

**Федеральное агентство связи**

**Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования**

**ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ**

**ЭЛЕКТРОННАЯ  
БИБЛИОТЕЧНАЯ СИСТЕМА**

**Самара**

ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ

Кафедра Информатики и вычислительной техники

Методические указания на проведение лабораторной работы  
«Помехоустойчивое кодирование»

по дисциплине «Информатика»  
(специальности 090302, 200700, 210400)



Автор-составитель  
доц., к.т.н. **Алексеев А.П.**  
Под общей редакцией Алексеева А.П.  
Рецензент д.т.н., проф. Карташевский В.Г.

Самара, 2012 г.

## Лабораторная работа Помехоустойчивое кодирование

### 1. Подготовка к работе

По указанной литературе изучить принципы помехоустойчивого кодирования и порядок работы с программой Electronics Workbench (EWB), ответить на контрольные вопросы.

### 2. Контрольные вопросы

- 2.1. Можно ли исправить возникшую ошибку в передаваемых данных, если для помехоустойчивого кодирования используется лишь единственный бит паритета?
- 2.2. Перечислите программы, предназначенные для моделирования радиоэлектронных устройств.
- 2.3. Запишите формулы для формирования контрольных битов с помощью кода Хэмминга.
- 2.4. Для чего используется QR код?
- 2.5. Как рассчитывается слово синдрома в коде Хэмминга?
- 2.6. К чему приведет искажение контрольного бита при использовании кода Хэмминга?
- 2.7. Как составить схему цифрового устройства по заданному логическому выражению?
- 2.8. Нужно ли исправлять контрольные биты?
- 2.9. Почему коды BCH называют избыточными?
- 2.10. Как проверить, правильно ли сформирован код BCH на передающей стороне?
- 2.11. Что такое вес остатка?

### 3. Задания на выполнение лабораторной работы

#### 3.1.Задание 1. Сформировать бит четности (бит паритета) для заданного байта передаваемых данных

Заданные значения информационных битов и контрольного бита следует записать в отчет.

Таблица 3.1.1

Вариант	Байт	Бит паритета
1	10101011	
2	10101100	
3	10101101	
4	10101110	
5	10101111	
6	10110001	
7	10110010	
8	10110011	
9	10110100	
10	10110101	
11	10110110	
12	10110111	
13	10111000	
14	10111001	
15	10111010	
16	10111011	
17	10111100	

#### 3.2. Задание 2. Исследовать помехоустойчивый код с формированием бита четности

Проведите моделирование процесса передачи информации (одной тетрады - половины байта). Исходные данные приведены в таблицах 3.2.1 и 3.2.2.

Таблица 3.2.1

Вариант	Тетрада
1	0000
2	0001
3	0010
4	0011
5	0100
6	0101
7	0110
8	0111
9	1000
10	1001
11	1010
12	1011
13	1100
14	1101
15	1110
16	1111

Моделирование следует выполнить четырежды при заданных значениях помех (табл.3.2.2). Результаты моделирования в виде таблицы поместите в отчет. Прокомментируйте полученные результаты.

Таблица 3.2.2

Вариант	$S_8S_7S_6S_5$	$S_8S_7S_6S_5$	$S_8S_7S_6S_5$	$S_8S_7S_6S_5$
1	0000	1000	0011	1110
2	0000	0100	0110	0111
3	0000	0010	1100	1011
4	0000	0001	1001	1101
5	0000	1000	0011	1110
6	0000	0100	0110	0111
7	0000	0010	1100	1011
8	0000	0001	1001	1101
9	0000	1000	0011	1110
10	0000	0100	0110	0111
11	0000	0010	1100	1011
12	0000	0001	1001	1101
13	0000	1000	0011	1110
14	0000	0100	0110	0111
15	0000	0010	1100	1011
16	0000	0001	1001	1101

### 3.3. Задание 3. Обнаружить и исправить ошибку в принятых данных с помощью кода Хэмминга

Расчетным путем определите, в каком разряде принятых данных произошло искажение. Исходные данные для разных вариантов приведены в таблице 3.3.1. Процесс вычисления искаженного бита следует подробно описать в отчете.

Таблица 3.3.1

Вар.	$b_8$	$b_7$	$b_6$	$b_5$	$k_8$	$b_4$	$b_3$	$b_2$	$k_4$	$b_1$	$k_2$	$k_1$
1	1	0	0	1	1	0	0	1	0	0	1	1
2	1	0	0	0	0	0	1	0	0	1	1	1
3	1	1	1	0	1	0	1	1	0	0	1	1
4	1	1	0	0	0	1	1	0	0	1	1	0
5	1	1	0	1	0	0	0	1	0	1	0	1
6	1	1	1	0	0	1	0	0	0	1	1	1
7	1	0	0	1	1	0	1	0	0	0	0	0
8	0	0	0	0	1	0	1	1	1	1	0	0
9	1	1	1	0	1	0	0	0	1	0	1	0
10	1	0	1	0	0	0	0	0	0	1	0	0
11	1	0	1	0	0	1	1	1	1	0	0	0
12	1	0	0	0	0	0	1	1	1	0	0	0
13	1	1	0	0	0	1	1	0	1	0	1	0
14	1	1	0	0	1	0	0	1	0	0	0	0
15	1	0	1	0	1	1	0	0	0	0	1	0
16	0	0	1	0	0	1	1	0	1	0	1	1

### 3.4. Задание 4. Выполнить моделирование помехоустойчивой передачи информации с помощью кода Хэмминга

С помощью программы Electronics Workbench собрать схему для моделирования процесса передачи информации с использованием помехоустойчивого кодирования (рис.4.4.1). Провести моделирование процесса передачи данных, приведенных в табл. 3.3.1 (для своего варианта).

