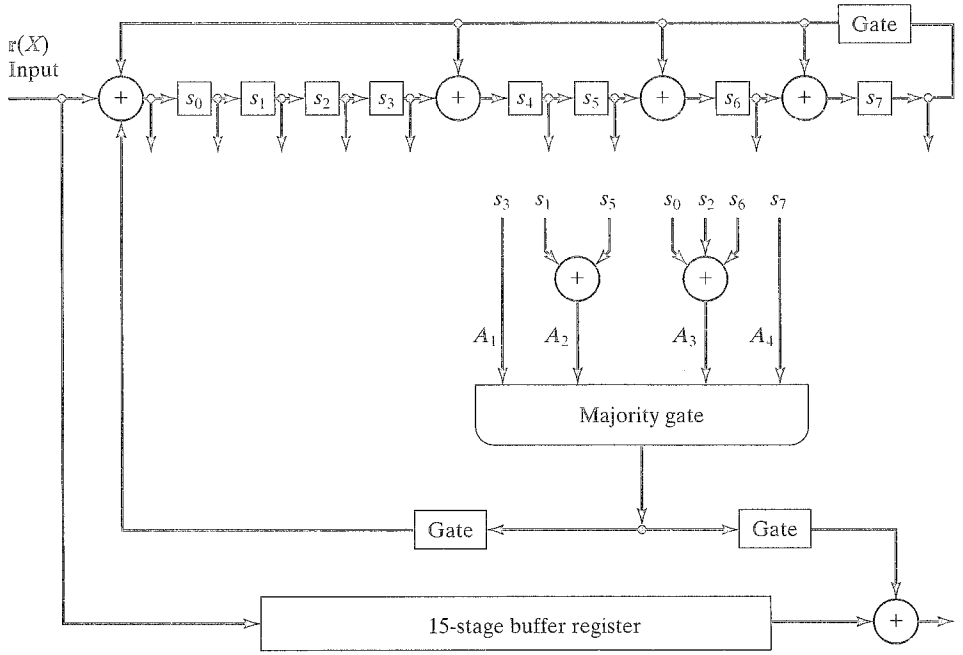

FIGURE 8.4: Type-I one-step majority-logic decoder for (15, 7) BCH code.

Because the majority of these four sums is 1, the output of the majority-logic gate is 1, which is the value of e_{14}. Simultaneously, the buffer and syndrome registers are shifted once; the highest-order received digit $r_{14} = 1$ is then corrected by the output of the majority-logic gate, and the new contents in the syndrome register are (0 0 0 1 0 1 1 1). The new parity-check sums are now

$$A_1^{(1)} = 1, \quad A_2^{(1)} = 1, \quad A_3^{(1)} = 1, \quad A_4^{(1)} = 1.$$

Again, the output of the majority-logic gate is 1, which is the value of e_{13}. Both the buffer and syndrome registers are shifted once more; the received digit r_{13} will be corrected, and the syndrome register will contain only zeros. At this point, both errors have been corrected, and the next 13 received digits are error-free.

One-step majority-logic decoding is most efficient for codes that are completely orthogonalizable, or for codes with large J compared with $d_{\min} - 1$. When J is small compared with $d_{min} - 1$, one-step majority-logic decoding becomes very inefficient, and much of the error-correcting capability of the code is sacrificed. Given a code C, one would like to know the maximum number of parity-check sums orthogonal on an error digit that can be formed. This question is answered by Theorem 8.1.

THEOREM 8.1 Let C be an (n, k) cyclic code whose dual code C_d has minimum distance δ. Then, the number of parity-check sums orthogonal on an

error digit that can be formed, J, is upper bounded by

$$J \leq \left\lfloor \frac{n-1}{\delta - 1} \right\rfloor. \tag{8.11}$$

Proof. Suppose that there exist J vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_J$ in the dual code of C that are orthogonal on the highest-order digit position, X^{n-1}. Because each of these J vectors has a weight of at least δ, the total number of 1's in these J vectors is at least $J\delta$; however, because of the orthogonal structure of these J vectors, the total number of 1's in them cannot exceed $J + (n-1)$. Therefore, we have $J\delta \leq J + (n-1)$. This implies that $J \leq (n-1)/(\delta - 1)$. Because J is an integer, we must have $J \leq \lfloor (n-1)/(\delta - 1) \rfloor$. Q.E.D.

The dual code of the (15, 7) BCH code has a minimum distance of 4. Therefore, the maximum number of parity-check sums orthogonal on an error digit is upper bounded by $\lfloor 14/3 \rfloor = 4$. This proves our claim in Example 8.1 that $J = 4$ is the maximum number of parity-check sums orthogonal on an error digit that can be formed for the (15, 7) BCH code.

If it is possible to form J parity-check sums orthogonal on an error digit for a cyclic code, then the code has a minimum distance of at least $J + 1$ (Massey bound [2]). The proof of this statement is left as a problem.

As we pointed out earlier in this section, one-step majority-logic decoding is most effective for cyclic codes that are completely orthogonalizable. Unfortunately, there exist very few good cyclic codes in this category. The double-error-correcting (15, 7) code considered in Example 8.1 is the only known BCH code that is completely orthogonalizable in one step. Several small classes of one-step majority-logic decodable cyclic codes are presented in the next two sections. Two of the classes are proved to be completely orthogonalizable.

8.2 A CLASS OF ONE-STEP MAJORITY-LOGIC DECODABLE CODES

In this section we present a class of one-step majority-logic decodable cyclic codes whose construction is based on a certain symmetry property.

Let C be an (n, k) cyclic code generated by $g(X)$, where $n = 2^m - 1$. We may extend each vector $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ in C by adding an *overall parity-check digit*, denoted by v_∞, to its left. The overall parity-check digit v_∞ is defined as the modulo-2 sum of all the digits of $\mathbf{v}$ (i.e., $v_\infty = v_0 + v_1 + \cdots + v_{n-1}$). Adding v_∞ to $\mathbf{v}$ results in the following vector of $n + 1 = 2^m$ components:

$$\mathbf{v}_e = (v_\infty, v_0, v_1, \cdots, v_{n-1}).$$

The overall parity-check digit is 1 if the weight of $\mathbf{v}$ is odd, and it is 0 if the weight of $\mathbf{v}$ is even. The 2^k extended vectors form an $(n+1, k)$ linear code, denoted by C_e, which is called an *extension* of C. Clearly, the codewords of C_e have even weight.

Let α be a primitive element in the Galois field $GF(2^m)$. We may number the components of a vector $\mathbf{v}_e = (v_\infty, v_0, v_1, \cdots, v_{2^m-2})$ in C_e by the elements of $GF(2^m)$ as follows: the component v_∞ is numbered $\alpha^\infty = 0$, the component v_0 is numbered $\alpha = 1$, and for $1 \leq i < 2^m - 1$, the component v_i is numbered α^i. We

call these numbers the *location numbers*. Let Y denote the location of a component of v_e. Consider a permutation that carries the component of v_e at the location Y to the location $Z = aY + b$, where a and b are elements from the field of $GF(2^m)$, and $a \neq 0$. This permutation is called an *affine permutation* [14]. Application of an affine permutation to a vector of 2^m components results in another vector of 2^m components.

EXAMPLE 8.3

Consider the following vector of 16 components, which are numbered with the elements of $GF(2^4)$ (using Table 2.8):

α^∞	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
(1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0).

Now, we apply the affine permutation

$$Z = \alpha Y + \alpha^{14}$$

to the components of the preceding vector. The resultant vector is

α^∞	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
(0	0	1	1	0	0	0	0	1	0	1	0	1	0	0	1).

For example, the component at the location $Y = \alpha^8$ is carried to the location

$$Z = \alpha \cdot \alpha^8 + \alpha^{14} = \alpha^9 + \alpha^{14} = \alpha^4.$$

An extended cyclic code C_e of length 2^m is said to be *invariant* under the group of affine permutations if every affine permutation carries every codeword in C_e into another codeword in C_e. In the following discussion we state a necessary and sufficient condition for an extended cyclic code of length 2^m to be invariant under the affine permutations.

Let h be a nonnegative integer less than 2^m. The *radix*-2 (binary) expansion of h is

$$h = \delta_0 + \delta_1 2 + \delta_2 2^2 + \cdots + \delta_{m-1} 2^{m-1},$$

where $\delta_i = 0$ or 1 for $0 \leq i < m$. Let h' be another nonnegative integer less than 2^m whose radix-2 expansion is

$$h' = \delta'_0 + \delta'_1 2 + \delta'_2 2^2 + \cdots + \delta'_{m-1} 2^{m-1}.$$

The integer h' is said to be a *descendant* of h if $\delta'_i \leq \delta_i$ for $0 \leq i < m$.

EXAMPLE 8.4

Let $m = 5$. The integer 21 has the following radix-2 expansion:

$$21 = 1 + 0 \cdot 2 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4.$$

The following integers are proper descendants of 21:

$$\begin{aligned}
20 &= 0+0\cdot 2+1\cdot 2^2+0\cdot 2^3+1\cdot 2^4,\\
17 &= 1+0\cdot 2+0\cdot 2^2+0\cdot 2^3+1\cdot 2^4,\\
16 &= 0+0\cdot 2+0\cdot 2^2+0\cdot 2^3+1\cdot 2^4,\\
5 &= 1+0\cdot 2+1\cdot 2^2+0\cdot 2^3+0\cdot 2^4,\\
4 &= 0+0\cdot 2+1\cdot 2^2+0\cdot 2^3+0\cdot 2^4,\\
1 &= 1+0\cdot 2+0\cdot 2^2+0\cdot 2^3+0\cdot 2^4,\\
0 &= 0+0\cdot 2+0\cdot 2^2+0\cdot 2^3+0\cdot 2^4.
\end{aligned}$$

Let $\Delta(h)$ denote the set of all nonzero proper descendants of h. The following theorem characterizes a necessary and sufficient condition for the extension C_e of a cyclic code C of length 2^m-1 to be invariant under the affine group of permutations.

THEOREM 8.2 [4, 5] Let C be a cyclic code of length $n = 2^m - 1$ generated by $g(X)$. Let C_e be the extended code obtained from C by appending an overall parity-check digit. Let α be a primitive element of the Galois field $GF(2^m)$. Then, the extended code C_e is invariant under the affine permutations if and only if for every α^h that is a root of the generator polynomial $g(X)$ of C and for every h' in $\Delta(h)$, $\alpha^{h'}$ is also a root of $g(X)$, and $\alpha^0 = 1$ is not a root of $g(X)$.

The proof of this theorem is omitted here. For a proof, the reader is referred to [4] and [5]. A cyclic code of length $2^m - 1$ whose generator polynomial satisfies the conditions given in Theorem 8.2 is said to have the *doubly transitive invariant (DTI) property*. RM codes and extended primitive BCH codes are invariant under the affine permutations [4, 5].

Given a code C_e of length $n = 2^m$ that is invariant under the affine permutations, the code C obtained by deleting the first digit from each vector of C_e is cyclic. To see this, we apply the permutation $Z = \alpha Y$ to a vector $(v_\infty, v_0, v_1, \cdots, v_{2^m-2})$ in C_e. This permutation keeps the component v_∞ at the same location α^∞ but *cyclically shifts* the other $2^m - 1$ components one place to the right. The resultant vector is

$$(v_\infty, v_{2^m-2}, v_0, v_1, \cdots, v_{2^m-3}),$$

which is also in C_e. Clearly, if we delete v_∞ from each vector of C_e, we obtain a cyclic code of length $2^m - 1$.

Now, we are ready to present a class of one-step majority-logic decodable codes whose dual codes have the DTI property. Let J and L be two factors of $2^m - 1$ such that $J \cdot L = 2^m - 1$. Clearly, both J and L are odd. The polynomial $X^{2^m-1} + 1$ can be factored as follows:

$$X^{2^m-1} + 1 = (1 + X^J)(1 + X^J + X^{2J} + \cdots + X^{(L-1)J}).$$

Let

$$\pi(X) = 1 + X^J + X^{2J} + \cdots + X^{(L-1)J}. \tag{8.12}$$

From Theorem 2.12 we know that the $2^m - 1$ nonzero elements of $GF(2^m)$ form the $2^m - 1$ roots of $X^{2^m-1} + 1$. Let α be a primitive element of $GF(2^m)$. Because $(\alpha^L)^J = \alpha^{2^m-1} = 1$, the polynomial $X^J + 1$ has $\alpha^0 = 1, \alpha^L, \alpha^{2L}, \cdots, \alpha^{(J-1)L}$ as all its roots. Therefore, the polynomial $\pi(X)$ has α^h as a root if and only if h is not a multiple of L, and $0 < h < 2^m - 1$.

Now, we form a polynomial $H(X)$ over $GF(2)$ as follows: $H(X)$ has α^h as a root if and only if (1) α^h is a root of $\pi(X)$ and (2) for every h' in $\Delta(h)$, $\alpha^{h'}$ is also a root of $\pi(X)$. Let α^i be a root of $H(X)$. Let $\phi_i(X)$ be the minimal polynomial of α^i. Then,

$$H(X) = \text{LCM}\{\text{minimal polynomials } \phi_i(X) \text{ of the roots of } H(X)\}. \tag{8.13}$$

It is clear that $H(X)$ divides $\pi(X)$ and is a factor of $X^{2^m-1} + 1$. Let C' be the cyclic code of length $2^m - 1$ generated by $H(X)$. It follows from Theorem 8.2 that C' has the DTI property. Thus, the extended code C'_e of C' is invariant under the group of affine permutations. Let C be the dual code of C'. Then, C is also cyclic. Since $H(X)$ divides $X^{2^m-1} + 1$, we have

$$X^{2^m-1} + 1 = G(X)H(H).$$

Let k be the degree of $H(X)$. Then, the degree of $G(X)$ is $2^m - 1 - k$. The generator polynomial of C is

$$g(X) = X^{2^m-k-1}G(X^{-1}), \tag{8.14}$$

which is the reciprocal of $G(X)$. Next, we will show that the code C is one-step majority-logic decodable and is capable of correcting $t_{ML} = \lfloor J/2 \rfloor$ or fewer errors where $J = (2^m - 1)/L$.

First, we need to determine J vectors from C' (the dual of C) that are orthogonal on the digit at location α^{2^m-2}. Because $\pi(X)$ is a multiple of $H(X)$ and has degree less than $2^m - 1$, it is a code polynomial in C' generated by $H(X)$. Clearly, the polynomials $X\pi(X), X^2\pi(X), \cdots, X^{J-1}\pi(X)$ are also code polynomials in C'. From (8.12) we see that, for $i \neq j$, $X^i\pi(X)$ and $X^j\pi(X)$ do not have any common term. Let $v_0, v_1, \cdots, v_{J-1}$ be the J corresponding codewords of $\pi(X), X\pi(X), \cdots, X^{J-1}\pi(X)$. The weight of each of these vectors is L. Adding an overall parity-check digit to each of these vectors, we obtain J vectors $u_0, u_1, \cdots, u_{J-1}$ of length 2^m that are codewords in the extension C'_e of C'. Since L is odd, the overall parity-check digit of each u_i is 1. Thus, the J vectors $u_0, u_1, \cdots, u_{J-1}$ have the following properties:

1. They all have 1 at location α^∞ (the overall parity-check digit position).
2. One and only one vector has a 1 at the location α^j for $0 \leq j < 2^m - 1$.

Therefore, they form J vectors orthogonal on the digit at location α^∞. Now, we apply the affine permutation

$$Z = \alpha Y + \alpha^{2^m-2}$$

to $u_0, u_1, \cdots, u_{J-1}$. This permutation carries $u_0, u_1, \cdots, u_{J-1}$ into J vectors $z_0, z_1, \cdots, z_{J-1}$, which are also in C'_e (since C'_e is invariant under the group of affine permutations). Note that the permutation $Z = \alpha Y + \alpha^{2^m-2}$ carries the component of u_i at location α^∞ to location α^{2^m-2}. Thus, the vectors $z_0, z_1, \cdots, z_{J-1}$ are

orthogonal on the digit at location α^{2^m-2}. Deleting the digit at location α^{∞} from $\mathbb{z}_0, \mathbb{z}_1, \cdots, \mathbb{z}_{J-1}$, we obtain J vectors $\mathbb{w}_0, \mathbb{w}_1, \cdots, \mathbb{w}_{J-1}$ of length $2^m - 1$, which are vectors in C' and are orthogonal on the digit at location α^{2^m-2}. From these J vectors we can form J parity-check sums orthogonal on the error digit e_{2^m-2}. Therefore, the cyclic code C generated by

$$\mathbb{g}(X) = X^{2^m-k-1}G(X^{-1})$$

is one-step majority-logic decodable and is capable of correcting $t_{ML} = \lfloor J/2 \rfloor$ or fewer errors. For convenience, we call this code a *type*-0 one-step majority-logic decodable DTI code.

EXAMPLE 8.5

Let $m = 5$. We can factor the polynomial $X^{2^4-1} + 1 = X^{15} + 1$ as

$$X^{15} + 1 = (1 + X^5)(1 + X^5 + X^{10}).$$

Thus, $J = 5$, $L = 3$, and $\pi(X) = 1 + X^5 + X^{10}$. Let α be a primitive element in $GF(2^4)$ (use Table 2.8) whose minimal polynomial is $\phi_1(X) = 1 + X + X^4$. Because $\alpha^{15} = 1$, the polynomial $1 + X^5$ has $1, \alpha^3, \alpha^6, \alpha^9$, and α^{12} as all its roots. The polynomial $\pi(X)$ has $\alpha, \alpha^2, \alpha^4, \alpha^5, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{11}, \alpha^{13}$ and α^{14} as roots. Next, we determine the polynomial $H(X)$. From the conditions on the roots of $H(X)$, we find that $H(X)$ has $\alpha, \alpha^2, \alpha^4, \alpha^5, \alpha^8$, and α^{10} as its roots. The roots $\alpha, \alpha^2, \alpha^4$, and α^8 are conjugates, and they have the same minimal polynomial, $\phi_1(X) = 1 + X + X^4$. The roots α^5 and α^{10} are conjugates, and they have $\phi_5(X) = 1 + X + X^2$ as their minimal polynomial. Hence,

$$\begin{aligned} H(X) &= \phi_1(X)\phi_5(X) = (1 + X + X^4)(1 + X + X^2) \\ &= 1 + X^3 + X^4 + X^5 + X^6. \end{aligned}$$

We can easily check that $H(X)$ divides $\pi(X)$, and, in fact, $\pi(X) = (1 + X^3 + X^4)H(X)$. Also, $H(X)$ divides $X^{15} + 1$, and $X^{15} + 1 = (1 + X^3 + X^4 + X^5 + X^8 + X^9)H(X)$. Thus, $G(X) = 1 + X^3 + X^4 + X^5 + X^8 + X^9$. The polynomial $H(X)$ generates a (15, 9) cyclic code C', which has the DTI property. The polynomials $\pi(X)$, $X\pi(X)$, $X^2\pi(X)$, $X^3\pi(X)$, and $X^4\pi(X)$ are code polynomials in C'. The dual code of C', C, is generated by

$$\mathbb{g}(X) = X^9G(X^{-1}) = 1 + X + X^4 + X^5 + X^6 + X^9.$$

Thus, C is a (15, 6) cyclic code.

To decode C, we need to determine parity-check sums orthogonal on e_{14}. The vectors corresponding to $\pi(X)$, $X\pi(X)$, $X^2\pi(X)$, $X^3\pi(X)$, and $X^4\pi(X)$ are

	Location Numbers														
	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbb{v}_0 =$	(1	0	0	0	0	1	0	0	0	0	1	0	0	0	0)
$\mathbb{v}_1 =$	(0	1	0	0	0	0	1	0	0	0	0	1	0	0	0)
$\mathbb{v}_2 =$	(0	0	1	0	0	0	0	1	0	0	0	0	1	0	0)
$\mathbb{v}_3 =$	(0	0	0	1	0	0	0	0	1	0	0	0	0	1	0)
$\mathbb{v}_4 =$	(0	0	0	0	1	0	0	0	0	1	0	0	0	0	1),

which are codewords in C'. Adding an overall parity-check digit to these vectors, we obtain the following vectors:

	Location Numbers														
	α^{∞}	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbb{u}_0 =$	(1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0)
$\mathbb{u}_1 =$	(1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0)
$\mathbb{u}_2 =$	(1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0)
$\mathbb{u}_3 =$	(1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0)
$\mathbb{u}_4 =$	(1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1),

which are vectors in C'_e (the extension of C'). Now, we apply the affine permutation $Z = \alpha Y + \alpha^{14}$ to permute the components of $\mathbb{u}_0$, $\mathbb{u}_1$, $\mathbb{u}_2$, $\mathbb{u}_3$, and $\mathbb{u}_4$. The permutation results in the following vectors:

	Location Numbers														
	α^{∞}	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbb{z}_0 =$	(0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1)
$\mathbb{z}_1 =$	(0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1)
$\mathbb{z}_2 =$	(0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1)
$\mathbb{z}_3 =$	(1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1)
$\mathbb{z}_4 =$	(0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1),

which are also in C'_e. Deleting the overall parity-check digits from these vectors, we obtain the following vectors in C':

	Location Numbers														
	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbb{w}_0 =$	(0	0	0	0	0	0	0	1	1	0	1	0	0	0	1)
$\mathbb{w}_1 =$	(0	1	0	0	0	1	0	0	0	0	0	0	0	1	1)
$\mathbb{w}_2 =$	(1	0	1	0	0	0	1	0	0	0	0	0	0	0	1)
$\mathbb{w}_3 =$	(0	0	0	0	1	0	0	0	0	1	0	0	0	0	1)
$\mathbb{w}_4 =$	(0	0	0	1	0	0	0	0	0	0	0	1	1	0	1)

We see that these vectors are orthogonal on the digit at location α^{14}.

Let $\mathbb{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14})$ be the received vector. Then, the parity-check sums orthogonal on e_{14} are

$$\begin{aligned}
A_0 &= \mathbb{r} \cdot \mathbb{w}_0 = r_7 + r_8 + r_{10} + r_{14}, \\
A_1 &= \mathbb{r} \cdot \mathbb{w}_1 = r_1 + r_5 + r_{13} + r_{14}, \\
A_2 &= \mathbb{r} \cdot \mathbb{w}_2 = r_0 + r_2 + r_6 + r_{14}, \\
A_3 &= \mathbb{r} \cdot \mathbb{w}_3 = r_4 + r_9 + r_{14}, \\
A_4 &= \mathbb{r} \cdot \mathbb{w}_4 = r_3 + r_{11} + r_{12} + r_{14}.
\end{aligned}$$

Therefore, the (15, 6) cyclic code C generated by $g(X) = 1 + X + X^4 + X^5 + X^6 + X^9$ is one-step majority-logic decodable and is capable of correcting $t_{ML} = \lfloor 5/2 \rfloor = 2$ or fewer errors. It also corrects many error patterns of three errors. The code has a minimum distance of at least $J + 1 = 5 + 1 = 6$; however, the minimum distance of the code is exactly 6. Hence, the code is completely orthogonalizable.

Recall that $\pi(X)$ has α^h as a root if and only if h is not a multiple of L, and $0 < h < 2^m - 1$. Therefore, $\pi(X)$ has the following consecutive powers of α as roots: $\alpha, \alpha^2, \cdots, \alpha^{L-1}$. Because any descendant h' of an integer h is less than h, if $h < L$ and h' in $\Delta(h)$, both α^h and $\alpha^{h'}$ are roots of $\pi(X)$. Consequently, the polynomial $H(X)$ also has $\alpha, \alpha^2, \cdots, \alpha^{L-1}$ as roots. Using the argument that proves the minimum distance of a BCH code, we can show that the minimum distance of C' generated by $H(X)$ is at least L; however, since $\pi(X)$ is a code polynomial of weight L in C', the minimum distance of C' is exactly L. It follows from Theorem 8.1 that the number of parity-check sums orthogonal on an error digit that can be formed for C is upper bounded by

$$\left\lfloor \frac{2^m - 2}{L - 1} \right\rfloor; \tag{8.15}$$

however, $J = (2^m - 1)/L$. Therefore, for large L, J is either equal to or close to the upper bound of (8.15).

In general, it is not known whether the type-0 DTI codes are completely orthogonalizable. There are a number of special cases for which we can prove that the codes are completely orthogonalizable.

