

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

Г. И. Никитин

СВЕРТОЧНЫЕ КОДЫ

Учебное пособие

Санкт-Петербург
2001

УДК 621.391.254.019.4 (075)

ББК 32.811.4

Н62

Никитин Г. И.

Н62 Сверточные коды: Учеб. пособие/ СПбГУАП. СПб., 2001. 80 с.: ил.

В пособии излагаются принципы построения, формирования и обработки помехоустойчивых сверточных кодов, их определение, свойства, разновидности и эффективность, а также рассматривается реализация кодирующих и декодирующих устройств этих кодов.

Приводятся примеры применения сверточных кодов в радиотехнических системах, в том числе в сотовых и спутниковых системах связи. Рассмотрен порядок выполнения лабораторной работы "Сверточные коды".

Предназначено для радиотехнических специальностей всех форм обучения.

Рецензенты:

кафедра электронных вычислительных машин
Санкт-Петербургского государственного университета путей сообщения;
кандидат технических наук доцент *В. Н. Яночкин*

Утверждено

редакционно-издательским советом университета
в качестве учебного пособия

© Санкт-Петербургский
государственный университет
аэрокосмического приборостроения, 2001

© Г. И. Никитин, 2001

Предисловие

Подавляющее число современных систем связи работает при передаче самого широкого спектра сообщений (от телеграфа до телевидения) в цифровом виде. Из-за наличия помех в каналах связи сбой при приеме любого элемента вызывает искажение цифровых данных, что может привести, особенно в космических системах связи, к катастрофическим последствиям. В настоящее время по каналам связи передаются цифровые данные со столь высокими требованиями к достоверности передаваемой информации, что удовлетворить эти требования традиционным совершенствованием антенно-фидерных трактов радиолиний, увеличением излучаемой мощности, снижением собственного шума приемника оказывается экономически невыгодным или просто невозможным.

Высокоэффективным средством борьбы с помехами в цифровых системах связи является применение помехоустойчивого кодирования, основанного на введении искусственной избыточности в передаваемое сообщение, что приводит к расширению используемой полосы частот и уменьшению информационной скорости передачи.

Теория и техника помехоустойчивого кодирования прошли несколько этапов в своем развитии от эмпирического использования простейших кодов с повторением, с постоянным весом, с одной проверкой на четность до создания основ математической теории – ответвления высшей алгебры и теории чисел с приложением теории к реальным системам связи.

Многообразие существующих кодов делится на два класса: блочные коды и непрерывные коды. В блочных кодах передаваемая информационная последовательность разбивается на отдельные блоки с добавлением к каждому блоку определенного числа проверочных символов. Кодовые комбинации кодируются и декодируются независимо друг от друга. В непрерывных кодах, называемых также цепными, рекуррентными, конволюционными или сверточными, передаваемая информационная последовательность не разделяется на блоки, а проверочные сим-

волы размещаются в определенном порядке между информационными. Процессы кодирования и декодирования также осуществляются в непрерывном режиме.

В связи с прогрессом в теории и технике кодирования в современных системах связи используются в той или иной степени помехоустойчивые коды. Так, в системах персонального радиовызова (пейджинговые системы) используются блочные циклические коды, в сотовых системах связи применяются как блочные, так и сверточные коды, в подавляющем большинстве спутниковых систем связи, в основном, используются непрерывные сверточные коды.

Настоящее учебное пособие является дополнением лекционного курса “Радиотехнические системы передачи информации” [1]. В процессе изучения курса РТС ПИ студенты выполняют 4 лабораторных работы, связанные с теорией и практикой кодирования: “Первичные коды” [2], “Эффективные коды” [3], “Корректирующие коды” [4] и “Циклические коды” [5]. Изучив материалы данного учебного пособия, студенты должны выполнить 5-ю лабораторную работу “Сверточные коды”.

Материал пособия написан в соответствии с образовательным стандартом *для инженеров*: направление 654200 – “Радиотехника”, специализация 200700 – радиотехника; *для магистров*: направление 552500 – “Радиотехника”; *для всех форм обучения*: направление 654100 – “Электроника и микроэлектроника”, специализация 200400 – промышленная электроника.

Автор выражает признательность и благодарность студенту группы 2502 К. Б. Тисленко за помощь при подготовке материалов пособия, за написание раздела “Порядок выполнения лабораторной работы”, за подготовку программы выполнения работы на компьютерах.

1. ПЕРВЫЙ РЕКУРРЕНТНЫЙ КОД ФИНКА

Первый непрерывный рекуррентный код был предложен в 1955 г. нашим соотечественником Л. М. Финком [6], который долгие годы заведовал кафедрой в Ленинградской военной академии связи, а затем работал в электротехническом институте связи им. М. А. Бонч-Бруевича.

Однако западные специалисты имели слабое представление о работах наших отечественных ученых в области кодирования и поэтому лишь спустя 4 года (в 1959 г.) “вновь открытый” рекуррентный код был назван по имени его западного автора – кодом Хегельбергера [7].

В этой связи целесообразно упомянуть самокритичное высказывание в предисловии к русскому изданию капитальной монографии У. Питерсона и Э. Уэлдона “Коды, исправляющие ошибки” [8]. Авторы по существу извиняются за недостаточное внимание к публикациям наших отечественных ученых в области теории и практики кодирования и отмечают следующее: “Без сомнения, наиболее серьезным пробелом данной монографии является отсутствия обзора последних работ, выполненных в Советском Союзе. Основные из этих работ включены в книгу В. Д. Колесника и Е. Т. Мирончикова “Декодирование циклических кодов” (1968), заслуживающую самой высокой оценки [9]... В связи с этим, может быть, самым подходящим для данной книги явилось бы название “Коды, исправляющие ошибки, в Западном мире.”

Отметим, что В. Д. Колесник и Е. Т. Мирончиков в свое время закончили ЛИАП и в настоящее время (оба доктора технических наук и профессора) работают в ГУАП.

Итак, рассмотрим идею построения рекуррентного кода Финка в изложении самого Л. М. Финка, скромно умолчавшего в монографии “Теория передачи дискретных сообщений” [10] о своем авторстве.

“В этом коде последовательность кодовых символов не разделяется на отдельные кодовые комбинации. В поток информационных символов включаются корректирующие символы, так что между каждыми двумя информационными символами помещается один корректирую-

щий. Обозначая информационные символы через a_i , а корректирующие через b_i получаем такую последовательность символов:

$$a_1 b_1 a_2 b_2 a_3 b_3 \dots a_k b_k a_{k+1} b_{k+1} \dots$$

Информационные символы определяются передаваемым сообщением, а корректирующие формируются по следующему правилу:

$$b_i = a_{k-s} + a_{k+s+1} \pmod{2}, \quad (1.1)$$

где s – произвольное целое число, называемое шагом кода ($s = 0, 1, 2, \dots$).

Очевидно, что при ошибочном приеме некоторого корректирующего символа b_i соотношение (1.1) в принятой последовательности не будет выполнено для $i = k$. В случае же ошибочного приема информационного символа a_i соотношение (1.1) не будет выполняться при двух значениях k , а именно при $k_1 = i - s - 1$ и при $k_2 = i + s$. Отсюда легко вывести правило исправления ошибок при декодировании. В принятой кодовой последовательности для каждого b_k проверяется соотношение (1.1). Если оно оказалось не выполненным при двух значениях k ($k = k_1$ и $k = k_2$) и при этом

$$k_2 - k_1 = 2s + 1, \quad (1.2)$$

то информативный элемент a_{k_1+s+1} должен быть заменен на противоположный.

Очевидно, что избыточность такого кода равна $1/2$. Он позволяет исправлять все ошибочно принятые символы, кроме некоторых неудачных сочетаний. Так, если $s = 0$, он обеспечивает правильное декодирование, когда между двумя ошибочно принятыми символами имеется не менее трех (а в некоторых случаях двух) правильно принятых символов (при этом учитываются как информационные, так и корректирующие символы)."

Для наглядности в табл. 1.1, 1.2 и 1.3 графически показаны процессы форматирования (кодирования) кодов Финка при шагах $s = 0, 1$ и 2 соответственно. Там же представлены варианты декодирования принятых последовательностей с искаженными за счет помех различными символами. Искаженные символы и результаты их декодирования отображены в таблицах жирным шрифтом. Для всех рассматриваемых значений $s = 0, 1$ и 2 принята исходная информативная последовательность из 10 символов 0001101011 (для $s = 2$ – с добавлением до 14).

Как видно из рассмотренных примеров формирования кодов после суммирования по модулю 2, в соответствии с выражением (1.1), последовательность проверочных символов получается различной и определяется значением шага s .

Таблица. 1.1

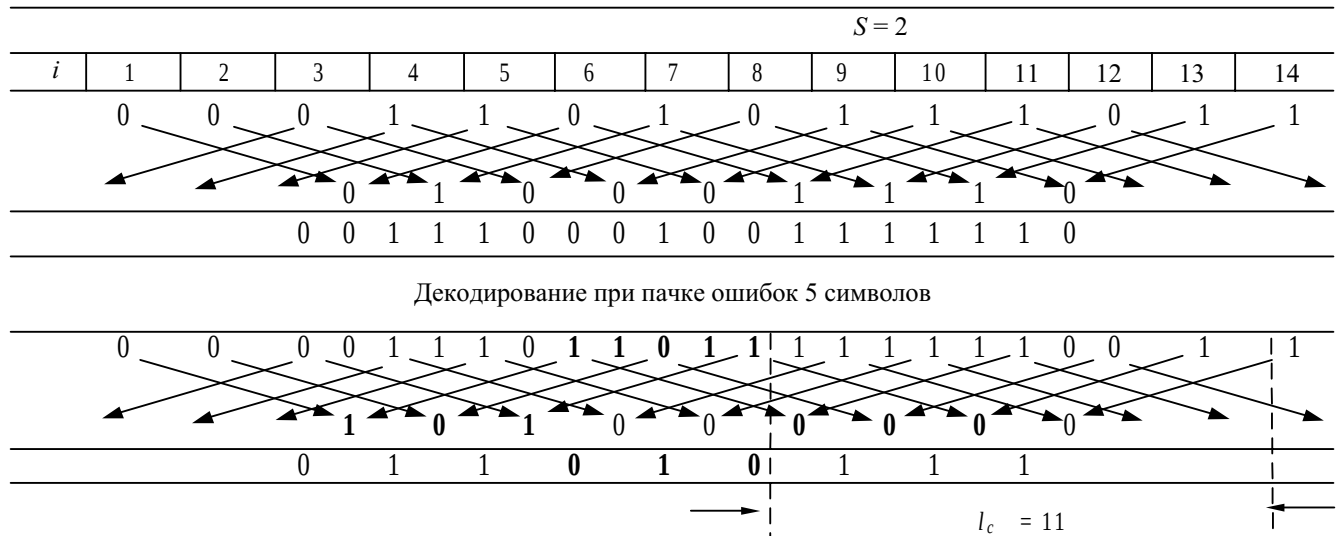
$S = 0$										
Маркировка символов, текущее значение i	a_1b_1 1	a_2b_2 2	a_3b_3 3	a_4b_4 4	a_5b_5 5	a_6b_6 6	a_7b_7 7	a_8b_8 8	a_9b_9 9	$a_{10}b_{10}$ 10
Информационные символы, суммирование по mod 2, проверочные символы	0 ↙ 0	0 ↘ 0	0 ↙ 1	1 ↘ 0	1 ↙ 1	0 ↘ 1	1 ↙ 1	0 ↘ 1	1 ↙ 1	1 ↘ 0
Выходная последовательность	0	0	0	0	0	1	1	0	1	0
Декодирование при ошибке в одном информационном символе										
Принятая последовательность, суммирование по mod 2, результат суммирования	0 ↙ 0	0 ↘ 0	0 ↙ 1	1 ↘ 1	0 ↙ 0	1 ↘ 1	1 ↙ 1	0 ↘ 1	1 ↙ 1	0 ↘ 0
Декодированная последовательность	0	0	0	1	1	0	1	0	1	
Декодирование при ошибке в одном проверочном символе										
Принятая последовательность, суммирование по mod 2, результат суммирования	0 ↙ 0	0 ↘ 0	0 ↙ 1	1 ↘ 0	0 ↙ 1	0 ↘ 1	1 ↙ 1	1 ↘ 1	0 ↙ 1	1 ↘ 0
Декодированная последовательность	0	0	0	1	1	0	1	0	1	
Декодирование при ошибке в двух проверочных символах										
Принятая последовательность, суммирование по mod 2, результат суммирования	0 ↙ 0	0 ↘ 0	0 ↙ 0	1 ↘ 0	0 ↙ 1	1 ↘ 0	1 ↙ 1	1 ↘ 1	0 ↙ 1	1 ↘ 0
Декодированная последовательность	0	0	0	1	1	0	1	0	1	
$\longrightarrow \quad \quad l_c = 3 \quad \quad \longleftarrow$										

Таблица 1.2

$S = 1$										
Текущее значение i	1	2	3	4	5	6	7	8	9	10
Информационные символы, суммирование по mod 2, проверочные символы	0	0	0	1	1	0	1	0	1	1
Выходная последовательность		0	1	0	1	1	0	1	0	0
Декодирование при ошибке в одном информационном символе										
Принятая последовательность, суммирование по mod 2, результат суммирования	0	0	1	0	1	0	0	1	0	1
Декодированная последовательность		0	0	1	1	0	1	0		
Декодирование при пачке ошибок 3 символов										
Принятая последовательность, суммирование по mod 2, результат суммирования	0	0	1	0	0	1	0	0	1	1
Декодированная последовательность		0	0	1	1	0	1	0		

$\longrightarrow$ | $l_c = 7$ | $\longleftarrow$

Таблица. 1.3



Полученные проверочные контрольные символы встраиваются между соседними информативными, образуя соответствующую входную последовательность, подлежащую передаче по каналу связи.

На стороне приема осуществляется та же самая процедура получения проверочных символов, что и на стороне передачи (1.1), и производится сравнение их с принятыми проверочными символами. Если при приеме ошибок нет, то результат суммирования по модулю 2 (сравнение) будет состоять из последовательности, содержащей одни нули. Эта последовательность, так же как в блочных циклических кодах, называется синдромом. Напомним [5], что термин “синдром” заимствован из медицинской практики (с греческого – вместе бегущий) и означает сочетание симптомов болезни, характерное для данного заболевания.

В теории кодирования синдром, который также называют опознавателем ошибок, означает совокупность признаков, характерных для определенных ошибок. Синдром полностью определяется комбинацией ошибок, которые приводят к появлению в синдромной последовательности 1 на соответствующих позициях. В табл. 1.1–1.3 непосредственно синдромы только с 1 на позициях, определяющих конфигурацию ошибок, не указаны. Их заменяют результаты суммирования по модулю 2 по алгоритму (1.1) информационных символов a_i , принимаемых с ошибками, причем символы, определяющие конфигурацию ошибок, выделены в таблицах жирным шрифтом.

Как видно из табл. 1.1, при $s = 0$ ошибка при приеме одного 5-го информационного символа приводит при декодировании к ошибке двух проверочных символов 4-го и 5-го, что позволяет исправить находящийся между ними 5-й информационный символ, в соответствии с выражениями (1.1) и (1.2).

Декодирование при ошибке в одном проверочном символе при разных значениях шага s вызывает в результате суммирования различие только в одном символе, что не влияет на правильный прием информативной последовательности.

Декодирование при ошибке в двух информационных символах (табл. 1.1, $s = 0$) показывает, что для верного декодирования ошибочные символы (4-й и 6-й) должны быть разнесены, чтобы не участвовать в формировании одинаковых проверочных символов. Для этого номера набора искаженных проверочных символов в результате суммирования должны стыковаться без перекрытия и, по крайней мере, без пропуска. В табл. 1.1 показана ситуация, когда в последовательности провероч-

ных символов неверно декодируется группа из 4 подряд следующих символов с 3-го по 6-й, которые отмечены жирным шрифтом.

Это условие “стыковки” требует наличия трех ($l_c = 3$) верно принимаемых символов между двумя искаженными информационными символами. Назовем этот интервал l_c – расстоянием стыковки.

В этом случае, когда соотношение (1.1) оказывается не выполненным для группы проверочных символов с 3-го по 6-й, в соответствии с выражением (1.2) 4-й и 6-й информационные элементы заменяются на противоположные, т. е. происходит верное исправление ошибочно принятых информационных символов с номерами $i = 4$ и $i = 6$. Однако в интервал стыковки $l_c = 3$ попадает верно принимаемый 5-й информационный символ, для которого из-за ошибочно принятых соседних (левого 4-го и правого 6-го) информационных символов также будет выполняться условие (1.2) и он, верно принятый, будет в декодере заменен на противоположный.

Для того чтобы этого не произошло необходимо увеличить число верно принимаемых символов между ошибочными информационными по крайней мере на два, чтобы пары неверно декодируемых проверочных символов не стыковались. Это приводит к тому, что стыковочное расстояние $l_c = 3$ требуется увеличить по крайней мере (для $s = 0$) на $l_d = 2$ (два дополнительных символа). Таким образом, для однозначного декодирования информационных символов требуется между двумя ошибочно принимаемыми символами иметь защитный интервал $l_0 = l_c + l_d$ из верно принимаемых символов с учетом проверочных. Для кода Финка с шагом $s = 0$ значение $l_0 = 5$. Увеличение шага ($s > 0$) способствует возможности исправления не только ошибочных символов в принимаемой последовательности, но и групп подряд следующих символов. При $s = 0$ код Финка не способен исправлять даже два подряд следующих ошибочных символа, а при шаге $s = 1$ появляется возможность исправления групп из трех соседних символов.

Подобные “серийные” ошибки возникают в результате воздействия в каналах передачи помех импульсного характера, длительность которых больше длительности одного символа. Аналогичная ситуация возникает и в каналах с кратковременными замираниями сигнала (мультипликативная помеха), в частности, в системах подвижной связи [11]. При этих условиях ошибки уже не независимы, а возникают “пачками”, общая длительность которых соответствует длительности помехи.

В этой связи интересно привести высказывание редактора перевода монографии [7] Добрушина Р.А.: “При переводе книги возникли серь-

езные терминологические трудности, связанные с отсутствием или неоднозначностью русской терминологии. Так, после долгих размышлений в качестве перевода английского термина *burst of error* был выбран термин “пачка ошибок”, хотя в русской литературе встречаются и другие варианты (“пакет ошибок”, “вспышка ошибок” и т. п.). В дальнейшем, присоединяясь к мнению Добрушина Р. Л., будем пользоваться термином пачка (пакет) ошибок.

Пачка ошибок длиной b определяется вектором ошибки, в котором все единицы заключены в последовательности b символов при условии, что крайние символы этой последовательности – единицы. Так, векторы ошибок при пачке ошибок длиной $b = 4$ могут выглядеть следующим образом (для последовательности длиной $n = 10$): 0001101000, 0100100000, 0000011110 и т. п.

Именно с целью возможности исправления пачек ошибок в коде Финка применяется отличный от нуля шаг s . При достаточно большом значении s символы, входящие в одну и ту же проверку на четность (1.1), будут разнесены по времени на столько, что состояние канала за это время успеет измениться. При этом пакет ошибок будет, как правило, захватывать только символы, не связанные друг с другом проверками на четность (1.1).

В табл. 1.2 для кода Финка с шагом $s = 1$, наряду с очевидными представлениями процесса формирования кода и декодирования при ошибке в одном информационном символе, рассмотрено декодирование принятой последовательности при пачке ошибок из трех символов. Как видно, искаженные (жирные) символы не связаны друг с другом общими проверками на четность (1.1). В связи с этим они все (2 информативных и один проверочный между ними) будут декодированы правильно. Возвращаясь к коду с шагом $s = 0$, из табл. 1.1 видно, что соседний с информационным искаженным символом проверочный символ (слева или справа), при условии его искажения, приведет к неверному декодированию принимаемой последовательности. Таким образом, при шаге $s = 0$ в коде Финка исправляемая пачка ошибок вырождается в один символ $b = 1$.

