**Методические указание к лабораторной работе**

**Тема: Исследование корректирующего кода**

1. **Лабораторное задание**
   1. Ознакомиться с интерфейсом программы и схемами кодера и декодера при (n,k)=(7,4).
   2. Задать исходную комбинацию на входе кодера циклического кода (7,4) и произвести кодирование.
   3. Затем в канале указать ошибки в любых битах получившейся в результате кодирования комбинации.
   4. Произвести декодирование получившейся комбинации с ошибкой, с помощью декодера и сравнить с исходной.
2. Ознакомление с методами построения корректирующих кодов. Экспериментальное исследование обнаруживающей и исправляющей способности циклического кода.

|  |  |
| --- | --- |
| № Комбинации | Разрешенные 7-элементные кодовые комбинации |
| 1 | 0001 011 |
| 2 | 1000 101 |
| 3 | 1100 010 |
| 4 | 0110 001 |
| 5 | 1011 000 |
| 6 | 0101 100 |
| 7 | 0010 110 |
| 8 | 1001 110 |
| 9 | 0100 111 |
| 10 | 1010 011 |
| 11 | 1101 001 |
| 12 | 1110 100 |
| 13 | 0111 010 |
| 14 | 0011 101 |
| 15 | 1111 111 |
| 16 | 0000 000 |

Где 0100 111 – разрешенная комбинация, информационные элементы которой соответствуют в десятичной счисления номеру бригады (лабораторной установки №4).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Пункт работы | Переданные информационные | Комбинация на выходе передатчика | Сочетание ошибок | Комбинация на входе декодера | Обнаружение ошибки (да/нет) | Номер исправленного | Принятые информационные |
| 5а | 1100 | 1101001 | 0001000 | 1100001 | да | 4 | 1101 |
| 5б | 1100 | 1100010 | 0000100 | 1100110 | да | 5 | 1100 |
| 6 | 1011 | 1011000 | 0100000 | 1111000 | да | 2 | 1011 |
| 7.1 | 1100 | 1100010 | 0010100 | 1110110 | да | 3,5 | 1110 |
| 7.2 | 1100 | 1100010 | 0010001 | 1110011 | да | 3,7 | 1010 |
| 8.1 | 1100 | 1100010 | 1100010 | 0000000 | Нет | 1,2,6 | 0000 |
| 8.2 | 1100 | 1100010 | 1010011 | 0110001 | Нет | 2,3,7 | 0110 |
| 8.3 | 1100 | 1100010 | 1111111 | 0011101 | Нет | 1-7 | 0011 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 1100 | 1100010 | 1100001 | 0000011 | Да | 1,2,7 | 0001 |

1. **Порядок выполнения работы**
   1. Изучить теоретические сведения к данной лабораторной работе, приведенные в пункте 3.
   2. Запустить программу, двойным кликом мыши по ярлыку с названием «Циклический код (7.4)».

На экране появится окно с изображением кодера циклического кода (7,4) (Рисунок 3): Интерфейс программы в окне с кодером циклического кода (7;4):

1. Исходная кодовая комбинация;
2. Закодированная кодовая комбинация;
3. Ячейки формирователя проверочных групп (ФПГ);
4. Кнопка для начала кодирования (впоследствии становится кнопкой «Такт»);
5. Кнопка для перехода в канал связи, чтобы внести ошибку.

