



**Кафедра  
проектирования радиоэлектронных систем  
летательных аппаратов**

**В.И. Шульгин**

**Основы теории передачи  
информации**

**Часть 2  
Помехоустойчивое кодирование**

Учебное пособие

<http://k501.xai.edu.ua/>

*МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ*

Национальный аэрокосмический университет  
им. Н.Е. Жуковского  
« Харьковский авиационный институт »

В.И. Шульгин

**Основы теории передачи  
информации**

Часть 2

**Помехоустойчивое кодирование**

Учебное пособие

Харьков “ХАИ” 2003

Основы теории передачи информации. Ч. 2. Помехоустойчивое кодирование / В.И. Шульгин.-Учеб. пособие. – Харьков: Нац. аэрокосм. ун-т « Харьк. авиац. ин-т » , 2003. - 87 с.

Изложены основы теории помехоустойчивого кодирования информации в системах связи. Рассмотрены наиболее широко распространенные в настоящее время методы кодирования – с использованием блочных и сверточных линейных кодов, а также методы их декодирования.

Приведено большое число практических задач, позволяющих закрепить изучаемый материал.

Для студентов, изучающих курсы “Основы теории связи”, “Основы теории передачи информации”, “Системы передачи информации”.

Рис. 31. Табл.11. Библиогр.: 17 назв.

Рецензенты: д-р техн. наук, проф. В.В. Пискорж,  
д-р физ.-мат. наук, проф. А.В. Мартыненко

# Оглавление

1. Основы помехоустойчивого кодирования .....	3
1.1. Основные принципы. Типы кодов .....	3
1.2. Линейные блочные коды.....	5
1.2.1. Код с проверкой на четность .....	6
1.2.2. Итеративный код .....	9
1.2.3. Порождающая матрица линейного блочного кода .....	10
1.2.4. Проверочная матрица.....	14
1.2.5. Дуальные коды.....	15
1.2.6. Синдром и обнаружение ошибок.....	16
1.2.7. Синдромное декодирование линейных блочных кодов .....	18
1.2.8. Мажоритарное декодирование линейных блочных кодов .....	21
1.2.9. Декодирование методом максимального правдоподобия .....	23
1.2.10. Вес и расстояние Хемминга. Способность кодов обнаруживать и исправлять ошибки .....	27
1.3. Полиномиальные коды .....	32
1.3.1. Циклические коды .....	34
1.3.2. Кодирование с использованием циклических кодов .....	34
1.3.3. Вычисление синдрома и исправление ошибок в циклических кодах .....	39
1.3.4. Неалгебраические методы декодирования циклических кодов.....	41
2. Сверточные коды .....	45
2.1. Кодирование с использованием сверточных кодов .....	45
2.2. Синдромное декодирование сверточных кодов .....	49
2.3. Кодовое дерево и решетчатая диаграмма .....	51
2.4. Декодирование сверточных кодов. Алгоритм Витерби .....	53
2.5. Алгоритмы поиска по решетке.....	56
3. Применение корректирующего кодирования в системах связи .....	58
3.1. Каскадные коды.....	58
3.2. Кодирование с перемежением .....	59
4. Задачи и практические вопросы к курсу .....	62
Библиографический список .....	85

# 1. Основы помехоустойчивого кодирования

Итак, мы рассмотрели основы экономного кодирования данных, или кодирования источника в системах передачи информации. Задача кодера источника – *представить подлежащие передаче данные в максимально компактной и, по возможности, неискаженной форме.*

При передаче информации по каналу связи с помехами в принятых данных могут возникать ошибки. Если такие ошибки имеют небольшую величину или возникают достаточно редко, информация может быть использована потребителем. При большом числе ошибок полученной информацией пользоваться нельзя.

Для уменьшения количества ошибок, возникающих при передаче информации по каналу с помехами, может быть использовано кодирование в канале, или *помехоустойчивое кодирование.*

Возможность использования кодирования для уменьшения числа ошибок в канале была теоретически показана К. Шенноном в 1948 году в его работе "Математическая теория связи". В ней было сделано утверждение, что *если скорость создания источником сообщений (производительность источника) не превосходит некоторой величины, называемой пропускной способностью канала, то при соответствующем кодировании и декодировании можно свести вероятность ошибок в канале к нулю.*

Вскоре, однако, стало ясно, что фактические ограничения на скорость передачи устанавливаются не пропускной способностью канала, а сложностью схем кодирования и декодирования. Поэтому усилия разработчиков и исследователей в последние десятилетия были направлены на поиски эффективных кодов, создание практически реализуемых схем кодирования и декодирования, которые по своим характеристикам приближались бы к предсказанным теоретически.

## 1.1. Основные принципы. Типы кодов

Кодирование с исправлением ошибок представляет собой *метод обработки сообщений, предназначенный для повышения надежности передачи по цифровым каналам.* Хотя различные схемы кодирования очень непохожи друг на друга и основаны на различных математических теориях, всем им присущи два общих свойства.

Первое – *использование избыточности.* Закодированные последовательности всегда содержат дополнительные, или избыточные, символы. *Количество символов в кодовой последовательности  $Y$  всегда больше, чем необходимо для однозначного представления любого сообщения  $\lambda_i$  из алфавита.*

Второе — *свойство усреднения*, означающее, что *избыточные символы зависят от нескольких информационных символов*, то есть информация, содержащаяся в кодовой последовательности  $X$ , перераспределяется также и на избыточные символы.

Существует два больших класса корректирующих кодов — *блочные и сверточные*. Определяющее различие между этими кодами состоит в отсутствии или наличии памяти кодера.

**Кодер для блочных кодов** делит непрерывную информационную последовательность  $X$  на блоки-сообщения длиной  $k$  символов.

*Кодер канала преобразует блоки-сообщения  $X$  в более длинные двоичные последовательности  $Y$ , состоящие из  $n$  символов и называемые кодовыми словами. Символы  $(n-k)$ , добавляемые к каждому блоку-сообщению кодером, называются *избыточными*. Они не несут никакой дополнительной информации, и их функция состоит в обеспечении возможности обнаруживать (или исправлять) ошибки, возникающие в процессе передачи.*

Как мы ранее показали,  $k$ -разрядным двоичным словом можно представить  $2^k$  возможных значений из алфавита источника, им соответствует  $2^k$  кодовых слов на выходе кодера.

*Такое множество  $2^k$  кодовых слов называется **блочным кодом**.*

Термин "*без памяти*" означает, что *каждый блок из  $n$  символов зависит только от соответствующего информационного блока из  $k$  символов и не зависит от других блоков*.

**Кодер для свёрточных кодов** работает с информационной последовательностью без разбиения ее на независимые блоки. В каждый момент времени кодер из небольшого текущего блока информационных символов размером в  $b$  символов (блока-сообщения) образует блок, состоящий из  $v$  кодовых символов (кодовый блок), причем  $v > b$ . При этом кодовый  $v$ -символьный блок зависит не только от  $b$ -символьного блока-сообщения, присутствующего на входе кодера в настоящий момент, но и от предшествующих  $m$  блоков-сообщений. В этом, собственно, и состоит наличие памяти в кодере.

Блочное кодирование удобно использовать в тех случаях, когда исходные данные по своей природе уже сгруппированы в какие-либо блоки или массивы.

При передаче по радиоканалам чаще используется сверточное кодирование, которое лучше приспособлено к побитовой передаче данных. Кроме этого, при одинаковой избыточности сверточные коды, как правило, обладают лучшей исправляющей способностью.

## 1.2. Линейные блочные коды

Для блочного кода с  $2^k$  кодовыми словами длиной в  $n$  символов, если он только не обладает специальной структурой, аппарат кодирования и декодирования является очень сложным. Поэтому ограничим свое рассмотрение лишь кодами, которые могут быть реализованы на практике.

Одним из условий реализуемости блочных кодов при больших  $k$  является условие их линейности.

*Что такое линейный код?*

Блочный код длиной  $n$  символов, состоящий из  $2^k$  кодовых слов, называется линейным  $(n, k)$ -кодом при условии, что все его  $2^k$  кодовых слов образуют  $k$ -мерное подпространство векторного пространства  $n$ -последовательностей двоичного поля  $GF(2)$ .

Если сказать проще, то двоичный код является линейным, если сумма по модулю 2 ( $\text{mod}2$ ) двух кодовых слов также является кодовым словом этого кода.

Работая с двоичными кодами, мы постоянно будем сталкиваться с элементами двоичной арифметики, поэтому определим основные понятия.

Полем называется множество математических объектов, которые можно складывать, вычитать, умножать и делить.

Возьмем простейшее поле, состоящее из двух элементов – нуля - 0 и единицы - 1. Определим для него операции сложения и умножения:

$$\begin{array}{ll} 0+0=0, & 0 \cdot 0=0; \\ 0+1=1, & 0 \cdot 1=0; \\ 1+0=1, & 1 \cdot 0=0; \\ 1+1=0, & 1 \cdot 1=1. \end{array}$$

Определенные таким образом операции сложения и умножения называются сложением по модулю 2 ( $\text{mod}2$ ) и умножением по модулю 2.

Отметим, что из равенства  $1+1=0$  следует, что  $-1=1$  и, соответственно,  $1+1=1-1$ , а из равенства  $1 \cdot 1=1$  – что  $1:1=1$ .

Алфавит из двух символов 0 и 1 вместе со сложением и умножением по  $\text{mod}2$  называется полем из двух элементов и обозначается как  $GF(2)$ . К полю  $GF(2)$  применимы все методы линейной алгебры, в том числе матричные операции.

Еще раз обратим внимание на то, что все действия над символами в двоичных кодах выполняются по модулю 2.

Желательным качеством линейных блочных кодов является систематичность.

Систематический код имеет формат, изображенный на рис. 1.1, то есть содержит неизменяемую информационную часть длиной  $k$  символов и избыточную (проверочную) длиной  $n - k$  символов.

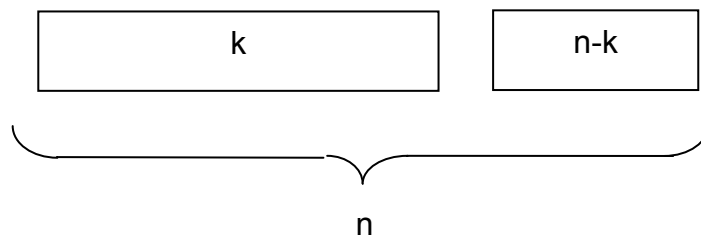


Рис. 1.1

Блочный код, обладающий свойствами линейности и систематичности, называется линейным блочным систематическим  $(n, k)$ -кодом.

### 1.2.1. Код с проверкой на четность

Самым простым линейным блочным кодом является  $(n, n-1)$ -код, построенный с помощью одной общей проверки на четность. Например, кодовое слово  $(4,3)$ -кода можно записать в виде вектора-столбца:

$$\bar{U}^T = (m_0, m_1, m_2, m_0+m_1+m_2), \quad (1.1)$$

где  $m_i$  - символы информационной последовательности, принимающие значения 0 и 1, а суммирование производится по модулю 2 ( $mod 2$ ).

Поясним основную идею проверки на четность.

Пусть информационная последовательность источника имеет вид

$$m = (1 \ 0 \ 1). \quad (1.2)$$

Тогда соответствующая ей кодовая последовательность будет выглядеть следующим образом :

$$U = (U_0, U_1, U_2, U_3) = (1 \ 0 \ 1 \ 0), \quad (1.3)$$

где проверочный символ  $U_3$  формируется путем суммирования по  $mod 2$  символов информационной последовательности  $m$  :

$$U_3 = m_0 + m_1 + m_2. \quad (1.4)$$

Нетрудно заметить, что если число единиц в последовательности  $m$  четно, то результатом суммирования будет 0, если нечетно — 1, то есть проверочный символ дополняет кодовую последовательность таким образом, чтобы количество единиц в ней было четным.

При передаче по каналам связи в принятой последовательности возможно появление ошибок, то есть символы принятой последовательности могут отличаться от соответствующих символов переданной кодовой последовательности (ноль переходит в единицу, а 1 – в 0).

Если ошибки в символах имеют одинаковую вероятность и независимы, то вероятность того, что в  $n$ -позиционном коде произойдет только одна ошибка, составит

$$P_1 = n \cdot P_{ош} \cdot (1 - P_{ош})^{n-1} \quad (1.5)$$

(то есть в одном бите ошибка есть, а во всех остальных  $n - 1$  битах ошибки нет).

Вероятность того, что произойдет две ошибки, определяется уже числом возможных сочетаний ошибок по две (в двух произвольных битах ошибка есть, а во всех остальных  $n - 2$  битах ошибки нет):

$$P_2 = C_n^2 \cdot P_{ош}^2 \cdot (1 - P_{ош})^{n-2}, \quad (1.6)$$

и аналогично для ошибок более высокой кратности.

Если считать, что вероятность ошибки на символ принятой последовательности  $P_{ош}$  достаточно мала ( $P_{ош} \ll 1$ ), а в противном случае передача информации не имеет смысла, то вероятность выпадения ровно  $l$  ошибок составит  $P_l \cong P_{ош}^l$ .

Отсюда видно, что наиболее вероятными являются одиночные ошибки, менее вероятными — двойные, еще меньшую вероятность будут иметь трехкратные ошибки и т. д.

Если при передаче рассматриваемого (4,3)-кода произошла одна ошибка, причем неважно, в какой его позиции, то общее число единиц в принятой последовательности  $r$  уже не будет четным.

Таким образом, *признаком отсутствия ошибки в принятой последовательности может служить четность числа единиц*. Поэтому такие коды и называются *кодами с проверкой на четность*.

Правда, если в принятой последовательности  $r$  произошло две ошибки, то общее число единиц в ней снова станет четным и ошибка обнаружена не будет. Однако вероятность двойной ошибки значительно меньше вероятности одиночной, поэтому наиболее вероятные одиночные ошибки таким кодом обнаруживаться все же будут.

На основании общей идеи проверки на четность и проверочного уравнения (1.4) легко организовать схему кодирования - декодирования для произвольного кода с простой проверкой на четность.

Схема кодирования может выглядеть следующим образом (рис. 1.2):

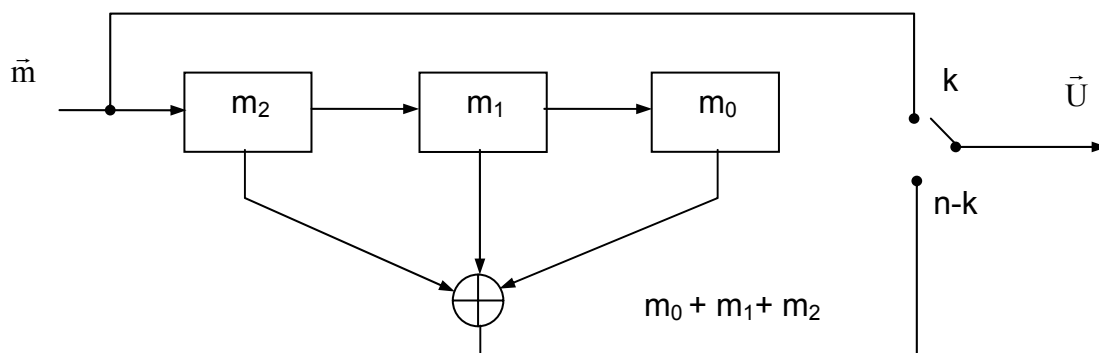


Рис. 1.2

Декодирующее устройство для кода с проверкой на четность изображено на рис. 1.3.

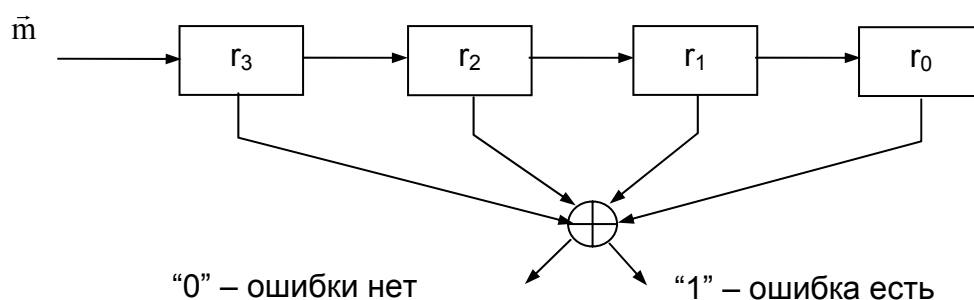


Рис. 1.3

Декодер, как это видно из рис. 1.3, проверяет на четность общее число единиц в принятой последовательности и выдает на своем выходе нуль или единицу в зависимости от того, выполнена проверка или нет.

Отметим следующий момент. Если посимвольно сложить два кодовых слова, принадлежащих рассматриваемому  $(4, 3)$ -коду:

$$a = (a_0, a_1, a_2, a_0 + a_1 + a_2), \text{ и } b = (b_0, b_1, b_2, b_0 + b_1 + b_2), \quad (1.7)$$

то получим

$$c = (a_0 + b_0, a_1 + b_1, a_2 + b_2, a_0 + b_0 + a_1 + b_1 + a_2 + b_2) = (c_0, c_1, c_2, c_0 + c_1 + c_2), \quad (1.8)$$

то есть проверочный символ в новом слове  $c$  определяется по тому же правилу, что и в слагаемых. Поэтому  $c$  также является кодовым словом данного кода.

Этот пример отражает важное свойство линейных блочных кодов — замкнутость, означающее, что *сумма двух кодовых слов данного кода также является кодовым словом*.

Несмотря на свою простоту и не очень высокую эффективность, коды с проверкой на четность широко используются в системах передачи и хранения информации. Они ценятся за невысокую избыточность: достаточно добавить к передаваемой последовательности всего один избыточный символ — и можно узнать, есть ли в принятой последовательности ошибка. Правда, определить место этой ошибки и, следовательно, исправить ее, пока нельзя. Можно лишь повторить передачу слова, в котором была допущена ошибка, и тем самым ее исправить.

### 1.2.2. Итеративный код

Еще одна простая схема кодирования, которая также часто используется, может быть построена следующим образом.

Предположим, что нужно передать, к примеру, девять информационных символов  $m = (m_0, m_1, \dots, m_8)$ . Эти символы можно расположить в виде квадратной матрицы, как это показано в табл. 1.1, и добавить к каждой строке и каждому столбцу этой таблицы по проверочному символу (проверка на четность).

Таблица 1.1

$m_0$	$m_1$	$m_2$	$P_1 = m_0 + m_1 + m_2$
$m_3$	$m_4$	$m_5$	$P_2 = m_3 + m_4 + m_5$
$m_6$	$m_7$	$m_8$	$P_3 = m_6 + m_7 + m_8$
$m_0 + m_3 + m_6$	$m_1 + m_4 + m_7$	$m_2 + m_5 + m_8$	$m_0 + m_0 + m_1 + m_1 + \dots + m_8 + m_8$

Таким образом, по строкам и по столбцам этой таблицы будет выполняться правило четности единиц.

Если в процессе передачи по каналу с помехами в этой таблице произойдет одна ошибка (например в символе  $m_4$ ), то проверка на четность в соответствующей строке и столбце (в нашем примере -  $P_2$  и  $P_5$ ) не будет выполняться.

Иными словами, координаты ошибки однозначно определяются номерами столбца и строки, в которых не выполняются проверки на четность. Таким образом, этот код, используя *различные проверки на четность* (по строкам и по столбцам), способен не только обнаруживать, но и исправлять ошибки (если известны координаты ошибки, то ее исправление состоит просто в замене символа на противоположный: если 0, то на 1, если 1 — то на 0).

Описанный метод кодирования, называемый *итеративным*, оказывается полезным в случае, когда данные естественным образом формируются в виде массивов, например, на шинах ЭВМ, в памяти, имеющей табличную структуру, и т.д. При этом размер таблицы в принципе не имеет значения ( $3 \times 3$  или  $20 \times 20$ ), однако в первом случае будет исправляться одна ошибка на  $3 \times 3 = 9$  символов, а во втором – на  $20 \times 20 = 400$  символов.

Обратим внимание еще на один момент. Если в простом коде с проверкой на четность для *обнаружения ошибки* приходится добавлять к информационной последовательности *всего один символ*, то для того, чтобы код стал *исправлять однократную ошибку*, понадобилось к девяти информационным символам добавить еще *семь проверочных*.

Таким образом, избыточность этого кода оказалась очень большой, а исправляющая способность – сравнительно низкой. Поэтому усилия специалистов в области помехоустойчивого кодирования всегда были направлены на поиск таких кодов и методов кодирования, которые *при минимальной избыточности* обеспечивали бы *максимальную исправляющую способность*.

### 1.2.3. Порождающая матрица линейного блочного кода

Только что в качестве примера были рассмотрены два простейших корректирующих кода - код с простой проверкой на четность, позволяющий обнаруживать однократную ошибку в принятой последовательности, и блочный итеративный код, исправляющий одну ошибку с помощью набора проверок на четность по строкам и столбцам таблицы. Однако формальное правило, по которому осуществляется кодирование, то есть *преобразование информационной последовательности в кодовое слово*, по-настоящему еще не определено. Так как же задаются блочные коды?

Простейшим способом описания, или задания, корректирующих кодов является *табличный способ*, при котором каждой информационной последовательности просто назначается кодовое слово из таблицы кода (табл. 1.2)

Таблица 1.2

<i>m</i>	<i>U</i>
<i>000</i>	<i>0000</i>
<i>001</i>	<i>0011</i>
<i>010</i>	<i>0101</i>
<i>011</i>	<i>0110</i>
<i>100</i>	<i>1001</i>
<i>101</i>	<i>1010</i>
<i>110</i>	<i>1100</i>
<i>111</i>	<i>1111</i>

Такой способ описания кодов, кстати, применим для любых, а не только линейных кодов. Однако при больших  $k$  размер кодовой таблицы оказывается слишком большим, чтобы им пользоваться на практике (для кода с простой проверкой на четность двухбайтового слова размер таблицы составит  $\sim 2^5 * 2^{16} = 2000000$  двоичных символов).

Другим способом задания линейных блочных кодов является использование так называемой *системы проверочных уравнений*, определяющих правило, по которому символы информационной последовательности преобразуются в кодовые символы. Для того же примера система проверочных уравнений будет выглядеть следующим образом:

$$\begin{aligned} U_0 &= m_0, \\ U_1 &= m_1, \\ U_2 &= m_2, \\ U_3 &= m_0 + m_1 + m_2. \end{aligned} \tag{1.9}$$

Однако наиболее удобным и наглядным способом описания линейных блочных кодов является их задание с использованием *порождающей матрицы*, являющейся компактной формой представления системы проверочных уравнений:

$$\underline{G} = \left( \begin{array}{cccc|cccc} 1 & 0 & 0 & \dots & 0 & P_{00} & P_{01} & \dots & P_{0, n-k-1} \\ 0 & 1 & 0 & \dots & 0 & P_{10} & P_{11} & \dots & P_{1, n-k-1} \\ & & \dots & & & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & 1 & P_{k-1, 0} & P_{k-1, 1} & \dots & P_{k-1, n-k-1} \end{array} \right). \tag{1.10}$$

единичная матрица  $\mathbf{I}$   
к\*к
матрица  $\mathbf{P}$   
к\*(n-k)

**Определение.** *Линейный блочный систематический  $(n,k)$ -код полностью определяется матрицей  $\underline{G}$  размером  $k * n$  с двоичными матричными элементами. При этом каждое кодовое слово является линейной комбинацией строк матрицы  $\underline{G}$ , а каждая линейная комбинация строк  $\underline{G}$  - кодовым словом.*

Пусть  $m = (m_0, m_1, \dots, m_{k-1})$  будет тем блоком-сообщением, который необходимо закодировать с использованием данного кода.

Тогда соответствующим ему кодовым словом  $U$  будет

$$U = m \cdot \underline{G}. \tag{1.11}$$

С учетом структуры матрицы  $\underline{G}$  символы кодового слова  $U$  будут такими:

для  $i = 0, 1, 2, \dots, k-1$

$$U_i = m_i; \quad (1.12)$$

для  $i = k, k+1, \dots, n$

$$U_i = m_0 \cdot P_{0j} + m_1 \cdot P_{1j} + m_2 \cdot P_{2j} + \dots + m_{k-1} \cdot P_{k-1,j}. \quad (1.13)$$

Иными словами,  $k$  крайних левых символов кодового слова совпадает с символами кодируемой информационной последовательности, а остальные  $(n - k)$  символов являются линейными комбинациями символов информационной последовательности.

Определенный таким образом код называется линейным блочным систематическим  $(n, k)$ -кодом с обобщенными проверками на четность, а задающая его матрица  $\underline{G}$  называется порождающей матрицей кода.

В качестве примера рассмотрим известный  $(7, 4)$ -код Хемминга, являющийся классической иллюстрацией простейших кодов с исправлением ошибок.

Пусть  $m = (m_0, m_1, m_2, m_3)$  будет тем сообщением, или информационной последовательностью, которую нужно закодировать.

Порождающая матрица  $\underline{G}$  для  $(7, 4)$ -кода Хемминга имеет вид

$$\underline{G}_{(7,4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \quad (1.14)$$

Тогда символы соответствующего кодового слова определяются следующим образом :

$$U = m \cdot \underline{G} = (m_0 m_1 m_2 m_3) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} =$$

$$= (m_0, m_1, m_2, m_3, m_0 + m_2 + m_3, m_0 + m_1 + m_2, m_1 + m_2 + m_3), \quad (1.15)$$

ИЛИ

$$\begin{aligned}
 U_0 &= m_0, \\
 U_1 &= m_1, \\
 U_2 &= m_2, \\
 U_3 &= m_3, \\
 U_4 &= m_0 + m_2 + m_3, \\
 U_5 &= m_0 + m_1 + m_2, \\
 U_6 &= m_1 + m_2 + m_3.
 \end{aligned}
 \tag{1.16}$$

Например, пусть  $m = (1\ 0\ 1\ 1)$ , тогда соответствующее кодовое слово будет иметь вид  $U = (1\ 0\ 1\ 1\ 1\ 0\ 0)$ . Или другой пример: пусть  $m = (1\ 0\ 0\ 0)$ , тогда  $U = (1\ 0\ 0\ 0\ 1\ 1\ 0)$ .

Интересно отметить, что в соответствии с приведенным выше определением строки матрицы  $\underline{G}$  сами являются кодовыми словами данного кода, а все остальные кодовые слова - линейными комбинациями строк порождающей матрицы.

На основании порождающей матрицы  $\underline{G}_{(7,4)}$  (1.15) или приведенной системы проверочных уравнений (1.16) легко реализовать схему кодирования для рассматриваемого (7,4)-кода Хемминга (рис. 1.4).

Кодер работает точно так же, как и при простой проверке на четность, но теперь выполняет не одну общую, а несколько частичных проверок, формируя, соответственно, несколько проверочных символов.

