Министерство образования Республики Беларусь
Учреждение образования
"Белорусский государственный университет
информатики и радиоэлектроники"

Кафедра сетей и устройств телекоммуникаций

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе
по дисциплине "Теория электросвязи"
для студентов специальности 45 01 03
"Телекоммуникационные системы"
дневной формы обучения

Минск 2004

УДК 621.391.2(075.8)
ББК 32.811.4 я 7
П 55

Составитель:
И.И. Астровский

П 55 **Помехоустойчивое** кодирование: Метод. указания к лаб. работе по дисц. "Теория электросвязи" для студ. спец. 45 01 03 "Телекоммуникационные системы" дневной формы обучения /Сост. И.И. Астровский. – Мн.: БГУИР, 2004. − 23 с.

Приведены задания и методические указания по выполнению лабораторной работы на IBM-совместимой ПЭВМ в диалоговом режиме.

Материалы рекомендуются для курсов, связанных с кодированием и цифровой обработкой сигналов.

УДК 621.391.2(075.8)
ББК 32.811.4 я 7

1. ЦЕЛЬ РАБОТЫ

Изучить принципы помехоустойчивой передачи информации. Исследовать принципы кодирования и декодирования линейных кодов: с проверкой на четность, «квазициклического» и циклического.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ И СООТНОШЕНИЯ

2.1. КОДИРОВАНИЕ

Кодированием называется представление различных сообщений в виде условных комбинаций, состоящих из определенного числа элементарных символов, причем каждому сообщению соответствует одна единственная условная комбинация (последовательность) символов (или цифр, или знаков, или импульсов).

Получатель по принятой кодовой комбинации (последовательности) восстанавливает переданную информацию. Операция, в результате действия которой осуществляется преобразование принятого кода в сообщение, называется **декодированием**.

Код строится из символов (элементов). Число различных символов (m) называется **основанием** кода. Код, состоящий из двух различных символов, называется кодом с основанием два (m = 2). Код, состоящий из трех символов, называется кодом с основанием три и т.д. Например, кодовая комбинация 10-1-10-11 состоит из 1, 0, -1, т.е. m = 3.

Число различных символов можно выбирать равным числу передаваемых знаков алфавита сообщения. Если число знаков достаточно велико (например, 32 буквы русского алфавита), то при передаче такого сообщения необходимо большое количество различных символов ($m \geq 32$). Техническая реализация такого кода встречает большие затруднения.

Поэтому наибольшее распространение получили коды, у которых основание значительно меньше числа букв алфавита сообщения. Тогда каждая буква алфавита передается не одним элементом кода, а их комбинацией. Число элементов или знаков комбинации кода называется значностью кода (n) или числом разрядов кода.

Коды, все комбинации которых имеют одинаковое число знаков, называются равномерными.

Для равномерного кода с основанием m и значностью n число кодовых комбинаций (M) определяется числом возможных комбинаций из m по n , т.е.

$$M=m^n,$$

где m - основание кода; n − значность кода; M − число комбинаций.

Например, для кода с m = 2, n = 3 существует восемь кодовых комбинаций ($M = 2^3$), состоящих из трех символов (000, 001, 010, 100, 101, 110, 111).

2.2. ПРИНЦИПЫ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Теория помехоустойчивого кодирования возникла в 50-е годы и наиболее полно разработана для двоичных кодов. В дальнейшем мы рассматриваем только двоичные коды (m = 2).

Помехоустойчивое кодирование является эффективным средством повышения достоверности передачи сообщений в каналах с помехами. Для этого применяют специальные коды, корректирующие ошибки. Код называется **корректирующим**, если он позволяет обнаруживать и исправлять ошибки в кодовой комбинации. В корректирующем коде должны содержаться дополнительные (избыточные) символы, предназначенные для корректирования ошибок. Чем больше избыточность кода (æ), тем выше его корректирующая способность:

$$æ = 1 - \frac{M_0}{M} = \frac{M - M_0}{M}. \qquad (1)$$

В некорректирующем коде число комбинаций M выбирается равным числу сообщений (алфавиту) источника M_0 и избыточность кода отсутствует (æ = 0).

Корректирующие коды строятся так, чтобы число комбинаций M превышало число сообщений источника M_0 ($M > M_0$). В этом случае лишь M_0 комбинаций используются для передачи информации. Они называются **разрешенными**. Остальные $M - M_0$ комбинации называются **запрещенными**. На приемном конце при декодировании известно, какие комбинации являются разрешенными, а какие − запрещенными. Поэтому если принята запрещенная комбинация, то ошибка может быть обнаружена и при определенных условиях исправлена.

Различие между комбинациями равномерного кода принято характеризовать кодовым расстоянием Хэмминга (d), равным наименьшему числу символов, которыми комбинации A_i и A_j отличаются одна от другой. Для любого кода $d \le n$. Кодовое расстояние обычно определяют количеством единиц в сумме этих комбинаций A_i и A_j по модулю два.

Например:

$$\begin{array}{lc} A_i & 1\ 0\ 1\ 0\ 0 \\ \oplus & \oplus \\ A_j & 0\ 0\ 1\ 1\ 1 \\ \hline D & 1\ 0\ 0\ 1\ 1 \end{array}$$

Сумма единиц в векторе D равна d = 3.

Чтобы в результате ошибки A_i преобразовалось в A_j, должно исказиться три символа. При искажении меньшего числа символов A_i перейдет в запрещенную комбинацию, и ошибка будет обнаружена. Ошибка всегда обнаруживается, если ее кратность g, т.е. число искаженных символов кодовой комбинации, меньше кодового расстояния:

$$g \leq d - 1. \qquad (2)$$

При $g > d$ ошибки тоже могут быть обнаружены, но имеется вероятность, что ошибочная комбинация совпадает с какой-либо разрешенной комбинацией. Минимальное кодовое расстояние, при котором обнаружатся любые одиночные ошибки, равняется $d = 2$.

Для того чтобы ошибка была исправлена, необходимо, чтобы ее кратность удовлетворяла неравенству

$$g \leq \frac{d-1}{2}. \qquad (3)$$

Минимальное кодовое расстояние, при котором возможно исправление любых одиночных ошибок, равняется трем, т.е. $d = 3$, так как

$$g \leq \frac{3-1}{2}.$$

Основными характеристиками корректирующего кода являются вероятность некорректированных ошибок Рош, избыточность æ и число символов n. Эти показатели позволяют понять, насколько удается повысить помехоустойчивость кода и какой ценой это достигается. Общая задача создания кода состоит в достижении наименьших значений Рош и æ .

