Linear and Systematic Block Codes

The parity bits of linear block codes are linear combination of the message. Therefore, we can represent the encoder by a linear system described by matrices.

Basic Definitions

- Linearity:

$$\text{If} \quad \mathbf{m}_1 \rightarrow \mathbf{c}_1 \text{ and } \mathbf{m}_2 \rightarrow \mathbf{c}_2$$
$$\text{then } \mathbf{m}_1 \oplus \mathbf{m}_2 \rightarrow \mathbf{c}_1 \oplus \mathbf{c}_2$$

where **m** is a k-bit information sequence

c is an n-bit codeword.

$\oplus$ is a bit-by-bit mod-2 addition without carry

- Linear code: The sum of any two codewords is a codeword.
- Observation: The all-zero sequence is a codeword in every

linear block code.

Basic Definitions (cont'd)

- Def: The weight of a codeword $\mathbf{c}_i$, denoted by $w(\mathbf{c}_i)$, is the number of of nonzero elements in the codeword.
- Def: The minimum weight of a code, $w_{\min}$, is the smallest weight of the nonzero codewords in the code.
- Theorem: In any linear code, $d_{\min} = w_{\min}$
- Systematic codes

n-k check bits	k information bits

Any linear block code can be put in systematic form

linear Encoder.

By linear transformation

$c = m \cdot G = m_0 g_0 + m_1 g_0 + \ldots\ldots + m_{k-1} g_{k-1}$

The code *C* is called a *k*-dimensional subspace.

G is called a generator matrix of the code.

Here *G* is a $k \times n$ matrix of rank *k* of elements from GF(2), g_i is the *i*-th row vector of *G*.

The rows of *G* are linearly independent since *G* is assumed to have rank *k*.

Example:

(7, 4) Hamming code over GF(2)

The encoding equation for this code is given by

$$c_0 = m_0$$
$$c_1 = m_1$$
$$c_2 = m_2$$
$$c_3 = m_3$$
$$c_4 = m_0 + m_1 + m_2$$
$$c_5 = m_1 + m_2 + m_3$$
$$c_6 = m_0 + m_1 + m_3$$

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

Linear Systematic Block Code:

An (*n, k*) *linear systematic code is completely specified by a k* × *n* generator matrix of the following form.

$$G = \begin{bmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} = [I_k P]$$

where I_k *is the k* × *k identity matrix.*

Linear Block Codes

- the number of codeworde is 2^k since there are 2^k distinct messages.
- The set of vectors $\{g_i\}$ are linearly independent since we must have a set of unique codewords.
- linearly independent vectors mean that no vector g_i can be expressed as a linear combination of the other vectors.
- These vectors are called baises vectors of the vector space C.
- The dimension of this vector space is the number of the basis vector which are *k*.
- $G_i \in C \rightarrow$ the rows of G are all legal codewords.

Hamming Weight

the minimum hamming distance of a linear block code is equal to the minimum hamming weight of the nonzero code vectors.

Since each $g_i \in C$,we must have $W_h(g_i) \geq d_{min}$ this a necessary condition but not sufficient.

Therefore, if the hamming weight of one of the rows of G is less than d_{min}, $\rightarrow d_{min}$ is not correct or G not correct.

Generator Matrix

- All 2^k codewords can be generated from a set of k linearly independent codewords.
- The simplest choice of this set is the k codewords corresponding to the information sequences that have a single nonzero element.
- Illustration: The generating set for the (7,4) code:

1000 ===> 1101000

0100 ===> 0110100

0010 ===> 1110010

0001 ===> 1010001

Generator Matrix (cont'd)

- Every codeword is a linear combination of these 4 codewords.

  That is: **c** = **m G**, where

$$\mathbf{G} = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}}_{k \times (n-k)} & \underbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{k \times k} \end{bmatrix} = \left[\mathbf{P} \mid \mathbf{I}_{\mathbf{k}} \right]$$

- Storage requirement reduced from $2^k(n+k)$ to $k(n-k)$.

Parity-Check Matrix

For $\mathbf{G} = [\ \mathbf{P} \mid \mathbf{I}_k\ ]$, define the matrix $\mathbf{H} = [\mathbf{I}_{n\text{-}k} \mid \mathbf{P}^T]$

(The size of **H** is $(n\text{-}k)\mathrm{x}n$).

It follows that $\mathbf{G}\mathbf{H}^T = \mathbf{o}$.

Since $\mathbf{c} = \mathbf{m}\mathbf{G}$, then $\mathbf{c}\mathbf{H}^T = \mathbf{m}\mathbf{G}\mathbf{H}^T = \mathbf{o}$.

