

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Радиоуправления и связи»

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
по дисциплине

«Методы первичного кодирования в телекоммуникациях»
Специальность подготовки – 11.04.02 «Инфокоммуникационные технологии и
системы связи»

Специальность (профиль) подготовки
Интеллектуальные системы и сети телекоммуникаций

Квалификация выпускника – магистр
Форма обучения – очная

Рязань 2025г.

4770

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
РИЯНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ

ЦИКЛИЧЕСКИХ КОДОВ

Методические указания к лабораторным работам

Рязань 2014

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

Изложены основные понятия и определения алгебраической теории кодирования и показано ее 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 материале, индивидуальные задания и программы двух лабораторных работ.

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

Табл. 6. Илл. 11. Библиогр.: 3 назв.

Исходников кодирования, циклические коды, декодирующие механизмы, кодирующие, декодирующие, дистанционные устройства, кодирующие устройства, декодирующие устройства

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета

Рецензент: кафедра радиоуправления и связи Рязанского государственного радиотехнического университета (д-р, профессор д-р тех. 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 любого кодирования заключается во введении и передаче сообщения на бытовом уровне информации по правилам, которые известны на передающей и приемной сторонах.

Циклические коды - наиболее распространенный тип бытовых кодов, используемый для повышения достоверности передачи цифровой информации, по линиям связи. Циклические коды имеют блочную структуру и относятся к систематическим разделимым кодам. Каждый блок передаваемой информации состоит из информационной и проверочной частей. 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 $a_i, i=1, 2, \dots, k$; $b_j, j=1, \dots, n-k$ - соответственно символы информационной и проверочной частей блока. Тогда кодовая комбинация систематического кода может быть представлена в виде:

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

где $a_1, a_2, \dots, a_k$ - информационная часть, $b_1, b_2, \dots, b_{n-k}$ - проверочная часть. Проверочная часть b определяется линейным соотношением:

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

здесь знак $\oplus$ означает сложение по модулю два; $c_{ij} \in \{1, 0\}$ - произвольный вектор для j -го проверочного символа. Для полного определения систематического кода необходимо расшифровать $(n-k)$ произвольных и векторов, образующих и произвольную матрицу $\|c_{ij}\|$. Каждая строка этой матрицы есть соответствующий произвольный вектор.

Все строки произвольной матрицы могут быть получены циклическим сдвигом одной из них, называемой образующей данного кода. Основная проблема при разработке экономических и удобных для реализации методов кодирования - нахождение производящей матрицы или образующей 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 комбинированные циклического кода представляются в виде многочлена фиксированной переменной x :

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

В случае двичного кода коэффициенты этого полинома могут принимать только два значения: 0 и 1. В такой записи, например, восьмичленное комбинированное 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 кода, в которой все символы кода представлены коэффициентами и многочлена, причем старшим символом кода соответствуют старшие коэффициенты многочлена.

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

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

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

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

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

где $r(x)$ — 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 $P(x)$. Следовательно эта операция обозначается так:

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

Каждый многочлен $G(x)$ по модулю полинома $P(x)$ называется остатком от деления $G(x)$ на $P(x)$. Простое произведение $G(x)F(x)$ на полином $P(x)$ по модулю $P(x)$: $[G(x)F(x)] = R_{P(x)}[G(x)F(x)]$.

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 кодовым полем или полем Галуа. Поле многочленов замкнуто относительно операции сложения и умножения по модулю неприводимого

многочлена в том смысле, что произведение указанных операций к двум элементам поля дает результат, также принадлежащий полю. Обозначение поля: $GF(2^n)$, где 2^n — порядок поля, n — степень кода.

Таким образом, поле Галуа $GF(2^n)$ — элемент и которого является полином степени не выше $n-1$, описывает все коды кодами кода значности n . Для образования избыточного кода способного обнаруживать и исправлять ошибки, возникающие при передаче, необходимо сократить число используемых кодов, сохранив значность кода. Эти коды называются разрешенными, их число меньше 2^n . Остальные коды называются запрещенными. Выделение разрешенных кодовых комбинаций можно осуществлять разными способами. При этом образованный избыточный код обладает высокой 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, попарном сравнении. В случае двичного кода кодовое расстояние находят путем суммирования по модулю двух кодовых комбинаций и подсчета числа ненулевых единиц.

Для обнаружения ошибок необходимо выдержать условие: $d \geq t_0 + 1$, где t_0 — кратность обнаруживаемых ошибок.

Для исправления ошибок необходимо, чтобы расстояние от принятого кода с ошибками и запрещенной комбинации до переданной было меньше, чем до любой другой разрешенной. Это возможно, если: $d \geq t_0 + 1$, где t_0 — кратность исправляемых ошибок. В этих выражениях t_0 и t_0 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. Выделенные таким образом коды называются идеалами, в $g(x)$ — порождающим многочленом идеала.

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

5. Циклическим называется такой полиномиальный код, который обладает следующими свойствами: если комбинация $\bar{A} = (a_0, a_1, \dots, a_{n-1})$ — разрешенная комбинация циклического кода, то и комбинация $\bar{A}' = (a_{n-1}, a_0, \dots, a_{n-2})$ — также разрешенная комбинация этого кода. Представим комбинации $\bar{A}$ и $\bar{A}'$ в виде многочленов $A(x)$ и

$\bar{d}(x)$, необходимо установить связь между ними в следующем виде (не забываем о прямых приведении подобных членов по модулю два)

$$\bar{d} = x d(x) + a_{n-1}x^n + a_{n-2}x^{n-1} + \dots + a_0(x^n + 1). \quad (1)$$

Пусть $d(x)$ принадлежит идеалу порожденному многочленом $g(x)$. Для того чтобы циклический сдвиг этой комбинации также принадлежал этому идеалу, необходимо деление многочлена $\bar{d}(x)$ 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.

$$R_0(x) | x^n + 1 = 0. \quad (2)$$

Поделим $g(x)$, являющийся делителем двучлена $x^n + 1$, называемся порождающим многочленом циклического кода. При этом степень $\deg(g(x)) = n - k$, где k — число информационных символов. Свойство (2) является обязательным при выборе порождающего многочлена циклического кода.

