

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**  
**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**Кафедра «Системы передачи информации»**

**В. Н. ФОМИЧЕВ, П. М. БУЙ**

# **ПЕРЕДАЧА ДИСКРЕТНЫХ СООБЩЕНИЙ**

**Лабораторный практикум**  
**по дисциплинам «Передача дискретных сообщений»**  
**и «Передача цифровых сигналов»**

**Гомель 2014**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

Кафедра «Системы передачи информации»

В. Н. ФОМИЧЕВ, П. М. БУЙ

# ПЕРЕДАЧА ДИСКРЕТНЫХ СООБЩЕНИЙ

*Одобрено методической комиссией электротехнического факультета  
в качестве лабораторного практикума по дисциплинам  
«Передача дискретных сообщений» и «Передача цифровых сигналов»*

Гомель 2014

УДК 656.251 (075.8)

ББК 39.278

Ф76

Рецензент – академик Международной академии связи, канд. техн. наук,  
профессор *Н. Ф. Семенюта*

**Фомичев, В. Н.**

Ф76      Передача дискретных сообщений : лабораторный практикум /  
В. Н. Фомичев, П. М. Буй ; М-во образования Респ. Беларусь, Бело-  
рус. гос. ун-т трансп. – Гомель : БелГУТ, 2014. – 55 с.  
ISBN 978-985-554-284-2

Изложены цель лабораторных работ, краткие сведения из теории, порядок выполнения, содержание отчета и контрольные вопросы.

Предназначен для студентов специальности 1-37 02 04 «Автоматика, телемеханика и связь на железнодорожном транспорте» специализации 1-37 02 04 02 «Системы передачи и распределения информации», а также может быть использован при изучении других курсов, связанных с передачей дискретных сообщений.

**УДК 656.251 (075.8)**

**ББК 39.278**

**ISBN 978-985-554-284-2**

© Фомичев В. Н., Буй П. М., 2014

© Оформление. УО «БелГУТ», 2014

## ОБЩИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы основаны на применении персональных компьютеров (ПК) в учебном процессе. Они представляют собой программы, имитирующие работу кодека определенного кода с возможностью внесения ошибки в закодированную кодовую комбинацию.

Эти программы разработали в 2010–2013 годах студенты электротехнического факультета УО «БелГУТ»:

- 1) инверсный код; код с четным (нечетным) числом единиц – А. В. Восковцов, А. В. Новак, В. Д. Шурьяков;
- 2) код с постоянным весом; корреляционный код – А. С. Тимошенко;
- 3) линейный код – М. М. Павлов;
- 4) код Хэмминга; циклический код – О. Н. Головина, В. В. Клиновенко.

Для запуска программ необходимо произвести следующие действия. С разрешения преподавателя включить ПК. Подождать, пока производится тестирование всех узлов ПК и загрузка операционной системы *MS Windows*. Затем следует подвести указатель “мыши” на кнопку “Пуск” и нажать левую кнопку манипулятора. В закладке “Главное меню” следует с помощью “мыши” выбрать следующий порядок папок “Лабораторные работы”, “Кодеки”. В папке “Кодеки” содержатся ярлыки программ кодеков циклических, кодов Хэмминга, линейных кодов, а также папка “Простейшие коды”, в которой размещены ярлыки программ кодеков простейших кодов (инверсного кода, кода с четным (нечетным) числом единиц, кода с постоянным весом и корреляционного кода). Следует выбрать требуемый ярлык, и нажатием левой кнопки “мыши” запустить кодек в работу.

## ИССЛЕДОВАНИЕ ПРОСТЕЙШИХ КОДОВ

**Цель работы.** Изучить принципы построения простейших кодов. Убедиться на практике в возможности обнаружения ошибок при использовании простейших кодов. Исследовать корректирующие свойства простейших кодов.

**Подготовка к работе.** Ознакомьтесь с принципом построения простейших кодов, их кодеров и декодеров [1, с. 78–79; 5, с. 312–325].

## Краткие сведения из теории

**Код с повторением (инверсный код).** Это блочный систематический код, имеющий две разновидности. В одной из них имеет место  $S$ -кратное повторение комбинации простого кода  $a_1, a_2, \dots, a_k$ :

$$\underbrace{a_1, a_2, \dots, a_k}_{1 \text{ раз}} \quad \underbrace{a_1, a_2, \dots, a_k}_{2 \text{ раза}} \quad \dots \quad \underbrace{a_1, a_2, \dots, a_k}_{5 \text{ раз}}$$

Вторая разновидность кода с повторением характеризуется  $S$ -кратной передачей каждого символа:

$$\underbrace{a_1, a_1, \dots, a_1}_{S \text{ раз}} \quad \underbrace{a_2, a_2, \dots, a_2}_{S \text{ раз}} \quad \underbrace{a_k, a_k, \dots, a_k}_{S \text{ раз}}$$

Код с повторением имеет длину  $n = S \cdot k$ , число проверочных символов  $r = k(S - 1)$ , минимальное кодовое расстояние  $d_{\min} = S$ . Избыточность этих кодов равна  $R = (S - 1) / S$ . Обычно применяется первая разновидность кода с повторением, имеющая в условиях коррелированных ошибок повышенную помехоустойчивость. Это обусловлено тем, что входящие в одну проверку на четность символы достаточно далеко отстоят друг от друга и с малой вероятностью поражаются одним пакетом ошибок. Число повторений обычно равно 2 ( $d_{\min} = 2$ ) и реже 3 ( $d_{\min} = 3$ ). Код с повторением характеризуется довольно высокими обнаруживающими свойствами при действии пакетов ошибок. Так, при  $S = 2$  всегда обнаруживаются пакеты ошибок длиной до  $n/2$ .

Существует особая разновидность кода с двукратным повторением, обеспечивающая вдвое большее минимальное кодовое расстояние ( $d_{\min} = 4$ ). Это так называемый код *Бауэра* (инверсный код). При четном числе единиц в исходной комбинации повторно она передается без изменения значения символов, а при нечетном – наоборот, с инверсией символов, т. е. заменой единиц на нули и нулей на единицы. Так как  $d_{\min} = 4$ , то гарантированно обнару-

руживаются трехкратные ошибки, а также большинство ошибок более высокой кратности, за исключением случаев, когда одновременно произойдут ошибки в двух символах, стоящих на одинаковых позициях исходной и повторяемой половинах кодовой комбинации. Вероятность таких ошибок  $P_{\text{но}} \approx C_{n/2}^2 P^4$ , где  $p$  и  $q$  – вероятности соответственно неправильного и правильного приемов символа кодовой комбинации,  $p = 1 - q$ .

Обнаружение ошибок в инверсном коде производится путем суммирования по модулю 2 первой и второй половин принятой кодовой комбинации. Причем, если в первых  $n/2$ -символах число единиц будет четным, то остальные  $n/2$ -символы принимаются в прямом виде, а при нечетном – в инвертированном. При приеме кодовой комбинации без ошибки значения всех просуммированных символов результирующей комбинации, равны нулю; появление символов, равных единице, свидетельствует о наличии ошибки.

При небольших длинах кодовой комбинаций (до 10–14) инверсный код по помехоустойчивости не уступает более сложным кодам с такой же величиной избыточности.

**Пример.** Закодировать инверсным кодом следующие кодовые комбинации:

1. 01100

2. 11010

Результат кодирования: 1. 01100 01100

2. 11010 00101

**Код с четным (нечетным) числом единиц** (код с одной проверкой на четность, паритетный код). Блочный систематический  $(k + 1, k)$ -код, комбинации которого образуются путем добавления одного проверочного символа ( $r = 1$ ) к информационным  $k$  символам исходного кода с тем, чтобы общее число единиц в новой  $(k + 1)$ -символьной комбинации было четным (нечетным), т. е. он определяется из соотношения  $b_1 = a_1 \oplus a_2 \oplus a_3 \oplus \dots \oplus a_k$ , или нечетным  $b_1 = \overline{a_1 \oplus a_2 \oplus a_3 \oplus \dots \oplus a_k}$ . Данный код имеет  $d_{\text{min}} = 2$  так же, как и код с двукратным повторением, но обладает гораздо меньшей избыточностью. Код позволяет обнаруживать ошибки нечетной кратности  $\delta = 1, 3, \dots$ , так как ошибки превращают разрешенные кодовые комбинации в запрещенные. Ошибки четной кратности код не обнаруживает (комбинации из разрешенных снова переходят в разрешенные).

Вероятность необнаруженных ошибок равна сумме вероятностей появления ошибок четной кратности, т. е.

$$P_{\text{но}} = C_n^2 p^2 q^{n-2} + C_n^4 p^4 q^{n-4} + C_n^6 p^6 q^{n-6} + \dots,$$

где  $n$  – длина кодовой комбинации,  $n = k + r$ .

В вычислительной технике и системах передачи дискретных сообщений код с четным (нечетным) числом единиц получил широкое применение. Чаще всего он образуется путем добавления одного проверочного символа к комбинациям международного кода № 2 или комбинациям кода № 5.

При воздействии пакетных ошибок помехоустойчивость данного кода значительно ниже, чем у кода с двукратным повторением. Существует также код с двумя проверками на четность. Независимо от длины кодовой комбинации этот код имеет два проверочных символа, один из которых выбирается из условия четности всех информационных символов, а второй – из условия четности всех нечетных (или четных) по номеру информационных символов. Так, при  $k = 5$  значения проверочных символов следующие:

$$b_1 = a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5, \quad b_2 = a_1 \oplus a_3 \oplus a_5.$$

Так же, как код с четным числом единиц, рассматриваемый код имеет  $d_{\min} = 2$ , но обладает большей помехоустойчивостью, поскольку обнаруживает часть ошибок четной кратности – все смежные, т. е. рядом расположенные ошибки.

Пример. Закодировать исходную ( $k = 5$ ) кодовую комбинацию 10010 корректирующим кодом с четным и нечетным числом единиц.

Результат кодирования: 1. 100100  
2. 100101

**Код с постоянным весом** – это блочный несистематический код, разрешенные кодовые комбинации которого имеют постоянный вес (постоянное число единиц).

Число разрешенных кодовых комбинаций кода

$$C_n^W = \frac{n!}{W!(n-W)!},$$

где  $W$  – вес кодовой комбинации (количество единиц в кодовой комбинации).

Данный код имеет  $d_{\min} = 2$  и позволяет обнаруживать все ошибки нечетной кратности и часть четных ошибок.

Не обнаруживаются ошибки четной кратности, когда одновременно в кодовой комбинации одинаковое число нулей переходит в единицы ( $0 \rightarrow 1$ ), а единиц – в нули ( $1 \rightarrow 0$ ). Для случая, когда в кодовой комбинации появляется одна пара таких ошибок, вероятность необнаружения ошибки

$$P_{\text{но}} = P_{1 \rightarrow 0} P_{0 \rightarrow 1} = C_W^1 P q^{W-1} C_{n-W}^1 P q^{n-W-1} = W(n-W) P^2 q^{n-2}.$$

Пример. Составить кодовые комбинации корректирующего кода с постоянным весом при  $n = 7$  и  $W = 3$ .

- Кодовые комбинации будут иметь вид:
1. 1011000
  2. 0101010

**Корреляционный код.** Блочный систематический  $(2k, k)$ -код, в комбинациях которого имеется взаимная связь (корреляция) между символами: каждая единица комбинации исходного кода преобразуется в единицу и нуль ( $1 \rightarrow 10$ ), а нуль – в нуль и единицу ( $0 \rightarrow 01$ ). Данная взаимосвязь символов в закодированной кодовой комбинации обеспечивает обнаружение большинства ошибок, за исключением случаев, когда ошибки происходят одновременно в двух рядом расположенных символах (01 или 10), соответствующих одному символу (0 или 1) исходного кода. Вероятность появления необнаруженной ошибки

$$P_{\text{но}} = C_{n/2}^1 P^2 q^{2n-2}.$$

Код имеет большую избыточность

$$R = r / n = 0,5.$$

Обнаружение ошибки производится суммированием по модулю 2 двух рядом расположенных информационного и проверочного (нечетного и четного) символов. Если в результате получим единицу, значит, ошибки нет, нуль – ошибка в одном из символов принятой пары.

**Пример.** Закодировать корреляционным кодом кодовые комбинации:

1. 10110
2. 01010

Результат кодирования:

1. 1001101001
2. 0110011001

### Порядок выполнения работы

1 Перевести в двоичную систему счисления свой порядковый номер в журнале преподавателя и выполнить его кодирование всеми простейшими кодами:

- инверсным кодом;
- кодом с четным (нечетным) числом единиц при  $k = 5$ ;
- кодом с постоянным весом при  $n = 5, 7$ ;
- корреляционным кодом при  $k = 5$ .

2 Для запуска программ, имитирующих работу кодеков простейших кодов необходимо ознакомиться с разделом “**Общие указания по выполнению лабораторных работ**” и выполнить указанные в нем действия.

#### **Код с повторением (инверсный код)**

1 После появления на экране монитора заставки “**Кодек инверсного кода**” (рисунок 1) необходимо нажать левой кнопкой “мыши” на кнопку “**Вы-**



**бор полинома**”. Здесь под полиномом понимается представление двоичного числа в виде коэффициентов (“1” или “0”) перед соответствующими частями полинома. Например, число 27 может быть представлено следующим полиномом

$$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0,$$

что соответствует информационной последовательности 11011.

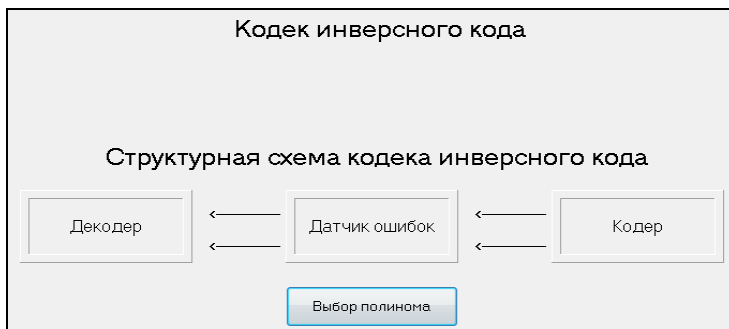


Рисунок 1

2 После нажатия на кнопку **“Выбор полинома”** требуется ввести в кодер свой порядковый номер по журналу преподавателя, переведенный в двоичную систему счисления (рисунок 2). Ввод символов кода осуществляется нажатием клавиш “1” или “0” основного ряда клавиатуры в соответствии с заданной кодовой комбинацией.

3 После правильного набора кодовой комбинации следует нажать на кнопку **“Кодер”**. В результате откроется окно кодера с введенной кодовой комбинацией (рисунок 3).

Полином                      Степень полинома

110011|                      5

Кодер

Рисунок 2

Далее следует нажимать клавишу **“Такт”**, в соответствии со степенью полинома. Это будет соответствовать поступлению в схему кодера тактовых импульсов. В результате закодированная инверсным кодом исходная комбинация будет последовательно, начиная с первого символа и до последнего, выводиться в канал связи с памятью, который в нашем случае заменяет датчик ошибок.