### 3.5. Задание 5. Сформировать код БЧХ

Для своего варианта расчетным путем сформировать помехоустойчивый циклический код БЧХ, у которого число информационных разрядов  $k = 7$ , число контрольных разрядов  $r = 8$ , а порождающий полином  $g(x) = x^8 + x^7 + x^6 + x^4 + 1$ .

Таблица 3.5.1

Вариант	Данные	Вариант	Данные
1	10100001	9	10101001
2	10100010	10	10101010
3	10100011	11	10101011
4	10100100	12	10101100
5	10100101	13	10101101
6	10100110	14	10101110
7	10100111	15	10101111
8	10101000	16	11101000

### 3.6. Задание 6. Выполнить декодирование кода БЧХ

Декодировать сообщение, сформированное в предыдущем задании, предварительно выполнив искажения в указанных разрядах принимаемых данных.

Таблица 3.6.1

Вариант	Искаженные разряды	Вариант	Искаженные разряды
1	14,7	9	12,7
2	14,8	10	12,8
3	14,9	11	12,9
4	14,10	12	12,10
5	14,11	13	12,11
6	14,12	14	12,13
7	14,13	15	11,10
8	13,10	16	11,9

## 4. Методические указания

### 4.1. Методические указания к заданию 3.1

При работе устройств вычислительной техники и телекоммуникационной аппаратуры возможно появление ошибок в обрабатываемых цифровых данных. Причинами сбоев могут быть мощные электромагнитные помехи, резкое изменение напряжения питания, старение радиоэлементов, ненадежный контакт разъемов, радиоактивное излучение естественных и искусственных источников и т.п. Сбои проявляются в виде случайного изменения одного или нескольких битов машинного слова (вместо единицы в отдельных разрядах передается ноль или наоборот).

Автоматическое обнаружение и исправление ошибок сопровождается введением избыточности в передаваемые или хранимые данные. Для этих целей разработаны специальные коды, в которые помимо информационных битов  $b_1b_2\dots b_n$  дополнительно вводят контрольные (проверочные) биты  $k_1k_2\dots k_m$ . Контрольные биты позволяют проверять целостность (неискаженность) информационных битов машинного слова, а наиболее сложные коды могут не только обнаружить, но и исправить неверно принятые биты.

Разработанные помехоустойчивые коды позволяют решать разные задачи: обнаружить одиночную ошибку, обнаружить и исправить единственную ошибку, обнаружить и исправить несколько ошибок. Первые коды называются обнаруживающими, а вторые – корректирующими кодами.

Простейший код, предназначенный для обнаружения одной ошибки (точнее – для обнаружения нечетного числа ошибок), основан на добавлении к информационным битам одного контрольного бита. При этом контрольный бит должен быть таким, чтобы суммарное число единиц в образованном машинном слове было четным. Добавляемый бит называется **битом паритета**.

Проверочный бит  $k$  для  $n$ -битного слова  $b_n\dots b_2b_1$  вычисляется по формуле:

$$k = \begin{cases} 1, & \text{если } b_1 \oplus b_2 \oplus \dots \oplus b_n = 1 \\ 0, & \text{если } b_1 \oplus b_2 \oplus \dots \oplus b_n = 0 \end{cases}$$

В результате такого преобразования формируется  $(n+1)$  – битное слово  $b_1b_2\dots b_nk$ , число единиц в котором будет четное.

В задании 3.1. требуется для своего варианта определить необходимый бит паритета, который нужно добавить к информационным битам.

Например, для варианта 17 (табл.1) значение байта равно 10111100. Число информационных единиц в этом байте нечетное, поэтому бит паритета нужно установить равным единице. В результате этого получается машинное слово 101111001.



## 4.2. Методические указания к заданию 3.2

Схема исследований показана на рис. 4.2.1.