2.2. Выбор порождающего многочлена циклического (и, к) кода

Выбор порождающего многочлена определяет корректность способности кода. При этом необходимо учитывать конечное количество кодов. 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 деления разрезанной кодовой комбинации на порождающего многочлена без остатка. Неудачной 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 выбрать такие порождающие многочлены, которые при делении ошибочных комбинаций порождают количество различных остатков, соответствующее с числом возможных ошибок. Многоточие, обладающие такими свойствами, называют примитивными. Существует два способа нахождения порождающего многочлена.

А. По заданным n и k

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

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

При этом возможны следующие многочлены:

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

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

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

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

$$g_7(x) = x^7 + 1.$$

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

Двучлен $x^{25} + 1$ разлагается на множители следующим образом:

$$x^{25} + 1 = (x + 1)(x^2 + x + 1)(x^4 + x^3 + 1)(x^4 + x^3 + x^2 + x + 1). \quad (4)$$

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

Б. По заданным n и d

В этом случае используется методика, разработанная Боулом, Чоудхури и Хокингом. Код, построенный по этой методике, получил название кодов БЧХ. Порождающий многочлен для кода БЧХ определяется равенством

$$g(x) = HOK[m_1(x), m_2(x), \dots, m_{r-d-2}(x)],$$

где НОК — наименьшее общее кратное, $r = 0$ при четном d и $r = 1$ при нечетном d , $m_i(x)$ — так называемые минимальные функции, или минимальные многочлены. Минимальные функции являются неприводимыми многочленами и имеют вид [1]:

$$\text{— для } n = 7$$

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

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

$$\text{— для } n = 15$$

$$m_0(x) = x + 1, \quad m_1(x) = x^4 + x + 1, \quad m_2(x) = x^4 + x^2 + 1, \quad m_3(x) = x^4 + x^3 + x^2 + x + 1,$$

$$m_4(x) = x^4 + x^3 + x^2 + x + 1, \quad m_5(x) = x^4 + x + 1, \quad m_6(x) = x^4 + x^2 + x + 1,$$

$$m_7(x) = x^4 + x^3 + x^2 + x + 1, \quad m_8(x) = x^4 + x + 1, \quad m_9(x) = x^4 + x^2 + x + 1,$$

$$m_0(x) = x^6 + x^5 + x^2 + x + 1, \quad m_{10}(x) = x^7 + x + 1, \quad m_1(x) = x^4 + x^3 + 1, \\ m_7(x) = x^4 + x^3 + x^2 + x + 1, \quad m_3(x) = x^4 + x^3 + 1, \quad m_4(x) = x^4 + x^3 + 1.$$

Рассмотрим пример. Пусть требуется построить циклический код с тремя информационными символами ($k=3$) и заданным расстоянием $d=4$. Выясним пороговый многочлен в рассматриваемом примере. Так как d — нечетное значение, примем $r=0$, тогда:

$$g(x) = \text{НОК}[m_1(x), m_3(x), m_4(x)] = m_3(x)m_4(x) = x^4 + x^3 + x^2 + 1 = g_3(x).$$

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

$$g(x) = \text{НОК}[m_1(x), m_3(x)] = m_1(x) = x^3 + x + 1 = g_2(x).$$

2.3. Принципы кодирования и декодирования циклических кодов

Кодирование циклических кодов можно выполнить тремя способами.

Первый способ кодирования основан на правиле:

$$P_j(x) = J_j(x)g(x). \quad (5)$$

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

Второй способ образования кодов — циклический код основан на соотношении

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

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

Пример 1.

Пусть кодированию подлежат $J_1(x) = x+1$ (011) и $J_2(x) = x^2$ (100).

Пороговой многочлен $g(x) = g_3(x)$.

Согласно (5) имеем:

$$K_1(x) = x^5 + x^2 + x + 1 \quad (010011)x$$

$$K_2(x) = x^5 + x^3 + x^2 + x^2 \quad (1110100).$$

Правило (6) дает:

$$K_3(x) = x^4(x+1) + x^5 + x^2 + x^2 + x^2 + x \quad (0111010)x$$

$$K_4(x) = x^4x^2 + x^5 + x^2 + x^2 + x^2 + x^2 + x \quad (1001110)x$$

Видно, что правило (5) приводит к неразделенному коду, а правило (6) — к разделенному, 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 ин-

формационные символы связаны линейным соотношением. Для осуществления этого метода кодирования необходимо представить комбинированный циклический код 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) и прежние обозначения или a_{n-k-j} и новый b_j занимают в кодовой рамке $(n-k-j)$ -ю позицию, определяется рекуррентным соотношением:

$$a_{n-k-j} = \sum_{i=0}^{k-1} a_{n-k-j} b_i, \quad j = 1, 2, \dots, n-k. \quad (7)$$

где b_j — коэффициенты генераторного (проверочного) многочлена.

$$h(x) = \frac{x^n + 1}{g(x)} = b_0 + b_1x + \dots + b_kx^k.$$

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

$$b_1 = a_1 = \sum_{i=0}^2 a_{n-k-i} b_i = a_6 b_0 \oplus a_5 b_1 \oplus a_4 b_2 = 0 \oplus 1 \oplus 1 \oplus 1 = 1,$$

$$b_2 = a_2 = \sum_{i=0}^2 a_{n-k-i} b_i = a_5 b_0 \oplus a_4 b_1 \oplus a_3 b_2 = 1 \oplus 1 \oplus 1 \oplus 1 = 0.$$

Из примера видно, что проверочные символы a_1, a_2, a_3, a_4 соотносятся с соответствующим символом и 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 раздельных 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 сов-

дают, то ошибка отсутствует. В противном случае фиксируется новая ошибка.

Третий способ, используемый для прецизионных кодов, основан на свойствах деления многочленов, отбрасывающих разрядные коды-граммы, на порождающий многочлен без остатка. Если остаток от деления нулевой, то ошибок нет. В противном случае принятая кодовая рамка является заперженной (имеет место ошибка). Исправление ошибок осуществляется путем анализа полученного остатка.

2.4. Принципы построения делительных устройств

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

многочлен $g(x) = x^2 + x + 1$.

