Лабораторная работа №6

«Помехоустойчивое кодирование. Код Хэмминга»

Цель работы: научиться строить порождающую и проверочную матрицу для кода Хэмминга. Научиться строить код Хэмминга по матрице. Научиться вычислять синдром.

1

Условие: Построить систематический код Хэмминга для m = 4. Вычислить n и k. Вычислить размеры порождающей и проверочной матрицы. Построить проверочную матрицу. Получить проверочную подматрицу порождающей матрицы Сформировать порождающую матрицу.

Решение:

Вычислим n и k:

$n = 2^m - 1 = 2^4 - 1 = 15$
$k = 2^m - 1 - m = 2^4 - 1 - 4 = 11$

Проверочная матрица:

$$H_{4\times 15} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Удалим из проверочной матрицы столбцы, номер которых равен значению какой-либо степени числа 2, тем самым получив вспомогательную матрицу:

$$R_{4\times 11} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Повернём вспомогательную матрицу, а затем с её помощью построим порождающую, поставив в неё столбцы из вспомогательной матрицы на места, номер которых является степенью числа 2 (а остальные столбцы – столбцы единичной матрицы $k \times k$):

$$R_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$G_{k \times n} = G_{11 \times 15} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

2

Условие: Взять произвольное ненулевое двоичное информационное слово i длиной 11. Получить кодовое слово C, полученное перемножением информационного слова и порождающей матрицы.

Решение:

Возьмём $i = 01100111011$.

$C = i \times G = 110111010111011$

3

Условие: Внести в кодовое слово произвольную ошибку (инвертировать любой бит слова). Вычислить синдром. Локализовать ошибку и исправить её. Получить информационной слово. Убедиться в идентичности полученного информационного и изначального слов.

Адаменко

Решение:

Исходное кодовое слово: $C = 110111010111011$. Добавим ошибку в первый бит, получим $C_1 = 010111010111011$. Найдём синдром:

$S = H \times C_1^T = 0001_2 = 1_{10}$

Инвертируем бит с ошибкой:

$C = 110111010111011$

Удалим корректирующие биты и получим информационное слово:

$i = 01100111011$

4

Условие: Создать подпрограмму для реализации алгоритма помехоустойчивого кодирования по Хэммингу для m=4. На вход подпрограммы передаётся информационное слово. Подпрограмма возвращает кодовое слово.

5

Условие: Создать подпрограмму для декодирования кодового слова с учётом возможной ошибки. На вход подпрограммы передаётся кодовое слово. Подпрограмма вычисляет синдром, и при наличии ошибки исправляет её. Далее подпрограмма выделяет информационные биты и возвращает информационное слово.

6

Условие: Создать программу, демонстрирующую работу подпрограмм. Программа позволяет пользователю ввести информационное слово. Далее вызывается первая подпрограмма, и слово кодируется, затем заносится или не заносится случайная ошибка. Далее программа вызывает вторую подпрограмму и декодирует кодовое слово, исправляя ошибку. Результат каждого этапа выводится на экран.