

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ 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-201

Машуров Дмитрий Русланович

Проверил: Бондаренко Т.В.

Белгород 2020

Лабораторная работа №6.

Обнаружение и исправление однократной ошибки в сообщении

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 M_1 по буквам, используя русский или латинский алфавит, размер сообщения не менее 4 букв. Определить размер n в битах закодированного сообщения M .
Например, в качестве кода можно использовать порядковый номер буквы в алфавите. Если $M_1 = \text{“АБ”}$, то $M = 000001000010$ и размер сообщения $n = 12$.
2. Определить количество k контрольных разрядов кода Хемминга, необходимых для кодирования сообщения M 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.

Дополнительное задание:

Составить программу, выполняющую построение кода Хемминга для произвольного сообщения, состоящего из символов русского и английского алфавита. (Сообщение необходимо закодировать)

Смоделировать процесс передачи сообщения, реализовав в программе случайное возникновение однократной, двукратной и k – кратной ошибки в случайно выбранных битах сообщения.

Реализовать в программе проверку сообщения в коде Хемминга на наличие однократной ошибки и поиск позиции бита с ошибкой.

Реализовать исправление ошибки и вывод откорректированного сообщения для пользователя.

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 M_1 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 M_1 по буквам с помощью порождающей матрицы G .
3. Смоделировать коррекцию ошибки: внести в линейно-групповой код одной из букв сообщения M_1 однократную ошибку, выполнить проверку сообщения на наличие ошибки и найти бит с ошибкой в сообщении. Провести аналогичную проверку для двукратной ошибки.

Дополнительное задание:

Составить программу, выполняющую построение порождающей матрицы линейно-группового кода заданной размерности. (Порождающая матрица должна подбираться программой автоматически, не допускается использование матрицы заданной пользователем заранее).

Реализовать получение линейно-группового кода произвольного сообщения, состоящего из символов русского и английского алфавита. (Сообщение необходимо закодировать).

Смоделировать процесс передачи сообщения, реализовав в программе случайное возникновение однократной, двукратной и k – кратной ошибки в случайно выбранных битах сообщения.

Реализовать в программе проверку сообщения в линейно-групповом коде на наличие ошибки и поиск позиции бита с ошибкой.

Реализовать исправление ошибки и вывод откорректированного сообщения для пользователя.

Выполнение работы:

Часть 1

1. Выполню кодирования текстового сообщения M_1 по буквами, используя русский или латинский алфавит, размер сообщения не менее 4 букв. Определю размер n в битах закодированного сообщения M

$M_1 = \text{"димон"} , M = 100101\ 100001\ 101110\ 110000\ 101111, n = 30$

2. Определю k контрольных разрядов для сообщения M длиной $n = 30$:

$$2^K \geq n + K + 1$$

$$2^K \geq 31 + K$$

Подбор значения k :

$$2^5 \not\geq 31 + 5$$

$$2^6 \geq 31 + 6$$

Следовательно $k = 6$

3. Определить позиции и значения k контрольных разрядов кода Хемминга:

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	30	31	32	33	34	35	36
	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1
1	0		1		0		1		0		1		0		0		1		0		1		0		1		0		0		0		1		1	
2		0	1			0	1			1	1			0	0			1	0			1	0			0	0			1	0			1	1	
4				0	0	0	1					0	0	0	0					1	1	1	0					0	0	1	0					1
8								0	0	1	1	0	0	0	0									1	1	0	0	0	0	1	0					
16																0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0					
32																																0	1	1	1	1

Подсчитав количество единиц в контрольных разрядах, получим:

$$K_1 = (1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_2 = (1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_4 = (1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_8 = (1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_{16} = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_{32} = (1 + 1 + 1 + 1) \bmod 2 = 0$$

	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	30	31	32	33	34	35	36
	0	1	1	0	0	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1
1	0		1		0		1		0		1		0		0		1		0		1		0		1		0		0		0		1		1	
2		0	1			0	1			1	1			0	0			1	0			1	0			0	0			1	0			1	1	
4				0	0	0	1					0	0	0	0					1	1	1	0					0	0	1	0					1
8								0	0	1	1	0	0	0	0									1	1	0	0	0	0	0	1	0				
16																0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0					
32																																0	1	1	1	1

2 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	30	31	32	33	34	35	36
0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1

$$3 = 000011_2$$

$$7 = 000111_2$$

$$10 = 001010_2$$

$$11 = 001011_2$$

$$17 = 010001_2$$

$$18 = 010010_2$$

$$20 = 010100_2$$

$$21 = 010101_2$$

$$22 = 010110_2$$

$$24 = 011000_2$$

$$25 = 011001_2$$

$$30 = 011110_2$$

$$33 = 100001_2$$

$$34 = 100010_2$$

$$35 = 100011_2$$

$$36 = 100100_2$$

	0	0	0	0	1	1
	0	0	0	1	1	1
	0	0	1	0	1	0
	0	0	1	0	1	1
	0	1	0	0	0	1
	0	1	0	0	1	0
	0	1	0	1	0	0
	0	1	0	1	0	1
xor	0	1	0	1	1	0
	0	1	1	0	0	0
	0	1	1	0	0	1
	0	1	1	1	1	0
	1	0	0	0	0	1
	1	0	0	0	1	0
	1	0	0	0	1	1
	1	0	0	1	0	0
	0	0	1	0	1	0

$$K_1 = 0, K_2 = 1, K_4 = 0, K_8 = 1, K_{16} = 0, K_{32} = 0$$

4. Запишу полученное сообщение $(n + k)$ разрядов в коде Хемминга
(красный цвет – контрольные разряды):