Вычисляем частное и остаток, используя алгоритм Эвклида:

$$r_1(x) = \frac{x^4 + x^3 + 0 \cdot x^2 + x^3 + x + 1}{x^2 + x + 1}$$

$$\oplus \frac{x^4 + 0 \cdot x^3 + x^3 + x^2}{x^2 + x + 1} = B_1(x)$$

$$r_4(x) = \frac{x^4 + x^3 + 0 \cdot x^2 + x^3 + 1}{x^2 + 0 \cdot x^3 + x^2 + x} = B_4(x)$$

$$\oplus \frac{x^4 + 0 \cdot x^3 + x^2 + x}{x^2 + 0 \cdot x^3 + x^2 + x + 1} = B_5(x)$$

$$r_5(x) = \frac{x^3 + x^2 + 0 \cdot x + 1}{x^2 + 0 \cdot x^3 + x^2 + x + 1}$$

$$\oplus \frac{x^3 + 0 \cdot x^2 + x + 1}{x^2 + x}$$

$$r_6(x) = \frac{x^2 + x}{x^2 + x} \quad (8)$$

В устройстве деления последовательно реализуются все операции этого алгоритма. Основной устройством деления является регистр сдвига с логическими обратными связями. Исходное состояние регистра нулевое. Число ячеек памяти определяется степенью полинома делителя. Многочлен - делитель поступает на вход регистра, начиная со старшего коэффициента. Деление начинается после того, как этот старший коэффициент достигнет последней ячейки памяти регистра. В рассматриваемом примере после трех тактов работы регистр содержит $r_3(x) = x^3 + x^4 + 0 \cdot x^2$ [см. алгоритм деления (8)]. После 4-го такта

коэффициент при x^3 поступает на выход схемы деления, а в первую ячейку регистра заносится коэффициент при x^2 . Таким образом, содержимое регистра теперь

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

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 полным $r_1(x) = x^4 + x^3 + 0 \cdot x^2$ алгоритма (8). Очевидно, что для этого необходимо изменить содержание первой и второй ячеек регистра на противоположное. Изменение содержания этих ячеек эквивалентно сложению по модулю 2 полного (9) и многочлена $B_1(x) = 0 \cdot x^4 + x^3 + x^2$, формируемого ЛОС:

$$\oplus \frac{x^4 + 0 \cdot x^3 + x^2}{0 \cdot x^4 + x^3 + x^2} = B_1(x)$$

$$x^4 + x^3 + 0 \cdot x^2 = r_2(x).$$

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

$$x^5 + 0 \cdot x^4 + x. \quad (10)$$

В цепи ЛОС формируются полным $B_5(x) = 0 \cdot x^5 + x^2 + x$ и при суммировании 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, и 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 составить схему деления по первому остатку $r_4(x)$. Схема делительного устройства показана на рис. 1, где обозначено: $\square$ - ячейка памяти, $\oplus$ - сумматор по модулю 2.

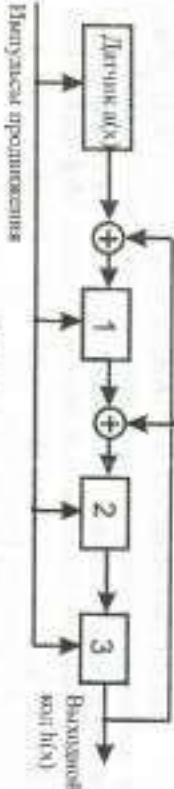


Рис. 1

Основные правила построения делительных устройств:

- 1) устройство деления многочленов есть регистр сдвига с логическими и обратными связями от выхода к входу и с ячейкой ячеек памяти, равным степени делителя;
- 2) ячейки памяти образуются от входа к выходу, начиная с единицы;

- 3) сумматоры по мод2 своим и первым входам и подключаются к выходам тех ячеек памяти, номера которых совпадают со степенью ведущих коэффициентов многочлена делителя, кроме последней ячейки;
- 4) выход последней ячейки подключается ко вторым входам всех сумматоров по мод2;
- 5) выходы сумматоров по мод2 подключаются к входам следующих ячеек памяти.

2.5. Кодирование устройств численных кодов

В разделе 2.3 установлены два алгоритма кодирования, используемые на практике. Первый из них, определяемый выражением (6), требует выполнения умножения информационного многочлена $J_i(x)$ на делитель x^{n-k} и вычисления остатка от деления произведения $x^{n-k} J_i(x)$ на порождающий многочлен $g(x)$. Умножение на один член x^{n-k} не требует специального устройства, так как такое умножение означает приращивание $n-k$ нулей со стороны младших разрядов к кодовой строке кодовой комбинации. Эта операция выполняется за счет организации работы датчика $J_i(x)$. Вычисление остатка осуществляется по схеме деления на $g(x)$. Информационные символы с приращиванием $n-k$ нулей и поступают, начиная со старшего разряда, одновременно в схему деления и в канал связи. Регистр схемы деления будет содержать остаток, т.е. проверочные символы передаваемой кодовой комбинации. Для вывода их из регистра требуется $n-k$ тактов. Недостаток этого кодера состоит в том, что форма нулевой кодовой комбинации о кода имеет разрыв между информационными и проверочными символами и $n-k$ символов, что снижает скорость передачи информации. Недостаток устраняется использованием специальной схемы деления [1].

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

После сброса обратной связи построена такая схема, что на ее выходе формируется первый проверочный символ

$$a_{n-k-1} = \sum_{i=0}^{k-1} a_{n-i-1} b_i = a_{n-1} b_0 \oplus a_{n-2} b_1 \oplus \dots \oplus a_{n-k} b_{k-1}.$$

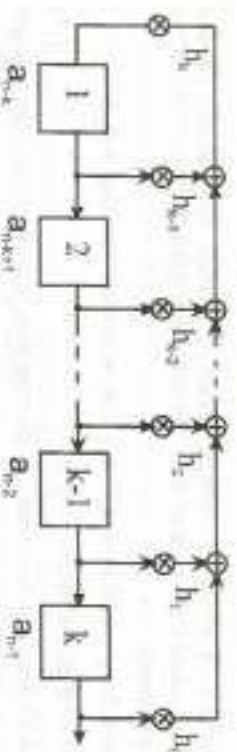


Рис. 2

После поступления первого информационного продолжения (на рис. 2 не показан) этот символ записывается в первую ячейку, а информационные символы смещаются на одну ячейку влево. Теперь на выходе ЛОС действует второй проверочный символ:

$$a_{n-k-2} = \sum_{i=0}^{k-1} a_{n-i-2} b_i = a_{n-2} b_0 \oplus a_{n-3} b_1 \oplus \dots \oplus a_{n-k-1} b_{k-1}.$$

На последующих тактах работы последовательно формируются оставшиеся проверочные символы и затем весь поток. Рассмотрим работу кодирующего устройства на примере кодера для кода (7.3).