The type-0 DTI codes may be modified so that the resultant codes are also one-step majority-logic decodable, and $J - 1$ parity-check sums orthogonal on an error digit can be formed. Recall that the polynomial $H(X)$ does not have $(X + 1)$ as a factor (i.e., it does not have $\alpha^0 = 1$ as a root). Let

$$H_1(X) = (X + 1)H(X). \tag{8.16}$$

The cyclic code C_1' generated by $H_1(X)$ is a subcode of C' generated by $H(X)$. In fact, C_1' consists of the *even-weight* vectors of C' as codewords. Recall that the J orthogonal vectors $\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{J-1}$ in C' are obtained from the vectors $\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_{J-1}$ by deleting the digit at location α^∞. Because $\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_{J-1}$ are orthogonal on the digit at location α^{2^m-2}, there is one and only one vector $\mathbf{z}_i$ that has a 1 at location α^∞. Since $\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_{J-1}$ all have weight $L + 1$ which is even, all but one of the orthogonal vectors $\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{J-1}$ have weight $L + 1$. These $J - 1$ even-weight orthogonal vectors are in C_1'. Therefore, the dual code of C_1', denoted by C_1, is one-step majority-logic decodable, and $J - 1$ parity-check sums orthogonal on an error digit can be formed. Let

$$G_1(X) = \frac{G(X)}{X + 1}. \tag{8.17}$$

Then, the generator polynomial for C_1 is

$$\mathbf{g}_1(X) = X^{2^m-k-2}G_1(X^{-1}) = \frac{\mathbf{g}(X)}{X + 1}, \tag{8.18}$$

where $\mathbf{g}(X)$ is given by (8.14). C_1 is called a *type*-1 DTI code, and its dimension is one greater than that of its corresponding type-0 DTI code.

EXAMPLE 8.6

For $m = 4$ and $J = 5$, the type-1 DTI code C_1 that corresponds to the type-0 DTI code given in Example 8.5 is generated by

$$g_1(X) = \frac{1 + X + X^4 + X^5 + X^6 + X^9}{1 + X}$$
$$= 1 + X^4 + X^6 + X^7 + X^8.$$

It is interesting to note that this code is the (15, 7) BCH code. From Example 8.5 we see that w_3 has odd weight, and therefore it is not a vector in C_1' (the dual of C_1). Hence, the four orthogonal vectors in C_1' are

$$w_0 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1),$$
$$w_1 = (0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1),$$
$$w_2 = (1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1),$$
$$w_4 = (0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1),$$

which are the same four orthogonal vectors given in Example 8.1.

Because the dual code of type-1 DTI code C_1 has a minimum distance of $L + 1$, the number of parity-check sums orthogonal on an error digit that can be formed is upper bounded by

$$\left\lfloor \frac{2^m - 2}{L} \right\rfloor = \left\lfloor \frac{2^m - 1}{L} - \frac{1}{L} \right\rfloor = \left\lfloor J - \frac{1}{L} \right\rfloor = J - 1.$$

Therefore, the number of parity-check sums orthogonal on an error digit that can be formed for a type-1 DTI code is equal to its upper bound. Since J is odd, $\lfloor J/2 \rfloor = \lfloor (J-1)/2 \rfloor$. Thus, both type-0 and type-1 DTI codes have the same majority-logic error-correcting capability.

In general, there is no simple formula for enumerating the number of parity-check digits of the one-step majority-logic decodable DTI codes (type-0 or type-1); however, for two special cases, exact formulas for $n - k$ can be obtained [6]:

Case I. For $m = 2sl$ and $J = 2^l + 1$, the number of parity-check digits of the type-1 DTI code of length $2^m - 1$ is

$$n - k = (2^{s+1} - 1)^l - 1.$$

Case II. For $m = \lambda l$ and $J = 2^l - 1$, the number of parity-check digits of the type-1 DTI code of length $2^m - 1$ is

$$n - k = 2^m - (2^\lambda - 1)^l - 1.$$

A list of one-step majority-logic decodable type-1 DTI codes is given in Table 8.1.

Short-length DTI codes are comparable with BCH codes in efficiency. For example, there exists a (63, 37) one-step majority-logic decodable type-1 DTI code that is capable of correcting four or fewer errors. The corresponding four-error-correcting BCH code of the same length is a (63, 39) code that has two information

TABLE 8.1: Some one-step majority-logic decodable type-1 DTI codes.

n	k	t_{ML}	n	k	t_{ML}
15	9	1	2047	1211	11
	7	2		573	44
63	49	1	4095	3969	1
	37	4		3871	2
	13	10		3753	4
255	225	1		3611	6
	207	2		3367	32
	175	8		2707	17
	37	25		2262	19
	21	42		2074	22
511	343	3		1649	45
	139	36		1393	52
1023	961	1		1377	136
	833	5		406	292
	781	16		101	409
	151	46		43	682
	30	170			

digits more than the (63, 37) type-1 DTI code; however, the decoding circuit for the (63, 39) BCH code is much more complex than for the (63, 37) DTI code. For large block lengths, the DTI codes are much less efficient than the BCH codes of the same length and the same error-correcting capability.

8.3 OTHER ONE-STEP MAJORITY-LOGIC DECODABLE CODES

There are two other small classes of one-step majority-logic decodable cyclic codes: the maximum-length codes and the difference-set codes. Both classes have been proved to be completely orthogonalizable.

8.3.1 Maximum-Length Codes

For any integer $m \geq 3$, there exists a nontrivial maximum-length code with the following parameters:

Block length:	$n = 2^m - 1$,
Number of information digits:	$k = m$,
Minimum distance:	$d = 2^{m-1}$.

The generator polynomial of this code is

$$\mathbf{g}(X) = \frac{X^n + 1}{\mathbf{p}(X)}, \tag{8.19}$$

where $\mathbf{p}(X)$ is a primitive polynomial of degree m. This code consists of the all-zero codeword and $2^m - 1$ codewords of weight 2^{m-1} (see Problem 8.11). Maximum-length

codes were first shown to be majority-logic decodable by Yale [7] and Zierler [8] independently. The dual code of the maximum-length code is a $(2^m - 1, 2^m - m - 1)$ cyclic code generated by the reciprocal of the parity polynomial $\mathbb{p}(X)$,

$$\mathbb{p}^*(X) = X^m \mathbb{p}(X^{-1}).$$

Because $\mathbb{p}^*(X)$ is also a primitive polynomial of degree m, the dual code is thus a Hamming code. Therefore, the null space of the maximum-length code contains vectors of weight 3 (this is the minimum weight). Now, consider the following set of distinct code polynomials:

$$Q = \{\mathbb{w}(X) = X^i + X^j + X^{n-1} : 0 \leq i < j < n - 1\} \tag{8.20}$$

in the Hamming code generated by $\mathbb{p}^*(X)$. No two polynomials in Q can have any common terms except the term X^{n-1}. Otherwise, the sum of these two polynomials would be a code polynomial of only two terms in the Hamming code. This is impossible, since the minimum weight of a Hamming code is 3. Therefore, the set Q contains polynomials orthogonal on the highest-order-digit position X^{n-1}. To find $\mathbb{w}(X)$, we start with a polynomial $X^{n-1} + X^j$ for $0 \leq j < n - 1$ and then determine X^i such that $X^{n-1} + X^j + X^i$ is divisible by $\mathbb{p}^*(X)$, as follows: Divide $X^{n-1} + X^j$ by $\mathbb{p}^*(X)$ step-by-step with long division until a single term X^i appears at the end of a certain step. Then, $\mathbb{w}(X) = X^{n-1} + X^j + X^i$ is a polynomial orthogonal on digit position X^{n-1}. Clearly, if we started with $X^{n-1} + X^i$, we would obtain the same polynomial $\mathbb{w}(X)$. Thus, we can find $(n - 1)/2 = 2^{m-1} - 1$ polynomials orthogonal on digit position X^{n-1}. That is, $J = 2^{m-1} - 1$ parity-check sums orthogonal on e_{n-1} can be formed. Because the maximum-length code generated by $\mathbb{g}(X)$ of (8.19) has a minimum distance of exactly 2^{m-1}, it is completely orthogonalizable. The code is capable of correcting $t_{ML} = 2^{m-2} - 1$ or fewer errors with one-step majority-logic decoding.

EXAMPLE 8.7

Consider the maximum-length code with $m = 4$ and parity polynomial $\mathbb{p}(X) = 1 + X + X^4$. This code has block length $n = 15$ and minimum distance $d = 8$. The generator polynomial of this code is

$$\begin{aligned}\mathbb{g}(X) &= \frac{X^{15} + 1}{\mathbb{p}(X)} \\ &= 1 + X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{11}.\end{aligned}$$

It is a (15, 4) code. The null space of this code is generated by

$$\mathbb{p}^*(X) = X^4 \mathbb{p}(X^{-1}) = X^4 + X^3 + 1.$$

We divide $X^{14} + X^{13}$ by $\mathbb{p}^*(X) = X^4 + X^3 + 1$ with long division as follows:

$$\begin{array}{r|l} & X^{10} \\ \hline X^4 + X^3 + 1 & X^{14} + X^{13} \\ & X^{14} + X^{13} + X^{10} \\ \hline & X^{10} \text{ (stop).} \end{array}$$

A single term, X^{10}, appears at the end of the first step of the long division. Then, $w_1(X) = X^{14} + X^{13} + X^{10}$ is a polynomial orthogonal on X^{14}. Now, we divide $X^{14} + X^{12}$ by $p^*(X)$:

$$
\begin{array}{rl}
 & X^{10} + X^9 + X^6 \\
\cline{2-2}
X^4 + X^3 + 1 \;\big| & X^{14} \qquad\quad + X^{12} \\
 & X^{14} + X^{13} \qquad\quad + X^{10} \\
\cline{2-2}
 & \qquad X^{13} + X^{12} + X^{10} \\
 & \qquad X^{13} + X^{12} \qquad\quad + X^9 \\
\cline{2-2}
 & \qquad\qquad\quad X^{10} + X^9 \\
 & \qquad\qquad\quad X^{10} + X^9 + X^6 \\
\cline{2-2}
 & \qquad\qquad\qquad\quad X^6 \text{ (stop).}
\end{array}
$$

Then, $w_2(X) = X^{14} + X^{12} + X^6$ is another polynomial orthogonal on X^{14}. The rest of the polynomials orthogonal on X^{14} can be found in the same manner; they are

$$
\begin{aligned}
w_3(X) &= 1 + X^{11} + X^{14}, \quad & w_4(X) &= X^4 + X^9 + X^{14}, \\
w_5(X) &= X + X^8 + X^{14}, \quad & w_6(X) &= X^5 + X^7 + X^{14}, \\
w_7(X) &= X^2 + X^3 + X^{14}.
\end{aligned}
$$

From the set of polynomials orthogonal on X^{14}, we obtain the following seven parity-check sums orthogonal on e_{14}:

$$
\begin{aligned}
A_1 &= e_{10} + e_{13} + e_{14}, \\
A_2 &= e_6 + e_{12} + e_{14}, \\
A_3 &= e_0 + e_{11} + e_{14}, \\
A_4 &= e_4 + e_9 + e_{14}, \\
A_5 &= e_1 + e_8 + e_{14}, \\
A_6 &= e_5 + e_7 + e_{14}, \\
A_7 &= e_2 + e_3 + e_{14}.
\end{aligned}
$$

In terms of syndrome bits, we have $A_1 = s_{10}$, $A_2 = s_6$, $A_3 = s_0$, $A_4 = s_4 + s_9$, $A_5 = s_1 + s_8$, $A_6 = s_5 + s_7$, and $A_7 = s_2 + s_3$. The code is capable of correcting three or fewer errors by one-step majority-logic decoding.

8.3.2 Difference-Set Codes

The formulation of difference-set codes is based on the construction of a *perfect difference set*. Let $P = \{l_0, l_1, l_2, \cdots, l_q\}$ be a set of $q + 1$ nonnegative integers such that

$$0 \le l_0 < l_1 < l_2 < \cdots < l_q \le q(q + 1).$$

From this set of integers it is possible to form $q(q + 1)$ *ordered differences* as follows:

$$D = \{l_j - l_i : j \ne i\}.$$

Obviously, half the differences in D are positive, and the other half are negative. The set P is said to be a *perfect simple difference set of order* q if and only if it has the following properties:

1. All the positive differences in D are distinct.
2. All the negative differences in D are distinct.
3. If $l_j - l_i$ is a negative difference in D, then $q(q+1)+1+(l_j - l_i)$ is not equal to any positive difference in D.

Clearly, it follows from the definition that $P' = \{0, l_1 - l_0, l_2 - l_0, \cdots, l_q - l_0\}$ is also a perfect simple difference set.

EXAMPLE 8.8

Consider the set $P = \{0, 2, 7, 8, 11\}$ with $q = 4$. The $4 \cdot 5 = 20$ ordered differences are

$$D = \{2, 7, 8, 11, 5, 6, 9, 1, 4, 3, -2, -7, -8, -11, -5, -6, -9, -1, -4, -3\}.$$

It can be checked easily that P satisfies all three properties of a perfect simple difference set.

Singer [9] has constructed perfect difference sets for order $q = p^s$, where p is a prime and s is any positive integer (see also [10]). In the following discussion we shall be concerned only with $q = 2^s$.

Let $P = \{l_0 = 0, l_1, l_2, \cdots, l_{2^s}\}$ be a perfect simple difference set of order 2^s. We define the polynomial

$$\mathbb{z}(X) = 1 + X^{l_1} + X^{l_2} + \cdots + X^{l_{2^s}}. \tag{8.21}$$

Let $n = 2^s(2^s + 1) + 1$ and $\mathbb{h}(X)$ be the greatest common divisor of $\mathbb{z}(X)$ and $X^n + 1$; that is,

$$\begin{aligned} \mathbb{h}(X) &= \mathrm{GCD}\{\mathbb{z}(X), X^n + 1\} \\ &= 1 + h_1 X + h_2 X^2 + \cdots + h_{k-1} X^{k-1} + X^k. \end{aligned} \tag{8.22}$$

Then a difference-set code of length n is defined as the cyclic code generated by

$$\begin{aligned} \mathbb{g}(X) &= \frac{X^n + 1}{\mathbb{h}(X)} \\ &= 1 + g_1 X + g_2 X^2 + \cdots + X^{n-k}. \end{aligned} \tag{8.23}$$

This code has the following parameters:

Code length: $n = 2^{2s} + 2^s + 1$

Number of parity-check digits: $n - k = 3^s + 1$

Minimum distance: $d = 2^s + 2$.

Difference-set codes were discovered by Rudolph [11] and Weldon [12] independently. The formula for the number of parity-check digits was derived by Graham and MacWilliams [13].

EXAMPLE 8.9

In Example 8.8 we showed that the set $P = \{0, 2, 7, 8, 11\}$ is a perfect simple difference set of order $q = 2^2$. Let $\mathbf{z}(X) = 1 + X^2 + X^7 + X^8 + X^{11}$. Then,

$$\begin{aligned}\mathbf{h}(X) &= \text{GCD}\{1 + X^2 + X^7 + X^8 + X^{11}, 1 + X^{21}\} \\ &= 1 + X^2 + X^7 + X^8 + X^{11}.\end{aligned}$$

The generator polynomial of the difference-set code of length $n = 21$ is

$$\begin{aligned}\mathbf{g}(X) &= \frac{X^{21} + 1}{\mathbf{h}(X)} \\ &= 1 + X^2 + X^4 + X^6 + X^7 + X^{10}.\end{aligned}$$

Thus, the code is a (21, 11) cyclic code.

Let $\mathbf{h}^*(X) = X^k\mathbf{h}(X^{-1})$ be the reciprocal polynomial of $\mathbf{h}(X)$. Then, the $(n, n-k)$ cyclic code generated by $\mathbf{h}^*(X)$ is the null space of the difference-set code generated by $\mathbf{g}(X)$ of (8.23). Let

$$\begin{aligned}\mathbf{z}^*(X) &= X^{l_{2^s}}\mathbf{z}(X^{-1}) \\ &= 1 + \cdots + X^{l_{2^s}-l_2} + X^{l_{2^s}-l_1} + X^{l_{2^s}}.\end{aligned} \tag{8.24}$$

Because $\mathbf{z}(X)$ is divisible by $\mathbf{h}(X)$, $\mathbf{z}^*(X)$ is divisible by $\mathbf{h}^*(X)$. Thus, $\mathbf{z}^*(X)$ is in the null space of the difference-set code generated by $\mathbf{g}(X)$ of (8.23). Let

$$\begin{aligned}\mathbf{w}_0(X) &= X^{n-1-l_{2^s}}\mathbf{z}^*(X) \\ &= X^{n-1-l_{2^s}} + \cdots + X^{n-1-l_2} + X^{n-1-l_1} + X^{n-1}.\end{aligned}$$

Obviously, $\mathbf{w}_0(X)$ is divisible by $\mathbf{h}^*(X)$ and is also in the null space of the difference-set code generated by $\mathbf{g}(X)$ of (8.23). Now, let

$$\begin{aligned}\mathbf{w}_i(X) = {} & X^{l_i-l_{i-1}-1} + X^{l_i-l_{i-2}-1} + \cdots + X^{l_i-l_1-1} + X^{l_i-1} \\ & + X^{n-1-l_{2^s}+l_i} + X^{n-1-l_{2^s-1}+l_i} + \cdots + X^{n-1}\end{aligned} \tag{8.25}$$

be the vector obtained by shifting $\mathbf{w}_0(X)$ cyclically to the right l_i times. Because $\{l_0 = 0, l_1, l_2, \cdots, l_{2^s}\}$ is a perfect difference set, no two polynomials $\mathbf{w}_i(X)$ and $\mathbf{w}_j(X)$ for $i \neq j$ can have any common term except X^{n-1}. Thus, $\mathbf{w}_0(X), \mathbf{w}_1(X), \cdots, \mathbf{w}_{2^s}(X)$ form a set of $J = 2^s + 1$ polynomials orthogonal on the digit at position X^{n-1}. Since the code generated by $\mathbf{g}(X)$ of (8.23) is proved to have a minimum distance of $2^s + 2$, it is completely orthogonalizable and is capable of correcting $t_{ML} = 2^{s-1}$ or fewer errors.

EXAMPLE 8.10

Consider the code given in Example 8.9, which is specified by the perfect difference set $P = \{0, 2, 7, 8, 11\}$ of order 2^2. Thus, we have

$$\mathbf{z}^*(X) = X^{11}\mathbf{z}(X^{-1}) = 1 + X^3 + X^4 + X^9 + X^{11}$$

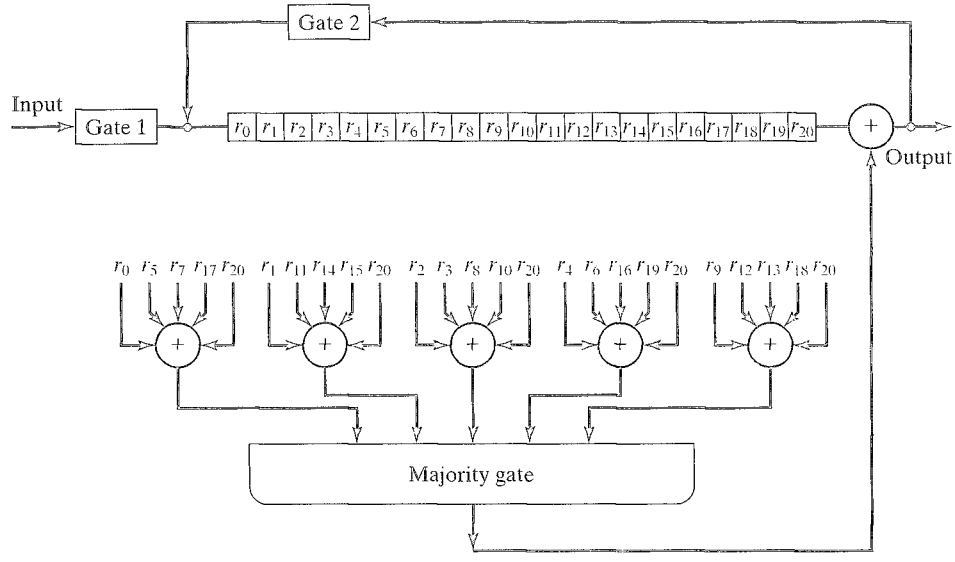

FIGURE 8.5: Type-II majority-logic decoder for the (21, 11) difference-set code.

and

$$\mathrm{w}_0(X) = X^9 \mathrm{z}^*(X) = X^9 + X^{12} + X^{13} + X^{18} + X^{20}.$$

By shifting $\mathrm{w}_0(X)$ cyclically to the right 2 times, 7 times, 8 times, and 11 times, we obtain

$$\begin{aligned}
\mathrm{w}_1(X) &= X + X^{11} + X^{14} + X^{15} + X^{20}, \\
\mathrm{w}_2(X) &= X^4 + X^6 + X^{16} + X^{19} + X^{20}, \\
\mathrm{w}_3(X) &= 1 + X^5 + X^7 + X^{17} + X^{20}, \\
\mathrm{w}_4(X) &= X^2 + X^3 + X^8 + X^{10} + X^{20}.
\end{aligned}$$

Clearly, $\mathrm{w}_0(X)$, $\mathrm{w}_1(X)$, $\mathrm{w}_2(X)$, $\mathrm{w}_3(X)$, and $\mathrm{w}_4(X)$ are five polynomials orthogonal on X^{20}. From these five orthogonal polynomials, we can form the following five parity-check sums orthogonal on e_{20}:

$$\begin{aligned}
A_1 &= s_9 &&= e_9 + e_{12} + e_{13} + e_{18} + e_{20}, \\
A_2 &= s_1 &&= e_1 + e_{11} + e_{14} + e_{15} + e_{20}, \\
A_3 &= s_4 + s_6 &&= e_4 + e_6 + e_{16} + e_{19} + e_{20}, \\
A_4 &= s_0 + s_5 + s_7 &&= e_0 + e_5 + e_7 + e_{17} + e_{20}, \\
A_5 &= s_2 + s_3 + s_8 &&= e_2 + e_3 + e_8 + e_{10} + e_{20}.
\end{aligned}$$

A type-II majority-logic decoder for this code is shown in Figure 8.5. The construction of a type-I decoder for this code is left as an exercise.

Difference-set codes are nearly as powerful as the best known cyclic codes in the range of practical interest. Unfortunately, there are relatively few codes with useful parameters in this class. A list of the first few codes with their generator

TABLE 8.2: A list of binary difference-set cyclic codes.

s	n	k	d	t	Generator polynomial, $g(X)$*	Associated difference set
1	7	3	4	1	0, 2, 3, 4	0, 2, 3
2	21	11	6	2	0, 2, 4, 6, 7, 10	0, 2, 7, 8, 11
3	73	45	10	4	0, 2, 4, 6, 8, 12, 16, 22, 25, 28	0, 2, 10, 24, 25, 29, 36, 42, 45
4	273	191	18	8	0, 4, 10, 18, 22, 24, 34, 36, 40, 48, 52, 56, 66, 67, 71, 76, 77, 82	0, 18, 24, 46, 50, 67, 103, 112, 115, 126, 128, 159, 166, 167, 186, 196, 201
5	1057	813	34	16	0, 1, 3, 4, 5, 11, 14, 17, 18, 22, 23, 26, 27, 28, 32, 33, 35, 37, 39, 41, 43, 45, 47, 48, 51, 52, 55, 59, 62, 68, 70, 71, 72, 74, 75, 76, 79, 81, 83, 88, 95, 98, 101, 103, 105, 106, 108, 111, 114, 115, 116, 120, 121, 122, 123, 124, 126, 129, 131, 132, 135, 137, 138, 141, 142, 146, 147, 149, 150, 151, 153, 154, 155, 158, 160, 161, 164, 165, 166, 167, 169, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 186, 188, 189, 191, 193, 194, 195, 198, 199, 200, 201, 202, 203, 208, 209, 210, 211, 212, 214, 216, 222, 224, 226, 228, 232, 234, 236, 242, 244	0, 1, 3, 7, 15, 31, 54, 63, 109, 127, 138, 219, 255, 277, 298, 338, 348, 439, 452, 511, 528, 555, 597, 677, 697, 702, 792, 897, 905, 924, 990, 1023

*Each generator polynomial is represented by the exponents of its nonzero terms. For example, $\{0, 2, 3, 4\}$ represents $g(X) = 1 + X^2 + X^3 + X^4$.

polynomials and their corresponding perfect simple difference sets is given in Table 8.2.