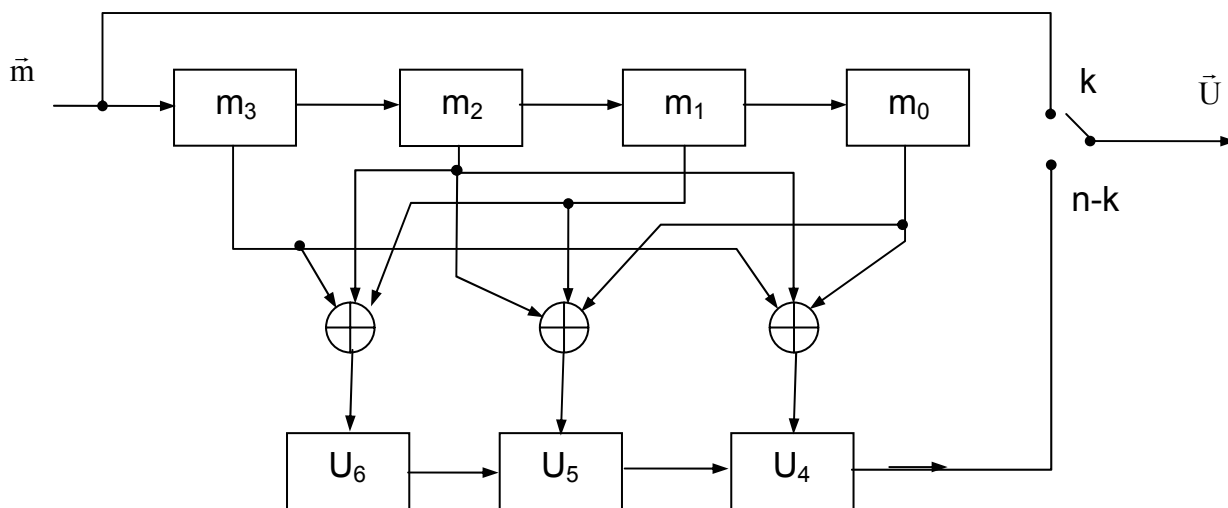


Рис. 1.4

### 1.2.4. Проверочная матрица

Линейный систематический блочный код может быть определен также с использованием так называемой проверочной матрицы  $\underline{H}$ , обладающей следующим свойством:

- если некоторая последовательность  $U$  является кодовым словом, то

$$U * \underline{H}^T = \underline{0}. \quad (1.17)$$

Другими словами, проверочная матрица  $\underline{H}$  ортогональна любой кодовой последовательности данного кода.

Проверочная матрица имеет размерность  $(n-k) * n$  и следующую структуру :

$$\underline{H} = \left( \begin{array}{cccc|cccc} P_{00} & P_{10} & \dots & P_{k-1,0} & 1 & 0 & 0 & \dots & 0 \\ P_{01} & P_{11} & \dots & P_{k-1,1} & 0 & 1 & 0 & \dots & 0 \\ P_{02} & P_{12} & \dots & P_{k-1,2} & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ P_{0,n-k-1} & P_{1,n-k-1} & \dots & P_{k-1,n-k-1} & 0 & 0 & 0 & \dots & 1 \end{array} \right), \quad (1.18)$$

$\underline{P}^T \qquad \qquad \qquad \underline{I}^1_{(n-k) \times (n-k)}$

где  $\underline{P}^T$  - транспонированная подматрица  $\underline{P}$  из порождающей матрицы  $\underline{G}$  ;  
 $\underline{I}^1_{(n-k) \times (n-k)}$  - единичная матрица соответствующего размера.

Видно, что единичная и проверочная подматрицы в  $\underline{G}$  и  $\underline{H}$  поменялись местами, кроме того, изменился их размер.

Для рассматриваемого нами в качестве примера (7,4)-кода Хемминга проверочная матрица  $\underline{H}$  имеет вид

$$H_{(7,4)} = \left( \begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right). \quad (1.19)$$

Проверочная матрица позволяет легко определить, является ли принятая последовательность кодовым словом данного кода.

Пусть, к примеру, принята последовательность символов  $c = (1011001)$ , тогда

$$c * H^T = (1011001) \left( \begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right)^T = (1 \ 1 \ 0) \neq \underline{0}.$$

Отсюда можно сделать вывод, что последовательность  $c = (1011001)$  не является кодовым словом данного кода.

Рассмотрим другой пример. Допустим, принята последовательность  $d = (0010111)$ , тогда

$$d \cdot \underline{H}^T = (0010111) \left| \begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right|^T = (0 \ 0 \ 0) \neq \underline{0},$$

то есть двоичная последовательность  $d$  принадлежит коду с проверочной матрицей  $\underline{H}$ .

### 1.2.5. Дуальные коды

Рассматривая матрицы  $\underline{G}$  и  $\underline{H}$ , можно сделать следующие интересные выводы. Каждая из них содержит множество линейно независимых векторов, то есть каждая из матриц может рассматриваться как базис некоторого линейного пространства. Кроме того, каждое из этих пространств является подпространством векторного пространства, состоящего из всех наборов двоичных символов длиной  $n$ .

Скалярное произведение каждой строки матрицы  $\underline{G}$  на каждую строку матрицы  $\underline{H}$  равно нулю, то есть

$$\underline{H} \cdot \underline{G}^T = \underline{0} \quad \text{и} \quad \underline{G} \cdot \underline{H}^T = \underline{0}. \quad (1.20)$$

Следовательно, можно "поменять ролями" эти две матрицы и использовать  $\underline{H}$  как порождающую матрицу, а  $\underline{G}$  как проверочную матрицу некоторого другого кода.

Коды, связанные таким образом, называются дуальными друг другу, т.е., задав каким-либо образом линейный блочный код, мы автоматически задаем и второй, дуальный ему код. Правда, если исходный код был получен так, чтобы иметь минимальную избыточность при заданной исправляющей способности, то гарантировать хорошее качество дуального ему кода мы не можем. Такие коды обычно имеют исправляющую способность, одинаковую с исходными, но большую, чем у них, избыточность.

Например, если рассмотренный в качестве примера  $(7,4)$ -код Хемминга имеет избыточность  $7/4$  и при этом позволяет исправлять одну ошибку в кодовом слове из 7 символов (об этом будем подробно говорить в следующих разделах настоящего пособия), то дуальный ему код  $(7,3)$  также исправляет одну ошибку на 7 символов, но уже имеет избыточность  $7/3$ , то есть на 3 информационных символа содержит 4 проверочных.

### 1.2.6. Синдром и обнаружение ошибок

Прежде чем говорить об обнаружении и исправлении ошибок корректирующими кодами, определим само понятие ошибки и методы их описания.

Пусть  $U = (U_0, U_1, \dots, U_n)$  является кодовым словом, переданным по каналу с помехами, а  $r = (r_0, r_1, \dots, r_n)$  - принятой последовательностью, возможно, отличающейся от переданного кодового слова  $U$ . Отличие  $r$  от  $U$  состоит в том, что некоторые символы  $r_i$  принятой последовательности могут отличаться от соответствующих символов  $U_i$  переданного кодового слова. Например,  $U = (0001000)$ , а  $r = (0000000)$ , то есть произошла ошибка в четвертом символе кодового слова, 1 перешла в 0. Или другой пример: передано кодовое слово  $U = (001111)$ , а принятая последовательность имеет вид  $r = (101111)$ , то есть ошибка возникла в первом бите кодового слова, при этом 0 перешел в единицу.

Для описания возникающих в канале ошибок используют вектор ошибки, обычно обозначаемый как  $e$  и представляющий собой двоичную последовательность длиной  $n$  с единицами в тех позициях, в которых произошли ошибки.

Так, вектор ошибки  $e = (0001000)$  означает однократную ошибку в четвертой позиции (четвертом бите), вектор ошибки  $e = (1100000)$  - двойную ошибку в первом и втором битах и т.д.

Тогда при передаче кодового слова  $U$  по каналу с ошибками принятая последовательность  $r$  будет иметь вид

$$r = U + e, \quad (1.21)$$

например:

$$\begin{aligned} U &= (0001000), \\ e &= (0001000), \\ r &= (0000000). \end{aligned} \quad (1.22)$$

Приняв вектор  $r$ , декодер сначала должен определить, имеются ли в принятой последовательности ошибки. Если ошибки есть, то он должен выполнить действия по их исправлению.

Чтобы проверить, является ли принятый вектор кодовым словом, декодер вычисляет  $(n-k)$ -последовательность, определяемую следующим образом:

$$S = (S_0, S_1, \dots, S_{n-k-1}) = r \cdot \underline{H}^T. \quad (1.23)$$

При этом  $r$  является кодовым словом тогда, и только тогда, когда  $S = (00..0)$ , и не является кодовым словом данного кода, если  $S \neq \underline{0}$ . Следовательно,  $S$  используется для обнаружения ошибок, ненулевое значение  $S$  служит признаком наличия ошибок в принятой последовательности. Поэтому вектор  $S$  называется синдромом принятого вектора  $r$ .

Некоторые сочетания ошибок, используя синдром, обнаружить невозможно. Например, если переданное кодовое слово  $U$  под влиянием помех превратилось в другое действительное кодовое слово  $V$  этого же кода, то синдром  $S = V * \underline{H}^T = \underline{0}$ . В этом случае декодер ошибки не обнаружит и, естественно, не попытается ее исправить.

Сочетания ошибок такого типа называются необнаруживаемыми. При построении кодов необходимо стремиться к тому, чтобы они обнаруживали наиболее вероятные сочетания ошибок.

Для рассматриваемого в качестве примера линейного блочного систематического  $(7,4)$ -кода Хемминга синдром определяется следующим образом:

пусть принят вектор  $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ , тогда

$$S = r * \underline{H}_{(7,4)}^T = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) * \begin{vmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}^T =$$

$$= (r_0 + r_2 + r_3 + r_4), (r_0 + r_1 + r_2 + r_5), (r_1 + r_2 + r_2 + r_6), \quad (1.23)$$

или

$$\begin{aligned} S_0 &= r_0 + r_2 + r_3 + r_4, \\ S_1 &= r_0 + r_1 + r_2 + r_5, \\ S_2 &= r_1 + r_2 + r_2 + r_6. \end{aligned} \quad (1.24)$$

Основываясь на полученных соотношениях, можно легко организовать схему для вычисления синдрома. Для  $(7,4)$ -кода Хемминга она приведена на рис. 1.5.

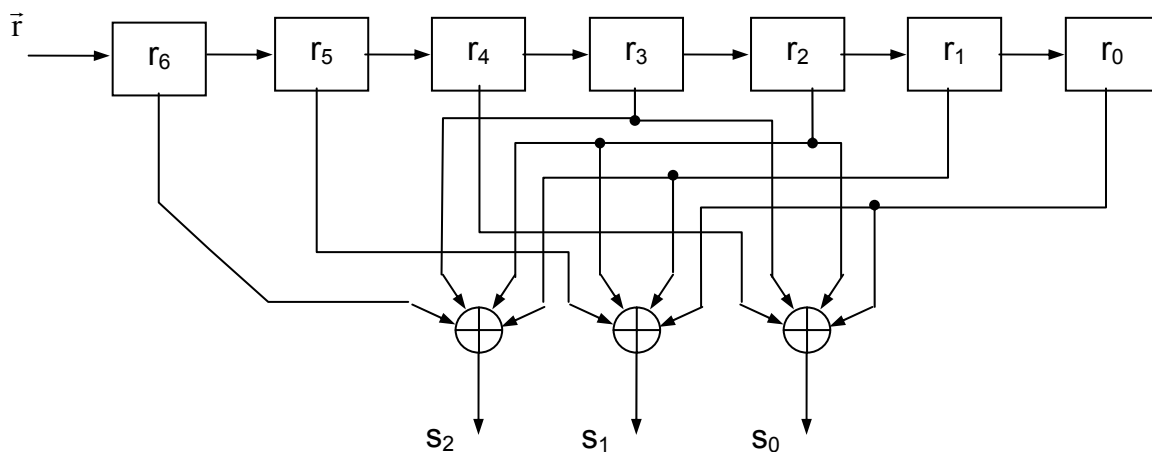


Рис. 1.5

### 1.2.7. Синдромное декодирование линейных блочных кодов

Покажем, как можно использовать синдром принятого вектора не только для обнаружения, но и для исправления ошибок.

Пусть  $U = (U_0, U_1, \dots, U_{n-1})$ ,  $e = (e_0, e_1, \dots, e_{n-1})$  и  $r = (r_0, r_1, r_2, \dots, r_{n-1})$  являются передаваемым кодовым словом, вектором-ошибкой и принятым вектором соответственно. Тогда

$$r = U + e \quad (1.25)$$

и синдром

$$S = r \cdot H^T = (U + e) \cdot H^T = U \cdot H^T + e \cdot H^T = 0 + e \cdot H^T = e \cdot H^T, \quad (1.26)$$

поскольку для любого кодового слова  $U \cdot H^T = 0$ .

Таким образом, синдром принятой последовательности  $r$  зависит только от ошибки, имеющей место в этой последовательности, и совершенно не зависит от переданного кодового слова. Задача декодера, используя эту зависимость, определить элементы (координаты) вектора ошибок. Найдя вектор ошибки можно восстановить кодовое слово как

$$U^* = r + e. \quad (1.27)$$

На примере одиночных ошибок при кодировании с использованием линейного блочного (7,4)-кода покажем, как вектор ошибки связан с синдромом, и как, имея синдром, локализовать и устранить ошибки, возникшие при передаче.

*Найдем значения синдрома для всех возможных одиночных ошибок в последовательности из семи символов:*

$$e_- = (0000000),$$

$$e_- \cdot \underline{H}^T = (0000000) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (000);$$

$$e_0 = (1000000),$$

$$e_0 \cdot \underline{H}^T = (1000000) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (110);$$

$$e_1 = (0100000),$$

$$e_1 \cdot \underline{H}^T = (0100000) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (011);$$

$$e_2 = (0010000),$$

$$e_2 \cdot \underline{H}^T = (0010000) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (111);$$

$$e_3 = (0001000),$$

$$e_3 \cdot \underline{H}^T = (0001000) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (101);$$

$$e_4 = (0000100),$$

$$e_4 \cdot \underline{H}^T = (0000100) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (100);$$

$$e_5 = (0000010),$$

$$e_5 \cdot \underline{H}^T = (0000010) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (010);$$

$$e_6 = (0000001),$$

$$e_6 \cdot \underline{H}^T = (0000001) \cdot \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (001).$$

Все возможные для (7,4)-кода одиночные ошибки и соответствующие им векторы синдрома приведены в табл. 1.3.

Таблица 1.3

Вектор ошибки	Синдром ошибки	Десятичный код синдрома
<b>1000000</b>	<b>110</b>	6
<b>0100000</b>	<b>011</b>	3
<b>0010000</b>	<b>111</b>	7
<b>0001000</b>	<b>101</b>	5
<b>0000100</b>	<b>100</b>	4
<b>0000010</b>	<b>010</b>	2
<b>0000001</b>	<b>001</b>	1

Из этой таблицы видно, что существует *однозначное соответствие между сочетанием ошибок (при одиночной ошибке) и его синдромом*, то есть, зная синдром, можно совершенно однозначно определить позицию кода, в которой произошла ошибка.

Например, если синдром, вычисленный по принятому вектору, равен (**111**), это значит, что произошла одиночная ошибка в третьем символе, если **S = (001)** – то в последнем, и т.д.

Если место ошибки определено, то устранить ее уже не представляет никакого труда.

Полная декодирующая схема для (7,4)-кода Хемминга, использующая синдром вектора **r** не только для обнаружения, но и для исправления ошибок, приведена на рис. 1.6.

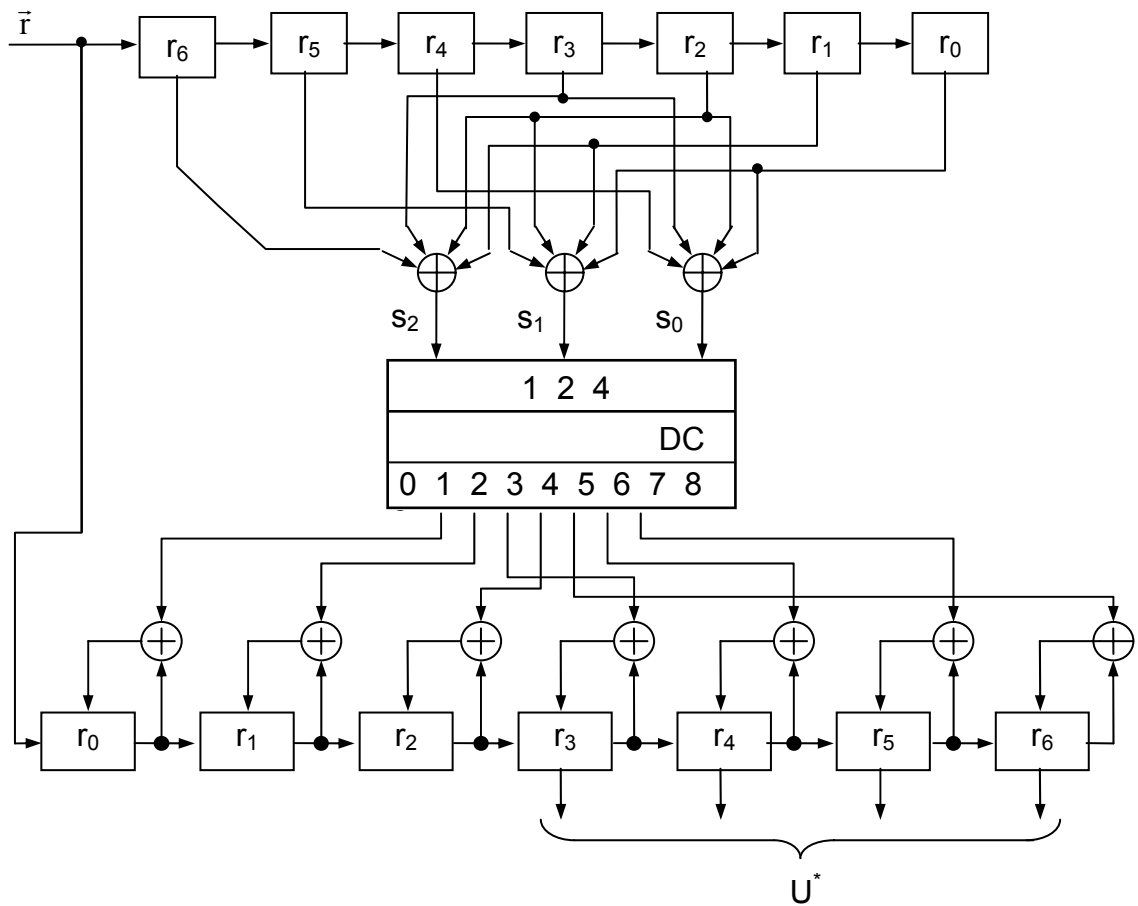


Рис. 1.6

А теперь посмотрим, что произойдет, если в принятой последовательности будет не одна, а, например, две ошибки. Пусть ошибки возникли во второй и шестой позициях  $e = (0100010)$ . Соответствующий синдром определится как

$$S = (0100010) \begin{vmatrix} 1011100 \\ 1110010 \\ 0111001 \end{vmatrix}^T = (001).$$

Однако синдром  $S = (001)$  соответствует также и одиночной ошибке в седьмой позиции ( $e_6$ ). Следовательно, наш декодер не только не исправит ошибок в позициях, в которых они произошли, но и внесет ошибку в ту позицию, где ее не было. Таким образом, видно, что  $(7,4)$ -код не обеспечивает исправления двойных ошибок, а также ошибок большей кратности.

Это, однако, обусловлено не тем, каким образом производится декодирование, а свойствами самого кода. Несколько позднее будет показано, от чего зависит исправляющая способность кода, то есть сколько ошибок он может исправить.

### 1.2.8. Мажоритарное декодирование линейных блочных кодов

Идею мажоритарного декодирования линейных блочных кодов можно продемонстрировать на очень простом примере.

Пусть  $m$  – кодируемая информационная последовательность, состоящая всего из одного символа  $m_0 = 0$  или  $1$ , а соответствующее ей кодовое слово помехоустойчивого (избыточного) кода имеет вид  $U = (m_0, m_0, m_0)$ , то есть  $(000)$ , если  $m_0 = 0$ , или  $(111)$ , если  $m_0 = 1$  (код с трехкратным повторением).

Допустим, передано кодовое слово  $U = (111)$  и в первом символе произошла одна ошибка, то есть принята последовательность  $r = (011)$ .

*Вопрос: какая последовательность передавалась,  $U = (000)$  или  $U = (111)$ ?*

Здравый смысл подсказывает, что, скорее всего, передавалось кодовое слово  $U = (111)$ , так как в противном случае ошибка должна была бы исказить два символа, чтобы кодовое слово  $U = (000)$  превратилось в последовательность вида  $r = (011)$ . Может быть, и не отдавая себе отчета в правиле принятия решения (о том, что передавалось), мы приняли решение по большинству – мажоритарно.

В общем случае, для линейных блочных кодов с более сложной структурой, решение будет не таким простым, но идея мажоритарного декодирования – та же: решение принимается по большинству. Как и в жизни, полагается, что большинство дает более правильный ответ.

Рассмотрим более сложный пример – мажоритарное декодирование для  $(7,4)$ -кода Хемминга.

Пусть передано кодовое слово  $(7,4)$ -кода –  $U = (U_0, U_1, U_2, U_3, U_4, U_5, U_6)$ , символы которого сформированы в соответствии с системой проверочных уравнений (правилом кодирования) вида:

$$\begin{aligned}U_0 &= m_0, \\U_1 &= m_1, \\U_2 &= m_2, \\U_3 &= m_3, \\U_4 &= m_0 + m_2 + m_3, \\U_5 &= m_0 + m_1 + m_2, \\U_6 &= m_1 + m_2 + m_3.\end{aligned}\tag{1.28}$$

На входе декодера наблюдается принятая последовательность  $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ , и необходимо ее декодировать, то есть определить вид передаваемой информационной последовательности  $m$ .

Поскольку невозможно быть абсолютно уверенными в правильности декодирования, мы можем говорить лишь об оценке информационной последовательности  $m^*$ .

Для начала предположим, что *ошибок в принятой последовательности  $r$  нет*, то есть  $r = U$ .

Тогда по принятой последовательности  $r$  можно легко найти оценку переданной информационной последовательности  $m^*$ , причем не единственным способом.

Во-первых, можно сразу записать

$$\begin{aligned} m_0^{*1} &= r_0, \\ m_1^{*1} &= r_1, \\ m_2^{*1} &= r_2, \\ m_3^{*1} &= r_3, \end{aligned} \tag{1.29}$$

то есть в качестве ответа или результата декодирования, взять первые четыре символа принятой последовательности.

Но это не единственный способ. Учитывая, что для элементов поля  $GF(2)$  справедливо условие  $m_i + m_i = 0$  ( то есть  $1+1 = 0$  и  $0+0 = 0$ ), можно записать еще несколько систем уравнений для определения  $m_i^*$ :

$$\begin{aligned} m_0^{*2} &= r_2 + r_3 + r_4, & m_0^{*3} &= r_1 + r_2 + r_5, \\ m_1^{*2} &= r_0 + r_2 + r_5, & m_1^{*3} &= r_2 + r_3 + r_6, \\ m_2^{*2} &= r_0 + r_3 + r_4, & m_2^{*3} &= r_0 + r_1 + r_5, \\ m_3^{*2} &= r_0 + r_2 + r_4; & m_3^{*3} &= r_1 + r_2 + r_6; \\ & & & \\ m_0^{*4} &= r_1 + r_4 + r_6, & m_0^{*5} &= r_3 + r_5 + r_6, \\ m_1^{*4} &= r_0 + r_4 + r_6, & m_1^{*5} &= r_3 + r_4 + r_5, \\ m_2^{*4} &= r_4 + r_5 + r_6, & m_2^{*5} &= r_1 + r_3 + r_6, \\ m_3^{*4} &= r_0 + r_5 + r_6; & m_3^{*5} &= r_1 + r_4 + r_5. \end{aligned} \tag{1.30}$$

Таким образом, получилось пять независимых систем уравнений для определения одних и тех же компонент вектора  $m^*$ , причем, они будут совместными (иметь одинаковые решения) только при отсутствии ошибок в принятой последовательности  $r$ , то есть при  $r = U$ . В противном случае решения для  $m_i^*$ , даваемые различными системами, будут разными.

Однако можно заметить следующее: в выражениях для  $m_i^*$  каждый из элементов принятой последовательности  $r_i$  присутствует не более двух раз (то есть не более чем в двух уравнениях из пяти).

Если считать, что в принятой последовательности возможна *только одиночная ошибка* (а с ошибкой большей кратности этот код не справляется), то *ошибочными будут решения не более чем двух уравнений из пяти для каждого из элементов  $m_i^*$ , остальные три уравнения дадут правильное решение*. Тогда правильный ответ может быть получен по "*большинству голосов*", или *мажоритарно*.

