4770

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ

Методические указания к лабораторным работам

Рязань 2014

УДК 621.396.21

Кодирование и декодирование циклических кодов: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост.: В.В. Езерский, А.В. Егоров.- Рязань: РГРТУ, 2014. - 28 с.

Изложены основные понятия и определения алгебраической теории кодирования и показано её применение для построения циклических кодов. Рассмотрены принципы построения основных элементов кодирующих и декодирующих устройств, а также основы схемной реализации таких устройств. Приведены необходимые сведения о лабораторном макете, индивидуальные задания и программы двух лабораторных работ.

Предназначены для студентов дневного факультета специальностей 210303 "Многоканальные телекоммуникационные системы" и 210304 "Системы связи с подвижными объектами".

Табл.6. Ил. 11. Библиогр.: 3 назв.

Избыточное кодирование, циклические коды, образующий многочлен, кодирование, декодирование, делительные устройства, кодирующие устройства, декодирующие устройства

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра радиоуправления и связи Рязанского государственного радиотехнического университета (зав. кафедрой д-р техн. наук, проф. С.Н. Кириллов)

1. ЦЕЛЬ РАБОТЫ

Целью работы является изучение принципов кодирования и декодирования циклических кодов, а также знакомство с техническими методами и устройствами, реализующими эти принципы.

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Общая идея избыточного кодирования заключается во внесении в передаваемые сообщения избыточной информации по правилам, которые известны на передающей и приёмной сторонах.

Циклические коды - наиболее распространенный тип избыточных кодов, используемый для повышения достоверности передачи дискретной информации, по линиям связи. Циклические коды имеют блочную структуру и относятся к систематическим разделимым кодам. Каждый блок передаваемой информации состоит из информационной и проверочной части. Проверочные символы блока являются линейной комбинацией информационных символов того же блока. Число элементов в блоке (n) в информационной (k) и проверочной (n-k) частях фиксировано и известно заранее. Наличие проверочной части позволяет на приемной стороне линии связи обнаружить ошибку и даже исправить ее.

Пусть a_i, $i=\{1,2...,k\}$; b_j, $j=\{1,...,n$-$k\}$ - соответственно символы информационной и проверочной частей блока. Тогда кодовая комбинация систематического кода может быть представлена в виде:

$a_1, a_2..., a_k, b_1, b_2..., b_{n-k}$

где $a_1, a_2,..., a_k$ - информационная часть, $b_1, b_2,..., b_{n-k}$ - проверочная часть. Проверочный символ b_j определяется линейным соотношением:

$$b_j = \sum_{i=1}^{k} c_{ij}a_i = c_{1j}a_1 \oplus c_{2j}a_2 \oplus ... \oplus c_{kj}a_k \ ;$$

здесь знак $\oplus$ означает сложение по модулю два; $c_{ij} \in \{1,0\}$ - производящий вектор для j-го проверочного символа. Для полного определения систематического кода необходимо располагать (n-k) производящими векторами, образующими производящую матрицу $\|c_{ij}\|$. Каждая строка этой матрицы есть соответствующий производящий вектор.

Все строки производящей матрицы могут быть получены циклическим сдвигом одной из них, называемой образующей данного кода. Основная проблема при разработке экономичных и удобных для реализации методов кодирования - нахождение производящей матрицы или образующего кода. Решение этой задачи требует знания специальной алгебраической теории кодирования. Некоторые сведения из этой теории приводятся ниже.

2.1. Основные понятия и определения алгебраической теории кодирования

1. n-разрядные комбинации циклического кода представляются в виде многочлена фиктивной переменной x:

$$G(x)=g_{n-1}x^{n-1}+g_{n-2}x^{n-2}+...+g_1x+g_0 .$$

В случае двоичного кода коэффициенты этого полинома могут принимать только два значения: 0 и 1. В такой записи, например, восьмиэлементная комбинация 10011101 представляется в виде полинома 7-й степени:

$$G(x)=x^7+x^4+x^3+x^2+1 .$$

Таким образом, многочлен $G(x)$ – это условная форма представления кодограммы, в которой все символы кода представлены коэффициентами многочлена, причем старшим символам кода соответствуют старшие коэффициенты многочлена.

2. Многочлены можно складывать, делить и умножать по определенным правилам.

Пусть $F(x)=f_{m-1}x^{m-1}+...+f_0$ - некоторый многочлен степени m-1.

Сложение многочленов $G(x)$ и $F(x)$ осуществляется по обычным правилам сложения алгебраических многочленов, однако подобные члены приводятся по модулю два.

Деление многочлена $G(x)$ на $F(x)$ определяется алгоритмом Эвклида

$G(x)=F(x)h(x)+r(x)$;

где $h(x)$ – частное от деления; $r(x)$ –остаток.

Простое произведение многочленов находится по обычным правилам умножения алгебраических многочленов с приведением подобных членов по модулю два.

Произведение двух многочленов $G(x)$ и $F(x)$ по модулю полинома $P(x)$ - это остаток $r(x)$ от деления произведения $G(x)F(x)$ на полином $P(x)$. Символически эта операция обозначается так:

$R_{P(x)}=[G(x)F(x)]=r(x)$.

k-й степенью многочлена $G(x)$ по модулю полинома $P(x)$ называется остаток от деления k-кратного простого произведения $G(x)G(x)$ на полином $P(x)$: $[G(x)]^k=R_{P(x)}[G(x)G(x)..G(x)]$.

3. Полином $P(x)$ называется неприводимым, если разложение его на множители вида $P(x)=P_1(x)P_2(x)$ невозможно в классе многочленов с коэффициентами 0 и 1. Другими словами, $P(x)$ является неприводимым, если уравнение $P(x)=0$ не имеет корней, равных 0 или 1.

4. Множество многочленов степени не выше n-1 называется конечным полем или полем Галуа. Поле многочленов замкнуто относительно операции сложения и умножения по модулю неприводимого

многочлена в том смысле, что применение упомянутых операций к двум элементам поля дает результат, также принадлежащий полю. Обозначение поля - GF(2^n), где 2 - порядок поля, n - значность кода.

Таким образом, поле Галуа GF(2^n), элементами которого являются полиномы степени не выше n-1, описывает все кодограммы кода значности n. Для образования избыточного кода, способного обнаруживать и исправлять ошибки, возникающие при передаче, необходимо сократить число используемых кодограмм, сохранив значность кода. Эти кодограммы называются разрешенными, их число меньше 2^n. Остальные кодограммы называют запрещёнными. Выделение разрешенных кодовых комбинаций можно осуществить разными способами. При этом образованный избыточный код обладает разной корректирующей способностью. Корректирующая способность кода тем выше, чем больше кодовое расстояние d.

Кодовым расстоянием называют минимальное количество разрядов кода, в которых одна кодовая комбинация отличается от другой при их попарном, поразрядном сравнении. В случае двоичного кода кодовое расстояние находят путем суммирования по mod2 двух кодовых комбинаций и подсчета числа получившихся единиц.

Для **обнаружения** ошибок необходимо выполнить условие: $d\geq t_0+1$, где t_0 - кратность обнаруживаемых ошибок.

Для **исправления** ошибок необходимо, чтобы расстояние от принимаемой с ошибками запрещенной комбинации до переданной было меньше, чем до любой другой разрешенной. Это возможно, если: $d\geq t_u+1$, где t_u - кратность исправляемых ошибок. В этих выражениях t_0 и t_u дают число гарантированно обнаруживаемых ошибок.

Один из эффективных способов избыточного кодирования состоит в следующем. Фиксируется некоторый многочлен $g(x)$ из поля GF(2^n), и выделяются те многочлены, которые делятся на многочлен $g(x)$ без остатка. Выделенное таким образом подмножество поля называется идеалом, а $g(x)$ – порождающим многочленом идеала.

Количество различных элементов в идеале и их свойства определяются видом порождающего многочлена. Построенный таким образом код называется полиномиальным.

5. Циклическим называется такой полиномиальный код, который обладает следующим свойством: если комбинация $A=(a_0,a_1,..,a_{n-1})$ - разрешённая комбинация циклического кода, то и комбинация $\overline{A}=(a_{n-1},a_0,..,a_{n-2})$ - также разрешённая комбинация этого кода. Представив комбинации A и $\overline{A}$ в виде многочленов $A(x)$ и

$\overline{A}(x)$, нетрудно установить связь между ними в следующем виде (не забывая о правилах приведения подобных членов по модулю два):

$$\overline{A} = xA(x) + a_{n-1}x^n + a_{n-1}) = xA(x) + a_{n-1}(x^n + 1). \qquad (1)$$

Пусть $A(x)$ принадлежит идеалу с порождающим многочленом $g(x)$. Для того чтобы циклический сдвиг этой комбинации также принадлежал этому идеалу, необходимо деление многочлена $\overline{A}(x)$ без остатка на $g(x)$. Из выражения (1) следует, что это условие будет выполнено в том случае, если двучлен x^n+1 делится на $g(x)$ без остатка.

$$R_{g(x)}\left[x^n + 1\right] = 0. \qquad (2)$$

Полином $g(x)$, являющийся делителем двучлена x^n+1, называется порождающим многочленом циклического кода. При этом степень $\deg[g(x)]=n-k$, где k – число информационных символов. Свойство (2) является отправным при выборе порождающего многочлена циклического кода.