Other one-step majority-logic decodable cyclic codes will be presented in Section 8.5.

8.4 MULTIPLE-STEP MAJORITY-LOGIC DECODING

The one-step majority-logic decoding for a cyclic code is based on the condition that a set of J parity-check sums orthogonal on a single error digit can be formed. This decoding method is effective for codes that are completely orthogonalizable or for codes with large J compared with their minimum distance $d_{\min}$. Unfortunately, only several small classes of cyclic codes are known to be in this category; however, the concept of parity-check sums orthogonal on a single error digit can be generalized in such a way that many cyclic codes can be decoded by employing several *levels* of majority-logic gates.

Let $E = \{e_{i_1}, e_{i_2}, \cdots, e_{i_M}\}$ be a set of M error digits, where $0 \leq i_1 < i_2 < \cdots < i_M < n$. The integer M is called the *size* of E.

DEFINITION 8.2 A set of J parity-check sums $A_1, A_2, \cdots, A_J$ is said to be orthogonal on the set E if and only if (1) every error digit e_{i_l} in E is checked by every check-sum A_j for $1 \leq j \leq J$, and (2) no other error digit is checked by more than one check-sum.

For example, the following four parity-check sums are orthogonal on the set $E = \{e_6, e_8\}$:

$$\begin{aligned}
A_1 &= e_0 + e_2 + e_6 + e_8, \\
A_2 &= e_3 + e_4 + e_6 + e_8, \\
A_3 &= e_1 + e_6 + e_7 + e_8, \\
A_4 &= e_5 + e_6 + e_8.
\end{aligned}$$

Following the same argument employed for one-step majority-logic decoding, we can correctly determine the sum of error digits in E, $e_{i_1} + e_{i_2} + \cdots + e_{i_M}$ from the check-sums $A_1, A_2, \cdots, A_J$ orthogonal on E provided that there are $\lfloor J/2 \rfloor$ or fewer errors in the error pattern $\mathbf{e}$. This sum of error digits in E may be regarded as an *additional* check-sum and so can be used for decoding.

Consider an (n, k) cyclic code C that is used for error control in a communication (or storage) system. Let $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1})$ denote the error vector that occurs during the transmission of a codeword $\mathbf{v}$ in C. Let $E_1^1, E_2^1, \cdots, E_i^1, \cdots$ be some properly selected sets of error digits of $\mathbf{e}$. Let $S(E_i^1)$ denote the modulo-2 sum of the error digits in E_i^1. Suppose that for each set E_i^1 it is possible to form at least J parity-check sums orthogonal on it. Then, the sum $S(E_i^1)$ can be estimated from these J orthogonal check-sums. The estimation can be done by a J-input majority-logic gate with the J orthogonal check-sums as inputs. The estimated value of $S(E_i^1)$ is the output of a majority-logic gate, which is 1 if and only if more than half of the inputs are 1; otherwise, it is 0. The estimation is correct provided that there are $\lfloor J/2 \rfloor$ or fewer errors in the error vector $\mathbf{e}$. The sums $S(E_1^1), S(E_2^1), \cdots, S(E_i^1), \cdots$ (possibly together with other check-sums) are then used to estimate the sums of error digits in the second selected sets, $E_1^2, E_2^2, \cdots, E_i^2, \cdots$, whose size is smaller than that of the first selected sets. Suppose that for each set E_i^2 it is possible to form J or more check-sums orthogonal on it. Then, the sum $S(E_i^2)$ can be determined correctly from the check-sums orthogonal on E_i^2 provided that there are no more than $\lfloor J/2 \rfloor$ errors in $\mathbf{e}$. Once the sums, $S(E_1^2), S(E_2^2), \cdots, S(E_i^2), \cdots$, are determined, they (maybe together with other check-sums) are used to estimate the sums of error digits in the third selected sets, $E_1^3, E_2^3, \cdots, E_i^3, \cdots$, whose size is smaller than that of the second selected sets. The process of estimating check-sums from known check-sums is called *orthogonalization* [2]. The orthogonalization process continues until a set of J or more check-sums orthogonal on only a single error digit, say e_{n-1}, is obtained. Then, the value of e_{n-1} can be estimated from these orthogonal check-sums. Because of the cyclic structure of the code, other error digits can be estimated in the same manner and by the same circuitry. A code is said to be *L-step orthogonalizable* (or *L-step majority-logic decodable*) if L steps of orthogonalization are required to

make a decoding decision on an error digit. The decoding process is called *L-step majority-logic decoding*. A code is said to be *completely L-step orthogonalizable* if J is 1 less than the minimum distance of the code (i.e., $J = d_{\min} - 1$). Because majority-logic gates are used to estimate selected sums of error digits at each step of orthogonalization, a total of L levels of majority-logic gates are required for decoding. The number of gates required at each level depends on the structure of the code.

The following two examples are used to illustrate the notions of multiple-step majority-logic decoding.

EXAMPLE 8.11

Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$. This is a Hamming code. The parity-check matrix (in systematic form) is found as follows:

$$\mathbb{H} = \begin{bmatrix} \mathbb{h}_0 \\ \mathbb{h}_1 \\ \mathbb{h}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

We see that the vectors $\mathbb{h}_0$ and $\mathbb{h}_2$ are orthogonal on digit positions 5 and 6 (or X^5 and X^6). We also see that the vectors $\mathbb{h}_0 + \mathbb{h}_1$ and $\mathbb{h}_2$ are orthogonal on digit positions 4 and 6. Let $E_1^1 = \{e_5, e_6\}$ and $E_2^1 = \{e_4, e_6\}$ be two selected sets. Let $\mathbb{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector. Then, the parity-check sums formed from $\mathbb{h}_0$ and $\mathbb{h}_2$ are

$$\begin{aligned} A_1 &= \mathbb{r} \cdot \mathbb{h}_0 = e_0 + e_3 + e_5 + e_6 \\ A_2 &= \mathbb{r} \cdot \mathbb{h}_2 = e_2 + e_4 + e_5 + e_6 \end{aligned}$$

and the parity-check sums formed from $\mathbb{h}_0 + \mathbb{h}_1$ and $\mathbb{h}_2$ are

$$\begin{aligned} B_1 &= \mathbb{r} \cdot (\mathbb{h}_0 + \mathbb{h}_1) = e_0 + e_1 + e_4 + e_6 \\ B_2 &= \mathbb{r} \cdot \mathbb{h}_2 = e_2 + e_4 + e_5 + e_6. \end{aligned}$$

The parity-check sums A_1 and A_2 are orthogonal on the set $E_1^1 = \{e_5, e_6\}$, and the parity-check sums B_1 and B_2 are orthogonal on the set $E_2^1 = \{e_4, e_6\}$. Therefore, the sum $S(E_1^1) = e_5 + e_6$ can be estimated from A_1 and A_2, and the sum $S(E_2^1) = e_4 + e_6$ can be estimated from B_1 and B_2. The sums $S(E_1^1)$ and $S(E_2^1)$ will be correctly estimated provided that there is no more than one error in the error vector $\mathbb{e}$. Now, let $E_1^2 = \{e_6\}$. We see that $S(E_1^1)$ and $S(E_2^1)$ are orthogonal on e_6. Hence, e_6 can be estimated from $S(E_1^1)$ and $S(E_2^1)$. The value of e_6 will be estimated correctly provided that there is no more than one error in $\mathbb{e}$. Therefore, the (7, 4) Hamming code can be decoded with two steps of orthogonalization, and it is two-step majority-logic decodable. Because its minimum distance is 3 and $J = 2$, it is two-step completely orthogonalizable. A type-II decoder for this code is shown in Figure 8.6.

Let $\mathbb{s} = (s_0, s_1, s_2) = \mathbb{r} \cdot \mathbb{H}^T$ be the syndrome of the received vector $\mathbb{r}$. Then, we can form the parity-check sums A_1, A_2, B_1, and B_2 from the syndrome digits as follows:

$$\begin{aligned} A_1 &= s_0, & A_2 &= s_2, \\ B_1 &= s_0 + s_1, & B_2 &= s_2. \end{aligned}$$

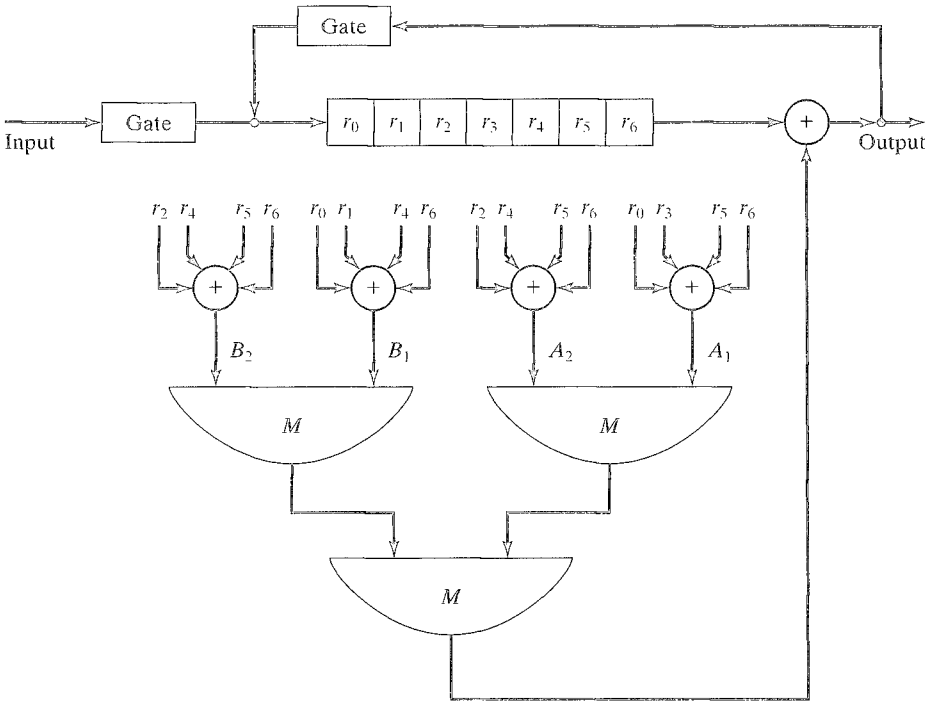

FIGURE 8.6: Type-II two-step majority-logic decoder for the (7, 4) Hamming code.

Based on these check-sums, we may construct a type-I majority-logic decoder for the (7, 4) Hamming code.

EXAMPLE 8.12

Consider the triple-error-correcting (15, 5) BCH code whose generator polynomial is

$$g(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}.$$

The parity-check matrix (in systematic form) is

$$\mathbb{H} = \begin{bmatrix} \mathbb{h}_0 \\ \mathbb{h}_1 \\ \mathbb{h}_2 \\ \mathbb{h}_3 \\ \mathbb{h}_4 \\ \mathbb{h}_5 \\ \mathbb{h}_6 \\ \mathbb{h}_7 \\ \mathbb{h}_8 \\ \mathbb{h}_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Let

$$E_1^1 = \{e_{13}, e_{14}\}, \quad E_2^1 = \{e_{12}, e_{14}\},$$
$$E_3^1 = \{e_{11}, e_{14}\}, \quad E_4^1 = \{e_{10}, e_{14}\},$$
$$E_5^1 = \{e_5, e_{14}\}, \quad E_6^1 = \{e_2, e_{14}\}$$

be six selected sets of error digits. For each of the preceding sets it is possible to find six parity-check sums orthogonal on it. Let $\mathbf{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14})$ be the received vector. By taking proper combinations of the rows $\mathbf{H}$, we find the following parity-check sums orthogonal on E_1^1, E_2^1, E_3^1, E_4^1, E_5^1, and E_6^1:

1. Check-sums orthogonal on $E_1^1 = \{e_{13}, e_{14}\}$:

$$\begin{aligned}
A_{11} &= \mathbf{r} \cdot \mathbf{h}_4 &&= e_4 + e_{10} + e_{13} + e_{14} \\
A_{12} &= \mathbf{r} \cdot \mathbf{h}_7 &&= e_7 + e_{12} + e_{13} + e_{14} \\
A_{13} &= \mathbf{r} \cdot \mathbf{h}_9 &&= e_9 + e_{11} + e_{13} + e_{14} \\
A_{14} &= \mathbf{r} \cdot (\mathbf{h}_0 + \mathbf{h}_8) &&= e_0 + e_8 + e_{13} + e_{14} \\
A_{15} &= \mathbf{r} \cdot (\mathbf{h}_1 + \mathbf{h}_5) &&= e_1 + e_5 + e_{13} + e_{14} \\
A_{16} &= \mathbf{r} \cdot (\mathbf{h}_3 + \mathbf{h}_6) &&= e_3 + e_6 + e_{13} + e_{14}.
\end{aligned}$$

2. Check-sums orthogonal on $E_2^1 = \{e_{12}, e_{14}\}$:

$$\begin{aligned}
A_{21} &= \mathbf{r} \cdot \mathbf{h}_0 &&= e_0 + e_{10} + e_{12} + e_{14} \\
A_{22} &= \mathbf{r} \cdot \mathbf{h}_3 &&= e_3 + e_{11} + e_{12} + e_{14} \\
A_{23} &= \mathbf{r} \cdot \mathbf{h}_7 &&= e_7 + e_{13} + e_{12} + e_{14} \\
A_{24} &= \mathbf{r} \cdot (\mathbf{h}_1 + \mathbf{h}_2) &&= e_1 + e_2 + e_{12} + e_{14} \\
A_{25} &= \mathbf{r} \cdot (\mathbf{h}_4 + \mathbf{h}_8) &&= e_4 + e_8 + e_{12} + e_{14} \\
A_{26} &= \mathbf{r} \cdot (\mathbf{h}_6 + \mathbf{h}_9) &&= e_6 + e_9 + e_{12} + e_{14}.
\end{aligned}$$

3. Check-sums orthogonal on $E_3^1 = \{e_{11}, e_{14}\}$:

$$\begin{aligned}
A_{31} &= \mathbf{r} \cdot \mathbf{h}_3 &&= e_3 + e_{12} + e_{11} + e_{14} \\
A_{32} &= \mathbf{r} \cdot \mathbf{h}_9 &&= e_9 + e_{13} + e_{11} + e_{14} \\
A_{33} &= \mathbf{r} \cdot (\mathbf{h}_0 + \mathbf{h}_5) &&= e_0 + e_5 + e_{11} + e_{14} \\
A_{34} &= \mathbf{r} \cdot (\mathbf{h}_1 + \mathbf{h}_8) &&= e_1 + e_8 + e_{11} + e_{14} \\
A_{35} &= \mathbf{r} \cdot (\mathbf{h}_2 + \mathbf{h}_4) &&= e_2 + e_4 + e_{11} + e_{14} \\
A_{36} &= \mathbf{r} \cdot (\mathbf{h}_6 + \mathbf{h}_7) &&= e_6 + e_7 + e_{11} + e_{14}.
\end{aligned}$$

4. Check-sums orthogonal on $E_4^1 = \{e_{10}, e_{14}\}$:

$$\begin{aligned}
A_{41} &= \mathbf{r} \cdot \mathbf{h}_0 &&= e_0 + e_{12} + e_{10} + e_{14} \\
A_{42} &= \mathbf{r} \cdot \mathbf{h}_4 &&= e_4 + e_{13} + e_{10} + e_{14} \\
A_{43} &= \mathbf{r} \cdot (\mathbf{h}_1 + \mathbf{h}_6) &&= e_1 + e_6 + e_{10} + e_{14} \\
A_{44} &= \mathbf{r} \cdot (\mathbf{h}_3 + \mathbf{h}_5) &&= e_3 + e_5 + e_{10} + e_{14} \\
A_{45} &= \mathbf{r} \cdot (\mathbf{h}_7 + \mathbf{h}_8) &&= e_7 + e_8 + e_{10} + e_{14} \\
A_{46} &= \mathbf{r} \cdot (\mathbf{h}_2 + \mathbf{h}_9) &&= e_2 + e_9 + e_{10} + e_{14}.
\end{aligned}$$

5. Check-sums orthogonal on $E_5^1 = \{e_5, e_{14}\}$:

$$\begin{aligned}
A_{51} &= \mathbb{r} \cdot (\mathbb{h}_0 + \mathbb{h}_5) &&= e_0 + e_{11} + e_5 + e_{14} \\
A_{52} &= \mathbb{r} \cdot (\mathbb{h}_1 + \mathbb{h}_5) &&= e_1 + e_{13} + e_5 + e_{14} \\
A_{53} &= \mathbb{r} \cdot (\mathbb{h}_3 + \mathbb{h}_5) &&= e_3 + e_{10} + e_5 + e_{14} \\
A_{54} &= \mathbb{r} \cdot (\mathbb{h}_4 + \mathbb{h}_5 + \mathbb{h}_6) &&= e_4 + e_6 + e_5 + e_{14} \\
A_{55} &= \mathbb{r} \cdot (\mathbb{h}_2 + \mathbb{h}_5 + \mathbb{h}_7) &&= e_2 + e_7 + e_5 + e_{14} \\
A_{56} &= \mathbb{r} \cdot (\mathbb{h}_5 + \mathbb{h}_8 + \mathbb{h}_9) &&= e_8 + e_9 + e_5 + e_{14}.
\end{aligned}$$

6. Check-sums orthogonal on $E_6^1 = \{e_2, e_{14}\}$:

$$\begin{aligned}
A_{61} &= \mathbb{r} \cdot (\mathbb{h}_1 + \mathbb{h}_2) &&= e_1 + e_{12} + e_2 + e_{14} \\
A_{62} &= \mathbb{r} \cdot (\mathbb{h}_2 + \mathbb{h}_4) &&= e_4 + e_{11} + e_2 + e_{14} \\
A_{63} &= \mathbb{r} \cdot (\mathbb{h}_0 + \mathbb{h}_2 + \mathbb{h}_6) &&= e_0 + e_6 + e_2 + e_{14} \\
A_{64} &= \mathbb{r} \cdot (\mathbb{h}_2 + \mathbb{h}_3 + \mathbb{h}_8) &&= e_3 + e_8 + e_2 + e_{14} \\
A_{65} &= \mathbb{r} \cdot (\mathbb{h}_2 + \mathbb{h}_5 + \mathbb{h}_7) &&= e_5 + e_7 + e_2 + e_{14} \\
A_{66} &= \mathbb{r} \cdot (\mathbb{h}_2 + \mathbb{h}_9) &&= e_9 + e_{10} + e_2 + e_{14}.
\end{aligned}$$

From the foregoing orthogonal check-sums, the sums $S(E_1^1) = e_{13} + e_{14}$, $S(E_2^1) = e_{12} + e_{14}$, $S(E_3^1) = e_{11} + e_{14}$, $S(E_4^1) = e_{10} + e_{14}$, $S(E_5^1) = e_5 + e_{14}$, and $S(E_6^1) = e_2 + e_{14}$ can be correctly estimated provided that there are no more than three errors in the error vector $\mathbb{e}$. Let $E_1^2 = \{e_{14}\}$. We see that the error sums $S(E_1^1)$, $S(E_2^1)$, $S(E_3^1)$, $S(E_4^1)$, $S(E_5^1)$, and $S(E_6^1)$ are orthogonal on e_{14}. Hence, e_{14} can be estimated from these sums. Therefore, the (15, 5) BCH code is two-step orthogonalizable. Because $J = 6$, it is capable of correcting three or fewer errors with two-step majority-logic decoding. It is known that the code has a minimum distance of exactly 7. Hence, it is two-step completely orthogonalizable.

The type-II decoder for the (15, 5) BCH code is shown in Figure 8.7, where seven six-input majority-logic gates (connected in a tree form) are used. Construction of a type-I majority-logic decoder for the (15, 5) BCH code is left as an exercise (see Problem 8.12).

A general type-II L-step majority-logic decoder is shown in Figure 8.8. The error correction procedure is as follows:

Step 1. The received vector $\mathbb{r}(X)$ is read into the buffer register.

Step 2. Parity-check sums (no more than $(J)^L$ of them) orthogonal on certain properly selected sets of error digits are formed by summing appropriate sets of received digits. These check-sums are then fed into the first-level majority-logic gates (there are at most $(J)^{L-1}$ of them). The outputs of the first-level majority-logic gates are used to form inputs to the second-level majority-logic gates (there are at most $(J)^{L-2}$ of them). The outputs of the second-level majority-logic gates are then used to form inputs to third-level majority-logic gates (there are at most $(J)^{L-3}$ of them). This process continues until the last level is reached; there is only one gate at the last level. The J inputs to this

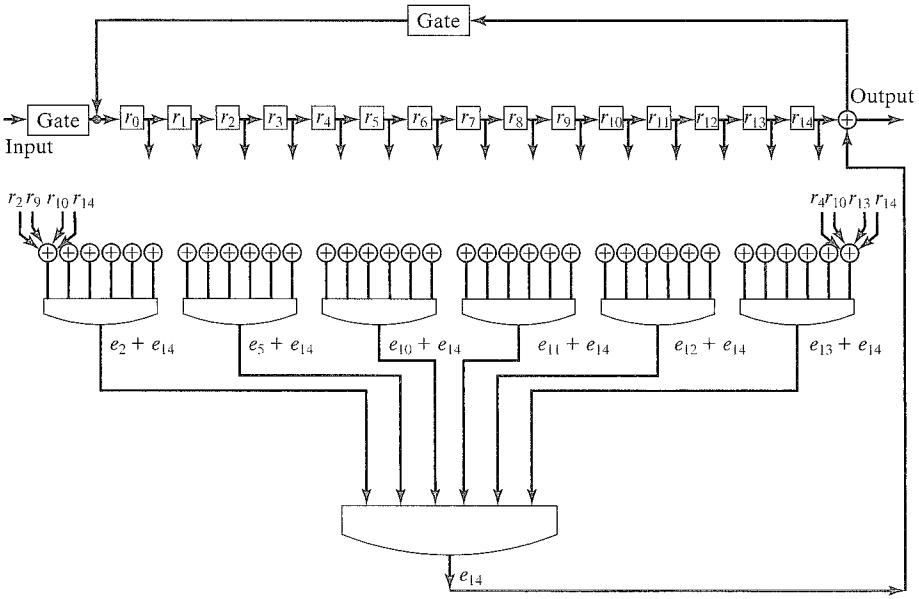

FIGURE 8.7: Type II two-step majority-logic decoder for the (15, 5) BCH code.

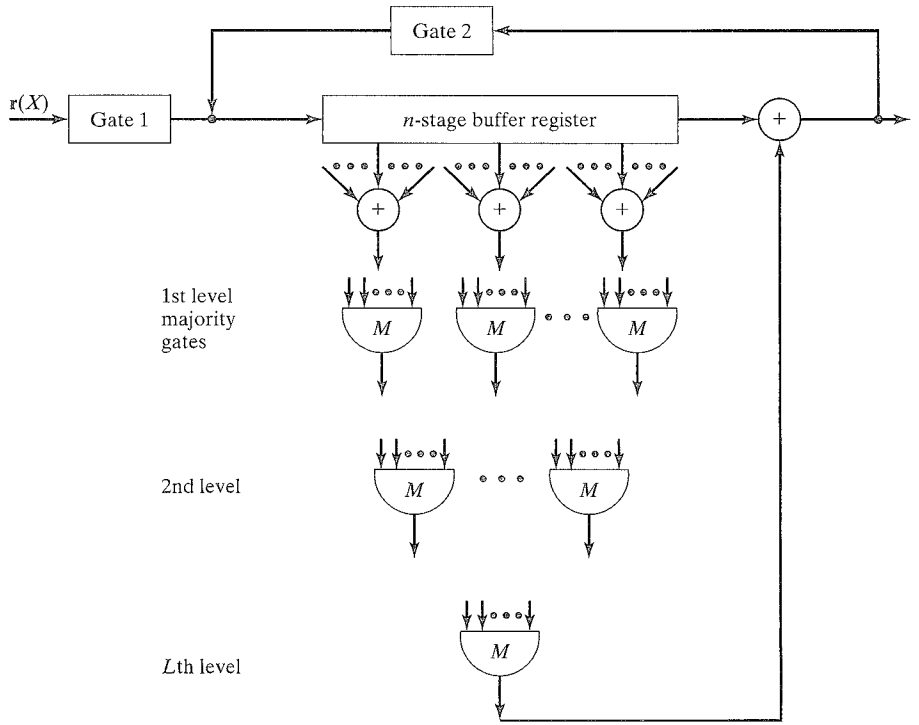

FIGURE 8.8: General type-II L-step majority-logic decoder.

gate are check-sums orthogonal on the highest-order error digit e_{n-1}. The output of this gate is used to correct the received digit r_{n-1}.