Устройством, которое принимает решение по "большинству", является так называемый *мажоритарный селектор*. При этом схема мажоритарного декодера для одного из символов принятой последовательности (7,4)-кода Хемминга может выглядеть, например, следующим образом (рис. 1.7):

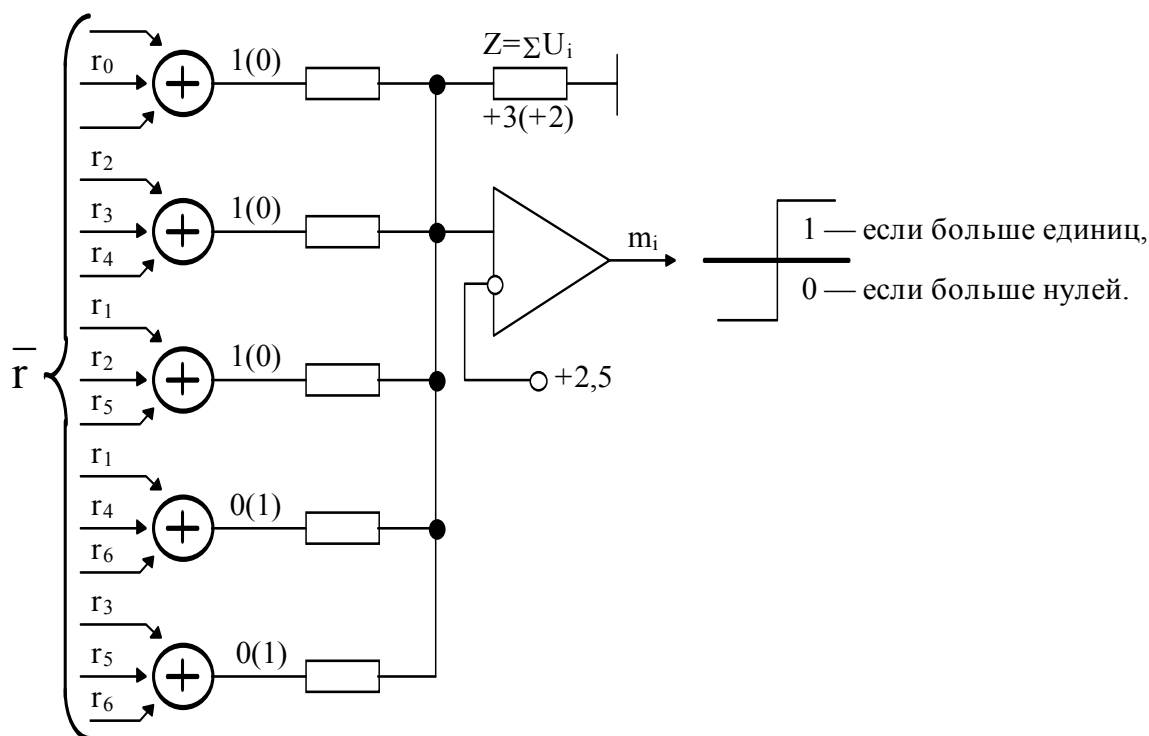


Рис. 1.7

Здесь мажоритарный селектор выполнен в виде аналогового сумматора и компаратора напряжений (напряжение на выходе компаратора = 1, если на его входе больше единиц, и равно 0 в противном случае). Однако возможна и чисто цифровая реализация мажоритарного селектора: он просто выдаст на своем выходе 1, если на его входе больше единиц, и 0 – в противном случае.

### 1.2.9. Декодирование методом максимального правдоподобия

Итак, рассмотрено несколько различных способов декодирования линейных блочных кодов, и, наверное, существует еще множество других способов. Возникает вопрос: а есть ли среди них наилучший, при использовании которого останется неисправленным наименьшее число ошибок?

Попытаемся определить наилучшее, или *оптимальное*, правило декодирования.

Пусть  $U = (U_0, U_1, \dots, U_i, \dots, U_n)$  является переданным кодовым словом некоторого двоичного блочного  $(n, k)$ -кода, а  $r = (r_0, r_1, \dots, r_i, \dots, r_n)$  – последовательность, принятая на выходе канала с помехами.

Принятая последовательность из-за действия шумов может отличаться от переданной, то есть по отдельным символам приемник мог принять неправильные решения (вместо нулей – единицы и наоборот).

Декодер канала на основе принятой последовательности должен принять решение относительно переданного кодового слова. Процедура принятия такого решения и называется декодированием.

Если декодер не в состоянии правильно воспроизвести действительное кодовое слово, то есть  $m^* \neq m$ , то при декодировании возникнет ошибка. Эта ошибка случайна, ее вероятность зависит от характеристик канала связи, характеристик кода, метода кодирования и декодирования. Желательно, чтобы вероятность ошибочного декодирования была как можно меньшей.

*Как должен работать декодер, чтобы вероятность ошибочного декодирования была минимальной?*

Сначала рассмотрим ситуацию, когда приемник не принимает решений относительно того, какой из символов  $r_i$  (0 или 1) в данный момент принят, то есть он отдает декодеру весь принятый сигнал  $S(t)$  и предоставляет право принимать решения самому декодеру.

Пусть  $U_l, (l = 0, 1, 2, 3..2^k - 1)$  –  $l$ -е кодовое слово используемого кода;

$U_{li}$  –  $i$ -й символ этого кодового слова;

$S(t)$  – принятый сигнал, содержащий одно из кодовых слов и помеху.

Какое кодовое слово содержится в принятом сигнале, мы не знаем. Известна только априорная вероятность передачи  $l$ -го кодового слова –  $P_l$ .

Оптимальный декодер должен учитывать всю имеющуюся информацию об используемом коде, канале связи и помехах, действующих в этом канале, и обеспечивать максимальную вероятность правильных ответов о том, какие кодовые слова были переданы по каналу связи. Такой критерий оптимальности – максимум апостериорной (послеопытной) вероятности правильных решений – называется критерием Байеса.

Оптимальный по критерию Байеса декодер должен выбирать в качестве решения кодовое слово  $U^* = U_k$ , которое максимизирует условную вероятность  $P(U_k/S)$  — вероятность того, что была передана последовательность  $U_k$ , если принята данная реализация сигнала  $S$ .

Поскольку

$$P(U_k/S) \cdot P(S) = P(S/U_k) \cdot P(U_k), \quad (1.31)$$

то

$$P(U_k/S) = P(S/U_k) \cdot P(U_k) / P(S). \quad (1.32)$$

Если считать, что все кодовые слова равновероятны –  $P(U_k) = \text{const}$ , а также учитывая, что безусловная плотность  $P(S)$  не зависит от  $U_k$ , то максимуму  $P(U_k/S)$  соответствует максимум  $P(S/U_k)$ , так называемой

функции правдоподобия — условной вероятности того, что сигнал примет свое значение  $S$ , если передавалось кодовое слово  $U_k$ .

В дальнейшем мы еще вернемся к подробному рассмотрению вопросов оптимального приема сигналов и покажем, как определяется вид функции правдоподобия, сейчас же можно сказать, что значение  $P(U_k / S)$  будет максимально, если минимальна величина

$$d_k = \Sigma \int \{ S(t) - U_k \}^2 dt, \quad (1.33)$$

или

$$d_k = \Sigma \Sigma \{ S_i - U_{ki} \}^2, \quad (1.34)$$

если принятый сигнал дискретизован и  $S_i$  —  $i$ -й отсчет принятого сигнала.

Сумма квадратов разностей между значениями принятого сигнала  $S_i$  и символами  $k$ -го кодового слова называется невязкой, или евклидовым расстоянием между этим кодовым словом и принятым сигналом.

Если помех в канале связи нет или они невелики, то при передаче  $l$ -го кодового слова принятый сигнал  $S$  будет совпадать с этим кодовым словом или незначительно отличаться от него. Тогда невязка будет равна нулю или минимальна именно для  $l = k$ .

Таким образом, оптимальный декодер должен вычислить евклидовы расстояния между принятым сигналом  $S$  и всеми возможными кодовыми словами  $U_k$  данного кода и принять решение в пользу кодового слова  $U_l$ , минимизирующего  $d_l^2$ , то есть наиболее похожего на принятый сигнал.

Структурная схема декодера максимального правдоподобия, реализующего правило декодирования (1.33) – (1.34), приведена на рис. 1.8.

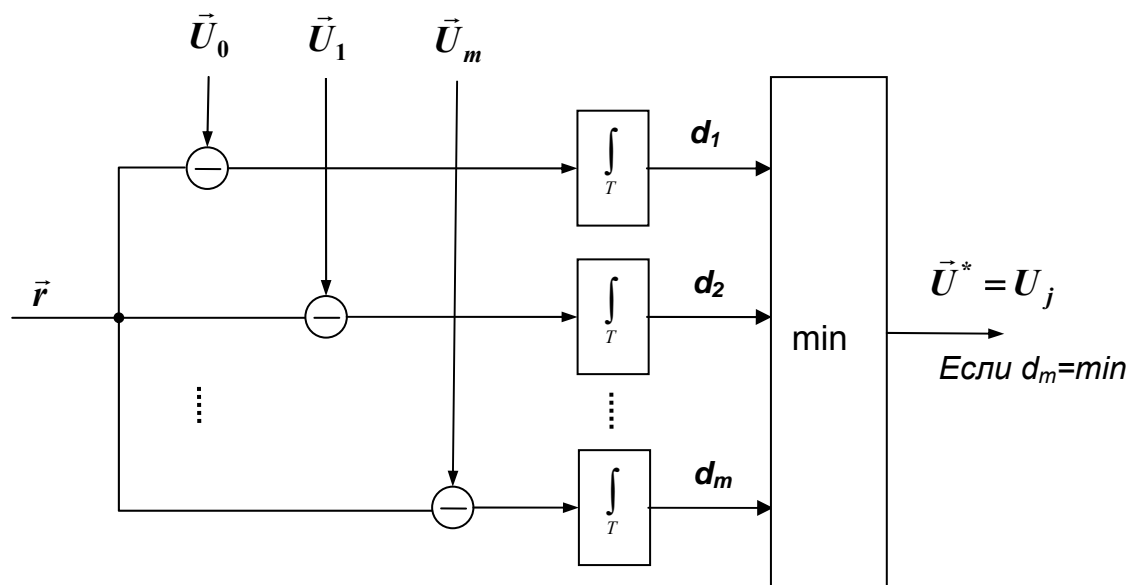


Рис. 1.8

Рассмотренный нами оптимальный декодер является так называемым *мягким декодером*, поскольку он выносит решения относительно  $U_l$  непосредственно на основе принятого сигнала.

На практике чаще применяется так называемое *жесткое декодирование*, когда в приемнике сначала принимается решение относительно значения символов принятой последовательности, а уже затем – относительно значения кодового слова.

В этом случае оптимальный декодер (*жесткий декодер максимального правдоподобия*) должен вычислить расстояния

$$d^*_k = \sum \{ r_l - U_k \}^2 \quad (1.35)$$

между *принятой последовательностью  $r$*  и всеми возможными кодовыми словами  $U_k$  данного кода и принять решение в пользу кодового слова, в минимальной степени отличающегося от принятой последовательности.

Таким образом, при жестком декодировании максимального правдоподобия по принятому сигналу сначала определяются символы принятой последовательности  $r$ , а потом эта последовательность поочередно сравнивается со всеми кодовыми словами данного кода. Решение принимается в пользу кодового слова, максимально похожего на принятую последовательность.

Поскольку в процессе мягкого декодирования информация о сигнале учитывается в большей мере (решение принимается по всему сигналу сразу, а не по частям, для каждого символа в отдельности, и только потом - для всей принятой последовательности), то качество мягкого декодирования должно быть, по идее, выше. Однако реализация жесткого декодера является гораздо более простой – действия выполняются над нулями и единицами. Поэтому такие декодеры используются чаще, хотя и несколько проигрывают мягким декодерам в вероятности правильного декодирования.

В заключение нужно сказать, что при декодировании блочных кодов декодеры максимального правдоподобия применяются достаточно редко из-за их сложности при больших размерах кода. Правда, при современных мощностях микропроцессорных устройств это уже не представляет непреодолимой трудности. Скорее, нужно выбирать между усложнением алгоритма декодирования и выигрышем в повышении вероятности правильного декодирования.

Для сверточных же кодов декодирование с использованием метода максимального правдоподобия – стек-алгоритм, алгоритм Фано и алгоритм Витерби - это основные способы декодирования.

### 1.2.10. Вес и расстояние Хемминга. Способность кодов обнаруживать и исправлять ошибки

Рассмотрим, чем определяется способность блочного кода обнаруживать и исправлять ошибки, возникшие при передаче.

Пусть  $U = (U_0, U_1, U_2, \dots, U_{n-1})$  - двоичная последовательность длиной  $n$ .

Число единиц (ненулевых компонент) в этой последовательности называется *весом Хемминга* вектора  $U$  и обозначается  $w(U)$ .

Например, вес Хемминга вектора  $U = (1001011)$  равен четырем, для вектора  $U = (1111111)$  величина  $w(U)$  составит 7 и т.д.

Таким образом, чем больше единиц в двоичной последовательности, тем больше ее вес Хемминга.

Далее, пусть  $U$  и  $V$  будут двоичными последовательностями длиной  $n$ .

Число разрядов, в которых эти последовательности различаются, называется *расстоянием Хемминга* между  $U$  и  $V$  и обозначается  $d(U, V)$ .

Например, если  $U = (1001011)$ , а  $V = (0100011)$ , то  $d(U, V) = 3$ .

Задав линейный код, то есть определив все  $2^k$  его кодовых слов, можно вычислить расстояние между всеми возможными парами кодовых слов. Минимальное из них называется *минимальным кодовым расстоянием кода* и обозначается  $d_{min}$ .

Можно проверить и убедиться, что минимальное кодовое расстояние для рассматриваемого нами в примерах  $(7,4)$ -кода равно трем:  $d_{min(7,4)} = 3$ . Для этого нужно записать все кодовые слова  $(7,4)$ -кода Хемминга (всего 16 слов), вычислить расстояния между их всеми парами и взять наименьшее значение. Однако можно определить  $d_{min}$  блочного кода и более простым способом.

Доказано, что расстояние между нулевым кодовым словом и одним из кодовых слов, входящих в порождающую матрицу (строки порождающей матрицы линейного блочного кода сами являются кодовыми словами, по определению), равно  $d_{min}$ . Но расстояние от любого кодового слова до нулевого равно весу Хемминга этого слова. Тогда  $d_{min}$  равно *минимальному весу Хемминга для всех строк порождающей матрицы кода*.

Если при передаче кодового слова по каналу связи в нем произошла *одиночная ошибка*, то расстояние Хемминга между переданным словом  $U$  и принятым вектором  $r$  будет равно единице. Если при этом одно кодовое слово не перешло в другое (а при  $d_{min} > 1$  и при одиночной ошибке это невозможно), то ошибка *будет обнаружена* при декодировании.

В общем случае если блочный код имеет минимальное расстояние  $d_{min}$ , то он *может обнаруживать любые сочетания ошибок при их числе, меньшем или равном  $d_{min} - 1$* , поскольку никакое сочетание ошибок при их

числе, меньшем, чем  $d_{min} - 1$ , не может перевести одно кодовое слово в другое.

Но ошибки могут иметь кратность и большую, чем  $d_{min} - 1$ , и тогда они останутся необнаруженными.

При этом среднюю вероятность необнаруживаемой ошибки можно определить следующим образом.

Пусть вероятность ошибки в канале связи равна  $P_{ош}$ . Тогда вероятность того, что при передаче последовательности длины  $n$  в ней произойдет одна ошибка, равна

$$P_1 = n P_{ош} \cdot (1 - P_{ош})^{n-1}, \quad (1.36)$$

соответственно, вероятность  $l$ -кратной ошибки -

$$P_l = C_n^l P_{ош}^l \cdot (1 - P_{ош})^{n-l}, \quad (1.37)$$

где  $C_n^l$  - число возможных комбинаций из  $n$  символов кодовой последовательности по  $l$  ошибок.

По каналу связи передаются кодовые слова с различными весами Хемминга. Положим, что  $a_i$  — число слов с весом  $i$  в данном коде (всего слов в коде длиной  $n$  -  $A = \sum_{i=0}^{n-1} d_i = 2^k$ ).

А теперь определим, что такое *необнаруживаемая ошибка*. Обнаружение ошибки производится путем вычисления синдрома принятой последовательности. Если принятая последовательность не является кодовым словом (тогда синдром не равен нулю), то считается, что ошибка есть. Если же синдром равен нулю, то полагаем, что ошибки нет (принятая последовательность является кодовым словом). Но тем ли, которое передавалось? Или же в результате действия ошибок переданное кодовое слово перешло в другое кодовое слово данного кода:

$$r = U + e = V, \quad (1.38)$$

то есть сумма переданного кодового слова  $U$  и вектора ошибки  $e$  даст новое кодовое слово  $V$ ? В этом случае, естественно, ошибка обнаружена быть не может.

Но из определения двоичного линейного кода следует, что *если сумма кодового слова и некоторого вектора  $e$  есть кодовое слово, то вектор  $e$  также представляет собой кодовое слово. Следовательно, необнаруживаемые ошибки будут возникать тогда, когда сочетания ошибок будут образовывать кодовые слова.*

Вероятность того, что вектор  $e$  совпадает с кодовым словом, имеющим вес  $i$ , равна

$$P_i = P_{ош}^i \cdot (1 - P_{ош})^{n-i}. \quad (1.39)$$

Тогда полная вероятность возникновения необнаруживаемой ошибки

$$P(E) = \sum_{i=1}^n a_i \cdot P_{ош}^i (1 - P_{ош})^{n-i}. \quad (1.40)$$

*Пример:* рассматриваемый нами (7,4)-код содержит по семь кодовых слов с весами  $w = 3$  и  $w = 4$  и одно кодовое слово с весом  $w = 7$ , тогда

$$P(E)_{(7,4)} = 7 \cdot P_{ош}^3 (1 - P_{ош})^4 + 7 \cdot P_{ош}^4 \cdot (1 - P_{ош})^3 + P^7 \quad (1.41)$$

или, при  $P_{ош} = 10^{-3}$ ,  $P(E) \cong 7 \cdot 10^{-9}$ .

Другими словами, если по каналу передается информация со скоростью  $V = 1$  кбит/с и в канале в среднем *каждую секунду* будет происходить искажение одного символа, то в среднем семь принятых слов на  $10^9$  переданных будут проходить через декодер без обнаружения ошибки (одна необнаруживаемая ошибка за 270 часов).

Таким образом, использование даже такого простого кода позволяет на несколько порядков снизить вероятность необнаруживаемых ошибок.

Теперь предположим, что линейный блочный код используется для исправления ошибок. Чем определяются его возможности по исправлению?

Рассмотрим пример, приведенный на рис. 1.9. Пусть  $U$  и  $V$  представляют пару кодовых слов кода с кодовым расстоянием  $d$ , равным минимальному —  $d_{min}$  для данного кода.

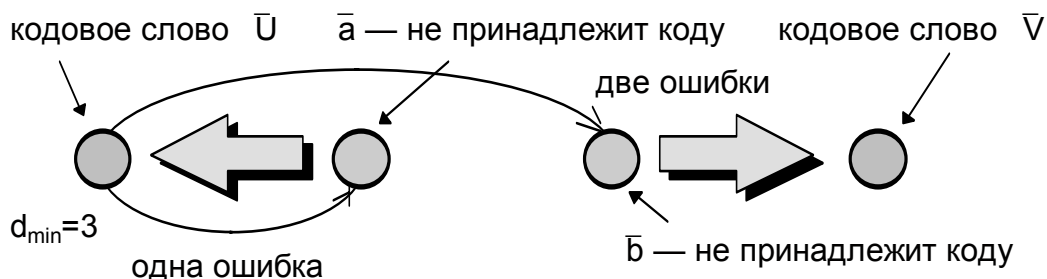


Рис. 1.9

Предположим, передано кодовое слово  $U$ , в канале произошла *одиночная ошибка* и принят вектор  $a$  (не принадлежащий коду).

Если декодирование производится оптимальным способом, то есть по методу максимального правдоподобия, то в качестве оценки  $U^*$  нужно выбрать *ближайшее к  $a$  кодовое слово*.

Таковым в данном случае будет  $U$ , следовательно, ошибка будет устранена.

Представим теперь, что произошло две ошибки и принят вектор  $b$ .

Тогда при декодировании по максимуму правдоподобия в качестве оценки будет выбрано ближайшее к  $\mathbf{b}$  кодовое слово, и им будет  $\mathbf{V}$ . Произойдет ошибка декодирования.

Продолжив рассуждения для  $d_{min} = 4$ ,  $d_{min} = 5$  и т.д., нетрудно сделать вывод, что ошибки будут устранены, если их кратность  $l$  не превышает величины

$$l < INT((d_{min} - 1)/2), \quad (1.41)$$

где  $INT(X)$  — целая часть  $X$ .

Так, используемый нами в качестве примера (7,4)-код имеет  $d_{min} = 3$  и, следовательно, позволяет исправлять лишь одиночные ошибки:

$$l = INT((d_{min} - 1)/2) = INT((3-1)/2) = 1. \quad (1.42)$$

Таким образом, возможности линейных блочных кодов по обнаружению и исправлению ошибок определяются их минимальным кодовым расстоянием. Чем больше  $d_{min}$ , тем большее число ошибок в принятой последовательности можно исправить.

А теперь определим вероятность того, что возникшая в процессе передачи ошибка не будет все же исправлена при декодировании.

Пусть, как и ранее, вероятность ошибки в канале будет равна  $P_{ош}$ . Ошибки, возникающие в различных позициях кода, считаем независимыми.

Вероятность того, что принятый вектор  $\mathbf{r}$  будет иметь какие-нибудь (одиночные, двукратные, трехкратные и т.д.) ошибки, можно определить как

$$P_{ош} = P_1 + P_2 + P_3 + \dots + P_n, \quad (1.43)$$

где  $P_1$  — вероятность того, что в  $\mathbf{r}$  присутствует одиночная ошибка;

$P_2$  — вероятность того, что ошибка двойная и т.д.;

$P_n$  — вероятность того, что все  $n$  символов искажены.

Определим вероятность ошибок заданной кратности:

$$\begin{aligned} P_1 &= \text{Вер}\{\text{ошибка в 1-й позиции ИЛИ ошибка во 2-й позиции ..ИЛИ в } n\text{-й позиции}\} = \\ &= P_{ош}(1 - P_{ош})^{n-1} + P_{ош}(1 - P_{ош})^{n-1} + \dots + P_{ош}(1 - P_{ош})^{n-1} = n \cdot P_{ош}(1 - P_{ош})^{n-1}; \end{aligned} \quad (1.44)$$

$$\begin{aligned} P_2 &= \text{Вер}\{\text{ошибка в 1-й И во 2-й позиции ИЛИ ошибка во 2-й И в 3-й позиции...}\} = \\ &= P_{ош}^2(1 - P_{ош})^{n-2} + \dots + P_{ош}^2(1 - P_{ош})^{n-2} = C_2^n P_{ош}^2(1 - P_{ош})^{n-2}. \end{aligned} \quad (1.45)$$

Аналогичным образом

$$P_3 = C_3^n P_{ош}^3(1 - P_{ош})^{n-3} \text{ и т.д.} \quad (1.46)$$

Декодер, как мы показали, исправляет все ошибки, кратность которых не превышает

$$l \leq INT \left[ \frac{d_{min} - 1}{2} \right], \quad (1.47)$$

то есть все ошибки кратности  $J \leq l$  будут исправлены.

Тогда ошибки декодирования - это ошибки с кратностью, большей кратности исправляемых ошибок  $l$ , и их вероятность

$$P(N) = \sum_{j=l-1}^n C_j^n \cdot P_{ош}^j \cdot (1 - P_{ош})^{n-j}. \quad (1.48)$$

Для (7,4)-кода Хемминга минимальное расстояние  $d_{min} = 3$ , т.е.  $l = 1$ . Следовательно, ошибки кратности 2 и более исправлены не будут и

$$P(N)_{(7.4)} = \sum_{j=2}^7 C_j^7 \cdot P_{ош}^j \cdot (1 - P_{ош})^{j-n}. \quad (1.49)$$

Если  $P_{ош} \ll 1$ , можно считать  $(1 - P_{ош}) \approx 1$  и, кроме того,  $P_{ош}^3 \ll P_{ош}^2$ . Тогда

$$P(N)_{(7.4)} \approx C_2^7 \cdot P_{ош}^2 \approx 21P^2. \quad (1.50)$$

Так, например, при вероятности ошибки в канале  $P_{ош} = 10^{-3}$  вероятность неисправления ошибки  $P(N) \approx 2 \cdot 10^{-5}$ , то есть при такой вероятности ошибок в канале кодирование (7,4)-кодом позволяет снизить вероятность оставшихся неисправленными ошибок примерно в пятьдесят раз.

Если же вероятность ошибки в канале будет в сто раз меньше  $P_{ош} = 10^{-5}$ , то вероятность ее неисправления составит уже  $P(N) \approx 2 \cdot 10^{-9}$ , или в 5000 раз меньше!

Таким образом, выигрыш от помехоустойчивого кодирования (который можно определить как отношение числа ошибок в канале к числу оставшихся неисправленными ошибок) существенно зависит от свойств канала связи.

Если вероятность ошибок в канале велика, то есть канал не очень хороший, ожидать большого эффекта от кодирования не приходится, если же вероятность ошибок в канале мала, то корректирующее кодирование уменьшает ее в значительно большей степени.

Другими словами, помехоустойчивое кодирование существенно улучшает свойства хороших каналов, в плохих же каналах оно большого эффекта не дает.

### 1.3. Полиномиальные коды

Представление кодового слова  $(n,k)$ -кода в виде последовательности  $U = (U_0, U_1, \dots, U_{n-1})$  длиной  $n$  символов или их задание с помощью системы проверочных уравнений и порождающей матрицы не является единственно возможным. Еще один удобный и широко используемый способ представления того же кодового слова состоит в том, что элементы  $U_0, U_1, \dots, U_{n-1}$  являются коэффициентами многочлена от  $X$ , то есть

$$U(x) = f(x) = U_0 + U_1 \cdot X + U_2 \cdot X^2 + \dots + U_{n-1} \cdot X^{n-1} . \quad (1.51)$$

Используя это представление, можно определить полиномиальный код как множество всех многочленов степени, не большей  $n - 1$ , содержащих в качестве общего множителя некоторый фиксированный многочлен  $g(x)$ .

Многочлен  $g(x)$  называется порождающим многочленом кода.

Представление кодовых слов в такой форме позволяет свести действия над комбинациями символов к действию над полиномами.

Определим действия над полиномами в поле двоичных символов  $GF(2)$ .

Суммой двух полиномов  $f(x)$  и  $g(x)$  из  $GF(2)$  называется полином из  $GF(2)$ , определяемый следующим образом:

$$f(x) + g(x) = \sum_{i=0}^{n-1} (f_i + g_i) \cdot x^i . \quad (1.52)$$

Другими словами, сложению двоичных полиномов соответствует сложение по *mod2* коэффициентов при одинаковых степенях  $x$ .

Например:

$$\begin{array}{r} X^3 + X^2 + 0 \cdot X + 1 \\ X + 1 \\ \hline X^3 + X^2 + X + 0 \end{array} = X^3 + X^2 + X , \quad (1.53)$$

$$\begin{array}{r} X^3 + X^2 + 0 \cdot X + 1 \\ X^2 + X + 1 \\ \hline X^3 + 0 + X + 0 \end{array} = X^3 + X . \quad (1.54)$$

Произведением двух полиномов из  $GF(2)$  называется полином из  $GF(2)$ , определяемый следующим образом :

$$f(x) \cdot g(x) = \sum_{i=0}^{n-1} \left( \sum_{j=0}^i f_j \cdot g_{i-j} \right) \cdot x^i , \quad (1.55)$$

то есть произведение получается по обычному правилу перемножения степенных функций, однако получаемые коэффициенты при данной степени  $X$  складываются по модулю 2.

