

КОДИРОВАНИЕ ИНФОРМАЦИИ



КОДИРОВАНИЕМ НАЗЫВАЕТСЯ ПРОЦЕСС ПРЕОБРАЗОВАНИЯ ОДНОГО НАБОРА ЗНАКОВ В ДРУГОЙ НАБОР ЗНАКОВ. ОБРАТНЫЙ ПРОЦЕСС – **ДЕКОДИРОВАНИЕ**.

ПРОГРАММИРОВАНИЕ – КОДИРОВАНИЕ ИНФОРМАЦИИ НА ЯЗЫКЕ, "ПОНЯТНОМ" КОМПЬЮТЕРУ

ЯЗЫКИ: ФОРМАЛЬНЫЕ И ЕСТЕСТВЕННЫЕ

КОД: РАВНОМЕРНЫЙ И НЕРАВНОМЕРНЫЙ

КОДИРОВАНИЕ

КОД СТРОИТСЯ НА БАЗЕ АЛФАВИТА, СОСТОЯЩЕГО ИЗ БУКВ, ЦИФР И ДРУГИХ СИМВОЛОВ. КОД ХАРАКТЕРИЗУЕТСЯ **ДЛИНОЙ** И **СТРУКТУРОЙ**

ИСПОЛЬЗУЕМЫЙ ДЛЯ КОДИРОВАНИЯ КОНЕЧНЫЙ НАБОР ОТЛИЧНЫХ ДРУГ ОТ ДРУГА ЗНАКОВ, В КОТОРОМ ОПРЕДЕЛЕН ПОРЯДОК, НАЗЫВАЕТСЯ **АЛФАВИТОМ**

КОДИРОВАНИЕ ИНФОРМАЦИИ

СИСТЕМЫ СЧИСЛЕНИЯ

ИЗМЕРЕНИЕ

НЕПОЗИЦИОННЫЕ

ПОЗИЦИОННЫЕ

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

МОЩНОСТЬ АЛФАВИТА - КОЛИЧЕСТВО ВХОДЯЩИХ В НЕГО ЗНАКОВ

ИНФОРМАЦИОННЫЙ ОБЪЕМ ТЕКСТА - КОЛИЧЕСТВО ИНФОРМАЦИИ, ЗАКЛЮЧЕННОЕ В ЭТОМ ТЕКСТЕ

КОЛИЧЕСТВО ИНФОРМАЦИИ - ЭТО ДЛИНА СООБЩЕНИЯ, С ПОМОЩЬЮ КОТОРОГО ЕЕ МОЖНО ЗАКОДИРОВАТЬ

ДЕСЯТИЧНАЯ

ДВОИЧНАЯ

ВОСЬМЕРИЧНАЯ

ШЕСТНАДЦАТЕРИЧНАЯ

ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ В ДРУГУЮ



Равномерные коды

Равномерные коды – все кодовые слова (коды отдельных букв) имеют одинаковую длину.

М	А	Ы	Л	У	пробел
000	001	010	011	100	101

МАМА МЫЛА ЛАМУ:

000 001 000 001 101 000 010 011 001 101 011 001 000 100

 **Равномерные коды позволяют однозначно декодировать сообщения!**

 **сообщения получаются длинными**

Закодируйте свое имя с помощью кодовой таблицы
(*Windows-1251*):

	0	1	2	3	4	5	6	7	8	9	А	В	С	Д	Е	Ф
С	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Д	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

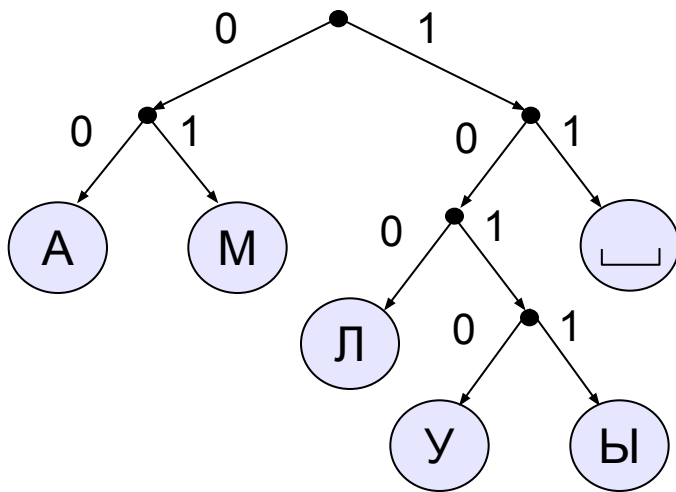
В А С Я
ВАСЯ С2 С0 Д1 ДФ

! Код равномерный, разделитель **НЕ** нужен!

Неравномерные коды

Неравномерные коды – кодовые слова имеют разную длину.

М	А	Ы	Л	У	пробел
01	00	1011	100	1010	11



01|00|01|00|11|01|10|11|10|00|01|11|00|00|00|11|01|0
М | А | М | А | _ | М | Ы | Л | А | _ | Л | А | М | У

Префиксный код – ни одно кодовое слово не совпадает с началом другого кодового слова (*условие Фано*).



Любой префиксный код позволяет однозначно декодировать сообщения!

Закодируйте свое имя с помощью азбуки Морзе.