Рассмотрение классификации корректирующих кодов не является нашей задачей, отметим только, что в лабораторной работе изучаются коды, относящиеся к линейным систематическим кодам.

Линейные р а з д е л и м ы е - это коды, в которых символы подразделяются на информационные и проверочные (контрольные). Информационные символы содержат передаваемую информацию. Контрольные символы являются избыточными и служат исключительно для коррекции ошибок. Число контрольных символов равно r, информационных – равно k. Их сумма определяет длину кодовых слов n = r+k. Особенность линейных кодов состоит в том, что контрольные символы образуются как линейные комбинации информационных символов.

Линейные коды разделяются на подклассы. Изучаемые коды относятся к **систематическим**. Все двоичные систематические коды являются групповыми и обладают тем свойством, что сумма по модулю два любой пары комбинаций снова дает комбинацию, принадлежащую этой группе. Из подклассов систематических кодов мы будем рассматривать коды с четным числом единиц и циклические коды. По причинам, о которых будет сказано несколько позже, мы рассмотрим и «квазициклические» коды.

2.3. КОД С ЧЕТНЫМ ЧИСЛОМ ЕДИНИЦ

Код с четным числом единиц (код с проверкой на четность) является одним из простейших систематических кодов. Каждая его комбинация содержит помимо информационных символов $\{a_1, a_2,..., a_k\}$ один контрольный символ $\{c\}$, равный 0 или I и выбираемый так, чтобы сумма единиц в комбинации всегда была четной. Следовательно, для любой кодовой комбинации сумма всех символов по модулю два будет равна нулю:

$$c \oplus \sum_{i=1}^{k} a_i = 0. \qquad (4)$$

Из формулы (4) можно выразить контрольный символ с через информационные:

$$c = \sum_{i=1}^{k} a_i. \qquad (5)$$

Например, для кода 111 $c = 1 \oplus 1 \oplus 1 = 1$. Закодированная последовательность будет 1111.

Кодер для кода с четным числом единиц $a_1, a_2,..., a_k$, с , работающий по алгоритму (5), отличается простотой и содержит сумматор по модулю два для получения контрольного символа с .

Декодер осуществляет проверку принятого кода $\{a_1^*, a_2^*,..., a_k^*, c^*\}$ на четность и работает по алгоритму

$$c^* \oplus \sum_{i=1}^{k} a_i^* = 0. \qquad (6)$$

Нарушение четности имеет место при появлении однократных, трехкратных и в общем случае ошибок нечетной кратности, что и дает возможность их обнаружить. Появление четных ошибок не нарушает четности символов в коде и нуля суммы (6), поэтому такие ошибки не обнаруживаются.

Декодер кода с четным числом единиц содержит в своем составе сумматор по модулю два. Если сигнал на выходе сумматора по модулю два равняется нулю, то это значит, что сигнал ошибки декодера будет равен нулю, а на выходе декодера появляется исходная кодовая последовательность. Если на выходе сумматора по модулю два будет единица, то сигнал ошибки будет равен единице и кодовая последовательность не поступит на выход декодера.

К достоинствам кода с четным числом единиц следует отнести простоту кодирующих и декодирующих устройств, а также малую избыточность кода. К недостаткам − низкую корректирующую способность кода.

2.4. ЦИКЛИЧЕСКИЕ КОДЫ

Недостатком большинства линейных кодов является сложность процедуры декодирования.

Упрощение алгоритмов и устройств декодирования возможно, если наложить на код кроме свойства линейности еще некоторые ограничения и использовать эти ограничения при декодировании. Таким ограничением может быть,

например, свойство цикличности. Линейные коды, удовлетворяющие этому ограничению, называются циклическими.

Циклическим кодом называется такой код, который вместе с каждым кодовым словом (a_0, a_1,…, a_{n-1}) содержит также и его циклическую перестановку (a_1, a_2,…, a_0). При таком определении для задания кода достаточно задать только одно кодовое слово. Остальные кодовые слова образуются из исходного путем циклического сдвига и сложением всех линейных комбинаций циклического сдвига.

Исходное кодовое слово принято задавать в виде порождающего полинома (генераторного полинома) $q(x)$ степени r. Например, $q(x)=1+x+x^3$ или в двоичной записи 1101.

Для того чтобы полином был генераторным полиномом циклического кода, необходимо, чтобы он был делителем полинома вида x^n+1, где n - значность кода. При делении полиномов действия производятся по правилам арифметики по модулю два, в которой вычитание равносильно сложению.

Для построения циклического кода необходимо знать разложение полинома x^n+1 на множители.

Если $n=2^m-1$, то многочлен x^n+1 представляется как произведение неприводимых многочленов степени не выше m. Каждый из этих многочленов, а также их произведения могут быть использованы как порождающие полиномы циклического кода.

Пример. Пусть $n=2^3-1=7$.

Полином x^7+1 разлагается на следующие неприводимые:

$x^7+1=(x+1)(x^3+x+1)(x^3+x^2+1)$, которые можно использовать как порождающие полиномы:

$q_1(x)=x+1$ порождает код (7, 6);

$q_2(x)=x^3+x+1$ порождает код (7, 4);

$q_3(x)=x^3+x^2+1$ порождает код (7, 4);

$q_1(x)\ q2(x)$ и $q_1(x)\ q3(x)$ порождают код (7, 3).

Порождающая матрица циклического кода имеет в качестве строк векторы $q(x)$, $xq(x)$,..., $x^{k-1}q(x)$:

$$G=\begin{bmatrix} q(x) \\ xq(x) \\ \dots \\ x^{k-1}q(x) \end{bmatrix}=\begin{bmatrix} q_0 & q_1 & \dots & q_r & 0 & \dots & 0 \\ 0 & q_0 & q_1 & \dots & q_r & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & q_0 & \dots & q_r \end{bmatrix},$$

где q_0, ..., q_r - коэффициенты генераторного полинома.

