Министерство образования и науки
Российской Федерации

ПСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

А.И. Спиридонов

ОСНОВЫ ТЕОРИИ ИНФОРМАЦИИ И КОДИРОВАНИЯ

УЧЕБНОЕ ПОСОБИЕ

Санкт-Петербург
2004

УДК 681.3.053

Рекомендовано к изданию
научно-методическим советом
Псковского государственного политехнического института

Рецензенты:
- Ильин С.Н., зам. генерального директора ОАО «СКБ ВТ»
- Григорьев О.Н., доцент каф. Электроэнергетики ППИ

Учебное пособие «Основы теории информации и кодирования» по дисциплине «Теория кодирования» (ЕН.Р.02) предназначено для студентов Псковского государственного политехнического института специальности 230101 «Электронные вычислительные машины, комплексы, системы и сети» очной, очно-заочной и очной с сокращенным сроком реализации форм обучения.

В пособии изложены основные положения теории информации и кодирования, принципы построения цифровых, эффективных и корректирующих кодов, способы реализации кодеров и декодеров.

Учебное пособие может использоваться студентами родственных специальностей и специализаций.

Спиридонов А.И. «Основы теории информации и кодирования». Учебное пособие. – СПб/Псков, Изд. СПбГПУ, 2004 – 140 с.

Содержание

ВВЕДЕНИЕ **5**

1. ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ. **7**

1.1. Понятия информации, сообщения и сигнала. 7
1.1.1. Понятие информации. *7*
1.1.2. Понятие сообщения. *8*
1.1.3. Понятие сигнала. *10*
1.2. Измерение информации. 12
1.2.1. Структурные меры информации. *13*
1.2.2. Статистическая мера информации. *17*
1.3. Квантование сигналов 23
1.3.1. Дискретизация сигналов. *25*
1.3.2. Квантование по уровню. *28*

2. КОДИРОВАНИЕ ИНФОРМАЦИИ. **31**

2.1. Цифровое кодирование информации. 31
2.1.1. Позиционные системы счисления. *32*
2.1.2. Смешанные системы счисления. *34*
2.1.3. Перевод числа из одной системы счисления в другую. *36*
2.1.4. Коды, не базирующиеся на системах счисления. *39*
2.2. Эффективное кодирование. 41
2.2.1. Избыточность сообщений. *41*
2.2.2. Теоретические основы эффективного кодирования. *43*
2.2.3. Построение эффективного кода по методам Шеннона-Фано и Хаффмена. *45*
2.2.4. Кодирование укрупненными блоками. *50*
2.3. Теоретические основы помехоустойчивого кодирования. 52
2.3.1. Теоремы Шеннона о помехоустойчивом кодировании. *53*
2.3.2. Классификация помехоустойчивых кодов. *54*
2.3.3. Общие принципы использования избыточности при построении корректирующих кодов. *55*
2.3.4. Избыточность корректирующих кодов. *60*
2.4. Коды, обнаруживающие ошибки. 61
2.5. Линейные коды, обнаруживающие и исправляющие ошибки. 65
2.5.1. Построение двоичного линейного кода. *68*
2.5.2. Кодирование. *74*
2.5.3. Синдромный метод декодирования. *76*
2.5.4. Кодирующее и декодирующее устройства. *79*
2.5.5 Коды Хэмминга. *84*
2.5.6. Матричное представление линейных кодов. *86*

2.5.7. Мажоритарное декодирование. *95*
2.6. Циклические коды..... 98
2.6.1. Выбор образующего многочлена. *101*
2.6.2. Формирование разрешенных кодовых комбинаций. *104*
2.6.3. Декодирование циклических кодов. *107*
2.6.4. Циклические коды с $d > 3$. *110*
2.6.5. Кодирующее и декодирующее устройства. *112*
2.6.6. Мажоритарное декодирование. *125*
2.6.7. Матричное представление циклических кодов. *129*
2.7. Краткие сведения о других кодах..... 131
2.8. Понятие об адаптивном кодировании..... 135
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ. 140

Введение

Во второй половине 20 в. человечество вступило в новый этап своего развития, заключающийся в переходе от индустриального общества к информационному. Процесс, обеспечивающий этот переход, получил название информатизации и состоит в реализации комплекса мер, направленных на обеспечение полного и своевременного использования достоверных знаний во всех общественно значимых видах человеческой деятельности.

Теоретическим фундаментом процесса информатизации является информатика. Впервые термин «информатика» (informatics) появился во Франции в 60-х годах после того, как Французская академия приняла его вместо понятия «обработка информации». После этого он стал широко использоваться в Европе и только в США и в Англии для обозначения рассматриваемой области деятельности используется термин «computer science». Буквальный перевод «computer science» – вычислительная наука, но это неверно по смыслу. По мнению академика А.А. Дородницына «computer science» – наука о преобразовании информации, в самом своем существе базирующаяся на применении вычислительной техники (ВТ).

Появление информатики как самостоятельной научной дисциплины обусловлено возникновением и распространением новой технологии сбора, передачи, обработки и преобразования информации, основанной на широком использовании средств ВТ и фиксации данных на машинных носителях. Но информатика – это не просто вычислительная технология, а вычислительная технология, органически встроенная в среду применения и преобразующая ее в соответствии с возможностями ЭВМ и потребностями самой среды применения. Воздействие вычислительной технологии на среду применения – главный отличительный элемент информатики как особой дисциплины.

Информатика, являясь новой научной областью, при решении своих задач опирается на такие традиционные

науки, как семиотика, математическая логика, электроника и т.д. В частности, теория информации и кодирования используется информатикой при кодировании информации и ее передаче.

Теория кодирования является одним из разделов теории информации, изучающей способы отображения сообщений с помощью символов некоторого алфавита. Являясь составной частью теории информации, теория кодирования базируется на ее основных положениях, поэтому в настоящем методическом пособии сначала рассматриваются элементы теории информации, а затем вопросы кодирования.

1. Элементы теории информации.

1.1. Понятия информации, сообщения и сигнала.

1.1.1. Понятие информации.

Информация наряду с материей и энергией является первичным понятием мира и поэтому в строгом смысле не может быть определена. Сам термин «информация» происходит от лат. «informatio» – разъяснение, осведомление.

Имеется множество определений понятия информации. В наиболее общем понимании информация есть отражение реального мира; это сведения, которые один реальный объект содержит о другом реальном объекте. В узком практическом понимании информация есть все сведения, представляющие интерес, подлежащие регистрации и обработке.

Сама по себе информация может быть отнесена к категории абстрактных понятий типа математических, но ряд ее особенностей приближает ее к материальным объектам. Так, информацию можно получить, записать, удалить, передать; информация не может возникнуть из ничего. Однако при распространении информации проявляется такое ее свойство, которое не присуще материальным объектам: при передаче информации из одной системы в другую количество информации в передающей системе не уменьшится, хотя в принимающей системе оно обычно увеличивается.

Понятие информации связано не с реальными вещами, а их моделями, отражающими сущность реальных вещей в той степени, в какой это необходимо для практических целей. В общем виде информация о различных природных явлениях и процессах может быть представлена в виде тех или иных полей. Математически такие поля описываются с помощью функций типа:

$$x = f(T, N), \qquad (1.1)$$

где x – физическая величина, характеризующая поле в момент времени T в точке пространства N.

Если все величины, входящие в приведенное соотношение, могут принимать непрерывный ряд значений, измеряемых вещественными числами, то такую информацию называют непрерывной. Если же установить минимальные шаги измерения всех величин, характеризующих поле x, то получим так называемое дискретное представление информации.

Практически точность любых измерений, как и человеческое восприятие, всегда ограничена, поэтому фактически, даже имея дело с непрерывной информацией, человек воспринимает ее в дискретном виде. Кроме того, любая непрерывная информация может быть аппроксимирована дискретной информацией с любой степенью точности, поэтому дискретную информацию можно считать универсальной формой ее представления.

1.1.2. Понятие сообщения.

Информация, зафиксированная в определенной форме, называется **сообщением**. Как и информация, сообщения бывают непрерывными и дискретными.

Непрерывное сообщение – это некоторая физическая, чаще всего электрическая величина, принимающая любые значения в заданном интервале и изменяющаяся в произвольные моменты времени.

Дискретное сообщение представляет собой последовательность отдельных элементов, разграниченных во времени и выбранных из некоторого набора. Физическая природа этих элементов безразлична; важно лишь, чтобы набор элементов был конечным и фиксированным.

Набор, из которого выбирают элементы, слагающие дискретное сообщение, называют **алфавитом**, а сами элементы – **буквами, знаками** или **символами**. Число различных букв в алфавите называют его **объемом.**

Дискретные сообщения часто разбивают на отдельные блоки конечной длины. Такие блоки, по аналогии с обычным языком, называют **словами.**

Длина блоков часто стандартизируется. В этих случаях представление информации называют **однородным**. Однородное представление информации имеет место в ЭВМ, где информация представляется в виде блоков равной длины, определяемой разрядностью ячеек памяти машины.

Процесс описания смыслового содержания информации с помощью символов (букв) называется **кодирование**. Обратный процесс, т.е. выявление смыслового содержания информации в принятых символах, называют **декодированием.**

При обработке информации часто возникает необходимость представлять буквы одного алфавита с большим объемом в другом алфавите с меньшим объемом. Операция перехода от первичного алфавита к вторичному также называется кодированием.

Поскольку объем вторичного алфавита меньше объема первичного, то каждому знаку первичного алфавита соответствует некоторая последовательность знаков вторичного алфавита, обычно называемая **кодовой комбинацией**. Число символов в кодовой комбинации называется ее **значностью**, а число ненулевых символов – **весом**. Операцию сопоставления кодовой комбинации соответствующего ей знака первичного алфавита также называют декодированием.

Для того чтобы потребитель информации мог распознать сообщение, т.е. отождествить символы с какими-либо объектами или процессами реального мира, он должен обладать определенными сведениями (алфавит, правило построения кода и т.п.).

Сведения, которыми располагает потребитель информации до ее получения и на знание которых рассчитывает отправитель, называют **априорными** (доопытными).

Сведения, которыми располагает потребитель после информационного обмена, называют **апостериорными** (послеопытными).

1.1.3. Понятие сигнала.

Сама по себе информация не материальна, но она является свойством материи и не может существовать без своего материального носителя – средства переноса информации в пространстве и во времени. В качестве носителей информации могут выступать самые разнообразные процессы и объекты, такие как электромагнитные волны, давление, механическое перемещение каких-либо объектов, бумага и т.д.

Все возможные носители сообщений, параметры которых содержат информацию, называют **сигналами.**

Формируют сигналы путем изменения одного или нескольких параметров носителя по закону, отображающему сообщение. Параметры сигнала, используемые для передачи сообщения, называют **информационными**. Правило, по которому производится изменение информационного параметра носителя при передаче сообщения, называют **правилом кодирования** или просто **кодом.**

Для передачи информации требуется, чтобы сигналы кроме информационных параметров имели и параметры **селекции** (отбора), позволяющие выделить полезный сигнал из совокупности других сигналов и помех.

Управление информационным параметром носителя в соответствии с передаваемым сообщением называется **модуляцией**. В зависимости от типа и числа параметров носителя может быть большое число различных методов модуляции. Например, если в качестве переносчика сообщений используется гармоническое колебание

$$x(t) = a \cdot \cos(\omega t + \varphi),$$

где a – амплитуда, ω – угловая частота, φ – начальная фаза, то может быть три метода модуляции: амплитудная (АМ), частотная (ЧМ) и фазовая (ФМ).

Применяют и комбинированные методы модуляции, когда в соответствии с передаваемым сообщением одновременно изменяется несколько параметров носителя. Примером

может служить амплитудно-квадратурная модуляция, представляющая собой сочетание фазовой и амплитудной модуляций (применяется в модемах). Во всех случаях комбинированной модуляции один из параметров носителя не должен изменяться, чтобы играть роль параметра селекции.

Если при модуляции информационный параметр изменяется скачкообразно и имеет конечное число значений, то модуляцию называют **манипуляцией**. При гармоническом носителе манипулированные сигналы обозначают: АМн, ЧМн, ФМн. Примеры манипулированных сигналов приведены на рис. 1.1.

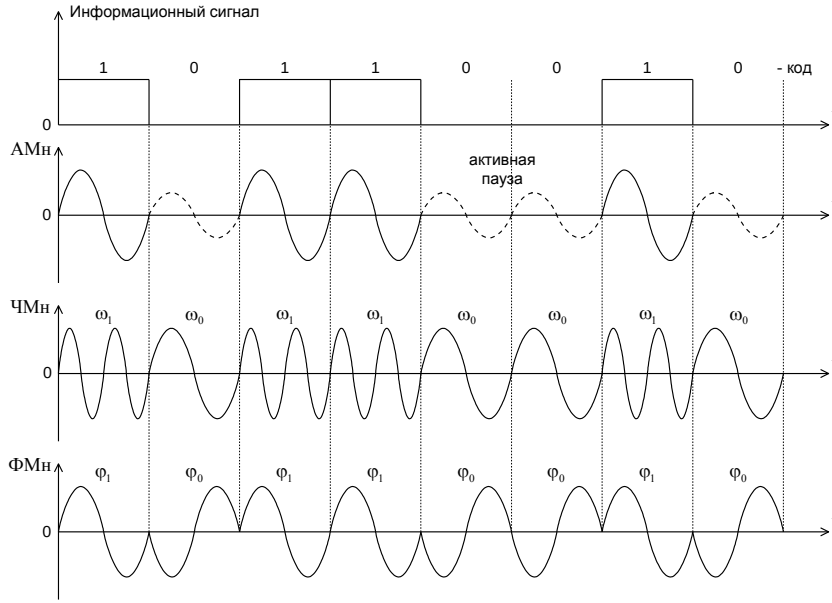


Рис. 1.1. Простейшие манипулированные сигналы.

Переносчиками информации может быть как постоянный, так и переменный ток.

В случае применения постоянного тока для передачи дискретной информации обычно используются импульсы

прямоугольной формы. Так представляется информационный сигнал, отображающий сообщение, поступающее, например, с выхода ЭВМ в двоичном коде.

Амплитудная модуляция связана с изменением амплитуды высокочастотного колебания. При АМн с пассивной паузой сигнал передается только при передаче "1" (может быть и наоборот), а при АМн с активной паузой сигнал передается всегда, но отличается амплитудами.

Частотная манипуляция состоит в том, что символы "0" и "1" передаются синусоидами, имеющими различные частоты. На рис. 1.1 разность $\Delta\omega = \omega_1 - \omega_0$ сделана слишком большой для наглядности, в реальных системах $\Delta\omega$ значительно меньше.

При фазовой манипуляции фаза синусоидального сигнала изменяется на 180° при изменении информационного сигнала от "0" к "1" и от "1" к "0".

1.2. Измерение информации.

Ценность теоретико-информационных представлений заключается в том, что благодаря их общности и абстрактности существенно разнородные физические и технические характеристики можно выразить через некоторые универсальные понятия теории информации и сопоставить их не только качественно, но и количественно.

Существуют три направления в теории информации, связанные с ее количественной оценкой – структурное, статистическое и прагматическое.

Структурная теория рассматривает дискретное строение массивов информации и ее измерение либо простым подсчетом информационных элементов, либо комбинаторным методом.

Статистическая теория оперирует понятием энтропии, характеризующей неопределенность сообщений и учитывающей вероятность их появления.

Прагматическая теория изучает вопросы о практическом использовании информации, о ее ценности для достижения поставленной цели.

В инженерной практике наибольшее применение находят первые две теории, поэтому ограничимся только их рассмотрением.

Независимо от направления теории информации при количественном ее определении должны выполняться следующие общие принципы:

- количественная мера информации не должна зависеть от физической природы объекта или процесса;
- мера должна быть такой, чтобы количество информации в данной мере было тем больше, чем больше априорная неопределенность ситуации;
- количественная мера должна удовлетворять принципу аддитивности, согласно которому количество информации должно изменяться прямо пропорционально длительности сообщения.

Рассмотрим меры информации, удовлетворяющие перечисленным требованиям.

1.2.1. Структурные меры информации.

Структурные меры информации применяются для оценки возможностей аппаратуры или сообщений вне зависимости от условий их применения.

В структурной теории различают геометрическую, комбинаторную и аддитивную меры информации.

Геометрическая мера используется для определения потенциального, т.е. максимально возможного количества в заданных структурных габаритах. Это количество называют **информационной емкостью** исследуемой части системы. Определение количества информации геометрическим методом сводится к измерению длины, площади или объема геометрической модели исследуемой системы. Единицей измерения при этом являются **кванты**, представляющие собой не-

делимые части. Физический смысл их может быть разным. Например, если ЗУ (запоминающее устройство) имеет информационную емкость 1 кбит (1024×1), то под квантом здесь подразумевается одна элементарная ячейка памяти, запоминающая 1 бит информации.

Комбинаторную меру применяют, когда требуется оценить возможность передачи информации при помощи различных комбинаций информационных элементов.

При формировании сообщений применяют следующие основные виды соединения букв:

– сочетания из h букв по ℓ; комбинации здесь отличаются составом букв, а максимальное их число определяется выражением:

$$Q = C_h^{\ell} = \frac{h!}{\ell!(h-\ell)!};$$

– сочетания с повторениями также различаются составом букв, но буквы в них могут повторяться по ℓ раз:

$$Q = \left(C_h^{\ell}\right)_{\text{повт.}} = \frac{(h+\ell-1)!}{\ell!(h-\ell)!};$$

– перестановки h букв различаются их порядком:

$$Q = P_h = 1 \cdot 2 \cdot 3 \ldots h = h!;$$

– перестановки с повторениями букв, причем одна из букв повторяется α, другая – β, наконец, последняя – γ раз:

$$Q = \left(P_h\right)_{\text{повт.}} = \frac{(\alpha+\beta+\ldots+\gamma)!}{\alpha!\beta!\ldots\gamma!};$$

– размещения из h букв по ℓ различаются и составом букв, и их порядком:

$$Q = A_h^{\ell} = \frac{h!}{(h-\ell)!};$$

– размещения с повторением букв до ℓ раз:

$$Q = \left(A_h^{\ell}\right)_{\text{повт.}} = h^{\ell}.$$

Количество информации в комбинаторной мере совпадает с числом возможных соединений букв.

Аддитивная мера (мера Хартли) используется для определения количества информации в сообщении без учета реальной вероятности появления отдельных букв (принимается одинаковой), возможной статистической связи между буквами и свойств канала связи.

Пусть передается сообщение длиной n, а объем используемого алфавита равен m. Очевидно, что при такой структуре может быть получено

$$N = m^n \qquad (1.2)$$

различных сообщений.

Чем больше N, тем больше априорная неопределенность ситуации и, следовательно, тем больше количество информации будет получено в результате приема конкретного сообщения. Однако непосредственно использовать выражение (1.2) для количественной оценки информации в сообщении нельзя, так как здесь не выдержан принцип аддитивности – пропорциональность между длиной сообщения n и количеством содержащейся в нем информации N. Но поскольку N однозначно определяется структурой сообщения, то для количественной оценки информации в нем целесообразно использовать некоторую функцию $I = f(N)$, такую, чтобы для I выполнялся принцип аддитивности.

В математической форме требование аддитивности записывается так:

$$\frac{dI}{dn} = k = \text{const}. \qquad (1.3)$$

Дифференцируя (1.2), получим: $\frac{dN}{dn} = m^n \ln m$, откуда

$$dN = N \cdot \ln m \, dn. \qquad (1.4)$$

Для того чтобы удовлетворялось требование аддитивности, выразим dn из (1.3) и подставим полученное значение dn в (1.4). Имеем:

$$dN = N \cdot \ln m \frac{dI}{k},$$

откуда $dI = \frac{k}{\ln m} \cdot \frac{dN}{N}$. (1.5)

Интегрируя (1.5) по N, получаем:

$$I = f(N) = \frac{k}{\ln m} = \ln N. \quad (1.6)$$

Выражение (1.6) справедливо для любого основания логарифма, так как содержит отношение натуральных логарифмов, а $\log_a x = \frac{\log_b x}{\log_b a}$.

Итак, для произвольного основания логарифма имеем:

$$I = \frac{k}{\log_a m} \cdot \log_a N. \quad (1.7)$$

Из (1.7) видим, что количество информации зависит не только от N, но и от объема используемого алфавита. Желательно убрать эту зависимость. Сделать это можно, если учесть, что в (1.3) не накладывается никаких ограничений на величину k, кроме того, что $k = \text{const}$. Поэтому можно принять $k = \log_a m$ и тогда получаем:

$$I = \log_a N$$

или

$$I = n \log_a m. \quad (1.8)$$

За единицу количества информации в данной мере принят 1 бит, соответствующий количеству информации, определенному по (1.8) при $a = 2$ для простейшей ситуации: n=1, m=2.

Выражение (1.8) впервые получено Р. Хартли, поэтому рассмотренную меру называют также мерой Хартли. Она позволяет определить потенциальное количество информации в сообщении заданной структуры.

1.2.2. Статистическая мера информации.

Статистическая теория, в отличие от структурной, позволяет оценивать информационные системы в конкретных условиях их применения, например, при передаче сообщений по каналам связи с шумами.

Статистическая теория оперирует понятием энтропии, введенной американским ученым Шенноном и определяемой выражением:

$$H(A) = -\sum_{i=1}^{m} P(a_i) \log_a P(a_i), \qquad (1.9)$$

где m – число возможных состояний объекта А; $P(a_i)$ – вероятность нахождения объекта А в i-ом состоянии, $i = \overline{1,m}$; $H(A)$ – энтропия объекта А.

В (1.9) предполагается, что имеет место ансамбль событий, т.е. полная группа событий ($\sum_{i=1}^{m} P(a_i) = 1$) с известным распределением вероятностей.

Если в (1.9) основание логарифма $a = 2$, то единицей измерения энтропии является бит, при $a = 10$ – дит, если логарифм натуральный – нат.

Энтропия $H(A)$ характеризует неопределенность состояния объекта А и использована Шенноном в статистической теории для определения количества информации $I(A)$, получаемое об объекте А в результате информационного обмена:

$$I(A) = H(A)_{нач.} - H(A)_{кон.}, \qquad (1.10)$$

где $H(A)_{нач.}$ – априорная энтропия объекта А, т.е. неопределенность состояния объекта А, имеющаяся у получателя информации до информационного обмена; $H(A)_{кон.}$ – апостериорная неопределенность объекта А, т.е. остающаяся у получателя после информационного обмена.

Единицы измерения $I(A)$ те же, что и для энтропии $H(A)$.

Априорная энтропия $H(A)_{нач.}$ определяется по формуле (1.9) после предварительного анализа объекта А с целью выяснения числа возможных состояний объекта А – m и вероятностей нахождения объекта в этих состояниях – $P(a_i)$, $i = \overline{1,m}$.

Для определения апостериорной энтропии $H(A)_{кон.}$ необходимо привлекать понятия объединения и условной энтропии. Рассмотрим эти понятия.

Объединением называется совокупность двух и более ансамблей дискретных случайных переменных. Рассмотрим объединение двух объектов А и В. Пусть число возможных состояний объекта А равно m, а объекта В – n. Возможные состояния объектов обозначим соответственно: a_i, $i = \overline{1,m}$; a_j, $j = \overline{1,n}$.

Результатом объединения явится сложный объект (А, В). Его состояния (a_i, b_j)представляют собой всевозможные комбинации состояний объектов А и В. Обозначим вероятность нахождения сложного объекта (А, В) в этих состояниях через $p(a_i, b_j)$.

Из теории вероятностей известно, что для независимых объектов

$$p(a_i, b_j) = p(a_i) \cdot p(b_j), \qquad (1.11)$$

а для зависимых объектов

$$p(a_i, b_j) = p(a_i) \cdot p(b_j / a_i) = p(b_j) \cdot p(a_i / b_j), \qquad (1.12)$$

где $p(a_i)$ и $p(b_j)$ – безусловные вероятности нахождения объектов А и В в соответствующих состояниях, а $p(b_j / a_i)$ и $p(a_i / b_j)$ – условные вероятности: $p(b_j / a_i)$ – вероятность того, что объект В примет конкретное состояние b_j, если объект А принял состояние a_i; $p(a_i / b_j)$ – вероятность того, что объект А примет состояние a_i, если объект В принял состояние b_j.

Предположим, что объект B принял состояние b_j и определим энтропию объекта A при условии, что B находится в этом конкретном состоянии. Такая энтропия называется условной, обозначается $H(A/b_j)$ и определяется выражением:

$$H(A/b_j) = -\sum_{i=1}^{m} p(a_i/b_j)\log_a p(a_i/b_j). \qquad (1.13)$$

Энтропия $H(A/b_j)$ зависит от того, в каком конкретном состоянии находится объект B. Умножив каждую условную энтропию $H(A/b_j)$ на $p(b_j)$ и сложив полученные произведения, получим выражение для средней условной энтропии:

$$\begin{aligned} H(A/B) &= \sum_{j=1}^{n} p(b_j)\cdot H(A/b_j) = \\ &= -\sum_{i=1}^{m}\sum_{j=1}^{n} p(b_j)\cdot p(a_i/b_j)\log_a p(a_i/b_j) \end{aligned} \qquad (1.14)$$

Учитывая (1.12), получаем:

$$H(A/B) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i, b_j)\log_a p(a_i/b_j). \qquad (1.15)$$

По смыслу $H(A/B)$ – неопределенность состояния объекта A, остающаяся после того, как состояние объекта B полностью выяснено.

Если отождествить с A передаваемое сообщение о состоянии объекта A, а с B – принимаемое сообщение, тогда средняя условная энтропия $H(A/B)$ будет характеризовать неопределенность передаваемого сообщения о состоянии объекта A, остающуюся после получения конкретного сообщения, т.е. после выяснения состояния объекта B. По смыслу это соответствует $H(A)_{кон.}$. Итак:

$$H(A)_{кон.} = H(A/B), \qquad (1.16)$$

$$\text{а } I(A) = H(A) - H(A/B). \qquad (1.17)$$

Используя полученные результаты, найдем выражения для количества информации, получаемого при передаче сообщений по каналам связи с шумами.

Рассмотрим сначала предельные случаи.

Если помех нет или их уровень настолько низок, что они не в состоянии уничтожить сигнал или имитировать сигнал при отсутствии передачи, то можно быть уверенным, что при передаче сигнала a_i на приемной стороне канала связи будет принят сигнал b_j, соответствующий сигналу a_i. Между передаваемыми и принимаемыми сигналами в этом случае существует жесткая связь, поэтому $p(a_i/b_j)=1$. Условная энтропия $H(A/B)$, определяемая выражением (1.15), при этом равна 0, т.к. $\log_a p(a_i/b_j)=\log_a 1=0$. Количество информации, получаемое в результате информационного обмена, $I(A)=H(A)$, т.е. максимально возможное.

При высоком уровне помех статистическая связь между передаваемыми и принимаемыми сообщениями отсутствует, т.е. при передаче любого сигнала a_i может быть принят любой сигнал b_j. В этом случае $p(a_i/b_j)=p(a_i)$ и $p(b_j/a_i)=p(b_j)$. Условная энтропия $H(A/B)$, определяемая (1.15), с учетом (1.12) может быть записана так:

$$H(A/B)=-\sum_{i=1}^{m}\sum_{j=1}^{n}p(b_j)p(a_i/b_j)\log_a p(a_i/b_j)=$$
$$=-\sum_{i=1}^{m}\sum_{j=1}^{n}p(b_j)p(a_i)\log_a p(a_i)=H(A)\cdot\sum_{j=1}^{n}p(b_j)$$

Но $\sum_{j=1}^{n}p(b_j)=1$, т.к. состояния объекта B составляют полную группу событий. Следовательно, $H(A/B)=H(A)$ и $I(A)=H(A)-H(A)=0$.