Например:

$$\begin{array}{r} X^3 + X^2 + 0 + 1 \\ \hline X + 1 \\ X^3 + X^2 + 0 + 1 \\ \hline X^4 + X^3 + 0 + X \\ X^4 + 0 + X^2 + X + 1 = X^4 + X^2 + X + 1, \end{array} \quad (1.56)$$

$$\begin{array}{r} X^3 + X^2 + 0 + 1 \\ \hline X^2 + X \\ X^4 + X^3 + 0 + X \\ \hline X^5 + X^4 + 0 + X^2 \\ X^5 + 0 + X^3 + X^2 + X = X^5 + X^3 + X^2 + X. \end{array} \quad (1.57)$$

Наконец, можно сформулировать теорему о делении полиномов: для каждой пары полиномов  $C(x)$  и  $d(x)$ ,  $d(x) \neq 0$  существует единственная пара полиномов  $q(x)$  — частное и  $\rho(x)$  — остаток, такие, что

$$C(x) = q(x) \cdot d(x) + \rho(x), \quad (1.58)$$

где степень остатка  $\rho(x)$  меньше степени делителя  $d(x)$ .

Иными словами, деление полиномов производится по правилам деления степенных функций, при этом операция вычитания заменяется суммированием по **mod2**.

Например:

$$\begin{array}{r} X^4 + 0 + X^2 + X + 1 \\ \hline X^4 + X^3 \\ \hline X^3 + X^2 + X + 1 \\ \hline X^3 + X^2 \\ \hline X + 1 \\ \hline X + 1 \\ \hline 0 \leftarrow \text{остаток } \rho(x). \end{array} \quad (1.59)$$

Еще раз напомним, что при сложении по **mod2** сумма двух единиц (то есть двух элементов полинома с одинаковыми степенями) будет равна нулю, а не привычным в десятичной системе счисления двум. И, кроме этого, операции вычитания и сложения по **mod2** совпадают.

### 1.3.1. Циклические коды

Частным и наиболее широко распространенным классом полиномиальных кодов являются *циклические коды*.

Линейный  $(n, k)$ -код называется *циклическим*, если в результате циклического сдвига кодового слова получается другое кодовое слово данного кода. Другими словами, если  $U = (U_0, U_1, \dots, U_{n-1})$  является кодовым словом, то и  $V = (U_{n-1}, U_0, U_1, \dots, U_{n-2})$ , полученное циклическим сдвигом  $U$ , является кодовым словом данного кода.

Циклические коды привлекательны по двум причинам.

Во-первых, операции кодирования и вычисления синдрома для них выполняются очень просто с использованием сдвиговых регистров.

Во-вторых, этим кодам присуща алгебраическая структура, и можно найти более простые и эффективные способы их декодирования.

Основные свойства циклических кодов:

1. В циклическом  $(n, k)$ -коде каждый ненулевой полином должен иметь степень, по крайней мере  $(n-k)$ , но не более  $n-1$ .
2. Существует только один кодовый полином  $g(x)$  степени  $(n-k)$

$$g(x) = 1 + g_1 \cdot x + g_2 \cdot x^2 + \dots + g_{n-k-1} \cdot x^{n-k-1} + x^{n-k}, \quad (1.60)$$

вида  
называемый порождающим полиномом кода.

3. Каждый кодовый полином  $U(x)$  является кратным  $g(x)$ , то есть

$$U(x) = m(x) \cdot g(x). \quad (1.61)$$

### 1.3.2. Кодирование с использованием циклических кодов

Предположим, надо закодировать некоторую информационную последовательность

$$m = (m_0, m_1, m_2, \dots, m_{k-1}). \quad (1.62)$$

Соответствующий ей полином выглядит следующим образом:

$$m(x) = m_0 + m_1 \cdot x + m_2 \cdot x^2 + \dots + m_{k-1} \cdot x^{k-1}. \quad (1.63)$$

Умножив  $m(x)$  на  $x^{n-k}$ :

$$x^{n-k} \cdot m(x) = m_0 \cdot x^{n-k} + m_1 \cdot x^{n-k+1} + \dots + m_{k-1} \cdot x^{n-1}, \quad (1.64)$$

получим полином степени  $n-1$  или меньшей.

Воспользовавшись теоремой о делении полиномов, можно записать

$$x^{n-k} \cdot m(x) = q(x) \cdot g(x) + \rho(x), \quad (1.65)$$

где  $q(x)$  и  $\rho(x)$  — частное и остаток от деления полинома  $x^{n-k} \cdot m(x)$  на порождающий полином  $g(x)$ .

Поскольку степень  $g(x)$  равна  $(n-k)$ , то степень  $\rho(x)$  должна быть  $(n-k-1)$  или меньше, а сам полином  $\rho(x)$  будет иметь вид

$$\rho(x) = \rho_0 + \rho_1 \cdot x + \rho_2 \cdot x^2 + \dots + \rho_{n-k-1} \cdot x^{n-k-1}. \quad (1.66)$$

С учетом правил арифметики в  $GF(2)$  данное выражение можно переписать следующим образом:

$$\rho(x) + x^{n-k} \cdot m(x) = q(x) \cdot g(x), \quad (1.67)$$

откуда видно, что полином  $\rho(x) + x^{n-k} \cdot m(x)$  является кратным  $g(x)$  и имеет степень  $n-1$  или меньшую. Следовательно, полином  $\rho(x) + x^{n-k} \cdot m(x)$  — это кодовый полином, соответствующий кодируемой информационной последовательности  $m(x)$ .

Раскрыв последнее выражение, получим

$$\rho(x) + m(x) \cdot x^{n-k} = \rho_0 + \rho_1 x + \rho_2 x^2 + \dots + \rho_{n-k-1} x^{n-k-1} + m_0 x^{n-k} + m_1 \cdot x^{n-k+1} + \dots + m_{k-1} x^{n-1},$$

что соответствует кодовому слову

$U =$	$(\rho_0, \rho_1 \dots \rho_{n-k-1},$	$m_0, m_1 \dots m_{k-1}),$
	<i>проверочные символы</i>	<i>информационные символы</i>

Таким образом, кодовое слово состоит из неизменной информационной части  $m$ , перед которой расположено  $(n-k)$  проверочных символов. Проверочные символы являются коэффициентами полинома  $\rho(x)$ , то есть остатка от деления  $m(x) \cdot x^{n-k}$  на порождающий полином  $g(x)$ .

Чтобы полученный результат был понятнее, напомним, что умножению некоторого двоичного полинома на  $x^{n-k}$  соответствует сдвиг двоичной последовательности  $m = (m_0, m_1 \dots m_{k-1})$  на  $n-k$  разрядов вправо.

Рассмотрим пример. С использованием кода, задаваемого порождающим полиномом  $g(x) = 1 + x + x^3$ , закодируем произвольную последовательность, например  $m = (0111)$ .

Последовательности  $m = (0111)$  соответствует полином  $m(x) = x + x^2 + x^3$ .

Умножим  $m(x)$  на  $x^{n-k}$  :

$$m(x) \cdot x^{n-k} = m(x) \cdot x^3 = (x + x^2 + x^3) \cdot x^3 = x^4 + x^5 + x^6. \quad (1.68)$$

Разделим  $m(x) \cdot x^{n-k}$  на порождающий полином  $g(x)$  :

$$\begin{array}{r} X^6 + X^5 + X^4 \\ \hline X^6 + 0 + X^4 + X^3 \\ \hline X^5 + 0 + X^3 \\ \hline X^5 + 0 + X^3 + X^2 \\ \hline X^2 = \rho(x). \end{array} \left| \begin{array}{l} X^3 + X + 1 \\ \hline X^3 + X^2 = g(x) \end{array} \right. \quad (1.69)$$

Таким образом, кодовый полином, соответствующий информационной последовательности  $m = (0111)$ , будет иметь следующий вид:

$$U(x) = 0 \cdot X^0 + 0 \cdot X^1 + 1 \cdot X^2 + 0 \cdot X^3 + 1 \cdot X^4 + 1 \cdot X^5 + 1 \cdot X^6, \quad (1.70)$$

а соответствующее кодовое слово  $U = (0010111)$ .

Итак, циклический  $(n,k)$ -код  $k$ -разрядной информационной последовательности  $m = (m_0, m_1 \dots m_{k-1})$  получают следующим образом:

- информационную последовательность  $m$  умножают на  $x^{n-k}$ , то есть сдвигают вправо на  $n-k$  разрядов;
- полином полученной последовательности делят на порождающий полином кода  $g(x)$ ;
- полученный остаток от деления  $m(x) \cdot x^{n-k}$  на  $g(x)$  прибавляют к  $m(x) \cdot x^{n-k}$ , то есть записывают в младших  $n-k$  разрядах кода.

Алгоритм кодирования, основанный на делении полиномов, можно реализовать, используя схему деления. Она представляет собой регистр сдвига, в котором цепи обратной связи замкнуты в соответствии с коэффициентами порождающего полинома  $g(x)$  (рис. 1.10).

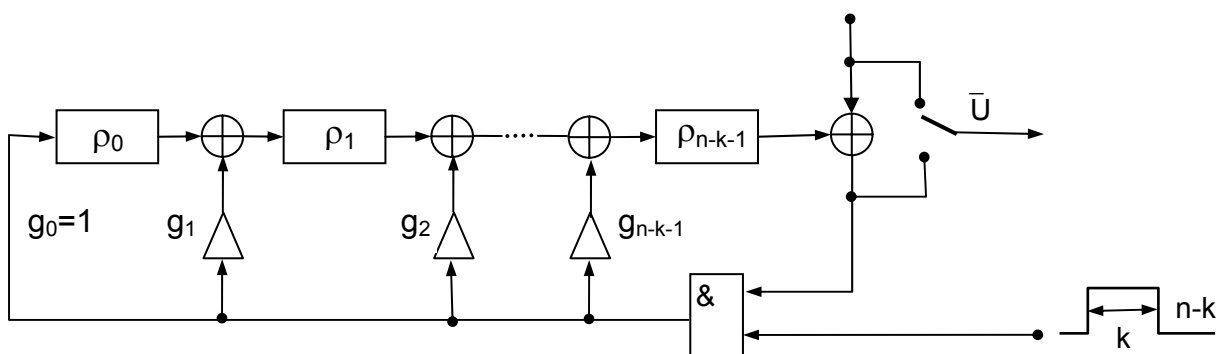


Рис. 1.10

Кодирование в схеме рис. 1.10 выполняется следующим образом:

-  $k$  символов информационной последовательности  $m$  через переключатель  $\Pi$ , находящийся в верхнем положении, один за другим передаются в канал и одновременно с этим через открытую схему  $\Pi$  записываются в регистр проверочных символов, в котором благодаря наличию цепей обратной связи  $g_0 \dots g_{n-k-1}$  формируется остаток от деления  $x^{n-k} \cdot m(x)$  на  $g(x)$  — проверочные символы;

- начиная с  $(k+1)$ -го такта переключатель переводится в нижнее положение, и из сдвигового регистра выводятся  $(n-k)$  проверочных символов; цепь обратной связи при этом разомкнута ( схема  $\Pi$  - закрыта ).

Для циклического  $(7,4)$ -кода Хемминга (а этот код обладает свойством цикличности), используемого в качестве примера и имеющего порождающий полином  $g(x) = 1 + x + x^3$ , схема кодирования выглядит следующим образом (рис. 1.11):

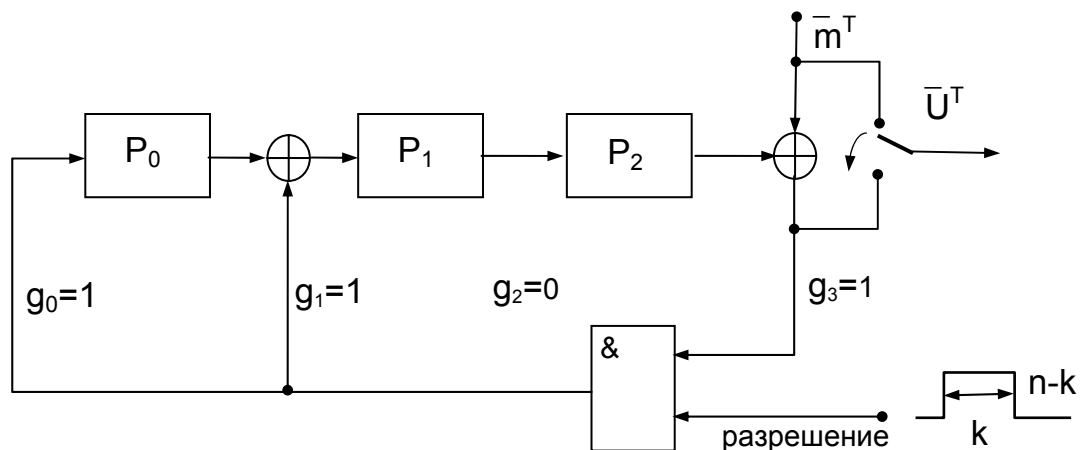


Рис. 1.11

В этой схеме, в отличие от обобщенной схемы кодера рис. 1.10, отсутствуют элементы в цепях, где значения коэффициентов обратной связи  $g_i$  равны нулю, там же, где коэффициенты передачи  $g_i$  равны единице, цепь просто замкнута.

На примере этого же кода и соответствующего ему кодера рассмотрим в динамике процесс кодирования произвольной последовательности  $m$ .

Пусть  $m = (1001)$ . Тогда последовательность состояний ячеек сдвигового регистра с обратными связями в процессе кодирования будет определяться табл. 1.4 .

Таблица 1.4

Номер такта	Положение переключателя	Уровень разрешения	Вход $m$	Состояние ячейки регистра			Выход $U$
				$P_0$	$P_1$	$P_2$	
0	↑	1	1	1	1	0	1
1	↑	1	0	0	1	1	0
2	↑	1	0	1	1	1	0
3	↑	1	1	0	1	1	1
4	↓	0	0	0	0	1	1
5	↓	0	0		0	0	1
6	↓	0	0			0	0
7	↓	0	0				

Еще одним важным свойством циклического  $(n, k)$ -кода, вытекающим из теоремы о существовании циклических кодов, является то, что его порождающий полином делит без остатка двучлен  $x^n + 1$ , то есть

$$x^n + 1 = g(x) \cdot h(x) + 0. \quad (1.71)$$

Полином  $h(x)$  — частное от деления  $x^n + 1$  на  $g(x)$  — называется проверочным полиномом.

Поскольку  $h(x)$  однозначно связан с  $g(x)$ , он также определяет код. Следовательно, с помощью проверочного полинома  $h(x)$  тоже можно производить кодирование. Схема кодирования на основании проверочного полинома  $h(x)$  приведена на рис. 1.12.

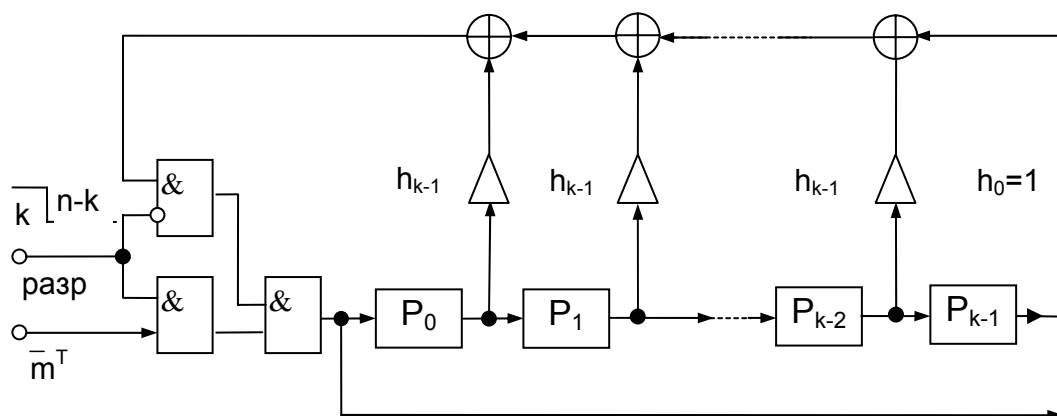


Рис. 1.12

Процедура кодирования на основании  $h(x)$  выглядит следующим образом :

1. На входе "Разрешение" устанавливается 1, при этом открывается нижняя схема **И** и закрывается верхняя.

2. Сообщение  $m$  последовательно записывается в  $k$ -разрядный сдвиговый регистр и одновременно с этим передается в канал.

3. По окончании ввода  $k$  информационных символов на входе "Разрешение" устанавливается 0, замыкая через верхнюю схему И цепь обратной связи.

4. Производится  $(n-k)$  сдвигов, при этом формируются и выдаются в канал  $(n-k)$  проверочных символов.

Для циклического  $(7,4)$ -кода с порождающим полиномом  $g(x) = 1 + x + x^3$  проверочный полином  $h(x)$  имеет вид

$$h(x) = \frac{x^{n-1}}{1 + x + x^3} = 1 + x + x^2 + x^4. \quad (1.72)$$

С учетом этого схема кодирования на основании полинома  $h(x)$  для  $(7,4)$ -кода выглядит следующим образом (рис. 1.13):

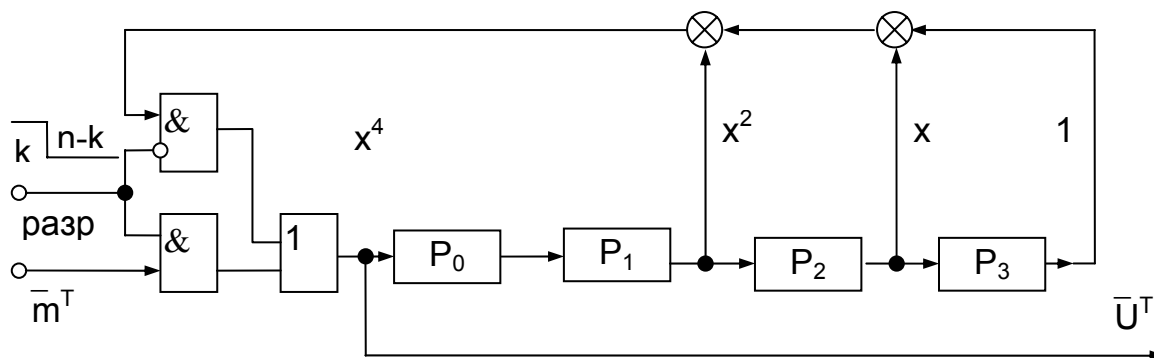


Рис. 1.13

### 1.3.3. Вычисление синдрома и исправление ошибок в циклических кодах

Вычисление синдрома для циклических кодов является довольно простой процедурой.

Пусть  $U(x)$  и  $r(x)$  - полиномы, соответствующие переданному кодовому слову и принятой последовательности.

Разделив  $r(x)$  на  $g(x)$ , получим

$$r(x) = q(x) \cdot g(x) + s(x), \quad (1.73)$$

где  $q(x)$  — частное от деления,  $s(x)$  — остаток от деления.

Если  $r(x)$  является кодовым полиномом, то он делится на  $g(x)$  без остатка, то есть  $s(x) = 0$ .

Следовательно,  $s(x) \neq 0$  является условием наличия ошибки в принятой последовательности, то есть синдромом принятой последовательности.

Синдром  $s(x)$  имеет в общем случае вид

$$S(x) = S_0 + S_1 \cdot x + \dots + S_{n-k-1} \cdot x^{n-k-1}. \quad (1.74)$$

Очевидно, что схема вычисления синдрома является схемой деления, подобной схемам кодирования рис. 1.10 или 1.11.

При наличии в принятой последовательности  $r$  хотя бы одной ошибки вектор синдрома  $S$  будет иметь, по крайней мере, один нулевой элемент, при этом факт наличия ошибки легко обнаружить, объединив по **ИЛИ** выходы всех ячеек регистра синдрома.

Покажем, что синдромный многочлен  $S(x)$  однозначно связан с многочленом ошибки  $e(x)$ , а значит, с его помощью можно не только обнаруживать, но и локализовать ошибку в принятой последовательности.

Пусть

$$e(x) = e_0 + e_1 \cdot x + e_2 \cdot x^2 + \dots + e_{n-1} \cdot x^{n-1} \quad (1.75)$$

— полиномом вектора ошибки.

Тогда полином принятой последовательности

$$r(x) = U(x) + e(x). \quad (1.76)$$

Прибавим в этом выражении слева и справа  $U(x)$ , а также учтем, что

$$r(x) = q(x) \cdot g(x) + S(x), \quad U(x) = m(x) \cdot g(x), \quad (1.77)$$

тогда

$$e(x) = [m(x) + q(x)] \cdot g(x) + S(x) = f(x) \cdot g(x) + S(x), \quad (1.78)$$

то есть синдромный полином  $S(x)$  есть остаток от деления полинома ошибки  $e(x)$  на порождающий полином  $g(x)$ .

Отсюда следует, что по синдрому  $S(x)$  можно однозначно определить вектор ошибки  $e(x)$ , а следовательно, исправить эту ошибку.

На рис. 1.14 приведена схема синдромного декодера с исправлением однократной ошибки для циклического (7,4)-кода. По своей структуре она подобна схеме, приведенной на рис. 1.6, с той лишь разницей, что вычисление синдрома принятой последовательности производится здесь не умножением на проверочную матрицу, а делением на порождающий полином. При этом она сохраняет и недостаток, присущий всем синдромным декодерам: необходимость иметь запоминающее устройство, ставящее в соответствие множеству возможных синдромов  $S$  множество векторов ошибок  $e$ . Цикличность структуры кода в этом случае не учитывается.

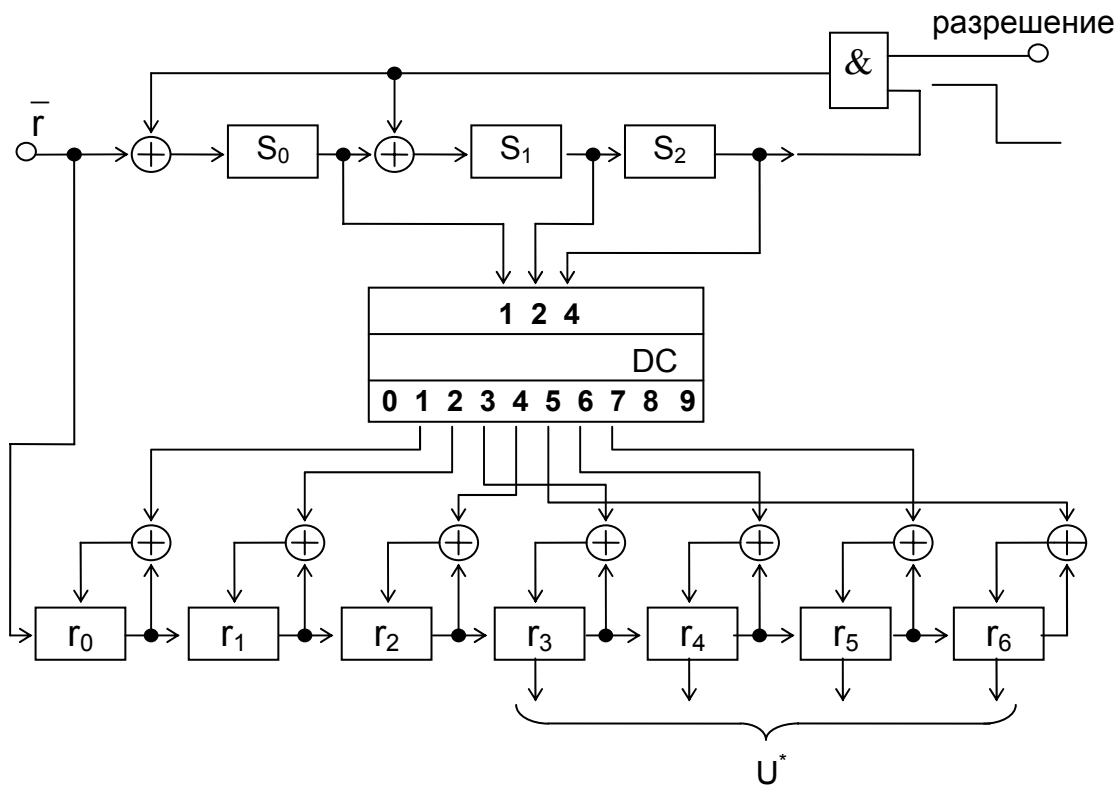


Рис. 1.14

#### 1.3.4. Неалгебраические методы декодирования циклических кодов

Все методы декодирования линейных блочных кодов можно разбить на две группы: *алгебраические и неалгебраические*.

В основе алгебраических методов лежит решение систем уравнений, задающих значение и расположение ошибок. Рассмотренные синдромные декодеры относятся именно к этой группе методов.

При неалгебраических методах та же цель достигается с помощью простых структурных понятий теории кодирования, позволяющих находить комбинации ошибок более простым путем.

Одним из неалгебраических методов является декодирование с использованием алгоритма *Меггитта*, пригодного для исправления как одиночных, так и  $l$ -кратных ошибок (на практике  $l \leq 3$ ).

При декодировании в соответствии с алгоритмом Меггитта также вычисляется синдром принятой последовательности  $S(x)$ , однако используется он иначе, нежели в рассмотренных ранее синдромных декодерах.

*Идея, лежащая в основе декодера Меггитта, очень проста и основывается на следующих свойствах циклических кодов:*

- существует взаимно-однозначное соответствие между множеством всех исправляемых ошибок и множеством синдромов;

- если  $S(x)$  — синдромный многочлен, соответствующий многочлену ошибок  $e(x)$ , то  $x \cdot S(x) \bmod g(x)$  — синдромный многочлен, соответствующий  $x \cdot e(x) \bmod (x^n + 1)$ .

Равенство  $a(x) = b(x) \bmod p(x)$  читается как “ $a(x)$ , сравнимо с  $b(x)$  по модулю  $p(x)$ ” и означает, что  $a(x)$  и  $b(x)$  имеют одинаковые остатки от деления на полином  $p(x)$ .

Таким образом, второе условие означает, что если комбинация ошибок циклически сдвинута на одну позицию вправо, то для получения нового синдрома нужно сдвинуть содержимое регистра сдвига с обратными связями, содержащего  $S(x)$ , также на одну позицию вправо.

Следовательно, основным элементом декодера Меггитта является сдвиговый регистр. Структурная схема декодера Меггитта для циклических кодов произвольной длины приведена на рис. 1.15.