Проверочная матрица строится на основе полинома $h(x)=\dfrac{x^n+1}{q(x)}$ степени k и имеет вид

$$H = \begin{bmatrix} \overleftarrow{h(x)} \\ \overleftarrow{xh(x)} \\ \dots \\ \overleftarrow{x^{r-1}h(x)} \end{bmatrix} = \begin{bmatrix} 0 & \dots & \dots & 0 & h_k & \dots & h_2 & h_1 & h_0 \\ 0 & \dots & 0 & h_k & \dots & h_2 & h_1 & h_0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_k & \dots & h_2 & h_1 & h_0 & 0 & \dots & \dots & 0 \end{bmatrix},$$

где h_0, h_1, … - коэффициенты полинома $h(x)$, называемого проверочным полиномом.

При таком построении кода умножение сообщения на порождающую матрицу эквивалентно умножению сообщения, представленного в виде полинома, на генераторный полином. Действительно, пусть сообщение имеет вид $a(x)=a_0 + a_1x + a_2x^2 +\dots$, $a \in 0,1$.

Тогда

$$(a_0\ a_1\dots)\ G = [a_0q_0, (a_0q_1+ a_1q_0), (a_0q_2+ a_1q_1+ a_2q_0)\ \dots],$$

что эквивалентно умножению полиномов:

$$a(\mathrm{x})\ \mathrm{q}(\mathrm{x}) = a_0\mathrm{q}_0, (a_0q_1+ a_1\mathrm{q}_0)\ x+\dots$$

Из способа кодирования следует, что любое кодовое слово должно делиться на $q(x)$. Пусть, например, генераторный полином равен $q(x) = 1+x+x^3$. Этот полином является делителем полинома x^7+1, т.е. $n=7$, а $h(x) = 1+x+x^2+x^4$.

Пусть передается сообщение 1010, т.е. $v(x) = 1+x^2$. После кодирования получим следующее кодовое слово:

$$v(x)q(x) = (q+x^2)(1+x+x^3) = 1+x^2+x+2x^3+x^5 = 1+x+x^2+x^5,$$

т.е. 1110010. Аналогично можно найти другие кодовые слова. Например, сообщению 0001 соответствует код 0001101, сообщению 1111 - 1001011.

Рассмотренный способ кодирования не позволяет разделить информационные и проверочные символы, а код, который при этом получается, называется неразделимым кодом. Использование неразделимых кодов по многим причинам неудобно. Циклический неразделимый код можно сделать разделимым следующим образом. Допишем к информационной последовательности n-k нулей, что эквивалентно умножению ее полинома на x^{n-k}, после чего разделим на генераторный полином:

$$\frac{v(x)x^{n-k}}{q(x)} = c + \frac{R(x)}{q(x)},$$

где R - некоторый остаток.

Умножим обе части на $q(x)$ и перенесем остаток в левую часть:

$$v(x)x^{n-k} + R(x) = cq(x).$$

Так как с − целое, то $v(x)x^r + R(x)$ делится на $q(x)$ и, следовательно, является кодовым словом. Таким образом, для того чтобы закодировать сообщение, необходимо приписать к информационным символам остаток от деления $v(x)x^{n-k}/q(x)$. Можно показать, что при таком способе кодирования множество кодовых слов остается тем же самым, а изменяется только таблица соответствия сообщений и кодовых слов.

Пример

а) Сообщение $v(x) = 0001$.

$$
\begin{array}{llll}
x^6 & & & \\
\underline{x^6} & +x^4 & +x^3 & \\
 & x^4 & +x^3 & \\
 & \underline{x^4} & +x^2 & +x \\
 & & x^3 & +x^2 & +x \\
 & & \underline{x^3} & +x & +1 \\
 & & & x^2 & +1
\end{array}
\qquad
\begin{array}{|lll}
x^3 & +x & +1 \\
\hline
x^3 & +x & +1
\end{array}
$$

Остаток 101. Кодовое слово 1010001.

б) Сообщение $v(x) = 0010$.

Проверить самостоятельно. Остаток 111. Кодовое слово 1110010.

2.5. КОДИРУЮЩИЕ УСТРОЙСТВА ЦИКЛИЧЕСКИХ КОДОВ

При кодировании разделимым кодом существует два варианта построения кодирующего устройства: *к*- и *r*-разрядным регистром сдвига. Обычно используется схема с минимальным числом ячеек в зависимости от соотношения между *k* и *r*.

Рассмотрим основное проверочное соотношение

$$
(a_0,\ a_1,\ \ldots\ a_{n-1})
\begin{bmatrix}
\ldots & \ldots & \ldots & h_k & \ldots & h_2 & h_1 & h_0 \\
\ldots & \ldots & h_k & \ldots & h_2 & h_1 & h_0 & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
h_k & \ldots & h_2 & h_1 & h_0 & \ldots & \ldots & \ldots
\end{bmatrix}
= 0 .
$$

Поскольку $h_k = 1$, то из произведения вектора сообщения на первую строку матрицы получим выражение для первого проверочного символа

$$a_{n-k-1} = \sum_{i=0}^{k-1} a_{n-1-i} h_i .$$

Из произведения на вторую строку получим

$$a_{n-k-2} = \sum_{i=0}^{k-1} a_{n-2-i} h_i$$

и т.д.

$$a_{n-k-j} = \sum_{i=0}^{k-1} a_{n-1-j-i} h_i ,$$

т. е. можно найти *n-k* контрольных символов по *k* информационным символам. Схема, выполняющая эту операцию, приведена на рис. 1 и работает следующим образом: сначала ключ К находится в положении 1 и на вход подаются информационные символы. После *К* тактов информационные символы занимают все *К* ячеек регистра. Затем ключ переводится в положение 2 и регистр совершает еще *n* тактов, при каждом из которых на выходе появляется очередной символ кодового слова.

Уже при первом из этих n тактов в первой ячейке формируется контрольный символ. После n-k тактов весь кодовый вектор сформировался, n-k символов выданы на выход, а остальные k символов находятся в регистре. Теперь ключ возвращается в положение 1 и в регистр вводится k информационных символов следующего вектора, а оставшиеся в регистре k символов предыдущего вектора выводятся наружу.