Информационные характеристики реальных каналов связи лежат между двумя этими предельными случаями. Несмотря на то, что часть информации поражается помехами, статистическая связь между передаваемыми и принимаемыми сообщениями сохраняется. Свойства канала связи при этом задаются канальной матрицей вида $p(A,B)$:

$$p(A,B)=\begin{vmatrix} p(a_1,b_1) & p(a_1,b_2) & \dots & p(a_1,b_n) \\ p(a_2,b_1) & p(a_2,b_2) & \dots & p(a_2,b_n) \\ \vdots & & & \\ p(a_m,b_1) & p(a_m,b_2) & \dots & p(a_m,b_n) \end{vmatrix} \qquad (1.18)$$

Матрица $p(A,B)$ всегда квадратная $(m=n)$. Вероятности, расположенные по диагонали и соответствующие $i=j$, определяют правильный прием, остальные – ложный.

Необходимые для расчета $I(A)$ безусловные вероятности $p(a_i)$ и $p(b_j)$ определяются из самой канальной матрицы:

$$p(a_i)=\sum_{j=1}^{n} p(a_i,b_j);\ p(b_j)=\sum_{i=1}^{m} p(a_i,b_j) \qquad (1.19)$$

Необходимые для расчета $H(A/B)$ условные вероятности $p(a_i/b_j)$ находятся из (1.12) по известным вероятностям $p(a_i,b_j)$ и $p(b_j)$.

Матрица $p(A,B)$ определяется экспериментально и периодически уточняется на основании результатов тестирования канала связи.

Выражение (1.17) определяет среднее количество информации, приходящееся на один знак сообщения. Для определения среднего количества информации в сообщении, состоящем из k знаков, найденное по (1.17) значение необходимо умножить на k.

Кроме отмеченного, при определении $I(A)$ необходимо учитывать статистические свойства источника информации. Осуществляется это следующим образом.

Пусть источник вырабатывает развернутое во времени дискретное сообщение $a_1a_2a_3\dots$ и т.д., где символы при знаках "a" указывают порядок их появления в сообщении. Тогда, если знаки независимы друг от друга, все условные вероятности появления знаков будут равны безусловным:

$$p(a_k/a_{k-1},a_{k-2},\dots,a_1)=p(a_k),\quad k=\overline{1,m}.$$

Если имеется связь только между двумя соседними знаками, то

$$p(a_k/a_{k-1}, a_{k-2}, \ldots, a_1) = p(a_k/a_{k-1}).$$

Корреляция может распространяться и на большее число предшествующих знаков, но у встречающихся на практике источников это число конечно. Такие источники называются **эргодическими.**

Для эргодических источников может быть найдено конечное число конечных характеристик состояний $S_1, S_2, S_3, \ldots$, таких, что условная вероятность появления очередного знака зависит только от того, в каком из этих состояний находился источник до его генерации. Вырабатывая очередной знак, источник переходит из одного состояния в другое или возвращается в исходное. Рассмотрим частные случаи.

Если корреляционные связи между знаками отсутствуют, то у источника имеется только одно характерное состояние S_1. Вероятность появления знака a_i в момент, когда источник находится в этом состоянии, равна $p(a_i)$. Выработав знак a_i, источник возвращается в исходное состояние S_1. Энтропия такого источника определяется выражением (1.9).

Когда корреляция имеется только между соседними знаками, число характерных состояний источника совпадает с объемом используемого алфавита. Находясь в одном из этих состояний, источник, выработав очередной знак, либо возвращается в исходное состояние, либо переходит в другое характерное состояние (номер характерного состояния, в котором окажется источник после генерации очередного знака, совпадает с номером этого знака). Для описания такого источника нужно знать условные вероятности появления знаков $p(a_i/a_j)$ для всех i и j. Найдем энтропию такого источника. Обозначим через $p(S_\ell/S_k)$ вероятность того, что источник, находясь в состоянии S_k, после генерации очередного знака перейдет в состояние S_ℓ. Тога энтропия источника в S_k состоянии будет равна:

$$H(S_k) = -\sum_{\ell/k} p(S_\ell/S_k) \log_a (S_\ell/S_k). \qquad (1.20)$$

Суммирование в (1.20) осуществляется по всем возможным переходам из S_k состояния в S_ℓ.

Умножив $H(S_k)$ на вероятность нахождения источника в S_k состоянии – $P(S_k)$ и сложив полученные произведения, получим выражение для средней энтропии источника при наличии корреляции между соседними знаками:

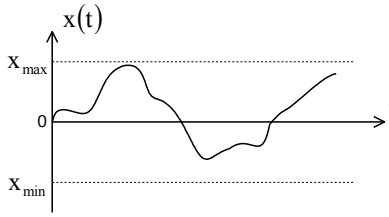
$$H(A) = \sum_{k=1}^{m} P(S_k) H(S_k), \qquad (1.21)$$

где $P(S_k)$ определяется как среднее значение вероятностей генерации знака a_k, определенное по всем характерным состояниям источника.

1.3. Квантование сигналов.

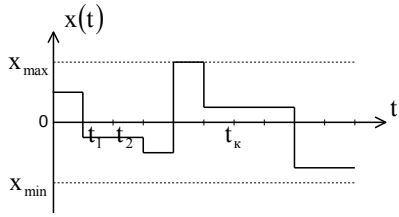
Квантование сигналов по времени и по уровню используется при переходе от аналогового представления сигнала к цифровому, что дает значительные преимущества при передаче, хранении и обработке информации.

Для выяснения сущности процессов квантования рассмотрим разновидности функций $x(t)$, используемых при описании реальных сигналов.

1. 

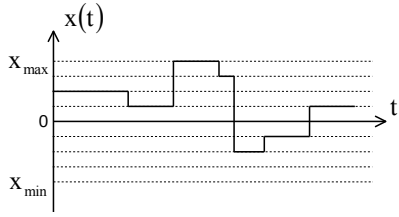
Непрерывная функция непрерывного аргумента.

Функция $x(t)$ может измениться в любой момент времени на интервале своего существования и принять любое значение из области своего изменения (x_{max}, x_{min}).

2. 

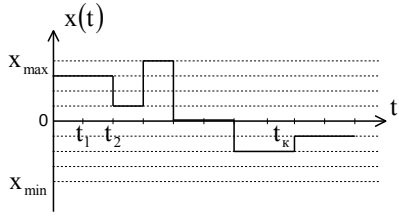
Непрерывная функция дискретного аргумента.

Функция $x(t)$ может изменяться только в строго определенные моменты времени, но принимает при этом любые значения из области своего изменения (x_{max}, x_{min}).

3. 

Дискретная функция непрерывного аргумента.

В данном случае вся область изменения функции $x(t)$ разбита на ряд дискретных уровней, которые может принимать функция в любой момент времени.

4. 

Дискретная функция дискретного аргумента.

Здесь как значения, которые может принимать функция $x(t)$, так и моменты времени, когда происходят ее изменения, являются дискретными.

Сигналы, описываемые функциями 1-го типа, называются непрерывными (аналоговыми); 2-го и 3-го типов – дискретно-непрерывными; 4-го типа – дискретными.

Операцию, переводящую непрерывный сигнал во вторую разновидность, называют квантованием по времени или дискретизацией.

Операцию, переводящую непрерывный сигнал в третью разновидность, называют квантованием по уровню.

Совместное применение операций дискретизации и квантования по уровню позволяет преобразовать непрерывный сигнал $x(t)$ в дискретный по координатам x и t, т.е. перевести его в четвертую разновидность.

1.3.1. Дискретизация сигналов.

Дискретизация по времени должна производиться так, чтобы при обработке сигнала по отсчетным значениям $x(t_k)$ можно было воспроизвести функцию $v(t)$, отличающуюся от исходной функции $x(t)$ в заданных пределах:

$$\varepsilon(t) = x(t) - v(t), \qquad (1.22)$$

где $\varepsilon(t)$ – погрешность дискретизации, т.е. погрешность восстановления сигнала $x(t)$.

Выбор критерия оценки погрешности дискретизации сигнала зависит от назначения системы, использующей дискретизированный сигнал.

Наиболее часто отклонение воспроизводимой функции $v(t)$ от сигнала $x(t)$ на интервале дискретизации $\Delta t_k = t_k - t_{k-1}$ оценивается следующими критериями:

– критерий наибольшего отклонения

$$\varepsilon_m = \max_{t \in \Delta t_k} |\varepsilon(t)|, \qquad (1.23)$$

где $\varepsilon(t)$ – текущая погрешность, определяемая выражением (1.22);

– среднеквадратичный критерий

$$\overline{\varepsilon^2} = \sqrt{\frac{1}{\Delta t_k} \int_{\Delta t_k} \varepsilon^2(t) dt}, \qquad (1.24)$$

где черта сверху означает усреднение по вероятностному множеству;

– интегральный критерий

$$\bar{\varepsilon} = \int_{\Delta t_k} \varepsilon(t)dt; \qquad (1.25)$$

– вероятностный критерий

$$p[\varepsilon(t) < \varepsilon_0] = p_0, \qquad (1.26)$$

где ε_0 – допустимое значение погрешности; p_0 – допустимая вероятность того, что погрешность не превысит значения ε_0.

Более подробно критерии оценки погрешности дискретизации рассмотрены в [6].

В соответствии с признаком регулярности отсчетов выделяют равномерную и неравномерную дискретизацию. Дискретизация называется равномерной, если на всем отрезке обработки сигнала

$$\Delta t_k = t_k - t_{k-1} = \text{const}.$$

Методы равномерной дискретизации применяются наиболее часто благодаря простоте как алгоритмов, так и аппаратуры дискретизации и восстановления сигнала.

Выбор частоты равномерной дискретизации часто осуществляется на основании теоремы В.А. Котельникова, сформулированной им применительно к передаче сигналов по каналам связи.

Теорема Котельникова утверждает, что непрерывный сигнал $x(t)$, имеющий спектр частот от 0 до f_c, полностью определяется последовательностью своих мгновенных значений, зафиксированных через равные интервалы времени $\Delta t = \frac{1}{2f_c}$, и описывается выражением:

$$x(t) = \sum_{k=-\infty}^{\infty} x(k \cdot \Delta t) \cdot \frac{\sin 2\pi f_c (t - k\Delta t)}{2\pi f_c (t - k\Delta t)}. \qquad (1.27)$$

Доказательство теоремы Котельникова приводится в работах [6] и [10].

Выражение (1.27) носит название ряда Котельникова и представляет собой разложение непрерывной функции $x(t)$ по системе ортогональных функций вида $\frac{\sin x}{x}$. Множители $x(k \cdot \Delta t)$ называются отсчетами функции $x(t)$ и представляют собой мгновенные значения непрерывного сигнала $x(t)$ в дискретные моменты времени $t_k = k \cdot \Delta t$. Множитель $\varphi(t) = \frac{\sin x}{x}$, где $x = 2\pi f_c (t - k\Delta t)$, называется функцией отсчетов.

По физическому смыслу каждое слагаемое ряда Котельникова представляет собой отклик идеального фильтра нижних частот с частотой среза f_c на короткий импульс, приходящий в момент времени $t_k = k \cdot \Delta t$ и имеющий площадь, пропорциональную мгновенному значению функции $x(t)$ в тот же момент времени. Отсюда следует, что для передачи сигнала $x(t)$ с ограниченным спектром по каналу связи, необходимо через равные интервалы $\Delta t = \frac{1}{2} f_c$ взять отсчеты мгновенных значений $x(t)$ и передать по каналу связи короткие импульсы, площади которых пропорциональны этим отсчетам. На приемном конце эти импульсы нужно пропустить через фильтр нижних частот и тем самым восстановить исходный сигнал $x(t)$ как сумму откликов фильтра. Значение всей суммы в момент времени $t_k = k \cdot \Delta t$ определяется только k-м слагаемым, так как все остальные слагаемые обращаются в нуль. Внутри промежутка времени Δt значения непрерывного сигнала определяются всеми слагаемыми.

Теорема Котельникова, как следует из определения, относится к сигналам с ограниченным спектром и, следовательно, к сигналам с неограниченной длительностью. Реальные же сигналы, используемые для передачи информации, имеют конечную длительность T, а это значит, что они имеют бесконечно широкий спектр частот. Тем не менее, можно предположить, что вне интервала времени T реальный непрерывный

сигнал $x(t)$ также имеет место, но все его мгновенные значения лежат ниже некоторого уровня, ограниченного чувствительностью устройства обработки информации. Это позволяет считать спектры реальных сигналов ограниченными. Однако просуммировать бесконечное число слагаемых, входящих в (1.27), не представляется возможным. Ограничение же числа слагаемых значением $m = 2Tf_c$ приводит к появлению ошибки воспроизведения сигнала:

$$\varepsilon(t) = x(t) - \sum_{k=1}^{m} x(k \cdot \Delta t) \cdot \frac{\sin 2\pi f_c (t - k\Delta t)}{2\pi f_c (t - k\Delta t)}. \qquad (1.28)$$

Величина этой ошибки равна нулю в точках отсчета и достигает максимума ε_m в середине интервала дискретизации.

1.3.2. Квантование по уровню.

Сущность квантования по уровню заключается в том, что весь диапазон возможных изменений функции $x(t)$ разбивается на n интервалов (шагов) квантования.

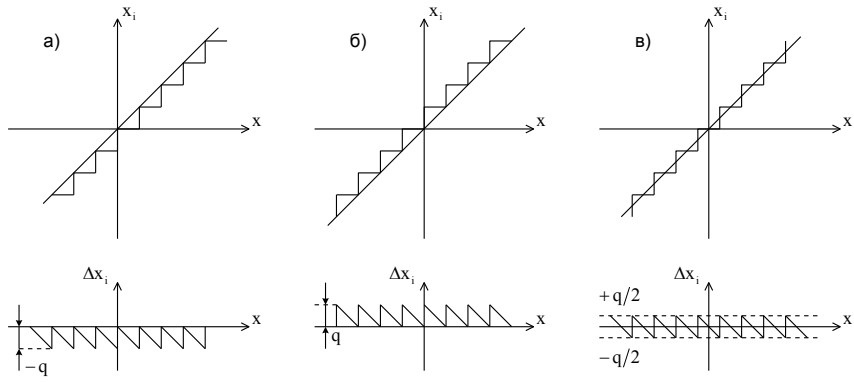
В результате квантования любое из значений x округляется до одного из уровней квантования. Округление может производиться до ближайшего меньшего или большего уровня. Квантование по уровню может быть равномерным или неравномерным. При равномерном квантовании шаги квантования одинаковы и определяются выражением:

$$q = x_i - x_{i-1} = \frac{x_{max} - x_{min}}{n}, \quad i = \overline{1,n}, \qquad (1.29)$$

где n – число интервалов квантования.

Так как в процессе квантования по уровню значение сигнала $x(t)$ отображается уровнем x_i, а каждому уровню x_i может быть поставлен в соответствие свой номер (число), то при передаче или хранении можно вместо истинного значения уровня квантования x_i использовать соответствующее число i. Истинное значение уровня квантования легко восстановить, зная масштаб по оси x.

Устройство для квантования сигналов по уровню, называемое квантизатором, представляет собой нелинейный элемент с амплитудной характеристикой типа: а) – при отождествлении сигнала с ближайшим меньшим уровнем квантования; б) – с ближайшим большим уровнем; в) – с ближайшим уровнем.



Погрешность квантования $\Delta x_i = x_i - x$, называемая также шумом квантования, является периодической функцией, изменяющейся в зависимости от значения x в пределах: а) $(0 \div -q)$; б) $(0 \div +q)$; в) $(+q/2 \div -q/2)$.

Так как x – случайная величина с плотностью распределения $W(x)$, то погрешность квантования на i-ом уровне квантования будет также случайной, зависящей от x.

Определим погрешность квантования для квантизатора типа в).

Полагая, что шаг квантования $q << (x_{max} - x_{min})$, можно считать, что плотность $W(x)$ постоянна в интервале q и равна $W(x_i)$, т.е.

$$W(x_i) \cdot q = \int_{x_i - q/2}^{x_i + q/2} W(x) dx .$$

Тогда математическое ожидание шума квантования на i-ом уровне:

$$M[\Delta x_i] = \int_{x_i - q/2}^{x_i + q/2} (x_i - x) W(x) dx =$$
$$= x_i W(x_i) \cdot q - W(x_i) \cdot \frac{x^2}{2}\Big|_{x_i - q/2}^{x_i + q/2} = 0 \qquad (1.30)$$

Дисперсия шума квантования:

$$D[\Delta x_i] = \int_{x_i - q/2}^{x_i + q/2} (x_i - x)^2 W(x) dx =$$
$$= W(x_i) \int_{x_i - q/2}^{x_i + q/2} (x_i - x)^2 dx = \frac{q^2}{12} [W(x_i) \cdot q] \qquad (1.31)$$

Просуммировав выражения для $D[\Delta x_i]$ по всем уровням x_i, получим суммарную дисперсию погрешности квантования:

$$D_\Sigma = \sum_{i=1}^{n} D[\Delta x_i] = \frac{q^2}{12} \sum_{i=1}^{n} W(x_i) \cdot q . \qquad (1.32)$$

Но $\sum_{i=1}^{n} W(x_i) \cdot q = 1$, т.к. каждое слагаемое в отдельности представляет собой вероятность попадания случайной величины в интервал $(x_i - q/2) \div (x_i + q/2)$.

Следовательно,

$$D_\Sigma = \frac{q^2}{12} . \qquad (1.33)$$

Но именно таким выражением определяется дисперсия случайной величины, равномерно распределенной в интервале q. Следовательно, погрешность квантования можно считать равномерно распределенной в интервале квантования.

Среднеквадратичная ошибка квантования:

$$\sigma[\Delta] = \sqrt{D_\Sigma} = \frac{q}{2\sqrt{3}} \qquad (1.34)$$

Зная допустимое значение $\sigma[\Delta]_{доп}$, из (1.34) можно определить максимально допустимый шаг квантования:

$$q_{max} \le 2\sqrt{3}\, \sigma[\Delta]_{доп} \qquad (1.35)$$

2. Кодирование информации.

Кодирование – процесс описания смыслового содержания информации с помощью символов. Осуществляется кодирование с целью представления сообщений в форме, удобной для передачи по данному каналу связи. Рассматриваемые ниже принципы кодирования справедливы как для систем, основная функция которых – передача информации в пространстве (системы связи), так и для систем, основная функция которых – передача информации во времени (системы хранения информации). В последних линией связи считается среда, в которой хранится информация.

В настоящее время существует большое число методов кодирования сообщений. Выбор конкретного метода определяется целью, достижение которой планируется. Например, кодирование информации может осуществляться так, чтобы обеспечить наибольшую простоту и надежность аппаратной реализации информационных устройств. В других случаях стремятся обеспечить минимальное время при передаче или минимальный объем запоминающих устройств. Может быть поставлена цель достижения заданной достоверности при передаче и хранении информации в условиях сильных помех и т.д. В дальнейшем будет отмечаться назначение конкретных методов кодирования.

2.1. Цифровое кодирование информации.

Цифровое кодирование обеспечивает наибольшую простоту информационных устройств и используется при кодировании информации без учета статистических свойств источника и помех в канале связи.

При таком кодировании любому дискретному сообщению ставится в соответствие определенное число, выраженное в какой-либо системе счисления.

Системой счисления называется способ наименования и записи чисел. В любой системе счисления для представления чисел выбираются некоторые символы, называемые базисными числами, а все остальные числа получаются в результате выполнения каких-либо операций с базисными. Символы, используемые для записи чисел, могут быть любыми, важно лишь, чтобы они были разными и значение каждого из них было известно.

2.1.1. Позиционные системы счисления.

В настоящее время наиболее распространенным принципом образования систем счисления является позиционный, при котором значение (вес) каждого символа зависит от его положения (позиции) в ряду символов, представляющих число. Самый младший разряд располагается в конце кодовой комбинации и имеет вес, равный единице, если отображаемое число является целым (в дальнейшем рассматривается кодирование только целых чисел). Вес единицы каждого следующего разряда больше веса предыдущего в m раз, где m – основание системы счисления (m указывает на число единиц какого-либо разряда, объединяемых в единицу более старшего разряда). Полное число получается в результате суммирования по всем разрядам:

$$Q = \sum_{i=1}^{\ell} a_i m^{i-1} = a_\ell m^{\ell-1} + a_{\ell-1} m^{\ell-2} + \ldots + a_1 m^0, \qquad (2.1)$$

где i – номер разряда; ℓ – количество разрядов; a_i – коэффициент, принимающий любые целочисленные значения в пределах от 0 до $m-1$ и показывающий, сколько единиц i-го разряда содержится в числе.

Кодовая комбинация, отображающая число Q, записывается в виде последовательности коэффициентов a_i:

$$(a_\ell\ a_{\ell-1}\ a_{\ell-2} \ldots a_1)_m \qquad (2.2)$$

Для указания того, в какой системе счисления записано число, принято при его изображении основание системы счисления указывать в виде нижнего индекса при нем.

В современном мире наиболее распространенным является представление чисел посредством арабских цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Позиционные системы счисления различаются выбором базисных чисел a_i.

В **десятичной системе** $(m=10)$ $a_i=\overline{0,9}$. В соответствии с (2.1) кодовые комбинации нужно читать так:

$$348_{10}=3\cdot10^2+4\cdot10^1+8\cdot10^0$$

В **восьмеричной системе** $(m=8)$ $a_i=\overline{0,7}$.

$$265_8=2\cdot8^2+6\cdot8^1+5\cdot8^0=128+48+5=181_{10}$$

В **шестнадцатеричной системе** $(m=16)$ $a_i=\overline{0,15}$. Как и в предыдущих системах счисления, для обозначения первых десяти базисных чисел здесь используются арабские цифры, а для изображения цифр, больших 9, применяются буквы: A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

$$F17B_{16}=15\cdot16^3+1\cdot16^2+7\cdot16^1+11\cdot16^0=$$

$$=61440+256+112+11=61819_{10}$$

В **двоичной системе** $(m=2)$ $a_i=0,1$.

$$1011_2=1\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0=8+2+1=11_{10}.$$

Чем больше основание системы счисления, тем меньше количество разрядов требуется для представления данного числа, а следовательно, тем меньше времени требуется для его передачи и меньше объем запоминающего устройства для хранения. Однако с ростом основания существенно повышаются требования к линиям связи и аппаратуре распознавания сигналов. Кроме этого, арифметические и логические элементы вычислительных устройств должны иметь большое число устойчивых состояний. В связи с этими обстоятельствами в современной вычислительной технике, в устройствах автоматики и связи широко используется двоичная система счисления. Логические элементы, соответствующие этой системе, должны иметь всего два устойчивых состояния. Задача различения сигналов, отображающих символы, сводится в этом случае к задаче обнаружения (есть сигнал или нет его), что

значительно проще. Просто выполняются арифметические операции:

0 + 0 = 0	0 - 0 = 0	0 × 0 = 0
0 + 1 = 1	1 - 0 = 1	0 × 1 = 0
1 + 0 = 1	1 - 1 = 0	1 × 0 = 0
1 + 1 = 1 0	1 0 - 1 = 1	1 × 1 = 1

Наиболее распространенная при кодировании и декодировании логическая операция – суммирование по модулю. При $m = 2$ она также наиболее проста и определяется равенствами:

0 ⊕ 0 = 0	1 ⊕ 1 = 0
0 ⊕ 1 = 1	1 ⊕ 0 = 1

Все вышеперечисленное является причиной того, что большинство современных ЭВМ используют двоичную систему счисления.

Неудобство использования двоичного кода заключается в громоздкости записи чисел. Поэтому если возникает необходимость кодирования информации «вручную», например, при составлении программы на машинном языке, то предпочтение отдается восьмеричной или шестнадцатеричной системам.

Кроме этого, двоичный код неудобен при вводе и выводе информации, т.к. трудно оперировать с непривычными двоичными числами. Человек привык работать и легко ориентируется при использовании десятичного кода. Для устранения этого неудобства применяется двоично-десятичный код, являющийся одним из вариантов смешанной системы счисления.

2.1.2. Смешанные системы счисления.

В смешанной системе счисления числа, заданные в системе счисления с основанием p, изображаются с помощью цифр другой системы счисления с основанием $q < p$ по сле-

дующему правилу: каждый коэффициент p-ичного разложения записывается в q-ичной системе. При этом p называется старшим основанием, а q – младшим основанием, а сама смешанная система называется $(q-p)$-ичной. Для того чтобы запись числа в смешанной системе была однозначной, для представления любой p-ичной цифры отводится одно и то же количество q-ичных разрядов, достаточное для представления любого базисного числа p-ичной системы. Так, в смешанной двоично-десятичной системе счисления для изображения каждой десятичной цифры отводится четыре двоичных разряда. Здесь имеется избыточность, т.к. двоичная тетрада позволяет изобразить не 10, а 16 чисел, и следовательно, существует целый ряд двоично-десятичных систем, отличающихся весами двоичных разрядов внутри одной тетрады. В наиболее часто используемой (2-10)-ной системе веса двоичных разрядов внутри тетрады естественны, т.е. 8 – 4 – 2 – 1.

Пример:

$(892)_{10} = (1000.1001.0010)_{2-10}$.

Существуют и другие варианты двоично-десятичного кода. Например, двоично-десятичные коды с весами 5 – 1 – 2 – 1 и 2 – 4 – 2 – 1 широко используются при поразрядном уравновешивании в цифровых измерительных приборах [6].

В смешанных системах счисления особого внимания заслуживает случай, когда

$$p = q^k, \qquad (2.3)$$

где k – целое положительное число. В этом случае запись какого-либо числа в смешанной системе совпадает с изображением этого числа в системе с основанием q. Доказательство этого утверждения приведено в [3].

Примеры:

1) $265_8 = (010.110.101)_{2-8} = 010110101_2$;

2) $E4A_{16} = (1110.0100.1010)_{2-16} = 111001001010_2$.

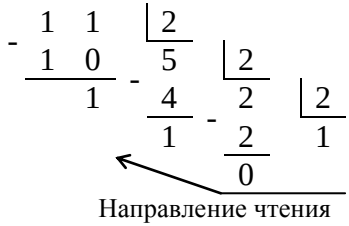
В обоих случаях условие (2.3) выполняется: в первом – $p=8,\ q=2,\ k=3$; во втором – $p=16,\ q=2,\ k=4$. Справедливость полученных равенств легко проверяется если по (2.1) найти десятичные эквиваленты для исходного числа и его отображения в двоичном коде.

2.1.3. Перевод числа из одной системы счисления в другую.

1. Перевод целого числа десятичного кода в любую другую систему счисления осуществляется по следующему правилу: нужно делить исходное десятичное число на величину основания той системы счисления, в которую переводится число, а затем частное от деления снова делить на то же основание и так до тех пор, пока частное не окажется меньше делителя. Полученные остатки от деления и последнее частное будут представлять собой разряды числа в новой системе счисления. Чтение нового кода осуществляется с конца, т.е. последнее частное дает цифру старшего разряда.