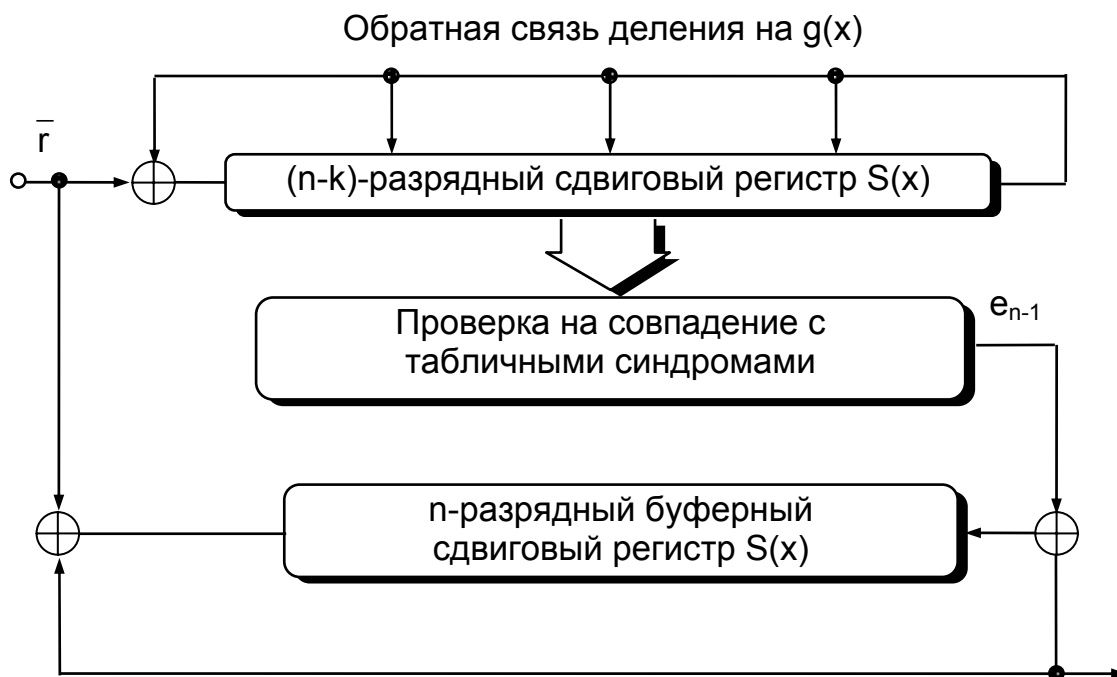


Рис. 1.15

Декодер работает следующим образом. Перед началом работы содержимое всех разрядов регистров равно нулю. Принимаемая последовательность  $r$  в течение первых  $n$  тактов вводится в буферный регистр и одновременно с этим в  $(n-k)$ -разрядном сдвиговом регистре с обратными связями производится ее деление на порождающий полином  $g(x)$ .

Через  $n$  тактов в буферном регистре оказывается принятое слово  $r$ , а в регистре синдрома — остаток от деления вектора ошибки на порождающий полином.

Вычисленный синдром сравнивается со всеми табличными синдромами, и в случае совпадения с одним из них старший разряд буферного регистра исправляется путем прибавления к его значению единицы.

После этого синдромный и буферный регистры один раз циклически сдвигаются. Это реализует умножение  $S(x)$  на  $x$  и деление на  $g(x)$ . Содержимое ячеек синдромного регистра теперь равно остатку от деления  $x \cdot S(x)$  на  $g(x)$  или синдрому ошибки  $x \cdot e(x) \bmod (X^n + 1)$ .

Если новый синдром совпадает с одним из табличных, то исправляется очередная ошибка и т.д. Через  $n$  тактов все позиции будут, таким образом, исправлены.

Рассмотрим работу декодера Меггитта для циклического (7,4)-кода, исправляющего одиночную ошибку. Схема декодера изображена на рис. 1.16.

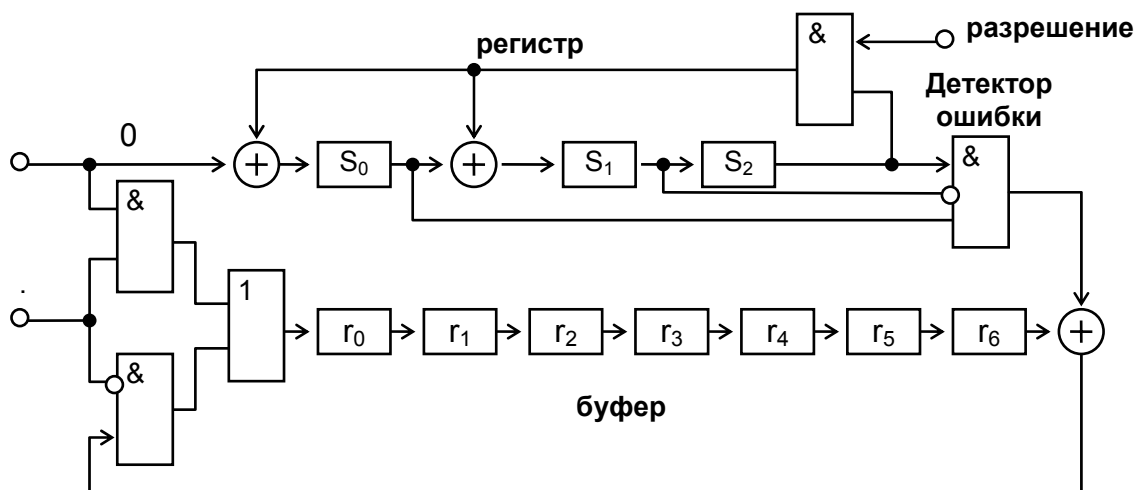


Рис. 1.16

Принцип работы декодера заключается в том, что независимо от того, в какой позиции произошла ошибка, осуществляется ее сдвиг в последнюю ячейку буферного регистра с соответствующим пересчетом синдрома и ее исправление в этой позиции.

Полином ошибки в старшем разряде для (7,4)-кода Хемминга имеет вид  $e_6(x) = x^6$ , соответствующий ему синдромный полином  $S_6(x) = 1 + x^3$  ( $S = (101)$ ), детектор ошибки настроен на это значение синдрома.

Пусть, например, передана последовательность  $U = (1001011)$ , ей соответствует кодовый полином  $U(x) = 1 + x^3 + x^5 + x^6$ . Произошла ошибка в третьей позиции  $e(x) = x^3$ , принятый вектор имеет вид  $r = (1000011)$ , его полином  $r(x) = 1 + x^5 + x^6$ .

Когда принятая последовательность полностью входит в буферный регистр, в регистре синдрома оказывается синдром, соответствующий ошибке  $e(x) = x^3 S_3 = (110)$ . Поскольку он не совпадает с табличным для  $e_6$ , на выходе детектора ошибок будет  $0$  и исправления не происходит.

Далее производится однократный циклический сдвиг принятой последовательности в буферном регистре, однократное деление синдрома  $x \cdot S_3$  на порождающий полином в регистре с обратными связями и проверка на совпадение синдрома с заданным.

Последовательность состояний регистров декодера в процессе декодирования показана на рис. 1.17.

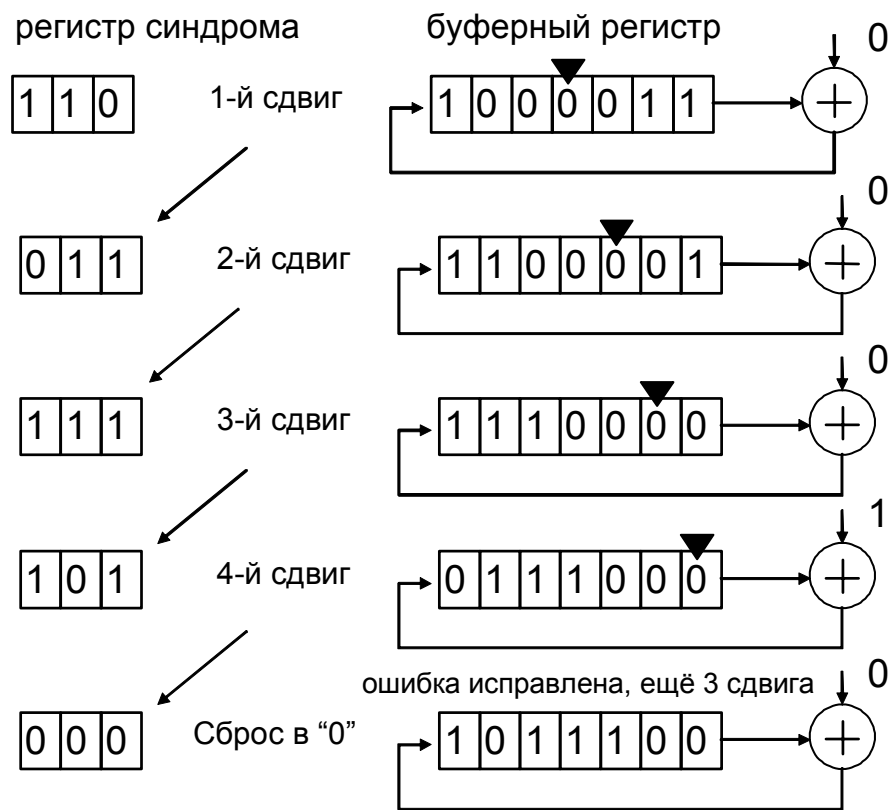


Рис. 1.17

Таким образом, исправление ошибки в декодере Меггитта осуществляется за  $2 \cdot n$  тактов: в течение  $n$  тактов производится ввод принятой последовательности в буферный регистр, в течение  $l$  тактов — исправление ошибки, и еще в течение  $n - l$  — восстановление буферного регистра в исходное состояние с исправленным словом. Простота декодера достигается увеличением времени декодирования.

Следует отметить, что в связи с успехами в разработке БИС и устройств памяти в значительной степени снимается вопрос о размерах таблиц, связывающих значения синдрома и вектора ошибки (для синдромных декодеров) и даже значения кодовых слов и принятых последовательностей

(для декодера максимального правдоподобия). Поэтому в перспективе возможно снижение интереса к кодам, обладающим специальной структурой, и к методам их декодирования.

## 2. Сверточные коды

Методы кодирования и декодирования, рассмотренные ранее, относились к блочным кодам. При использовании таких кодов информационная последовательность разбивается на отдельные блоки, которые кодируются независимо друг от друга. Таким образом, закодированная последовательность становится последовательностью независимых слов одинаковой длины.

При использовании сверточных кодов поток данных разбивается на гораздо меньшие блоки длиной  $k$  символов (в частном случае  $k_0 = 1$ ), которые называются *кадрами информационных символов*.

Кадры информационных символов кодируются *кадрами кодовых символов* длиной  $n_0$  символов. При этом кодирование кадра информационных символов в кадр кодового слова производится с учетом *предшествующих  $m$  кадров информационных символов*. Процедура кодирования, таким образом, связывает между собой последовательные кадры кодовых слов. Передаваемая последовательность становится одним полубесконечным кодовым словом.

Развитие теории и практики сверточных кодов заметно отличается от развития блочных кодов. При построении блочных кодов и методов их декодирования широко использовались алгебраические методы. В случае сверточных кодов это не так. Большинство хороших сверточных кодов было найдено путем просмотра с помощью ЭВМ большого числа кодов и последующего выбора кодов с хорошими свойствами. Декодирование сверточных кодов производится методами, близкими к методам максимального правдоподобия, причем в этом случае они реализуются достаточно просто.

### 2.1. Кодирование с использованием сверточных кодов

Основными характеристиками сверточных кодов являются величины:

- $k_0$  – размер кадра информационных символов;
- $n_0$  – размер кадра кодовых символов;
- $m$  – длина памяти кода;
- $k = (m+1) \cdot k_0$  - информационная длина слова;
- $n = (m+1) \cdot n_0$  - кодовая длина блока.

Кодовая длина блока - это *длина кодовой последовательности, на которой сохраняется влияние одного кадра информационных символов.*

Наконец, сверточный код имеет еще один важный параметр - скорость  $R = k/n$ , которая характеризует степень избыточности кода, вводимой для обеспечения исправляющих свойств кода.

Как и блочные, сверточные коды могут быть систематическими и несистематическими и обозначаются как линейные сверточные  $(n,k)$ -коды.

Систематическим сверточным кодом является такой код, для которого в выходной последовательности кодовых символов содержится без изменения породившая его последовательность информационных символов. В противном случае сверточный код является несистематическим.

Примеры схем кодеров для систематического  $(8,4)$  и несистематического сверточных  $(6,3)$ -кодов приведены на рис. 2.1 и 2.2.

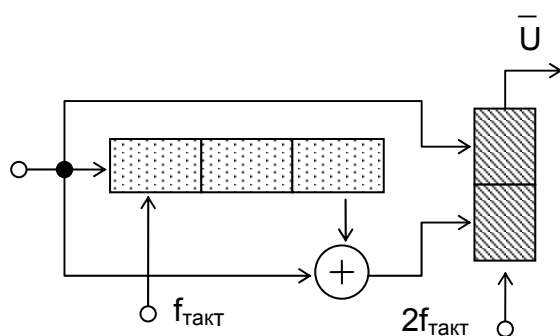


Рис. 2.1

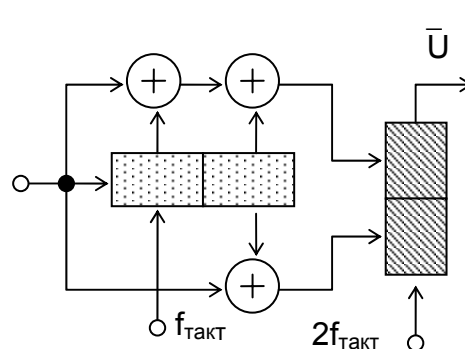


Рис. 2.2

Возможны различные способы описания сверточных кодов, например, с помощью порождающей матрицы. Правда, в силу бесконечности кодируемой последовательности и порождающая матрица будет иметь бесконечные размеры. Точнее, она будет состоять из бесконечного числа матриц  $\underline{G}$  для обычного блочного кода, расположенных вдоль главной диагонали полубесконечной матрицы. Вся остальная ее часть заполняется нулями.

Однако более удобным способом описания сверточного кода является его задание с помощью *импульсной переходной характеристики эквивалентного фильтра* или соответствующего ей *порождающего полинома.*

Импульсная переходная характеристика фильтра (*ИПХ*) (а кодер сверточного кода, по сути дела, является фильтром) есть реакция на единичное воздействие вида  $\delta = (10000....$ . Для кодеров, изображенных на рис. 2.1 и 2.2, соответствующие импульсные характеристики будут иметь вид:

$$H_a = (11.00.00.01.00.00\dots, \quad (2.1)$$

$$H_b = (11.10.11.00.00.00\dots \quad (2.2)$$

Еще одна форма задания сверточных кодов – это использование порождающих полиномов, однозначно связанных с *ИПХ* эквивалентного фильтра:

$$H_a(x) = 1 + x + x^7, \quad (2.3)$$

$$H_b(x) = 1 + x + x^2 + x^4 + x^5. \quad (2.4)$$

При этом кодовая последовательность  $U$  на выходе сверточного кодера получается в результате свертки входной информационной последовательности  $m$  с импульсной переходной характеристикой  $H$ .

Рассмотрим примеры кодирования последовательностей с использованием импульсной характеристики эквивалентного фильтра.

Пусть  $m = (110 \dots$ , тогда для кодера с *ИПХ*  $H = (11.00.00.01.00.00\dots$

$$\begin{array}{r} U = 11.00.00.01.00.00\dots \\ \quad 11.00.00.01.00\dots \\ \hline \end{array}$$

$$U = 11.11.00.01.01.00.00\dots,$$

или  $m = (1011000\dots$

$$\begin{array}{r} U = 11.00.00.01.00.00.00\dots \\ \quad 11.00.00.01.00\dots \\ \quad 11.00.00.01\dots \\ \hline \end{array}$$

$$U = 11.00.11.10.00.01.01.00\dots$$

Иногда удобнее рассматривать полный порождающий полином сверточного кода  $G(x)$  как совокупность нескольких многочленов меньших степеней, соответствующих ячейкам выходного регистра кодера. Так, для кодера рис. 2.2 соответствующие частичные порождающие полиномы будут следующими:

$$G_1(x) = 1 + x + x^2, \quad (2.5)$$

$$G_2(x) = 1 + x^2. \quad (2.6)$$

Пусть, например, кодируется последовательность  $m = (1010\dots$ . Соответствующий информационный полином запишется как  $m(x) = 1 + x^2$ . Тогда на входе первой ячейки выходного регистра при кодировании будет последовательность  $U_1 = (11011000\dots$ , которой соответствует полином  $U_1(x) = 1 + x + x^3 + x^4$ , а на входе второй ячейки —  $U_2 = (10001000\dots$  и, соответственно, полином  $U_2(x) = 1 + x^4$ .

Легко заметить, что при этом справедливы равенства

$$U_1(x) = m(x) \cdot G_1(x), \quad (2.7)$$

$$U_2(x) = m(x) \cdot G_2(x). \quad (2.8)$$

Такая форма записи удобна, поскольку видна структура кодирующего устройства (по набору полиномов можно сразу синтезировать устройство). На практике полиномы задаются набором своих коэффициентов. В табл. 2.1 приведены примеры кодеров сверточных кодов, заданных своими коэффициентами при степенях порождающих полиномов.

Таблица 2.1

$G_1$	$G_2$
<b>111</b>	<b>101</b>
<b>1111</b>	<b>1101</b>
<b>11101</b>	<b>10011</b>
<b>111011</b>	<b>10100111</b>
<b>110001</b>	<b>11111001</b>

В качестве примера кодера с  $k_0 \neq 1$  можно привести кодер сверточного (12,9) кода Вайнера-Эша с параметрами:  $k_0 = 3$ ,  $n_0 = 4$ ,  $k = 9$ ,  $n = 12$ ,  $R = 3/4$  (рис. 2.3).

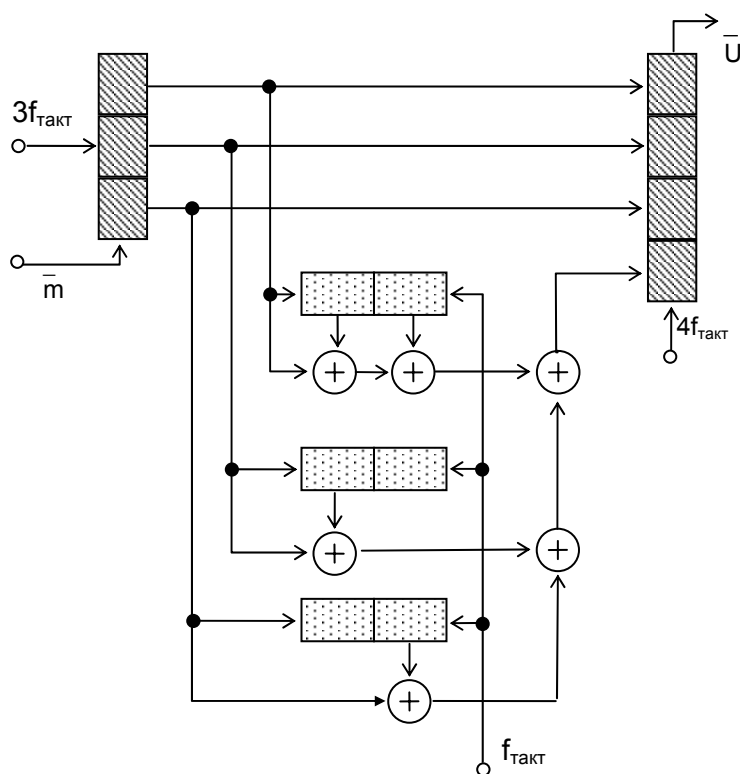


Рис. 2.3

Пусть, например,  $m = (100.000.000....$  Тогда кодовое слово  $U = (1001.0001.0001.0000....)$ . Видно, что это – систематический код, в котором к трем информационным символам добавляется один проверочный, зависящий от значений информационных символов не только текущего кадра, но и двух предшествующих кадров. При этом влияние одного кадра информационных символов распространяется на 12 символов кодовой последовательности, то есть кодовая длина блока для этого кода  $n = 12$ .

## 2.2. Синдромное декодирование сверточных кодов

Предположим, что нами принята полубесконечная последовательность  $r$ , состоящая из слова сверточного кода и вектора ошибки:

$$r = U + e. \quad (2.9)$$

Аналогично тому, как это делается для блочных кодов, можно вычислить синдром принятой последовательности:

$$S = r \cdot \underline{H} = e \cdot \underline{H}. \quad (2.10)$$

Однако из-за бесконечной длины принятой последовательности (а сверточный код представляет собой непрерывную бесконечную последовательность двоичных символов) синдром также будет иметь бесконечную длину и его прямое вычисление не имеет смысла.

Вместе с тем можно заметить, что для рассмотренных нами сверточных кодов влияние одного информационного кадра распространяется всего на несколько кодовых кадров. Поэтому декодер может *просматривать не весь синдром, а вычислять его компоненты по мере поступления кадров кодовой последовательности, исправлять текущие ошибки и сбрасывать те компоненты синдрома, которые вычислены давно.*

Для исправления ошибок при этом декодер должен содержать таблицу сегментов синдромов и сегментов конфигураций ошибок, образующих данные конфигурации синдрома. Если декодер находит в таблице полученный сегмент синдрома, он исправляет начальный сегмент кодового слова.

Схема декодера для сверточного  $(12,9)$ -кода Вайнера-Эша изображена ниже (рис. 2.4). Исправление ошибок с помощью данного декодера производится на сегментах из трех кодовых кадров -  $n = 12$ .

Декодер работает следующим образом. Во входной регистр записывается первый кадр принимаемой последовательности  $r$  (четыре символа). По первым трем (информационным) символам кадра по тем же правилам, что и при кодировании, определяется значение контрольного бита, который далее сравнивается с четвертым (проверочным) символом принятого кадра.

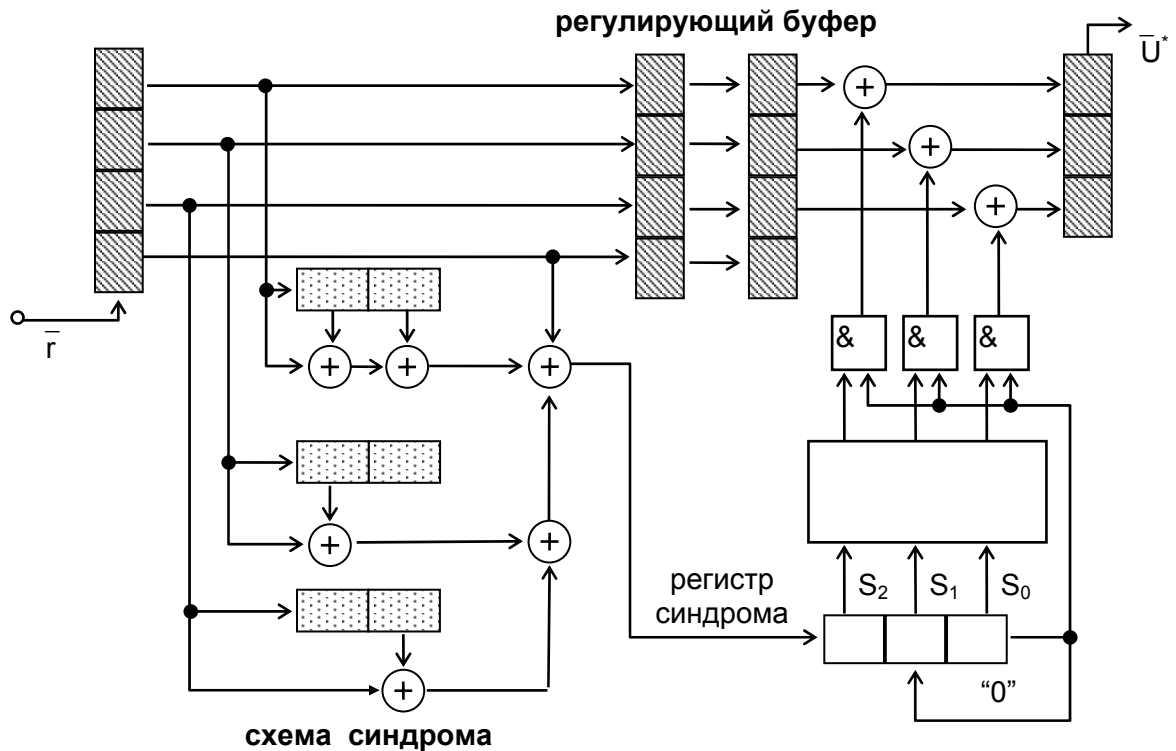


Рис. 2.4

При совпадении контрольного и проверочного битов (а это будет, если ошибки в первом кадре нет) в первую ячейку синдромного регистра записывается  $0$ , если же в кадре ошибка есть, — то  $1$ . Далее первый кадр принятой последовательности переносится в регулирующий буфер, а во входной регистр заносится очередной кадр принятой последовательности.

После аналогичных проверок для второго и третьего кадров принятой последовательности на выходе регулирующего буфера оказывается первый кадр принятой последовательности, в регистре синдрома — трехбитовый синдром, соответствующий принятому сегменту из трех кадров, а на выходе адресной логики — дешифрованный по синдрому  $S$  вектор ошибки  $e$ .

Исправленный кадр записывается в выходной регистр, при этом исправление производится только при наличии единицы на входах схем “И”, что соответствует присутствию ошибки именно в первом кадре. После исправления ошибки регистр синдрома сбрасывается. Потеря второго и третьего битов синдрома при этом не имеет значения, так как ни во втором, ни в третьем кадрах ошибки быть не должно (она была в первом кадре, а значит, во фрагменте из трех кадров ошибки больше быть не должно).

Все возможные конфигурации ошибок в первых трех кадрах и соответствующие им первые три бита синдрома приведены в табл. 2.2.



На диаграмме рис. 2.6 изображены входные и выходные последовательности кодера: *входные* — направлением движения по дереву (вверх - 0, вниз - 1), *выходные* — символами вдоль ребер дерева. При этом кодирование состоит в движении вправо по кодовому дереву.

Например, входная последовательность  $m = (01000\dots)$  кодируется как  $U = (0011101100000\dots)$ , последовательность  $m = (1010100000\dots)$  - как  $U = (1110001000\dots)$ .

Если внимательно посмотреть на строение приведенного на рис. 2.6 кодового дерева, можно заметить, что начиная с четвертого ребра его структура повторяется, т.е. каким бы ни был первый шаг, начиная с четвертого ребра значения выходных символов кодера повторяются. Одинаковые же узлы могут быть объединены, и тогда начиная с некоторого сечения размер кодового дерева перестанет увеличиваться.

Другими словами, *выходная кодовая последовательность в определенный момент перестает зависеть от значений входных символов, введенных в кодер ранее.*

Действительно, из рис. 2.6 видно, что, когда третий входной символ вводится в кодер, первый входной символ покидает сдвиговый регистр и не сможет в дальнейшем оказывать влияния на выходные символы кодера.

С учетом этого неограниченное кодовое дерево, изображенное на рис. 2.6, переходит в ограниченную решетчатую диаграмму (кодовое дерево со сливающимися узлами) рис. 2.7.