А	● —	П	● — — ●	Ь	— ● ● —
Б	— ● ● ●	Р	● — ●	Ы	— ● — —
В	● — — —	С	● ● ●	Й	● — — —
Г	— — — ●	Т	—		
Д	— ● ●	У	● ● —	1	● — — — — —
Е	●	Ф	● ● — ●	2	● ● — — — —
Ж	● ● ● —	Х	● ● ● ●	3	● ● ● — — —
З	— — — ● ●	Ц	— ● — — ●	4	● ● ● ● —
И	● ●	Ч	— — — — ●	5	● ● ● ● ●
К	— ● — —	Ш	— — — — —	6	— ● ● ● ●
Л	● — — ● ●	Щ	— — — ● —	7	— — — ● ● ●
М	— — —	Э	● ● — — ● ●	8	— — — — ● ●
Н	— — ●	Ю	● ● — — —	9	— — — — — ●
О	— — — —	Я	● — — ● —	0	— — — — — —

ВАСЯ



Код неравномерный, нужен разделитель!

Двоичное кодирование



- можно закодировать (почти) **все виды** информации;
- нужны только устройства с **двумя состояниями**;
- почти **нет ошибок** при передаче данных;
- **компьютеру легче** обрабатывать данные.



- **человеку сложно** воспринимать двоичные коды.



Декодирование – это восстановление сообщения из последовательности кодов.

М	А	Ы	Л	У	пробел
00	1	01	0	10	11

МАМА МЫЛА ЛАМУ → 00 1 00 1 11 00 01 0 1 11 0 1 00 10

Приняли сообщение:

0010011100010111010010 → ???

ЛЛАЛЛАААЛЛЛАЛАААЛАЛЛАЛ



Не все коды допускают однозначное декодирование!

Помехоустойчивое кодирование - кодирование, предназначенное для передачи данных по каналам с помехами, обеспечивающее исправление возможных ошибок передачи вследствие помех.

Для обнаружения ошибок используют коды обнаружения ошибок, для исправления - помехоустойчивые коды.

Помехоустойчивое кодирование предполагает **введение в передаваемое сообщение**, наряду с информационными, так называемых **проверочных разрядов**, формируемых в устройствах защиты от ошибок (кодерах - на передающем конце, декодерах - на приемном).

Лабораторная работа №5

ПОМЕХОУСТОЙЧИВЫЙ КОД ХЭММИНГА

Цель работы: изучение принципов помехоустойчивого кодирования, получение навыков моделирования помехоустойчивых кодов с помощью Electronics Workbench.

ЗАДАНИЕ 1.

Формирование бита чётности

Простейший код, предназначенный для обнаружения одной ошибки (точнее – для обнаружения нечётного 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 чётным.**

Добавляемый бит называется **битом паритета.**

Проверочный бит k для n -битного двоичного слова $b_1b_2\dots b_n$ вычисляется по формуле:

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

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Пример.

Пусть дан байт **10111100**.

$$1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

тогда $k=1$ и кодовая комбинация равна

101111001

Сформировать бит чётности (бит паритета) для заданного байта передаваемых данных.

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	

ЗАДАНИЕ 2.

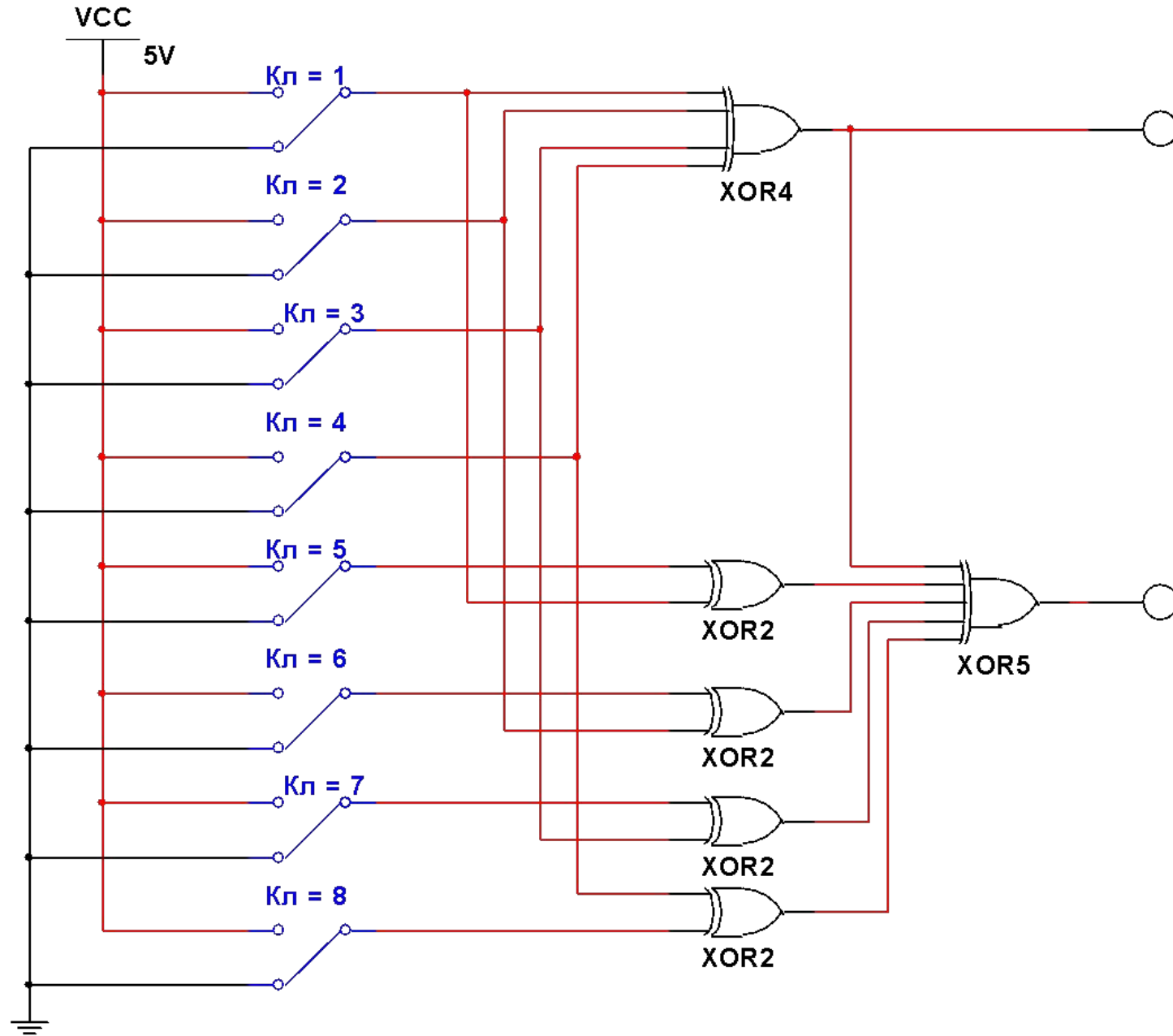
Исследование помехоустойчивого кода с формированием бита чётности