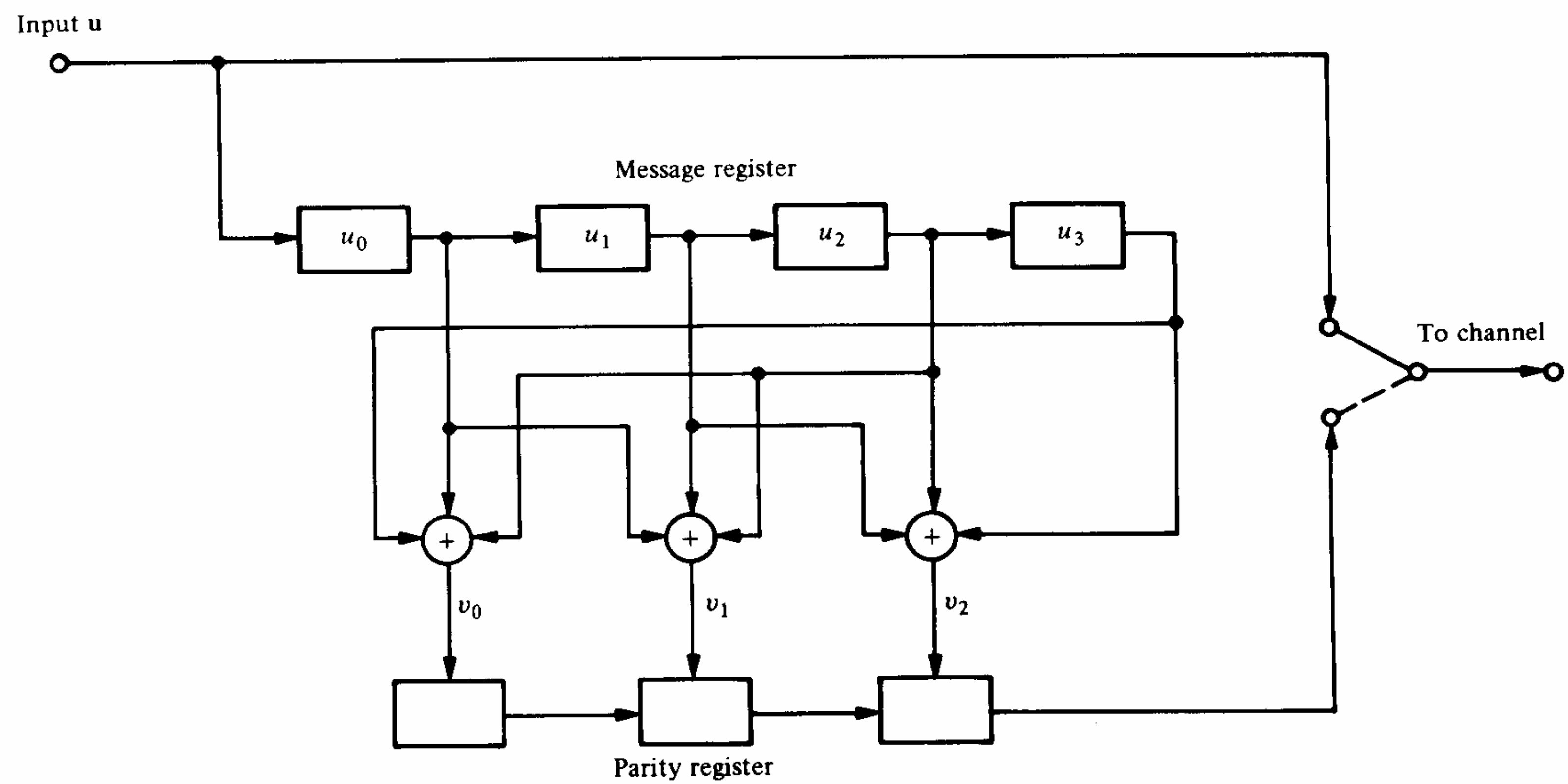
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Encoding Using H Matrix

$$\begin{bmatrix} c_1 & c_2 & c_3 & \underbrace{c_4 & c_5 & c_6 & c_7}_{\text{information}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \mathbf{0}$$

$$\begin{aligned} c_1 + c_4 + c_6 + c_7 &= 0 \\ c_2 + c_4 + c_5 + c_6 &= 0 \\ c_3 + c_5 + c_6 + c_7 &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} c_1 &= c_4 + c_6 + c_7 \\ c_2 &= c_4 + c_5 + c_6 \\ c_3 &= c_5 + c_6 + c_7 \end{aligned}$$

Encoding Circuit



The Encoding Problem (Revisited)

- Linearity makes the encoding problem a lot easier, yet: How to construct the G (or H) matrix of a code of minimum distance $d_{\min}$?
- The general answer to this question will be attempted later. For the time being we will state the answer to a class of codes: the Hamming codes.

