Binary Linear Block Codes

Content

1. **Block Coding**
2. **Linear Systematic Block Code**
3. **Parity-Check Matrix**
4. **Error Pattern**
5. **Syndrome and Error Detection**
6. **Syndrome Circuit**
7. **Syndrome and Error Pattern**
8. **Standard Array**
9. **Decoding and Correctable Error Patterns**
10. **Syndrome Decoding**

11. **MLD for a BSC Based on a Standard Array**
12. **Hamming Distance**
13. **Minimum Distance of a Block Code**
14. **Weight Distribution**
15. **Error Detection with a Linear Block Code**
16. **Error Correcting Capability**
17. **Bounded Distance Decoding**
18. **Hamming Bound**
19. **Hamming Codes**
20. **Extended Linear Codes**
21. **Shortened Linear Codes**

1. Linear Block Codes

- A message of ***k*** bits is encoded into a codeword (code vector) of *n* bits .

$$\bar{c} = (\underbrace{c_0, c_1, \cdots c_{k-1}}_{\textbf{message}}) \longleftrightarrow \bar{v} = (\underbrace{v_0, v_1, \cdots v_{n-1}}_{\textbf{codeword}})$$

- The 2^k codewords corresponding to the 2^k distinct messages form an (*n*, *k*) block code. For the code to be useful, all the 2^k codewords must be distinct.
- An (*n*, *k*) block code is said to be linear if the vector sum of two codewords is a codeword.
- An (*n*, *k*) linear block code is simply a *k*-dimensional subspace of the vector space V_n of all the binary *n*-tuples .

- An (n, k) linear block code is spanned by k linearly independent vectors, $\overline{g}_0, \overline{g}_1, ..., \overline{g}_{k-1}$. The 2^k codewords are simply the 2^k linear combinations of these k vectors.
- Encoding can be done as follows : The codeword for message $\overline{c} = (c_0, c_1, ..., c_{k-1})$ is

$$\overline{v} = c_0 \overline{g}_0 + c_1 \overline{g}_1 + ... + c_{k-1} \overline{g}_{k-1}$$

$$= (c_0, c_1, \cdots c_{k-1}) \cdot \begin{bmatrix} \overline{g}_0 \\ \overline{g}_1 \\ \vdots \\ \overline{g}_{k-1} \end{bmatrix}$$

- We may arrange the k vectors , $\overline{g}_0, \overline{g}_1, ..., \overline{g}_{k-1}$, as rows of a $k \times n$ matrix,

$$\overline{G} = \begin{bmatrix} \overline{g}_0 \\ \overline{g}_1 \\ \vdots \\ \overline{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \cdots & \cdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

- $\overline{G}$ is called a generator matrix of the code.

- Example 3-1: Let $k = 3$ and $n = 6$. Table 3-1 gives a (6, 3) linear block code.

Table 3-1

Message $(c_0, \ c_1, \ c_2)$	Codeword $(v_0, v_1, v_2, v_3, v_4, v_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(0 1 1 1 0 0)
(0 1 0)	(1 0 1 0 1 0)
(1 1 0)	(1 1 0 1 1 0)
(0 0 1)	(1 1 0 0 0 1)
(1 0 1)	(1 0 1 1 0 1)
(0 1 1)	(0 1 1 0 1 1)
(1 1 1)	(0 0 0 1 1 1)

A generator matrix for this code is

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

The codeword for the message $\overline{c} = (101)$ is

$$\overline{v} = \overline{c} \cdot \overline{G}$$

$$= 1\cdot(011100)+0\cdot(101010)+1\cdot(110001)$$

$$= (011100)+(000000)+(110001)$$

$$= (101101)\ .$$

2. Linear Systematic Block Code

- An (n, k) linear block code is said to be systematic if it has the following structure: Every codeword consists two parts The message part consist of the k unaltered message bits and the parity-check part consists of n - k parity-check bits as shown in Figure 3-1

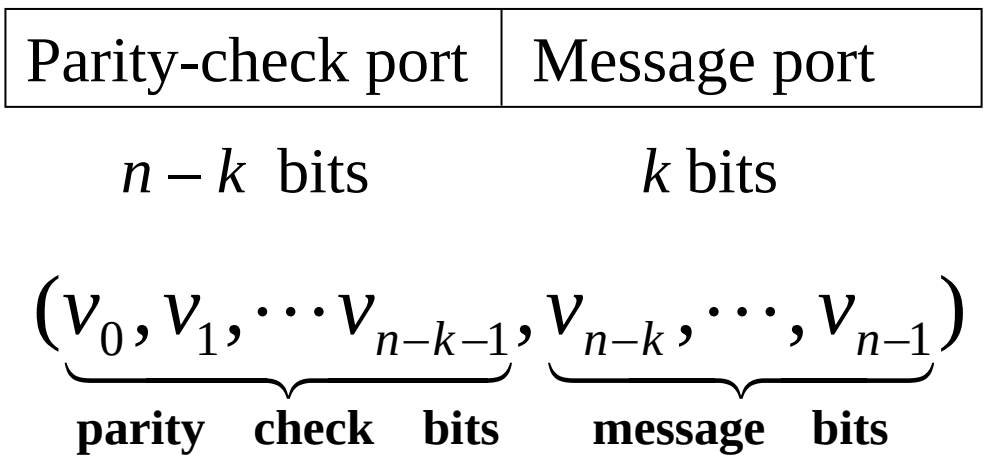


Figure. 3-1 systematic format

- The (6, 3) code given by Table 3-1 is a linear systematic block code.
- An (n, k) linear systematic code is completely specified a $k \times n$ generator matrix of the following form .

$$\overline{G} = \begin{bmatrix} \overline{g}_0 \\ \overline{g}_1 \\ \overline{g}_2 \\ \vdots \\ \overline{g}_{k-1} \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ P_{10} & P_{11} & \cdots & P_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ P_{20} & P_{21} & \cdots & P_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{k-1,0} & P_{k-1,1} & \cdots & P_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

where p_{ij} = 0 or 1 .

- Let $\overline{I_k}$ denote the $k \times k$ identity matrix . Then

$$\overline{G} = \left[\overline{P} \quad \overline{I_k}\right]$$

- Let $\bar{c} = (c_0, c_1, ..., c_{k-1})$ be the message to be encoded .

The corresponding codeword is then ,

$$\bar{v} = (v_0, v_1, \cdots, v_{n-1})$$
$$= \bar{c} \cdot \overline{G} = c_0 \overline{g}_0 + c_1 \overline{g}_1 + ... + c_{k-1} \overline{g}_{k-1}$$

- It is easy to see that

(1) $v_{n-k+i} = c_i$, for $0 \le i \le k$ (3-1)

(2) $v_j = c_0 p_{0,j} + c_1 p_{1,j} + \ldots + c_{k-1} p_{k-1,j}$ (3-2)

for $0 \le j \le n - k - 1$

- We see that the k code bits, v_{n-k} , v_{n-k+1} , … , v_{n-1} are identical to the k message bits. The n - k code bits, v_0 ,v_1,…, v_{n-k-1} are parity-check bits.
- Each parity–check bit is a sum (modulo-2) of some message bits.
- (3-2) gives n - k equations which are called **parity-check equations**. These parity-check equations completely specify the code.
- Example 3-2: Consider the (6, 3) code given in Table 3-1. Its generator matrix in systematic form is

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

Let $\bar{c} = (c_0, c_1, c_2)$ be the message to be encoded. Then the codeword is

$$\bar{v} = (v_0, v_1, v_2, v_3, v_4, v_5) = \bar{c} \cdot \bar{G}$$

We find that

$v_5 = c_2$

$v_4 = c_1$

$v_3 = c_0$

$v_2 = c_0 + c_1$

$v_1 = c_0 + c_2$

$v_0 = c_1 + c_2$

} parity-check equations

The parity-check equations actually tell us how to implement the encoder.

