МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по проведению практических и лабораторных занятий
по дисциплине «Теория информации и кодирования»

Направление подготовки
09.03.01 Информатика и вычислительная техника

профиль Интеллектуальные системы обработки информации

Ростов на Дону
2022

Енгибарян И.А.

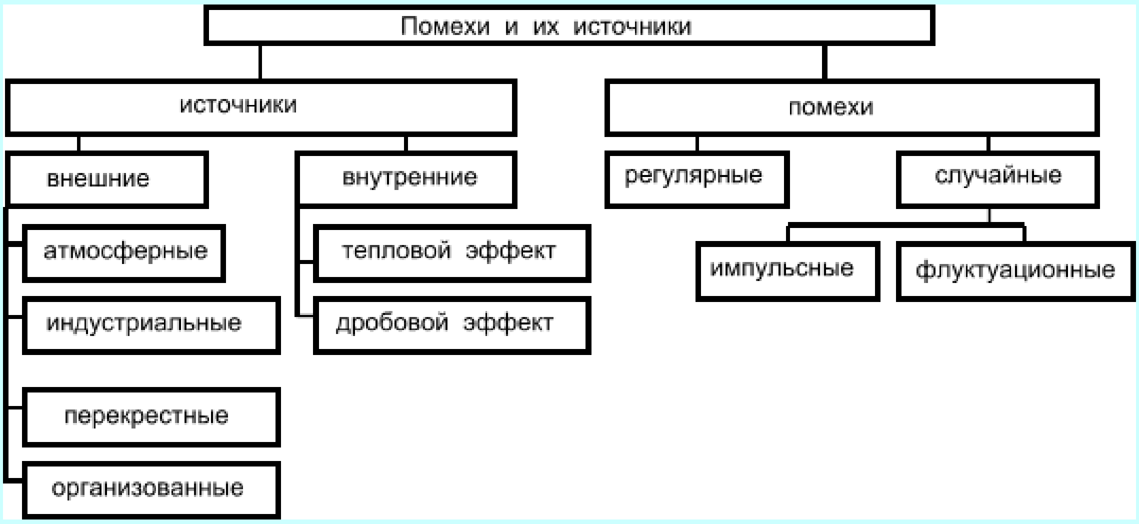
Методические указания по проведению практических и лабораторных занятий по дисциплине «Теория информации и кодирования»

Рассмотрены и одобрены
на заседании кафедры ИТСС
Протокол № 5 от 19.12.2022 г.

Содержание

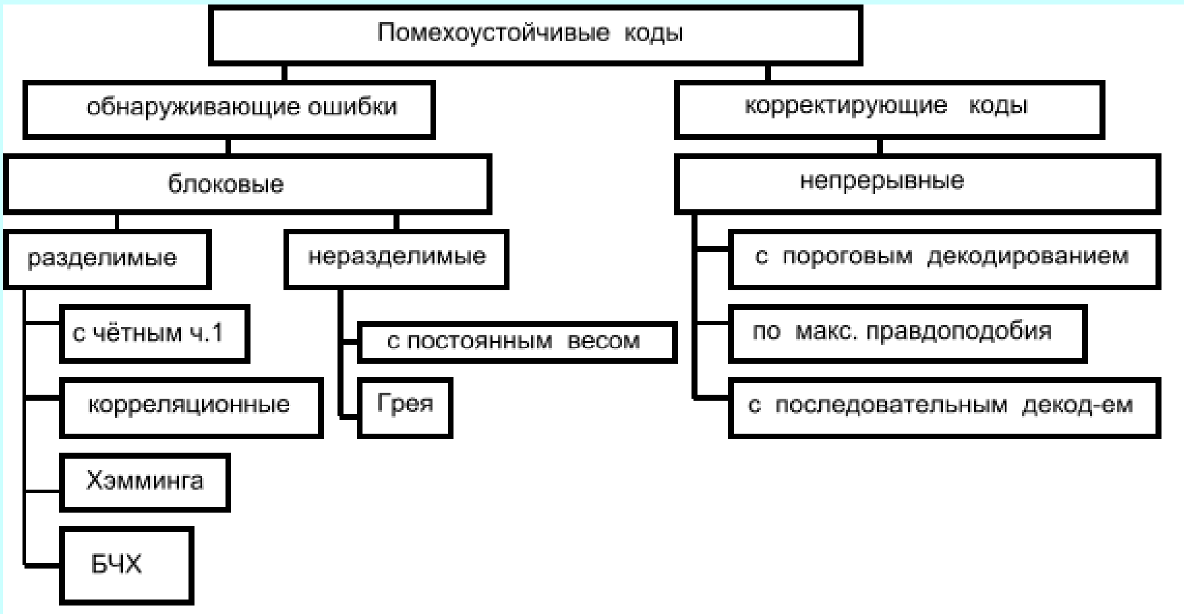
Помехоустойчивое кодирование

Под помехой понимается любое воздействие, накладывающееся на полезный сигнал и затрудняющее его прием. Ниже приведена классификация помех и их источников.



Внешние источники помех вызывают в основном импульсные помехи, а внутренние – флуктуационные. Помехи, накладываясь на видеосигнал, приводят к двум типам искажений: краевые и дробления. Краевые искажения связаны со смещением переднего или заднего фронта импульса. Дробление связано с дроблением единого видеосигнала на некоторое количество более коротких сигналов.

Приведем классификацию помехоустойчивых кодов.



Построение помехоустойчивых кодов в основном связано с добавлением к исходной комбинации (k – символов) контрольных (r – символов) см. на рис.5.1. Закодированная комбинация будет составлять n – символов. Эти коды часто называют (n, k) – коды.

Рис.5.1. Получение(n, k)-кодов.

где

k - число символов в исходной комбинации;

r - число контрольных символов.

Коды с обнаружением ошибок

1. Код с проверкой на четность.

Такой код образуется путем добавления к передаваемой комбинации, состоящей из k информационных символов, одного контрольного символа (0 или 1), так, чтобы общее число единиц в передаваемой комбинации было четным.

Пример 5.1. Построим коды для проверки на четность, где k – исходные комбинации, r – контрольные символы.

k	r	n
11011	0	110110
11100	1	111001

Определим, каковы обнаруживающие свойства этого кода. Вероятность P_{oo} обнаружения ошибок будет равна

$$P_{00} = C_n^1 p^1 (1-p)^{n-1} + C_n^3 p^3 (1-p)^{n-3} + C_n^5 p^5 (1-p)^{n-5} \ldots$$

Так как вероятность ошибок $p << 1$ является весьма малой величиной, то можно ограничится $P_{00} = C_n^1 P^1 (1-p)^{n-1}$

Вероятность появления всевозможных ошибок, как обнаруживаемых так и не обнаруживаемых, равна $P_\Sigma = 1 - Q$, где $Q = (1-p)^n$ - вероятность отсутствия искажений в кодовой комбинации. Тогда $P_\Sigma = 1 - (1-p)^n$.

При передаче большого количества кодовых комбинаций N_k , число кодовых комбинаций, в которых ошибки обнаруживаются, равно: $N_{00} = N_k \cdot P_{00} = N_k C_n^1 p (1-p)^{n-1}$

Общее количество комбинаций с обнаруживаемыми и не обнаруживаемыми ошибками равно $N_\Sigma = N_k \cdot P_\Sigma = N_k (1 - (1-p)^n)$

Тогда коэффициент обнаружения $K_{обн}$ для кода с четной защитой будет равен

$$K_{обн} = \frac{N_{00}}{N_\Sigma} = \frac{P_{00}}{P_\Sigma}$$

Например, для кода с k=5 и вероятностью ошибки $p = 10^{-2}$ коэффициент обнаружения составит $K_{обн} = 0.9$. То есть 90% ошибок обнаруживаем, при этом избыточность будет составлять $L = 1 - \frac{5}{6} = 0.17$ или 17%.

2. Код с постоянным весом.

Этот код содержит постоянное число единиц и нулей. Число кодовых комбинаций составит $N = C_n^k = \frac{n!}{k!(n-k)!}$

Пример 5.2. Коды с двумя единицами из пяти и тремя единицами из семи.

$N = C_5^2 = 10$	$N = C_7^3 = 35$
11000 10010 00101	0000111 1001001 1010100

Этот код позволяет обнаруживать любые одиночные ошибки и часть многократных ошибок. Не обнаруживаются этим кодом только ошибки смещения, когда одновременно одна единица переходит в ноль и один ноль переходит в единицу, два ноля и две единицы меняются на обратные символы и т.д.