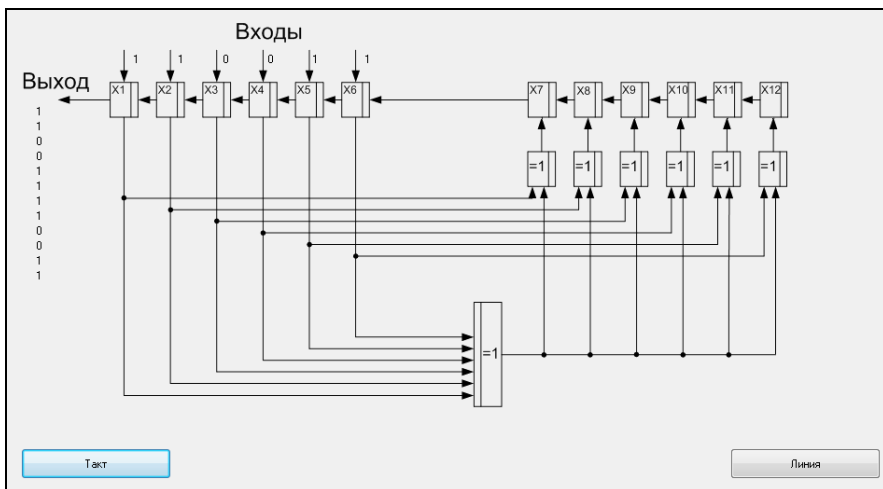


Рисунок 3

4 После завершения ввода закодированной кодовой комбинации в канал связи с памятью, нажатием клавиши “Линия” в окне датчика ошибок можно имитировать появление ошибок в закодированной кодовой комбинации или оставить ее прежней, чтобы убедиться, что декодер работает правильно. Для внесения ошибки требуется нажать на нужный номер символа, который надо изменить (рисунок 4).



Рисунок 4

После внесения ошибки цвет символа изменится на красный.

5 Далее следует нажать клавишу “Декодер” (рисунок 5). Окно представляет собой схему декодера инверсного кода, на выходах регистра сдвига которого показаны значения символов принятой кодовой комбинации и результат ее декодирования (“ОШИБКА!”, “ОШИБКА НЕ ОБНАРУЖЕНА”).

Подача закодированной кодовой комбинации в декодер производится посимвольно последовательным нажатием клавиши “Такт”. После завершения ввода кодовой комбинации из датчика ошибок в декодер появится результат ее декодирования.

6 Для исследования корректирующих свойств кода требуется возвращение в датчик ошибок.

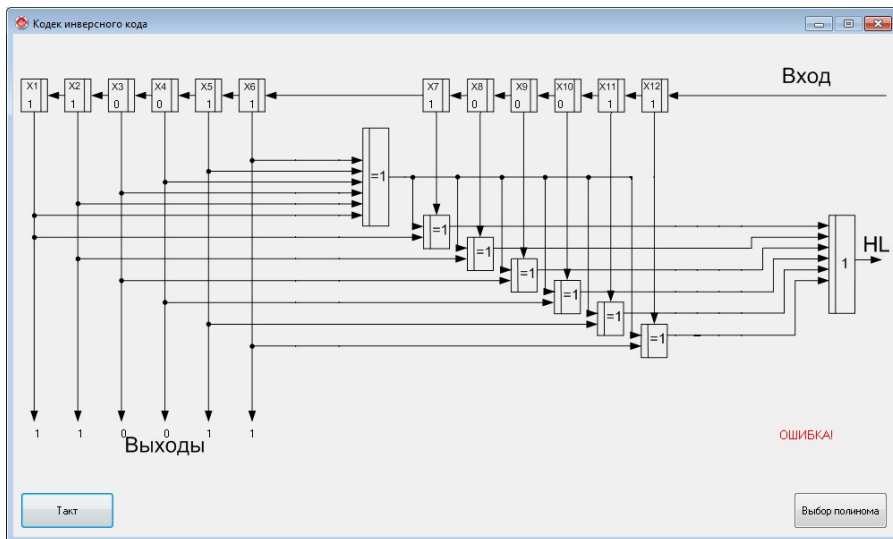


Рисунок 5

Результаты исследований сводятся в таблицу 1.

Таблица 1

Название кода	Кодовая комбинация ( $V$ )	Ошибка ( $E_i$ )	Результат (ошибка обнаружена, исправлена или нет)

7 После завершения исследований требуется выйти из программы, нажав комбинацию клавиш **Alt+F4**.

### **Код с четным (нечетным) числом единиц**

1 После появления на экране монитора заставки “**Кодек кода с проверкой на четность**” (рисунок 6) и структурной схемы кодека следует нажать кнопку “**Выбор полинома**”.

2 После нажатия клавиши “**Выбор полинома**” требуется ввести в кодер свой порядковый номер по журналу преподавателя, переведенный в двоичную систему счисления (рисунок 7).

3 Нажатием клавиши “**Кодер**”, откроется окно с принципиальной схемой кодера (рисунок 8).

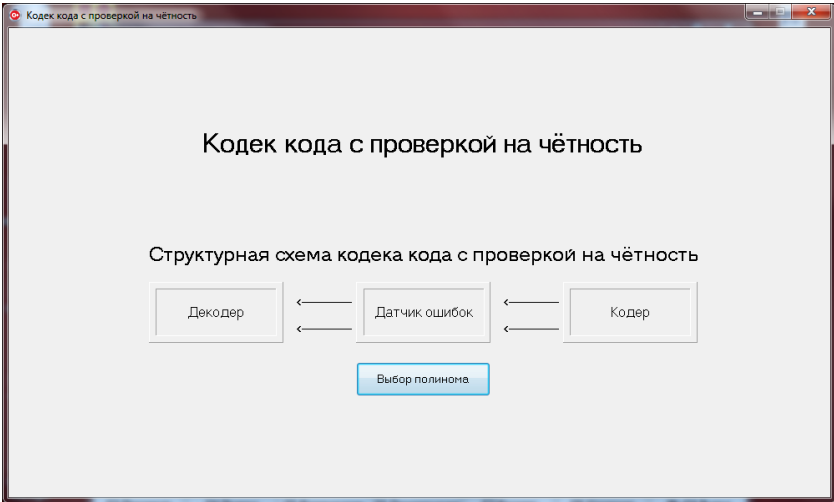


Рисунок 6

Полином: 110011|

Степень полинома: 5

Кодер

Рисунок 7

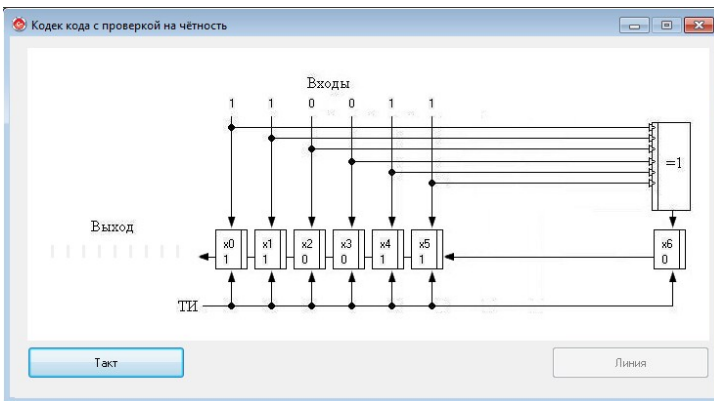


Рисунок 8

Для подачи в линию закодированной комбинации нажимайте “мышью” кнопку “**Такт**”, что соответствует подаче в кодер тактовых импульсов.

4 Далее нажимаем кнопку “**Линия**”. В появившемся окне (рисунок 9) выбираем, какой символ необходимо исказить. Это производится нажатием на искажаемый символ комбинации. Цвет этого символа изменится на красный.

5 Нажатием кнопки “**Декодер**”, попадаем в окно с принципиальной схемой декодера (рисунок 10). Декодер декодирует кодовую комбинацию, и при обнаружении ошибки на мониторе высвечивается надпись “**ОШИБКА**”.

6 Для исследования корректирующих свойств кода нужно повторить испытание с внесением нужных искажений в линии (пункт 4).

Результаты исследований сводятся в таблицу 1.

7 После завершения исследований требуется выйти из программы нажатием комбинации клавиш **Alt+F4**.



Рисунок 9

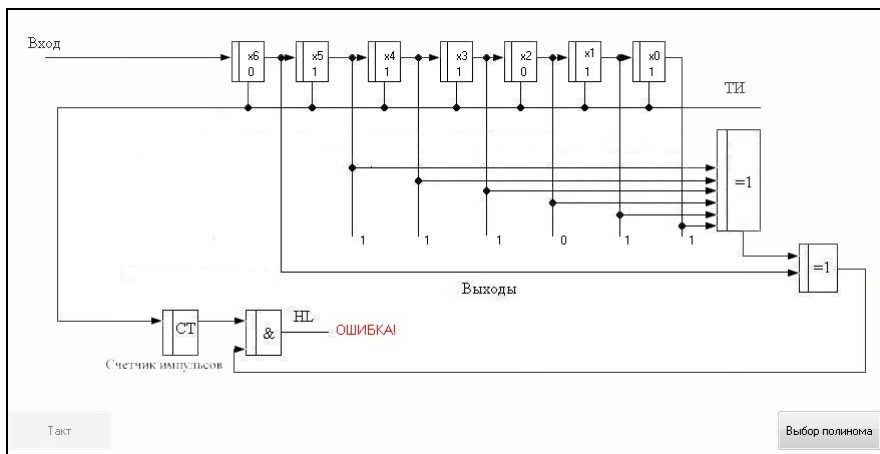


Рисунок 10

### **Код с постоянным весом**

1 После появления на экране монитора заставки “**Код с постоянным весом**” следует выбрать его разновидность (рисунок 11).

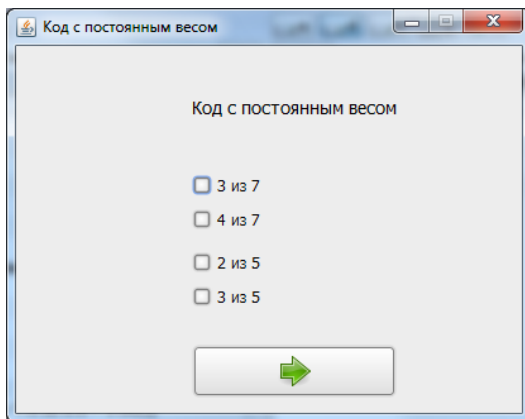


Рисунок 11

2 После выбора разновидности кода, в появившемся окне **“Кодер”** следует ввести кодовую комбинацию (рисунок 12). Требуется ввести в кодер свой порядковый номер по журналу преподавателя, переведенный в двоичную систему счисления. Ввод символов кода осуществляется нажатием клавиш **“1”** или **“0”** основного ряда клавиатуры в соответствии с заданной кодовой комбинацией.

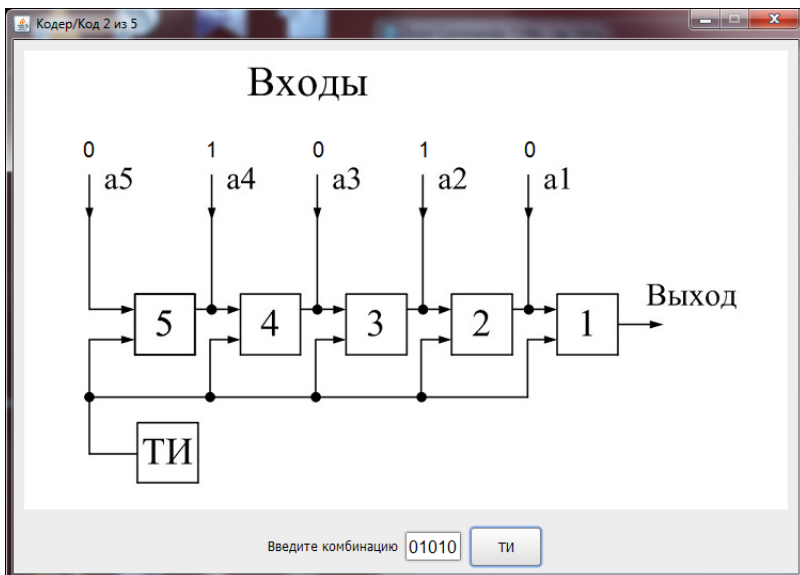


Рисунок 12

3 Нажимая кнопку “ТИ” имитируем подачу символов в линию передачи.

4 Если кодовая комбинация введена правильно, то появится окно “Линия передачи”, в которой можно исказить любой нужный нам символ, для исследования корректирующих свойств кода (рисунок 13).

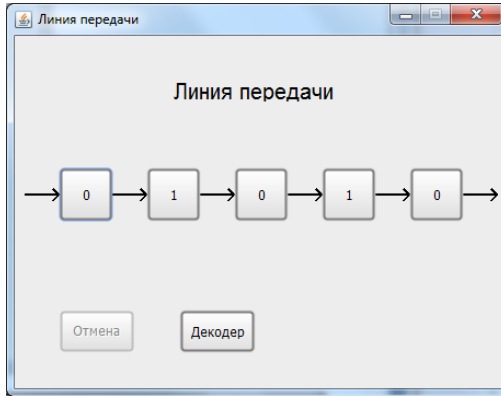


Рисунок 13

5 Из линии передач кодовая комбинация попадает декодер (рисунок 14). Принятие символов в декодере осуществляется нажатием кнопки “ТИ”. Декодер декодирует кодовую комбинацию и при обнаружении ошибки загорается лампочка с надписью “ОШИБКА”.

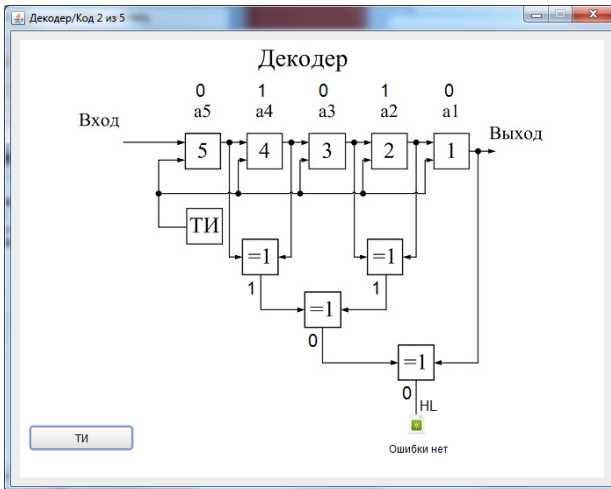


Рисунок 14

6 Для исследования корректирующих свойств кода с постоянным весом, нужно возвратиться в “**Линию передачи**”, в которой можно ввести ошибки различной кратности. Результаты исследований сводятся в таблицу 1.

7 После завершения исследований требуется выйти из программы.

### **Корреляционный код**

1 После появления на экране монитора окна “Кодер корреляционного кода” со структурной схемой кодера (рисунок 15), требуется ввести кодовую комбинацию и нажатием кнопки “**ТИ**” отправить эту комбинацию в линию передачи.

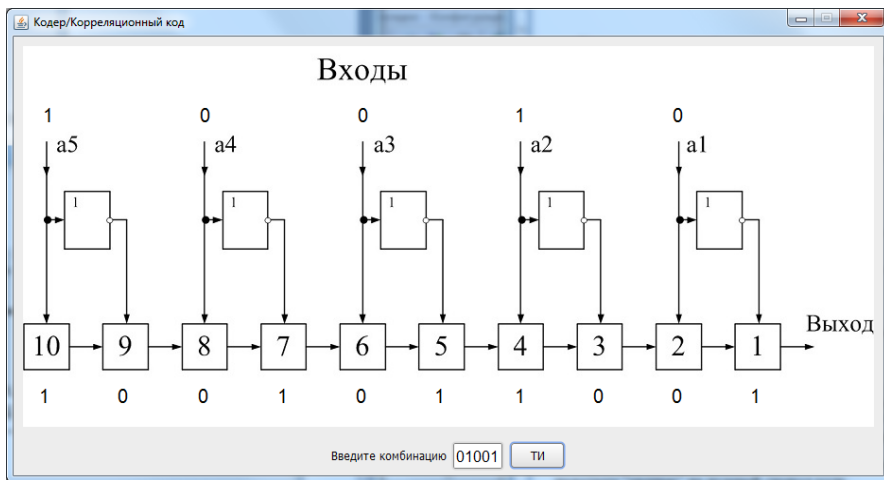


Рисунок 15

2 В окне “**Линия передачи**” (рисунок 16) возможно изменить любой из символов кодовой комбинации нажатием “мышью” на нужный символ кодовой комбинации.

