Министерство образования и науки РФ

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Лабораторная работа №3

«Исследование методов помехоустойчивого кодирования: Хэмминга, циклического, сверточного»

Выполнил:

ст.гр. ИСб-22д

Воронин И.Ю.

Проверил:

Маслова М.А

Севастополь

2015

1.Цель работы

Исследование методов помехоустойчивого кодирования: Хэмминга, циклического, сверточного.

2.Вариант задания

1. Составить кодовую таблицу алфавита из 32 символов русского языка, используя для этого код Хэмминга (7,4). Записать образующую и проверочную матрицы и систему проверочных уравнений.

2. Записать слово из 4-5 букв и закодировать его с помощью составленной кодовой таблицы.

3. Проиллюстрировать процесс декодирования фрагмента сообщения с учетом влияния источника помех, приводящего к возникновению однократных ошибок в процессе передачи данных.

4. \*Составить кодовую таблицу алфавита из 32 символов русского языка, используя для этого циклический код. Применять образующие полиномы P1(x) = x3+x+1 и P2(x) = x3 +x2+1. Записать образующую и проверочную матрицы.

5. Записать слово из 4-5 букв и закодировать его с помощью составленной кодовой таблицы.

6. Проиллюстрировать процесс декодирования фрагмента сообщения с учетом влияния источника помех, приводящего к возникновению однократных ошибок в процессе передачи данных.

7. Записать произвольную информационную последовательность, содержащую 10-12 бит. Закодировать ее сверточным кодом с R=0,5.

8. Проиллюстрировать процесс декодирования фрагмента полученной последовательности с помощью дерева и сети Витерби для случаев безошибочной передачи и однократной ошибки.

9. Сделать выводы по работе

3.Ход работы

Составим двоичный безизбыточный код для 32 двух букв русского алфавита.(табл.3.1)

|  |  |  |  |
| --- | --- | --- | --- |
| А | 00000 | р | 10000 |
| Б | 00001 | с | 10001 |
| В | 00010 | т | 10010 |
| Г | 00011 | у | 10011 |
| Д | 00100 | ф | 10100 |
| Е | 00101 | х | 10101 |
| Ж | 00110 | ц | 10110 |
| З | 00111 | ч | 10111 |
| И | 01000 | ш | 11000 |
| Й | 01001 | щ | 11001 |
| К | 01010 | ъ | 11010 |
| Л | 01011 | ы | 11011 |
| М | 01100 | ь | 11100 |
| Н | 01101 | э | 11101 |
| О | 01110 | ю | 11110 |
| П | 01111 | я | 11111 |

Таблица 3.1-Алфавит.

* **Код Хэмминга**

Закодируем слово «свет» при помощи кода Хэмминга. Пусть длина информационного слова будет равна 10 бит. Следовательно, в этой двоичной последовательности может уместиться два символа нашей последовательности.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| С | в |  | е | т |
| 10001 | 00010 |  | 00101 | 10010 |

Таблица 3.2-Исходный вид сообщения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| С | в |  | е | т |
| 00100000100010 | |  | 00000100110010 | |

Таблица 3.3-Сообщение с контрольными битами.

В таблице (3.3) показано, что в каждом информационном сообщении будет содержаться 4 контрольных бита. Бит будет принимать значение 1, когда в области его действия будет нечётное количество единиц; 0, когда чётное. Другими словами, к ячейкам будет применяться исключающие «ИЛИ». В таблице (3.4.) показано, какие ячейки охватывает каждый контрольный бит.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| X |  | X |  | X |  | X |  | X |  | X |  | X |  | 1 |
|  | X | X |  |  | X | X |  |  | X | X |  |  | X | 2 |
|  |  |  | X | X | X | X |  |  |  |  | X | X | X | 4 |
|  |  |  |  |  |  |  | X | X | X | X | X | X | X | 8 |

Таблица 3.4 – Ячейки контрольных битов.

Проведя подсчёт, получим следующий вид сообщения. (табл.3.5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| с | в |  | е | т |
| 11110000100010 | |  | 00000101110010 | |

Таблица 3.5 – Итоговый вид сообщения.

Допустим, что нами было получено данное сообщение после влияния помех в следующем виде. (табл.3.6)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 11110**1**00100010 | |  | 000001011100**0**0 | |

Таблица 3.6 – Искажённое сообщение.

