**Лабораторная работа 10**

**Линейные блочные кода**

**Цель работы:** Изучение принципов линейного блочного кодирования и декодирования данных с использованием матриц генератора и проверочной матрицы.

**Теоретическая часть**

Линейные блочные коды представляют собой важный инструмент в цифровой связи и обеспечении целостности данных. Они позволяют добавлять избыточность к исходным данным для обеспечения их надёжной передачи по каналу с шумами. Основная идея линейных блочных кодов состоит в использовании матриц генератора и проверочной матрицы для преобразования информационных символов в кодовые слова и проверки их на наличие ошибок.

* **Матрица генератора (G)** используется для кодирования сообщения. Она преобразует информационный вектор длиной в кодовое слово длиной, умножая его на матрицу . Формула кодирования представляется как:

*C*=*mG*

где c — кодовое слово, m — вектор информационных символов, G — матрица генератора.

* **Проверочная матрица (H)** Позволяет обнаружить ошибки в кодовом слове. Для этого кодовое слово умножается на транспонированную проверочную матрицу:

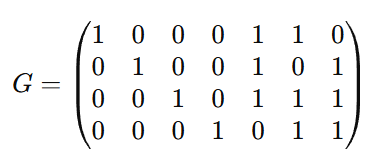
Формула для синдрома:

*s*=*c*⋅*HT*

где *s* — синдром ошибки. Если s=0, ошибок нет.

**Пример: Кодирование и обнаружение ошибок**

Пусть у нас есть матрица генератора *G* для кода (7,4) Хэмминга:



Для кодирования сообщения m=(1,0,1,1):

Изображение выглядит как текст, Шрифт, типография

Автоматически созданное описание

**Описание задания**

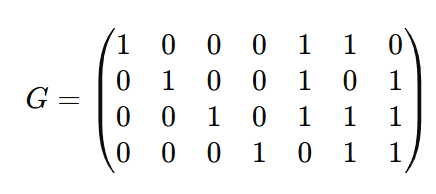
1. **Создание матрицы генератора**

Необходимо создать **матрицу генератора** для линейного блочного кода, которая будет преобразовывать информационные символы в кодовые слова.

Матрица генератора *G* обычно состоит из двух частей:

* Единичная матрица *Ik​*, размером *k*×*k*, которая отвечает за исходные информационные биты.
* Дополнительные столбцы, которые формируют проверочные биты.

Пусть мы рассматриваем код (7,4), где *n*=7 — длина кодового слова, а *k*=4 — количество информационных бит. В таком случае матрица генератора *G* будет иметь размер 4×7:



Здесь первые четыре столбца — это единичная матрица *I*4​, а последние три столбца отвечают за добавление проверочных бит.

Элементы подматрицы *P* выбираются так, чтобы можно было сгенерировать проверочные биты, обеспечивающие обнаружение и исправление ошибок. Мы можем выбрать любую подматрицу *P*, которая будет обеспечивать линейную независимость кодовых слов. Линейная независимость кодовых слов — это свойство линейного блочного кода, которое означает, что никакое кодовое слово не может быть выражено как линейная комбинация других кодовых слов.

Рассмотрим линейный блочный код (6,3), у которого есть следующие кодовые слова:

c1=(1,0,1,1,0,1),

c2=(0,1,0,1,1,0),

c3=(1,1,0,0,1,1).

Чтобы эти кодовые слова были линейно независимыми, никакое одно кодовое слово не должно быть получено путём сложения других кодовых слов. То есть нельзя найти такие коэффициенты α1,α2,α3​, которые не равны 0, чтобы было верно:

α1⋅c1+α2⋅c2+α3⋅c3=0

В бинарной арифметике, если сложение любых двух (или больше) кодовых слов даёт новое кодовое слово, это кодовое слово также должно быть линейно независимым от остальных.

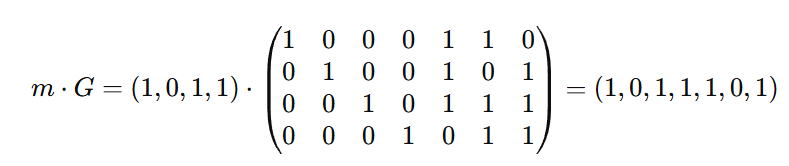
1. **Кодирование сообщений**

Теперь необходимо **закодировать несколько сообщений** длиной 4 бита (так как k=4) с использованием матрицы генератора *G*.

Пусть сообщение имеет вид m=(m1,m2,m3,m4). Для кодирования используется операция умножения вектора сообщения на матрицу генератора:

*c*=*m*⋅*G*

Например, возьмём сообщение *m*=(1,0,1,1):



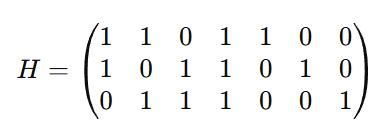
Полученное кодовое слово: *c*=(1,0,1,1,1,0,1).

1. **Создание проверочной матрицы**

Теперь создаём **проверочную матрицу** *H*, которая будет использоваться для обнаружения ошибок. Проверочная матрица должна удовлетворять следующему условию:

*G*⋅*HT*=0

Для кода (7,4) проверочная матрица *H* будет иметь размер 3×7, где 3=*n*−*k*. Пример проверочной матрицы для кода Хэмминга:



1. **Обнаружение ошибок**

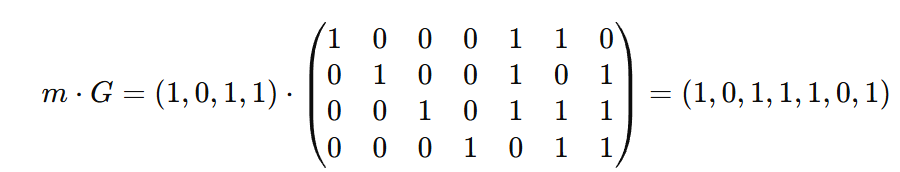
Моделируем передачу закодированных сообщений через канал с возможностью появления ошибок. Например, в закодированном слове изменился один бит.

Пусть было передано кодовое слово *c*=(1,0,1,1,1,0,1), но в процессе передачи второй бит изменился, и приёмник получил *c*′=(1,1,1,1,1,0,1).

Чтобы обнаружить ошибку, умножаем полученное кодовое слово *c*′ на транспонированную проверочную матрицу *HT* для получения синдрома:

*s*=*c*′⋅*HT*

В нашем примере:



Полученный синдром указывает на то, что ошибка произошла во втором бите. Этот бит можно исправить, изменив его обратно на 0, и мы восстановим правильное кодовое слово (1,0,1,1,1,0,1).

*Матричное умножение производится следующим образом: каждый элемент результирующего вектора является суммой произведений соответствующих элементов строки вектора c′ и столбца матрицы HT, с последующим вычислением по модулю 2 (сложение и умножение по модулю 2).*

**Практическое задание**

1. **Кодирование сообщений**: Создайте информационные векторы длиной 3 бита и с их помощью сформируйте кодовые слова длиной 5 бит с использованием матрицы генератора.
2. **Обнаружение ошибок**: Получите кодовые слова, в которых случайным образом изменён один бит, и с помощью проверочной матрицы найдите синдром ошибки.
3. **Исправление ошибок**: На основе полученного синдрома укажите, какой бит был изменён, и восстановите правильное кодовое слово.

**Контрольные вопросы**

1. Что такое матрица генератора и для чего она используется?
2. Какова роль проверочной матрицы в процессе декодирования?
3. Как с помощью синдрома можно определить наличие ошибок в кодовом слове?
4. В чём состоит основное преимущество использования линейных блочных кодов?