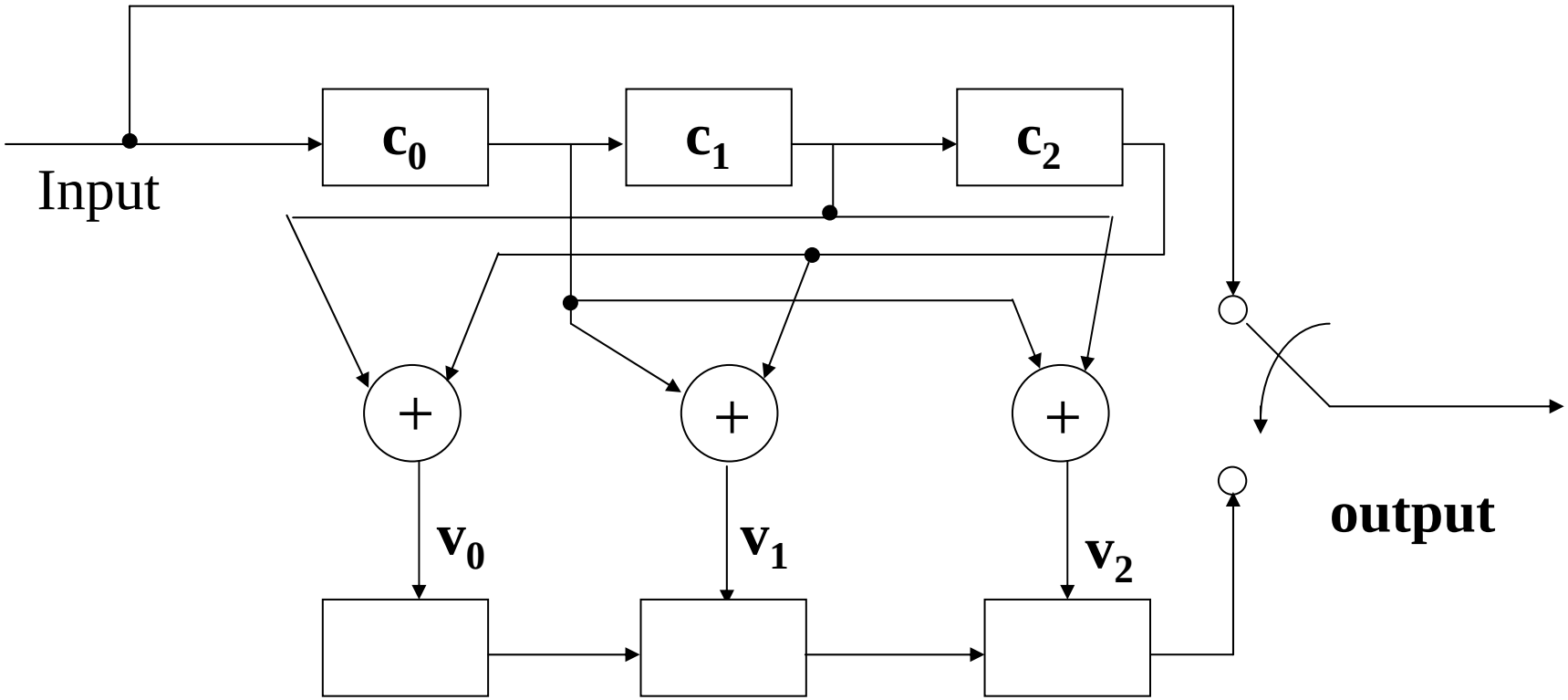


Figure 3-2 A (6, 3) code encoder

3. Parity-Check Matrix

- An (n, k) linear code can also be specified by an $(n - k) \times n$ matrix $\overline{H}$.
- Let $\bar{v} = (v_0, v_1, ..., v_{n-1})$ be an n-tuple. Then it is a codeword if and only if

$$\bar{v} \cdot \overline{H}^T = (0\,0 ... 0)$$

i.e., the inner product of $\bar{v}$ and $\overline{H}$ is zero .

- The matrix $\overline{H}$ is called a parity–check matrix .
- For an (n, k) systematic code with generator matrix $\overline{G} = [\overline{P}\ \ \overline{I}_k]$, the parity-check matrix is

$$\overline{H} = [\bar{I}_{n-k}\ \ \overline{P}^T]$$

where $\bar{I}_{n-k}$ is an $(n\text{-}k) \times (n\text{-}k)$ identity matrix and $\overline{P}^T$ is the transpose of $\overline{P}$

- Parity-check matrix is used for decoding.
- Example 3-3: Consider a (7, 4) linear systematic code with generator matrix

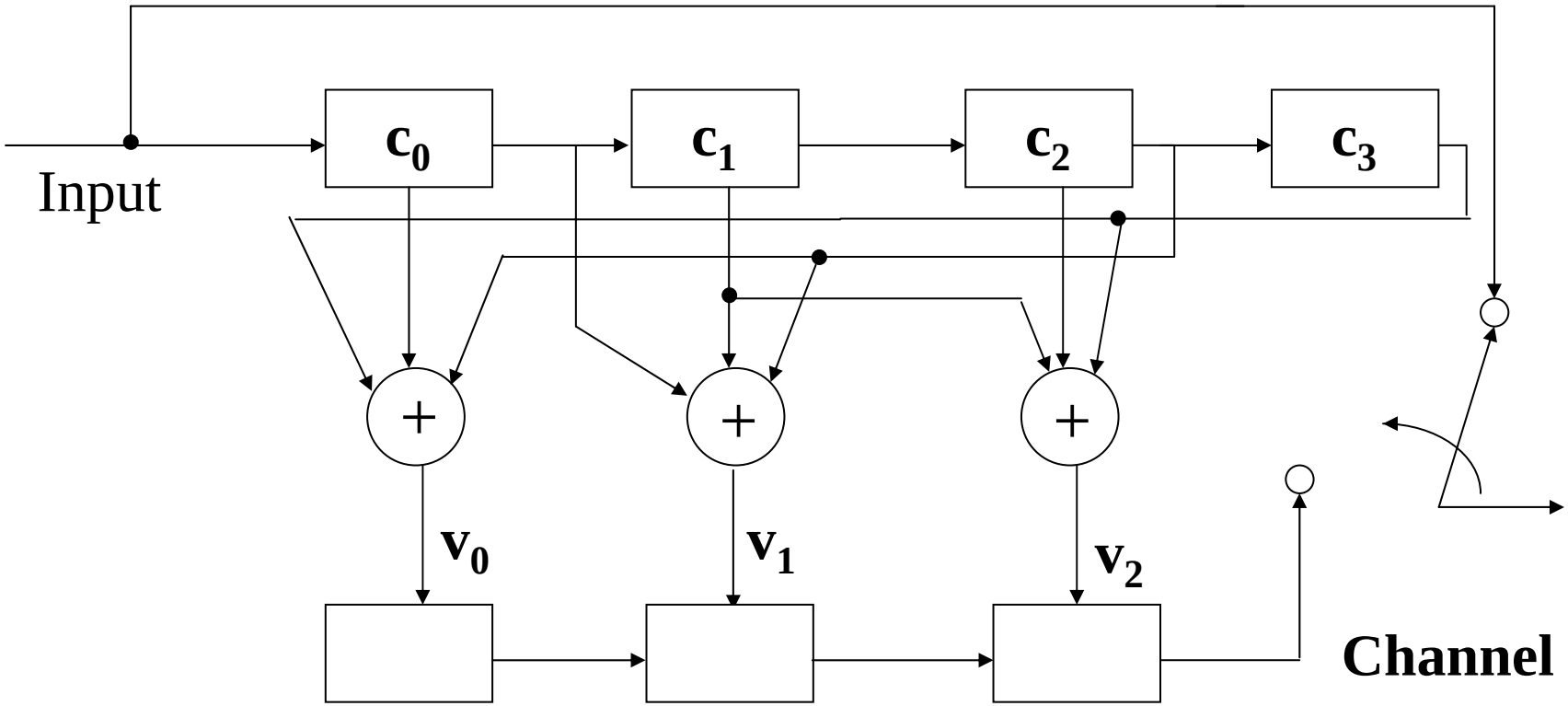
$$\overline{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then the parity-check matrix in systematic form is

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

The parity-check equations are:

$$v_2 = c_1 + c_2 + c_3$$
$$v_1 = c_0 + c_1 + c_2$$
$$v_0 = c_0 + c_2 + c_3$$



Parity register

Figure 3-2.1 A (7, 4) code encoder

4. Error Pattern

- Suppose a codeword $\bar{v} = (v_0, v_1, ..., v_{n-1})$ in a block code C is transmitted .
- Let $\bar{r} = (r_0, r_1, ..., r_{n-1})$ be the corresponding received vector.
- If $r_j \neq v_j$, we say that there is a transmission error at the j-th position of $\bar{v}$.
- The difference between $\bar{r}$ and $\bar{v}$ gives the pattern of errors. This difference is defined as follows:

$$\begin{aligned} \bar{e} &= (r_0, r_1, \ldots, r_{n-1}) + (v_0, v_1, \ldots, v_{n-1}) \\ &= (r_0 + v_0,\ r_1 + v_1, \ldots,\ r_{n-1} + v_{n-1}) \\ &= (e_0, e_1, \ldots, e_{n-1}) \end{aligned}$$

where $r_i + v_i$ is carried out in modulo-2 addition .

- The vector $\bar{e}$ is called an error pattern (or vector) $e_j = 1$ indicates that the j-th position of $\bar{r}$ has an error.
- Obviously, we have $\bar{r} = \bar{v} + \bar{e}$.
- There are a total 2^n possible error patterns. Among these error patterns, only 2^{n-k} of them are correctable by an (n, k) linear code. To minimize the probability of a decoding error, it is desired to design a code which corrects the 2^{n-k} most probable error patterns.

5. Syndrome and Error Detection

- To test whether a received vector $\bar{r}$ contains transmission errors, we compute the following $(n - k)$ tuples

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

- Then $\bar{r}$ is a codeword in code C if and only if $\bar{s} = \bar{0}$
- Hence, if $\bar{s} \neq \bar{0}$, $\bar{r}$ is not a codeword and contains transmission errors. In this case, we say that the presence of errors is being detected.