3 По нажатии кнопки “**Декодер**” попадаем в окно “**Декодер корреляционного кода**” (рисунок 17), где нажатием кнопки “**ТИ**” имитируем прием декодером кодовой комбинации. Декодер декодирует кодовую комбинацию и при обнаружении ошибки загорается лампочка с надписью “**ОШИБКА**”.

4 Для исследования корректирующих свойств корреляционного кода нужно возвратиться в “**Линию передачи**”, в которой можно ввести ошибки различной кратности. Результаты исследований сводятся в таблицу 1.

5 После завершения исследований требуется выйти из программы.



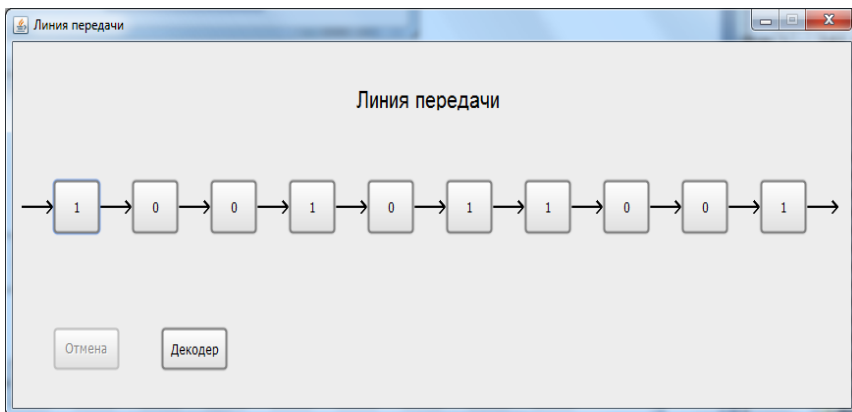


Рисунок 16

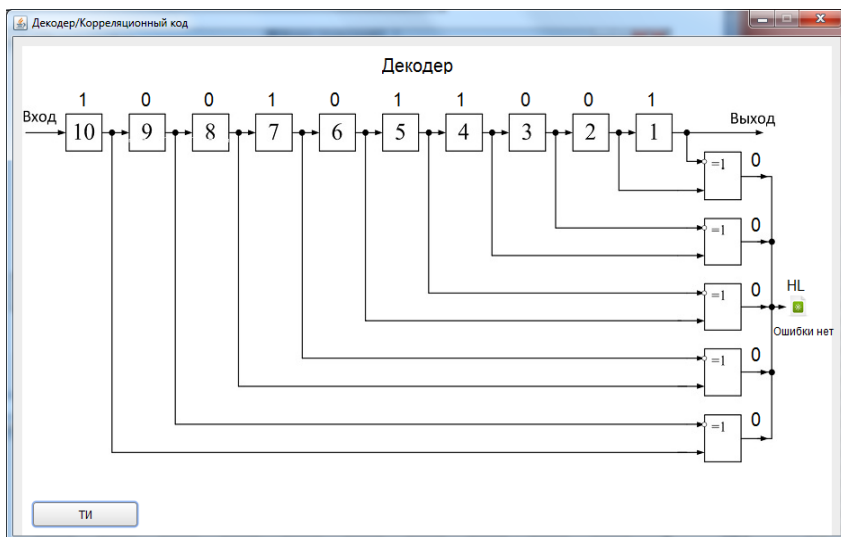


Рисунок 17

### Содержание отчета

- 1 Структурная схема кодера или декодера на определенный тип простейшего кода (по заданию преподавателя).
- 2 Таблица с результатами исследований корректирующих свойств кодов.
- 3 Выводы по результатам исследований.
- 4 Ответы на контрольные вопросы.

## Контрольные вопросы

- 1 Область применения корректирующих кодов.
- 2 Понятие “разрешимые” и “запрещенные” кодовые комбинации.
- 3 Определение блочного, равномерного, делимого, систематического кодов.
- 4 Параметры корректирующих кодов.
- 5 Принцип кодирования и декодирования кода с повторением, его корректирующая способность.
- 6 Принцип кодирования и декодирования кода с четным (нечетным) числом единиц, его корректирующая способность.
- 7 Принцип кодирования и декодирования кода с постоянным весом, его корректирующая способность.
- 8 Принцип кодирования и декодирования корреляционного кода, его корректирующая способность.

## Лабораторная работа № 2

### ИССЛЕДОВАНИЕ ЛИНЕЙНЫХ КОДОВ

**Цель работы.** Изучить принципы построения линейных кодов. Убедиться на практике в возможности обнаружения и исправления ошибок при использовании линейных кодов. Исследовать корректирующие свойства линейных кодов.

**Подготовка к работе.** Ознакомиться с принципом построения линейных кодов, их кодеров и декодеров [1, с. 80–86].

#### Краткие сведения из теории

Линейным называют такие блочные делимые  $(n, k)$ -коды, в которых проверочные символы представляют собой комбинации из информационных символов.

Так как линейные коды образуют алгебраическую группу (замкнутое множество) по отношению к операции сложения по модулю 2 ( $\oplus$ ) они являются также групповыми кодами.

Пусть комбинация  $a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_r$ , где  $a_1, a_2, \dots, a_k$  – информационные,  $b_1, b_2, \dots, b_r$  – проверочные символы, является разрешенной кодовой комбинацией линейного  $(n, k)$ -кода. В линейных кодах проверочные символы являются линейной комбинацией из информационных символов. Это означает, что значение любого проверочного символа

$$b_i = C_1 a_1 \oplus C_2 a_2 \oplus \dots \oplus C_k a_k,$$

где  $C_1, C_2, \dots, C_k$  – коэффициенты, равные 0 или 1.

Нулевая (т. е. состоящая из одних нулей) кодовая комбинация в любом линейном коде является разрешенной кодовой комбинацией, поскольку линейная комбинация нулей есть нуль.

Сложим по модулю 2 две разрешенные кодовые комбинации линейного кода:

$$\begin{aligned} & a_1 a_2 \dots a_k b_1 b_2 \dots b_r \oplus a'_1 a'_2 \dots a'_k b'_1 b'_2 \dots b'_r = \\ & = (a_1 \oplus a'_1)(a_2 \oplus a'_2) \dots (a_k \oplus a'_k)(b_1 \oplus b'_1)(b_2 \oplus b'_2) \dots (b_r \oplus b'_r) \end{aligned}$$

Видно, что

$$b_i \oplus b'_i = C_1(a_1 \oplus a'_1) \oplus C_2(a_2 \oplus a'_2) \oplus \dots \oplus C_k(a_k \oplus a'_k)$$

Таким образом, проверочные символы двух разрешенных кодовых комбинаций образуются по тому же правилу, что и для каждой разрешенной кодовой комбинации. Отсюда сумма двух разрешенных кодовых комбинаций линейного кода также является разрешенной кодовой комбинацией. Такое положение справедливо и при суммировании по модулю 2 любого количества разрешенных кодовых комбинаций, располагая лишь ограниченным их количеством. Все разрешенные кодовые комбинации линейного кода могут быть найдены следующим образом. Выберем несколько  $q$  ненулевых разрешенных кодовых комбинаций. Сложим их по всевозможным сочетаниям: по две, по три, по четыре, ..., по  $q$  комбинаций. Каждое сложение дает нам новую разрешенную кодовую комбинацию.

Процесс получения множества разрешенных кодовых комбинаций можно выразить как

$$x_i = C_1 x_1 \oplus C_2 x_2 \oplus \dots \oplus C_q x_q,$$

где  $x_1, x_2, \dots, x_q$  – исходные разрешенные кодовые комбинации;

$C_1, C_2, \dots, C_k$  – коэффициенты, принимающие значения 1 или 0.

Придавая различным коэффициентам  $C_i$  значения 1 или 0, можно сложить исходные разрешенные кодовые комбинации во всевозможных сочетаниях.

Общее количество полученных кодовых комбинаций составит  $C_q^2 + C_q^3 + \dots + C_q^q$ . Учитывая  $q$  исходных разрешенных кодовых комбинаций и одну нулевую, общее количество разрешенных комбинаций

$$C_q^0 + C_q^1 + C_q^2 + \dots + C_q^q = 2^k, \quad (1)$$

поскольку линейный код содержит  $2^k$  разрешенных кодовых комбинаций. Это условие удовлетворяется при  $q = k$ .

Соотношение (1) справедливо в том случае, если кодовые комбинации, получившиеся в результате сложения, не совпадают ни с исходными кодовыми комбинациями, ни с нулевой. Для этого исходные кодовые комбинации должны быть выбраны специальным образом:

а) все исходные кодовые комбинации должны быть различны. В противном случае при сложении будет получаться нулевая кодовая комбинация;

б) нулевая кодовая комбинация не должна входить в число исходных, поскольку в результате сложения нулевой кодовой комбинации с какой-либо исходной будет получаться та же исходная кодовая комбинация;

в) все исходные кодовые комбинации должны быть линейно независимы, т. е. должно соблюдаться неравенство  $C_1x_1 \oplus C_2x_2 \oplus \dots \oplus C_kx_k \neq 0$  при всех значениях  $C_i$ , за исключением  $C_1 = C_2 = \dots = C_k = 0$ . В противном случае производные кодовые комбинации будут совпадать с исходными;

г) каждая исходная кодовая комбинация, как и любая ненулевая разрешенная кодовая комбинация, должна иметь вес не менее  $d_{\min}$ . Справедливость этого условия легко понять, если учесть, что нулевая кодовая комбинация также является разрешенной, а кодовое расстояние между парой любых кодовых комбинаций кода не должно быть меньше  $d_{\min}$ ;

д) кодовое расстояние между любыми парами исходных кодовых комбинаций не должны быть меньше  $d_{\min}$ .

Подобранные определенным образом  $k$  исходных разрешенных кодовых комбинаций однозначно определяют линейный код. Исходные кодовые комбинации, записанные в виде матрицы, состоящей из  $n$  столбцов и  $k$  строк, образуют порождающую (образующую, производящую) матрицу

$$G_{n,k} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} & b_{11} & b_{12} & \dots & b_{1r} \\ a_{21} & a_{22} & \dots & a_{2k} & b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} & b_{k1} & b_{k2} & \dots & b_{kr} \end{pmatrix}$$

Так как информационные символы занимают обычно первые  $k$  позиций в закодированных кодовых комбинациях, то порождающую матрицу, содержащую  $k$  линейно независимых комбинаций, целесообразно представить состоящей из единичной матрицы размерности  $k \cdot k$ , полностью характеризующей все комбинации простого (исходного) кода,

$$I_k = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

и матрицы-дополнения  $R$  размерности  $r \cdot k$ , содержащей  $r$  столбцов и  $k$  строк проверочных символов:

$$G_{n,k} = \|I_k, R_{r,k}\| = \begin{vmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kr} \end{vmatrix}. \quad (2)$$

Порождающая матрица данного вида называется *канонической*.

При определении значений проверочных символов матрицы-дополнения следует исходить из основных свойств линейного кода.

Поскольку каждая строка единичной матрицы  $k \cdot k$  содержит одну единицу, то вес каждой строки матрицы-дополнения не должен быть меньше  $d_{\min} - 1$  ( $d_{\min}$  – заданное минимальное кодовое расстояние), а сумма по модулю 2 двух любых ее строк должна иметь вес не менее  $d_{\min} - 2$ . В общем случае комбинации матрицы-дополнения находят, перебирая различные комбинации, имеющие вес не менее  $d_{\min} - 1$ . Построенные таким образом исходные  $n$ -символьные комбинации  $(n,k)$ -кода должны быть линейно независимыми.

Процесс кодирования сводится к определению  $r$ -проверочных символов посредством линейных операций над известными  $k$ -информационными символами. Каждый проверочный символ определяется с помощью линейных уравнений. Выбор позиции 1 и 0 в матрице-дополнении  $R_{r,k}$  однозначно определяет позиции тех информационных символов, которые должны участвовать в образовании проверочных символов. Проверочную матрицу линейного кода  $H_{n,r}$ , имеющую  $n$  столбцов и  $r$  строк, можно построить из канонической порождающей матрицы  $G_{n,k}$  (2):

$$H_{n,r} = \|R_{k,r}^T, I_r\| = \begin{vmatrix} b_{11} & b_{21} & \dots & b_{k1} & 1 & 0 & \dots & 0 \\ b_{12} & b_{22} & \dots & b_{k2} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{1r} & b_{2r} & \dots & b_{kr} & 0 & 0 & \dots & 1 \end{vmatrix},$$

где  $T$  – символ транспонированной матрицы.

Транспонированной называют матрицу, строками которой являются столбцы, а столбцами – строки транспонируемой матрицы.

При заданном в лабораторной работе  $d_{\min} = 3$  зависимость между числом проверочных и информационных символов имеет вид

$$r \geq \log_2(n+1), \quad (3)$$

где  $n$  – общее количество символов в закодированной кодовой комбинации,

$$n = k + r;$$

$r$  – наименьшее целое число, при котором удовлетворяет неравенство (3).

Таким образом, при  $k = 5, r = 4$ .

Линейные уравнения задаются следующим образом:

$$b_i = C_1 a_1 \oplus C_2 a_2 \oplus C_3 a_3 \oplus C_4 a_4 \oplus C_5 a_5,$$

где  $C_1, C_2, C_3, C_4, C_5$  – коэффициенты, равные 0 или 1;

$i$  – принимает значение от 1 до 4.

Если линейные уравнения имеют вид

$$b_1 = a_1 \oplus a_2 \oplus a_4 \oplus a_5,$$

$$b_2 = a_1 \oplus a_3 \oplus a_4 \oplus a_5,$$

$$b_3 = a_3 \oplus a_4,$$

$$b_4 = a_1 \oplus a_2 \oplus a_4,$$

то порождающая матрица в каноническом виде будет следующей:

$$G_{n,k} = G_{9,5} = \left\| I_5 R_{4,5} \right\| = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

$\underbrace{\hspace{15em}}_{I_5} \quad \underbrace{\hspace{15em}}_{R_{4,5}}$

Алгоритм получения закодированной кодовой комбинации линейного кода заключается в умножении заданной  $k$ -символьной кодовой комбинации простого кода  $U = a_1, a_2, \dots, a_k$ , представленной в виде матрицы-строки, на каноническую порождающую матрицу  $G$ .

$$V = UG = \left\| a_1, a_2, \dots, a_k \right\| \begin{pmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kr} \end{pmatrix}.$$

Первые  $k$  символов закодированной комбинации линейного кода  $V = a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_r$  длиной  $n = k + r$  повторяют информационные символы исходной кодовой комбинации  $U$ , а последние  $r$  являются проверочными и определяются из линейного уравнения

$$b_i = \sum_{j=1}^k a_j R_{j,i} \quad (i = 1, 2, \dots, r).$$

Если исходная кодовая комбинация простого кода имеет вид

$$U = 10001,$$

то закодированная линейным кодом, она будет следующей:

$$V = UG = \|10001\| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{vmatrix} = 100010001.$$

Проверочная матрица будет иметь вид

$$H_{9,4} = \|R_{5,4}^T I_4\| = \begin{vmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Декодирование принятой кодовой комбинации производится путем определения синдрома  $S = S_4 S_3 S_2 S_1$  по результатам вычислений  $r$ -проверочных уравнений:

$$\begin{aligned} S_1 &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus b_1, \\ S_2 &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus b_2, \\ S_3 &= a_3 \oplus a_4 \oplus b_3, \\ S_4 &= a_1 \oplus a_2 \oplus a_4 \oplus b_4. \end{aligned}$$

Определим синдром при декодировании переданной без ошибок кодовой комбинации 100010001:

$$\begin{aligned} S_1 &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus b_1 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0, \\ S_2 &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus b_2 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0, \\ S_3 &= a_3 \oplus a_4 \oplus b_3 = 0 \oplus 0 \oplus 0 = 0, \\ S_4 &= a_1 \oplus a_2 \oplus a_4 \oplus b_4 = 1 \oplus 0 \oplus 0 \oplus 1 = 0. \end{aligned}$$

Таким образом, синдром кодовой комбинации, принятой без ошибок, равен нулю,  $S = 0000$ .

