Linear Block Codes[1]

Manjunatha. P

manjup.jnnce@gmail.com

Professor

Dept. of ECE

J.N.N. College of Engineering, Shimoga

September 27, 2013

1. Generator and Parity check Matrices

1. Generator and Parity check Matrices
2. Encoding circuits

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities
6. Standard array and Syndrome decoding

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities
6. Standard array and Syndrome decoding
7. Decoding circuits

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities
6. Standard array and Syndrome decoding
7. Decoding circuits
8. Hamming Codes

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities
6. Standard array and Syndrome decoding
7. Decoding circuits
8. Hamming Codes
9. Reed Muller codes

1. Generator and Parity check Matrices
2. Encoding circuits
3. Syndrome and Error Detection
4. Minimum Distance Considerations
5. Error detecting and Error correcting capabilities
6. Standard array and Syndrome decoding
7. Decoding circuits
8. Hamming Codes
9. Reed Muller codes
10. The (24, 12) Golay code
11. Product codes and Interleaved codes

Introduction to Linear Block Codes

- Transmission through noisy channel.

- Transmission through noisy channel.
- Transmission errors can occur, 1's become 0's and 0's become 1's.

- Transmission through noisy channel.
- Transmission errors can occur, 1's become 0's and 0's become 1's.
- To correct the errors, some redundancy bits are added to the information sequence, at the receiver the correlation is exploited to locate transmission errors.

- Transmission through noisy channel.
- Transmission errors can occur, 1's become 0's and 0's become 1's.
- To correct the errors, some redundancy bits are added to the information sequence, at the receiver the correlation is exploited to locate transmission errors.
- Here, only binary transmission is considered.

- Transmission through noisy channel.
- Transmission errors can occur, 1's become 0's and 0's become 1's.
- To correct the errors, some redundancy bits are added to the information sequence, at the receiver the correlation is exploited to locate transmission errors.
- Here, only binary transmission is considered.

Type of Errors

Type of Errors

Single-bit error

Type of Errors

Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.

Type of Errors

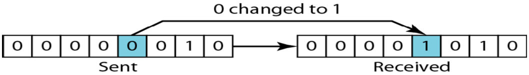
Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.

Type of Errors

Single-bit error

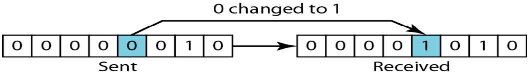
- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.

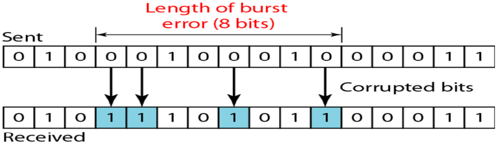


Type of Errors

Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.

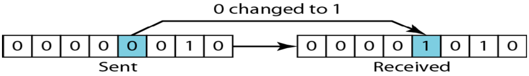


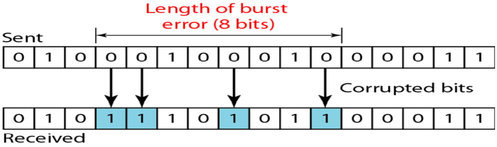


Type of Errors

Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.





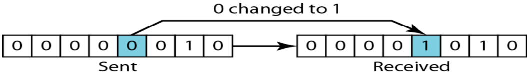
Burst error

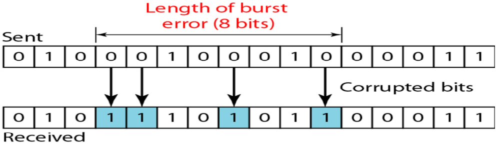
- 2 or more bits in the data unit have changed..

Type of Errors

Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.





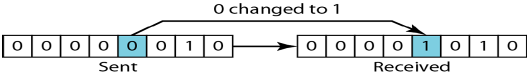
Burst error

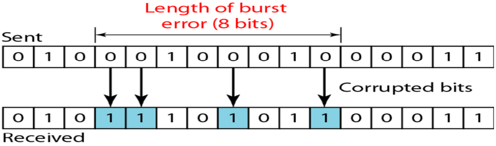
- 2 or more bits in the data unit have changed..
- More likely to occur than the single-bit error because the duration of noise is normally longer than the duration of 1 bit

Type of Errors

Single-bit error

- Only 1 bit in the data unit (packet, frame, cell) has changed.
- Either 1 to 0, or 0 to 1.





Burst error

- 2 or more bits in the data unit have changed..
- More likely to occur than the single-bit error because the duration of noise is normally longer than the duration of 1 bit

- The output of an information source is a sequence of binary digits "0" or "1"

- The output of an information source is a sequence of binary digits "0" or "1"
- Information sequence is segmented into message block of fixed length, denoted by u.

- The output of an information source is a sequence of binary digits "0" or "1"
- Information sequence is segmented into message block of fixed length, denoted by u.
- Each message block consists of k information digits. There are a total of 2^k distinct message.

- The output of an information source is a sequence of binary digits "0" or "1"
- Information sequence is segmented into message block of fixed length, denoted by u.
- Each message block consists of k information digits. There are a total of 2^k distinct message.
- The encoder transforms each input message u into a binary n-tuple v with $n > k$

- The output of an information source is a sequence of binary digits "0" or "1"
- Information sequence is segmented into message block of fixed length, denoted by u.
- Each message block consists of k information digits. There are a total of 2^k distinct message.
- The encoder transforms each input message u into a binary n-tuple v with $n > k$
- This n-tuple v is referred to as the code word (or code vector) of the message u.

- The output of an information source is a sequence of binary digits "0" or "1"
- Information sequence is segmented into message block of fixed length, denoted by u.
- Each message block consists of k information digits. There are a total of 2^k distinct message.
- The encoder transforms each input message u into a binary n-tuple v with $n > k$
- This n-tuple v is referred to as the code word (or code vector) of the message u.

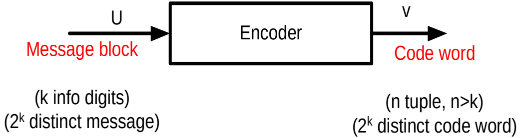


Figure: The encoder

- There are distinct 2^k code words.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.
- With this structure, the encoding complexity will be greatly reduced.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.
- With this structure, the encoding complexity will be greatly reduced.
- Definition : A block code of length n and 2^k code word is called a linear (n, k) code iff its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuple over the field GF(2).

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.
- With this structure, the encoding complexity will be greatly reduced.
- Definition : A block code of length n and 2^k code word is called a linear (n, k) code iff its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuple over the field GF(2).
- A binary block code is linear iff the module-2 sum of two code word is also a code word.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.
- With this structure, the encoding complexity will be greatly reduced.
- Definition : A block code of length n and 2^k code word is called a linear (n, k) code iff its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuple over the field GF(2).
- A binary block code is linear iff the module-2 sum of two code word is also a code word.
- The block code given in Table 3.1 is a (7, 4) linear code.

- There are distinct 2^k code words.
- This set of 2^k code words is called a block code.
- For a block code to be useful, there should be a one-to-one correspondence between a message u and its code word v.
- A desirable structure for a block code to possess is the linearity.
- With this structure, the encoding complexity will be greatly reduced.
- Definition : A block code of length n and 2^k code word is called a linear (n, k) code iff its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuple over the field GF(2).
- A binary block code is linear iff the module-2 sum of two code word is also a code word.
- The block code given in Table 3.1 is a (7, 4) linear code.

Message	Codewords
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 1 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space Vn of all the binary n tuple, it is possible to find k linearly independent code word, $g_0, g_0, \ldots g_{k-1}$ in C

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space Vn of all the binary n tuple, it is possible to find k linearly independent code word, $g_0, g_0, \ldots g_{k-1}$ in C

$$v = u_0 g_0 + u_1 g_1 + \ldots u_{k-1} g_{k-1} \tag{1}$$

- where $u_i = 0$ or 1 for $0 \leq i < k$

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space Vn of all the binary n tuple, it is possible to find k linearly independent code word, $g_0, g_0, \ldots g_{k-1}$ in C

$$v = u_0 g_0 + u_1 g_1 + \ldots u_{k-1} g_{k-1} \tag{1}$$

- where $u_i = 0$ or 1 for $0 \leq i < k$

Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space Vn of all the binary n tuple, it is possible to find k linearly independent code word, $g_0, g_0, \dots g_{k-1}$ in C

$$v = u_0 g_0 + u_1 g_1 + \dots u_{k-1} g_{k-1} \tag{1}$$

- where $u_i = 0$ or 1 for $0 \leq i < k$

Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & & & & \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{2}$$

- Since an (n, k) linear code C is a k-dimensional subspace of the vector space Vn of all the binary n tuple, it is possible to find k linearly independent code word, $g_0, g_0, \ldots g_{k-1}$ in C

$$v = u_0 g_0 + u_1 g_1 + \ldots u_{k-1} g_{k-1} \tag{1}$$

- where $u_i = 0$ or 1 for $0 \leq i < k$

Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & & & & \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{2}$$

where $g_i = (g_{i0}, g_{i1}, \ldots g_{in-1})0$ or 1 for $0 \leq i < k$

If $u = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded, the corresponding code word

$$v = U.G = \begin{pmatrix} u_0, & u_1, \ldots, & u_{k-1} \end{pmatrix} . \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} \tag{3}$$

$v = u_0 g_0 + u_1 g_1 + \ldots u_{k-1} g_{k-1}$

- Because the rows of G generate the (n, k) linear code C, the matrix G is called a generator matrix for C
- Note that any k linearly independent code words of an (n, k) linear code can be used to form a generator matrix for the code
- It follows from (3.3) that an (n, k) linear code is completely specified by the k rows of a generator matrix G

- The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix
- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

- The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix
- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

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

- The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix
- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

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

- If $u = (1101)$ is the message to be encoded, its corresponding code word, according to (3.3), would be

- The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix
- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

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

- If $u = (1101)$ is the message to be encoded, its corresponding code word, according to (3.3), would be
- $v = 1.g_0 + 1.g_1 + 0.g_2 + 1.g_3$

- The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix
- If u = (1 1 0 1) is the message to be encoded, its corresponding code word, according to (3.3), would be

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

- If $u = (1101)$ is the message to be encoded, its corresponding code word, according to (3.3), would be
- $v = 1.g_0 + 1.g_1 + 0.g_2 + 1.g_3$
- =1.(1 1 0 1 0 0 0)+ 1.(0 1 1 0 1 0 0)+ 0.(1 1 1 0 0 1 0)+1. (1 0 1 0 0 0 1)
- =(0 0 0 1 1 0 1)

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1
- where a code word is divided into two parts.