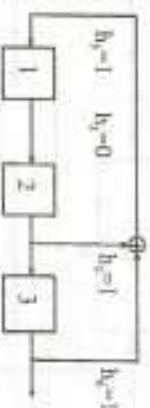


Рис. 3

порождающего многочлена $g(x) = x^4 + x^3 + x + 1$. Генераторный многочлен при этом таков: $h(x) = x^3 + x + 1$. Соответствующее кодирующее устройство приведено на рис. 3.

Таблица 1

N	ДП	2	3
0	a_4	a_5	a_6
1	$a_5 = a_4 \oplus a_0$	a_6	a_7
2	$a_6 = a_5 \oplus a_1$	a_7	a_8
3	$a_7 = a_6 \oplus a_2$	a_8	a_9
4	$a_8 = a_7 \oplus a_3$	a_9	a_{10}
5	$a_9 = a_8 \oplus a_4$	a_{10}	a_{11}
6	$a_{10} = a_9 \oplus a_5$	a_{11}	a_{12}
7	$a_{11} = a_{10} \oplus a_6$	a_{12}	a_{13}

В исходном состоянии содержимым регистра являются $a_4, a_5, \dots, a_6$. Из схемы видно, что проверочный символ есть сумма по мод2 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 формируется периодический повторяющийся код

циклического кода. Из таблицы также установим, что каждый символ циклического кода может быть представлен разными линейными соотношениями:

$$\begin{aligned} a_6 &= a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5; & a_5 &= a_2 \oplus a_4 \oplus a_6 \oplus a_1; \\ a_4 &= a_1 \oplus a_6 \oplus a_2 \oplus a_3 \oplus a_5. \end{aligned} \quad (11)$$

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 кодовыми по каналу декодера. Команда на повторную передачу передается на передающей конец канала связи по обратному каналу.

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

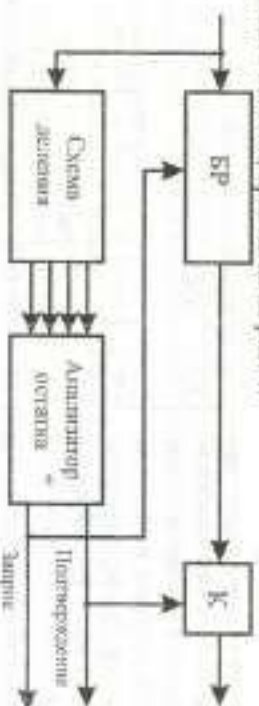


Рис. 4

Принцип работы устройства основан на проверке основного свойства кодовым циклического кода - делении на порождающий многочлен. Принятая кодовая последовательность на входе схемы деления и одновременно записывается в буферный регистр (БР). Когда вся кодовая последовательность записана в БР, в схеме деления образуется остаток от деления кодовыми на порождающий многочлен, называемый синдромом.

$$S(x) = S_0x^{n-k} + S_1x^{n-k-1} + \dots + S_{k-1}.$$

Если ошибки отсутствуют, то регистр схемы деления содержит один нуль (нулевой синдром, т.е. $S_0 = S_1 = \dots = S_{n-k} = 0$). Анализатор синдрома выделяет сигнал "Подтверждение", по которому разрешается выдвигать кодовыми потребителю. В противном случае выдвигается сигнал "Запрос", который стирает содержимое буферного регистра.

Более сложное реализуется декодером 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 и многочлена $K_0(x)$ на порождающий многочлен $g(x)$ называется 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 синдромом и номером ошибки, так имеет место ошибка. Структура декодера показана на рис. 5.

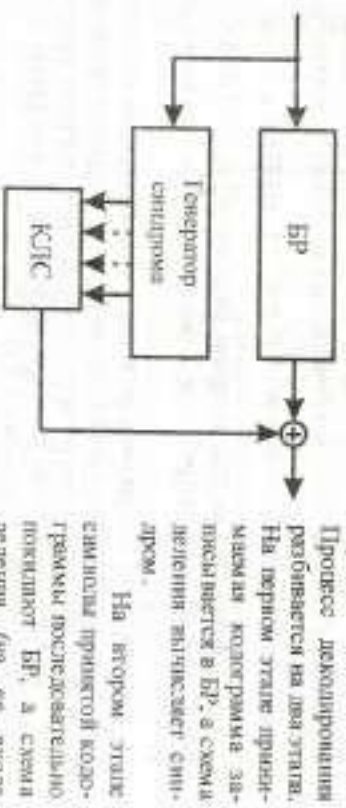


Рис. 5

При заданном синдроме. Если синдром отличен от нуля, то на входе второго этапа содержимое регистра схемы деления есть разные двоичные числа разрядности $n-k$. Результат работы схемы деления на очередном этапе 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 реализовать 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 и старшем разряде (по крайней мере разряд, расположенный в выходной ячейке БР). Так как кодовыми циклического кода делится на $g(x)$ без остатка, то, очевидно, что опознаватель есть остаток от деления x^n на

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

Рассмотрим пример, иллюстрирующий принцип построения КЛС, исправляющей ошибки в комбинациях двоичного (7,4) кода с образующим многочленом $g(x) = x^3 + x + 1$. Означительный шифр,

разделяет x^7 на $g(x)$. Он равен x^6 (001). Схема деления, выполняемая синхронно, построенная по правилам разд. 2.6, приведена на рис. 1. Пусть на вход этой схемы поступает кодовое слово двоичного 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 принятое двоичное слово, 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 потребуются 7 тактов работы делительного устройства.