2.2. Выбор порождающего многочлена циклического (n, k) кода

Выбор порождающего многочлена определяет корректирующие способности кода. При этом необходимо учитывать конечное назначение кода. Например, если для обнаружения ошибок необходимо только зафиксировать факт ошибки, то для ее исправления надо определить номер ошибочного разряда. Основой для этих действий является свойство делимости разрешенной кодовой комбинации на порождающего многочлен без остатка. Ненулевой остаток от деления ошибочной кодовой комбинации на порождающего многочлен называют синдромом. Возникновение ненулевого остатка является признаком ошибки, а конкретный вид синдрома в ряде случаев однозначно связан с местом возникновения ошибки. В таком случае для исправления ошибок необходимо выбирать такие порождающего многочлены, которые при делении ошибочных комбинаций порождают количество различных остатков, совпадающее с числом возможных ошибок. Многочлены, обладающие таким свойством, называют примитивными. Существует два способа нахождения порождающего многочлена.

А. <u>По заданным n и k</u>

Выбор порождающего многочлена циклического кода производится на основе разложения двучлена x^n+1 на множители - полиномы с коэффициентами 0 и 1. Рассмотрим эту процедуру на примере циклического кода (7,3). Двучлен x^7+1 разлагается на множители следующим образом:

$$x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1). \qquad (3)$$

При этом возможны следующие многочлены:

$g_1(x) = x+1, \quad g_2(x) = x^3 + x + 1, \quad g_3(x) = x^3 + x^2 + x + 1,$

$g_4(x) = (x+1)(x^3 + x + 1) = x^4 + x^3 + x^2 + 1,$

$g_5(x) = (x+1)(x^3 + x^2 + 1) = x^4 + x^2 + x + 1,$

$g_6(x) = (x^3 + x + 1)(x^3 + x^2 + 1) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1,$

$g_7(x) = x^2 + 1.$

Поскольку число проверочных символов $n-k$ равно степени порождающего многочлена, то требуемый код (7,3) может быть образован многочленом степени 4, т.е. $g_4(x)$ и $g_5(x)$. Как видим, этот метод не обеспечивает однозначного выбора порождающего многочлена. При этом корректирующая способность полученных кодов будет различной.

Двучлен $x^{15}+1$ разлагается на множители следующим образом:

$$x^{15} + 1 = (x+1)(x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + 1)(x^4 + x^3 + x^2 + x + 1). \quad (4)$$

Комбинируя различные сочетания скобок, можно получить 28 возможных порождающего многочленов. В этом случае неоднозначность выбора $g(x)$ проявляется ещё сильнее.

Б. <u>По заданным n и d</u>

В этом случае используется методика, разработанная Боузом, Чоудхури и Хоквингемом. Коды, построенные по этой методике, получили название кодов БЧХ. Порождающий многочлен для кода БЧХ определяется равенством

$$g(x) = НОК[m_r(x), m_{r+1}(x), ..., m_{r+d-2}(x)],$$

где НОК - наименьшее общее кратное, $r=0$ при четном d и $r=1$ при нечетном d. $m_1(x)$ - так называемые <u>минимальные функции</u>, или <u>минимальные многочлены</u>. Минимальные функции являются неприводимыми полиномами и имеют вид [1]:

-для $n=7$

$m_0(x) = x+1, \; m_1(x) = x^3 + x + 1, \; m_2(x) = x^3 + x + 1, \; m_3(x) = x^3 + x^2 + 1,$

$m_4(x) = x^3 + x + 1, \quad m_5(x) = x^3 + x^2 + 1, \; m_6(x) = x^3 + x + 1;$

-для n=15

$m_0(x) = x+1, \qquad m_1(x) = x^4 + x + 1, \quad m_2(x) = x^4 + x + 1,$

$m_3(x) = x^4 + x^3 + x^2 + x + 1, \; m_4(x) = x^4 + x + 1, \quad m_5(x) = x^2 + x + 1,$

$m_6(x) = x^4 + x^3 + x^2 + x + 1, \; m_7(x) = x^4 + x + 1, \quad m_8(x) = x^4 + x + 1,$

$m_9(x)=x^4+x^3+x^2+x+1,\quad m_{10}(x)=x^2+x+1,\quad m_{11}(x)=x^4+x^3+1,$
$m_{12}(x)=x^4+x^3+x^2+x+1,\ m_{13}(x)=x^4+x^3+1,\ m_{14}(x)=x^4+x^3+1.$

Рассмотрим пример. Пусть требуется построить циклический код с тремя информационными символами ($k=3$) и кодовым расстоянием $d=4$. Вычислим порождающий многочлен в рассматриваемом примере. Так как d имеет четное значение, примем $r=0$, тогда:

$g(x)=\text{НОК}[m_0(x), m_1(x), m_2(x)]=m_0(x)m_1(x)=x^4+x^3+x^2+1=g_4(x).$

Если проделать то же самое для $d=3$, то $r=1$ и

$g(x)=\text{НОК}[m_1(x), m_2(x)]=m_1(x)=x^3+x+1=g_3(x).$

2.3. Принципы кодирования и декодирования циклических кодов

Кодирование циклических кодов можно выполнять тремя способами.

Первый способ кодирования основан на правиле:

$$P_j(x)=J_j(x)g(x). \tag{5}$$

Главный недостаток этого способа состоит в том, что он приводит к неразделимому коду, в котором информационные и проверочные символы не занимают постоянных мест в блоке (кодограмме).

Второй способ образования кодограмм циклического кода основан на соотношении

$$K_j(x)=\left[x^{n-k}J_j(x)\right]\oplus R_{g(x)}\left[x^{n-k}J_j(x)\right]. \tag{6}$$

Нетрудно убедиться, что $K_j(x)$ делится на $g(x)$ без остатка и, следовательно, кодограммы (6) образуют циклический код.

Пример 1.

Пусть кодированию подлежат $J_3(x)=x+1$ (011) и $J_4(x)=x^2$ (100).

Порождающий многочлен $g(x)=g_4(x)$.

Согласно (5) имеем:

$K_3(x)=x^5+x^2+x+1$ (0100111);

$K_4(x)=x^6+x^5+x^4+x^2$ (1110100).

Правило (6) дает:

$K_3(x)=x^4(x+1)+x^3+x=x^5+x^4+x^3+x$ (0111010);

$K_4(x)=x^4x^2+x^3+x^2+x=x^6+x^3+x^2+x$ (1001110).

Видно, что правило (5) приводит к неразделимому коду, а правило (6) – к разделимому, когда на первых позициях стоят информационные символы, а на последующих позициях - проверочные.

Третий способ кодирования основан на том, что циклический код является систематическим (линейным) и его проверочные и информационные символы связаны линейными соотношениями. Для использования этого метода кодирования необходимо представить комбинацию циклического кода в другом виде:

$$a_0, a_1, \dots, a_{n-k-1}, a_{n-k}, \dots, a_{n-1},$$

где $a_0, a_1, \dots, a_{n-k-1}$ - проверочная часть, $a_{n-k}, \dots, a_{n-1}$ - информационная часть.

Можно показать, что j-й проверочный символ (b_j в прежнем обозначении или a_{n-k-j} в новом), занимающий в кодограмме (n-k-j)-ю позицию, определяется рекуррентным соотношением:

$$a_{n-k-j}=\sum_{i=0}^{k-1} a_{n-i-j}h_i,\ j=1,2,\dots,n-k, \tag{7}$$

где h_j - коэффициенты генераторного (проверочного) многочлена

$$h(x)=\frac{x^n+1}{g(x)}=h_0+h_1x+\dots+h_kx^k.$$

Таким образом, согласно (7) любой символ циклического кода является взвешенной суммой k других символов кода (суммирование выполняется по модулю два).