Рассмотрим код с тремя единицами из семи. Для этого кода возможны смещения трех типов.

$$\left.\begin{matrix} 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{matrix}\right\} см1 \qquad \left.\begin{matrix} 00 \rightarrow 11 \\ 11 \rightarrow 00 \end{matrix}\right\} см2 \qquad \left.\begin{matrix} 000 \rightarrow 111 \\ 111 \rightarrow 000 \end{matrix}\right\} см3$$

Вероятность появления не обнаруживаемых ошибок смещения

$P_{НО} = P_{см1} + P_{см2} + P_{см3}$, где $\qquad P_{см1} = C_3^1 * C_4^1 * p^2 * (1-p)^5$

$P_{см2} = C_3^2 * C_4^2 * p^4 * (1-p)^3 \qquad P_{см3} = C_3^3 * C_4^3 * p^6 * (1-p)^1$

При p<<1 $P_{см3} << P_{см2} << P_{см1}$, тогда $P_{НО} = 12p^2(1-p)^5$

Вероятность появления всевозможных ошибок как обнаруживаемых, так и не обнаруживаемых будет составлять $P_{\Sigma} = 1 - Q = 1 - (1-p)^7$

Вероятность обнаруживаемых ошибок $P_{00} = P_{\Sigma} - P_{НО}$. Тогда коэффициент обнаружения будет равен $K_{обн} = \frac{P_{oo}}{P_{\Sigma}} = \frac{1-(1-p)^n - 12 * p^2 * (1-p)^5}{1-(1-p)^n}$

Например, код C_7^3 при $p = 10^{-2}$ коэффициент обнаружения составит $K_{обн} = 0.985$, избыточность L=27%.

3. Корреляционный код (Код с удвоением). Элементы данного кода заменяются двумя символами, единица ‘1’ преобразуется в 10, а ноль ‘0’ в 01.

Вместо комбинации 1010011 передается 10011001011010. Ошибка обнаруживается в том случае, если в парных элементах будут одинаковые символы 00 или 11 (вместо 01 и 10).

Например, при k=5, n=10 и вероятности ошибки $p = 10^{-2}$, $K_{обн} = 0.995$. Но при этом избыточность будет составлять 50%.

4. Инверсный код. К исходной комбинации добавляется такая же комбинация по длине. В линию посылается удвоенное число символов. Если в исходной комбинации четное число единиц, то добавляемая комбинация повторяет исходную комбинацию, если нечетное, то добавляемая комбинация является инверсной по отношению к исходной.

k	r	n
11011	11011	1101111011
11100	00011	1110000011

Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в первой основной группе символов. Если число единиц четное, то контрольные символы принимаются без изменения, если нечетное, то контрольные символы инвертируются. На втором этапе контрольные символы суммируются с информационными символами по модулю два. Нулевая сумма говорит об отсутствии ошибок. При ненулевой сумме, принятая комбинация бракуется. Покажем суммирование для принятых комбинаций без ошибок (1, 3) и с ошибками (2, 4).

1 11011	2 11111	3 11100	4 11000
11011	00100	11100	11100
00000	11011	00000	00100

Обнаруживающие способности данного кода достаточно велики. Данный код обнаруживает практически любые ошибки, кроме редких ошибок смещения, которые одновременно происходят как среди информационных символов, так и среди соответствующих контрольных. Например, при k=5, n=10 и $p = 10^{-2}$. Коэффициент обнаружения будет составлять $K_{обн} = 1 - 10^{-5}$.

5. Код Грея. Код Грея используется для преобразования угла поворота тела вращения в код. Принцип работы можно представить по рис.5.2. На пластине, которая вращается на валу, сделаны отверстия, через которые может проходить свет. Причём, диск разбит на сектора, в которых и сделаны эти отверстия. При вращении, свет проходит через них, что приводит к срабатыванию фотоприёмников. При снятии информации в виде двоичных кодов может произойти существенная ошибка. Например, возьмем две соседние цифры 7 и 8. Двоичные коды этих цифр отличаются во всех разрядах.

7 0111 –> 1111

8 1000 –> 0000

Если ошибка произойдет в старшем разряде, то это приведет к максимальной ошибке, на 360^0. А код Грея, это такой код в котором все соседние комбинации отличаются только одним символом, поэтому при переходе от изображения одного числа к изображению соседнего происходит изменение только на единицу младшего разряда. Ошибка будет минимальной.

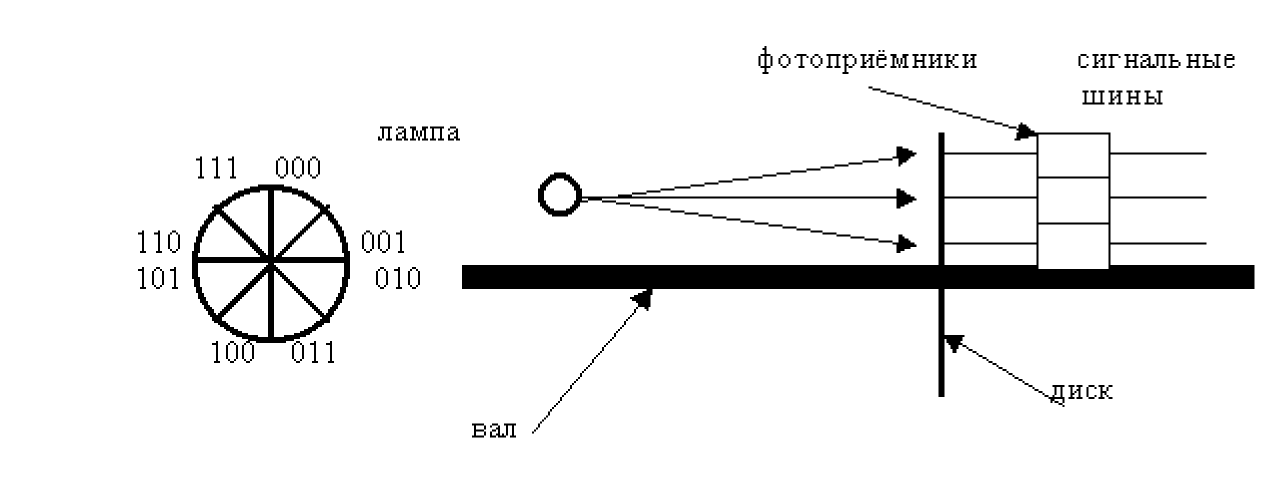


Рис.5.2. Схема съема информации угла поворота вала в код

Код Грея записывается следующим образом

Номер	Код Грея
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

Разряды в коде Грея не имеют постоянного веса. Вес k–разряда определяется следующим образом $\pm\sum_{i=0}^{k} 2^i = \pm(2^{k+1}-1)$.

При этом все нечетные единицы, считая слева направо, имеют положительный вес, а все четные единицы отрицательный.

Например, $1110 = 1 \cdot (2^{3+1} - 1) - 1 \cdot (2^{2+1} - 1) + 1 \cdot (2^{1+1} - 1) = 15 - 7 + 3 = 11_{10}$

Непостоянство весов разрядов затрудняет выполнение арифметических операций в коде Грея, поэтому необходимо уметь делать перевод кода Грея в обычный двоичный код и наоборот. Алгоритм перевода чисел можно представить следующим образом.

Пусть $A_n A_{n-1} \ldots A_0$-- двоичный код, $a_n a_{n-1} \ldots a_0$-- код Грея

Тогда переход из двоичного кода в код Грея выполнится по следующему алгоритму

$$a_i = \begin{cases} A_n, \ \textit{при} \ \ i = n \\ A_i, \ \textit{если} \ \ A_{i+1} = 0 \\ \overline{A}_i, \ \textit{если} \ \ A_{i+1} = 1 \end{cases}$$

Например, $1101_2 = 1011_{\text{Г}}$.

Обратный переход из кода Грея в двоичный код

$$A_i = \begin{cases} a_n, \textit{при} \ \ i = n \\ a_i, \ \textit{если } i - \textit{символу предшествует четное число } 1 \\ \overline{a}_i, \ \textit{если } i - \textit{символу предшествует нечетное число } 1 \end{cases}$$

Например, $1101_{\text{Г}} = 1011_2$.

Корректирующие коды

Корректирующими называются коды позволяющие обнаруживать и исправлять ошибки. Идею представления корректирующих кодов можно представить с помощью N-мерного куба. Возьмем трехмерный куб (рис.5.3), длина ребер, в котором равна одной единице. Вершины такого куба отображают двоичные коды. Минимальное расстояние между вершинами определяется минимальным количеством ребер, находящихся между вершинами. Это расстояние называется кодовым (или хэмминговым) и обозначается буквой d.