Step 3. The received digit r_{n-1} is read out of the buffer and is corrected by the last-level majority-logic gate.

Step 4. At the end of step 3, the buffer register has been shifted one place to the right. Now, the second-highest-order received digit r_{n-2} is in the rightmost stage of the buffer register, and it will be corrected in exactly the same manner as was the highest-order received digit r_{n-1}. The decoder repeats steps 2 and 3.

Step 5. The received vector is decoded digit by digit in the manner described until a total of n shifts.

A general type-I decoder for an L-step majority-logic decoder code is shown in Figure 8.9. Its decoding operation is identical to that of the type-I decoder for

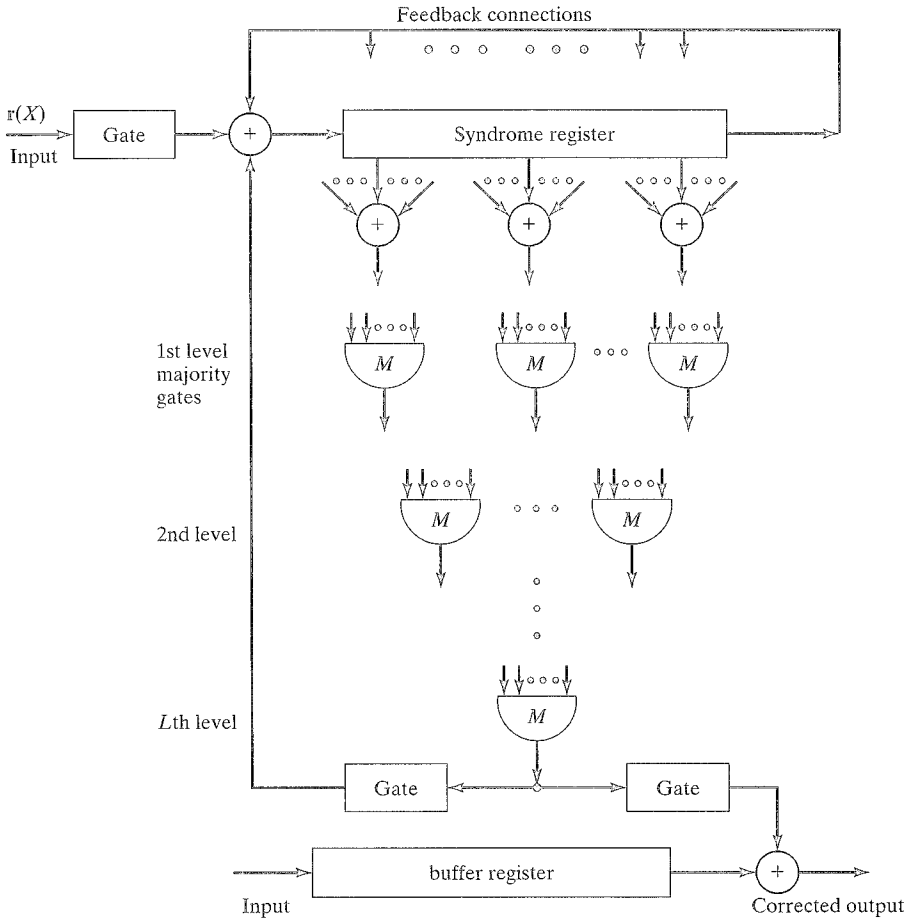

FIGURE 8.9: General type-I L-step majority-logic decoder.

a one-step majority-logic decodable code except that L levels of orthogonalization are required.

An L-step majority-logic decoder requires L levels of majority-logic gates. At the ith level, no more than $(J)^{L-i}$ gates are required. Thus, the total number of majority-logic gates needed is upper bounded by $1 + J + J^2 + \cdots + J^{L-1}$. In fact, Massey [2] has proved that for an (n, k) L-step majority-logic decodable code no more than k majority-logic gates are ever required. Unfortunately, for a given L-step majority-logic decodable cyclic code, there is no known systematic method for minimizing the number of majority-logic gates except the trial-and-error method. For almost all the known classes of L-step majority-logic decodable codes, the rules for forming orthogonal parity-check sums require a total of $1 + J + J^2 + \cdots + J^{L-1}$ majority-logic gates. Thus, the complexity is an exponential function of L. For large L, the decoder is likely to be impractical. Fortunately, there are many cyclic codes with useful parameters that can be decoded with a reasonably small L.

Several large classes of cyclic codes have been found to be L-step majority-logic decodable. The construction and the rules for orthogonalization of these codes are based on the properties of finite geometries, which are the subject of the next four sections.

8.5 EUCLIDEAN GEOMETRY

Consider all the m-tuples $(a_0, a_1, \cdots, a_{m-1})$, with components a_i's from the Galois field $GF(2^s)$. There are $(2^s)^m = 2^{ms}$ such m-tuples. These 2^{ms} m-tuples form a vector space over $GF(2^s)$. The vector addition and scalar multiplication are defined in the usual way:

$$(a_0, a_1, \cdots, a_{m-1}) + (b_0, b_1, \cdots, b_{m-1}) = (a_0 + b_0, a_1 + b_1, \cdots, a_{m-1} + b_{m-1}),$$

$$\beta \cdot (a_0, a_1, \cdots, a_{m-1}) = (\beta \cdot a_0, \beta \cdot a_1, \cdots, \beta \cdot a_{m-1}),$$

where additions $a_i + b_i$ and multiplication $\beta \cdot a_i$ are carried out in $GF(2^s)$. In combinatorial mathematics, the 2^{ms} m-tuples over $GF(2^s)$ are also known to form an m-dimensional *Euclidean geometry* over $GF(2^s)$, denoted by $\mathrm{EG}(m, 2^s)$ [14–16]. Each m-tuple $\mathbf{a} = (a_0, a_1, \cdots, a_{m-1})$ is called a *point* in $\mathrm{EG}(m, 2^s)$. The all-zero m-tuple, $\mathbf{0} = (0, 0, \cdots, 0)$, is called the *origin* of the geometry $\mathrm{EG}(m, 2^s)$.

Let $\mathbf{a}$ be a nonorigin point in $\mathrm{EG}(m, 2^s)$ (i.e., $\mathbf{a} \neq \mathbf{0}$). Then, the 2^s points $\{\beta\mathbf{a} : \beta \in GF(2^s)\}$ constitute a *line* (or 1-*flat*) in $\mathrm{EG}(m, 2^s)$. For convenience, we use the notation $\{\beta\mathbf{a}\}$ to represent this line. Because this line contains the origin (with $\beta = 0$), we say that $\{\beta\mathbf{a}\}$ passes through the origin. Let $\mathbf{a}_0$ and $\mathbf{a}$ be two linearly independent points in $\mathrm{EG}(m, 2^s)$ (i.e., $\beta_0\mathbf{a}_0 + \beta\mathbf{a} \neq \mathbf{0}$ unless $\beta_0 = \beta = 0$). Then, the collection of the following 2^s points,

$$\{\mathbf{a}_0 + \beta\mathbf{a}\},$$

with $\beta \in GF(2^s)$, constitutes a line in $\mathrm{EG}(m, 2^s)$ that passes through the point $\mathbf{a}_0$. Line $\{\beta\mathbf{a}\}$ and the line $\{\mathbf{a}_0 + \beta\mathbf{a}\}$ do not have any point in common. Suppose that they have a common point. Then, for some β' and β'' in $GF(2^s)$,

$$\beta'\mathbf{a} = \mathbf{a}_0 + \beta''\mathbf{a}.$$

As a result, $\mathbb{a}_0 + (\beta'' - \beta')\mathbb{a} = 0$. This implies that $\mathbb{a}_0$ and $\mathbb{a}$ are linearly dependent, which is a contradiction to our assumption that $\mathbb{a}_0$ and $\mathbb{a}$ are two linearly independent points in EG$(m, 2^s)$. Therefore, $\{\beta\mathbb{a}\}$ and$\{\mathbb{a}_0 + \beta\mathbb{a}\}$ do not have any common points. We say that $\{\beta\mathbb{a}\}$ and $\{\mathbb{a}_0 + \beta\mathbb{a}\}$ are *parallel* lines. Note that $\{\beta\mathbb{a}\}$ is simply a one-dimensional subspace of the vector space of all the 2^{ms} m-tuples over $GF(2^s)$, and $\{\mathbb{a}_0 + \beta\mathbb{a}\}$ is simply a coset of $\{\beta\mathbb{a}\}$. Let $\mathbb{b}_0$ be a point not on line $\{\beta\mathbb{a}\}$ or on line $\{\mathbb{a}_0 + \beta\mathbb{a}\}$. The line $\{\mathbb{b}_0 + \beta\mathbb{a}\}$ passes through the point $\mathbb{b}_0$ and is parallel to both $\{\beta\mathbb{a}\}$ and $\{\mathbb{a}_0 + \beta\mathbb{a}\}$. In EG$(m, 2^s)$, for every line passing through the origin, there are $2^{(m-1)s} - 1$ lines parallel to it. A line $\{\beta\mathbb{a}\}$ and the $2^{(m-1)s} - 1$ lines parallel to it are said to form a *parallel bundle*. The $2^{(m-1)s}$ lines in a parallel bundle are parallel to each other. Basically, the $2^{(m-1)s}$ lines in a parallel bundle simply correspond to a one-dimensional subspace of the vector space of all the m-tuples over $GF(2)$ and its $2^{(m-1)s} - 1$ cosets.

Let $\mathbb{a}_1$ and $\mathbb{a}_2$ be two linearly independent points in EG$(m, 2^s)$. The lines $\{\mathbb{a}_0 + \beta\mathbb{a}_1\}$ and $\{\mathbb{a}_0 + \beta\mathbb{a}_2\}$ have only one point, $\mathbb{a}_0$, in common. Suppose that they have another point besides $\mathbb{a}_0$ in common. Then, for some $\beta' \neq 0$ and $\beta'' \neq 0$, we have

$$\mathbb{a}_0 + \beta'\mathbb{a}_1 = \mathbb{a}_0 + \beta''\mathbb{a}_2.$$

This equality implies that $\beta'\mathbb{a}_1 - \beta''\mathbb{a}_2 = 0$ and that $\mathbb{a}_1$ and $\mathbb{a}_2$ are linearly dependent. This is a contradiction to the hypothesis that $\mathbb{a}_1$ and $\mathbb{a}_2$ are linearly independent points in EG$(m, 2^s)$. Therefore, $\{\mathbb{a}_0 + \beta\mathbb{a}_1\}$ and $\{\mathbb{a}_0 + \beta\mathbb{a}_2\}$ have only one point in common, and they both pass through the point $\mathbb{a}_0$. We say that $\{\mathbb{a}_0 + \beta\mathbb{a}_1\}$ and $\{\mathbb{a}_0 + \beta\mathbb{a}_2\}$ intersect at the point $\mathbb{a}_0$. Given a point $\mathbb{a}_0$ in EG$(m, 2^s)$, there are

$$\frac{2^{ms} - 1}{2^s - 1} \tag{8.26}$$

lines in EG$(m, 2^s)$ that intersect at $\mathbb{a}_0$ (including the line $\{\beta\mathbb{a}_0\}$ that passes through the origin). This is an important property that will be used to form orthogonal parity-check sums for the codes presented in the next section. Another important structural property of lines is that any two points are connected by a line. Let $\mathbb{a}_1$ and $\mathbb{a}_2$ be two points in EG$(m, 2^s)$. Suppose that $\mathbb{a}_1$ and $\mathbb{a}_2$ are linearly dependent. Then, $\mathbb{a}_2 = \beta_i\mathbb{a}_1$ for some element β_i in $GF(2^s)$. In this case $\mathbb{a}_1$ and $\mathbb{a}_2$ are connected by the line $\{\beta\mathbb{a}_1\}$. Suppose that $\mathbb{a}_1$ and $\mathbb{a}_2$ are linearly independent. Let $\mathbb{a}_3 = \mathbb{a}_1 + \mathbb{a}_2$. Then, $\mathbb{a}_2 = \mathbb{a}_1 + \mathbb{a}_3$, and $\mathbb{a}_1$ and $\mathbb{a}_2$ are connected by the line $\{\mathbb{a}_1 + \beta\mathbb{a}_3\}$. The total number of lines in EG$(m, 2^s)$ is

$$\frac{2^{(m-1)s}(2^{ms} - 1)}{2^s - 1}.$$

EXAMPLE 8.13

Let $m = 3$ and $s = 1$. Consider the Euclidean geometry EG(3, 2) over $GF(2)$. There are eight points and 28 lines. Each point $\mathbb{a}_i$ is a 3-tuple over $GF(2)$. Each line consists of two points $\{\mathbb{a}_i, \mathbb{a}_j\}$. The points and the lines are given in Table 8.3. Lines $\{\mathbb{a}_0, \mathbb{a}_1\}$, $\{\mathbb{a}_2, \mathbb{a}_3\}$, $\{\mathbb{a}_4, \mathbb{a}_5\}$, and $\{\mathbb{a}_6, \mathbb{a}_7\}$ are parallel and they form a parallel bundle. The lines that intersect at the point $\mathbb{a}_2$ are $\{\mathbb{a}_0, \mathbb{a}_2\}$, $\{\mathbb{a}_1, \mathbb{a}_2\}$, $\{\mathbb{a}_2, \mathbb{a}_3\}$, $\{\mathbb{a}_2, \mathbb{a}_4\}$, $\{\mathbb{a}_2, \mathbb{a}_5\}$, $\{\mathbb{a}_2, \mathbb{a}_6\}$, and $\{\mathbb{a}_2, \mathbb{a}_7\}$.

TABLE 8.3: Points and lines in EG(3, 2).

(a) Points in EG(3, 2)

$\mathbf{a}_0 = (000)$,	$\mathbf{a}_1 = (001)$,	$\mathbf{a}_2 = (010)$,	$\mathbf{a}_3 = (011)$,
$\mathbf{a}_4 = (100)$,	$\mathbf{a}_5 = (101)$,	$\mathbf{a}_6 = (110)$,	$\mathbf{a}_7 = (111)$.

(b) Lines in EG(3, 2)

$\{\mathbf{a}_0, \mathbf{a}_1\}$	$\{\mathbf{a}_1, \mathbf{a}_2\}$	$\{\mathbf{a}_2, \mathbf{a}_4\}$	$\{\mathbf{a}_3, \mathbf{a}_7\}$
$\{\mathbf{a}_0, \mathbf{a}_2\}$	$\{\mathbf{a}_1, \mathbf{a}_3\}$	$\{\mathbf{a}_2, \mathbf{a}_5\}$	$\{\mathbf{a}_4, \mathbf{a}_5\}$
$\{\mathbf{a}_0, \mathbf{a}_3\}$	$\{\mathbf{a}_1, \mathbf{a}_4\}$	$\{\mathbf{a}_2, \mathbf{a}_6\}$	$\{\mathbf{a}_4, \mathbf{a}_6\}$
$\{\mathbf{a}_0, \mathbf{a}_4\}$	$\{\mathbf{a}_1, \mathbf{a}_5\}$	$\{\mathbf{a}_2, \mathbf{a}_7\}$	$\{\mathbf{a}_4, \mathbf{a}_7\}$
$\{\mathbf{a}_0, \mathbf{a}_5\}$	$\{\mathbf{a}_1, \mathbf{a}_6\}$	$\{\mathbf{a}_3, \mathbf{a}_4\}$	$\{\mathbf{a}_5, \mathbf{a}_6\}$
$\{\mathbf{a}_0, \mathbf{a}_6\}$	$\{\mathbf{a}_1, \mathbf{a}_7\}$	$\{\mathbf{a}_3, \mathbf{a}_5\}$	$\{\mathbf{a}_5, \mathbf{a}_7\}$
$\{\mathbf{a}_0, \mathbf{a}_7\}$	$\{\mathbf{a}_2, \mathbf{a}_3\}$	$\{\mathbf{a}_3, \mathbf{a}_6\}$	$\{\mathbf{a}_6, \mathbf{a}_7\}$

Now, we extend the concept of lines to planes in EG$(m, 2^s)$. Let $\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_\mu$ be $\mu + 1$ linearly independent points in EG$(m, 2^s)$, where $\mu < m$. The $2^{\mu s}$ points of the form

$$\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \beta_2 \mathbf{a}_2 + \cdots + \beta_\mu \mathbf{a}_\mu,$$

with $\beta_i \in GF(2^s)$ for $1 \leq i \leq \mu$, constitute a μ-*flat* (or a μ-dimensional *hyperplane*) in EG$(m, 2^s)$ that passes through the point $\mathbf{a}_0$. We denote this μ-flat by $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu\}$. The μ-flat that consists of the $2^{\mu s}$ points

$$\beta_1 \mathbf{a}_1 + \beta_2 \mathbf{a}_2 + \cdots + \beta_\mu \mathbf{a}_\mu$$

passes through the origin. We can readily prove that the μ-flats $\{\beta_1 \mathbf{a}_1 + \beta_2 \mathbf{a}_2 + \beta_3 \mathbf{a}_3 + \cdots + \beta_\mu \mathbf{a}_\mu\}$ and $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \beta_2 \mathbf{a}_2 + \cdots + \beta_\mu \mathbf{a}_\mu\}$ do not have any point in common. We say that these two μ-flats are parallel. For any μ-flat passing through the origin, there are $2^{(m-\mu)s} - 1$ μ-flats in EG$(m, 2^s)$ parallel to it. These $2^{(m-\mu)s}$ parallel μ-flats form a parallel bundle. Note that a μ-flat in EG$(m, 2^s)$ is either a μ-dimensional subspace of the vector space of all the 2^{ms} m-tuples over $GF(2^s)$ or a coset of a μ-dimensional subspace. The $2^{(m-\mu)s}$ parallel μ-flats in a parallel bundle simply correspond to a μ-dimensional subspace of the vector space of all the m-tuples over $GF(2)$ and its $2^{(m-1)s} - 1$ cosets.

If $\mathbf{a}_{\mu+1}$ is not a point in the μ-flat $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu\}$, then the $(\mu + 1)$-flat $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu + \beta_{\mu+1} \mathbf{a}_{\mu+1}\}$ contains the μ-flat $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu\}$. Let $\mathbf{b}_{\mu+1}$ be a point not in $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_{\mu+1} \mathbf{a}_{\mu+1}\}$. Then, the two $(\mu + 1)$-flats $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu + \beta_{\mu+1} \mathbf{a}_{\mu+1}\}$ and $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu + \beta_{\mu+1} \mathbf{b}_{\mu+1}\}$ intersect on the μ-flat $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu\}$ (i.e., they have the points in $\{\mathbf{a}_0 + \beta_1 \mathbf{a}_1 + \cdots + \beta_\mu \mathbf{a}_\mu\}$ as all their common points). Given a μ-flat F in EG$(m, 2^s)$, the number of $(\mu+1)$-flats in EG$(m, 2^s)$ that intersect on F is

$$\frac{2^{(m-\mu)s} - 1}{2^s - 1}. \tag{8.27}$$

TABLE 8.4: 2-Flats in EG(3, 2).

$\{a_0, a_1, a_2, a_3\}$	$\{a_4, a_5, a_6, a_7\}$	$\{a_0, a_1, a_4, a_5\}$	$\{a_2, a_3, a_6, a_7\}$
$\{a_0, a_2, a_4, a_6\}$	$\{a_1, a_3, a_5, a_7\}$	$\{a_0, a_1, a_6, a_7\}$	$\{a_2, a_3, a_4, a_5\}$
$\{a_0, a_2, a_5, a_7\}$	$\{a_1, a_3, a_4, a_6\}$	$\{a_0, a_4, a_3, a_7\}$	$\{a_1, a_2, a_5, a_6\}$
$\{a_0, a_3, a_5, a_6\}$	$\{a_1, a_2, a_4, a_7\}$		

Any point outside the μ-flat F is contained in one and only one of the $(\mu+1)$-flats that intersect on F. The number of μ-flats in $\mathrm{EG}(m, 2^s)$ is

$$2^{(m-\mu)s} \prod_{i=1}^{\mu} \frac{2^{(m-i+1)s} - 1}{2^{(\mu-i+1)s} - 1}.$$

EXAMPLE 8.14

Consider the geometry EG(3, 2) over $GF(2)$ given in Example 8.13. There are fourteen 2-flats, which are given in Table 8.4. The 2-flats that intersect on the line $\{a_1, a_3\}$ are $\{a_0, a_1, a_2, a_3\}$, $\{a_1, a_3, a_5, a_7\}$, and $\{a_1, a_3, a_4, a_6\}$. The 2-flats $\{a_0, a_1, a_2, a_3\}$ and $\{a_4, a_5, a_6, a_7\}$ are parallel.

Next, we show that the elements in the Galois field $GF(2^{ms})$ actually form an m-dimensional Euclidean geometry $\mathrm{EG}(m, 2^s)$. Let α be a primitive element of $GF(2^{ms})$. Then, the 2^{ms} elements in $GF(2^{ms})$ can be expressed as powers of α as follows: $\alpha^\infty = 0, \alpha^0 = 1, \alpha^1, \alpha^2, \cdots, \alpha^{2^{ms}-2}$. It is known that $GF(2^{ms})$ contains $GF(2^s)$ as a subfield. Every element α^i in $GF(2^{ms})$ can be expressed as

$$\alpha^i = a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + \cdots + a_{i,m-1}\alpha^{m-1},$$

where $a_{ij} \in GF(2^s)$ for $0 \leq j < m$. There is a *one-to-one correspondence* between the element α^i and the m-tuple $(a_{i0}, a_{i1}, \cdots, a_{i,m-1})$ over $GF(2^s)$. Therefore, the 2^{ms} elements in $GF(2^{ms})$ may be regarded as the 2^{ms} points in $\mathrm{EG}(m, 2^s)$, and $GF(2^{ms})$ as the geometry $\mathrm{EG}(m, 2^s)$. In this case, a μ-flat passing through the point α^{l_0} consists of the following $2^{\mu s}$ points:

$$\alpha^{l_0} + \beta_1\alpha^{l_1} + \cdots + \beta_\mu\alpha^{l_\mu},$$

where $\alpha^{l_0}, \alpha^{l_1}, \cdots, \alpha^{l_\mu}$ are $\mu + 1$ linearly independent elements in $GF(2^{ms})$, and $\beta_i \in GF(2^s)$.

EXAMPLE 8.15

Consider the Galois field $GF(2^4)$ given by Table 2.8. Let $m = 2$. Let α be a primitive element whose minimal polynomial is $\phi(X) = 1 + X + X^4$. Let $\beta = \alpha^5$. We see that $\beta^0 = 1, \beta^1 = \alpha^5, \beta^2 = \alpha^{10}$, and $\beta^3 = \alpha^{15} = 1$. Therefore, the order of β is 3. We can readily check that the elements

$$0,\ 1,\ \beta,\ \beta^2$$

TABLE 8.5: Elements in $GF(2^4)^*$.