Пример 2. Вычислим генераторный (проверочный) многочлен

$$h(x)=\frac{x^7+1}{x^4+x^3+x^2+1}=x^3+x^2+1.$$

Для информационного многочлена $J_1(x)=x+1$ согласно (7) имеем

$$b_1=a_3=\sum_{i=0}^{2} a_{6-i}h_i=a_6h_0\oplus a_5h_1\oplus a_4h_2=0\cdot1\oplus1\cdot0\oplus1\cdot1=1,$$

$$b_2=a_2=\sum_{i=0}^{2} a_{5-i}h_i=a_5h_0\oplus a_4h_1\oplus a_3h_2=1\cdot1\oplus1\cdot0\oplus1\cdot1=0.$$

Из примера видно, что проверочные символы a_3, a_2, a_1, a_0 совпадают с соответствующими символами кодового слова в примере 1.

Обнаружение ошибок в принятых комбинациях циклических кодов может быть осуществлено разными способами.

Наиболее простой - это хранить на приемной стороне весь список разрешённых кодовых комбинаций и сравнивать с ними принятую комбинацию. Если она не совпадает ни с одной из имеющихся в списке - произошла ошибка.

Второй способ заключается в том, что по принятой информационной части кодовой комбинации вычисляются проверочные разряды и сравниваются с принятыми проверочными разрядами. Если они совпадают, то ошибка отсутствует. В противном случае фиксируется наличие ошибки.

Третий способ, используемый для циклических кодов, основан на свойствах делимости многочленов, описывающих разрешённые кодограммы, на порождающий многочлен без остатка. Если остаток от деления нулевой, то ошибки нет. В противном случае принятая кодограмма является запрещённой (имеют место ошибки). Исправление ошибок осуществляется путем анализа полученного остатка.

2.4. Принципы построения делительных устройств

Пусть требуется разделить многочлен $a(x)=x^5+x^4+x^2+x+1$ на многочлен $g(x)=x^3+x+1$.

Вычислим частное и остаток, используя алгоритм Эвклида:

$$
\begin{array}{lll}
r_3(x) & x^5+x^4+0\cdot x^3+x^2+x+1 & \big|\ x^3+x+1 \\
\oplus & \underline{x^5+0\cdot x^4+x^3+x^2 \quad B_4(x)} & \ \ x^2+x+1=h(x) \\
r_4(x) & x^4+x^3+0\cdot x^2+x+1 & \\
\oplus & \underline{x^4+0\cdot x^3+x^2+x \quad B_5(x)} & \\
r_5(x) & x^3+x^2+0\cdot x+1 & \\
\oplus & \underline{x^3+0\cdot x^2+x+1 \quad B_6(x)} & \\
r_6(x) & x^2+x\,. & \qquad (8)
\end{array}
$$

В устройстве деления последовательно реализуются все операции этого алгоритма. Основой устройств деления является регистр сдвига с логическими обратными связями. Исходное состояние регистра нулевое. Число ячеек памяти определяется степенью полинома делителя. Многочлен - делимое поступает на вход регистра, начиная со старшего коэффициента. Деление начинается после того, как этот старший коэффициент достигает последней ячейки памяти регистра. В рассматриваемом примере после трех тактов работы регистр содержит $r_3(x)=x^5+x^4+0\cdot x^3$ [см. алгоритм деления (8)]. После 4-го такта коэффициент при x^5 поступает на выход схемы деления, а в первую ячейку регистра запишется коэффициент при x^2. Таким образом, содержимое регистра теперь

$$x^4+0\cdot x^3+x^2. \qquad (9)$$

Коэффициент при x^5 не только поступает на выход схемы деления, но и одновременно воздействует на вход логической обратной связи (ЛОС) (если этот коэффициент единица) с тем, чтобы выражение (9) совпало с полиномом $r_4(x)=x^4+x^3+0\cdot x^2$ алгоритма (8). Очевидно, что для этого необходимо изменить содержание первой и второй ячеек регистра на противоположное. Изменение содержания этих ячеек эквивалентно сложению по mod2 полинома (9) и многочлена $B_4(x)=0\cdot x^4+x^3+x^2$, формируемого ЛОС:

$$
\begin{array}{ll}
 & x^4+0\cdot x^3+x^2 \\
\oplus & \underline{0\cdot x^4+x^3+x^2=B_4(x)} \\
 & x^4+x^3+0\cdot x^2=r_4(x)\,.
\end{array}
$$

Легко убедиться, что после 5 тактов работы регистр содержит коэффициенты многочлена

$$x^3+0\cdot x^2+x\,. \qquad (10)$$

В цепи ЛОС формируются полином $B_5(x)=0\cdot x^3+x^2+x$ и промежуточный остаток от деления $r_5(x)=x^3+x^2+0\cdot x$, что соответствует промежуточному результату в алгоритме (8).

После шести тактов работы формируется частное, а регистр содержит остаток от деления многочленов $r_6(x)$, отметим, что при формировании всех промежуточных остатков $r_i(x)$ структура ЛОС не меняется, что позволяет легко составлять схемы деления по первому остатку $r_4(x)$. Схема делительного устройства показана на рис. 1, где обозначено: □ - ячейка памяти, ⊕ - сумматор по модулю два.

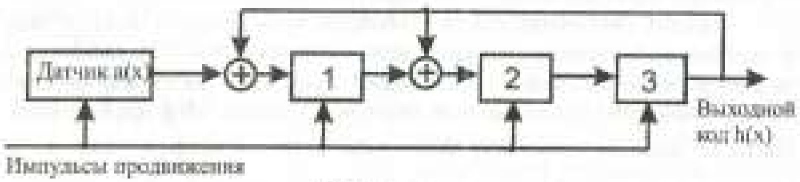

Рис. 1

Основные правила построения делительных устройств:

1) устройство деления многочленов есть регистр сдвига с логическими обратными связями от выхода к входу и с числом ячеек памяти, равным степени многочлена делителя;

2) ячейки памяти нумеруются от входа к выходу, начиная с единицы;

3) сумматоры по mod2 своими первыми входами подключаются к выходам тех ячеек памяти, номера которых совпадают со степенью ненулевых коэффициентов многочлена делителя, кроме последней ячейки;
4) выход последней ячейки подключается ко вторым входам всех сумматоров по mod2;
5) выходы сумматоров по mod2 подключаются к входам следующих ячеек памяти.

2.5. Кодирующие устройства циклических кодов

В разделе 2.3 установлены два алгоритма кодирования, используемые на практике. **Первый** из них, определяемый выражением (6), требует выполнения умножения информационного многочлена $J_j(x)$ на одночлен x^{n-k} и вычисления остатка от деления произведения $x^{n-k}J_j(x)$ на порождающий многочлен $g(x)$. Умножение на одночлен x^{n-k} не требует специального устройства, так как такое умножение означает приписывание n-k нулей со стороны младших разрядов к кодируемой кодовой комбинации. Эта операция выполняется за счет организации работы датчика $J_j(x)$. Вычисление остатка осуществляется схемой деления на $g(x)$. Информационные символы с приписанными n-k нулями поступают, начиная со старшего разряда, одновременно в схему деления и в канал связи. Когда все k информационных символов и n-k нулей поступят в канал связи, регистр схемы деления будет содержать остаток, т.е. проверочные символы передаваемой кодограммы. Для вывода их из регистра требуется n-k тактов. Недостаток этого кодера состоит в том, что формируемая кодограмма циклического кода имеет разрыв между информационными и проверочными символами в n-k символов, что снижает скорость передачи информации. Недостаток устраняется использованием специальной схемы деления [1].

Второй тип кодирующих устройств, применяемых на практике, функционирует на основе рекуррентных соотношений (7) и реализуется в виде регистра сдвига с k ячейками памяти (по числу информационных символов) и логической обратной связью. Информационные символы заносятся в регистр так, чтобы старший коэффициент a_{n-1} оказался в последней ячейке (см. рис. 2), а a_{n-k} - в первой.

