Important Linear Block Codes [1, 2]

Manjunatha. P

manjup.jnnce@gmail.com

Professor

Dept. of ECE

J.N.N. College of Engineering, Shimoga

November 6, 2013

Important Linear Block Codes

Important Linear Block Codes

1. Hamming Codes

Important Linear Block Codes

1. Hamming Codes
    - Have minimum distance of 3 and capable of correcting any single error.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes

Important Linear Block Codes

1. Hamming Codes
    - Have minimum distance of 3 and capable of correcting any single error.
    - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
    - A large class of codes for multiple random error correction.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
   - A large class of codes for multiple random error correction.
   - Are simple in construction and rich in structural properties.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
   - A large class of codes for multiple random error correction.
   - Are simple in construction and rich in structural properties.
   - Decoding is done using hard or soft decision decoding algorithms.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
   - A large class of codes for multiple random error correction.
   - Are simple in construction and rich in structural properties.
   - Decoding is done using hard or soft decision decoding algorithms.
   - Soft decision decoding achieve very good error performance with low decoding complexity.

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
   - A large class of codes for multiple random error correction.
   - Are simple in construction and rich in structural properties.
   - Decoding is done using hard or soft decision decoding algorithms.
   - Soft decision decoding achieve very good error performance with low decoding complexity.
3. The (24, 12) Golay code

Important Linear Block Codes

1. Hamming Codes
   - Have minimum distance of 3 and capable of correcting any single error.
   - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
   - A large class of codes for multiple random error correction.
   - Are simple in construction and rich in structural properties.
   - Decoding is done using hard or soft decision decoding algorithms.
   - Soft decision decoding achieve very good error performance with low decoding complexity.
3. The (24, 12) Golay code
   - Used for error control in many communication systems.

Important Linear Block Codes

1. Hamming Codes
    - Have minimum distance of 3 and capable of correcting any single error.
    - Hamming codes can be decoded easily using a table-lookup scheme.
2. Reed Muller codes
    - A large class of codes for multiple random error correction.
    - Are simple in construction and rich in structural properties.
    - Decoding is done using hard or soft decision decoding algorithms.
    - Soft decision decoding achieve very good error performance with low decoding complexity.
3. The (24, 12) Golay code
    - Used for error control in many communication systems.
4. Product codes and Interleaved codes

Hamming Codes

- These codes and their variations have been widely used for error control in digital communication and data storage systems.

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
- Code length: $n = 2^m - 1$

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
- Code length: $n = 2^m - 1$
- Number of information symbols: $k = 2^m - m - 1$

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
- Code length: $n = 2^m - 1$
- Number of information symbols: $k = 2^m - m - 1$
- Number of parity-check symbols: $n - k = m$

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
- Code length: $n = 2^m - 1$
- Number of information symbols: $k = 2^m - m - 1$
- Number of parity-check symbols: $n - k = m$
- Error-correcting capability: $t = 1 (d_{min} = 3)$

- These codes and their variations have been widely used for error control in digital communication and data storage systems.
- For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters :
- Code length: $n = 2^m - 1$
- Number of information symbols: $k = 2^m - m - 1$
- Number of parity-check symbols: $n - k = m$
- Error-correcting capability: $t = 1(d_{min} = 3)$
- The parity-check matrix H of this code consists of all the nonzero m-tuple as its columns $(2^m - 1)$.

- In systematic form, the columns of H are arranged in the following form :

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- where I_m is an $m \times m$ identity matrix.

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- where I_m is an $m \times m$ identity matrix.
- The submatrix Q consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more.

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- where I_m is an $m \times m$ identity matrix.
- The submatrix Q consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more.
- The columns of Q may be arranged in any order without affecting the distance property and weight distribution of the code.

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- where I_m is an $m \times m$ identity matrix.
- The submatrix Q consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more.
- The columns of Q may be arranged in any order without affecting the distance property and weight distribution of the code.
- For example let m=3, the parity check matrix of a Hamming code of length 7 is in the form.

- In systematic form, the columns of H are arranged in the following form :

$$H = [I_m \quad Q]$$

- where I_m is an $m \times m$ identity matrix.
- The submatrix Q consists of $2^m - m - 1$ columns which are the m-tuples of weight 2 or more.
- The columns of Q may be arranged in any order without affecting the distance property and weight distribution of the code.
- For example let m=3, the parity check matrix of a Hamming code of length 7 is in the form.

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

- In systematic form, the generator matrix of the code is

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m - m - 1}]$$

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.
- Since the columns of H are nonzero and distinct, no two columns add to zero.

