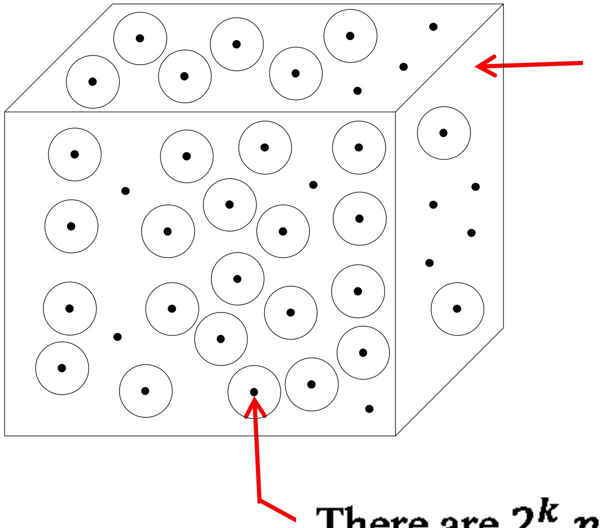
Chapter 2 An Introduction of Channel Coding

- 2.1 Channel Coding
- 2.2 Block Codes
- 2.3 Cyclic Codes
- 2.4 The Parity-Check Matrix

§ 2.1 Channel Coding

- Channel Coding: map a k-dimensional message vector to an n-dimensional codeword vector, and $k < n$.
- If it is a binary channel code, there are at most 2^k n-dimensional codewords. The redundancy of $2^n - 2^k$ enables the error-correction capability of the code.



The n-dimensional binary space that can accommodate at most 2^n binary vectors.

There are 2^k n-dimensional codeword vectors filling the space.

§ 2.1 Channel Coding

- Let $r = {}^{k}/_{n}$ be the code rate and C be the channel capacity. It is known if $r < C$, the estimation error probability of the source data X ($P_e[\hat{X} \neq X]$) can approach zero.
- A channel code is a specific capacity approaching operational strategy.
- **Code classification** (according to the encoder's structure)

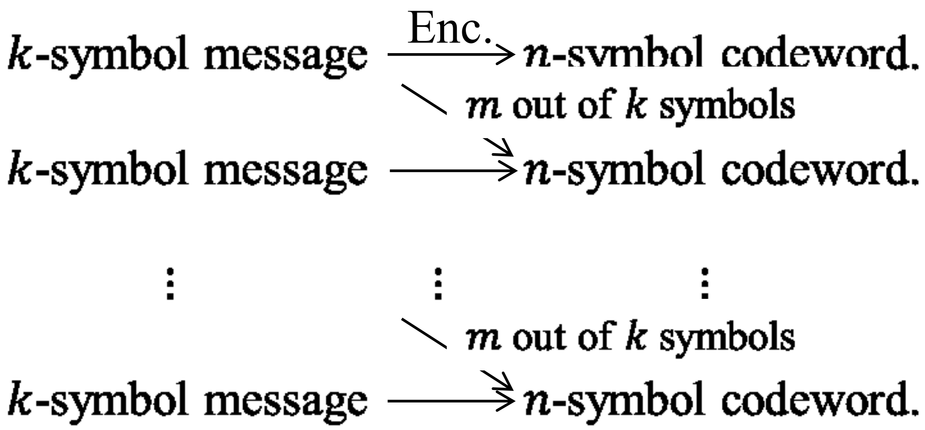
1. Block codes:

$$k\text{-symbol message} \xrightarrow{\text{Enc.}} n\text{-symbol codeword.}$$

- Encoder is memoryless and can be implemented with a combinatorial logic circuit.
- **Linear Block Code**: If c_i and c_j belong to a block code, $c' = a \cdot c_i + b \cdot c_j$ also belongs to the block code. $(a, b) \in F_q$ in which the block code is defined.
- Examples: Reed-Solomon code, algebraic-geometric code, Hamming code, low-density parity-check code.