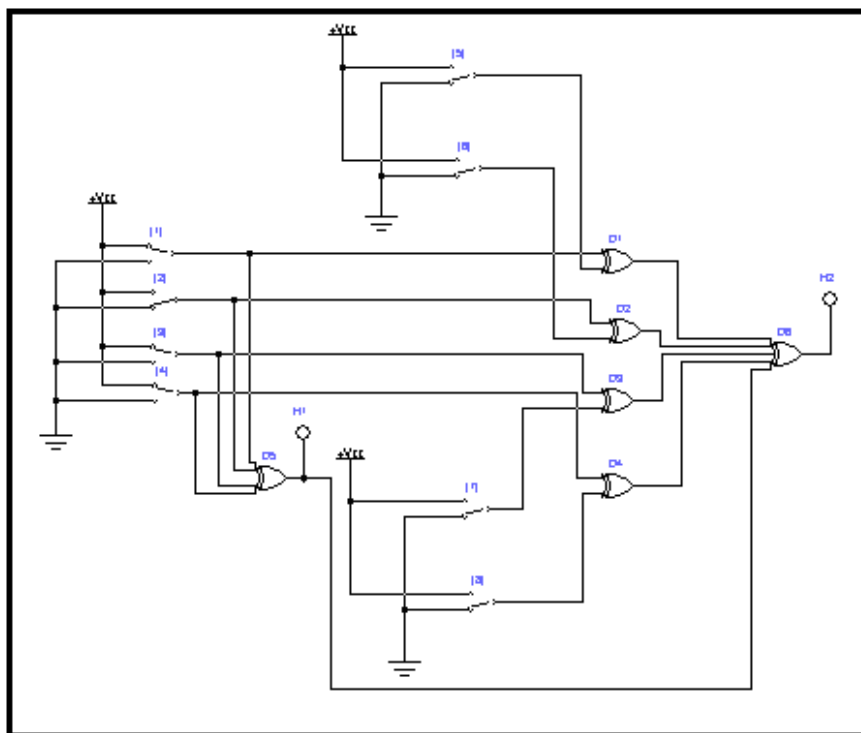


Рис. 4.2.1. Схема исследований кода с формированием бита паритета

Переключатели  $S1...S4$  имитируют передаваемую тетраду (четыре информационных бита). Переключатели  $S5...S8$  симулируют помехи, возникающие при передаче или хранении цифровых данных. Нижнее положение переключателей  $S5...S8$  соответствует отсутствию искажений в соответствующем разряде. Схема  $D5$  формирует бит четности (если на вход подается четное число единиц, то выходной сигнал равен нулю). При этом светодиод  $H1$  не горит.

Логическая схема  $D6$  на приеме осуществляет контроль искажений. Если светодиод  $H2$  не горит, то считается, что искажений нет. Фактически это не так. Появление четного числа искажений в машинном слове не фиксируется (не обнаруживается) данным устройством. Это является существенным недостатком данного метода контроля искажений. Такой код способен лишь сигнализировать о наличии редко появляющихся одиночных сбоев (точнее, нечетного числа сбоев).

## 4.3. Методические указания к заданию 3.3

Рассмотрим пример нахождения неправильно принятого бита с помощью кода Хэмминга.

Места расположения информационных битов (ИБ) и контрольных битов (КБ) в передаваемых данных при использовании кода Хэмминга приведена в следующей таблице. В верхней строке таблицы указан порядковый номер каждого бита в машинном слове.

Таблица 4.3.1.

№ раз.	12	11	10	9	8	7	6	5	4	3	2	1
ИБ	$b_8$	$b_7$	$b_6$	$b_5$		$b_4$	$b_3$	$b_2$		$b_1$		
КБ					$k_8$				$k_4$		$k_2$	$k_1$

Форма записи машинного слова, приведенная в предыдущей таблице, выбрана такой с целью повышения наглядности (из методических соображений). Фактически данные представляют одним двенадцатибитным машинным словом:  $b_8b_7b_6b_5k_8b_4b_3b_2k_4b_1k_2k_1$ .

*Пример.*

Предположим, что в процессе передачи некоторых данных произошло искажение одного информационного бита и на приеме получены указанные в таблице 4.3.2 данные. Требуется найти и исправить искаженный информационный бит.

Таблица 4.3.2.

Разряд	12	11	10	9	8	7	6	5	4	3	2	1
Слово	$b_8$	$b_7$	$b_6$	$b_5$	$k_8$	$b_4$	$b_3$	$b_2$	$k_4$	$b_1$	$k_2$	$k_1$
ИБ	1	0	0	0		1	1	0		1		
КБ					0				1		0	0

*Решение.*

Вычислим значения контрольных битов на приеме. Будем обозначать проверочные биты на приеме со штрихом (чтобы отличить их от контрольных битов, сформированных на передающей стороне). Расчет производится по формулам [1]:

$$\begin{aligned} k_1^{\wedge} &= b_1 \oplus b_2 \oplus b_4 \oplus b_5 \oplus b_7; \\ k_2^{\wedge} &= b_1 \oplus b_3 \oplus b_4 \oplus b_6 \oplus b_7; \end{aligned} \quad (4.3.1)$$

$$k_4^{\wedge} = b_2 \oplus b_3 \oplus b_4 \oplus b_8;$$

$$k_8^{\wedge} = b_5 \oplus b_6 \oplus b_7 \oplus b_8.$$

Используя формулу (4.3.1) и верхнюю строчку таблицы 4.3.2, получим конкретные значения контрольных битов для рассматриваемого примера на приеме:

$$\begin{aligned} k_1^{\wedge} &= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0; \\ k_2^{\wedge} &= 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1; \end{aligned} \quad (4.3.2)$$

$$k_4^{\wedge} = 0 \oplus 1 \oplus 1 \oplus 1 = 1;$$

$$k_8^{\wedge} = 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$

Результаты расчетов (4.3.2) показывают, что контрольные биты, сформированные на передающей и приемной сторонах, различаются:

$$k_1^{\wedge} = k_1; \quad k_2^{\wedge} \neq k_2; \quad k_4^{\wedge} = k_4; \quad k_8^{\wedge} \neq k_8.$$

Различие контрольных битов, сформированных на передающей и приемной сторонах, говорит о том, что в процессе передачи произошло искажение информационного бита. Теперь необходимо определить, какой именно бит был принят неверно.