	2-tuples over $GF(2^2)$
$0 = 0$	$(0, 0)$
$1 = 1$	$(1, 0)$
$\alpha = \alpha$	$(0, 1)$
$\alpha^2 = \beta + \alpha$	$(\beta, 1)$
$\alpha^3 = \beta + \beta^2\alpha$	(β, β^2)
$\alpha^4 = 1 + \alpha$	$(1, 1)$
$\alpha^5 = \beta$	$(\beta, 0)$
$\alpha^6 = \beta\alpha$	$(0, \beta)$
$\alpha^7 = \beta^2 + \beta\alpha$	(β^2, β)
$\alpha^8 = \beta^2 + \alpha$	$(\beta^2, 1)$
$\alpha^9 = \beta + \beta\alpha$	(β, β)
$\alpha^{10} = \beta^2$	$(\beta^2, 0)$
$\alpha^{11} = \beta^2\alpha$	$(0, \beta^2)$
$\alpha^{12} = 1 + \beta^2\alpha$	$(1, \beta^2)$
$\alpha^{13} = 1 + \beta\alpha$	$(1, \beta)$
$\alpha^{14} = \beta^2 + \beta^2\alpha$	(β^2, β^2)

*Elements in $GF(2^4)$ are expressed in the form $a_{i0} + a_{i1}\alpha$, where α is a primitive element in $GF(2^4)$, and a_{ij} is an element in $GF(2^2) = \{0, 1, \beta, \beta^2\}$ with $\beta = \alpha^5$.

form a field of four elements, $GF(2^2)$. Therefore, $GF(2^2)$ is a subfield of $GF(2^4)$. Table 8.5 shows that every element α^i in $GF(2^4)$ is expressed in the form

$$\alpha^i = a_{i0} + a_{i1}\alpha,$$

with a_{i0} and a_{i1} in $GF(2^2) = \{0, 1, \beta, \beta^2\}$. We may regard $GF(2^4)$ as the Euclidean geometry $\mathrm{EG}(2, 2^2)$ over $GF(2^2)$. Then, the points

$$\alpha^{14} + 0 \cdot \alpha = \alpha^{14}, \qquad \alpha^{14} + 1 \cdot \alpha = \alpha^7,$$
$$\alpha^{14} + \beta \cdot \alpha = \alpha^8, \qquad \alpha^{14} + \beta^2 \cdot \alpha = \alpha^{10},$$

form a line passing through the point α^{14}. The other four lines in $\mathrm{EG}(2, 2^2)$ passing through α^{14} are

$$\{\alpha^{14}, \alpha^{13}, \alpha, \alpha^5\}, \qquad \{\alpha^{14}, \alpha^0, \alpha^6, \alpha^2\},$$
$$\{\alpha^{14}, \alpha^9, \alpha^4, 0\}, \qquad \{\alpha^{14}, \alpha^{12}, \alpha^{11}, \alpha^3\}.$$

The field $GF(2^{ms})$ may be regarded either as an extension field of $GF(2^s)$ or as an extension field of $GF(2^m)$. Therefore, $GF(2^{ms})$ may be regarded either as the m-dimensional Euclidean geometry $\mathrm{EG}(m, 2^s)$ over $GF(2^s)$ or as the s-dimensional Euclidean geometry $\mathrm{EG}(s, 2^m)$ over $GF(2^m)$.

8.6 EUCLIDEAN GEOMETRY CODES

Let

$$\mathbf{v} = (v_0, v_1, \cdots, v_{2^{ms}-2}),$$

be a $(2^{ms} - 1)$-tuple over the binary field $GF(2)$. Let α be a primitive element of the Galois field $GF(2^{ms})$. We may number the components of $\mathbf{v}$ with the nonzero elements of $GF(2^{ms})$ as follows: the component v_i is numbered α^i for $0 \le i \le 2^m - 2$. Hence, α^i is the location number of v_i. Now, we regard $GF(2^{ms})$ as the m-dimensional Euclidean geometry over $GF(2^s)$, $\mathrm{EG}(m, 2^s)$. Let F be a μ-flat in $\mathrm{EG}(m, 2^s)$ that does not pass through the origin, $\alpha^\infty = 0$. Based on this μ-flat F, we may form a vector over $GF(2)$ as follows:

$$\mathbf{v}_F = (v_0, v_1, \cdots, v_{2^{ms}-2}),$$

whose ith component v_i is 1 if its location number α^i is a point in F; otherwise, v_i is 0. In other words, the location numbers for the nonzero components of $\mathbf{v}_F$ form the points of the μ-flat F. The vector $\mathbf{v}_F$ is called the *incidence vector* of the μ-flat F. The incidence vector $\mathbf{v}_F$ for the μ-flat F simply displays the points contained in F. A very interesting structural property of the incidence vectors of the μ-flats in $\mathrm{EG}(m, 2^s)$ not passing through the origin is their cyclic structure: a cyclic shift of the incidence vector of a μ-flat not passing through the origin is the incidence vector of another μ-flat not passing through the origin (see Problem 8.33).

EXAMPLE 8.16

Let $m = 2$ and $s = 2$. Consider the field $GF(2^4)$, which is regarded as the Euclidean geometry over $GF(2^2)$, $\mathrm{EG}(2, 2^2)$. From Example 8.15, the four 1-flats (or lines) passing through the point α^{14} but not the origin are

$$L_1 = \{\alpha^{14}, \alpha^7, \alpha^8, \alpha^{10}\}, \quad L_2 = \{\alpha^{14}, \alpha^{13}, \alpha, \alpha^5\},$$
$$L_3 = \{\alpha^{14}, \alpha^0, \alpha^6, \alpha^2\}, \quad L_4 = \{\alpha^{14}, \alpha^{12}, \alpha^{11}, \alpha^3\}.$$

The incidence vectors for these four 1-flats are

	Location Numbers														
	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbf{v}_{L_1} =$	(0	0	0	0	0	0	0	1	1	0	1	0	0	0	1)
$\mathbf{v}_{L_2} =$	(0	1	0	0	0	1	0	0	0	0	0	0	0	1	1)
$\mathbf{v}_{L_3} =$	(1	0	1	0	0	0	1	0	0	0	0	0	0	0	1)
$\mathbf{v}_{L_4} =$	(0	0	0	1	0	0	0	0	0	0	0	1	1	0	1)

Suppose we cyclically shift the incidence vector $\mathbf{v}_{L_2}$ of line L_2. We obtain the following vector:

$$(1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1),$$

which is the incidence vector of line $\{\alpha^0, \alpha^2, \alpha^6, \alpha^{14}\}$. If we cyclically shift the incidence vector $\mathbf{v}_{L_1}$ of line L_1, we obtain the following vector:

$$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0),$$

which is the incidence vector of line $\{\alpha^0, \alpha^8, \alpha^9, \alpha^{11}\}$.

Definition 8.3 A (μ, s)th-order binary Euclidean geometry (EG) code of length $2^{ms} - 1$ is the largest cyclic code whose null space contains the incidence vectors of all the $(\mu + 1)$-flats of EG$(m, 2^s)$ that do not pass through the origin.

Basically, the (μ, s)th-order EG code of length $2^{ms} - 1$ is the dual code (or null space) of the code (or space) spanned by the incidence vectors of the $(\mu + 1)$-flats of EG$(m, 2^s)$ that do not pass through the origin. Due to the cyclic structural property of the incidence vectors of the flats in EG$(m, 2^s)$ not passing through the origin, the code spanned by the incidence vectors of $(\mu + 1)$-flats not passing through the origin is cyclic and hence its dual code, the (μ, s)th-order EG code, is also cyclic.

The generator polynomial of a (μ, s)th-order EG code is given in terms of its roots in $GF(2^{ms})$. Let h be a nonnegative integer less than 2^{ms}. Then, we can express h in radix-2^s form as follows:

$$h = \delta_0 + \delta_1 2^s + \delta_2 2^{2s} + \cdots + \delta_{m-1} 2^{(m-1)s},$$

where $0 \leq \delta_i < 2^s$ for $0 \leq i < m$. The 2^s-weight of h, denoted by $W_{2^s}(h)$, is defined as the real sum of the coefficients in the radix-2^s expansion of h; that is,

$$W_{2^s}(h) = \sum_{i=0}^{m-1} \delta_i. \tag{8.28}$$

As an example, let $m = 3$ and $s = 2$. Then, we can expand the integer $h = 45$ in radix-2^2 form as follows:

$$45 = 1 + 3 \cdot 2^2 + 2 \cdot 2^{2\times 2},$$

with $\delta_0 = 1$, $\delta_1 = 3$, and $\delta_2 = 2$. The 2^2-weight of 45 is then

$$W_{2^2}(45) = 1 + 3 + 2 = 6.$$

Consider the difference $h - W_{2^s}(h)$, which we can express as follows:

$$h - W_{2^s}(h) = \delta_1(2^s - 1) + \delta_2(2^{2s} - 1) + \cdots + \delta_{m-1}(2^{(m-1)s} - 1).$$

It is clear from this difference that h is divisible by $2^s - 1$ if and only if its 2^s-weight, $W_{2^s}(h)$, is divisible by $2^s - 1$. Let $h^{(l)}$ be the remainder resulting from dividing $2^l h$ by $2^{ms} - 1$; that is,

$$2^l h = q(2^{ms} - 1) + h^{(l)},$$

with $0 \leq h^{(l)} < 2^{ms} - 1$. Clearly, $h^{(l)}$ is divisible by $2^s - 1$ if and only if h is divisible by $2^s - 1$. Note that $h^{(0)} = h$.

Now, we state a theorem (without proof) that characterizes the roots of the generator polynomial of a (μ, s)th-order EG code. The proof of this theorem can be found in [26, 27], and [33].

Theorem 8.3 Let α be a primitive element of the Galois field $GF(2^{ms})$. Let h be a nonnegative integer less than $2^{ms} - 1$. The generator polynomial $\mathbf{g}(X)$ of the (μ, s)th-order EG code of length $2^{ms} - 1$ has α^h as a root if and only if

$$0 < \max_{0 \leq l < s} W_{2^s}(h^{(l)}) \leq (m - \mu - 1)(2^s - 1). \tag{8.29}$$

EXAMPLE 8.17

Let $m = 2, s = 2$, and $\mu = 0$. Then, the Galois field $GF(2^4)$ may be regarded as the Euclidean geometry $\mathrm{EG}(2, 2^2)$ over $GF(2^2)$. Let α be a primitive element in $GF(2^4)$ (use Table 2.8). Let h be a nonnegative integer less than 15. It follows from Theorem 8.3 that the generator polynomial $g(X)$ of the (0, 2)th-order EG code of length 15 has α^h as a root if and only if

$$0 < \max_{0 \le l < 2} W_{2^2}(h^{(l)}) \le 3.$$

The nonnegative integers less than 15 that satisfy this condition are 1, 2, 3, 4, 6, 8, 9, and 12. Therefore, $g(X)$ has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9$, and α^{12} as all its roots. The elements $\alpha, \alpha^2, \alpha^4$, and α^8 have the same minimal polynomial, $\phi_1(X) = 1 + X + X^4$, and the elements $\alpha^3, \alpha^6, \alpha^9$, and α^{12} have the same minimal polynomial, $\phi_1(X) = 1 + X + X^2 + X^3 + X^4$. Thus, the generator polynomial of the (0, 2)th-order EG code of length 15 is

$$\begin{aligned} g(X) &= (1 + X + X^4)(1 + X + X^2 + X^3 + X^4) \\ &= 1 + X^4 + X^6 + X^7 + X^8. \end{aligned}$$

It is interesting to note that the (0, 2)th-order EG code is the (15, 7) BCH code considered in Example 8.1. It is one-step majority-logic decodable.

EXAMPLE 8.18

Let $m = 3, s = 2$, and $\mu = 1$. Then, the Galois field $GF(2^6)$ may be regarded as the Euclidean geometry $\mathrm{EG}(3, 2^2)$ over $GF(2^2)$. Let α be a primitive element in $GF(2^6)$ (use Table 6.2). Let h be a nonnegative integer less than 63. It follows from Theorem 8.3 that the generator polynomial $g(X)$ of the (1, 2)th-order EG code of length 63 has α^h as a root if and only if

$$0 < \max_{0 \le l < 2} W_{2^2}(h^{(l)}) \le 3.$$

The nonnegative integers less than 63 that satisfy this condition are

$$1,\ 2,\ 3,\ 4,\ 6,\ 8,\ 9,\ 12,\ 16,\ 18,\ 24,\ 32,\ 33,\ 48.$$

Thus, $g(X)$ has the following roots:

$$\alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}, \alpha^{16}, \alpha^{18}, \alpha^{24}, \alpha^{32}, \alpha^{33}, \alpha^{48}.$$

From Table 6.3 we find that

1. $\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$, and α^{32} have $\phi_1(X) = 1 + X + X^6$ as their minimal polynomial.
2. $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{33}$, and α^{48} have $\phi_3(X) = 1 + X + X^2 + X^4 + X^6$ as their minimal polynomial.
3. α^9, α^{18}, and α^{36} have the same minimal polynomial, $\phi_9(X) = 1 + X^2 + X^3$.

Therefore, the generator polynomial of the (1, 2)th-order EG code of length 63 is

$$\begin{aligned} g(X) &= (1 + X + X^6)(1 + X + X^2 + X^4 + X^6)(1 + X^2 + X^3) \\ &= 1 + X^2 + X^4 + X^{11} + X^{13} + X^{14} + X^{15}. \end{aligned}$$

Hence, the (1, 2)th-order EG code of length 63 is a (63, 48) cyclic code. Later we will show that this code is two-step orthogonalizable and is capable of correcting any combination of two or fewer errors.

Decoding of the (μ, s)th-order EG code of length $2^{ms} - 1$ is based on the structural properties of the Euclidean geometry EG$(m, 2^s)$. From Definition 8.3 we know that the null space of the code contains the incidence vectors of all the $(\mu + 1)$-flats of EG$(m, 2^s)$ that do not pass through the origin. Let $F^{(\mu)}$ be a μ-flat passing through the point $\alpha^{2^{ms}-2}$. From (8.27) we know that there are

$$J = \frac{2^{(m-\mu)s} - 1}{2^s - 1} - 1 \tag{8.30}$$

$(\mu + 1)$-flats not passing through the origin that intersect on $F^{(\mu)}$. The incidence vectors of these J $(\mu + 1)$-flats are orthogonal on the digits at the locations that correspond to the points in $F^{(\mu)}$. Therefore, the parity-check sums formed from these J incidence vectors are orthogonal on the error digits at the locations corresponding to the points in $F^{(\mu)}$. If there are $\lfloor J/2 \rfloor$ or fewer errors in the received vector, the sum of errors at the locations corresponding to the points in $F^{(\mu)}$ can be determined correctly. Let us denote this error sum with $S(F^{(\mu)})$. In this manner the error sum $S(F^{(\mu)})$ can be determined for every μ-flat $F^{(\mu)}$ passing through the point $\alpha^{2^{ms}-2}$. This forms the first step of orthogonalization.

We then use the error sums $S(F^{(\mu)})$'s corresponding to all the μ-flats $F^{(\mu)}$ that pass through the point $\alpha^{2^{ms}-2}$ for the second step of orthogonalization. Let $F^{(\mu-1)}$ be a $(\mu - 1)$-flat passing through the point $\alpha^{2^{ms}-2}$. From (8.27) we see that there are

$$J_1 = \frac{2^{(m-\mu+1)s} - 1}{2^s - 1} - 1 > J$$

μ-flats not passing through the origin that intersect on $F^{(\mu-1)}$. The error sums corresponding to these J_1 μ-flats are orthogonal on the error digits at the locations corresponding to the points in $F^{(\mu-1)}$. Let $S(F^{(\mu-1)})$ denote the sum of error digits at the locations corresponding to the points in $F^{(\mu-1)}$. Then, $S(F^{(\mu-1)})$ can be determined from the J_1 error sums $S(F^{(\mu)})$'s that are orthogonal on $S(F^{(\mu-1)})$. Since $J_1 > J$, if there are no more than $\lfloor J/2 \rfloor$ errors in the received vector, the error sum $S(F^{(\mu-1)})$ can be determined correctly. In this manner the error sum $S(F^{(\mu-1)})$ can be determined for every $(\mu - 1)$-flat $F^{(\mu-1)}$ passing through the point $\alpha^{2^{ms}-2}$ but not the origin. This completes the second step of orthogonalization.

The error sums $S(F^{(\mu-1)})$'s now are used for the third step of orthogonalization. Let $F^{(\mu-2)}$ be a $(\mu - 2)$-flat passing through the point $\alpha^{2^{ms}-2}$ but not the origin. From (8.27) we see that there are

$$J_2 = \frac{2^{(m-\mu+2)s} - 1}{2^s - 1} - 1 > J_1 > J$$

error sums $S(F^{(\mu-1)})$'s orthogonal on the error sum $S(F^{(\mu-2)})$. Hence, $S(F^{(\mu-2)})$ can be determined correctly. The error sums $S(F^{(\mu-2)})$'s are then used for the next step of orthogonalization. This process continues until the error sums corresponding to all the 1-flats (lines) passing through the point $\alpha^{2^{ms}-2}$ but not the origin are determined. There are

$$J_\mu = \frac{2^{ms}-1}{2^s-1} > J_{\mu-1} > \cdots > J_1 > J$$

such error sums orthogonal on the error digit $e_{2^{ms}-2}$ at the location $\alpha^{2^{ms}-2}$. Thus, $e_{2^{ms}-2}$ can be determined correctly from these orthogonal error sums provided that there are no more than $\lfloor J/2 \rfloor$ errors in the received vector. Because the code is cyclic, other error digits can successively be decoded in the same manner.

Because the decoding of each error digit requires $\mu + 1$ steps of orthogonalization, the (μ, s)th-order EG code of length $2^{ms} - 1$ is therefore $(\mu + 1)$-step majority-logic decodable. The code is capable of correcting

$$t_{ML} = \left\lfloor \frac{2^{(m-\mu)s}-1}{2(2^s-1)} - \frac{1}{2} \right\rfloor \tag{8.31}$$

or fewer errors. Therefore, its minimum distance is at least

$$2t_{ML} + 1 = 2^s(2^{(m-\mu-2)s} + \cdots + 2^s + 1). \tag{8.32}$$

Note that at each step of orthogonalization we need only J orthogonal error sums to determine an error sum for the next step. For $\mu = 0$, a $(0, s)$th-order EG code is one-step majority-logic decodable.

EXAMPLE 8.19

Let $m = 2$, $s = 2$ and $\mu = 0$. Consider the $(0, 2)$th-order EG code of length 15. From Example 8.17 we know that this code is the (15, 7) BCH code (also a type-1 DTI code). The null space of this code contains the incidence vectors of all the 1-flats (lines) in EG$(2, 2^2)$ that do not pass through the origin. To decode e_{14}, we need to determine the incidence vectors of the 1-flat passing through the point α^{14}, where α is a primitive element in $GF(2^4)$. There are

$$J = \frac{2^{2\cdot 2}-1}{2^2-1} - 1 = 4$$

such incidence vectors, which are given in Example 8.16. These four vectors are orthogonal on the digit position α^{14}. In fact, these are exactly the four orthogonal vectors $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$, and $\mathbf{w}_4$ given in Example 8.1.

EXAMPLE 8.20

Let $m = 4, s = 1$, and $\mu = 1$. Consider the (1, 1)th-order EG code of length $2^4 - 1 = 15$. Let α be a primitive element of $GF(2^4)$ given by Table 2.8. Let h be

a nonnegative integer less than 15. It follows from Theorem 8.3 that the generator polynomial $g(X)$ of this code has α^h as a root if and only if

$$0 < W_2(h^{(0)}) \leq 2.$$

Note that $h^{(0)} = h$. From the preceding condition we find that $g(X)$ has the following roots: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^8, \alpha^9, \alpha^{10}$, and α^{12}. From Table 2.9 we find that

$$\begin{aligned} g(X) &= (1 + X + X^4)(1 + X + X^2 + X^3 + X^4)(1 + X + X^2) \\ &= 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}. \end{aligned}$$

It is interesting to note that this EG code is actually the (15, 5) BCH code studied in Example 8.12.

The null space of this code contains the incidence vectors of all the 2-flats of the EG(4, 2) that do not pass through the origin. Now, we will show how to form orthogonal check-sums based on the structure of EG(4, 2). First, we treat $GF(2^4)$ as the geometry EG(4, 2). A 1-flat passing through the point α^{14} consists of the points of the form $\alpha^{14} + a\alpha^i$ with $a \in GF(2)$. There are thirteen 1-flats passing through α^{14} but not the origin, $\alpha^\infty = 0$; they are

$$\{\alpha^{13}, \alpha^{14}\}, \{\alpha^{12}, \alpha^{14}\}, \{\alpha^{11}, \alpha^{14}\}\{\alpha^{10}, \alpha^{14}\}, \{\alpha^9, \alpha^{14}\}, \{\alpha^8, \alpha^{14}\},$$
$$\{\alpha^7, \alpha^{14}\}, \{\alpha^6, \alpha^{14}\}, \{\alpha^5, \alpha^{14}\}, \{\alpha^4, \alpha^{14}\}, \{\alpha^3, \alpha^{14}\}, \{\alpha^2, \alpha^{14}\}, \{\alpha, \alpha^{14}\}.$$

For each of these 1-flats, there are

$$J = \frac{2^{(4-1)\cdot 1} - 1}{2^1 - 1} - 1 = 6$$

2-flats not passing through the origin that intersect on it. Each of these 2-flats consists of the points of the form $\alpha^{14} + a\alpha^i + b\alpha^j$, with a and b in $GF(2)$. The six 2-flats that intersect on the 1-flat $\{\alpha^{13}, \alpha^{14}\}$ are

$$\{\alpha^4, \alpha^{10}, \alpha^{13}, \alpha^{14}\}, \quad \{\alpha^7, \alpha^{12}, \alpha^{13}, \alpha^{14}\}, \quad \{\alpha^9, \alpha^{11}, \alpha^{13}, \alpha^{14}\},$$
$$\{\alpha^0, \alpha^8, \alpha^{13}, \alpha^{14}\}, \quad \{\alpha^1, \alpha^5, \alpha^{13}, \alpha^{14}\}, \quad \{\alpha^3, \alpha^6, \alpha^{13}, \alpha^{14}\}.$$

The incidence vectors of these six 2-flats are

	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$w_{11} =$	(0	0	0	0	1	0	0	0	0	0	1	0	0	1	1)
$w_{12} =$	(0	0	0	0	0	0	0	1	0	0	0	0	1	1	1)
$w_{13} =$	(0	0	0	0	0	0	0	0	0	1	0	1	0	1	1)
$w_{14} =$	(1	0	0	0	0	0	0	0	1	0	0	0	0	1	1)
$w_{15} =$	(0	1	0	0	0	1	0	0	0	0	0	0	0	1	1)
$w_{16} =$	(0	0	0	1	0	0	1	0	0	0	0	0	0	1	1).

Clearly, these six vectors are orthogonal on digits at locations α^{13} and α^{14}. Let r be the received vector. The parity-check sums formed from these six

orthogonal vectors are

$$\begin{aligned}
A_{11} &= \mathbb{w}_{11} \cdot r = e_4 + e_{10} + e_{13} + e_{14} \\
A_{12} &= \mathbb{w}_{12} \cdot r = e_7 + e_{12} + e_{13} + e_{14} \\
A_{13} &= \mathbb{w}_{13} \cdot r = e_9 + e_{11} + e_{13} + e_{14} \\
A_{14} &= \mathbb{w}_{14} \cdot r = e_0 + e_8 + e_{13} + e_{14} \\
A_{15} &= \mathbb{w}_{15} \cdot r = e_1 + e_5 + e_{13} + e_{14} \\
A_{16} &= \mathbb{w}_{16} \cdot r = e_3 + e_6 + e_{13} + e_{14}.
\end{aligned}$$

We see that these six check-sums orthogonal on $\{e_{e_{13}}, e_{14}\}$ are exactly the same check sums given in Example 8.12. Thus, we can determine the error sum $e_{13} + e_{14}$ corresponding to the 1-flat $\{\alpha^{13}, \alpha^{14}\}$ from these six check-sums.

In the same manner we can determine the error sums corresponding to the other twelve 1-flats passing through α^{14}. Because $J = 6$, we need to determine only six error sums corresponding to any six 1-flats passing through α^{14}. We then use these error sums to determine e_{14}. Thus, the (1, 1)th-order EG code of length 15 is a two-step majority-logic decodable code.

Except for certain special cases, there is no simple formula for enumerating the number of parity-check digits of EG codes. Complicated combinatorial expressions for the number of parity-check digits of EG codes can be found in [17] and [18]. One special case is $\mu = m - 2$. The number of parity-check digits for a $(m - 2, s)$th-order EG code of length $2^{ms} - 1$ is

$$n - k = \begin{pmatrix} m+1 \\ m \end{pmatrix}^s - 1. \tag{8.33}$$

This result was obtained independently by Smith [19] and by MacWilliams and Mann [20].