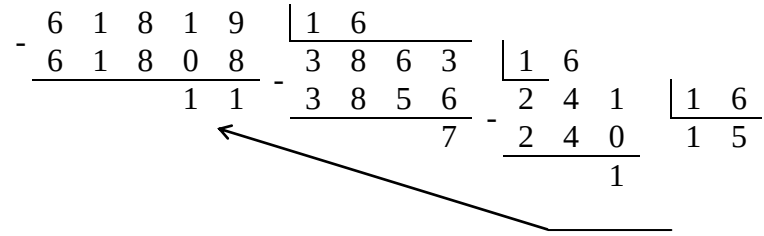
Примеры:

1) перевести десятичное число 11 в двоичную систему счисления:



Искомое двоичное число 1011, т.е. $11_{10}=1011_2$.

2) перевести десятичное число 61819 в шестнадцатеричную систему:



Поскольку $11_{10} = B_{16}$, а $15_{10} = F_{16}$, искомое шестнадцатеричное число запишется в виде $F17B$, т.е. $61819_{10} = F17B_{16}$.

2. Для перевода чисел из любой системы счисления в десятичную можно использовать запись этого числа в виде полинома (2.1). Выполнив сложение, получаем искомое десятичное число.

Пример.

Перевести восьмеричное число 745 в десятичную систему счисления:

$$745_8 = 7 \cdot 8^2 + 4 \cdot 8^1 + 5 \cdot 8^0 = 485_{10}.$$

Перевод в десятичный код может быть выполнен также по схеме Горнера. Процедура перевода следующая. Старший разряд исходного кода нужно умножить на основание переводимой системы счисления и полученное произведение сложить со следующим символом кода. Полученную сумму вновь умножить на основание и результат сложить со следующим символом и так продолжать до последнего (младшего) разряда кода. Для числа 745_8 указанные вычисления будут следующими:

$$\left[(7 \times 8) + 4\right] \times 8 + 5 = 60 \times 8 + 5 = 485_{10}.$$

Схема Горнера особенно удобна при переводе чисел из двоичной системы. Имеем: $745_8 = 111.100.101_2$. Переведем полученное двоичное число в десятичное:

1	1	1	1	0	0	1	0	1
1	3	7	15	30	60	121	242	485

Искомое десятичное число 485, т.е. $111100101_2 = 485_{10}$.

3. Перевод чисел из любой недесятичной системы счисления в другую недесятичную систему выполняется в следующей последовательности:

– по методике, изложенной в п. 2, перевести исходный код в десятичный;

– по методике, изложенной в п. 1, перевести полученный десятичный код во вторую систему счисления.

Решение упрощается, если основания подлежащих переводу систем удовлетворяют условию (2.3), т.е. $p = q^k$, где $p > q$, k – целое положительное число. При этом, если исходное число представлено в системе счисления с основанием p, для получения его изображения в системе с основанием q достаточно каждый символ исходного кода представить k-значным числом в системе счисления с основанием q. Примеры такого перевода приведены в п. 2.1.2 для чисел 265_8 и $E4A_{16}$. Если же исходное число представлено в системе счисления с основанием q, для получения его изображения в системе с основанием p нужно в последовательности символов исходного кода, начиная с младших разрядов, выделять группы по k знаков и каждой из этих групп поставить в соответствие число в системе счисления с основанием p. Если последняя группа окажется неполной, ее нужно дополнить нулями со стороны старших разрядов.

Примеры:

1) перевести двоичное число 10011000111100 в восьмеричную систему счисления ($k = 3$):

$$\underbrace{10}_{2}\ \underbrace{011}_{3}\ \underbrace{000}_{0}\ \underbrace{111}_{7}\ \underbrace{100}_{4}$$

Получаем: $10011000111100_2 = 23074_8$;

2) перевести то же двоичное число в шестнадцатеричную систему счисления ($k = 4$):

$$\underbrace{10}_{2}\ \underbrace{0110}_{6}\ \underbrace{0011}_{3}\ \underbrace{1100}_{12}$$

Получаем: $10011000111100_2 = 263C_{16}$.

Последние примеры показывают, что если $p_1 = q^{k_1}$, а $p_2 = q^{k_2}$, перевод чисел из системы счисления с основанием p_1 в систему с основанием p_2 и наоборот может выполняться через промежуточную систему счисления с основанием q. В частности, так можно осуществлять перевод чисел из восьмеричной системы в шестнадцатеричную и наоборот через двоичный код, т.е. сначала исходное число представляется в двоичном коде, затем полученная двоичная комбинация разбивается на группы, каждой из которых ставится число во второй системе счисления.

2.1.4. Коды, не базирующиеся на системах счисления.

Среди кодов указанного типа наибольшее распространение имеют код Грея, часто называемый также рефлексно-двоичным, и унитарный (числоимпульсный) код.

Код Грея применяется в технике аналого-кодового преобразования, где он позволяет свести к единице младшего разряда погрешность неоднозначности при считывании. Достигается это за счет того, что смежные числа этого кода отличаются только в одном разряде. Комбинации кода Грея, соответствующие десятичным числам от 0 до 15, приведены в таблице 2.1.

Таблица 2.1.

Q_{10}	Код Грея	Q_{10}	Код Грея	Q_{10}	Код Грея
0	0000	6	0101	11	1110
1	0001	7	0100	12	1010
2	0011	8	1100	13	1011
3	0010	9	1101	14	1001
4	0110	10	1111	15	1000
5	0111				

Формируются комбинации кода Грея из обычного двоичного кода путем суммирования по модулю 2 исходной ко-

довой комбинации с такой же комбинацией, сдвинутой вправо на 1 разряд. Младший разряд в сдвинутой комбинации при этом отбрасывается.

Пример:

$13_{10} = 1101_2$

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ \oplus\ \ 1\ 1\ 0 \\ \hline 1\ 0\ 1\ 1 \end{array}$$ – код Грея

Перевод числа из кода Грея в обычный двоичный осуществляется по следующему правилу: все предшествующие нули и первая единица со стороны старших разрядов остаются без изменения; последующие символы (0 и 1) остаются без изменения, если число единиц, им предшествующих в коде Грея, четно, и инвертируются, если число единиц нечетно.

Пример:

1 1 0 1 0 1 1 0 – код Грея

1 0 0 1 1 0 1 1 – двоичный код

Проверка:

$$\begin{array}{r} 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1 \\ \oplus\ \ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \end{array}$$ – код Грея

Унитарный код. Кодовые комбинации здесь отличаются друг от друга числом единиц. В таблице 2.2, в качестве примера, приведены комбинации пятиразрядного унитарного кода.

Таблица 2.2.

Q_{10}	0	1	2	3	4	5
Унитарный код	00000	10000	11000	11100	11110	11111

Данный код характеризуется простотой реализации, поэтому и применяется при небольших значениях Q.

2.2. Эффективное кодирование.

Целью эффективного кодирования является устранение избыточности в сообщении, что дает выигрыш во времени передачи и требуемом объеме запоминающего устройства. Применяется эффективное кодирование в случаях, когда уровень помех в канале связи незначителен и их влиянием на работу информационной системы можно пренебречь.

2.2.1. Избыточность сообщений.

Энтропия $H(A)$, определяемая выражением (1.9), характеризует среднее количество информации, приходящееся на один знак дискретного сообщения, и является максимальной, когда все знаки сообщения равновероятны и независимы друг от друга. При учете вероятностей появления отдельных знаков сообщения, а также статистических связей между знаками, энтропия уменьшается. Это значит, что при наличии связей между знаками часть информации не является непредвиденной для получателя и ее можно было бы не передавать по каналу связи, а восстанавливать на приемной стороне на основании известной связи между знаками. Таким образом, появляется возможность передачи сообщений в сокращенном виде.

Величина, показывающая, какая часть сообщения при данных условиях может быть устранена без потери информации, называется избыточностью.

Абсолютная величина избыточности определяется выражением:

$$\Delta D = H(A)_{max} - H(A), \qquad (2.4)$$

где $H(A)_{max}$ – максимальная энтропия сообщения, определяемая при равновероятных и статистически независимых знаках в сообщении; $H(A)$ – реальная энтропия сообщения.

Относительная избыточность, называемая информационной, определяется выражением:

$$D = \frac{\Delta D}{H(A)_{max}} = 1 - \frac{H(A)}{H(A)_{max}}. \qquad (2.5)$$

Информационная избыточность показывает относительную недогруженность одного знака алфавита и является безразмерной величиной, изменяющейся в пределах: $0 \le D \le 1$.

В соответствии с (2.5), избыточность, обусловленная неравновероятным появлением знаков в сообщении, будет равна:

$$D_P = 1 - \frac{\sum_{i=1}^{M} P(a_i) \log_a P(a_i)}{\log_a M}, \qquad (2.6)$$

где M – объем используемого алфавита; $\log_a M = H(A)_{max}$.

Избыточность, вызванная статистической связью между знаками сообщения:

$$D_S = 1 - \frac{\sum_{k=1}^{M} P(S_k) \cdot H(S_k)}{-\sum_{k=1}^{M} P(a_i) \cdot \log_a (a_i)}, \qquad (2.7)$$

где $P(S_k)$ – вероятность нахождения источника в S_k характерном состоянии; $H(S_k)$ – энтропия источника в S_k состоянии, определяемая выражением (1.20).

Полная информационная избыточность равна [8]:

$$D = D_p + D_s - D_p \cdot D_s \qquad (2.8)$$

Подставляя (2.6) и (2.7) в (2.8), после преобразований, получаем:

$$D = 1 - \frac{\sum_{k=1}^{M} P(S_k) H(S_k)}{\log_a M} \qquad (2.9)$$

Статистический анализ текстов на русском языке показывает, что $D_P \cong 0{,}13$, $D_S \cong 0{,}73$, $D \cong 0{,}75$. Из этих данных следует, что основная избыточность здесь обусловлена наличием статистической связи между буквами.

2.2.2. Теоретические основы эффективного кодирования.

Пусть требуется передать достаточно длинное сообщение, составленное из ансамбля знаков $\{a_k\}$, $k=\overline{1,M}$, где M – объем первичного алфавита: $\sum_{k=1}^{M} P(a_k)=1$.

При кодировании каждому знаку первичного алфавита ставится в соответствие своя кодовая комбинация во вторичном алфавите, содержащем m символов. Обозначим через n_k длину кодовой комбинации, соответствующей знаку a_k, а через $L_{ср}$ – среднюю длину кодовой комбинации при выбранном способе кодирования. Тогда:

$$L_{ср}=\sum_{k=1}^{M} n_k \cdot p(a_k). \qquad (2.10)$$

При эффективном кодировании стремятся к тому, чтобы $L_{ср} \to L_{min}$. Это значит, что при определенном объеме передаваемой информации необходимо увеличивать ее количество на каждый символ сигнала, т.е. повышать, как обычно говорят, «плотность» упаковки информации.

Теоретической основой для определения минимально возможной средней длины кодовых комбинаций служит теорема Шеннона о кодировании при отсутствии помех. Теорема читается так: при кодировании множества сигналов с энтропией $H(A)$ в алфавите, насчитывающем m символов, при условии отсутствия помех, средняя длина кодовых комбинаций не может быть меньше частного от деления указанной энтропии на количество информации в одном символе, т.е.

$$L_{ср} \geq \frac{H(A)}{\log_a m}. \qquad (2.11)$$

Справедливость теоремы Шеннона можно видеть из следующих рассуждений. Количество информации, получаемое в результате информационного обмена, как показано в п. 1.2.2, определяется выражением:

$$I(A) = H(A) - H(A/B).$$

При отсутствии помех $H(A/B) = 0$. Следовательно, среднее количество информации, передаваемое одним знаком первичного алфавита, равно:

$$I(A) = H(A) = -\sum_{k=1}^{M} P(a_k) \log_a a_k .$$

Очевидно, что кодирование будет тем эффективнее, чем большее количество информации будет приходиться на каждый символ кодовой комбинации. Так как во вторичном алфавите m символов, а максимальное количество информации в символе будет в случае, когда вероятности появления символов одинаковы, то максимально возможное количество информации, приходящееся на I символ кодовой комбинации, будет равно $\log_a m$.

Рассуждая аналогично, можно прийти к выводу, что для эффективного кодирования длина кодовой комбинации n_k, передающей информацию о знаке a_k, должна приближаться к частному от деления количества информации, содержащегося в знаке a_k, на $\log_a m$:

$$n_k \geq \frac{-\log_a P(a_k)}{\log_a m}. \qquad (2.12)$$

Если $P(a_k)$ не являются целочисленными степенями m, то рассчитанная таким образом величина n_k окажется дробной. В таком случае минимальная избыточность достигается, если n_k будет выбрано в соответствии с выражением:

$$n_k = E\left[\frac{-\log_a P(a_k)}{\log_a m}\right], \qquad (2.13)$$

где E – знак округления до ближайшего большего числа. Кодирование при этом не будет оптимальным, т.е. средняя длина кодовых комбинаций будет больше минимально возможной. Однако в теории информации доказывается, что при кодировании более крупными блоками можно сколь угодно приблизиться к нижней границе.

Из выражения (2.12) следует, что при эффективном кодировании длина конкретной кодовой комбинации n_k тем больше, чем меньше вероятность появления знака a_k в сообщении, и наоборот. Это значит, что часто встречающиеся знаки следует кодировать более короткими комбинациями, а редко встречающиеся – более длинными. Благодаря этому и достигается выигрыш в величине $L_{ср}$.

Дополнительное сокращение времени, затрачиваемого на передачу сообщения, можно достигнуть с помощью специального кода, позволяющего вести передачу без промежутков между кодовыми комбинациями. Чтобы на приемной стороне можно было отличить одну комбинацию от другой, т.е. чтобы код был разделимым (однозначно декодируемым), он должен строится так, чтобы ни одна более длинная кодовая комбинация не являлась продолжением более короткой. Например, при указанном кодировании нельзя применять комбинации 110 и 11010, так как вторая комбинация получена из первой путем добавления 10.

Возможность построения такого кода определяется с помощью неравенства Крафта. В нем утверждается, что такой код существует, если

$$\sum_{k=1}^{M} m^{-n_k} \le 1 \qquad (2.14)$$

(доказательство неравенства Крафта приведено в [10]).

Итак, неравенство Крафта определяет возможность построения разделимого кода при выбранных m и n_k, а теорема Шеннона – минимально возможную среднюю длину кодовых комбинаций, отображающих буквы первичного алфавита.

2.2.3. Построение эффективного кода по методам Шеннона-Фано и Хаффмена.

Из теоремы Шеннона следует, что для обеспечения минимальной средней длины кодовых комбинаций эффектив-

ный код должен строиться так, чтобы символы кодовых комбинаций были равновероятны и статистически независимы друг от друга. Если требования независимости и равновероятности символов по каким-либо причинам невыполнимы, то, чем лучше они выполняются, тем ближе код к оптимальному. Именно эти соображения и используются при построении эффективного кода по методикам Шеннона-Фано и Хаффмена.

Код Шеннона-Фано строится следующим образом. Буквы первичного алфавита вписываются в таблицу в порядке убывания вероятностей. Затем они разделяются на две группы так, чтобы суммы вероятностей появления знаков в каждой из групп были по возможности одинаковы. Для всех букв верхней группы в качестве первого символа кодовой комбинации принимается 1, а для букв нижней группы – 0. В дальнейшем каждая из полученных групп разбивается на две по возможности равновероятных подгруппы и символ 1 или 0 (в зависимости от подгруппы) берется вторым элементом кодовой комбинации и т.д. Процесс повторяется до тех пор, пока в каждой подгруппе останется по одной букве.

Пример 1.

Построить по методике Шеннона-Фано эффективный код для букв первичного алфавита, $M = 8$. Вероятности появления букв в сообщениях:

$$P(a_1) = P(a_2) = \frac{1}{4};\ P(a_3) = P(a_4) = \frac{1}{8};$$

$$P(a_5) = P(a_6) = P(a_7) = P(a_8) = \frac{1}{16}.$$

Процедура построения кода и результат приведены в таблице 2.3.

Таблица 2.3.

Буква	$P(a_k)$	**Разбиения**	**Кодовая комбинация**	n_k
a_1	$\frac{1}{4}$		11	2
a_2	$\frac{1}{4}$		10	2
a_3	$\frac{1}{8}$		011	3
a_4	$\frac{1}{8}$		010	3
a_5	$\frac{1}{16}$		0011	4
a_6	$\frac{1}{16}$		0010	4
a_7	$\frac{1}{16}$		0001	4
a_8	$\frac{1}{16}$		0000	4

Из таблицы видно, что ни одна более длинная кодовая комбинация не является продолжением более короткой, следовательно, полученный код является разделимым.

Для определения того, является ли полученный код оптимальным или нет, необходимо сравнить значения реально полученной средней длины кодовой комбинации, вычисляемой в соответствии с (2.10), с минимально достижимой по теореме Шеннона. Имеем:

$$L_{ср} = \sum_{k=1}^{M} n_k \cdot P(a_k) = 2 \cdot 2 \cdot 0{,}25 + 2 \cdot 3 \cdot 0{,}125 + 4 \cdot 4 \cdot 0{,}625 =$$

$$= 2{,}75 \text{ символа}.$$

Так как вторичный код является двоичным ($m = 2$), то минимальная средняя длина кодовой комбинации, определяемая по (2.11) и выраженная в символах, численно будет совпадать с энтропией первичного алфавита, выраженной в битах:

$$H(A) = -\sum_{k=1}^{M} P(a_k)\log_2 P(a_k) = 2 \cdot 0{,}25 \cdot 2 + 2 \cdot 0{,}125 \cdot 3 +$$

$$+ 4 \cdot 0{,}0625 \cdot 4 = 2{,}75 \text{ бит.}$$

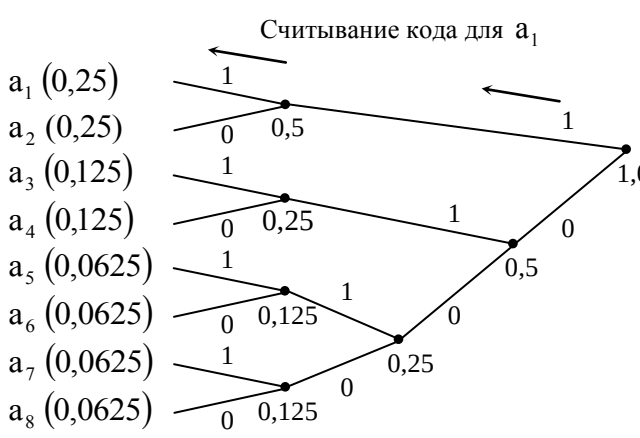
Таким образом, реально полученная средняя длина кодовой комбинации в точности равна энтропии первичного алфавита, выраженной в битах, следовательно, построенный код является оптимальным.

Для построения эффективного кода по методике **Хаффмена** буквы первичного алфавита вписываются в столбец в порядке убывания вероятностей (последнее не является обязательным, но желательно) и строится кодовое дерево в следующей последовательности. Сначала две буквы с наименьшими вероятностями объединяются в новую букву с вероятностью, равной сумме вероятностей объединенных букв. В дальнейшем новая буква и оставшиеся исходные буквы рассматриваются как равноценные и выполняется такая же процедура объединения. Этот процесс продолжается до тех пор, пока не получается единственная вспомогательная буква с вероятностью, равной 1.

При каждом парном объединении одна из ветвей кодового дерева обозначается символом 1, а другая 0. Код для каждой буквы представляет собой последовательности нулей и единиц, обозначающих ветви кодового дерева, по которым нужно пройти от вершины кодового дерева до данной буквы.

Пример 2.

Построить по методике Хаффмена эффективный код для букв первичного алфавита, приведенных в примере 1.



Знаки	Кодовые комбинации
a_1	11
a_2	10
a_3	011
a_4	010
a_5	0011
a_6	0010
a_7	0001
a_8	0000

Сравнивая этот код с ранее полученным по методике Шеннона-Фано, можно видеть, что длины кодовых комбинаций n_k, $k=\overline{1,8}$, в обоих кодах совпадают. Следовательно, код, полученный по методике Хаффмена, также является оптимальным.

Если при парном объединении букв верхнюю и нижнюю ветви кодового дерева обозначить по-другому (поменять местами 0 и 1), кодовые комбинации, отображающие буквы первичного алфавита, будут другими. Однако код остается разделимым, а значения n_k остаются прежними, что и является принципиальным.

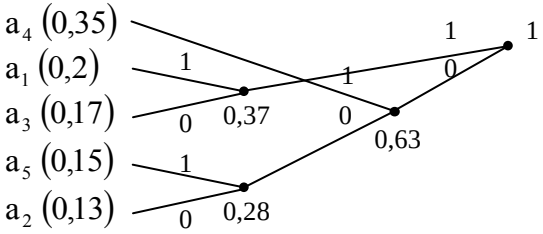
Пример 3.

Для передачи сообщений используется алфавит из 5 знаков. Вероятность появления знаков: $P(a_1)=0{,}2$; $P(a_2)=0{,}13$; $P(a_3)=0{,}17$; $P(a_4)=0{,}35$. Определить вероятность появления знака a_5 и закодировать знаки первичного алфавита эффективным кодом по методике Хаффмена.

Решение. Эффективному кодированию подлежат только знаки, составляющие полную группу событий, т.е. $\sum_{i=1}^{M} P(a_i)=1$.

Следовательно, $P(a_5)=1-\sum_{i=1}^{4} P(a_i)=0{,}15$.

Строим кодовое дерево:



Знаки	Кодовые комбинации
a_1	11
a_2	000
a_3	10
a_4	01
a_5	001

Проверка построенного кода на оптимальность осуществляется так же, как и в примере 1.

Для количественной оценки степени близости построенного кода к оптимальному используется коэффициент эффективности ψ, определяемый выражением:

$$\psi = \frac{L_{cp.\,min}}{L_{cp.}}, \qquad (2.15)$$

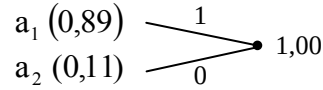
где $L_{cp.\,min}$ – теоретически достижимая средняя минимальная величина кодовой комбинации, определяемая выражением (2.11), $L_{cp.}$ – реально достигнутая средняя длина, определяемая по формуле (2.10).

2.2.4. Кодирование укрупненными блоками.

В случае, когда побуквенное кодирование не приводит к получению оптимального или близкого к нему кода, нужно применять кодирование укрупненными блоками.

Пусть для передачи сообщения используется всего две буквы, причем вероятности появления этих букв существенно различны: $P(a_1) = 0{,}89$; $P(a_2) = 0{,}11$.

Выполнив кодирование по методике Хаффмена,



получаем, что средняя длина кодовой комбинации равна 1 символу $(a_1 - 1\ a_2 - 0)$. В то же время энтропия первичного источника

$$H(A) = -0{,}89 \log_2 0{,}89 - 0{,}11 \log_2 0{,}11 = 0{,}5 \text{ бита}.$$

Таким образом, реально полученная средняя длина кодовой комбинации при побуквенном кодировании вдвое превышает минимально достижимую величину, определяемую теоремой Шеннона, т.е. $\psi = 0{,}5$.

Произведем кодирование блоков из двух букв:

Блоки	Вероятность	Разбиения	Кодовая комбинация	n_k
$a_1 \cdot a_1$	$0{,}89 \cdot 0{,}89 = 0{,}792$		1	1
$a_1 \cdot a_2$	$0{,}89 \cdot 0{,}11 = 0{,}098$		0 1	2
$a_2 \cdot a_1$	$0{,}11 \cdot 0{,}89 = 0{,}098$		0 0 1	3
$a_2 \cdot a_2$	$0{,}11 \cdot 0{,}11 = 0{,}012$		0 0 0	3

Средняя длина кодовой комбинации, отображающей блок из двух букв:

$$L_{\text{ср.бл.2}} = 0{,}792 \cdot 1 + 0{,}098 \cdot 2 + 0{,}098 \cdot 3 + 0{,}012 \cdot 3 = 1{,}318 \text{ символа}$$

Следовательно, на одну букву сообщения приходится в среднем

$$L_{\text{ср.}} = \frac{L_{\text{ср.бл.2}}}{2} = \frac{1{,}318}{2} \cong 0{,}66 \text{ символа},$$

т.е. в данном случае по сравнению с побуквенным кодированием получено сокращение средней длины кодовой комбинации на 0,34 символа.

Здесь коэффициент эффективности построенного кода

$$\psi = \frac{0{,}5}{0{,}66} \cong 0{,}76.$$

Если полученный результат является неудовлетворительным, необходимо проводить кодирование более крупными блоками – по три буквы в блоке, четыре и т.д. Приемлемая величина ψ должна быть задана.

2.3. Теоретические основы помехоустойчивого кодирования.

Целью помехоустойчивого кодирования является обеспечение заданной достоверности передачи информации при наличии помех в канале связи за счет внесения избыточности в сообщения.

Помехи могут быть естественными и искусственными. Естественные помехи – грозовые разряды, космическое излучение и т.д. Искусственные помехи образуются вследствие деятельности человека и связаны в основном с прерыванием цепей электрического тока (электросварка, системы зажигания двигателей внутреннего сгорания) или электромагнитными волнами.

Достоверность передачи информации характеризуется вероятностью ошибочного приема 1 символа кодовой комбинации.

Требуемая достоверность обеспечивается за счет внесения избыточности в сообщения. Необходимо иметь в виду, что в данном случае избыточность имеет совершенно иную природу, чем та, что устранялась методами эффективного кодирования. Рассмотренная ранее избыточность была заложена в первичном алфавите источника сообщений и характере передаваемых сообщений; эта избыточность не согласована со статистическими свойствами помех в канале связи и поэтому не может быть использована для повышения достоверности принимаемого сообщения. При помехоустойчивом кодировании избыточность обусловлена введением специальных – контрольных символов, количество и состав которых зависит от свойств канала связи. Первичный код при этом может быть оптимальным.

2.3.1. Теоремы Шеннона о помехоустойчивом кодировании.

Существует несколько теорем Шеннона, посвященных кодированию при наличии помех. Первая из них читается так: если скорость создания информации источником меньше пропускной способности канала, то среди кодов, обеспечивающих сколь угодно малую вероятность ошибки, существует код, при котором скорость передачи каналом сколь угодно близка к скорости создания информации источником.

Конкретизируем понятия, используемые в данной теореме.

Скорость создания информации источником определяет количество информации, вырабатываемое источником в единицу времени, и равна

$$H(C) = \frac{H(A)}{\bar{\tau}}, \text{ бит/с}, \qquad (2.16)$$

где $H(A)$ – энтропия источника сообщений; $\bar{\tau}$ – средняя длительность передаваемых знаков.

Скорость передачи информации определяет количество информации, переносимое одним знаком сообщения в единицу времени. При знаках равной длительности τ скорость передачи информации

$$C = n[H(A) - H(A/B)], \qquad (2.17)$$

где n – количество знаков (букв), вырабатываемых источником в единицу времени.

Пропускная способность канала определяется как максимум скорости передачи информации, взятый по всем возможным распределениям вероятностей появления знаков на входе канала, т.е.

$$H(K) = \max_{P(a_i)}(C). \qquad (2.18)$$

Используя введенные обозначения, сущность 1-ой теоремы Шеннона можно сформулировать так: если $H(C) < H(K)$, то среди помехоустойчивых кодов существует код, обеспечивающий сколь угодно малую вероятность ошибки при передаче без задержки.