Для определения и исправления неверно принятого бита требуется вычислить так называемый синдром  $S = s_8 s_4 s_2 s_1$ , где

$$s_1 = k_1 \oplus k_1; \quad s_2 = k_2 \oplus k_2; \quad s_4 = k_4 \oplus k_4; \quad s_8 = k_8 \oplus k_8. \quad (4.3.3)$$

Используя результаты (4.3.2) и нижнюю строчку таблицы 4.3.2, вычислим для рассматриваемого примера четыре бита синдрома:

$$s_1 = 0 \oplus 0 = 0; \quad s_2 = 1 \oplus 0 = 1; \quad s_4 = 1 \oplus 1 = 0; \quad s_8 = 1 \oplus 0 = 1.$$

Синдром  $S = 1010_2$  из двоичной системы счисления (СС) переведем в десятичную СС  $s = 10_{10}$ . Десятичное число 10 говорит о том, что десятый разряд принятых данных ( $b_6$ ) искажен, и этот бит нужно исправить (проинвертировать). Таким образом, после корректировки принятые данные будут иметь вид, показанный в таблице 4.3.3. Напомним, что счет разрядов ведется справа налево.

Таблица 4.3.3.

Разряд	12	11	<b>10</b>	9	8	7	6	5	4	3	2	1
Слово	$b_8$	$b_7$	$b_6$	$b_5$	$k_8$	$b_4$	$b_3$	$b_2$	$k_4$	$b_1$	$k_2$	$k_1$
ИБ	1	0	<b>1</b>	0		1	1	0		1		
КБ					0				1		0	0

#### 4.4. Методические указания к заданию 3.4

Структурная схема устройства для моделирования передачи данных с использованием кода Хэмминга показана на рис. 4.4.1.

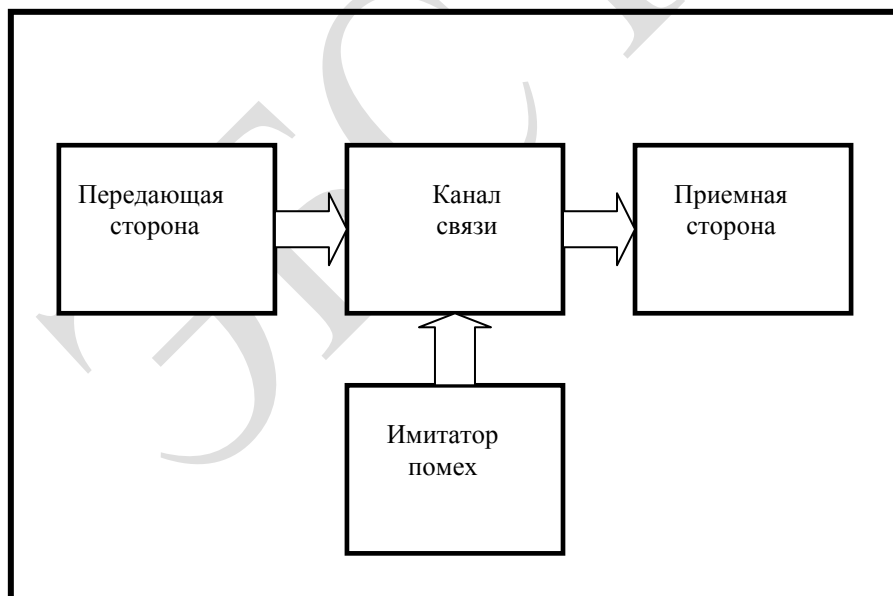


Рис. 4.4.1. Структурная схема устройства моделирования

На передающей стороне формируются контрольные биты в соответствии с выражением (4.3.1). На приемной стороне вычисляется синдром в соответствии с вы-

ражениями (4.3.1) и (4.3.3). Имитатор помех позволяет исказить любой бит данных, передаваемых по каналу связи.

Рассмотрим поочередно конструкцию каждого блока, указанного на структурной схеме.

На рис. 4.4.2 показана схема передатчика.