- If $\bar{s} = \bar{0}$, $\bar{r}$ is a codeword . In this case, $\bar{r}$ is assumed to be error-free and accepted by the receiver. A **decoding error** is committed if $\bar{r}$ is a codeword which is different from the actually transmitted codeword .
- The $(n-k)$ tuples, $\bar{s} = (s_0, s_1, ..., s_{n-k-1})$, is called the **syndrome** of $\bar{r}$.
- Example 3-4: Consider a (7, 4) linear code with parity-check matrix

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

Let $\bar{r} = (0100001)$. The syndrome of $\bar{r}$ is

$$\bar{s} = (s_0, s_1, s_2) = \bar{r} \cdot \bar{\bar{H}}^T$$

$$= (0100001) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$= (1\,1\,1) \neq \bar{0}$$

Hence $\bar{r}$ is not a codeword .

6. Syndrome Circuit

- Consider the (7, 4) code given in Example 3-4 .
- Let $\bar{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector .

  The syndrome of $\bar{r}$ is

$$\bar{s} = (s_0, s_1, s_2) = \bar{r} \cdot \bar{H}^T$$

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

- Then

$$S_0 = r_0 + r_3 + r_5 + r_6$$
$$S_1 = r_1 + r_3 + r_4 + r_5$$
$$S_2 = r_2 + r_4 + r_5 + r_6 \ .$$

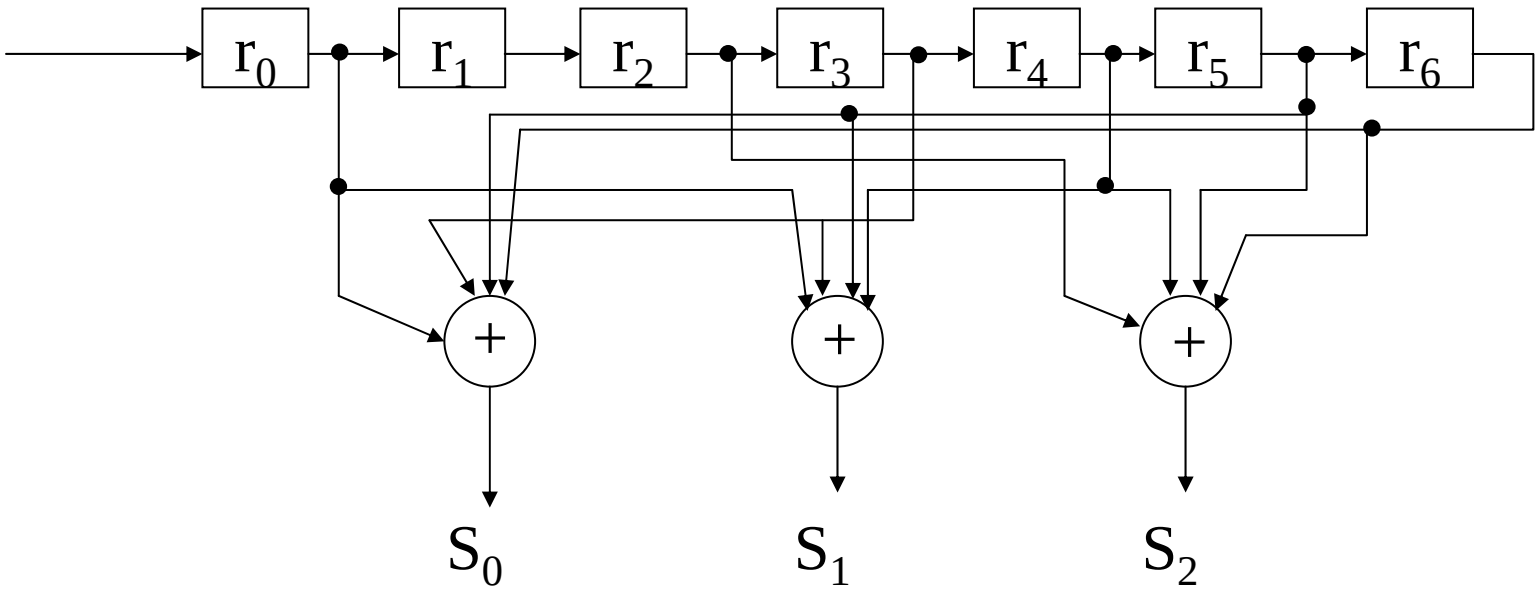


Figure 3-3 syndrome circuit for the (7,4) code given in Example 3-4

7. Syndrome and Error Pattern

- Let $\bar{r} = \bar{v} + \bar{e}$ be the received vector where $\bar{v}$ and $\bar{e}$ are the transmitted codeword and error pattern respectively.
- Then the syndrome of $\bar{r}$ is

$$\begin{aligned} \bar{s} &= \bar{r} \cdot \bar{\bar{H}}^T = (\bar{v} + \bar{e}) \cdot \bar{\bar{H}}^T \\ &= \bar{v} \cdot \bar{\bar{H}}^T + \bar{e} \cdot \bar{\bar{H}}^T = \bar{e} \cdot \bar{\bar{H}}^T \end{aligned} \qquad (3\text{-}6)$$

- (3-6) gives a relationship between the unknown error pattern and the syndrome.
- In fact (3-6) gives the following $n - k$ linear equations:

$$\begin{aligned} s_0 &= e_0 + e_{n-k}p_{0,,0} + e_{n-k+1}p_{0,1} + \cdots + e_{n-1}p_{0,k-1} \\ s_1 &= e_1 + e_{n-k}p_{1,0} + e_{n-k+1}p_{1,,1} + \cdots + e_{n-1}p_{1,k-1} \\ &\vdots \\ s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{n-k-1,0} + e_{n-k+1}p_{n-k-1,1} + \cdots + e_{n-1}p_{n-k-1,k-1} \end{aligned} \qquad (3\text{-}7)$$

- Any method solving these $n - k$ equations is a decoding method.
- Since there are more unknowns than equations, these equations do not have a unique solution. In fact, there are 2^k possible solutions. The true error pattern is just one of them.
- To minimize the probability of a decoding error, the most probable error pattern which satisfies the equations is chosen as the true error pattern.

- Example 3-5: Let

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

Suppose $\bar{v} = (1000001)$ is transmitted and $\bar{r} = (1001001)$ is received. Then the syndrome of $\bar{r}$ is

$$\bar{s} = (s_0, s_1, s_2) = \bar{r} \cdot \overline{\overline{H}}^T = (110)$$

Let $\bar{e} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$ be the error pattern .

Since

$$\bar{s} = \bar{e} \cdot \overline{\overline{H}}^T$$

We have the following 3 equations:

$$1 = e_0 + e_3 + e_5 + e_6$$
$$1 = e_1 + e_3 + e_4 + e_5$$
$$0 = e_2 + e_4 + e_5 + e_6$$

The solutions are:

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

Note the true error pattern,

$$\bar{e} = \bar{r} + \bar{v}$$

$$= (1\ 0\ 0\ 1\ 0\ 0\ 1) + (1\ 0\ 0\ 0\ 0\ 0\ 1)$$

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

is just one of the 16 possible solutions. It is also the most probable solution.

8. Standard Array

- Consider an (n, k) linear code C.
- Let $\bar{v} = \bar{0}, \bar{v}_2, ..., \bar{v}_{2^k}$, be the 2^k codewords in C.
- We form an array with vectors from V_n(the vector space of all binary n-tuples) .
- First we arrange the 2^k codewords from C as the top row of the array with $\bar{e} = \bar{0}$ as the element.

- Suppose we have formed the (j –1)-th row of the array.
- Choose a vector e_j from V_n which is not in the previous j-1 row.
- From the j-th row by adding $\bar{e}_j$ to each codeword in the top row and placing $\bar{e}_j + \bar{v}_i$ under $\bar{v}_i$.
- The array is completed when no vectors can be chosen from V_n.
- This array is called a **standard array**.

$$
\begin{array}{ccccccc}
\bar{v}_1 = \bar{0} & & \bar{v}_2 & \cdots & \bar{v}_i & \cdots & \bar{v}_{2^k} \\
\bar{e}_2 & & \bar{e}_2 + \bar{v}_2 & \cdots & \bar{e}_2 + \bar{v}_i & \cdots & \bar{e}_2 + \bar{v}_{2^k} \\
\bar{e}_3 & & \bar{e}_3 + \bar{v}_2 & \cdots & \bar{e}_3 + \bar{v}_i & \cdots & \bar{e}_3 + \bar{v}_{2^k} \\
\vdots & & & & & & \vdots \\
\bar{e}_{2^{n-k}} & & \bar{e}_{2^{n-k}} + \bar{v}_2 & \cdots & \bar{e}_{2^{n-k}} + \bar{v}_i & \cdots & \bar{e}_{2^{n-k}} + \bar{v}_{2^k}
\end{array}
$$

$$
\begin{array}{cccccc}
\bar{v}_1 = \bar{0} & \bar{v}_2 & \cdots & \bar{v}_i & \cdots & \bar{v}_{2^k} \\
\bar{e}_2 & \bar{e}_2 + \bar{v}_2 & \cdots & \bar{e}_2 + \bar{v}_i & \cdots & \bar{e}_2 + \bar{v}_{2^k} \\
\bar{e}_3 & \bar{e}_3 + \bar{v}_2 & \cdots & \bar{e}_3 + \bar{v}_i & \cdots & \bar{e}_3 + \bar{v}_{2^k} \\
\vdots & & & & & \vdots \\
\bar{e}_{2^{n-k}} & \bar{e}_{2^{n-k}} + \bar{v}_2 & \cdots & \bar{e}_{2^{n-k}} + \bar{v}_i & \cdots & \bar{e}_{2^{n-k}} + \bar{v}_{2^k}
\end{array}
$$

- Each row is called a **coset**.
- There are exactly 2^{n-k} cosets.
- The first element of each coset is called the **coset leader**.
- Every vector in V_n appears one and only once in the array.
- Example 3-6: A standard array for the (6, 3) code given in Example 3-1 is shown below:

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

Properties of a standard array:

- All the 2^k vector in a coset have the same syndrome which is the syndrome of the coset leader.

$$\bar{s} = (\bar{e}_j + \bar{v}_i) \cdot \overline{\overline{H}}^T = \bar{e}_j \cdot \overline{\overline{H}}^T$$

- Different cosets have different syndromes.
- There is one-to-one correspondence between a coset and an ($n - k$)-tuple syndrome. That is, there is a one-to-one correspondence between a coset leader and an ($n - k$)-tuple syndrome.
- The above properties justify our claim that the $n - k$ equations of (3-7) for a given syndrome have 2^k solutions.

