Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Защита информации и надёжность информационных систем

Студент: Скалкович С.Л.

ФИТ 3 курс 2 группа

Преподаватель: Нистюк О.А.

**Лабораторная работа №4**

ИЗБЫТОЧНОЕ КОДИРОВАНИЕ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ.

**Итеративные коды**

**Цель**: приобретение практических навыков кодирования/декодирования двоичных данных при использовании итеративных кодов.

**Задачи:**

1. Закрепить теоретические знания по использованию итеративных кодов для повышения надежности передачи и хранения в памяти компьютера двоичных данных.

2. Разработать приложение для кодирования/декодирования двоичной информации итеративным кодом с различной относительной избыточностью кодовых слов.

3. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**1. Теоретические сведения**

Итеративные коды относятся к классу кодов произведения. Кодом произведения двух исходных (базовых) помехоустойчивых кодов называется такой многомерный помехоустойчивый код, кодовыми последовательностями которого являются все двумерные таблицы со строками кода (k1) и столбцами кода (k2).

Итеративные коды могут строиться на основе использования дву-, трехмерных матриц (таблиц) и более высоких размерностей. Каждая из отдельных последовательностей информационных символов кодируется определенным линейным кодом (групповым или циклическим). Получаемый таким образом итеративный код также является линейным.

Простейшим из итеративных кодов является двумерный код с проверкой на четность по строкам и столбцам. Итеративные коды, иногда называемые прямоугольными кодами либо композиционными, являются одними из самых простых (с точки зрения аппаратной реализации) избыточных кодов, позволяющих исправлять ошибки в информационных словах.

Кодовые слова записываются в виде таблицы. Проверочные символы вычисляются исходя из того, что строки и столбцы должны содержать четное (нечетное) число единиц. На рисунке 1.1 представлена матрица, поясняющая принцип формирования избыточных символов итеративного кода.

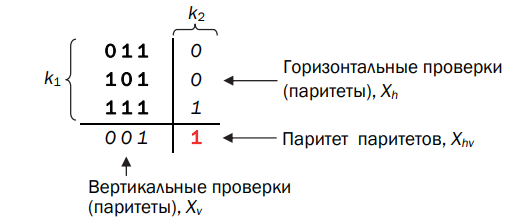


Рисунок 1.1 – Матрица для итеративного кода

Как видно, избыточные символы (называемые также паритетами) в приведенном кодовом слове в принятом порядке (Xh, Xv, Xhv) записываются сверху вниз, справа налево. Возможен обратный или иной порядок. Важно только, чтобы при декодировании сообщения использовался аналогичный порядок следования паритетов. Символ Xhv (паритет паритетов) равен сумме по модулю 2 символов информационного слова Xk, а также проверочных символов Xv и Xh.

Принято считать рассматриваемый код многомерным, если количество измерений, по которым вычисляются и анализируются паритеты, не менее 3. Таким образом, простейшим многомерным линейным итеративным кодом является код трехмерный.

Пример реализации трехмерного случая иллюстрирует рис. 1.2. Дополнительно к двум кодам на основе кодов простой четности (по вертикали и горизонтали) избыточные символы вычисляются по диагонали: Xd. Приведенную на рис. 1.2 трехмерную структуру итеративных кодов можно дополнить достаточно большим числом разнообразных проверок на четность по диагоналям, тем самым получив набор кодов с высокими корректирующими возможностями.



Рисунок 1.2 – Принцип формирования избыточных символов для линейного итеративного кода с диагональными проверками

1. **Практическая часть**

В соответствии с вариантом №2 необходимо использовать следующие параметры, представленные на рисунке 2.1.



Рисунок 2.1 – Варианты заданий

Сперва необходимо создать матрицу размерностью 4x5. Исходное сообщение, генерируется случайным образом каждый раз при запуске программы, в ходе отчета исходное слово Xk = 01011111111110011100. Матрица изображена на рисунке 2.2.

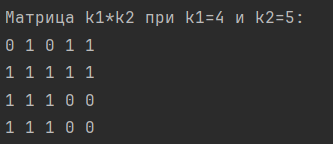


Рисунок 2.2 – Матрица размерностью 4x5

Далее мы вычисляем паритеты и создаем проверочные биты сообщения. Проверочные биты представлены на рисунке 2.3.



Рисунок 2.3 – Вывод вертикального и горизонтального паритета для матрицы 4x5

Следующим шагом создается матрица для полученного сообщения с случайно сгенерированной ошибкой. Далее вычисляются проверочные биты для этой новой матрицы. Результат выводится на рисунке 2.4.

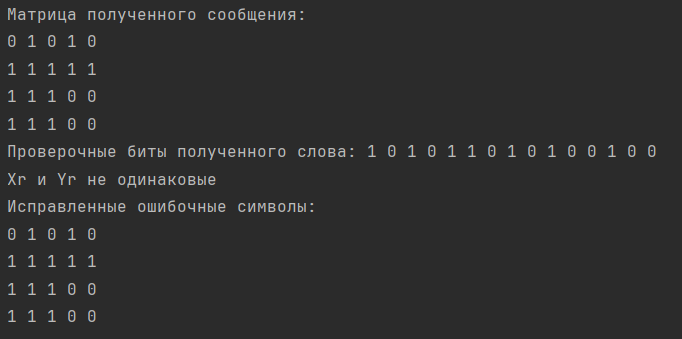


Рисунок 2.4 – Матрица для полученного сообщения с ошибкой

В результате, для вычисления ошибки необходимо сравнить проверочные символы обеих матриц и отметить несовпадения. Благодаря этому можно определить положение ошибки.

**Вывод**

Итеративные коды являются одним из классов кодов произведения. Они отличаются тем, что их можно легко представить в виде матрицы, и их реализация возможна как на аппаратном, так и на программном уровне.

Однако, основная ограниченность итеративных кодов заключается в их способности исправлять только одиночные ошибки. Это означает, что они могут обнаруживать и исправлять ошибки, которые произошли в результате единичного неверного бита или символа в передаваемом сообщении. Однако, когда в сообщении возникают несколько ошибок или ошибки происходят в более чем одном бите, итеративные коды могут не справиться с исправлением таких ошибок.

Таким образом, хотя итеративные коды обладают преимуществами легкого представления в виде матрицы и возможности реализации на разных уровнях, их способность исправлять только одиночные ошибки ограничивает их применимость в более сложных сценариях с множеством ошибок.