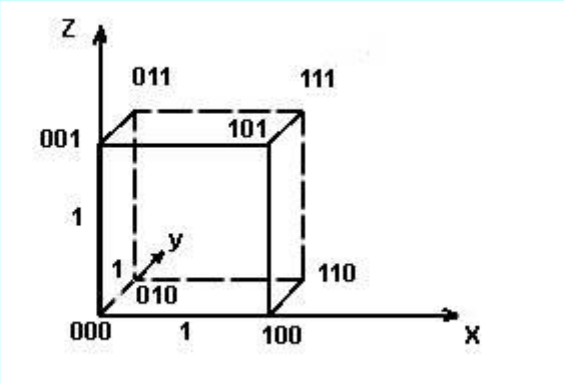


Рис.5.3. Представление двоичных кодов с помощью куба

Иначе, кодовое расстояние – это то минимальное число элементов, в которых одна кодовая комбинация отличается от другой. Для определения кодового расстояния достаточно сравнить две кодовые комбинации по модулю 2. Так, сложив две комбинации

10110101101

<u>11001010101</u>

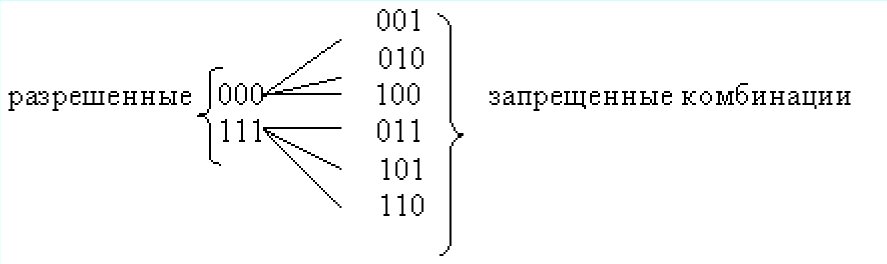
01111111000

определим, что расстояние между ними d=7.

Для кода с N=3 восемь кодовых комбинаций размещаются на вершинах трехмерного куба. Такой код имеет кодовое расстояние d=1, и для передачи используются все восемь кодовых комбинаций 000,001,..,111. Такой код является не помехоустойчивым, он не в состоянии обнаружить ошибку.

Если выберем комбинации с кодовым расстоянием d=2, например, 000,110,101,011, то такой код позволит обнаруживать однократные ошибки. Назовем эти комбинации разрешенными, предназначенными для передачи информации. Все остальные 001,010,100,111 - запрещенные.

Любая одиночная ошибка приводит к тому, что разрешенная комбинация переходит в ближайшую, запрещенную комбинацию (см. рис.5.3). Получив запрещенную комбинацию, мы обнаружим ошибку. Выберем далее вершины с кодовым расстоянием d=3



Такой код может исправить одну одиночную ошибку или обнаружить две ошибки. Таким образом, увеличивая кодовое расстояние можно увеличить помехоустойчивость кода. В общем случае кодовое расстояние определяется по формуле $d=t+l+1$ где t - число исправляемых ошибок , l - число обнаруживаемых ошибок. Обычно $l>t$.

Большинство корректирующих кодов являются линейными кодами. Линейные коды - это такие коды, у которых контрольные символы образуются путем линейной комбинации информационных символов. Кроме того, корректирующие коды являются групповыми кодами. Групповые коды (G_n) - это такие коды, которые имеют одну основную операцию. При этом, должно соблюдаться условие замкнутости (то есть, при сложении двух элементов группы получается элемент принадлежащий этой же группе). Число разрядов в группе не должно увеличиваться. Этому условию удовлетворяет операция поразрядного сложения по модулю 2. В группе, кроме того, должен быть нулевой элемент.

Пример 5.3. Ниже приведены кодовые комбинации, являющиеся группой или нет.

1) 1101 1110 0111 1011 – не группа, так как нет нулевого элемента.

2) 0000 1101 1110 0111 – не группа, так как не соблюдается условие замкнутости (1101+1110=0011).

3) 000 001 010 011 100 101 110 111 - группа

4) 000 001 010 111 - подгруппа

Большинство корректирующих кодов образуются путем добавления к исходной k – комбинации r – контрольных символов. В итоге в линию передаются n=k+r символов. При этом корректирующие коды называются (n, k) кодами. Как можно определить необходимое число контрольных символов?

Для построения кода способного обнаруживать и исправлять одиночную ошибку необходимое число контрольных разрядов будет составлять $n-k \geq \log(n+1)$. Это равносильно известной задаче о минимуме числа контрольных вопросов, на которые могут быть даны ответы вида “да” или “нет”, для однозначного определения одного из элементов конечного множества.

Если необходимо исправить две ошибки, то число различных исходов будет составлять C_n^2. Тогда $n-k \geq \log(1+C_n^1+C_n^2)$, в этом случае обнаруживаются однократные и двукратные ошибки. В общем случае, число контрольных символов должно быть не меньше

$$n-k \geq \log(1+C_n^1+C_n^2+\ldots+C_n^t) = \log\sum_{i=0}^{t} C_n^i \qquad (5.1)$$

Эта формула называется неравенством Хэмминга, или нижней границей Хэмминга для числа контрольных символов.

Код Хэмминга

Код Хэмминга, являющийся групповым (n, k) кодом, с минимальным расстоянием d=3 позволяет обнаруживать и исправлять однократные ошибки. Для построения кода Хэмминга используется матрица Н. $H = \left| A_k E_{n-k} \right|$, где A_k– транспонированная подматрица, E_{n-k} – единичная подматрица порядка n–k.

Если X - исходная последовательность, то произведение X·H=0. Пусть *E*– вектор ошибок. Тогда (X+E)·H = X·H+E·H = 0+E·H=E·H - синдром или корректор, который позволяет обнаружить и исправить ошибки. Контрольные символы $e_1, e_2, ..., e_r$ образуются из информационных символов, путем линейной комбинации $e_j = a_{j1}x_1 + a_{j2}x_2 + ... + a_{jk}x_k$, где a_j={0,1} – коэффициенты, взятые из подматрицы *A* матрицы *H*.

Рассмотрим Построение кода Хэмминга для k=4 символам. Число контрольных символов *r=n–k* можно определить по неравенству Хэмминга $n - k \geq \log(n+1)$ для однократной ошибки. Но так, как нам известно, только исходное число символов k, то проще вычислить по эмпирической формуле

$$r = n - k = [\log\{(k+1) + [\log(k+1)]\}], \qquad (5.2)$$

где [.] – означает округление до большего ближайшего целого значения. Вычислим для k=4 $r = n - k = [\log\{(4+1) + [\log(4+1)]\}] = 3$. Получим код (n,k)=(7,4); n=7; k=4; r=n-k=3; d=3. Построим матрицу Н.

$$H = \begin{vmatrix} 1\,2\,3\,4\,5\,6\,7 \\ 0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1 \end{vmatrix} \Rightarrow \begin{vmatrix} 3 & 5 & 6 & 7 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ x_3 & x_5 & x_6 & x_7 & e_4 & e_2 & e_1 \end{vmatrix} = \left| A_k, E_{n-k} \right|$$

Контрольные символы e_j определим по формуле $e_j = a_{j1}x_1 + a_{j2}x_2 + ... + a_{jk}x_k$. Например, $e_4 = 0 \cdot x_3 + 1 \cdot x_5 + 1 \cdot x_6 + 1 \cdot x_7$. Для простоты оставляем только слагаемые с единичными коэффициентами. В результате получим систему линейных уравнений, с помощью которых вычисляются контрольные разряды. Каждый контрольный разряд является как бы дополнением для определенных информационных разрядов для проверки на четность.

$$\left. \begin{array}{l} e_4 = x_5 + x_6 + x_7 \\ e_2 = x_3 + x_6 + x_7 \\ e_1 = x_3 + x_5 + x_7 \end{array} \right|$$

При декодировании вычисляем корректор $K=k_4k_2k_1$

$$\left. \begin{array}{l} k_4 = e_4 + x_5 + x_6 + x_7 \mod 2 \\ k_2 = e_2 + x_3 + x_0 + x_7 \mod 2 \\ k_1 = e_1 + x_3 + x_5 + x_7 \mod 2 \end{array} \right|$$

Если корректор равен нулю, следовательно, ошибок нет. Если корректор не равен нулю, то местоположение вектор–столбца матрицы H, совпадающего с вычисленным корректором, указывает место ошибки. При передаче может возникнуть двойная и более ошибка. Корректор также не будет равен нулю. В этом случае произойдет исправление случайного символа и нами будет принят неверный код. Для исключения такого автоматического исправления вводится еще один символ $e_0 = e_1 + e_2 + x_3 + e_4 + x_5 + x_6 + x_7$ для проверки всей комбинации на четность. Кодовое расстояние d=4. Тогда матрица H будет иметь вид

$$\left| \begin{array}{cccc} 3\,5\,6\,7 & 4\,2\,1\,0 \\ 0\,1\,1\,1 & 1\,0\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0\,0 \\ 1\,1\,0\,1 & 0\,0\,1\,0 \\ 1\,1\,1\,1 & 1\,1\,1\,1 \end{array} \right|$$

Пример 5.4. Дана 1101 - исходная комбинация (k=4). Закодировать ее в коде Хэмминга.