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.
- Since the columns of H are nonzero and distinct, no two columns add to zero.
- Since H consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say h_i and h_j, must also be a column in H, say h_l

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.
- Since the columns of H are nonzero and distinct, no two columns add to zero.
- Since H consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say h_i and h_j, must also be a column in H, say h_l

$$h_i + h_j + h_l = 0$$

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.
- Since the columns of H are nonzero and distinct, no two columns add to zero.
- Since H consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say h_i and h_j, must also be a column in H, say h_l

$$h_i + h_j + h_l = 0$$

- The minimum distance of a Hamming code is exactly 3

- In systematic form, the generator matrix of the code is

$$G = [Q^T \quad I_{2^m-m-1}]$$

- where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1)x(2^m - m - 1)$ identity matrix.
- Since the columns of H are nonzero and distinct, no two columns add to zero.
- Since H consists of all the nonzero m-tuples as its columns, the vector sum of any two columns, say h_i and h_j, must also be a column in H, say h_l

$$h_i + h_j + h_l = 0$$

- The minimum distance of a Hamming code is exactly 3
- The code is capable of correcting all the error patterns with a single error or of detecting all the error patterns of two or fewer errors.

- If we form the standard array for the Hamming code of length $2^m - 1$

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.
- The number of $(2^m - 1)$-tuples of weight 1 is $(2^m - 1)$

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.
- The number of $(2^m - 1)$-tuples of weight 1 is $(2^m - 1)$
- Since $n - k = m$, the code has 2^m cosets

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.
- The number of $(2^m - 1)$-tuples of weight 1 is $(2^m - 1)$
- Since $n - k = m$, the code has 2^m cosets
- The zero vector 0 and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.
- The number of $(2^m - 1)$-tuples of weight 1 is $(2^m - 1)$
- Since $n - k = m$, the code has 2^m cosets
- The zero vector 0 and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array
- A t-error-correcting code is called a perfect code if its standard array has all the error patterns of t or fewer errors and no others as coset leader

- If we form the standard array for the Hamming code of length $2^m - 1$
- All the $(2^m - 1)$-tuple of weight 1 can be used as coset leaders.
- The number of $(2^m - 1)$-tuples of weight 1 is $(2^m - 1)$
- Since $n - k = m$, the code has 2^m cosets
- The zero vector 0 and the $(2^m - 1)$-tuples of weight 1 form all the coset leaders of the standard array
- A t-error-correcting code is called a perfect code if its standard array has all the error patterns of t or fewer errors and no others as coset leader
- Decoding of Hamming codes can be accomplished easily with the table-lookup scheme

- We may delete any l columns from the parity-check matrix H of a Hamming code

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix H'

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix H'
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix H'
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:
- Code length: $n = 2^m - l - 1$

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix H'
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:
- Code length: $n = 2^m - l - 1$
- Number of information symbols: $k = 2^m - m - l - 1$

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix $H^{'}$
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:
- Code length: $n = 2^m - l - 1$
- Number of information symbols: $k = 2^m - m - l - 1$
- Number of parity-check symbols: $n - k = m$

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix $H^{'}$
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:
- Code length: $n = 2^m - l - 1$
- Number of information symbols: $k = 2^m - m - l - 1$
- Number of parity-check symbols: $n - k = m$
- Minimum distance: $d_{min} \geq 3$

- We may delete any l columns from the parity-check matrix H of a Hamming code
- This deletion results in an $m \times (2^m - l - 1)$ matrix $H^{'}$
- Using H' as a parity-check matrix, we obtain a shortened Hamming code with the following parameters:
- Code length: $n = 2^m - l - 1$
- Number of information symbols: $k = 2^m - m - l - 1$
- Number of parity-check symbols: $n - k = m$
- Minimum distance: $d_{min} \geq 3$
- If we delete columns from H properly, we may obtain a shortened Hamming code with minimum distance 4

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \quad Q^{'}]$$

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \quad Q^{'}]$$

Q consists of $2^{m-1} - m$ columns of odd weight.

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \quad Q^{'}]$$

Q consists of $2^{m-1} - m$ columns of odd weight.

- Since all columns of H have odd weight, no three columns add to zero.

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \;\; Q^{'}]$$

  Q consists of $2^{m-1} - m$ columns of odd weight.

- Since all columns of H have odd weight, no three columns add to zero.
- However, for a column h_i of weight 3 in $Q^{'}$, there exists three columns h_j, h_l, and h_s in I_m such that $hi + h_j + h_l + h_s = 0$.

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \;\; Q^{'}]$$

Q consists of $2^{m-1} - m$ columns of odd weight.

- Since all columns of H have odd weight, no three columns add to zero.
- However, for a column h_i of weight 3 in $Q^{'}$, there exists three columns h_j, h_l, and h_s in I_m such that $hi + h_j + h_l + h_s = 0$.
- Thus, the shortened Hamming code with H as a parity-check matrix has minimum distance exactly 4.

- For example, if we delete from the submatrix Q all the columns of even weight, we obtain an $mx2^{m-1}$ matrix.

$$H^{'} = [I_m \;\; Q^{'}]$$

Q consists of $2^{m-1} - m$ columns of odd weight.