Рассмотрим содержание регистра схемы деления на последующих тактах работы при условии, что на ее вход поступает только нуль. В исходном состоянии регистр схемы деления содержит синдром ("1" в ячейке 3). Состояние регистра на дополнительных тактах работы отражено в табл. 2 (столбцы 1, 2, 3).

Таблица 2

ЯИ			
Такт	1	2	3
0	0	0	1
1a	0	0	1
1b	1	1	0
2a	0	1	1
2b	0	1	1
3a	0	0	1
3b	1	1	1
4a	0	1	1
4b	1	0	1
5a	0	1	0
5b	1	0	0
6a	0	1	0
6b	0	1	0
7a	0	0	1
7b	0	0	1

Каждый такт разделение на два: содержимое ячеек на подплате "а" вычисляется без учета ЛОС, на подплате "б" учитывается действие ЛОС. В последней колонке записан символ, действующий на выходе делительного устройства и одновременно на входе ЛОС. Отметим, что для выхода ошибочного символа на БР требуется 5 разрядов тактов (колов рамка в(х) записывается в БР начиная со старшего разряда). Из таблицы можно видеть, что описанная 100 содержится в регистре схемы деления на 5-м такте, т.е. в момент выхода ошибочного символа из БР.

Примеры, выполняются для этого условия для случая ошибки в 5-м разряде, т.е. $a(x) = x^6 + x^5 + x^3 + x^2 + 1$ (1111001). Синдром $S(x) = x^2 + x + 1$ (110). Из табл. 3 можно видеть, что опознаватель 100 содержится в регистре схемы деления на 3-м до-

полнительном такте и на этом же такте ошибка выводится из БР.

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

Таблица 3

ЯИ			
Такт	1	2	3
0	0	0	1
1a	0	0	1
1b	1	1	1
2a	0	1	1
2b	1	0	1
3a	0	1	0
3b	1	0	0
4a	0	1	0
4b	0	1	0

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

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

Система контрольных проверок вида (11), построенная для одного символа a_i двоичного кода, может быть использована для декодирования всех символов этой комбинации. Действительно, контрольные проверки удовлетворяют любой комбинации двоичного кода, а следовательно, и кодовым, полученные двоичные и проверочные символы. Таким образом, для декодирования символа a_i достаточно произвести / сдвиги принятой комбинации, не изменяя их схемы вычисления проверочных соотношений, на макоритарного элемента.

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

$$a_6 = a_7 \oplus a_4 = a_1 \oplus a_2 \oplus a_3; \quad a_5 = a_1 \oplus a_2 \oplus a_4 = a_3 \oplus a_7; \quad a_4 = a_7 \oplus a_4 = a_1 \oplus a_2 \oplus a_3 \oplus a_5 \oplus a_6. \quad (12)$$

Макоритарный декодер МД-1 первой разновидности представляет собой БР, дополненный устройством и, выполняющим и проверки (12) относительно какого-либо одного символа (например, a_4), и макоритарным элементом (МЭ). При этом используется и триггерная проверка (вид $a_4 - a_4$). Схема декодера приведена на рис. 6. МЭ выдает решение о значении проверочного символа по большинству результатов проверок, действующих на его входах. Если результаты проверок даются 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, дупликации).

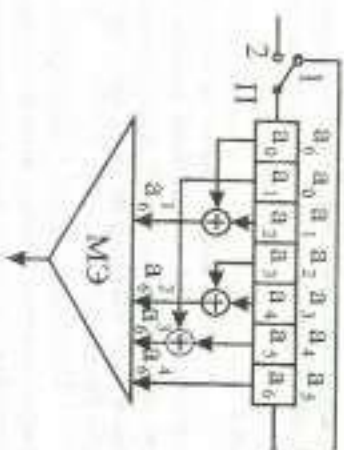


Рис. 6

В практических схемах место МЭ используют пороговый элемент (ПЭ), который не фиксирует появление несправильной ошибки.

Для проверки остальных символов кода достаточно привести циклическую перестановку заданной кодовой таблицы. С этой целью переключатель 2 устанавливается в положение 1 и подается на нулевой элемент.

На рис. 6 приведен результат такого сдвига на один такт. Теперь декодер реализует следующие соотношения: $a_0 \oplus a_6 \oplus a_2 \oplus a_4 \oplus a_1$. Сравнив их с (12), можно убедиться, что декодер осуществляет проверку относительно символа a_7 . Таким образом, для проверки и исправления информационных символов потребуются 4 такта работы.

В риске отрисовки примем известна 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 кодовой таблицы) не могут быть исправлены (голоса разделились поровну), но будут обнаружены.

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

Максимальный декодер МД-2 использует тот факт, что элемент синдрома есть сумма по модулю 2 определенных символов принятой комбинации. Действительно, пусть для некоторого циклического кода генератор синдрома представлен на рис. 7.

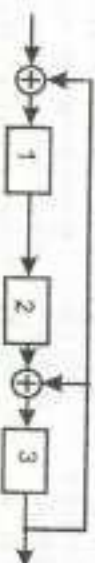


Рис. 7

Формирование синдрома при поступлении кодовой таблицы

$a(x) = a_6x^6 + a_5x^5 + a_4x^4 + \dots + a_1x + a_0$ отображается в табл. 4.

На 7-м такте работы схема образует синдром (S_1, S_2, S_3).

Таблица 4

ЛП	1	2	3	Вых
Такт	1	2	3	Вых
3	a_6	a_5	a_4	
4а	a_3	a_2	a_1	a_0
4б	$a_7 + a_6$	a_4	$a_3 + a_6$	
5а	a_2	$a_3 + a_6$	a_4	$a_5 + a_6$
5б	$a_2 + a_1 + a_6$	$a_3 + a_6$	$a_4 + a_3 + a_6$	
6а	a_1	$a_3 + a_6$	$a_4 + a_3$	$a_5 + a_1 + a_6$
6б	$a_1 + a_4 + a_3 + a_6$	$a_3 + a_6$	$a_4 + a_3 + a_6$	
7а	a_0	$a_1 + a_4 + a_3 + a_6$	$a_3 + a_6$	$a_4 + a_1 + a_6$
7б	$a_0 + a_1 + a_4 + a_6$	$a_1 + a_4 + a_3 + a_6$	$a_3 + a_6$	$a_4 + a_1 + a_6$
8а	0	$a_2 + a_3 + a_4 + a_6$	$a_1 + a_4 + a_3 + a_6$	$a_5 + a_2 + a_1 + a_6$
8б	$a_2 + a_3 + a_4 + a_6$	$a_1 + a_4 + a_3 + a_6$	$a_3 + a_6$	$a_4 + a_2 + a_1 + a_6$

Примечание: табл. 4 не имеет:

$S_1 = a_0 \oplus a_2 \oplus a_4 \oplus a_6$, $S_2 = a_1 \oplus a_3 \oplus a_5 \oplus a_7$, $S_3 = a_2 \oplus a_4 \oplus a_6 \oplus a_8$.