For $s = 1$, we obtain another special subclass of EG codes, which happens to be the class of RM codes of length $2^m - 1$ in cyclic form [11, 21–29]. A μth-order cyclic RM is simply a $(\mu, 1)$th-order EG code. If we add an overall parity bit to each codeword of this code, we obtain the μ-th order RM code of length 2^m presented in Section 4.3. Let α be a primitive element of the Galois field $GF(2^m)$. Let h be a nonnegative integer less than 2^m. It follows from Theorem 8.3 that the generator polynomial $\mathbb{g}(X)$ of the μth-order cyclic RM code of length $2^m - 1$ has α^h as a root if and only if

$$0 < W_2(h) \leq m - \mu - 1. \tag{8.34}$$

The μth-order cyclic RM code of length $2^m - 1$ has the following parameters:

$$k = \sum_{i=0}^{\mu} \begin{pmatrix} m \\ i \end{pmatrix},$$

$$d_{min} = 2^{m-\mu} - 1,$$

$$J = 2^{m-\mu} - 2.$$

Because $J = d_{min} - 1$, cyclic RM codes are completely orthogonalizable. The cyclic structure of RM codes was proved independently by Kasami et al. [21, 22] and Kolesnik and Mironchikov [23].

Except for RM codes and other special cases, EG codes in general are not completely orthogonalizable. For moderate-length n, the error-correcting capability of an EG code is slightly inferior to that of a comparable BCH code; however, the majority-logic decoding for EG codes is more simply implemented than the decoding for BCH codes. Thus, for moderate n, EG codes provide rather effective error control. For large-length n, EG codes become much inferior to the comparable BCH codes, and the number of majority-logic gates required for decoding becomes prohibitively large. In this case, BCH codes are definitely superior to the EG codes in error-correcting capability and decoding complexity. A list of EG codes with $n \leq 1023$ is given in Table 8.6. See [24] for a more extensive list.

TABLE 8.6: A list of EG codes.

m	s	μ	n	k	J	t_{ML}
3	1	1	7	4	2	1
4	1	2	15	11	2	1
4	1	1	15	5	6	3
2	2	0	15	7	4	2
5	1	3	31	26	2	1
5	1	2	31	16	6	3
5	1	1	31	6	14	7
6	1	4	63	57	2	1
6	1	3	63	42	6	3
6	1	2	63	22	14	7
6	1	1	63	7	31	15
3	2	1	63	48	4	2
3	2	0	63	13	20	10
2	3	0	63	37	8	4
7	1	5	127	120	2	1
7	1	4	127	99	6	3
7	1	3	127	64	14	7
7	1	2	127	29	30	15
7	1	1	127	8	62	31
8	1	6	255	247	2	1
8	1	5	255	219	6	3
8	1	4	255	163	14	7
8	1	3	255	93	30	15
8	1	2	255	37	62	31
8	1	1	255	9	126	63
4	2	2	255	231	4	2
4	2	1	255	127	20	10
4	2	0	255	21	84	42
2	4	0	255	175	16	8
9	1	7	511	502	2	1
9	1	6	511	466	6	3
9	1	5	511	382	14	7

TABLE 8.6: (*continued*)

m	s	μ	n	k	J	t_{ML}
9	1	4	511	256	30	15
9	1	3	511	130	62	31
9	1	2	511	46	126	63
9	1	1	511	10	254	127
3	3	1	511	448	8	4
3	3	0	511	139	72	36
10	1	8	1023	1013	2	1
10	1	7	1023	968	6	3
10	1	6	1023	848	14	7
10	1	5	1023	638	30	15
10	1	4	1023	386	62	31
10	1	3	1023	176	126	63
10	1	2	1023	56	254	127
10	1	1	1023	11	510	255
5	2	3	1023	988	4	2
5	2	2	1023	748	20	10
5	2	1	1023	288	84	42
5	2	0	1023	31	340	170
2	5	0	1023	781	32	16

A very special subclass of EG codes is the subclass of codes with $m = 2$ and $\mu = 0$. A code in this subclass is a $(0, s)$th-order EG code of length $n = 2^{2s} - 1$. The null space of this code contains the incidence vectors of all the lines in EG$(2, 2^s)$ not passing through the origin. It follows from (8.30) that 2^s check-sums orthogonal on any code digit can be formed. Therefore, the minimum distance of the code is at least $2^s + 1$. It follows from (8.29) that the generator polynomial $g(X)$ of this code has $\alpha^1, \alpha^2, \cdots, \alpha^{2^s}$ and their conjugates as roots. The polynomial $X^{2^{2s}-1} + 1$ can be factored as follows:

$$X^{2^{2s}-1} + 1 = (X^{2^s-1} + 1)(X^{2^s(2^s-1)} + \cdots + X^{2^s-1} + 1).$$

The first factor, $X^{2^s-1} + 1$, has $\alpha^0 = 1, \alpha^{2^s+1}, \alpha^{2(2^s+1)}, \cdots, \alpha^{(2^s-2)(2^s+1)}$ as all its roots. Then, the second factor, $v(X) = 1 + X^{2^s-1} + X^{2(2^s-1)} + \cdots + X^{2^s(2^s-1)}$, has $\alpha^1, \alpha^2, \cdots, \alpha^{2^s}$ as roots. Therefore, $v(X)$ must be a multiple of $g(x)$ (or divisible by $g(X)$) and hence it is a code polynomial of the $(0, s)$th-order EG code of length $n = 2^{2s} - 1$. This code polynomial $v(x)$ has a weight of exactly $2^s + 1$. This weight together with the bound that the minimum distance of the code is at least $2^s + 1$ imply that the minimum distance of the $(0, s)$th-order EG code of length $n = 2^{2s} - 1$ is exactly $2^s + 1$. Therefore, the code is one-step completely orthogonalizable. It follows from (8.33) that the number of parity-check digits of the $(0, s)$th-order EG code constructed based on the two-dimensional Euclidean geometry EG$(2, 2^s)$ is

$$n - k = 3^s - 1. \tag{8.35}$$

It is interesting to note that a $(0, s)$th-order EG code of length $n = 2^{2s} - 1$ is also a type-I DTI code given in Section 8.2.

EXAMPLE 8.21

Consider the (0, 6)th-order EG code of length $n = 2^{2\times 6} - 1 = 4095$ constructed based on the lines in EG(2, 2^6) not passing through the origin. This code is a (4095, 3367) cyclic code with a minimum distance of 65 and a rate of 0.83. Let α be a primitive element of $GF(2^{12})$. The generator polynomial of this code has $\alpha^1, \alpha^2, \cdots, \alpha^{64}$ as roots. It is one-step completely orthogonalizable and can correct up to 32 errors with one-step majority-logic decoding, either type I or type II. The error performance of this code on an AWGN channel with BPSK signaling and one-step majority-logic decoding is shown in Figure 8.10. At the BER of 10^{-5}, it achieves a 4-dB coding gain over the uncoded BPSK. The majority-logic decoder for this code can easily be implemented in hardware with a feedback shift register of 728 flip-flops and a majority-logic circuit with 64 inputs. This hardware implementation can achieve very high decoding speed and is quite suitable for high-speed optical networks operating at 10 Gbits or for high-speed satellite communications. Consider the NASA standard (255, 223, 33) RS code over $GF(2^8)$. For binary transmission,

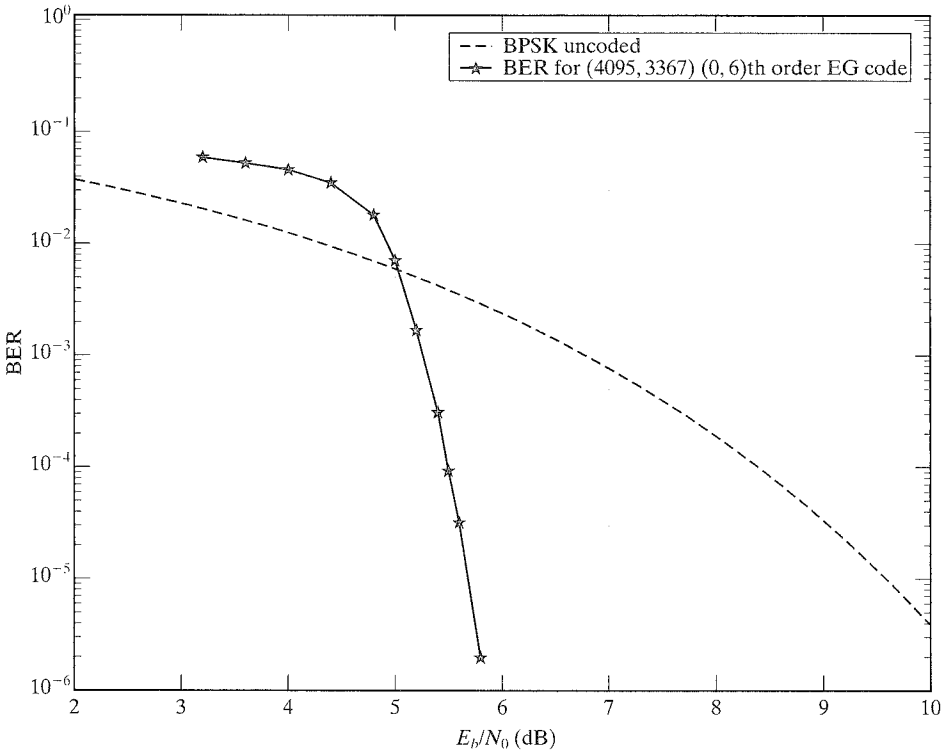

FIGURE 8.10: Bit-error performance of the (4095, 3367) (0, 6)th-order EG code with majority-logic decoding.

each code symbol is expanded into an 8-bit byte. This symbol-to-binary expansion results in a (2040, 1784) binary code. At the receiving end, the received digits are grouped back into symbols in $GF(2^8)$ for decoding. From Figures 7.3 and 8.10 we see that the (4095, 3367) (0, 6)th-order EG code outperforms the (255, 223, 33) RS code by more than 0.5 dB at the BER of 10^{-5}. Even though the (4095, 3367) EG code is twice as long as the (255, 223, 33) RS code in binary form, its decoding complexity is much simpler, because majority-logic decoding requires only simple binary logic operations, whereas decoding of the (255, 223, 33) RS code with algebraic decoding algorithms requires computations in $GF(2^8)$ to find the error-location polynomial and the error-value enumerator. The (4095, 3367) EG code can also be decoded with several other hard- or soft-decision decoding methods to achieve better error performance at the expense of increasing decoding complexity. This topic will be discussed in Chapter 17.

EG codes were first studied by Rudolph [3]. Rudolph's work was later extended and generalized by other coding theorists [24–30]. Improvements for decoding EG codes were suggested by Weldon [31] and by Chen [32]. Chen proved that any EG code can be decoded in *no more* than three steps. Chen's decoding algorithm is based on further structure of the Euclidean geometry, which is not covered in this introductory book.

There are several classes of generalized EG codes [24, 28–30, 34] that all contain EG codes as subclasses. We will not cover these generalizations here; however, we present a simple generalization using parallel flats next.

8.7 TWOFOLD EG CODES

Let F and F_1 be any two parallel μ-flats in $\mathrm{EG}(m, 2^s)$. We say that F and F_1 form a $(\mu, 2)$-frame in $\mathrm{EG}(m, 2^s)$, denoted by $\{F, F_1\}$. Because F and F_1 do not have any point in common, the $(\mu, 2)$-frame $\{F, F_1\}$ consists of $2^{\mu s+1}$ points. Let F_2 be another μ-flat parallel to F and F_1. Then, the two $(\mu, 2)$-frames $\{F, F_1\}$ and $\{F, F_2\}$ intersect on F. Let L be a $(\mu+1)$-flat that contains the μ-flat F. Then, L contains $2^s - 1$ other μ-flats that are parallel to F. Each of these $2^s - 1$ μ-flats together with F forms a $(\mu, 2)$-frame. There are $2^s - 1$ such $(\mu, 2)$-frames that intersect on F. Clearly, these $2^s - 1$ $(\mu, 2)$-frames are all contained in the $(\mu+1)$-flat L. Any point in L but outside F is *in on one and only one of these* $2^s - 1$ $(\mu, 2)$*-frames*. Because there are

$$\frac{2^{(m-\mu)s} - 1}{2^s - 1}$$

$(\mu + 1)$-flats that intersect on F, there are

$$(2^s - 1) \cdot \frac{2^{(m-\mu)s} - 1}{2^s - 1} = 2^{(m-\mu)s} - 1 \tag{8.36}$$

$(\mu, 2)$-frames that intersect on F. Any point outside F is in on one and only one of these $(\mu, 2)$-frames. We say that these $(\mu, 2)$-frames are orthogonal on the μ-flat F. If F does not pass through the origin, there are

$$2^{(m-\mu)s} - 2 \tag{8.37}$$

$(\mu, 2)$-frames that are orthogonal on F and do not pass through the origin.

Again, we regard the Galois field $GF(2^{ms})$ as the geometry $\mathrm{EG}(m, 2^s)$. Let α be a primitive element of $GF(2^{ms})$. For any $(2^{ms}-1)$-tuple

$$\mathbf{v} = (v_0, v_1, \cdots, v_{2^{ms}-2})$$

over $GF(2)$, we again number its components with the nonzero elements of $GF(2^{ms})$ as usual (i.e., v_i is numbered with α^i for $0 \leq i < 2^{ms}-1$). For each $(\mu, 2)$-frame Q in $\mathrm{EG}(m, 2^s)$, we define its incidence vector as follows:

$$\mathbf{v}_Q = (v_0, v_1, \cdots, v_{2^{ms}-2}),$$

where the ith component is

$$v_i = \begin{cases} 1 & \text{if } \alpha^i \text{ is a point in } Q, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 8.4 A (μ, s)th-order twofold EG code of length $2^{ms}-1$ is the largest cyclic code whose null space contains the incidence vectors of all the $(\mu, 2)$-frames in $\mathrm{EG}(m, 2^s)$ that do not pass through the origin.

We now state a theorem (without proof) [34] that characterizes the roots of the generator polynomial of a (μ, s)th-order twofold EG code.

Theorem 8.4 Let α be a primitive element of the Galois field $GF(2^{ms})$. Let h be a nonnegative integer less than $2^{ms}-1$. The generator polynomial $\mathbf{g}(X)$ of the (μ, s)th-order twofold EG code of length $2^{ms}-1$ has α^h as a root if and only if

$$0 < \max_{0 \leq l < s} W_{2^s}(h^{(l)}) < (m-\mu)(2^s-1). \tag{8.38}$$

EXAMPLE 8.22

Let $m = 2$, $s = 3$, and $\mu = 1$. Consider the $(1, 3)$th-order twofold EG code of length 63. Let α be a primitive element of $GF(2^6)$ given by Table 6.2. Let h be a nonnegative integer less than 63. It follows from (8.38) that the generator polynomial $\mathbf{g}(X)$ of the $(1, 3)$th-order twofold EG code of length 63 has α^h as a root if and only if

$$0 < \max_{0 \leq l < 3} W_{2^3}(h^{(l)}) < 7.$$

The nonnegative integers less than 63 that satisfy this condition are

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 8,\ 9,\ 10,\ 12,\ 16,\ 17,\ 20,\ 24,\ 32,\ 33,\ 34,\ 40,\ 48.$$

Thus, the generator polynomial $\mathbf{g}(X)$ has the following roots:

$$\alpha_1,\ \alpha^2,\ \alpha^3,\ \alpha^4,\ \alpha^5,\ \alpha^6,\ \alpha^8,\ \alpha^9,\ \alpha^{10},\ \alpha^{12},$$
$$\alpha^{16},\ \alpha^{17},\ \alpha^{20},\ \alpha^{24},\ \alpha^{32},\ \alpha^{33},\ \alpha^{34},\ \alpha^{40},\ \alpha^{48}.$$

From Table 6.3 we find that:

1. The roots $\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$, and α^{32} have the same minimal polynomial, $\phi_1(X) = 1 + X + X^6$.

2. The roots $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}$ and α^{33} have the same minimal polynomial, $\phi_3(X) = 1 + X + X^2 + X^4 + X^6$.
3. The roots $\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}$, and α^{34} have the same minimal polynomial, $\phi_5(X) = 1 + X + X^2 + X^5 + X^6$.

Therefore,

$$\begin{aligned} g(X) &= \phi_1(X) \cdot \phi_3(X) \cdot \phi_5(X) \\ &= 1 + X + X^2 + X^3 + X^6 + X^7 + X^9 + X^{15} + X^{16} + X^{17} + X^{18}. \end{aligned}$$

Therefore, the (1, 3)th-order twofold EG code of length 63 with $m = 2$ is a (63, 45) cyclic code. In fact, it is the (63, 45) BCH code with a minimum distance equal to 7.

To decode the (μ, s)th-order twofold EG code of length $2^{ms} - 1$, we first form the parity-check sums from the incidence vectors of all the $(\mu, 2)$-frames in EG $(m, 2^s)$ that do not pass through the origin (note that these incidence vectors are in the null space of the code). Let $F^{(\mu)}$ be a μ-flat that passes through the point $\alpha^{2^{ms}-2}$. From (8.37) we see that there are

$$J = 2^{(m-\mu)s} - 2 \tag{8.39}$$

$(\mu, 2)$-frames not passing through the origin that are orthogonal on $F^{(\mu)}$. The incidence vectors of these $(\mu, 2)$-frames are orthogonal on the digits at the locations that correspond to the points in $F^{(\mu)}$. Therefore, the parity-check sums formed from these J incidence vectors are orthogonal on the error digits at the locations that correspond to the points in $F^{(\mu)}$. Let $S(F^{(\mu)})$ denote the sum of error digits at the locations corresponding to the points in $F^{(\mu)}$. Then, we can correctly determine this error sum, $S(F^{(\mu)})$, from the J check-sums orthogonal on it provided that there are no more than

$$\left\lfloor \frac{J}{2} \right\rfloor = 2^{(m-\mu)s-1} - 1$$

errors in the received vector. In this manner we can determine the error sums, $S(F^{(\mu)})$'s that correspond to all the μ-flats passing through the point $\alpha^{2^{ms}-2}$. This completes the first step of orthogonalization. After this step, the rest of orthogonalization steps are the same as those for a (μ, s)th-order EG code. Therefore, a total of $\mu + 1$ steps of orthogonalization are needed to decode a (μ, s)th-order twofold EG code.

We can easily check that at each decoding step there are at least $J = 2^{(m-\mu)s} - 2$ error sums orthogonal on an error sum for the next step. Thus, the (μ, s)th-order twofold EG code of length $2^{ms} - 1$ is capable of correcting.

$$t_{ML} = \left\lfloor \frac{J}{2} \right\rfloor = 2^{(m-\mu)s-1} - 1 \tag{8.40}$$

or fewer errors with majority-logic decoding. It has been proved [34] that the minimum distance of the (μ, s)th-order twofold EG code of length $2^{ms} - 1$ is exactly $2^{(m-\mu)s} - 1$. Therefore, the class of twofold EG codes is completely orthogonalizable.

EXAMPLE 8.23

Consider the decoding of the (1, 3)th-order twofold EG of length 63 with $m = 2$ and $s = 3$. In Example 8.22 we showed that this code is a (63, 45) cyclic code (also a BCH code). The null space of this code contains the incidence vectors of all the (1, 2)-frames in EG(2, 2^3) that do not pass through the origin. Regard $GF(2^6)$ as the geometry EG(2, 2^3). Let α be a primitive element of $GF(2^6)$ (use Table 6.2). From (8.26) we see that there are nine lines in EG(2, 2^3) that intersect at the point α^{62}. Eight of these lines do not pass through the origin. From (8.37) we see that for each of these eight lines there are six (1, 2)-frames intersecting on it. The incidence vectors of these six (1, 2)-frames are in the null space of the code, and they will be used to form parity-check sums for decoding the error digit e_{62} at location α^{62}. Because $J = 6$, we need to find only six lines in EG(2, 2^3) that intersect at the point α^{62} and do not pass through the origin.

Let $\beta = \alpha^9$. Then, $0, 1, \beta, \beta^2, \beta^3, \beta^4, \beta^5$, and $\beta^6 (\beta^7 = 1)$ form a subfield $GF(2^3)$ of the field $GF(2^6)$ (use Table 6.2). Then, each line in EG(2, 2^3) that passes through α^{62} consists of the following points:

$$\alpha^{62} + \eta\alpha^j$$

where $\eta \in \{0, 1, \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6\}$. Six lines passing through the point α^{62} are as follows:

$$L_1 = \{\alpha^{11}, \alpha^{16}, \alpha^{18}, \alpha^{24}, \alpha^{48}, \alpha^{58}, \alpha^{59}, \alpha^{62}\},$$
$$L_2 = \{\alpha^{1}, \alpha^{7}, \alpha^{31}, \alpha^{41}, \alpha^{42}, \alpha^{45}, \alpha^{57}, \alpha^{62}, \}$$
$$L_3 = \{\alpha^{23}, \alpha^{33}, \alpha^{34}, \alpha^{37}, \alpha^{49}, \alpha^{54}, \alpha^{56}, \alpha^{62}\},$$
$$L_4 = \{\alpha^{2}, \alpha^{12}, \alpha^{19}, \alpha^{21}, \alpha^{27}, \alpha^{51}, \alpha^{61}, \alpha^{62}\},$$
$$L_5 = \{\alpha^{0}, \alpha^{3}, \alpha^{15}, \alpha^{20}, \alpha^{22}, \alpha^{28}, \alpha^{52}, \alpha^{62}\},$$
$$L_6 = \{\alpha^{9}, \alpha^{10}, \alpha^{13}, \alpha^{25}, \alpha^{30}, \alpha^{32}, \alpha^{38}, \alpha^{62}\}.$$

For each of these lines we form six (1, 2)-frames intersecting on it. A (1, 2)-frame that contains the line $\{\alpha^{62} + \eta\alpha^j\}$ is of the form

$$(\{\alpha^{62} + \eta\alpha^j\}, \{\alpha^{62} + \alpha^i + \eta\alpha^j\}),$$

where α^i is not in $\{\alpha^{62} + \eta\alpha^j\}$. Line L_1 consists of the points $\{\alpha^{62} + \eta\alpha\}$. The point α^9 is not in L_1. Then, the line $\{\alpha^{62} + \alpha^9 + \eta\alpha\}$ is parallel to $\{\alpha^{62} + \eta\alpha\}$. Thus,

$$(\{\alpha^{62} + \eta\alpha\}, \{\alpha^{62} + \alpha^9 + \eta\alpha\})$$

forms a (1, 2)-frame containing the line L_1. This (1, 2)-frame consists of the following points:

$$\{\alpha^{11}, \alpha^{16}, \alpha^{18}, \alpha^{21}, \alpha^{24}, \alpha^{31}, \alpha^{32}, \alpha^{35}, \alpha^{47}, \alpha^{48}, \alpha^{52}, \alpha^{54}, \alpha^{58}, \alpha^{59}, \alpha^{60}, \alpha^{62}\}.$$

In this manner, for each line L_i, we can form six (1, 2)-frames orthogonal on it. The incidence vectors of these 36 (1, 2)-frames are given in Tables 8.7A through 8.7F.

TABLE 8.7A: Polynomials orthogonal on $\{e_{11}, e_{16}, e_{18}, e_{24}, e_{48}, e_{58}, e_{59}, e_{62}\}$.

$\mathbb{w}_{11}(X)^* = (11, 16, 18, 21, 24, 31, 32, 35, 47, 48, 52, 54, 58, 59, 60, 62)$
$\mathbb{w}_{12}(X) = (11, 12, 16, 18, 22, 23, 24, 26, 38, 43, 45, 48, 51, 58, 59, 62)$
$\mathbb{w}_{13}(X) = (0, 6, 11, 16, 18, 24, 30, 40, 41, 44, 48, 56, 58, 59, 61, 62)$
$\mathbb{w}_{14}(X) = (4, 5, 8, 11, 16, 18, 20, 24, 25, 27, 33, 48, 57, 58, 59, 62)$
$\mathbb{w}_{15}(X) = (3, 11, 13, 14, 16, 17, 18, 24, 29, 34, 36, 42, 48, 58, 59, 62)$
$\mathbb{w}_{16}(X) = (2, 7, 9, 11, 15, 16, 18, 24, 39, 48, 49, 50, 53, 58, 59, 62)$

*In Tables 8.7A through 8.7F, the integers inside the parentheses are powers of X.