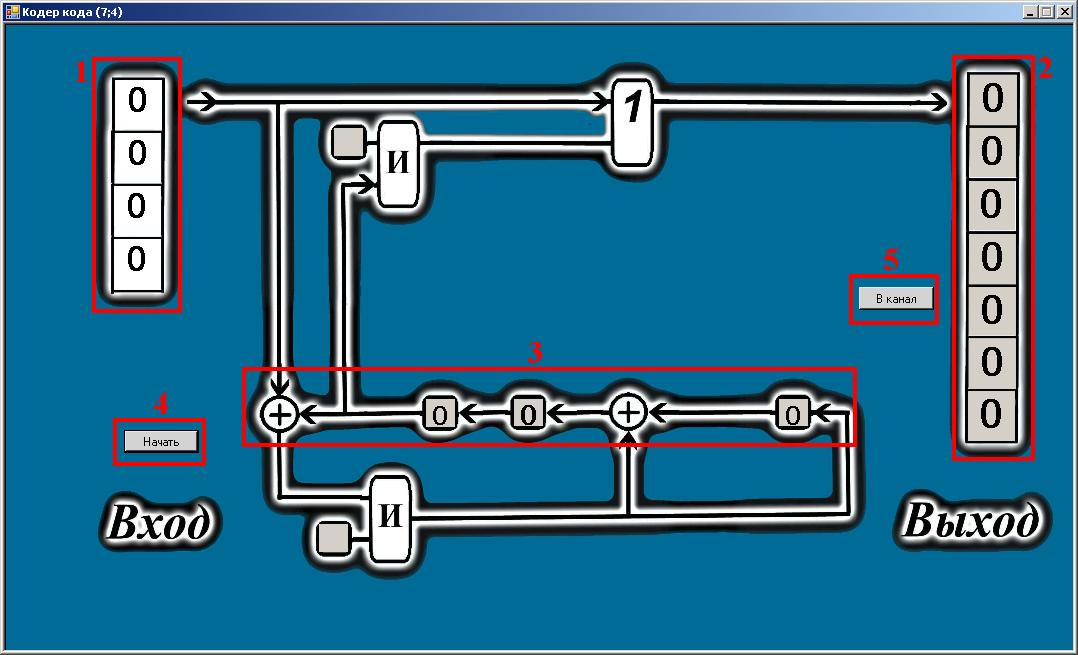


Рисунок 3 – Модель кодера циклического кода (7,4)

4.3 На входе кодера необходимо ввести свою исходную информационную кодовую комбинацию, после чего нажать на кнопку «Начать».

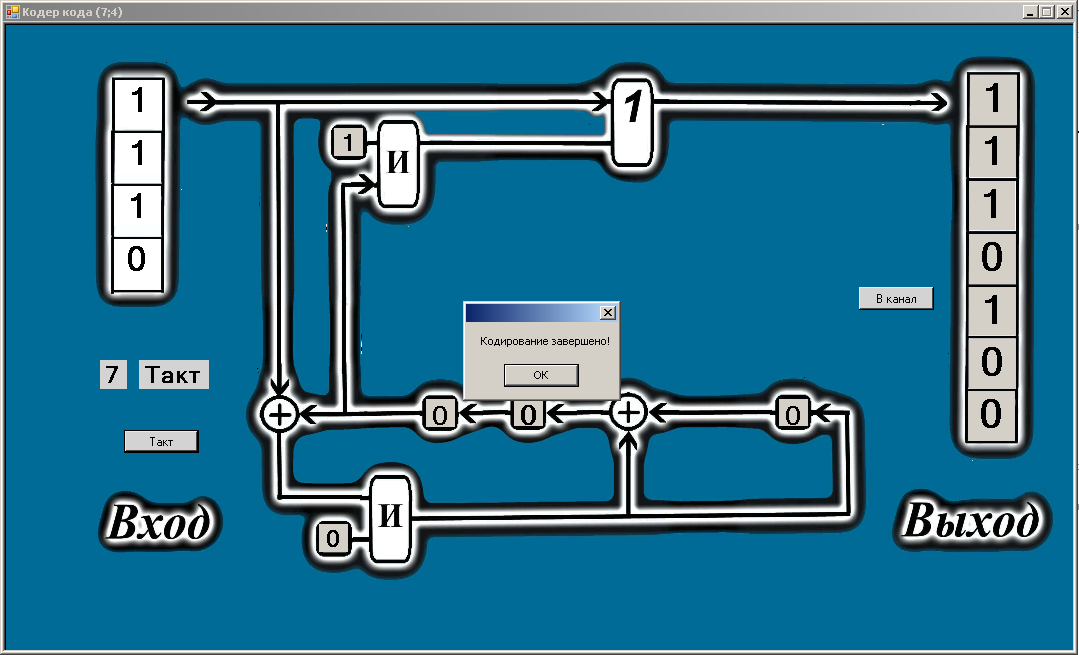


Рисунок 4 – Модель кодера циклического кода (7,4)