По формуле (5.2) находим число контрольных символов r=3. Берем регистр из 7 ячеек памяти. Размещаем исходную комбинацию в ячейках 3, 5, 6, 7.

1 2 3 4 5 6 7

* * 1 * 1 0 1

Находим контрольные символы

$e_4 = 5 + 6 + 7 = 1 + 0 + 1 = 0$

$e_2 = 3 + 6 + 7 = 1 + 0 + 1 = 0$

$e_1 = 3 + 5 + 7 = 1 + 1 + 1 = 1$

Закодированная комбинация будет иметь вид

1 2 3 4 5 6 7

1 0 1 0 1 0 1

Допустим, что при передаче возникла ошибка, и мы приняли неверную комбинацию

1 2 3 4 5 6 7

1 0 1 0 1 $\underline{1}$ 1

Проверяем ее

$к_4 = 4 + 5 + 6 + 7 = 0 + 1 + 1 + 1 = 1$

$к_2 = 2 + 3 + 6 + 7 = 0 + 1 + 1 + 1 = 1$

$к_1 = 1 + 3 + 5 + 7 = 1 + 1 + 1 + 1 + 0$

$K = k_4, k_2, k_1 = 110$- в шестом разряде ошибка.

Если бы нам понадобилось построить код и для проверки двойных ошибок, необходимо было бы ввести еще один дополнительный нулевой разряд.

$$e_0 = e_1 + e_2 + x_3 + e_4 + x_5 + x_6 + x_7 = 1 + 0 + 1 + 0 + 1 + 0 + 1 = 0$$

Получим следующий код

0 1 2 3 4 5 6 7

0 1 0 1 0 1 0 1

При передаче и возникновении ошибки код будет иметь вид

0 1 2 3 4 5 6 7

0 1 0 1 0 1 1 1

Проверка в этом случае показала бы, что корректор K=110, а проверка всей комбинации на четность $E_0 = 0+1+0+1+0+1+1+1=1$. Это указывает на одиночную ошибку. Допускается автоматическое исправление ошибки.

Существует следующий алгоритм декодирования кода Хэмминга с d=4

Корректор – K	**Значение E_0**	**Вывод**
K=0	E_0=0	Ошибок нет
K#0	E_0#0	Произошла одиночная ошибка
K#0	E_0=0	Произошла двойная ошибка. Исправление запрещено.
K=0	E_0#0	Произошла тройная или более нечетная ошибка

Код (7,4) является минимально возможным кодом с достаточно большой избыточностью. Эффективность кода (k/n) растет с увеличением длины кода

Длина кода – n	7	15	31	63
Число информационных разрядов – k	4	11	26	57
Число контрольных разрядов – r	3	4	5	6
Эффективность кода k/n	0,57	0,73	0,84	0,9

Техническая реализация кода Хэмминга

Схемы кодирования и декодирования представлены на рис.5.4 и 5.5 соответственно. Устройство кодирования строится на регистре в n разрядов и r сумматоров “по модулю 2”. В ячейки 3, 5, 6, 7 вводятся исходные символы.

В следующем такте формируются проверочные разряды. Закодированная комбинация может быть передана в линию последовательно или параллельно.

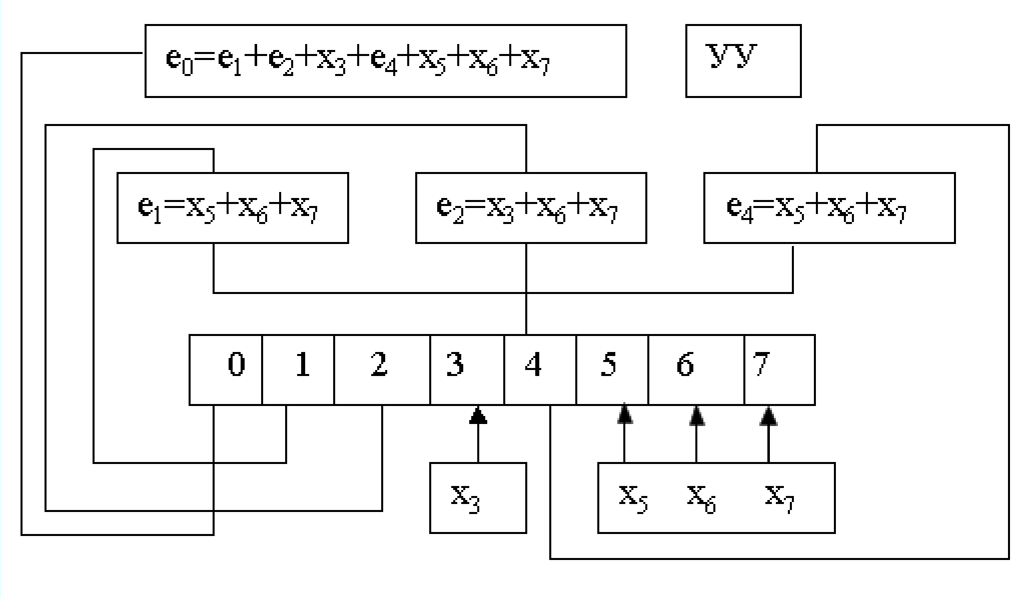


Рис.5.4. Схема кодирования

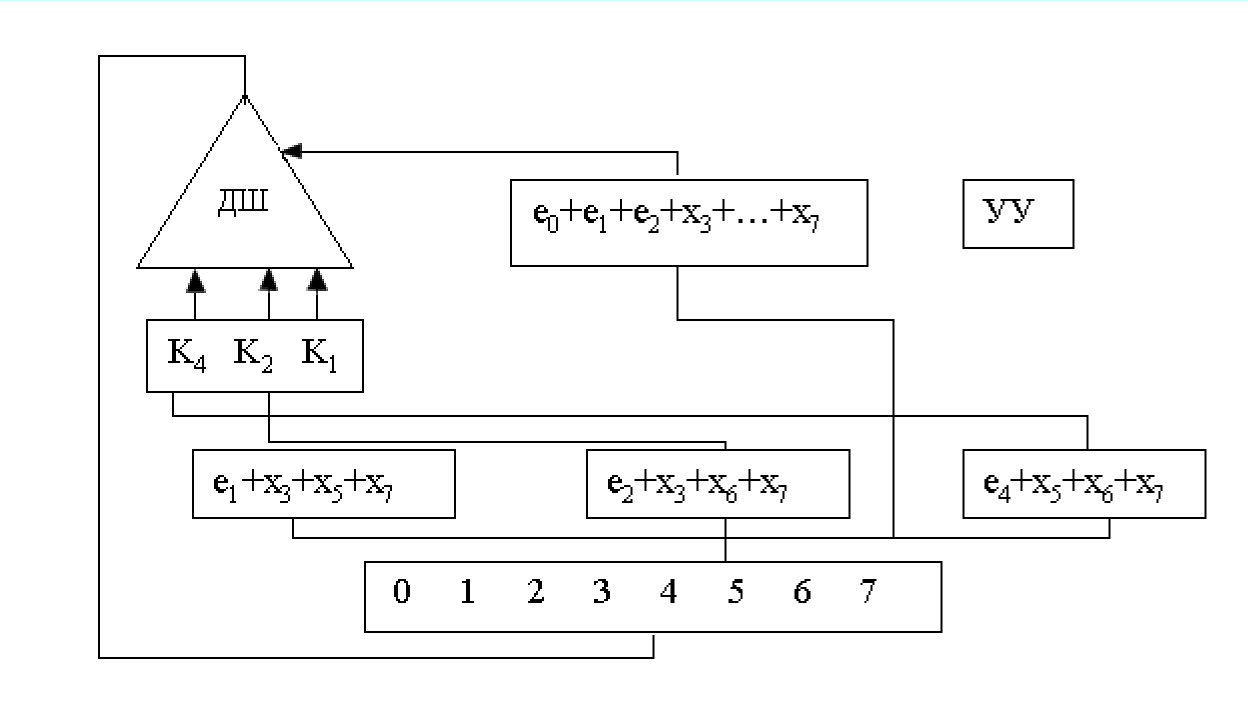


Рис.5.5. Схема декодирования

Циклические коды

Циклическими кодами называют специальную группу кодов, для построения которых могут быть использованы циклические свойства квадратных матриц, а также коды, которые описываются неприводимыми, образующими (порождающими) многочленами (полиномами). Например, для кодовой комбинации 101101 полиномиальное представление таково:

$A(X) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 = x^5 + x^3 + x^2 + 1.$

Циклические коды относятся к систематическим (n, k) кодам, в которых контрольные r и информационные k разряды расположены на строго определенных местах: $n = k + r$.