Для шага $s = 1$ и пачке ошибок из 3 символов стыковочное расстояние $l_c = 7$, а число необходимых дополнительных правильно принимаемых символов для однозначного декодирования $l_d = 6$. Это приводит к тому, что правильное декодирование очередного (даже одиночного) информативного символа или очередной пачки ошибок из трех символов возможно при защитном (однозначном) интервале с неискаженными помехами символами $l_0 = l_c + l_d = 7 + 6 = 13$.

В табл. 1.3 рассмотрены процессы формирования кода Финка с шагом $s = 2$, что дает возможность верного декодирования пачки ошибок, состоящей из 5 символов. При этом стыковочное расстояние $l_c = 11$, дополнительное $- l_d = 10$ и результирующий защитный интервал $l_0 = 21$ верно принимаемому символу между крайними пораженными помехами символами.

Таблица 1.4

В табл. 1.4 приводится сводка результатов для значений длительности исправляемой пачки ошибок b и защитного интервала l_0 по числу символов при трех значениях шага $s = 0, 1, 2$ кода Финка.

s	b	l_c	l_d	l_0
0	1	3	2	5
1	3	7	6	13
2	5	11	10	21
s_i	$2s_i+1$	$2b+1$	$2b$	$4b+1$

На основании данных табл. 1.4 устанавливается связь между шагом кода Финка s , длиной b исправляемой пачки ошибок

$$b = 2s+1 \quad (1.3)$$

и защитным интервалом

$$l_0 = 4b + 1. \quad (1.4)$$

Необходимость иметь интервал времени с неискаженными символами после прохождения пачки ошибок относится к недостаткам всех рекуррентных кодов.

Обратим внимание на то, что для верного декодирования принимаемой кодовой последовательности на стороне приема должна быть обеспечена жесткая синхронизация по тактовым импульсам, определяющим порядок следования информативных и проверочных символов. В противном случае, при сбое синхронизации, верное декодирование становится невозможным, поскольку принимаемые информационные символы могут быть восприняты как проверочные и наоборот. При рассмотрении вопросов, связанных с любым кодированием, всегда априорно предполагается наличие правильной синхронизации как по тактовым, так и по цикловым (при блочном кодировании) импульсам.

Как отмечалось ранее, код Финка остался незамеченным в Западном мире. Первые сверточные (рекуррентные) коды, исправляющие пачки ошибок, были найдены позже Хегельбергером. Вайнер и Эш в 1963 г. создали значительную часть математического аппарата теории сверточных кодов [13] и нашли хорошие коды, исправляющие одиночные ошибки. В дальнейшем методы построения сверточных кодов были обобщены на случай двоичных кодов.

2. ДВОИЧНЫЕ СВЕРТОЧНЫЕ КОДЕРЫ

Сверточное кодирование, рассмотренное в целом ряде монографий [6–27], удобнее всего описывать, характеризуя действие соответствующего кодирующего устройства. Сверточный кодер представляет собой устройство, воспринимающее за каждый такт работы в общем случае k входных информационных символов, и выдающее на выход за тот же такт n выходных символов, подлежащих передаче по каналу связи.

Отношение $R = k/n$ называют относительной скоростью кода. Выходные символы, создаваемые кодером на данном такте, зависят от m информационных символов, поступивших на этом и предыдущем тактах. Таким образом, выходные символы сверточного кодера однозначно определяются его входным сигналом и состоянием, зависящим от $m-k$ предыдущих информационных символов. Обратим внимание на то, что в коде Финка выходные символы кодера зависят как от предыдущих, так и от последующих информационных символов, в соответствии с алгоритмом (1.1), поскольку шаг кода $\pm s$ – двузначный.

Основными элементами сверточного кодера являются: регистр сдвига, сумматоры по модулю 2 и коммутатор.

Регистр сдвига является динамическим запоминающим устройством (рис 2.1), в котором хранятся двоичные символы 0 или 1. Число триггерных ячеек m в регистре сдвига и определяет память кода. В момент поступления на вход регистра нового информационного символа символ, хранящийся в крайнем правом разряде, выводится из регистра и сбрасывается. Каждый из остальных, хранящихся в регистре символов перемещается на один разряд вправо, освобождая тем самым крайний левый разряд, куда и поступает новый информационный символ.

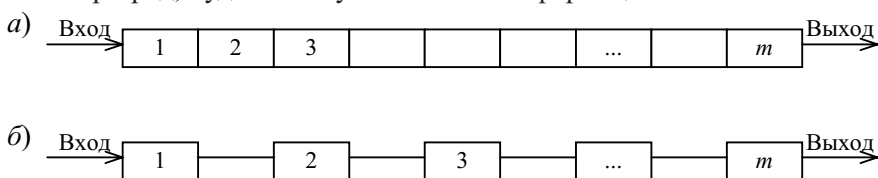


Рис. 2.1. Регистр сдвига

Используются два различных изображения регистров сдвига: с со-
стыкованными впритык ячейками (рис.2.1, *а*) и с непосредственными
последовательными связями между ячейками (рис.2.1, *б*), что дает воз-
можность на схемах встраивать между соответствующими ячейками до-
бавочные элементы, в частности, схемы суммирования по модулю 2 [5].

Заметим, что при изображении сдвигающего регистра, как правило,
не указывают шину сдвига, на которую поступают тактовые импульсы,
продвигающие записанную информацию по регистру.

Сумматор по модулю 2 осуществляет сложение поступающих на его
входы символов 0 и 1. Правило сложения по модулю 2 следующее: сум-
ма двоичных символов равна 0, если число единиц среди поступающих
на входы символов четно, и равна 1, если это число нечетно. В частно-
сти, для двухвходовой схемы $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$, где
знаком $\oplus$ обозначается сложение по модулю 2.

Изображение схем может быть таким, как показано на рис. 2.2.

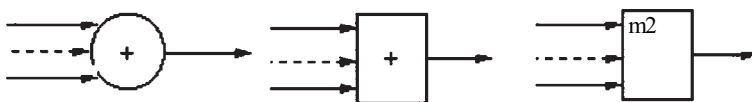


Рис. 2.2. Варианты изображений схем суммирования по модулю 2

Коммутатор осуществляет последовательное считывание поступающих
на его входы (контакты) символов и устанавливает на выходе очередность
посылки кодовых символов в канал связи. Встречаются 3 варианта изобра-
жений коммутаторов в схемах сверточных кодеров (рис. 2.3)

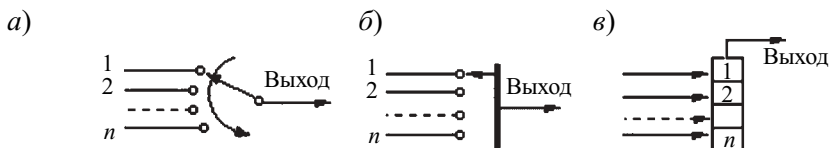


Рис. 2.3. Варианты изображений коммутаторов

На рис. 2.3, *а* представлена кольцевая схема коммутатора, на рис. 2.3, *б*
– линейная, на рис. 2.3, *в* – в виде регистра сдвига. Тактовая частота
переключения и число контактов коммутатора в сверточных кодерах
определяется относительной скоростью кода $R = k/n$, где k – число ин-
формативных символов; n – число передаваемых в канал связи сим-
волов за один такт поступления на кодер информационного символа. В

соответствии с этим число контактов (ячеек регистра сдвига коммутатора) должно быть равно n , а частота переключения должна быть в n раз больше входной тактовой частоты. Так, при скорости $R = 1/2$ у коммутатора должно быть 2 контакта и переключение должно производиться с удвоенной тактовой частотой.

Для примера на рис. 2.4 приведены кодеры кода Финка с различными шагами $S = 0, 1, 2$, работающие со скоростью $R = 1/2$.

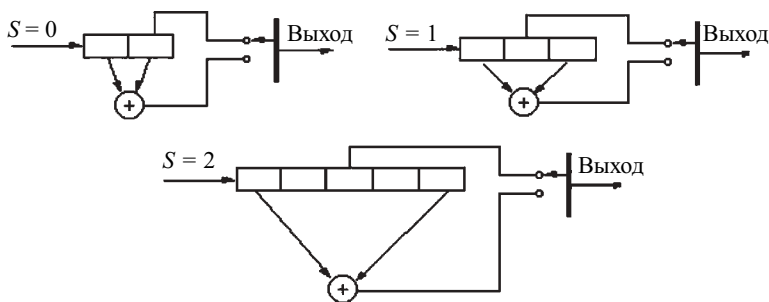


Рис. 2.4. Кодеры для кода Финка

По аналогии с блоковыми кодами, сверточные коды можно классифицировать на систематические и несистематические. Систематическим сверточным кодом является такой код, для которого в выходной последовательности кодовых символов содержится без изменения породившая ее последовательность информационных символов. В противном случае сверточный код является несистематическим. На рис. 2.5, *a* и *б* представлены, соответственно, примеры кодеров систематического и несистематического сверточного кода для $R = 1/2$. В каждом из этих кодеров входные двоичные информационные символы поступают в сдвигающий регистр, состоящий из трех ячеек, находившийся в исходном нулевом состоянии. После прихода на вход сдвигающего регистра очередного информационного символа коммутатор опрашивает два выхода в каждом из кодеров и формирует тем самым два выходных кодовых символа. В случае систематического сверточного кода (рис. 2.5, *a*) первым из выходных кодовых символов, получаемых за каждый цикл опроса коммутатора, всегда будет очередной информационный символ, поступивший в сдвигающий регистр. Из рис. 2.5, *б* можно видеть, что выходная последовательность кодовых символов не содержит входные информационные символы в неизменном виде, поэтому кодер будет порождать несистематический сверточный код.

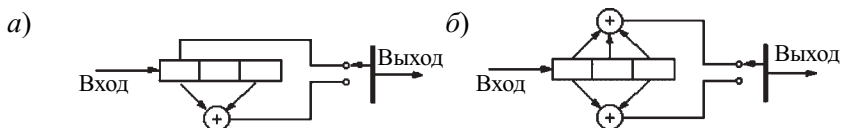


Рис. 2.5. Примеры кодеров систематического (а) и несистематического (б) сверточного кода

В общем случае сдвигающий регистр кодера сверточного кода (рис. 2.6) содержит m ячеек, а коммутатор делает один цикл опроса при приходе $1 \leq k < m$ очередных информационных символов, где m кратно k , опрашивая за один цикл $n \geq 2$ выходов кодера. При этом, очевидно, влияние любого входного информационного символа будет распространяться на $l_{\Pi} = mn/k$ выходных кодовых символов. Эта величина называется полной длиной кодового ограничения и играет роль, аналогичную блоковой длине кода при блочном кодировании. Длина кодового ограничения и конкретный выбор связей с ячейками сдвигающего регистра на сумматоры по модулю 2 будут определять корректирующие свойства получаемого сверточного кода.

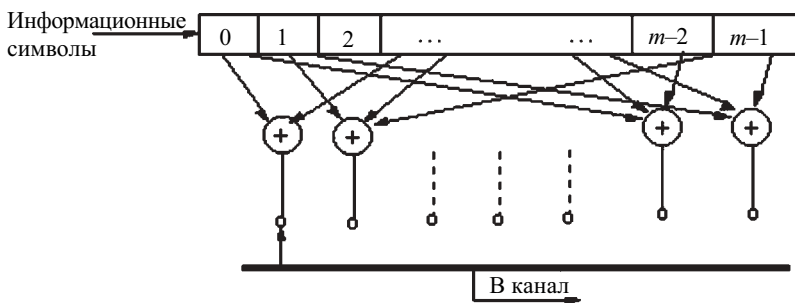


Рис. 2.6. Общий вид двоичного сверточного кодера

Для того чтобы задать структуру сверточного кодера, необходимо указать, какие разряды регистра сдвига связаны с каждым из сумматоров по модулю 2, счет разрядов ведется слева направо. Связи j -го сумматора по модулю 2 описываются путем задания j -й порождающей последовательности

$$g_j = (g_{j0}, g_{j1}, g_{j2}, \dots, g_{jm-1}), \quad (2.1)$$

где компонента

$$g_{ji} = \begin{cases} 1, & \text{если } i\text{-й разряд регистра связан с } j\text{-м сумматором;} \\ 0, & \text{в противном случае.} \end{cases} \quad (2.2)$$

Типичные параметры сверточных кодов: $k, n = 1, 2, \dots, 8$; $R = k/n = 1/4, \dots, 7/8$; $m = 2, \dots, 10$ [15].

Наиболее часто на практике применяются сверточные коды со скоростью $R = 1/2$ (рис. 2.5). При значении $R = 1/n$ структура кодера остается неизменной, но два сумматора по модулю 2 заменяются на n сумматоров, образующих n выходных символов для каждого информационного символа, поступающего на вход кодера. На рис. 2.7, а приведена схема сверточного кодера для скорости $R = 1/3$.

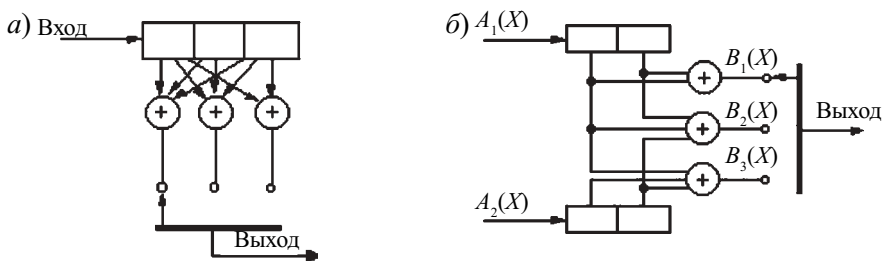


Рис. 2.7. Примеры кодеров для скоростей 1/3(а) и 2/3(б)

При скоростях $R = k/n$, где $k > 1$, как правило, в схеме кодера используют k регистров сдвига. Типичным является пример, показанный на рис. 2.7, б для кода с $R = 2/3$. В кодер одновременно вводятся два символа (один для входной последовательности $A_1(X)$, другой – для $A_2(X)$) и сумматоры, по модулю 2 вычисляют три символа для выходных последовательностей $B_1(X)$, $B_2(X)$, $B_3(X)$.

Сверточный код удобно задавать посредством порождающих (производящих) многочленов, определяемых видом последовательностей (2.1), подобно тому, как это делается для линейных блочных циклических кодов [5]. Порождающие многочлены полностью определяют структуру двоичного кодера сверточного кода. Выходные кодовые символы можно представить в виде свертки последовательности информационных символов и порождающих многочленов кода, задающих линейные рекуррентные правила кодирования, но об этом в следующем разделе.

3. ПРЕДСТАВЛЕНИЕ СВЕРТОЧНЫХ КОДОВ С ПОМОЩЬЮ МНОГОЧЛЕНОВ

В отличие от боковых кодов, каждый из которых описывается единственным порождающим многочленом, сверточный код требует для своего описания несколько порождающих многочленов, число которых определяется количеством n выходных символов, передаваемых за каждый такт в канал связи.

Представим последовательность информационных символов, поступающих на вход кодера, в виде многочлена (полинома)

$$A(X) = a_0 + a_1X + a_2X^2 + \dots, \quad (3.1)$$

где X^i – символ оператора задержки на i тактов работы сдвигающего регистра; $a_i = 0; 1$ – информационные двоичные символы.

Многочлены, описывающие n последовательностей кодовых символов, поступающих на вход коммутатора кодера и далее в канал связи,

$$B_j(X) = b_0^{(j)} + b_1^{(j)}X + b_2^{(j)}X^2 + \dots, \quad (3.2)$$

$b_i^{(j)} = 0; 1$ – двоичные кодовые символы на j -м входе коммутатора кодера.

В силу линейности сверточного кода

$$B_j(X) = G_j(X)A(X), \quad (3.3)$$

$$\text{где} \quad G_j(X) = g_0 + g_1X + g_2X^2 + \dots + g_{m-1}X^{m-1} - \quad (3.4)$$

j -й порождающий многочлен сверточного кода; $g_i = 0; 1$ – его двоичные коэффициенты (2.1), равные 1, если i -я ячейка ($i = 0, \dots, m-1$) сдвигающего регистра через схему суммирования связана с j -м входом коммутатора кодера, и равные нулю в противном случае.

Например, для кодера систематического сверточного кода (рис. 2.5, а) порождающие многочлены будут

$$G_1(X) = 1; \quad G_2(X) = 1 + X^2, \quad (3.5)$$

а для кодера несистематического сверточного кода (рис. 2.5, б)

$$G_1(X) = 1 + X + X^2; \quad G_2(X) = 1 + X^2. \quad (3.6)$$

Порождающие многочлены могут быть объединены в матрицу размера $k \times n$, называемую порождающей матрицей из многочленов. На-

пример, порождающие матрицы для кодеров (рис. 2.4), в соответствии с (3.5) и (3.6), записываются в виде

$$\begin{aligned} \mathbf{G}_1(X) &= [1 \ 1+X^2] \text{ и} \\ \mathbf{G}_2(X) &= [1+X+X^2 \ 1+X^2]. \end{aligned} \quad (3.7)$$

Строка в матрице соответствует одному из k символов входной последовательности (в данном случае $k = 1$), а число многочленов в строке равно числу схем суммирования по модулю 2. При $k > 1$ некоторые порождающие многочлены могут равняться нулю.

Так, для схемы кодера (рис. 2.7, б) при скорости $R = 2/3$, выходной код описывается шестью порождающими полиномами, задаваемыми шестью наборами связей между двумя регистрами и тремя сумматорами. Связь между входными символами и выходными последовательностями может быть представлена в следующей матричной форме:

$$[B_1(X)B_2(X)B_3(X)] = [A_1(X)A_2(X)] \begin{bmatrix} G_{11}(X)G_{12}(X)G_{13}(X) \\ G_{21}(X)G_{22}(X)G_{23}(X) \end{bmatrix}. \quad (3.8)$$

В данном случае порождающая матрица многочленов (полиномов) имеет вид

$$\mathbf{G}(X) = \begin{bmatrix} 1+X & 1+X & \\ 0 & X & +X \end{bmatrix}. \quad (3.9)$$

Рассмотрим, в качестве примера, для схемы кодера (рис. 2.5, б), как кодируется последовательность информационных символов 101. Этой последовательности соответствует многочлен

$$A(X) = 1+X^2.$$

Тогда на выходе первого сумматора по модулю 2 кодера (рис. 2.5, б) последовательность кодовых символов будет 11011, ей соответствует многочлен

$$B_1(X) = 1+X+X^3+X^4.$$

На выходе второго сумматора по модулю 2 этого кодера последовательность кодовых символов будет 10001, а ей соответствует многочлен

$$B_2(X) = 1+X^4.$$

В итоге на выходе кодера будет сформирована последовательность $B(X)$ выходных символов за 5 тактов нахождения входной последовательности 101 в трехразрядном регистре:

$B_1(X)$	1	1	0	1	1
$B_2(X)$	1	0	0	0	1
$B(X)$	11	10	00	10	11
Такты	1	2	3	4	5

Нетрудно видеть, что справедливы равенства

$$\begin{aligned} B_1(X) &= A(X)G_1(X); \\ B_2(X) &= A(X)G_2(X), \end{aligned} \quad (3.10)$$

где после перемножения многочленов подобные члены приводятся по модулю 2. [5]

Используя представление сверточного кода с помощью порождающих многочленов, часто задают сверточный код посредством последовательностей коэффициентов производящих многочленов, записанных в двоичной или восьмеричной форме. Последняя запись является более компактной и используется при большой длине сдвигающего регистра кодера. Например, структуру кодера (рис. 2.5, б) можно записать в виде в двух последовательностей: 111; 101, которые в двоичной форме представляют коэффициенты соответствующих порождающих многочленов (3.6), а в восьмеричной форме – 7;5.