С помощью кнопки «такт» происходит выполнение следующего такта. Содержимое ячеек формирователя проверочной группы каждого такта и значение элемента на входе нужно свести в таблицу 2:

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Такт | Вход | № ячейки ФПГ | | |
| 1 | 2 | 3 |
| 0 | - | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 |
| 5 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 |

4.4 По истечению 7 тактов формируется закодированная кодовая комбинация, которую нужно записать в отчет. Затем программа выдает сообщение «Кодирование завершено!», следует нажать ОК и перейти к каналу с помощью кнопки «В канал».

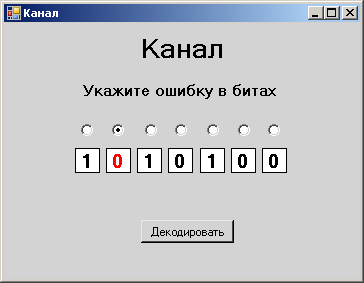


Рисунок 5 – Окно канала

4.5 В канале необходимо указать ошибку в битах закодированной кодовой комбинации (в любом бите на выбор студента), после чего необходимо нажать кнопку «Декодировать» для перехода к декодеру Меггита.

4.6 На рисунке 6 приведена схема декодера Меггита для циклического кода (7,4), по которой будет происходить декодирование кодовой комбинации с ошибкой. Бит с ошибкой выделен красным цветом. Для начала декодирования необходимо нажать кнопку «Начать».

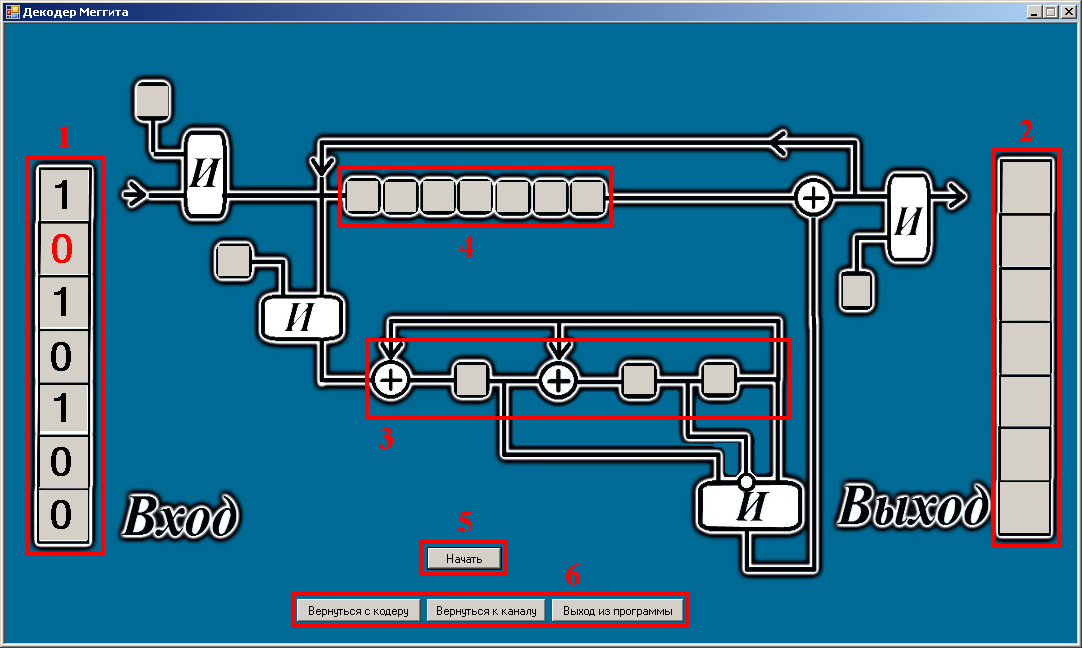


Рисунок 6 – Модель декодера Меггита

Интерфейс программы в окне с декодером Меггита:

1. Закодированная кодовая комбинация с ошибкой;
2. Результат декодирования;
3. Ячейки ФПГ;
4. Ячейки регистра задержки (РЗ);
5. Кнопка для начала декодирования (впоследствии становится кнопкой «Такт»);
6. Группа кнопок для дальнейшего действия:

* Кнопка «Вернуться к кодеру» - если нужно задать другую исходную комбинацию и произвести её кодирование;
* Кнопка «Вернуться к каналу» - если необходимо задать ошибку в другом бите закодированной комбинации;
* Кнопка «Выход из программы» - если лабораторная работа выполнена, чтобы закрыть окно программы.

4.7 С помощью кнопки «такт» осуществляется переход к следующему такту. В вверху указывается действие текущего такта:

* 0 такт – «Начальное состояние…»
* 1-7 такт – «Идет загрузка в регистр…»
* 8-14 такт – «Процесс исправления ошибки…»
* 15-21 такт – «Выгрузка данных из регистра на выход…»

После выполнения 21 такта на экране высвечивается сообщение «Процесс декодирования завершен!!!» (Рисунок 7). Необходимо нажать ОК и зафиксировать в отчете кодовую комбинацию на выходе и выделить в ней информационные элементы.

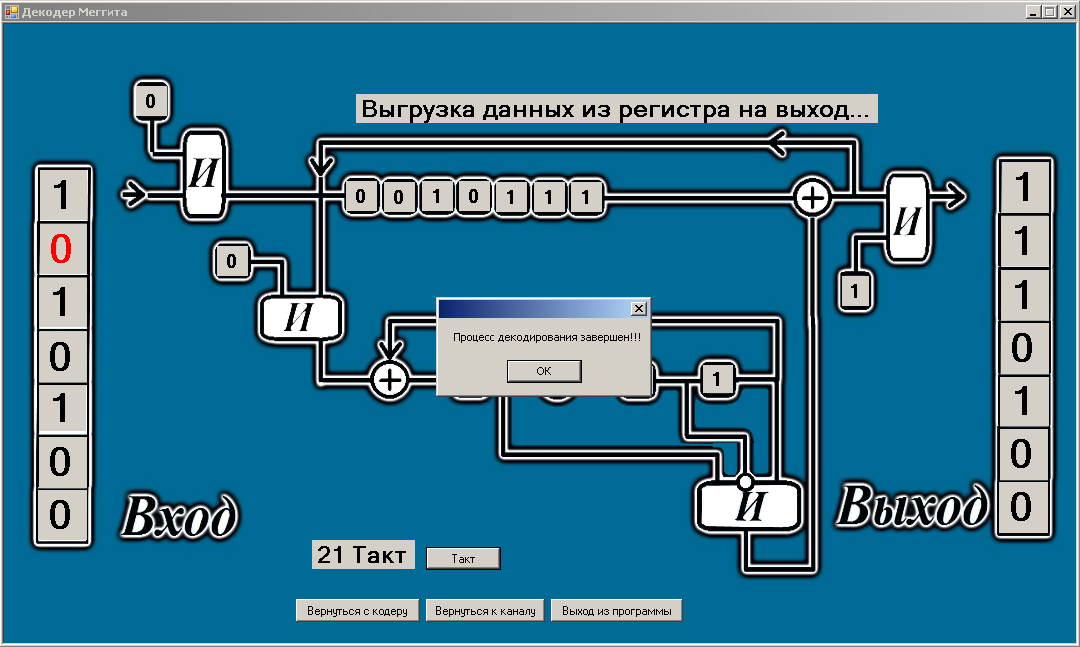


Рисунок 7 – Модель декодера Меггита

Содержимое ячеек формирователя проверочной группы каждого такта и значение элемента на входе при выполнении действий загрузки в регистр и процесса исправления ошибки нужно свести в таблицу 3:

Таблица 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Такт | Вход | № ячейки ФПГ | | |
| 1 | 2 | 3 |
| 0 | - | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |
| 4 | 1 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 1 |
| 6 | 1 | 0 | 1 | 0 |
| 7 | 0 | 0 | 0 | 1 |
| 8 | - |  |  |  |
| 9 | - |  |  |  |
| 10 | - |  |  |  |
| 11 | - |  |  |  |
| 12 | - |  |  |  |
| 13 | - |  |  |  |
| 14 | - |  |  |  |
| 15 | - |  |  |  |

4.8 После окончания выполнения лабораторной работы следует закрыть программу, затем выключить компьютер.

1. **Содержание отчёта**

Отчет должен содержать:

* цель работы;
* предварительный расчет;
* структурные схемы кодера и декодера Меггита;
* результаты кодирования и декодирования в виде таблиц 2 и 3;
* выводы.

**Основные теоретические положения**

Принципы помехоустойчивого (корректирующего) кодирования. Кодовое расстояние

В теории помехоустойчивого кодирования важным является вопрос об использовании избыточности для корректирования возникающих при передаче ошибок. Здесь удобно рассмотреть блочные коды, в которых всегда имеется возможность выделить отдельные кодовые комбинации. Напомним, что для равномерных кодов число возможных комбинаций равно , где – значимость кода. В обычном не корректирующем коде без избыточности число комбинаций *K* выбирается равным числу сообщений алфавита источника , и все комбинации используются для передачи информации. Корректирующие коды строятся так, чтобы число комбинаций превышало число сообщений источника . Однако в этом случае лишь комбинаций из общего числа используется для передачи информации. Эти комбинации называются *разрешенными*, а остальные комбинаций носят название *запрещенных*. На приемном конце на декодирующем устройстве известно, какие комбинации являются разрешенными и какие – запрещенными. Поэтому, если переданная разрешенная комбинация в результате ошибки преобразуется в некоторую запрещенную комбинацию, то такая ошибка будет *обнаружена*, а при определенных условиях *исправлена*. Известно, что ошибки, приводящие к образованию другой разрешенной комбинации, не образуются.

Различие между комбинациями равномерного кода принято характеризовать расстоянием, равным числу символов, которыми отличаются комбинации одна от другой. Расстояние между двумя комбинациями и определяется количеством единиц в сумме этих комбинаций при сложении по модулю два.

*Например:*

Для любого кода минимальное расстояние между разрешенными комбинациями в данном коде называется кодовым расстоянием , или расстоянием по Хэмингу.

Введем коэффициенты, характеризующие обнаруживающую и исправляющую способность кодов (, ).

– вероятность обнаружения ошибок,

– общая вероятность ошибок.

Принцип направления ошибок следующий. Мы разбиваем все множество комбинаций на ряд непересекающихся подпространство по количеству разрешенных комбинаций. Все запрещенные комбинации находятся в соответственных подпространствах разрешенных комбинаций. Если передавалась комбинация и она незначительно исказилась помехами и попала в подпространство , то считаем, что искажений не было. Коэффициент исправления:

– вероятность исправления ошибок.

в раз больше .

*Кратность ошибки* – число искаженных элементов в одной комбинации. Если в комбинации искажено два элемента, то говорят, что ошибка двукратная. Кратность ошибок в значительной степени определяется каналом. Есть каналы, в которых ошибки случайны и независимы (в таких каналах с увеличением кратности ошибок вероятность увеличивается), а есть каналы, в которых ошибки пакетируются.



Рисунок 1 Модели каналов

а) с независимыми ошибками;

б) с кратными ошибками (пакетирование)

Для правильности выбора кода необходимо знать статистику ошибок в канале.

Обнаруживающая и исправляющая способность кодов

Для лучшего понимания сущности обнаружения и исправления ошибок воспользуемся пространственными представлениями.

Для обнаружения ошибок все пространство кодовых слов подразделяется на два подпространства – разрешенных и запрещенных комбинаций (кодовых слов).