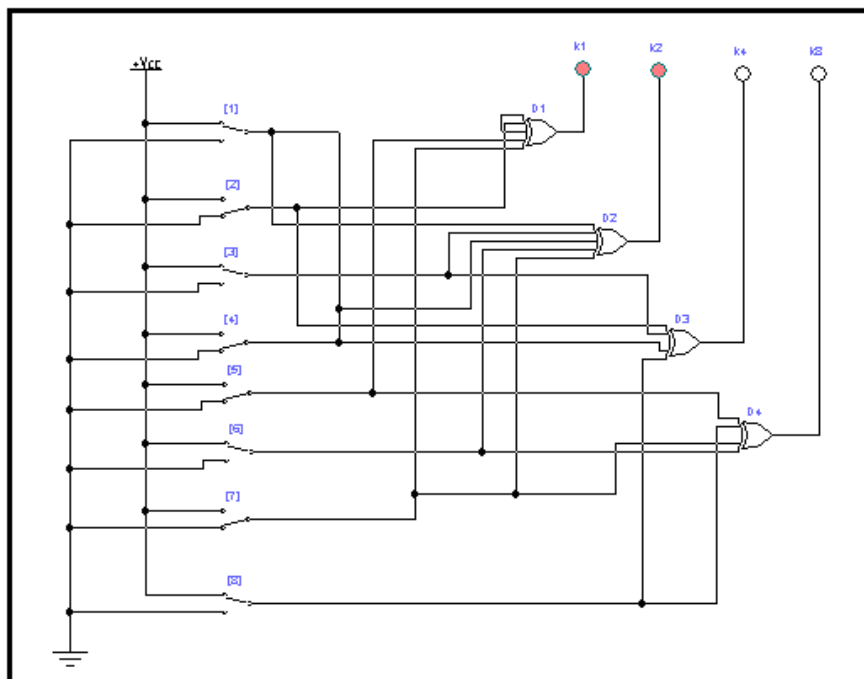


Рис. 4.4.2. Передающая сторона (Передатчик)

Переключатели 1...8 имитируют информационные биты передаваемых данных. С помощью четырех логических элементов Исключающее ИЛИ D1...D4 формируются контрольные биты  $k_1, k_2, k_4, k_8$ . В канал связи передается 12 бит (восемь информационных и четыре контрольных бита). Схематично передаваемые по каналу связи данные показаны на рис. 4.4.3. В устройствах телекоммуникации информация передается не с помощью двенадцатипроводной линии (параллельный код), а преобразуется в последовательный код, например, с помощью регистра сдвига.

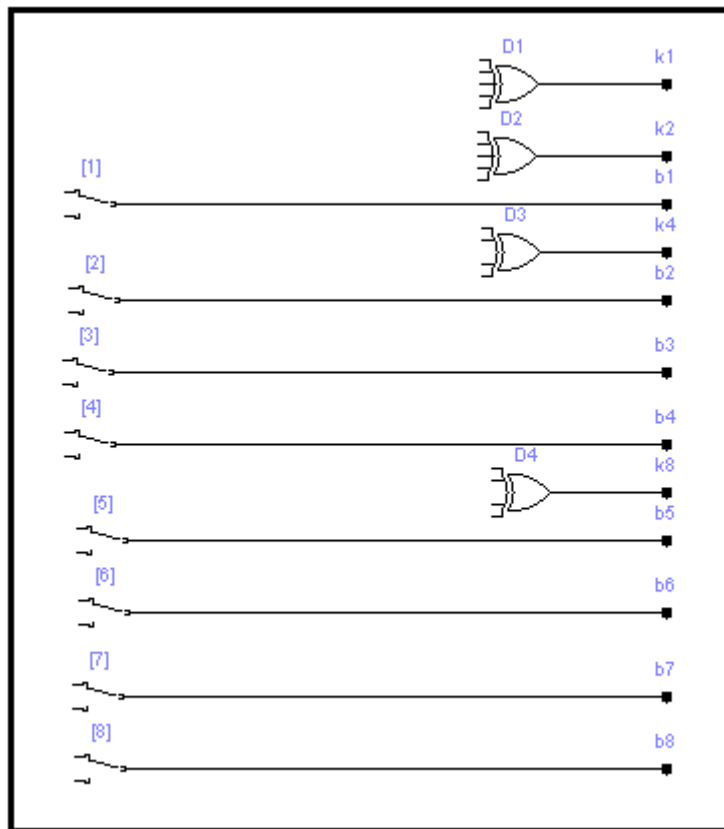


Рис. 4.4.3. Схематичное изображение разрядов передаваемых данных

Имитатор помех позволяет исследователю моделировать возникновение сбоев в любом разряде передаваемых данных. Конструкция имитатора помех сходна с конструкцией блока формирования помех, показанного на рис. 4.2.1. В имитаторе помех содержится 12 логических схем Иключающее ИЛИ и 12 переключателей, с помощью которых можно изменить любой бит. Фрагмент этого блока показан на рисунке 4.4.3. Переключатели Q, W, R могут изменять значения битов (соответственно  $k_1, k_2, b_8$ ). Нижнее положение переключателей не изменяет значения передаваемых битов, а верхнее положение приводит к инверсии соответствующего бита. Указанное положение переключателей Q, W и R приведет к тому, что биты  $k_1, k_2$  будут переданы без искажений ( $k_{p1} = k_1, k_{p2} = k_2$ ), а бит  $k_8$  будет проинвертирован ( $k_{p8} = \bar{k}_8$ ).