Если код используется для исправления ошибок (а данный код может исправлять однократные ошибки), то на приемной стороне должно быть заранее определено соответствие между видом синдрома и видом исправленной однократной ошибки.

Допустим, ошибка появилась во втором символе (кодовая комбинация

ошибки  $E = 010000000$ ). Тогда принятая с ошибкой кодовая комбинация будет иметь вид  $V = V \oplus E = 110010001$ . Определим синдром:

$$\begin{aligned} S_1 &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus b_1 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1, \\ S_2 &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus b_2 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0, \\ S_3 &= a_3 \oplus a_4 \oplus b_3 = 0 \oplus 0 \oplus 0 = 0, \\ S_4 &= a_1 \oplus a_2 \oplus a_4 \oplus b_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1. \end{aligned}$$

Таким образом, при появлении ошибки во 2-м символе синдром  $S = 1001$ .

Вычисление синдрома возможно также умножением матрицы-строки принятой  $n$ -символьной комбинации  $V = a_1, a_2, \dots, a_n$  на транспонированную проверочную матрицу  $H^T$ , т. е.  $S = V \cdot H^T$ ,

$$H_{r,n}^T = \left\| \begin{array}{cccc} b_{11} & b_{12} & \dots & b_{1r} \\ b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{kr} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{array} \right\|,$$

а для приведенного примера

$$H_{4,9}^T = \left\| \begin{array}{cccc} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right\|.$$

Учитывая, что принятая кодовая комбинация  $V' = V \oplus E$ , где  $V$  – закодированная линейным кодом и переданная в линию кодовая комбинация (без ошибки);  $E$  – кодовая комбинация ошибки, то синдром



$$S = V \cdot H^T = (V \oplus E) \cdot H^T = \underbrace{V \cdot H^T}_{0} \oplus E \cdot H^T = E \cdot H^T.$$

Таким образом, вид синдрома определяется видом исправляемой (однократной) ошибки.

Для нашего примера имеет место соответствие между однократной ошибкой в символе и видом синдрома, представленное в таблице 1

Таблица 1

Ошибка в символе	1	2	3	4	5	6	7	8	9
Синдром	1101	1001	0110	1111	1100	1000	0100	0010	0001

Принятую кодовую комбинацию  $V$  для исправления ошибки суммируют с соответствующей данному синдрому комбинацией ошибки  $E$ , т. е.

$$V \oplus E = (V \oplus E) \oplus E = V \oplus (E \oplus E) = V.$$

### Порядок выполнения работы

1 Перевести в двоичную систему счисления свой порядковый номер в журнале преподавателя и выполнить кодирование линейным кодом при  $K = 5$ .

2 Выполнить действия, предложенные в разделе “Общие указания по выполнению лабораторных работ”.

3 При запуске программы на экране ПК откроется окно со структурной схемой кодека линейного кода (рисунок 1).



Рисунок 1

Для продолжения работы с программой необходимо нажать кнопку “Начать”.

4 На экране появится окно “Введите коэффициенты  $C_1, C_2, C_3, C_4, C_5$ ” (рисунок 2). Ввод коэффициентов можно произвести вручную, а также при помощи формы “Заполнить уравнения”.

Введите коэффициенты C1, C2, C3, C4, C5

$b_1 = C_1 * a_1 + C_2 * a_2 + C_3 * a_3 + C_4 * a_4 + C_5 * a_5$   
 $b_2 = C_1 * a_1 + C_2 * a_2 + C_3 * a_3 + C_4 * a_4 + C_5 * a_5$   
 $b_3 = C_1 * a_1 + C_2 * a_2 + C_3 * a_3 + C_4 * a_4 + C_5 * a_5$   
 $b_4 = C_1 * a_1 + C_2 * a_2 + C_3 * a_3 + C_4 * a_4 + C_5 * a_5$

Введите исходную кодовую комбинацию

Рисунок 2

При желании можно очистить все заполненные поля (коэффициенты линейных уравнений и исходную кодовую комбинацию) воспользовавшись формой “Очистить”.

5 Для продолжения работы с программой необходимо нажать на кнопку “Продолжить”. В открывшемся окне “**Кодер**” отображается структурная схема выбранного варианта кодера, порождающая матрица в каноническом виде и проверочная матрица. Кодирование осуществляется путем однократного нажатия на кнопку “Такт” в правой нижней части окна (рисунок 3).

6 Для имитации процесса передачи закодированной кодовой комбинации в линию связи необходимо подать 9 тактовых импульсов последовательным нажатием кнопки “Такт”. Для продолжения работы с программой необходимо нажать кнопку “Далее”.

7 Датчик ошибок предназначен для внесения ошибок в любой символ закодированной кодовой комбинации, что позволяет исследовать корректирующую способность линейного кода. Рассмотрим варианты без внесения ошибки (рисунок 4), с внесением однократной ошибки (рисунок 5), двукратной ошибки (рисунок 6).

Для внесения ошибки необходимо нажать на кнопку с номером символа, в который вносится ошибка. При этом его значение инвертируется и выделяется красным цветом.

Датчик ошибок позволяет вводить ошибку любой кратности. Также для всех заданных вариантов рассмотрим схемы и результаты декодирования и исправления/обнаружения внесенных ошибок. Для продолжения необходимо нажать кнопку “Далее”.

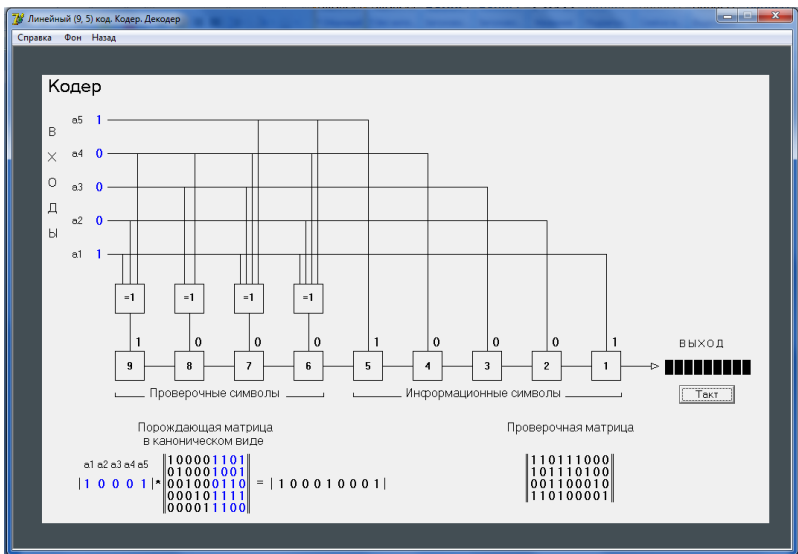


Рисунок 3



Рисунок 4



Рисунок 5



Рисунок 6

8 В окне “Декодер” отображается структурная схема декодера для выбранного варианта линейного кода. Закодированная кодовая комбинация подается на вход декодера; в программе реализована возможность просмотра структурной схемы каждого из 5 решающих устройств путем нажатия на блоки “РУ” на структурной схеме декодера (рисунок 7).

Декодирование происходит путем девятикратного нажатия кнопки “Такт”; в левой нижней части окна выводится информация о внесенных ошибках (номер символа), обнаруженных ошибках (да/нет) и исправленных ошибках (да (номер символа)/нет).

При рассмотрении случая без внесения ошибки на выходе декодера появится исходная кодовая комбинация, в левой нижней части окна информация указывает на отсутствие вносимых ошибок (рисунок 8).

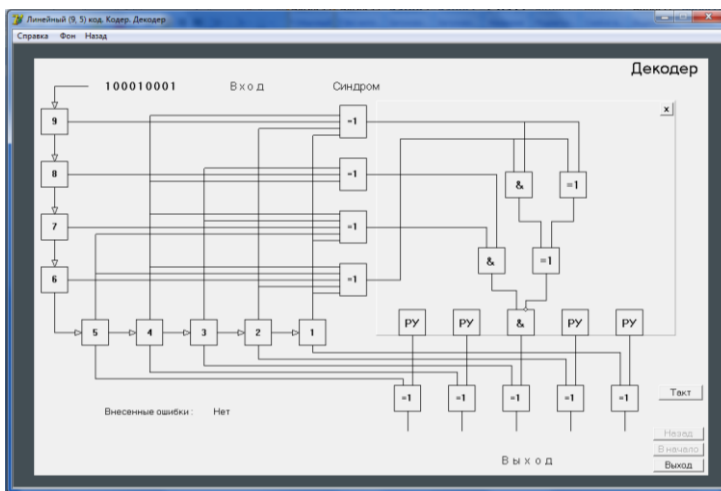


Рисунок 7

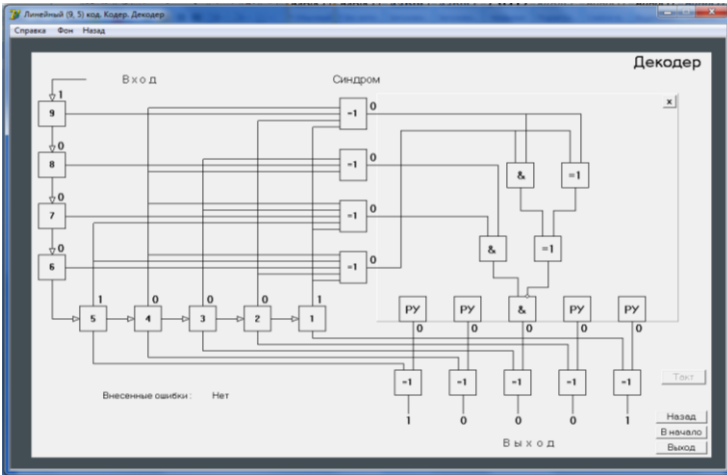


Рисунок 8

При внесении однократной ошибки символ, в который была внесена ошибка, в регистре датчика ошибок отображен красным цветом. На выходе декодера появится исходная кодовая комбинация; в левой нижней части окна представлена информация о том, что в 5-й символ была внесена ошибка. Она была обнаружена и успешно исправлена, что соответствует свойствам данного кода, у которого  $d_{\min} = 3$ , и он исправляет все однократные ошибки (рисунок 9).

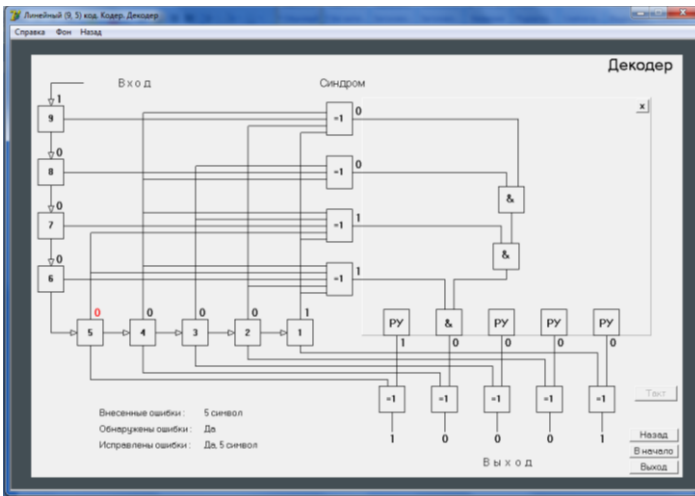


Рисунок 9

При внесении двукратной ошибки символы, в которые были внесены ошибки, отображаются красным цветом. На выходе декодера появится неверно декодированная кодовая комбинация; в левой нижней части окна появится информация сообщающая, что в 5-й и 8-й символы были внесены ошибки, они были обнаружены, но не исправлены (рисунок 10).

Данный результат является закономерным, т. к.  $d_{\min} = 3$ , исправляются только однократные ошибки, ошибки же более высоких порядков могут быть обнаружены, но не исправляются.

9 На каждом этапе работы с программой реализована возможность возвращения к предыдущему окну и выход из программы (при использовании меню в верхней части окна); в окне “Декодер” в правой нижней части окна при использовании кнопок “Назад”, “В начало”, “Выход” реализованы возможности: возвращения к предыдущему окну программы, возвращения к началу работы с программой, выхода из программы.

Данная программная имитация макета кодека позволяет в лабораторных условиях исследовать свойства линейного (9,5)-кода.

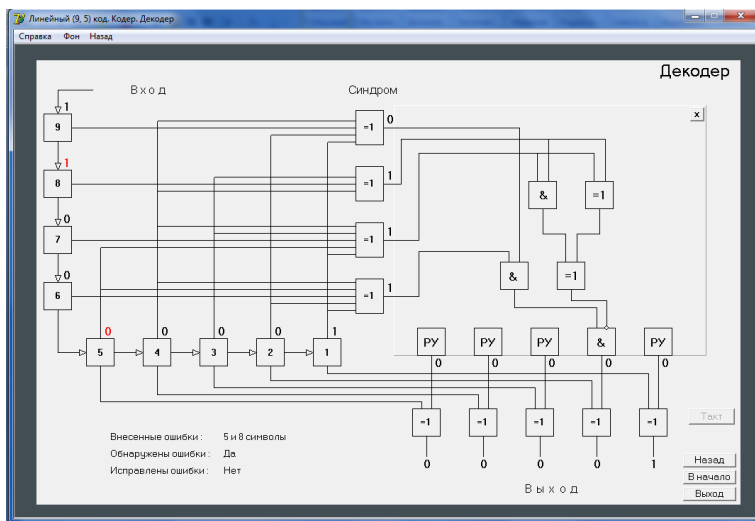


Рисунок 10

10 Результаты исследований сводятся в таблицу 2.

Таблица 2

$V$ (кодовая комбинация)	$E_i$ (ошибка)	Результат (ошибка обнаружена, исправлена или нет)

## Содержание отчета

- 1 Структурная схема кодера, декодера заданного варианта линейного кода.
- 2 Таблица с результатами исследований корректирующих свойств линейного кода.
- 3 Выводы по результатам исследований.
- 4 Ответы на контрольные вопросы.

## Контрольные вопросы

- 1 Понятие минимального кодового расстояния  $d_{\min}$  и его связь с числом обнаруживаемых и исправляемых ошибок.
- 2 Назначение и принцип действия сумматоров по модулю 2 в интегральном исполнении.
- 3 Принцип кодирования линейного кода.
- 4 Принцип декодирования линейного кода.

## Лабораторная работа № 3

### ИССЛЕДОВАНИЕ КОДОВ ХЭММИНГА

**Цель работы.** Изучить принципы построения  $(n,k)$ -кодов Хэмминга. Убедиться на практике в возможности обнаружения и исправления ошибок при использовании кодов Хэмминга. Исследовать корректирующие свойства кодов Хэмминга.