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

$S_1 + S_2 = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_8$.

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

Если же имеет место ошибка в любом другом разряде, то равным и единице оказываются только две суммы из четырех. Это обстоятельство можно использовать для исправления ошибки с помощью схемы, приведенной на рис. 8. На вход ПЭ подается четыре суммы: S_1, S_2, S_3, S_4 . Если проверочный первый 7-й разряд ошибочен, то все эти суммы равны единице и на выходе ПЭ появится "1" ($4 > 3$).

3.2. Заданный генератор и схема формирования пакета импульсов

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

Для удобства наблюдения кодовых слов с помощью осциллографа, а также для обеспечения работы декодирующего устройства в пакете предусмотрена схема формирования пакета импульсов сдвиг. Схема формирует пакеты по 7 или 15 импульсов, раздельных интервалом, равным длительности пакета. Таким образом, кодовое 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 поступает импульсы ЗГ. Таким образом, на выходе схемы "И" формируются пакеты по 7 или 15 импульсов, обеспечивающие работу кодирующего устройства.

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

Таким образом, осуществляется переключением П2. С помощью этого переключателя и показаний "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 сдвиг оказывается укороченным, что полностью дезориентирует работу кодирующего устройства.

Схема установки ошибок состоит из схемы "И" на 5 входов, двух выходной схемы "ИЛИ" и сумматора по мод2; 4 из 5 входов схемы "ИЛИ" через переключатели П3-П6 могут подключаться к первым или пятерым 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 ошибок ставит в соответствие определенный номер позиции, в котором происходит ошибка. Табл. 5 позволяет определить кодовые тумблеров для введения ошибки в определенный разряд кодовой комбинации. Подключение схемы внешнего сброса осуществляется с помощью коммутированных тумблеров и гнезд К15-К22, расположенных на лицевой панели макета.

Таблица 5

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

4. ОПИСАНИЕ МАКЕТА ДЕКОДИРУЮЩЕГО УСТРОЙСТВА

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

- набор триггеров Т1-Т10 с включением между ними сумматоров по мод2 (М2), декодировать составленную схему генератора синхронизации для всех циклических кодов длиной $n=7$ и кодов (15,7), (15,6), (15,5);
- буферный регистр, содержащий 15 триггеров в ячейки памяти (Т11-Т25);

- набор сумматоров по мод2, 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 заданного генератора из макета кодирующего устройства;
- формирование импульсов обнуления генератора синхронизации перед поступлением очередной кодовой комбинации;
- ввод кодовой комбинации из макета кодера;
- формирование сигналов, разрешающих срабатывание схемы обнаружения и исправления ошибок (например, после того, как закончилось формирование синдрома);

Кроме того, в состав макета входит сумматор по мод2, с помощью которого осуществляется исправление ошибок на выходе БР, кода (БЛ), автоматически замыкающий БР в кольцо для осуществления циклических сдвигов, схема "ИЛИ", включаемая на выходе генератора синхронизации для согласования полярности сигналов ЛОС, и сумматор по мод2 на входе генератора синхронизации, через который осуществляется коррекция синдрома.

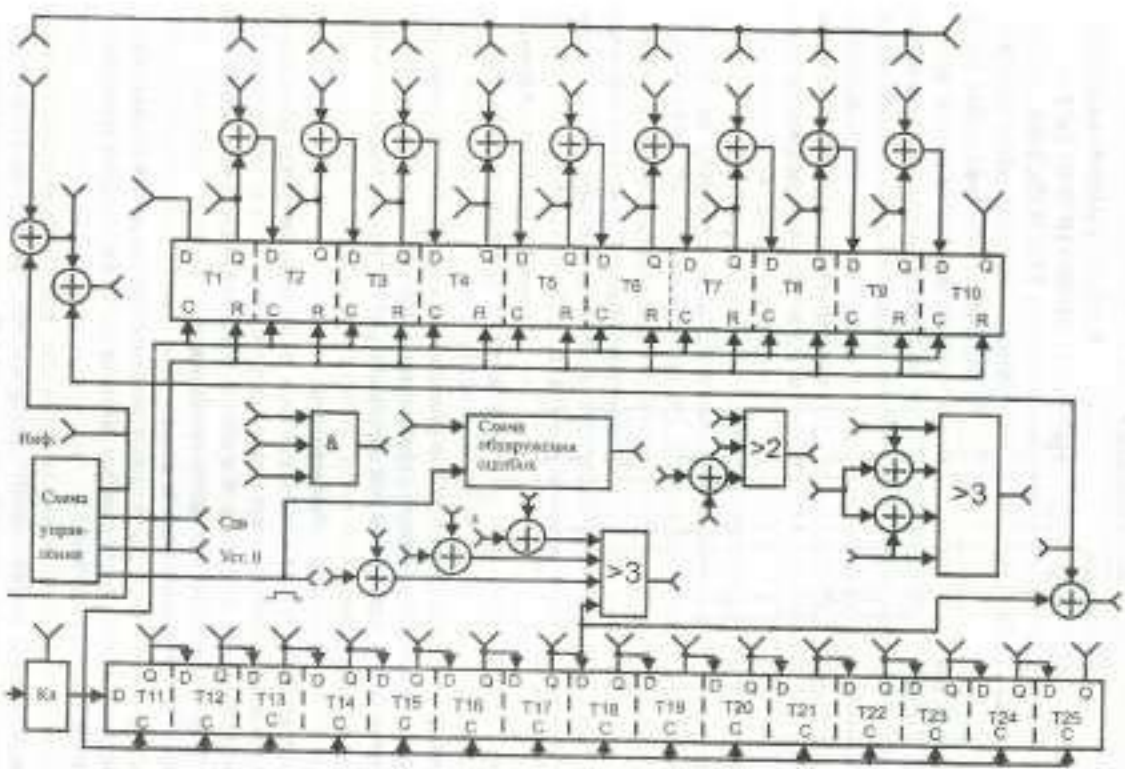


Рис. 10

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

Реализация той или иной схемы декодирования осуществляется с помощью коммутационных гнезд и шнуров. На лицевой панели ма-

кета нанесены функциональная схема и коммутировочные гнезда для подключения осциллографа, что обеспечивает создание декодирующего устройства.