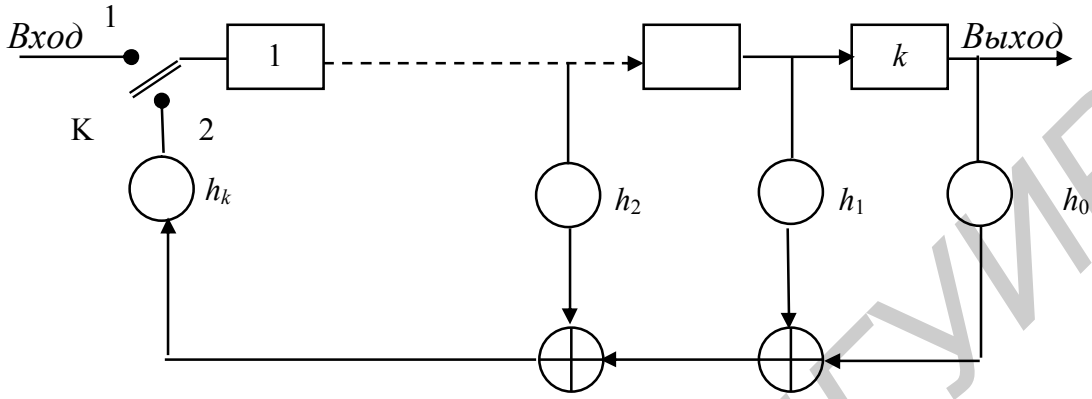

Рис.1 Кодирующее устройство с k-разрядным регистром

Второй вариант кодирующего устройства содержит (n-k)-разрядный регистр сдвига и определяет остаток от деления $v(x)x^n$ на генераторный полином $q(x)$ в соответствии с правилами образования кода.

Схема такого кодирующего устройства изображена на рис. 2 и работает следующим образом: вначале ключ $К_1$ находится в положении 1, а ключ $К_2$ замкнут. Информационные символы, подаваемые на вход через ключ $К_1$, поступают на выход, а через ключ $К_2$ – в кодирующее устройство, где через k шагов образуется n-k контрольных символов. После этого ключ $К_1$ перекладывается в положение 2, а ключ $К_2$ размыкается. Затем регистр делает еще n-k тактов, выдавая контрольные символы на выход.

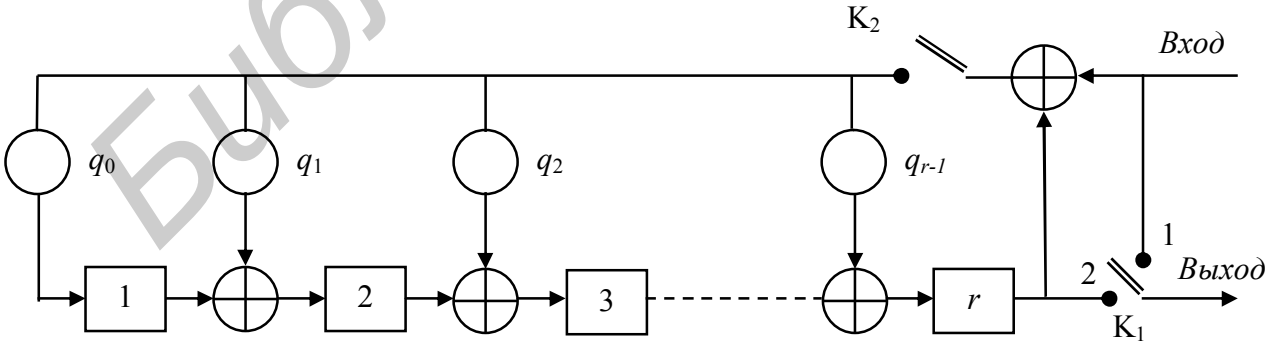

Рис.2. Кодирующее устройство с r - разрядным регистром

2.6. ДЕКОДИРОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ ПО СИНДРОМУ

Ошибки, возникшие при передаче, могут быть исправлены, если каждому вектору ошибки сопоставить свой синдром, однако декодирующее устройство получается довольно сложным.

При использовании циклических кодов декодирующее устройство может быть упрощено за счет перехода к последовательному режиму работы. Общая схема декодирующего устройства показана на рис. 3,а.

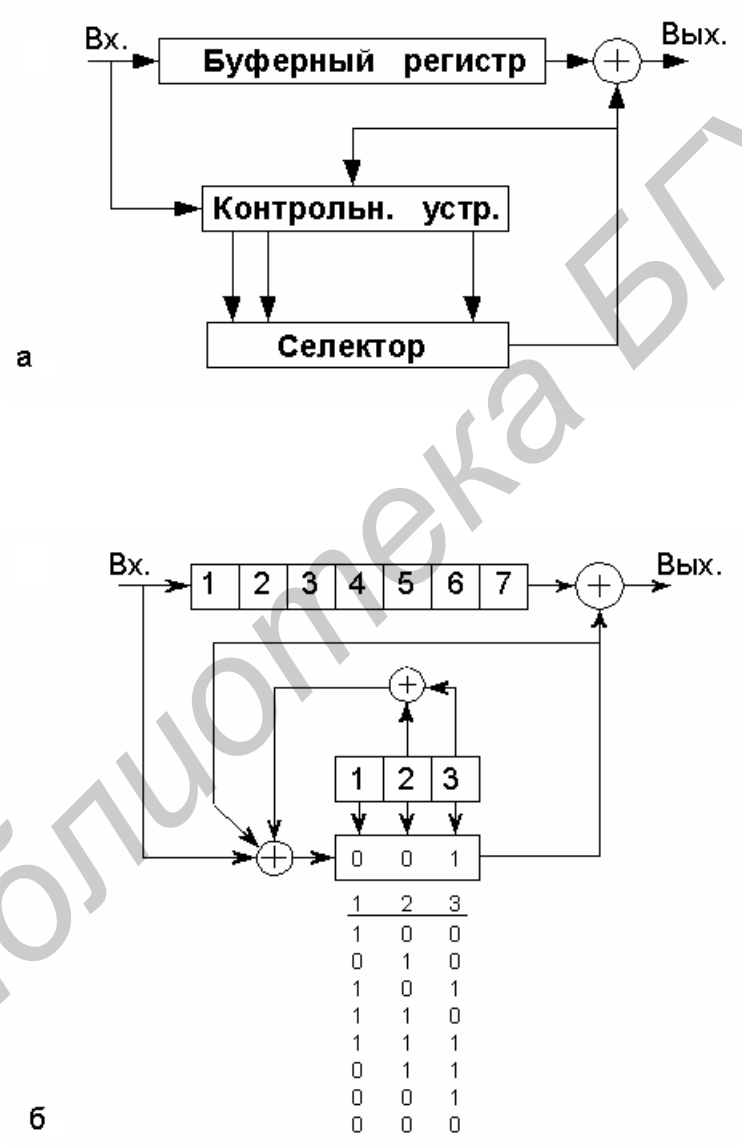

Рис. 3. Декодирование циклических кодов

Последовательность y, поступающая из приемника, записывается в n-разрядный буферный регистр, так что через n тактов все слово оказывается записанным в регистр. Одновременно последовательность y поступает в контрольное устройство, которое производит вычисление синдрома yH^T. Ранее