- Since all columns of H have odd weight, no three columns add to zero.
- However, for a column h_i of weight 3 in $Q^{'}$, there exists three columns h_j, h_l, and h_s in I_m such that $hi + h_j + h_l + h_s = 0$.
- Thus, the shortened Hamming code with H as a parity-check matrix has minimum distance exactly 4.
- The distance 4 shortened Hamming code can be used for correcting all error patterns of single error and simultaneously detecting all error patterns of double errors

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^{T} corresponds to a column in H)

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^T corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s ($exH^{'T}$ corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s
- Decoding can be accomplished in the following manner :

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^T corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s
- Decoding can be accomplished in the following manner :
  1. If the syndrome s is zero, we assume that no error occurred

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^T corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s
- Decoding can be accomplished in the following manner :
  1. If the syndrome s is zero, we assume that no error occurred
  2. If s is nonzero and it contains odd number of 1's, we assume that a single error occurred.

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^T corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s
- Decoding can be accomplished in the following manner :
  1. If the syndrome s is zero, we assume that no error occurred
  2. If s is nonzero and it contains odd number of 1's, we assume that a single error occurred.
  3. The error pattern of a single error that corresponds to s is added to the received vector for error correction

- When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1s (exH'^T corresponds to a column in H)
- When double errors occurs, the syndrome is nonzero, but it contains even number of 1s
- Decoding can be accomplished in the following manner :
  1. If the syndrome s is zero, we assume that no error occurred
  2. If s is nonzero and it contains odd number of 1's, we assume that a single error occurred.
  3. The error pattern of a single error that corresponds to s is added to the received vector for error correction
  4. If s is nonzero and it contains even number of 1's, an uncorrectable error pattern has been detected

Reed-Muller Codes

- Reed-Muller codes are among the oldest known codes and have found widespread applications.

- Reed-Muller codes are among the oldest known codes and have found widespread applications.
- They were discovered by Muller and provided with a decoding algorithm by Reed in 1954.

- Reed-Muller codes are among the oldest known codes and have found widespread applications.
- They were discovered by Muller and provided with a decoding algorithm by Reed in 1954.
- These codes were initially given as binary codes, but modern generalizations to q-ary codes exist.

- Reed-Muller codes are among the oldest known codes and have found widespread applications.
- They were discovered by Muller and provided with a decoding algorithm by Reed in 1954.
- These codes were initially given as binary codes, but modern generalizations to q-ary codes exist.
- Reed-Muller codes have many interesting properties that are worth examination; they form an infinite family of codes, and larger Reed-Muller codes can be constructed from smaller ones.

- Reed-Muller codes are among the oldest known codes and have found widespread applications.
- They were discovered by Muller and provided with a decoding algorithm by Reed in 1954.
- These codes were initially given as binary codes, but modern generalizations to q-ary codes exist.
- Reed-Muller codes have many interesting properties that are worth examination; they form an infinite family of codes, and larger Reed-Muller codes can be constructed from smaller ones.
- This particular observation leads us to show that Reed-Muller codes can be defined recursively.

- Reed-Muller codes are among the oldest known codes and have found widespread applications.
- They were discovered by Muller and provided with a decoding algorithm by Reed in 1954.
- These codes were initially given as binary codes, but modern generalizations to q-ary codes exist.
- Reed-Muller codes have many interesting properties that are worth examination; they form an infinite family of codes, and larger Reed-Muller codes can be constructed from smaller ones.
- This particular observation leads us to show that Reed-Muller codes can be defined recursively.
- One of the major advantages of Reed-Muller codes is their relative simplicity to encode messages and decode received transmissions.

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$
- Dimension: $k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$
- Dimension: $k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$
- Minimum distance: $d_{min} = 2^{m-r}$

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$
- Dimension: $k(r, m) = 1 + \left(\begin{array}{c} m \\ 1 \end{array} \right) + \left(\begin{array}{c} m \\ 2 \end{array} \right) + \ldots + \left(\begin{array}{c} m \\ r \end{array} \right)$
- Minimum distance: $d_{min} = 2^{m-r}$
- where $\left(\begin{array}{c} m \\ i \end{array} \right) = \frac{m!}{i!(m-i)!}$

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$
- Dimension: $k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$
- Minimum distance: $d_{min} = 2^{m-r}$
- where $\binom{m}{i} = \frac{m!}{i!(m-i)!}$
- Example m=5 and r=2 then n=32, k(2,5)=16 and $d_{min} = 8$

- For any integers m and r with $0 \leq r \leq m$ there exist a binary rth order RM code, denoted by RM(r,m), with the following parameters
- Code length: $n = 2^m$
- Dimension: $k(r,m) = 1 + \begin{pmatrix} m \\ 1 \end{pmatrix} + \begin{pmatrix} m \\ 2 \end{pmatrix} + \ldots + \begin{pmatrix} m \\ r \end{pmatrix}$
- Minimum distance: $d_{min} = 2^{m-r}$
- where $\begin{pmatrix} m \\ i \end{pmatrix} = \frac{m!}{i!(m-i)!}$
- Example m=5 and r=2 then n=32, k(2,5)=16 and $d_{min} = 8$
- There exists a (32, 16) RM code with a distance of 8.
- For $1 \leq i \leq m$ let V_i be 2^m tuple over $GF(2)$ of the following form:

$$V_i = (\underbrace{0\ldots0}_{2^{i-1}}, \underbrace{1\ldots1}_{2^{i-1}}, \underbrace{0\ldots0}_{2^{i-1}} \ldots \underbrace{1\ldots1}_{2^{i-1}})$$

which consists of 2^{m-i+1} alternating all zero and all one 2^{i-1} tuples.