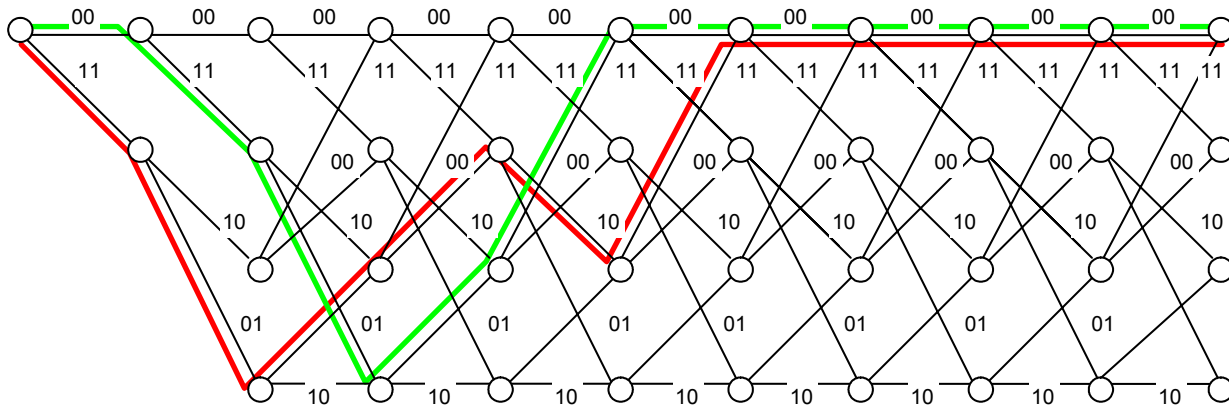


Рис. 2.7

Кодирование сверточными кодами с использованием решетчатой диаграммы, как и в случае с кодовым деревом, представляет собой чрезвычайно простую процедуру: *очередные символы входной последовательности определяют направление движения из узлов решетки: если 0, то идем по верхнему ребру, если 1 - по нижнему ребру.* Однако в отличие от кодового дерева решетчатая диаграмма не разрастается по ширине с каждым шагом, а имеет начиная с третьего сечения постоянную ширину.

В качестве примера закодируем с помощью решетчатой диаграммы несколько информационных последовательностей.

Пусть  $m=(011000\dots)$ . Соответствующая ей кодовая последовательность будет иметь вид:

$$U = (001101011100\dots,$$

или  $m = (110100\dots)$ , тогда

$$U = (1101010010110000\dots;$$

или  $m = (000000\dots)$ , тогда

$$U = (1101010010110000$$

и т.д., проще не придумаешь.

## 2.4. Декодирование сверточных кодов. Алгоритм Витерби

Наилучшей схемой декодирования корректирующих кодов, как уже отмечалось, является декодирование методом максимального правдоподобия, когда декодер определяет набор условных вероятностей  $P(r/U_i)$ , соответствующих всем возможным кодовым векторам  $U_i$ , и решение принимает в пользу кодового слова, соответствующего максимальному  $P(r/U_i)$ .

Для двоичного симметричного канала без памяти (канала, в котором вероятности передачи  $0$  и  $1$ , а также вероятности ошибок вида  $0 \rightarrow 1$  и  $1 \rightarrow 0$  одинаковы, ошибки в  $j$ -м и  $i$ -м символах кода независимы) декодер максимального правдоподобия сводится к декодеру минимального хеммингова расстояния. Последний вычисляет расстояние Хемминга между принятой последовательностью  $r$  и всеми возможными кодовыми векторами  $U_i$  и выносит решение в пользу того вектора, который оказывается ближе к принятому.

Естественно, что в общем случае такой декодер оказывается очень сложным и при больших размерах кодов  $n$  и  $k$  практически нереализуемым.

Характерная структура сверточных кодов (повторяемость структуры за пределами окна длиной  $n$ ) позволяет создать вполне приемлемый по сложности декодер максимального правдоподобия.

Впервые идея такого декодера была предложена Витерби. Работает он следующим образом.

Предположим, на вход декодера поступил сегмент последовательности  $r$  длиной  $b$ , превышающей кодовую длину блока  $n$ . Назовем  $b$  окном декодирования. Сравним все кодовые слова данного кода (в пределах сегмента длиной  $b$ ) с принятым словом и выберем кодовое слово, ближайшее

к принятому. Первый информационный кадр выбранного кодового слова примем в качестве оценки информационного кадра декодированного слова.

После этого в декодер вводится  $n_0$  новых символов, а введенные ранее самые старые  $n_0$  символов сбрасываются, и процесс повторяется для определения следующего информационного кадра.

Таким образом, декодер Витерби последовательно обрабатывает кадр за кадром, двигаясь по решетке, аналогичной используемой кодером. В каждый момент времени декодер не знает, в каком узле находится кодер, и не пытается его декодировать. Вместо этого декодер по принятой последовательности определяет наиболее правдоподобный путь к каждому узлу и определяет расстояние между каждым таким путем и принятой последовательностью. Это расстояние называется мерой расходимости пути. В качестве оценки принятой последовательности выбирается сегмент, имеющий наименьшую меру расходимости. Путь с наименьшей мерой расходимости называется выжившим путем.

Рассмотрим работу декодера Витерби на простом примере. Полагаем, что кодирование производится с использованием сверточного (6,3)-кода (схема кодера приведена на рис. 2.5, решетчатая диаграмма, соответствующая этому кодеру, – на рис. 2.7).

Пользуясь решетчатой диаграммой кодера, попытаемся, приняв некоторый сегмент  $r$ , проследить наиболее вероятный путь кодера. При этом для каждого сечения решетчатой диаграммы будем отмечать меру расходимости пути к каждому ее узлу.

Предположим, что передана кодовая последовательность  $U = (0000000...)$ , а принятая последовательность имеет вид  $r = (1000100000...)$ , то есть в первом и в третьем кадрах кодового слова возникли ошибки. Как мы уже убедились, процедура и результат декодирования не зависят от передаваемого кодового слова и определяются только ошибкой, содержащейся в принятой последовательности. Поэтому проще всего считать, что передана нулевая последовательность, то есть  $U = (0000000...)$ .

Приняв первую пару символов (10), определим меру расходимости для первого сечения решетки (см. рис. 2.7), приняв следующую пару символов (00), — для второго сечения и т.д. При этом из входящих в каждый узел путей оставляем путь с меньшей расходимостью, поскольку путь с большей на данный момент расходимостью уже не сможет стать в дальнейшем короче.

Заметим, что для рассматриваемого примера начиная с четвертого уровня метрика (или мера расходимости) нулевого пути меньше любой другой метрики. Поскольку ошибок в канале больше не было, ясно, что в конце концов в качестве ответа будет выбран именно этот путь. Из этого примера также видно, что выжившие пути могут достаточно долго отличаться друг от друга.

Однако на шестом - седьмом уровне первые семь ребер всех выживших путей совпадут друг с другом. В этот момент согласно алгоритму Витерби и принимается решение о переданных символах, так как все выжившие пути выходят из одной вершины, т.е. соответствуют одному информационному символу.

Процедура декодирования последовательности с двумя ошибками иллюстрируется последовательностью, представленной на рис. 2.8.

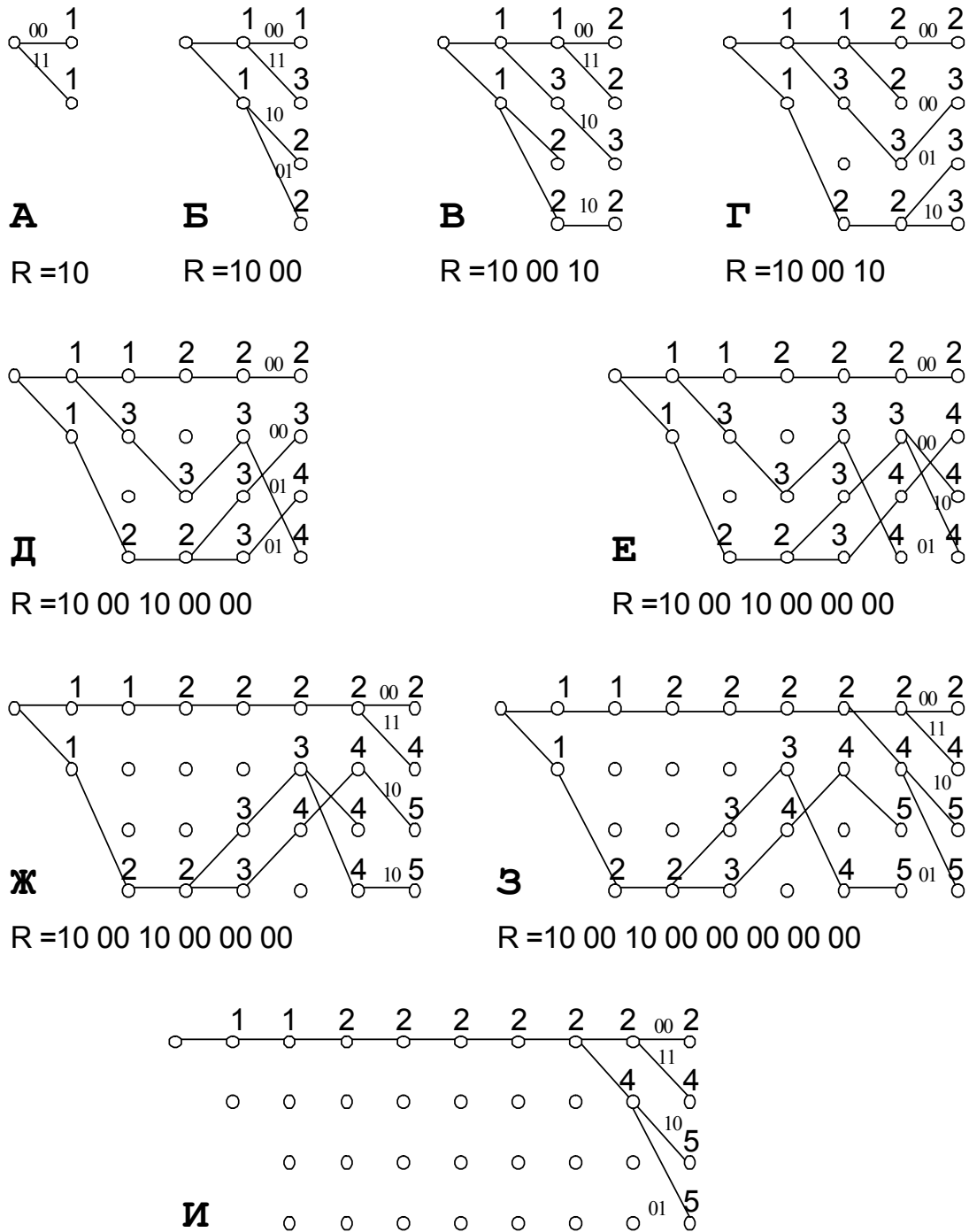


Рис. 2.8

Глубина, на которой происходит слияние выживших путей, не может быть вычислена заранее; она является случайной величиной, зависящей от кратности и вероятности возникающих в канале ошибок. Поэтому на практике обычно не ждут слияния путей, а устанавливают фиксированную глубину декодирования.

Из рис. 2.8 видно, что уже на уровне  $E$  степень различия метрик правильного и неправильного путей достаточно велика ( $d_{np} = 2$ ,  $d_{om} = 4$ ), то есть в данном случае можно было бы ограничить глубину декодирования величиной  $b \leq 6$ . Но иногда более длинный к данному сечению путь может оказаться в конечном итоге самым коротким, поэтому особенно увлекаться уменьшением размера окна декодирования  $b$  с целью упрощения работы декодера не стоит.

На практике глубину декодирования обычно выбирают в диапазоне  $n < b \leq n + l$ , где  $l$  - число исправляемых данным кодом ошибок.

Из рис. 2.8 видно также, что, несмотря на наличие в принятом фрагменте двух ошибок, его декодирование произошло без ошибки и в качестве ответа будет принята переданная нулевая последовательность.

## 2.5. Алгоритмы поиска по решетке

Характеристики сверточных кодов, как и любых других кодов, улучшаются по мере увеличения их размера, в данном случае - кодовой длины блока  $n$ . При этом, однако, декодер Витерби становится нереализуемо сложным. Так, при  $n = 10$  он должен помнить уже не менее  $2^{10} = 1024$  выживших путей.

Для уменьшения сложности декодера максимального правдоподобия при больших  $n$  была разработана стратегия, *игнорирующая маловероятные пути поиска по решетке, как только они становятся маловероятными. Однако решение о том, чтобы окончательно отбросить данный путь, не принимается. Время от времени декодер возвращается назад и продолжает оставленный путь.*

Подобные стратегии поиска наиболее вероятного пути по решетке известны под общим названием *последовательного декодирования*.

В отличие от оптимальной процедуры Витерби последовательный декодер, просмотрев первый кадр, переходит в очередной узел решетки с *наименьшей на данный момент расходимостью*. Из этого узла он анализирует следующий кадр, выбирая ребро, ближайшее к данному кадру, и переходит в следующий узел и так далее.

При отсутствии ошибок эта процедура работает очень хорошо, однако при возникновении ошибок на каком-либо шаге декодер может случайно выбрать неправильную ветвь. Если он продолжит следовать по неправильному пути, то очень скоро обнаружит, что происходит слишком много ошибок, и расходимость пути начнет быстро нарастать. Но это будут

ошибки декодера, а не канала. Поэтому декодер возвращается на несколько кадров назад и начинает исследовать другие пути, пока не найдет наиболее правдоподобный. Затем он будет следовать вдоль этого нового пути.

Наиболее простым последовательным алгоритмом декодирования является *алгоритм Фано*. Для его реализации необходимо знать среднюю вероятность появления ошибок в канале связи  $P_{ош}$ .

Пока декодер следует по правильному пути, вероятное число ошибок в первых  $l$  кадрах (это будет мерой расходимости пути за  $l$  кадров) примерно равно

$$d_l = P_{ош} \cdot n_0 \cdot l. \quad (2.11)$$

Если же ошибок становится намного больше, то декодер принимает решение, что он на ложном пути, и возвращается на несколько кадров назад.

Процедура декодирования последовательности  $r = (01000000\dots$ , содержащей ошибку во втором символе, показана на рис. 2.9.

Пусть  $U = (0000000000000000\dots$  и принята последовательность  $r = (010000000000\dots$ , то есть возникла ошибка во второй позиции кода. Проследим путь декодера Фано по решетке рис. 2.9.

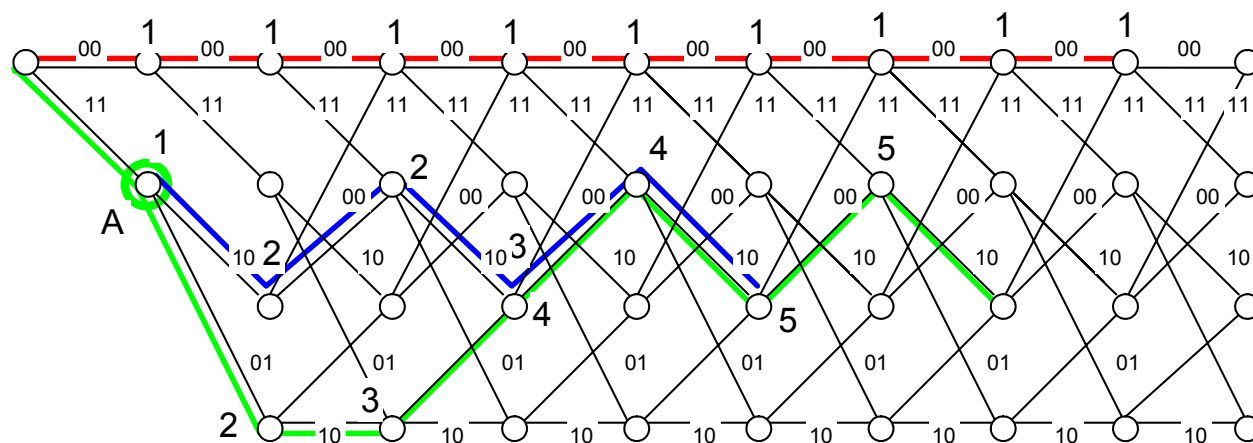


Рис. 2.9

Видно, что если на первом шаге выбран верный путь, то расходимость не увеличивается и декодирование выполняется без ошибки. Однако на первом шаге мог быть выбран и неправильный путь (вниз), имеющий такую же метрику. При этом расходимость начинает быстро нарастать, что свидетельствует о неправильном выборе пути. Необходимо вернуться, но на сколько шагов? Предположим, что возвращение произошло в точку  $A$  и из нее осуществляется поиск по неисследованному пути. Очень быстро, однако, декодер убедится в неправильности выбора и снова должен вернуться. Но теперь уже не имеет смысла возвращаться в точку  $A$ , нужно сделать по крайней мере еще один шаг назад и снова пойти по неисследованному пути. В данном случае это будет правильный путь.

Декодер Фано гораздо проще в реализации, нежели декодер Витерби, и почти не уступает ему в вероятности ошибочного декодирования. Но он обладает и серьезным недостатком. Если вероятность ошибок в канале велика, то возрастает и вероятность принятия неправильных решений при переходе из одного узла решетки в другой, а следовательно, возрастает число возвратов к предыдущим узлам, что требует затрат времени. Входная последовательность при этом во избежание потери данных должна сохраняться в специальном накапливающем регистре. В конечном итоге могут произойти переполнение этого регистра и отказ декодера.

### 3. Применение корректирующего кодирования в системах связи

Приведенные ниже несколько типичных примеров применения помогут понять, как эффективнее на практике осуществлять кодирование в системах связи. Один из типичных случаев характерен для систем, в которых требуются очень мощные коды. Обычный метод реализации таких кодов состоит в каскадировании двух или более простых кодов.

Еще одной интересной проблемой является кодирование для каналов, в которых ошибки возникают не независимо, а пакетами. Рассмотренные ранее методы кодирования при этом становятся совершенно неэффективными, поскольку все их характеристики определялись исходя из того, что ошибки в отдельных символах независимы друг от друга.

#### 3.1. Каскадные коды

Каскадные коды были впервые предложены Фorni в качестве метода практической реализации кода с большой длиной блока и высокой корректирующей способностью. Эта цель достигается введением нескольких уровней кодирования, обычно – двух.

Основную идею каскадного кодирования с двумя уровнями иллюстрирует рис. 3.1.

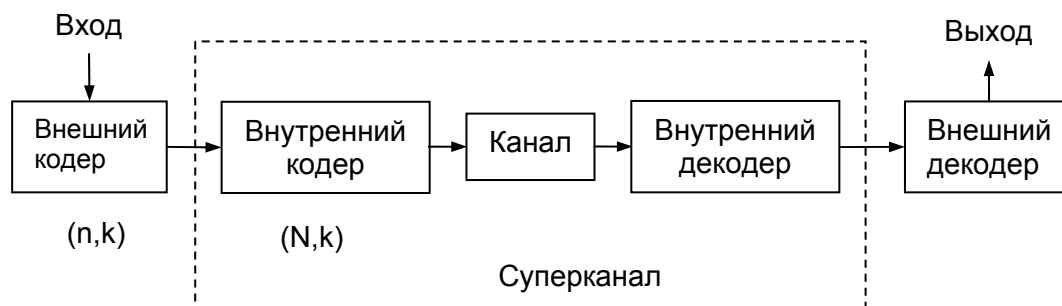


Рис. 3.1

В этой схеме комбинацию внутреннего кодера, канала и внутреннего декодера иногда называют *суперканалом*, аналогично, комбинацию

внешнего и внутреннего кодеров - *суперкодером*, а комбинацию внутреннего и внешнего декодеров – *супердекодером*.

Длина каскадного кода получается равной  $N1 = N \cdot n$  двоичных символов, где  $N$  - длина внешнего кода, а  $n$  - длина внутреннего кода. При этом информационная длина кода составляет  $K1 = K \cdot k$  двоичных символов, а скорость кода  $R1 = R \cdot r$ . Несмотря на то, что общая длина кода получается большой и, соответственно, значительно возрастает его исправляющая способность, его декодирование может выполняться с помощью двух декодеров, рассчитанных на длины составляющих его кодов  $n$  и  $N$ . Это позволяет многократно снизить сложность декодера в сравнении с тем, если бы такая исправляющая способность достигалась одноуровневым кодированием.

Простейшей иллюстрацией к каскадному кодированию является итеративный код, рассмотренный в параграфе 1.2.2. Этот код состоит из простых кодов с проверкой на четность (по строкам и столбцам), но в то же время обладает исправляющей способностью. На практике, конечно, используются гораздо более сложные коды.

Обычно внешнее кодирование выполняется *блочными кодами*, а внутреннее – более приспособленными для побитовой передачи по радиоканалу *сверточными кодами*. Каскадное кодирование широко применяется на практике, в частности, при помехоустойчивом кодировании речевой информации в системе сотовой связи формата **GSM**.

### 3.2. Кодирование с перемежением

Все рассмотренные ранее методы кодирования и примеры расчета их эффективности относились к каналам без памяти, то есть к каналам, в которых вероятность ошибки постоянна и не зависит от времени. На практике же свойства каналов связи таковы, что ошибки обычно группируются так называемыми пакетами. При постоянной некоторой средней вероятности ошибок на большом интервале времени значение  $P_{ош}$  на отдельных коротких интервалах может значительно превышать среднее значение -  $P_{ош\ ср}$ . Если для исправления таких ошибок использовать традиционные методы кодирования-декодирования, это потребует применения сложных кодов с большой исправляющей способностью и, соответственно, большой избыточностью.

Одно из возможных решений в таких случаях может состоять в использовании достаточно простого кода, рассчитанного на исправление одиночных ошибок, вместе с парой устройств, выполняющих перемежение закодированных символов перед их передачей в канал и восстановление (деперемежение) после приема. При такой обработке кодовой и принятой последовательностей ошибки на входе декодера распределяются более равномерно.

Структурная схема системы с перемежением показана на рис. 3.2.

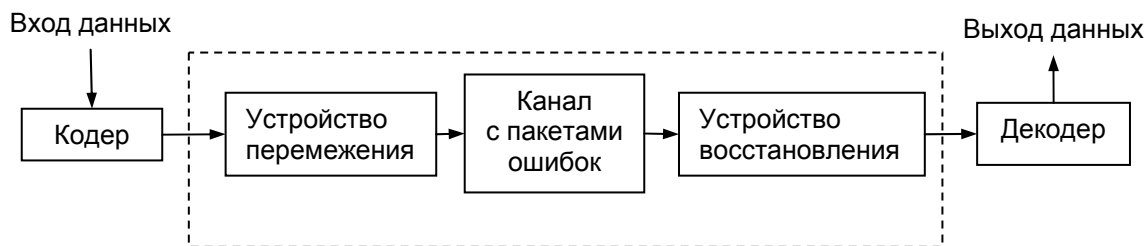


Рис. 3.2

Устройство *перемежения* в этой схеме переупорядочивает (переставляет) символы передаваемой последовательности некоторым детерминированным образом. С помощью устройства *восстановления* производится обратная перестановка, восстанавливающая исходный порядок следования символов. Используются различные способы перемежения-восстановления. Первый способ – *периодическое перемежение*. Он проще, но при изменении характера помех может оказаться неустойчивым. Более сложное – *псевдослучайное перемежение*, которое обладает при нестационарных ошибках гораздо большей устойчивостью.

### *Периодическое перемежение*

При периодическом перемежении функция перестановок периодична с некоторым периодом. Перемежение может быть блочным, когда перестановки выполняются над блоком данных фиксированного размера, или сверточным, когда процедура выполняется над непрерывной последовательностью.

Типичное блочное устройство перемежения работает следующим образом. Кодовые символы записываются в матрицу, имеющую  $N$  строк и  $M$  столбцов построчно, а читаются из нее по столбцам. На приемной стороне операция выполняется в обратном порядке: запись производится по столбцам, а чтение – по строкам. При этом происходит восстановление исходного порядка следования символов. Естественно, что процедуры перемежения и деперемежения должны быть засинхронизированы.

При таком перемежении достигается следующее: любой пакет ошибок длиной  $m \leq M$  переходит на выходе устройства восстановления в одиночные ошибки, каждая пара которых разделена не менее чем  $N$  символами. Правда, при этом любая периодическая с периодом  $M$  *одиночная ошибка превращается в пакет*, но вероятность такого преобразования очень мала, хотя и существует. В этом, собственно, и состоит главный недостаток периодического перемежения: если появилась помеха с частотой следования ошибок, совпадающей с периодом перемежения или кратной ему, то до тех пор, пока характеристики помехи не изменятся, из одиночных ошибок будут возникать неисправляемые пакеты ошибок.

## *Псевдослучайное перемежение*

При псевдослучайном перемежении блоки из  $L$  символов записываются в память с произвольной выборкой (ЗУПВ), а затем считываются из нее псевдослучайным образом. Порядок перестановок, одинаковый для устройств перемежения и восстановления, можно записать в ПЗУ и использовать его для адресации ЗУПВ.

Как и для периодического перемежения, существует вероятность того, что ошибки будут следовать таким образом (синхронно с перемежением), что одиночные ошибки будут группироваться в пакеты. Но такая вероятность чрезвычайно мала (если, конечно, это не организованная помеха и противник не знает порядка перемежения). Случайное же совпадение порядка следования перестановок при перемежении и импульсов помехи при достаточной длине  $L$  практически невероятно.

Что касается характера псевдослучайного перемежения, то для этого могут использоваться любые псевдослучайные последовательности - линейные и нелинейные последовательности максимальной длины, последовательности, основанные на линейном сравнении, а также любые алгоритмы формирования псевдослучайных чисел с необходимым периодом повторения.

На этом краткий экскурс в теорию помехоустойчивого кодирования завершён, более детально в различных аспектах практического применения корректирующего кодирования для повышения помехоустойчивости систем связи можно разобраться с использованием литературных источников [4,5].

Как мы уже отмечали, помехоустойчивое кодирование, вообще-то, не является обязательной операцией при передаче информации. Эта процедура (и соответствующий ей элемент структурной схемы *РТС ПИ*) может отсутствовать. Однако это может привести к очень существенным потерям в помехоустойчивости системы, значительному уменьшению скорости передачи и снижению качества передачи информации. Поэтому практически все современные системы (за исключением, быть может, самых простых) должны включать и обязательно включают помехоустойчивое кодирование данных.

#### 4. Задачи и практические вопросы к курсу

1. По каналу связи с нормальным белым шумом передается информация со скоростью  $V_{nu} = 10 \text{ кбит/с}$ . При этом средняя вероятность ошибок в канале составляет  $P_{ош} = 10^{-6}$ .

Для улучшения качества передачи информации рассматривается несколько вариантов решения:

- кодирование корректирующим кодом с порождающей матрицей  $G = | 111 |$ ;
- кодирование сверточным (6,3)-кодом;
- кодирование (8,4)-кодом Хемминга;
- уменьшение на 20% скорости передачи .