**Подготовка к работе.** Ознакомиться с принципом построения кодов Хэмминга, их кодеров и декодеров [1, с. 86–89].

### Краткие сведения из теории

Коды Хэмминга с минимальным кодовым расстоянием  $d_{\min} = 3$  и  $d_{\min} = 4$  относятся к оптимальным систематическим кодам, исправляющим все варианты одиночных ошибок. Оптимальным  $(n,k)$ -кодом является такой код, который обеспечивает наименьшую вероятность неправильного декодирования среди всех кодов с теми же значениями  $n$  и  $k$ . Коды Хэмминга совершенны, так как исправляют только все одиночные ошибки. Коды с минимальным кодовым расстоянием  $d_{\min} = 4$  образуются на основе кодов с  $d_{\min} = 3$  путем введения еще одного проверочного символа, дополняющего закодированную кодовую комбинацию до четного числа единиц. Количество необходимых проверочных символов  $r$  в кодовой комбинации при заданном числе  $k$  определяется следующим образом.

При передаче информации закодированная кодовая комбинация может быть принята без ошибок (один случай) или может быть искажена любой из  $n$  ее символов ( $n$  случаев). Следовательно, с помощью проверочных символов необходимо контролировать все эти  $(n + 1)$  случаи. Значит, число про-

верочных символов в кодовой комбинации можно определить из неравенства  $2^r \geq n + 1 = k + r + 1$ . Так как  $r = n - k$ , то

$$2^k \leq \frac{2^n}{(n+1)}, \quad (1)$$

где  $n$  и  $k$  могут принимать только целые значения.

Это выражение является исходным при определении длины закодированной кодовой комбинации по заданному числу информационных символов.

Порядок формирования проверочных символов определяется задаваемым алгоритмом обнаружения и исправления искаженного символа, т. е. требованием, предъявляемым к синдрому.

Синдром – двоичное  $r$ -разрядное число, записанное по результатам  $r$ -проверок ( $S = S_r \dots S_3 S_2 S_1$ ), указывает номер искаженного символа закодированной кодовой комбинации.

Исходя из изложенного требования к формированию проверочных символов определяют, какие из символов закодированной кодовой комбинации должны охватываться каждой из  $r$ -проверок.

Пусть  $S = 1$ . Это значит, что один из символов закодированной кодовой комбинации, охватываемых первой проверкой, искажен. Наличие "1" в младшем разряде синдрома  $S_r \dots S_3 S_2 S_1$  указывает, что номер искомого искаженного символа нечетный, т. к. единицу в первом разряде имеют все нечетные десятичные номера символов, представленные в двоичной системе счисления. Следовательно, они и должны охватываться первой проверкой:

$$S_1 = a_1 \oplus a_3 \oplus a_5 \oplus a_7 \oplus \dots$$

Вторая проверка  $S_2$  охватывает те символы принятой кодовой комбинации, которые имеют десятичный номер, представленный в двоичной системе счисления, с "1" во втором разряде. Таковыми являются символы с номерами 2, 3, 6, 7, 10, 11, 14, 15, ..., которые и должны охватываться второй проверкой:

$$S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7 \oplus a_{10} \oplus \dots$$

Аналогичным образом можно найти символы, охваченные третьей, четвертой и т. д. проверками (таблица 1).

Таблица 1

Номер проверки	Проверяемые символы
1	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, ...
2	2, 3, 6, 7, 10, 11, 14, 15, 18, ...
3	4, 5, 6, 7, 12, 13, 14, 15, 20, ...
4	8, 9, 10, 11, 12, 13, 14, 15, 24, ...



В качестве проверочных символов удобно взять такие, которые входят только один раз в каждую проверку. Номера проверочных символов в закодированной кодовой комбинации определяются соотношением  $2^{i-1}$ , где  $i = 1, 2, 3, \dots$ , т. е.  $a_1, a_2, a_4, a_8$  и т. д.

В качестве примера рассмотрим преобразование пятисимвольной кодовой комбинации простого кода ( $k = 5$ ) в закодированную кодом Хэмминга кодовую комбинацию, исправляющую одиночные ошибки. Согласно условию (4) получим  $n = 9, r = 4$ . Причем символы закодированной кодом Хэмминга комбинации  $a_1, a_2, a_4, a_8$  будут проверочными,  $a_3, a_5, a_6, a_7, a_9$  – информационными. Для пятисимвольной кодовой комбинации простого кода вида 10110 ( $a_3 = 1, a_5 = 0, a_6 = 1, a_7 = 0, a_9 = 1$ ) определим численные значения проверочных символов путем суммирования по модулю 2 определенных информационных символов:

$$\begin{aligned} a_1 &= a_3 \oplus a_5 \oplus a_7 \oplus a_9 = 0 \oplus 1 \oplus 0 \oplus 1 = 0, \\ a_2 &= a_3 \oplus a_6 \oplus a_7 = 0 \oplus 1 \oplus 0 = 1, \\ a_4 &= a_5 \oplus a_6 \oplus a_7 = 1 \oplus 1 \oplus 0 = 0, \\ a_8 &= a_9 = 1. \end{aligned}$$

Тогда закодированная кодом Хэмминга пятисимвольная кодовая комбинация простого кода будет иметь вид 110110010.

Допустим, что закодированная кодовая комбинация 110110010 при передаче по линии связи подверглась воздействию помехи и в 7-м символе появилась ошибка – 111110010. В результате четырех проверок при декодировании получим:

$$\begin{aligned} S_1 &= a_1 \oplus a_3 \oplus a_5 \oplus a_7 \oplus a_9 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1, \\ S_2 &= a_2 \oplus a_3 \oplus a_6 \oplus a_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1, \\ S_3 &= a_4 \oplus a_5 \oplus a_6 \oplus a_7 = 0 \oplus 1 \oplus 1 \oplus 1 = 1, \\ S_4 &= a_8 \oplus a_9 = 1 \oplus 1 = 0. \end{aligned}$$

Следовательно, синдром, представленный в виде двоичного числа  $S = S_4 S_3 S_2 S_1 = 0111$  (что соответствует десятичному числу 7), и указывает номер ошибочного символа, значение которого необходимо заменить на противоположное.

Структурная схема кодера (9,5)-кода Хэмминга приведена на рисунке 1. Информационные символы пятисимвольного простого кода параллельным кодом поступают в 3-, 5-, 6-, 7-, 9-й триггеры регистра и одновременно на входы сумматоров по модулю 2  $w_1 \dots w_4$ , где в соответствии с алгоритмом кодирования происходит формирование проверочных символов  $a_1, a_2, a_4, a_8$ , которые записываются в триггеры 1-, 2-, 4-, 8-го регистра.

Далее все символы закодированной кодовой комбинации под действием тактовых импульсов ТИ поступают в линию с памятью.

Структурная схема декодера (9,5)-кода Хэмминга приведена на рисунке 2. Она состоит из 9-триггерного регистра (принимающего последовательным кодом информацию с линии связи), схемы обнаружения ошибок  $w_1 \dots w_4$  (сумматоры по модулю 2), дешифратора (ДС) и схемы исправления ошибок  $w_5 \dots w_{13}$  (сумматоры по модулю 2).

Поступающая из линии связи закодированная кодовая комбинация записывается в 1–9-й триггеры регистра. После приема всех символов кодовой комбинации на выходах сумматоров  $w_1-w_4$  образуется двоичная кодовая комбинация синдрома:  $S = S_4 S_3 S_2 S_1$ .

При отсутствии ошибок в принятой кодовой комбинации  $S = 0000$ . Если кодовая комбинация принята с обнаруженными ошибками, то синдром не будет нулевым. Для исправления ошибок в декодере имеется дешифратор синдрома и схема исправления ошибок.

Дешифратор предназначен для преобразования  $r$ -символьной кодовой комбинации синдрома  $S$ , формируемой сумматорами  $w_1-w_4$ , по результатам четырех проверок на четность, в  $n$ -символьную комбинацию ошибки  $E_i$  с единицей на позиции ошибочного символа принятой кодовой комбинации.

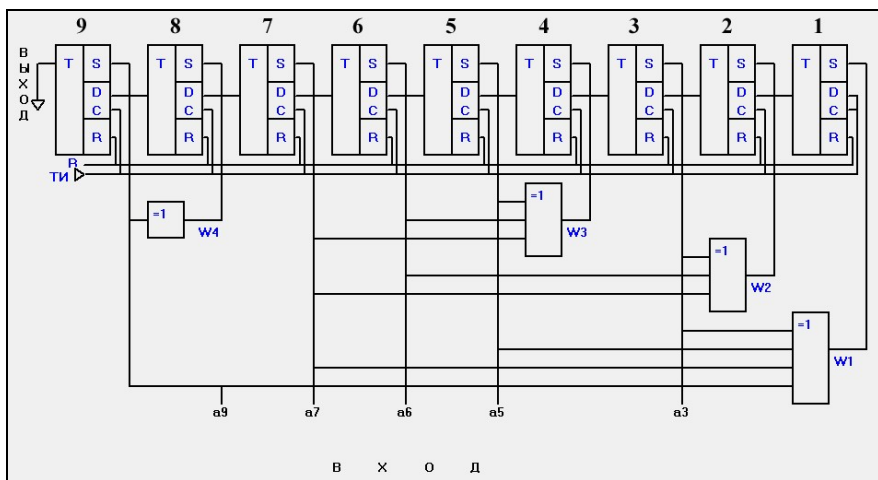


Рисунок 1

Схема исправления ошибок состоит из  $n$  сумматоров по модулю 2, на входы которых поступают символы принятой кодовой комбинации из регистра и исправляющие символы  $E_i$  из дешифратора. Так как принятая с ошибкой кодовая комбинация  $V' = V \oplus E_i$ , где  $V$  – переданная кодовая комбинация,  $E_i$  – кодовая комбинация ошибки, то после схемы исправления ошибок получим

$$V' \oplus E_i = (V \oplus E_i) \oplus E_i = V \oplus (E_i \oplus E_i) = V,$$

где  $V'$  – принятая кодовая комбинация.

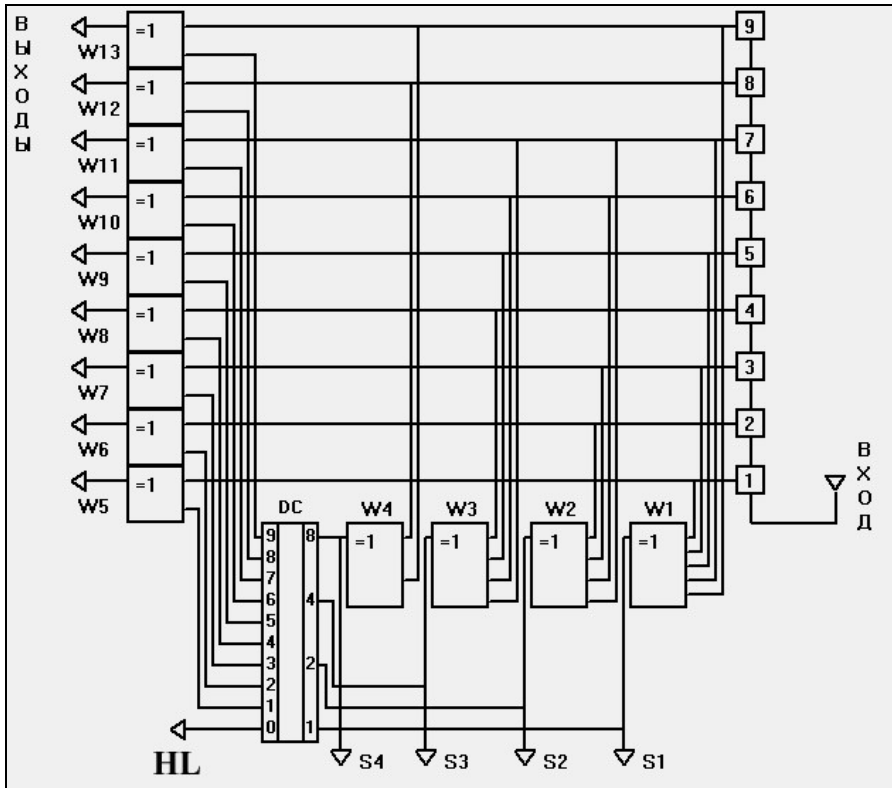


Рисунок 2

### Порядок выполнения работы

1 Перевести в двоичную систему счисления свой порядковый номер в журнале преподавателя и выполнить кодирование первичной пятисимвольной кодовой комбинации кодом Хэмминга.

2 Выполнить действия, предложенные в разделе “Общие указания по выполнению лабораторных работ”.

3 При запуске программы в первоначальном варианте на экране ПК откроется окно “Структурная схема кодека кодов Хэмминга” (рисунок 3).

Для продолжения работы с программой необходимо нажать кнопку “Полиномы” в правой верхней части окна.



Рисунок 3

При этом откроется окно с формой для ввода кодовой последовательности (рисунок 4). Максимальное число символов в кодовой последовательности для данной программы составит пять символов. После ввода кодовой комбинации необходимо нажать кнопку “>>” для ввода информации в память программы. По завершении вышеописанных действий появится возможность перейти к окну “Кодер” (рисунок 5), для этого в правой верхней части окна следует нажать кнопку “Кодер”.

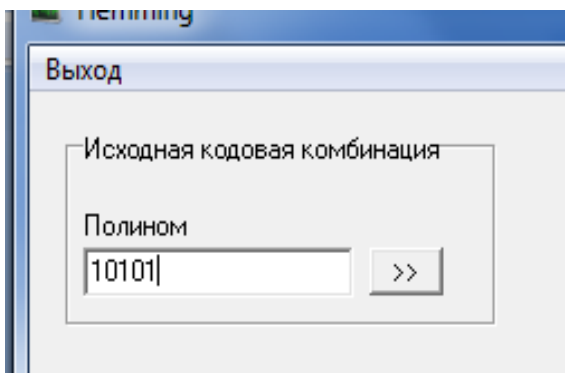


Рисунок 4

4 На входе кодера появится введенная информационная последовательность  $(a_3, a_5, a_6, a_7, a_9)$ , которая параллельным кодом поступает в триггеры 3-, 5-, 6-, 7-, 9-го регистра и одновременно на входы сумматоров  $w_1-w_4$ , где в соответствии с алгоритмом кодирования происходит формирование проверочных символов  $a_1, a_2, a_4, a_8$ , которые записываются в триггеры 1-, 2-, 4-, 8-го регистра.

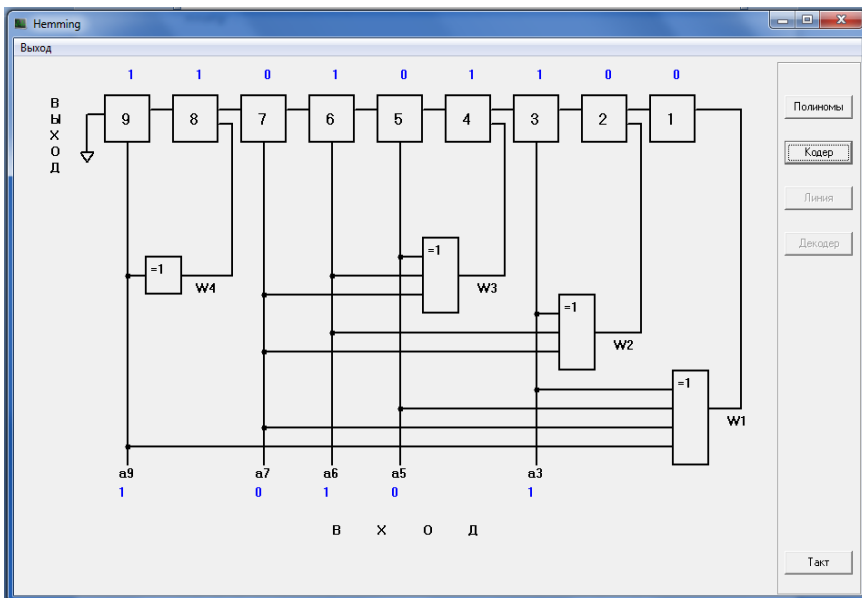


Рисунок 5

Далее все символы закодированной кодовой комбинации под действием тактовых импульсов ТИ (путем нажатия кнопки “Такт” в правой нижней части окна) поступают на выход кодера и подаются в линию с памятью (датчик ошибок).