Example 1

- Consider the code R(1,3) with generator matrix:

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r, m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-1} = 4$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-1} = 4$

$$\mathrm{G_{RM}}(1,3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r, m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-1} = 4$

$$\mathrm{G_{RM}}(1,3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Let $a = (a_0, a_1, a_2, \ldots a_{n-1})$ $b = (b_0, b_1, b_2, \ldots b_{n-1})$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-1} = 4$

$$\mathrm{G_{RM}}(1,3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Let $a = (a_0, a_1, a_2, \ldots a_{n-1})$ $b = (b_0, b_1, b_2, \ldots b_{n-1})$
- $a.b = (a_0.b_0, a_1.b_1, \ldots a_{n-1}.b_{n-1})$

Example 1

- Consider the code R(1,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-1} = 4$

$$\mathrm{G_{RM}}(1,3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Let $a = (a_0, a_1, a_2, \ldots a_{n-1})$ $b = (b_0, b_1, b_2, \ldots b_{n-1})$
- $a.b = (a_0.b_0, a_1.b_1, \ldots a_{n-1}.b_{n-1})$
- where . denotes logic product i.e. $a_i.b_i = 1$ if and if only $a_i = b_i = 1$

- The rth order RM code, RM(r, m) of length 2^m is generated by the following set of independent vectors:

$$\begin{aligned} G_{RM}(r,m) &= (V_0, V_1, V_2, \ldots V_m, V_1.V_2, V_1.V_3, V_{m-1}.V_m \\ &= \textit{up to products of degree } r) \end{aligned}$$

- There are

$$k(r,m) = 1 + \begin{pmatrix} m \\ 1 \end{pmatrix} + \begin{pmatrix} m \\ 2 \end{pmatrix} + \ldots + \begin{pmatrix} m \\ r \end{pmatrix}$$

- vectors in $G_{RM}(r,m)$. Therefore the dimension of the code is $k(r,m)$
- The vectors in $G_{RM}(r,m)$ are arranged as rows of a matrix, then the matrix is a generator matrix of the $RM(r,m)$ code. Hence $G_{RM}(r,m)$ is called as the generator matrix

Example 2

- Consider the code R(2,3) with generator matrix:

Example 2

- Consider the code R(2,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$

Example 2

- Consider the code R(2,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} + \frac{3.2.1}{(1)2.1} = 7$

Example 2

- Consider the code R(2,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} + \frac{3.2.1}{(1)2.1} = 7$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-2} = 2$

Example 2

- Consider the code R(2,3) with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r, m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} + \frac{3.2.1}{(1)2.1} = 7$
- Minimum distance: $d_{min} = 2^{m-r} = 2^{3-2} = 2$

$$\mathrm{G_{RM}}(2,3) = \begin{bmatrix} V_0 \\ V_3 \\ V_2 \\ V_1 \\ V_3.V_2 \\ V_3.V_1 \\ V_2.V_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$V_3.V_2$=(1 1 0 0 0 0 0 0)

Reed Decoding

- Consider an example R(1,3) code with generator matrix:

- Consider an example R(1,3) code with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$

- Consider an example R(1,3) code with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$

$$G_{RM} = \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- The rows of the matrix are labeled as $V_0, V_1, V_2 \ \ and \ \ V_3$.
- Consider a message $m = (a_0, a_1, a_2, a_3)$ to be encoded. $V = m * G_{RM}(1;3) = V = a_0 V_0 + a_1 V_1 + a_2 V_2 + a_3 V_3$.

- Consider an example R(1,3) code with generator matrix:
- Code length: $n = 2^m = 2^3 = 8$
- $k(r,m) = 1 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = 1 + \frac{3.2.1}{(2.1).1} = 4$

$$G_{RM} = \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- The rows of the matrix are labeled as V_0, V_1, V_2 *and* V_3.
- Consider a message $m = (a_0, a_1, a_2, a_3)$ to be encoded.
  $V = m * G_{RM}(1;3) = V = a_0 V_0 + a_1 V_1 + a_2 V_2 + a_3 V_3$.
- Written as a vector,
  $V = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

Reed Decoding

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3).$

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

- If no errors occur, a received vector $r = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ can be used to solve for the a_i other than a_0 in several ways (4 ways for each) namely:

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

- If no errors occur, a received vector $r = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ can be used to solve for the a_i other than a_0 in several ways (4 ways for each) namely:
  $a_0 = y_0$ $a_0 + a_1 = y_1 \Rightarrow y_0 + y_1 = a_1$
  $a_0 + a_2 = y_2$ $a_0 + a_1 + a_2 = y_3 \Rightarrow y_2 + y_3 = a_1$
  $a_0 + a_3 = y_4$ $a_0 + a_1 + a_3 = y_5 \Rightarrow y_4 + y_5 = a_1$
  $a_0 + a_2 + a_3 = y_6$ $a_0 + a_1 + a_2 + a_3 = y_7 \Rightarrow y_6 + y_7 = a_1$