§ 2.1 Channel Coding

2. Convolutional codes:



k-symbol message $\xrightarrow{\text{Enc.}}$ n-symbol codeword.

m out of k symbols

k-symbol message $\longrightarrow$ n-symbol codeword.

⋮ ⋮ ⋮

m out of k symbols

k-symbol message $\longrightarrow$ n-symbol codeword.

- Encoder has a memory of order m and can be implemented with a sequential logic circuit.
- Examples: Convolutional code, Trellis coded modulation, Turbo code.

§ 2.1 Channel Coding

Start with Error-Dectection:

The simplest class of block code is the **parity check code**, which cannot correct errors but can **detect** a single error.

For each binary message a parity check bit is added so that there are an **even** number of 1s in each codeword.

If k = 3 then there are 8 possible messages. The eight codeword will be:

000 →	000**0**
001 →	001**1**
010 →	010**1**
011 →	011**0**
100 →	100**1**
101 →	101**0**
110 →	110**0**
111 →	111**1**

When there are odd number of 1,
the decoder (detector) knows error
has been introduced.

§ 2.2 Block Codes

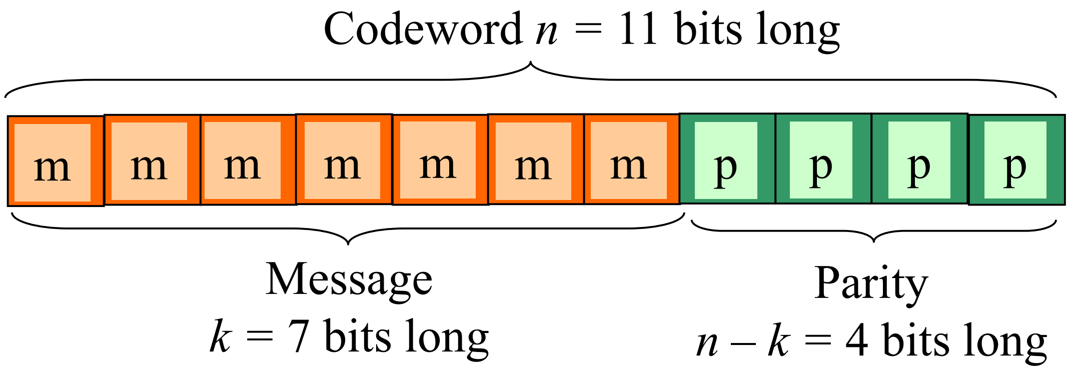
- All block codes are defined by their codeword length n, their message length (or dimension) k and their minimum Hamming distance d. A block code is denoted as an $(n,\ k,\ d)$ code.
- Code rate: $r = \frac{k}{n}$.
- Encoding of a Block code can be written as:

$$\bar{c} = \bar{m} \cdot \mathbf{G}.$$

$\bar{m}$ — k-dimensional message vector
$\mathbf{G}$ — a generator matrix of size $k \times n$
$\bar{c}$ — n-dimensional codeword vector

§ 2.2 Block Codes

Hamming Distance



Codeword n = 11 bits long

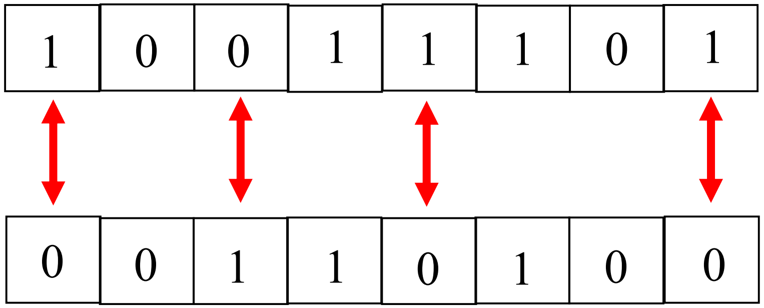
m	m	m	m	m	m	m	p	p	p	p

Message
k = 7 bits long

Parity
$n - k$ = 4 bits long

m = message bit
p = parity check bit

The Hamming distance between any two codewords is the total number of positions where the two codewords differ.