Выполнить моделирование процесса передачи информации.

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

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	1000	1001	1101
13	0000	0001	0011	1110
14	0000	0100	0110	0111
15	0000	0010	1100	1011
16	0000	0001	1001	1101

Схема исследований кода с формированием бита паритета



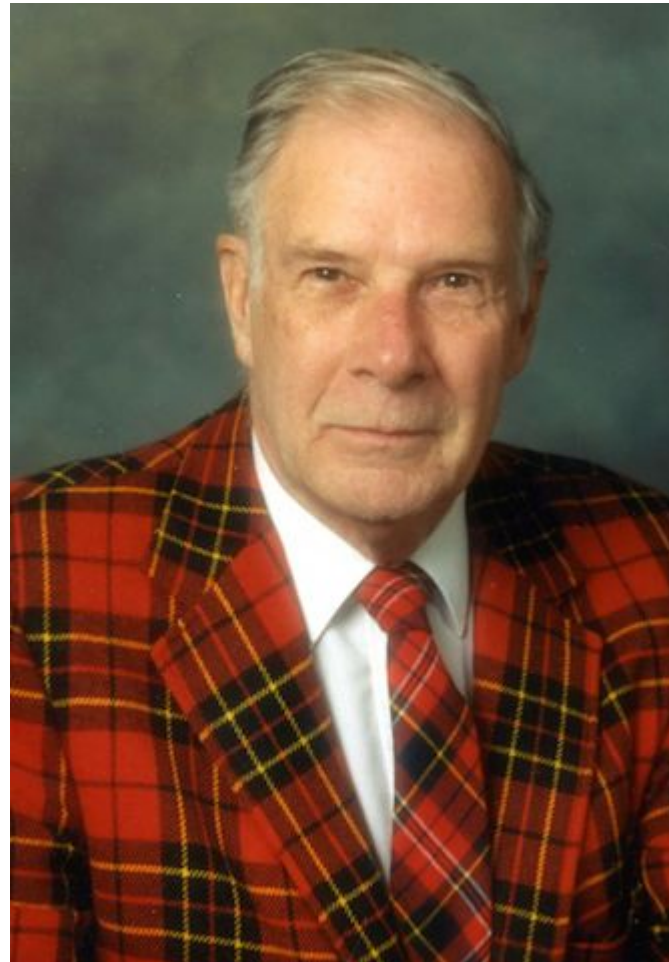
ЗАДАНИЕ 3.

Исправление ошибки с помощью кода Хэмминга

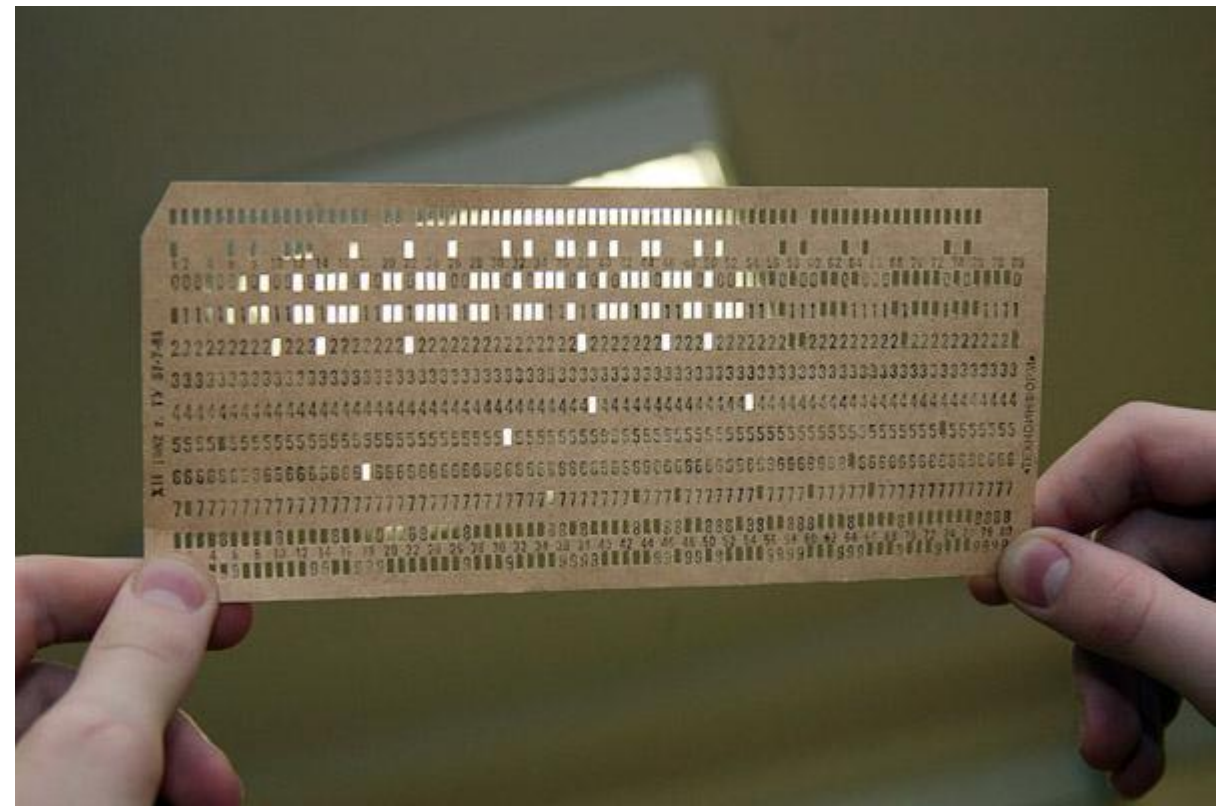
Расчётным путём (вручную) определить, в каком разряде принятого кода Хэмминга произошло искажение. Процесс вычисления искажённого бита следует описать в отчёте.