Рисунок 2 Мханизм обнаружения ошибок

Следует заметить, что если из-за воздействия помех одна разрешенная кодовая комбинация преобразуется в другую разрешенную кодовую комбинацию, то такая ошибка, хотя она и присутствует, обнаружена не будет.

Для исправления ошибок все пространство кодовых слов разбивается на подпространств (непересекающихся).



Рисунок 3 Механизм исправления ошибок

В каждом подпространстве находится одна разрешенная комбинация (обозначена кружком «○») и некоторое количество запрещенных из общего количества (обозначенных точками «•»). Все запрещенные кодовые комбинации распределяются по подпространствам по принципу «близости» к разрешенной кодовой комбинации данного подпространства (т.е. отличающиеся в одном или двух и т.д. знаках от разрешенной кодовой комбинации).

Исправление ошибок производится в два этапа:

1. Определяется кодовое расстояние между пришедшей кодовой комбинацией и всеми разрешенными кодовыми комбинациями.

2. Решение принимается в пользу той разрешенной кодовой комбинции, для которой кодовое расстояние будет наименьшим (т.е. реализуется критерий идеального наблюдателя).

Для получения некоторых количественных соотношений рассмотрим две комбинации и , расстояние между которыми условно обозначено на рис. 4, а, где промежуточные комбинации отличаются друг от друга одним символом.



Рисунок 4 Иллюстрация для определения кодовых расстояний при обнаружении и исправлении ошибок

В общем случае некоторая пара разрешенных комбинаций и , разделенных кодовым расстоянием , изображена на прямой рис. 4, б, где точками указаны запрещенные комбинации. Для того, чтобы в результате ошибки комбинация преобразовалась в другую разрешенную комбинацию , должно исказиться символов. При искажении меньшего числа символов комбинация перейдет в запрещенную комбинацию и ошибка будет обнаружена. Отсюда следует, что ошибка всегда обнаруживается, если ее кратность, т.е. число искаженных символов в кодовой комбинации

Если , то некоторые ошибки также обнаруживаются. Однако полной гарантии обнаружения ошибок здесь нет, т.к. ошибочная комбинация в этом случае может совпасть с какой-либо разрешенной комбинацией. Минимальное кодовое расстояние, при котором обнаруживаются любые одиночные ошибки, .

Процедура исправления ошибок в процессе декодирования сводится к определению переданной комбинации по известной принятой. Расстояние между переданной разрешенной комбинацией и принятой запрещенной комбинацией равно кратности ошибок . Если ошибки в символах комбинации происходят независимо относительно друг друга, то вероятность искажения некоторых символов в *n*-значной комбинации будет равна:

где – вероятность искажения одного символа.

Т.к. обычно , то вероятность многократных ошибок уменьшается с увеличением их кратности, при этом более вероятные меньше расстояния . В этих условиях исправление ошибок может производиться по следующему правилу. Если принята запрещенная комбинация, то считается переданной ближайшая разрешенная комбинация. Например, пусть образовалась запрещенная комбинация (рис. 4, б), тогда принимается решение, что была передана комбинация . Это правило декодирования для указанного распределения ошибок является оптимальным, т.к. оно обеспечивает исправление максимального числа ошибок. В общем случае оптимальное правило декодирования зависит от распределения ошибок (их статистики). Напомним, что аналогичное правило используется в теории потенциальной помехоустойчивости при оптимальном приеме дискретных сигналов, когда решение сводится к выбору того переданного сигнала, который в наименьшей степени отличается от принятого. Нетрудно определить, что при таком правиле декодирования будут исправляться все ошибки кратности:

Минимальное значение , при котором еще возможно исправление любых одиночных, равно 3.

Возможно также построение таких кодов, в которых часть ошибок исправляется, а часть только обнаруживается. Так, в соответствии с рис. 4, в, ошибки кратности исправляются, а ошибки, кратность которых лежит в пределах , только обнаруживаются. Что касается ошибок, кратность которых сосредоточена в пределах , то они обнаруживаются, однако при их исправлении принимается ошибочное решение – считается переданной комбинацией вместо или наоборот.