было доказано, что все слова циклического кода должны делиться на порождающий полином $q(x)$. Если принятое слово принадлежит коду, то остаток от его деления на полином $q(x)$ равен нулю, ненулевой остаток свидетельствует об ошибке. По виду остатка можно определить ошибку.

Контрольное устройство совместно с селектором производит эти операции. В контрольном устройстве производится вычисление остатка от деления $y(x)$ на $q(x)$. Селектор анализирует полученный остаток и выдает исправляющий сигнал в тот момент, когда ошибочный символ покидает буферный регистр; одновременно в контрольное устройство выдается сигнал, обозначающий, что осталась некоторая более простая комбинация ошибок. Если после $2n$ сдвигов, т.е. когда последний символ покидает буферный регистр, состояние контрольного устройства будет ненулевым, это означает, что произошла некорректируемая ошибка.

Работу декодирующего устройства проследим на следующем примере. Пусть принимается информация, закодированная циклическим кодом (7;4), имеющим следующий генераторный полином:

$$q(x) = 1 + x + x^3 .$$

Схема декодирующего устройства показана на рис.3,б и состоит из семиразрядного буферного регистра, трехразрядного контрольного устройства и селектора, настроенного на комбинацию 001. Единичный сигнал на выходе селектора появляется только при этой комбинации в ячейках контрольного устройства. Пусть передается кодовая комбинация, состоящая из одних нулей, т.е. 0000000, но в результате ошибки в канале связи принимается комбинация 0000001. Состояния ячеек контрольного устройства в последовательные моменты времени, соответствующие этой комбинации, показаны на рис.3,б.

Через семь тактов весь принятый вектор будет записан в буферном регистре, причем искаженный первый символ будет записан в седьмую ячейку буферного регистра. При этом в контрольном устройстве находится комбинация 001. На восьмом такте искаженный символ покидает буферный регистр. Одновременно с этим с селектора выдается единичный сигнал и происходит исправление ошибки, а в контрольном устройстве остается комбинация 000. Можно доказать, что схема будет работать аналогичным образом при любой одиночной ошибке в принятом кодовом слове и позволяет исправлять все одиночные ошибки.

В общем случае декодирующее устройство получается значительно более сложным, чем в рассмотренном примере, так как с увеличением кратности исправляемых ошибок число селектируемых комбинаций возрастает. В таблице основных показателей декодирующих устройств приведено в качестве примера количество селектируемых комбинаций для кода длиной $n = 63$.

Из этой таблицы видно, что уже при исправлении двойных ошибок сложность селектора превышает сложность контрольного устройства. Поэтому рассмотренное декодирующее устройство находит в основном применение для

исправления ошибок малой кратности (первой и второй) и для обнаружения ошибок.

Основные показатели декодирующих устройств

Показатель	Количество		
Кратность исправляемых ошибок	1	2	3
Число селектируемых комбинаций	1	63	1955
Количество элементов задержки в контрольном устройстве	6	12	18

2.7. МАЖОРИТАРНОЕ ДЕКОДИРОВАНИЕ

Рассмотрим основное проверочное соотношение

$$(a_0, a_1, \ldots, a_{n-1})H^T = 0,$$

где

$$\mathrm{H} = \begin{bmatrix} h_{00} & h_{01} & \ldots & h_{0,n-1} \\ \ldots & \ldots & \ldots & \ldots \\ h_{r-1,0} & h_{r-1,1} & \ldots & h_{r-1,n-1} \end{bmatrix}$$

проверочная матрица.

Эта запись сводится к системе уравнений:

$$a_0 h_{0,0} + a_1 h_{0,1} + \ldots + a_{n-1} h_{0,\,n-1} = 0,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_0 h_{r-1,\,0} + a_1 h_{r-1,\,1} + \ldots + a_{n-1}\, h_{r-1,\,n-1} = 0.$$

Выберем из них те, для которых $h_{i\,0} \neq 0$. Тогда

$$\begin{aligned} a_0 &= a_1 h_{i_1 1} + a_2 h_{i_1 2} + \ldots + a_{n-1} h_{i_1 n-1} \\ a_0 &= a_1 h_{i_2 1} + a_2 h_{i_2 2} + \ldots + a_{n-1} h_{i_2 n-1} \end{aligned},$$

и т. д.

Теперь символ a_0 можно определить по принципу большинства, полагая $a_0 = 0$, если правая часть большинства уравнений последней системы равна нулю, и $a_0 = 1$ в противном случае. Такую же процедуру можно проделать для символов $a_1, a_2, \ldots, a_{n-1}$.

В случае если код циклический, система проверочных уравнений для символов $a_1, a_2, \ldots, a_{n-1}$ получается из системы для символа a_0 циклическим сдвигом.

Для декодирования символа a_i достаточно произвести циклическую перестановку кодового слова на i позиций и использовать ту же самую систему проверок.

Если код не является циклическим, то для декодирования каждого символа должна быть найдена своя система проверок.

Система проверок может быть разделенной, λ-связанной и квазиразделенной.

Разделенные проверки. Система проверок называется разделенной, если:

1. Некоторый символ, например a_j, входит в каждую контрольную проверку.

2. Любой другой символ a_i, $i \neq j$ входит не более чем в одну контрольную проверку.