Вариант	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

Код Хемминга – это блочный код, позволяющий исправлять одиночные и фиксировать двойные ошибки, разработанный Ричардом Хеммингом в 40-х годах прошлого столетия.



В это время он работал в лаборатории Bell Labs на электромеханической счетной машине Bell Model V.



Идея кода Хемминга заключается в разбиении данных на блоки фиксированной длины и вводе в эти блоки контрольных бит, дополняющих до четности несколько пересекающихся групп, охватывающих все биты блока.

Ричард Хемминг рассчитал минимальное количество проверочных бит, позволяющих однозначно исправлять однократные ошибки.

$$k = 2^k - m - 1$$

m - длина информационного блока;

k - количество контрольных бит.

Рассмотрим пример кодирования бинарной последовательности данных, состоящей из 8 элементов: **10101101**.

1. Определим необходимое количество контрольных разрядов.

$$k=2^k - m - 1$$

m - длина информационного блока;

k - количество контрольных бит.

$$k=3$$

$$3=2^3 - 8 - 1, 3 > -1$$

$$k=4$$

$$4=2^4 - 8 - 1, 4 < 7$$

2. Определим расположение проверочных бит в результирующей закодированной последовательности.

Обозначим информационные биты символом ИБ, а контрольные биты символом КБ.

Контрольные биты будут занимать четыре позиции с порядковыми номерами, равными степени двойки: $2^0, 2^1, 2^2, 2^3$
 $\Rightarrow 1, 2, 4, 8$.

№ разряда	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

3. Определим, какие группы контролируют проверочные биты.

ИБ3: $3 = 2^0 + 2^1 = 1 + 2 \Rightarrow$ информационный бит проверяется контрольными битами 1 и 2.

ИБ5: $5 = 2^0 + 2^2 = 1 + 4 \Rightarrow$ Информационный бит проверяется контрольными битами 1 и 4.

ИБ6: $6 = 2^1 + 2^2 = 2 + 4 \Rightarrow$ Информационный бит проверяется контрольными битами 2 и 4.

ИБ7: $7 = 2^0 + 2^1 + 2^2 = 1 + 2 + 4 \Rightarrow$ Информационный бит проверяется контрольными битами 1, 2 и 4.

ИБ9: $9 = 2^0 + 2^3 = 1 + 8 \Rightarrow$ Информационный бит проверяется контрольными битами 1 и 8.

ИБ10: $10 = 2^1 + 2^3 = 2 + 8 \Rightarrow$ Информационный бит проверяется контрольными битами 2 и 8.

ИБ11: $11 = 2^0 + 2^1 + 2^3 = 1 + 2 + 8 \Rightarrow$ Информационный бит проверяется контрольными битами 1, 2 и 8.

ИБ12: $12 = 2^2 + 2^3 = 4 + 8 \Rightarrow$ Информационный бит проверяется контрольными битами 4 и 8.

4. Рассчитаем значения контрольных бит.

Для этого определим группы для всех контрольных бит, а результат запишем в соответствующие контрольные биты:

$$\begin{aligned} \text{КБ1} &= \text{ИБ3} \oplus \text{ИБ5} \oplus \text{ИБ7} \oplus \text{ИБ9} \oplus \text{ИБ11} = \\ &= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0 \end{aligned}$$

$$\begin{aligned} \text{КБ2} &= \text{ИБ3} \oplus \text{ИБ6} \oplus \text{ИБ7} \oplus \text{ИБ10} \oplus \text{ИБ11} = \\ &= 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0 \end{aligned}$$

$$\text{КБ4} = \text{ИБ5} \oplus \text{ИБ6} \oplus \text{ИБ7} \oplus \text{ИБ12} = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$\text{КБ8} = \text{ИБ9} \oplus \text{ИБ10} \oplus \text{ИБ11} \oplus \text{ИБ12} = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

Итоговая кодовая комбинация

№ разряда	12	11	10	9	8	7	6	5	4	3	2	1
ИБ	1	0	1	0		1	1	0		1		
КБ					0				1		0	0
Код	1	0	1	0	0	1	1	0	1	1	0	0

Рассмотрим пример нахождения искажённого бита с помощью кода Хэмминга.

Места расположения информационных битов (ИБ) и контрольных битов (КБ) в передаваемых данных указаны в таблице.

№ разряда	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

Пример.

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

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

Вычислим значения контрольных битов на приёме.