9. Decoding and Correctable Error Patterns

- Recall that every column of a standard array consists of one and only one codeword, and all the other vectors are sums of the codeword and the coset leaders. The *j*-th column is

$$D_j = \{\bar{v}_j, \bar{e}_2 + \bar{v}_j, \bar{e}_3 + \bar{v}_j, ..., \bar{e}_{2^{n-k}} + \bar{v}_j\}$$

- The 2^k columns can be used as the decoding regions.
- Let $\bar{r}$ be the received vector. If $\bar{r}$ is found in the *j*-th column D_j, then is decoded into the codeword $\bar{v}_j$.
- To minimize the probability of a decoding error, the error patterns that are most likely to occur for a given channel should be chosen as the coset leaders.

10. Syndrome Decoding

- The decoding in the previous section can be simplified by using the one-to-one relationship between an ($n - k$) tuple syndrome and a coset leader (correctable error pattern).
- Suppose a codeword is transmitted and is the error pattern. Then the received vector is

$$\bar{r} = \bar{v} + \bar{e}$$

- At the receiving end, if we can estimate $\bar{e}$, then the transmitted codeword is obtained by adding $\bar{e}$ to $\bar{r}$

$$\bar{v} = \bar{r} + \bar{e}$$

- As a result, decoding can be done in 3 steps:

  (1) Compute the syndrome of $\bar{r}$, i.e.

$$\bar{s} = \bar{r} \cdot \bar{\bar{H}}^T$$

  (2) Find the coset leader $\bar{e}$ whose syndrome is equal to $\bar{s}$. Then $\bar{e}$ is assumed to be the error pattern caused by the channel.

  (3) Decoding the received vector $\bar{r}$ into the codeword

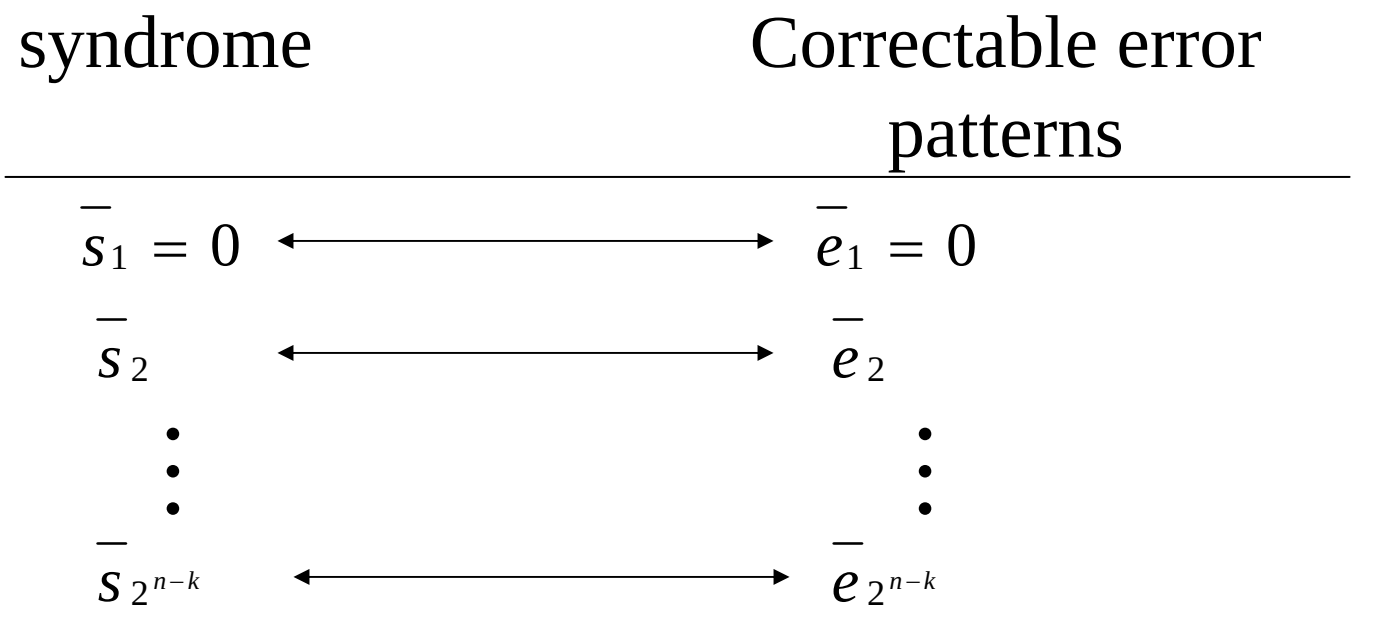
$$\bar{v} = \bar{r} + \bar{e}$$

- This decoding process is called the syndrome decoding.

.

Table–look–up Implementation

- Syndrome decoding can be done by using a table which consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes,



- This can be implemented with a ROM or a combinational logic circuit (CLC).
- For CLC implementation, each error bit is regarded as a switching function of the syndrome variables $s_0, s_1, ..., s_{n-k-1}$

$$e_i = f(s_0, s_1, ..., s_{n-k-1})$$

- For large n - k , the decoder would become very complex .

- **Example 3-7**: Consider a (6, 3) linear systematic code generated by

$$\overline{G} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \overline{P} & \overline{I}_3 \end{bmatrix}$$

Its parity-check matrix is

$$\overline{H} = \begin{bmatrix} \overline{I}_3 & \overline{P}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Encoding

$$(c_0, c_1, c_2) \longleftrightarrow (v_0, v_1, v_2, c_0, c_1, c_2)$$

Where

$$v_0 = c_1 + c_2$$
$$v_1 = c_0 + c_2$$
$$v_2 = c_0 + c_1$$

An encoding circuit is shown in Figure 3-2.

Syndrome look-up table

Syndrome $(s_0,\ s_1,\ s_2)$	Correctable error patterns $(e_0, e_1, e_2, e_3, e_4, e_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(1 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0)
(0 1 1)	(0 0 0 1 0 0)
(1 0 1)	(0 0 0 0 1 0)
(1 1 0)	(0 0 0 0 0 1)
(1 1 1)	(1 0 0 1 0 0)

Combinational logic circuit implementation

$$e_0 = s_0 \cap \bar{s}_1 \cap \bar{s}_2 + s_0 \cap s_1 \cap s_2$$

$$e_1 = \bar{s}_0 \cap s_1 \cap \bar{s}_2$$

$$e_2 = \bar{s}_0 \cap \bar{s}_1 \cap s_2$$

$$e_3 = \bar{s}_0 \cap s_1 \cap s_2 + s_0 \cap s_1 \cap s_2$$

$$e_4 = s_0 \cap \bar{s}_1 \cap s_2$$

$$e_5 = s_0 \cap s_1 \cap \bar{s}_2$$

A complete decoder is shown in Figure 3-4.