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1
- where a code word is divided into two parts.
- The message part consists of k information digits.

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1
- where a code word is divided into two parts.
- The message part consists of k information digits.
- The redundant checking part consists of n - k parity-check digits.

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1
- where a code word is divided into two parts.
- The message part consists of k information digits.
- The redundant checking part consists of n - k parity-check digits.
- A linear block code with this structure is referred to as a linear systematic block code.

- A desirable property for a linear block code is the systematic structure of the code words as shown in Fig. 3.1
- where a code word is divided into two parts.
- The message part consists of k information digits.
- The redundant checking part consists of n - k parity-check digits.
- A linear block code with this structure is referred to as a linear systematic block code.

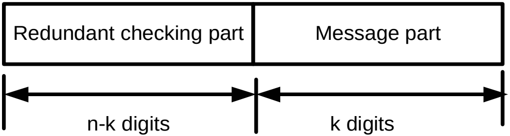


Figure: Systematic format of a codeword

A linear systematic (n, k) code is completely specified by a k x n matrix G of the following form:

A linear systematic (n, k) code is completely specified by a k x n matrix G of the following form:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{k-1} \end{bmatrix}$$

$$= \left[\begin{array}{cccc|ccccc} \multicolumn{4}{c}{\overbrace{\hphantom{p_{00} \quad p_{01} \quad \cdots \quad p_{0,n-k-1}}}^{P \quad Matrix}} & \multicolumn{5}{c}{\overbrace{\hphantom{1 \quad 0 \quad 0 \quad \ldots \quad 0}}^{kxk \quad Identity \quad Matrix}} \\ p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \ldots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \ldots & 0 \\ p_{20} & p_{21} & \cdots & p_{2,n-k-1} & 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & & & & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \ldots & 1 \end{array} \right] \tag{4}$$

where $p_{ij} = 0$ or 1

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

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \tag{5}$$

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \tag{5}$$

It follows from (4) (5) that the components of V are

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \tag{5}$$

It follows from (4) (5) that the components of V are

$$v_{n-k+i} = u_i \qquad for \;\; 0 \leq i < k \qquad (6a)$$

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \qquad (5)$$

It follows from (4) (5) that the components of V are

$$v_{n-k+i} = u_i \qquad for \;\; 0 \le i < k \qquad (6a)$$

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \ldots, + u_{k-1} p_{k-1,j} \quad for \;\; 0 \le j < n-k \quad (6b) \quad (6)$$

- Equation (6a) shows that the rightmost k digits of a code word v are identical to the information digits $u_0, u_1, \ldots u_{k-1}$ to be encoded

- Let $u = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot G \qquad (5)$$

It follows from (4) (5) that the components of V are

$$v_{n-k+i} = u_i \qquad \textit{for} \quad 0 \leq i < k \qquad (6a)$$

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \ldots, + u_{k-1} p_{k-1,j} \quad \textit{for} \quad 0 \leq j < n-k \quad (6b) \quad (6)$$

- Equation (6a) shows that the rightmost k digits of a code word v are identical to the information digits $u_0, u_1, \ldots u_{k-1}$ to be encoded
- Equation (6b) shown that the leftmost n-k redundant digits are linear sums of the information digits.

- Let $u = (u_0, u_1, \dots, u_{k-1})$ be the message to be encoded.
- The corresponding code word is.

$$v = (v_0, v_1, \dots, v_{n-1}) = (u_0, u_1, \dots, u_{k-1}) \cdot G \tag{5}$$

It follows from (4) (5) that the components of V are

$$v_{n-k+i} = u_i \qquad for \;\; 0 \leq i < k \qquad (6a)$$

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \dots, + u_{k-1} p_{k-1,j} \quad for \;\; 0 \leq j < n-k \quad (6b) \tag{6}$$

- Equation (6a) shows that the rightmost k digits of a code word v are identical to the information digits $u_0, u_1, \dots u_{k-1}$ to be encoded
- Equation (6b) shown that the leftmost n-k redundant digits are linear sums of the information digits.
- The n-k equations given by (6b) are called parity-check equations of the code.

- The matrix G given in example 3.1

- The matrix G given in example 3.1
- Let $u = (u_0, u_1, u_2, u_3)$ be the message to be encoded.

- The matrix G given in example 3.1
- Let $u = (u_0, u_1, u_2, u_3)$ be the message to be encoded.
- Let $v = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word.

- The matrix G given in example 3.1
- Let $u = (u_0, u_1, u_2, u_3)$ be the message to be encoded.
- Let $v = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word.

Solution :

$$v = u \cdot G = \begin{pmatrix} u_0, & u_1, & u_2, & u_3 \end{pmatrix} . \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- The matrix G given in example 3.1
- Let $u = (u_0, u_1, u_2, u_3)$ be the message to be encoded.
- Let $v = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word.

Solution :

$$v = u \cdot G = \begin{pmatrix} u_0, & u_1, & u_2, & u_3 \end{pmatrix} . \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

By matrix multiplication, we obtain the following digits of the code word v
$v_6 = u_3$, $v_5 = u_2$, $v_4 = u_2$, $v_3 = u_0$,$v_2 = u_1 + u_2 + u_3$, $v_1 = u_0 + u_1 + u_2$,
$v_0 = u_0 + u_2 + u_3$

- The matrix G given in example 3.1
- Let $u = (u_0, u_1, u_2, u_3)$ be the message to be encoded.
- Let $v = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word.

Solution :

$$v = u \cdot G = \begin{pmatrix} u_0, & u_1, & u_2, & u_3 \end{pmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

By matrix multiplication, we obtain the following digits of the code word v
$v_6 = u_3,\ v_5 = u_2,\ v_4 = u_2,\ v_3 = u_0, v_2 = u_1 + u_2 + u_3,\ v_1 = u_0 + u_1 + u_2,$
$v_0 = u_0 + u_2 + u_3$
The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1)

- For any k x n matrix G with k linearly independent rows, there exists an (n-k)x n matrix H with n-k linearly independent rows such that any vector in the row space of G is orthogonal to the rows of H and any vector that is orthogonal to the rows of H is in the row space of G.

- For any k x n matrix G with k linearly independent rows, there exists an (n-k)x n matrix H with n-k linearly independent rows such that any vector in the row space of G is orthogonal to the rows of H and any vector that is orthogonal to the rows of H is in the row space of G.
- An n-tuple v is a codeword in the code C generated by G if and only if

$$v \cdot H^T = 0$$

- For any k x n matrix G with k linearly independent rows, there exists an (n-k)x n matrix H with n-k linearly independent rows such that any vector in the row space of G is orthogonal to the rows of H and any vector that is orthogonal to the rows of H is in the row space of G.
- An n-tuple v is a codeword in the code C generated by G if and only if

$$v \cdot H^T = 0$$

- This matrix H is called a parity-check matrix of the code.

- For any k x n matrix G with k linearly independent rows, there exists an (n-k)x n matrix H with n-k linearly independent rows such that any vector in the row space of G is orthogonal to the rows of H and any vector that is orthogonal to the rows of H is in the row space of G.
- An n-tuple v is a codeword in the code C generated by G if and only if

$$v \cdot H^T = 0$$

- This matrix H is called a parity-check matrix of the code.
- The 2^{n-k} linear combinations of the rows of matrix H form an (n, n-k) linear code C_d

- For any k x n matrix G with k linearly independent rows, there exists an (n-k)x n matrix H with n-k linearly independent rows such that any vector in the row space of G is orthogonal to the rows of H and any vector that is orthogonal to the rows of H is in the row space of G.
- An n-tuple v is a codeword in the code C generated by G if and only if

$$v \cdot H^T = 0$$

- This matrix H is called a parity-check matrix of the code.
- The 2^{n-k} linear combinations of the rows of matrix H form an (n, n-k) linear code C_d
- This code is the null space of the (n, k) linear code C generated by matrix G.
- C_d is called the dual code of C

If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$H = [I_{n-k} P^T] =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \dots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \quad (7)$$

If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$H = [I_{n-k} P^T] =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \dots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \quad (7)$$

- Let h_j be the j^{th} row of H then inner product of the i^{th} row of G is

If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$H = [I_{n-k} P^T] =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \dots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \quad (7)$$

- Let h_j be the j^{th} row of H then inner product of the i^{th} row of G is

$$g_i.h_j = p_{ij} + p_{ij} = 0$$

for $0 \leq i < k$ and $0 \leq j < n-k$

If the generator matrix of an (n,k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$H = [I_{n-k} P^T] =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \dots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \tag{7}$$

- Let h_j be the j^{th} row of H then inner product of the i^{th} row of G is

$$g_i . h_j = p_{ij} + p_{ij} = 0$$

for $0 \leq i < k$ and $0 \leq j < n - k$

- This implies that

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

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

- Let $u = (u_0, u_1, \ldots u_{k-1})$ be the message to be encoded.
- In systematic form the corresponding code word would be

- Let $u = (u_0, u_1, \dots u_{k-1})$ be the message to be encoded.
- In systematic form the corresponding code word would be $v = (v_0, v_1, \dots v_{n-1}) = (v_0, v_1, \dots v_{n-k-1}, u_0, u_1, \dots u_{k-1})$
- Using the fact that $v.H^T = 0$, we obtain

- Let $u = (u_0, u_1, \ldots u_{k-1})$ be the message to be encoded.
- In systematic form the corresponding code word would be $v = (v_0, v_1, \ldots v_{n-1}) = (v_0, v_1, \ldots v_{n-k-1}, u_0, u_1, \ldots u_{k-1})$
- Using the fact that $v.H^T = 0$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} \ldots + u_{k-1} p_{k-1j} = 0 \tag{8}$$

- Let $u = (u_0, u_1, \ldots u_{k-1})$ be the message to be encoded.
- In systematic form the corresponding code word would be $v = (v_0, v_1, \ldots v_{n-1}) = (v_0, v_1, \ldots v_{n-k-1}, u_0, u_1, \ldots u_{k-1})$
- Using the fact that $v.H^T = 0$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} \ldots + u_{k-1} p_{k-1j} = 0 \tag{8}$$

for $0 \leq j < n - k$

- Rearranging the equation of (8), we obtain the same parity-check equations of (6b)