Расчёт производится по формулам:

$$\begin{aligned} \text{КБ}'1 &= \text{ИБ3} \oplus \text{ИБ5} \oplus \text{ИБ7} \oplus \text{ИБ9} \oplus \text{ИБ11} = \\ &= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0 \end{aligned}$$

$$\begin{aligned} \text{КБ}'2 &= \text{ИБ3} \oplus \text{ИБ6} \oplus \text{ИБ7} \oplus \text{ИБ10} \oplus \text{ИБ11} = \\ &= 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1 \end{aligned}$$

$$\text{КБ}'4 = \text{ИБ5} \oplus \text{ИБ6} \oplus \text{ИБ7} \oplus \text{ИБ12} = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$\text{КБ}'8 = \text{ИБ9} \oplus \text{ИБ10} \oplus \text{ИБ11} \oplus \text{ИБ12} = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

Контрольные биты, сформированные на передающей и приёмной сторонах, **различаются**:

$$\text{КБ1} = \text{КБ}'1 = 0 = 0, \quad \text{КБ2} = \text{КБ}'2 = 0 \neq 1,$$

$$\text{КБ4} = \text{КБ}'4 = 1 = 1, \quad \text{КБ8} = \text{КБ}'8 = 0 \neq 1,$$

Для определения неверно принятого бита требуется вычислить так называемый синдром $S=s_8s_4s_2s_1$, где

$$s_1 = \text{ИБ1} \oplus \text{ИБ}'1 = 0 \oplus 0 = 0,$$

$$s_2 = \text{ИБ2} \oplus \text{ИБ}'2 = 0 \oplus 1 = 1,$$

$$s_4 = \text{ИБ4} \oplus \text{ИБ}'4 = 1 \oplus 1 = 0,$$

$$s_8 = \text{ИБ8} \oplus \text{ИБ}'8 = 0 \oplus 1 = 1.$$

Переведём синдром $S=1010$ из двоичной системы счисления в десятичную $S=1*2^3+0*2^2+1*2^1+0*2^0=10$.

Десятичное число 10 говорит о том, что десятый разряд принятых данных (b_6) искажён, и этот бит нужно исправить (проинвертировать).

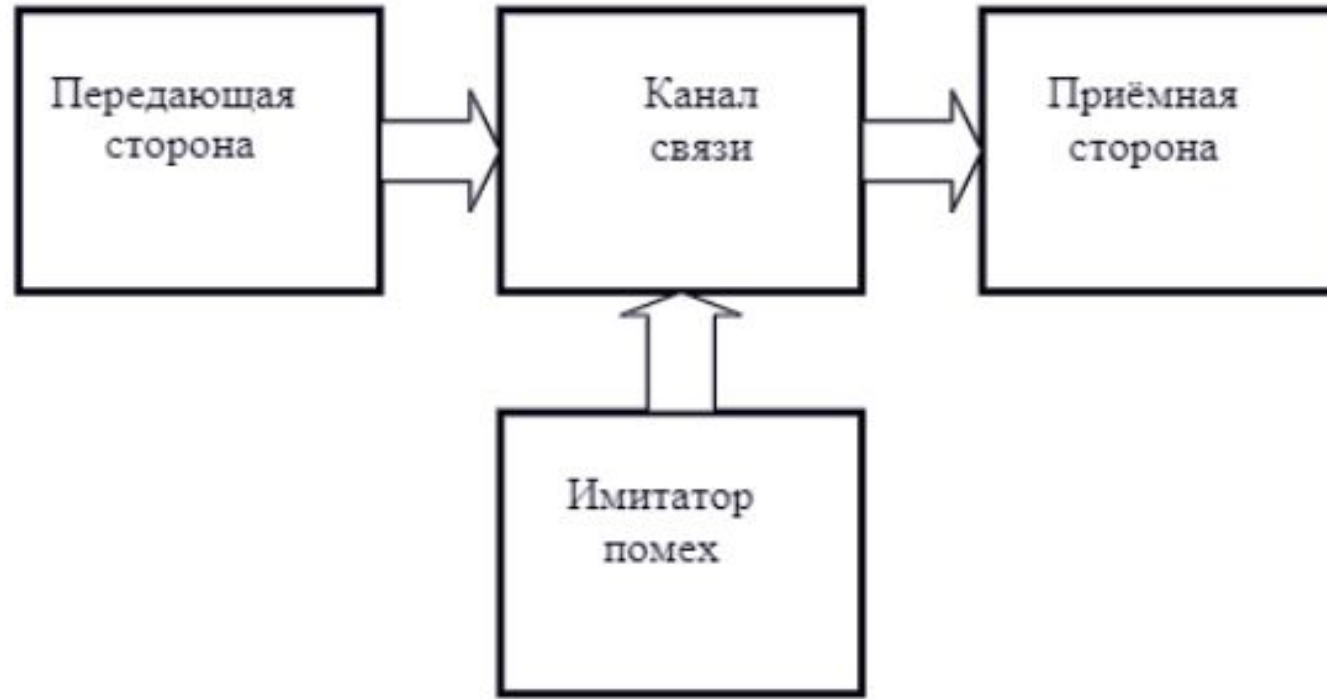
Таким образом, после корректировки принятые данные будут иметь вид, показанный в таблице.

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	1	0		1	1	0		1		
КБ					0				1		0	0

Код	1	0	1	0	0	1	1	0	1	1	0	0
-----	---	---	---	---	---	---	---	---	---	---	---	---

ЗАДАНИЕ 4.