011000110110000011011101100001001111

5. Исправление ошибок (синий бит – ошибочный):

1-кратная ошибка:

Поступило сообщение: 100100 100001 101110 110000 101111

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	30	31	32	33	34	35	36
	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1
1	0			1		0		1		0		1			0		1		0		1		0		1		0		0		0		1		1	
2			0	1			0	1			0	1			0	0			1	0			1	0			0	0			1	0			1	1
4					0	0	0	1					0	0	0	0					1	1	1	0					0	0	1	0				
8								0	0	0	1	0	0	0	0	0									1	1	0	0	0	0	1	0				
16																0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0					
32																																0	1	1	1	1

$$K_1 = (1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_2 = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_4 = (1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_8 = (1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_{16} = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_{32} = (1 + 1 + 1 + 1) \bmod 2 = 0$$

Значения K_2 и K_8 не совпадают с изначальными, значит ошибочный разряд находится в $2 + 8$, то есть в 10

2 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	30	31	32	33	34	35	36
	0	1	1	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1
1	0		1		0		1		0		1		0		0		1		0		1		0		1		0		0		0		1		1	
2		0	1			0	1			0	1			0	0			1	0			1	0			0	0			1	0			1	1	
4				0	0	0	1					0	0	0	0					1	1	1	0					0	0	1	0					
8								0	0	0	1	0	0	0	0									1	1	0	0	0	0	1	0					
16																0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0					
32																																0	1	1	1	1

$$3 = 000011_2$$

$$7 = 000111_2$$

$$11 = 001011_2$$

$$17 = 010001_2$$

$$18 = 010010_2$$

$$20 = 010100_2$$

$$21 = 010101_2$$

$$22 = 010110_2$$

$$24 = 011000_2$$

$$25 = 011001_2$$

$$30 = 011110_2$$

$$33 = 100001_2$$

$$34 = 100010_2$$

$$35 = 100011_2$$

$$36 = 100100_2$$

	0	0	0	0	1	1
	0	0	0	1	1	1
	0	0	1	0	1	1
	0	1	0	0	0	1
	0	1	0	0	1	0
	0	1	0	1	0	0
	0	1	0	1	0	1
	0	1	0	1	1	0
xor	0	1	1	0	0	0
	0	1	1	0	0	1
	0	1	1	1	1	0
	1	0	0	0	0	1
	1	0	0	0	1	0
	1	0	0	0	1	1
	1	0	0	1	0	0
	0	0	0	0	0	0

$$K_1 = 0, K_2 = 0, K_4 = 0, K_8 = 0, K_{16} = 0, K_{32} = 0$$

Как можно заметить, значения K_2 и K_8 не совпадают с изначальными, поэтому ошибка находится в разряде $2+8$, т.е. в 10 разряде

2-кратная ошибка:

Поступило сообщение: 100101 000001 101110 110100 101111

Код Хэмминга не может исправить 2-кратную ошибку – только определить, поэтому:

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	30	31	32	33	34	35	36
	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	0	1	0	0	1	0	0	1	1	1	1
1	0		1		0		1		0		0		0		0		1		0		1		0		1		1		0		0		1		1	
2		0	1			0	1			1	0			0	0			1	0			1	0			0	1			1	0			1	1	
4				0	0	0	1					0	0	0	0					1	1	1	0				0	0	1	0					1	
8								0	0	1	0	0	0	0	0										1	1	0	1	0	0	1	0				
16																0	1	1	0	1	1	1	0	1	1	0	1	0	0	1	0					
32																															0	1	1	1	1	1

$$K_1 = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_2 = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_4 = (1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 0$$

$$K_8 = (1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_{16} = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1) \bmod 2 = 1$$

$$K_{32} = (1 + 1 + 1 + 1) \bmod 2 = 0$$

Можно заметить, что значение K_{16} не совпадает с изначальным, но поскольку K_{16} является контрольным разрядом, то мы не можем однозначно сказать, в каком разряде находится ошибка