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

The total number of positions where these two codewords differ is 4. Therefore, the Hamming distance is 4.

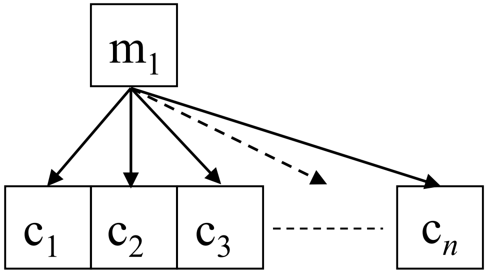
§ 2.2 Block Codes

Repetition Codes

A repetition encoder takes a **single** message bit and gives a codeword that is the message bit repeated *n* times, where *n* is an **odd** number

A message bit **0** will be encoded to give the codeword **0000...000**
A message bit **1** will be encoded to give the codeword **1111...111**

- This is the simplest type of error-correcting code as it only has **two codewords**
- We can easily see that it has a minimum Hamming distance $d = n$
- Hence it is an $(n, 1, n)$ block code



The generator matrix of the code is simply

$$\mathbf{G} = [1\ 1\ 1\ 1\ \dots\ 1]$$

§ 2.2 Block Codes

Majority Decoding

To recover the transmitted codeword of a repetition code, a simple decoder known as a **Majority Decoder** is used

1. The number of 0s and 1s in the received word are counted
2. If the number of 0s > number of 1s (*i.e*. a majority) , then the message bit was a 0. Else if the number of 1s > number of 0s, then the message bit was a 1

Example: Say our message bit was a 1 and it was encoded by the (5, 1, 5) repetition code then the codeword will be **c** = 11111.

- If after transmission we receive the word **r** = 10011 then the number of 1s > number of 0s and so the majority decoder decides that the original message was 1
- However, if we receive the word **r** = 00011 then the number of 0s > number of 1s and the Majority decoder **incorrectly** decides that the original message was 0

In general, a (n, 1, n) repetition code can correct up to $\frac{n-1}{2}$ errors

§ 2.2 Block Codes

The Minimum Hamming Distance and Error Correction of a Block Code

Take the (3, 1, 3) repetition code with codewords 000 and 111

If we add **one** error, the possible received words are

Codewords

000	111
001	**110**
010	**101**
100	**011**

All received words are **unique** so we can see that this code can correct one error.

If we add **two** errors, the possible received words are

Codewords

000	111
011	**100**
110	**001**
101	**010**

However, the received words also appear in the first list and so are not unique. Hence, this code cannot correct two or more errors.

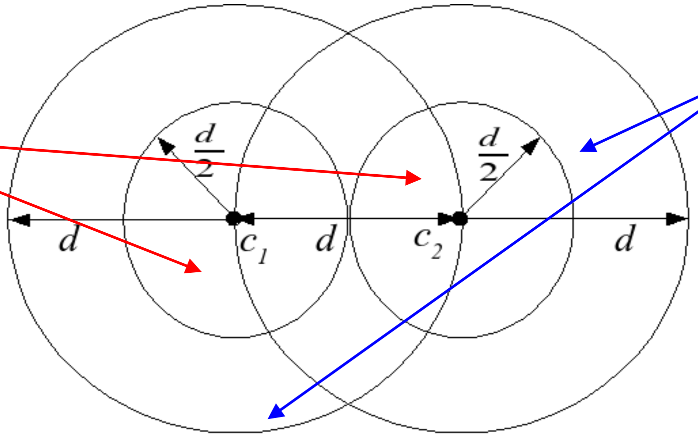
§ 2.2 Block Codes

The Minimum Hamming Distance and Error Correction of a Block Code

The minimum Hamming distance: for any two codewords c_i and c_j picked up from the codebook $\boldsymbol{C}$, the minimum Hamming distance d is defined as:

$$d = \min_{(c_i, c_j) \in \boldsymbol{C}} \{ d_{Ham}(c_i, c_j) \}.$$

- In general, a block code can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors, where $\lfloor x \rfloor$ means that x is rounded down to the nearest integer, *e.g.* $\lfloor 2.5 \rfloor = 2$
- A block code can **detect** d – 1 errors



A block code can ***correct*** received words with up to $d/2$ errors.