Какое из предложенных решений обеспечит больший эффект, если в первых трех случаях скорость передачи информации должна сохраниться прежней?

2. Порождающая матрица линейного блочного кода имеет вид

$$G = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}.$$

Определить для данного кода  $H$ ,  $d_{min}$ . Изобразить схему кодера и декодера.

Определить число ошибок, не исправленных данным кодом за 1 час работы, если  $V_{nu} = 10 \text{ кбит/с}$ ,  $P_{ош} = 10^{-4}$ .

3. В канале связи с шумами производится кодирование информации с использованием кода с порождающей матрицей вида  $G = | 11111 |$ .

На входе приемного устройства на интервале времени, соответствующем длине кодовой последовательности, присутствует колебание вида  $U(t)$  (рис. 4.1)

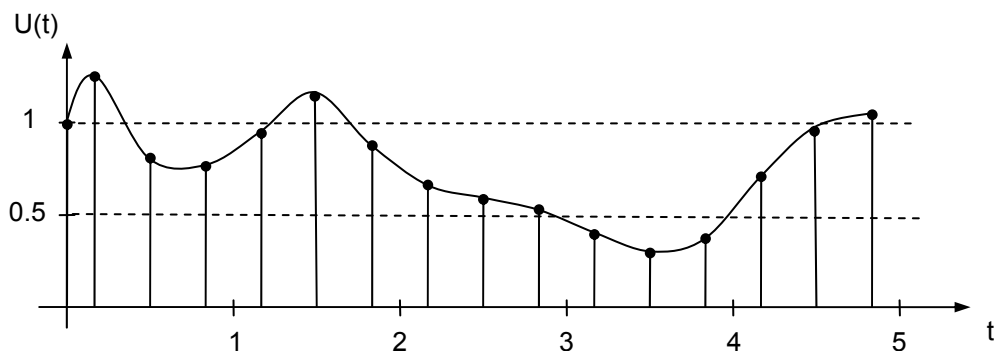


Рис. 4.1

Сигнал  $U(t)$  в приемном устройстве подвергается дискретизации (по 3 отсчета на интервал, соответствующий одному символу кодовой последовательности).

Какое решение относительно  $m$  примет:

- жесткий мажоритарный декодер;
- мягкий декодер максимального правдоподобия?

4. Для кодера сверточного кода со схемой, показанной на рис. 2.5, определить  $k_0, n_0, m, n, k, R, b$ , изобразить кодовое дерево и решетчатую диаграмму, закодировать последовательность  $m = (1100100000.....)$ , декодировать  $r$  с ошибкой во втором кадре с использованием алгоритма Витерби.

5. Предложить вариант схем кодера и декодера сверточного (9,6)-кода Вайнера-Эша (по аналогии с (12,9)-кодом), исправляющего одиночную ошибку на сегменте из трех кодовых кадров. Проиллюстрировать работу кодера и декодера на примере.

6. Схема кодера линейного блочного кода приведена на рис. 4.2. Найти для него  $H, G, d_{min}, P_{no}, P_{nu}$ . Изобразить схему синдромного декодера.

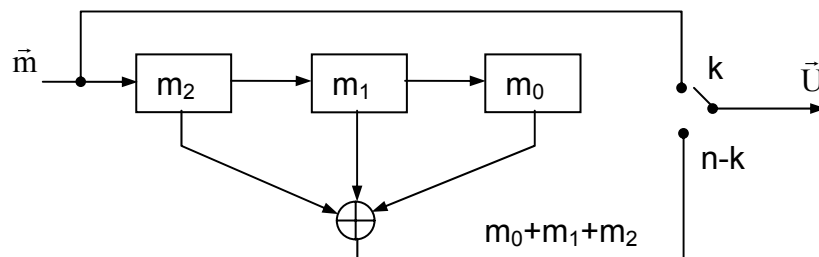


Рис. 4.2

7. Для сверточного кода со схемой рис. 2.5 ((6,3)-код) определить  $d_{min}$ , закодировать последовательность  $m = (1100000000.....)$ , декодировать принятую последовательность с двойной ошибкой в третьем кадре с использованием алгоритма Витерби и Фано.

8. Изобразить схему кодера и декодера Меггитта для циклического (8,4)-кода. Привести пример кодирования и декодирования с одиночной ошибкой.

9. По каналу связи с шумами передается двоичная информация со скоростью  $V_{nu} = 1 \text{ Мбит/с}$ . При этом в среднем 1 раз в минуту в канале возникает ошибка.

Для уменьшения частоты ошибок предложено использовать несколько вариантов кодирования:

- (7,4)-кодом Хемминга;
- сверточным (6,3)-кодом;
- кодом с порождающей матрицей вида  $G = | III |$ .

Какое из предложенных решений обеспечит больший эффект, если скорость передачи информации  $V_{ин}$  должна остаться неизменной?

10. Предложить варианты схем кодера и декодера сверточного (15,12)-кода (по аналогии с (12,9)-кодом Вайнера-Эша, исправляющего одиночную ошибку на сегменте из трех кодовых кадров).

Привести пример кодирования и декодирования.

11. В цифровой двоичной системе связи информация передается со скоростью  $2 \text{ Мбит/с}$ , при этом в среднем один раз в минуту в канале возникает ошибка.

Как изменится частота появления ошибок, если в канале производить кодирование сверточным (6,3)-кодом и для сохранения скорости передачи информации в 2 раза повысить скорость передачи двоичных символов?

12. Изобразить схему декодера Меггитта для циклического (7,3)-кода. Привести пример кодирования и декодирования с одиночной ошибкой.

13. Порождающая матрица блочного кода имеет вид  $G = | III |$ . Найти  $H, d_{min}, P_{но}, P_{ни}$  данного кода. Изобразить схему кодера и декодера.

14. В канале связи с шумами производится кодирование информации с использованием блочного кода с порождающей матрицей  $G = | 11111 |$ . На входе приемного устройства присутствует колебание  $U(t)$  вида, показанного на рис. 4.3. Сигнал  $U(t)$  подвергается дискретизации, причем на интервал длительностью в один символ приходится два отсчета  $U(t)$ .

Какое решение относительно  $m$  вынесет по принятой реализации:

- мягкий декодер максимального правдоподобия;
- жесткий мажоритарный декодер?

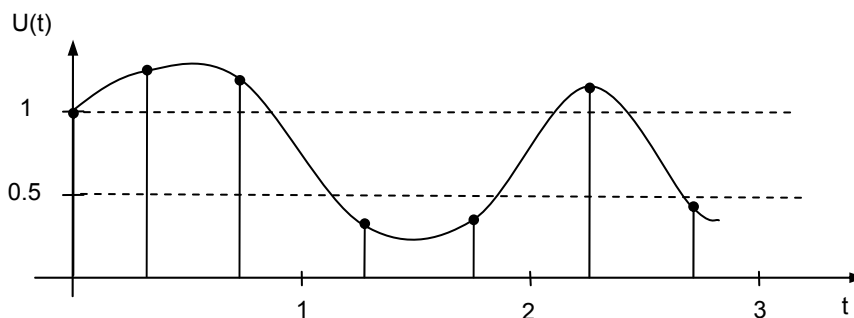


Рис. 4.3

15. Двоичный циклический код, заданный порождающим полиномом

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

позволяет исправлять пакеты ошибок длиной 2 (двойные ошибки в соседних символах).

Определить длину кода. Сконструировать декодер Меггитта для данного кода.

16. Порождающая матрица линейного блочного кода  $G$  имеет вид

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Изобразить схему кодирования и декодирования с использованием данного кода. Определить время до первой не обнаруживаемой кодом ошибки, если скорость передачи составляет  $V_{nu} = 1 \text{ Мбит/с}$ , а вероятность ошибки в канале равна  $P_{om} = 10^{-7}$ .

17. Изобразить схему, построить кодовое дерево и решетчатую диаграмму для несистематического сверточного кода с  $R = 1/3$ ,  $m = 2$  и имеющего порождающие полиномы вида

$$g_1(x) = 1 + X + X^2, \quad g_2(x) = 1 + X + X^2 \text{ и } g_3(x) = 1 + X^2.$$

18. По двоичному каналу связи передается информация со скоростью 9600 бит/с.

Сколько времени понадобится для передачи 1000 с. русского текста (энтропия  $H(\lambda) = 2$  бит/букву) с использованием примитивного равномерного двоичного кода и кода без избыточности (одна страница - 2000 букв)?

19. Итеративный код задан матрицей вида

$$U = \begin{pmatrix} m0 & m1 & p0 \\ m2 & m3 & p1 \\ p2 & p3 & p4 \end{pmatrix}.$$

Записать порождающую матрицу эквивалентного ему линейного блочного систематического  $(n,k)$ -кода. Определить исправляющую способность кода, найти вероятность неисправления ошибки, если вероятность ошибок в канале составляет  $P_{om} = 10^{-4}$ .

20. Итеративный код задан матрицей вида

$$U = \begin{vmatrix} m_0 & m_1 & m_2 & m_0 + m_1 & m_1 + m_2 \\ m_3 & m_4 & m_5 & m_3 + m_4 & m_4 + m_5 \\ m_6 & m_7 & m_8 & m_6 + m_7 & m_7 + m_8 \\ m_0 + m_3 & m_1 + m_4 & m_2 + m_5 & p_1 & p_2 \\ m_5 + m_6 & m_4 + m_7 & m_5 + m_8 & p_3 & p_4 \end{vmatrix}.$$

Проверочные символы  $P_1 \dots P_4$  формируются путем суммирования всех информационных символов, входящих в соответствующие столбцы и строки матрицы, например  $p_1 = m_0 + m_1 + m_3 + m_4 + m_6 + m_7 + m_0 + m_3 + m_1 + m_4 + m_2 + m_5$ .

Записать порождающую матрицу эквивалентного линейного блочного систематического  $(n, k)$ -кода. Определить исправляющую способность кода. Найти вероятность не исправляемой данным кодом ошибки, если вероятность ошибки в канале составляет  $P_{ош} = 10^{-3}$ .

21. Исправляющий двойные ошибки циклический  $(15, 7)$ -код БЧХ имеет порождающий полином вида

$$G(x) = X^8 + X^7 + X^6 + X^4 + 1.$$

Построить кодер и декодер Меггитта для этого кода.

22. Дискретный источник выдает символы из ансамбля  $\{ a_i \}$  объемом  $K = 50$ .

Какое минимальное число разрядов должен иметь равномерный двоичный код, предназначенный для кодирования символов данного ансамбля? Записать примеры кодовых слов. Какова избыточность примитивного кода, если энтропия источника составляет 3 бит/букву?

23. Ансамбль дискретных символов  $\{ a_i \}$  объемом  $K = 32$  имеет энтропию  $H(A) = 2$  бит/символ.

Найти минимальное количество кодовых символов, которое надо израсходовать на кодирование символа источника равномерным примитивным двоичным кодом. Какое избыточное количество символов по сравнению с оптимальным кодом приходится использовать на один символ источника при примитивном кодировании?

24. Закодировать двоичным кодом Шеннона-Фано ансамбль  $\{ a_i \}$ , если вероятность символов имеет значения, приведенные ниже.

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
0.25	0.25	0.125	0.125	0.0625	0.0625	0.0625	0.0625

Найти среднее число символов кодовой комбинации. Определить избыточность кода.

25. В цифровой системе телевидения высокой четкости (ТВЧ) передача одного кадра изображения размером  $1500 \times 1000$  элементов с числом градаций яркости  $M = 256$  производится за  $T_k = 40$  мс.

Какую полосу частот будет занимать цифровой телевизионный сигнал при использовании примитивной КИМ? Как изменится эта величина, если степень корреляции соседних элементов изображения составляет 0.95 и производится кодирование с полным устранением избыточности?

26. Некоторый дискретный источник выдает независимые символы из ансамбля  $\{ a_i \}$  ( $i = 1, 2, \dots, 9$ ) с вероятностями, определенными следующим образом:

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$
0.2	0.15	0.15	0.12	0.1	0.1	.08	0.06	0.04

Закодировать символы данного ансамбля кодом Хаффмена. Построить кодовое дерево и определить среднюю длину кодового слова.

27. Циклический (15,4)-код задан порождающим полиномом вида

$$g(x) = X^{11} + X^8 + X^5 + X^3 + X^2 + X + 1.$$

Построить кодер и декодер Меггитта для этого кода. Определить минимальное хеммингово расстояние и исправляющую способность кода. Найти вероятность неисправления ошибки, если вероятность ошибки в канале составляет  $P_{ош} = 10^{-5}$ .

28. Показать, что хороший декодер линейного блочного кода должен производить нелинейные операции, для чего доказать:

а) что процедура вычисления синдрома линейна по отношению к вектору ошибок, т.е. если  $S = F(e)$ , то  $F(e_1 + e_2) = F(e_1) + F(e_2)$ ;

б) что линейный декодер - это такой декодер, у которого функция  $e = f(S)$ , связывающая синдром и оценку вектора ошибок, удовлетворяет условию  $f(S_1 + S_2) = f(S_1) + f(S_2)$ ;

в) что если мы хотим, чтобы декодер исправлял все одиночные ошибки, то функция  $e = f(S)$ , связывающая синдром и оценку вектора ошибок, должна быть нелинейной.

Доказать, что линейный декодер может исправлять не более  $n-k$  из  $n$  возможных одиночных ошибок.

29. Полиномиальный (17,9)-код задан порождающим многочленом вида

$$g(x) = X^8 + X^5 + X^4 + X^3 + 1.$$

Определить минимальное расстояние Хемминга для данного кода. Сколько ошибок может исправить этот код? Построить кодер и декодер Меггитта для данного кода. Определить вероятность не исправляемой кодом ошибки, если вероятность ошибки в канале составляет  $P_{ош} = 10^{-3}$ .

30. Кодом с проверкой на четность называется код, который образуется путем добавления к  $k$ -разрядной информационной последовательности одного символа так, чтобы число единиц в полученном коде было четно.

Построить кодер и декодер для (8,7)-кода с проверкой на четность. Определить вероятность необнаруживаемой ошибки, если вероятность ошибки приема символа составляет  $P_{ош} = 10^{-3}$ . Изобразить схемы кодера и декодера.

31. Исправляющий три ошибки (23,12)-код является циклическим с порождающим полиномом

$$G(x) = X^{11} + X^{10} + X^6 + X^5 + X^4 + X^2 + 1,$$

или

$$G(x) = X^{11} + X^9 + X^7 + X^6 + X^5 + X + 1.$$

Найти проверочный многочлен  $h(x)$  для данного кода. Построить кодер на основе  $g(x)$ . Построить декодер Меггитта для данного кода.

32. Рассмотреть линейный блочный код, кодовое слово которого формируется по правилу :

$$U = (x_0, x_1, x_2, x_3, x_4, x_0+x_1+x_2+x_3+x_4, x_0+x_2+x_3+x_4, x_0+x_1+x_2+x_4, x_0+x_1+x_2+x_3).$$

Найти проверочную матрицу кода и параметры  $n, k$ .

33. Добавить к коду из предыдущей задачи общую проверку на четность и построить соответствующую проверочную матрицу.

Чему равно минимальное кодовое расстояние полученного кода?

34. Построить для кодов из предыдущих задач порождающие матрицы по проверочным.

35. Двоичный код, предназначенный для кодирования сообщений источника с алфавитом  $M = 8$ , содержит следующие кодовые слова:

$$U_1=00000; U_2=10011; U_3=01010; U_4=11001;$$

$$U_5=00101; U_6=10110; U_7=01111; U_8=11100.$$

Является ли данный код линейным и систематическим?

Определить возможности кода по обнаружению и исправлению ошибок.

Если код является линейным, построить порождающую и проверочную матрицы кода, схемы кодирования и декодирования.

36. Спроектировать блоки кодирования и декодирования данных для системы передачи информации.

Исходные данные:

- источник выдает информацию блоками по 4 бита;
- производительность источника = 0,4 Мбит/с;
- используется (7,3)-код Хемминга;
- затухание сигнала на трассе  $D = 150 \text{ дБ}$ ;
- коэффициент усиления передающей и приемной антенн  $G = 32$ ;
- чувствительность приемника радиолинии  $N_0 = 10^{-18} \text{ Вт/Гц}$ ;
- прием сообщения - посимвольный с использованием ортогональных сигналов.

Определить мощность передатчика системы связи, обеспечивающую вероятность безошибочного приема блока сообщения  $P = 0,999999$ .

37. Одним из способов улучшения корректирующих свойств кодов является добавление общей проверки на четность (если число единиц в кодовом слове нечетно, добавляется 1, если четно - 0), что эквивалентно перекодированию кодовых слов следующим образом:

$$U_1 = m * G_1, \quad U_2 = U_1 * G_2.$$

Записать выражение для перекодирующей матрицы  $G_2$ , соответствующей исходному (7,3)-коду Хемминга.

Записать выражение для проверочной матрицы нового кода. Изобразить схемы кодирующего и декодирующего устройств.

Как изменится вероятность необнаружения ошибки  $P(E)$  в сравнении с исходным (7,3)-кодом, если вероятность ошибки в канале  $P_{om} = 10^{-5}$  ?

38. По каналу связи передается информация со скоростью  $V = 2 \text{ кбит/с}$ . Используются двоичные сигналы типа 1 и -1. Мощность сигнала на входе приемника составляет  $P = 10^{-14} \text{ Вт}$ . Спектральная плотность мощности помех, приведенная ко входу,  $N_0 = 10^{-18} \text{ Вт/Гц}$ . При передаче используется корректирующий (7,3)-код Хемминга.

Определить число необнаруживаемых и неисправленных ошибок, проходящих по каналу связи за один час работы.

39. Для кодирования информации в системе связи используется (7,3)-код Хемминга. Скорость передачи - 1 кбит/с. В канале связи действует нормальная "белая" помеха со спектральной плотностью  $N_0 = 10^{-18}$  Вт/Гц.

При какой мощности сигнала на входе приемника вероятность неисправленной ошибки составит  $P_{ин} = 10^{-8}$  ?

40. Порождающая матрица кода имеет вид

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Найти проверочную матрицу кода; изобразить схемы кодирующего и декодирующего устройств; найти минимальное кодовое расстояние кода; определить возможности кода по обнаружению и исправлению ошибок.

Определить вероятность пропуска необнаруживаемой ошибки  $P(E)$ , если  $P_{ош} = 10^{-6}$ .

41. Порождающая матрица кода имеет вид

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Найти проверочную матрицу кода; изобразить схемы кодирующего и декодирующего устройств; найти  $d$  кода; определить возможности кода по обнаружению и исправлению ошибок.

42. Проверочная матрица имеет вид

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

Найти порождающую матрицу кода; изобразить схемы кодирующего и декодирующего устройств; найти  $d_{min}$  кода; определить возможности кода по обнаружению и исправлению ошибок.

43. С борта космического аппарата (КА) "Марс 2002" передается телевизионное изображение поверхности планеты. Размер кадра изображения - 512\*512 элементов, каждый элемент квантуется на 128

уровней и кодируется с использованием сверточного (6,3)-кода. Мощность передатчика КА - 100 Вт, коэффициент усиления антенны -  $G = 50$ .

Передача двоичных символов кодовых последовательностей осуществляется с использованием противоположных сигналов. Прием сигнала производится на антенну площадью  $S = 100 \text{ м}^2$ , чувствительность приемника -  $N = 10^{-22} \text{ Вт/Гц}$ .

За какое время может быть принят один кадр изображения, если прием производится посимвольно и отношение сигнал/шум по мощности на выходе приемника должно составить не менее 500? (Расстояние до КА  $R = 50$  млн. км.)

Как изменится это время, если вместо помехоустойчивого кодирования использовать передачу элементов изображения ортогональными сигналами и осуществлять прием в целом?

44. В ходе радиолокационного зондирования Венеры, осуществляемого радиолокатором с синтезированной апертурой, производилась передача радиоизображения поверхности кадрами размером  $2048 \times 128$  элементов, причем каждый элемент квантовался на 64 уровня. Передача осуществлялась с использованием двоичной цифровой радиопередачи, использующей противоположные сигналы.

Параметры радиопередачи связи:

- мощность передатчика  $P = 50 \text{ Вт};$
- коэффициент усиления антенны передатчика  $G = 40;$
- площадь приемной антенны  $S = 300 \text{ м}^2;$
- чувствительность приемного устройства (спектральная плотность шумов, приведенная ко входу)  $N_0 = 10^{-20} \text{ Вт/Гц};$
- максимальное расстояние  $R = 10^8 \text{ км}.$

Требуемое отношение сигнал/шум по мощности на выходе приемника радиопередачи изображения  $(P_c/P_{ш}) = 1000$ .

Сколько времени понадобится на передачу одного кадра изображения при посимвольном приеме? Как изменится это время, если кодирование элементов изображения производится сверточным (6,3)-кодом, исправляющим двойные ошибки?

45. В цифровой двоичной системе связи информация передается со скоростью 1 Мбит/с, при этом в среднем один раз в минуту в канале происходит ошибка.

Как изменится частота ошибок, если в канале использовать кодирование (7,3)-кодом Хемминга и для сохранения скорости передачи информации длительность передаваемых символов будет уменьшена в (7/3) раза? Определить величину выигрыша (проигрыша) по частоте ошибок за счет кодирования при средней вероятности ошибок в канале без кодирования  $P_{ош} = 10^{-3} \dots 10^{-6}$ .

46. Линейный блочный код задан порождающей матрицей  $G$  вида

$$G = \left| \begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right|.$$

Изобразить схему кодера и синдромного декодера для этого кода. Составить таблицу декодирования с исправлением одиночных ошибок для декодера максимального правдоподобия.

47. Составить структурные схемы кодера и синдромного декодера для циклического (7,4)-кода, заданного порождающим полиномом

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

Описать процесс кодирования и декодирования с исправлением одиночных ошибок.

48. Двоичный циклический код, заданный порождающим полиномом

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

позволяет исправлять пакеты ошибок длиной 2 (двойные ошибки в соседних символах).

Чему равна длина этого кода?

Найти минимальное кодовое расстояние данного кода.

Сконструировать систематический кодер для этого кода.

Сконструировать декодер, позволяющий исправлять пакеты по 2 ошибки.

49. Построить кодирующее и декодирующее по схеме Меггитта устройства для циклического (15,11)-кода Хемминга, имеющего порождающий многочлен вида

$$g(x) = 1 + x + x^4.$$

Сколько ошибок в принятой последовательности может обнаружить и исправить данный код?

50. Спроектировать блоки кодирования и декодирования для системы передачи данных со следующими исходными условиями:

- источник выдает непрерывный поток двоичных символов;
- производительность источника  $V = 0,3 \text{ Мбит/с}$ ;
- используется сверточный (12,9)-код Вайнера-Эша;
- декодирование сверточного кода - синдромное;

- затухание сигнала на трассе  $D = 160 \text{ дБ};$
- коэффициент усиления передающей и приемной антенн  $G = 32;$
- чувствительность приемника  $N_0 = 10^{-21} \text{ Вт/Гц};$
- прием сигнала посимвольный с использованием ортогональных (противоположных) сигналов;

Определить мощность передатчика системы связи, обеспечивающего вероятность безошибочного приема символа сообщения  $P = 10^{-10}.$

51. Напряжение полезного сигнала на входе приемника  $U = 1 \text{ мкВ};$  входное сопротивление приемника  $R = 50 \text{ Ом};$  сигналы противоположные; спектральная плотность помех на входе приемника  $N_0 = 10^{-13} \text{ Вт/Гц}.$

При какой длительности интервала наблюдения вероятность ошибочного различения двух противоположных сигналов составит  $P_{ош}=10^{-5}$ ? Как изменится величина  $P_{ош},$  если использовать не противоположные, а ортогональные сигналы?

52. Действующее значение напряжения на входе приемника  $S = 1 \text{ мкВ};$  ширина полосы пропускания приемного устройства  $F = 10 \text{ кГц};$  спектральная плотность помех, приведенная ко входу,  $N_0 = 10^{-17} \text{ Вт/Гц};$  входное сопротивление приемника  $R = 100 \text{ Ом}.$

Определить отношение сигнал/шум по мощности на входе приемника.

Определить вероятность ошибки при различении двух ортогональных сигналов длительностью  $\tau = 50 \text{ мс}.$

Построить график  $P_{ош}(\tau)$  для  $\tau = 1 \text{ мс} \dots 1 \text{ с}.$

53. Напряженность поля полезного сигнала в точке приема составляет  $E = 1 \text{ мкВ/м},$  действующая высота антенны  $h = 2.5 \text{ м},$  входное сопротивление приемника  $R = 75 \text{ Ом}.$  Полоса пропускания входных цепей приемника  $F = 50 \text{ кГц},$  спектральная плотность помех, приведенная ко входу,  $N_0 = 10^{-17} \text{ Вт/Гц}.$

Определить отношение сигнал/шум по мощности на входе приемника, а также отношение сигнал/шум по энергии на выходе, если время приема составляет  $T = 10 \text{ мс}.$

При какой длине интервала наблюдения сигнала вероятность правильного приема составит 0.9999?

54. Радиолиния связи системы охранной сигнализации предназначена для передачи сигналов 0 - 1 (0 - охрана объекта не нарушена, 1 - охрана нарушена); сигналы 0 - 1 – ортогональные; прием сигнала ведется в соответствии с оптимальным правилом различения двух известных сигналов на фоне нормального белого шума.

Радиолиния имеет следующие параметры:

- напряженность поля сигнала в точке приема - 0.5 мкВ/м ;

- действующая высота антенны приемника - 1 м;
- входное сопротивление приемника - 100 Ом;
- полоса пропускания приемника - 2 кГц;
- чувствительность приемника равна 1 мкВ при отношении сигнал/шум по мощности в полосе приема 0 дБ.

Каким должен быть интервал когерентной обработки сигнала (время интегрирования в корреляторе), чтобы ложные тревоги в системе возникали с вероятностью не более  $10^{-9}$  ?

55. Оптимальный приемник-различитель двух известных сигналов производит прием сигналов (сигналы - противоположные) на фоне нормального шума с равномерной в полосе  $\Delta F_{np}$  спектральной плотностью  $N_0 = 10^{-12} \text{ Вт/Гц}$ . Мощность сигнала на входе приемника  $P_{np} = 10^{-10} \text{ Вт}$ , интервал приема сигнала  $T$  (время интегрирования) - 0.1 с.

Определить мощность шумов на входе приемника при полосе  $\Delta F_{np} = 10 \text{ кГц}$ . Определить вероятность ошибки при различении сигналов. Как изменится мощность шума, если полосу пропускания приемника сузить до  $\Delta F_{np} = 100 \text{ Гц}$ ? Как при этом изменится вероятность ошибки различения?