2 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	30	31	32	33	34	35	36
	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	0	1	0	0	1	0	0	1	1	1	1
1	0		1		0		1		0		0		0		0		1		0		1		0		1		1		0		0		1		1	
2		0	1				0	1			1	0			0	0		1	0			1	0			0	1			1	0			1	1	
4				0	0	0	1						0	0	0	0	0				1	1	1	0				0	0	1	0					1
8								0	0	1	0	0	0	0	0	0									1	1	0	1	0	0	1	0				
16																0	1	1	0	1	1	1	0	1	1	0	1	0	0	1	0					
32																															0	1	1	1	1	1

$$3 = 000011_2$$

$$7 = 000111_2$$

$$10 = 001010_2$$

$$17 = 010001_2$$

$$18 = 010010_2$$

$$20 = 010100_2$$

$$21 = 010101_2$$

$$22 = 010110_2$$

$$24 = 011000_2$$

$$25 = 011001_2$$

$$27 = 011011_2$$

$$30 = 011110_2$$

$$33 = 100001_2$$

$$34 = 100010_2$$

$$35 = 100011_2$$

$$36 = 100100_2$$

	0	0	0	0	1	1
	0	0	0	1	1	1
	0	0	1	0	1	0
	0	1	0	0	0	1
	0	1	0	0	1	0
	0	1	0	1	0	0
	0	1	0	1	0	1
	0	1	0	1	1	0
xor	0	1	1	0	0	0
	0	1	1	0	0	1
	0	1	1	0	1	1
	0	1	1	1	1	0
	1	0	0	0	0	1
	1	0	0	0	1	0
	1	0	0	0	1	1
	1	0	0	1	0	0
	0	1	1	0	1	0

Можно заметить, что значение K_{16} не совпадает с изначальным, но поскольку K_{16} является контрольным разрядом, то мы не можем однозначно сказать, в каком разряде находится ошибка

k-кратная ошибка:

Выполняется аналогично как и с 2-кратной ошибкой

Часть 2

1. Число информационных бит единичной матрицы $n_{\text{и}} = \log_2 64 = 6$
Число проверочных бит $n_{\text{к}} = \log_2(n + 1 + \log_2(n + 1)) =$
 $(\log_2(6 + 1 + \log_2(6 + 1))) \approx 3,29 \rightarrow n_{\text{к}} = 4$

Единичная матрицы порядка $n_{\text{и}} = 6$

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Порождающая матрица G:

a_1	a_2	a_3	a_4	a_5	a_6	p_1 (1 \oplus 2 \oplus 6)	p_2 (2 \oplus 3 \oplus 4)	p_3 (1 \oplus 3 \oplus 4 \oplus 5)	p_4 (4 \oplus 5 \oplus 6)
1	0	0	0	0	0	1	0	1	0
0	1	0	0	0	0	1	1	0	0
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	1	1
0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0	0	1

$$d_{min} = 3$$

$$w_{min} = d_{min} - 1 = 2$$

Порождающая матрица G удовлетворяет требованиям:

- 1) Все вектора различны и линейно независимы
- 2) Нулевой вектор отсутствует
- 3) $w(v_i) \geq d_{min}$
- 4) $d(v_i, v_j) \geq d_{min}$

2. Закодирую сообщение M_1

$$100101 \rightarrow 0100$$

$$100001 \rightarrow 0011$$

$$101110 \rightarrow 1000$$

$$110000 \rightarrow 0110$$

$$101111 \rightarrow 0001$$

10010101001000010011101110100011000001111011110001

3. Добавлю в код однократную ошибку:

10000101001000010011101110100011000001111011110001

Матрица M' :

	a_1	a_2	a_3	a_4	a_5	a_6	p_1	p_2	p_3	p_4	S_1	S_2	S_3	S_4
a_1	1	0	0	0	0	0	0	0	0	0	1	0	1	0
a_2	0	1	0	0	0	0	0	0	0	0	1	1	0	0
a_3	0	0	1	0	0	0	0	0	0	0	0	1	1	0
a_4	0	0	0	1	0	0	0	0	0	0	0	1	1	1
a_5	0	0	0	0	1	0	0	0	0	0	0	0	1	1
a_6	0	0	0	0	0	1	0	0	0	0	1	0	0	1
p_1	0	0	0	0	0	0	1	0	0	0	1	0	0	0
p_2	0	0	0	0	0	0	0	1	0	0	0	1	0	0
p_3	0	0	0	0	0	0	0	0	1	0	0	0	1	0
p_4	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Проверю часть с ошибкой:

1000010100

$$S_1 = a_1 + a_2 + a_6 + p_1 = 1 + 1 = 0$$

$$S_2 = a_2 + a_3 + a_4 + p_2 = 1$$

$$S_3 = a_1 + a_3 + a_4 + a_5 + p_3 = 1$$

$$S_4 = a_4 + a_5 + a_6 + p_4 = 1$$

Значит ошибка находится в разряде a_4

Добавлю в код двукратную ошибку:

00000101001000010011101110100011000001111011110001

Проверю часть с ошибкой:

0000010100

Найду вектор-синдром, определяющий двукратную ошибку, для a_1 и a_4 :

xor	1	0	1	0
	0	1	1	1
	1	1	0	1

$$\begin{aligned}
S_1 &= a_1 + a_2 + a_6 + p_1 = 1 \\
S_2 &= a_2 + a_3 + a_4 + p_2 = 1 \\
S_3 &= a_1 + a_3 + a_4 + a_5 + p_3 = 0 \\
S_4 &= a_4 + a_5 + a_6 + p_4 = 1
\end{aligned}$$

Полученные значения совпадают с вектор-синдромом, значит ошибка действительно находится в ячейках a_1 и a_4 . Может быть исправлена если в построенной матрице M'' все векторы будут различны