5 После этого путем нажатия кнопки “Линия” открывается окно “Линия” (рисунок 6). Данная форма имитирует реальную линию связи, при передаче по которой в закодированную кодовую комбинацию может быть внесена ошибка.

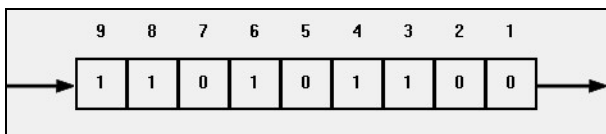


Рисунок 6

Ошибку в кодовую последовательность можно внести нажатием на кнопку под порядковым номером символа. При этом значение символа кодовой последовательности инвертируется.

Рассмотрим варианты без внесения ошибки (см. рисунок 6), с внесением однократной ошибки (рисунок 7), двукратной ошибки (рисунок 8).

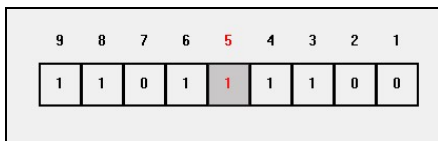


Рисунок 7

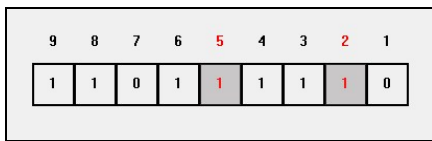


Рисунок 8

Также далее для всех заданных вариантов рассмотрим схемы и результаты декодирования и исправления/обнаружения внесенных ошибок.

6 Для того, чтобы перейти к окну “Декодер” (рисунок 9) необходимо нажать кнопку “Декодер” в правой части окна.

Рассмотрим работу декодера для оговоренных вариантов: без внесения ошибки (рисунок 10), с внесением однократной ошибки (рисунок 11), с внесением двухкратной ошибки (рисунок 12).

Нажатием кнопки “Такт” кодовая комбинация подается в регистр. На выходах схемы обнаружения ошибок можем видеть значение синдрома, в данном случае имеем нулевой синдром ( $S = 0000$ ). На выводе “0” дешифратора  $DC$  также нулевое значение, этот вывод предназначен для индикации наличия/отсутствия ошибки. На выходе декодера появится исходная закодированная кодовая комбинация.

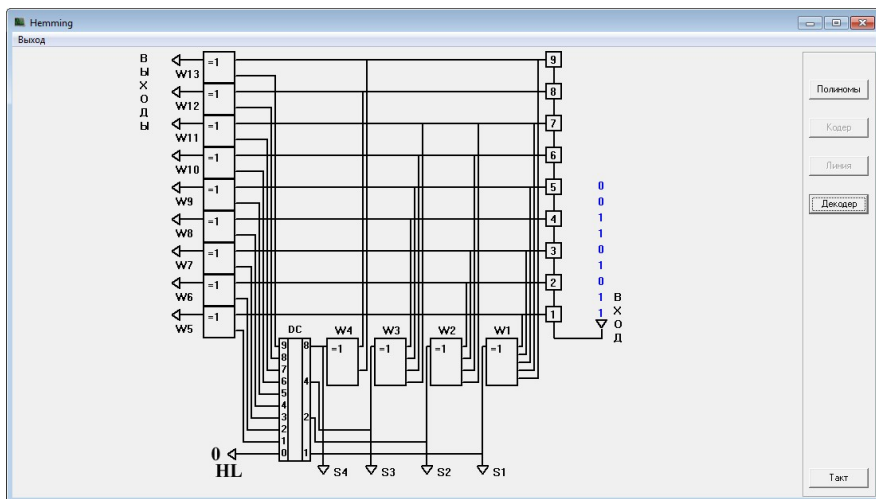


Рисунок 9

При внесении однократной ошибки символ, в который была внесена ошибка, в регистре отображен красным цветом. На выходе схемы обнаружения ошибки синдром  $S = 0101$ , что соответствует значению 5 в десятичной системе счисления.

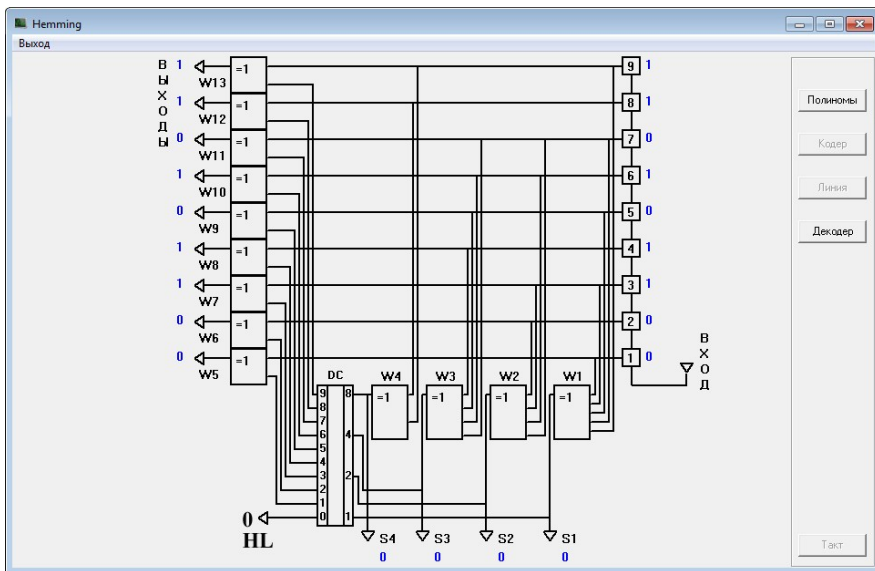


Рисунок 10

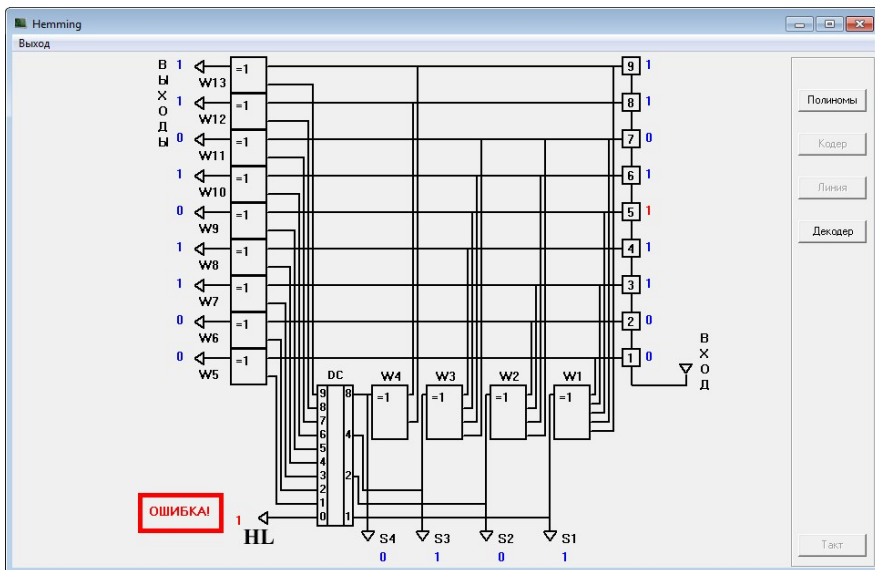


Рисунок 11

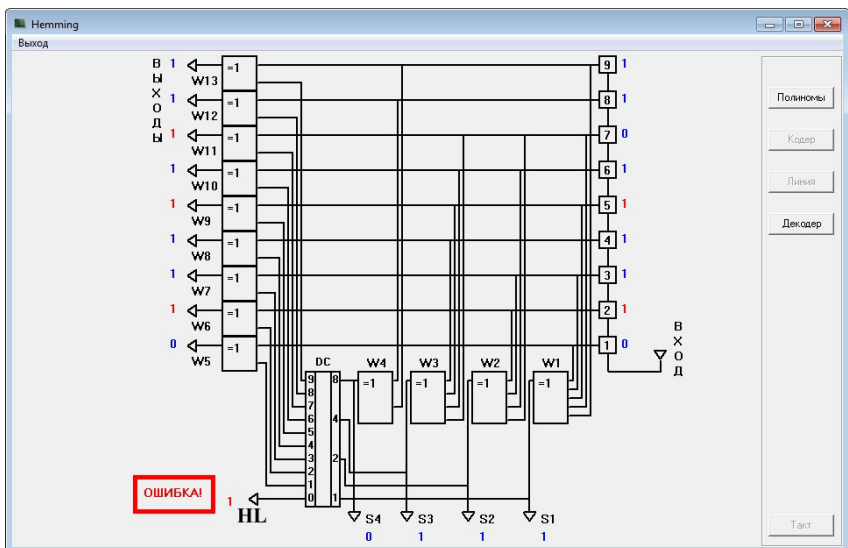


Рисунок 12

На выводе “0” дешифратора значение 1 и сообщение о наличии ошибки в принятой кодовой комбинации. На выходе декодера исходная закодированная кодовая комбинация, т. е. однократная ошибка была исправлена, что соответствует свойствам кода Хэмминга ( $d_{\min} = 3$  исправляет все однократные ошибки).

При внесении двукратной ошибки символы, в которые были внесены ошибки, отображаются красным цветом. На выводе “0” дешифратора значение 1, что соответствует обнаружению ошибок. Однако согласно свойствам кода двукратная ошибка не может быть исправлена. На выводе декодера кодовая комбинация с ошибками отображается красным цветом.

Данная программная имитация макета кодера/декодера позволяет в лабораторных условиях исследовать свойства кода Хэмминга.

7 Результаты исследований сводятся в таблицу 2.

Таблица 2

$V$ (кодовая комбинация)	$E_i$ (ошибка)	Результат (ошибка обнаружена, исправлена или нет)

### Содержание отчета

- 1 Структурная схема кодера, декодера (по заданию преподавателя).
- 2 Таблица с результатами исследований корректирующих свойств кода.



3 Выводы по результатам исследований.

4 Ответы на контрольные вопросы.

### Контрольные вопросы

1 Определение систематического, оптимального и совершенного кодов.

2 Принцип кодирования кодом Хэмминга.

3 Принцип обнаружения и исправления ошибок кодом Хэмминга.

4 Назначение и принцип действия дешифратора в интегральном исполнении.

## Лабораторная работа № 4

### ИССЛЕДОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ

**Цель работы.** Изучить принципы построения циклических  $(n,k)$ -кодов. Убедиться на практике в возможности обнаружения и исправления ошибок при использовании циклических кодов. Исследовать корректирующие свойства циклических кодов.

**Подготовка к работе.** Ознакомиться с принципами построения циклических кодов, их кодеров и декодеров [1, с. 89–96].

#### Краткие сведения из теории

Циклические коды являются разновидностью систематических кодов и обладают всеми их свойствами. Первоначально они были созданы с целью упрощения схем кодирования и декодирования. Впоследствии обнаружилось высокие корректирующие свойства этих кодов, что и обеспечило им широкое распространение на практике. Упрощение кодирующих (кодеров) и декодирующих (декодеров) устройств достигается за счет наложения дополнительных ограничений на подгруппу разрешенных кодовых комбинаций циклического кода. В коде на подгруппу разрешенных кодовых комбинаций, помимо условия замкнутости по отношению к операции сложения, накладывается также ограничение по отношению к операции умножения (циклического сдвига символов кодовых комбинаций на одну позицию вправо или влево).

При построении циклических кодов кодовые комбинации принято представлять в виде многочленов переменной  $x$ :

$$P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0,$$

где  $a_{n-1}, \dots, a_1, a_0$  – коэффициенты, принимающие значения 0 или 1.

Основное свойство рассматриваемых кодов состоит в том, что циклический сдвиг разрешенной кодовой комбинации также является разрешенной кодовой комбинацией. Циклический сдвиг эквивалентен умножению на  $x$  кодовой комбинации, записанной в виде многочлена:

$$xP(x) = a_{n-1}x^n + a_{n-2}x^{n-1} + \dots + a_1x^2 + a_0x.$$

Так как в кодовой комбинации, имеющей длину  $n$ , степень многочлена не может превышать  $n - 1$ , то следует принимать  $x^n = 1$ . Тогда окончательный результат циклического сдвига многочлена на одну позицию влево

$$xP(x) = a_{n-2}x^{n-1} + \dots + a_1x^2 + a_0x + a_{n-1}.$$

Принцип построения кодовых комбинаций циклических кодов заключается в следующем. Умножим каждую кодовую комбинацию  $A(x)$  простого  $k$ -символьного кода на  $x^r$ , где  $r = n - k$ , а затем разделим на порождающий (образующий) многочлен  $q(x)$  степени  $r$ . В результате умножения  $A(x)$  на  $x^r$  степень каждого слагаемого  $x_i$ , входящего в многочлен  $A(x)$ , повысится на  $r$ . При делении произведения  $x^r A(x)$  на  $q(x)$  получим частное  $Q(x)$  такой же степени, что и  $A(x)$ . Если произведение  $x^r A(x)$  не делится нацело на  $q(x)$ , то получим остаток  $R(x)$ :

$$\frac{x^r A(x)}{q(x)} = Q(x) + \frac{R(x)}{q(x)}.$$

Поскольку частное  $Q(x)$  имеет ту же степень, что и  $A(x)$ , то оно также является кодовой комбинацией простого  $k$ -символьного кода. Умножая обе части равенства на  $q(x)$  после переноса влево произведения  $Q(x)q(x)$  получим

$$F(x) = Q(x)q(x) = x^r A(x) + R(x).$$

Таким образом, кодовая комбинация циклического кода может быть получена двумя способами:

- 1) неразделимый код – умножением  $k$ -символьной кодовой комбинации простого кода на образующий многочлен  $q(x)$ ;
- 2) делимый код – умножением кодовой комбинации простого кода на одночлен  $x^r$  и добавлением к этому произведению остатка от деления произведения  $x^r A(x)$  на  $q(x)$ .

Неразделимость кода значительно усложняет процесс декодирования, поэтому на практике используется второй способ построения кодовых комбинаций.

В соответствии с принципом построения циклического кода в основе схем кодирования и декодирования лежат регистры сдвига, выполняющие операции умножения и деления многочленов. В лабораторной работе рассмотрены кодирующие и декодирующие устройства на следующие виды порождающих многочленов:  $q_1(x) = x^4 + x + 1$  (рисунки 1, 2),  $q_2(x) = x^4 + x^3 + 1$  (рисунки 3, 4),  $q_3(x) = x^4 + x^3 + x^2 + x + 1$  (рисунки 5, 6).