На выходах триггеров через выходные светодиоды, также выведенные на лицевую панель. В однополярном режиме они позволяют контролировать правильность работы собранной схемы декодирования.

5. ВАРИАНТЫ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ

Мастера лабораторной панели обеспечивают реализацию кодирующих и декодирующих устройств для циклических кодов с 7 и 15 сим-волами. Варианты задания сведены в табл. 6.

Таблица 6

Бригады	Параметры КУ		ДКУ	
№1	1	$n=7, k=3$	$g(x)=x^4+x^2+x+1$	МД-2
	2	$n=7, d=4$	ВЧХ	МД-1
	3	$n=15, d=5$	ВЧХ	ДОН
	4	$n=7, k=4$	$g(x)=x^3+x^2+1$	МД-1
№2	1	$n=7, d=3$	ВЧХ	МД-2
	2	$n=15, d=6$	ВЧХ	МД-1
	3	$n=7, d=2$	ВЧХ	ДОН
	4	$n=7, k=3$	$g(x)=x^4+x^2+x+1$	МД-1
№3	1	$n=7, d=4$	ВЧХ	МД-2
	2	$n=7, d=3$	ВЧХ	МД-1
	3	$n=15, k=6$	$g(x)=x^4+x^2+x^2+x^2+x+1$	ДОН
	4	$n=15, k=7$	$g(x)=x^4+x^2+x^2+x+1$	МД-1
№4	1	$n=15, k=7$	$g(x)=x^4+x^2+x^2+x+1$	МД-2
	2	$n=15, k=6$	$g(x)=x^4+x^2+x^2+x^2+x+1$	МД-1
	3	$n=15, d=7$	ВЧХ	ДОН
	4	$n=15, k=5$	$g(x)=x^{10}+x^5+1$	МД-1

В каждой бригаде задание распределяется согласно алфавитному списку бригад. Номер бригады определяется внутри подгруппы. Тип кодирующего устройства (КУ), который следует построить, одинаков для всех вариантов. Он определяется методом кодирующего устройства. Тип декодирующего устройства (ДКУ) указан в соответствующей графе табл. 6 (МД - макротипный декодер, ДОН - декодер с односторонним).

6. ЛАБОРАТОРНАЯ РАБОТА № 4

Подготовка к лабораторной работе.

1. Изучите все разделы настоящего описания, а также соответствующие разделы рекоммендуемой литературы.
2. Постройте 3-4 следующих друг за другом комбинаций десятичного кода, соответствующего вашему варианту задания. Убедитесь, в том, что они переходят друг в друга при циклических сдвигах.
3. Нарисуйте функциональную схему десятичного устройства по образцу и поточному варианту.
4. Нарисуйте функциональную схему кодирующего устройства с ячейкой и нажмите и постройте таблицу (см. табл. 1). Установите проверочные соотношения. Заполните таблицу для кодирования одной из построенных комбинаций комбинаций.
5. Нарисуйте схему декодирующего устройства. Проведите проверку работы схемы на каждый такт. Определите отклонения (для ДКУ с оптимизацией см. табл. 2) или соответствия проверки (для ДКУ МД-2 см. табл. 4).
6. Нарисуйте схему декодирующего устройства, отмечаящую выходу заданно. С оставьте таблицу, отражающую работу декодера на каждый такт для выбранной комбинации.

Нормы выполнения лабораторной работы.

- А. Проверка результатов кодирования циклических кодов на ПЭВМ. Запустите с «Рабочего стола» компьютерную программу СЖСКОД и следуйте ее инструкциям в соответствии со своим вариантом лабораторного задания.
- Б. Исследование работы устройства формирования десятичного кода в системе схемотехнического моделирования МСРО-САР 5.

В системе МСРО-САР 5 доши файл LABCOD.R, содержащий принципиальную схему устройства кодирования десятичных кодов, которая приведена на рис. 11 и состоит из двух универсальных логических элементов сдвигающих регистров с параллельной загрузкой, элементов суммирования по модулю 2, обозначенных на схеме значком $\oplus$, генератора тактовой частоты и генератора импульсов загрузки информации в регистры десятичного кода.

В схеме выполнена основная часть соединений. В зависимости от варианта задания (т.е. требуемой схемы кодера и информации кода) необходимо проанализировать некоторые соединения в принципиальной схеме с помощью редактора схем МСРО-САР. Для этого, войдя в среду МС 5, вызвать через меню ФП.Е файл LABCOD.R.