Каждая проверка системы разделенных проверок позволяет представить a_j в виде линейной комбинации символов, которые не входят более ни в одну из проверок. Поэтому одиночное искажение кодового слова может нарушить только одну проверку, в которую входит искаженный символ. Две ошибки могут нарушить две проверки и т.д. Если искажена одна проверка, то для принятия решения по большинству необходимо не менее двух правильных проверок. Если искажено две проверки, то необходимо, чтобы было три правильных и т.д. Если искажено t символов, то необходимо, чтобы $t + 1$ проверки были правильными. Поэтому для исправления t ошибок следует использовать $2t + 1$ проверок.

Пример. Рассмотрим код (7;3) с проверочной матрицей

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Согласно основному проверочному соотношению, имеем

$$a_0 + a_1 + a_3 = 0,$$
$$a_1 + a_2 + a_4 = 0,$$
$$a_0 + a_1 + a_2 + a_5 = 0,$$
$$a_0 + a_2 + a_6 = 0.$$

Оставляя в левой части только символ a_0 и складывая второе и третье равенства, получим

$$a_0 = a_1 + a_3,$$
$$a_0 = a_4 + a_5,$$
$$a_0 = a_2 + a_6.$$

Кроме того, всегда справедливо соотношение $a_0 = a_0$. Искажение любого символа нарушает не более одного проверочного соотношения, следовательно, символ можно определить, применяя решение по большинству. Случай, когда два выражения дают $a_0 = 1$, а два другие $a_0 = 0$, соответствует обнаружению ошибок. Остальные символы определяются после циклических сдвигов принятой последовательности, например,

$$a_1 = a_0 + a_3,$$
$$a_1 = a_2 + a_4,$$
$$a_1 = a_5 + a_6.$$

Схема декодирования рассмотренного кода приведена на рис.4 и работает следующим образом.

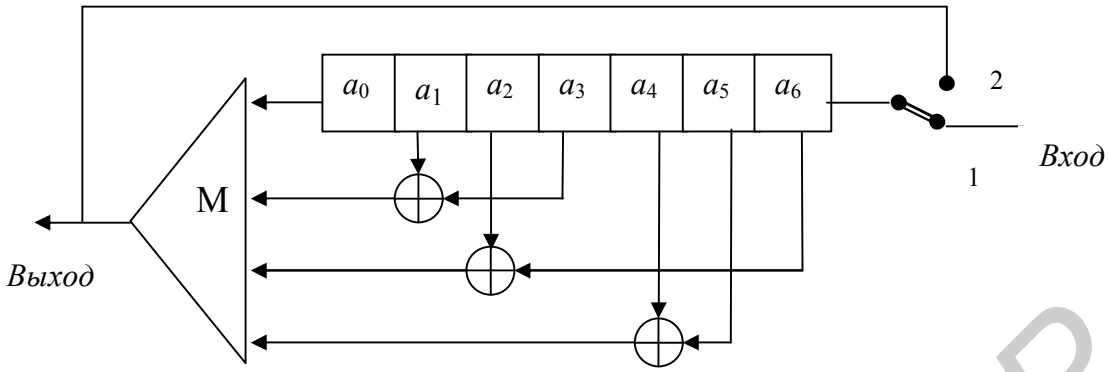

Рис. 4 . Мажоритарное декодирование кода с разделенными проверками

При приеме сообщения ключ находится в положении 1 и принятый вектор символ за символом записывается в регистр. После того как символы будут записаны, ключ переводится в положение 2 и начинается процесс декодирования. На первом шаге определяется символ a_0, а затем a_1 и т.д.

Связанные проверки. Системой λ-связанных проверок называется множество контрольных проверок, которые удовлетворяют следующим условиям:

1. В каждую проверку входит один и тот же символ, например a_i.

2. Любой символ a_j входит не более чем в λ проверок.

3. Существует символ a_j, $i \neq j$, который входит точно в λ проверок. Число λ называется показателем связности.

Если символ a_j искажен и входит в λ проверок, то будут нарушены λ проверочных уравнений. Поэтому для исправления одиночных ошибок следует использовать 2λ +1 уравнений, а для исправления t-кратной ошибки $2\lambda t$ + 1 уравнений.

Пример. Рассмотрим код с проверочной матрицей

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Система проверочных уравнений имеет следующий вид:

$a_0 = a_1 + a_2 + a_4$ (первая строка);

$a_0 = a_3 + a_4 + a_5$ (сумма первой и второй строк);

$a_0 = a_2 + a_3 + a_6$ (третья строка);

$a_0 = a_1 + a_5 + a_6$ (сумма второй и третьей строк);

$a_0 = a_0$.

Показатель связности λ = 2. Схема декодирования приведена на рис.5 и работает аналогично схеме рис.4.

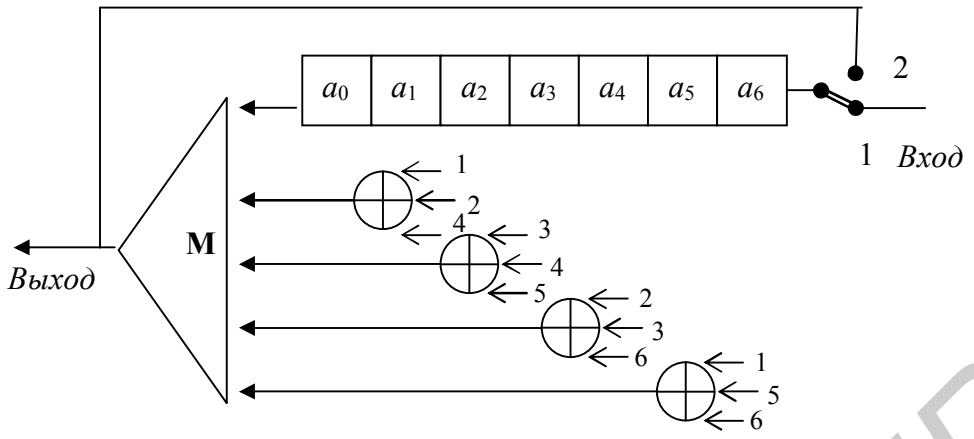

Рис. 5. Декодирование кода с λ-связанными проверками

Квазиразделенные проверки. Система проверок называется квазиразделенной, если она разделена относительно некоторой суммы символов.