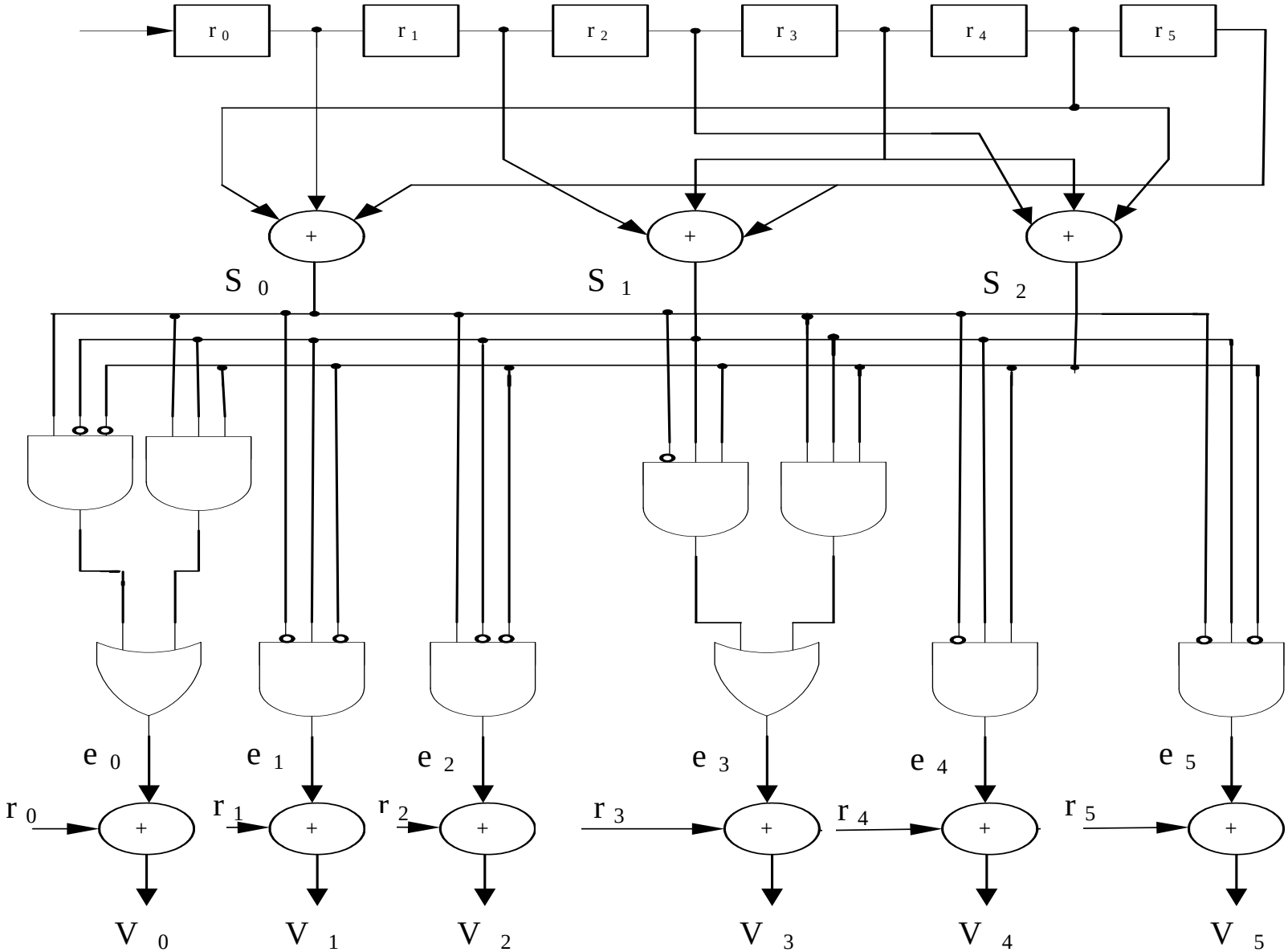

r 0
r 1
r 2
r 3
r 4
r 5
+
+
+
S 0
S 1
S 2
e 0
e 1
e 2
e 3
e 4
e 5
r 0
r 1
r 2
r 3
r 4
r 5
+
+
+
+
+
+
V 0
V 1
V 2
V 3
V 4
V 5

Figure 3-4

HW #3

1. Consider a systematic (8, 4) code whose parity-check equations are

$$v_0 = u_1 + u_2 + u_3$$
$$v_1 = u_0 + u_1 + u_2$$
$$v_2 = u_0 + u_1 + u_3$$
$$v_3 = u_0 + u_2 + u_3$$

   construct an encoder for the code.

2. Construct a syndrome circuit for the code given in Problem 1.

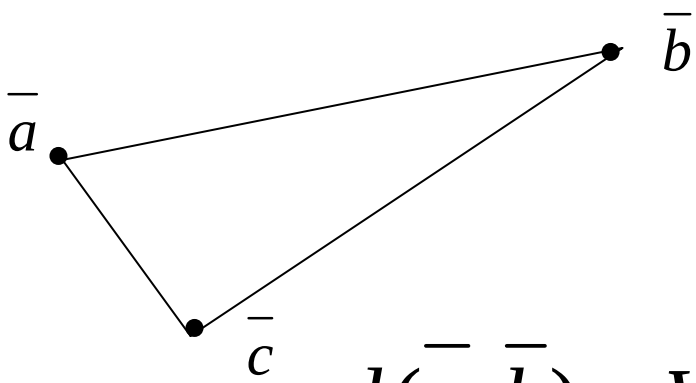
11. MLD for a BSC Based on a Standard Array

- To minimize the probability of a decoding error, the error patterns that are most likely to occur for a given channel should be chosen as the coset leaders (correctable error patterns).
- The Hamming weight of a binary n-tuple is defined as the number of ones in it. For example, the Hamming weight of $\underline{v} = (101100101)$ is 5, i.e. $W(\underline{v}) = 5$.
- In a BSC, an error pattern of smaller weight is more probable than an error pattern of larger weight.
- When a standard array is formed, each coset leader should be chosen to be a vector of least weight from the remaining available vector.

- Choosing coset leader in this manner, each coset leader has minimum weight in each coset.
- As a result, the syndrome decoding is the MLD for a BSC.

12. Hamming Distance

- The Hamming distance between two binary n-tuples $\bar{a}$ and $\bar{b}$, denoted $d(\bar{a},\bar{b})$, is defined as the number of places where $\bar{a}$ and $\bar{b}$ differ.
- **Example 3-8:** Let $\bar{a} = (1001011)$ and $\bar{b} = (0100011)$. Then, $d(\bar{a},\bar{b}) = 3$.
- Triangle inequality: $d(\bar{a},\bar{c}) + d(\bar{c},\bar{b}) \geq d(\bar{a},\bar{b})$



$$d(\bar{a},\bar{b}) = W(\bar{a} + \bar{b})$$

- For example, let $\bar{a} = (1001011)$ and $\bar{b} = (0100011)$. Then $\bar{a} + \bar{b} = (1101000)$. We see that

$$d(\bar{a}, \bar{b}) = W(\bar{a} + \bar{b}) = 3$$

13. Minimum Distance of a Block Code

- Let C be a linear block code. The minimum distance of C, denote d_{min} , is definer as follows:

$$d_{\min} \stackrel{\Delta}{=} \min\{d(\bar{v}, \bar{w}) : \bar{v}, \bar{w} \in C, \bar{v} \neq \bar{w}\}$$

- The minimum weight of C , denoted W_{min} , is defined as follows:

$$w_{\min} \stackrel{\Delta}{=} \min\{w(\bar{v}) : \bar{v} \in C, \bar{v} \neq 0\}$$

- Note that

$$
\begin{aligned}
d_{\min} &= \min\{d(\bar{v}, \bar{w}) : \bar{v}, \bar{w} \in C, \bar{v} \neq \bar{w}\} \\
&= \min\{d(\bar{v} + \bar{w}); \bar{v}, \bar{w} \in C, \bar{v} \neq \bar{w}\} \\
&= \min\{w(\bar{x}) : \bar{x} \in C, \bar{x} \neq \bar{0}\} \\
&= w_{\min}
\end{aligned}
$$

- How to determine the Hamming distance in a block code.
- Let C be an (n, k) linear code with parity-check matrix $\overline{H}$.
- For each code vector of Hamming weight l, there exist l columns of $\overline{H}$ such that sum of these l columns is equal to the zero vector.
- Conversely, if there exist l columns of $\overline{H}$ whose vector sum is the zero vector, there exists a code vector of Hamming weight l in C.
- Example 3-9 : in example 3-4: no two or fewer columns of sum to 0. The 0^{th}, 2^{nd} and 6^{th} columns sum to 0. Thus the $l = 3$. The minimum distance of the (6, 3) linear code is 3

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

14. Weight Distribution

- Let C be an (n, k) linear code.
- Let A_i be the number of codewords in C with (Hamming) weight i.
- Then the set, $\{A_0, A_1, A_2, ..., A_n\}$, is called the weight distribution (or spectrum) of C.
- $A_0 = 1$ and $A_0 + A_1 + \ldots + A_n = 2^k$
- The weight distribution of the (6, 3) linear code given in Example 3-1 is $A_0 = 1$, $A_1 = 0$, $A_2 = 0$, $A_3 = 4$, $A_4 = 3$, $A_5 = 0$, $A_6 = 0$.

15. Error Detection with a Linear Block Code

- Consider an (n, k) linear code C with minimum distance d_{min} .
- Suppose a codeword $\bar{v}$ is transmitted and an non-zero error pattern $\bar{e}$ is added to $\bar{v}$ during the transmission. Then the received sequence is $\bar{r} = \bar{v} + \bar{e}$
- If $\bar{r} = \bar{v} + \bar{e}$ is not a codeword, then its syndrome

$$\bar{s} = \bar{r} \bullet \bar{\bar{H}}^T \neq \bar{0}$$

In this case, the existence of errors in $\bar{r}$ is detected. The error pattern is then called a detectable error pattern.

- However, if $\bar{r} = \bar{v} + \bar{e}$ happens to be a codeword, then the syndrome of $\bar{r}$ is zero. In this case, $\bar{r}$ is assumed to be error-free and accepted by the receiver. A decoding error is committed. Since the existence of errors in $\bar{r}$ is not detected, the error pattern $\bar{e}$ is called an **undetectable** error pattern.
- Due to linear structure of the code, any error pattern which is identical to an nonzero codeword is an undetectable error pattern. Any error pattern which is not identical to an nonzero codeword is detectable.
- There are 2^k-1 undetectable error pattern and $2^n - 2^k + 1$ detectable error patterns.