Проведём переподсчёт контрольных битов. Получим:

* 10100100100010 – 2 бита не совпадают.
* 10010100110000 – 3 бита не совпадают.

Из этого следует, что в каждом из сообщений есть ошибка. Чтобы найти бит, который необходимо инвертировать, сложим номера позиций неправильных контрольных битов.

* 2 + 4 = 6, следовательно, 10100**~~1~~**00100010
* 1 + 4 + 8 = 13, следовательно, 100101001100**~~0~~**0

Проведя передосчёт заново после инвертирования, получим сообщение с контрольными битами:

* 11110000100010
* 00000101110010

Без контрольных битов:

* 10001 - с; 00010 - в;
* 00101 - е; 10010 – т;

Объём передаваемых данных возрос с 20 бит, до 28 бит, следовательно, коэффициент сжатия равен 1,4.

* **Циклический код**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Буква | КОД | b1 | b2 | b3 | b4 | 5 | 6 | 7 |
|
| А | 00000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Б | 00001 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| В | 00010 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| Г | 00011 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Д | 00100 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| E | 00101 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| Ж | 00110 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| З | 00111 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| И | 001000 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Й | 01001 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| К | 01010 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| Л | 01011 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| М | 01100 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Н | 01101 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| О | 01110 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| П | 01111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Р | 10000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| С | 10001 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Т | 10010 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| У | 10011 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Ф | 10100 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| Х | 10101 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| Ц | 10110 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Ч | 10111 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| Ш | 11000 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Щ | 11001 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ъ | 11010 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| Ы | 11011 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| Ь | 11100 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Э | 11101 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| Ю | 11110 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| Я | 11111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Буква | КОД | 1 | 2 | 3 | 4 | b5 | 6 | 7 |
|
| А | 00000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Б | 00001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| В | 00010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Г | 00011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Д | 00100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 00101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ж | 00110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| З | 00111 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| И | 001000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Й | 01001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| К | 01010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Л | 01011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| М | 01100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Н | 01101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| О | 01110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| П | 01111 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Р | 10000 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| С | 10001 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Т | 10010 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| У | 10011 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ф | 10100 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Х | 10101 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ц | 10110 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ч | 10111 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ш | 11000 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Щ | 11001 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ъ | 11010 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ы | 11011 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ь | 11100 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Э | 11101 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Ю | 11110 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Я | 11111 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

Закодируем сообщение «хомяк».

0101100 0001011 1110100 0000000 1100010 0000000 1111111 0001011 1010011 0000000

* **Свёрточный код**

Составим произвольное сообщение из двоичного кода.

110101110010

При помощи свёрточного кода со скоростью ½ закодируем сообщения. (рис.3.1). Схема автомата Мура, применяемая в данном случае, содержит два сумматора по mod2 (S1 и S2) и два регистра (R1 и R2).