56. Непрерывное сообщение  $A(t)$  с шириной полосы  $2F_m = 10 \text{ кГц}$  передается по каналу связи с использованием простейшей КИМ и посимвольного приема. Мощность полезного сигнала на входе приемника  $P = 10^{-13} \text{ Вт}$ . Спектральная плотность мешающих шумов, приведенная ко входу,  $N_0 = 10^{-18} \text{ Вт/Гц}$ , сигнал - типа 0 - 1.

Определить оптимальное с точки зрения отношения сигнал/шум на выходе приемника число уровней квантования сигнала.

57. На геостационарной орбите ( $H = 40000 \text{ км}$ ) установлен спутник непосредственного ТВ вещания, работающий в обычном ТВ стандарте:

- однополосная модуляция;
- $F = 6 \text{ МГц}$ ;
- $P = 1 \text{ кВт}$ ;
- $G = 100$ .

Спектральная плотность мешающих шумов, приведенная ко входу приемника,  $N_0 = 10^{-21} \text{ Вт/Гц}$ .

Какую площадь должна иметь антенна приемника ТВ сигнала, чтобы отношение сигнал/шум в канале изображения было не хуже  $(P_c/P_{ш}) = 50 \text{ дБ}$ .

58. Задача 57 при условии: частотная модуляция  $m_{чм} = 3$ .

59. Какую мощность должен иметь передатчик радиолинии связи (портативный радиотелефон) с амплитудной модуляцией, если:

- чувствительность приемника при отношении сигнал/шум по мощности в полосе приема  $(P_c/P_{ш}) = +10 \text{ дБ}$  - 10 мкВ;

- действующая высота антенны приемника - 0.5 м ;
- полоса пропускания приемника - 20 кГц ;
- глубина амплитудной модуляции  $m_{ам}$  - 0.3;
- затухание сигнала в канале связи - не более 90 дБ;

и требуется обеспечить отношение сигнал/шум  $(P_c / P_{ш})_{вых}$  на выходе приемника не менее + 30 дБ.

60. Над охраняемой территорией на высоте  $H = 300$  км пролетает разведывательный спутник, собирающий информацию с автоматических передающих станций. Передатчик станции работает на ненаправленную антенну ( $G = 1$ ) и передает информацию блоками по 6 бит за 1 мс. Прием сигнала на спутнике осуществляется в целом. Площадь приемной антенны  $S = 1$  м<sup>2</sup>. Спектральная плотность помех, приведенная ко входу приемника,  $N_0 = 10^{-17}$  Вт/Гц.

Какую мощность должен иметь передатчик, чтобы вероятность правильного приема блока составляла не менее  $P_{пр} = 0,99999999$  ?

Как изменится величина требуемой мощности, если перейти на посимвольный прием блока?

Как изменится вероятность правильного приема блока, если мощность передатчика не изменится, но прием производится посимвольно ?

61. Какую мощность должен иметь передатчик, установленный на геостационарном спутнике ( $H = 40000$  км), работающий на антенну с усилением  $G = 10$  и передающий информацию с полосой  $2F_m = 25$  кГц с использованием:

- 1) АМ;
- 2) ЧМ,  $m_{чм} = 10$ ;
- 3) ВИМ,  $\tau = 10$  мкс;  $\Delta F = 10$  МГц,

чтобы отношение сигнал/шум на выходе приемника было не менее 50 дБ? Площадь приемной антенны  $S = 10$  м<sup>2</sup>, спектральная плотность шумов приемника  $N_0 = 10^{-18}$  Вт/Гц.

62. В системе связи с ВИМ передается сообщение  $\lambda(t)$  с полосой  $2F_m = 10$  кГц. Для передачи используются 3 вида сигналов:

- 1) простой импульс  $\tau = 1$  мкс;  $S_0 = 1$  В;
- 2) ЛЧМ-импульс с  $\tau = 10$  мкс;  $S_0 = 0,1$  В;  $\Delta F = 10$  МГц;
- 3) ЛЧМ-импульс с  $\tau = 45$  мкс;  $S_0 = 0,5$  В;  $\Delta F = 1$  МГц.

При использовании какого сигнала отношение сигнал/шум на выходе приемника будет наибольшим?

63. Во сколько раз можно уменьшить мощность передатчика дискретной системы связи при переходе с посимвольного приема на прием в целом с использованием ортогональных сигналов, если передается одно из 128 известных сообщений и необходимо обеспечить вероятность правильного приема сообщения  $P = 0,999999$  ?

64. Полоса системы связи с ВИМ  $\Delta F = 1 \text{ МГц}$ . Спектральная мощность шумов приемника  $N_0 = 10^{-16} \text{ Вт/Гц}$ . Мощность сигнала на входе приемника  $P = 10^{-10} \text{ Вт}$ . Полоса передаваемого сообщения -  $2F_m = 20 \text{ кГц}$ .

Определить максимально достижимое отношение сигнал/шум в системе. При каком значении параметра сигнала  $\tau$  оно будет обеспечено?

65. С космического аппарата, находящегося в районе Марса ( $R = 50$  млн. км) передается телевизионное изображение. Размер кадра -  $1000 * 1000$  элементов. Точность квантования элемента изображения  $\delta A / A = 5 * 10^{-3}$ . Осуществляется передача с использованием простейшей КИМ, прием - посимвольный, сигналы 0-1 - противоположные.

Параметры радиолинии:

- $P = 100 \text{ Вт}$ ;
- $G = 100$ ;
- $S = 250 \text{ м}^2$ ;
- $N_0 = 10^{-21} \text{ Вт/Гц}$ .

За какое время может быть передан кадр изображения, если допустима вероятность ошибки в одном элементе изображения  $P_{ош} = 0.00001$ ? Как изменится это время, если прием элемента изображения производится в целом и сигналы  $S_j$  - ортогональные?

66. При работе двоичной ШСС, передающей информацию ортогональными сигналами со скоростью 1 кбит/с и занимающей полосу  $\Delta F = 1 \text{ МГц}$ , в среднем один раз в минуту допускается ошибка, обусловленная действием на входе приемника нормального белого шума. В общем канале связи может работать большое число систем связи с ШШС при условии, что сигналы различных систем ортогональны. Полагаем, что сигналы мешающих ШСС занимают ту же полосу, что и полезный сигнал, и уровни всех мешающих сигналов одинаковы и равны полезному сигналу.

При каком числе мешающих сигналов частота ошибок возрастет до одной ошибки в секунду?

67. На входе приемника двоичной ШСС, использующей для передачи символов 0 и 1 противоположные сигналы и передающей информацию со скоростью 10 кбит/с, спектральная плотность помех  $N_n$  в 50 раз превышает спектральную плотность полезного сигнала  $N_s$ .

Какую полосу должен иметь ШПС, чтобы вероятность ошибок при приеме не превышала  $P_{ош} = 0.001$ ?

68. В двоичной ШСС, использующей для передачи символов 0 и 1 ортогональные сигналы, спектральная плотность помех на входе приемника  $N_n$  в 100 раз превышает спектральную плотность сигнала  $N_s$ . Вероятность ошибок при передаче составляет  $P = 10^{-3}$ . Полоса частот, занимаемых системой,  $\Delta F = 10 \text{ МГц}$ .

Какова максимальная скорость передачи информации (кбит/с) при заданной вероятности  $P_{ош}$  ?

69. Объем информации, накапливаемой метеоспутником за один виток вокруг Земли, составляет 10000 Мбайт. Сброс информации производится при пролете над станциями сбора метеоданных, при этом интервал устойчивой радиовидимости составляет 8 минут. Передача информации производится байтами (8-битовыми словами) с использованием ортогональных сигналов. Прием слова производится в целом. Система связи имеет следующие параметры:

- коэффициент усиления бортовой передающей антенны  $G = 100$ ;
- мощность передатчика  $P = 10 \text{ Вт}$ ;
- высота орбиты спутника  $R = 5000 \text{ км}$ ;
- чувствительность приемника станции наблюдения  $N_0 = 10^{-21} \text{ Вт/Гц}$ ;
- затухание сигнала в тропосфере  $D = 6 \text{ дБ}$ .

Какую площадь должна иметь антенна станции наблюдения, чтобы вероятность ошибок при приеме одного слова не превышала  $P = 0.000001$  ?

70. Радиотелефон сотовой системы связи с модуляцией ВИМ - ШПС имеет следующие параметры :

- длительность импульса ВИМ-ШПС - 10 мкс;
- ширина полосы импульса ВИМ-ШПС - 10 МГц;
- частота дискретизации сообщения  $F_{дискр} = 2F_{max} = 10 \text{ кГц}$  ;
- индекс модуляции  $m_{вим}$  - 0.5 ;
- минимальное энергетическое отношение сигнал/белый шум на входе приемника - 300.

Какое число радиотелефонов может одновременно работать в отведенной полосе частот (10 мГц), если необходимо обеспечить отношение сигнал/шум на выходе приемника ( $P_c/P_{ш}$ )<sub>вим</sub> не менее 30 дБ? (Учсть , что при несовпадении импульсов ВИМ-ШПС по времени они не создают взаимных помех).

71. Для повышения отношения сигнал/шум на выходе системы с КИМ при посимвольном приеме иногда используется так называемая КИМ с усиленным старшим разрядом. В такой КИМ для уменьшения вероятности ошибки в старшем разряде кода, вносящей наибольший вклад в ошибку приема сообщения, производится дополнительное кодирование этого разряда, например, путем увеличения длительности символа, соответ-

ствующего этому разряду, или путем его многократной передачи и мажоритарного (по большинству) приема.

В системе цифрового вещания используется простейшая 13-разрядная КИМ с посимвольным приемом. Отношение сигнал/шум на выходе приемника с КИМ (с учетом шумов квантования и ошибок приема символов) составляет 76 дБ.

Как изменится отношение сигнал/шум на выходе приемника КИМ, если старший разряд кода подвергается трехкратной передаче и решение о его значении выносится по правилу "два из трех"?

72. Предыдущая задача при условии, что импульс, соответствующий старшему разряду кода, удлинится вдвое в сравнении с остальными разрядами, что обеспечивает меньшую вероятность ошибки при его приеме.

73. Задача 71 при условии, что дублируется и мажоритарно принимается не только старший ( $N$ -й), но и  $N-1$ -й разряд кода.

74. Как изменится отношение сигнал/шум на выходе системы с условиями задачи 71, если передача производится с использованием не 13, а 15-разрядных слов? Прием, как и ранее, посимвольный.

75. Как изменится отношение сигнал/шум на выходе приемника КИМ, если от посимвольной передачи и приема перейти к передаче блоками по три символа (пятнадцатиразрядное слово передавать как пять блоков по три символа) с кодированием блоков ортогональными сигналами  $S_1 \dots S_8$ ?

Предложить метод объединения символов исходной кодовой последовательности в блоки, обеспечивающий наилучшее качество приема сообщения.

76. При разработке системы низовой радиотелефонной связи предложены два альтернативных варианта ее построения:

1. КИМ - система с кодово-импульсной модуляцией и посимвольным приемом. Передачу символов 0 и 1 предполагается производить с использованием противоположных сигналов.

2. ВИМ-ШПС - система с время-импульсной модуляцией, в качестве импульса ВИМ использующая широкополосный шумоподобный сигнал  $S_{шум}(t)$ .

В обоих случаях система должна отвечать следующим требованиям:

- частота дискретизации передаваемого сообщения  $F_{дискр} = 10$  кГц;
- минимальное энергетическое отношение сигнал/шум на входе приемника, приведенное к интервалу дискретизации,

$$\mu = P_s * T_{дискр} / N_0 \geq 50;$$

- отношение сигнал/шум по мощности на выходе приемника в худшем случае должно быть не менее + 30 дБ.

Какой из вариантов системы, удовлетворяющих заданным требованиям, будет менее широкополосным ?

Какая из систем обеспечит лучшее отношение сигнал/шум на выходе приемника, если полоса частот, занимаемая системой, не должна превышать 100 кГц?

77. Для условий предыдущей задачи рассмотреть варианты использования амплитудной и дискретной частотной модуляции.

78. Как изменится степень предпочтения различных вариантов построения системы (задачи 77,78), если энергетическое отношение сигнал/шум в канале увеличить до 200 ?

79. Предприятие коммерческой связи предоставляет услуги по передаче дискретных сообщений по международной линии связи.

Передача производится с использованием алфавита  $\{ A_i \}$  размером в 128 букв, простейшей КИМ без избыточного кодирования с посимвольным приемом. Параметры линии связи таковы, что *при скорости передачи  $V = 33$  кбит/с в среднем допускается одна ошибка в секунду.*

Плата, взимаемая с заказчика за передачу одной страницы текста ( 2 тыс. букв), составляет 50 коп .

Штраф, выплачиваемый заказчику за одну допущенную ошибку на страницу текста, - 2 грн .

Какова *максимальная* прибыль, которую может получить предприятие связи за 1 час непрерывной работы?

80. Как изменится величина максимальной прибыли, если в канале применить помехоустойчивое кодирование (11,7)-кодом Хемминга, исправляющим одиночные ошибки ?

Как изменится размер прибыли (задача 79), если размер используемого алфавита  $\{ A_i \}$  уменьшится до 32 букв ?

81. По каналу связи с помехами с использованием двоичной системы связи (0-1) передается текстовая информация. Размер словаря сообщения (количество букв в тексте)  $K = 32$ . Средняя энтропия передаваемых сообщений составляет 3 бита/букву.

При скорости передачи  $V_n = 9.6$  кбит/с вероятность ошибки в канале составляет  $P_{ош} = 10^{-4}$ .

Сумма, взимаемая с заказчика за передачу 1 кбита информации, составляет 20 коп.

Штраф, выплачиваемый заказчику за одну допущенную ошибку, - 1 грн.

Выбрать и обосновать наилучший способ передачи информации и найти скорость передачи, обеспечивающие максимальную прибыль за 1 час работы системы:

- а) примитивное двоичное кодирование, передача с использованием сигналов типа 0 - 1;
- б) то же, сигналы - ортогональные;
- в) то же, сигналы - типа -1 +1;
- г) варианты “а” и “б”, но с применением экономного кодирования;
- д) варианты “а” и “б” с применением корректирующего кодирования (7,4)-кодом Хемминга;
- е) варианты “а” и “б” с применением сверточного (6,3)-кода;
- ж) одновременное применение экономного и корректирующего кодирования сверточным (6,3)-кодом.

82. По каналу связи с помехами с применением двоичной системы связи передается текстовая информация. Словарь сообщения (количество букв в тексте)  $K = 128$ . Средняя энтропия передаваемых сообщений составляет 4 бита/букву.

При скорости передачи  $V_n = 100$  кбит/сек вероятность ошибки в канале составляет  $P_{ош} = 10^{-5}$ .

Сумма, взываемая с заказчика за передачу 1 кбита информации, составляет 5 коп. Штраф, выплачиваемый заказчику за одну ошибку, - 1 грн.

Предложить способ передачи информации и найти скорость передачи, обеспечивающие максимальную прибыль за 1 час работы системы:

- а) примитивное двоичное кодирование, передача с использованием сигналов типа 0 - 1;
- б) то же, сигналы - ортогональные;
- в) то же, сигналы - типа -1 +1;
- г) варианты “а” и “б” но с применением экономного кодирования;
- д) варианты “а” и “б” с использованием корректирующего кодирования (7,4)-кодом Хемминга;
- е) варианты “а” и “б” с применением сверточного (6,3)-кода;
- ж) одновременное применение экономного и корректирующего кодирования сверточным (6,3)-кодом;
- з) вариант “ж”, но код (7,4);
- и) переход на восьмеричную передачу с применением ортогональных сигналов;
- к) любая из предложенных комбинаций, обеспечивающая наибольший эффект.

83. По каналу связи с помехами с использованием двоичной системы связи передается текстовая информация. Размер словаря сообщения  $K = 256$  символов. Средняя энтропия передаваемых сообщений составляет 5 бит/букву.

При скорости передачи  $V_n = 100 \text{ кбит/с}$  вероятность ошибки в канале составляет  $P_{ош} = 10^{-5}$ .

Какой из способов передачи даст лучший эффект в смысле числа ошибок:

- кодирование сверточным (6,3)-кодом;
- безызбыточное кодирование;
- и то и другое?

84. С борта орбитальной станции «АЛЬФА» с использованием цифровой системы передачи данных передается телевизионное изображение поверхности планеты. Размер кадра изображения  $N \times M$  элементов. Изображение подвергается скалярному квантованию и сжатию с применением эффективного алгоритма Хаффмена. Поток двоичных данных кодируется с использованием  $(n,k)$ -сверточного кода.

Сколько времени понадобится для приема одного кадра изображения, если прием производится посимвольно, декодирование принятой последовательности - с использованием алгоритма Витерби и отношение сигнал/шум по мощности на выходе приемника должно составить не менее  $B$  дБ?

Как изменится это время, если отказаться от корректирующего кодирования и считать, что  $P_{ош} = P_{неиспр}$  ?

При какой скорости передачи данных отношение сигнал/шум на выходе приемника ухудшится вдвое в сравнении с максимально достижимым?

Выполнить задание с использованием исходных данных, приведенных в табл. 4.1.

Таблица 4.1

Номер варианта / Исходные данные	1	2	3
Размер кадра изображения элементов $M \times N$	512x512	512x1024	512x2048
Число уровней квантования	256	512	256
Средняя энтропия источника $H(Y)$ бит/пиксел	2	3	2,5
Параметры сверточного кода $(n,k)$	(6,3)	(8,4)	(4,2)
Мощность передатчика $P$ , Вт	10	12	15
Коэффициент усиления передающей антенны $G$	200	100	150
Несущая частота радиолинии $F_0$ , ГГц	11	12	14
Затухание сигнала в тропосфере $D$ , дБ	4	6	8
Высота орбиты $H$ , км	500	400	1000
Площадь приемной антенны $S$ , м <sup>2</sup>	100	200	50
Чувствительность приемника $N_0$ , Вт/Гц	$10^{-20}$	$5 \times 10^{-21}$	$10^{-21}$
Отношение сигнал/шум по мощности на выходе приемника $B$ , дБ	40	50	44

85. При разработке системы высококачественной радиотелефонной связи предложены два альтернативных варианта ее построения:

- КИМ - система с кодово-импульсной модуляцией и посимвольным приемом; передачу символов 0 и 1 предполагается производить с использованием противоположных (КИМ-ФМ) или ортогональных (КИМ-ЧМ) сигналов;

- ВИМ - ШПС - система с время-импульсной модуляцией, в качестве импульса ВИМ использующая широкополосный сигнал  $S_{имп}(t)$ .

В обоих случаях система должна отвечать следующим требованиям:

- минимальное энергетическое отношение сигнал/шум на входе приемника, приведенное к интервалу дискретизации,  $\mu = P_s \cdot T_{дискр} / N_0 > \mu_{мин}$ ;

- отношение сигнал/шум по мощности (с учетом шумов квантования при КИМ) на выходе приемника в худшем случае должно быть не менее  $B_{мин}$  дБ.

Изобразить функциональную схему приемника РТС для обоих вариантов построения.

Обосновать выбор числа уровней квантования сигнала в системе с КИМ, обеспечивающего наибольшее отношение сигнал/шум на выходе приемника при заданном  $\mu$ .

Какой из вариантов системы, удовлетворяющих заданным требованиям, будет менее широкополосным?

Какая из систем обеспечит лучшее отношение сигнал/шум на выходе приемника, если полоса занимаемых частот не должна превышать  $\Delta F$  ?

Как изменится степень предпочтения, если полосу частот, занимаемую системой, сократить вдвое?

Как изменится степень предпочтения различных вариантов построения системы, если отношение сигнал/шум в канале  $\mu$  увеличить вдвое?

Выполнить задание с использованием исходных данных табл. 4.2.

Таблица 4.2

Номер варианта / Исходные данные	1	2	3
Частота дискретизации сигнала $F_{дискр}$ , кГц	10	8	15
Энергетическое отношение сигнал/шум $\mu$	50	100	150
Отношение сигнал/шум по мощности на выходе $B$ , дБ	50	60	55
Допустимая полоса частот $\Delta F$ , кГц	100	100	150
Индекс время-импульсной модуляции $m_{вим}$	0.5	0.7	0.6
Способ передачи сигнала с КИМ	<b>КИМ-ЧМ</b>	<b>КИМ-ФМ</b>	<b>КИМ-ЧМ</b>

86. Изобразить функциональную схему, построить кодовое дерево и решетчатую диаграмму для линейного сверточного  $(n,k)$ -кода, заданного порождающими полиномами  $G_1(X)$  и  $G_2(X)$ .

Закодировать последовательность вида  $m = (101101\dots)$ , внести в принятую последовательность одиночную ошибку и с использованием алгоритма Витерби (Фано) декодировать принятую последовательность  $r$ .

Определить вероятность пропуска неисправляемой ошибки  $P_{неиспр}$  при вероятности ошибки в канале  $P_{ош}$ .

Изобразить временные диаграммы работы схемы при подаче на ее вход кодируемой последовательности  $m$  вида  $(101101\dots)$ .

Выполнить задание с использованием данных табл. 4.3.

Таблица 4.3

Параметры кода $n, k, r$	6,3,1/2	4,2,1/2	8
Коэффициенты порождающего полинома $G1(X)$	111	10	1000
Коэффициенты порождающего полинома $G2(X)$	101	11	1001
Принятая последовательность $r$	11010001010010 .....	110101000100 ...	10100010100010 .....
Вероятность ошибки в канале $P_{ош}$	$10^{-5}$	$10^{-6}$	$10^{-4}$

87. По каналу связи с помехами с использованием двоичной системы связи передается текстовая информация.

Энергетические параметры канала связи таковы, что при скорости передачи  $V_n$ , кбит/с, вероятность ошибки в канале (обусловленной действием белого шума) составляет  $P_{ош}$ .

Сумма, взимаемая с заказчика за безошибочную передачу 1 кбита информации, составляет 0.05 грн. Штраф, выплачиваемый заказчику за одну ошибку, допущенную при приеме переданного сообщения, - 1 грн.

Изменяя скорость передачи информации, способ передачи и приема, параметры используемых сигналов, обеспечить максимальную величину чистой прибыли, получаемой за 1 час непрерывной работы системы.

Мощность передатчика и чувствительность приемника считать неизменными.

Оптимизацию проводить в рамках следующих вариантов построения системы:

- используется примитивное двоичное кодирование, передача ортогональными сигналами;
- то же, сигналы противоположные;

- применяется экономное кодирование с полным устранением избыточности в сообщении;
- используется помехоустойчивое кодирование (7,4)-кодом Хемминга;
- применяется помехоустойчивое кодирование (6,3)-сверточным кодом.
- одновременно используются все рассмотренные меры, обеспечивающие повышение чистой прибыли от системы.

Как изменится величина прибыли, если размер алфавита источника увеличится вдвое?

Выполнить задание с использованием данных табл. 4.4.

Таблица 4.4

Номер варианта / Исходные данные	1	2	3
Размер алфавита источника $K$	128	256	128
Средняя энтропия передаваемого сообщения $H$ , бит/букву	3	4	3,5
Исходная скорость передачи $V_n$ , кбит/с	9,6	19,2	2,4
Вероятность ошибки в канале при заданной скорости $V_n - P_{ош}$	$10^{-6}$	$10^{-5}$	$10^{-7}$

При решении задач можно пользоваться приведенной ниже таблицей значений интеграла вероятности  $1 - \Phi(Z)$  – табл. 4.5.

Таблица 4.5

$Z$	$1 - \Phi(Z)$
0	0.5
0.1	0.46
0.2	0.42
0.5	0.31
0.8	0.21
1.0	0.16
1.2	0.115
1.5	0.067
2.0	0.0227
2.2	0.0122
2.5	0.0062
3.0	0.0013
3.5	$2 \cdot 10^{-4}$
4.0	$3.16 \cdot 10^{-5}$
4.5	$4.43 \cdot 10^{-6}$
5.0	$2.86 \cdot 10^{-7}$
6.0	$9.86 \cdot 10^{-10}$
7.0	$1.28 \cdot 10^{-12}$
8.0	$6.22 \cdot 10^{-16}$
9.0	$1.13 \cdot 10^{-19}$

### *Библиографический список*

1. Лезин Ю.С. Введение в теорию и технику радиотехнических систем. - М.: Радио и связь, 1986.
2. Зюко А.Г., Коробов Ю.Ф. Теория передачи сигналов. - М.: Сов. радио, 1972.
3. Радиотехнические системы/ Под ред. Ю.М. Казаринова - М.: Сов. радио, 1968.
4. Чердынцев В.А. Радиотехнические системы. – Минск: Вышэйш. шк., 1988.
6. Пенин П.И. Системы передачи цифровой информации. - М.: Сов. радио, 1976.
7. Мордухович Л.Г., Степанов А.Г. Системы радиосвязи (курсовое проектирование). - М.: Радио и связь, 1987.
8. Гуткин Л.С. Проектирование радиосистем и радиоустройств. - М.: Радио и связь, 1986.
9. Пестряков В.Б., Кузенков В.Д. Радиотехнические системы. - М.: Радио и связь, 1988.
10. Калинин А.Н., Черенков Е.П. Распространение радиоволн и работа радиолиний. - М.: Радио и связь, 1971.
11. Справочник по радиорелейной связи/ Под ред. С.В. Бородича - М.: Радио и связь, 1981.
12. Коржик В.И., Финк Л.М., Шелкунов К.Н.. Расчет помехоустойчивости систем передачи дискретных сообщений. - М.: Радио и связь, 1981.
13. Тепляков И.П., Рошин Б.В. Радиосистемы передачи информации. - М.: Радио и связь, 1982.
14. Банкет В.Л., Дорофеев В.П. Цифровые методы в спутниковой связи. - М.: Радио и связь, 1988.
15. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. – М.: Радио и связь, 1987.
16. Кузьмин И.В. Основы теории информации и кодирования. - Минск: Вышэйш. шк., 1986.
17. Хемминг Р.В. Теория информации и теория кодирования. - М.: Радио и связь, 1983.

Шульгин Вячеслав Иванович

ОСНОВЫ ТЕОРИИ ПЕРЕДАЧИ ИНФОРМАЦИИ

Часть 2

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Редактор Л.А. Кузьменко

Св. план, 2003

Подписано в печать 14.02.2003

Формат 60x84 1/16. Бум. офс. №2. Офс. печ.

Усл. печ. л. 4,7. Уч.-изд. л. 5,37. Т. 30 экз. Заказ 51. Цена свободная

---

Национальный аэрокосмический университет им. Н.Е. Жуковского

« Харьковский авиационный институт »

61070, Харьков-70, ул. Чкалова,17

<http://www.khai.edu>

Издательский центр «ХАИ»

61070, Харьков-70, ул. Чкалова,17

[izdat@khai.edu](mailto:izdat@khai.edu)