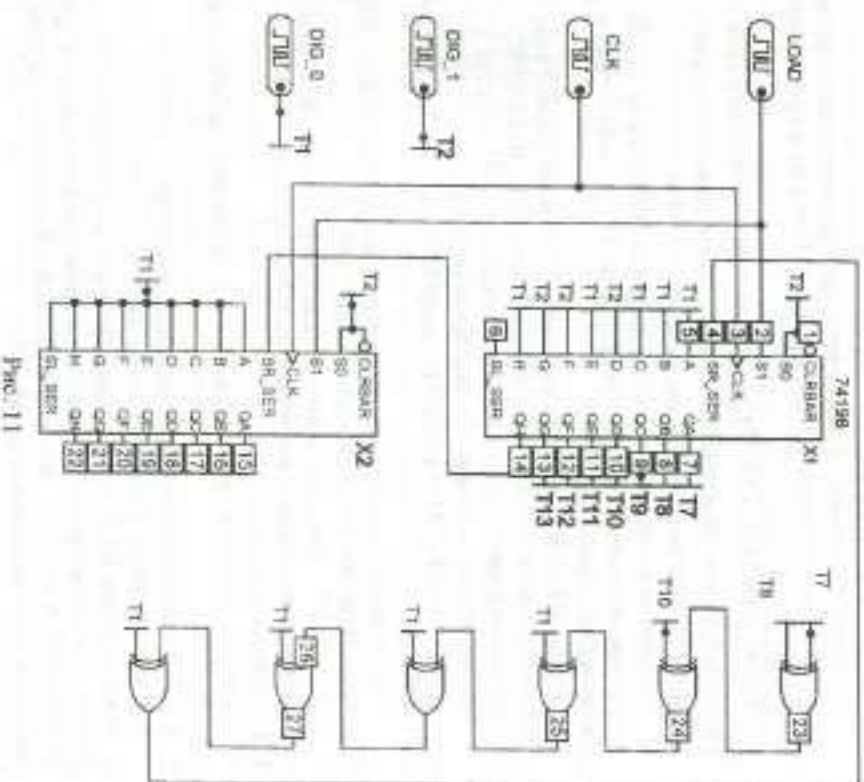


Рис. 11

1. Выполнить соединения в устройстве кодирования, соответствующие конкретной принципиальной схеме, т.е. подключить к схеме суммирования по модулю 2 требуемые выходы регистров 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 $\oplus$.

- на незадействованные входы схемы суммирование подать уровень логического «0», установив значение VALUE при их редактировании равным T1.
- 2. Входя регистр X1 (A - H) в зависимости от входного 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 слева 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 ОК,
 - эту же операцию повторить для всех входов (A - H) регистра X1.
- 3. Провести временной анализ устройства кодирования:
 - войти в меню ANALYSIS и выбрать функцию TRANSCIENT ANALYSIS,
 - в окне задано условий анализа задать значения: Time Range 2000ns, Maxchip Time Step 50ns, Number of point 0,
 - в колонке Y expression набрать имена контрольных точек принципиальной схемы, стоявшие в коробках требуется проследить (например, D(1)D(10) и т.д.),
 - в колонке X expression установить 1 (т.е. исследовать временных зависимостей)
 - в колонке X Range установить значение 20e-007,0,
 - в колонке Y Range установить значение N/A,
 - нажать кнопку RUN и меню анализа, при этом в случае отсутствия ошибок при редактировании схемы и задании условий анализа в окне TRANSCIENT ANALYSIS появится временные диаграммы в характерных точках принципиальной схемы,
 - для изменения условий анализа войдите в меню TRANSCIENT, выберите режим Limit и измените значения временных параметров имени контрольных точек,
 - для изменения соединений в принципиальной схеме закройте окно TRANSCIENT ANALYSIS, соответствующей кнопкой отредактируйте схему, как было описано в п. 1.2.

7. ЛАБОРАТОРНАЯ РАБОТА № 5

Подготовка к лабораторной работе.

1. Изучившая материалы лабораторной работы №4, подготовьте схемы кодирующего и декодирующего устройств.
2. Тщательно изучите функциональные схемы микров кодировщика и декодирующего устройств.
3. Проработайте реализацию логических схем кодирования и декодирования на микретах. Составьте схему коммутации.

4. Составьте методички и подписание эксперта по разделу 7.2. *Порядок выполнения работы.*
- Исследование устройства формирования циклического кода на микретах.
 1. Включите осциллограф и его корпус соедините с корпусом микрета.
 2. Установите переключатели на ливневой панели микрета кодирующего устройства в положение, соответствующее нашему варианту задания.
 3. С помощью коммутационных шнуров соберите кодирующее устройство с к в ч е т в и для исследуемого кода.
 4. По решению преподавателя включите тумблер питания.
 5. Переключите 3Г в однократный режим, обнулите делитель, схемы формирования пакета импульсов и регистр кодирующего устройства.
 6. Введите в регистр одну из комбинаций информационного кода, для которой найден (согласно п. 2 раздела 6.1) проверочные символы, контролируя состояние триггеров по индикаторам (ветвиодим)
 7. Последовательно нажмите кнопку "Пуск", просканируйте правильность работы кодера, сравнив полученное регистром на выходе пакеты работы с таблицей, построенной согласно п.4 раздела 6.1.
 8. Переключите 3Г в режим "AB TOMAT". Завершите осциллограммы в характерных точках микрета.
 9. Подключите схему включения ошибок и по указанно преподавателя внесите ошибку в козовую комбинацию. Запишите осциллограмму кодера с внесенной ошибкой и без нее.
 10. Переключите 3Г в однократный режим.
 11. Подключите к микрету кодирующего устройства микрет декодера.
 12. Соберите схему декодирования, отключившему нашему заданию.
 13. Нажав на кнопку "Пуск" микрета декодера, проверьте правильность работы схемы декодирования, сопоставив содержимое ее регистра с таблицей, построенной согласно п.6 раздела 6.1.
 14. Переключите 3Г в режим «AB TOMAT». Нарисуйте осциллограммы напряжений в характерных точках схемы декодирования и схемы управления. Обсудите по осциллограммам работу микрета.
 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 полученных параметров.
 4. Выводы о результатах работы.
- Примечание. Ответ оформляется каждым студентом самостоятельно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

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

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	1
2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	1
3. ОПИСАНИЕ МАКЕТА КОДИРУЮЩЕГО УСТРОЙСТВА	18
4. ОПИСАНИЕ МАКЕТА ДЕКОДИРУЮЩЕГО УСТРОЙСТВА	21
5. ВАРИАНТЫ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ	23
6. ЛАБОРАТОРНАЯ РАБОТА №4	24
7. ЛАБОРАТОРНАЯ РАБОТА №5	26
8. СОДЕРЖАНИЕ ОТЧЕТА	27

Копирование и доводирование циклических кодов

Составители: Езерский Виктор Витольдович

Егоров Алексей Владимирович

Редактор Н.А. Орлова

Корректор С.В. Мамулина

Подписано в печать 28.04.14. Формат бумаги 60х84 1/16.

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

Тираж 100 экз. Заказ 2335

Издательский государственный радиотехнический университет,

390005, Рыбин, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