- Therefore a_1 is
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

- If no errors occur, a received vector $r = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ can be used to solve for the a_i other than a_0 in several ways (4 ways for each) namely:
  $a_0 = y_0$ $a_0 + a_1 = y_1 \Rightarrow y_0 + y_1 = a_1$
  $a_0 + a_2 = y_2$ $a_0 + a_1 + a_2 = y_3 \Rightarrow y_2 + y_3 = a_1$
  $a_0 + a_3 = y_4$ $a_0 + a_1 + a_3 = y_5 \Rightarrow y_4 + y_5 = a_1$
  $a_0 + a_2 + a_3 = y_6$ $a_0 + a_1 + a_2 + a_3 = y_7 \Rightarrow y_6 + y_7 = a_1$
- Therefore a_1 is
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$
- Similarly a_2 and a_3 are determined and are as follows

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

- If no errors occur, a received vector $r = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ can be used to solve for the a_i other than a_0 in several ways (4 ways for each) namely:
  $a_0 = y_0 \ a_0 + a_1 = y_1 \Rightarrow y_0 + y_1 = a_1$
  $a_0 + a_2 = y_2 \ a_0 + a_1 + a_2 = y_3 \Rightarrow y_2 + y_3 = a_1$
  $a_0 + a_3 = y_4 \ a_0 + a_1 + a_3 = y_5 \Rightarrow y_4 + y_5 = a_1$
  $a_0 + a_2 + a_3 = y_6 \ a_0 + a_1 + a_2 + a_3 = y_7 \Rightarrow y_6 + y_7 = a_1$
- Therefore a_1 is
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$
- Similarly a_2 and a_3 are determined and are as follows
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$
  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- If one error has occurred in r, then when all the calculations above are made, 3 of the 4 values will agree for each a_i, so the correct value will be obtained by majority decoding.

Reed Decoding

$v = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$.

- If no errors occur, a received vector $r = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ can be used to solve for the a_i other than a_0 in several ways (4 ways for each) namely:
  $a_0 = y_0$ $a_0 + a_1 = y_1 \Rightarrow y_0 + y_1 = a_1$
  $a_0 + a_2 = y_2$ $a_0 + a_1 + a_2 = y_3 \Rightarrow y_2 + y_3 = a_1$
  $a_0 + a_3 = y_4$ $a_0 + a_1 + a_3 = y_5 \Rightarrow y_4 + y_5 = a_1$
  $a_0 + a_2 + a_3 = y_6$ $a_0 + a_1 + a_2 + a_3 = y_7 \Rightarrow y_6 + y_7 = a_1$

- Therefore a_1 is
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$
- Similarly a_2 and a_3 are determined and are as follows
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$
  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- If one error has occurred in r, then when all the calculations above are made, 3 of the 4 values will agree for each a_i, so the correct value will be obtained by majority decoding.
- Finally, a_0 can be determined as the majority of the components of $r + a_1 v_1 + a_2 v_2 + a_3 v_3$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$,
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$

- Calculate $a_1, a_2,$ and a_3 using majority decoding.:

  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$,

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:

  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$

  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$

- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:

  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$

  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$

  $a_3 = 0 = 1 = 1 = 1$ so $a_3 = 1$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7$,
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$
  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:
  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$
  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$
  $a_3 = 0 = 1 = 1 = 1$ so $a_3 = 1$
- Finally, a_0 can be determined as the majority of the components of $r + a_1 v_1 + a_2 v_2 + a_3 v_3$

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,

  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$

  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$

  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:

  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$

  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$

  $a_3 = 0 = 1 = 1 = 1$ so $a_3 = 1$
- Finally, a_0 can be determined as the majority of the components of $r + a_1 v_1 + a_2 v_2 + a_3 v_3$
- and $y_0 = a_0$ $y_1 = a_0 + a_1$ hence $a_0 = 1$ since 10101101 + 01010101 + 00001111 = 11110111.

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$
  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,\ and\ a_3$ using majority decoding.:
  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$
  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$
  $a_3 = 0 = 1 = 1 = 1$ so $a_3 = 1$
- Finally, a_0 can be determined as the majority of the components of $r + a_1v_1 + a_2v_2 + a_3v_3$
- and $y_0 = a_0\ y_1 = a_0 + a_1$ hence $a_0 = 1$ since 10101101 + 01010101 + 00001111 = 11110111.
- $v = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3$. In this case $a_2 = 0$ Therefore

Example

- Suppose that the transmitted code is v = 10100101 and received code as 10101101. Using,
  $a_1 = y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7,$
  $a_2 = y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7$
  $a_3 = y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7$
- Calculate $a_1, a_2,$ *and* a_3 using majority decoding.:
  $a_1 = 1 = 1 = 0 = 1$ so $a_1 = 1$
  $a_2 = 0 = 0 = 1 = 0$ so $a_2 = 0$
  $a_3 = 0 = 1 = 1 = 1$ so $a_3 = 1$
- Finally, a_0 can be determined as the majority of the components of $r + a_1 v_1 + a_2 v_2 + a_3 v_3$
- and $y_0 = a_0 \; y_1 = a_0 + a_1$ hence $a_0 = 1$ since 10101101 + 01010101 + 00001111 = 11110111.
- $v = a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3$. In this case $a_2 = 0$ Therefore
- V = 11111111 + 01010101 + 00001111 = 10100101.