Вторая теорема Шеннона о кодировании при наличии помех утверждает, что если скорость создания информации источником больше пропускной способности канала, то никакой код не может сделать вероятность ошибки сколь угодно малой.

Таким образом, согласно 2-ой теореме Шеннона, при $H(C) > H(K)$ потери информации в канале связи неизбежны, при этом величина минимальных потерь равна разности $H(C) - H(K)$.

В случае помехоустойчивого кодирования, так же как и в случае эффективного, теоремы Шеннона не указывают путей построения помехоустойчивых кодов, а лишь указывают на возможность и условия практически идеальной передачи информации при наличии помех.

2.3.2. Классификация помехоустойчивых кодов.

В настоящее время помехоустойчивые (корректирующие) коды используются как для обнаружения ошибок, так и для их исправления. Классификация корректирующих кодов представлена на рис. 2.1.

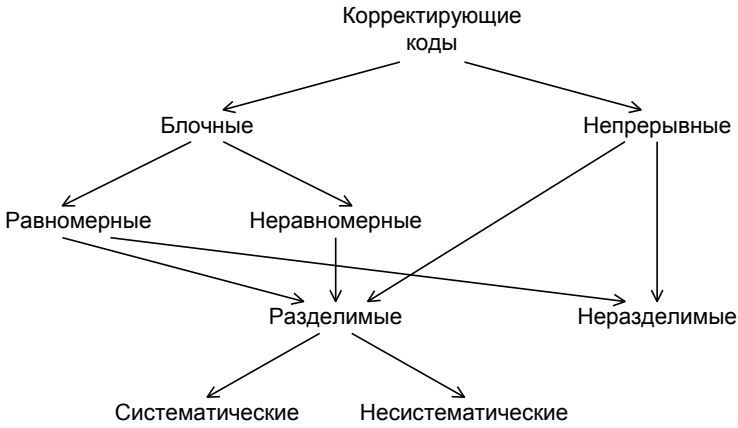


Рис. 2.1. Классификация корректирующих кодов.

Как видно из рис. 2.1, все корректирующие коды делятся на два больших класса – блочные и непрерывные.

Блочные коды – коды, в которых каждому знаку сообщения в соответствие ставится кодовая комбинация из n символов, т.е. блок из n символов.

Блочный код называется равномерным, если значение n остается постоянным для всех знаков сообщения. В противном случае код называется неравномерным.

Непрерывные коды представляют собой непрерывную последовательность элементов, не подразделяемую на блоки. Процессы кодирования и декодирования здесь имеют непрерывный характер.

Блочные и непрерывные коды, в свою очередь, могут быть разделимыми и неразделимыми.

При кодировании разделимыми кодами в выходных последовательностях символов можно разграничить информационные символы и проверочные. В неразделимых кодах такое разграничение невозможно.

Разделимые коды делятся на систематические и несистематические.

Систематическими называются коды, в которых контрольные (проверочные) элементы представляют собой различные линейные комбинации информационных элементов. Несистематические коды таким свойством не обладают.

В настоящее время среди корректирующих кодов наибольшее распространение имеют блочные разделимые систематические коды.

2.3.3. Общие принципы использования избыточности при построении корректирующих кодов.

Предположим, что на вход кодера (кодирующее устройство) подана безизбыточная кодовая комбинация, состоящая из k информационных символов. В кодере по определенному

алгоритму к поступившим k символам добавляется m контрольных символов. В результате на выходе кодера будет получена n-значная комбинация, где $n = m + k$.

При двоичном кодировании число различных кодовых комбинаций на входе кодера равно 2^k, а возможное число n-значных кодовых комбинаций – 2^n. Очевидно, что из теоретически возможного множества n-значных комбинаций на выходе кодера реально существуют только 2^k комбинаций, соответствующих входным. Эти комбинации составляют подмножество разрешенных кодовых комбинаций. Остальные $(2^n - 2^k)$ возможных комбинаций для передачи не используются и образуют подмножество запрещенных кодовых комбинаций.

В процессе передачи информации по каналу связи в результате действия помех любая из 2^k разрешенных комбинаций может трансформироваться в любую из множества 2^n. Таким образом, число возможных исходов передачи будет равно $2^k \cdot 2^n$. Из них:

2^k случаев безошибочной передачи;

$2^k\left(2^k - 1\right)$ случаев перехода одних разрешенных комбинаций в другие разрешенные комбинации, что соответствует необнаруживаемым ошибкам;

$2^k\left(2^n - 2^k\right)$ случаев перехода в запрещенные кодовые комбинации, что соответствует обнаруживаемым ошибкам.

На приемной стороне разрешенных и запрещенных кодовых комбинаций известны, поэтому сам факт появления комбинации в подмножестве запрещенных является фактом обнаружения ошибки. Однако локализовать и исправить ошибку при этом невозможно.

Для исправления ошибки все множество запрещенных комбинаций разбивается на 2^k непересекающихся подмножеств M_i, $i = \overline{1, 2^k}$, каждое из которых ставится в соответствие одной из разрешенных комбинаций A_i. При получении запрещенной комбинации, принадлежащей подмножеству M_i, принимается

решение, что передавалась комбинация A_i. В данном случае ошибка будет исправлена, когда запрещенная комбинация действительно образовалась из разрешенной комбинации A_i.

Способ разбиения запрещенных кодовых комбинаций на непересекающиеся подмножества при данном виде исправления ошибок зависит от того, какие ошибки должны исправляться корректирующим кодом.

Большинство корректирующих кодов предназначено для исправления взаимно независимых ошибок определенной кратности или пакетов ошибок.

Кратностью ошибки называется количество искаженных символов в кодовой комбинации. При взаимно независимых ошибках вероятность искажения любых r символов в n-разрядной кодовой комбинации равна:

$$P_r = C_n^r P^r (1-P)^{n-r}, \qquad (2.19)$$

где P – вероятность искажения одного символа.

Так как $P << 1$, то из (2.19) следует, что наиболее вероятны ошибки малой кратности, т.е. при взаимно независимых ошибках наиболее вероятен переход в кодовую комбинацию, отличающуюся от данной в наименьшем числе символов.

Под пакетом ошибок подразумевается следующее. В реальных каналах связи длительность импульсов помехи часто превышает длительность одного символа, в результате одновременно могут исказиться несколько расположенных рядом символов кодовой комбинации.

Длиной пакета (пачки) ошибок называется число следующих друг за другом символов, левее и правее которых в кодовой комбинации искаженных символов не содержится.

Пример:

1 0 0 1 0 0 1 1 1 0 0 0 1 – исходная кодовая комбинация
1 0 0 0 1 0 0 1 0 1 0 0 1 – искаженная кодовая комбинация

длина пакета ошибок

Теоретические исследования в области помехоустойчивого кодирования показали, что корректирующая способность кода прежде всего определяется кодовым расстоянием кода. Кодовое расстояние двух любых комбинаций характеризует степень различия этих комбинаций и равно числу символов, в которых комбинации отличаются одна от другой. Чтобы определить кодовое расстояние (d), достаточно сложить кодовые комбинации по модулю 2 и подсчитать число единиц в сумме. Например:

$$\begin{array}{cccccccccc} & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ \oplus & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \qquad d = 7$$

Под кодовым расстоянием кода подразумевается минимальное кодовое расстояние, определенное по множеству всех разрешенных комбинаций данного кода.

Рассмотрим на примере трехзначных кодовых комбинаций зависимость корректирующей способности кода от d.

Пусть $d = 1$. В этом случае все кодовые комбинации из множества 000, 001, 010, 011, 100, 101, 110, 111 являются разрешенными и любая одиночная ошибка переводит одну разрешенную комбинацию в другую разрешенную. Это случай равнодоступного кода, не обладающего корректирующей способностью.

Если $d = 2$, тогда множество всех возможных комбинаций 2^n, $n = 3$, разбивается на два непересекающихся подмножества: разрешенное и запрещенное. Пусть подмножество разрешенных комбинаций составляют комбинации 000, 011, 101, 110, $d = 2$. Тогда комбинации 001, 010, 100, 111 составят подмножество запрещенных кодовых комбинаций. Анализируя выделенные подмножества, можно видеть, что в данном случае однократная ошибка всегда переводит комбинацию из подмножества разрешенных в подмножество запрещенных, т.е. при $d = 2$ код способен обнаруживать все одиночные ошибки.

В общем случае при необходимости обнаруживать ошибки кратности r кодовое расстояние кода должно быть больше r по крайней мере на 1, т.е.

$$d_{min} \geq r+1. \qquad (2.20)$$

Исправлять ошибки код с $d=2$ не может, т.к. одна и та же запрещенная комбинация в данном случае может быть получена при однократной ошибке из различных разрешенных комбинаций. Для исправления одиночной ошибки каждой разрешенной кодовой комбинации необходимо сопоставить свое подмножество запрещенных комбинаций. Чтобы эти подмножества не пересекались, кодовое расстояние применяемого кода должно быть не менее 3.

При $n=3$ из множества всех возможных комбинаций 2^n можно выделить только две возможные комбинации с $d=3$, т.е. подмножество разрешенных кодовых комбинаций содержит не более 2-х комбинаций. Каждой из выбранных разрешенных комбинаций в соответствие ставится свое подмножество запрещенных кодовых комбинаций, отличающихся от разрешенной только в одном символе. Например, если в качестве разрешенных выбраны комбинации 000 и 111, то подмножество запрещенных кодовых комбинаций для 000 будет $\{100, 010, 001\}$, а для 111 – $\{011, 101, 110\}$.

Итак, код с $d=3$ способен исправлять однократную ошибку. В общем случае для исправления ошибки кратности S минимальное кодовое расстояние между разрешенными кодовыми комбинациями, т.е. кодовое расстояние кода должно удовлетворять соотношению:

$$d_{min} \geq 2S+1. \qquad (2.21)$$

В точке приема декодирование может производится таким образом, что принятая кодовая комбинация отождествляется с той разрешенной, которая отличается от полученной в наименьшем числе символов. Такое декодирование называется декодированием по методу максимального правдоподобия. Этот метод эффективно используется при обнаружении и исправле-

нии ошибок заданной кратности, однако при исправлении пакетов ошибок более эффективными являются другие методы.

Понятие кодового расстояния и его связь с корректирующей способностью кода впервые установлены Хэммингом, поэтому в литературе кодовое расстояние часто называют хэмминговым расстоянием.

2.3.4. Избыточность корректирующих кодов.

Избыточность является основной характеристикой корректирующих кодов, указывающей степень удлинения кодовой комбинации для достижения определенной корректирующей способности.

Если на каждые n символов последовательности на выходе кодирующего устройства приходится k информационных и $m = n - k$ проверочных, то относительная избыточность кода определяется соотношением:

$$R = \frac{m}{k} \qquad (2.22)$$

Коды, обеспечивающие заданную корректирующую способность при минимально возможной избыточности, называются оптимальными.

В общем виде задача построения оптимальных кодов до сих пор не решена. Кроме того, даже в тех частных случаях, где решение этой задачи известно, применение оптимальных кодов может приводить к существенному усложнению кодирующей и декодирующей аппаратуры, поэтому от применения оптимальных кодов часто отказываются, а используют коды, близкие к оптимальным, но требующие значительно меньше аппаратурных затрат.

Следует иметь в виду, что применение корректирующих кодов не гарантирует безошибочного приема, а только дает возможность повысить вероятность получения правильного результата. Любой корректирующий код предназначен для исправления ошибок, наиболее вероятных для заданного ка-

нала. Если уровень помех в канале будет больше предполагаемого, то эффективность применения этого кода резко снизится.

2.4. Коды, обнаруживающие ошибки.

Код с проверкой на четность.

Указанный код имеет в каждой кодовой комбинации всего один избыточный символ, т.е. $n = k + 1$. Общее число возможных выходных кодовых комбинаций в данном случае равно 2^{k+1}. За подмножество разрешенных кодовых комбинаций можно принять, например подмножество 2^k комбинаций, содержащих четное число единиц. Тогда подмножество оставшихся 2^k комбинаций, содержащее нечетное число единиц, будет представлять собой подмножество запрещенных. При кодировании к каждой последовательности из k информационных символов добавляется один символ (0 или 1), выбираемый таким образом, чтобы общее число единиц в последовательности оказалось четным. В таком случае искажение одного, а также любого нечетного числа символов переводит разрешенную кодовую комбинацию в подмножество запрещенных комбинаций с нечетным числом единиц и, таким образом, ошибка обнаруживается.

Код относится к классу разделимых и систематических. Проверочный символ в данном случае получается в результате суммирования информационных символов по модулю 2.

Пример.

Пусть безизбыточная кодовая комбинация имеет вид 11011010.

Проверочный символ $m = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$, а кодовая комбинация на выходе кодера – 110110101.

Код с числом единиц, кратным трем.

Этот код образуется путем добавления к k информационным символам двух контрольных символов (m = 2), имеющих такие значения, при которых сумма единиц в разрешенных комбинациях кратна трем.

Пример.

k	m	k + m = n
0 0 0 1 1 0	1 0	0 0 0 1 1 0 1 0
1 0 0 0 1 1	0 0	1 0 0 0 1 1 0 0
1 1 0 1 1 0	1 1	1 1 0 1 1 0 1 1

На приемной стороне вес полученных кодовых комбинаций проверяется на кратность 3 (вес кодовой комбинации – число ненулевых символов в комбинации).

Такой код позволяет обнаруживать все одиночные ошибки и любое четное количество ошибок одного типа (например, переход от 0 к 1).

Код относится к классу разделимых и систематических.

Код с постоянным весом.

Код с постоянным весом содержит в комбинациях постоянное число единиц и нулей. Общее число кодовых комбинаций в двоичном коде с постоянным весом равно:

$$N = C_n^{\ell} = \frac{n!}{\ell!(n-\ell)!},$$

где ℓ – число единиц в кодовой комбинации длиною n.

Наиболее часто применяются пятиразрядный код с двумя единицами ($N = C_5^2 = 10$) и семиразрядный код с тремя единицами ($N = C_7^3 = 35$).

Правильность принятых кодовых комбинаций здесь определяется путем подсчета количества единиц и, если это число не соответствует требуемому, делается вывод, что произошла ошибка. Эти коды могут обнаруживать все одиночные ошибки, а также многократные ошибки за исключением слу-

чаев, когда одна единица переходит в нуль, а один из нулей – в единицу (такое двойное искажение называется смещением). Очевидно, что при двойных смешениях ошибки также не обнаруживаются, однако ошибки такой высокой кратности маловероятны, поэтому помехозащищенность таких кодов является достаточно высокой.

Коды с постоянным весом относятся к классу неразделимых.

Корреляционный код.

Корреляционный код по-другому называют кодом с удвоением элементов. Здесь каждый символ двоичного безизбыточного кода кодируется двумя элементами, причем 1 преобразуется в 10, а $0 \rightarrow 01$. Таким образом, вместо комбинации 01101, например, в канал связи передается 01.10.10.01.10. Ошибка обнаруживается, если в парных элементах будут содержаться одинаковые символы, т.е. 11 или 00 вместо 10 и 01. При правильном приеме четные элементы отбрасываются и остается первоначальная комбинация.

Код обладает высокой помехоустойчивостью, так как ошибка не обнаруживается только тогда, когда два рядом стоящих различных элемента, соответствующих одному информационному символу, будут искажены так, что 1 перейдет в 0, а 0 – в 1. Очевидно, что вероятность появления такого события невысокая.

Корреляционный код относится к группе разделимых и систематических.

Комбинированный инверсный код.

В таком коде к исходной k-разрядной комбинации добавляется еще k контрольных символов. Правило образования кода следующее: если в исходной комбинации четное число единиц, то добавляемая комбинация повторяет исход-

ную, если нечетное, то дополнительная комбинация – инвертированная исходная.

Пример.

k	m = k	Инверсный код
1 1 0 0 0 1 0 0 1 0 1 0	0 0 1 1 1 0 0 0 1 0 1 0	1 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0 1 0 0 0 1 0 1 0

Прием комбинированного инверсного кода осуществляется в два этапа.

На 1-ом этапе суммируются единицы в первой (основной) группе символов k. Если число единиц в этой части комбинации четное, то последующие контрольные символы принимаются без изменения; если число символов окажется нечетное, то контрольные символы инвертируются.

На 2-ом этапе контрольные символы m сравниваются с информационными k $(k = m)$ путем суммирования по модулю 2. Если передача прошла без ошибок, то результат суммирования будет нулевым. В противном случае в сумме по модулю 2 появляются единицы, что и является индикатором ошибки.

Корректирующая способность кода очень высокая. Например, если в информационной части исказилось нечетное число разрядов, то ошибка будет не обнаружена только в случае, когда в проверочной части разряды, соответствующие ошибочным информационным, сохраняются неизмененными., а все остальные разряды инвертируются. Если же в информационной части искажается четное число разрядов, то ошибка не обнаруживается, когда будут искажены одноименные разряды и в проверочной части. Вероятность появления таких событий мала.

Рассматриваемый код способен не только обнаруживать многократные ошибки, но и исправлять однократные. При этом индикатором ошибочного разряда в информационной части является 0 при 1 во всех остальных разрядах суммы по $\mathrm{mod}\,2$, а

индикатором ошибочного разряда в проверочной части является 1 при 0 во всех остальных разрядах указанной суммы.

Пример.

Пусть исходная комбинация имеет вид:

$\underbrace{1\,1\,0\,0\,0\,1}_{k}.\underbrace{0\,0\,1\,1\,1\,0}_{m}$

Предположим, что произошла однократная ошибка в информационной части, т.е. принята комбинация

1 1 0 <u>1</u> 0 1 . 0 0 1 1 1 0

После декодирования получаем:

$$\begin{array}{rl} & 1\,1\,0\,1\,0\,1 \quad -\text{чт} \\ \oplus & 0\,0\,1\,1\,1\,0 \\ \hline & 1\,1\,1\,\underline{0}\,1\,1 \end{array}$$

Вывод: ошибка в 4-м разряде информационной части.

Если принята комбинация 1 1 0 0 0 1 . 0 0 1 1 <u>0</u> 0, т.е. произошла ошибка в контрольной части, после декодирования получаем:

$$\begin{array}{rl} & 1\,1\,0\,0\,0\,1 \quad -\text{нч} \\ \oplus & 1\,1\,0\,0\,1\,1 \\ \hline & 0\,0\,0\,0\,\underline{1}\,0 \end{array}$$

Вывод: ошибка в 5-м разряде проверочной части.

Комбинированный инверсный код относится к группе разделимых и систематических.

2.5. Линейные коды, обнаруживающие и исправляющие ошибки.

Линейными называют коды, в которых проверочные символы представляют собой линейные комбинации инфор-

мационных символов, т.е. линейные коды являются систематическими.

Основой математического описания линейных кодов является теория алгебраических систем (теория групп, колец, полей, векторных пространств, матриц).

Кодовые комбинации в линейных кодах рассматриваются как элементы множества.

Алгебраическими системами называют множества, для которых определены некоторые алгебраические операции. Под алгебраической операцией понимается однозначное сопоставление двум элементам третьего элемента по определенным правилам. Обычно такой операцией является сложение ($a + b = c$) или умножение ($a \cdot b = c$).

Рассмотрим кратко такие алгебраические системы, как группа и кольцо.

Группой называется множество элементов (кодовых комбинаций), объединенных определенным законом композиции. Это значит, что задана некоторая операция G, в результате применения которой к любым двум комбинациям группы образуется третья комбинация, также принадлежащая этой группе. Иными словами, группа – это множество кодовых слов, обладающих свойством замкнутости относительно операции G.

Если операция, определенная в группе, коммутативна, т.е. справедливо равенство $a + b = b + a$ (для группы с операцией сложения) или равенство $a \cdot b = b \cdot a$ (для группы с операцией умножения), то группа называется **коммутативной** или абелевой.

Группа, состоящая из конечного числа элементов, называется **конечной.**

Чтобы множество n-разрядных кодовых комбинаций, было конечной группой, при выполнении основной операции число разрядов в результирующей комбинации не должно увеличиваться. Этому условию удовлетворяет операция символического поразрядного сложения по заданному модулю q, при которой цифры одинаковых разрядов элементов группы

(кодовых комбинаций) складываются обычным порядком, а результатом сложения считается остаток от деления полученного числа на модуль q.

Кроме отмеченного, элементы группы должны удовлетворять следующим требованиям:

– для любых трех элементов группы a, b и c должен выполняться закон ассоциативности, т.е. $(a+b)+c=a+(b+c)$ (если основная операция – сложение) и $a\cdot(b\cdot c)=(a\cdot b)\cdot c$ (если основная операция – умножение);

– существует нулевой элемент 0 со свойством $0+a=a$ (в группе операций сложения) и единичный элемент 1 со свойством $1\cdot a=a$ (в группе с операцией умножения);

– для каждого элемента a группы существует противоположный элемент, обозначаемый $-a$, со свойством $-a+a=0$ (для группы по сложению), и обратный элемент, обозначаемый a^{-1}, со свойством $a\cdot a^{-1}=1$ (для группы по умножению).

В теории двоичных кодов рассматриваются аддитивные группы, т.е. группы с операцией сложения (по модулю 2).

Кольцом называется множество элементов R, на котором определены две операции – сложения и умножения, такие, что:

– множество R является коммутативной группой по сложению;

– произведение элементов $a\in R$ и $b\in R$ есть элемент множества R (замкнутость по отношении к умножению);

– для любых трех элементов a, b и c из R справедливо равенство $a\cdot(b\cdot c)=(a\cdot b)\cdot c$ (ассоциативный закон для умножения);

– для любых трех элементов a, b и c из R выполняются соотношения $a\cdot(b+c)=a\cdot b+a\cdot c$ и $(b+c)\cdot a=b\cdot a+c\cdot a$ (дистрибутивные законы).

Более подробно теория рассмотренных и прочих алгебраических систем излагается в работах [1] и [6].

2.5.1. Построение двоичного линейного кода.

Когда речь идет о линейных кодах, кодовые комбинации принято называть кодовыми векторами (КВ).

Линейный код обычно обозначают как $(n, k)-$код, где n – значность КВ, k – число информационных символов. Следовательно, число проверочных (контрольных) символов $m = n - k$.

Построение линейного $(n, k)-$кода начинается с выбора числа информационных разрядов в кодовых векторах. Это число выбирается, исходя из требуемого объема кода Q, т.е. максимального числа сообщений, которое требуется передавать.

В случае передачи двоичным кодом величина k должна удовлетворять неравенству:

$$2^k - 1 \geq Q \qquad (2.23)$$

(единица вычитается из 2^k потому, что нулевая комбинация обычно не используется при передаче, т.к. не изменяет состояния канала).

После выбора k определяется число контрольных разрядов m, необходимое для получения требуемой корректирующей способности кода.

Если требуется исправлять все одиночные ошибки (кодовое расстояние кода $d \geq 3$), величина m выбирается из следующих соображений. Под действием помех может быть искажен любой символ в n-значном КВ, т.е. для каждого кодового вектора возможно $(n+1)$ исходов передачи ($+1$ учитывает правильную передачу). С помощью контрольных символов нужно различать все возможные исходы передачи. Это возможно, если выполняется условие:

$$2^m \geq C_n^1 + 1 = k + m + 1, \qquad (2.24)$$

где C_n^1 – число сочетаний из n по 1.

Уравнение (2.24) является трансцендентным относительно m, поэтому при небольших k величину m определя-

ют простым подбором, принимая максимальное значение m, удовлетворяющие (2.24).

При больших k для определения m при $d = 3$ можно использовать эмпирическое соотношение:

$$m = E\log_2[(k+1) + E\log_2(k+1)], \qquad (2.25)$$

где E – знак округления до ближайшего большего числа.

Если необходимо исправлять не только все единичные, но и все двойные независимые ошибки, величина m должна выбираться в соответствии с условием:

$$2^m \geq C_n^1 + C_n^2 + 1. \qquad (2.26)$$

В общем случае для исправления всех независимых ошибок кратности до S включительно

$$2^m \geq C_n^1 + C_n^2 + \ldots + C_n^S + 1. \qquad (2.27)$$

После определения m составляется образующая матрица, состоящая из k строк и n столбцов. В общем виде образующая матрица имеет вид:

$$G = \left|\begin{matrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{k1} & a_{k2} & a_{k3} & \ldots & a_{kn} \end{matrix}\right| \begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{matrix} \qquad (2.28)$$

Кодовые векторы, входящие в образующую матрицу, являются исходными разрешенными. Поскольку двоичный линейный код является групповым по сложению, остальные разрешенные КВ получаются путем суммирования по модулю 2 строк образующей матрицы сначала попарно, затем по три, наконец, всех k – строк.

В качестве строк образующей матрицы могут быть взяты любые КВ, отвечающие следующим условиям. Они должны быть:

1) n-значными;

2) отстоящими друг от друга на заданное кодовое расстояние;

3) ненулевыми;

4) иметь вес не менее заданного кодового расстояния кода;

5) линейно-независимыми.

КВ матрицы G являются разрешенными, что и объясняет первые два требования.

Третье требование связано с тем, что любой групповой код по сложению имеет нулевой элемент, представляющий собой КВ из одних нулей. При сложении такого КВ с любым другим, результат будет совпадать с вторым слагаемым, что недопустимо, т.к. при этом не выполняется требование замкнутости по операции сложения.

Четвертое требование объясняется тем, что КВ образующей матрицы G являются разрешенными и, следовательно, должны удовлетворять условию (2.21). Кодовое расстояние между двумя любыми КВ равно весу w вектора, полученного в результате их сложения по модулю 2. Но нулевой КВ, не входящий в G, также является разрешенным. Отсюда следует, что вес КВ, входящих в матрицу G, должен быть не менее заданного кодового расстояния кода.

Пятое требование гарантирует, что ни один из векторов образующей матрицы (2.28) не будет результатом суммирования по модулю 2 каких-либо других векторов этой же матрицы (в противном случае не будет получено требуемое число разрешенных КВ).

Линейно-независимыми являются КВ, для которых выполняется неравенство:

$$a_1 v_1 \oplus a_2 v_2 \oplus \ldots \oplus a_k v_k \neq 0, \qquad (2.29)$$

где v_i, $i = \overline{1,k}$ – кодовые векторы; a_i – коэффициенты, принимающие произвольные значения 0 и 1 за исключением $a_1 = a_2 = \ldots = a_i = \ldots = a_k = 0$.

Последним шагом в построении линейного кода является составление проверочной (контрольной) матрицы, имеющей n столбцов и m строк. В общем виде контрольная матрица имеет вид:

$$H = \left|\begin{matrix} \beta_{11} & \beta_{12} & \beta_{13} & \ldots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \beta_{23} & \ldots & \beta_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \beta_{m1} & \beta_{m2} & \beta_{m3} & \ldots & \beta_{mn} \end{matrix}\right| \begin{matrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{matrix} \qquad (2.30)$$

Элементы β, составляющие контрольную матрицу, представляют собой элементы КВ, ортогональных любым разрешенным кодовым векторам. Если обозначить через V-разрешенные КВ данного линейного кода, а через U-вектор контрольной матрицы, то условие ортогональности КВ V и U в математической форме записывается так:

$$V \cdot U = \sum_{i=1}^{n} a_i \beta_i = 0, \qquad (2.31)$$

где a_i и β_i, $i = \overline{1,n}$, – соответственно, элементы разрешенных КВ и векторов контрольной матрицы.