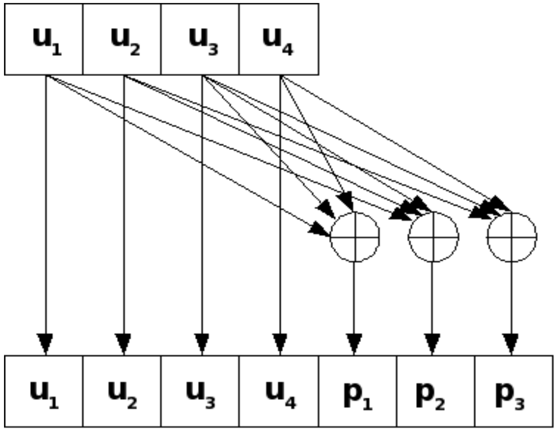
A block code can ***detect*** up to d – 1 errors

§ 2.2 Block Codes

The (7, 4, 3) Hamming Code

This code can correct 1 error

Notice that only 16 of 128 possible sequences of length 7 bits are used for transmission.



(7, 4, 3) Hamming Code

The parity bits are calculated by

$$p_1 = u_1 \oplus u_3 \oplus u_4$$

$$p_2 = u_1 \oplus u_2 \oplus u_3$$

$$p_3 = u_2 \oplus u_3 \oplus u_4$$

The encoding can be written as

$$\bar{c} = \bar{m} \cdot \mathbf{G},$$

and

$$\mathbf{G} =$$

This is a systematic encoding as the message symbols appear in the codeword.

Message	Codeword
0000	0000 000
0001	0001 101
0010	0010 111
0011	0011 010
0100	0100 011
0101	0101 110
0110	0110 100
0111	0111 001
1000	1000 110
1001	1001 011
1010	1010 001
1011	1011 100
1100	1100 101
1101	1101 000
1110	1110 010
1111	1111 111

§ 2.3 Cyclic Codes

- A cyclic code is a block code which has the property that cyclically shifting a codeword results in another codeword

- Cyclic codes have the advantage that simple encoders can be constructed using shift registers and low complexity decoding algorithms exist to decode them

- A cyclic code is constructed by first choosing a generator polynomial $g(x)$ and multiplying this by a message polynomial $m(x)$ to generate a codeword polynomial $c(x)$.

§ 2.3 Cyclic Codes

Cyclic Hamming Code

- The (7, 4, 3) Hamming code is actually a cyclic code and can be constructed using the generator polynomial $g(x) = x^3 + x^2 + 1$.

- For example, to encode the binary message 1010 we first write it as the message polynomial $m(x) = x^3 + x$ and then multiply it with $g(x)$ modulo-2