Пример. Рассмотрим код с проверочной матрицей

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

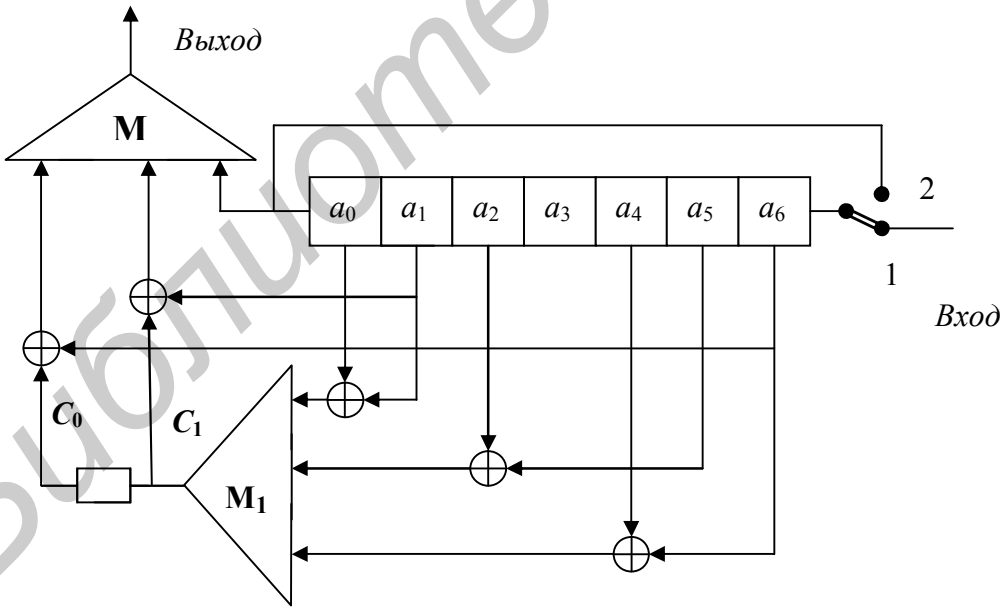

Рис. 6. Декодирование кода с квазиразделенными проверками

Для суммы $a_0 + a_1$ можно записать:

$a_0 + a_1 = a_2 + a_5$,

$a_0 + a_1 = a_4 + a_6$,

$a_0 + a_4 = a_0 + a_1$.

Предположим, что вычислено значение $a_0 + a_1 = C_0$, после чего в регистре, где хранятся a_0 и a_1, произведен циклический сдвиг, при котором вычисляется $a_1 + a_2 = C_1$. Тогда для a_1 получим

$a_1 = a_0 + C_0$,

$a_1 = a_2 + C_1$,

$a_1 = a_1$.

Декодирующее устройство приведено на рис.6. После ввода сигнала в регистр входной ключ переводится в верхнее положение и делается сдвиг содержимого регистра. В этом такте на выходе появляется скорректированный символ a_1. В следующем такте декодируется символ a_2 и т.д. Символ a_0 декодируется последним.

2.8. «КВАЗИЦИКЛИЧЕСКИЕ» КОДЫ

В этом разделе рассмотрен пример, который показывает, сколь внимательно следует соблюдать теоретические положения при построении кодов.

Данная лабораторная работа разработана на основе лабораторной работы №3 "Помехоустойчивое кодирование" [1]. Выполнение работы предусматривалось на специальном макете. Макет был разработан в 1986 г. Ленинградским электротехническим институтом связи им. проф. М.А.Бонч-Бруевича, приобретался и, по нашим сведениям, используется до настоящего времени различными учебными заведениями на территории бывшего СССР, в том числе и учебными заведениями Республики Беларусь. Безусловно, возможности макета по сравнению с ПЭВМ весьма скромны.

В описании лабораторной работы №3 приводятся "Основные теоретические сведения и соотношения", которые можно использовать и при выполнении данной лабораторной работы.

В [1] приводится пример построения разделимого якобы циклического кода (10,5) с порождающим полиномом $G(z)=z^5+z^4+z^2+1$. В этом коде значность кода n=10, число информационных символов k=5; число проверочных символов r=5. Очевидно, что $n \neq 2^m-1$. Ближайшие $n \in \{7,15,31,...\}$. **Заметим, что при n=15 построить циклический код можно.**

Несмотря на то что алгоритм построения кода полностью соответствует алгоритму построения разделимого циклического кода, получить циклический код с указанным порождающим полиномом при n=10 нельзя. В [1] не учтено важнейшее условие построения циклического кода. Нельзя и при n=7 (поясните, почему). **Для того чтобы полином был генераторным полиномом циклического кода, необходимо, чтобы он был делителем полинома вида x^n+1, где n - значность кода**.

Полином пятой степени G(z) мог бы быть порождающим для кода с $n=2^m-1 = 2^5-1 = 31$, если бы он делил полином $z^{31}+1$. Тогда бы из множества всех слов полученного циклического кода можно было бы выбрать подмножество кодовых слов с 21 нулём впереди и при таком условии объявить подмножество с n=10 принадлежащим циклическому коду. Заметим, что такое возможно сделать при n=15, если выбрать слова с пятью нулями впереди.

Безусловно, декодирующее устройство такого кода строилось бы по правилам для циклического кода (31,5) или (15,5) и было бы достаточно сложным. Но указанный полином G(z) не делит ни полином $z^{31}+1$, ни полином $z^{10}+1$ без остатка. Это легко проверяется в системе MATLAB, и проверка входит в задание на выполнение данной лабораторной работы.

Следовательно, полином $G(z)=z^5+z^4+z^2+1$ при условиях, указанных в [1] не может быть порождающим полиномом для циклического кода и к нему не может быть применена схема для декодирования циклического кода, основным достоинством которой является относительная простота. Нарушив основное правило построения циклического кода, авторы работы [1], естественно, не сообщая об этом, получили линейный код с порождающей матрицей в приведённо-ступенчатой форме $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$,

который является линейным групповым, **но не циклическим**.

Для декодирования этого кода могут быть применены общие методы декодирования линейных кодов, и в частности, метод мажоритарного декодирования. По приведённой порождающей матрице G для каждого информационного символа можно построить систему из семи трехсвязных проверочных уравнений и таким образом получить возможность исправлять одну ошибку в принятом кодовом слове при наличии пяти проверочных символов.