Other Constructions of Reed-Muller Codes:

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).
- The Kronecker product of A and B denoted by $A \otimes B$ is the $mnxmn$ matrix obtained from A by replacing every entry a_{ij} with the matrix $a_{ij}B$.

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).
- The Kronecker product of A and B denoted by $A \otimes B$ is the $mnxmn$ matrix obtained from A by replacing every entry a_{ij} with the matrix $a_{ij}B$.
- If $a_{ij} = 1$ then $a_{ij}B = B$ and for $a_{ij} = 0$ then $a_{ij}B$ is an nxn zero matrix.

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).
- The Kronecker product of A and B denoted by $A \otimes B$ is the $mnxmn$ matrix obtained from A by replacing every entry a_{ij} with the matrix $a_{ij}B$.
- If $a_{ij} = 1$ then $a_{ij}B = B$ and for $a_{ij} = 0$ then $a_{ij}B$ is an nxn zero matrix.
- Generator matrix of $2x2$ over Galois field GF(2)is:

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).
- The Kronecker product of A and B denoted by $A \otimes B$ is the $mnxmn$ matrix obtained from A by replacing every entry a_{ij} with the matrix $a_{ij}B$.
- If $a_{ij} = 1$ then $a_{ij}B = B$ and for $a_{ij} = 0$ then $a_{ij}B$ is an nxn zero matrix.
- Generator matrix of $2x2$ over Galois field GF(2)is:

$$G_{(2,2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Other Constructions of Reed-Muller Codes:

- Let $A = [a_{ij}]$ be and mXm matrix and $B = [b_{ij}]$ be an nxn matrix over GF(2).
- The Kronecker product of A and B denoted by $A \otimes B$ is the $mnxmn$ matrix obtained from A by replacing every entry a_{ij} with the matrix $a_{ij}B$.
- If $a_{ij} = 1$ then $a_{ij}B = B$ and for $a_{ij} = 0$ then $a_{ij}B$ is an nxn zero matrix.
- Generator matrix of 2x2 over Galois field GF(2)is:

$$G_{(2,2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

- The two fold Kronecker product of $G_{(2,2)}$ is:

$$G_{(2^2,2^2)} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The three fold Kronecker product of $G_{(2,2)}$ is:

The three fold Kronecker product of $G_{(2,2)}$ is:

$$G_{(2^3,2^3)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

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

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

The four fold Kronecker product of $G_{(2,2)}$ is:

The four fold Kronecker product of $G_{(2,2)}$ is:

$$G_{(2^4,2^4)} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

The (24, 12) Golay Code

- Golay code constructed by M.J.E. Golay in 1949.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.
- Has abundant and beautiful algebraic structure.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.
- Has abundant and beautiful algebraic structure.
- The (23, 12) Golay code can be extended by adding an overall parity-check bit to each codework.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.
- Has abundant and beautiful algebraic structure.
- The (23, 12) Golay code can be extended by adding an overall parity-check bit to each codework.
- This extension results in a (24, 12) code with minimum distance of 8.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.
- Has abundant and beautiful algebraic structure.
- The (23, 12) Golay code can be extended by adding an overall parity-check bit to each codework.
- This extension results in a (24, 12) code with minimum distance of 8.
- This code is capable of correcting all errors of there or fewer errors, and detecting all error patterns of four errors.

- Golay code constructed by M.J.E. Golay in 1949.
- Has a minimum distance of 7 and is capable of correcting any combination of three or fewer random error in the block of 23 digits.
- Has abundant and beautiful algebraic structure.
- The (23, 12) Golay code can be extended by adding an overall parity-check bit to each codework.
- This extension results in a (24, 12) code with minimum distance of 8.
- This code is capable of correcting all errors of there or fewer errors, and detecting all error patterns of four errors.
- It is not a perfect code anymore however, it has many interesting structural properties.

- A generator matrix in systematic form for this code is as follows:

- A generator matrix in systematic form for this code is as follows:

$$G = [P \quad I_{12}]$$

- A generator matrix in systematic form for this code is as follows:

$$G = [P \quad I_{12}]$$

- where $I_{12}$is the identity matrix of dimension 12 and P is:

- A generator matrix in systematic form for this code is as follows:

$$G = [P \quad I_{12}]$$

- where $I_{12}$is the identity matrix of dimension 12 and P is:

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

- The P matrix has the following properties:

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P
  4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.