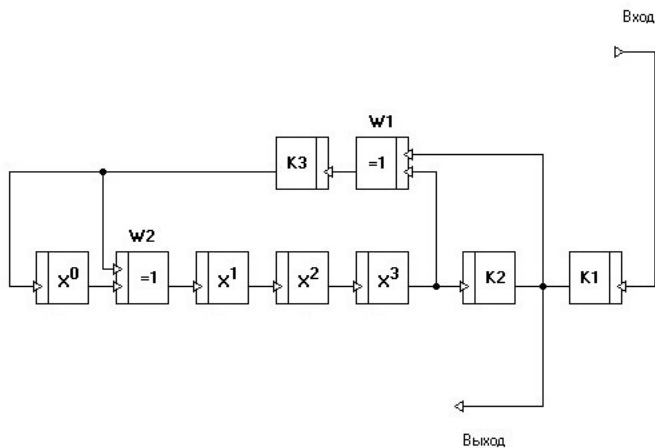


Рисунок 1

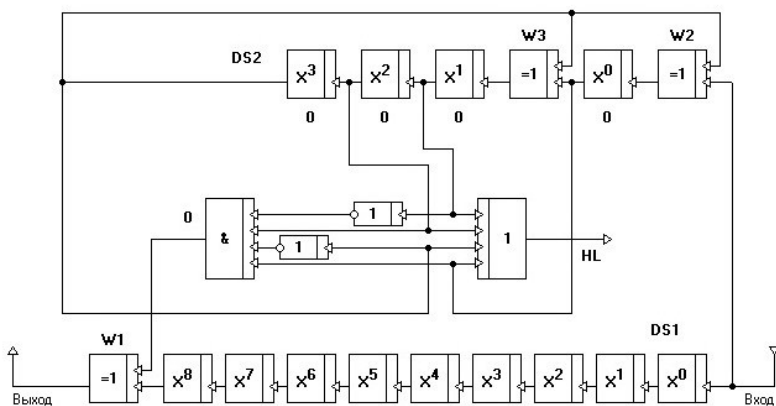


Рисунок 2

Из схем видно, что в состав кодера входят  $r$ -триггерный регистр сдвига, три ключа (K1, K2, K3) и двухвходовые сумматоры по модулю 2 ( $w$ ), причем их число на единицу меньше числа ненулевых членов образующего многочлена  $q(x)$ . Структура логических обратных связей в регистре сдвига, число элементов которого равно максимальной степени образующего многочлена, полностью определяется видом порождающего многочлена.

Сумматоры  $w_1, \dots, w_{r-1}$  присутствуют в схеме в том случае, если соответствующие коэффициенты образующего многочлена  $a_1, \dots, a_{r-1}$  равны 1.

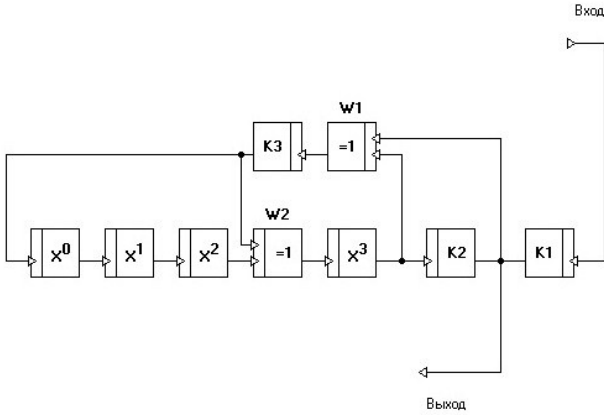


Рисунок 3

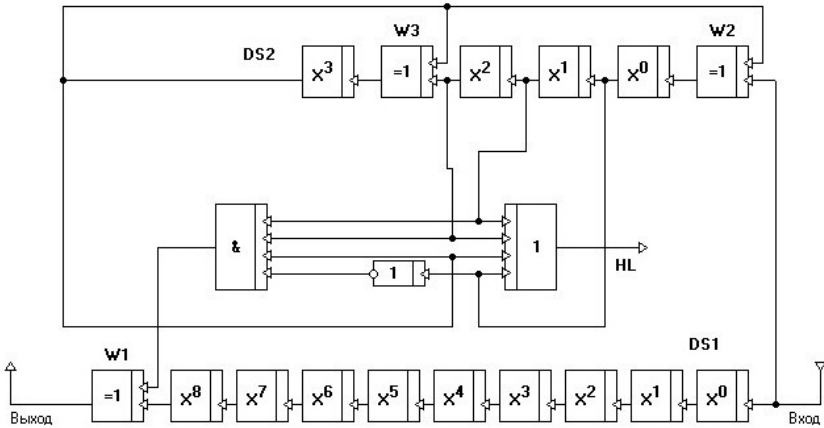


Рисунок 4

Схема кодирующего устройства циклического кода работает следующим образом. Сначала ключ K2 закрыт, а ключи K1 и K3 открыты. Информационный многочлен  $A(x)$  поступает одновременно на выход кодера, начиная со старшего члена (идет умножение на одночлен  $x^r$ ) и сумматор  $w_1$ . В процессе ее прохождения за  $k$ -тактовых импульсов в ячейках регистра сдвига накапливается  $r$  проверочных символов (осуществляется деление  $x^r A(x)$  на образующий многочлен  $q(x)$ ). После  $k$ -го тактового импульса ключи K1 и K3 закрываются, а K2 открывается. Далее полученные проверочные символы  $r$  тактовыми импульсами поступают на выход кодера.

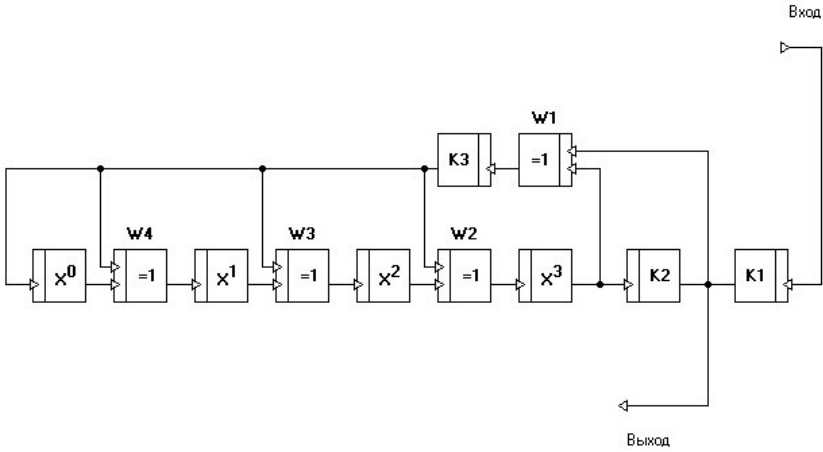


Рисунок 5

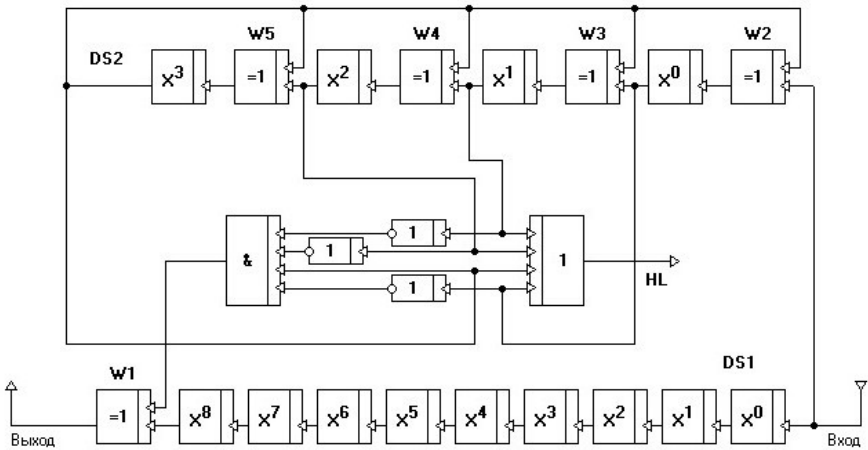


Рисунок 6

Принцип действия кодера (см. рисунок 1) на примере информационной кодовой комбинации  $A(x) = x^4 + x^2 + 1 = 10101$ , показан в таблице 1. Закодированная кодовая комбинация запишется следующим образом:  $F(x) = x^8 + x^6 + x^4 + x^3 + x = 101011010$ .

Признаком, указывающим на наличие ошибки в принятой комбинации, служит остаток, получаемый в результате деления принятого многочлена  $F(x)$  на образующий многочлен  $q(x)$ .

Таблица 1

Такты	Информация		Состояние ячеек памяти (триггеров) кодера			
	На входе кодера	На выходе кодера	$x^3$	$x^2$	$x^1$	$x^0$
1	1	1	0	0	1	1
2	0	0	0	1	1	0
3	1	1	1	1	1	1
4	0	0	1	1	0	1
5	1	1	1	0	1	0
6	–	1	0	1	0	–
7	–	0	1	0	–	–
8	–	1	0	–	–	–
9	–	0	–	–	–	–

*Примечание* – Прочерк (–) означает, что ячейка памяти информации не несет.

Так как  $F'(x) = F(x) \oplus E(x)$ , то  $F'(x)/q(x) = [F(x) \oplus E(x)]/q(x) = F(x)/q(x) \oplus \oplus E(x)/q(x)$ . Так как  $F(x)$  делится на  $q(x)$  без остатка, то неделимость  $F'(x)$  на  $q(x)$  определяется тем, что  $E(x)$  не делится на  $q(x)$ . Таким образом, деление принятой комбинации с ошибкой  $F'(x)$  на образующий многочлен  $q(x)$  можно заменить делением многочлена ошибки  $E(x)$  на этот многочлен.

На рисунке 41 приведена структурная схема декодирующего устройства с порождающим многочленом вида  $q_1(x) = x^4 + x + 1$  с обнаружением и исправлением ошибок. В состав декодера входят: DS1 – буферный регистр, служащий для хранения принимаемой информации до ее исправления; DS2 – декодирующий регистр, служащий для обнаружения ошибки; DS – дешифратор синдрома, служащий для опознавания синдрома; HL – сигнализация ошибки.

Принимаемая кодовая комбинация с ошибкой декодируется с целью определения ошибки в комбинации. Если в полученном от деления на образующий многочлен  $q(x)$  остатке имеется хотя бы одна единица, т. е. он является ненулевым, то будет зафиксирована ошибка.

Дешифратор синдрома строится с таким расчетом, чтобы момент появления “1” на его выходе совпал с моментом прохождения ошибочного символа через сумматор  $w_1$  включенный на выходе буферного регистра, в результате чего произойдет его исправление. Последовательность работы декодера при исправлении ошибки в четвертом символе принятой комбинации показана в таблице 2.

На 9-м тактовом импульсе в элементах памяти декодирующего регистра завершается формирование комбинации синдрома. Комбинация синдрома (жирные цифры) отлична от нуля, что свидетельствует о наличии ошибки.

Далее в буферный и декодирующий регистры подаются еще  $k$  ( $k = 5$ ) тактовых импульсов, которые, во-первых, считывают информационные

символы с буферного регистра через сумматор  $w_1$  в приемник, а во-вторых, переформируют информацию в ячейках декодирующего регистра.

Таблица 2

Такты	Информация на входе декодера	Состояние ячеек памяти декодирующего регистра				Информация на выходе дешифтрора синдрома	Информация на выходе декодера
		$x^3$	$x^2$	$x^1$	$x^0$		
1	1	0	0	0	1	0	
2	0	0	0	1	0	0	
3	1	0	1	0	1	1	
4	1	1	0	1	1	0	
5	1	0	1	0	0	0	
6	1	1	0	0	1	0	
7	0	0	0	0	1	0	
8	1	0	0	1	1	0	
9	0	0	1	1	0	0	
10	–	1	1	0	0	0	1
11	–	1	0	1	1	0	0
12	–	0	1	0	1	1	1
13	–	1	0	1	0	0	0
14	–	0	1	1	1	0	1

Как видно из таблицы 2, на 12-м тактовом импульсе в ячейках декодирующего регистра сформирована комбинация 0101, она поступает на вход дешифратора синдрома, на выходе которого возникает сигнал “1”.

В этот же момент времени на сумматор  $w_1$  поступает искаженный 4-й информационный символ, который, проходя через сумматор, меняет знак на противоположный. Исправленная комбинация будет иметь вид: 10101.

### Порядок выполнения работы

1 Перевести в двоичную систему счисления свой порядковый номер в журнале преподавателя и выполнить кодирование разделимым циклическим кодом при степени порождающего полинома  $r = 4$ .

2 Выполнить действия, предложенные в разделе “Общие указания по выполнению лабораторных работ”.

3 При запуске программы на экране ПК откроется окно “Структурная схема кодека циклического кода” (рисунок 7).

Для продолжения работы с программой необходимо нажать кнопку “Полиномы” в правой верхней части окна.

При этом откроется окно с формами для выбора степени и допустимого образующего полинома и выбора степени и ввода информационного полинома (рисунок 8). В форме “Порождающий многочлен” необходимо выбрать степень полинома (3, 4 или 5) и ниже в контейнере “Допустимые об-

разующие полиномы” выбрать один из приведенных полиномов. По умолчанию задается 4-я степень и первый в контейнере порождающий полином. Далее в форме “Информационный полином” необходимо выбрать степень информационного полинома от 2 до 25 (по умолчанию задана 4-я степень) и ввести информационную последовательность (рисунок 9).

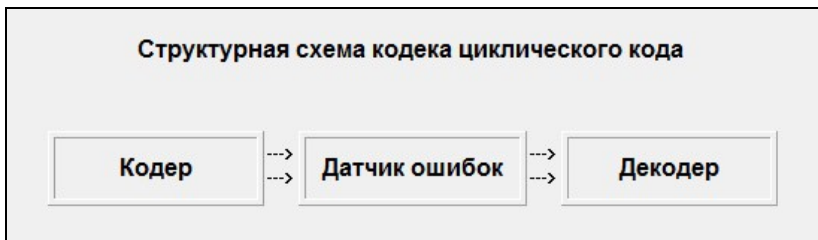


Рисунок 7

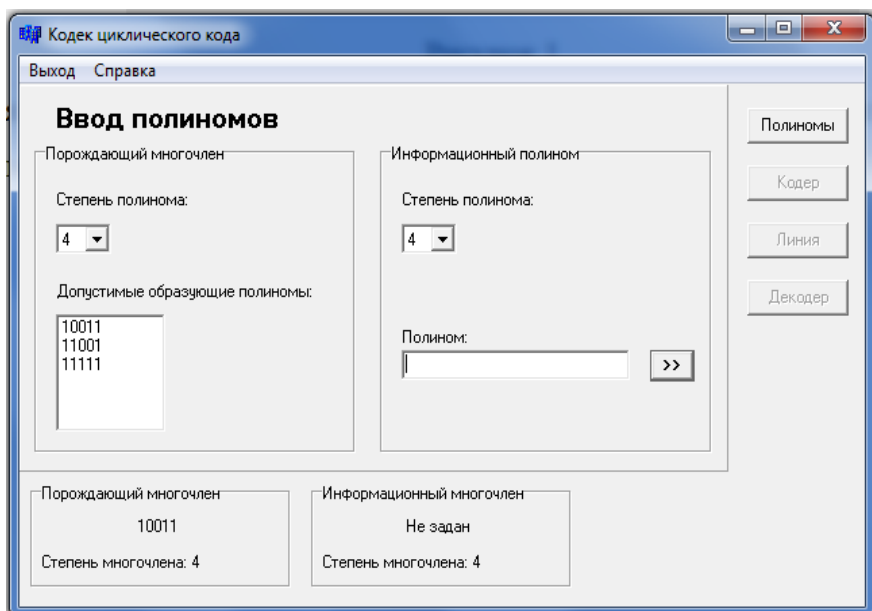


Рисунок 8

4 После ввода кодовой комбинации необходимо нажать кнопку “>>” для ввода информации в память программы. По завершению вышеописанных действий появится возможность перейти к окну “Кодер” (рисунок 10), для этого в правой верхней части окна следует нажать кнопку “Кодер”.