Рассмотрим алгебру циклических кодов. Допустим, необходимо перемножить три многочлена $(x^3+x^2+1) \cdot (x^3+x+1) \cdot (x+1)$. Действия производятся также как в обычной алгебре, только сложение проводится по модулю 2.

$$
\begin{array}{r}
x^3+x^2+1 \\
\underline{x^3+x+1} \\
x^3+x^2+0+1 \\
x^4+x^3+0+x \\
\underline{x^6+x^5+0+x^3} \\
x^6+x^5+x^4+x^3+x^2+x+1 \\
\underline{x+1} \\
x^6+x^5+x^4+x^3+x^2+x+1 \\
\underline{x^7+x^6+x^5+x^4+x^3+x^2+x} \\
x^7+0+0+0+0+0+0+1=x^7+1
\end{array}
\qquad
\begin{array}{r}
1101 * 1011 \\
1101 \\
\underline{1101} \\
1111111 * 11 \\
\underline{1111111} \\
10000001 \;=\; x^7+1
\end{array}
$$

При делении операция вычитания заменяется операцией сложения по модулю 2. Например, необходимо разделить многочлен седьмой степени на многочлен третей степени $(x^7+x^5+x^4+x+1) / (x^3+x^2+1)$

Операция деления может быть произведена или в виде многочленов или в виде двоичных кодов.

$$
\begin{array}{l|l}
x^7+0+x^5+x^4+0+0+x+1 & \underline{x^3+x^2+1} \\
\underline{x^7+x^6+0+x^4} & x^4+x^3+1 \\
\quad x^6+x^5+0+0 & \\
\quad \underline{x^6+x^5+0+x} & \\
\qquad\qquad x^3+0+x+1 & \\
\qquad\qquad \underline{x^3+x^2+0+1} & \\
\qquad\qquad\quad x^2+x &
\end{array}
\qquad\qquad
\begin{array}{l|l}
-10110011 & \underline{1101} \\
\ \underline{1101} & 11001 \\
\quad 1100 & \\
\quad \underline{1101} & \\
\qquad 1011 & \\
\qquad \underline{1101} & \\
\qquad\ 110 &
\end{array}
$$

Схема деления реализуется на регистрах сдвига со встроенными сумматорами по модулю 2. Вид схемы определяется многочленом, на который производится деление. В процессе деления с помощью такого устройства находится остаток.

Пример 5.5. Построить схему деления на многочлен $g(x)=x^3+x+1$ (1011)

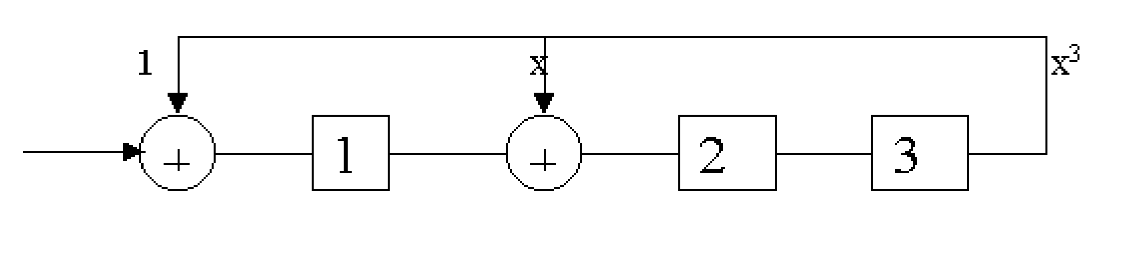


Рис.5.6. Схема деления на многочлен $g(x)=x^3+x+1$

Пусть на вход подается комбинация 10110001

В процессе алгебраического деления получается остаток 001

$$
\begin{array}{l}
10110001 \\
\underline{1011} \\
\qquad 001
\end{array}
$$

Процесс деления с помощью устройства показан в таблице 5.1.

Таблица 5.1

Вх	1	2	3
1	1	0	0

0	0	1	0
1	1	0	1
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
1	1	0	0

Циклический код получают следующим образом: заданный многочлен h(x) сначала умножается на одночлен x^{n-k}, затем делится на образующий многочлен g(x). В результате получим

$$\frac{h(x)x^{n-k}}{g(x)} = Q(x) + \frac{R(x)}{g(x)} \qquad (5.3)$$

или

$$F(x) = Q(x) \cdot g(x) = x^{n-k}h(x) + R(X) \qquad (5.4)$$

Таким образом, циклический код можно построить умножением кодовой комбинации h(x), являющейся заданной, на одночлен x^{n-k} добавлением к этому произведению остатка R(x). При декодировании, принятую кодовую комбинацию необходимо разделить на g(x). Наличие остатка указывает на ошибку.

Образующий полином g(x) является сомножителем при разложении двучлена x^n+1. Сомножителями разложения двучлена являются неприводимые полиномы (таблица 5.3).

Образующий полином выбирают следующим образом. По заданной кодовой комбинации k определяют число контрольных символов из соотношения r = log (n + 1) или по эмпирической формуле

$$r = [\log\{(k + 1) + [\log(k + 1)]\}] \qquad (5.5)$$

Соотношение значений n, k, r можно определить по таблице 5.2.

Таблица 5.2 зависимостей между n, k и r

n	3	5	6	7	9...15	17...31	33...63	65...127

k	1	2	3	4	5...11	12...26	27...57	28...120
r	2	3	3	3	4	5	6	7

Из таблицы неприводимых полиномов (табл.5.3) выбирают самый короткий многочлен со степенью, равной числу контрольных символов; его и принимают за образующий полином.

Пример 5.6. Пусть требуется закодировать комбинацию вида 1101, что соответствует $h(x) = x^3 + x^2 + 1$. По формуле (5.5) определяем число контрольных символов $r = 3$. Из таблицы 5.3 возьмем многочлен $g(x) = x^3 + x + 1$, т.е. 1011.

Решение:

Умножим h(x) на x^r.

$h(x)x^r = (x^3 + x^2 + 1)x^3 = x^6 + x^5 + x^3 \rightarrow 11010000$

Разделим полученное произведение на образующий полином g(x)

$$\frac{h(x)x^r}{g(x)} = \frac{x^6 + x^5 + x^3}{x^3 + x + 1} = x^3 + x^2 + x + 1\frac{1}{x^3 + x + 1} \rightarrow 1111 + \frac{001}{1011}$$

При делении необходимо учитывать, что вычитание производится по модулю 2. Остаток суммируем с $h(x)x^r$. В результате получим закодированное сообщение:

$F(x) = (x^3 + x^2 + 1)\,(x^3 + x + 1) = (x^3 + x^2 + 1)x^3 + 1 \rightarrow 1101001$

В полученной кодовой комбинации циклического кода информационные символы h(x) = 1101, а контрольные R(x) = 001. Закодированное сообщение делится на образующий полином без остатка.

Сообщение, которое закодировано, является одной из комбинаций 4-разрядного кода, так как весь ансамбль сообщений (вся группа) содержит N=16 сообщений. Это значит, что если все сообщения передаются в закодированном виде, то каждое из них необходимо кодировать так же, как и комбинацию h(x) = 1101. Однако выполнять дополнительные 15 расчетов (а в общем случае 2^{n-k}-1 расчет) нет необходимости. Это можно сделать проще, путем составления образующей (порождающей) матрицы.