- The P matrix has the following properties:
    1. It is symmetrical with respect to its diagonal
    2. The i^{th} column is the transpose of the i^{th} row
    3. $P.P^T = I_{12}$ where P^T is the transpose of P
    4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.
    5. It follows from the second property that

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P
  4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.
  5. It follows from the second property that

$$P^T = P$$

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P
  4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.
  5. It follows from the second property that

$$P^T = P$$

- Consequently the parity check matrix in systematic form for the (24, 12)

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P
  4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.
  5. It follows from the second property that

$$P^T = P$$

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

- The P matrix has the following properties:
  1. It is symmetrical with respect to its diagonal
  2. The i^{th} column is the transpose of the i^{th} row
  3. $P.P^T = I_{12}$ where P^T is the transpose of P
  4. The sub matrix obtained by deleting the last row and last column is formed by cyclically shifting the first row to the left 11 times.
  5. It follows from the second property that

$$P^T = P$$

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

$$\begin{aligned} H &= [I_{12} \;\; P^T] \\ H &= [I_{12} \;\; P] \end{aligned}$$

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.
4. Compute $s \bullet P$.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.
4. Compute $s \bullet P$.
5. If $w(s \bullet P) = 2$ or 3, then we set $e* = (0; s \bullet P)$. And go to step 8.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.
4. Compute $s \bullet P$.
5. If $w(s \bullet P) = 2$ or 3, then we set $e* = (0; s \bullet P)$. And go to step 8.
6. If $w(s \bullet P + p_i) = 2$ for some p_i, then we set $e* = (u^{(i)}; sP + p_i)$

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.
4. Compute $s \bullet P$.
5. If $w(s \bullet P) = 2$ or 3, then we set $e* = (0; s \bullet P)$. And go to step 8.
6. If $w(s \bullet P + p_i) = 2$ for some p_i, then we set $e* = (u^{(i)}; sP + p_i)$
7. Otherwise, with $s \neq 0$, declare an uncorrectable error pattern.

Decoding Algorithm:

- Denote p_i to be the i^{th} row of P, and u(i) to be the 12-tuple in which only the i^{th} component is nonzero.
- The decoding algorithm consists of the following steps:

1. Compute the syndrome $s = r \bullet H^T$.
2. If $w(s) \leq 3$, then we set $e* = (s, 0)$. And go to step 8.
3. If $w(s + p_i) \leq 2$ for some row p_i in P, then we set $e* = (s + pi, u^{(i)})$. And go to step 8.
4. Compute $s \bullet P$.
5. If $w(s \bullet P) = 2$ or 3, then we set $e* = (0; s \bullet P)$. And go to step 8.
6. If $w(s \bullet P + p_i) = 2$ for some p_i, then we set $e* = (u^{(i)}; sP + p_i)$
7. Otherwise, with $s \neq 0$, declare an uncorrectable error pattern.
8. $v* = r + e*$:

Example:

- Suppose the (24,12) Golay code is used error control.

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r
- $s = r \bullet H^T = (111011111100)$

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r
- $s = r \bullet H^T = (111011111100)$
- Because $w(s) > 3$, go to step 3. We find that
- $s + p_{11} = (111011111100) + (111111111110) = (000100000010)$

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r
- $s = r \bullet H^T = (111011111100)$
- Because $w(s) > 3$, go to step 3. We find that
- $s + p_{11} = (111011111100) + (111111111110) = (000100000010)$
- and $s + p_{11} = 2$ So set $e = (s + p_{11}, u^{(11)}) = (000100000010000000000001)$

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r
- $s = r \bullet H^T = (111011111100)$
- Because $w(s) > 3$, go to step 3. We find that
- $s + p_{11} = (111011111100) + (111111111110) = (000100000010)$
- and $s + p_{11} = 2$ So set $e = (s + p_{11}, u^{(11)}) = (000100000010000000000001)$
- and decode r into as

Example:

- Suppose the (24,12) Golay code is used error control.
- Let r=(1 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1) received sequence.
- To decode r, compute S of r
- $s = r \bullet H^T = (111011111100)$
- Because $w(s) > 3$, go to step 3. We find that
- $s + p_{11} = (111011111100) + (111111111110) = (000100000010)$
- and $s + p_{11} = 2$ So set $e = (s + p_{11}, u^{(11)}) = (000100000010000000000001)$
- and decode r into as
- $v* = r + e = (100100110110110000000000)$

Product Codes

- Let C_1 be an (n_1, k_1) linear code and C_2 be an (n_2, k_2) linear code.
- Then an $(n_1 n_2, k_1 k_2)$ linear code is formed such that each codeword is rectangular array of (n_1) columns and (n_2) rows in which every row is codeword in C_1 and every column is codeword in C_2.
- This two dimensional codeword is called direct product of C_1 and C_2.
- The (k_1, k_2) digits in the right corner of the array are information symbols.
- The (k_1, k_2) digits in the upper right corner of the array are information symbols.
- The digits in the upper left corner of the array are computed from the parity check rules for C_1 on rows and the digits in the lower right corner are computed from the parity check rules for C_2 on columns.
- The digits in the lower left corner of the array are parity check rules for C_2 on columns or parity check rules for C_1 on rows.