- Let $u = (u_0, u_1, \ldots u_{k-1})$ be the message to be encoded.
- In systematic form the corresponding code word would be $v = (v_0, v_1, \ldots v_{n-1}) = (v_0, v_1, \ldots v_{n-k-1}, u_0, u_1, \ldots u_{k-1})$
- Using the fact that $v.H^T = 0$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} \ldots + u_{k-1} p_{k-1j} = 0 \tag{8}$$

for $0 \leq j < n - k$

- Rearranging the equation of (8), we obtain the same parity-check equations of (6b)
- An (n, k) linear code is completely specified by its parity check

$$v.H^T = (v_0, v_1, \dots v_{n-k-1}, u_0, u_1, \dots u_{k-1}).H^T = 0$$

$$H^T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ p_{00} & p_{01} & p_{02} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,n-k-1} \\ \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & p_{k-1,2} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

$$v.H^T = (v_0, v_1, \ldots v_{n-k-1}, u_0, u_1, \ldots u_{k-1}).H^T = 0$$

$$H^T = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & 1 \\ p_{00} & p_{01} & p_{02} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,n-k-1} \\ \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & p_{k-1,2} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

$v_0(1) + v_1(1) + v_{n-1}(1) + u_0(p_{00}\ p_{01}\ \ldots\ p_{0,n-k-1})+$
$+u_1(p_{10}\ p_{11}\ p_{12} \ldots p_{1,n-k-1}) + u_2(p_{20}\ p_{21}\ p_{22} \ldots p_{2,n-k-1})$
$\ldots + u_{k-1}(p_{k-1,0}\ p_{k-1,1} \ldots p_{k-1,2}\ p_{k-1,n-k-1}) = 0$
$v_j + u_0 p_{0j} + u_1 p_{1j} \ldots + u_{k-1} p_{k-1j} = 0$ for $0 \leq j < n-k$

- Consider the generator matrix of a (7,4) linear code given in example 3.1

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

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

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

Ezample:

- Take any code given in table.
- $v.H^T = 0$ is

Ezample:

- Take any code given in table.
- $v.H^T = 0$ is

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

Summaries

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

Summaries

- For any (n, k) linear block code C, there exists a k n matrix G whose row space given C.
- There exist an (n - k) n matrix H such that an n-tuple v is a code word in C if and only if $v \cdot H^T = 0$

Summaries

- For any (n, k) linear block code C, there exists a k n matrix G whose row space given C.
- There exist an (n - k) n matrix H such that an n-tuple v is a code word in C if and only if $v \cdot H^T = 0$
- If G is of the form given by (4), then H may take form

Summaries

- For any (n, k) linear block code C, there exists a k n matrix G whose row space given C.
- There exist an (n - k) n matrix H such that an n-tuple v is a code word in C if and only if $v \cdot H^T = 0$
- If G is of the form given by (4), then H may take formgiven by (7), and vice versa

Summaries

- For any (n, k) linear block code C, there exists a k n matrix G whose row space given C.
- There exist an (n - k) n matrix H such that an n-tuple v is a code word in C if and only if $v \cdot H^T = 0$
- If G is of the form given by (4), then H may take formgiven by (7), and vice versa

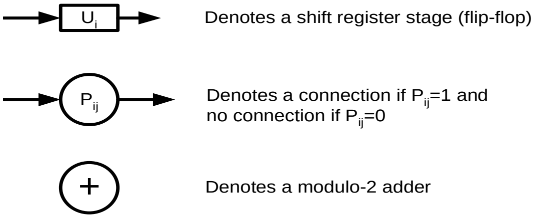
- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily.

- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily.
- The encoding circuit is shown in Fig. 3.2

- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily.
- The encoding circuit is shown in Fig. 3.2
- The complexity of the encoding circuit is linear proportional to the block length.

- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily.
- The encoding circuit is shown in Fig. 3.2
- The complexity of the encoding circuit is linear proportional to the block length.
- The encoding circuit for the (7,4) code given in Table 3.1 is shown in Fig 3.3

- Based on the equation of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily.
- The encoding circuit is shown in Fig. 3.2
- The complexity of the encoding circuit is linear proportional to the block length.
- The encoding circuit for the (7,4) code given in Table 3.1 is shown in Fig 3.3



$v_{n-k+i} = u_i \qquad for \quad 0 \leq i < k$

$v_j = u_0 p_{0j} + u_1 p_{1j} + \ldots, + u_{k-1} p_{k-1,j} \quad for \quad 0 \leq j < n-k$

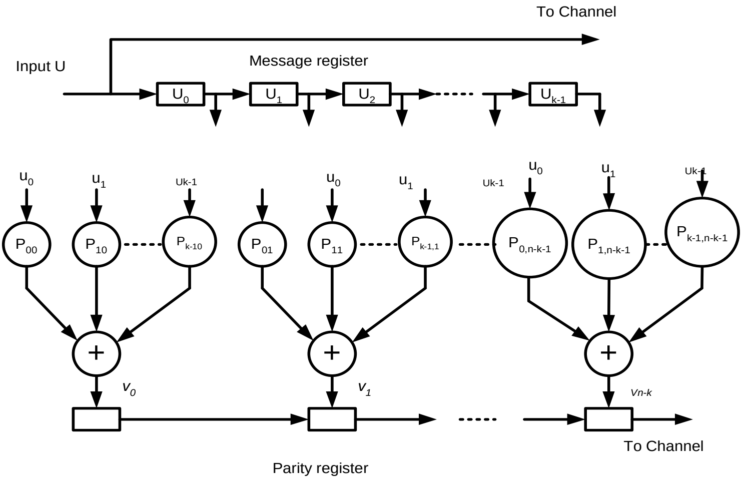


Figure: The encoding circuit for a liner system (n,k) code

The encoding circuit for a liner system (n,k) code

$v_6 = u_3,\ v_5 = u_2,\ v_4 = u_2,\ v_3 = u_0, v_2 = u_1 + u_2 + u_3,\ v_1 = u_0 + u_1 + u_2,$
$v_0 = u_0 + u_2 + u_3$

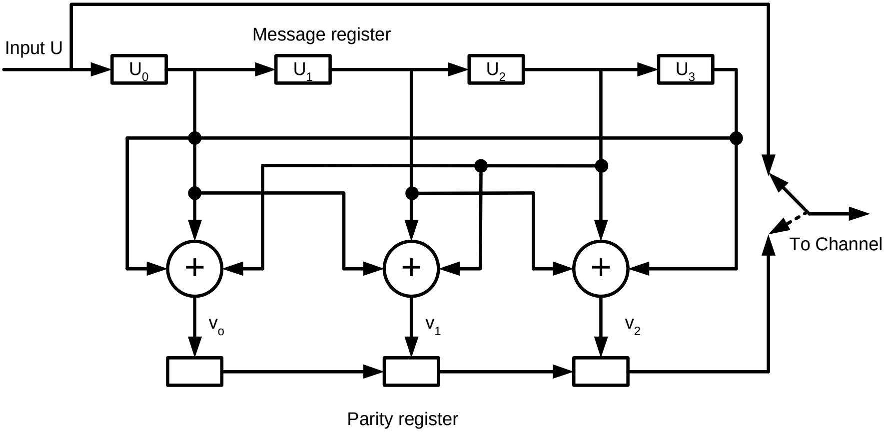


Figure: The encoding circuit for the (7,4)systematic code

Syndrome and Error Detection

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.
- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector.

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.
- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector.

$$e = r + v = (e_0, e_1, \ldots, e_{n-1}) \tag{9}$$

- $e_i = 1$ for $r_i \neq v_i$ or $e_i = 0$ for $r_i = v_i$

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.
- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector.

$$e = r + v = (e_0, e_1, \ldots, e_{n-1}) \tag{9}$$

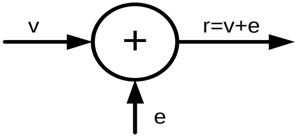
- $e_i = 1$ for $r_i \neq v_i$ or $e_i = 0$ for $r_i = v_i$
- The n-tuple e is called the error vector (or error pattern)

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.
- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector.

$$e = r + v = (e_0, e_1, \ldots, e_{n-1}) \tag{9}$$

- $e_i = 1$ for $r_i \neq v_i$ or $e_i = 0$ for $r_i = v_i$
- The n-tuple e is called the error vector (or error pattern)
- Received vector r is the vector sum of transmitted codeword and the error vector

$$r = v + e$$



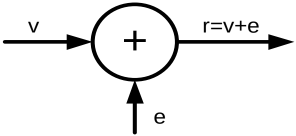
- The decoder first determine whether r contains errors.

- $v = (v_0, v_1, \ldots, v_{n-1})$ be a codeword transmitted over a noisy channel.
- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector.

$$e = r + v = (e_0, e_1, \ldots, e_{n-1}) \tag{9}$$

- $e_i = 1$ for $r_i \neq v_i$ or $e_i = 0$ for $r_i = v_i$
- The n-tuple e is called the error vector (or error pattern)
- Received vector r is the vector sum of transmitted codeword and the error vector

$$r = v + e$$



- The decoder first determine whether r contains errors.
- If errors are detected, correct errors (FEC) or Request for a retransmission of v (ARQ).