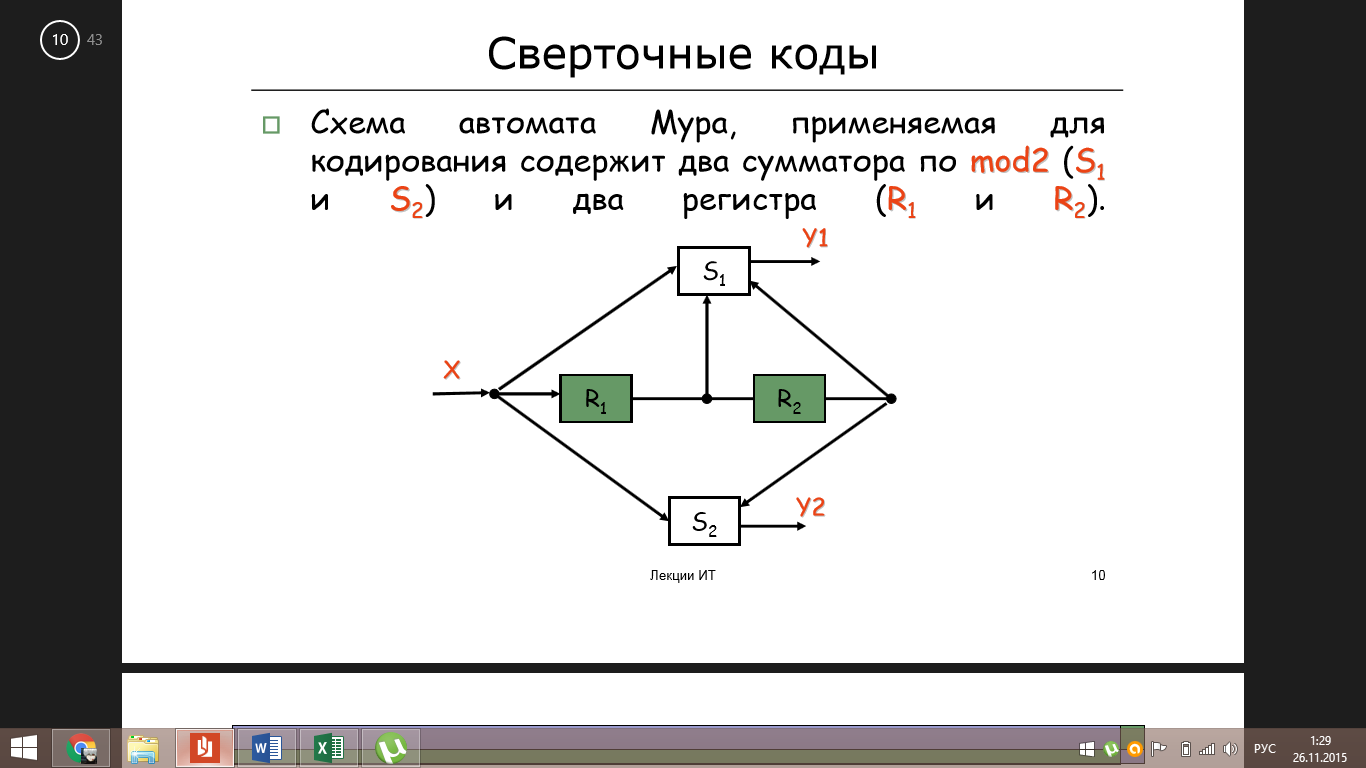


Рисунок 3.1-Схема Мура.

Полученное сообщение будет в 2 раза больше в объёме, так как k=1, а n=2. Получим таблицу итераций. (табл.3.7)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X | R1 | R2 | S1 | S2 | Y1 | Y2 |
| 1 | 0 | 0 | 1 ⨁ 0 ⨁ 0 | 1 ⨁ 0 | 1 | 1 |
| 1 | 1 | 0 | 1 ⨁ 1 ⨁ 0 | 1 ⨁ 0 | 0 | 1 |
| 0 | 1 | 1 | 0 ⨁1 ⨁ 1 | 0 ⨁ 1 | 0 | 1 |
| 1 | 0 | 1 | 1 ⨁ 0 ⨁ 1 | 1 ⨁ 1 | 0 | 0 |
| 0 | 1 | 0 | 0 ⨁ 1 ⨁ 0 | 0 ⨁ 0 | 1 | 0 |
| 1 | 0 | 1 | 1 ⨁ 0 ⨁ 1 | 1 ⨁ 1 | 0 | 0 |
| 1 | 1 | 0 | 1 ⨁ 1 ⨁ 0 | 1 ⨁ 0 | 0 | 1 |
| 1 | 1 | 1 | 1 ⨁ 1 ⨁ 1 | 1 ⨁ 1 | 1 | 0 |
| 0 | 1 | 1 | 0 ⨁ 1 ⨁ 1 | 0 ⨁ 1 | 0 | 1 |
| 0 | 0 | 1 | 0 ⨁ 0 ⨁ 1 | 0 ⨁ 1 | 1 | 1 |
| 1 | 0 | 0 | 1 ⨁ 0 ⨁ 0 | 1 ⨁ 0 | 1 | 1 |
| 0 | 1 | 0 | 0 ⨁ 1 ⨁ 0 | 0 ⨁ 0 | 1 | 0 |

На выходе получим сообщение: 11 01 01 00 10 00 01 10 01 11 11 10.

Выполним проверку. По формуле yi = x \* hi[x] при h1[x] = x2 + x+ 1 и h2[x]=x2+1. Получим полиномы:

y1 = x13 + x9 + x6 + x4 + x3 + x2 + x --> 100010010111**10**;

y1 = x13 + x12 + x11 + x7 + x5 + x4 + x3 + x -->111000101110**10**;