Логическая обратная связь построена таким образом, что на ее выходе формируется первый проверочный символ

$$a_{n-k-1}=\sum_{i=0}^{k-1} a_{n-i-1}h_i = a_{n-1}h_0 \oplus a_{n-2}h_1 \oplus ... \oplus a_{n-k}h_{k-1}.$$

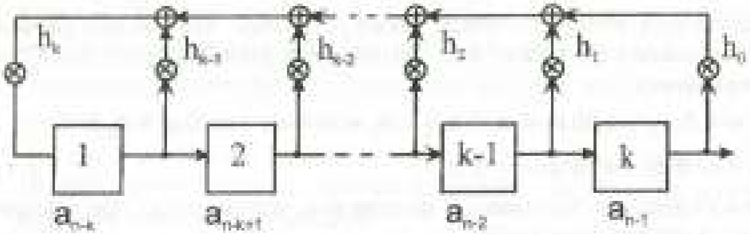

Рис. 2

После поступления первого импульса продвижения (на рис. 2 не показан) этот символ запишется в первую ячейку, а информационные символы сместятся на одну ячейку вправо. Теперь на выходе ЛОС действует второй проверочный символ:

$$a_{n-k-2}=\sum_{i=0}^{k-1} a_{n-i-2}h_i = a_{n-2}h_0 \oplus a_{n-3}h_1 \oplus ... \oplus a_{n-k-1}h_{k-1}.$$

На последующих тактах работы последовательно формируются оставшиеся проверочные символы и затем всё повторяется. Рассмотрим работу кодирующего устройства на примере кодера для кода (7,3), порожденного многочленом $g(x)=x^4+x^2+x+1$. Генераторный многочлен при этом таков: $h(x)=x^3+x+1$. Соответствующее кодирующее устройство приведено на рис. 3.

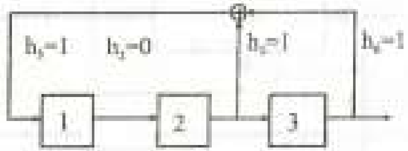

Рис. 3

Таблица1

N п/п	ЯП 1	2	3
0	a_4	a_5	a_6
1	$a_3=a_5\oplus a_6$	a_4	a_5
2	$a_2=a_4\oplus a_5$	a_3	a_4
3	$a_1=a_3\oplus a_4$	a_2	a_3
4	$a_0=a_2\oplus a_3$	a_1	a_2
5	$a_6=a_1\oplus a_2$	a_0	a_1
6	$a_5=a_0\oplus a_1$	a_6	a_0
7	$a_4=a_6\oplus a_0$	a_5	a_6

В исходном состоянии содержимым регистра являются $a_4, a_5, \dots, a_6$. Из схемы видно, что проверочный символ есть сумма по mod2 содержимого ячеек 2 и 3. Это позволяет легко вычислить содержимое регистра на любом такте работы. Результаты вычислений представлены в табл. 1.

Из табл. 1 можно видеть, что после семи тактов работы регистр вновь содержит информационные символы. Следовательно, далее содержимое регистра начинает повторяться и на выходе регистра сформируется периодически повторяющаяся кодограмма циклического кода. Из таблицы также установим, что каждый символ циклического кода может быть представлен разными линейными соотношениями:

$$a_6 = a_1 \oplus a_2 = a_3 \oplus a_5 = a_0 \oplus a_4; \quad a_5 = a_2 \oplus a_4 = a_3 \oplus a_6 = a_0 \oplus a_1;$$
$$a_4 = a_0 \oplus a_6 = a_2 \oplus a_5 = a_1 \oplus a_3. \qquad (11)$$

Эти соотношения, называемые проверками, используются при декодировании циклических кодов.

Простейшим декодирующим устройством является декодер, обнаруживающий ошибки (но не исправляющий). Исправление ошибок осуществляется обычно повторной передачей кодограммы по команде декодера. Команда на повторную передачу передается на передающий конец канала связи по обратному каналу.

Функциональная схема декодирующего устройства, обнаруживающего ошибки, приведена на рис. 4.

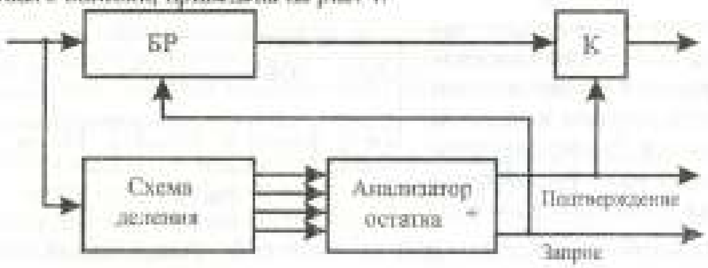

Рис. 4

Принцип работы устройства основан на проверке основного свойства кодограммы циклического кода - делимости на порождающий многочлен. Принимаемая кодограмма поступает на вход схемы деления и одновременно записывается в буферный регистр (БР). Когда вся кодограмма запишется в буферный регистр, в схеме деления образуется остаток от деления кодограммы на порождающий многочлен, называемый синдромом

$$S(x) = S_{n-k}x^{n-k} + S_{n-k-1}x^{n-k-1} + ... + S_0.$$

Если ошибки отсутствуют, то регистр схемы деления содержит одни нули (нулевой синдром, т.е. $S_0 = S_1 = ... = S_{n-k} = 0$). Анализатор синдрома вырабатывает сигнал "Подтверждение", по которому разрешает выдачу кодограммы потребителю. В противном случае вырабатывается сигнал "Запрос", который стирает содержимое буферного регистра.

Более сложно реализуются декодеры циклических кодов с исправлением ошибок. Существует несколько способов функционирования таких устройств. Для определения номеров элементов, в которых

произошла ошибка, существует несколько методов. Один из них основан на свойстве, что остаток, полученный при делении принятого с ошибками многочлена $K_0(x)$ на порождающий многочлен $g(x)$, называемый синдромом, равен остатку, полученному при делении на $g(x)$ соответствующего многочлена ошибок $E(x) = K_0(x) + K(x)$, где $K(x)$ исходный многочлен циклического кода, т.е. для данного кода (n,k) синдром зависит только от номера ошибочного разряда и не зависит от вида передаваемой комбинации.

Наиболее очевидный алгоритм исправления ошибок основан на связи между содержимым синдрома и номером позиции, где имеет место ошибка. Структура декодера показана на рис. 5.

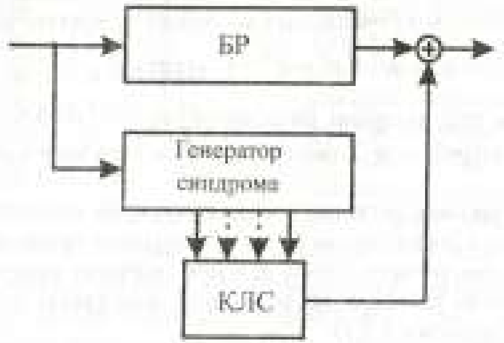

Рис. 5

Процесс декодирования разбивается на два этапа. На первом этапе принимаемая кодограмма записывается в БР, а схема деления вычисляет синдром.

На втором этапе символы принятой кодограммы последовательно покидают БР, а схема деления (на ее входе теперь действуют только нули) производит операции над синдромом. Если синдром отличен от нуля, то на n тактах второго этапа содержимое регистра схемы деления есть разные двоичные числа разрядностью n-k. Результат работы схемы деления на очередном i-м такте второго этапа фактически является продуктом деления принятого многочлена, сдвинутого влево на j разрядов, на порождающий многочлен. Комбинаторно-логическая схема (КЛС) строится с таким расчетом, чтобы реагировать на те двоичные числа, которые появляются в момент, когда ошибочные символы покидают БР.

Сложность КЛС зависит от числа исправляемых ошибок. Простейшие схемы получаются при реализации кодов, рассчитанных на исправление однократных ошибок.

Двоичное число (опознаватель), на которое должна реагировать КЛС, легко определить, заметив, что это число генерируется схемой деления за один такт работы из синдрома, соответствующего ошибке в старшем разряде (поскольку старший разряд расположен в выходной ячейке БР). Так как кодограммы циклического кода делятся на $g(x)$ без остатка, то, очевидно, что опознаватель есть остаток от деления x^n на

$g(x)$ (с учётом того, что всю кодовую комбинацию необходимо при этом сдвинуть в сторону старших разрядов на один разряд).

Рассмотрим пример, иллюстрирующий принцип построения КЛС, исправляющей ошибки в комбинациях циклического (7,4) кода с образующим многочленом $g(x)=x^3+x+1$. Опознаватель найдем, разделив x^7 на $g(x)$. Он равен x^0 (001). Схема деления, вычисляющая синдром, построенная по правилам разд.2.6, приведена на рис. 1. Пусть на вход этой схемы поступает кодограмма циклического кода $a(x)=x^6+x^5+x^3+1$ (1101001), в которой имеет место ошибка в 3-м разряде. С учетом этого многочлен, описывающий принятую диаграмму, имеет вид: $a(x)=x^6+x^5+x^3+x^2+1$ (1101101).

Разделив $a'(x)$ на $g(x)$, получим синдром $S(x)=x^2$ (100), на вычисление которого потребуется 7 тактов работы делительного устройства.

Рассмотрим содержимое регистра схемы деления на последующих тактах работы при условии, что на ее вход поступают только нули. В исходном состоянии регистр схемы деления содержит синдром ("1" в ячейке 3). Состояния регистра на дополнительных тактах работы отражены в табл. 2 (колонки 1,2,3).