Образующая матрица составляется на основе единичной транспонированной, к которой справа дописывается матрица дополнений:

$$H_{n,k} = \| I_k, C_{n,r} \| \qquad (5.6)$$

Матрица дополнений получается из остатков от деления единицы с нулями на образующий многочлен g(x). Комбинации единиц с нулями представляют собой векторы ошибок: 00...01, 00... 10, 00... 1...0 и т.д. Каждому вектору ошибок будет соответствовать свой остаток (опознаватель):

```
1000000…
1011
———
011          1-й остаток      011              1  || 0001   011 ||
1100         2-й остаток      110       H7,4 = 2  || 0010   110 ||
1011                                           3  || 0100   111 ||
———                                            4  || 1000   101 ||
1110         3-й остаток      111
1011
———
1010         4-й остаток      101
```

Получено 4 комбинации циклического кода, т.е. столько, сколько информационных разрядов, а так как в 4-разрядном двоичном коде всего N = 2^4 = 16 комбинаций, то остальные 11 ненулевых комбинаций находятся суммированием по модулю 2 всевозможных сочетаний строк образующей матрицы. Например, необходимо из исходных кодов 1101 и 1010 получить циклические помехозащищенные коды. Они получаются суммированием соответствующих строк образующей матрицы:

1. 1+3+4 = 1101001;
2. 2+4 = 1010011.

Декодирование циклических кодов

Для обнаружения и исправления ошибок принятая комбинация делится на образующий многочлен $g(x)$. Если остаток $R(x) = 0$, значит, комбинация принята без ошибок. Наличие остатка свидетельствует о том, что комбинация принята искаженной. Значение остатка совпадет с одним из опознавателей матрицы H**,** который и укажет на местоположение ошибки по вектору ошибок.

```
1100001
1011
 1110
 1011
  1010
  1011
   011 – ошибка в четвертом разряде
```

Если ошибка содержится в одном из поверочных разрядов, то одночлен одиночной ошибки будет иметь степень, меньшую, чем степень образующего многочлена и совпадет с остатком от деления. При этом номер разряда остатка прямо укажет на номер искаженного поверочного разряда.

```
1101011
1011
 1100
 1011
  1111
  1011
   1001
   1011
    010  – ошибка во 2–м контрольном разряде
```

Существует более общий алгоритм обнаружения и исправления ошибок:

1. Принятая комбинация делится на образующий многочлен $g(x)$. Если остаток $R(x)<>0$ то определяется вес остатка w. Если вес остатка равен или меньше числа исправляемых ошибок t ($w<=t$), то принятую комбинацию складывают по модулю 2 с остатком и получают исправленную комбинацию.

2. Если w>t, то производится циклический сдвиг на один символ влево и полученная после такого сдвига комбинация снова делится на образующий многочлен. Если вес полученного остатка w<=t, то циклически сдвинутую комбинацию складывают с остатком и затем после сложения циклически сдвигают в обратную сторону вправо на один символ (возвращают на прежнее место). В результате получаем исправленную комбинацию.

3. Если после циклического сдвига на один символ по прежнему w>t, то производят дополнительные циклические сдвиги влево. При этом после каждого сдвига осуществляется деление сдвинутой комбинации на g(x) и проверяется вес остатка. При w<=t сдвинутую комбинацию складывают с остатком и производят обратных циклических сдвигов вправо столько, сколько было сделано влево.

Пример 5.7. Необходимо проверить принятый код 1101110, для g(x)=1011 и t=1.

1. Принятый код 1101110 делим на g(x), находим остаток R(x)=111, w=3

2. Код 1101110 сдвигаем влево на один разряд, получаем 1011101. Делим на образующий многочлен g(x). Находим остаток R(x)=101, вес w=2

3. Снова производим сдвиг влево, получаем 0111011. Делим на g(x). Остаток R(x)=001. w=1

4. Складываем сдвинутую комбинацию с остатком 0111011+001=0111010

5. Производим два циклических сдвига вправо 0111010 –> 0011101 –>1001110

В результате получили исправленную комбинацию.

Аппаратурная реализация циклических кодов

Циклические коды реализуются с помощью сдвиговых регистров. Схема кодирования (образуется с помощью деления на образующий многочлен):

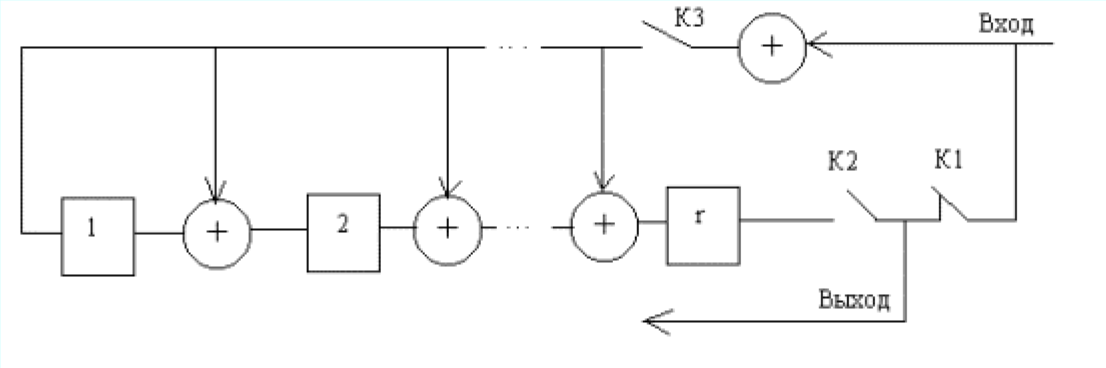


Рис. 5.7. Схема кодирования

Ключи к1 и к2 первоначально замкнуты, а ключ к3 – разомкнут. Исходная комбинация через ключ к1 поступает на выход и через входной сумматор на сдвиговый регистр, где и образуется контрольные символы. Затем ключ к2 замыкается, а к1 и к3 размыкаются. Контрольные символы подаются на выход в след, за информационными символами.

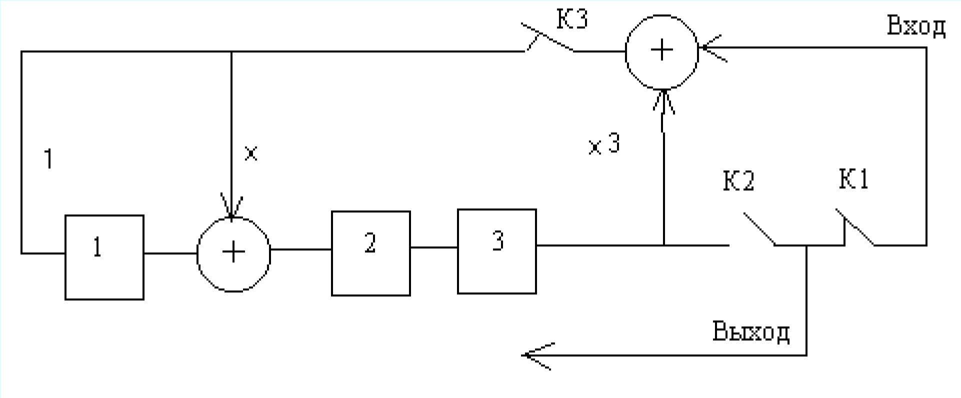


Рис.5.8. Схема кодирования методом деления на образующий многочлен $g(x)=x^3+x+1$

Пример 5.8. Пусть на вход подается комбинация 1101. Последовательность деления представлена в таблице. В результате на выходе будет получена комбинация 1101001

Вх	1	2	3
1	1	1	0
1	1	0	1
0	1	0	0
1	1	0	0

Пример 5.9. Пусть на вход подается комбинация 1001. В результате деления будет получена комбинация 1001011

Вх	1	2	3
1	1	1	0
1	1	0	1
0	1	0	0
1	1	0	0

Схема декодирования (образуется с помощью деления на образующий многочлен):

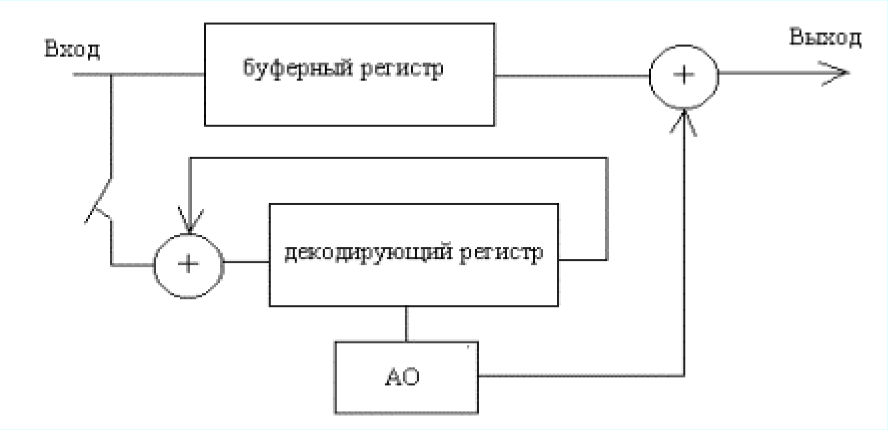


Рис.5.9. Схема декодирования циклического кода. АО – анализатор ошибок.