- When r is received, the decoder computes the following (n-k)-tuple:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}) \tag{10}$$

- When r is received, the decoder computes the following (n-k)-tuple:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}) \tag{10}$$

- which is called the syndrome of r
- $s = 0$ if and only if r is a code word and receiver accepts r as the transmitted code word

- When r is received, the decoder computes the following (n-k)-tuple:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}) \tag{10}$$

- which is called the syndrome of r
- $s = 0$ if and only if r is a code word and receiver accepts r as the transmitted code word
- $s \neq 0$ if and only if r is not a code word and the presence of errors has been detected

- When r is received, the decoder computes the following (n-k)-tuple:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}) \tag{10}$$

- which is called the syndrome of r
- $s = 0$ if and only if r is a code word and receiver accepts r as the transmitted code word
- $s \neq 0$ if and only if r is not a code word and the presence of errors has been detected
- When the error pattern e is identical to a nonzero code word (i.e., r contain errors but $s = r \cdot H^T = 0$), error patterns of this kind are called undetectable error patterns

- When r is received, the decoder computes the following (n-k)-tuple:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}) \tag{10}$$

- which is called the syndrome of r
- $s = 0$ if and only if r is a code word and receiver accepts r as the transmitted code word
- $s \neq 0$ if and only if r is not a code word and the presence of errors has been detected
- When the error pattern e is identical to a nonzero code word (i.e., r contain errors but $s = r \cdot H^T = 0$), error patterns of this kind are called undetectable error patterns
- Since there are 2^{k-1} nonzero code words, there are 2^{k-1} undetectable error patterns

$$S = (r_0, r_1, \ldots, r_{n-1}).H^T = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & 1 \\ p_{00} & p_{01} & p_{02} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,n-k-1} \\ \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & p_{k-1,2} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

- Based on Equation 10, the syndrome digits are as follows:

$$S = (r_0, r_1, \ldots, r_{n-1}).H^T = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & 1 \\ p_{00} & p_{01} & p_{02} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,n-k-1} \\ \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & p_{k-1,2} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

- Based on Equation 10, the syndrome digits are as follows:

$$s_0 = r_0 + r_{n-k}p_{00} + r_{n-k+1}p_{10} + \ldots + r_{n-1}p_{k-1,0}$$
$$s_1 = r_1 + r_{n-k}p_{01} + r_{n-k+1}p_{11} + \ldots + r_{n-1}p_{k-1,1}$$

$$s_{n-k-1} = r_{n-k-1} + r_{n-k}p_{0,n-k-1} \ldots + r_{n-1}p_{k-1,n-k-1} \tag{11}$$

- The syndrome s is the vector sum of the received parity digits

- The syndrome s is the vector sum of the received parity digits $(r_0, r_1, \ldots, r_{n-k-1})$ and the parity-check digits recomputed from the received information digits $(r_{n-k}, r_{n-k+1} \ldots, r_{n1})$
- A general syndrome circuit is shown in Fig. 5

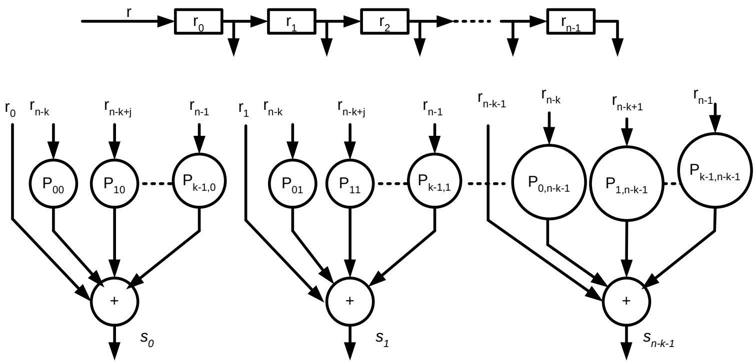


Figure: Syndrome circuit for a liner system (n,k) code

Example 3.4

- The parity-check matrix is given in example 3.3

Example 3.4

- The parity-check matrix is given in example 3.3
- Let $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector

Example 3.4

- The parity-check matrix is given in example 3.3
- Let $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector
- The syndrome is given by

Example 3.4

- The parity-check matrix is given in example 3.3
- Let $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector
- The syndrome is given by

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

- $s_0 = r_0 + r_3 + r_5 + r_6$
- $s_1 = r_1 + r_3 + r_4 + r_5$
- $s_2 = r_2 + r_4 + r_5 + r_6$

- $s_0 = r_0 + r_3 + r_5 + r_6$
- $s_1 = r_1 + r_3 + r_4 + r_5$
- $s_2 = r_2 + r_4 + r_5 + r_6$
- The syndrome circuit for this code is shown below

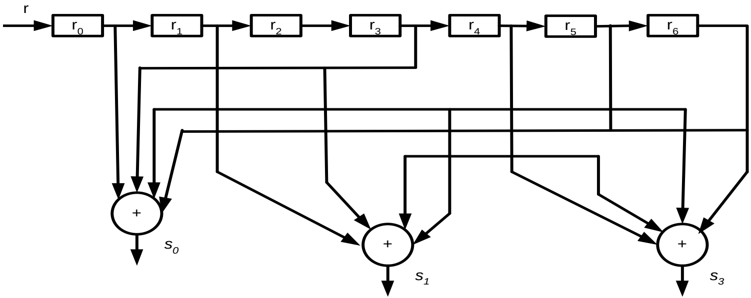


Figure: Syndrome circuit for a liner system (n,k) code

- Since r is the vector sum of v and e, it follows from (3.10) that

- Since r is the vector sum of v and e, it follows from (3.10) that

$$s = r.H^T = (v + e).H^T = v.H^T + e.H^T$$

- Since r is the vector sum of v and e, it follows from (3.10) that

$$s = r.H^T = (v + e).H^T = v.H^T + e.H^T$$

$$v.H^T = 0$$

- The relation between the syndrome and the error pattern is:

- Since r is the vector sum of v and e, it follows from (3.10) that

$$s = r.H^T = (v + e).H^T = v.H^T + e.H^T$$

$$v.H^T = 0$$

- The relation between the syndrome and the error pattern is:

$$s = e.H^T \tag{12}$$

- Since r is the vector sum of v and e, it follows from (3.10) that

$$s = r.H^T = (v + e).H^T = v.H^T + e.H^T$$

$$v.H^T = 0$$

- The relation between the syndrome and the error pattern is:

$$s = e.H^T \tag{12}$$

- If the parity-check matrix H is expressed in the systematic form as given by (3.7), multiplying out $e.H^T$ yield the following linear relationship between the syndrome digits and the error digits:

- Since r is the vector sum of v and e, it follows from (3.10) that

$$s = r.H^T = (v + e).H^T = v.H^T + e.H^T$$

$$v.H^T = 0$$

- The relation between the syndrome and the error pattern is:

$$s = e.H^T \tag{12}$$

- If the parity-check matrix H is expressed in the systematic form as given by (3.7), multiplying out $e.H^T$ yield the following linear relationship between the syndrome digits and the error digits:

$$s_0 = e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \ldots + e_{n-1}p_{k-1,0}$$
$$s_1 = e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \ldots + e_{n-1}p_{k-1,1}$$

$$s_{n-k-1} = e_{n-k-1} + e_{n-k}p_{0,n-k-1} \ldots + e_{n-1}p_{k-1,n-k-1} \tag{13}$$

- The syndrome digits are linear combinations of the error digits

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction
- Because the n-k linear equations of (3.13) do not have a unique solution but have 2^k solutions

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction
- Because the n-k linear equations of (3.13) do not have a unique solution but have 2^k solutions
- There are 2^k error pattern that result in the same syndrome, and the true error pattern e is one of them

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction
- Because the n-k linear equations of (3.13) do not have a unique solution but have 2^k solutions
- There are 2^k error pattern that result in the same syndrome, and the true error pattern e is one of them
- The decoder has to determine the true error vector from a set of 2^k candidates

- The syndrome digits are linear combinations of the error digits
- The syndrome digits can be used for error correction
- Because the n-k linear equations of (3.13) do not have a unique solution but have 2^k solutions
- There are 2^k error pattern that result in the same syndrome, and the true error pattern e is one of them
- The decoder has to determine the true error vector from a set of 2^k candidates
- To minimize the probability of a decoding error, the most probable error pattern that satisfies the equations of (3.13) is chosen as the true error vector

- Consider the code C(5,2) with the parity check matrix

- Consider the code C(5,2) with the parity check matrix

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

- Consider the code C(5,2) with the parity check matrix

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

- Let v=(0 0 1 1 1) be the transmitted codeword over BSC and r=(1 0 1 1 1) be received vector.

- Consider the code C(5,2) with the parity check matrix

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

- Let v=(0 0 1 1 1) be the transmitted codeword over BSC and r=(1 0 1 1 1) be received vector.
- The problem is to find the digits of an error pattern $e = (e_0, e_1, e_2, e_3, e_4)$ Compute the syndrome $S = (s_0, s_1, s_2)$ of r=(1 0 1 1 1)

- Consider the code C(5,2) with the parity check matrix

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

- Let v=(0 0 1 1 1) be the transmitted codeword over BSC and r=(1 0 1 1 1) be received vector.
- The problem is to find the digits of an error pattern $e = (e_0, e_1, e_2, e_3, e_4)$ Compute the syndrome $S = (s_0, s_1, s_2)$ of r=(1 0 1 1 1)

$$s = r.H^T = (1\ 0\ 1\ 1\ 1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = (1\ 0\ 0)$$

- Consider the code C(5,2) with the parity check matrix

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

- Let v=(0 0 1 1 1) be the transmitted codeword over BSC and r=(1 0 1 1 1) be received vector.
- The problem is to find the digits of an error pattern $e = (e_0, e_1, e_2, e_3, e_4)$ Compute the syndrome $S = (s_0, s_1, s_2)$ of r=(1 0 1 1 1)

$$s = r.H^T = (1\ 0\ 1\ 1\ 1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = (1\ 0\ 0)$$

- Solve the system for $e = (e_0, e_1, e_2, e_3, e_4)$ with s=(1 0 0) as

- Solve the system for $e = (e_0, e_1, e_2, e_3, e_4)$ with s=(1 0 0) as

$$H.e^T = s^T \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- Solve the system for $e = (e_0, e_1, e_2, e_3, e_4)$ with s=(1 0 0) as

$$H.e^T = s^T \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} e_0 + e_3 + e_4 &= 1 \\ e_1 + e_3 + e_4 &= 0 \\ e_2 + e_3 &= 0 \end{aligned}$$

- Solve the system for $e = (e_0, e_1, e_2, e_3, e_4)$ with s=(1 0 0) as

$$H.e^T = s^T \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} e_0 + e_3 + e_4 &= 1 \\ e_1 + e_3 + e_4 &= 0 \\ e_2 + e_3 &= 0 \end{aligned}$$

- There are $2^2 = 4$ error patterns that satisfy the above system depending on $e_3 e_4$=00 or 01 or 10 or 11, they are (1 0 0 0 0), (0 1 0 0 1), (0 1 1 1 0),(1 0 1 1 1)

- Solve the system for $e = (e_0, e_1, e_2, e_3, e_4)$ with s=(1 0 0) as

$$H.e^T = s^T \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} e_0 + e_3 + e_4 &= 1 \\ e_1 + e_3 + e_4 &= 0 \\ e_2 + e_3 &= 0 \end{aligned}$$

- There are $2^2 = 4$ error patterns that satisfy the above system depending on $e_3 e_4$=00 or 01 or 10 or 11, they are (1 0 0 0 0), (0 1 0 0 1), (0 1 1 1 0),(1 0 1 1 1)

- Now, since the channel is Binary Symmetric Channel (BSC), Then the most probable error pattern that satisfies the system above is e=(1 0 0 0 0) which has the smallest number of nonzero digits.
- The receiver decodes the received word r=(1 0 1 1 1) into the following codeword v*=r+e=(1 0 1 1 1)+(1 0 0 0 0)=(0 0 1 1 1)

- Now, since the channel is Binary Symmetric Channel (BSC), Then the most probable error pattern that satisfies the system above is e=(1 0 0 0 0) which has the smallest number of nonzero digits.
- The receiver decodes the received word r=(1 0 1 1 1) into the following codeword v*=r+e=(1 0 1 1 1)+(1 0 0 0 0)=(0 0 1 1 1)
- We see that the receiver has made a correct decoding.