Двоичные последовательности коэффициентов порождающих многочленов иногда называют генераторами кодов [16].

Следует заметить, что при формировании систематического кода первый из порождающих многочленов $G_1(X) = 1$, т. е. с первого разряда регистра сдвига информационный символ непосредственно поступает в канал. При этом генератор кода в двоичной форме будет соответствовать значению 100...0 с общим числом символов, равным числу разрядов m в сдвигающем регистре. В частности, для систематического кодера (рис. 2.5, а) с порождающими многочленами (3.5) двоичные генераторы кода будут 100 и 101 (в восьмеричной форме – 4;5).

Таким образом, выбор генераторов кода тождественен заданию номеров отводов m -разрядного регистра сдвига.

В общем случае последовательность коэффициентов j -го производящего многочлена ($j = 1, \dots, n$) при подобном представлении будет иметь вид

$$G^j = \{ g_0^j, g_1^j, \dots, g_{m-1}^j \}$$

и совпадает с порождающей последовательностью кода (2.1). Если

$$A = \{ a_0, a_1, a_2, \dots \} -$$

последовательность кодируемых информационных символов, а

$$B_j = \{b_0^j, b_1^j, b_2^j \dots\} -$$

последовательность кодовых символов на j -м входе коммутатора кодера, то для любого из них, появляющегося в μ -й момент времени ($\mu = 0, 1, 2, \dots$), можно записать

$$b_\mu^j = \sum_{i=0}^{m-1} a_{\mu-i} g_i^j. \quad (3.11)$$

Таким образом, каждый кодовый символ выходной последовательности кодера сверточного кода определяется сверткой кодируемой информационной и порождающей последовательностей, что и обуславливает название сверточных кодов.

Сверточные коды являются частным случаем итеративных или рекуррентных кодов. При рекуррентном кодировании разбиение кодируемой последовательности информационных символов на блоки не производится, а кодовые символы вычисляются.

Очевидно для рекуррентных и, в частности, для сверточных кодов понятие кодового слова не имеет смысла, так как кодовые символы вычисляются по текущему блоку последних информационных символов для каждого такта работы кодера. Поэтому подобные коды называют также цепными или скользящими.

Вообще говоря, сверточные коды можно рассматривать как обобщение блочных кодов, так как за каждый такт работы кодера по текущему блоку последних информационных символов формируется определенный блок выходных кодовых символов. Однако блоки выходных кодовых символов сверточного кода, формируемые в следующие друг за другом такты работы кодера, являются функционально зависимыми, тогда как при блочном кодировании зависимость между соседними блоками кодовых символов (кодowymi словами) отсутствует. Указанное положение обуславливает определенные преимущества сверточных кодов перед блочными, так как позволяет существенно проще, чем при блочном кодировании, реализовать достаточно большую эквивалентную блочковую длину кода, определяемую длиной кодовых ограничений. Как известно, большая блочковая длина кода требуется для достижения высокой эффективности кодирования. Упрощению аппаратной реализации способствует также непрерывность обработки информации, осуществляемой при кодировании и декодировании сверточных кодов.

4. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СВЕРТОЧНЫХ КОДОВ

Сверточный кодер как конечный автомат с памятью описывают диаграммой состояний. Диаграмма состояний представляет собой направленный граф, вершины которого отождествляются с возможными состояниями кодера, а ребра, помеченные стрелками, указывают возможные переходы между состояниями. Состояние 000...0 называется нулевым, остальные – ненулевые. Над каждым из ребер записывают кодовые символы, порождаемые кодером при соответствующем переходе из состояния в состояние. Так например, диаграмма состояний для кодера сверточного кода (рис. 2.5, б) будет иметь вид, показанный на рис. 4.1, где различные состояния кодера отмечены также буквами *a*, *b*, *c*, *d*.

Внутренним состоянием кодера считают символы, содержащиеся в $(m-1)$ разрядах регистра (начиная от входа кодера). Первые два разряда кодера (рис. 2.5, б) с $m = 3$ могут находиться в одном из четырех состояний – 00, 10, 11 и 01. Эти состояния соответствуют вершинам графа (рис. 4.1).

Диаграмма построена следующим образом. Первоначально кодер находится в состоянии 00 и поступление на вход символа 0 переводит его также в состояние 00. На выходе кодера будут символы 00. На диаграмме этот переход обозначают петлей 00 около состояния 00.

Далее при поступлении символа 1 кодер переходит в состояние 10 и на его выходе будут символы 11. Этот переход из состояния 00 в состояние 10 обозначают стрелкой (ребром). Затем возможно поступление символа 0 или 1. Кодер переходит в состояние 01 либо 11, а символы на выходе будут 10 или 01 соответственно. Построение диаграммы состояний заканчивается, когда просмотрены возможные переходы из каждого состояния во все остальные.

Если, например, в сдвигающем регистре кодера перед очередным опросом ком-

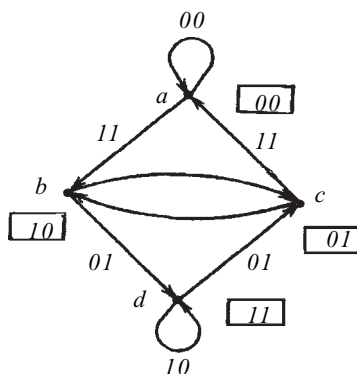


Рис. 4.1. Диаграмма состояний кодера (рис. 2.5, б)

мутатора содержится 011, то на данной диаграмме переходов это отображается переходом из состояния 11(*d*) в состояние 01(*c*); порожденные при этом кодовые символы будут 01, они записаны над соответствующим ребром, помеченным стрелкой. Определенная последовательность кодируемых информационных символов задает конкретную последовательность смены состояний и при этом порождаются кодовые символы, записанные над соответствующими ребрами, соединяющими состояния кодера на диаграмме состояний.

Рассматриваемую диаграмму состояний можно развернуть во времени, при этом получим так называемую решеточную (решетчатую) диаграмму. Так, например, решеточная диаграмма для кодера (рис. 2.5, б) диаграмма состояний которого представлена на рис. 4.1, показана на рис. 4.2. На ней принято, что штриховые линии (ветви) соответствуют переходам, происходящим при приходе информационного символа 1, а сплошные линии (ветви) — информационного символа 0. Из решеточной диаграммы видно, что ее структура после окончания “переходного процесса” в кодере становится повторяющейся. Так, на рис. 4.2 подобная повторяемость структуры решеточной диаграммы будет возможна после третьего такта работы кодера, так как при поступлении в кодер четвертого информационного символа первый символ покидает регистр сдвига и более не оказывает влияния на формирование кодовых символов. Важное значение решетчатого представления состоит в том, что с ростом числа входных символов число вершин в решетке не растет, а остается равным 2^{m-1} , где m — число ячеек в регистре сдвига, необходимое для кодирования.

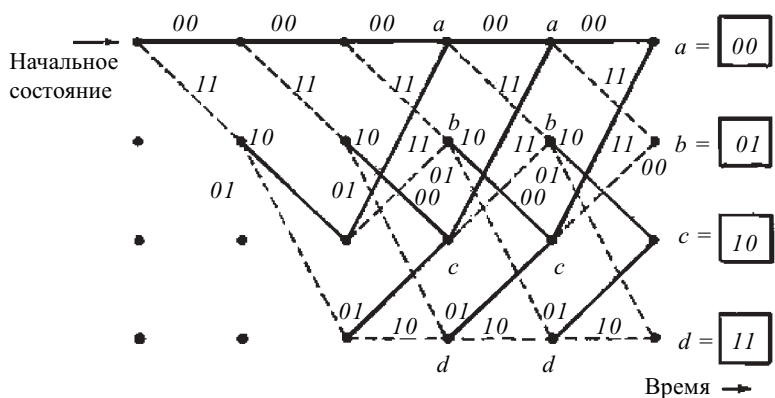


Рис. 4.2. Решеточная диаграмма кодера (рис. 2.5, б)

Решетчатая диаграмма показывает все разрешенные пути, по которым может продвигаться кодер при кодировании. Например, при поступлении на вход кодера последовательности 1011... путь по решетке (1 – пунктирная, 0 – сплошная линия) даст возможность получить конфигурацию выходной последовательности 11100001...

Каждой информационной последовательности символов соответствует определенный путь (определенная траектория) на диаграмме. Кодовая последовательность на выходе формируется путем считывания комбинаций над ветвями при прослеживании данной траектории. Таким образом, решетчатая диаграмма однозначно связывает информационную последовательность, последовательность состояний кодера и последовательность символов на его выходе.

Удобным графическим аппаратом для исследования кодирования и декодирования сверточных кодов является также кодовое дерево, строящееся на основании диаграммы состояний и решетчатой диаграммы [8, 14–15, 24–25]. Коды, допускающие подобное представление с помощью кодового дерева, называются древовидными. Таким образом, сверточные коды относятся к древовидным кодам.

5. ОСНОВНЫЕ ПАРАМЕТРЫ И ХАРАКТЕРИСТИКИ СВЕРТОЧНЫХ КОДОВ

При сверточном кодировании преобразование информационных последовательностей в выходные кодовые происходит непрерывно. Кодер двоичного сверточного кода содержит сдвигающий регистр из m разрядов и сумматоры по модулю 2 для образования кодовых символов выходной последовательности. Входы сумматоров соединены с определенными разрядами регистра. Коммутатор на выходе устанавливает очередность посылки кодовых символов в канал связи. В соответствии с такой идеологией формирования сверточных кодов можно перечислить ряд параметров и характеристик, определяющих структуру кодов.

1. *Число информационных символов, поступающих за один такт на вход кодера – k .*

2. *Число символов на выходе кодера – n , соответствующих k , поступившим на вход символам за такт.*

3. *Скорость кода определяется соотношением*

$$R = k/n \quad (5.1)$$

и характеризует избыточность, вводимую при кодировании. Так, для кодов Финка (рис. 2.4) и кодеров (рис. 2.5) значение $R = 1/2$. Для кодера (рис. 2.7, а) скорость кода $R = 1/3$, а для кодера (рис. 2.7, б) – $R = 2/3$, т. е. на вход кодера одновременно поступает $k = 2$ информационных символа, а на выходе образуется $n = 3$ кодовых символа за один входной такт. Типичными являются скорости $R = 1/n$ и $R = n-1/n$ ($R = 2/3$ на рис. 2.7, б). Большие скорости кода позволяют увеличить пропускную способность канала связи, зато снижение скорости уменьшает количество ошибок на выходе приемника.

4. *Избыточность кода*

$$\alpha = 1 - R = 1 - k/n. \quad (5.2)$$

В частности, при $R = 1/3$ число символов в выходной последовательности превышает их число во входной информационной последовательности в 3 раза. Однако несмотря на большую избыточность, сверточ-

ные коды находят широкое распространение благодаря хорошим корректирующим свойствам.

5. *Память кода*, называемая также входной длиной кодового ограничения или информационной длиной кодового слова [18], определяется максимальной степенью порождающего многочлена в составе порождающей матрицы $\mathbf{G}(X)$ (3.9):

$$l = k \max_{i,j} \lfloor \deg g_{i,j}(x) + 1 \rfloor, \quad (5.3)$$

где $\deg g_{i,j}(x)$ – степень (от *degree* – англ.) порождающего многочлена.

В свою очередь, максимальная степень одного из порождающих многочленов определяет число разрядов m в регистре сдвига кодера (рис. 2.6). С учетом младшего члена многочлена $X^0 = 1$ значение

$$m = \max \lfloor \deg g_{i,j}(x) + 1 \rfloor \quad (5.4)$$

и задает необходимое число разрядов в регистре сдвига.

Таким образом, память кода (входная длина кодового ограничения) задается простым выражением

$$l = km, \quad (5.5)$$

причем при единственном ($k = 1$) входном информационном символе, поступающем в кодер для формирования сверточного кода за один такт, $l = m$, т. е. память кода определяется числом ячеек в регистре сдвига, необходимым для кодирования.

Обратим внимание на то, что в ряде монографий [6–8, 18] при изображении сверточных кодеров младший член порождающих полиномов $X^0 = 1$ вводится в соответствующую схему суммирования по модулю 2 или непосредственно на контакт коммутатора не с первой ячейки регистра сдвига кодера, как это показано на рис. 2.6 и в других схемах сверточных кодеров, а непосредственно со входной шины. При этом число требуемых разрядов в регистре сдвига на один уменьшается. Число элементов памяти сдвигающего регистра определяет сложность кодера.

6. *Полная длина кодового ограничения (ДКО)*, называемая также ДКО по выходу кодера [23] или кодовой длиной блока [18], представляется числом кодовых символов, порождаемых кодером в промежутке времени между поступлением в него данного информативного символа и выходом в канал соответствующего символа, в формировании которого входной символ принимает участие. С учетом того, что каждые k запи-

санные в сдвигающем регистре кодера информационные символы представляются на выходе n символами, полная ДКО будет определяться выражением [18]

$$l_{\Pi} = n \max \left[\deg g_{ij}(x) + 1 \right], \quad (5.6)$$

или в соответствии с (5.4) и (5.5)

$$l_{\Pi} = nm = \frac{ln}{k}. \quad (5.7)$$

Значение l_{Π} показывает на какое максимальное число выходных символов влияет данный информационный символ и играет ту же роль, что и длина блочного кода [5]. Величина полной ДКО характеризует протяженность корреляционных связей в закодированной последовательности для одного информационного символа.

В табл. 5.1 приведены последовательности символов на выходах соответствующих кодеров при поступлении на вход информационной последовательности, содержащей только одну единицу – $A(X) = 1000\dots$. Это дает возможность определить выходной набор символов $B(X)$ на полной длине кодового ограничения, которая подчеркнута в табл. 5.1.

Таблица 5.1

№	№ рисунка кодера	R	Порождающие многочлены и матрицы	Выходная последовательность $B(X)$	L	l_{Π}	$d_{\min}$
1	2.5, а	1/2	$\begin{matrix} 1 \\ 1+X^2 \end{matrix}$	$\underline{1} \ \underline{1} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{1} \ 0 \ 0 \ 0 \ \dots$	3	6	3
2	2.5, б	1/2	$\begin{matrix} 1+X+X^2 \\ 1+X^2 \end{matrix}$	$\underline{1} \ \underline{1} \ \underline{1} \ \underline{0} \ \underline{1} \ \underline{1} \ 0 \ 0 \ 0 \ \dots$	3	6	5
3	2.7, а	1/3	$\begin{matrix} 1+X+X^2 \\ 1+X+X^2 \\ 1+X^2 \end{matrix}$	$\underline{1} \ \underline{1} \ \underline{1} \ \underline{1} \ \underline{1} \ \underline{0} \ \underline{1} \ \underline{1} \ \underline{1} \ 0 \ 0 \ 0 \ \dots$	3	9	8
4	2.7, б	2/3		$\underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{0} \quad 0$	2	6	5

В табл. 5.1 указаны порождающие многочлены для трех схем сверточных кодеров (рис. 2.5 и 2.7, а) и порождающая матрица для кодера (рис. 2.7, б), определены значения памяти кодов l , полной ДКО l_{Π} и минимального свободного расстояния $d_{\min}$ (см. далее п. 10). Построены

выходные последовательности $B(X)$ при подаче на вход кодеров информационной последовательности $A(X) = 10000\dots$, которые позволяют непосредственно определить l_{Π} и $d_{\min}$. Для кодера (рис. 2.7, б), обеспечивающего скорость $R = 2/3$, последовательность $A(X)$ подается на верхний вход, а на нижний вход – $00000\dots$

7. *Маркировка сверточного кода* отличается разнообразием, но во всех обозначениях кода присутствуют основные параметры – k и n , определяющие скорость кода $R = k/n$. Так, в [18] код обозначается (n, k) , в [21] – (k, n) , но в большинстве монографий в маркировку кода вводят третий параметр, определяющий входную длину кодового ограничения (память кода) – l [6, 11, 15, 27]. Будем придерживаться этой позиции и маркировать код тремя параметрами

$$(n, k, l), \quad (5.8)$$

что позволяет по обозначению кода сразу же определить скорость кода $R = k/n$ и число разрядов в сдвигающем регистре кодера, причем для наиболее часто применяемых кодов с $R = 1/n$ маркировка кода будет – (n, k, m) .

8. *Вес w двоичных кодовых последовательностей* определяется числом “единиц”, входящих в эту последовательность или кодовые слова. Например, $w = 4$ для кодовой комбинации 101011, а $w = 2$ – вес комбинации 1001.

9. *Кодовое расстояние d* показывает степень различия между i -й и j -й кодовыми комбинациями при условии их одинаковой длины. Для любых двух двоичных кодовых комбинаций кодовое расстояние равно числу несовпадающих в них символов. В общем виде кодовое расстояние может быть определено как суммарный результат сложения по модулю 2 одноименных разрядов кодовых комбинаций –

$$d_{ij} = \sum_{k=1}^L k_{ik} \oplus k_{jk}, \quad (5.9)$$

где k_{ik} и k_{jk} – k -е символы кодовых комбинаций i и j ; L – длина кодовой комбинации.

Впервые в теорию кодирования понятие кодового расстояния ввел Р. Хемминг, поэтому расстояние d называют хемминговым.

10. *Минимальное кодовое расстояние $d_{\min}$* – это наименьшее расстояние Хемминга для набора кодовых комбинаций постоянной длины. Пе-

ребрав все возможные пары кодовых комбинаций, определяем минимальное из них

$$d_{\min} = \min d_{ij}. \quad (5.10)$$

Однако это определение справедливо для блочных кодов, имеющих постоянную длину. Сверточные коды являются непрерывными и характеризуются многими минимальными расстояниями, определяемыми длинами начальных сегментов кодовых последовательностей, между которыми берется минимальное расстояние. Число символов в принятой для обработки длине сегмента L определяет на приемной стороне число ячеек в декодирующем устройстве.

Это число символов, которое декодер должен хранить в памяти для обработки принимаемой кодовой последовательности, называется шириной окна декодирования. Если ставится цель обнаружения и исправления как можно большего числа конфигураций ошибок, то в общем случае увеличение ширины окна декодирования всегда приводит к улучшению характеристик, однако к концу концов происходит насыщение. Ширина окна декодирования должна быть не меньше полной длины кодового ограничения l_{Π} (5.7) и зачастую в несколько раз ее превышает.

В соответствии с различной длиной L обрабатываемых в декодере сегментов, минимальное расстояние Хемминга для любых пар кодовых слов называется L -м минимальным свободным расстоянием сверточного кода и обозначается d_L . Если L равно l_{Π} , то оно называется просто минимальным свободным расстоянием и будем обозначать его, как и в блочных кодах

$$d_{ln} = d_{\min}. \quad (5.11)$$

Теоретически значение наибольшего минимального свободного расстояния d_L^* можно получить, устремляя длину L обрабатываемых в декодере символов к бесконечности

$$d_{\infty} = \max_L d_L = d_L^*, \quad (5.12)$$

причем очевидно, что

$$d_{l_{\Pi}} \leq d_{l_{\Pi}+1} \leq d_{l_{\Pi}+2} \leq \dots \leq d_{\infty}. \quad (5.13)$$

Значение d_L^* оказывается полезным при поиске структур хороших сверточных кодов, поскольку эта величина указывает предел, к достижению которого надо стремиться при поиске структуры кода [18, 20].