Приведём системы уравнений для определения первого и второго символов кодового слова источника информации:

$$\begin{cases} s(1) = b(9), \\ s(1) = b(7) + b(6) + b(3), \\ s(1) = b(7) + b(6) + b(5) + b(0), \\ s(1) = b(4) + b(2), \\ s(1) = b(8) + b(3) + b(1), \\ s(1) = b(8) + b(7) + b(5) + b(3) + b(2), \\ s(1) = b(8) + b(5) + b(1) + b(0). \end{cases}$$

Здесь символы s(i) – символы информационного вектора, читаемого слева направо; b(i) – символы принятого кодового слова, читаемые как коэффициенты полинома с возрастанием степеней справа налево (в связи с принятыми соглашениями при программировании объектов типа полином).

$$\begin{cases} s(2) = b(8), \\ s(2) = b(9) + b(6) + b(5) + b(4), \\ s(2) = b(6) + b(5) + b(2), \\ s(2) = b(7) + b(6) + b(1), \\ s(2) = b(9) + b(3) + b(1), \\ s(2) = b(4) + b(3) + b(2) + b(1), \\ s(2) = b(9) + b(7) + b(5) + b(3) + b(2). \end{cases}$$

Аналогичным образом составляются проверочные уравнения и для последующих символов кодовых слов источника информации. Этими уравнениями определяются структура и сложность декодирующего устройства.

Для сравнения отметим, что в лабораторной работе рассматривается раздел с применением циклического кода (7,4) с порождающим полиномом z^3+z+1. Здесь при наличии трех проверочных символов также исправляются одиночные ошибки. Декодер использует только одну систему из пяти двухсвязных проверочных уравнений для определения одного символа. Остальные символы получаются путем циклического сдвига принятого декодером слова, записанного в регистр сдвига, и применением той же системы уравнений. Относительная простота такого декодера очевидна. Он решает только одну систему уравнений:

$$\begin{cases} s(i) = b(i+5), \\ s(i) = b(i+4) + b(i+3) + b(i+1), \\ s(i) = b(i+4) + b(i+2) + b(i-1), \\ s(i) = b(i+2) + b(i+1) + b(i), \\ s(i) = b(i+3) + b(i) + b(i-1), \end{cases}$$

где $1 \le i \le 4$.

В противоположность действительно циклическим кодам код, описанный в [1], назван в рассматриваемой лабораторной работе "**квазициклическим**".

3. ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ

3.1. Изучите:

3.1.1. Принципы передачи информации по линиям связи.

3.1.2. Вопросы экономного и помехоустойчивого кодирования.

3.1.3. Блочное кодирование.

3.1.4. Какие коды называются линейными, способы их задания и декодирования.

3.1.5. Коды с проверкой на четность, способы их задания и декодирования.

3.1.6. Циклические коды, способы их задания и декодирования.

3.2. Ответьте на вопросы:

3.2.1. По каким параметрам производится классификация кодов?

3.2.2. Как задаётся код с проверкой на четность? Линейный? Неразделимый циклический? Разделимый циклический код?

3.2.3. Как оценивается отношение сигнал/помеха на входе и выходе устройств обработки?

3.2.4. Положим, что источнику сообщений выделяется 1000 отсчетов дискретного времени для передачи пакета информации. Варьируя число информационных символов в пакете, источник сообщений изменяет количество отсчетов на символ и тем самым изменяет энергию передаваемых элементов сигнала. Учитывая число проверочных символов, оцените энергию передаваемых в линию связи элементов сигнала в виде импульсов с одиночной амплитудой для кодов (6,5), (10,5), (7,4), задаваясь числом информационных символов в пакете 8. Затем задайтесь числом 17.

3.2.5. Каким образом осуществляется обнаружение и исправление ошибок?

3.2.6. Что такое мажоритарное декодирование? Как оно осуществляется для линейных кодов? Для циклических? В чем сходство и различие?

Закрепите на практике технику сложения, умножения и деления бинарных полиномов.

4. ЛАБОРАТОРНОЕ ЗАДАНИЕ

Начать выполнение лабораторной работы следует с раздела главного меню "Предварительное задание". ПЭВМ поможет выяснить, сколь успешно Вы справились с предварительным заданием, работая без ПЭВМ.

Наибольшим приоритетом обладает задание преподавателя, затем ПЭВМ (см. главное меню, а также задания ПЭВМ по ходу выполнения лабораторной работы).

Не задерживайтесь на выполнении пунктов, сущность которых уже выяснена или не до конца ясна. При наличии времени у вас будет возможность возвратиться в любой пункт задания и с учетом накопленного опыта более успешно справляться с возникающими затруднениями.

5. СОДЕРЖАНИЕ ОТЧЕТА

5.1. Решение задач предварительного задания.

5.2. Результаты выполнения лабораторного задания.

5.3. Анализ результатов и выводы.

ЛИТЕРАТУРА

1. Лабораторные работы по курсу "Теория передачи сигналов" для студентов 4-го курса /Под ред. А.С.Сухорукова.−М.: ВЗЭИС, 1985.

2. Лосев В.В. Помехоустойчивое кодирование в радиотехнических системах передачи информации. Ч.1: Линейные коды. Ч.2: Циклические коды. − Мн.: МРТИ,1984.

3. Клюев Л.Л. Теория электрической связи.−Мн.: Дизайн ПРО, 1998.

4. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. − М.: Мир, 1976.

5. Касами Т., Токура Н., Ивадари Е., Иногаки Я. Теория кодирования. − М.: Мир, 1978.

6. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки. − М.: Связь, 1979.

7. Колесник В.Д., Мирончиков Е. Т. Декодирование циклических кодов. − М.: Связь, 1968.

8. Финк Л. М. Теория передачи дискретных сообщений. − М.: Сов. радио, 1970.

Св.план 2004, поз.104

Учебное издание

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе
по дисциплине "Теория электросвязи"
для студентов специальности 45 01 03
"Телекоммуникационные системы"
дневной формы обучения

Составитель:
Астровский Иван Иванович

Редактор Т.А. Лейко
Корректор Е.Н.Батурчик

Подписано в печать 30.01.2004.	Формат 60x84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 1,51.
Уч.−изд. л. 1,0.	Тираж 70 экз.	Заказ 527.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Лицензия ЛП №156 от 30.12.2002.
Лицензия ЛВ №509 от 03.08.2001.
220013, Минск, П.Бровки, 6