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

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

$$s = r.H^T = (1 \quad 1 \quad 1)$$

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

$$s = r.H^T = (1 \ \ 1 \ \ 1)$$

- The receiver attempts to determine the true error vector

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

$$s = r.H^T = (1 \ \ 1 \ \ 1)$$

- The receiver attempts to determine the true error vector$e = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$, which yields the syndrome above

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

$$s = r.H^T = (1 \;\; 1 \;\; 1)$$

- The receiver attempts to determine the true error vector$e = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$, which yields the syndrome above

$$\begin{aligned} 1 &= e_0 + e_3 + e_5 + e_6 \\ 1 &= e_1 + e_3 + e_4 + e_5 \\ 1 &= e_2 + e_4 + e_5 + e_6 \end{aligned}$$

- We consider the (7,4) code whose parity-check matrix is given in example 3.3
- Let v = (1 0 0 1 0 1 1) be the transmitted code word
- Let r = (1 0 0 1 0 0 1) be the received vector
- The receiver computes the syndrome

$$s = r.H^T = (1 \;\; 1 \;\; 1)$$

- The receiver attempts to determine the true error vector$e = (e_0, e_1, e_2, e_3, e_4, e_5, e_6)$, which yields the syndrome above

$$\begin{aligned} 1 &= e_0 + e_3 + e_5 + e_6 \\ 1 &= e_1 + e_3 + e_4 + e_5 \\ 1 &= e_2 + e_4 + e_5 + e_6 \end{aligned}$$

- There are $2^4 = 16$ error patterns that satisfy the equations above.

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

- The error vector e = (0 0 0 0 0 1 0) has the smallest number of nonzero components

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

- The error vector e = (0 0 0 0 0 1 0) has the smallest number of nonzero components
- If the channel is a Binary Symmetric Channel (BSC), e = (0 0 0 0 0 1 0) is the most probable error vector that satisfies the equation above

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

- The error vector e = (0 0 0 0 0 1 0) has the smallest number of nonzero components
- If the channel is a Binary Symmetric Channel (BSC), e = (0 0 0 0 0 1 0) is the most probable error vector that satisfies the equation above
- Taking e = (0 0 0 0 0 1 0) as the true error vector, the receiver decodes the received vector r = (1 0 0 1 0 0 1) into the following code word

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

- The error vector e = (0 0 0 0 0 1 0) has the smallest number of nonzero components
- If the channel is a Binary Symmetric Channel (BSC), e = (0 0 0 0 0 1 0) is the most probable error vector that satisfies the equation above
- Taking e = (0 0 0 0 0 1 0) as the true error vector, the receiver decodes the received vector r = (1 0 0 1 0 0 1) into the following code word
- $v^* = r + e = (1\ 0\ 0\ 1\ 0\ 0\ 1) + (0\ 0\ 0\ 0\ 0\ 1\ 0)$

(0000010),(1101010),(0110110),(1011110),
(1110000),(0011000),(1000100),(0101100),
(1010011),(0111011),(1100111),(0001111),
(0100001),(1001001),(0010101),(1111101)

- The error vector e = (0 0 0 0 0 1 0) has the smallest number of nonzero components
- If the channel is a Binary Symmetric Channel (BSC), e = (0 0 0 0 0 1 0) is the most probable error vector that satisfies the equation above
- Taking e = (0 0 0 0 0 1 0) as the true error vector, the receiver decodes the received vector r = (1 0 0 1 0 0 1) into the following code word
- v* = r + e = (1 0 0 1 0 0 1) + (0 0 0 0 0 1 0)= (1 0 0 1 0 1 1)
- where v* is the actual transmitted code word

The Minimum Distance of a Block Code

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

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the Hamming weight (or simply weight) of v, denoted by w(v), is defined as the number of nonzero components of v.
- For example, the Hamming weight of v = (1 0 0 0 1 1 0) is 3.

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the Hamming weight (or simply weight) of v, denoted by w(v), is defined as the number of nonzero components of v.
- For example, the Hamming weight of v = (1 0 0 0 1 1 0) is 3.
- Let v and w be two n-tuple, the Hamming distance between v and w, denoted $d(v, w)$, is defined as the number of places where they differ.

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the Hamming weight (or simply weight) of v, denoted by w(v), is defined as the number of nonzero components of v.
- For example, the Hamming weight of v = (1 0 0 0 1 1 0) is 3.
- Let v and w be two n-tuple, the Hamming distance between v and w, denoted $d(v, w)$, is defined as the number of places where they differ.
- For example, the Hamming distance between v = (1 0 0 1 0 1 1) and w = (0 1 0 0 0 1 1) is 3

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the Hamming weight (or simply weight) of v, denoted by w(v), is defined as the number of nonzero components of v.
- For example, the Hamming weight of v = (1 0 0 0 1 1 0) is 3.
- Let v and w be two n-tuple, the Hamming distance between v and w, denoted $d(v, w)$, is defined as the number of places where they differ.
- For example, the Hamming distance between v = (1 0 0 1 0 1 1) and w = (0 1 0 0 0 1 1) is 3
- The Hamming distance is a metric function that satisfied the triangle inequality.

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple, the Hamming weight (or simply weight) of v, denoted by w(v), is defined as the number of nonzero components of v.
- For example, the Hamming weight of v = (1 0 0 0 1 1 0) is 3.
- Let v and w be two n-tuple, the Hamming distance between v and w, denoted $d(v, w)$, is defined as the number of places where they differ.
- For example, the Hamming distance between v = (1 0 0 1 0 1 1) and w = (0 1 0 0 0 1 1) is 3
- The Hamming distance is a metric function that satisfied the triangle inequality.

$$d(v, w) + d(w, x) \geq d(v, x) \qquad (3.14) \qquad (14)$$

- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, v and w, is equal to the Hamming weight of the sum of v and w, that is.

- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, v and w, is equal to the Hamming weight of the sum of v and w, that is.

$$d(v, w) = w(v + w) \quad (3.15) \qquad (15)$$

- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, v and w, is equal to the Hamming weight of the sum of v and w, that is.

$$d(v, w) = w(v + w) \qquad (3.15) \qquad (15)$$

- For example, the Hamming distance between v = (1 0 0 1 0 1 1) and w = (1 1 1 0 0 1 0) is 4 and the weight of v + w = (0 1 1 1 0 0 1) is also 4.

- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, v and w, is equal to the Hamming weight of the sum of v and w, that is.

$$d(v, w) = w(v + w) \qquad (3.15) \qquad (15)$$

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

- From the definition of Hamming distance and the definition of module-2 addition that the Hamming distance between two n-tuple, v and w, is equal to the Hamming weight of the sum of v and w, that is.

$$d(v, w) = w(v + w) \qquad (3.15) \qquad (15)$$

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

$$d_{min} = min\{d(v, w) : v, w \in C, v \neq w\} \qquad (3.16) \qquad (16)$$

- If C is a linear block, the sum of two vectors is also a code vector.

- If C is a linear block, the sum of two vectors is also a code vector.
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C.

- If C is a linear block, the sum of two vectors is also a code vector.
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C.

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

- If C is a linear block, the sum of two vectors is also a code vector.
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C.

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

- The parameter $w_{min} \equiv \{w(x) : x \in C, x \neq 0\}$ is called the minimum weight of the linear code C.

- If C is a linear block, the sum of two vectors is also a code vector.
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C.

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

- The parameter $w_{min} \equiv \{w(x) : x \in C, x \neq 0\}$ is called the minimum weight of the linear code C.

Theorem 3.1

- *The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words.*

- If C is a linear block, the sum of two vectors is also a code vector.
- From (3.15) that the Hamming distance between two code vectors in C is equal to the Hamming weight of a third code vector in C.

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

- The parameter $w_{min} \equiv \{w(x) : x \in C, x \neq 0\}$ is called the minimum weight of the linear code C.

Theorem 3.1

- *The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words.*
- The (7,4) code has minimum weight of 3.

Theorem 3.2

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

Proof

- Let the parity-check matrix be

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l and v has l nonzero components.