TABLE 8.7B: Polynomials orthogonal on $\{e_1, e_7, e_{31}, e_{41}, e_{42}, e_{45}, e_{57}, e_{62}\}$.

$\mathbb{w}_{21}(X) = (1, 7, 13, 23, 24, 27, 31, 39, 41, 42, 44, 45, 46, 52, 57, 62)$
$\mathbb{w}_{22}(X) = (1, 7, 22, 31, 32, 33, 36, 41, 42, 45, 48, 53, 55, 57, 61, 62)$
$\mathbb{w}_{23}(X) = (0, 1, 7, 12, 17, 19, 25, 31, 41, 42, 45, 49, 57, 59, 60, 62)$
$\mathbb{w}_{24}(X) = (1, 5, 6, 7, 9, 21, 26, 28, 31, 34, 41, 42, 45, 57, 58, 62)$
$\mathbb{w}_{25}(X) = (1, 3, 7, 8, 10, 16, 31, 40, 41, 42, 45, 50, 51, 54, 57, 62)$
$\mathbb{w}_{26}(X) = (1, 4, 7, 14, 15, 18, 30, 31, 35, 37, 41, 42, 43, 45, 57, 62)$

TABLE 8.7C: Polynomials orthogonal on $\{e_{23}, e_{33}, e_{34}, e_{37}, e_{49}, e_{54}, e_{56}, e_{62}\}$.

$\mathbb{w}_{31}(X) = (4, 9, 11, 17, 23, 33, 34, 37, 41, 49, 51, 52, 54, 55, 56, 62)$
$\mathbb{w}_{32}(X) = (5, 15, 16, 19, 23, 31, 33, 34, 36, 37, 38, 44, 49, 54, 56, 62)$
$\mathbb{w}_{33}(X) = (1, 13, 18, 20, 23, 26, 33, 34, 37, 42, 43, 46, 49, 54, 56, 58, 62)$
$\mathbb{w}_{34}(X) = (0, 2, 8, 23, 32, 33, 34, 37, 42, 43, 46, 49, 54, 56, 58, 62)$
$\mathbb{w}_{35}(X) = (14, 23, 24, 25, 28, 33, 34, 37, 40, 45, 47, 49, 53, 54, 56, 62)$
$\mathbb{w}_{36}(X) = (6, 7, 10, 22, 23, 27, 29, 33, 34, 35, 37, 49, 54, 56, 59, 62)$

TABLE 8.7D: Polynomials orthogonal on $\{e_2, e_{14}, e_{19}, e_{21}, e_{27}, e_{51}, e_{61}, e_{62}\}$.

$\mathbb{w}_{41}(X) = (2, 7, 8, 11, 14, 19, 21, 23, 27, 28, 30, 36, 51, 60, 61, 62)$
$\mathbb{w}_{42}(X) = (0, 2, 14, 19, 21, 24, 27, 34, 35, 38, 50, 51, 55, 57, 61, 62)$
$\mathbb{w}_{43}(X) = (2, 14, 15, 19, 21, 25, 26, 27, 29, 41, 46, 48, 51, 54, 61, 62)$
$\mathbb{w}_{44}(X) = (1, 2, 3, 9, 14, 19, 21, 27, 33, 43, 44, 47, 51, 59, 61, 62)$
$\mathbb{w}_{45}(X) = (2, 6, 14, 16, 17, 19, 20, 21, 27, 32, 37, 39, 45, 51, 61, 62)$
$\mathbb{w}_{46}(X) = (2, 5, 10, 12, 14, 18, 19, 21, 27, 42, 51, 52, 53, 56, 61, 62)$

To decode the code, the incidence vectors of the 36 (1, 2)-frames given in Tables 8.7A through 8.7F are used to form parity-check sums. Let $S(L_i)$ denote the sum of error digits at the locations corresponding to the points on line L_i for $1 \leq i \leq 6$. Then, for each error sum $S(L_i)$, there are six parity-check sums orthogonal

TABLE 8.7E: Polynomials orthogonal on $\{e_0, e_3, e_{15}, e_{20}, e_{22}, e_{28}, e_{52}, e_{62}\}$.

$w_{51}(X) = (0, 3, 6, 11, 13, 15, 19, 20, 22, 28, 43, 52, 53, 54, 57, 62)$
$w_{52}(X) = (0, 3, 8, 9, 12, 15, 20, 22, 24, 28, 29, 31, 37, 52, 61, 62)$
$w_{53}(X) = (0, 1, 3, 15, 20, 22, 25, 28, 35, 36, 39, 51, 52, 56, 58, 62)$
$w_{54}(X) = (0, 3, 15, 16, 20, 22, 26, 27, 28, 30, 42, 47, 49, 52, 55, 62)$
$w_{55}(X) = (0, 2, 3, 4, 10, 15, 20, 22, 28, 34, 44, 45, 48, 52, 60, 62)$
$w_{56}(X) = (0, 3, 7, 15, 17, 18, 20, 21, 22, 28, 33, 38, 40, 46, 52, 62)$

TABLE 8.7F: Polynomials orthogonal on $\{e_9, e_{10}, e_{13}, e_{25}, e_{30}, e_{32}, e_{38}, e_{62}\}$.

$w_{61}(X) = (3, 5, 9, 10, 11, 13, 25, 30, 32, 35, 38, 45, 46, 49, 61, 62)$
$w_{62}(X) = (9, 10, 13, 17, 25, 27, 28, 30, 31, 32, 38, 43, 48, 50, 56, 62)$
$w_{63}(X) = (0, 1, 4, 9, 10, 13, 16, 21, 23, 25, 29, 30, 32, 38, 53, 62)$
$w_{64}(X) = (8, 9, 10, 13, 18, 19, 22, 25, 30, 32, 34, 38, 39, 41, 47, 62)$
$w_{65}(X) = (7, 9, 10, 12, 13, 14, 20, 25, 30, 32, 38, 44, 54, 55, 58, 62)$
$w_{66}(X) = (2, 9, 10, 13, 25, 26, 30, 32, 36, 37, 38, 40, 52, 57, 59, 62)$

on it. Thus, $S(L_i)$ can be determined correctly provided that there are three or fewer errors in the error vector.

The error sums $S(L_1)$, $S(L_2)$, $S(L_3)$, $S(L_4)$, $S(L_5)$, and $S(L_6)$ are orthogonal on e_{62}. Consequently, e_{62} can be determined from these error sums. Thus, the (1, 3)th-order twofold (63, 45) EG code is two-step majority-logic decodable. Because its minimum distance $d_{min} = 7$ and $J = 6$, it is completely orthogonalizable.

There is no simple formula for enumerating the number of parity-check digits for a general twofold EG code; however, for $\mu = m - 1$, the number of parity-check digits for the $(m - 1, s)$th-order twofold EG code of length $2^{ms} - 1$ is [34]

$$n - k = \binom{m+1}{m}^s - \binom{m}{m-1}^s. \tag{8.41}$$

A list of twofold EG codes is given in Table 8.8. We see that the twofold EG codes are more efficient than their corresponding RM codes and are comparable to their corresponding BCH codes. For example, for error-correcting capability $t = 7$, there is a two-step majority-logic decodable (255, 191) twofold EG code; the corresponding RM code is a (255, 163) code that is five-step majority-logic decodable (using Chen's decoding algorithm [32], it may be decoded in two steps); the corresponding BCH code is a (255, 199) code. Twofold EG codes form a special subclass of multifold EG codes presented in [30] and [34].

TABLE 8.8: Twofold EG codes*.

m	s	μ	n	k	J	t_{ML}
3	2	1	63	24	14	7
2	3	1	63	45	6	3
4	2	1	255	45	62	31
4	2	2	255	171	14	7
2	4	1	255	191	14	7
3	3	1	511	184	62	31
3	3	2	511	475	6	3
5	2	1	1023	76	254	127
5	2	2	1023	438	62	31
5	2	3	1023	868	14	7
2	5	1	1023	813	30	15

*The (63, 24) and (63, 45) codes are BCH codes.

8.8 PROJECTIVE GEOMETRY AND PROJECTIVE GEOMETRY CODES

Like Euclidean geometry, a projective geometry may be constructed from the elements of a Galois field. Consider the Galois field $GF(2^{(m+1)s})$ that contains $GF(2^s)$ as a subfield. Let α be a primitive element in $GF(2^{(m+1)s})$. Then, the powers of $\alpha, \alpha^0, \alpha^1, \cdots, \alpha^{2^{(m+1)s}-2}$ form all the nonzero elements of $GF(2^{(m+1)s})$. Let

$$n = \frac{2^{(m+1)s} - 1}{2^s - 1} = 2^{ms} + 2^{(m-1)s} + \cdots + 2^s + 1. \tag{8.42}$$

Then, the order of $\beta = \alpha^n$ is $2^s - 1$. The 2^s elements $0, 1, \beta, \beta^2, \cdots, \beta^{2^s-2}$ form the Galois field $GF(2^s)$.

Consider the first n powers of α:

$$\Gamma = \{\alpha^0, \alpha^1, \alpha^2, \cdots, \alpha^{n-1}\}.$$

No element α^i in Γ can be a product of an element in $GF(2^s)$ and another element α^j in Γ [i.e., $\alpha^i \neq \eta \cdot \alpha^j$ for $\eta \in GF(2^s)$]. Suppose that $\alpha^i = \eta\alpha^j$. Then, $\alpha^{i-j} = \eta$. Because $\eta^{2^s-1} = 1$, we obtain $\alpha^{(i-j)(2^s-1)} = 1$. This is impossible, since $(i-j)(2^s-1) < 2^{(m+1)s} - 1$, and the order of α is $2^{(m+1)s} - 1$. Therefore, we conclude that for α^i and α^j in Γ, $\alpha^i \neq \eta\alpha^j$ for any $\eta \in GF(2^s)$. Now, we partition the nonzero elements of $GF(2^{(m+1)s})$ into n disjoint subsets as follows:

$$\begin{aligned}
&\{\alpha^0, \beta\alpha^0, \beta^2\alpha^0, \cdots, \beta^{2^s-2}\alpha^0\},\\
&\{\alpha^1, \beta\alpha^1, \beta^2\alpha^1, \cdots, \beta^{2^s-2}\alpha^1\},\\
&\{\alpha^2, \beta\alpha^2, \beta^2\alpha^2, \cdots, \beta^{2^s-2}\alpha^2\},\\
&\vdots\\
&\{\alpha^{n-1}, \beta\alpha^{n-1}, \beta^2\alpha^{n-1}, \cdots, \beta^{2^s-2}\alpha^{n-1}\},
\end{aligned}$$

where $\beta = \alpha^n$, a primitive element in $GF(2^s)$. Each set consists of $2^s - 1$ elements, and each element is a multiple of the first element in the set. No element in one set can be a product of an element of $GF(2^s)$ and an element from a different set. Now, we represent each set by its first element as follows:

$$(\alpha^i) \triangleq \{\alpha^i, \beta\alpha^i, \cdots, \beta^{2^s-2}\alpha^i\},$$

with $0 \leq i < n$. For any α^j in $GF(2^{(m+1)s})$, if $\alpha^j = \beta^l \cdot \alpha^i$ with $0 \leq i < n$, then α^j is represented by (α^i). If each element in $GF(2^{(m+1)s})$ is represented as an $(m+1)$-tuple over $GF(2^s)$, then (α^i) consists of $2^s - 1$ $(m+1)$-tuples over $GF(2^s)$. The $(m+1)$-tuple for α^i represents the $2^s - 1$ $(m+1)$-tuples in (α^i). All the $(m+1)$-tuples representing the elements in (α^i) are multiples of the $(m+1)$-tuple representing α^i. The $(m+1)$-tuple over $GF(2^s)$ that represents (α^i) may be regarded as a point in a finite geometry over $GF(2^s)$. Then, the points

$$(\alpha^0), (\alpha^1), (\alpha^2), \cdots, (\alpha^{n-1})$$

are said to form an m-dimensional projective geometry over $GF(2^s)$, denoted by $\mathrm{PG}(m, 2^s)$ [15, 16]. In this geometry, the $2^s - 1$ elements in $\{\alpha^i, \beta\alpha^i, \cdots, \beta^{(2^s-2)}\alpha^i\}$ are considered to be the same point in $\mathrm{PG}(m, 2^s)$. This is a major difference between a projective geometry and a Euclidean geometry. A projective geometry does not have an origin.

Let (α^i) and (α^j) be any two *distinct* points in $\mathrm{PG}(m, 2^s)$. Then, the *line* (1-flat) passing through (or connecting) (α^i) and (α^j) consists of points of the following form:

$$(\eta_1\alpha^i + \eta_2\alpha^j), \tag{8.43}$$

where η_1 and η_2 are from $GF(2^s)$ and are not both equal to zero. There are $(2^s)^2 - 1$ possible choices of η_1 and η_2 from $GF(2^s)$ (excluding $\eta_1 = \eta_2 = 0$); however, there are always $2^s - 1$ choices of η_1 and η_2 that result in the same point. For example,

$$\eta_1\alpha^i + \eta_2\alpha^j, \beta\eta_1\alpha^i + \beta\eta_2\alpha^j, \cdots, \beta^{2^s-2}\eta_1\alpha^i + \beta^{2^s-2}\eta_2\alpha^j$$

represent the same point in $\mathrm{PG}(m, 2^s)$. Therefore, a line in $\mathrm{PG}(m, 2^s)$ consists of

$$\frac{(2^s)^2 - 1}{2^s - 1} = 2^s + 1$$

points. To generate the $2^s + 1$ distinct points on the line $\{(\eta_1\alpha^i + \eta_2\alpha^j)\}$, we simply choose η_1 and η_2 such that no choice (η_1, η_2) is a multiple of another choice (η_1', η_2') [i.e., $(\eta_1, \eta_2) \neq (\delta\eta_1', \delta\eta_2')$ for any $\delta \in GF(2^s)$].

EXAMPLE 8.24

Let $m = 2$ and $s = 2$. Consider the projective geometry $\mathrm{PG}(2, 2^2)$. This geometry can be constructed from the field $GF(2^6)$, which contains $GF(2^2)$ as a subfield. Let

$$n = \frac{2^6 - 1}{2^2 - 1} = 2^{2\cdot 2} + 2^2 + 1 = 21.$$

Let α be a primitive of $GF(2^6)$ (use Table 6.2). Let $\beta = \alpha^{21}$. Then 0,1,β, and β^2 form the field $GF(2^2)$. The geometry PG$(2, 2^2)$ consists of the following 21 points:

$$(\alpha^0),\ (\alpha^1),\ (\alpha^2),\ (\alpha^3),\ (\alpha^4),\ (\alpha^5),\ (\alpha^6),$$
$$(\alpha^7),\ (\alpha^8),\ (\alpha^9),\ (\alpha^{10}),\ (\alpha^{11}),\ (\alpha^{12}),\ (\alpha^{13}),$$
$$(\alpha^{14}),\ (\alpha^{15}),\ (\alpha^{16}),\ (\alpha^{17}),\ (\alpha^{18}),\ (\alpha^{19}),\ (\alpha^{20}).$$

Consider the line passing through the point (α) and (α^{20}) that consists of five points of the form $(\eta_1\alpha + \eta_2\alpha^{20})$, with η_1 and η_2 from $GF(2^2) = \{0, 1, \beta, \beta^2\}$. The five distinct points are

$$\begin{aligned}
&(\alpha),\\
&(\alpha^{20}),\\
&(\alpha + \alpha^{20}) = (\alpha^{57}) = (\beta^2\alpha^{15}) = (\alpha^{15}),\\
&(\alpha + \beta\alpha^{20}) = (\alpha + \alpha^{41}) = (\alpha^{56}) = (\beta^2\alpha^{14}) = (\alpha^{14}),\\
&(\alpha + \beta^2\alpha^{20}) = (\alpha + \alpha^{62}) = (\alpha^{11}).
\end{aligned}$$

Thus, $\{(\alpha), (\alpha^{11}), (\alpha^{14}), (\alpha^{15}), (\alpha^{20})\}$ is the line in PG$(2, 2^s)$ that passes through the points (α) and (α^{20}).

Let (α^l) be a point not on the line $\{(\eta_1\alpha^i + \eta_2\alpha^j)\}$. Then, the line $\{(\eta_1\alpha^i + \eta_2\alpha^j)\}$ and the line $\{(\eta_1\alpha^l + \eta_2\alpha^j)\}$ have (α^j) as a common point (the only common point). We say that they intersect at (α^j). The number of lines in PG$(m, 2^s)$ that intersect at a given point is

$$\frac{2^{ms} - 1}{2^s - 1} = 1 + 2^s + \cdots + 2^{(m-1)s}. \tag{8.44}$$

Let $(\alpha^{l_1}), (\alpha^{l_2}), \cdots, (\alpha^{l_{\mu+1}})$ be $\mu + 1$ linearly independent points (i.e., $\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots + \eta_{\mu+1}\alpha^{l_{\mu+1}} = 0$ if and only if $\eta_1 = \eta_2 = \cdots = \eta_{\mu+1} = 0$). Then, a μ-flat in PG$(m, 2^s)$ consists of points of the form

$$(\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots + \eta_{\mu+1}\alpha^{l_{\mu+1}}), \tag{8.45}$$

where $\eta_i \in GF(2^s)$, and not all $\eta_1, \eta_2, \cdots, \eta_{\mu+1}$ are zero. There are $2^{(\mu+1)s} - 1$ choices for $\eta_1, \eta_2, \cdots, \eta_{\mu+1}$ ($\eta_1 = \eta_2 = \cdots = \eta_{\mu+1} = 0$ is not allowed). Because there are always $2^s - 1$ choices of η_1 to $\eta_{\mu+1}$ resulting in the same point in PG$(m, 2^s)$, there are

$$\frac{2^{(\mu+1)s} - 1}{2^s - 1} = 1 + 2^2 + \cdots + 2^{\mu s} \tag{8.46}$$

points in a μ-flat in PG$(m, 2^s)$. Let $\alpha^{l'_{\mu+1}}$ be a point not in the μ-flat:

$$\{(\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots + \eta_{\mu+1}\alpha^{l_{\mu+1}})\}.$$

Then, the μ-flat $\{(\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots + \eta_\mu\alpha^{l_\mu} + \eta_{\mu+1}\alpha^{l_{\mu+1}})\}$ and the μ-flat $\{(\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots + \eta_\mu\alpha^{l_\mu} + \eta_{\mu+1}\alpha^{l'_{\mu+1}})\}$ intersect on the $(\mu - 1)$-flat $\{(\eta_1\alpha^{l_1} + \eta_2\alpha^{l_2} + \cdots +$

$\eta_\mu \alpha^{l_\mu})\}$. The number of μ-flats in PG$(m, 2^s)$ that intersect on a given $(\mu-1)$-flat in PG$(m, 2^s)$ is

$$\frac{2^{(m-\mu+1)s}-1}{2^s-1} = 1 + 2^s + \cdots + 2^{(m-\mu)s}. \tag{8.47}$$

Every point outside a $(\mu-1)$-flat F is in one and only one of the μ-flats intersecting on F.

Let $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$ be an n-tuple over $GF(2)$, where

$$n = \frac{2^{(m+1)s}-1}{2^s-1} = 1 + 2^s + \cdots + 2^{ms}.$$

Let α be a primitive element in $GF(2^{(m+1)s})$. We may number the components of $\mathbf{v}$ with the first n powers of α as follows: v_i is numbered α^i for $0 \le i < n$. As usual, α^i is called the location number of v_i. Let F be a μ-flat in PG$(m, 2^s)$. The incidence vector for F is an n-tuple over $GF(2)$,

$$\mathbf{v}_F = (v_0, v_1, \cdots, v_{n-1}),$$

whose ith component

$$v_i = \begin{cases} 1 & \text{if } (\alpha^i) \text{ is a point in } F, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 8.5 A (μ, s)th-order binary projective geometry (PG) code of length $n = (2^{(m+1)s}-1)/(2^s-1)$ is defined as the largest cyclic code whose null space contains the incidence vectors of all the μ-flats in PG$(m, 2^s)$.

Let h be a nonnegative integer less than $2^{(m+1)s}-1$ and $h^{(l)}$ be the remainder resulting from dividing $2^l h$ by $2^{(m+1)s}-1$. Clearly, $h^{(0)} = h$. The 2^s-weight of h, $W_{2^s}(h)$, is defined by (8.28). The following theorem characterizes the roots of the generator polynomial of a (μ, s)th-order PG code (the proof is omitted). The proof of this theorem can be found in [27, 35], and [36].

Theorem 8.5 Let α be a primitive element $GF(\alpha^{(m+1)s})$. Let h be a nonnegative integer less than $2^{(m+1)s}-1$. Then, the generator polynomial $\mathbf{g}(X)$ of a (μ, s)th-order PG code of length $n = (2^{(m+1)s}-1)/(2^s-1)$ has α^h as a root if and only if h is divisible by 2^s-1, and

$$0 \le \max_{0 \le l < s} W_{2^s}(h^{(l)}) = j(2^s-1), \tag{8.48}$$

with $0 \le j \le m-\mu$.

EXAMPLE 8.25

Let $m = 2$, $s = 2$, and $\mu = 1$. Consider the $(1, 2)$th-order PG code of length

$$n = \frac{2^{(2+1)\cdot 2}-1}{2^2-1} = 21.$$

Let α be a primitive element of $GF(2^6)$. Let h be a nonnegative integer less than 63. It follows from Theorem 8.5 that the generator polynomial $g(X)$ of the (1, 2)th-order PG code of length 21 has α^h as a root if and only if h is divisible by 3, and

$$0 \leq \max_{0 \leq l < 2} W_{2^2}(h^{(l)}) = 3j,$$

with $0 \leq j \leq 1$. The integers that are divisible by 3 and satisfy the preceding condition are 0, 3, 6, 9, 12, 18, 24, 33, 36, and 48. Thus, $g(X)$ has $\alpha^0 = 1, \alpha^3, \alpha^6, \alpha^9, \alpha^{12}, \alpha^{18}, \alpha^{24}, \alpha^{33}, \alpha^{36}$, and α^{48} as roots. From Table 6.3 we find that (1) $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{33}$, and α^{48} have the same minimal polynomial, $\phi_3(X) = 1 + X + X^2 + X^4 + X^6$; and (2) α^9, α^{18}, and α^{36} have $\phi_9(X) = 1 + X^2 + X^3$ as their minimal polynomial. Thus,

$$\begin{aligned} g(X) &= (1 + X)\phi_3(X)\phi_9(X) \\ &= 1 + X^2 + X^4 + X^6 + X^7 + X^{10}. \end{aligned}$$

Hence, the (1, 2)th-order PG code of length 21 is a (21, 11) cyclic code. It is interesting to note that this code is the (21, 11) difference-set code considered in Example 8.9.