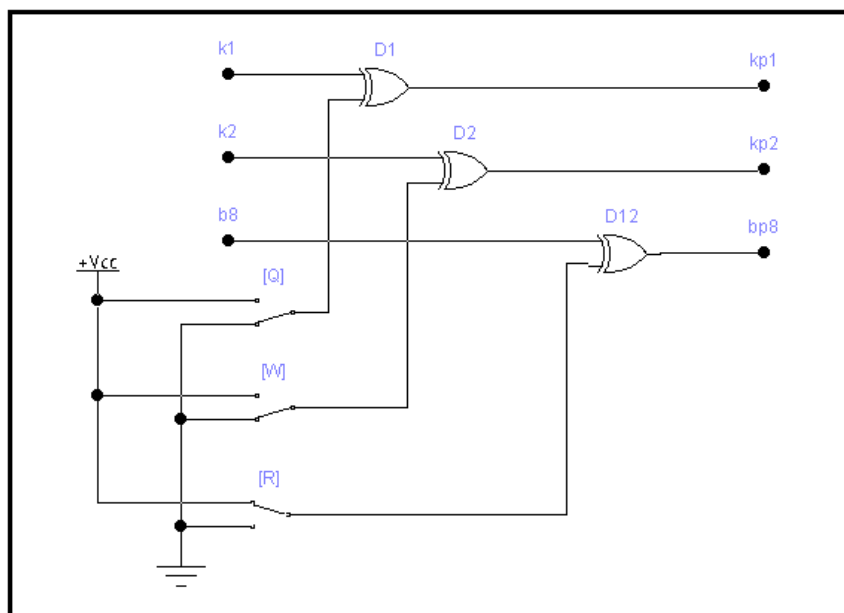


Рис. 4.4.4. Фрагмент имитатора помех

Рассмотренная конструкция имитатора помех не единственная. Этот блок можно построить иначе, например, с помощью генератора слов (рис. 4.4.5). Если генератор слов (Word Generator) формирует во всех разрядах логические нули, то искажений не происходит. Естественно, что логическая единица в каком-либо разряде управляющего слова приводит к искажению соответствующего бита (инверсии).

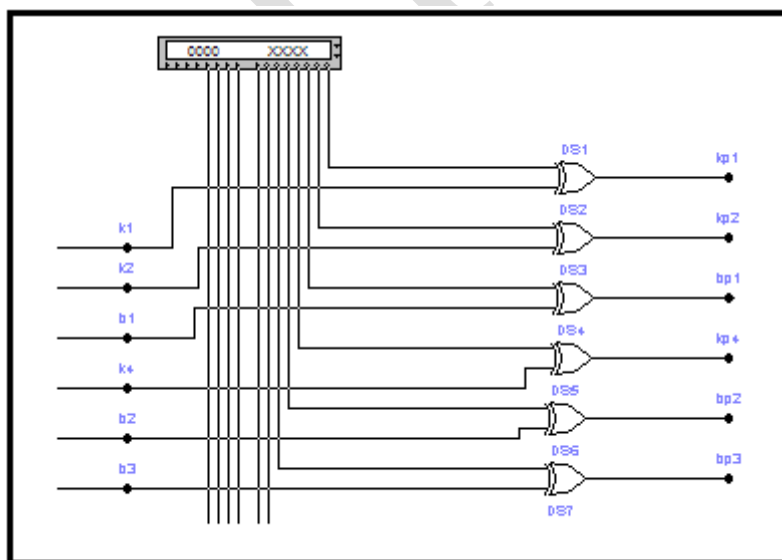


Рис. 4.4.5. Реализация имитатора помех с помощью генератора слов

На приемной стороне поступившие данные должны быть обработаны так, чтобы можно было автоматически исправить искаженный бит. На рис. 4.4.6 показана принципиальная схема приемника. Четыре логических элемента DP1...DP4 производят расчет контрольных битов на приеме. Следующие четыре логические схемы Исключающее ИЛИ DSS1...DSS4 вычисляют слово синдрома. Значение синдрома в

шестнадцатеричной СС отображается на индикаторе Н1. Если индикатор показывает нуль, то это означает, что передача данных произошла без искажений. При наличии сбоев в канале связи показания индикатора изменяются от 1 до СН. Данная конструкция приемника только указывает номер искаженного разряда, но коррекции принятых данных не производит.

Полная принципиальная схема устройства моделирования показана на рис. 4.4.7. Восемь светодиодов позволяют анализировать работу устройства в местах формирования контрольных битов.

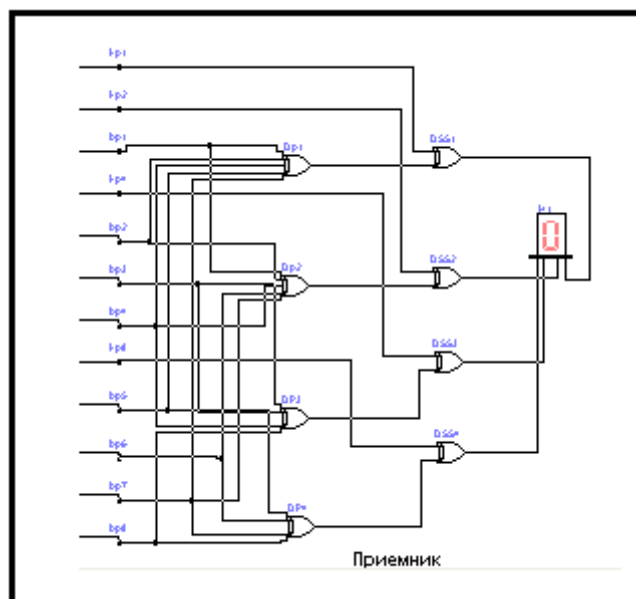


Рис. 4.4.6. Приемная сторона (Приемник)