Таблица 2

	ЯП			
Такт	1	2	3	Вых
0	0	0	1	
1а	0	0	0	1
1б	1	1	0	
2а	0	1	1	0
2б	0	1	1	
3а	0	0	1	1
3б	1	1	1	
4а	0	1	1	1
4б	1	0	1	
5а	0	1	0	1
5б	1	0	0	
6а	0	1	0	0
6б	0	1	0	
7а	0	0	1	0
7б	0	0	1	

Каждый такт разделен на два: содержимое ячеек на подтакте "а" вычисляется без учета ЛОС, на подтакте "б" учитывается действие ЛОС. В последней колонке записан символ, действующий на выходе делительного устройства и одновременно на входе ЛОС. Отметим, что для вывода ошибочного символа из БР требуется 5 рабочих тактов [кодограмма a'(x) записывается в БР начиная со старшего разряда]. Из таблицы можно видеть, что опознаватель 100 содержится в регистре схемы деления на 5-м такте, т.е. в момент вывода ошибочного символа из БР.

Проверим, выполняется ли это условие для случая ошибки в 5-м разряде, т.е. $a'(x)=x^6+x^5+x^3+x^2+1$ (1111001). Синдром $S(x)=x^2+x$ (110). Из табл.3 можно видеть, что опознаватель 100 содержится в регистре сдвига схемы деления на 3-м дополнительном такте и на этом же такте ошибка выводится из БР.

Легко убедиться, что для рассматриваемого кода комбинация 100 является опознавателем любой одиночной ошибки (в любом разряде).

Таблица 3

	ЯП			
Такт	1	2	3	Вых
0	0	0	1	
1а	0	0	1	1
1б	1	1	1	
2а	0	1	1	1
2б	1	0	1	
3а	0	1	0	1
3б	1	0	0	
4а	0	1	0	0
4б	0	1	0	

Аналогичным образом можно построить КЛС для исправления ошибок более высокой кратности. Однако сложность таких схем значительна, из-за чего они используются весьма редко.

Последний метод декодирования, рассматриваемый здесь, называется мажоритарным. Он основывается на реализации проверок вида (11) и подсчете их результатов. Решение о значении проверяемого символа принимается по большинству результатов контрольных проверок (отсюда и название метода - по аналогии с мажоритарной системой голосования).

Система контрольных проверок вида (11), построенная для одного символа a_i циклического кода, может быть использована для декодирования всех символов этой комбинации. Действительно, контрольным проверкам удовлетворяет любая кодограмма циклического кода, а следовательно, и кодограммы, полученные циклическими перестановками исходной. Таким образом, для декодирования символа: a_{i+j} достаточно произвести j сдвигов принятой кодограммы, не изменяя ни схемы вычисления проверочных соотношений, ни мажоритарного элемента.

Существует две разновидности мажоритарных декодеров. Рассмотрим первую из них на примере некоторого циклического кода длиной n=7, для которого проверки имеют вид:

$$a_6=a_3\oplus a_4=a_1\oplus a_5=a_0\oplus a_2;\; a_5=a_1\oplus a_6=a_0\oplus a_4=a_2\oplus a_3. \qquad (12)$$

Мажоритарный декодер МД-1 первой разновидности представляет собой БР, дополненный устройствами, реализующими проверки (12) относительно какого-либо одного символа (например, a_6), и мажоритарным элементом (МЭ). При этом используется и тривиальная проверка (вида $a_6 = a_6$). Схема декодера приведена на рис. 6. МЭ выносит решение о значении проверяемого символа по большинству результатов проверок, действующих на его входах. Если результаты проверок делятся поровну (например, $a_6^1=a_6^2=1$, $a_6^3=a_6^4=0$), то МЭ выдает сигнал, свидетельствующий о наличии неуправляемой ошибки (например, двукратной).

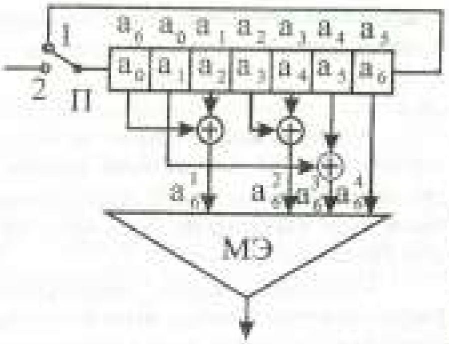

Рис. 6

В практических схемах вместо МЭ используют пороговый элемент ПЭ, который не фиксирует появление неисправляемой ошибки.

Для проверки остальных символов кода достаточно произвести циклическую перестановку записанной кодограммы. С этой целью переключатель 2 устанавливается в положение 1 и подается импульс сдвига.

На рис. 6 приведен результат такого сдвига на один такт. Теперь декодер реализует следующие соотношения: $a_5 a_1 \oplus a_6 a_0 \oplus a_4 a_2 \oplus a_3$. Сравнив их с (12), можно убедиться, что декодер осуществляет проверки относительно символа a_3. Таким образом, для проверки и исправления информационных символов потребуется k тактов работы.

В рассмотренном примере имеется 4 соотношения проверки. Нетрудно заметить, что любая однократная ошибка нарушает только одну контрольную проверку. Следовательно, МЭ исправит однократную ошибку. Двукратные ошибки (ошибки в двух символах кодограммы) не могут быть исправлены (голоса разделяются поровну), но будут обнаружены.

В проверках (12) каждый символ a_i участвует один раз. Такие проверки называют разделёнными. Следует иметь в виду, что разделенные проверки получаются не всегда, т.е. один или несколько символов могут входить в проверки не один раз. В этом случае и однократная ошибка может нарушать более чем одну проверку.

Мажоритарный декодер МД-2 использует тот факт, что элементы синдрома есть суммы по mod2 определенных символов принимаемой комбинации. Действительно, пусть для некоторого циклического кода генератор синдрома представлен на рис. 7.

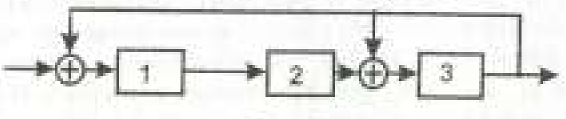

Рис. 7

Формирование синдрома при поступлении кодограммы $a(x) = a_6 x^6 + a_5 x^5 + a_4 x^4 + \ldots + a_1 x + a_0$ отображается в табл. 4.

На 7-м такте работы схема образует синдром (S1, S2, S3).

Таблица 4

	ЯП			
Такт	1	2	3	Вых
3	a_4	a_5	a_6	
4а 4б	a_3 a_3+a_6	a_4 a_4	a_5 a_5+a_6	a_6
5а 5б	a_2 $a_2+a_5+a_6$	a_3+a_6 a_3+a_6	a_4 $a_4+a_5+a_6$	a_5+a_6
6а 6б	a_1 $a_1+a_4+a_5+a_6$	$a_2+a_5+a_6$ $a_2+a_5+a_6$	a_3+a_6 $a_3+a_4+a_5$	$a_4+a_5+a_6$
7а 7б	a_0 $a_0+a_3+a_4+a_5$	$a_1+a_4+a_5+a_6$ $a_1+a_4+a_5+a_6$	$a_2+a_5+a_6$ $a_2+a_3+a_4+a_6$	$a_3+a_4+a_5$
8а 8б	0 $a_3+a_5+a_4+a_6$	$a_0+a_3+a_4+a_5$ $a_0+a_3+a_4+a_5$	$a_1+a_4+a_5+a_6$ $a_1+a_2+a_3+a_5$	$a_2+a_3+a_4+a_6$

Причем согласно табл. 4 имеем:

$S_1 = a_0 \oplus a_3 \oplus a_4 \oplus a_5$, $S_2 = a_1 \oplus a_4 \oplus a_5 \oplus a_6$, $S_3 = a_2 \oplus a_3 \oplus a_4 \oplus a_6$.

Если ошибок нет, то, очевидно, что эти суммы равны нулю (нулевой синдром – основной признак отсутствия ошибок). Значит, равны нулю суммы:

$S_1 + S_2 = a_0 \oplus a_1 \oplus a_3 \oplus a_6$, $S_1 + S_3 = a_0 \oplus a_2 \oplus a_5 \oplus a_6$.

Наличие ошибки только в старшем разряде приводит к тому, что последние 4 суммы оказываются равными единице (S_1=0, так как в нее a_6 не входит).