Hamming Codes

- Hamming codes constitute a class of single-error correcting codes defined as:

  $n = 2^r - 1$, $k = n - r$, $r > 2$

- The minimum distance of the code $d_{min} = 3$
- Hamming codes are perfect codes.
- Construction rule:

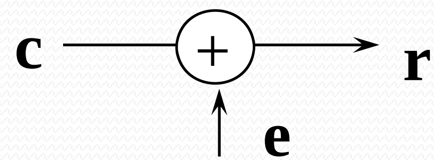
  The H matrix of a Hamming code of order r has as its columns all non-zero r-bit patterns.

  Size of H: $r \text{ x} (2^r - 1) = (n-k) \text{x} n$

Decoding

- Let **c** be transmitted and **r** be received, where

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$



e $\equiv$ error pattern = $e_1 e_2 \ldots\ldots e_n$, where

$$e_i = \begin{cases} 1 & \text{if the error has occured in the } i^{th} \text{ location} \\ 0 & \text{otherwise} \end{cases}$$

The weight of **e** determines the number of errors.

If the error pattern can be determined, decoding can be achieved by:

$$\mathbf{c} = \mathbf{r} + \mathbf{e}$$

Decoding (cont'd)

Consider the (7,4) code.

(1) Let 1101000 be transmitted and 1100000 be received.

Then: **e** = 0001000 (an error in the fourth location)

(2) Let **r** = 1110100. What was transmitted?

	c	**e**
#2	0110100	1000000
#1	1101000	0011100
#3	1011100	0101000

The first scenario is the most probable.

Standard Array

correctable error patterns

$$\begin{array}{ccccc}
c_0 & c_1 & c_2 & \cdots & c_{2^k-1} \\
e_1 + c_0 & e_1 + c_1 & e_1 + c_2 & \cdots & e_1 + c_{2^k-1} \\
e_2 + c_0 & e_2 + c_1 & e_2 + c_2 & \cdots & e_2 + c_{2^k-1} \\
\vdots & \vdots & \vdots & & \vdots \\
e_{2^{n-k}-1} + c_0 & e_{2^{n-k}-1} + c_1 & e_{2^{n-k}-1} + c_2 & & e_{2^{n-k}-1} + c_{2^k-1}
\end{array}$$

Standard Array (cont'd)

1. List the 2^k codewords in a row, starting with the all-zero codeword c_0.
2. Select an error pattern $\mathbf{e}_1$ and place it below c_0. This error pattern will be a correctable error pattern, therefore it should be selected such that:

   (i) it has the smallest weight possible (most probable error)

   (ii) it has not appeared before in the array.
3. Repeat step 2 until all the possible error patterns have been accounted for. There will always be $2^n / 2^k = 2^{n-k}$ rows in the array. Each row is called a *coset*. The leading error pattern is the *coset leader*.

Standard Array Decoding

- For an (n,k) linear code, standard array decoding is able to correct exactly 2^{n-k} error patterns, including the all-zero error pattern.
- Illustration 1: The (7,4) Hamming code

  # of correctable error patterns = 2^3 = 8

  # of single-error patterns = 7

  Therefore, all single-error patterns, and only single-error patterns can be corrected. (Recall the Hamming Bound, and the fact that Hamming codes are perfect.

Standard Array Decoding (cont'd)

Illustration 2: The (6,3) code defined by the H matrix:

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

$$c_1 = c_5 + c_6$$

$$c_2 = c_4 + c_6$$

$$c_3 = c_4 + c_5$$

Codewords

000000
110001
101010
011011
011100
101101
110110
000111

$d_{\min} = 3$

Standard Array Decoding (cont'd)

- Can correct all single errors and one double error pattern

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

The Syndrome

- Huge storage memory (and searching time) is required by standard array decoding.
- Define the syndrome

  $$\mathbf{s} = \mathbf{v}\mathbf{H}^T = (\mathbf{c} + \mathbf{e})\, \mathbf{H}^T = \mathbf{e}\mathbf{H}^T$$

- The syndrome depends only on the error pattern and not on the transmitted codeword.
- Therefore, each coset in the array is associated with a unique syndrome.

The Syndrom (cont'd)

Error Pattern	Syndrome
0000000	000
1000000	100
0100000	010
0010000	001
0001000	110
0000100	011
0000010	111
0000001	101

Syndrome Decoding

Decoding Procedure:

1. For the received vector **v**, compute the syndrome $\mathbf{s} = \mathbf{v}\mathbf{H}^T$.
2. Using the table, identify the error pattern **e**.
3. Add **e** to **v** to recover the transmitted codeword **c**.

<u>Example</u>:

v = 1110101 ==> **s** = 001 ==> **e** = 0010000

Then, **c** = 1100101

- Syndrome decoding reduces storage memory from $nx2^n$ to $2^{n-k}(2n-k)$. Also, It reduces the searching time considerably.