Моделирование работы кода Хэмминга

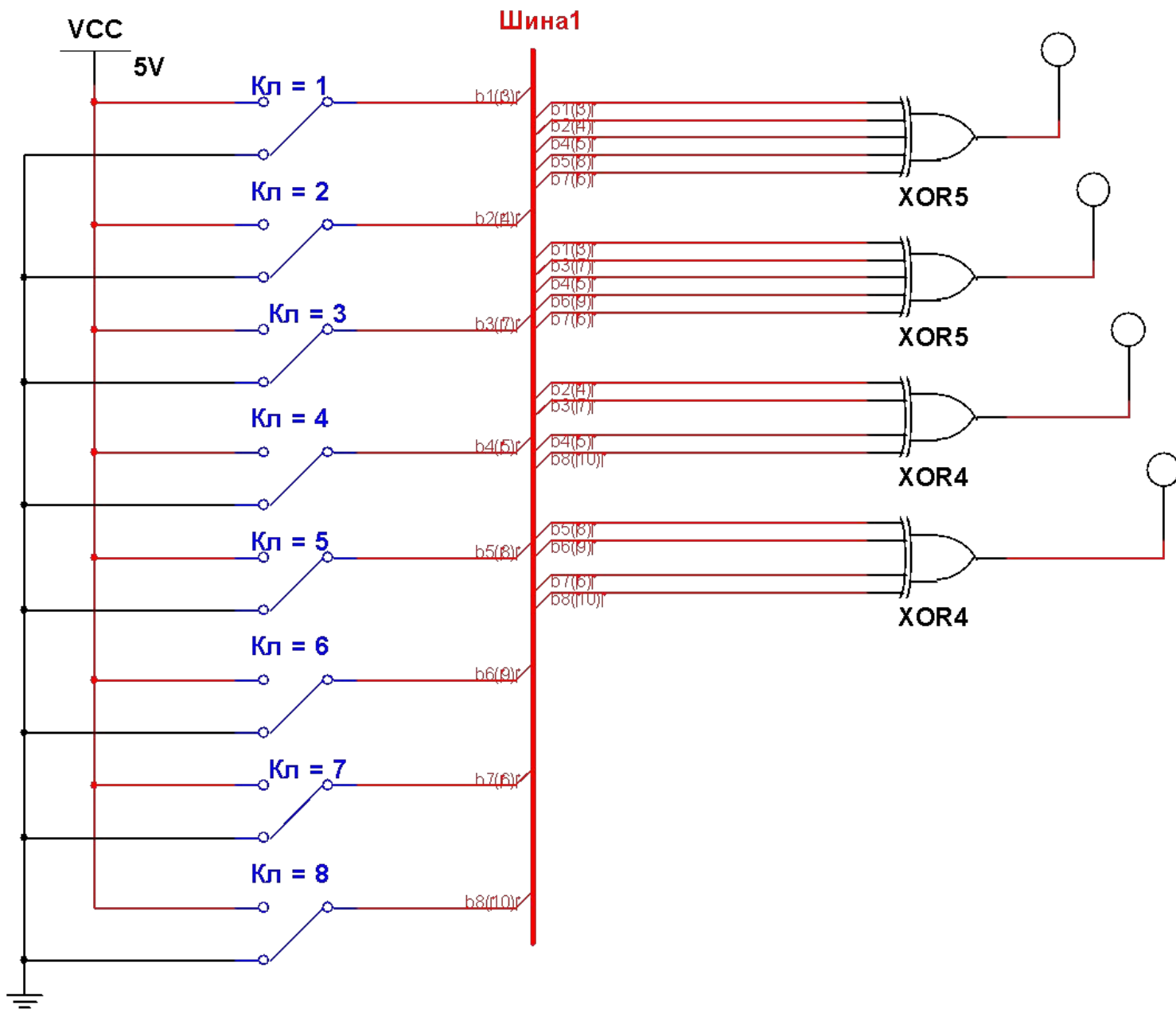
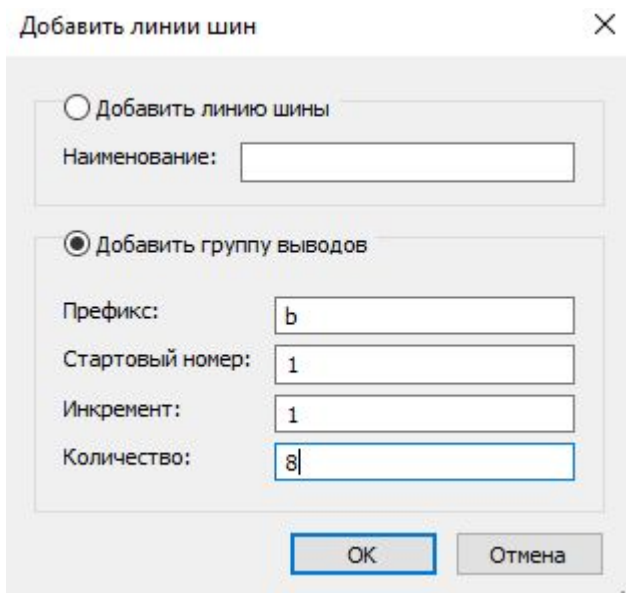
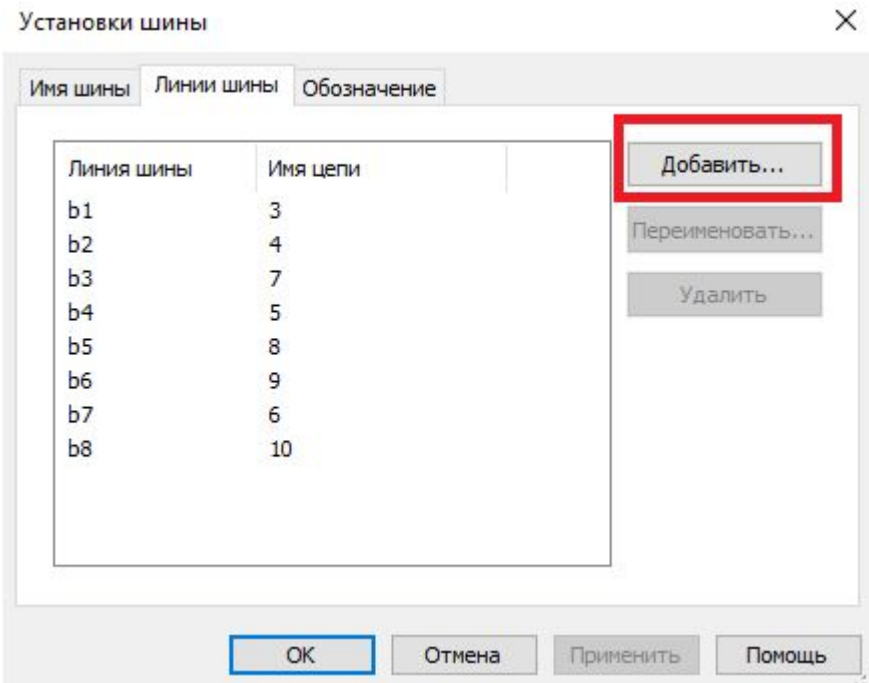


На передающей стороне формируются контрольные биты.