Кроме того, что любой вектор контрольной матрицы должен быть ортогонален любому разрешенному КВ, матрица H в целом должна удовлетворять следующему требованию: в контрольной матрице не должно быть нулевых и одинаковых столбцов.

После построения контрольной матрицы линейный код является полностью определенным. На этапе кодирования КВ формируется так, чтобы он был ортогонален каждому из векторов контрольной матрицы, а на этапе декодирования принятый КВ, возможно содержащий ошибки, проверяется на ортогональность векторам матрицы H.

Пример.

Построить линейный $(n, k)-$код, позволяющий исправлять все одиночные ошибки, если требуемый объем кода $Q = 12$.

Решение.

1. Определяем требуемое число информационных разрядов. Согласно (2.23) имеем: $2^k \geq Q+1$, $2^k \geq 13$, откуда $k = 4$.

2. В соответствии с (2.25) определяем требуемое число контрольных разрядов:

$$m = E\log_2[(k+1)+E\log_2(k+1)] = E\log_2[5+3] = 3, \; m = 3.$$

Следовательно, $n = k + m = 7$, а код имеет формат (7, 4).

3. Составляем образующую матрицу G.

Так как линейный код должен исправлять однократные ошибки, то кодовое расстояние между комбинациями образующей матрицы должно удовлетворять условию (2.21):

$d_{min} \geq 2S+1=3$, S – кратность исправляемой ошибки, $S=1$.

Теперь, учитывая изложенные выше общие требования к КВ образующей матрицы, переходим к построению самой матрицы. Существует несколько подходов к построению G. В самом общем случае при построении G предварительно не определяют места расположения информационных и контрольных символов. Это становится ясным только после построения контрольной матрицы. Такой подход осуществляется в [10]. В некоторых работах предлагается сразу же назначать места расположения контрольных символов. Так, например, в [7] для кода (7.4) предлагается такое построение кодовых векторов:

$m_1 m_2 k_4 m_3 k_3 k_2 k_1$.

По мнению автора такая структура КВ несколько упрощает алгоритм определения контрольных символов.

В дальнейшем при построении G будем следовать методике, изложенной в [10].

Пусть образующая матрица имеет вид:

$$G = \left|\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 1 & 0 & 1 & v_1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & v_2 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & v_3 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & v_4 \end{array}\right|$$

Входящие в нее векторы удовлетворяют всем сформулированным выше условиям.

4. Строим контрольную матрицу H.

Ранее отмечалось, что все КВ контрольной матрицы должны быть ортогональны всем разрешенным комбинациям данного линейного кода.

Кодовые векторы образующей матрицы – разрешенные.

Контрольная матрица H содержит три КВ и, следовательно, для ее построения в соответствии с (2.31) достаточно иметь три разрешенных КВ. Возьмем в качестве таковых векторы v_1, v_2 и v_3 образующей матрицы G.

Определим условия, при которых любой КВ матрицы H будет ортогонален любому из указанных разрешенных КВ.

В соответствии с (2.31) имеем:

$$\begin{array}{ccccccccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & -\ u \\ \times & 0 & 1 & 0 & 0 & 1 & 0 & 1 & -\ v_1 \\ \hline & 0 & \oplus\ \beta_2 & \oplus\ 0 & \oplus\ 0 & \oplus\ \beta_5 & \oplus\ 0 & \oplus\ \beta_7 & = \beta_2 \oplus \beta_5 \oplus \beta_7 = 0 \end{array}$$

$$\begin{array}{ccccccccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & -\ u \\ \times & 0 & 0 & 1 & 0 & 1 & 1 & 0 & -\ v_2 \\ \hline & 0 & \oplus\ 0 & \oplus\ \beta_3 & \oplus\ 0 & \oplus\ \beta_5 & \oplus\ \beta_6 & \oplus\ 0 & = \beta_3 \oplus \beta_5 \oplus \beta_6 = 0 \end{array}$$

$$\begin{array}{ccccccccc} & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & -\ u \\ \times & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -\ v_3 \\ \hline & \beta_1 & \oplus\ 0 & \oplus\ 0 & \oplus\ \beta_4 & \oplus\ \beta_5 & \oplus\ 0 & \oplus\ 0 & = \beta_1 \oplus \beta_4 \oplus \beta_5 = 0 \end{array}$$

Итак, для того, чтобы любой КВ матрицы H был ортогонален любому разрешенному КВ, векторы контрольной матрицы должны удовлетворять следующим условиям:

$$\left.\begin{array}{ll} \beta_2 \oplus \beta_5 \oplus \beta_7 = 0 & (1) \\ \beta_3 \oplus \beta_5 \oplus \beta_6 = 0 & (2) \\ \beta_1 \oplus \beta_4 \oplus \beta_5 = 0 & (3) \end{array}\right\} \qquad (2.32)$$

Теперь необходимо подобрать β_i, $i = \overline{1,7}$, принимающие значения 0 и 1, так, чтобы они удовлетворяли соотношениям (2.32). При этом необходимо следить, чтобы в контрольной матрице не оказалось нулевых и одинаковых столбцов.

Пусть $\beta_2 = \beta_5 = 1$, тогда из (1) следует, что $\beta_7 = 0$.

Если $\beta_3 = 1$, тогда, согласно (2), $\beta_6 = 0$.

При $\beta_1 = 1$, $\beta_4 = 0$, что следует из (3).

Итак, первый КВ контрольной матрицы определен: $U_1 = 1110100$.

Определение 2-го КВ контрольной матрицы осуществляется аналогично.

Пусть $\beta_2 = \beta_7 = 1$, тогда $\beta_5 = 0$

$\beta_3 = \beta_6 = 1$

$\beta_1 = 0$, тогда $\beta_4 = 0$.

Принять $\beta_1 = 1$ было нельзя, так как тогда первые два символа в столбцах 1, 2 и 3 разрядов были бы единичными, а это привело бы в дальнейшем к появлению одинаковых столбцов в контрольной матрице, что недопустимо. Итак, $U_2 = 0110011$.

При выборе 3-го КВ контрольной матрицы нужно ввести ряд ограничений, предотвращающих появление одинаковых и нулевых столбцов. В рассматриваемом случае указанные ограничения имеют вид:

$\beta_2 \neq \beta_3$, $\beta_6 \neq \beta_7$, $\beta_4 = 1$.

Тогда, если $\beta_2 = \beta_6 = 1$, то $\beta_3 = \beta_7 = 0$. $U_3(2) - \beta_5 = 1$, а из $(3) - \beta_1 = 0$. Итак, $U_3 = 0101110$.

Контрольная матрица в окончательном виде:

$$H = \left|\begin{array}{ccccccc|c} 1 & 1 & 1 & 0 & 1 & 0 & 0 & U_1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & U_2 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & U_3 \end{array}\right| \qquad (2.33)$$

2.5.2. Кодирование.

Разрешенные КВ должны быть ортогональны векторам контрольной матрицы. Исходя из этого, найдем условия, которым должны удовлетворять КВ кода, построенного по контрольной матрице (2.33):

$$\begin{array}{ccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & - & v \\ \times\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & - & u_1 \\ \hline a_1 \oplus a_2 \oplus a_3 \oplus 0 \oplus a_5 \oplus 0 \oplus 0 & = & a_1 \oplus a_2 \oplus a_3 \oplus a_5 = 0 & & & & & & \end{array}$$

$$\begin{array}{l} a_1\ \ a_2\ \ a_3\ \ a_4\ \ a_5\ \ a_6\ \ a_7 - v \\ \times\ 0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1 - u_2 \\ \hline 0 \oplus a_2 \oplus a_3 \oplus 0 \oplus 0 \oplus a_6 \oplus a_7 = a_2 \oplus a_3 \oplus a_6 \oplus a_7 = 0 \end{array}$$

$$\begin{array}{l} a_1\ \ a_2\ \ a_3\ \ a_4\ \ a_5\ \ a_6\ \ a_7 - v \\ \times\ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0 - u_3 \\ \hline 0 \oplus a_2 \oplus 0 \oplus a_4 \oplus a_5 \oplus a_6 \oplus 0 = a_2 \oplus a_4 \oplus a_5 \oplus a_6 = 0 \end{array}$$

Итак, любой формируемый КВ должен удовлетворять условиям:

$$\left.\begin{array}{l} a_1 \oplus a_2 \oplus a_3 \oplus a_5 = 0 \quad (1) \\ a_2 \oplus a_3 \oplus a_6 \oplus a_7 = 0 \quad (2) \\ a_2 \oplus a_4 \oplus a_5 \oplus a_6 = 0 \quad (3) \end{array}\right\} \qquad (2.34)$$

Для любой информационной части равенства (2.34) достигаются путем подбора контрольных разрядов, в качестве которых выбираются разряды, встречающиеся только в одной проверке. Таковыми являются 1, 4 и 7 разряды (соответствуют столбцам контрольной матрицы H, содержащим только одну 1). Разряд a_1 обеспечивает ортогональность с вектором U_1; a_4 – с U_3; a_7 – с U_2.

Из системы проверочных равенств (2.34) определяем, какими должны быть проверочные символы при формировании конкретной комбинации безизбыточного кода:

$$\left.\begin{array}{l} a_1 = a_2 \oplus a_3 \oplus a_5 \\ a_4 = a_2 \oplus a_5 \oplus a_6 \\ \underbrace{a_7}_{\text{Контрольные разряды}} = \underbrace{a_2 \oplus a_3 \oplus a_6}_{\text{Информационные разряды}} \end{array}\right\} \qquad (2.35)$$

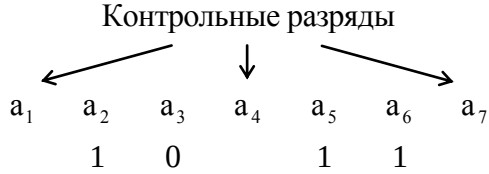
Пример.

Закодировать число 11_{10} линейным кодом, использующим контрольную матрицу (2.33).

Решение.

$11_{10} = 1011_2$.

Начиная со старших разрядов, располагаем полученную безизбыточную двоичную комбинацию на отведенные ей позиции (2, 3, 5 и 6 разряды).



Подставляя значения информационных символов в уравнения (2.35), находим значения проверочных символов:

$$a_1 = 1 \oplus 0 \oplus 1 = 0$$

$$a_4 = 1 \oplus 1 \oplus 1 = 1$$

$$a_7 = 1 \oplus 0 \oplus 1 = 0$$

Подставив полученные значения контрольных разрядов на отведенные позиции, получаем:

a_1	a_2	a_3	a_4	a_5	a_6	a_7
0	1	0	1	1	1	0

2.5.3. Синдромный метод декодирования.

Декодирование линейных кодов, так же как и кодирование, основано на использовании контрольной матрицы H. Наиболее распространенными методами декодирования являются метод проверки на четность и метод синдромного декодирования.

Метод проверки четности основан на использовании проверочных равенств (2.34).

Если все проверочные равенства выполняются, делается вывод, что кодовая комбинация принята без искажений. Если же в принятой комбинации есть ошибка, то по крайней мере одна, а возможно и большее число проверок в (2.34) дадут результат 1.

В таком случае делается вывод, что символы, вошедшие в проверки с нулевым результатом, приняты верно, а ошибочным является символ, входящий одновременно во все проверки, давшие единичный результат.

Недостатком данного метода является необходимость выполнения логических заключений, что усложняет декодирующее устройство.

Метод синдромного декодирования основан на использовании так называемых синдромов ошибок, представляющих собой m-значные двоичные числа, разряды которых являются скалярными произведениями векторов ошибок на векторы контрольной матрицы.

Вектор ошибки представляет собой n-значное двоичное число, содержащее 1 в разряде, где есть ошибка, и 0 во всех остальных разрядах. В качестве примера найдем синдром ошибки в 3-м разряде:

$$\begin{array}{r} 0010000 \\ \times\ 1110100 \\ \hline 0010000 \end{array} \begin{array}{l} -\ \ell_3 \\ -\ u_1 \end{array} \qquad \sum_{i=1}^{7} \ell_{i3} \cdot u_{i1} = 1$$

$$\begin{array}{r} 0010000 \\ \times\ 0110011 \\ \hline 0010000 \end{array} \begin{array}{l} -\ \ell_3 \\ -\ u_2 \end{array} \qquad \sum_{i=1}^{7} \ell_{i3} \cdot u_{i2} = 1$$

$$\begin{array}{r} 0010000 \\ \times\ 0101110 \\ \hline 0000000 \end{array} \begin{array}{l} -\ \ell_3 \\ -\ u_3 \end{array} \qquad \sum_{i=1}^{7} \ell_{i3} \cdot u_{i3} = 0$$

Итак, синдромом ошибки в 3-м разряде является комбинация 110. Действуя аналогично, можно найти синдромы ошибок во всех остальных разрядах. Характерным является то, что синдром ошибки i-го разряда совпадает с i-м столбцом контрольной матрицы, поэтому в запоминающем устройстве декодера достаточно хранить только контрольную матрицу.

При приеме с помощью принятого КВ и векторов контрольной матрицы вычисляется синдром. Если ошибки нет, то синдром будет нулевой. Во всех остальных случаях получается синдром, совпадающий с синдромом ошибки в соответствующем разряде. Это следует из того, что искаженную кодовую комбинацию $\hat{v}_j$ можно представить как сумму по мо-

дулю 2 вектора ошибки ℓ_i и неискаженного КВ v_j, т.е. $\hat{v}_j = v_j \oplus \ell_i$. Умножив искаженный КВ на векторы контрольной матрицы u_k, $k = \overline{1,3}$, получим:

$$\hat{v}_j \cdot u_k = (v_j \oplus \ell_i) \cdot u_k = v_j \cdot u_k \oplus \ell_i \cdot u_k .$$

Но $v_j \cdot u_k = 0$ в силу ортогональности разрешенных КВ и векторов контрольной матрицы. Следовательно, $\hat{v}_j \cdot u_k = \ell_i \cdot u_k$, что и требуется. Таким образом, определив синдром принятой комбинации можно однозначно указать номер разряда, в котором произошла ошибка.

Пример 1.

Из канала связи поступила комбинация $v = 1110010$. Контрольная матрица имеет вид:

$$H = \left|\begin{array}{ccccccc|c} 1 & 0 & 1 & 1 & 1 & 0 & 0 & u_1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & u_2 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & u_3 \end{array}\right| \qquad (2.36)$$

Определить, какое число передавалось.

Решение.

1. Определяем синдром:

$$\begin{array}{r} 1110010 \quad - v \\ \times\ 1011100 \quad - u_1 \\ \hline 1010000 \end{array} \qquad \sum_{i=1}^{7} v \cdot u_1 = 0$$

$$\begin{array}{r} 1110010 \quad - v \\ \times\ 1110010 \quad - u_2 \\ \hline 1110010 \end{array} \qquad \sum_{i=1}^{7} v \cdot u_2 = 0$$

$$\begin{array}{r} 1110010 \quad - v \\ \times\ 0111001 \quad - u_3 \\ \hline 0110000 \end{array} \qquad \sum_{i=1}^{7} v \cdot u_3 = 0$$

Итак, синдром ошибки 000, значит, передача прошла без искажений.

2. По матрице H определяем расположение контрольных разрядов и отбрасываем их:

$$v - 1\ 1\ 1\ 0\ \underbrace{0\ 1\ 0}$$

контрольные разряды, т.к. соответствуют столбцам H с одной 1.

3. Находим десятичный эквивалент:

$\underbrace{1\ 1\ 1\ 0_2} = 14_{10}$

информационная часть принятого КВ.

Пример 2.

Из канала связи поступила комбинация $v = 1010010$. Контрольная матрица та же, что и в примере 1. Определить, какое число передавалось.

Решение.

1. Определяем синдром:

$$\begin{array}{r} 1010010 \quad -\ v \\ \times\ 1011100 \quad -\ u_1 \\ \hline 1010000 \end{array} \qquad \begin{array}{r} 1010010 \quad -\ v \\ \times\ 1110010 \quad -\ u_2 \\ \hline 1010010 \end{array} \qquad \begin{array}{r} 1010010 \quad -\ v \\ \times\ 0111001 \quad -\ u_3 \\ \hline 0010000 \end{array}$$

$$\sum_{i=1}^{7} v \cdot u_1 = 0 \qquad \sum_{i=1}^{7} v \cdot u_2 = 1 \qquad \sum_{i=1}^{7} v \cdot u_3 = 1$$

Итак, синдром ошибки 011. Не нулевой, следовательно, есть ошибка.

2. По контрольной матрице выясняем, что синдром совпадает со вторым столбцом, значит ошибочным является 2-ой разряд принятого КВ.

3. Инвертируем искаженный разряд:

$1\underline{0}10010 \rightarrow 1\underline{1}10010$.

4. Отбрасываем контрольные символы и переводим информационную часть в десятичное число:

$1110_2 = 14_{10}$.

2.5.4. Кодирующее и декодирующее устройства.

Линейные корректирующие коды наиболее часто используются для исправления и обнаружения ошибок в цифровых устройствах обработки и хранения информации, где надежность кодирующих и декодирующих устройств соизмерима с надежностью самого вычислительного канала. В связи

с этим кодирующее и декодирующее устройства здесь должны:

– иметь минимальную сложность;
– иметь минимальные задержки.

В наиболее полной степени эти требования выполняются при параллельной передаче информации, что в таких системах является приемлемым вследствие сравнительно малых расстояний между взаимодействующими блоками. Именно на этот способ передачи информации и ориентированы рассматриваемые ниже устройства. Схемы построены для контрольной матрицы (2.36).

Кодирующее устройство (кодер).

Кодирующее устройство реализует следующие уравнения для выполнения проверочных символов:

$$\left.\begin{aligned} a_5 &= a_1 \oplus a_3 \oplus a_4 \\ a_6 &= a_1 \oplus a_2 \oplus a_3 \\ a_7 &= a_2 \oplus a_3 \oplus a_4 \end{aligned}\right\} \qquad (2.37)$$

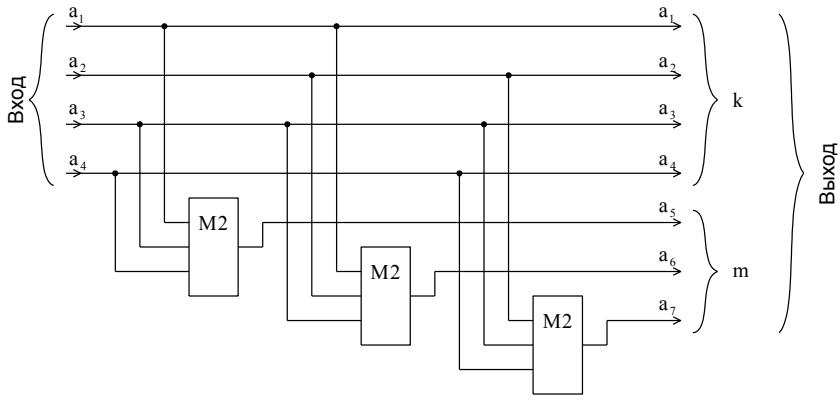


Рис. 2.2. Функциональная схема кодера.

На вход кодера подается безизбыточная k-значная ($k=4$) комбинация. В кодере с помощью сумматоров по модулю 2 (М2) в соответствии с (2.37) вычисляются контрольные символы a_5, a_6, a_7 и присоединяются к информационным. В результате на выходе появляется n-значная ($n=k+m$) комбинация линейного кода. Задержка между моментом появления входной комбинации и окончанием формирования выходной определяется только временем распространения сигнала в одном сумматоре М2 (сумматоры работают параллельно).

Декодирующее устройство (декодер).

Декодер выполняет следующие функции:

– вычисляет синдром ошибки в принятом КВ;
– дешифрирует синдром ошибки;
– инвертирует ошибочный разряд.

Функциональная схема декодера представлена на рис. 2.3.

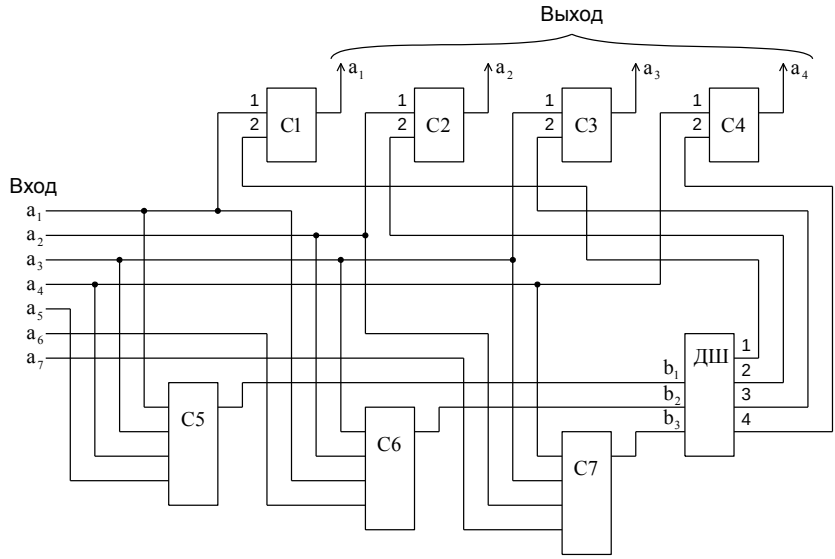


Рис. 2.3. Функциональная схема декодера.

На вход декодера подается подлежащий декодированию КВ.

Схема вычисления синдрома собрана на сумматорах по модулю 2 С5…С7 и реализует следующие уравнения:

$$\left.\begin{aligned} S_1 &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \\ S_2 &= a_1 \oplus a_2 \oplus a_3 \oplus a_6 \\ S_3 &= a_2 \oplus a_3 \oplus a_4 \oplus a_7 \end{aligned}\right\} \qquad (2.38)$$

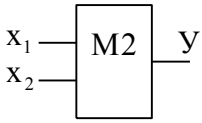
Если ошибка отсутствует, вектор синдрома состоит из одних нулей. На всех выходах дешифратора ДШ при этом будут нули.

При наличии ошибки вектор синдрома будет совпадать с одним из столбцов контрольной матрицы (2.36) и 1 появится только на выходе ДШ, соответствующем ошибочному разряду. В целом состояния выходов ДШ определяются следующей таблицей:

Входы			Выходы			
S_1	S_2	S_3	1	2	3	4
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	1	0	0	0
1	1	1	0	0	1	0

Исправление (инвертирование) искаженного информационного разряда осуществляется с помощью сумматоров по модулю 2 С1…С4, выполняющих функции управляемых ин-

верторов. Возможность такого использования сумматоров видна из таблицы его состояний:



Вход		Выход
x_1	x_2	у
0	0	0
1	0	1
0	1	1
1	1	0

Если рассматривать вход x_2 в качестве управляющего, а x_1 – информационного, то из таблицы видно, что при $x_2 = 0$ информационный сигнал передается на выход без инверсии, а при $x_2 = 1$ – инвертируется. В рассматриваемой схеме декодера роль управляющих сигналов выполняют сигналы, поступающие с выходов ДШ.

Минимальная задержка между моментом появления КВ на входе декодера и моментом, когда исправленная информационная часть КВ может быть передана на обработку, определяется выражением:

$$t_3 = t_1 + t_2 + t_3,$$

где t_1 – время распространения сигнала в 4-х входовом сумматоре (С5…С6); t_2 – время распространения сигнала в ДШ; t_3 – время распространения сигнала в 2-х входовом сумматоре (С1…С4).

2.5.5 Коды Хэмминга.

Все коды Хэмминга являются блочными, равномерными, разделимыми и систематическими. Выбор проверочных символов (процесс кодирования), а также обнаружение и исправление ошибок (процесс декодирования) осуществляются с помощью проверок на четность, т.е. подсчетом сумм по модулю 2.

К кодам Хэмминга относят:

– код с обнаружением однократных ошибок ($d = 2$);

– код с исправлением однократных ошибок ($d = 3$);

– код с исправлением однократных и обнаружением двойных ошибок ($d = 4$).

Код с обнаружением однократной шибки получают добавлением к безизбыточной кодовой комбинации одного проверочного символа (0 или 1), выбираемого из условия четности числа единиц в полной кодовой комбинации. Это обычный код с проверкой на четность, рассмотренный в п. 2.4.

Код Хэмминга, исправляющий однократные ошибки, это линейный групповой код с $d = 3$. Все, что было сказано выше по поводу этих кодов (методика построения, процессы кодирования и декодирования, схемы кодера и декодера), имеет непосредственное отношение к коду Хэмминга с $d = 3$. Согласно выражениям (2.20) и (2.21), данный код способен либо исправлять однократные ошибки, либо обнаруживать двойные. Если декодер работает в режиме исправления ошибки, а произошла двойная ошибка, то результат декодирования будет ошибочен. В связи с этим, если вероятность искажения двух символов в кодовой комбинации велика, то целесообразно применение кода Хэмминга с $d = 4$, позволяющего исправить однократную ошибку, если была только одиночная ошибка, и, кроме того, обнаружить двойную ошибку, если исказились два символа.

Для того чтобы код был способен исправлять ошибки кратности S и одновременно обнаруживать r ошибок, минимальное кодовое расстояние кода должно быть равно:

$$d = S + r + 1,\ r \geq S. \qquad (2.39)$$

В рассматриваемом случае минимальное кодовое расстояние кода должно быть равно $d = 4$. такой код строится на базе кода, исправляющего одиночные ошибки, путем добавления дополнительного контрольного символа к закодированной комбинации, который позволяет произвести проверку на четность всей комбинации. Поэтому контрольный символ должен быть равен единице, если число единиц в исходном КВ нечетное, и нулю, если число единиц четное.

При проверке принятой комбинации возможны следующие варианты:

– ошибок нет; это показывает как общая проверка принятой комбинации на четность, так и частные проверки КВ, в процессе которых дополнительный контрольный символ отбрасывается;

– одиночная ошибка в КВ; общая проверка на четность показывает наличие ошибки, а частные проверки КВ указывают на номер ошибочного разряда;

– искажение дополнительного контрольного символа; общая проверка на четность показывает наличие ошибки, а частные проверки КВ – на ее отсутствие;

– две ошибки; общая проверка на четность указывает на отсутствие ошибок, а частные проверки КВ – на наличие ошибок; в отличие от предыдущих вариантов ошибку здесь исправить нельзя, т.к. невозможно определить место расположения ошибок.

2.5.6. Матричное представление линейных кодов.

В п. 2.5.1 образующая матрица (2.28) составлена путем простого подбора КВ в соответствии с предъявляемыми к ним требованиями. Такое решение задачи приемлемо при небольшом объеме кода, но становится мало пригодном при его существенном увеличении. Соответственно, возникнут трудности и при составлении контрольной матрицы H.

Для упрощения указанных операций групповые коды удобно задавать матрицами, размерность которых определяется параметрами кода k и m. Число строк матрицы равно k, число столбцов равно $n = k + m$.