Probability of an undetectable error

- Let $\{A_i : 0 \le i \le n\}$ be the weight distribution of C .
- The probability of an undetected error for C is

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

- Note that P_{ud} (E) depends on the weight distribution of the code .
- The probability P_{ud} (E) can also be computed from the weight distribution of the dual code $C^{\perp}$ of C . Let $\{B_i : 0 \le i \le n\}$ be the weight distribution of $C^{\perp}$. Then

$$P_{ud}(E) = 2^{-(n-k)} \sum_{i=0}^{n} B_i (1-2p)^i - (1-p)^n$$

- It has been proved that there exist (n, k) linear block codes with

$$P_{ud}(E) \leq 2^{-(n-k)}$$

- A code is said to be a **good error-detecting code** if the above bound holds.

Error detecting capability

- Since the minimum distance of the code is d_{min}, two codewords in C differ at least d_{min} places .
- As a result , no error pattern with weight $d_{min} - 1$ or less (i.e. , $d_{min} - 1$ or fewer errors) will change a transmitted codeword into another codeword .

- There exists at least one error pattern with d_{min} errors which is not detectable.
- The parameter $d_{min} - 1$ is called the **random error-detecting capability** of the code.

16. Error Correcting Capability

- We have shown that, with syndrome decoding, a linear code is capable of correcting 2^{n-k} error patterns.
- These error patterns are coset leaders.
- To achieve MLD, each coset leader must have the smallest weight in coset which contains it. That is, the more probable error patterns should be chosen as the coset leaders.
- For a code with minimum distance d_{min}, we want to know what kind of error patterns can be used as coset leaders.

- It is possible to show that all the error patterns of weight $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or less (i.e., t or fewer errors) can be used as coset leaders.
- Therefore, if a codeword is transmitted and there are t or fewer transmission errors, the received vector will be decoded into the transmitted codeword based on the syndrome decoding (i.e., minimum distance decoding). Errors are hence corrected.
- However, there exists at last one error pattern with t+1 errors can not be used as a coset leader and hence is not correctable. When this error pattern occurs, an incorrect decoding will be made.

- This is to say that all the error patterns of t or fewer errors are guaranteed to be correctable. No such guarantee can be made for the other error patterns.
- For this reason, the parameter $t = \lfloor (d_{\min} - 1)/2 \rfloor$ is called the **random error correcting capability** of the code. The code is called a t-error correcting code.
- The number of guaranteed correctable error pattern is

$$N_t = \sum_{i=0}^{t} \binom{n}{i}$$

- In general, N_t is a small fraction of the 2^{n-k} correctable error patterns.

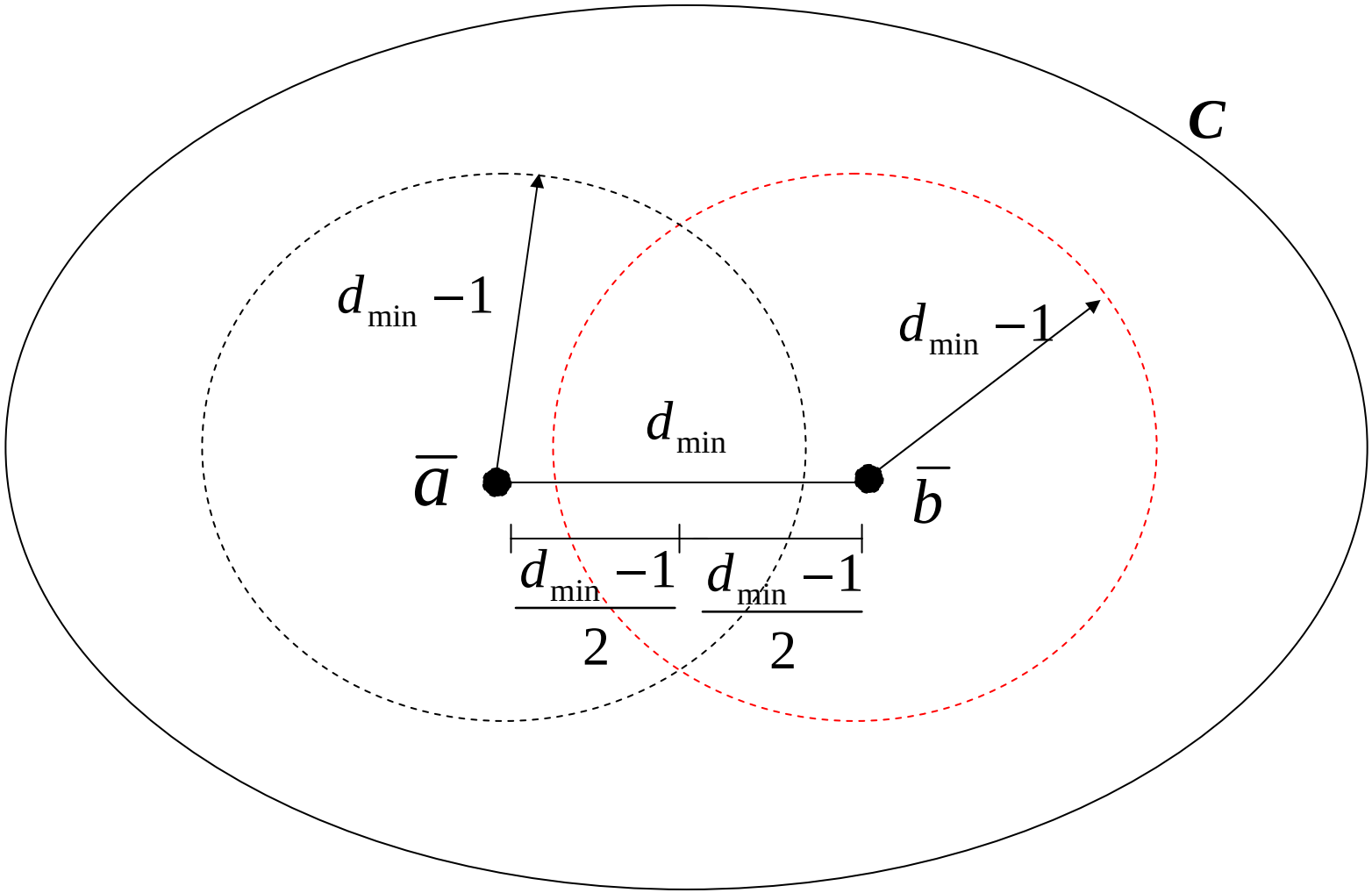
- A code is said to be perfect if $N_t = 2^{n-k}$. There are not too many perfect codes .

Probability of an erroneous decoding

- An upper bound

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

Example 3-10: Consider the (6, 3) code given in Example 3-1. Its minimum distance $d_{min} = 3$. Hence its error correcting capability is $t = \lfloor (3-1)/2 \rfloor = 1$. The code is capable of correcting any error pattern with single error. From its standard array (Table 3 –2), we see that the code is also capable of correcting an error pattern of double errors.


C
$d_{\min} - 1$
$d_{\min} - 1$
$d_{\min}$
$\overline{a}$
$\overline{b}$
$\frac{d_{\min} - 1}{2}$
$\frac{d_{\min} - 1}{2}$

17. Bounded Distance Decoding

- For large n - k, a decoder for correcting all the 2^{n-k} correctable error patterns is very complex and expensive.
- To simplify the decoding complexity, we may design a decoder which only corrects the error patterns which are guaranteed by the error-correcting capability t of the code, and raises flag to other detected but uncorrected error patterns. This kind of decoding is called the bounded distance decoding.
- Probability of a decoding error

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

- Decoded bit-error probability

$$P_b \le \frac{1}{n} \sum_{j=t+1}^{n} (j+t) \binom{n}{j} p^j (1-p)^{n-j}$$

- For large signal-to-noise rations,

$$P_b \approx \frac{d_{\min}}{n} P(E)$$

- P_b is normally called decoding bit-error-rate (BER).

18. Hamming Bound

- For given n and k, it is desired to construct an (n, k) code with minimum distance d_{min} as large as possible.
- Let t_0 be the **largest** integer which satisfies the following inequality,

$$n - k \geq \log_2[1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t_0}]$$

- The error-correcting capability $t = \lfloor (d_{\min} - 1)/2 \rfloor$ of an (n, k) code is **at most** equal to $t_0, i.e., t \leq t_0$
- Hence $d_{\min} \leq 2t_0 + 2$. This is an **upper** bound on the minimum distance of an (n , k) linear code.

19. Hamming Codes

- First class of codes devised for error correction.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters:

Code length:	$n = 2^m - 1$
Dimension:	$k = 2^m - m - 1$
Number of parity-check symbols:	$n - k = m$
Error correcting capability:	$t = 1$
Minimum distance:	$d_{min} = 3$

- The parity-check matrix in systematic form is given as follows:

$$\overline{H} = [\overline{I}_m \quad \overline{P}^T]$$

where $\overline{I}_m$ is an $m \times m$ identity matrix and the submatrix $\overline{P}^T$ consists of $2^m - m - 1$ columns which are m-tuples of weight 2 or more.