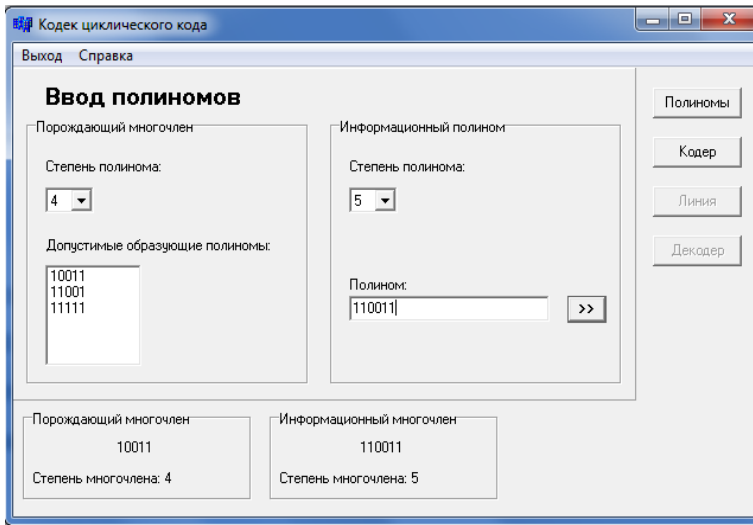


Рисунок 9

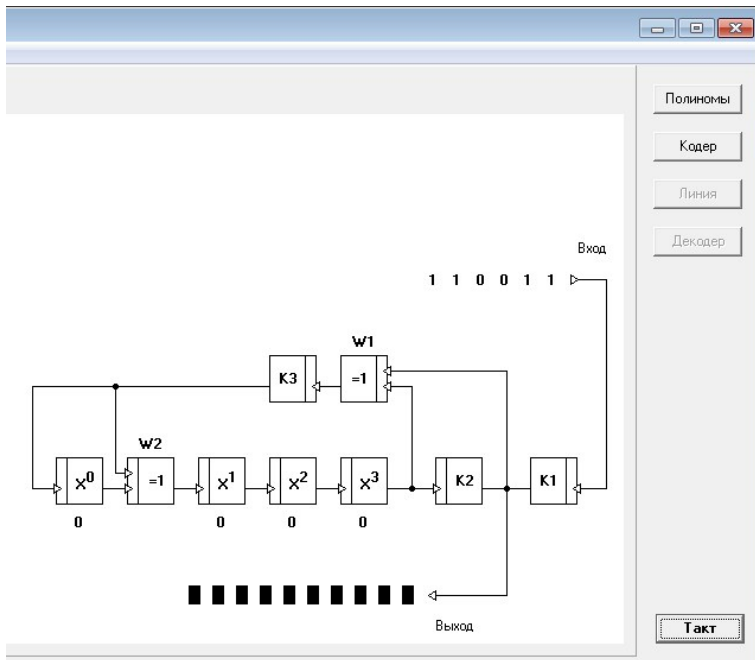


Рисунок 10

На вход кодера подается введенная информационная последовательность, кодирование осуществляется путем последовательного нажатия на кнопку “Такт” в правом нижнем углу окна. Символы закодированной кодовой комбинации поступают на выход кодера и подаются в линию с памятью и датчиком ошибок (рисунок 11).

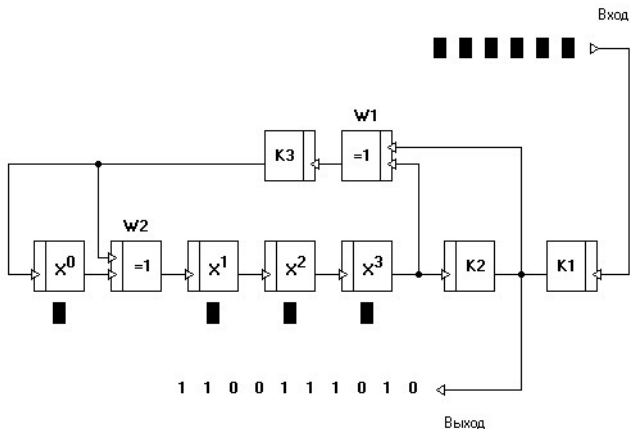


Рисунок 11

В левой нижней части окна находится информация о степени и виде порождающего и информационного многочленов. После вышеописанных действий появляется возможность перехода к окну “Линия”, для этого в правой верхней части окна необходимо нажать кнопку “Линия”.

5 Датчик ошибок имитирует реальную линию связи, при передаче по которой в закодированную кодовую комбинацию может быть внесена ошибка.

Ошибку в кодовую последовательность можно внести нажатием на кнопку под порядковым номером символа. При этом значение символа инвертируется.

Рассмотрим варианты без внесения ошибки (рисунок 12), с внесением однократной ошибки (рисунок 13), двукратной ошибки (рисунок 14).

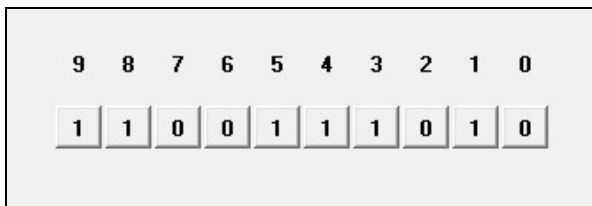


Рисунок 12

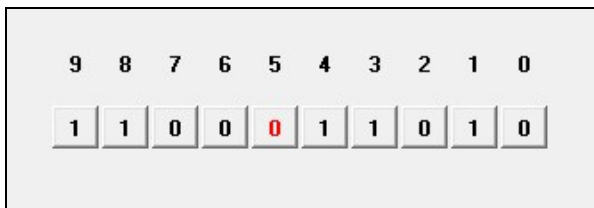


Рисунок 13

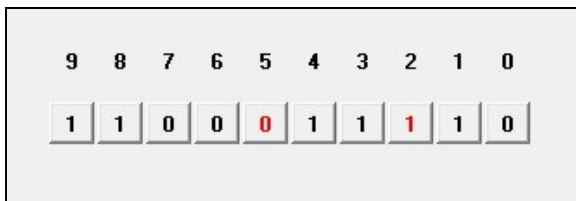


Рисунок 14

Также далее для всех заданных вариантов рассмотрим схемы и результаты декодирования и обнаружения/исправления внесенных ошибок.

6 Для того чтобы перейти к окну “Декодер” (рисунок 15), необходимо нажать кнопку “Декодер” в правой верхней части окна.

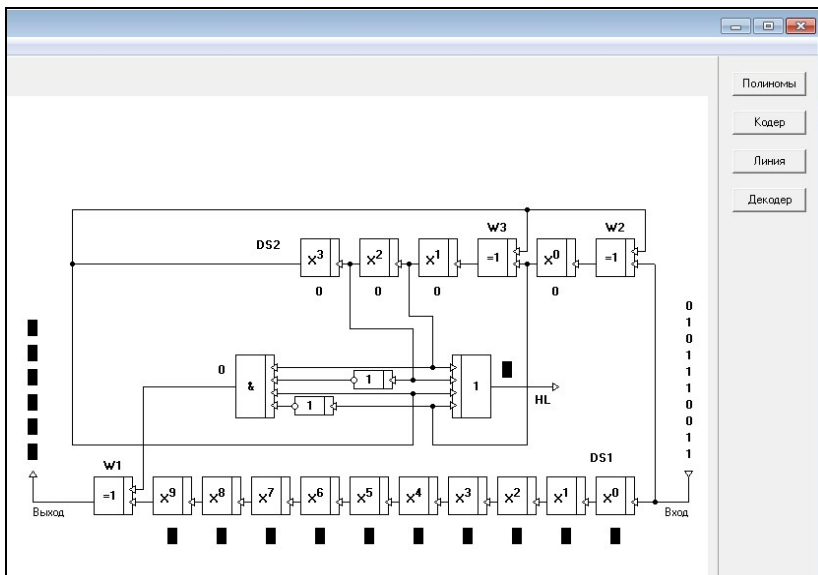


Рисунок 15

Декодирование кодовой комбинации (рисунок 16) происходит путем последовательного нажатия кнопки “Такт”, при этом на выходе декодера появляется правильно декодированная информационная последовательность (вариант без внесения ошибок).

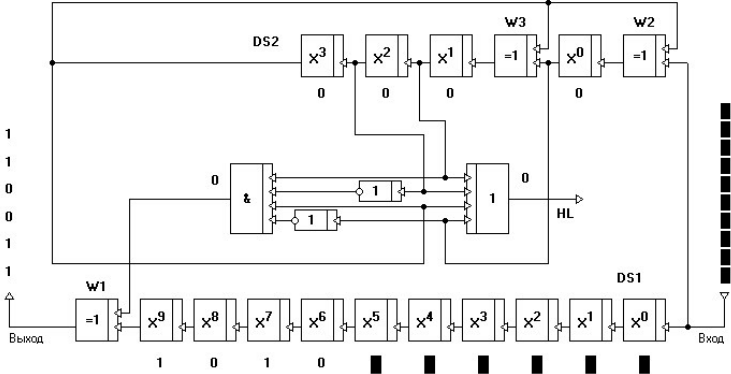


Рисунок 16

При внесении однократной ошибки символ, в который была внесена ошибка, на входе декодера отображается красным цветом. После декодирования на схеме декодера выводится сообщение о том, что в закодированной последовательности была ошибка, которая была успешно исправлена (рисунок 17).

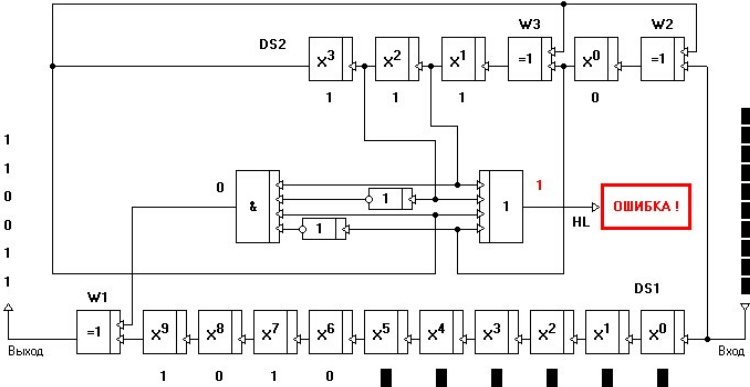


Рисунок 17

При рассмотрении варианта внесения двукратной ошибки символы, в которые была внесена ошибка, отображаются красным цветом. После декодирования на схеме декодера выводится сообщение о том, что была обнаружена ошибка, однако она не была исправлена, неверно декодированный символ на выходе декодера отображен красным цветом (рисунок 18).

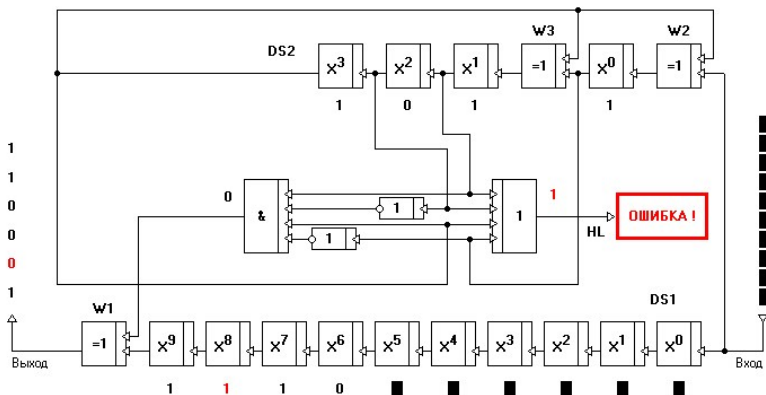


Рисунок 18

С использованием разработанной имитационной модели были проведены исследования корректирующих свойств циклических кодов, результаты исследований показали, что одиночная ошибка обнаруживается и исправляется, а двойная только обнаруживается, что соответствует теоретическим свойствам циклических кодов.

На каждом этапе работы программы можно перейти к любому из этапов, выполненных ранее, повторно выполнить текущий, либо завершить работу с программой. В программе реализована возможность вызова справки при помощи меню в верхней части программы или с помощью нажатия клавиши F1.

Данная программная имитация макета кодера/декодера позволяет в лабораторных условиях исследовать свойства циклического кода.

### Содержание отчета

- 1 Структурная схема кодера или декодера на определенный вид порождающего многочлена (по заданию преподавателя).
- 2 Таблица с результатами исследований корректирующих свойств кода (таблица 3).
- 3 Выводы по результатам исследований.
- 4 Ответы на контрольные вопросы.

Таблица 3

$V$ (кодвая комбинация)	$E_i$ (ошибка)	Результат (ошибка обнаружена, исправлена или нет)

**Контрольные вопросы**

- 1 Определение циклического  $(n,k)$ -кода.
- 2 Как определяется корректирующая способность циклического кода.
- 3 Свойство «разрешенных» комбинаций циклического кода.
- 4 Принцип кодирования циклических кодов.
- 5 Принцип обнаружения и исправления ошибок в циклических кодах.
- 6 Свойства циклических кодов при обнаружении одиночных и двойных ошибок.
- 7 Свойства циклических кодов при обнаружении пакетов ошибок.
- 8 Выбор порождающего многочлена при построении циклических кодов.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 **Кудряшов, В. А.** Передача дискретной информации на железнодорожном транспорте / В. А. Кудряшов, Н. Ф. Семенюта. – М. : Транспорт, 1999. – 328 с.

2 **Нейман, В. Н.** Системы и сети передачи данных на железнодорожном транспорте / В. Н. Нейман. – М. : Маршрут, 2005. – 470 с.

3 Передача дискретных сообщений / под ред. В. П. Шувалова. – М. : Радио и связь, 1990. – 464 с.

4 **Шварцман, В. О.** Теория передачи дискретной информации / В. О. Шварцман, Г. А. Емельянов. – М. : Связь, 1979. – 424 с.

5 **Шляпобергский, В. И.** Основы техники передачи дискретных сообщений / В. И. Шляпобергский – М. : Связь, 1973. – 480 с.

6 Передача дискретной информации и телеграфия / В. С. Гуров [и др.]. – М. : Связь, 1974. – 526 с.

## ОГЛАВЛЕНИЕ

Общие указания по выполнению лабораторных работ .....	3
Лабораторная работа № 1 Исследование простейших кодов .....	4
Лабораторная работа № 2 Исследование линейных кодов .....	17
Лабораторная работа № 3 Исследование кодов Хэмминга .....	30
Лабораторная работа № 4 Исследование циклических кодов .....	40
Список рекомендуемой литературы .....	54



Учебное издание

**Фомичев** Владимир Николаевич  
**Буй** Павел Михайлович

## ПЕРЕДАЧА ДИСКРЕТНЫХ СООБЩЕНИЙ

Лабораторный практикум по дисциплинам  
«Передача дискретных сообщений» и «Передача цифровых сигналов»

Редактор *А. А. Павлюченкова*  
Технический редактор *В. Н. Кучерова*

Подписано в печать 22.05.2014 г. Формат бумаги 60x84<sup>1/16</sup>  
Бумага офсетная. Гарнитура Times. Печать на ризографе.  
Усл. печ. л. 3,25. Уч.-изд. л. 2,83. Тираж 200 экз.  
Зак № . Изд. № 24

Издатель и полиграфическое исполнение  
Белорусский государственный университет транспорта:  
ЛИ № 02330/0552508 от 09.07.2009 г.  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 2/104 от 01.04.2014  
246653, г. Гомель, ул. Кирова, 34