Теорией и практикой установлено, что для упрощения процесса кодирования наиболее удобно, чтобы порождающая матрица $\underline{M}_{n,k}$ состояла из двух матриц: единичной матрицы размерности $k \times k$ и дописываемой справа матрицы-дополнения

(контрольной подматрицы) размерности $k \times m$, которая соответствует m проверочным разрядам:

$$\underline{M}n,k = \left| \underline{I}k \;\; \underline{P}m,k \right| \qquad (2.40)$$

Единичной матрицей I называется квадратная матрица, у которой по одной из диагоналей расположены только единицы, а все остальные элементы равны нулю. Учитывая это, запишем порождающую матрицу $\underline{M}n,k$ в виде:

$$\underline{M}n,k = \begin{array}{c} \begin{array}{ccccccccccc} a_1 & a_2 & a_3 & \dots & a_k & p_1 & p_2 & p_3 & \dots & p_m \end{array} \\ \left| \begin{array}{ccccccccccc} 1 & 0 & 0 & \dots & 0 & p_{11} & p_{12} & p_{13} & \dots & p_{1m} \\ 0 & 1 & 0 & \dots & 0 & p_{21} & p_{22} & p_{23} & \dots & p_{2m} \\ 0 & 0 & 1 & \dots & 0 & p_{31} & p_{32} & p_{33} & \dots & p_{3m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & p_{k1} & p_{k2} & p_{k3} & \dots & p_{km} \end{array} \right| \end{array} \qquad (2.41)$$

Характерной особенностью данной матрицы является то, что в порождаемых ею разрешенных кодовых комбинациях первые k символов совпадают с исходными информационными, а последующие m символов оказываются проверочными. Это утверждение следует из того, что разрешенные КВ, соответствующие произвольной комбинации $\underline{A}k_i$ из k информационных символов, получают путем умножения вектора $\underline{A}k_i$ на порождающую матрицу $\underline{M}n,k$:

$$\underline{A}n_i = \underline{A}k_i \underline{M}n,k . \qquad (2.42)$$

В общем случае, при перемножении матрицы $\underline{A} = \left| a_{ij} \right|$ размерности $\ell \times n$ и матрицы $\underline{B} = \left| b_{jk} \right|$ размерности $n \times m$, элементами C_{jk} матрицы-произведения размерности $\ell \times m$ являются суммы произведений элементов i-й строки матрицы $\underline{A}$ на соответствующие элементы k-го столбца матрицы $\underline{B}$:

$$C_{jk} = \sum_{j=1}^{n} a_{ij} b_{jk} . \qquad (2.43)$$

Пусть $\underline{A}ki = (a_1, a_2, \dots, a_k)$. Умножив вектор-строку $\underline{A}ki$ на матрицу $\underline{M}n,k$, получим вектор $\underline{A}ni = (a_1, a_2, \dots, a_k, a_{k+1}, \dots, a_n)$,

где проверочные символы a_j $(k+1 \le j \le n)$ являются линейными комбинациями информационных, определяемыми в соответствии с выражением:

$$a_j = \sum_{i=1}^{k} a_i p_{ij} . \qquad (2.44)$$

Таким образом, при использовании порождающей матрицы (2.41) формирование разрешенных КВ линейного кода сводится к добавлению к k-значной информационной комбинации m контрольных символов, определяемых в соответствии с выражением (2.44).

При составлении матрицы $\underline{P}_{m,k}$ необходимо учитывать, что для обеспечения требуемого кодового расстояния кода вес каждой строки матрицы $\underline{P}_{m,k}$ должен быть не менее $W_{\underline{P}_{m,k}} = d - W_{\underline{I}_k}$, где d – требуемое кодовое расстояние кода; $W_{\underline{I}_k}$ – вес соответствующей строки матрицы $\underline{I}_k$ ($W_{\underline{I}_k} = 1$). Если бы в матрице (2.41) левая половина была бы не единичной матрицей, а имела вес $W > 1$, это усложнило бы как построение кода, так и техническую реализацию кодера и декодера.

Дополнительно следует иметь в виду, что чем больше вес строк матрицы $\underline{P}_{m,k}$, тем ближе порождаемый код к совершенному (плотноупакованному, оптимальному), обеспечивающему максимально возможную корректирующую способность при заданной избыточности. В связи с этим рекомендуется из всех возможных m-значных кодовых комбинаций в качестве строк матрицы $\underline{P}_{m,k}$ выбирать комбинации, обладающие наибольшим весом. Это обеспечивает требуемое кодовое расстояние между КВ матрицы (2.41), а использование в (2.41) единичной матрицы $\underline{I}_k$ гарантирует их линейную независимость.

Пример.

Построить порождающую матрицу линейного группового кода, исправляющего однократную ошибку (d = 3), если требуемый объем кода $Q = 100$.

Решение.

Для передачи 100 сообщений необходимо соблюдать неравенство (2.23):

$2^k \geq 100+1$, откуда $k=7$.

В соответствии с (2.25) требуемое число контрольных разрядов:

$$m = E\log_2[(k+1)+E\log_2(k+1)] = E\log_2[8+3] = E\log_2 11 = 4$$

Таким образом, число строк порождающей матрицы $k=7$, число столбцов $n=k+m=11$.

Учитывая вышеизложенное, записываем порождающую матрицу $\underline{M}_{11,7}$ в виде:

$$\underline{M}_{11,7} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{vmatrix} \qquad (2.45)$$

По известной матрице $\underline{M}_{n,k}$ контрольная матрица $\underline{H}$ определяется в соответствии с выражением:

$$\underline{H} = \left|\underline{P}^T_{m,k}\, \underline{I}_m\right|, \qquad (2.46)$$

где $\underline{P}^T_{m,k}$ – транспонированная матрица $\underline{P}_{m,k}$ (в транспонированной матрице строками являются столбцы, а столбцами – строки исходной матрицы); $\underline{I}_m$ – единичная матрица размерности $m \times m$.

Для порождающей матрицы (2.45) контрольная матрица $\underline{H}$ имеет вид:

$$H = \begin{array}{c} \begin{matrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} \end{matrix} \\ \begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \end{array} \qquad (2.47)$$

В соответствии с (2.47) контрольные символы определяются выражениями:

$$\left.\begin{aligned} a_8 &= a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_6 \\ a_9 &= a_1 \oplus a_2 \oplus a_3 \oplus a_5 \oplus a_7 \\ a_{10} &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7 \\ a_{11} &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \end{aligned}\right\} \qquad (2.48)$$

Элементы синдрома вычисляются по уравнениям:

$$\left.\begin{aligned} S_1 &= a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_8 \\ S_2 &= a_1 \oplus a_2 \oplus a_3 \oplus a_5 \oplus a_7 \oplus a_9 \\ S_3 &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7 \oplus a_{10} \\ S_4 &= a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{11} \end{aligned}\right\} \qquad (2.49)$$

Пример.

Закодировать число 83_{10} линейным кодом, использующим контрольную матрицу (2.47).

Решение.

$$\begin{array}{lccccccc} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ 83_{10} = & 1 & 0 & 1 & 0 & 0 & 1 & 1_2 \end{array}$$

В соответствии с (2.48) получаем:

$$a_8 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$a_9 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$a_{10} = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$a_{11} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1$$

Кодовый вектор, отображающий заданное число:

$v = 1010011.1101.$

Предположим, что при передаче найденного КВ произошло искажение 4-го разряда, т.е. принятый КВ имеет вид:

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
1	0	1	$\underline{1}$	0	1	1	1	1	0	1

По (2.49) находим синдром:

$$S_1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$S_2 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$S_3 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$S_4 = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

Найденный синдром 1011 совпадает с 4-м столбцом контрольной матрицы (2.47), следовательно, декодирование выполнено верно.

Построение контрольной матрицы $\underline{H}$ для кодов с большей корректирующей способностью осуществляется аналогично.

Пример.

Построить контрольную матрицу линейного кода, исправляющего одиночные и двойные ошибки. Требуемое число информационных разрядов $k = 8$.

Решение.

Для построения порождающей матрицы необходимо определить число контрольных разрядов. При заданной корректирующей способности кода величина m определяется по (2.26), т.е. $2^m \geq C_n^1 + C_n^2 + 1$. Расчеты показывают, что при $k = 8$ минимальное значение m, при котором выполняется неравенство (2.26), равно 7. Следовательно, $n = k + m = 8 + 7 = 15$. В соответствии с вышеизложенным, матрица $\underline{M}_{15,8}$ имеет вид:

$$\underline{M}_{15,8} = \underbrace{\begin{vmatrix} 1&0&0&0&0&0&0&0&1&1&1&1&1&1&1 \\ 0&1&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&1&0&0&0&0&0&1&0&1&1&1&1&1 \\ 0&0&0&1&0&0&0&0&1&1&0&1&1&1&1 \\ 0&0&0&0&1&0&0&0&1&1&1&0&1&1&1 \\ 0&0&0&0&0&1&0&0&1&1&1&1&0&1&1 \\ 0&0&0&0&0&0&1&0&1&1&1&1&1&0&1 \\ 0&0&0&0&0&0&0&1&1&1&1&1&1&1&0 \end{vmatrix}}_{\underline{I}_8 \qquad\qquad \underline{P}_{7,8}} \qquad (2.50)$$

При построении матрицы $\underline{M}_{15,8}$ учтено, что для получения требуемой корректирующей способности кодовое расстояние кода, согласно (2.21), $d \geq 5$. Следовательно, вес векторов подматрицы $\underline{P}_{7,8}$ должен быть не менее 4.

Контрольная матрица $\underline{H}$ имеет вид:

$$\underline{H} = \begin{array}{c} \begin{array}{ccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{array} \\ \left| \begin{array}{ccccccccccccccc} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right| \\ \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\underline{P}^{T}_{7,8}} \qquad \underbrace{\qquad\qquad\qquad\qquad\qquad}_{\underline{I}_7} \end{array} \quad \begin{array}{l} \text{номера} \\ \text{разрядов} \end{array} \qquad (2.51)$$

В соответствии с (2.51) составляем выражения для контрольных разрядов.

Обозначим $\sum_{i=1}^{8} a_i = b$. Тогда получаем:

$$\left.\begin{array}{ll} a_9 = b \oplus a_2 & a_{13} = b \oplus a_6 \\ a_{10} = b \oplus a_3 & a_{14} = b \oplus a_7 \\ a_{11} = b \oplus a_4 & a_{15} = b \oplus a_8 \\ a_{12} = b \oplus a_5 & \end{array}\right\} \qquad (2.52)$$

Элементы синдрома вычисляются по уравнениям:

$$\left.\begin{array}{ll} S_1 = b \oplus a_2 \oplus a_9 & S_5 = b \oplus a_6 \oplus a_{13} \\ S_2 = b \oplus a_3 \oplus a_{10} & S_6 = b \oplus a_7 \oplus a_{14} \\ S_3 = b \oplus a_4 \oplus a_{11} & S_7 = b \oplus a_8 \oplus a_{15} \\ S_4 = b \oplus a_5 \oplus a_{12} & \end{array}\right\} \qquad (2.53)$$

При однократных ошибках синдром совпадает с соответствующим столбцом контрольной матрицы, а синдром двойной ошибки равен сумме по модулю 2 синдромов ошибочных разрядов. Очевидно, что при использовании матрицы (2.51) синдромы одиночных и двойных ошибок не совпадают, что позволяет однозначно исправлять как одиночные, так и двойные ошибки.

Рассмотрим примеры.

Пример 1.

Закодировать по (2.51) безизбыточный КВ

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	1	0	1	0	0	1	0

Решение.

Определяем $b = \sum_{i=1}^{8} a_i = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$.

Определяем по (2.52) контрольные разряды:

$a_9 = 0 \oplus 1 = 1 \qquad a_{13} = 0 \oplus 0 = 0$

$a_{10} = 0 \oplus 0 = 0 \qquad a_{14} = 0 \oplus 1 = 1$

$a_{11} = 0 \oplus 1 = 1 \qquad a_{15} = 0 \oplus 0 = 0$

$a_{12} = 0 \oplus 0 = 0$

Полный КВ имеет вид:

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
$v=1$	1	0	1	0	0	1	0	1	0	1	0	0	1	0	

Пример 2.

При передаче вектора v произошло искажение 5-го разряда, т.е. принятый КВ имеет вид:

1 1 0 1 $\underline{1}$ 0 1 0 . 1 0 1 0 0 1 0

Выполнить декодирование.

Решение.

Определяем по принятому КВ $b = \sum_{i=1}^{8} a_i = 1$. Определяем по (2.53) синдром:

$$S_1 = 1 \oplus 1 \oplus 1 = 1$$

$$S_2 = 1 \oplus 0 \oplus 0 = 1$$

$$S_3 = 1 \oplus 1 \oplus 1 = 1$$

$$S_4 = 1 \oplus 1 \oplus 0 = 0$$

$$S_5 = 1 \oplus 0 \oplus 0 = 1$$

$$S_6 = 1 \oplus 1 \oplus 1 = 1$$

$$S_7 = 1 \oplus 0 \oplus 0 = 1$$

Найденный синдром совпадает с 5-м столбцом матрицы (2.51), что подтверждает правильность декодирования.

Пример 3.

При передаче КВ произошло искажение 5-го и 7-го разрядов, т.е. принятый КВ имеет вид:

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8		a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
1	1	0	1	$\underline{1}$	0	$\underline{0}$	0	.	1	0	1	0	0	1	0

Выполнить декодирование.

Решение.

По принятому КВ определяем $b = \sum_{i=1}^{8} a_i = 0$. Определяем синдром:

$$S_1 = 0 \oplus 1 \oplus 1 = 0$$

$$S_2 = 0 \oplus 0 \oplus 0 = 0$$

$$S_3 = 0 \oplus 1 \oplus 1 = 0$$

$$S_4 = 0 \oplus 1 \oplus 0 = 1$$

$$S_5 = 0 \oplus 0 \oplus 0 = 0$$

$$S_6 = 0 \oplus 0 \oplus 1 = 1$$

$$S_7 = 0 \oplus 0 \oplus 0 = 0$$

Полученный синдром не нулевой, следовательно, есть ошибка. В контрольной матрице (2.51) таких столбцов нет, значит произошла двойная ошибка.

Проверка:

$$\begin{array}{rcl} & 1\ 1\ 1\ 0\ 1\ 1\ 1 & \text{– синдром ошибки 5-го разряда} \\ \oplus & 1\ 1\ 1\ 1\ 1\ 0\ 1 & \text{– синдром ошибки 7-го разряда} \\ \hline & 0\ 0\ 0\ 1\ 0\ 1\ 0 & \text{– синдром ошибки 5-го и 7-го разрядов} \end{array}$$

Вывод – декодирование выполнено верно.

При практической реализации синдромного метода декодирования все возможные синдромы ошибок нужно заранее рассчитать и хранить в каком-нибудь запоминающем устройстве, например, в ПЗУ. В последнем случае на адресные входы ПЗУ нужно подавать синдром, а матрица ПЗУ должна быть закодирована так, чтобы 1 появлялись только на выходах, соответствующих ошибочным разрядам (при этом совмещаются функции ЗУ и ДШ).

Исправление ошибок осуществляется с помощью сумматоров по модулю 2 (см. п. 2.5.4).

2.5.7. Мажоритарное декодирование.

Для линейных кодов, рассчитанных на исправление многократных ошибок, часто более простыми оказываются декодирующие устройства, построенные по мажоритарному принципу. Этот метод декодирования называют также принципом голосования.

Метод мажоритарного декодирования основывается на составлении для каждого информационного символа системы $2\lambda S$, где S – кратность исправляемой ошибки, λ – связанных нетривиальных проверок. В качестве уравнений проверок используются линейные комбинации строк контрольной матрицы. Под системой λ-связанных проверок для символа a_i понимают систему уравнений, удовлетворяющую двум условиям:

– символ a_i входит в каждое уравнение проверки;

– любой символ a_j, $j \neq i$, входит не более чем в λ уравнений проверок.

Если $\lambda = 1$, то проверки называют раздельными. Под системой раздельных проверок для символа a_i понимают систему уравнений, удовлетворяющих двум условиям:

– символ a_i входит в каждое уравнение проверок;

– любой символ a_j, $j \neq i$, входит не более чем в одно уравнение проверки.

Кроме нетривиальных проверок в систему проверок может включаться тривиальная проверка $a_i = a_i$.

Результаты вычислений каждого из уравнений, входящих в систему проверок символа a_i, подаются на соответствующий этому символу мажоритарный элемент. Последний представляет собой схему, имеющую $2\lambda S + 1$ входов (S – кратность исправляемой ошибки) и один выход, на котором появляется 1, когда на входах число 1 больше половины, и 0 – в противном случае.

Пример 1.

Разработать систему проверок для мажоритарного декодирования кода (7,4), исправляющего однократную ошибку. Контрольная матрица кода имеет вид:

$$H = \begin{vmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

Решение.

Кодирующее устройство реализует следующие уравнения:

$a_5 = a_1 \oplus a_2 \oplus a_3$

$a_6 = a_2 \oplus a_3 \oplus a_4$

$a_7 = a_1 \oplus a_2 \oplus a_4$

Декодирующее устройство реализует систему уравнений проверок для каждого информационного символа a_i, $i = \overline{1,4}$. Система уравнений проверок составляется путем сложения по модулю 2 строк матрицы H. Полученная система проверок имеет вид:

$a_1 = a_1$	$a_2 = a_2$	$a_3 = a_3$	$a_4 = a_4$
$a_1 = a_5 \oplus a_6 \oplus a_4$	$a_2 = a_5 \oplus a_6 \oplus a_7$	$a_3 = a_5 \oplus a_1 \oplus a_2$	$a_4 = a_5 \oplus a_6 \oplus a_1$
$a_1 = a_5 \oplus a_2 \oplus a_3$	$a_2 = a_5 \oplus a_1 \oplus a_3$	$a_3 = a_5 \oplus a_7 \oplus a_4$	$a_4 = a_6 \oplus a_2 \oplus a_3$
$a_1 = a_6 \oplus a_7 \oplus a_3$	$a_2 = a_6 \oplus a_3 \oplus a_4$	$a_3 = a_6 \oplus a_2 \oplus a_4$	$a_4 = a_7 \oplus a_1 \oplus a_2$
$a_1 = a_7 \oplus a_2 \oplus a_4$	$a_2 = a_7 \oplus a_1 \oplus a_4$	$a_3 = a_6 \oplus a_7 \oplus a_1$	$a_4 = a_5 \oplus a_7 \oplus a_3$

Можно видеть, что некоторые символы входят в систему уравнений 2 раза. Это означает, что связность проверок $\lambda = 2$. Требуемое число нетривиальных проверок и тривиальной проверки $2\lambda S + 1 = 2 \cdot 2 \cdot 1 + 1 = 5$ выполнено.

Пример 2.

Произвести декодирование по разработанной системе проверок кодового вектора

a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	1	0	1	0	0	1

$a_1 = 1$	$a_2 = 1$	$a_3 = 0$	$a_4 = 1$
$a_1 = 0 \oplus 0 + 1 = 1$	$a_2 = 0 \oplus 0 \oplus 1 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 0 \oplus 0 \oplus 1 = 1$
$a_1 = 0 \oplus 1 + 0 = 1$	$a_2 = 0 \oplus 1 \oplus 0 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 0 \oplus 1 \oplus 0 = 1$
$a_1 = 0 \oplus 1 \oplus 0 = 1$	$a_2 = 0 \oplus 0 \oplus 1 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 1 \oplus 1 \oplus 1 = 1$
$a_1 = 1 \oplus 1 \oplus 1 = 1$	$a_2 = 1 \oplus 1 \oplus 1 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 0 \oplus 1 \oplus 0 = 1$
Выходы МЭ: $a_1 = 1$	$a_2 = 1$	$a_3 = 0$	$a_4 = 1$

Полученная в результате декодирования комбинация совпадает с информационной частью принятого КВ, все проверки дают правильный результат, следовательно, передача прошла без искажений.

Предположим, что при передаче произошло искажение 2-го разряда, т.е. принятый КВ имеет вид:

a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	0	0	1	0	0	1

$a_1 = 1$	$a_2 = 0$	$a_3 = 0$	$a_4 = 1$
$a_1 = 0 \oplus 0 \oplus 1 = 1$	$a_2 = 0 \oplus 0 \oplus 1 = 1$	$a_3 = 0 \oplus 1 \oplus 0 = 1$	$a_4 = 0 \oplus 0 \oplus 1 = 1$
$a_1 = 0 \oplus 0 \oplus 0 = 0$	$a_2 = 0 \oplus 1 \oplus 0 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 0 \oplus 0 \oplus 0 = 0$
$a_1 = 0 \oplus 1 \oplus 0 = 1$	$a_2 = 0 \oplus 0 \oplus 1 = 1$	$a_3 = 0 \oplus 0 \oplus 1 = 1$	$a_4 = 1 \oplus 1 \oplus 0 = 0$
$a_1 = 1 \oplus 0 \oplus 1 = 0$	$a_2 = 1 \oplus 1 \oplus 1 = 1$	$a_3 = 0 \oplus 1 \oplus 1 = 0$	$a_4 = 0 \oplus 1 \oplus 0 = 1$
Выходы МЭ: $a_1 = 1$	$a_2 = \underline{1}$	$a_3 = 0$	$a_4 = 1$

Итак, однократная ошибка исправлена.

Появление ошибок более высокой кратности приводит к ложному декодированию. Например, при одновременном искажении 2-го и 3-го символов искаженная информационная часть подтверждается декодированием (ошибка была бы зафиксирована в проверочных разрядах, если бы выполнялась и их проверка), но не все проверки дают правильный результат. Последнее может являться сигналом ошибочного декодирования.

В работе [9] отмечается, что любой линейный код допускает мажоритарное декодирование, однако сложность декодирующего устройства может быть достаточно велика за счет многошаговой процедуры вычисления разрядов кода. Поэтому для сокращения объема декодирующего устройства целесообразно использовать коды с раздельными независимыми проверками, реализующими одноступенчатое мажоритарное декодирование, т.е. содержат на пути прохождения сигналов в декодирующем устройстве один мажоритарный элемент.

2.6. Циклические коды.

Групповой год называется циклическим, если все КВ, составляющие образующую матрицу, могут быть получены циклическим сдвигом одной образующей комбинации кода.

При циклическом сдвиге все символы кодовой комбинации перемещаются справа налево на одну позицию, причем крайний левый символ каждый раз переносится в конец комбинации.

Запишем, например, образующую (производящую) матрицу G, получающуюся при циклическом сдвиге комбинации 0 0 1 0 1 1:

$$G = \left|\begin{array}{cccccc|c} 0 & 0 & 1 & 0 & 1 & 1 & v_1 \\ 0 & 1 & 0 & 1 & 1 & 0 & v_2 \\ 1 & 0 & 1 & 1 & 0 & 0 & v_3 \\ 0 & 1 & 1 & 0 & 0 & 1 & v_4 \\ 1 & 1 & 0 & 0 & 1 & 0 & v_5 \\ 1 & 0 & 0 & 1 & 0 & 1 & v_6 \end{array}\right| \qquad (2.54)$$

КВ, входящие в образующую матрицу G, являются разрешенными. Остальные разрешенные КВ получаются путем суммирования по модулю 2 всех возможных комбинаций КВ, входящих в образующую матрицу G.

При описании циклических кодов n-разрядные кодовые комбинации представляются в виде многочленов фиктивной переменной x, в которых коэффициентами при переменной x являются цифры 0 и 1, составляющие КВ. Например, КВ $v_1 = 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1$ в виде многочлена представляется так:

$$v_1(x) = 0 \cdot x^5 + 0 \cdot x^1 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0$$

Члены с нулевыми коэффициентами при записи опускаются, т.е. $v_1(x)$ имеет вид:

$$v_1(x) = x^3 + x + 1$$

В общем случае, если число элементов КВ равно n, соответствующий ему многочлен имеет вид:

$$v(x) = a_n \cdot x^{n-1} + a_{n-1} \cdot x^{n-2} + \ldots + a_1 \cdot x^0, \qquad (2.55)$$

где a_n, a_{n-1},…, a_1 – коэффициенты, принимающие значения 0 и 1.

Наибольшая степень x в слагаемом с ненулевым коэффициентом называется степенью многочлена.

Замена кодовых комбинаций многочленами позволяет заменить действия над КВ действиями над многочленами, с которыми можно производить все обычные алгебраические операции, за исключением операций сложения и умножения.

Теория циклических кодов базируется на теории групп и теории колец, где определены операции символического сложения и умножения.

Сложение, как уже отмечалось, должно осуществляться по модулю 2.

Операция символического умножения задается так:

– многочлены перемножаются по обычным правилам, но с приведением подобных членов по модулю 2;

– если старшая степень произведения не превышает $n-1$ (см. 2.5.5), то оно и является результатом символического умножения;

– если старшая степень произведения больше или равна n, то многочлен произведения делится на многочлен x^n+1 (сам остаток в этом случае называется вычетом).

Учитывая изложенные особенности операций сложения и умножения многочленов, правила циклического сдвига КВ образующей матрицы можно сформулировать следующим образом.

Циклический сдвиг КВ с нулем в старшем разряде (слева) равносилен умножению многочлена, отображающего этот КВ, на x с одновременным вычитанием из результата многочлена x^n+1.

Пусть исходным является вектор v_3 матрицы (2.54). Тогда, согласно сформулированному правилу, $v_4(x)=v_3(x)\cdot x-$ $-(x^6+1)=(x^5+x^3+x^2)\cdot x-(x^6+1)=x^6+x^4+x^3-x^6-1=x^4+$ $+x^3+1$ (знак перед 1 заменен с "–" на "+" потому, что "–" здесь не имеет смысла). $v_4=0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1$.

В случаях, когда степень произведения равна n, операция вычитания из произведения многочлена x^n+1 эквивалентна делению этого произведения на x^n+1, что предусмотрено общим правилом символического умножения.

За разрешенные кодовые комбинации в циклических кодах принимаются комбинации, которые делятся без остатка на некоторый заранее выбранный образующий многочлен.

При декодировании принятый КВ делится на образующий многочлен. Если принятый КВ имеет ошибку, то деление производится с остатком. Сам факт появления остатка свиде-

тельствует об ошибке, а анализ остатка позволяет локализовать и исправить ошибку.

Из сказанного следует, что образующий многочлен в циклических кодах выполняет такую же роль, что и контрольная матрица в рассмотренных выше линейных кодах.

2.6.1. Выбор образующего многочлена.

Многочлены бывают приводимыми и неприводимыми. Многочлен, который можно представить в виде произведения многочленов низших степеней, называют приводимым, в противном случае – неприводимым. В табл. 2.4 в качестве примера указаны все неприводимые многочлены до шестой степени включительно (на практике используются многочлены и более высоких степеней).

Таблица 2.4. Неприводимые многочлены и их двоичные эквиваленты.

Многочлен	Эквивалент	Многочлен	Эквивалент
$x+1$	11	$x^5+x^4+x^3+x+1$	1110111
x^2+x+1	111	$x^5+x^4+x^3+x^2+1$	111101
x^3+x+1	1011	x^6+x+1	1000011
x^3+x^2+1	1101	x^6+x^3+1	1001001
x^4+x+1	10011	$x^6+x^4+x^2+x+1$	1010111
x^4+x^3+1	11001	$x^6+x^4+x^3+x+1$	1011011
$x^4+x^3+x^2+x+1$	11111	x^6+x^5+1	1100001
x^5+x^2+1	100101	$x^6+x^5+x^2+x+1$	1100111
x^5+x^3+1	101001	$x^6+x^5+x^3+x^2+1$	1101101
$x^5+x^3+x^2+x+1$	101111	$x^6+x^5+x^4+x+1$	1110011
$x^5+x^4+x^2+x+1$	110111	$x^6+x^5+x^4+x^2+1$	1110101

Многочлен в поле двоичных чисел называется неприводимым, если он делится без остатка только на себя и на единицу. Приведенные в таблице 2.4 многочлены являются неприводимыми только для конечного поля двоичных чисел.