- The generator matrix is

$$\overline{G} = [\overline{P} \quad \overline{I}_k]$$

- It corrects all the error patterns with a single error and no others.
- They are widely used for error control.

- The weight distribution is known . The number of codewords of weight i, A_i is simply the coefficient of z^i in the expansion of the following polynomial.

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

which is called the **weight enumerator**.

- The dual code is an $(2^m - 1, m)$ linear code with weight enumerator,

$$B(z) = 1 + (2^m - 1)z^{2^{m-1}}$$

- For error detection , the probability of an undetected error is

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

- For error correction, the probability of a decoding error is

$$P(E) = \sum_{j=2}^{n} \binom{n}{j} p^j (1 - p)^{n-j}$$

Example 3-11: Let $m = 3$. There is a Hamming code of length $n = 2^3-1 = 7$ and dimension $k = 2^3-3-1 = 4$ whose parity-check matrix is given as follows:

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

Its generator matrix is

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

- It is a (7, 4) linear block code which is the same code given in Example 3-3 .
- Parity-check equations: message $\bar{u} = (u_0, u_1, u_2)$

$$v_0 = u_0 + u_2 + u_3 ,$$

$$v_1 = u_0 + u_1 + u_2 ,$$

$$v_2 = u_1 + u_2 + u_3 .$$

- Look-up decoding table

Syndromes (s_0, s_1, s_2)	Correctable Error Patterns $(e_0, e_1, e_2, e_3, e_4, e_5, e_6)$
(0 0 0)	(0 0 0 0 0 0 0)
(1 0 0)	(1 0 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0 0)
(1 1 0)	(0 0 0 1 0 0 0)
(0 1 1)	(0 0 0 0 1 0 0)
(1 1 1)	(0 0 0 0 0 1 0)
(1 0 1)	(0 0 0 0 0 0 1)

Syndromes (s_0, s_1, s_2)	Correctable Error Patterns $(e_0, e_1, e_2, e_3, e_4, e_5, e_6)$
(0 0 0)	(0 0 0 0 0 0 0)
(1 0 0)	(1 0 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0 0)
(1 1 0)	(0 0 0 1 0 0 0)
(0 1 1)	(0 0 0 0 1 0 0)
(1 1 1)	(0 0 0 0 0 1 0)
(1 0 1)	(0 0 0 0 0 0 1)

- Logic functions for the 7 error digits are:

$$e_0 = s_0 \cap \bar{s}_1 \cap \bar{s}_2 \qquad e_1 = \bar{s}_0 \cap s_1 \cap \bar{s}_2$$

$$e_2 = \bar{s}_0 \cap \bar{s}_1 \cap s_2 \qquad e_3 = s_0 \cap s_1 \cap \bar{s}_2$$

$$e_4 = \bar{s}_0 \cap s_1 \cap s_2 \qquad e_5 = s_0 \cap s_1 \cap s_2$$

$$e_6 = s_0 \cap \bar{s}_1 \cap s_2$$

A complete decoder

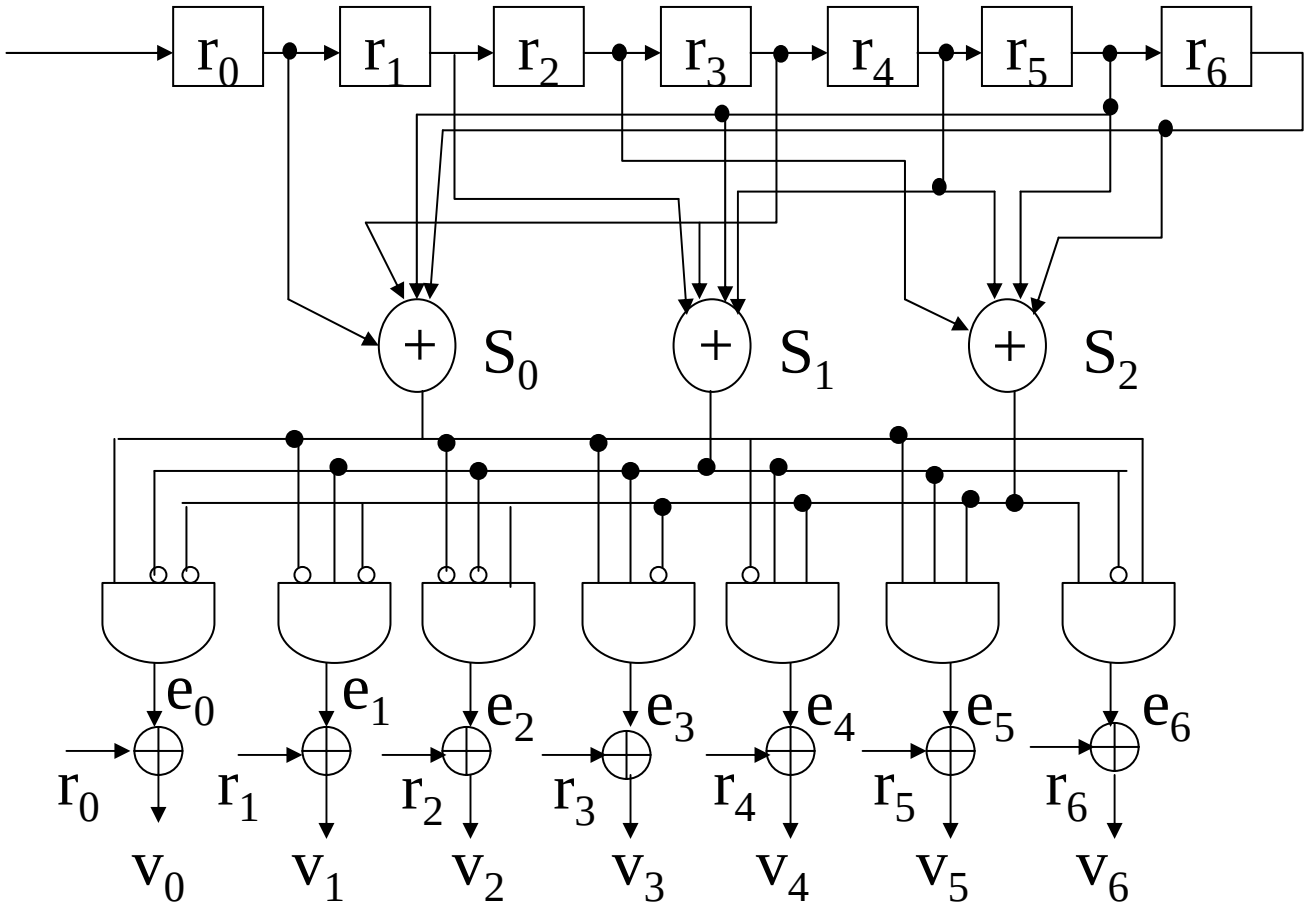


Figure 3-5 A complete decoder for (7,4) Hamming code

20. Extended Linear Codes

- Let C be an (n, k) linear code with both odd and even weight codewords.
- Then C can be extended by adding an overall parity check bit , denoted v_∞, to the left of each codeword $\bar{v} = (v_0, v_1, ..., v_{n-1})$ in C, where

$$v_\infty = v_0 + v_1 + ... + v_{n-1}$$

- The extended codeword is then

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

- Note that $v_\infty = 0$ if the weight of $\bar{v}$ is even and $v_\infty = 1$ if the weight of $\bar{v}$ is odd

- The resulting code, denoted C_e, is called an extension C. C_e is an $(n + 1, k)$ code and has only even weight codewords.
- If the minimum distance d_{min} of C is odd, then the minimum distance of C_e is $d_{min} + 1$ (even).
- Let $\overline{H}$ be the parity-check matrix of C. Then the parity check matrix of C_e is

$$\overline{H}_e = \left[\begin{array}{c|ccc} 1 & 1 & \cdots & 1 \\ \hline 0 & & & \\ \vdots & & \overline{H} & \\ 0 & & & \end{array}\right]$$

- An extended Hamming code has the following parameters:

$$n = 2^m$$

$$k = 2^m + m + 1$$

$$n - k = m + 1$$

$$d_{min} = 4$$

- This is called a distance-4 Hamming code.

21. Shortened Linear Codes

- Let C be an (n, k) linear code with parity-check matrix $\overline{H} = [\overline{I}_{n-k} \quad \overline{P}^T]$ and minimum distance d_{min}.
- Let $\overline{P}_S^T$ denote the matrix obtained by deleting λ columns from $\overline{P}^T$.
- Then the code with

$$\overline{H}_S = [\overline{I}_{n-k} \quad \overline{P}_s^T]$$

  as the parity-check matrix is an $(n - \lambda, k - \lambda)$ linear code.
- The code, denoted C_s, is called a shortened code of C.
- The minimum distance of C_s is at least d_{min}.
- Often C_s is obtained deleting the right-most columns from $\overline{P}^T$.

Distance-4 Hamming Codes

- Consider a hamming code of length $n = 2^m - 1$ with parity-check matrix

$$\overline{H} = [\overline{I}_m \quad \overline{P}^T]$$

where $\overline{P}^T$ consists all the m-tuple of weight 2 or more as columns.