Важной характеристикой кода, полностью определяющей его корректирующие свойства, является спектр свободных расстояний кода, т. е. перечисление количества кодовых последовательностей, имеющих данное значение свободного расстояния [15, 23].

Поскольку сверточный код является линейным кодом, то среди различных путей на решеточной диаграмме кода обязательно будет путь с нулевым весом (нулевой путь), т. е. путь, последовательность кодовых символов которого состоит полностью из нулей. Следовательно, минимальное свободное расстояние сверточного кода будет равно минимальному числу единиц, т. е. минимальному весу путей, которые расходятся и сливаются с нулевым путем. Решеточную диаграмму кода можно использовать для определения минимального свободного расстояния, если для каждой ее ветви записать вес соответствующих кодовых символов на выходе кодера, а затем подсчитать вес путей, расходящихся и сливающихся с нулевым путем. На рис. 5.1 показана подобная решеточная диаграмма с метками весов для кодера сверточного кода (рис. 2.5, б), построенная на базе решеточной диаграммы (рис. 4.2).

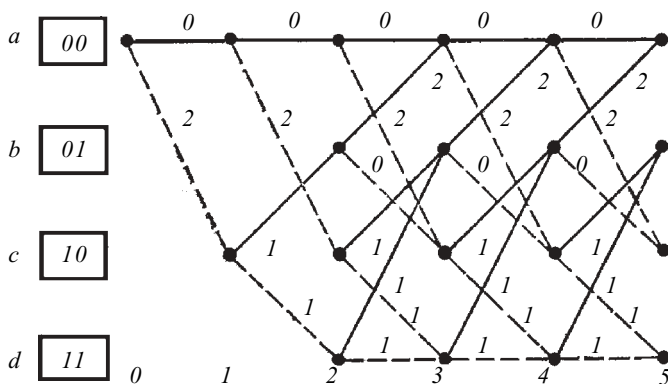


Рис. 5.1. Решеточная диаграмма с метками весов для кодера (рис. 2.5, б)

Из нее можно видеть, что из всех ненулевых путей, расходящихся и сливающихся с горизонтальным нулевым, будет один, имеющий вес 5, отходящий от нулевого пути на 3 ветви $[ab(11)-bc(10)-ca(11)]$,

два пути, отходящие от нулевого пути на 4 $[ab(11)-bd(01)-dc(01)-ca(11)]$ и 5 $[ab(11)-bc(10)-cb(00)-bc(10)-ca(11)]$ ветвей и имеющие вес $w = 6$ и т. д.

Таким образом, минимальный вес ненулевого пути для данного сверточного кода равен 5, следовательно, минимальное свободное расстояние этого кода $d_{\min} = 5$. Очевидно, этот код может исправить (см. далее п. 11) любые две ошибки, произошедшие в канале связи, так как эти две ошибки могут привести к тому, что принимаемая последовательность кодовых символов будет иметь хеммингово расстояние от переданной последовательности, равное двум, а от всех других последовательностей это расстояние будет по крайней мере не менее трех. Следовательно, при декодировании по минимуму хеммингова расстояния любые две ошибки этим кодом будут исправлены.

Очевидно потенциально корректирующая способность сверточного кода тем выше, чем больше его минимальное свободное расстояние. Поэтому структуру кодера сверточного кода, т. е. его порождающие многочлены, стремятся выбрать таким образом, чтобы максимизировать минимальное свободное расстояние кода. Это при прочих равных условиях обеспечивает наивысшую корректирующую способность сверточного кода при его декодировании по максимуму правдоподобия, в частности, при декодировании по алгоритму Витерби (см. разд. 8.3).

Наряду с оценкой минимального свободного расстояния сверточного кода по решетчатой диаграмме, можно значение $d_{\min}$ определить по последовательности выходных символов $B(X)$ – на полной длине кодового ограничения l_{Π} (5.7).

В табл. 5.1 на основании построения $B(X)$ на интервале l_{Π} в последней колонке приведены значения $d_{\min}$ соответствующих кодов, которые равны весу w последовательностей $B(X)$:

$$d_{\min} = W_{l_{\Pi}}, \quad (5.14)$$

где $W_{l_{\Pi}}$ – вес кодовой последовательности на полной длине кодового ограничения l_{Π} .

Обратим внимание на то, что для кодера на рис. 2.5, б (вторая строка табл. 5.1) выходная последовательность $B(x)$ по структуре полностью совпадает со структурой выходного кода при поступлении на вход кодера последовательности $A(X) = 1000\dots$ для решеточной диаграммы (рис. 4.2), так как путь по решетке $[ab(11)-bc(10)-ca(11)]$ определяет тождественную $B(X) = 111011000\dots$

Таким образом, минимальное свободное расстояние $d_{\min}$ для различных вариантов коротких сверточных кодов можно найти по (5.14), определив структуру выходной последовательности $B(X)$ на длине l_{Π} (табл.5.1.), что и делается при выполнении лабораторной работы.

11. *Корректирующая способность кода* определяется кратностью (количеством) исправляемых $g_{\text{И}}$ ошибок, под которой понимают гарантируемое число ошибок в кодовых комбинациях, исправляемых заданным кодом. Очевидно, что чем больше кратность $g_{\text{И}}$, тем эффективнее является код. Хемминг определил простую зависимость между значением минимального кодового расстояния и кратностью ошибки. Принятая, пораженная помехой кодовая последовательность отождествляется на приемной стороне с той из информационных, на которую она больше всего похожа, т. е. с той, от которой она отличается меньшим числом символов. Хемминг показал, что это для блочных кодов выполняется при условии

$$d_{\min} \geq 2g_{\text{И}} + 1, \quad (5.15)$$

откуда следует, что

$$g_{\text{И}} \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor, \quad (5.16)$$

где $\lfloor x \rfloor$ означает целую часть числа x .

При сверточном кодировании кратность исправляемых ошибок определяется аналогичным выражением с учетом минимального свободного расстояния d_L при обработке кодовых последовательностей с длиной L . Если в последовательности произошло не более $g_{\text{И}}$ ошибок, причем $g_{\text{И}}$ удовлетворяет неравенству

$$2g_{\text{И}} + 1 \leq d_L, \quad (5.17)$$

то ошибки исправляются.

В частности, при длине L кодовой последовательности, равной полной длине кодового ограничения l_{Π} , в соответствии с (5.11),

$$2g_{\text{И}} + 1 \leq d_{\min}, \quad (5.18)$$

что совпадает с (5.15) и (5.16) для блочных кодов, так как, постоянная длина блочного кода эквивалентна полной длине кодового ограничения для сверточных кодов.

12. *Мощность кода* определяет способность кодов исправлять множественные, одиночные и многократные ошибки, возникающие в кана-

ле связи. Мощность кода зависит от входной длины кодового ограничения l (5.5) и от вида образующих полиномов. В [23] показано, что вероятность исправления ошибок при декодировании в явном виде связана со свободным кодовым расстоянием d_L . Так, для рассмотренного ранее кодера (рис. 2.5, б), генерирующего код (2, 1, 3) с образующими полиномами G_1 и G_2 (табл. 5.1), величина $d_{\min} = 5$. Для кода (2, 1, 5) с образующими полиномами

$$\begin{aligned} G_1(X) &= 1 + X^3 + X^4, \\ G_2(X) &= 1 + X + X^3 + X^4, \end{aligned} \quad (5.19)$$

который используют при кодировании речевого сигнала в системе сотовой связи GSM, $d_{\min} = 7$. В еще более мощном коде (2, 1, 7) с образующими полиномами

$$\begin{aligned} G_1(X) &= 1 + X^2 + X^3 + X^5 + X^6, \\ G_2(X) &= 1 + X + X^2 + X^3 + X^6, \end{aligned} \quad (5.20)$$

применяемом в системах спутниковой связи, $d_{\min} = 10$.

Следовательно, коды с большим кодовым ограничением l являются более мощными.

13. *Энергетический выигрыш кода* η определяет выигрыш по помехоустойчивости при применении корректирующего кодирования. Как правило, в системах предполагают применение символов с двухпозиционной фазовой модуляцией ФМ-2, использующей противоположные сигналы с начальной фазой 0° при передаче символа “1” и 180° при передаче “0” (или наоборот).

Для получения заданного значения вероятности ошибочного приема одного символа p_1 в информационной последовательности надо обеспечить на выходе демодулятора приемника некоторое необходимое минимально допустимое отношение сигнал/шум.

При передаче информации с корректирующим кодированием уже вместо k информативных символов за заданное время требуется передача n символов с добавлением проверочных за то же время при том же уровне сигналов.

При этом придется сокращать длительность символов при передаче (при скорости $R = 1/3$ – в три раза), что потребует расширения полосы частот в n/k раз. Исходное заданное значение вероятности p_1 будет обеспечиваться уже при другом отношении сигнал/шум. Разница отношений сигнал/шум при применении кодирования и без него при ее поло-

жительном значении определяет энергетический выигрыш кода, выражаемый в децибелах.

Быстрая ориентировочная оценка энергетической эффективности для целей оперативного сравнения кодов производится по асимптотическому энергетическому выигрышу от кодирования (АЭВК)

$$\eta = 10 \lg R d_{\min} \text{ (дБ)}, \quad (5.21)$$

где $R = k/n$ – относительная скорость кода; $d_{\min}$ – минимальное кодовое расстояние.

Величина АЭВК характеризует ЭВК при вероятности $p_1 \rightarrow 0$ и является верхней границей реального ЭВК при $p_1 \neq 0$ [23].

В табл. 5.2 приведены основные характеристики коротких сверточных кодов со скоростью $R = 1/2$, с указанием значений АЭВК.

Коды заданы порождающими многочленами в восьмеричной форме записи и указаны значения памяти кода (5.5) и минимального кодового расстояния (5.11).

Таблица 5.2

№	Порождающие многочлены	R	l	$d_{\min}$	АЭВК, дБ
1	5,7	1/2	3	5	3,98
2	15,17		4	6	4,77
3	23,35		5	7	5,44
4	53,75		6	8	6,02
5	133,171		7	10	6,99
6	247,371		8	10	6,99

Выигрыш от кодирования может быть использован наиболее эффективным способом, например, путем уменьшения мощности передатчиков в системах связи, уменьшения размеров антенн или увеличения скорости передачи.

Как отмечается в [15], для получения значительного выигрыша от кодирования наиболее пригодны сверточные коды с малой длиной кодового ограничения и с декодированием по алгоритму Витерби. В частности, хорошо известный код с $R = 1/2$, $l = 6$, который имеет ЭВК 5дБ при $p_1 = 10^{-5}$, применяется во многих системах при различных скоростях передачи данных.

14. *Сложность кодеков сверточных кодов* определяет возможность практической реализации кодера на стороне передачи и декодера на стороне приема системы связи.

Сложность сверточного кодера определяется числом его простейших элементов, которыми являются разряды в регистре сдвига, сумматоры по модулю 2 и связи сумматоров с разрядами сдвигов. В большинстве случаев длина регистра сдвига имеет порядок нескольких десятков единиц, а каждый сумматор связывается приблизительно с половиной разрядов регистра. Поэтому можно считать, что сложность сверточного кодера линейно зависит от длины регистра m или от длины кодового ограничения l . Практическая реализация устройства, состоящего из нескольких десятков или сотен простейших элементов, не представляет труда.

Сложность декодеров определяется методом декодирования. В настоящее время используется три основных метода декодирования сверточных кодов: пороговое, аналогичное мажоритарному методу декодирования блочных кодов, последовательное и декодирование по алгоритму Витерби.

Наиболее простыми в реализации являются алгоритмы мажоритарного декодирования как блочных, так и сверточных кодов. Сложность реализации декодеров растет практически пропорционально полной длине кодового ограничения l_{Π} (5.7). Декодеры достаточно просты при исправлении ошибок невысокой кратности ($g_{\Pi} = 1, 2$). Однако дальнейшее увеличение кратности исправляемых ошибок приводит к значительному усложнению схемного построения декодеров, которое не оправдывается возрастанием величины ЭВК.

Наибольшую сложность имеют декодеры Витерби, объем вычислений (сложность) которых возрастает экспоненциально с ростом длины кодового ограничения. При использовании алгоритма Витерби увеличение ДКО на единицу более чем вдвое увеличивает объем декодера, но дает прирост ЭВК $-0,4 \dots -0,5$ дБ [23]. Поэтому практически используемые декодеры выполняются для кодов с ДКО $l \leq 7 \dots 8$. Повышение быстродействия таких декодеров возможно при распараллеливании процедур декодирования, а снижение объема – за счет перехода к микропроцессорной технике.

15. *Надежность кодирования* – определяется вероятностью правильного декодирования передаваемой информационной последовательности. Очевидно, что не всякий выбор связей в сверточном кодере приведет к хорошему построению кодера. Например, заведомо плохо связывать каждый из сумматоров с одними и теми же разрядами регистра.

В значительной степени надежность кодирования определяется совокупностью кодовых расстояний между последовательностями, порождаемыми кодером.

Чтобы построить хороший сверточный код, нужно найти определяющие коды, порождающие многочлены или матрицы, задающие связи между сумматорами и отводами регистра. Построение ведут исходя из получения максимального ЭВК, или минимальной ДКО l при заданном $d_{\min}$, или минимальной вероятности сбоя символа. Поиск кодов осуществляют, как правило, перебором на ЭВМ. К настоящему времени найдено большое число сверточных кодов с самыми различными параметрами для борьбы с различного рода помехами.

16. *Трудоемкость* – характеризует процесс декодирования сверточного кода и определяется числом вычислительных операций, затраченных на декодирование одного соответствующего блока принимаемой последовательности символов. Ясно, что если требуемое число операций в единицу времени при любом методе декодирования данного сверточного кода превышает быстроедействие существующей вычислительной техники, то такой код не имеет практического интереса. В этом случае потребуется, в частности, снижение технической скорости передачи.

Рассмотрев целый ряд параметров и характеристик сверточных кодов, перейдем к краткому обзору разновидностей этих кодов.

6. РАЗНОВИДНОСТИ СВЕРТОЧНЫХ КОДОВ

6.1. Систематические и несистематические коды

Если в последовательности формируемых кодером кодовых символов можно отделить $r = n - k$ избыточных символов от k информационных, то код называют систематическим. В систематическом кодере на k выходах будут информационные последовательности. На остальных $n - k$ выходах – последовательности проверочных символов, формируемых как линейные комбинации информационных.

Первые k выходов систематического кодера соединены непосредственно с первыми разрядами регистров. Коды Финка являются систематическими, на рис. 2.5 показаны: а) – систематический, б) – несистематический кодеры при скорости кода $R = 1/2$, т. е. при $k = 1$. Для кодеров с $k = 1$ при формировании систематических кодов один из порождающих многочленов либо $G_1(X) = 1$, либо $G_2(X) = 1$, чтобы информационная последовательность была частью выходной последовательности.

Систематические коды позволяют получить на приемной стороне оценку информационных символов, не производя декодирования или какой-либо иной обработки принимаемых символов. В кодовых несистематических последовательностях информация непосредственно не содержится, они должны строиться так, чтобы при отсутствии ошибок ее можно было легко восстановить в декодере.

Достоинством несистематических сверточных кодов является то, что их минимальное свободное расстояние выше, чем у систематических

Таблица 6.1

m	Несистематический код – $d_{\min}$	Систематический код – $d_{\min}$
2	3	3
3	5	4
4	6	4
5	7	5

при прочих равных условиях. Это объясняется тем, что систематический сверточный код можно получить из соответствующего несистематического, исключив один из сумматоров по модулю 2, что и приводит к уменьшению

минимального свободного расстояния кода.

В качестве примера (табл. 6.1) приведены максимальные значения минимального свободного расстояния для систематических и несистематических сверточных кодов со скоростью $R = 1/2$ при различной длине m сдвигающего регистра.

Однако несистематические сверточные коды, в отличие от систематических, могут быть катастрофическими.

6.2. Катастрофические сверточные коды

Катастрофическим называется такой сверточный код, для которого конечное число ошибок, приобретенных в канале связи, может привести к бесконечному числу ошибок в декодированных символах.

Рассмотрим кодер несистематического сверточного кода (рис. 6.1, а), диаграмма состояний которого приведена на рис. 6.1, б, где четыре различных состояния кодера обозначены буквами a, b, c, d . Положим, что на данный кодер поступила последовательность информационных символов, соответствующая такой последовательности смены состояний кодера: $abddd...dca$.

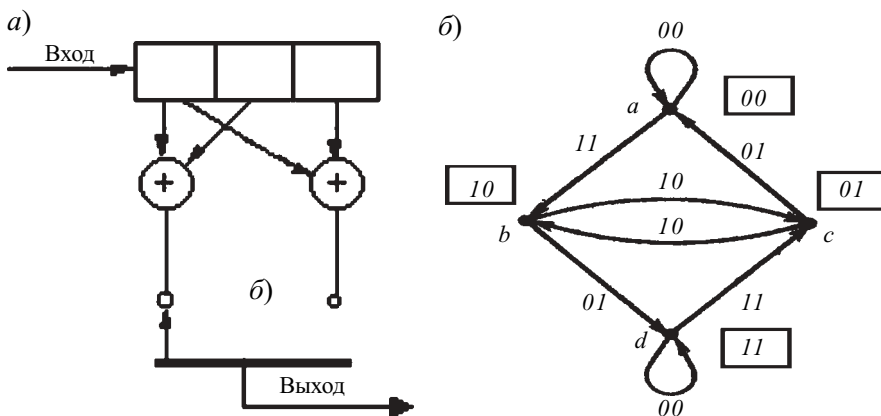


Рис. 6.1. Кодер несистематического катастрофического кода (а) и его диаграмма состояний (б)

Назовем ее истинным путем по диаграмме состояний кода. Тогда из диаграммы кодера (рис. 6.1, б) следует, что последовательность кодовых символов, передаваемая по каналу связи, будет иметь вес равный 6, вне зависимости от того, сколько раз петля находится в состоянии d . Если в канале связи четыре из шести единиц передаваемых кодовых

символов исказились и были приняты как нули, то принятая последовательность символов будет иметь меньшее расстояние Хемминга от нулевого пути, чем от истинного пути. Это приведет к тому, что при декодировании по минимуму расстояния Хемминга, т. е. по максимуму правдоподобия, нулевой путь будет выбран в качестве истинного. В результате при декодировании будет сделано произвольно большое число ошибок. Таким образом, кодер, показанный на рис. 6.1, *a*, порождает катастрофический сверточный код.

Очевидно, кодер будет порождать катастрофический сверточный код тогда, когда на диаграмме состояний этого кодера будет замкнутый путь нулевого веса, соответствующий ненулевым кодируемым информационным символам. Для двоичных сверточных кодов со скоростью $R = 1/n$, для которых каждый сумматор имеет четное число связей, всегда будет петля с нулевым весом, соответствующая состоянию из всех единиц, т. е. ненулевым информационным символам. Следовательно, все такие кодеры будут порождать катастрофический сверточный код. В общем случае необходимое и достаточное условие катастрофичности сверточного кода состоит в том, чтобы его порождающие многочлены имели общий делитель.

Например, для катастрофического кодера (рис. 6.1, *a*) с $R = 1/2$, $l = 3$ порождающие многочлены

$$\begin{aligned} G_1(X) &= 1 + X, \\ G_2(X) &= 1 + X^2. \end{aligned} \quad (6.1)$$

Очевидно, что при этом каждый сумматор имеет четное число связей (по 2) и эти генераторы кода имеют общий множитель $1 + X$. Заметим, что нулевая входная последовательность приводит к выходной последовательности 000000..., в то время как входная последовательность, состоящая целиком из единиц, приводит к выходной последовательности 11010000... Таким образом, две ошибки в первых символах приведут к бесконечному числу ошибок в информационных символах, что очевидно является катастрофой [15].