В основу циклического кодирования положено использование неприводимых многочленов, которые применительно к циклическим кодам называют образующими.

Обозначим образующий многочлен через $g(x)$.

Все КВ образующей матрицы (2.54) являются разрешенными и, следовательно, $g(x)$ нужно выбирать так, чтобы все КВ этой матрицы делились на $g(x)$ без остатка. Выясним условие, при котором это возможно.

Учитывая изложенные выше правила циклического сдвига КВ, любой КВ образующей матрицы можно представить в виде:

$$v_i(x) = v_{i-1}(x) \cdot x - C(x^n + 1), \qquad (2.56)$$

где C – коэффициент, принимающий значение 1, если степень $v_{i-1}(x) \cdot x$ равна n, и $C = 0$, если степень $v_{i-1}(x) \cdot x$ меньше n; $i = \overline{2, n}$.

Если $v_i(x)$ делится на $g(x)$ без остатка, то, как видно из (2.56), все многочлены матрицы будут делиться на $g(x)$ без остатка только в том случае, если на $g(x)$ будет делиться без остатка многочлен $x^n + 1$. Но многочлен $x^n + 1$ при $n \geq 2$ является приводимым, т.е. всегда может быть представлен в виде произведения некоторых неприводимых многочленов. Отсюда следует, что образующий многочлен $g(x)$ должен выбираться из разложения многочлена $x^n + 1$.

Выбор конкретного образующего многочлена осуществляется в соответствии с требуемой корректирующей способностью кода.

Рассмотрим сначала случай простейшего циклического кода, обнаруживающего все одиночные ошибки.

Любая принятая комбинация $\hat{v}(x)$, возможно, содержащая ошибку, может быть представлена в виде суммы по модулю 2 неискаженного КВ $v(x)$ и вектора ошибки $\ell_i(x)$, $i=\overline{1,n}$:

$$\hat{v}(x)=v(x)\oplus\ell_i(x).$$

Очевидно, что ошибка будет обнаружена только в том случае, если многочлен $\ell_i(x)$, соответствующий вектору ошибки, не делится на образующий многочлен $g(x)$.

Вектор одиночной ошибки имеет 1 в искаженном разряде и 0 во всех остальных разрядах, т.е. ему соответствует многочлен $\ell_i(x)=x^{i-1}$. Среди неприводимых многочленов, входящих в разложение x^n+1 и не входящих в разложение x^{i-1}, многочленом наименьшей степени является $x+1$. Остаток от деления любого многочлена на $x+1$ представляет собой многочлен нулевой степени и может принимать только два значения: 0 или 1. Так как $x+1\to 11$, то очевидно, что при четном числе единиц в исходном КВ остаток от деления многочлена, отображающего этот КВ, на $x+1$ будет 0, а при нечетном – 1. Следовательно, при любом числе информационных разрядов k требуется только один проверочный символ, обеспечивающий четность числа единиц в полной комбинации. Таким образом, циклический код с обнаружением одиночной ошибки является обычным кодом с проверкой на четность и способен обнаруживать не только одиночные, но и любое нечетное число ошибок.

Циклические коды с $d=2$ не имеют практического значения. В двоичных кодах всегда проще подобрать контрольный символ 0 или 1 таким образом, чтобы сумма единиц в КВ была четной, чем строить циклический код для получения того же результата.

В случаях, когда циклический код должен не только обнаруживать, но и исправлять одиночные ошибки, выбор $g(x)$ осуществляется из следующих соображений. Так как информацию об искаженном разряде несет только остаток от деле-

ния векторов ошибок на образующий многочлен, то $g(x)$ должен обеспечивать требуемое число различных остатков. Но при делении любого многочлена на образующий многочлен степени m наибольшая степень остатка равна $m-1$, т.е. остаток содержит m разрядов. Следовательно, при двоичном кодировании и степени образующего многочлена равной m возможно получение 2^m различных остатков, с помощью которых нужно различить ошибку в любом из разрядов n-значной кодовой комбинации, включая и правильную передачу. Следовательно, должно выполняться условие:

$$2^m \geq n+1 \qquad (2.57)$$

Отсюда $m \geq \log_2(n+1)$.

По известному m по таблицам выбирается конкретный многочлен. При этом необходимо, чтобы образующий многочлен степени m, выбранный из разложения многочлена x^n+1, не входил бы одновременно в разложение многочлена $x^\lambda+1$, где $\lambda < n$, т.к. число остатков в этом случае окажется равным λ. В связи с этим, после выбора конкретного многочлена, проверяют число различных остатков, для чего комбинацию в виде 1 с последующими нулями делят на образующий многочлен $g(x)$.

Итак, для кодов с $d=3$ образующий многочлен должен выбираться из числа неприводимых многочленов, входящих в разложение многочлена x^n+1, иметь степень не ниже определяемой выражением (2.57), быть по возможности более коротким, так как при этом упрощаются кодирующие и декодирующие устройства, но с числом ненулевых членов не менее заданного кодового расстояния кода.

2.6.2. Формирование разрешенных кодовых комбинаций.

Наиболее просто комбинации циклического кода можно получить, умножая многочлены $a(x)$, соответствующие комбинациям безизбыточного кода, на образующий многочлен

$g(x)$. Такой способ легко реализуется, но код при этом получается неразделимым. Применительно к циклическим кодам принято отводить под информационные символы k старших разрядов, а под проверочные $m = n - k$ младших разрядов. Чтобы получить такой разделимый код, применяется следующая процедура кодирования.

Многочлен $a(x)$, соответствующий k-разрядной комбинации безизбыточного кода, умножается на x^m, где $m = n - k$. Затем произведение $a(x) \cdot x^m$ делится на образующий многочлен $g(x)$. В общем случае при этом получается некоторое частное $q(x)$ и остаток $r(x)$. Последний складывается по модулю 2 с $a(x) \cdot x^m$ и в результате получается многочлен

$$v(x) = a(x) \cdot x^m \oplus r(x). \qquad (2.58)$$

Полученный таким образом многочлен $v(x)$ делится на образующий многочлен $g(x)$ без остатка. Действительно, многочлен $a(x) \cdot x^m$ можно записать в виде:

$$a(x) \cdot x^m = g(x) \cdot q(x) \oplus r(x). \qquad (2.59)$$

Так как операции сложения и вычитания по модулю 2 тождественны, то $r(x)$ из правой части равенства (2.59) можно перенести в левую. Тогда

$$a(x) \cdot x^m \oplus r(x) = g(x) \cdot q(x),$$

что и доказывает делимость $v(x)$ на $g(x)$ без остатка.

В комбинации $a(x) \cdot x^m$ m младших разрядов – нулевые, следовательно, разрешенные КВ циклического кода можно строить путем приписывания к комбинации безизбыточного кода $a(x)$ остатка от деления многочлена $a(x) \cdot x^m$ на образующий многочлен кода.

Пример.

Закодировать циклическим кодом, исправляющим однократную ошибку, комбинацию 1001.

Решение.

Согласно заданию: $a(x) \to 1001$, $k = 4$, требуемое кодовое расстояние кода $d \geq 3$.

Для того чтобы код был способен исправлять однократную ошибку, степень образующего многочлена m должна удовлетворять условию:

$$2^m \geq n + 1 = m + k + 1.$$

Получаем: $m = 3$, $n = m + k = 3 + 4 = 7$.

Из табл. 2.4 выбираем неприводимый многочлен степени $m = 3$ и числом ненулевых членов, равным 3 ($d = 3$):

$$g(x) = x^3 + x + 1 \to 1011.$$

Определим число различных остатков:

№ остатка		
	$\oplus$ 1 0 0 0 0 …… 1 0 1 1	
	$\oplus$ 1 0 1 1	
1	0 1 1	
2	1 1 0	
	$\oplus$ 1 1 0 0	
	$\oplus$ 1 0 1 1	
3	1 1 1	
	$\oplus$ 1 1 1 0	
	$\oplus$ 1 0 1 1	
4	1 0 1	
	$\oplus$ 1 0 1 0	
	$\oplus$ 1 0 1 1	
5	0 0 1	
6	0 1 0	
7	1 0 0	
	1 0 0 0	

В дальнейшем остатки повторяются.

Количество различных остатков равно 7, следовательно, выбранный образующий многочлен входит в разложение многочлена x^7+1 и не входит в разложение $x^{\lambda}+1$, где $\lambda<7$, что и требуется.

Согласно (2.58), для определения комбинации циклического кода, соответствующей безизбыточной комбинации $a(x)\to 1001$, необходимо найти остаток $r(x)$ от деления $a(x)\cdot x^3\to 1001000$ на образующий многочлен $g(x)\to 1011$ и сложить его по модулю 2 с $a(x)\cdot x^3$. Имеем:

$$
\begin{array}{llllllll}
 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\oplus & 1 & 0 & 1 & 1 & & & \\
\hline
 & & 1 & 0 & 0 & 0 & & \\
 & \oplus & 1 & 0 & 1 & 1 & & \\
\hline
 & & & & 1 & 1 & 0 &
\end{array}
\qquad
\begin{array}{|cccc}
\hline
1 & 0 & 1 & 1 \\
\end{array}
$$

$r(x)\to 110$, а искомая комбинация циклического кода – 1001110.

Разрешенные кодовые комбинации должны делиться на образующий многочлен без остатка. Проверим:

$$
\begin{array}{llllllll}
 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
\oplus & 1 & 0 & 1 & 1 & & & \\
\hline
 & & 1 & 0 & 1 & 1 & & \\
 & \oplus & 1 & 0 & 1 & 1 & & \\
\hline
 & & & & 0 & 0 & 0 &
\end{array}
\qquad
\begin{array}{|cccc}
\hline
1 & 0 & 1 & 1 \\
\end{array}
$$

Вывод: кодирование выполнено правильно.

2.6.3. Декодирование циклических кодов.

Декодирование циклических кодов проводится путем деления принятого КВ на образующий многочлен. Если ошибки нет, то деление выполняется без остатка. Появление остатка сигнализирует об ошибке, исправление которой осуществляется в следующей последовательности.

Подсчитывается вес остатка W. Если он равен или меньше кратности исправляемых ошибок, т.е. $W \le S$, то принятый КВ складывают по модулю 2 с остатком и получают исправленный КВ.

Если $W > S$, то производится циклический сдвиг на один символ влево и полученный после такого сдвига КВ снова делится на образующий многочлен. Если вес полученного остатка $W \le S$, то циклически сдвинутую комбинацию складывают с остатком и полученный КВ циклически сдвигают в обратную сторону. В результате на исходной позиции получают исправленный КВ.

Если после первого сдвига и последующего деления на $g(x)$ вновь оказывается $W > S$, делается еще один сдвиг влево, снова проверяют остаток и так до тех пор, пока не окажется $W \le S$. После сложения сдвинутой комбинации с остатком осуществляется ее сдвиг в обратную сторону на столько шагов, на сколько их было сделано до получения требуемого остатка.

Пример.

Предположим, что при передаче рассмотренного выше КВ $v(x) \to 1001110$ произошло искажение 2-го разряда, т.е. принятый КВ $\hat{v}(x)$ имеет вид: $1\underline{1}01110$. Определим истинный вид поступившего КВ.

Решение.

Согласно правилу, необходимо делить поступивший КВ на образующий многочлен и оценивать вес остатка. Выполняем:

$$
\begin{array}{rcccccccc}
 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & \big| \; 1\;0\;1\;1 \\
\oplus & 1 & 0 & 1 & 1 & & & & \\
\hline
 & & 1 & 1 & 0 & 1 & & & \\
\oplus & & 1 & 0 & 1 & 1 & & & \\
\hline
 & & & 1 & 1 & 0 & 1 & & \\
\oplus & & & 1 & 0 & 1 & 1 & & \\
\hline
 & & & & 1 & 1 & 0 & 0 & \\
\oplus & & & & 1 & 0 & 1 & 1 & \\
\hline
 & & & & & 1 & 1 & 1 &
\end{array}
$$

$r(x) \to 111$. Остаток не нулевой, следовательно, есть ошибка.

Вес $(W = 3) > (S = 1)$ – нужно осуществить циклический сдвиг принятой кодовой комбинации влево и повторить деление:

1) $1\ 1\ 0\ 1\ 1\ 1\ 0 \to$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 1\ 0\ 1 \\ \oplus\ 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 1 \end{array} \quad \Big| \overline{1\ 0\ 1\ 1}$$

$r(x) \to 101$. Опять $(W = 2) > (S = 1)$, поэтому выполняем еще один циклический сдвиг влево и повторяем деление:

2) $1\ 0\ 1\ 1\ 1\ 0\ 1 \to$

$$\begin{array}{r} 0\ 1\ 1\ 1\ 0\ 1\ 1 \\ \oplus\ 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 1\ 1 \\ 1\ 0\ 1\ 1 \\ \hline 0\ 0\ 1 \end{array} \quad \Big| \overline{1\ 0\ 1\ 1}$$

$r(x) \to 001$. Здесь $(W = 1) = (S = 1)$, поэтому складываем сдвинутую кодовую комбинацию с остатком по модулю 2:

$$\begin{array}{r} 0\ 1\ 1\ 1\ 0\ 1\ 1 \\ \oplus\ 0\ 0\ 1 \\ \hline 0\ 1\ 1\ 1\ 0\ 1\ 0 \end{array}$$

Полученную после суммирования комбинацию циклически сдвигаем вправо на 2 такта:

$0\ 1\ 1\ 1\ 0\ 1\ 0$ – 1-ый сдвиг вправо
$0\ 0\ 1\ 1\ 1\ 0\ 1$ – 2-ой сдвиг вправо
$1\ 0\ 0\ 1\ 1\ 1\ 0$

Исправленная кодовая комбинация – 1001110.

2.6.4. Циклические коды с $d > 3$.

Циклические коды с d = 4. Эти коды могут обнаруживать одиночные, двойные и тройные ошибки или, в случае применения мажоритарного декодирования, обнаруживать двойные и исправлять одиночные ошибки.

Степень образующего многочлена и, соответственно, число контрольных разрядов в этом коде должно быть на единицу больше, чем для кода с $d > 3$:

$$m_{d=4} = m_{d=3} + 1. \qquad (2.60)$$

Образующий многочлен $g(x)_{d=4}$ равен произведению двучлена $x+1$ на многочлен $g(x)_{d=3}$:

$$g(x)_{d=4} = (x+1)\cdot g(x)_{d=3} \qquad (2.61)$$

Это объясняется тем, что двучлен $x+1$ позволяет обнаружить все одиночные и тройные ошибки, а многочлен $g(x)_{d=3}$ – двойные ошибки. Так, для кода (7,3), обнаруживающего все тройные ошибки, можно выбрать $g(x)_{d=4} = (x+1)(x^3 + x + 1)$.

Дальнейшая процедура кодирования остается такой же, как и в кодах с $d = 3$.

Пример.

Требуется закодировать сообщение 100111011001 циклическим кодом с $d = 4$.

Решение.

Определяем степень образующего многочлена для $d = 3$ по (2.25):

$$m_{d=3} = E\log_2\left[(k+1) + E\log_2(k+1)\right] =$$

$$= E\log_2\left[(12+1) + E\log_2(12+1)\right] = E\log_2[13+4] = 5$$

Выбираем из таблицы 2.4 образующий многочлен $g(x)_{d=3} = x^5 + x^3 + 1$. Тогда $g(x)_{d=4} = (x+1)(x^5 + x^3 + 1) = x^6 + x^5 + x^4 + x^3 + x + 1 \rightarrow 1111011$, $m = 6$.

$$a(x)\cdot x^6 \rightarrow \underbrace{100111011001}_{k}\underbrace{000000}_{m}$$

Разделив $a(x)\cdot x^6$ на $g(x)_{d=4}$, находим остаток:

$$r(x)\rightarrow 100101.$$

Закодированная комбинация имеет вид:

$v(x)\rightarrow 100111011001.100101$, код (18,12).

Обнаружение ошибок производится по остаткам от деления принятой комбинации $\hat{v}(x)$ на образующий многочлен $g(x)_{d=4}$. Если остаток нулевой, то контрольные разряды отбрасываются, а информационная часть используется по назначению. Если в результате деления получается ненулевой остаток, то комбинация бракуется.

Пример.

Пусть в рассмотренной выше комбинации кода (18,12) при передаче исказились 5, 7 и 11 разряды. Произведем декодирование.

$$\begin{array}{l}
\hat{v}(x)\rightarrow\ 1\ 0\ 0\ 1\ \underline{0}\ 1\ \underline{1}\ 1\ 1\ 0\ \underline{1}\ 1\ 1\ 0\ 0\ 1\ 0\ 1 \,\big|\, \underline{1\ 1\ 1\ 1\ 0\ 1\ 1} \\
\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\ \underline{\quad 1\ 1\ 1\ 0\ 0\ 0\ 1} \\
\quad\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\quad\ \underline{\quad\ 1\ 1\ 0\ 1\ 0\ 1\ 0} \\
\quad\quad\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\quad\quad\ \underline{\quad\ 1\ 0\ 0\ 0\ 1\ 1\ 1} \\
\quad\quad\quad\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\quad\quad\quad\ \underline{\quad\ 1\ 1\ 1\ 1\ 0\ 0\ 1} \\
\quad\quad\quad\quad\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\quad\quad\quad\quad\ \underline{\quad\ 1\ 0\ 0\ 0\ 1\ 0\ 1} \\
\quad\quad\quad\quad\quad\quad \oplus\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\quad\quad\quad\quad\quad\quad\ \underline{\quad\ 1\ 1\ 1\ 1\ 1\ 1\ 0} \\
r(x)\rightarrow\quad\quad\quad\quad\quad\quad\quad\ 1\ 1\ 1\ 1\ 1\ 0
\end{array}$$

Остаток ненулевой, поэтому комбинация бракуется.

Процесс исправления однократной ошибки и одновременного обнаружения двойной будет рассмотрен в п. 2.6.6 на примере кода (7,3).

Циклические коды с $d \geq 5$ позволяют обнаруживать и исправлять любое число ошибок. В литературе эти коды известны как коды БЧХ (первые буквы фамилий Боуз, Чоудхури, Хоквинхем – разработчиков методики построения этих кодов). Построение кодов БЧХ отличается от построения циклических кодов с $d < 5$ только выбором образующего многочлена. Заданными при кодировании здесь являются кратность исправляемой ошибки S и длина кодового слова n (на величину n при этом накладывается ряд ограничений). Числа информационных символов k и контрольных символов m подлежат определению.

Методика построения кодов БЧХ рассмотрена, например, в работах [6] и [8]. Декодирование кодов БЧХ производится по той же методике, что и декодирование циклических кодов с $d < 5$.

Коды БЧХ, предназначенные для исправления взаимно независимых ошибок, могут использоваться также для обнаружения и исправления пакетов ошибок. Однако более эффективными при решении этих задач являются специализированные коды, например, коды Файра [6] и коды Рида-Соломона [1]. При исправлении пакетов ошибок эти коды имеют значительно меньшую избыточность, чем коды БЧХ.

2.6.5. Кодирующее и декодирующее устройства.

Основой кодера и декодера циклических кодов является схема деления многочлена на многочлен. В кодирующем устройстве находится остаток от деления многочлена $a(x) \cdot x^m$ на образующий многочлен $g(x)$, а в декодирующем устройстве находится остаток от деления принятого КВ на тот же образующий многочлен.

Существует два принципиально различных подхода к построению схем деления. Первый реализует последовательный метод деления и создается на базе регистров сдвига с обратными связями. Второй реализует параллельный метод деления и строится только на сумматорах по модулю два. Рассмотрим эти методы.

Схема деления на регистре сдвига с сумматорами по модулю 2.

Существуют различные варианты построения таких делителей, отличающихся числом ячеек памяти регистра (m или k) и схемой включения сумматоров в цепи обратной связи.

Рассмотрим схему деления на основе m-разрядного регистра сдвига. Строится эта схема по следующему правилу:

– число ячеек регистра равно степени образующего многочлена; ячейка регистра для старшей степени многочлена отсутствует, но всегда присутствует ячейка X^0;

– сумматоры ставятся перед ячейками регистра, соответствующими ненулевым членам образующего многочлена, при этом сумматор, соответствующий старшему члену образующего многочлена, отбрасывается;

– делимое, начиная со старшего разряда, поступает на вход первого сумматора, соответствующего члену X^0;

– выход последней ячейки, соответствующей X^{m-1}, соединен со вторыми входами всех сумматоров.

Составленная в соответствии с этими правилами схема деления на образующий многочлен $g(x) = x^3 + x + 1$ представлена на рис. 2.4.

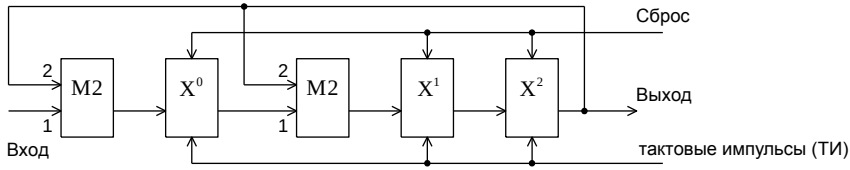


Рис. 2.4. Схема деления на базе регистра сдвига.

Чтобы проследить работу схемы, составим таблицу состояний ячеек регистра.

Таблица 2.5.

№ такта	Вход	Состояние ячеек регистра		
		X^0	X^1	X^2
Сброс		0	0	0
I	1	1	0	0
II	0	0	1	0
III	0	0	0	1
IV	1	0	1	0
V	0	0	0	1
VI	0	1	1	0
VII	0	0	1	1

Характерной особенностью построения регистра является то, что ячейки памяти имеют двухступенчатую структуру. Прием информации в первую ступень осуществляется по пе-

реднему фронту ТИ, а передача информации из первой ступени во вторую – по заднему фронту.

В таблице 2.5 символы 0 и1 характеризуют состояние второй ступени ячеек памяти, устанавливающееся после прохождения заднего фронта ТИ.

Перед началом деления все ячейки регистра устанавливаются в нулевое состояние под действием импульса «Сброс».

За первые m тактов ($m = 3$) коэффициенты многочлена-делимого, подаваемые на вход схемы деления, заполняют регистр, причем коэффициент при x в старшей степени появляется на выходе ячейки X^2. В следующем такте 1 с выхода ячейки X^2 по цепи обратной связи подается на вторые входы сумматоров по модулю 2, что равносильно вычитанию образующего многочлена из многочлена-делимого. Если после окончания предыдущего такта на выходе ячейки X^2, соответствующей старшей степени остатка, устанавливается 0, то в следующем такте образующий многочлен не вычитается. Коэффициенты делимого просто смещаются вперед по регистру на один разряд, что находится в полном соответствии с тем, как это делается при делении многочленов столбиком. После окончания
n-го такта ($n = 7$) в ячейках регистра хранятся коэффициенты остатка $r(x)$ от деления многочлена-делимого на образующий многочлен.

Схема деления на сумматорах по модулю 2.

В этом варианте КВ, отображающий многочлен-делимое, передается на схему деления параллельно. Если КВ передается поэлементно, его необходимо записать в обычный (без обратных связей) n-разрядный регистр сдвига и передать на схему деления с выходов этого регистра. Схема деления на образующий многочлен $g(x) = x^3 + x + 1$ представлена на рис. 2.5.

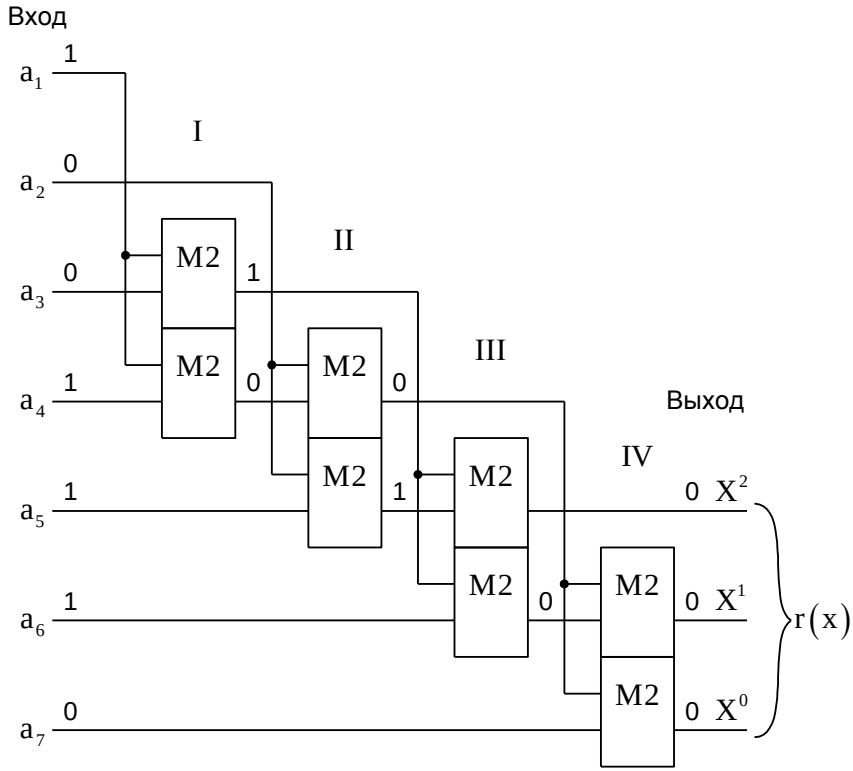


Рис. 2.5. Схема деления на сумматорах.

Деление осуществляется в соответствии с классической схемой деления многочлена на многочлен столбиком. В каждой тетраде (образующий многочлен – четырехзначный) устанавливаются сумматоры по модулю 2 в разрядах, соответствующих ненулевым членам $g(x)$, при этом сумматор для старшей степени $g(x)$ отсутствует. Старший разряд тетрады является управляющим для сумматоров данной ступени деления. Если он равен 1, то из делимого вычитается образующий многочлен, если старший разряд тетрады равен 0, тогда три младших разряда тетрады передаются на следующую ступень деления в неизменном виде. Требуемое число ступе-

ней деления равно k. Задержка появления $r(x)$ относительно момента поступления КВ на вход схемы деления определяется временем распространения сигнала в трех сумматорах по модулю 2.

Кодирующее устройство с делителем на базе регистра сдвига.

Упрощенная функциональная схема кодера для циклического кода (7,4) с $g(x) = x^3 + x + 1$ представлена на рис. 2.6.