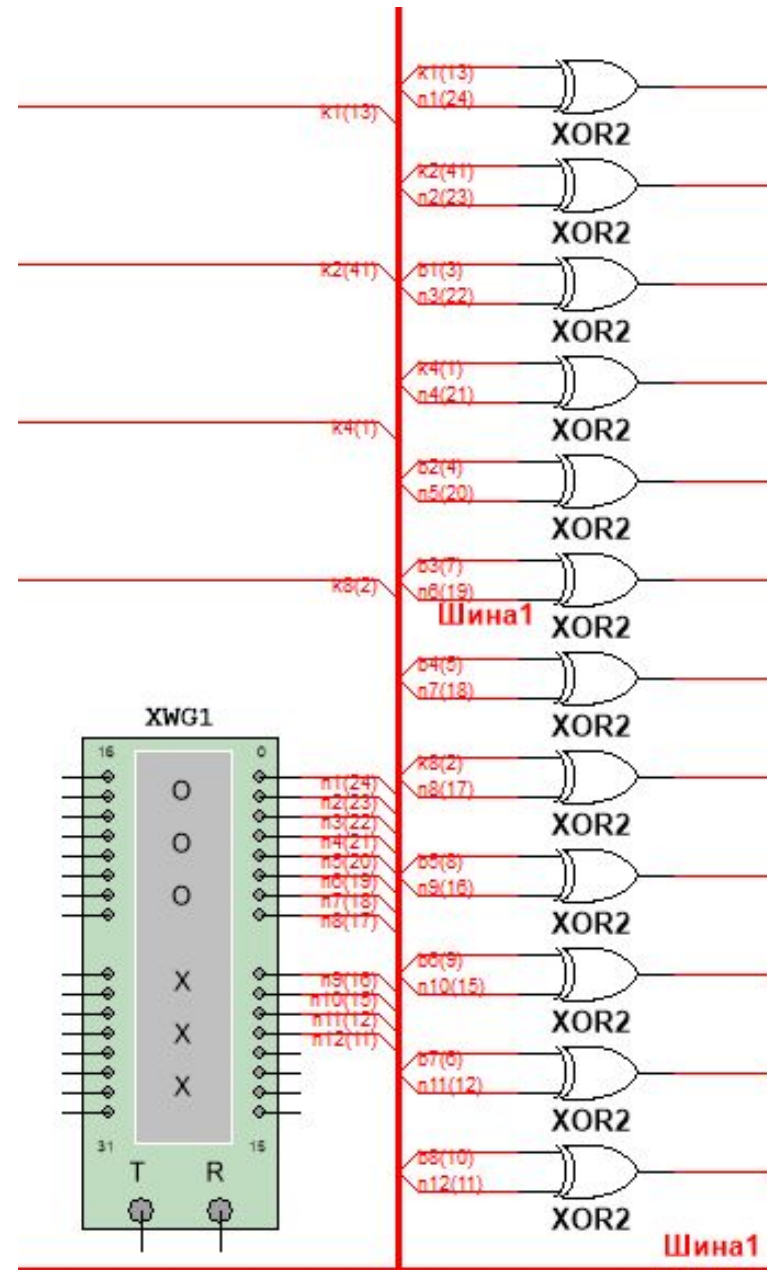
На приёмной стороне вычисляется синдром.

Имитатор помех позволяет исказить любой бит данных, передаваемых по каналу связи.