Достоинством систематических сверточных кодов является то, что они не могут быть катастрофическими. Действительно, для систематического сверточного кода каждый замкнутый путь, соответствующий ненулевым кодируемым информационным символам, никогда не будет иметь нулевой вес, так как кодируемые информационные символы обязательно присутствуют на выходе в неизменном виде.

Таким образом, при выборе структуры кодера сверточного кода порождающие многочлены кода (т. е. связи со сдвигающего регистра на сумматоры по модулю 2) стремятся выбрать таким образом, чтобы получить некатастрофический код с максимальным минимальным свободным расстоянием. Такие коды должны быть несистематическими, и порождающие их многочлены не должны иметь общих делителей. Для поиска структур таких кодеров широко используются ЭВМ.

6.3. Прозрачные сверточные коды

В настоящее время при передаче цифровых данных в большинстве систем связи применяют фазовую модуляцию и когерентный прием. При двухпозиционной фазовой модуляции (ФМ-2) символ “1” передается с начальной фазой 0, при передаче символа “1” – с фазой π . Опорная несущая, необходимая для когерентного приема в демодуляторах, выделяется, как правило, из принимаемого сигнала. При этом в опорном генераторе может спорадически возникать неоднозначность фаз, что приводит к так называемой “обратной работе”, когда все символы “1” принимаются как “0”, а все символы “0” – как “1”, т. е. возникает эффект негатива. Передача ФМ сигналов в каналах с неоднозначностью требует применения специальных мер для устранения негатива.

Впервые эту задачу решил Н. Т. Петрович в 1954 г., предложив использовать в качестве опорного напряжения предыдущую посылку для демодуляции последующей. Автор назвал этот метод – относительной фазовой телеграфией (ОФТ) [28]. В дальнейшем это название трансформировалось в относительную фазовую модуляцию (ОФМ) и в фазоразностную модуляцию (ФРМ) [29].

Аналогичным способом, допускающим использование кодов, исправляющих независимые ошибки в канале с фазовой неоднозначностью, является внешнее относительное кодирование (рис. 6.2). При этом кодер помехоустойчивого кода должен быть прозрачным для неоднозначности, т. е. должен обладать свойством инвариантности (прозрачности) по отношению к операции инвертирования кода. Это означает, что если значения входных символов поменялись на противоположные, то комбинированная выходная последовательность также инвертируется.

В этом случае в канале с ФМ-2 инверсия канальных символов, обусловленная поворотом фазы опорной несущей на π , должна приводить к инверсии всех декодированных символов на выходе декодера.



Рис. 6.2. Способ передачи по каналу с неоднозначностью с внешним относительным кодированием

Условия прозрачности сверточных кодов с $R = 1/2$, декодируемых по алгоритму Витерби, для двоичных кодов имеют вид

$$\sum_{s=0}^v G_s^{(i)} = 1, \quad (6.2)$$

где $G(X)$ – порождающий полином; v – максимальная степень $G(X)$; $i = 1, 2$ – номер полинома $G(X)$; $\sum$ – знак суммирования по модулю 2.

Выражение (6.2) означает, что число единиц, в каждом порождающем многочлене (полиноме) двоичного сверточного прозрачного к инверсии кода, должно быть нечетным.

При этом последовательность на выходе кодера, состоящая сплошь из символов 1, порождает последовательность на выходе кодера, также целиком состоящую из символов 1. В этом случае обращение полярности всех символов на выходе декодера приводит лишь к обращению полярности символов на выходе декодера.

При использовании кодов такого типа сверточную систему кодирования – декодирования следует дополнить разностным (относительным) кодированием–декодированием, как показано на рис. 6.2. Это приводит к правильной полярности декодированных данных. Разностный кодер отображает последовательность данных $A(X)$ в последовательность $A'(X)$, в которой полярность символа, соответствующая символу 1 в $A(X)$, меняется по сравнению с предыдущим, а полярность символа, соответствующая символу 0 в $A(X)$, остается неизменной.

Если в канале не меняется полярность передаваемой последовательности, то восстановленная последовательность $\hat{A}'(X)$ совпадает с $A'(X)$ (за исключением ошибок декодирования) и $\hat{A}(X)$ совпадает, конечно, с $A(X)$. Если в канале меняется полярность передаваемых данных, то при использовании прозрачного кода гарантируется, что последовательность $\hat{A}(X)$ будет совпадать с $A'(X)$. Однако ввиду разностного кодиро-

вания исходных данных, последовательность $\hat{A}(X)$ будет совпадать с $A(X)$.

Таким образом, при использовании прозрачного кода полярность символов определяется автоматически в разностном декодере, имеющем две ячейки в регистре сдвига (рис. 6.2).

Заметим, что в разностном декодере одна выходная ошибка приводит к треку из двух ошибок (спаренные ошибки) на выходе разностного декодера (как и в методе ФРМ). Соответствующие потери составляют при этом всего 0,1...0,2 дБ [15].

В табл. 6.2 наглядно показано, что при поступлении на разностный декодер последовательности символов $\hat{A}'(X)$, как в “позитиве”, так и в “негативе”, выходные последовательности $\hat{A}(X)$ будут одинаковы.

Жирным шрифтом в последовательностях $\hat{A}'(X)$ отмечены ошибки, которые приводят в выходной последовательности к треку из двух ошибок.

Таблица 6.2

ПОЗИТИВ		НЕГАТИВ	
$\oplus$	10110110 0 10	$\longleftarrow \hat{A}'(X) \longrightarrow$	01001001 1 01
	<u>10110110010</u>		<u>01001001101</u>
	1101101 0 11	$\longleftarrow \hat{A}(X) \longrightarrow$	1101101 0 11

В табл. 6.3 приведены основные характеристики прозрачных сверточных кодов для каналов с ФМ и числом фаз $M = 2$ и 4 при скорости кода $R = 1/2$. Указаны значения входной длины кодового ограничения l (5.5), величины минимального кодового расстояния $d_{\min}$ (5.10) и асимптотического ЭВК η (5.21). Порождающие многочлены $G(X)$ кодов представлены в восьмеричной форме записи.

Таблица 6.3

№	M	R	$G_1(X), G_2(X)$	l	$d_{\min}$	АЭВК, дБ
1	2	1/2	13,15	4567	66810	4,77
2			25,37			4,77
3			61,73			6,02
4			133,171			6,99
5	4	1/2	23,32	588	488	3,01
6			212,113			6,02
7			212,131			6,02

По величине АЭВК прозрачные двоичные коды не уступают в ряде случаев оптимальным кодам. Двоичные прозрачные коды используются и в каналах с многопозиционной ФМ [23].

6.4. Перфорированные сверточные коды

Практическая реализация сверточных кодов со скоростями $R = k/n$ встречает затруднения, особенно в случае больших скоростей передачи данных (несколько мегабит в секунду) [15]. Упрощение алгоритма обработки может быть получено при выборе кода с $R = 1/n$ и “выкалывании” или удалении некоторых символов в выходной последовательности для получения кода с $R = k/n$. В частности, кодовые последовательности требуемого кода с $R = 2/3$ могут быть получены из последовательностей кода с $R = 1/2$ путем периодического вычеркивания (перфорации) символов. Полученный код будет иметь три символа на выходе декодера для каждых двух информационных символов, т. е. его скорость будет равна $R = 2/3$. Такие коды называются перфорированными.

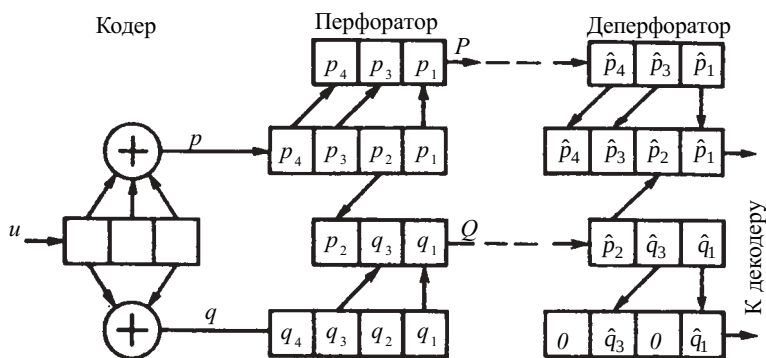


Рис. 6.3. Образование символов перфорированного кода

На рис. 6.3 показан процесс перфорации. При поступлении на вход кодера информационного символа u на его выходе образуется пара символов p и q . В перфораторе, состоящем из четырех тактируемых регистров, производится такое преобразование последовательностей p и q в кодовые последовательности P и Q , при котором символы q_2 и q_4 на выходе отсутствуют. Четырем информационным символам на входе кодера соответствуют восемь символов на его выходах $p_1 p_2 p_3 p_4$ и $q_1 q_2 q_3 q_4$ и шесть символов на выходах перфоратора $p_1 p_3 p_4$ и $q_1 q_3 p_2$. При этом скорость перфорированного кода $2/3$. На приемной стороне необходимо

обратное преобразование и вместо вычеркнутых символов в ячейки регистра вписываются, например, нули.

Декодирование кодов, полученных перфорацией исходного кода со скоростью $1/2$, можно производить декодером, рассчитанным на эту скорость. Необходимо лишь предусмотреть правильную расстановку символов перфорации.

Перфорированные коды имеют почти те же характеристики, что и наилучшие известные коды, поэтому из-за своей простоты в ряде случаев они более предпочтительны.

6.5. Сверточные коды с различными алгоритмами обработки и конфигурациями исправляемых ошибок

К настоящему времени разработано большое число различных сверточных кодов, обеспечивающих исправление ошибок различной конфигурации. Разработаны коды, исправляющие случайные ошибки, исправляющие одиночные и двойные ошибки, пачки ошибок, пачки ошибок с одновременным исправлением независимых ошибок [8, 14]. Предложен ряд алгоритмов декодирования сверточных кодов. Как правило, и коды, и алгоритмы получают название предложивших их авторов.

Первые непрерывные коды, исправляющие независимые ошибки, были описаны Элайесом (1955 г.) под названием конволюционных. Разработка таких кодов (под названием “рекуррентных”) была продолжена Килмером, который в работе 1961 г. рассмотрел пример непрерывного кода, обеспечивающего наименьшую вероятность ошибки, какую только можно получить с помощью систематического кода с заданным интервалом взаимных связей между символами (выходной ДКО – l_{Π}).

Первый найденный класс сверточных кодов, исправляющих многократные ошибки, составили самоортогональные коды Мессе. Им же разработан метод порогового декодирования [30]. Вайнер и Эш [13] нашли коды, исправляющие одиночные ошибки, и создали значительную часть математического аппарата теории сверточных кодов.

Последовательное декодирование предложил Возенкрафт [31]. Аналогичный алгоритм был предложен Фано. Нашим соотечественником Зигангировым был разработан алгоритм последовательного декодирования, получивший название стек-алгоритма [24]. Этот алгоритм во многих случаях лучше алгоритма Фано, он очень просто описывается и помогает проникнуть в существо различных принципиальных вопро-

сов последовательного декодирования. Основной недостаток последовательного декодирования – чувствительность к пакетным ошибкам.

После введения Элайесом сверточных кодов встал вопрос о том, какие коды (сверточные или блочные) потенциально лучше. Ответ на этот вопрос был дан Витерби в 1967 г., он получил верхние и нижние границы вероятности ошибки для сверточных кодов и с их помощью показал, что характеристики оптимальных сверточных кодов, как функции кодового ограничения, лучше соответствующих характеристик блочных кодов той же длины. Для получения этих границ Витерби использовал алгоритм декодирования сверточных кодов, который, как оказалось, действует как алгоритм декодирования по максимуму правдоподобия. Этот алгоритм стал называться алгоритмом Витерби.

Сверточные коды, исправляющие пакеты ошибок, как отмечалось в разд. 1, были предложены Финком, а затем Хегельбергером. В отечественной технической литературе эти коды получили название Финка-Хегельбергера [6]. В дальнейшем коды, исправляющие пакеты ошибок, изучались Килмером, а также Вайнером и Эшем [13].

Их результаты были использованы Берлекэмпом и Препарата для построения систематических сверточных кодов на базе теории матриц. Месси предложил процедуру их декодирования, и эти коды получили название Берлекэмпа-Препарата-Месси (БПМ-коды). Позже, в 1966 г., Ивадари разработал аналогичные коды, которые несколько проще в реализации, чем БПМ-коды.

Для исправления случайных ошибок и пачек ошибок Коленбург разработал так называемые диффузные коды, допускающие пороговое декодирование, вклад в исследование которых в дальнейшем внес Тонг [8, 14].

Если на начальном этапе сверточные коды были лишь объектом теоретических исследований, то в дальнейшем они стали применяться сначала в сочетании с последовательным декодированием в космической связи, а затем в сочетании с пороговым декодированием в спутниковых системах передачи данных, системах морской радиосвязи и других системах.

7. СОГЛАСОВАНИЕ МОДЕМА И КОДЕКА

Кодек помехоустойчивого кода состоит из кодера и декодера, систем синхронизации и устройств согласования с модемом (рис. 7.1).

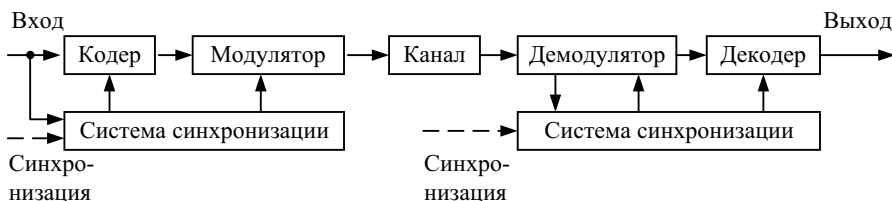


Рис. 7.1. Структурная схема модема и кодера

Системы синхронизации формируют синхропоследовательности, необходимые для работы кодера и модема. Во многих случаях синхросигналы на приемной стороне выделяются из информационных последовательностей.

Помехоустойчивые кодеки используются с модемами различных типов. На рис. 7.2 показан пример согласования кодера (рис. 2.5, б) со скоростью $1/2$ с фазовыми модуляторами ФМ-2 и ФМ-4.

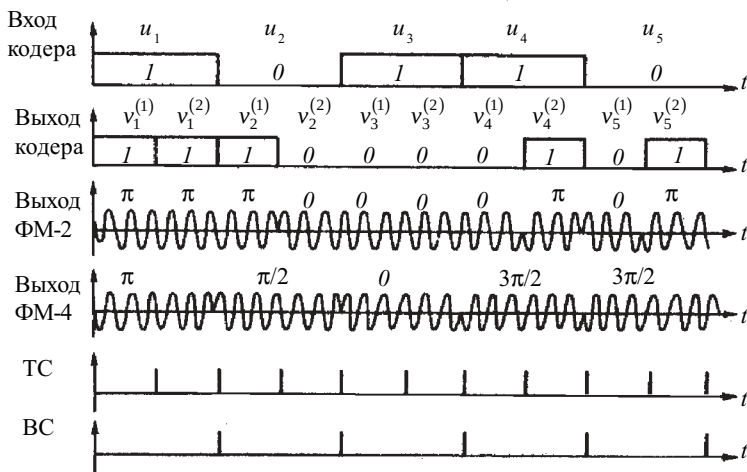


Рис. 7.2. Временные диаграммы сигналов

При использовании правила модуляции ($00 \rightarrow 0$, $10 \rightarrow \pi/2$, $11 \rightarrow \pi$ и $01 \rightarrow 3\pi/2$) на выходе модулятора будет четырехфазный сигнал, причем каждому значению фазы соответствует пара символов на выходе кодера.

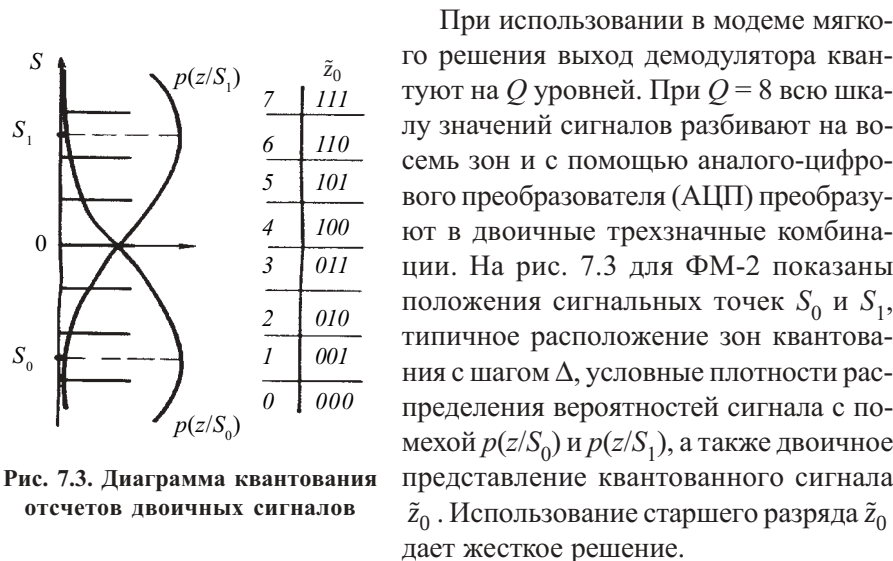


Рис. 7.3. Диаграмма квантования отсчетов двоичных сигналов

Границы принимаемых ФМ сигналов определяются импульсами тактовой синхронизации (ТС), а границы ветвей сверточного кода – импульсами ветвевой синхронизации (ВС) (рис. 7.2). При выборе способа согласования кодера и модема необходимо учитывать неоднозначность фазы, возникающую при восстановлении несущей в демодуляторе.

Схемы кодеров сверточных кодов достаточно просты. Основное внимание уделяется вопросам реализации декодеров и устройств, обеспечивающих их работу.

Применение в кодерах быстродействующих СБИС открыло путь к использованию мощных сверточных кодов при обработке сигналов в реальном масштабе времени.

8. МЕТОДЫ ДЕКОДИРОВАНИЯ СВЕРТОЧНЫХ КОДОВ

Развитие теории сверточных кодов происходило в трех направлениях в соответствии с тремя важнейшими методами декодирования сверточных кодов: метода порогового декодирования [30], метода последовательного декодирования [24, 31] и метода декодирования по максимуму правдоподобия (алгоритм Витерби) [17].

Метод порогового декодирования сверточных кодов в принципе аналогичен методу мажоритарного декодирования блочных кодов [5]. Достоинством этого метода является простота алгоритма, а, следовательно, и реализующих его устройств. Число операций, необходимых для декодирования одного информационного символа, для этого алгоритма не превосходит некоторой постоянной величины.

Метод последовательного декодирования является методом вероятностного декодирования, при котором число операций, необходимых для декодирования одного символа, является случайной величиной. При практически приемлемой сложности устройств метод последовательного декодирования по своим характеристикам приближается к методу декодирования по максимуму правдоподобия.

Метод декодирования по максимуму правдоподобия теоретически более эффективен, чем метод порогового декодирования, однако сложность устройств, необходимых для его реализации, возрастает экспоненциально с ростом длины кода.

8.1. Метод порогового декодирования

При пороговом декодировании сверточных кодов вычисляются синдромы (признаки места ошибочных символов), затем эти синдромы или последовательности, полученные посредством линейного преобразования синдромов, подаются на входы порогового элемента, где путем “голосования” (мажоритарный метод) и сравнения его результатов с порогом выносятся решение о значении декодируемого символа. Основное достоинство этого метода декодирования – простота реализации. Однако он не полностью реализует потенциальные корректирующие способности сверточного кода.

Кроме того, не все сверточные коды могут быть декодированы этим методом. Чтобы сверточный код допускал декодирование пороговым методом, он должен обладать свойством ортогональности.