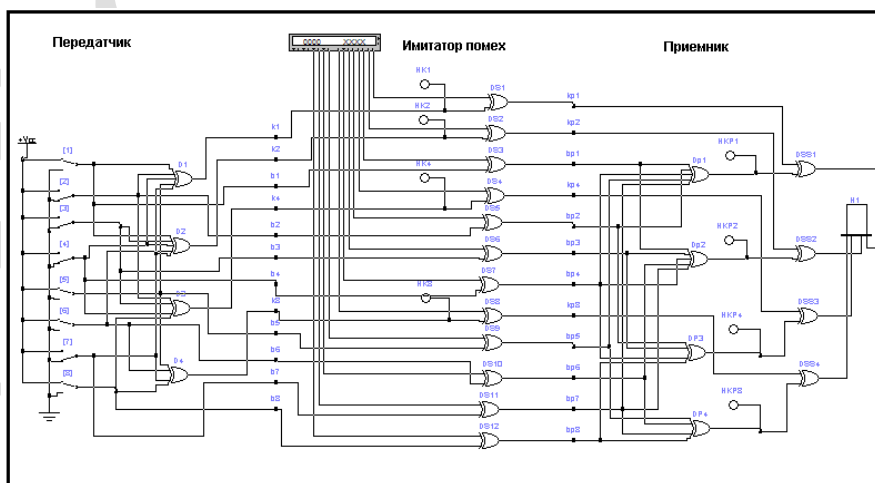


Рис. 4.4.7. Принципиальная схема устройства моделирования

#### 4.5. Методические указания к заданию 3.5

Устранить две и более ошибки в принятых данных позволяют циклические коды Боуза-Чоудхури-Хоквингема (БЧХ).

Циклический код БЧХ  $v(x)$  на передающей стороне формируется следующим образом [2]:

$$v(x) = x^{n-k}u(x) + (x^{n-k}u(x) \bmod g(x)), \quad (4.5.1)$$

где  $x$  – фиктивная переменная;  $u(x)$  – кодируемая последовательность данных (информационные биты);  $n$  – число бит в передаваемых данных (суммарное число информационных и контрольных бит);  $k$  – число информационных бит в машинном слове;  $\bmod$  – операция вычисления остатка от деления;  $+$  – операция соединения (конкатенации) информационных и контрольных битов;  $g(x)$  – порождающий полином.

В выражении (4.5.1) первое слагаемое описывает информационные биты, сдвинутые влево на  $x^{n-k}$  разрядов. Второе слагаемое описывает контрольные биты.

В данной лабораторной работе будет рассматриваться только одна разновидность кодов БЧХ длиной  $n = 15$ . Данный код формируется с помощью порождающего полинома восьмой степени ( $r = 8$ ):

$$g(x) = x^8 + x^7 + x^6 + x^4 + 1.$$

Число информационных разрядов в таком коде  $k = n - r = 15 - 8 = 7$ . Код позволяет исправить две ошибки ( $s = 2$ ).

Рассмотрим порядок построения циклического кода БЧХ на передающей стороне.

*Пример.*

Дано 7 информационных бит 1011011. Запишем исходные данные в виде полинома:

$$u(x) = x^6 + x^4 + x^3 + x + 1. \quad (4.5.2)$$

Порядок построения полинома (4.5.2) иллюстрирует следующая таблица:

Табл. 4.5.1

Номера разрядов	7	6	5	4	3	2	1
Слагаемые	$x^6$	$x^5$	$x^4$	$x^3$	$x^2$	$x$	1

Таблицу нужно трактовать следующим образом: если в соответствующем разряде данных информационный бит равен единице, то в полином нужно включить нижерасположенный член ряда. Этим объясняется отсутствие в полиноме (4.5.2) слагаемых  $x^2$  и  $x^5$ .

Для нахождения первого слагаемого в выражении (4.5.1) нужно полином (4.5.2) умножить на  $x^8$  (заметим, что  $n - k = 8$ ):

$$x^8 \cdot (x^6 + x^4 + x^3 + x + 1) = x^{14} + x^{12} + x^{11} + x^9 + x^8. \quad (4.5.3)$$

Смысл предыдущей операции очень прост: информационные разряды за счет умножения на  $x^8$  смещаются влево на восемь позиций (в сторону старших разрядов). Это сделано для того, чтобы разделить информационные и контрольные биты (информационные разряды будут располагаться в передаваемых данных слева, а кон-



трольные биты – справа).

Для нахождения контрольных битов нужно найти остаток от деления полинома (4.5.3) на порождающий полином  $g(x)$ :

$$\begin{aligned}(x^{14} + x^{12} + x^{11} + x^9 + x^8) \bmod (x^8 + x^7 + x^6 + x^4 + 1) = \\ = x^6 + x^5 + x^3 + x^2 + 1.\end{aligned}\quad (4.5.4)$$

Именно контрольные биты (4.5.4) позволяют на приемной стороне определить, есть ли искажения и при необходимости восстановить два неверно принятых бита.

Сформированный в соответствии с (4.5.1) помехоустойчивый код описывается полиномом:

$$v(x) = x^{14} + x^{12} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1. \quad (4.5.5)$$

Чтобы проверить верно ли сформирован полином (4.5.5), достаточно его разделить на порождающий полином  $g(x)$ . Если остаток от деления равен нулю, то код сформирован правильно.

В соответствии с полиномом (4.5.5) в линию нужно передать двоичный код:

$$101101101101101. \quad (4.5.6)$$

Несложно заметить, что первые семь бит сформированного кода полностью совпадают с информационными битами. Следующие восемь бит являются контрольными. Соотношение между числом информационных и контрольных битов наглядно показывает, почему такие коды называют избыточными.