Decoding of Hamming Codes

- Consider a single-error pattern $\mathbf{e}^{(i)}$, where i is a number determining the position of the error.
- $\mathbf{s} = \mathbf{e}^{(i)}\,\mathbf{H}^{\mathrm{T}} = \mathbf{H}_i^{\mathrm{T}}$ = the transpose of the i^{th} column of $\mathbf{H}$.
- Example:

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

Decoding of Hamming Codes (cont'd)

- That is, the (transpose of the) i^{th} column of H is the syndrome corresponding to a single error in the i^{th} position.
- Decoding rule:

  1. Compute the syndrome $\mathbf{s} = \mathbf{v}\mathbf{H}^T$
  2. Locate the error (*i.e.* find i for which $\mathbf{s}^T = \mathbf{H}_i$)
  3. Invert the i^{th} bit of $\mathbf{v}$.

Hardware Implementation

- Let $\mathbf{v} = v_0\ v_1\ v_2\ v_3\ v_4\ v_5\ v_6$ and $\mathbf{s} = s_0\ s_1\ s_2$
- From the **H** matrix:

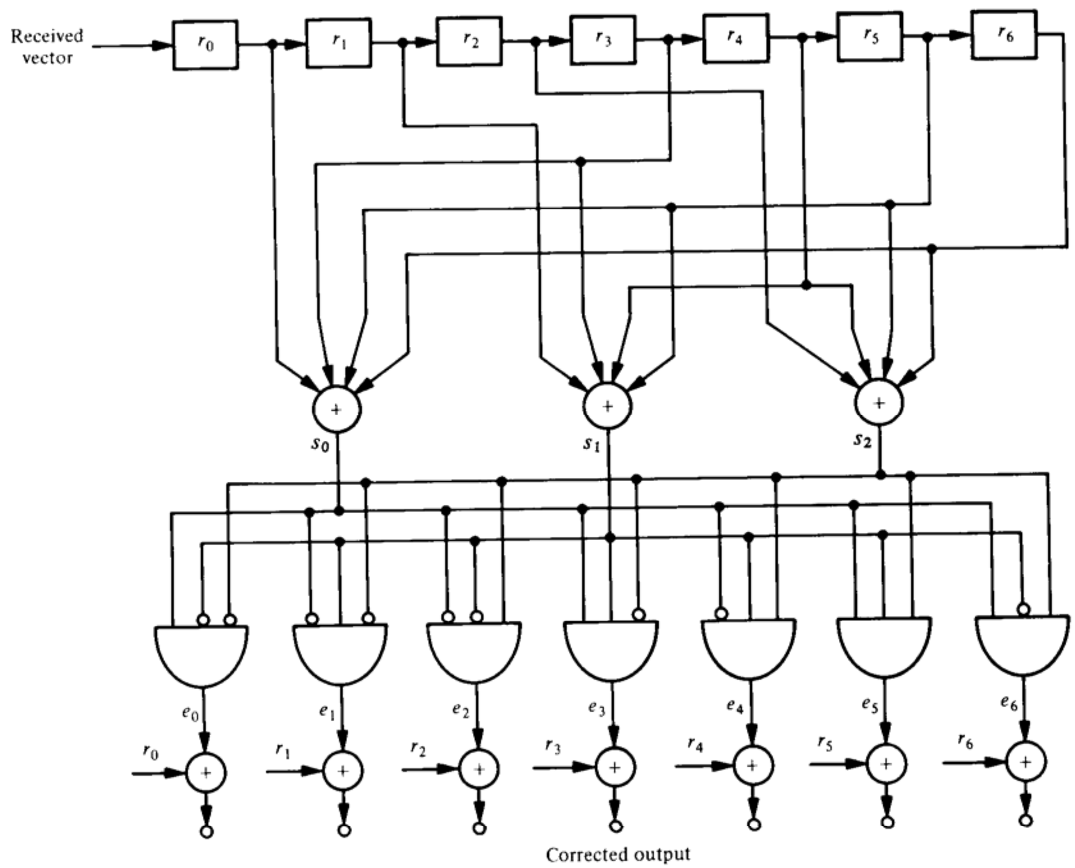
$$s_0 = v_0 + v_3 + v_5 + v_6$$

$$s_1 = v_1 + v_3 + v_4 + v_5$$

$$s_2 = v_2 + v_4 + v_5 + v_6$$

- From the table of syndromes and their corresponding correctable error patterns, a truth table can be construsted. A combinational logic circuit with s_0 , s_1 , s_2 as input and e_0 , e_1 , e_2 , e_3 , e_4 , e_5 , e_6 as outputs can be designed.

Decoding Circuit for the (7,4) HC



v rather than r