В принципе метод порогового декодирования сверточных кодов аналогичен методу мажоритарного декодирования циклических блочных кодов [5].

Общая схема декодера для сверточного кода ($R = 1/2$) представлена на рис. 8.1.

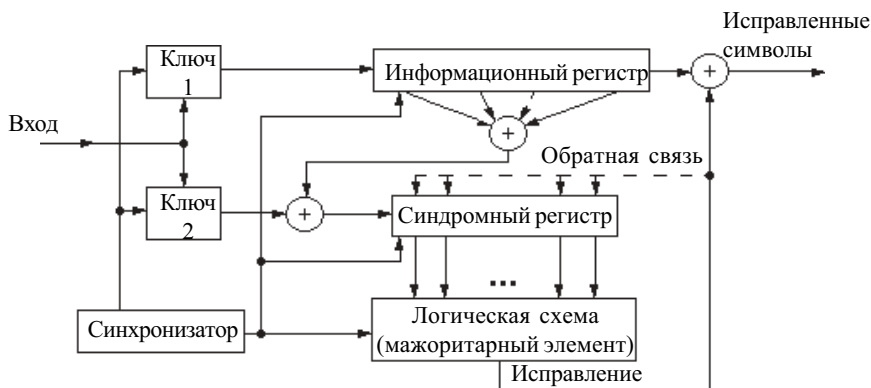


Рис. 8.1. Общая схема декодера для сверточного кода

Пороговое декодирование, как правило, применяется для систематических кодов. Декодер содержит аналог кодера, в котором по принимаемым информационным символам в сдвигающем регистре формируется копия проверочной последовательности. С этой целью синхронизатор декодера с помощью ключей 1 и 2 “расфасовывает” входную последовательность символов на 2 потока – информационный и проверочный, синхронизатор управляет работой всего декодера.

В формирователе синдрома (сумматоре по модулю 2) образуется последовательность синдромов S , которая поступает на вход синдромного регистра. В отсутствие в канале ошибок последовательности на входах формирователя синдрома всегда совпадают, и синдромная последовательность состоит из одних нулей. Различным наборам ошибок соответствуют определенные конфигурации синдромных последовательностей, в которых на определенных позициях появляются единичные символы. Закон формирования проверочных символов выбирается таким образом, чтобы по структуре синдромной последовательности можно было определить искаженные символы.

Логическая схема определяет по синдрому правильность записанного в информационном регистре блока информационных символов. Если имеется комбинация ошибок, которая может быть исправлена, то логическая схема исправляет ошибки в этом блоке путем подачи единичных символов на выходную схему суммирования по модулю 2 в моменты выхода из информационного регистра искаженных символов.

Ошибки, исправляемые в очередном информационном блоке, могут влиять на символы синдромов, соответствующих последующим блокам, поскольку сверточные коды непрерывны. Для того чтобы декодер смог полностью реализовать свои корректирующие возможности, следует исключить влияние этих ошибок. Это может быть достигнуто за счет обратной связи, которая на схеме (рис. 8.1) представлена пунктирной линией.

Обратная связь преобразует синдромный регистр прямого действия в нелинейный регистр сдвига с обратной связью. Это может привести к явлению, называемому *размножением ошибок*. Неисправимые ошибки в канале могут вызвать переход синдромного регистра в такое состояние, что и при отсутствии аддитивных ошибок в канале декодер будет продолжать всегда декодировать неправильно. Причина этого состоит в том, что выход нелинейного регистра сдвига с обратной связью, когда на его вход поступает нулевая последовательность, а начальное состояние – ненулевое, может быть периодическим.

Известно несколько способов ограничения эффекта размножения ошибок. Возможен анализ числа ненулевых синдромов на длине кодового ограничения: когда число ошибок превышает корректирующую способность кода, анализатор отключает цепи коррекции синдрома. Другой способ основан на использовании *самоортогональных кодов*, в которых распространение ошибок минимально. С этой же целью применяют *дефинитное декодирование*, при котором обратная связь на синдромный регистр вообще не используется. Очевидно, что без обратной связи не будет размножения ошибок.

При пороговом декодировании используют, в основном, систематические коды. При скорости кода, отличной от $1/2$, структура порогового декодера не отличается от описанной нами. Для практической реализации чаще всего выбирают самоортогональные коды, порождающие многочлены которых определяются на основе теории совершенных разностных множеств. Подробные таблицы самоортогональных кодов приведены в [14].

Методы порогового декодирования основаны на простых идеях, и они находят широкое применение на практике.

8.2. Метод последовательного декодирования

Если простота алгоритмов порогового декодирования и, следовательно, простота их реализации стимулировали исследования сравнительно коротких кодов и повышение их корректирующих способностей, то проводившиеся параллельно исследования методов последовательного декодирования имели одной из своих главных целей практическую реализацию теоремы Шеннона для канала с шумами [8].

При последовательном декодировании число операций, которое должен выполнить декодер, для того чтобы декодировать один символ, изменяется в зависимости от уровня шумов в канале. Число операций при последовательном декодировании является функцией скорости передачи и шумов в канале. При всех скоростях передачи, меньших определенной скорости, число операций при декодировании оказывается небольшим. Последовательный декодер строится по логической схеме, позволяющей проводить вычисления со средней скоростью, в несколько раз большей скорости передачи символов, и включает в свой состав буферное запоминающее устройство, предназначенное для хранения поступающих данных при повышении уровня шумов в канале. В случае если число возникших в канале ошибок превысит корректирующую способность кода, при последовательном декодировании и других методах декодирования могут возникать ошибки. Однако при последовательном декодировании ошибки могут возникать также и из-за того, что оказываются заполненными все ячейки памяти буферного запоминающего устройства, т. е. из-за переполнения буфера. Если вероятность ошибки из-за переполнения буфера оказывается значительно больше, чем вероятность возникновения ошибки из-за невозможности исправить или обнаружить ошибку, то именно она и будет определять характеристики алгоритма последовательного декодирования.

При рассмотрении алгоритмов последовательного декодирования удобно представлять сверточный код в виде кодового дерева, которое для кодера (рис. 2.5, б) приведено на рис. 8.2.

Кодовое дерево строится следующим образом. Исходному нулевому состоянию сдвигающего регистра кодера соответствует начальный узел дерева. Если входной информационный символ, поступающий в регистр,

равен 1, то ему приписывается линия (ребро дерева), идущая, как принято на этом рисунке, вниз, а если информационный символ равен 0, – то вверх. Тем самым получаем два новых узла, соответствующие следующему такту работы кодера, для каждого из которых дерево строится далее аналогичным образом, и т.д. Над каждым ребром дерева записываются n кодовых символов, получаемых при этом на выходе кодера. Совокупность нескольких последовательных ребер, соединяющих какие-либо два узла, составляет ветвь дерева. Узлы, соединенные одним ребром, называются соседними. Для двоичных кодов в общем случае из каждого узла дерева будет исходить 2^k ребер, каждому из которых сопоставлено n двоичных символов.

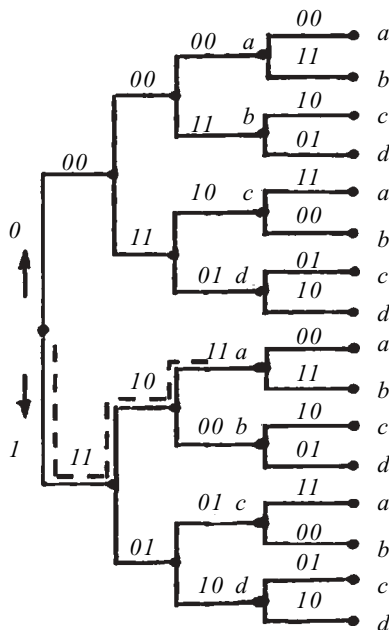


Рис.8.2. Кодовое дерево кодера (рис. 2.5, б)

Коды, допускающие подобное представление с помощью кодового дерева, называются древовидными. Таким образом, сверточные коды относятся к древовидным.

Можно заметить, что рассмотренную ранее решеточную диаграмму кода можно получить из кодового дерева, если объединить те узлы дерева, после которых над соответствующими ребрами оказываются одинаковые кодовые символы. Эти узлы помечены на рис. 8.2 одинаковыми буквами a, b, c, d и соответствуют состояниям кодера на диаграмме состояний (рис. 4.1).

Каждая последовательность кодируемых информационных символов порождает определенный путь по кодовому дереву. Например, информационная последовательность 1001.... порождает путь по кодовому дереву, показанный штриховой линией на рис. 8.2, которому соответствует последовательность кодовых символов 11101111... Очевидно, задача декодера заключается в отыскании истинного (правильного) пути, т. е. того пути, который в действительности был порожден кодером.

Таким образом, при алгоритмах последовательного декодирования декодер определяет наиболее правдоподобный путь по дереву, что позволяет исключить из анализа большую часть остальных путей, имеющих меньшее правдоподобие.

Предположим, что задан некоторый сверточный код, и попытаемся по принятой последовательности определить передававшуюся последовательность. Для этого сначала необходимо передвигаться поочередно вдоль каждого пути кодового дерева, сравнивая принятую последовательность с последовательностью, соответствующей пути, по которому происходит движение. Если при этом удастся обнаружить некоторый путь, которому соответствует последовательность, почти совпадающая с принятой последовательностью, то естественно считать, что эта последовательность и передавалась. Метод последовательного декодирования, предложенный Возенкрафтом [31], предусматривает вначале поиск на кодовом дереве пути из m -го ребра (m – число разрядов регистра сдвига кодера), соответствующая которому кодовая последовательность находится на каком-то определенном расстоянии от переданной последовательности, и далее декодирование первого информационного символа. Первым переданным информационным символом считается первый информационный символ первого ребра найденного пути. Поскольку далее декодирование второго и последующих информационных символов осуществляется точно так же, как и первого информационного символа, то этот алгоритм и был назван алгоритмом последовательного декодирования.

Одним из критериев близости пути на кодовом дереве к принятой последовательности является расстояние Хемминга между ними. Если расстояние Хемминга между последовательностями с длиной n_0 , равной числу обработанных двоичных символов, не превышает определенный порог D_0 , то последовательности можно считать совпадающими. В случае двоичного симметричного канала порог D_0 выбирается таким образом, чтобы вероятность того, что расстояние Хемминга между передававшейся и принятой последовательностями было больше чем D_0 , не превышало величины 2^{-L} , где L – некоторая заданная постоянная. При этом вероятность того, что расстояние Хемминга между правильным путем дерева и принятой последовательностью больше чем D_0 , при больших L оказывается чрезвычайно малой.

Предположим, что этот метод декодирования используется в двоичном симметричном канале с вероятностью перехода p . Напомним, что в

двоичном симметричном канале без памяти каждый переданный кодовый символ может быть принят ошибочно с вероятностью p и правильно с вероятностью $q = 1-p$, причем в случае ошибки вместо переданного символа 0 на приемной стороне воспроизводится символ 1, или – наоборот. Для канала без памяти вероятность ошибочного приема символа не зависит от того, какие символы переданы до него и как они были приняты. На рис. 8.3 показаны вероятности перехода для двоичного симметричного канала.

Поскольку в канале ошибки возникают независимо с вероятностью перехода p , то расстояние Хемминга d , вычисляемое вдоль правильного пути, как показано на рис. 8.4, почти всегда будет возрастать со скоростью p с ростом числа обработанных символов n_0 . В то же время расстояние Хемминга, вычисляемое вдоль любого другого пути кодового дерева, будет возрастать со скоростью $1/2$, поскольку различные пути кодового дерева отличаются друг от друга приблизительно в половине символов. Таким образом, расстояния Хемминга, вычисленные вдоль правильного и ошибочного пути кодового дерева, оказываются различными. При этом, даже если на начальном этапе декодирования путь кодового дерева был выбран неправильно, то через некоторое время ошибка будет обнаружена, поиск правильного пути будет продолжен и в конце концов правильный путь будет найден.

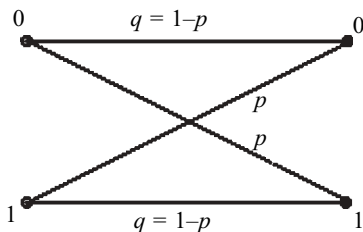


Рис. 8.3. Вероятности переходов для двоичного симметричного канала

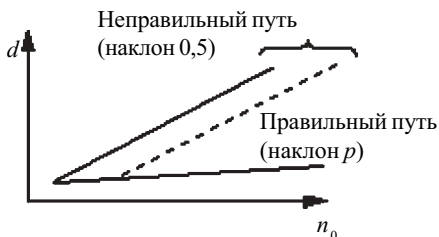


Рис. 8.4. Зависимость расстояния Хемминга d от числа декодированных символов – n_0

Этот алгоритм последовательного декодирования был обобщен Рейффеном [31] на случай произвольного дискретного канала без памяти. Обобщение было сделано путем замены расстояния Хемминга на другую функцию (“перекошенное” расстояние [19]), являющуюся обобщением расстояния Хемминга, которая была названа *ценой пути*.

Другим алгоритмом последовательного декодирования является алгоритм, предложенный Фано [32], который предусматривает вычисление и использование при декодировании наряду с ценой пути также некоторых порогов. Этот алгоритм является алгоритмом того же типа, что и алгоритм Возенкрафта, но для него среднее число операций, необходимых для декодирования одного символа, не зависит от длины кодового ограничения. Это упрощает анализ алгоритма Фано. Кроме того, достоинством алгоритма Фано является также то, что он допускает более простую техническую реализацию. Поскольку сложность декодера Фано не зависит от длины кодового ограничения, то это позволяет использовать его при больших ДКО, т. е. при малых вероятностях ошибки на декодированный символ. Подробное исследование количественных характеристик алгоритма Фано содержится в [8, 14–15].

Еще одним известным алгоритмом последовательного декодирования является алгоритм Зигангирова, предложенный в 1966 г., являющийся алгоритмом декодирования по максимуму правдоподобия.

Этот алгоритм иногда называют стэк-алгоритмом (от английского *stack* – кipa, стопа), так как для его реализации требуется память большой емкости, содержимое которой постоянно упорядочивается определенным образом, т. е. требуется “кipa памяти”.

В качестве метрики (расстояний Хемминга) путей по кодовому дереву при этом алгоритме также используется функция правдоподобия пути (цена пути), но с добавлением элемента смещения, величина которого определяется исходя из скорости используемого кода и пропускной способности канала, путем оптимизации вероятности ошибки декодирования и среднего числа операций при декодировании одного информационного символа [24].

Количество вычислений на один декодированный информационный символ, которые производит декодер при данном алгоритме декодирования, является случайной величиной, что характерно для всех алгоритмов последовательного декодирования, так как все они являются вероятностными. Количество производимых декодером вычислений тем меньше, чем ниже уровень шума в канале, т. е. чем меньше вероятность ошибок при приеме кодовых символов в первой решающей схеме приемника. Для компенсации этой неравномерности вычислений при всех алгоритмах последовательного декодирования требуется включение буферного запоминающего устройства (ЗУ) между собственно последовательным декодером и первой решающей схемой приемника. Однако в

том случае, когда объем производимых вычислений оказывается очень большим (“всплеск вычислений”, соответствующий “плохому” состоянию канала связи), возможно переполнение буферного ЗУ, что приводит к отказу в декодировании. Это является недостатком всех методов последовательного декодирования. Однако сложность этих методов либо слабо, либо совсем не зависит от длины кодовых ограничений, что позволяет использовать большую длину кодовых ограничений, а следовательно, достигнуть большой эффективности корректирующего сверточного кода.

Для борьбы с отказами в декодировании при методах последовательного декодирования, кроме очевидных способов увеличения объема буферного ЗУ и быстродействия декодера, можно через определенное заранее число информационных символов, соответствующее с заданной вероятностью времени наступления отказа в декодировании, передавать последовательность заранее известных символов, например последовательность нулей. Тем самым декодер устанавливается на заведомо правильный путь по дереву. В случае наличия обратного канала возможна организация переспроса при наступлении отказа в декодировании.

Из-за присущей сверточным кодам непрерывности в обработке информации их синхронизация при декодировании осуществляется гораздо проще, чем при блочном кодировании. В частности, не требуется синхронизации по кодовым словам, без которой правильное декодирование блоковых кодов, как правило, невозможно. Однако для всех методов декодирования сверточных кодов необходима надежная синхронизация по узлам кодового дерева (узловая синхронизация), т. е. синхронизация по группам символов, соответствующих одному циклу опроса коммутатора кодера.

Структурная схема декодера Зигангирова будет выглядеть аналогично схеме декодера Витерби (рис. 8.9) с той разницей, что на входе декодера включено буферное ЗУ, куда поступают символы с первой решающей схемы приемника.

8.3. Метод декодирования по алгоритму Витерби

Метод представляет собой декодирование по максимуму правдоподобия. Идея алгоритма Витерби состоит в том, что в декодере воспроизводят все возможные пути последовательных изменений состояний сигнала, сопоставляя получаемые при этом кодовые символы с приня-

тыми аналогами по каналу связи и на основе анализа ошибок между принятыми и требуемыми символами определяют оптимальный путь (оптимальной считается та последовательность, расстояние Хемминга которой от принятой последовательности минимально).

Декодирование по методу Витерби, по существу, представляет собой алгоритм поиска наивыгоднейшего, максимально правдоподобного пути на графе – решеточной диаграмме кода.

8.3.1. Декодирование в случае отсутствия ошибок при приеме

Продemonстрируем работу алгоритма на примере несистематического сверточного кода с маркировкой (2, 1, 3) с порождающими полиномами, задаваемыми выражением (3.6), кодер которого показан на рис. 2.5, б. Возьмем следующую последовательность символов 10110 и будем полагать, что все символы приняты без ошибок.

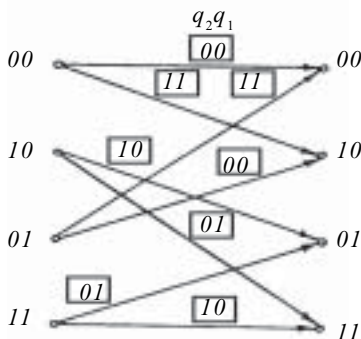


Рис. 8.5. Диаграмма переходов состояний сигнала

Рассмотрим графическое отображение изменений состояния сигнала (рис. 8.5), которое полностью соответствует диаграмме (рис. 4.1). Начальным всегда является состояние 00. В соответствии с диаграммой переходов состояний сигнала (рис. 8.5) в первый тактовый момент возможны два перехода: $00 \rightarrow 00$ и $00 \rightarrow 10$. Первому переходу соответствует на выходе кодера кодовая комбинация 00, второму – 11.

При приходе на вход кодера первой информационной 1 первой ветви $00 \rightarrow 00$ будут соответствовать две ошибки в приеме, а второй ветви $00 \rightarrow 10$ – нуль ошибок. Ошибка по каждой ветви служит метрикой d_H в смысле расстояния Хемминга, т. е. соответствует числу отличающихся от требуемых принятых символов. Эти метрики зафиксированы в узлах диаграммы (рис. 8.6).

В момент времени 2 (второй такт) сигнал может принять 4 состояния, которые определяются двумя возможными переходами из 00 и двумя переходами из 10. Сравнение с принятыми символами 01 дает следующие метрики соответствующих ветвей d_H : $00 \rightarrow 00 d_H = 1$, $00 \rightarrow 10 d_H = 1$, $10 \rightarrow 01 d_H = 0$, $10 \rightarrow 11 d_H = 2$.