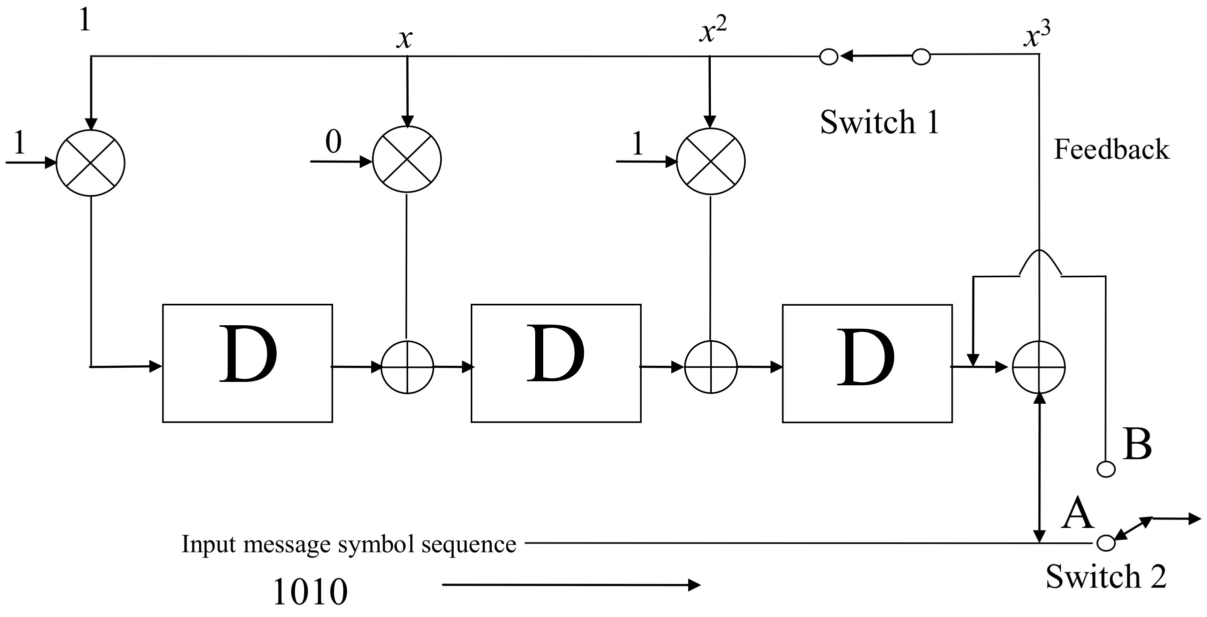
$$
\begin{aligned}
c(x) &= m(x)g(x) \\
&= (x^3 + x)(x^3 + x^2 + 1) \\
&= x^6 + x^5 + x^3 + x^4 + x^3 + x \qquad [(x^3 + x^3) \bmod 2 = 2x^3 \bmod 2 = 0] \\
&= x^6 + x^5 + x^4 + x
\end{aligned}
$$

This codeword polynomial corresponds to 1 1 1 0 0 1 0

- However, notice that the first four bits of this codeword are not the same as the original message 1010
- This is an example of a **non-systematic** code

§ 2.3 Cyclic Codes

Systematic Cyclic Hamming Code



An encoder for the systematic (7, 4, 3) cyclic Hamming code

1. For the first $k = 4$ message bits, switch 1 is closed and switch 2 is in position A
2. After the first 4 message bits have entered, switch 1 is open, switch 2 is in position B and the contents of memory elements are read out giving the parity check bits

§ 2.4 The Parity Check Matrix

- We need to know when a codeword is valid
- A parity check matrix **H** is defined as the **null space** of the generator matrix **G**, i.e. the inner product of the two matrices results in an all-zero matrix, $\mathbf{GH}^{\mathrm{T}} = \mathbf{0}$ (T is the transpose of the matrix)
- When a codeword is multiplied by the parity check matrix it should result in an all-zero vector, i.e.,

$$\bar{c}\cdot\mathbf{H}^{\mathrm{T}} = \bar{m}\cdot\mathbf{G}\cdot\mathbf{H}^{\mathrm{T}} = \underbrace{\underline{\mathbf{0}}}_{\text{Syndrone vector.}}.$$

- If $\hat{c}\cdot\mathbf{H}^{\mathrm{T}} = \mathbf{0}$, it implies $\hat{c}$ is a valid codeword.

§ 2.4 The Parity Check Matrix

- If the generator matrix is of the form $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$, where $\mathbf{I}_k$ **is a (*k* × *k*) identity matrix** and $\mathbf{P}$ is a parity matrix, then the parity check matrix is of the form $\mathbf{H} = [\mathbf{P}^T \mid \mathbf{I}_{n-k}]$

Taking the (7, 4, 3) Hamming code

$\mathbf{I}_4$ $\mathbf{P}$

$$\mathbf{G} = \left[\begin{array}{cccc:ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

The parity check matrix is →

$\mathbf{P}^T$ $\mathbf{I}_{n\text{-}k} = \mathbf{I}_{7\text{-}4} = \mathbf{I}_3$

$$\mathbf{H} = \left[\begin{array}{cccc:ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$