Decoding PG codes is similar to decoding EG codes. Consider the decoding of a (μ, s)th-order PG code of length $n = (2^{(m+1)s} - 1)/(2^s - 1)$. The null space of this code contains the incidence vectors of all the μ-flats of PG$(m, 2^s)$. Let $F^{(\mu-1)}$ be a $(\mu - 1)$-flat in PG$(m, 2^s)$ that contains the point (α^{n-1}). From (8.47) we see that there are

$$J = \frac{2^{(m-\mu+1)s} - 1}{2^s - 1}$$

μ-flats intersecting on the $(\mu - 1)$-flat $F^{(\mu-1)}$. The incidence vectors of these J μ-flats are orthogonal on the digits at the locations corresponding to the points in $F^{(\mu-1)}$. Therefore, the parity-check sums formed from these J incidence vectors are orthogonal on the error digits at the locations corresponding to the points in $F^{(\mu-1)}$. Let $S(F^{(\mu-1)})$ denote the sum of error digits at the locations corresponding to the points in $F^{(\mu-1)}$. Then, we can correctly determine this error sum, $S(F^{(\mu-1)})$, from the J check-sums orthogonal on it provided that there are no more than

$$\left\lfloor \frac{J}{2} \right\rfloor = \left\lfloor \frac{2^{(m-\mu+1)s} - 1}{2(2^s - 1)} \right\rfloor$$

errors in the received vector. In this manner we can determine the error sums, $S(F^{(\mu-1)})$'s, corresponding to all the $(\mu - 1)$-flats that contain the point (α^{n-1}). We then use these error sums to determine the error sums, $S(F^{(\mu-2)})$'s, corresponding to all the $(\mu-2)$-flats that contain (α^{n-1}). We continue this process until the error sums, $S(F^{(1)})$'s, corresponding to all the 1-flats that intersect on (α^{n-1}) are formed. These error sums, $S(F^{(1)})$'s, are orthogonal on the error digit e_{n-1} at the location α^{n-1}. Thus, we can determine the value of e_{n-1}. A total of μ steps of orthogonalization are required to decode e_{n-1}. Because the code is cyclic, we can decode other error

digits in the same manner. Thus, the code is μ-step decodable. At the rth step of orthogonalization with $1 \leq r \leq \mu$, the number of error sums, $S(F^{(\mu-r+1)})$'s, that are orthogonal in the error sum corresponding to a given $(\mu - r)$-flat $F^{(\mu-r)}$ is

$$J_{\mu-r+1} = \frac{2^{(m-\mu+r)s} - 1}{2^s - 1} \geq J.$$

Therefore, at each step of orthogonalization, we can always correctly determine the error sums needed for the next step provided that there are no more than $\lfloor J/2 \rfloor$ errors in the received vector. Thus, the μth-order PG code of length $n = (2^{(m+1)s} - 1)/(2^s - 1)$ is capable of correcting

$$t_{ML} = \left\lfloor \frac{J}{2} \right\rfloor = \left\lfloor \frac{2^{(m-\mu+1)s} - 1}{2(2^s - 1)} \right\rfloor \tag{8.49}$$

or fewer errors with majority-logic decoding. Its minimum distance is at least

$$2t_{ML} + 1 = 2^{(m-\mu)s} + \cdots + 2^s + 2. \tag{8.50}$$

EXAMPLE 8.26

Consider the decoding of the (1, 2)th-order (21, 11) PG code with $m = 2$ and $s = 2$. The null space of this code contains the incidence vectors of all the 1-flats (lines) in PG(2, 2^2). Let α be a primitive element in $GF(2^6)$. The geometry PG(2, 2^2) consists of 21 points, (α^0) to (α^{20}), as given in Example 8.24. Let $\beta = \alpha^{21}$. Then, 0, 1, β, and β^2 form the field $GF(2^2)$.

There are $2^2 + 1 = 5$ lines passing through the point (α^{20}), namely,

$$\{(\eta_1\alpha^0 + \eta_2\alpha^{20})\} = \{(\alpha^0), (\alpha^5), (\alpha^7), (\alpha^{17}), (\alpha^{20})\},$$
$$\{(\eta_1\alpha^1 + \eta_2\alpha^{20})\} = \{(\alpha^1), (\alpha^{11}), (\alpha^{14}), (\alpha^{15}), (\alpha^{20})\},$$
$$\{(\eta_1\alpha^2 + \eta_2\alpha^{20})\} = \{(\alpha^2), (\alpha^3), (\alpha^8), (\alpha^{10}), (\alpha^{20})\},$$
$$\{(\eta_1\alpha^4 + \eta_2\alpha^{20})\} = \{(\alpha^4), (\alpha^6), (\alpha^{16}), (\alpha^{19}), (\alpha^{20})\},$$
$$\{(\eta_1\alpha^9 + \eta_2\alpha^{20})\} = \{(\alpha^9), (\alpha^{12}), (\alpha^{13}), (\alpha^{18}), (\alpha^{20})\}.$$

The incidence vectors of these lines (in polynomial form) are

$$\begin{aligned}
w_1(X) &= 1 + X^5 + X^7 + X^{17} + X^{20}, \\
w_2(X) &= X + X^{11} + X^{14} + X^{15} + X^{20}, \\
w_3(X) &= X^2 + X^3 + X^8 + X^{10} + X^{20}, \\
w_4(X) &= X^4 + X^6 + X^{16} + X^{19} + X^{20}, \\
w_5(X) &= X^9 + X^{12} + X^{13} + X^{18} + X^{20}.
\end{aligned}$$

These vectors are orthogonal on digit position 20. They are exactly the orthogonal vectors for the (21, 11) difference-set code given in Examples 8.9 and 8.10.

TABLE 8.9: PG codes.

m	s	μ	n	k	J	t_{ML}
2	2	1	21	11	5	2
2	3	1	73	45	9	4
3	2	2	85	68	5	2
3	2	1	85	24	21	10
2	4	1	273	191	17	8
4	2	3	341	315	5	2
4	2	2	341	195	21	10
4	2	1	341	45	85	42
3	3	2	585	520	9	4
3	3	1	585	184	73	36
2	5	1	1057	813	33	16
5	2	4	1365	1328	5	2
5	2	3	1365	1063	21	10
5	2	2	1365	483	85	21
5	2	1	1365	76	341	170
2	6	1	4161	3431	65	32
6	2	5	5461	5411	5	2
6	2	4	5461	4900	21	10
6	2	3	5461	3185	85	42
6	2	2	5461	1064	341	170
6	2	1	5461	119	1365	682

For $\mu = 1$, we obtain a class of one-step majority-logic decodable PG codes. For $m = 2$, a $(1, s)$th-order PG code becomes a difference-set code. Thus, the difference-set codes form a subclass of the class of $(1, s)$th-order PG codes. For $s = 1$, a $(1, 1)$th-order PG code becomes a maximum-length code.

There is no simple formula for enumerating the number of parity-check digits for a general (μ, s)th-order PG code. Rather complicated combinatorial expressions for the number of parity-check digits of PG codes can be found in [17] and [18]; however for $\mu = m - 1$, the number of parity-check digits for the $(m - 1, s)$th-order of PG code of length $n = (2^{(m+1)s} - 1)/(2^s - 1)$ is

$$n - k = 1 + \binom{m+1}{m}^s. \tag{8.51}$$

This expression was obtained independently by Goethals Delsarte [35], Smith [19], and MacWilliams and Mann [20]. A list of PG codes is given in Table 8.9.

PG codes were first studied by Rudolph [3] and were later extended and generalized by many others [19, 24, 25, 27, 35, 36].

8.9 REMARKS

In this chapter we considered only finite geometries over Galois field $GF(2^s)$ and the construction of majority-logic decodable codes based on these geometries. Finite geometries over Galois field $GF(p^s)$, where p is a prime, can be constructed in

exactly the same way, simply by replacing 2 with p and $GF(2^s)$ with $GF(p^s)$. Construction of codes based on the flats and points in these finite geometries results in a much larger class of majority-logic decodable codes. Construction of finite geometries over $GF(p^s)$ and their application to the construction of low-density parity-check codes will be discussed in Chapter 17.

PROBLEMS

8.1 Consider the (31, 5) maximum-length code whose parity-check polynomial is $\mathbf{p}(X) = 1 + X^2 + X^5$. Find all the polynomials orthogonal on the digit position X^{30}. Devise both type-I and type-II majority-logic decoders for this code.

8.2 $P = \{0, 2, 3\}$ is a perfect simple difference set. Construct a difference-set code based on this set.

a. What is the length n of this code?

b. Determine its generator polynomial.

c. Find all the polynomials orthogonal on the highest-order digit position X^{n-1}.

d. Construct a type-I majority-logic decoder for this code.

8.3 Example 8.1 shows that the (15, 7) BCH code is one-step majority-logic decodable and is capable of correcting any combination of two or fewer errors. Show that the code is also capable of correcting some error patterns of three errors and some error patterns of four errors. List some of these error patterns.

8.4 Consider an (11, 6) linear code whose parity-check matrix is

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

(This code is not cyclic.)

a. Show that the minimum distance of this code is exactly 4.

b. Let $\mathbf{e} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10})$ be an error vector. Find the syndrome bits in terms of error digits.

c. Construct all possible parity-check sums orthogonal on each message error digit e_i for $i = 5, 6, 7, 8, 9, 10$.

d. Is this code completely orthogonalizable in one step?

8.5 Let $m = 6$. Express the integer 43 in radix-2 form. Find all the nonzero proper descendants of 43.

8.6 Let α be a primitive element of $GF(2^4)$ given by Table 2.8. Apply the affine permutation $Z = \alpha^3 Y + \alpha^{11}$ to the following vector of 16 components:

	Location Numbers															
	α^∞	α^0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
$\mathbf{u} =$	(1	1	1	0	0	1	0	1	0	1	1	0	0	0	0	1)

What is the resultant vector?

8.7 Let $m = 6$. Then, $2^6 - 1$ can be factored as follows: $2^6 - 1 = 7 \times 9$. Let $J = 9$ and $L = 7$. Find the generator polynomial of the type-I DTI code of length 63 and $J = 9$ (use Table 6.2). Find all the polynomials (or vectors) orthogonal on the digit position X^{62} (or α^{62}).

8.8 Find the generator polynomial of the type-I DTI code of length 63 and $J = 7$. Find all the polynomials orthogonal on the digit position X^{62}.

8.9 Show that the all-one vector is not a code vector in a maximum-length code.

8.10 Let $\mathbb{v}(X) = v_0 + v_1 X + \cdots + v_{2^m-2} X^{2^m-2}$ be a nonzero code polynomial in the $(2^m - 1, m)$ maximum-length code whose parity-check polynomial is $\mathbb{p}(X)$. Show that the other $2^m - 2$ nonzero code polynomials are cyclic shifts of $\mathbb{v}(X)$. (*Hint:* Let $\mathbb{v}^{(i)}(X)$ and $\mathbb{v}^{(j)}(X)$ be the ith and jth cyclic shifts of $\mathbb{v}(X)$, respectively, with $0 \le i < j < 2^m - 2$. Show that $\mathbb{v}^{(i)}(X) \neq \mathbb{v}^{(j)}(X)$.)

8.11 Arrange the 2^m code vectors of a maximum-length code as rows of a $2^m \times (2^m - 1)$ array.

- a. Show that each column of this array has 2^{m-1} ones and 2^{m-1} zeros.
- b. Show that the weight of each nonzero code vector is exactly 2^{m-1}.

8.12 Example 8.12 shows that the (15, 5) BCH code is two-step majority-logic decodable and is capable of correcting any combination of three or fewer errors. Devise a type-I majority-logic decoder for this code.

8.13 Show that the extended cyclic Hamming code is invariant under the affine permutations.

8.14 Show that the extended primitive BCH code is invariant under the affine permutations.

8.15 Let $P = \{l_0, l_1, l_2, \ldots, l_{2^s}\}$ be a perfect simple difference set of order 2^s such that

$$0 \le l_0 < l_1 < l_2 < \cdots < l_{2^s} \le 2^s(2^s + 1).$$

Construct a vector of $n = 2^{2s} + 2^s + 1$ components,

$$\mathbb{v} = (v_0, v_1, \ldots, v_{n-1}),$$

whose nonzero components are $v_{l_0}, v_{l_1}, \ldots, v_{l_{2^s}}$; that is,

$$v_{l_0} = v_{l_1} = \cdots = v_{l_{2^s}} = 1.$$

Consider the following $n \times 2n$ matrix:

$$\mathbb{G} = [\mathbb{Q}\ \mathbb{I}_n],$$

where (1) $\mathbb{I}_n$ is an $n \times n$ identity matrix, and (2) $\mathbb{Q}$ is an $n \times n$ matrix whose n rows are $\mathbb{v}$ and $n - 1$ cyclic shifts of $\mathbb{v}$. The code generated by $\mathbb{G}$ is a $(2n, n)$ linear (not cyclic) code whose parity-check matrix is

$$\mathbb{H} = [\mathbb{I}_n\ \mathbb{Q}^T].$$

- a. Show that $J = 2^s + 1$ parity-check sums orthogonal on any message error digit can be formed.
- b. Show that the minimum distance of this code is $d = J + 1 = 2^s + 2$. (This code is a half-rate *quasi-cyclic code* [20].)

8.16 Prove that if J parity-check sums orthogonal on any digit position can be formed for a linear code (cyclic or noncyclic), the minimum distance of the code is at least $J + 1$.

8.17 Consider the Galois field $GF(2^4)$ given by Table 2.8. Let $\beta = \alpha^5$. Then, $\{0, 1, \beta, \beta^2\}$ form the subfield $GF(2^2)$ of $GF(2^4)$. Regard $GF(2^4)$ as the two-dimensional Euclidean geometry over $GF(2^2)$, $\mathrm{EG}(2, 2^2)$. Find all the 1-flats that pass through the point α^7.

8.18 Consider the Galois field $GF(2^6)$ given by Table 6.2. Let $\beta = \alpha^{21}$. Then, $\{0, 1, \beta, \beta^2\}$ form the subfield $GF(2^2)$ of $GF(2^6)$. Regard $GF(2^6)$ as the three-dimensional Euclidean geometry EG$(3, 2^2)$.

a. Find all the 1-flats that pass through the point α^{63}.

b. Find all the 2-flats that intersect on the 1-flat, $\{\alpha^{63} + \eta\alpha\}$, where $\eta \in GF(2^2)$.

8.19 Regard $GF(2^6)$ as the two-dimensional Euclidean geometry EG$(2, 2^3)$. Let $\beta = \alpha^9$. Then, $\{0, 1, \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6\}$ form the subfield $GF(2^3)$ of $GF(2^6)$. Determine all the 1-flats that pass through the point α^{21}.

8.20 Let $m = 2$ and $s = 3$.

a. Determine the 2^3-weight of 47.

b. Determine $\max_{0 \le l < 3} W_{2^3}(47^{(l)})$.

c. Determine all the positive integers h less than 63 such that

$$0 < \max_{0 \le l < 3} W_{2^3}(h^{(l)}) \le 2^3 - 1.$$

8.21 Find the generator polynomial of the first-order cyclic RM code of length $2^5 - 1$. Describe how to decode this code.

8.22 Find the generator polynomial of the third-order cyclic RM code of length $2^6 - 1$. Describe how to decode this code.

8.23 Let $m = 2$ and $s = 3$. Find the generator polynomial of the $(0, 3)$th-order EG code of length $2^{2\times 3} - 1$. This code is one-step majority-logic decodable. Find all the polynomials orthogonal on the digit location α^{62} where α is a primitive element in $GF(2^{2\times 3})$. Design a type-I majority-logic decoder for this code.

8.24 Let $m = 3$ and $s = 2$. Find the generator polynomial of the $(1, 2)$th-order twofold EG code of length $2^{3\times 2} - 1$. Describe how to decode this code.

8.25 Prove that the $(m-2)$th-order cyclic RM code of length $2^m - 1$ is a Hamming code. (*Hint:* Show that its generator polynomial is a primitive polynomial of degree m.)

8.26 Prove that the even-weight codewords of the first-order cyclic RM code of length $2^m - 1$ form the maximum-length code of length $2^m - 1$.

8.27 Let $0 < \mu < m - 1$. Prove that the even-weight codewords of the $(m - \mu - 1)$th-order cyclic RM code of length $2^m - 1$ form the dual of the μth-order RM code of length $2^m - 1$. (*Hint:* Let $\mathbf{g}(X)$ be the generator polynomial of the $(m - \mu - 1)$th-order cyclic RM code C. Show that the set of even-weight codewords of C is a cyclic code generated by $(X + 1)\mathbf{g}(X)$. Show that the dual of the μth-order cyclic RM code is also generated by $(X + 1)\mathbf{g}(X)$.)

8.28 The μth-order cyclic RM code of length $2^m - 1$ has a minimum distance of $d_{\min} = 2^{m-\mu} - 1$. Prove that this RM code is a subcode of the primitive BCH code of length $2^m - 1$ and designed distance $2^{m-\mu} - 1$. (*Hint:* Let $\mathbf{g}(X)_{RM}$ be the generator polynomial of the RM code and let $\mathbf{g}(X)_{BCH}$ be the generator polynomial of the BCH code. Prove that $\mathbf{g}(X)_{BCH}$ is a factor of $\mathbf{g}(X)_{RM}$.)

8.29 Show that extended RM codes are invariant under the affine permutations.

8.30 Let $m = 3$, $s = 2$ and $\mu = 2$. Find the generator polynomial of the $(2, 2)$th-order PG code constructed based on the projective geometry PG$(3, 2^2)$. This code is two-step majority-logic decodable. Find all the orthogonal polynomials at each step of orthogonalization.

8.31 Let $\mathcal{L}$ be a line in the two-dimensional Euclidean geometry EG$(2, 2^s)$ that does not pass through the origin. Let $\mathbf{v}_{\mathcal{L}}$ be the incidence vector of $\mathcal{L}$. For $0 < i \le 2^{2s} - 2$, let $\mathbf{v}_{\mathcal{L}}^{(i)}$ be the ith cyclic shift of $\mathbf{v}_{\mathcal{L}}$. Prove that

$$\mathbf{v}_{\mathcal{L}}^{(i)} \ne \mathbf{v}_{\mathcal{L}}.$$

8.32 Let $\mathcal{L}$ be a line in the two-dimensional projective geometry PG(2, 2^s). Let $\mathbf{v}_{\mathcal{L}}$ be the incidence vector of $\mathcal{L}$. For $0 < i \leq 2^{2s} + 2^s$, let $\mathbf{v}_{\mathcal{L}}^{(i)}$ be the ith cyclic shift of $\mathbf{v}_{\mathcal{L}}$. Prove that

$$\mathbf{v}_{\mathcal{L}}^{(i)} \neq \mathbf{v}_{\mathcal{L}}.$$

8.33 Prove that a cyclic shift of the incidence vector of a μ-flat in EG(m, 2^s) not passing through the origin is the incidence vector of another μ-flat in EG(m, 2^s) not passing through the origin.

8.34 Consider the cyclic product code whose component codes are the (3, 2) cyclic code generated by $g_1(X) = 1 + X$ and the (7, 4) Hamming code generated by $g_2(X) = 1 + X + X^3$. The component code C_1 is completely orthogonalizable in one step, and the component code C_2 is completely orthogonalizable in two steps. Show that the product code is completely orthogonalizable in two steps. (In general, if one component code is completely orthogonalizable in one step, and the other component code is completely orthogonalizable in L steps, the product code is completely orthogonalizable in L steps [37].)

BIBLIOGRAPHY

1. I. S. Reed, "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Trans.*, IT-4: 38–49, September 1954.

2. J. L. Massey, *Threshold Decoding*, MIT Press, Cambridge, 1963.

3. L. D. Rudolph, "A Class of Majority Logic Decodable Codes," *IEEE Trans. Inform. Theory*, IT-13: 305–7, April 1967.

4. T. Kasami, L. Lin, and W. W. Peterson, "Some Results on Cyclic Codes Which Are Invariant under the Affine Group and Their Applications," *Inform. Control*, 2(5 and 6): 475–96, November 1968.

5. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2d ed. MIT Press, Cambridge, 1972.

6. S. Lin and G. Markowsky, "On a Class of One-Step Majority-Logic Decodable Cyclic Codes," *IBM J. Res. Dev.*, January 1980.

7. R. B. Yale, "Error-Correcting Codes and Linear Recurring Sequences," *Lincoln Laboratory Report*, pp. 33–77, Lincoln Labs., MIT, Cambridge, 1958.

8. N. Zierler, "On a Variation of the First-Order Reed-Muller Codes," *Lincoln Laboratory Report*, pp. 38–80, Lincoln Labs., MIT, Cambridge, 1958.

9. J. Singer, "A Theorem in Finite Projective Geometry and Some Applications to Number Theory," *AMS Trans.*, 43: 377–85, 1938.

10. T. A. Evans and H. B. Mann, "On Simple Difference Sets," *Sankhya*, 11: 464–81, 1955.

11. L. D. Rudolph, "Geometric Configuration and Majority Logic Decodable Codes," *M.E.E. thesis*, University of Oklahoma, Norman, 1964.

12. E. J. Weldon, Jr., "Difference-Set Cyclic Codes," *Bell Syst. Tech. J.*, 45: 1045–55, September 1966.

13. F. L. Graham and J. MacWilliams, "On the Number of Parity Checks in Difference-Set Cyclic Codes," *Bell Syst. Tech. J.*, 45: 1046–70, September 1966.

14. R. D. Carmichael, *Introduction to the Theory of Groups of Finite Order*, Dover, New York, 1956.

15. H. B. Mann, *Analysis and Design of Experiments,* Dover, New York, 1949.

16. A. P. Street and D. J. Street, *Combinatorics of Experimental Design,* Oxford Science Publications, Inc., Clarendon Press, Oxford, 1987.

17. N. Hamada, "On the p-rank of the Incidence Matrix of a Balanced or Partially Balanced Incomplete Block Design and Its Applications to Error-Correcting Codes," *Hiroshima Math. J.*, 3: 153–226, 1973.

18. S. Lin, "On the Number of Information Symbols in Polynomial Codes," *IEEE Trans. Inform. Theory*, IT-18: 785–94, November 1972.

19. K. J. C. Smith, "Majority Decodable Codes Derived from Finite Geometries," *Institute of Statistics Mimeo Series*, No. 561, University of North Carolina, Chapel Hill, 1967.

20. F. J. MacWilliams and H. B. Mann, "On the p-rank of the Design Matrix of a Different Set," *Inform. Control*, 12: 474–88, 1968.

21. T. Kasami, S. Lin, and W. W. Peterson, "Linear Codes Which Are Invariant under the Affine Group and Some Results on Minimum Weights in BCH Codes," *Electron. Commun. Jap.*, 50(9): 100–106, September 1967.

22. T. Kasami, S. Lin, and W. W. Peterson, "New Generalizations of the Reed–Muller Codes, Pt. I: Primitive Codes," *IEEE Trans. Inform. Theory*, IT-14: 189–99, March 1968.

23. V. D. Kolesnik and E. T. Mironchikov, "Cyclic Reed–Muller Codes and Their Decoding," *Probl. Inform. Transm.*, no. 4: 15–19, 1968.

24. C. R. P. Hartmann, J. B. Ducey, and L. D. Rudolph, "On the Structure of Generalized Finite Geometry Codes," *IEEE Trans. Inform. Theory*, IT-20(2): 240–52, March 1974.

25. D. K. Chow, "A Geometric Approach to Coding Theory with Applications to Information Retrieval," *CSL Report No. R-368*, University of Illinois, Urbana, 1967.

26. E. J. Weldon, Jr., "Euclidean Geometry Cyclic Codes," *Proc. Symp. Combinatorial Math.*, University of North Carolina, Chapel Hill, April 1967.

27. T. Kasami, S. Lin, and W. W. Peterson, "Polynomial Codes," *IEEE Trans. Inform. Theory,* 14: 807–14, 1968.

28. P. Delsarte, "A Geometric Approach to a Class of Cyclic Codes," *J. Combinatorial Theory*, 6: 340–58, 1969.

29. P. Delsarte, J. M. Goethals, and J. MacWilliams, "On GRM and Related Codes," *Inform. Control*, 16: 403–42, July 1970.

30. S. Lin and K. P. Yiu, "An Improvement to Multifold Euclidean Geometry Codes," *Inform. Control*, 28(3): 221–265 July 1975.

31. E. J. Weldon, Jr., "Some Results on Majority-Logic Decoding," Chap. 8 in *Error-Correcting Codes*, H. Mann, ed., John Wiley, New York, 1968.

32. C. L. Chen, "On Majority-Logic Decoding of Finite Geometry Codes," *IEEE Trans. Inform. Theory*, IT-17(3): 332–36, May 1971.

33. T. Kasami and S. Lin, "On Majority-Logic Decoding for Duals of Primitive Polynomial Codes," *IEEE Trans. Inform. Theory*, IT-17(3): 322–31, May 1971.

34. S. Lin, "Multifold Euclidean Geometry Codes," *IEEE Trans. Inform. Theory*, IT-19(4): 537–48, July 1973.

35. J. M. Goethals and P. Delsarte, "On a Class of Majority-Logic Decodable Codes," *IEEE Trans. Inform. Theory*, IT-14: 182–89, March 1968.

36. E. J. Weldon, Jr., "New Generations of the Reed–Muller Codes, Part II: Nonprimitive Codes," *IEEE Trans. Inform. Theory*, IT-14: 199–205, March 1968.

37. S. Lin and E. J. Weldon, Jr. "Further Results on Cyclic Product Codes," *IEEE Trans. Inform. Theory*, IT-16: 452–59, July 1970.