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l and v has l nonzero components.
- Let$v_{i1}, v_{i2}, \ldots, v_{il}$ be the l nonzero components of v,

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l and v has l nonzero components.
- Let$v_{i1}, v_{i2}, \ldots, v_{il}$ be the l nonzero components of v,
- where $0 \leq i_1 < i_2 < \ldots < i_l \leq n-1$, then $v_{i1} = v_{i2} = \ldots = v_{il} = 1$

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l and v has l nonzero components.
- Let$v_{i1}, v_{i2}, \ldots, v_{il}$ be the l nonzero components of v,
- where $0 \leq i_1 < i_2 < \ldots < i_l \leq n-1$, then $v_{i1} = v_{i2} = \ldots = v_{il} = 1$
- since v is code vector, we must have

Proof

- Let the parity-check matrix be

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

- where h_i represents the i_{th} column of H
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector of weight l and v has l nonzero components.
- Let$v_{i1}, v_{i2}, \ldots, v_{il}$ be the l nonzero components of v,
- where $0 \leq i_1 < i_2 < \ldots < i_l \leq n-1$, then $v_{i1} = v_{i2} = \ldots = v_{il} = 1$
- since v is code vector, we must have

$$\begin{aligned} 0 &= v.H^T \\ &= v_0 h_0 + v_1 h_1 + \ldots + v_{n-1} h_{n-1} \\ &= v_{i1} h_{i1} + v_{i2} h_{i2} + \ldots + v_{il} h_{il} \\ &= h_{i1} + h_{i2} + \ldots + h_{il} \end{aligned}$$

Proof

- Suppose that $h_{i1}, h_{i2}, \ldots, h_{il}$ are l columns of H such that

Proof

- Suppose that $h_{i1}, h_{i2}, \ldots, h_{il}$ are l columns of H such that

$$h_{i1} + h_{i2} + \ldots + h_{il} = 0 \quad (3.18) \qquad (17)$$

Proof

- Suppose that $h_{i1}, h_{i2}, \ldots, h_{il}$ are l columns of H such that

$$h_{i1} + h_{i2} + \ldots + h_{il} = 0 \qquad (3.18) \qquad (17)$$

- Let $x = (x1, x2, \ldots, xn-1)$ whose nonzero components are x_{i1}, x_{i2}, x_{il}

Proof

- Suppose that $h_{i1}, h_{i2}, \ldots, h_{il}$ are l columns of H such that

$$h_{i1} + h_{i2} + \ldots + h_{il} = 0 \qquad (3.18) \qquad (17)$$

- Let $x = (x1, x2, \ldots, xn-1)$ whose nonzero components are x_{i1}, x_{i2}, x_{il}

$$\begin{aligned} x.H^T &= x_0 h_0 + x_1 h_1 + \ldots + x_{n-1} h_{n-1} \\ &= x_{i1} h_{i1} + x_{i2} h_{i2} + \ldots + x_{il} h_{il} \\ &= h_{i1} + h_{i2} + \ldots + h_{il} \end{aligned}$$

Proof

- Suppose that $h_{i1}, h_{i2}, \ldots, h_{il}$ are l columns of H such that

$$h_{i1} + h_{i2} + \ldots + h_{il} = 0 \qquad (3.18) \qquad (17)$$

- Let $x = (x1, x2, \ldots, xn - 1)$ whose nonzero components are x_{i1}, x_{i2}, x_{il}

$$\begin{aligned} x.H^T &= x_0 h_0 + x_1 h_1 + \ldots + x_{n-1} h_{n-1} \\ &= x_{i1} h_{i1} + x_{i2} h_{i2} + \ldots + x_{il} h_{il} \\ &= h_{i1} + h_{i2} + \ldots + h_{il} \end{aligned}$$

- It following from (3.18) that $x.H^T = 0$, x is code vector of weight l in C

Corollary 3.2.1

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

Corollary 3.2.1

- Let C be a linear block code with parity-check matrix H. If no d-1 or fewer columns of H add to 0, the code has minimum weight at least d.
  Corollary 3.2.2
- The minimum weight of C is equal to the smallest number of columns of H that sum to 0.

Corollary 3.2.1

- Let C be a linear block code with parity-check matrix H. If no d-1 or fewer columns of H add to 0, the code has minimum weight at least d.
  Corollary 3.2.2
- The minimum weight of C is equal to the smallest number of columns of H that sum to 0.

Error-Detecting and Error-Correcting Capabilities of a Block Code

Error-Detecting and Error-Correcting Capabilities of a Block Code

- If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places.

- If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places.
- A block code with minimum distance d_{min} is capable of detecting all the error pattern of d_{min}- 1 or fewer errors.

- If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places.
- A block code with minimum distance d_{min} is capable of detecting all the error pattern of d_{min}- 1 or fewer errors.
- However, it cannot detect all the error pattern of d_{min} errors because there exists at least one pair of code vectors that differ in d_{min} places and there is an error pattern of d_{min} errors that will carry one into the other.

- If the minimum distance of a block code C is d_{min}, any two distinct code vector of C differ in at least d_{min} places.
- A block code with minimum distance d_{min} is capable of detecting all the error pattern of d_{min}- 1 or fewer errors.
- However, it cannot detect all the error pattern of d_{min} errors because there exists at least one pair of code vectors that differ in d_{min} places and there is an error pattern of d_{min} errors that will carry one into the other.
- The random-error-detecting capability of a block code with minimum distance d_{min} is d_{min}-1.

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.
- Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words.

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.
- Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words.
- If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word v into another code word w, thus w will be received and its syndrome is zero.

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.
- Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words.
- If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word v into another code word w, thus w will be received and its syndrome is zero.
- There are $2^k - 1$ undetectable error patterns.
- If an error pattern is not identical to a nonzero code word, the received vector r will not be a code word and the syndrome will not be zero.

- An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.
- Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words.
- If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word v into another code word w, thus w will be received and its syndrome is zero.
- There are $2^k - 1$ undetectable error patterns.
- If an error pattern is not identical to a nonzero code word, the received vector r will not be a code word and the syndrome will not be zero.
- These $2^n - 2^k$ error patterns are detectable error patterns.

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.
- Let Pu(E) denote the probability of an undetected error.

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.
- Let Pu(E) denote the probability of an undetected error.
- Since an undetected error occurs only when the error pattern is identical to a nonzero code vector of C .

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.
- Let Pu(E) denote the probability of an undetected error.
- Since an undetected error occurs only when the error pattern is identical to a nonzero code vector of C .

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \tag{18}$$

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.
- Let Pu(E) denote the probability of an undetected error.
- Since an undetected error occurs only when the error pattern is identical to a nonzero code vector of C .

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \tag{18}$$

where p is the transition probability of the BSC.

- Let A_i be the number of code vectors of weight i in C, the numbers $A_0, A_1, ..., A_n$ are called the weight distribution of C.
- Let Pu(E) denote the probability of an undetected error.
- Since an undetected error occurs only when the error pattern is identical to a nonzero code vector of C .

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \tag{18}$$

where p is the transition probability of the BSC.

- If the minimum distance of C is d_{min}, then A_1 to $A_{dmin} - 1$ are zero.

- Consider the (7,4) code given in table. The weight distribution is: $A_0 = 1, A_1 = A_2 = 0, A_3 = A_4 = 7, A_5 = A_6 = 0$,and $A_7 = 1$

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

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

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

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

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

- If $p = 10^{-2}$ then $P_u(E) = 7x10^{-6}$ this means, if 1 million codewords are transmitted over a BSC with $p = 10^{-2}$ on average seven erroneous codewords pass through the decoder without being detected.

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

$$d(v, r) + d(w, r) \geq 2t + 1$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

- Suppose that an t' errors occurs during the transmission of v. Then $d(v, r) = t'$.
- Since v and w are code vectors in C, we have

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

$$d(v, r) + d(w, r) \geq 2t + 1$$

and $d(v, r) = t'$

$$d(w, r) \geq 2t + 1 - t'$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

$$d(v, r) + d(w, r) \geq 2t + 1$$

and $d(v, r) = t^{'}$

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

$$if \quad t^{'} \leq t \Rightarrow d(w, r) > t$$

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

$$d(v, r) + d(w, r) \geq 2t + 1$$

and $d(v, r) = t^{'}$

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

$$if \quad t^{'} \leq t \Rightarrow d(w, r) > t$$

- The inequality above says that if an error pattern of t or fewer errors occurs, the received vector r is closer (in Hamming distance) to the transmitted code vector v than to any other code vector w in C.

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Consider a block code C with minimum distance d_{min} is used for random error correction and d_{min} is either odd or even.
- Let t be a positive integer such that:

$$2t + 1 \leq d_{min} \leq 2t + 2 \tag{19}$$

Fact 1:

- The code C is capable of correcting all the error patterns of t or fewer errors.

Proof:

- Let **v** and **r** be the transmitted code vector and the received vector, respectively and **w** be any other code vector in C.

$$d(v, r) + d(w, r) \geq d(v, w) \tag{20}$$

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

$$d(v, w) \geq d_{min} \geq 2t + 1. \tag{21}$$

Combining equation 20 and equation 21

$$d(v, r) + d(w, r) \geq 2t + 1$$

and $d(v, r) = t^{'}$

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

$$if \quad t^{'} \leq t \Rightarrow d(w, r) > t$$

- The inequality above says that if an error pattern of t or fewer errors occurs, the received vector r is closer (in Hamming distance) to the transmitted code vector v than to any other code vector w in C.
- For a BSC, this means that the conditional probability $P(r|v)$ is greater than the conditional probability $P(r|w)$ for $w \neq v$.

Fact 2:

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.
- Let e_1 and e_2 be two error patterns that satisfy the following conditions:

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.
- Let e_1 and e_2 be two error patterns that satisfy the following conditions:
- 1. $e_1 + e_2 = v + w$

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.
- Let e_1 and e_2 be two error patterns that satisfy the following conditions:
- 1. $e_1 + e_2 = v + w$
  2. e_1 and e_2 do not have nonzero components in common places.

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.
- Let e_1 and e_2 be two error patterns that satisfy the following conditions:
- 1. $e_1 + e_2 = v + w$
  2. e_1 and e_2 do not have nonzero components in common places.
- We have

Fact 2:

- The code is not capable of correcting all the error patterns of l errors with $l > t$, for there is at least one case where an error pattern of l errors results in a received vector which is closer to an incorrect code vector than to the actual transmitted code vector.
- Proof:
- Let v and w be two code vectors in C such that $d(v, w) = d_{min}$.
- Let e_1 and e_2 be two error patterns that satisfy the following conditions:
-
  1. $e_1 + e_2 = v + w$
  2. e_1 and e_2 do not have nonzero components in common places.
- We have

$$w(e_1) + w(e_2) = w(v + w) = d(v, w) = d_{min}.(3.23) \qquad (22)$$

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \tag{23}$$

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \tag{23}$$

- The Hamming distance between w and r is

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \qquad (23)$$

- The Hamming distance between w and r is

$$d(w, r) = w(w + r) = w(w + v + e_1) = w(e_2)(3.25) \qquad (24)$$

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \tag{23}$$

- The Hamming distance between w and r is

$$d(w, r) = w(w + r) = w(w + v + e_1) = w(e_2)(3.25) \tag{24}$$

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

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \tag{23}$$

- The Hamming distance between w and r is

$$d(w, r) = w(w + r) = w(w + v + e_1) = w(e_2)(3.25) \tag{24}$$

- Now, suppose that the error pattern e_1 contains more than t errors [i.e. $w(e_1) \geq t + 1$].
- Since $2t + 1 \leq d_{min} 2t + 2$, it follows from (3.23) that

- Suppose that v is transmitted and is corrupted by the error pattern e_1, then the received vector is

$$r = v + e_1$$

- The Hamming distance between v and r is

$$d(v, r) = w(v + r) = w(e_1).(3.24) \tag{23}$$

- The Hamming distance between w and r is

$$d(w, r) = w(w + r) = w(w + v + e_1) = w(e_2)(3.25) \tag{24}$$

- Now, suppose that the error pattern e_1 contains more than t errors [i.e. $w(e_1) \geq t + 1$].
- Since $2t + 1 \leq d_{min} 2t + 2$, it follows from (3.23) that

$$w(e_2) = d_{min} - w(e_1) \leq (2t + 2) - (t + 1) = t + 1$$

- Combining (3.24) and (3.25) and using the fact that $w(e1) \geq t + 1$ and $w(e2) \leq t + 1$, we have

- Combining (3.24) and (3.25) and using the fact that $w(e1) \geq t+1$ and $w(e2) \leq t+1$, we have

$$d(v,r) \geq d(w,r)$$

- Combining (3.24) and (3.25) and using the fact that $w(e1) \geq t + 1$ and $w(e2) \leq t + 1$, we have

$$d(v, r) \geq d(w, r)$$

- This inequality say that there exists an error pattern of $l(l > t)$ errors which results in a received vector that is closer to an incorrect code vector than to the transmitted code vector.