Исходная комбинация подается в буферный регистр и одновременно через ключ в декодирующий регистр. Если с приходом последнего символа, зафиксирован нулевой остаток, то ошибок нет, и, если не нулевой, то есть ошибка. Принятая комбинация подается через выходной сумматор, и искаженный сигнал исправляется анализатором ошибок (АО).

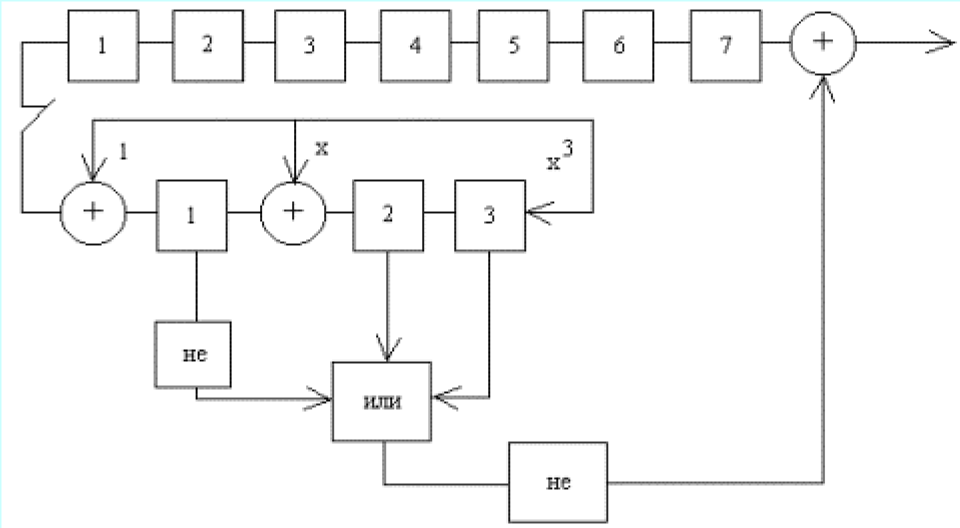


Рис.5.10. Пример схемы декодирования методом деления на полином $g(x)=x^3+x+1$

Пример 5.10. Пусть декодирующий регистр построен методом деления на образующий полином $g(x)=x^3+x+1$.

Пусть на вход подается комбинация 1101001

Вх	1	2	3
1	1	0	0
1	1	1	0
0	0	1	1
1	0	1	1
0	1	1	1
0	1	0	1

1	0	0	0

В результате деления получился нулевой остаток, следовательно, ошибок нет.

Теперь пусть на вход подается комбинация с ошибкой 1100001

Вх	1	2	3
1	1	0	0
1	1	1	0
0	0	1	1
0	1	1	1
0	1	0	1
0	1	0	0
1	1	1	0

В результате получился не нулевой остаток 110. Отключим ключ и начнем выводить комбинацию из буферного регистра. На четвертом такте

№	1	2	3
---	1	1	0
1	0	1	1
2	1	1	1
3	1	0	1
4	1	0	0

анализатор ошибок подаст на выходной сумматор сигнал исправления. Ошибка будет исправлена.

Для исправления многократных ошибок используется произведение образующих многочленов: $g(x^3) = g_1(x) \cdot g_2(x) K\ g_k(x)$

Таблица 5.3. Неприводимые полиномы

g(x)	**полином**	**g(x)**	**полином**
$g(x)$	$x+1$	$g_1(x^6)$	x^6+x+1
$g(x^2)$	x^2+x+1	$g_2(x^6)$	x^6+x^3+1
$g_1(x^3)$	x^3+x^1+1	$g_3(x^6)$	x^6+x^5+1
$g_2(x^3)$	x^3+x^2+1	……………	……………

$g_1(x^4)$	x^4+x+1	$g_1(x^7)$	x^7+x+1
$g_2(x^4)$	x^4+x^3+1	$g_2(x^7)$	x^7+x^3+1
$g_3(x^4)$	$x^4+x^3+x^2+x+1$	$g_3(x^7)$	$x^7+x^3+x^2+x+1$
$g_1(x^5)$	x^5+x^2+1	……………	……………
$g_2(x^5)$	x^5+x^3+1	$g_1(x^8)$	$x^8+x^4+x^3+x+1$
$g_3(x^5)$	$x^5+x^3+x^4+x+1$	$g_2(x^8)$	$x^8+x^4+x^3+x^2+1$
$g_4(x^5)$	$x^5+x^4+x^2+x+1$	……………	……………
$g_5(x^5)$	$x^5+x^4+x^3+x+1$	$g_1(x^9)$	$x^9+ x+1$
$g_6(x^5)$	$x^5+x^4+x^3+x^2+1$	$g_2(x^9)$	$x^9+x^4+ 1$

Вопросы

1. Каковы причины возникновения помех, и какие помехи вызываются этими причинами.
2. Назовите коды, предназначенные для обнаружения ошибок.
3. Дайте определение инверсного кода.
4. С какой целью используется код Грея.
5. Какие коды называются корректирующими.
6. Как определяется минимальное кодовое расстояние?
7. Дайте определение группового кода.
8. Из какого соотношения определяется необходимое число контрольных символов?
9. Нарисуйте схему кодирования кода Хэмминга.
10. Какие коды называются циклическими?
11. Для чего используется представление кодов в виде многочленов?
12. Приведите примеры алгебры циклических кодов.
13. Из каких соображений выбирается образующий многочлен?
14. Поясните процесс получения циклического кода математическим методом.
15. Как построить образующую матрицу циклического кода?
16. Нарисуйте схему кодирования циклического кода.
17. Поясните процесс декодирования циклического кода.

Упражнения

1. Определить величину кодового расстояния между двумя комбинациями 1101101, 1001011.
2. Определить величину кодового расстояния d, обеспечивающего исправление .s-кратных ошибок при s = 1, 3, 5.
3. Определить наименьшее количество проверочных элементов и n-k, необходимое для исправления трехкратных ошибок, если число элементов в кодовой комбинации равно 12.
4. Произвести перемножение трех многочленов в алгебре циклических кодов (x + 1), (x^3 + x +1) и ($x^3 + x^2$ + 1). Проделать аналогичную операцию и для двоичных эквивалентов.
5. Найти остаток при делении многочлена ($x^7+x^6+x^4+x+1$)... на многочлен (x^4+x+1). Аналогичную операцию проделать и для двоичных эквивалентов.
6. Закодировать в циклическом коде комбинации 1001, 1010, если образующий многочлен $g(x)=x^3+x+1$
7. Закодировать многочлен $x^7+x^4+x^3+x+1$ с проверкой на четность.
8. По заданному образующему многочлену $g(x)=x^4+x^3+1$ построить образующую (проверочную) матрицу H=I,C, усеченную до 6 разрядов.
9. Число информационных символов в кодовой комбинации k=11. Выбрать образующий многочлен g(x) с условием исправления одиночной ошибки и построить проверочную матрицу $H_{n,k}$.
10. Проверить принятую кодовую комбинацию $x^{14}+x^{11}+x^8+x^6+x^3+x^2+x$ на наличие одиночной ошибки, если образующий полином $g(x)=x^4+x^3+1$. При обнаружении ошибки исправить ее.
11. Закодировать в циклическом коде следующие комбинации:

1) 100011; 2) 110011; 3) 100101; 4) 100110; 5) 100111;

6) 110111; 7) 101001.

Практические задания

Исследование способов построения корректирующих кодов

Исходные комбинации задаются преподавателем для каждого обучаемого по табл. 2.

Для выполнения лабораторной работы необходимо:

1) заданную комбинацию закодировать в циклическом коде

- по формулам (5.3-5.4),

- по матрице (5.5), предварительно построив матрицу H;

2) проверить полученный циклический код;

3) в правильно закодированную комбинацию внести ошибку и по образующей матрице определить местоположение ошибки.