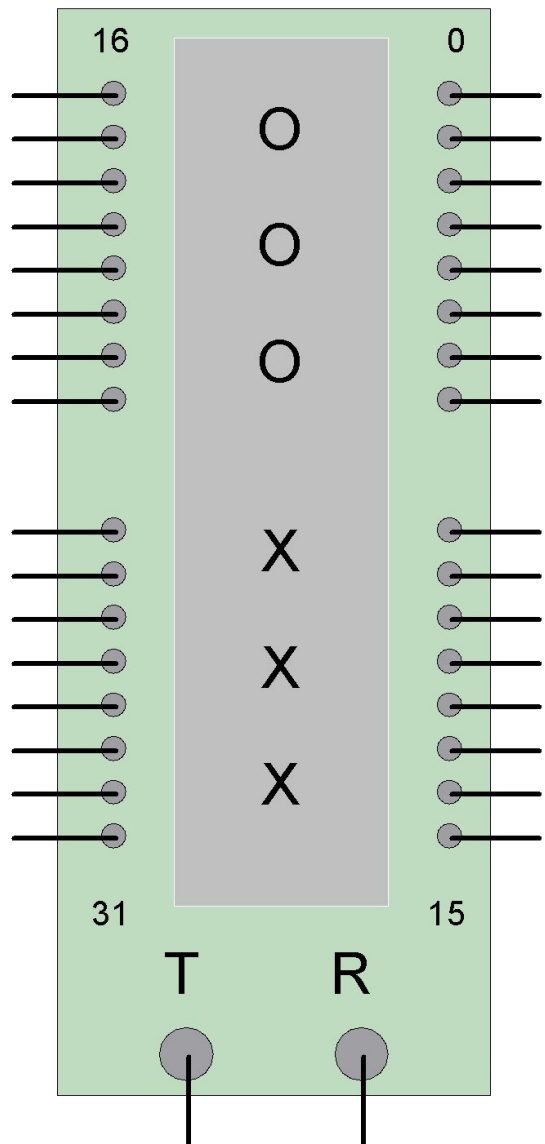
Пример схемы передающей стороны



Пример схемы канала связи



XWG1



Генератор & слов-XWG1

Управление: Циклически, Однократно, Пошагово, **Установки...**

Отображение: 16-ричное, Десятичное, Двоичное, ASCII

Запуск: Внутренний, Внешний

Частота: 10 Hz

Готовность Запуск

31 0

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

Установки

Конфигурация: Без изменений, Загрузить, Сохранить, Очистить буфер, Вверх, Вниз, Вправо, Влево

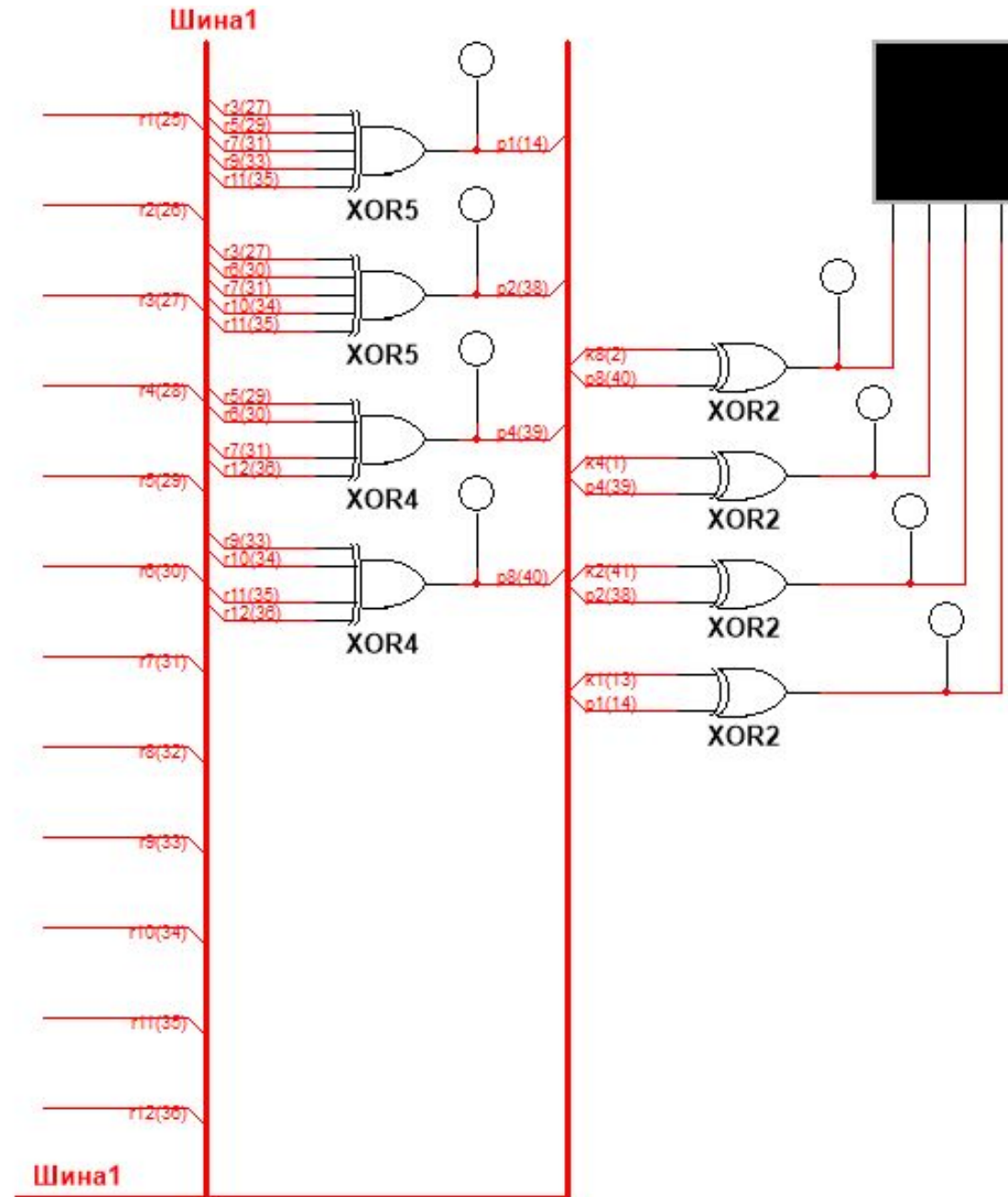
Отображение: 16-ричное, Десятичное

Размер буфера: [<= 0x2000] 0400

Инициализировать конфигурацию: 00000001

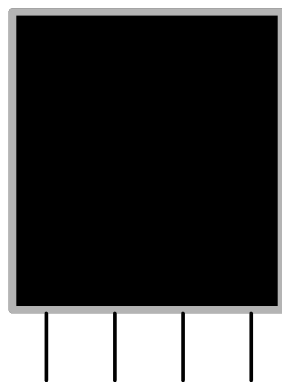
OK Отмена

Пример схемы принимающей стороны



Шестнадцатеричная система счисления

Десятичные и двоичные эквиваленты для шестнадцатеричных значений от 0 до F



**Семисегментный
индикатор**

Десятичные	Двоичные	Шестнадцатеричные
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Пример схемы системы передачи информации