- Suppose we delete all the columns of even weight from $\overline{P}^T$ This results in a $m \times 2^{m-1}$ matrix

$$\overline{H} = [\overline{I}_m \quad \overline{P}^T]$$

- The shortened code C_s with $\overline{H}_S$ as the parity-check matrix is a $(2^{m-1}, m)$ linear code with minimum distance $d_{min} = 4$
- C_s is called a distance-4 shortened Hamming code.

Example 3-12: Let $m = 4$. The (15, 11) Hamming code has the following parity-check matrix:

$$\overline{H} = \underbrace{\begin{bmatrix} 100000011101111 \\ 010001100110111 \\ 001010101011011 \\ 000111010011101 \end{bmatrix}}_{\overline{I}_4 \qquad\qquad \overline{P}^T}$$

- Deleting columns of even weight from $\overline{P}^T$, we have the following matrix

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

- $\overline{H}_s$ gives a (8, 4) shortened Hamming code.

- Another shortening:

A code is shortened by deleting several messages coordinates from the encoding process. In other words, for some shortened message coordinates, we delete their corresponding columns and rows in the generator matrix G. For example: the generator of the (8,4) linear code is

$$G = \begin{bmatrix} 11111111 \\ 01011100 \\ 00101110 \\ 00010111 \end{bmatrix}$$

If we like to delete the first bit in the codeword $\bar{v}$, then the new generator of this shortened code is shown in the following

$$G = \begin{bmatrix} 01011100 \\ 00101110 \\ 00010111 \end{bmatrix}$$

HW #4

1. In problem #1 in HW#3, show that the code has the minimum Hamming distance 4.
2. Find out all syndrome patterns in the above problem.
3. Determine the weight distribution of (8,4) linear code (mentioned in problem #1 in HW#3). Let the transition probability of a BSC be $p = 10^{-2}$. Compute the probability of an undetected error of this code.

22. Error Correction Performance and MATLAB Example

- In following figure, the comparison of error correction performance of a shorten Hamming code is shown.
- This shortened Hamming code with length 21, dimension 16, and minimum Hamming distance 3 is illustrated in MATLAB for error correcting. Each Chinese character is constituted by 2 bytes (8-bit). This shortened Hamming code is shorten from the (31, 26, 3) Hamming code.

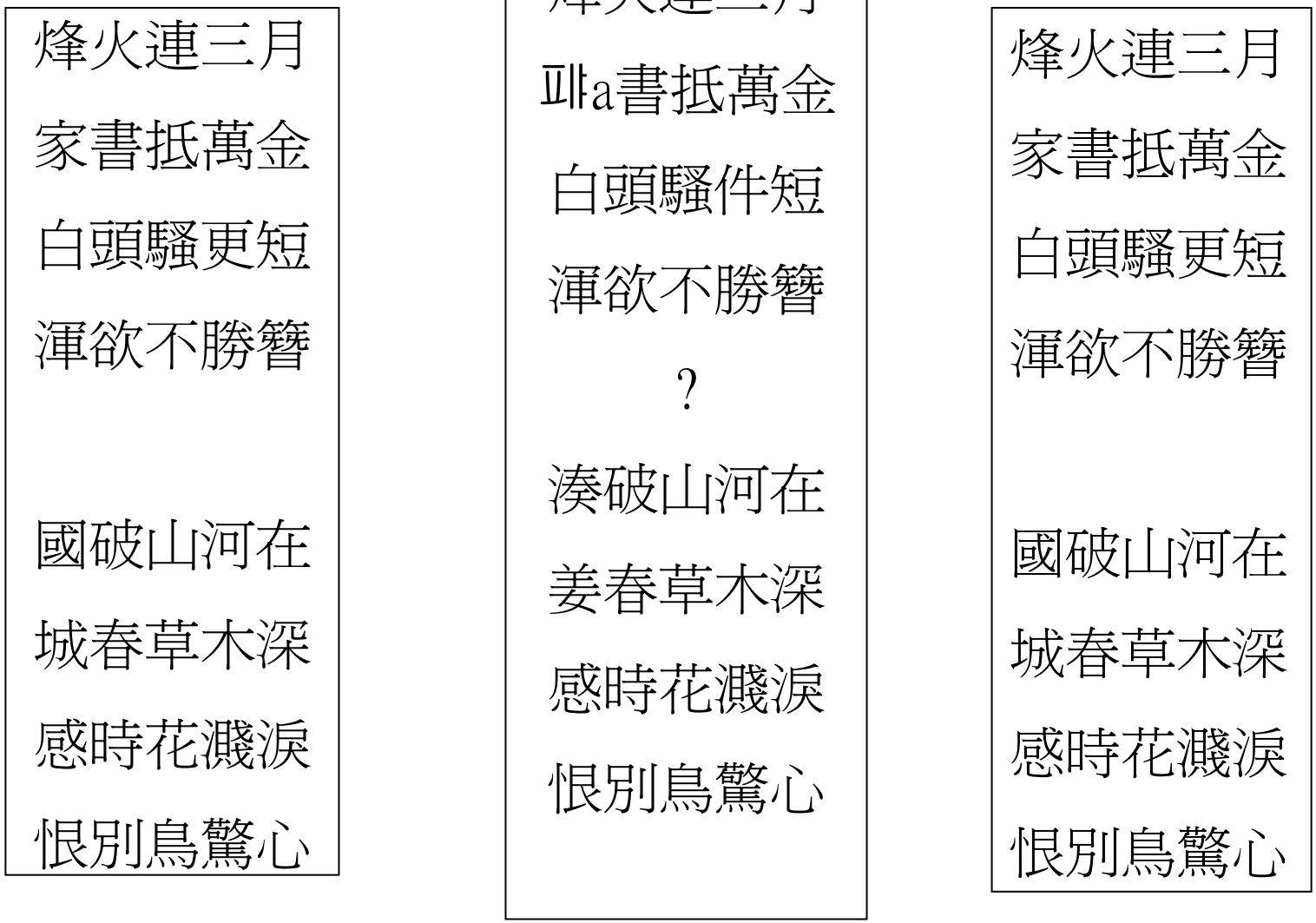


Figure 3-6: the original Chinese poem (left), degrade by AWGN (middle), recovered with (21, 16, 3) Hamming encoding/decoding (right)

```
% to demo performance of (21, 16, 3)Hamming
code, which shortened from (31, 26, 3) Hamming
code.|<--10 bits for parity check -->|<--16
bits for info-->|<--5 zero bits padded-->|

clear

fid = fopen('杜甫詩.txt','r');

A = fread(fid); % A is an array of integers

S = char(A'); % to chinese character

SNR = 6; % 6dB = SNR = 10log(Eb/No) =
         10log(signal_pw/(code_rate*2*
         noise_var)) ,assume signal_pw = 1

noise_var = 1/(2*code_rate*10^(SNR/10));

D = max(size(A));  % D must be even

tt = 1;   % correct 1 bit errors
```

```
Dmin = 2*tt+1; % minimum Hamming ditance
for i=1:1:D
    for j=1:1:8   % to get all bits in A
        MSG(i,j) = bitget(A(i),j);
    end
end
j = 1;
for i=1:2:D
    U(j,:) = [MSG(i,:), MSG(i+1,:),
zeros(1,10)];
    j = j+1;
end
```

```
M = 5;

NN = 2^M-1;

KK = NN-M;

V = encode(U,NN,KK,'hamming');

  % Hamming(31,26,3) encoder

for i=1:1:D/2

    r(i,:) = -2*V(i,1:21)+1 + sqrt(noise_var)*
                 randn(1,21);

    % the shorten Hamming code is transmitted
with length 21 and information 16 bits

    % olny first 21 bits are fetched.

end
```

```
for i=1:1:D/2
  for j=1:1:21
   if(r(i,j) > 0) y(j) = 0;
% hard decision output
   else y(j) = 1;
   end
 end
   Y(i,:) = [y, zeros(1,10)];
end
U_hat = decode(Y,NN,KK,'hamming');
% hamming(31,26,3) decoder
```

```
for i=1:1:D/2

    b(2*i-1,:) = Y(i, 6:13); % disregard
parity check bits and only the 16 info. bits
are fetched

    b(2*i,:) = Y(i, 14:21);

    MSG_hat(2*i-1,:) = U_hat(i,1:8); %the info.
16 bits are fetched

    MSG_hat(2*i,:) = U_hat(i,9:16);

end

for i=1:1:D

    B(i) = Bits2num(b(i,:),8);

    C(i) = Bits2num(MSG_hat(i,:),8);  %the
info. Bits

end
```

```
fid2 = fopen('杜甫詩(AWGN雜訊干擾).txt','w');
fid3 = fopen('杜甫詩(Hamming decoded).txt','w');
fprintf(fid2,'%c',char(B));
fprintf(fid3,'%c',char(C));
fclose('all');
```

HW #4-1

1. In the previous MATLAB program, we encode each Chinese character with Hamming encoding.
2. Now, for some reason, we would like to encode this file “杜甫詩.txt” with Hamming encoding by the line-by-line way. Please modify this program and adjust the SNR such that there are no errors in the “decoded file”.
3. What kind of the shortened Hamming code is used ?