Таблица 5.4. Исходные кодовые последовательности

Задания	**Варианты**		
	1	**2**	**3**
1	110100	100011	1001
2	1101010	110000	1010
3	11010000	1000	11010
4	110101	100100	1011
5	1101011	11000	1100
6	1101000	1101	10100
7	11010011	1001	11011
8	110110	100100	1101
9	1101100	110010	1110
10	1100111	1110	10101
11	11010010	1010	11100
12	110111	100101	1111
13	1101101	110011	1001
14	1100110	1111	10110
15	11101001	1100	11101
16	111000	100101	1010
17	1101111	110100	1011
18	1100100	0100	10111
19	11010100	1001	11110
20	111001	100110	1100

21	1110000	110101	1101
22	1100101	0110	11000
23	11010101	1011	11111

Содержание отчета

Для указанного варианта приводятся исходная и закодированная комбинация. Показывается методика кодирования и декодирования комбинаций, а также способы обнаружения ошибок.

Делаются необходимые выводы.

Практическое занятие 1. **Количественная оценка информации в сообщении**

Контрольные вопросы

1. Поясните понятие первичного алфавита.
2. Поясните понятие вторичного алфавита.
3. В чем измеряется неопределенность, приходящаяся на символ первичного алфавита при использовании двоичных логарифмов?
4. В чем измеряется неопределенность, приходящаяся на символ первичного алфавита при использовании десятичных логарифмов?
5. В чем измеряется неопределенность, приходящаяся на символ первичного алфавита при использовании натуральных логарифмов?
6. Приведите формулу количества информации в k сообщениях алфавита m для случая равновероятных и взаимонезависимых символов.
7. Приведите формулу количества информации в k сообщениях алфавита m для случая неравновероятных символов алфавита.
8. Поясните понятие полная группа событий.
9. Поясните понятие максимальная энтропия системы.
10. Поясните понятие безусловная вероятность события.
11. Поясните понятие условная вероятность события.

12.Поясните понятие совместная вероятность событий.

Практическое занятие 2. **Определение условной энтропии**

Контрольные вопросы

1. Приведите формулу условной вероятности события.
2. Поясните понятие условной энтропии.
3. Поясните понятие безусловная вероятность события.
4. Поясните понятие условная вероятность события.
5. Поясните понятие совместная вероятность событий.
6. Поясните понятие общая условная энтропия сообщения относительно другого сообщения.
7. Поясните понятие потери информации в каналах с шумами.
8. Поясните понятие частная условная энтропия сообщения относительно другого сообщения.
9. Приведите пример канальной матрицы.

Практическое занятие 3. **Вычисление информационных потерь при передаче сообщений по каналу связи с шумами**

Контрольные вопросы

1. Поясните термин канальная матрица.
2. Что даст суммирование элементов канальной матрицы вида p(ai,bj) по строкам?
3. Что даст суммирование элементов канальной матрицы вида p(ai,bj) по столбцам?
4. Чему равна энтропия канала при высоком уровне помех, когда можно ожидать прихода любого символа?
5. Чему равно количество информация при высоком уровне помех, когда можно ожидать прихода любого символа?

6. Что нужно задать для полного описания канала связи с точки зрения информации?
7. Поясните понятие потери информации в каналах с шумами.
8. Поясните случай, когда значение информации канала имеет отрицательный знак.

Практическое занятие 4. **Вычисление скорости передачи информации и пропускной способности каналов связи**

Контрольные вопросы

1. Чем определяется скорость передачи информации в условиях отсутствия помех?
2. Поясните разницу в терминах скорость передачи информации и скорость передачи сигналов.
3. Приведите формулу для скорости передачи информации для сообщений из равновероятных и взаимонезависимых символов равной длительности.
4. Скорость передачи информации определяется относительно первичного или вторичного алфавита?
5. Скорость передачи сигналов определяется относительно первичного или вторичного алфавита?
6. Приведите формулу для скорости передачи информации для сообщений из неравновероятных символов равной длительности.
7. Дайте определение пропускной способности канала связи.
8. Как вычисляется пропускная способность канала связи при наличии помех?
9. Приведите свойства симметричного двоичного канала.

Практическое занятие 5. **Определение избыточности сообщений**

Контрольные вопросы

1. Чем определяется избыточность источника сообщений?
2. Что такое информационная избыточность?
3. Что такое избыточность за счет неравновероятного распределения символов в сообщении?
4. Что такое избыточность за счет статистической связи символов в сообщении?
5. Поясните понятие достаточная длина кодовой комбинации.
6. Поясните понятие избыточность от округления.
7. Поясните понятие абсолютная корректирующая избыточность.
8. Поясните понятие относительная корректирующая избыточность.
9. Поясните понятие оптимальный код с точки зрения избыточности.

Лабораторная работа 1**. Границы помехоустойчивости и их геометрическая интерпретация.**

Контрольные вопросы

1. Поясните понятие помехоустойчивость.
2. Поясните понятие граница помехоустойчивости.
3. Поясните понятие регулярная помеха.
4. Поясните понятие случайная помеха.
5. Поясните понятие аддитивная помеха.
6. Поясните понятие мультипликативная помеха.
7. Как связаны помехоустойчивость и надежность системы передачи информации.
8. Как связаны помехоустойчивость и эффективность системы передачи информации.
9. Поясните понятие статическая помехоустойчивость.

10. Поясните понятие динамическая помехоустойчивость.

Практическое занятие 6. **Определение линейного блокового кода**

Контрольные вопросы

1. Поясните понятие блокового кода.
2. Перечислите основные способы задания линейных блоковых кодов.
3. Назовите линейную операцию для двоичного блокового кода.
4. Поясните понятие порождающей матрицы линейного блокового кода.
5. Поясните понятие проверочной матрицы линейного блокового кода.
6. Приведите пример линейного блокового кода.
7. Каков результат произведения кодового слова на транспонированную проверочную матрицу?
8. Что такое граница Синглтона?
9. Что такое эквивалентные блоковые коды?
10. Как определить кодовое расстояние через проверочную матрицу?

Лабораторная работа 2. **Исследование кода Хэмминга**

Контрольные вопросы

1. Поясните суть кода Хэмминга.
2. Перечислите основные способы задания кода Хэмминга.
3. Назовите линейную операцию для кода Хэмминга.
4. Поясните понятие порождающей матрицы кода Хэмминга
5. Поясните понятие проверочной матрицы кода Хэмминга.
6. Чему равно кодовое расстояние кода Хэмминга?
7. Что такое синдром кода Хэмминга?

8. Можно ли увеличить проверочные способности кода Хэмминга и как это сделать?

9. Можно ли использовать проверочные уравнения кода Хэмминга для построения кодера?

10. Можно ли использовать синдромные уравнения кода Хэмминга для построения декодера?

Лабораторная работа 3. **Исследование циклических кодов**

Контрольные вопросы

1. Поясните понятие циклического кода.
2. Перечислите основные способы задания циклического кода.
3. Назовите линейную операцию для циклического кода.
4. Поясните понятие порождающей матрицы циклического кода.
5. Поясните понятие проверочной матрицы циклического кода.
6. Поясните понятие порождающего полинома циклического кода.
7. Назовите известные циклические коды.
8. Поясните суть кода БЧХ.
9. Поясните суть систематического кодирования циклическим кодом.
10. Поясните суть несистематического кодирования циклическим кодом.

Лабораторная работа 4. **Декодирование циклических кодов**

Контрольные вопросы

1. Поясните понятие проверочной матрицы циклического кода.
2. Перечислите основные способы задания циклического кода.
3. Поясните понятие порождающей матрицы циклического кода.
4. Что такое синдром систематического циклического кода?
5. Как реализуется декодирование систематического циклического кода?

6. Как реализуется декодирование несистематического циклического кода?
7. Чему равно произведение порождающей и единичной транспонированной матриц циклического кода?
8. Поясните понятие порождающего полинома циклического кода.
9. Поясните суть операции деления полинома x^n-1 на порождающий полином.

Лабораторная работа 5. **Декодирование сверточных кодов с мягкими и жесткими решениями**

Контрольные вопросы

1. Поясните понятие сверточного кода.
2. Поясните понятие древовидного кода.
3. Поясните понятие решетчатого кода.
4. Перечислите основные способы задания сверточного кода.
5. Назовите линейную операцию для сверточного кода.
6. Приведите общую схему нерекурсивного сверточного кода.
7. Чему равна скорость сверточного кода?
8. Чему равна избыточность сверточного кода?
9. Приведите формулы для памяти сверточного кода.
10. Поясните понятие рекуррентного кода.