 Error-Detecting and Error-Correcting Capabilities of a Block Code

- Combining (3.24) and (3.25) and using the fact that $w(e1) \geq t+1$ and $w(e2) \leq t+1$, we have

$$d(v, r) \geq d(w, r)$$

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

- A block code with minimum distance d_{min} guarantees correcting all the error patterns of $t = [(d_{min} - 1)/2]$ or fewer errors, where $[(d_{min} - 1)/2]$ denotes the largest integer no greater than $(d_{min} - 1)/2$

- A block code with minimum distance d_{min} guarantees correcting all the error patterns of $t = [(d_{min} - 1)/2]$ or fewer errors, where $[(d_{min} - 1)/2]$ denotes the largest integer no greater than $(d_{min} - 1)/2$
- The parameter $t = [(d_{min} - 1)/2]$ is called the random-error correcting capability of the code

- A block code with minimum distance d_{min} guarantees correcting all the error patterns of $t = [(d_{min} - 1)/2]$ or fewer errors, where $[(d_{min} - 1)/2]$ denotes the largest integer no greater than $(d_{min} - 1)/2$
- The parameter $t = [(d_{min} - 1)/2]$ is called the random-error correcting capability of the code
- The code is referred to as a t-error-correcting code.

- A block code with minimum distance d_{min} guarantees correcting all the error patterns of $t = [(d_{min} - 1)/2]$ or fewer errors, where $[(d_{min} - 1)/2]$ denotes the largest integer no greater than $(d_{min} - 1)/2$
- The parameter $t = [(d_{min} - 1)/2]$ is called the random-error correcting capability of the code
- The code is referred to as a t-error-correcting code.
- A block code with random-error-correcting capability t is usually capable of correcting many error patterns of $t + 1$ or more errors.

- A block code with minimum distance d_{min} guarantees correcting all the error patterns of $t = [(d_{min} - 1)/2]$ or fewer errors, where $[(d_{min} - 1)/2]$ denotes the largest integer no greater than $(d_{min} - 1)/2$
- The parameter $t = [(d_{min} - 1)/2]$ is called the random-error correcting capability of the code
- The code is referred to as a t-error-correcting code.
- A block code with random-error-correcting capability t is usually capable of correcting many error patterns of $t + 1$ or more errors.
- For a t-error-correcting (n, k) linear code, it is capable of correcting a total 2^{n-k} error patterns.

Standard Array and Syndrome Decoding

- Let $V_1, V_2, V_3, \ldots, V_{2^k}$ be the code vector of C i.e $C = \{V_1, V_2, , \ldots, V_{2^k}\}$. Each code vector i.e for example $V_1 = (v_0, v_1, \ldots, v_{n-1})$
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \ldots, D_{2^k}$ such that the code vector v_i is contained in the subset D_i for $1 \leq i \leq 2^k$.
- Each subset D_i is one-to-one correspondence to a code vector v_i.
- If the received vector r is found in the subset D_i, r is decoded into v_i.
- Correct decoding is made if and only if the received vector r is in the subset D_i that corresponds to the actual code vector transmitted.

- A method to partition the 2^n possible received vectors into 2^k disjoint subsets such that each subset contains one and only one code vector is described here.

  Step 1.
- First, the 2^k code vectors of C are placed in a row with the all-zero code vector $v_1 = (0, 0, \ldots, 0)$ as the first (leftmost) element.

  $D_1, \quad D_2, \ldots, \quad D_i, \quad D_{2}k$

  $v_1 = (00...0) \quad v_2, \ldots, \quad v_i, \quad v_{2}k$

  Step 2.
- From the remaining $2^n - 2^k$ n-tuple, an n-tuple e_2 of minimum weight is chosen and is placed under the zero vector v_1.
- A second row is formed by adding e_2 to each code vector v_i in the first row and placing the sum $e_2 + v_i$ under v_i

Step 3.

- An unused n-tuple e_3 is chosen from the remaining n-tuples and is placed under e_2.
- Then a third row is formed by adding e_3 to each code vector v_i in the first row and placing $e_3 + v_i$ under v_i.
- Continue this process until all the n-tuples are used.
- Then we have an array of rows and columns as shown in Fig 3.6
- This array is called a standard array of the given linear code C

$$
\begin{array}{llllll}
v_1 = 0 & v_2 & \ldots & v_i & \ldots & v_{2^k} \\
e_2 & e_2 + v_2 & \ldots & e_2 + v_i & \ldots & e_2 + v_{2^k} \\
e_3 & e_3 + v_2 & \ldots & e_3 + v_i & \ldots & e_3 + v_{2^k} \\
\vdots & & & & & \\
e_l & e_l + v_2 & \ldots & e_l + v_i & \ldots & e_l + v_{2^k} \\
\vdots & & & & & \\
e_2^{n-k} & e_2^{n-k} + v_2 & \ldots & e_2^{n-k} + v_i & \ldots & e_2^{n-k} + v_{2^k}
\end{array}
$$

Theorem 3.3: *No two n-tuples in the same row of a standard array are identical. Every n-tuple appears in one and only one row.* **Proof:**

- The first part of the theorem follows from the fact that all the code vectors of C are distinct
- Suppose that two n-tuples in the l^{th} rows are identical, say $e_l + v_i = e_l + v_j$ with $i \neq j$
- This means that $v_i = v_j$, which is impossible, therefore no two n-tuples in the same row are identical

Proof

- It follows from the construction rule of the standard array that every n-tuple appears at least once
- Suppose that an n-tuple appears in both lth row and the mth row with $l < m$
- Then this n-tuple must be equal to el + vi for some i and equal to $e_m + v_j$ for some j
- As a result, $e_l + v_i = e_m + v_j$
- From this equality we obtain $em = el + (vi + vj)$
- Since vi and vj are code vectors in C, $vi + v$ is also a code vector in C, say vs
- This implies that the n-tuple em is in the lth row of the array, which contradicts the construction rule of the array that em, the first element of the mth row, should be unused in any previous row
- No n-tuple can appear in more than one row of the array

- From Theorem 3.3 we see that there are $2^n/2^k = 2^{n-k}$ disjoint rows in the standard array, and each row consists of 2^k distinct elements
- The 2_{n-k} rows are called the cosets of the code C
- The first n-tuple e_j of each coset is called a coset leader
- Any element in a coset can be used as its coset leader

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

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

Message is of

u_0	u_1	u_2
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The coded message is of $U.G =$
(000000, 011100,101010, 110001, 110110, 101101, 011011, 000111)
The standard array of this code is shown in Table.

The coded message is of $U.G =$ (000000, 011100,101010, 110001, 110110, 101101, 011011, 000111) The standard array of this code is shown in Table.

Coset leader							
000000	011100	101010	110001	110110	101101	011011	000111
100000	111100	001010	010001	010110	001101	111011	100111
010000	001100	111010	100001	100110	111101	001011	010111
001000	010100	100010	111001	111110	100101	010011	001111
000100	011000	101110	110101	110010	101001	011111	000011
000010	011110	101000	110011	110100	101111	011001	000101
000001	011101	101011	110000	110111	101100	011010	000110
100100	111000	001110	010101	010010	001001	111111	100011

Standard Array Decoding

- Consider (011100) is the transmitted codeword and the received word is (001100) which lies in 2^{nd} column whose coset leader e=(010000). So e is correctable error pattern. v=r+e= (001100)+(010000)=(011100).
- Consider (011100) is the transmitted codeword and the received word is (010100) which lies in 2^{nd} column whose coset leader e=(001000). So e is correctable error pattern. v=r+e= (010100)+(001000)=(011100).
- Consider (011100) is the transmitted codeword and the received word is (001010) which lies in 2^{nd} column whose coset leader e=(100000). v=r+e= (001010)+(100000)=(101010), in which there are 3 errors ocuur in the received vector that is equal to d_{min}, hence it is undetectable.
- Again consider (011100) is the transmitted codeword and the received word is (101100) and for this error pattern there is no coset leader in the standard array, so e is uncorrectable error pattern.

- A standard array of an (n, k) linear code C consists of 2^k disjoint columns
- Let Dj denote the jth column of the standard array, then

$$D_j = \{v_j, e_2 + v_j, e_3 + v_j, \ldots, e_{2^{n-k}} + v_j\} \qquad (3.27)$$

- v_j is a code vector of C and $e_2, e_3, \ldots e_{2^{n-k}}$ are the coset leaders
- The 2^k disjoint columns $D_1, D_2, \ldots, D_{2^k}$ can be used for decoding the code C.
- Suppose that the code vector v_j is transmitted over a noisy channel, from (3.27) we see that the received vector r is in D_j if the error pattern caused by the channel is a coset leader
- If the error pattern caused by the channel is not a coset leader, an erroneous decoding will result

- The decoding is correct if and only if the error pattern caused by the channel is a coset leader
- The 2^{n-k} coset leaders (including the zero vector 0) are called the correctable error patterns.
  Theorem 3.4 *Every (n, k) linear block code is capable of correcting 2^{n-k} error pattern.*
- To minimize the probability of a decoding error, the error patterns that are most likely to occur for a given channel should be chosen as the coset leaders
- When a standard array is formed, each coset leader should be chosen to be a vector of least weight from the remaining available vectors

Syndrome Decoding

- The syndrome of an n-tuple is an (n-k)-tuple and there are 2^{n-k} distinct (n-k)-tuples.
- From theorem 3.6 that there is a one-to-one correspondence between a coset and an (nk)-tuple syndrome
- Using this one-to-one correspondence relationship, we can form a decoding table, which is much simpler to use than a standard array
- The table consists of 2^{n-k} coset leaders (the correctable error pattern) and their corresponding syndromes
- This table is either stored or wired in the receiver

The decoding of a received vector consists of three steps:
Step 1.