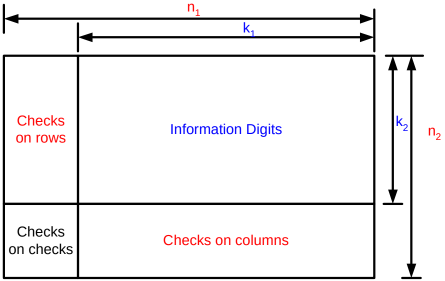

Figure: Code array for product code

- The product code $C_1 X C_2$ is encoded in two steps.
- A message of (k_1, k_2) information symbols is first arranged as shown in the upper right corner of Figure 2
  1. In the first step each row of the information array is encoded into a codeword in C_1. The encoded results an array of (k_2) rows and (n_1) columns as shown in the upper part of the the Figure.
  2. In the second step of encoding each of the n_1 columns of the array formed at the first encoding step is encoded into a codeword in C_2.
- This results in a code array of (n_2) rows and (n_1) columns as shown in Figure 2.
- The code array is also can be formed by first performing the column by column encoding and then the row encoding.
- Transmission can be carried out either column by column or row by row.

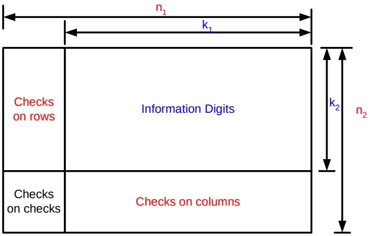

Figure: Code array for product code

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

- Consider an example u=(1 0 1 1 0 0 0 1 0 1 0 1 1 1 0 1)
- This can be arranged as 4X4 information array.
- The first four information symbols form the first row of the information array the second four information symbols form the second row and so on.
- In the first step of encoding a single (even) parity check symbol is added to each row of the information array. This results in a 4X5 array.
- In the first step of encoding a single (even) parity check symbol is added to each the five columns of the array. This results in a 5X5 array.
- At the receiver a single error occurs at the intersection of two and column.
- The erroneous row and column corrected by complementing the received symbol at the intersection.
- Parity failure cannot correct any double error pattern, but it can detect all the double error pattern
- When a double error pattern occurs, there are 3 possible distribution of the two errors: (1) they are in the same row (2)

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

Interleaved Codes[2]

- In data manipulation and transmission, errors may be caused by a variety of factors including noise corruption, limited channel bandwidth, and interference between channels and sources.
- Bursts (or clusters) of errors are defined as a group of consecutive error bits in the one-dimensional (1-D) case or connected error bits in multi-dimensional (M-D) cases.
- Several consecutive transmitted error bits in a mobile communication system caused by a multipath fading channel.
- A bursty channel is defined as a channel over which errors tend to occur in bunches, or “bursts,” as opposed to random patterns associated with a Bernoulli-distributed process.
- The main idea is to mix up the code symbols from different code-words so that when the code-words are reconstructed at the receiving end error bursts encountered in the transmission are spread across multiple codewords.
- Consequently, the errors occurred within one code-word may be small enough to be corrected by using a simple random error correction code.

- Consider a code in which each code-word contains four code symbols[2].
- Suppose there are 16 symbols, which correspond to four code-words.
- That is, code symbols from 1 to 4 form a code-word, from 5 to 8 another codeword, and so on.
- In block interleaving, first creates a 4X4 2-D array, called block interleaver as shown in Figure 1.
- The 16 code symbols are read into the 2-D array in a column-by-column (or row-by-row) manner.
- The interleaved code symbols are obtained by writing the code symbols out of the 2-D array in a row-by-row (or column by-column) fashion.
- This process has been depicted in Figure 1 (a), (b), and (c).
- Assume a burst of errors involving four consecutive symbols as shown in Figure 1 (c) with shades.
- After de-interleaving as shown in Figure 1 (d), the error burst is effectively spread among four code-words, resulting in only one code symbol in error for each of the four code-words

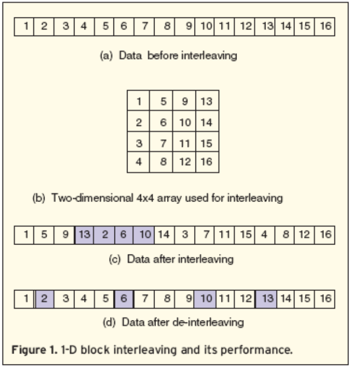

Figure: Block Interleaving

[2]

- Consider a (n,k) linear block code C, a new $(\lambda n, \lambda k)$ linear code is constructed by interleaving, that is arranging λ codewords in C into λ rows of rectangular array and then transmitting the array column by column.
- The resulting code denoted by C^{λ} is called and interleaved code.
- The parameter is referred as interleaving depth.
- The interleaving technique is effective for deriving long, powerful codes for correcting errors that cluster to form bursts.

Thank You

References

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

Y. Q. Shi, X. M. Zhang, Z.-C. Ni, and N. Ansari, "Interleaving for combating bursts of errors," *IEEE Circuits And Systems Magazine*, pp. 29–42, 2004.