Если же имеет место ошибка в любом другом разряде, то равными единице оказываются только две суммы из четырех. Это обстоятельство можно использовать для исправления ошибки с помощью схемы, приведенной на рис. 8. На вход ПЭ подаются четыре суммы: S_1, $S_1 \oplus S_2$, $S_1 \oplus S_3$, S_3. Если проверяемый первым 7-й разряд ошибочен, то все эти суммы равны единице и на выходе ПЭ появится "1" (4>3).

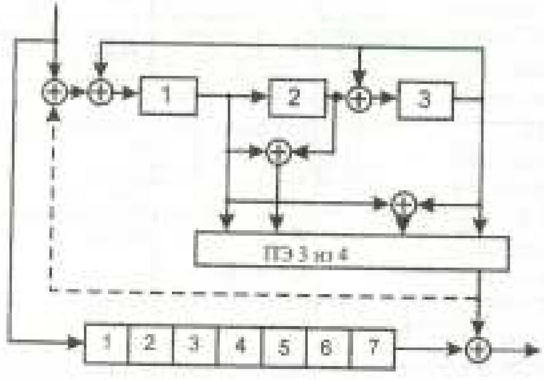

Рис. 8

Таким образом, на 8-м такте ошибка будет исправлена. При этом синдром должен стать, очевидно, нулевым (так как ошибка однократная и исправлена). Анализируя 8-ю строку табл.4, можно видеть, что содержимое только первой ячейки генератора синдрома отлично от нуля ($S_1 = a_2 \oplus a_3 \oplus a_4 \oplus a_6$ зависит от a_6, а S_2 и S_3 от a_6 не зависят). Суммирование S_1 с "1" на выходе ПЭ (осуществляется связью, показанной на рис. 8 пунктиром) переводит первую ячейку в состояние "0". Вывод остальных символов из БР теперь уже не будет сопровождаться коррекцией.

Если однократная ошибка имеет место в 6-м разряде, то на 8-м такте ПЭ даст отклик "0" и поэтому 7-й символ кодограммы покинет БР без исправления. Синдром также не корректируется, и, следовательно, содержимое генератора синдрома определяется 8-й строкой табл.4.

Ошибочный 6-й разряд обращает все четыре суммы в "1", а ошибка в любом другом разряде - только две из них. Следовательно, ПЭ распознает ошибку в 6-м разряде. На последующих тактах работы аналогичным образом проверяются остальные разряды комбинации.

3. ОПИСАНИЕ МАКЕТА КОДИРУЮЩЕГО УСТРОЙСТВА

Макет кодирующего устройства состоит из собственного кодирующего устройства, устройства формирования тактов импульсов продвижения, задающего генератора и устройства внесения ошибок. Функциональная схема макета приведена на рис. 9.

3.1. Кодирующее устройство

Кодирующее устройство реализовано на основе регистра с ячейками и функционирует в соответствии с алгоритмом (7) (см. также разд. 2.5). Всего таких ячеек 7. Информационные символы заносятся в регистр с помощью кнопок Кн1-Кн7, размещенных на лицевой панели макета. ЛОС кодера построена на основе сумматоров по mod2.

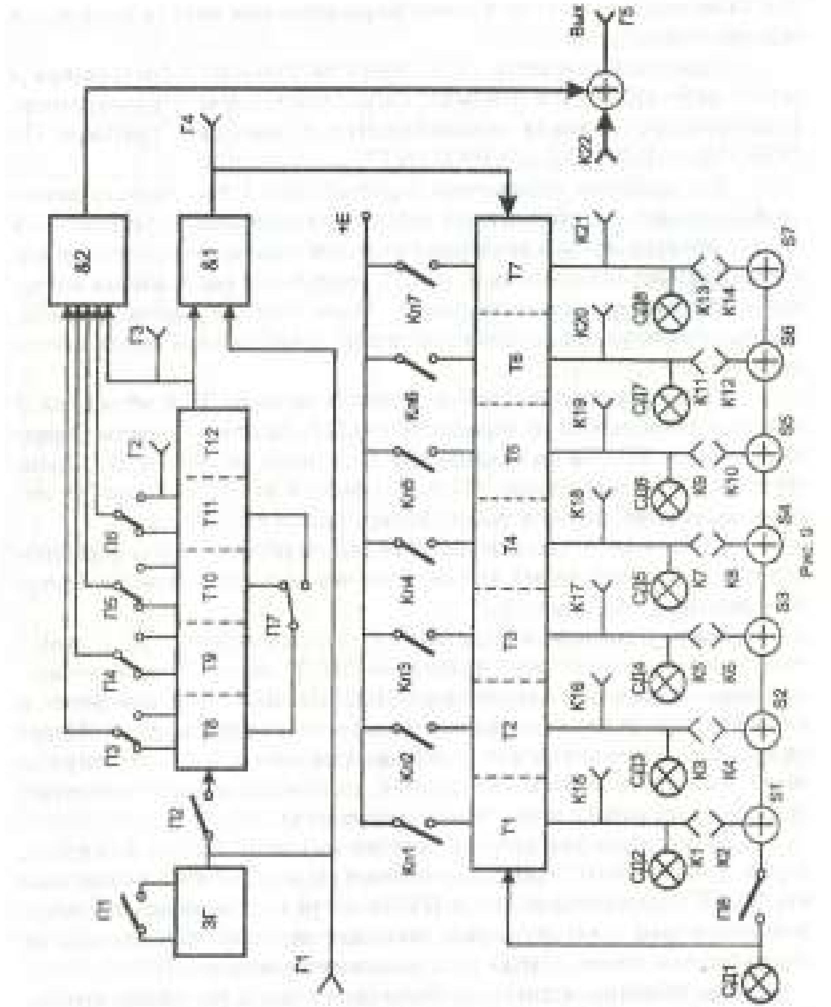

Рис. 9

Реализация кодирующего устройства для заданного кода обеспечивается с помощью внешней коммутации, осуществляемой на лицевой панели пакета (гнезда К1-К14).

Для индикации состояния ячеек регистра на их выходах включены светодиоды. В однократном режиме они позволяют проверить правильность работы кодера.

3.2. Задающий генератор и схема формирования пакета импульсов продвижения

Задающий генератор (ЗГ) собран по схеме мультивибратора и может работать в двух режимах: автоколебательном и однократном. Переключение режимов осуществляется с помощью тумблера П1 "РЕЖИМ - АВТОМАТ – ОДНОКРАТ".

Для удобства наблюдения кодовых слов с помощью осциллографа, а также для обеспечения работы декодирующего устройства в макете предусмотрена схема формирования пакета импульсов сдвига. Схема формирует пакеты по 7 или 15 импульсов, разделенных интервалом, равным длительности пакета. Таким образом, кодовые комбинации, генерируемые кодером, на экране осциллографа наблюдаются раздельно.

Схема представляет собой делитель частоты ЗГ (счетчик) на 7 или 15 в зависимости от переключателя П7. Делитель частоты управляет схемой "И". На ее второй вход поступают импульсы ЗГ. Таким образом, на выходе схемы "И" формируются пачки по 7 или 15 импульсов, обеспечивающие работу кодирующего устройства.

Следует заметить, что для правильной работы схемы формирования пакетов необходима предварительная установка делителя частоты в исходное состояние.

Такая установка осуществляется переключением П2. С помощью этого переключателя в положении "ВКЛ" выход ЗГ подключается к входу делителя, а в положении "ОБНУЛЕНИЕ" ЗГ отключается, и одновременно делитель устанавливается в исходное состояние. Нарушение исходного состояния делителя приводит к тому, что первый пакет импульсов сдвига оказывается укороченным, что полностью дезорганизует работу кодирующего устройства.

Схема установки ошибок состоит из схемы "И" на 5 входов, двухвходовой схемы "ИЛИ" и сумматора по mod2; 4 из 5 входов схемы "ИЛИ" через переключатели П3-П6 могут подключаться к прямым или инверсным промежуточным выходам делителя частоты или не подключаться совсем. Пятый вход подключен к выходу ЗГ.

Как известно, делитель частоты имеет число различных состояний, равное коэффициенту деления. В данном случае число состояний равно длине кодовых слов, генерируемых кодером, и каждому состоянию делителя схема установки ошибки ставит в соответствие определенный номер позиции, в котором произойдет ошибка. Табл. 5 позволяет определить положение тумблеров для введения ошибки в определенный разряд кодовой комбинации. Подключение схемы внесения ошибки осуществляется с помощью коммутационных шнуров и гнезд К15-К22, расположенных на лицевой панели макета.