#### 4.6. Методические указания к заданию 3.6

В процесс передачи по каналу связи (или записи в память цифрового устройства) сформированный код БЧХ  $v(x)$  может быть искажен и на прием поступит код  $f(x)$ . Декодирование полученных данных  $f(x)$  происходит в соответствии со следующим алгоритмом:

- 1) выполняется деление принятой последовательности  $f(x)$  на порождающий полином  $g(x)$ ;
- 2) вычисляется вес остатка  $w$  (количество единиц в остатке);
- 3) если  $w > s$ , где  $s$  – допустимое число исправляемых данным кодом ошибок, то производится циклический сдвиг влево на один разряд принятой последовательности  $f(x)$  и вновь выполняется шаг 1;
- 4) если  $w \leq s$ , то производится суммирование образованной последовательности с остатком;
- 5) производится циклический сдвиг полученной на шаге 4 последовательности вправо на количество разрядов, на которые сдвигалась влево в процессе декодирования принятая последовательность.

Рассмотрим процесс декодирования на примере кода, сформированного в предыдущем задании.

*Пример.*

Предположим, что в данных (4.5.6) произошло искажение разрядов 13 и 11. В результате на прием поступило двоичное число 111001101101101.

Запишем его в виде полинома:

$$f(x) = x^{14} + x^{13} + x^{12} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1.$$

Выполним декодирование. Результаты расчетов на каждой итерации поместим в таблицу 4.6.1. Так как данный код позволяет исправлять две ошибки ( $s = 2$ ), расчеты следует вести до тех пор, пока не будет выполнено условие  $w \leq 2$ .

В таблице 4.6.1 использованы такие обозначения:

$C^i(x)$  - частное от деления  $f^i(x)$  на порождающий полином  $g(x)$  на  $i$ -той итерации;  $P^i(x)$  - остаток от деления  $f^i(x)$  на порождающий полином  $g(x)$  на  $i$ -той итерации.

На пятой итерации вес остатка достиг значения, при котором следует прекратить дальнейшие расчеты ( $w = 2$ ). За пять проведенных итераций было выполнено четыре циклических сдвига влево.

Выполненные операции позволяют наглядно понять, почему этот код называют циклическим. При выполнении сдвига влево крайний левый разряд считался соседним с крайним правым разрядом. По этой причине на итерациях 2, 3, 4 последними слагаемыми полиномов  $f^i(x)$  являются единицы. Это объясняется тем, что в тех случаях  $x^{14} = 1$  и единица переходит в младший разряд последовательности. На пятой итерации полином  $f^5(x)$  не содержит слагаемого, равного 1. Это объясняется тем, что на предыдущей итерации  $x^{14} = 0$ .

Таблица 4.6.1

Ит.		Полиномы	Вес $w$
1	$f^1(x)$	$x^{14} + x^{13} + x^{12} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1$	
	$C^1(x)$	$x^6 + x^2$	
	$P^1(x)$	$x^6 + x^5 + x^3 + 1$	4
2	$f^2(x)$	$x^{14} + x^{13} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x + 1$	
	$C^2(x)$	$x^6 + x^4 + x^3 + 1$	
	$P^2(x)$	$x^7 + x^6 + x^4 + x$	4
3	$f^3(x)$	$x^{14} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	
	$C^3(x)$	$x^6 + x^5 + x$	
	$P^3(x)$	$x^6 + x^5 + x^4 + x^2 + 1$	5
4	$f^4(x)$	$x^{12} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + x + 1$	
	$C^4(x)$	$x^4 + x^2 + 1$	
	$P^4(x)$	$x^7 + x^6 + x^5 + x^3 + x$	5
5	$f^5(x)$	$x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x^2 + x$	
	$C^5(x)$	$x^5 + x^3 + x + 1$	
	$P^5(x)$	$x^2 + 1$	2

В соответствии с алгоритмом декодирования теперь следует перейти к шагу 4 и найти сумму величин на последней итерации:

$$\begin{aligned}
 R(x) &= f^5(x) + P^5(x) = \\
 &= x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x^2 + x + x^2 + 1 = \\
 &= x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x^2 + x + 1. \quad (4.6.1)
 \end{aligned}$$

Полученный полином (4,6,1) преобразуем в двоичное число:

$$R(x) = 011011011011111. \quad (4.6.2)$$

В соответствии в шагом 5 алгоритма декодирования последовательность (4.6.2) нужно циклически сдвинуть вправо на четыре разряда. Этапы сдвига показаны в табл. 4.6.2.

Таблица 4.6.2

Номера сдвигов	Числа
$R(x)$	011011011011011
1 сдвиг	101101101101101
2 сдвиг	110110110110110
3 сдвиг	011011011011011
4 сдвиг	101101101101101

Первые семь битов двоичного числа, полученного после четвертого циклического сдвига вправо, являются информационными битами, в которых были исправлены две ошибки.

Таким образом, в результате декодирования получено число 1011011, которое точно совпадает с переданным числом.

### Список литературы

1. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем: Учебник для вузов.- СПб: Питер, 2004. – 668 с.
2. Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование. – М.: ДМК Пресс, 2009. – 848 с.
3. Алексеев А.П. Информатика 2007.– СОЛОН-ПРЕСС, 2007. – 608 с.