Существуют двоичные системы связи, в которых решающее устройство выдает, кроме обычных символов 0 и 1, еще так называемый символ стирания θ. Этот символ соответствует приему сомнительных сигналов, когда затруднительно принять определенное решение в отношении того, какой из символов, 0 или 1, был передан. Принятый символ в этом случае стирается. Однако при использовании корректирующего кода возможно восстановление стертых символов.

Если в кодовой комбинации число символов оказалась равным , причем

а остальные символы приняты без ошибок, то такая комбинация полностью восстанавливается.

Корректирующая способность кода возрастает с увеличением . При фиксированном числе разрешенных комбинаций увеличение возможно лишь за счет роста количества запрещенных комбинаций

что, в свою очередь, требует избыточного числа символов , где – количество символов в комбинации кода без избыточности. Можно ввести понятие избыточности кода и количественно определить ее:

При независимых ошибках вероятность определенного сочетания ошибочных символов в *n*-значной кодовой комбинации выражается формулой (2), а количество всевозможных сочетаний ошибочных символов в комбинации зависит от ее длины и определяется известной формулой числа сочетаний:

Отсюда полная вероятность ошибки кратности , учитывающая все сочетания ошибочных символов, равняется:

Используя (7), можно записать формулу, определяющую вероятность отсутствия ошибок в кодовой комбинации, т.е. вероятность правильного приема:

и вероятность правильного корректирования ошибок:

Здесь суммирование производится по всем значениям кратности ошибок , которые обнаруживаются и исправляются. Таким образом, вероятность необнаруженных ошибок равна:

Вероятность , избыточность и число символов являются основными характеристиками корректирующего кода, определяющими, насколько удается повысить помехоустойчивость передачи дискретных сообщений и какой ценой это достигается.

*Циклические коды* (*систематический блочный код*).

Разрешенные комбинации циклического кода можно вычислить последовательно, применяя две операции: циклическую перестановку и суммирование по модулю 2. Циклическая перестановка менее трудоемкая, и это обстоятельство полезно использовать для упрощения процесса вычислений. Расширим производящий момент кода (7,4) до семи членов, применяя нулевые коэффициенты:

Запишем первую комбинацию: .

Применив шесть раз операцию циклической перестановки, можно записать следующие шесть комбинаций. Дальнейшее применение перестановки теряет смысл (они будут приводить к повторению уже имеющихся). Просуммируем по модулю два любые две комбинации из уже имеющихся. Получим комбинацию, содержащую четыре единицы и три нуля. Следующие шесть комбинаций получаются в результате шестикратной циклической перестановки ее элементов.

Чтобы получить пятнадцатую комбинацию, нужно найти среди ранее полученных четырнадцати любую пару взаимно противоположных и просуммировать их по модулю 2 (). Получим комбинацию, состоящую из семи единиц. Шестнадцатую комбинацию (состоящую из 7 нулей) не используют, ее можно вычислить, прибавив к любой из имеющихся комбинаций ее же по модулю 2.

Рассмотрим еще один способ нахождения разрешенных кодовых комбинаций, который широко применяется на практике.

Пусть – информационный полином, наивысшая степень которого (состоит из элементов). Умножим каждый информационный многочлен на выравнивающий полином (в нашем случае ), что соответствует приписыванию справа нулей. Разделим полученный результат на производящий полином , операция деления выполняется по модулю 2.

где – частное от деления, нас не интересует,

– остаток от деления.

Если остаток от деления добавить к , то получим комбинацию, делящуюся без остатка на производящий полином, т.е. разрешенную комбинацию данного кода.

*Пример:*

, ,

, .

, .

Разрешенная кодовая комбинация имеет вид:

, .

В приемнике (декодере) принятая кодовая комбинация делится на производящий полином. При нулевом остатке комбинация принята верно, либо содержит необнаруженную ошибку; остаток от деления указывает, что комбинация принята с обнаруженной ошибкой. По виду остатка может быть определен искаженный элемент в информационной части принятой комбинации (формируется так называемый *синдром*).