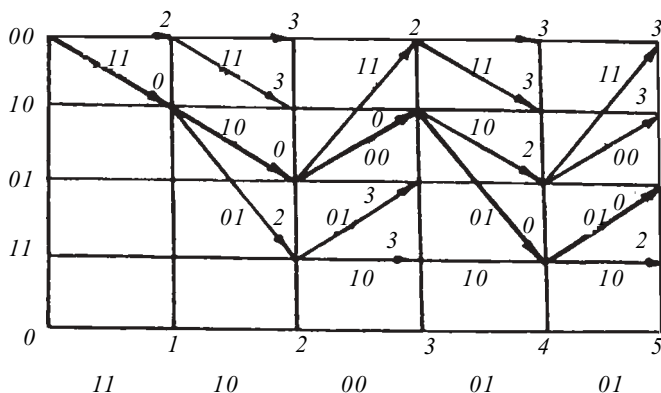


Рис. 8.6. Декодирование на основе алгоритма Витерби в отсутствие ошибок при приеме

Суммарная метрика $d_{\Sigma H}$ по каждому из возможных путей определяется как сумма метрик составляющих его ветвей. Значения суммарных метрик показаны на диаграмме рис. 8.6. Для момента времени 3 следует анализировать уже 8 возможных путей и сравнивать 8 соответствующих им метрик $d_{\Sigma H}$.

Алгоритм Витерби выбирает путь с наименьшей суммарной метрикой (“штрафом”) и может отбрасывать по ходу продвижения во времени те пути, которые имеют штраф, превышающий минимальный “штраф” в данный момент времени на некоторую пороговую величину. Для упрощения рис. 8.6 установлен пороговый уровень $d_{\Sigma H} = 3$ и не показываются ветви с большим значением $d_{\Sigma H}$. Оптимальным путем является путь с наименьшим “штрафом” (при отсутствии ошибок в канале передачи символов $d_{\Sigma H} = 0$). Последовательность бит на выходе декодера, соответствующая этому пути, отмеченному на рис. 8.6 утолщением ветвей, совпадает с передаваемым информационным сигналом.

8.3.2. Декодирование в случае наличия ошибок при приеме

Рассмотрим теперь случай, когда в канале связи возникли ошибки, например, при передаче первого и третьего информационного битов (на рис. 8.7 они обведены).

Теперь метрики первой и третьей ветвей оптимального пути не равны нулю и суммарная метрика оптимального пути $d_{\Sigma H} = 2$. Однако как следует из рис. 8.7, оптимальный путь с наименьшей суммарной метрикой восстанавливает передачу последовательности информационных

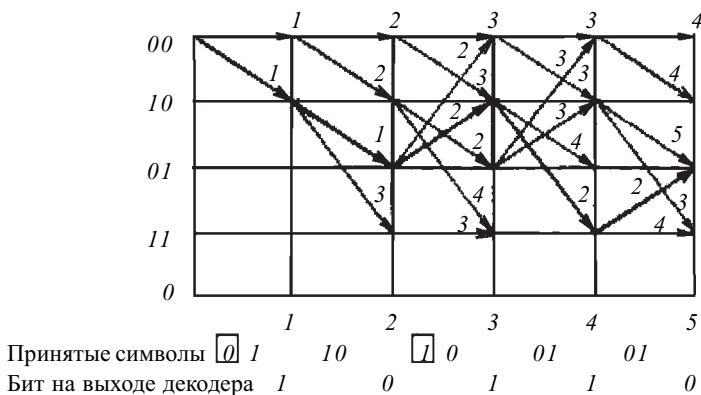


Рис. 8.7. Декодирование на основе алгоритма Витерби в случае приема с ошибками

бит, т. е. использованный код исправляет ошибки. Разумеется, алгоритм Витерби не во всех случаях дает верный результат на выходе декодера. На рис. 8.8 показан пример кода (2, 1, 3) для информационной последовательности, состоящей из всех нулей. При этом все символы на выходе кодера равны 0. Однако ошибки при приеме первых трех символов приводят к ошибке первого бита на выходе декодера.

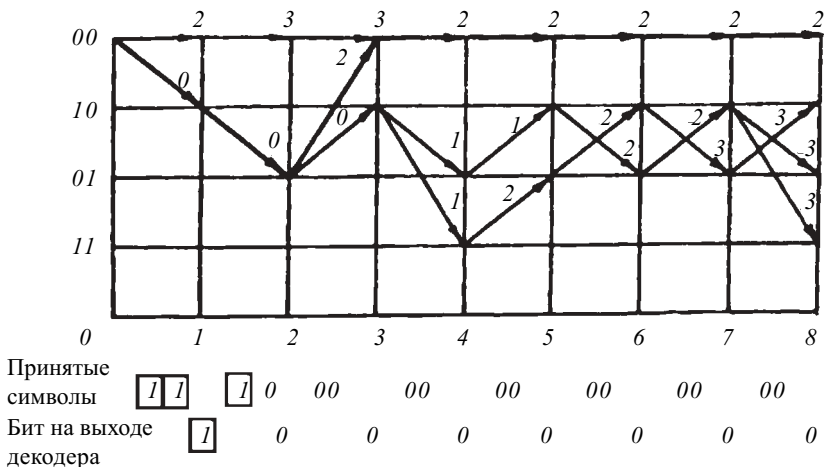


Рис. 8.8. Декодирование на основе алгоритма Витерби нулевой последовательности при наличии ошибок на выходе

Рассмотренный метод декодирования, когда в качестве метрики используют расстояние Хэмминга, называют декодированием с жестким решением. Здесь каждому символу на выходе демодулятора соответствует одно из двух значений: 0 или 1. Лучшие результаты в смысле восстановления исходного сигнала дает декодирование с мягким решением. В этом случае каждый символ на выходе демодулятора подвергают квантованию (см. разд. 7). Для оптимизации приема сигнала в канале с гауссовым шумом достаточно использовать 8-уровневые квантователи. При этом, например, уровню передачи логической единицы, будут соответствовать $q_i = 1 - 4$, а уровню передачи логического нуля $q_i = (-1) - (-4)$ (возможно и наоборот). Величина квантованного уровня определяет доверительную вероятность полученного результата. Более высоким уровням (3; 4) соответствует более высокая доверительная вероятность. При этом метрику отдельных ветвей вычисляют, суммируя совпадающие значения q_i и вычитая несовпадающие.

При декодировании с мягким решением в качестве оптимального пути выбирают путь с максимальной суммарной метрикой, что соответствует максимальной накопленной доверительной вероятности. Декодирование с мягким решением даже при трехкратной ошибке в приеме символов (рис. 8.8) может обеспечить оптимальный результат без ошибок на выходе декодера, что соответствует большей помехозащищенности канала связи, чем при использовании декодирования с жестким решением [27].

8.3.3. Схемное построение декодера Витерби

Обобщенная структурная схема декодера, работающего по алгоритму Витерби, показана на рис. 8.9. Для каждого такта работы, соответствующего приему кодовых символов, полученных за один цикл опроса коммутатора кодера, вычислитель метрики ребер (ВМР) вычисляет правдоподобие ребер, сливающихся в каждом узле. Например, в случае двоичного симметричного канала с жесткими решениями он вычисляет Хеммингово расстояние между каждым из путей, сливающихся в любом узле, и соответствующей последовательностью принимаемых кодовых символов, поступивших с выхода первой решающей схемы приемника, выносящей жесткие решения о значении каждого принимаемого кодового символа.

Вычислитель метрики путей (ВМП), для каждого из путей, выживших на предыдущем такте декодирования и хранимых в ЗУ путей, осу-

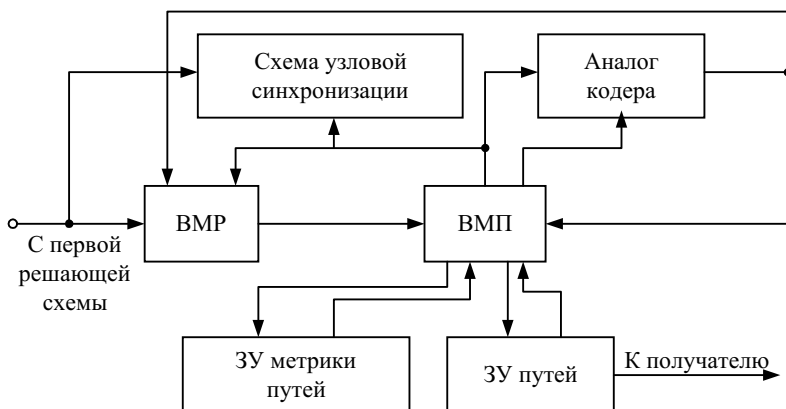


Рис. 8.9. Структурная схема декодера Витерби

ществляет следующие операции: вводит каждый из этих путей в аналог кодера, где генерируются 2^q его возможных продолжений; вычисляет правдоподобие каждого из этих продолжений, для чего суммирует метрики выживших путей, хранимые в ЗУ метрики путей, с новыми вычисленными значениями метрик ребер; заносит вновь полученные пути в ЗУ путей, а их метрики — в ЗУ метрики путей, а затем отбирает из них пути, каждый из которых максимально правдоподобен для одного из узлов и сохраняет его в ЗУ путей, а его метрику — в ЗУ метрики путей. Затем эта же последовательность операций повторяется для следующего такта работы и т. д.

Важным достоинством декодера Витерби является то, что когда в результате воздействия шумов в канале связи или по другим причинам при декодировании сделана ошибка в выборе пути на решетчатой диаграмме кода, т. е. выбран неправильный путь, то за несколько тактов, в течение которых могут происходить ошибки при декодировании, декодер вновь выходит на правильный путь (рис. 8.10). Это позволяет на-

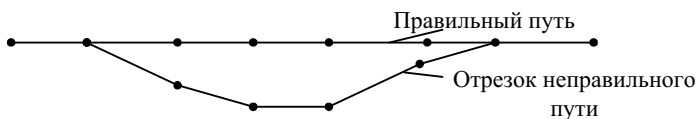


Рис. 8.10. Случай выбора декодером Витерби отрезка неправильного пути

чать процесс декодирования с любого момента времени, не заботясь о взаимной синхронизации кодера и декодера по началу работы. Действительно, отсутствие такой синхронизации эквивалентно нахождению декодера на неправильном пути по решетчатой диаграмме кода. В силу этого свойства декодер, начав декодирование в произвольный момент, через несколько тактов сам выйдет на правильный путь, автоматически установив указанную синхронизацию.

Однако для всех методов декодирования сверточных кодов, как уже указывалось, необходима надежная синхронизация по узлам кодового дерева (узловая синхронизация), т. е. синхронизация по группам символов, соответствующих одному циклу опроса коммутатора кодера. Для этого можно использовать тот факт, что функция правдоподобия лучшего пути по дереву при правильной узловой синхронизации растет существенно быстрее, чем при ее отсутствии.

Эффективность сверточных кодов зависит от выбранного кода и характеристики канала передачи. Итоговую оценку производят по величине коэффициента ошибок при заданном отношении сигнал/шум в канале и типе канала. Поскольку сверточные коды хорошо исправляют одиночные ошибки, но чувствительны к пакетам ошибок, то их обычно применяют вместе с перемежением передаваемых по каналу символов, как это делается в сотовых системах подвижной связи.

Интенсивно применяются сверточные коды и в спутниковых системах связи.

9. ПРИМЕНЕНИЕ СВЕРТОЧНЫХ КОДОВ В СОТОВЫХ И СПУТНИКОВЫХ СИСТЕМАХ СВЯЗИ

Сверточные коды нашли широкое применение в сотовых и в спутниковых системах связи.

Системы сотовой подвижной связи (ССПС) впервые стали эксплуатироваться в конце 70-х – начале 80-х годов в Скандинавских странах (NMT-450) и США (AMPS). Сотовый принцип топологии сети с повторным использованием частот в сотах во многом решил проблему дефицита частотного ресурса и в настоящее время является основным в создаваемых системах подвижной связи общего пользования. Стандартизация в области ССПС привела к тому, что на смену девяти отдельным аналоговым стандартам сотовой связи первого поколения пришли три цифровых стандарта второго поколения (GSM, D-AMPS, JDC), один из них – GSM признан “глобальным”.

В 1982 г. Европейская Конференция Администраций Почт и Электросвязи, объединяющая администрации связи 26 европейских стран, создала специальную группу Group Special Mobile для разработки единого стандарта цифровой сотовой связи в диапазоне 900 МГц. Аббревиатура GSM и дала название новому стандарту. Позднее, в связи с широким распространением этого стандарта во всем мире, GSM стали расшифровывать как Global System for Mobile Communications (глобальная система для мобильной связи). Результатом работы этой группы стали опубликованные в 1990г. требования к системе сотовой связи стандарта GSM, в котором используются самые современные разработки ведущих научно-технических центров. К ним, в частности, относятся временное разделение каналов, шифрование сообщений и защита данных абонента, использование блочного и сверточного кодирования с перемежением и т. п.

Каналы связи в стандарте GSM разделяются на физические и логические. Физический канал образуется путем комбинирования временного (ВРК) и частотного (ЧРК) разделения сигналов.

До формирования физического канала сообщения данные, представленные в цифровом виде, группируются и объединяются в логические каналы двух типов:

- канал связи – для передачи кодированной речи и данных;
- канал управления – для передачи сигналов управления и синхронизации.

Кодирование и перемежение являются важными ступенями тракта обработки информационных цифровых сигналов и сигналов управления. В цифровых ССПС осуществляется преобразование аналогового речевого сигнала в цифровую последовательность, которая подвергается шифрованию и кодированию, что необходимо для защиты информации от ошибок в процессе передачи и приема. Для этого используются:

- блочное кодирование – для быстрого обнаружения ошибок при приеме;
- сверточное кодирование – для исправления одиночных ошибок;
- перемежение – для преобразования пакета ошибок в одиночные ошибки.

В цифровых ССПС кодируются все передаваемые по радиоканалу сигналы. В аналоговых ССПС кодируют цифровые сигналы управления.

При кодировании преследуют различные цели. Самый низкий уровень имеет выявление (обнаружение) ошибок в полностью принятом сигнале. По сравнению с ним более высоким уровнем обладает обнаружение ошибок в отдельных сегментах сигнала, которое может быть выполнено с помощью простых блоковых кодов, например, с проверкой на четность. В современных системах используют коды с исправлением ошибок. Это могут быть блоковые коды (каналы сигнализации в NMT-450, DECT) и сверточные коды (GSM, системы с кодовым разделением – CDMA). Выбор кода определяет большое число факторов: характеристики каналов, скорость передачи, вид модуляции и т. п. Важное значение приобретает элементарно-технологическая база. Применение быстродействующих процессорных СБИС открыло путь к использованию мощных сверточных кодов при обработке сигналов в реальном времени. Сверточные коды хорошо исправляют случайные одиночные ошибки, но дают плохие результаты при пакетах ошибок. Поэтому сверточное кодирование и совмещают с перемежением (перетасовкой) информационных символов, которое обеспечивает преобразование пакетов ошибок в одиночные.

В ССПС основные свойства речевых каналов и каналов управления значительно отличаются друг от друга. Для речевых каналов необходима связь в реальном масштабе времени с короткими задержками при сравнительно низких требованиях к вероятности ошибки в канале. Для канала управления требуется абсолютная достоверность данных и исправление ошибок, но допускается более длительное время передачи и задержки.

В различных логических каналах используются различные сверточные коды, поскольку скорости передачи и требования по защите от ошибок также различны. Для упрощения процедур кодирования и декодирования при формировании кодов используются только несколько полиномов. Это позволяет использовать в стандарте GSM сверточный код с одной скоростью $R = 1/2$. В ряде режимов для выравнивания скорости в речевом канале до $R = 1/2$ применяют прореживание, т. е. периодический пропуск (перфорацию) кодированных символов. Поскольку сложность декодирования по наиболее выгодному, с точки зрения реализации, алгоритму Витерби возрастает экспоненциально с увеличением длины кодового ограничения l , то типовые значения ДКО малы и лежат в интервале $l = 3 - 10$.

В стандарте GSM в речевом канале, в зависимости от скорости передачи сообщения, применяются следующие пары порождающих сверточный код многочленов (при $R = 1/2$):

$$G_1(X) = 1 + X^3 + X^4; \quad G_2(X) = 1 + X^2 + X^3 + X^4; \quad G_3(X) = 1 + X + X^4.$$

$$G'_1(X) = 1 + X + X^2 + X^4; \quad G'_2(X) = 1 + X + X^4; \quad G'_3(X) = 1 + X + X^3 + X^4.$$

В каналах управления цифровая последовательность подвергается кодированию сверточным кодом, который задается полиномами первой пары – $G_1(X)$ и $G'_1(X)$.

На следующей ступени развития ССПС можно ожидать перехода к каскадному кодированию, например, внутреннему сверточному кодированию и внешнему, устраняющему ошибки при декодировании сверточных кодов, блоковому, с исправлением пакетов ошибок. Разумеется, такое сложное кодирование требует большой избыточности, что формально снижает эффективность использования систем, но существенно повышает достоверность информации [27].

Сверточные коды и рассмотренные алгоритмы декодирования (в первую очередь – по максимуму правдоподобия, алгоритм Витерби) находят основное применение в системах космической и спутниковой свя-

зи. Это объясняется тем, что каналы связи в этих системах близки по своим свойствам к каналам с белым гауссовским шумом, которые являются симметричными каналами без памяти. Для подобных систем характерны жесткие ограничения по мощности передаваемого сигнала, поэтому для них важно осуществить наиболее эффективное кодирование и декодирование, позволяющее уменьшить вероятность ошибки на декодированный информационный символ при малом энергетическом потенциале [20].

Для существенного улучшения помехоустойчивости при использовании сверточных кодов необходимо увеличивать скорость передачи символов, а следовательно, и ширину полосы, например, в 2 раза при относительной скорости передачи кода $1/2$ или в $4/3$ раза при относительной скорости $3/4$. Таким образом, применение сверточных кодов оказывается особенно выгодным в спутниковых системах связи, энергетический потенциал которых ограничивается мощностью бортового ретранслятора, т. е. в каналах, где определяющим фактором является ограничение мощности, а не полосы частот [16]. В системах с ограниченной энергетикой кодирование позволяет уменьшить необходимое отношение сигнал – шум, оптимальным образом распределить мощность ретранслятора между каналами и увеличить число каналов.

Большая задержка на трассах распространения в цифровых спутниковых системах связи (ССС) не позволяет использовать для повышения верности системы с автозапросом (с обратным каналом), в которых коды служат для обнаружения ошибок. Поэтому в СССР и используются, в основном, сверточные коды, решающие задачу непосредственного исправления ошибок.

Результаты расчетов вероятности ошибки для кодов со скоростью $1/2$ и перфорированных кодов со скоростями $2/3$ и $3/4$, взятые из монографии “Цифровые методы в спутниковой связи” [23], показаны на рис. 9.1.

Там же приведены зависимости вероятности ошибки приема одного бита информации от отношения сигнал – шум E_b/N_0 при фазовом методе модуляции без применения кодирования (ФМ-2) и с применением сверточных кодов с различными порождающими многочленами, представленными в 8-ричной форме записи. По кривым можно определить энергетический выигрыш от кодирования (ЭВК). Например, при использовании сверточного кода (133,171) выигрыш составляет 5,4 дБ при вероятности ошибки 10^{-5} .

Анализ показывает [23], что применение коротких сверточных кодов, декодируемых по алгоритму Витерби с мягким решением, позволяет получить ЭВК 4 – 6 дБ. Переход к жесткому решению снижает ЭВК примерно на 2 дБ. Квантование выхода демодулятора на четыре уровня снижает ЭВК на 0,7 – 0,8 дБ, а квантование на восемь уровней – на 0,25 дБ. Обычно ограничиваются квантованием на восемь уровней, используя практически полностью возможности мягкого решения.