Таблица 5

N разряда	Положение			
	П3	П4	П5	П6
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	1	0	0
5	0	0	1	0
6	1	0	1	0
7	0	1	1	0
8	1	1	1	0
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

4. ОПИСАНИЕ МАКЕТА ДЕКОДИРУЮЩЕГО УСТРОЙСТВА

Макет декодирующего устройства, функциональная схема которого приведена на рис.10, позволяет реализовать декодирование кодограммы циклических кодов с обнаружением и исправлением однократных ошибок. В состав макета входят следующие элементы и устройства:

- набор триггеров Т1-Т10 с включенными между ними сумматорами по mod2 (М2), позволяющий составить схему генератора синдрома для всех циклических кодов длиной n=7 и кодов (15,7), (15,6), (15,5);
- буферный регистр, содержащий 15 триггерных ячеек памяти (Т11-Т25);
- набор сумматоров по mod2, пороговые элементы, используемые для реализации декодирующих устройств мажоритарного типа;
- устройство управления, согласующее работу кодирующего и декодирующего макета.

Устройство управления обеспечивает:

- использование в качестве тактовых импульсов сигналов задающего генератора из макета кодирующего устройства;
- формирование импульса обнуления генератора синдрома перед поступлением очередной кодовой комбинации;
- ввод кодовой комбинации из макета кодера;
- формирование сигналов, разрешающих срабатывание схемы обнаружения и исправления ошибок (например, после того, как закончится формирование синдрома).

Кроме того, в состав макета входят сумматор по mod2, с помощью которого осуществляется исправление ошибок на выходе БР, ключ (Кл), автоматически замыкающий БР в кольцо для осуществления циклических сдвигов, схема "ИЛИ", включаемая на выходе генератора синдрома для согласования полярности сигналов ЛОС, и сумматор по mod2 на входе генератора синдрома, через который осуществляется коррекция синдрома.

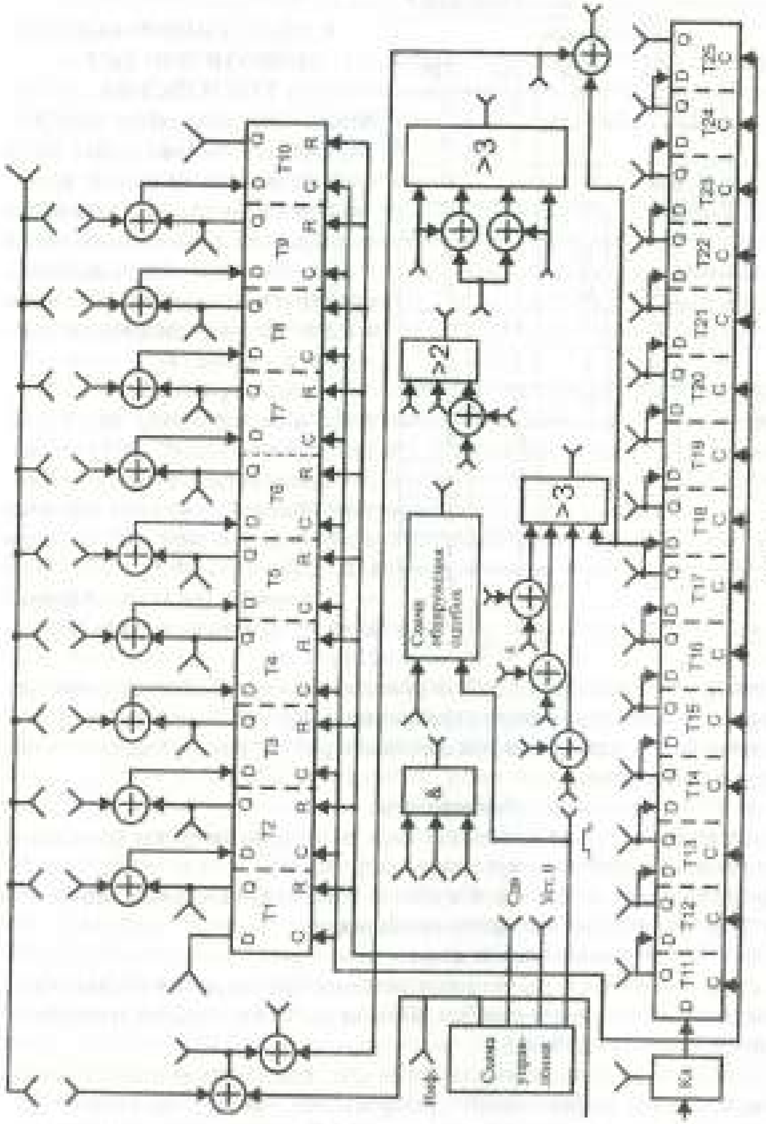

Рис. 10

Все необходимые сигналы от кодера на вход устройства управления подаются через разъем, расположенный на боковой стенке макета декодера.

Реализация той или иной схемы декодирования осуществляется с помощью коммутационных гнезд и шнуров. На лицевой панели макета нанесены функциональная схема и коммутационные гнезда для подключения осциллографа, что обеспечивает создание декодирующих устройств.

На выходах триггерных ячеек включены светодиоды, также выведенные на лицевую панель. В однократном режиме они позволяют контролировать правильность работы собранной схемы декодирования.

5. ВАРИАНТЫ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

Макеты лабораторной панели обеспечивают реализацию кодирующих и декодирующих устройств для циклических кодов с 7 и 15 символами. Варианты заданий сведены в табл.6.

Таблица 6

Бригада		Параметры КУ		ДКУ
№1	1	n=7, k=3	$g(x)=x^4+x^2+x+1$	МД-2
	2	n=7, d=4	БЧХ	МД-1
	3	n=15, d=5	БЧХ	ДОП
	4	n=7, k=4	$g(x)=x^3+x^2+1$	МД-1
№2	1	n=7, d=3	БЧХ	МД-2
	2	n=15, d=6	БЧХ	МД-1
	3	n=7, d=2	БЧХ	ДОП
	4	n=7, k=3	$g(x)=x^4+x^2+x+1$	МД-1
№3	1	n=7, d=4	БЧХ	МД-2
	2	n=7, d=3	БЧХ	МД-1
	3	n=15, k=6	$g(x)=x^9+x^7+x^6+x^3+x^2+1$	ДОП
	4	n=15, k=7	$g(x)=x^8+x^7+x^2+x+1$	МД-1
№4	1	n=15, k=7	$g(x)=x^8+x^7+x^5+x^4+x^3+x+1$	МД-2
	2	n=15, k=6	$g(x)=x^9+x^8+x^5+x^4+x^2+1$	МД-1
	3	n=15, d=7	БЧХ	ДОП
	4	n=15, k=5	$g(x)=x^{10}+x^5+1$	МД-1

В каждой бригаде задания распределяются согласно алфавитному списку бригады. Номер бригады определяется внутри подгруппы. Тип кодирующего устройства (КУ), который следует построить, одинаков для всех вариантов. Он определяется макетом кодирующего устройства. Тип декодирующего устройства (ДКУ) указан в соответствующей графе табл. 6 (МД - мажоритарный декодер, ДОП - декодер с опознавателем).

6. ЛАБОРАТОРНАЯ РАБОТА № 4

Подготовка к лабораторной работе.

1. Изучите все разделы настоящего описания, а также соответствующие разделы рекомендуемой литературы.
2. Постройте 3-4 следующих друг за другом комбинаций циклического кода, соответствующего вашему варианту задания. Убедитесь в том, что они переходят друг в друга при циклических сдвигах.
3. Нарисуйте функциональную схему делительного устройства по образующему многочлену вашего кода.
4. Нарисуйте функциональную схему кодирующего устройства с ячейками памяти и постройте таблицу, отражающую содержимое ячеек памяти на каждый такт работы (см. табл.1). Установите проверочные соотношения. Заполните таблицу для кодирования одной из построенных кодовых комбинаций.
5. Нарисуйте схему генератора синдрома. Проанализируйте работу схемы на каждый такт. Определите опознаватель (для ДКУ с опознавателем см. табл.2) или соотношения проверки (для ДКУ МД-2 см. табл.4).
6. Нарисуйте схему декодирующего устройства, отвечающую вашему заданию. Составьте таблицу, отражающую работу декодера на каждый такт для выбранной кодограммы.

Порядок выполнения лабораторной работы

А. Проверка результатов кодирования циклических кодов на ПЭВМ. Запустите с «Рабочего стола» компьютера программу CICLCOD и следуйте ее инструкциям в соответствии со своим вариантом лабораторного задания.

Б. Исследование работы устройства формирования циклического кода в системе схемотехнического моделирования MICRO-CAP 5.

