

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа № 6
дисциплина: Информатика
тема: «Обнаружение и исправление однократной ошибки в сообщении»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович
Проверил: Бондаренко Татьяна Владимировна

Белгород 2021

Цель работы: изучить основные принципы помехоустойчивого кодирования; изучить способ определения позиции и значения корректирующих бит кода Хемминга; получить практические навыки построения кода Хемминга, корректирующего однократные ошибки; изучить способ построения линейно-группового кода и возможность коррекции однократной ошибки с помощью линейно-группового кода.

Часть 1. Обнаружение и коррекция однократной ошибки в сообщении с помощью кода Хемминга

Задания к работе

1. Выполнить кодирование текстового сообщения М1 по буквам, используя русский или латинский алфавит, размер сообщения не менее 4 букв. Определить размер n в битах закодированного сообщения М.
2. Определить количество k контрольных разрядов кода Хемминга, необходимых для кодирования сообщения М размер n бит.
3. Определить позиции и значения k контрольных разрядов кода Хемминга: двумя способами:
 - подсчёт количества единиц в контролируемых контрольным битом разрядах сообщения;
 - использование двоичного представления номеров разрядов сообщения.
4. Записать полученное сообщение размера $(n + k)$ в коде Хемминга.
5. Смоделировать коррекцию ошибки: внести однократную, двукратную и k -кратную ошибки в произвольные биты сообщения и найти эти ошибки с помощью кода Хемминга, используя:
 - подсчёт количества единиц в контролируемых контрольным битом разрядах сообщения;
 - двоичное представление номеров разрядов сообщения.

Часть 2. Обнаружение и коррекция однократной ошибки в сообщении с помощью линейно-группового кода

Задание к работе:

1. Выполнить построение порождающей матрицы G линейно-группового кода, необходимой для кодирования сообщения М1 по буквам. Определить необходимое число информационных и проверочных столбцов матрицы G . Вычислить значение проверочных столбцов и доказать соответствие полученной порождающей матрицы G требованиям.
 2. Выполнить кодирование сообщения М1 по буквам с помощью порождающей матрицы G .
 3. Смоделировать коррекцию ошибки: внести в линейно-групповой код одной из букв сообщения М1 однократную ошибку, выполнить проверку сообщения на наличие ошибки и найти бит с ошибкой в сообщении.
- Провести аналогичную проверку для двукратной ошибки.

Часть 1. Обнаружение и коррекция однократной ошибки в сообщении с помощью кода Хемминга

1. Выполнил кодирование текстового сообщения **М1** по буквам, используя русский алфавит, размер сообщения — 4 буквы. Определил размер в **n** битах закодированного сообщения **М**.

$M = (\text{ИЛЪЯ}) = 001010.001101.011110.100001$

$n = 4 * 6 = 24$

2. Определил количество k контрольных разрядов кода Хемминга, необходимых для кодирования сообщения M рамера n бит.

$$2^k \geq M + k + 1$$

$$2^k \geq 25 + k$$

$$k = 5$$

3. Определил позиции и значения k контрольных разрядов кода Хемминга двумя способами:

- подсчёт количества единиц в контролируемых контрольным битом разрядах сообщения;
- использование двоичного представления номеров разрядов сообщения.

$$N = 24 + 5 = 29$$

1 способ:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | к1 | 6 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | | |
| к1 | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 6 | 0 |
| к2 | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 6 | 0 |
| к3 | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 6 | 0 |
| к4 | | | | | | | | X | X | X | X | X | X | X | X | | | | | | | | | X | X | X | X | X | X | 7 | 1 |
| к5 | | | | | | | | | | | | | | | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 5 | 1 |

Код Хемминга: 00101000.11010111.11010100.00100

2 способ:

| | |
|-----|-------|
| 3 | 00011 |
| 10 | 01010 |
| 12 | 01100 |
| 13 | 01101 |
| 14 | 01110 |
| 15 | 01111 |
| 18 | 10010 |
| 20 | 10100 |
| 21 | 10101 |
| 25 | 11001 |
| 27 | 11011 |
| XOR | 11000 |

Код Хемминга: 00101000.11010111.11010100.00100

4. Записал полученное сообщение размера $(n + k)$ в коде Хемминга:

Код Хемминга: 00101000.11010111.11010100.00100

5. Смоделировал коррекцию ошибки: внести однократную, двукратную и k-кратную ошибки в произвольные биты сообщения и найти эти ошибки с помощью кода Хемминга, используя:

- подсчёт количества единиц в контролируемых контрольным битом разрядах сообщения;
- двоичное представление номеров разрядов сообщения.

Однократная ошибка:

00101000.11010111.11010101.00100

Первый способ:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | κ1 | б |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | | |
| κ1 | X | | X | | X | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 6 | 0 | |
| κ2 | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 7 | 1 |
| κ3 | | | X | X | X | X | | | | X | | X | X | X | X | | | X | | X | X | X | X | | | | | X | X | 7 | 1 |
| κ4 | | | X | | | | | X | X | X | X | X | X | X | X | | | X | | X | | | | X | X | X | X | X | X | 7 | 1 |
| κ5 | | | X | | | | | | | X | | X | | | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 5 | 1 |

Различное значение принимают 2 и 4 бита, следовательно ошибка в 6 бите.

Второй способ:

| | |
|-----|-------|
| 3 | 00011 |
| 6 | 00110 |
| 10 | 01010 |
| 12 | 01100 |
| 13 | 01101 |
| 14 | 01110 |
| 15 | 01111 |
| 18 | 10010 |
| 20 | 10100 |
| 21 | 10101 |
| 25 | 11001 |
| 27 | 11011 |
| XOR | 11110 |

Различное значение принимают 2 и 4 бита, следовательно ошибка в 6 бите.

Двукратная ошибка: 00101000.11010111.11010100.00111

Первый способ:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | κ1 | 6 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | | |
| κ1 | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 6 | 0 |
| κ2 | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 6 | 0 |
| κ3 | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 6 | 0 |
| κ4 | | | | | | | | X | X | X | X | X | X | X | X | | | | | | | | | X | X | X | X | X | X | 7 | 1 |
| κ5 | | | | | | | | | | | | | | | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 5 | 1 |

Обнаружены 2 ошибки

2 способ:

| | |
|-----|-------|
| 3 | 00011 |
| 10 | 01010 |
| 12 | 01100 |
| 13 | 01101 |
| 14 | 01110 |
| 15 | 01111 |
| 18 | 10010 |
| 20 | 10100 |
| 21 | 10101 |
| 25 | 11001 |
| 27 | 11011 |
| XOR | 11000 |

Обнаружены 2 ошибки

Трехкратная ошибка: 00101000.11010111.11010100.00011

Первый способ:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | κ1 | 6 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | | |
| κ1 | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 5 | 1 |
| κ2 | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 5 | 1 |
| κ3 | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 6 | 0 |
| κ4 | | | | | | | | X | X | X | X | X | X | X | X | | | | | | | | | X | X | X | X | X | X | 7 | 1 |
| κ5 | | | | | | | | | | | | | | | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 5 | 1 |

Обнаружены ошибки

2 способ:

| | |
|-----|-------|
| 10 | 01010 |
| 12 | 01100 |
| 13 | 01101 |
| 14 | 01110 |
| 15 | 01111 |
| 18 | 10010 |
| 20 | 10100 |
| 21 | 10101 |
| 25 | 11001 |
| 27 | 11011 |
| XOR | 11011 |

Обнаружены ошибки