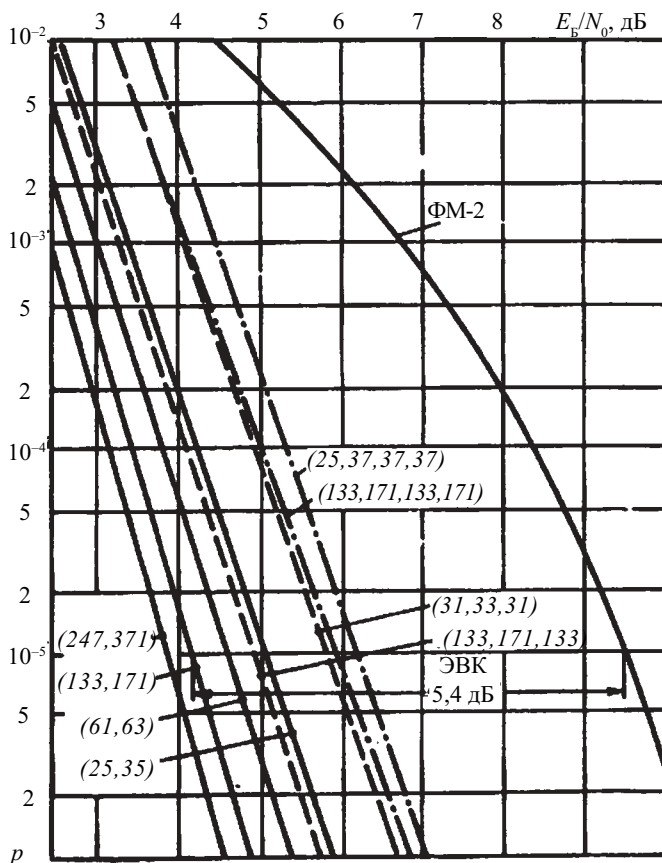


Рис. 9.1. Кривые помехоустойчивости декодирования сверточных кодов

В настоящее время известны 4 наиболее крупных зарубежных проекта глобальных систем спутниковой персональной связи – Iridium, Globalstar (на низких орбитах), Odyssey и ICO (на средних орбитах), а

также 2 проекта систем персональной связи на геостационарной орбите: Tritium-G (GEO-G) – глобальная связь с помощью трех искусственных спутников Земли (ИСЗ) и Tritium-R (GEO-R) – региональная персональная связь. В табл. 9.1 приведены данные для этих систем с относительной скоростью сверточных кодов и вероятностью ошибки при передаче данных [33].

Таблица 9.1

Наименование системы	Iridium	Globalstar	Odyssey	ICO	GEO-G	GEO-R
Скорость сверточного кода	3/4	1/3	3/4	1/3	3/5	3/4
Вероятность ошибки при передаче данных	$5 \cdot 10^{-6}$	10^{-6}	10^{-5}	10^{-5}	$5 \cdot 10^{-6}$	$5 \cdot 10^{-6}$

К сожалению, СССР Iridium прекратила функционирование, запущенные спутники сняты с рабочих орбит и уничтожены по причине банкротства системы из-за первоначальной ошибки в расчете числа пользователей, способных закупать дорогостоящие терминалы системы.

В широко известной международной СССР ИНМАРСАТ также использующей сверточные коды, в которых для передачи данных принята скорость кода $1/2$ при длине кодового ограничения 7 и порождающих полиномах [35]:

$$G_1(X) = 1 + X^2 + X^3 + X^5 + X^6,$$

$$G_2(X) = 1 + X + X^2 + X^3 + X^6.$$

В России в настоящее время разрабатывается проект многофункциональной космической телекоммуникационной системы (МКТС) “Ростелесат” [34].

Система предназначена для передачи различного вида информации в цифровой форме через ИСЗ в глобальном масштабе с любыми стандартными скоростями от 1,2 до 2048 кбит/с, а также для мониторинга земной поверхности.

В системе выбрана группировка из 24 ИСЗ с 4 орбитальными плоскостями по 6 ИСЗ в каждой при высоте круговых орбит 10360 км.

МКТС “Ростелесат” является интегральной системой и включает в себя 3 взаимосвязанные функционально законченные системы: низкоскоростную систему связи (НСС), высокоскоростную систему связи (ВСС) и систему мониторинга (СМ).

НСС обеспечивает непрерывную телефонную и другие виды цифровой связи со скоростями до 64 кбит/с (телекс, факс, данные, пейджинг и др.) между подвижными и стационарными абонентскими станциями, а также абонентами наземных систем связи, находящимися в любых точках земного шара. На линии “Земля – Космос” и на линии “Космос – Земля” используется многостанционный доступ с временным разделением на ряде частот (ВРК/ЧРК). В обоих направлениях применяется каскадное кодирование при внутреннем – сверточном коде и внешнем – блоковом коде Рида-Соломона.

ВСС обеспечивает передачу между мобильными и стационарными абонентами со скоростями до 2048 кбит/с (связь между локальными вычислительными сетями, доступ к Интернету, дистанционное обучение и другие услуги мультимедиа). На линии “вверх” применяется сверточный код со скоростью $3/4$ и длиной кодового ограничения 7; на линии “вниз” – каскадное кодирование: внутренний код – сверточный со скоростью $3/4$ и длиной кодового ограничения 7, внешний код – Рида-Соломона (126,112).

Таким образом, изучение сверточных кодов необходимо для понимания работы широкого спектра современных систем связи.

10. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ “СВЕРТОЧНЫЕ КОДЫ”

Лабораторная работа выполняется на персональном компьютере и состоит из трех частей. Выполнение каждой последующей части становится возможным лишь при правильном выполнении предыдущей. Для того чтобы приступить к работе студент должен получить у преподавателя два номера варианта и ввести их на экране монитора в ячейки под символом “№”. При этом в соседних клетках отобразятся исходные данные для выполнения работы, а именно: порождающий полином в восьмеричной форме, скорость кода и минимальное расстояние по Хеммингу, соответствующие заданным вариантам.

Варианты представлены в следующей таблице:

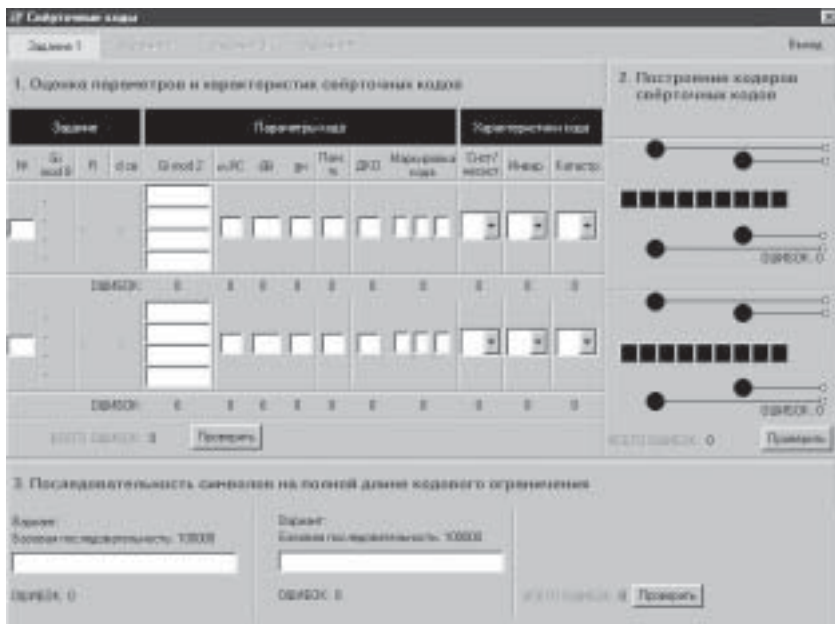
№	1	2	3	4	5	6	7	8	9	10
G_1	7	17	37	35	33	47	65	73	147	171
G_2	2	15	25	23	31	40	57	61	135	133
№	11	12	13	14	15	16	17	18	19	20
G_1	345	371	753	7	17	37	75	171	367	15
G_2	237	247	561	7	15	33	53	165	331	13
G_3	—	—	—	5	13	25	47	133	225	10

Передвижение по ячейкам первого задания осуществляется нажатием левой кнопки мыши, при наведенном на ячейку курсоре, или при нажатии клавиши “tab” на клавиатуре. Выделенная ячейка отображается мигающим в ней знаком “[]”. При наведении курсора на заголовки параметров кода и характеристик кода высвечиваются интерактивные подсказки. После заполнения ответами всех ячеек для двух заданных номеров необходимо нажать кнопку “Проверить”. На экране под каждым ответом высветится количество ошибок и суммарное их количество во всем разделе. При нулевом количестве ошибок программа автоматически перейдет к следующему заданию. Если ошибки были, необходимо их исправить и повторно нажать кнопку “Проверить”. Количе-

ство ошибок, допущенных при ответах, суммируется с каждой последующей попыткой и отображается в графе “всего ошибок”.

Целью выполнения первого задания является правильное заполнение ячеек соответствующими параметрами и характеристиками кодов, опираясь на материалы данного пособия. Обратим внимание на то, что значение коэффициента избыточности кода проставляется в десятичной форме с округлением второго знака после точки.

Во втором задании “Построение кодеров сверточных кодов” необходимо построить схему кодера, в соответствии с порождающими полиномами кодов. Связи в кодере задаются последовательным нажатием левой кнопки мыши на необходимом сумматоре по модулю 2 (кружок) и той ячейки регистра сдвига, с которой надо его связать. После задания всех связей нажимается кнопка “Проверить”. При устранении неверно поставленной связи процедура ее стирания аналогична (кружок сумматора – ячейка регистра сдвига). Коммутатор на выходе кодера не показан. Номера контактов коммутатора соответствуют номерам сумматоров, высвечиваемым на экране монитора при установке курсора на кружок сумматора.



В третьем задании надо определить последовательность символов на полной длине кодового ограничения при заданной базовой последовательности 100000 для построенных кодеров из второго задания.

После выполнения всех трех заданий нужно показать преподавателю экран выполненной работы для ее оценки.

На рисунке показан экран монитора при выполнении лабораторной работы для четырех порождающих полиномов, формирующих сверточный код.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Построение рекуррентного кода Финка, его характеристики и возможности.
2. Кодирование устройства кода Финка при различных значениях шага кода.
3. Примеры кодеров систематического и несистематического кодов.
4. Схемное построение кодеров для скорости $1/2$, $1/3$ и $2/3$.
5. Представление сверточных кодов с помощью многочленов.
6. Обоснование названия “сверточные коды.”
7. Диаграмма состояний кодера сверточного кода (СК).
8. Решетчатая диаграмма кодера СК.
9. Определение характеристик СК: скорость, избыточность, память и длина кодового ограничения.
10. Маркировка, вес и кодовое расстояние СК.
11. Минимальное кодовое расстояние, корректирующая способность СК.
12. Мощность и энергетический выигрыш кода.
13. Определение систематических и несистематических кодов.
14. Характеристика катастрофических СК.
15. Характеристика прозрачных кодов.
16. Особенности применения перфорированных СК.
17. Характеристика алгоритмов обработки в декодерах СК.
18. Принцип декодирования по алгоритму Витерби.
19. Примеры применения СК и их возможности.
20. Достоинства и недостатки СК по сравнению с блоковыми кодами.
21. Поясните принципы “мягкого” и “жесткого” декодирования СК.
22. Работа кодера, реализующего алгоритм Витерби.
23. Постройте схему кодера СК, заданного порождающими полиномами 15,17. Охарактеризуйте формируемый СК.
24. Определите полную длину кодового ограничения для СК с порождающими многочленами 5,7; 4,5; 7,7,5 и приведите схемы кодеров.

25. Определите, какие из перечисленных ниже сверточных кодов со скоростью $1/2$ являются прозрачными или катастрофическими:

1. $G_1(X) = X^2$, $G_2(X) = 1+X+X^3$.
2. $G_1(X) = 1+X^2+X^4$, $G_2(X) = 1+X+X^3+X^4$.
3. $G_1(X) = 1+X+X^2+X^4$, $G_2(X) = 1+X^3+X^4$.
4. $G_1(X) = 1+X^4+X^5+X^6$, $G_2(X) = 1+X+X^3+X^5$.

26. Приведите маркировку порождающих многочленов п.25 в двоичную и восьмеричную формы. Приведите рисунки соответствующих кодеров.

27. Методы декодирования СК.

28. Характеристика двоичного симметричного канала.

29. Применение СК в сотовых системах связи.

30. Применение СК в спутниковых системах связи.

31. Постройте схему кодера СК, заданного порождающими полиномами 23,35; 13,15,17. Охарактеризуйте формируемый СК.

32. Сопряжение кодеров СК с модемами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Журавлев А. К., Никитин Г. И. Радиотехнические системы передачи информации: Учеб. пособие / ЛИАП. Л., 1989. 86 с.
2. Никитин Г. И. Первичные коды: Метод. указ. / ЛИАП. Л., 1984. 28 с.
3. Никитин Г. И. Эффективные коды: Метод. указ. / ЛИАП. Л., 1987. 28 с.
4. Никитин Г. И., Антипова И. Б., Красновидов А. В. Корректирующие коды: Метод. указ. / ЛИАП. Л., 1989. 32 с.
5. Никитин Г. И., Поддубный С. С. Помехоустойчивые циклические коды: Учеб. пособие / СПбГУАП. СПб., 1998. 72 с.
6. Харкевич А. А. Борьба с помехами. М.: Физматгиз, 1963. 276 с.
7. Питерсон У. У. Коды, исправляющие ошибки. М.: Мир, 1964. 340 с.
8. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976. 600 с.
9. Колесник В. Д., МIRONЧИКОВ Е. Т. Декодирование циклических кодов. М.: Связь, 1968. 252 с.
10. Финк Л. М. Теория передачи дискретных сообщений. М.: Сов. радио, 1963. 576 с.
11. Громаков Ю. А. Стандарты и системы подвижной радиосвязи. М.: Эко-Трендз, 1998. 242 с.
12. Ильин В. А. Телеуправление и телеизмерение: Учеб. пособие для вузов. 3-е изд. М.: Энергоиздат, 1982. 560 с.
13. Вайнер Э., Эш Р. Анализ рекуррентных кодов. Кибернетический сборник. Вып. 5. М.: Мир, 1963.
14. Теория кодирования / Т. Касам., Н. Токура, Е.Ивадари, Я. Инагаки. М.: Мир, 1978. 576 с.
15. Кларк Д., Кейн Д. Кодирование с исправлением ошибок в системах цифровой связи. М.: Радио и связь, 1987. 392 с.
16. Спилкер Д. Цифровая спутниковая связь. М.: Связь, 1979. 592 с.
17. Витерби А. Д., Омура Д. К. Принципы цифровой связи и кодирования. М.: Радио и связь, 1982. 536 с.
18. Блейхут Р. Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986. 576 с.
19. Шварцман В. О., Емельянов Г. А. Теория передачи дискретной информации: Учебник для вузов связи. М.: Связь, 1979. 424 с.
20. Радиосистемы передачи информации: Учеб. пособие для вузов/ Под ред. И. М. Теплякова. М.: Радио и связь, 1982. 264 с.

21. Радиотехнические системы передачи информации: Учеб. пособие для вузов/ Под ред. *В. В. Калмыкова*. М.: Радио и связь, 1990. 304 с.
22. *Немировский М. С.* Цифровая передача информации в радиосвязи. М.: Связь, 1980. 256 с.
23. *Банкет В. Л., Дорофеев В. М.* Цифровые методы в спутниковой связи. М.: Радио и связь, 1988. 240 с.
24. *Зигангиров К. Ш.* Процедуры последовательного декодирования. М.: Связь, 1974. 208 с.
25. *Злотник Б. М.* Помехоустойчивые коды в системах связи. М.: Радио и связь, 1989. 232 с.
26. *Дмитриев В. И.* Прикладная теория информации: Учеб. пособие для вузов. М.: Высш. школа, 1989. 320 с.
27. Передача информации в системах подвижной связи/ *В. Ю. Бабков, М. А. Вознюк, В. А. Петраков, А. Е. Рыжков, М. А. Сиверс*; СПбГУТ. СПб., 1999. 152 с.
28. *Петрович Н. Т.* Способ телеграфной проводной и радиосвязи фазоманипулированными колебаниями. А. с. № 105692, приоритет от 12.02.54.
29. *Окунев Ю. Б.* Теория фазоразностной модуляции. М.: Связь, 1979. 216 с.
30. *Месси Дж.* Пороговое декодирование. М.: Мир, 1966. 132 с.
31. *Возенкрафт Д., Рейффен Б.* Последовательное декодирование. М.: ИЛ, 1963. 156 с.
32. *Фано Р.* Эвристическое обсуждение вероятностного декодирования/ Сб. Теория кодирования. М.: Мир, 1964. С. 166–198.
33. *Невдяев Л. М., Смирнов А. А.* Персональная спутниковая связь. М.: Эко-Трендз, 1998. 216 с.
34. *Горностаев Ю. М., Соколов В. В., Невдяев Л. М.* Перспективные спутниковые системы связи. М.: Горячая линия – Телеком, 2000. 132 с.
35. *Никитин Г. И.* Спутниковые системы связи: Учеб. пособие / СПбГУАП. СПб. 2000. 80 с.

Оглавление

Предисловие	3
1. ПЕРВЫЙ РЕКУРРЕНТНЫЙ КОД ФИНКА	5
2. ДВОИЧНЫЕ СВЕРТОЧНЫЕ КОДЕРЫ	14
3. ПРЕДСТАВЛЕНИЕ СВЕРТОЧНЫХ КОДОВ С ПОМОЩЬЮ МНОГОЧЛЕНОВ	19
4. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СВЕРТОЧНЫХ КОДОВ ...	23
5. ОСНОВНЫЕ ПАРАМЕТРЫ И ХАРАКТЕРИСТИКИ СВЕРТОЧНЫХ КОДОВ	26
6. РАЗНОВИДНОСТИ СВЕРТОЧНЫХ КОДОВ	38
6.1. Систематические и несистематические коды	38
6.2. Катастрофические сверточные коды	39
6.3. Прозрачные сверточные коды	41
6.4. Перфорированные сверточные коды	44
6.5. Сверточные коды с различными алгоритмами обработки и конфигурациями исправляемых ошибок	45
7. СОГЛАСОВАНИЕ МОДЕМА И КОДЕКА	47
8. МЕТОДЫ ДЕКОДИРОВАНИЯ СВЕРТОЧНЫХ КОДОВ	49
8.1. Метод порогового декодирования	49
8.2. Метод последовательного декодирования	52
8.3. Метод декодирования по алгоритму Витерби	57
8.3.1. Декодирование в случае отсутствия ошибок при приеме	58
8.3.2. Декодирование в случае наличия ошибок при приеме .	59
8.3.3. Схемное построение декодера Витерби	61
9. ПРИМЕНЕНИЕ СВЕРТОЧНЫХ КОДОВ В СОТОВЫХ И СПУТНИКОВЫХ СИСТЕМАХ СВЯЗИ	64
10. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ “СВЕРТОЧНЫЕ КОДЫ”	71
Контрольные вопросы	74
Библиографический список	76

Учебное издание

Никитин Герман Иванович

СВЕРТОЧНЫЕ КОДЫ

Учебное пособие

Редактор *В. П. Зуева*
Компьютерная верстка *А. Н. Колешко*

Лицензия ЛР № 020341 от 07.05.97. Сдано в набор 05.10.01. Подписано к печати 29.12.01.
Формат 60×84 1/16. Бумага тип. №3. Печать офсетная. Усл. печ. л. 4,65. Усл. кр.-отт. 4,77.
Уч. -изд. л. 5,0. Тираж 300 экз. Заказ №

Редакционно-издательский отдел
Лаборатория компьютерно-издательских технологий
Отдел оперативной полиграфии
СПбГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67