В системе MICRO-CAP 5 создан файл LABCOD.CIR, содержащий принципиальную схему устройства кодирования циклических кодов, которая приведена на рис. 11 и состоит из двух универсальных восьмиразрядных сдвиговых регистров с параллельной загрузкой, элементов суммирования по модулю 2, обозначенных на схеме значком , генератора тактовой частоты и генератора импульса загрузки информационных разрядов параллельного кода.

В схеме выполнена основная часть соединений. В зависимости от варианта задания (т.е. требуемой схемы кодера и информационного кода) необходимо произвести некоторые соединения в принципиальной схеме с помощью редактора схем MICRO-CAP. Для этого, войдя в среду MC 5, вызвать через меню FILE файл LABCOD.CIR.

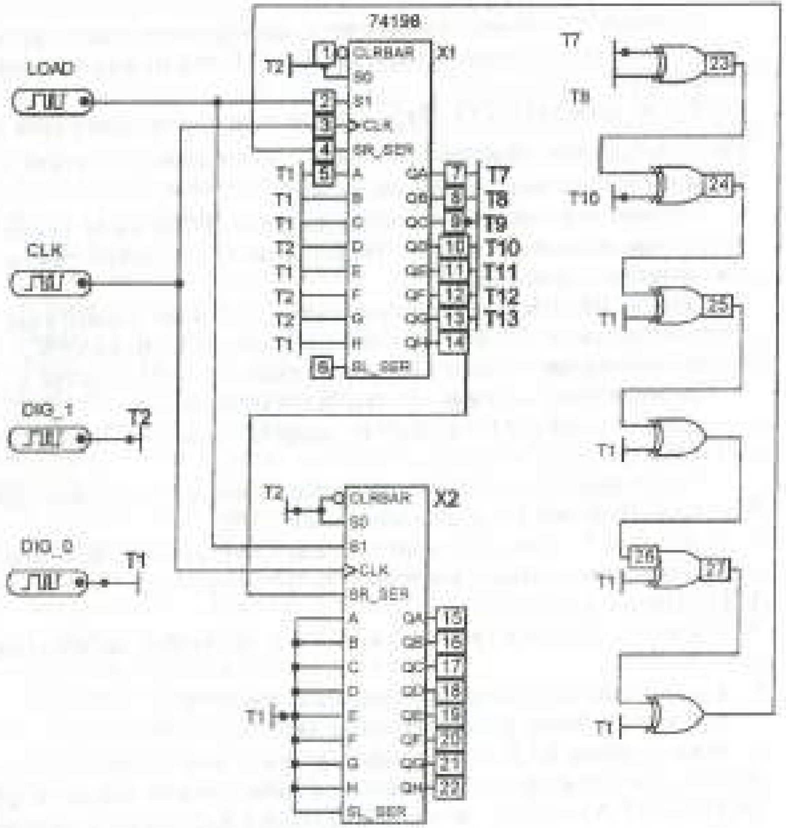

Рис. 11

1. Выполнить соединения в устройстве кодирования, соответствующие конкретной принципиальной схеме, т.е. подключить к схеме суммирования по модулю 2 требуемые выходы регистра в следующем порядке:

- включить режим (выбор объектов),
- двойным нажатием левой кнопки мыши на свободном входе элемента войти в режим редактирования компонента,
- в строке VALUE установить значение, соответствующее номеру выхода регистра X1, подключаемого к схеме суммирования (T7 – T13). Нажать ОК,
- повторить операцию до тех пор, пока все «суммируемые» выходы регистра не будут подключены к схеме суммирования ⊕,

- на незадействованные входы схемы суммирования подать уровень логического «0», установив значение VALUE при их редактировании равным T1.

2. Входы регистра X1 (A - H) в зависимости от вводимого кода подключить к уровню логического «0» или логической «1», для чего:

- включить режим - (выбор объектов),
- двойным нажатием левой кнопки мыши на обозначении T1 или T2, находящемся левее входов (A - H) регистра X1, включить режим редактирования компонента,
- в строке VALUE установить значение T1, если на данный вход подается логический «0», или T2, если - логическая «1», нажать OK,
- эту же операцию повторить для всех входов (A - H) регистра X1.

3. Провести временной анализ устройства кодирования:

- войти в меню ANALYSIS и выбрать функцию TRANSIENT ANALYSIS,
- в окне задания условий анализа задать значения: Time Range 2000n, Maximum Time Step 50n, Number of point 0,
- в колонке Y expression набрать имена контрольных точек принципиальной схемы, сигналы в которых требуется просмотреть (например, D(1) D(10) и т.д.),
- в колонке X expression установить t (т.е. исследование временных зависимостей),
- в колонке X Rangs установить значение 20e-007,0 ,
- в колонке Y Range установить значение N/A,
- нажать кнопку RUN в меню анализа, при этом в случае отсутствия ошибок при редактировании схемы и задании условий анализа в окне TRANSIENT ANALYSIS появятся временные диаграммы в характерных точках принципиальной схемы,
- для изменения условий анализа войдите в меню TRANSIENT, включите режим Limits и измените значения временных параметров или имена контрольных точек,
- для изменения соединений в принципиальной схеме закройте окно TRANSIENT ANALYSIS, соответствующей кнопкой отредактируйте схему, как было описано в п. 1,2.

7. ЛАБОРАТОРНАЯ РАБОТА № 5

Подготовка к лабораторной работе.

1. Используя материалы лабораторной работы №4, подготовьте схемы кодирующего и декодирующего устройств.
2. Тщательно изучите функциональные схемы макетов кодирующего и декодирующего устройств.
3. Продумайте реализацию ваших схем кодирования и декодирования на макетах. Составьте схему коммутации.

4. Составьте методики выполнения экспериментов по разделу 7.2.

Порядок выполнения работы.

Исследование устройства формирования циклического кода на макете.

1. Включите осциллограф и его корпус соедините с корпусом макета.
2. Установите переключатели на лицевой панели макета кодирующего устройства в положения, соответствующие вашему варианту задания.
3. С помощью коммутационных шнуров соберите кодирующее устройство с k ячейками для исследуемого кода.
4. По разрешению преподавателя включите тумблер питания.
5. Переведите ЗГ в однократный режим, обнулите делитель схемы формирования пакета импульсов и регистр кодирующего устройства.
6. Введите в регистр одну из комбинации информационного кода, для которой найдены (согласно п. 2 раздела 6.1) проверочные символы, контролируя состояние триггеров по индикаторам (светодиодам).
7. Последовательно нажимая кнопку "Пуск", проконтролируйте правильность работы кодера, сравнив содержимое регистра на каждый такт работы с таблицей, построенной согласно п.4 раздела 6.1.
8. Переведите ЗГ в режим "АВТОМАТ". Зарисуйте осциллограммы в характерных точках макета.
9. Подключите схему включения ошибки и по указанию преподавателя внесите ошибку в кодовую комбинацию. Зарисуйте осциллограмму кодограммы с внесенной ошибкой и без нее.
10. Переведите ЗГ в однократный режим.
11. Подключите к макету кодирующего устройства макет декодера.
12. Соберите схему декодирования, отвечающую вашему заданию.
13. Нажимая кнопку "Пуск" макета декодера, проверьте правильность работы схемы декодирования, сопоставив содержимое ее регистра с таблицей, построенной согласно п.6 раздела 6.1.
14. Переведите ЗГ в режим «АВТОМАТ». Нарисуйте осциллограммы напряжений в характерных точках схемы декодирования и схемы управлений. Объясните по осциллограммам работу макета.
15. Выключите установку и приведите в порядок рабочее место.

8. СОДЕРЖАНИЕ ОТЧЕТА

1. Результаты самостоятельной подготовки.
2. Осциллограммы с указанием точек схемы, где они сняты.
3. Результаты измерений, оформленные в виде таблиц, графиков с указанием размерности значений постоянных параметров.
4. Выводы о результатах работы.

Примечание. Отчет оформляется каждым студентом самостоятельно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976. 594 с.
2. Шварцман В.О., Емельянов Г. А. Теория передачи дискретной информации. М.: Связь, 1979. 424 с.
3 Темников Ф.Е., Афонин В. А., Дмитриев В.И. Теоретические основы информационной техники. М.: Энергия, 1979. 512 с.

СОДЕРЖАНИЕ

Кодирование и декодирование циклических кодов

Составители: Е з е р с к и й Виктор Витольдович

Е г о р о в Алексей Владимирович

Редактор Н.А. Орлова

Корректор С.В. Макушина

Подписано в печать 28.04.14. Формат бумаги 60x84 1/16.

Бумага газетная. Печать трафаретная. Усл. печ. л. 1,75.

Тираж 100 экз. Заказ 2835

Рязанский государственный радиотехнический университет.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.