The decoding of a received vector consists of three steps:
Step 1.

- Compute the syndrome S of the received word r,

$$S = r.H^T = H^T.r$$

The decoding of a received vector consists of three steps:
Step 1.

- Compute the syndrome S of the received word r,

$$S = r.H^T = H^T.r$$

Step 2.

- Locate the coset leader e_l whose syndrome is equal to $r.H^T$, then e_l is assumed to be the error pattern caused by the channel.

The decoding of a received vector consists of three steps:
Step 1.

- Compute the syndrome S of the received word r,

$$S = r.H^T = H^T.r$$

Step 2.

- Locate the coset leader e_l whose syndrome is equal to $r.H^T$, then e_l is assumed to be the error pattern caused by the channel.
  Step 3.
- Decode the received vector r into the code vector v. i.e., $v = r + e_l$
- The decoding scheme described above is called the syndrome decoding or table-lookup decoding

Example 3.8

- Consider the (7, 4) linear code given in Table 3.1, the parity-check matrix is given in example 3.3
- The code has $2^3 = 8$ cosets.
- There are eight correctable error patterns (including the all-zero vector)
- Since the minimum distance of the code is 3, it is capable of correcting all the error patterns of weight 1 or 0
- All the 7-tuples of weight 1 or 0 can be used as coset leaders.
- The number of correctable error pattern guaranteed by the minimum distance is equal to the total number of correctable error patterns.

Table: Decoding table for the (7,4) linear code.

Syndrome	**Coset Leader**
(100)	(1000000)
(010)	(0100000)
(001)	(0010000)
(110)	(0001000)
(011)	(0000100)
(111)	(0000010)
(101)	(0000001)

- Suppose that the code vector v = (1 0 0 1 0 1 1) is transmitted and r = (1 0 0 1 1 1 1) is received code vector.

- Suppose that the code vector v = (1 0 0 1 0 1 1) is transmitted and r = (1 0 0 1 1 1 1) is received code vector.
- For decoding r, we compute the syndrome of r.

- Suppose that the code vector v = (1 0 0 1 0 1 1) is transmitted and r = (1 0 0 1 1 1 1) is received code vector.
- For decoding r, we compute the syndrome of r.

$$S = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = (0\ 1\ 1)$$

- From Table 3.2 we find that (0 1 1) is the syndrome of the coset leader e = (0 0 0 0 1 0 0), then r is decoded into

- From Table 3.2 we find that (0 1 1) is the syndrome of the coset leader e = (0 0 0 0 1 0 0), then r is decoded into

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

- which is the actual code vector transmitted
- The decoding is correct since the error pattern caused by the channel is a coset leader.

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.
- We see that **two** errors have occurred during the transmission of v.

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.
- We see that **two** errors have occurred during the transmission of v.
- The error pattern is not correctable and will cause a decoding error.

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.
- We see that **two** errors have occurred during the transmission of v.
- The error pattern is not correctable and will cause a decoding error.
- When r is received, the receiver computes the syndrome.

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.
- We see that **two** errors have occurred during the transmission of v.
- The error pattern is not correctable and will cause a decoding error.
- When r is received, the receiver computes the syndrome.

$$s = r.H^T = (111)$$

- Suppose that v = (0 0 0 0 0 0 0) is transmitted and r = (1 0 0 0 1 0 0) is received code vector.
- We see that **two** errors have occurred during the transmission of v.
- The error pattern is not correctable and will cause a decoding error.
- When r is received, the receiver computes the syndrome.

$$s = r.H^T = (111)$$

- From the decoding table we find that the coset leader e = (0 0 0 0 0 1 0) corresponds to the syndrome s = (1 1 1).

- r is decoded into the code vector.

- r is decoded into the code vector.

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

- r is decoded into the code vector.

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

- Since v* is not the actual code vector transmitted, a decoding error is committed.

- r is decoded into the code vector.

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

- Since v* is not the actual code vector transmitted, a decoding error is committed.
- Using Table 3.2, the code is capable of correcting any single error over a block of seven digits.

- r is decoded into the code vector.

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

- Since v* is not the actual code vector transmitted, a decoding error is committed.
- Using Table 3.2, the code is capable of correcting any single error over a block of seven digits.
- When two or more errors occur, a decoding error will be committed.

- The table-lookup decoding of an (n, k) linear code may be implemented as follows.

- The table-lookup decoding of an (n, k) linear code may be implemented as follows.
- The decoding table is regarded as the truth table of n switch functions:

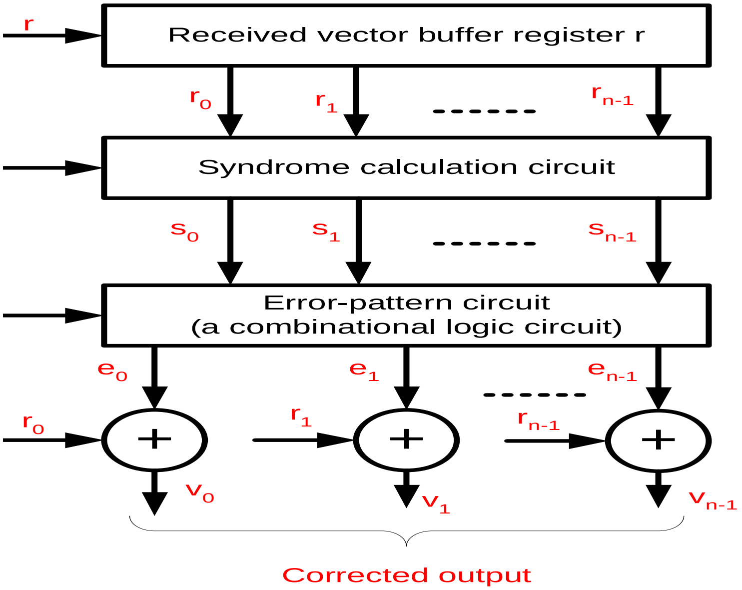
- The table-lookup decoding of an (n, k) linear code may be implemented as follows.
- The decoding table is regarded as the truth table of n switch functions:

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

- The table-lookup decoding of an (n, k) linear code may be implemented as follows.
- The decoding table is regarded as the truth table of n switch functions:

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

where $s_0, s_1, \ldots, s_{n-k-1}$ are the syndrome digits where $e_0, e_1, \ldots, e_{n-1}$ are the estimated error digits


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

Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.

Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.
- The decoding table is given by Table 3.2

Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.
- The decoding table is given by Table 3.2
- From this table we form the truth table (Table 3.3)

Example 3.9

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

Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.
- The decoding table is given by Table 3.2
- From this table we form the truth table (Table 3.3)
- The switching expression for the seven error digits are
- where $\wedge$ denotes the logic-AND operation

Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.
- The decoding table is given by Table 3.2
- From this table we form the truth table (Table 3.3)
- The switching expression for the seven error digits are
- where $\wedge$ denotes the logic-AND operation
- where $'$ denotes the logic-COMPLENENT of s

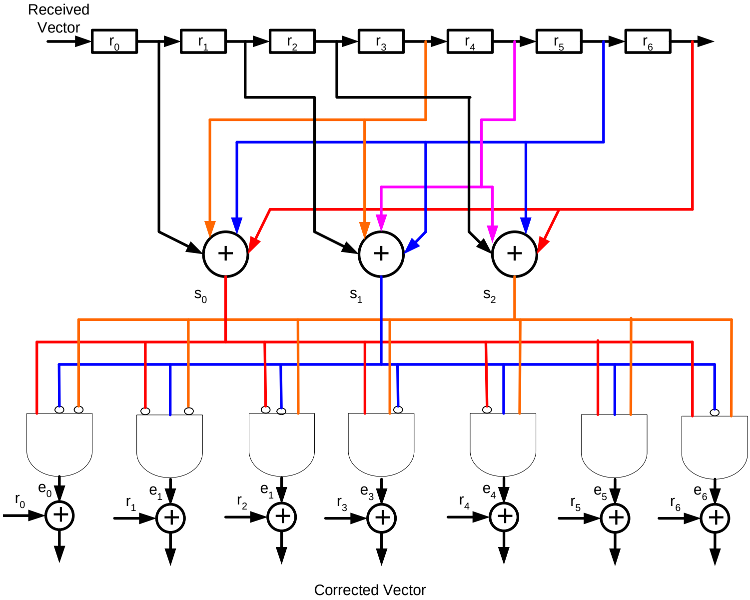
Example 3.9

- Consider the (7, 4) code given in Table 3.1. The syndrome circuit for this code is shown in Fig. 3.5.
- The decoding table is given by Table 3.2
- From this table we form the truth table (Table 3.3)
- The switching expression for the seven error digits are
- where $\wedge$ denotes the logic-AND operation
- where $'$ denotes the logic-COMPLENENT of s

$e_0 = s_0 \wedge s_1^{'} \wedge s_2^{'}$ $\quad e_1 = s_0^{'} \wedge s_1 \wedge s_2^{'}$ $\quad e_2 = s_0^{'} \wedge s_1^{'} \wedge s_2$

$e_3 = s_0 \wedge s_1 \wedge s_2^{'}$ $\quad e_4 = s_0^{'} \wedge s_1 \wedge s_2$ $\quad e_5 = s_0 \wedge s_1 \wedge s_2$

$e_6 = s_0 \wedge s_1^{'} \wedge s_2$


Received Vector
r0
r1
r2
r3
r4
r5
r6
s0
s1
s2
e0
e1
e2
e3
e4
e5
e6
Corrected Vector

Thank You

References

S. Lin and J. Daniel J. Costello, *Error Control Coding*, 2nd ed. Pearson/Prentice Hall, 2004.