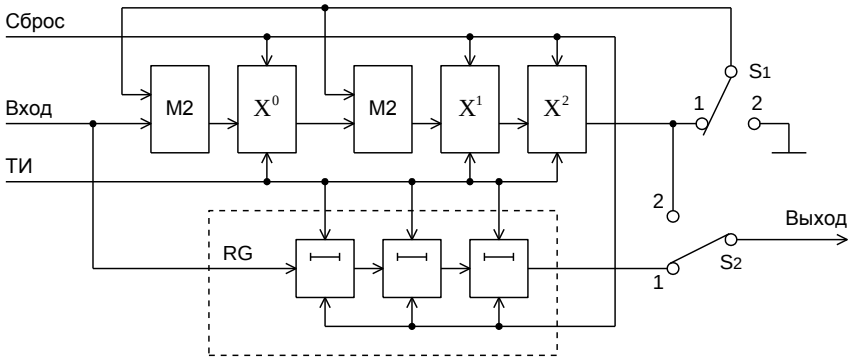


Рис. 2.6. Функциональная схема кодера с делителем на базе регистра сдвига.

Основой кодера является схема деления, представленная на рис. 2.4. Дополнительно введены: m-разрядный регистр сдвига RG, а также переключатели S_1 и S_2. В реальных условиях указанные переключатели являются электронными, поэтому требуется еще не показанное на схеме устройство управления переключателями. Регистр сдвига RG введен для устранения в выходных комбинациях разрыва в m тактов между информационными и контрольными символами, имеющего место в схеме деления рис. 2.4.

Схема работает следующим образом. В исходном состоянии переключатели S_1 и S_2 установлены в положение «1».

Под действием импульса «Сброс» ячейки памяти обоих регистров устанавливаются в нулевое состояние. Затем на вход кодера посимвольно поступает комбинация, отображающая многочлен $a(x)\cdot x^m$. Через m тактов на выходе RG появляется первый информационный символ и за последующие k тактов информационная часть КВ с выхода RG через переключатель S_2 передается на выход кодера. К этому же моменту времени заканчивается деление и в ячейках X^2, X^1 и X^0 находится остаток от деления многочлена $a(x)\cdot x^m$ на образующий многочлен. Для его присоединения к информационной части переключатель S_2 через n тактов после начала деления устанавливается в положение «2». Одновременно переключатель S_1 устанавливается в положение «2», чтобы не исказить остаток при его выводе из регистра схемы деления за счет обратной связи. Задержка появления первого символа КВ на выходе кодера относительно момента появления первого символа на входе в данной схеме составляет m тактов.

Кодирующее устройство с делителем на сумматорах.

Функциональная схема кодера даже проще схемы деления, представленной на рис. 2.5. Упрощение состоит в том, что при кодировании $a_5 = a_6 = a_7 = 0$, поэтому не требуются сумматоры по модулю 2, связанные с этими входами. Указанные сумматоры заменяются перемычками, соединяющими вторые входы сумматоров с их выходами.

Функциональная схема кодера с делителями на сумматорах по модулю 2 представлена на рис. 2.7.

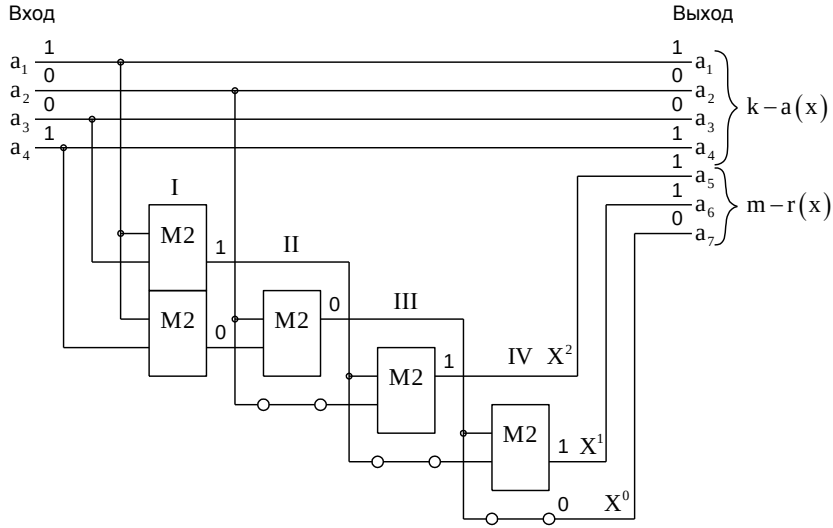


Рис. 2.7. Функциональная схема кодера с делителем на сумматорах.

На вход кодера подаются только информационные разряды. Задержка появления $r(x)$ относительно момента поступления информационной части КВ на вход кодера определяется временем распространения сигнала в трех сумматорах по модулю 2.

Сравнивая рассмотренный вариант кодера с предыдущим вариантом, можно видеть, что кодер с делителем на сумматорах имеет значительно более высокое быстродействие и при небольших значениях k проще в реализации, чем кодер с делителем на регистре сдвига с обратными связями. В то же время в реальных системах передачи информации, например, в сетях ЭВМ, размер передаваемых блоков может составлять тысячи бит. В этих условиях альтернативы последовательному методу деления нет.

Декодирующее устройство с делителем
на базе регистра сдвига.

Декодирование циклического кода заключается в делении принятой комбинации на образующий многочлен. Следовательно, декодер, прежде всего, должен включать в себя схему деления того или иного типа. Кроме того, декодер должен содержать блок памяти для хранения информационных разрядов принятой кодовой комбинации. Последнее связано с тем, что решение о пригодности поступившего КВ для использования по назначению может быть принято только после завершения процесса декодирования, а при последовательном методе декодирования поступивший КВ полностью разрушается.

Функциональная схема декодера с последовательным методом декодирования представлена на рис. 2.8.

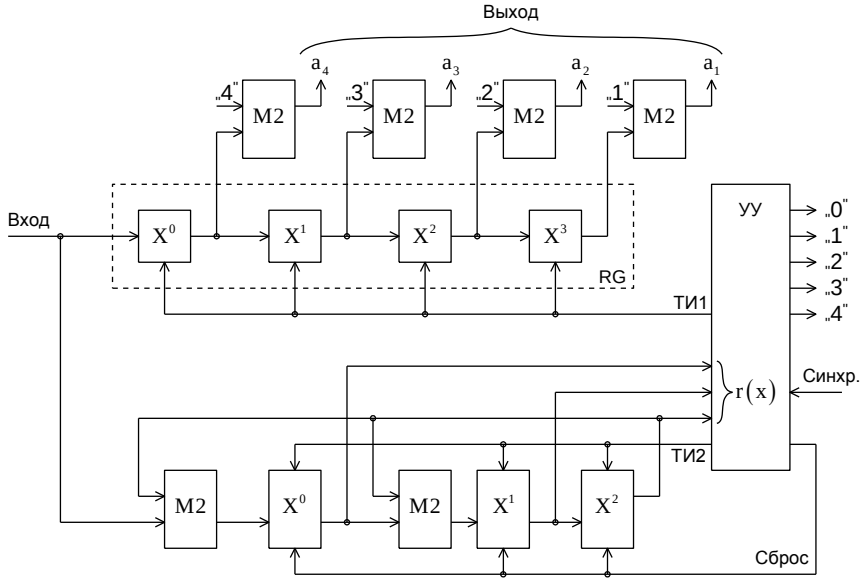


Рис. 2.8. Функциональная схема декодера.

Схема содержит: регистр сдвига RG для хранения информационной части поступающего КВ; сумматоры М2, подключенные к выходам RG и предназначенные для исправления ошибок; схему деления на образующий многочлен $g(x) = x^3 + x + 1$; устройство управления УУ.

Схема декодера работает следующим образом. Под действием сигнала синхронизации «Синхр.», поступающего на декодер извне, устройство управления УУ вырабатывает импульс «Сброс», устанавливающий ячейки памяти схемы деления в нулевое состояние. Одновременно на выходах УУ появляются тактовые импульсные последовательности ТИ1 и ТИ2. Подлежащий декодированию КВ, начиная со старших разрядов информационной части, поступает одновременно на вход приемного регистра RG и схему деления на образующий многочлен. Через k тактов импульсная последовательность ТИ1 на выходе УУ исчезает и регистр RG переходит в режим хранения информационной части КВ, а ТИ2 сохраняется и деление продолжается еще в течение m тактов. По истечении n тактов устройство управления анализирует остаток $r(x)$, поступающий на УУ с ячеек памяти схемы деления. Если в принятой комбинации ошибка отсутствует, то $r(x)$ будет нулевым и на выходе УУ «0» появляется 1, сигнализирующая о том, что с выходов $a_1 \ldots a_4$ комбинация может быть передана по назначению. Если $r(x)$ отличен от нуля, тогда деление продолжается, а в устройстве управления запускается счетчик дополнительных тактов. Деление продолжается до тех пор, пока в ячейках памяти схемы деления не появится комбинация 100 ($x^0 \to 1$). Номер дополнительного такта, после которого будет получен такой результат, укажет на номер ошибочного разряда. Состояние счетчика дополнительных тактов дешифрируется и если ошибка произошла в информационном разряде, на соответствующем выходе из множества «1»… «4» появляется 1, происходит инвертирование ошибочного разряда и с небольшой задержкой появляется 1 на выходе «0». Если

же ошибка произошла в контрольном разряде, то 1 появляется сразу на выходе «0».

Предположим, что при передаче КВ 1001110 произошло искажение 2-го разряда, т.е. принят КВ 1$\underline{1}$01110. Из табл. 2.6 видно, что остаток 100 получен после 2-го дополнительного такта, что и указывает на ошибку во 2-м разряде.

Таблица 2.6.

№ такта	Вход	Состояние ячеек регистра		
		X^0	X^1	X^2
Сброс	0	0	0	0
I	1	1	0	0
II	1	1	1	0
III	0	0	1	1
IV	1	0	1	1
V	1	0	1	1
VI	1	0	1	1
VII	0	1	1	1
1	0	1	0	1
2	0	1	0	0

Данный алгоритм декодирования полностью соответствует алгоритму, изложенному в п. 2.6.3. Поясним это соответствие.

Подлежащий декодированию КВ, возможно содержащий ошибку, представляется в виде:

$$\hat{v}(x) = v(x) \oplus \ell_i(x),$$

где $v(x)$ – многочлен, отображающий разрешенный КВ, а $\ell_i(x)$ – вектор ошибки i-го разряда, $i = \overline{1,n}$. При делении $\hat{v}(x)$ на образующий многочлен всегда получается остаток от деления $\ell_i(x)$ на $g(x)$. Найдем эти остатки для $g(x) = x^3 + x + 1$.

Результаты расчетов сведены в табл. 2.7.

Таблица 2.7.

i	$\ell_i(x)$	$\frac{r(x)}{X^2\ X^1\ X^0}$
1	1 0 0 0 0 0 0	1 0 1
2	0 1 0 0 0 0 0	1 1 1
3	0 0 1 0 0 0 0	1 1 0
4	0 0 0 1 0 0 0	0 1 1
5	0 0 0 0 1 0 0	1 0 0
6	0 0 0 0 0 1 0	0 1 0
7	0 0 0 0 0 0 1	0 0 1

Теперь выясним, какие остатки будут получаться при дополнительных тактах деления. Предположим, что после VII основного такта деления в ячейках регистра получен остаток $X^0\ X^1\ X^2 \to 1\,0\,0$. Остаток не нулевой, поэтому нужны дополнительные такты деления до получения такого же остатка. Из табл. 2.8 видно, что требуемый результат будет получен через

7 дополнительных тактов. В общем случае, если после VII такта будет получен остаток, соответствующий ошибке в i-м разряде, то, согласно табл. 2.8, требуется i дополнительных тактов, чтобы получить остаток $X^0\ X^1\ X^2 \to 1\,0\,0$.

Таблица 2.8.

№ дополн. такта	Делимое	Состояние ячеек регистра			i
		X^0	X^1	X^2	
		1	0	0	7
1	0 →	0	1	0	6
2	0 →	0	0	1	5
3	0 →	1	1	0	4
4	0 →	0	1	1	3
5	0 →	1	1	1	2
6	0 →	1	0	1	1
7	0 →	1	0	0	7

Согласно алгоритму п. 2.6.3, после окончания деления оценивается вес остатка W. Если $W=1$, то он складывается по модулю 2 с принятым КВ и получают исправленный КВ. Из табл. 2.7 видно, что $W=1$, если ошибка произошла в одном из контрольных разрядов (a_5, a_6, a_7). При сложении такого остатка с КВ единица остатка оказывается под ошибочным разрядом, что и требуется. Если же ошибка произошла в одном из информационных разрядов, то $W=2$, поэтому осуществляются циклические сдвиги принятого КВ влево до тех

пор, пока не окажется $W=1$. Но при циклическом сдвиге влево номер ошибочного разряда понижается и первый же остаток с $W=1$ будет соответствовать остатку при $i=7$. Количество необходимых сдвигов влево при этом совпадает с числом дополнительных тактов в рассмотренной схеме делителя.

Из вышеизложенного следует, что остаток от деления принятого КВ на образующий многочлен несет в себе информацию об ошибочном разряде. Поэтому для кодов, предназначенных для исправления однократной ошибки, схема, представленная на рис. 2.8, не является оправданной. Здесь не нужны дополнительные такты деления, а следует сразу же дешифрировать остаток после выполнения n тактов деления. Сложность дешифратора при этом будет такой же, как и в схеме рис. 2.8.

2.6.6. Мажоритарное декодирование.

Сущность мажоритарного декодирования излагается в п. 2.5.7 на примере линейных кодов.

Поскольку циклический код является разновидностью группового кода, то его проверочные символы должны выражаться через суммы по модулю 2 определенных информационных символов. Основная трудность состоит в нахождении систем контрольных проверок.

Существуют разные способы нахождения системы проверочных равенств. В работе [6], например, для определения проверочных символов используется соотношение

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

где h – двоичные коэффициенты генераторного многочлена $h(x)$, определяемого выражением:

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

где $g(x)$ – образующий многочлен циклического кода.

Более просто и наглядно данная задача решается, если построить схему деления на сумматорах по модулю 2 и непосредственно по схеме, прослеживая путь прохождения сигнала, составить интересующие соотношения. Рассмотрим этот вопрос на примере кода (7,3) с образующим многочленом $g(x)=(x+1)(x^3+x+1)$. Данный код имеет кодовое расстояние $d=4$, поэтому способен исправлять однократную ошибку и одновременно обнаруживать двойную.

Имеем:

$$g(x)=(x+1)(x^3+x+1)=x^4+x^3+x^2+1\rightarrow 11101.$$

Схема деления на этот образующий многочлен представлена на рис. 2.9.

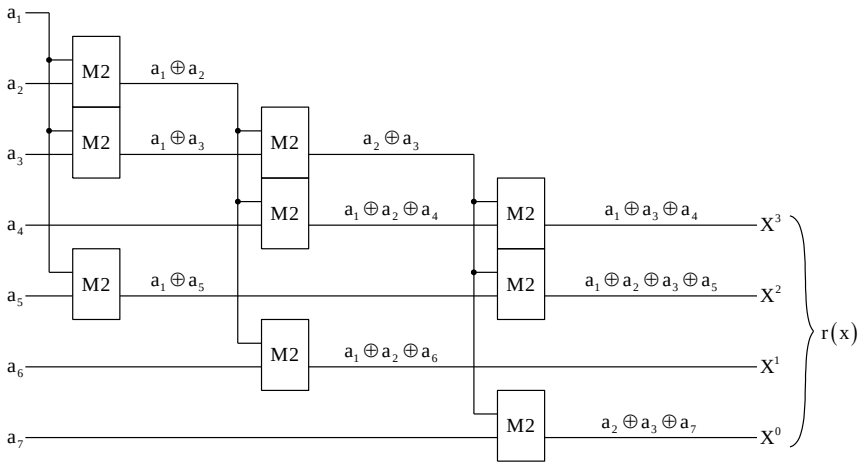


Рис. 2.9. Схема деления на $g(x)=(x+1)(x^3+x+1)$.

За разрешенные кодовые комбинации циклического кода принимают комбинации, которые делятся без остатка на образующий многочлен. Для рассматриваемого кода это будет иметь место только при выполнении следующих четырех условий:

1. $a_1 \oplus a_3 \oplus a_4 = 0$

2. $a_1 \oplus a_2 \oplus a_3 \oplus a_5 = 0 \qquad \rightarrow a_2 \oplus a_3 = a_1 \oplus a_5$

3. $a_1 \oplus a_2 \oplus a_6 = 0$

4. $a_2 \oplus a_3 \oplus a_7 = 0$

Используя полученные соотношения, выразим a_1 через различные символы. Добавляя тривиальную проверку $a_1 = a_1$, получает систему раздельных проверок для a_1:

$a_1 = a_3 \oplus a_4$

$a_1 = a_5 \oplus a_7$

$a_1 = a_2 \oplus a_6$

$a_1 = a_1$

Проверочные равенства для остальных символов в циклических кодах находятся по следующему правилу: каждый последующий символ определяется путем прибавления единицы к номеру предыдущего символа. Номер последнего символа при прибавлении к нему единицы заменяется на единицу.

Используя сформулированное правило, находим проверочные равенства для a_2 и a_3:

$a_2 = a_4 \oplus a_5$ $\qquad$ $a_3 = a_5 \oplus a_6$

$a_2 = a_6 \oplus a_1$ $\qquad$ $a_3 = a_7 \oplus a_2$

$a_2 = a_3 \oplus a_7$ $\qquad$ $a_3 = a_4 \oplus a_1$

$a_2 = a_2$ $\qquad$ $a_3 = a_3$

Пример.

Закодировать циклическим кодом (7,3) с $g(x) = (x+1)(x^3 + x + 1)$ безизбыточную комбинацию $a(x) \rightarrow 101$ и выполнить мажоритарное декодирование при появлении однократной и двойной ошибки.

Решение.

Находим разрешенный КВ:

$$
\begin{array}{rcccccc|ccccc}
 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\oplus & 1 & 1 & 1 & 0 & 1 & & & & & & & \\
\hline
 & \oplus & 1 & 0 & 0 & 1 & 0 & & & & & & \\
 & & 1 & 1 & 1 & 0 & 1 & & & & & & \\
\hline
 & & \oplus & 1 & 1 & 1 & 1 & 0 & & & & & \\
 & & & 1 & 1 & 1 & 0 & 1 & & & & & \\
\hline
 & & r(x)\rightarrow & 0 & 0 & 1 & 1 & & & & & &
\end{array}
$$

Итак,

$$
\begin{array}{rccccccc}
 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\
v(x)\rightarrow & 1 & 0 & 1 & 0 & 0 & 1 & 1
\end{array}
$$

Легко видеть, что при отсутствии искажений все равенства разработанной системы проверок выполняются.

Пусть при передаче $v(x)$ произошло искажение 3-го разряда, т.е.

$$
\begin{array}{rccccccc}
 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\
\hat{v}(x)\rightarrow & 1 & 0 & \underline{0} & 0 & 0 & 1 & 1
\end{array}
$$

Выполняем проверку:

$$
\begin{array}{lll}
a_1 = 0 \oplus 0 = 0 & a_2 = 0 \oplus 0 = 0 & a_3 = 0 \oplus 1 = 1 \\
a_1 = 0 \oplus 1 = 1 & a_2 = 1 \oplus 1 = 0 & a_3 = 1 \oplus 0 = 1 \\
a_1 = 0 \oplus 1 = 1 & a_2 = 0 \oplus 1 = 1 & a_3 = 0 \oplus 1 = 1 \\
a_1 = 1 & a_2 = 0 & a_3 = 0 \\
\hline
\text{Решение принято:} \quad a_1 = 1 & a_2 = 0 & a_3 = 1
\end{array}
$$

Итак, однократная ошибка исправлена.

Предположим, что произошло искажение 1-го и 5-го разрядов, т.е.

$$
\begin{array}{rccccccc}
 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\
\hat{v}(x)\rightarrow & \underline{0} & 0 & 1 & 0 & \underline{1} & 1 & 1
\end{array}
$$

Выполняем проверку:

	$a_1 = 1 \oplus 0 = 1$	$a_2 = 0 \oplus 1 = 1$	$a_3 = 1 \oplus 1 = 0$
	$a_1 = 1 \oplus 1 = 0$	$a_2 = 1 \oplus 0 = 1$	$a_3 = 1 \oplus 0 = 1$
	$a_1 = 0 \oplus 1 = 1$	$a_2 = 1 \oplus 1 = 0$	$a_3 = 0 \oplus 0 = 0$
	$a_1 = 0$	$a_2 = 0$	$a_3 = 1$
Решение не принимается:	$a_1 = ?$	$a_2 = ?$	$a_3 = ?$

Двойная ошибка фиксируется по равенству 1 и 0 в системе проверочных равенств, при этом не имеет значения как ошибки распределены по КВ. Однако следует иметь в виду, что при двойной ошибке по одному из символов решение принимается, если ошибочные разряды вошли в одну из проверок этого символа. Например, если бы искажение произошло в 1 и 6 разрядах, то решение по a_2 было бы принято. Отсюда следует, что, по крайней мере, два мажоритарных элемента должны фиксировать двойную ошибку, т.е. равенство 1 и 0 в системе проверок.

2.6.7. Матричное представление циклических кодов.

Образующая матрица циклического кода может быть построена на основании главной особенности этого кода – циклически сдвинутая разрешенная кодовая комбинация также является разрешенной. Именно это положение использовано при записи образующей матрицы (2.54).

Для того, чтобы образующая комбинация кода делилась на $g(x)$ без остатка, первая строка образующей матрицы формируется путем приписывания к представленному двоичным числом образующему многочлену кода $k-1$ нулей со стороны старших разрядов. Каждая следующая строка матрицы получается циклическим сдвигом этой строки на один разряд влево. Обычно такая матрица обозначается $\underline{M}\, g_{n,k}(x)$. Например, для кода (7,4) с $g(x) = x^3 + x^2 + 1$ матрица имеет вид:

$$\underline{\underline{M}}\, g_{7,4}(x) = \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{vmatrix} \qquad (2.62)$$

Матрица (2.62) легко строится, но недостаточно удобна при теоретических исследованиях. Поэтому чаще используется матрица $\underline{\underline{M}}_{n,k}$, состоящая из двух матриц: транспонированной единичной матрицы $\underline{I}^{T}_{k}$ (соответствующей k информационным разрядам) и дополнительной матрицы $\underline{C}_{m,k}$ (соответствующей проверочным разрядам):

$$\underline{\underline{M}}_{n,k} = \left| \underline{I}^{T}_{k}, \underline{C}_{m,k} \right| \qquad (2.63)$$

Для кода (7,4) матрица $\underline{I}^{T}_{k}$ имеет вид:

$$\underline{I}^{T}_{k} = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

Матрица $\underline{C}_{m,k}$ может быть построена путем вычисления остатков $r(x)$: для каждой строки матрицы $\underline{I}^{T}_{k}$, рассматриваемой как $a(x)$, соответствующий остаток $r(x)$ находится делением многочлена $a(x)\cdot x^{m}$ этой строки на образующий многочлен кода $g(x)$.

Учитывая изложенное, для $g(x) = x^3 + x^2 + 1$ матрица $\underline{C}_{3,4}$ записывается в виде:

$$\underline{C}_{3,4} = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix}$$

Матрица $\underline{M}_{7,4}$, составленная в соответствии с (2.63) для $g(x) = x^3 + x^2 + 1$, имеет вид:

$$\underline{M}_{7,4} = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix}$$

2.7. Краткие сведения о других кодах.

Итеративные коды.

Различные коды обладают разными корректирующими способностями. Для получения более совершенных кодов можно использовать комбинации двух и более кодов. Одним из таких классов кодов являются итеративные. Для них характерно, что операции кодирования проводятся над совокупностью информационных символов, располагаемых по нескольким, например q, координатам. В связи с этим итеративные коды также называют многомерными, многостепенными. Число информационных символов в кодовом векторе q-степенного кода равно:

$$k = \prod_{\gamma=1}^{q} k_\gamma ,$$

где k_γ – число информационных символов по координате γ.

Последовательность информационных символов по каждой из координат кодируется каким-либо линейным кодом. В общем случае каждый информационный символ входит одновременно в q различных кодовых векторов.

В простейшем случае, соответствующем $\gamma = 2$, итеративный код получается путем расположения информационных символов в виде следующей таблицы:

Каждая строка этой таблицы кодируется каким-либо кодом, а затем кодируется каждый столбец, причем не обязательно тем же кодом. Символы, расположенные в правом нижнем углу таблицы, получаются в результате проверки проверочных символов. Они могут быть построены на основе проверки по строкам и тогда будут удовлетворять проверке по столбцам, и наоборот.

В качестве примера рассмотрим итеративный код с проверкой на четность каждого столбца и каждой строки.

1	0	0	1	1	1
1	1	0	0	1	1
1	0	0	0	1	0
1	1	1	1	1	1
1	0	0	0	0	1
0	0	1	1	0	0
1	0	0	1	0	0

Передачу символов такого кода обычно осуществляют последовательно символ за символом, от одной строки к дру-

гой, либо параллельно целыми строками. Декодирование начинают сразу, не ожидая поступления всего блока информации.

Такой код обладает большими корректирующими способностями, чем обычный код с проверкой на четность (обнаружение одиночных и всех нечетных ошибок). Данный код позволяет исправить все одиночные ошибки, т.к. пересечение строки и столбца, содержащих ошибки, однозначно указывает ее место. Более того, код позволяет исправить любое нечетное число искаженных символов, расположенных в одной строке или столбце. Большинство ошибок другой конфигурации может быть обнаружено этим кодом. Необнаруженными оказываются только ошибки, имеющие четное число искаженных символов как по строкам, так и по столбцам.

Минимальное кодовое расстояние двумерного итеративного кода $d \geq d_1 + d_2$, где d_1 и d_2 – кодовые расстояния кодов, используемых для кодирования строк и столбцов.

Каскадные коды.

Каскадные коды, как и итеративные, состоят из двух или более кодов, но в отличие от них строятся ступенчатым образом: кодовые слова одного кода являются информационными символами для кода следующей ступени.

В теории кодирования доказывается, что кодовое расстояние каскадного кода $d \geq d_1 \cdot d_2$.

Достоинством каскадных кодов является возможность исправления не только одиночных ошибок, но и пакетов ошибок. Это достигается за счет того, что в качестве внутреннего кода используется код, исправляющий одиночные ошибки, а качестве внешнего кода – код, обнаруживающий и исправляющий пакеты ошибок.

Теоретически наиболее полно исследованы каскадные коды, в которых внутренними являются коды Хэмминга, а внешними – коды Рида-Соломона.

Цепной код.

Цепной код относится к группе непрерывных кодов, где операции кодирования и декодирования производятся непрерывно над последовательностью символов без деления ее на блоки.

В цепном коде после каждого информационного элемента следует проверочный элемент, формируемый путем сложения по модулю 2 двух информационных элементов, отстоящих друг от друга на шаг сложения ℓ. Шаг ℓ – это расстояние между двумя информационными элементами, формирующими проверочный элемент.

Обозначим через $a_0, a_1, \ldots, a_\ell, a_{\ell+1}, \ldots, a_{2\ell+1}, \ldots$ информационную последовательность символов. Проверочные символы обозначим через b. Согласно сказанному выше, имеем: $b_0 = a_0 \oplus a_\ell$; $b_1 = a_1 \oplus a_{\ell+1}$; …; $b_\ell = a_\ell \oplus a_{2\ell}$; …

Закодированная цепным кодом последовательность имеет вид:

$a_0\ b_0\ a_1\ b_1\ a_2\ b_2 \ldots a_\ell b_{2\ell} \ldots .$

Процесс декодирования цепного кода состоит в следующем:

– из поступающей последовательности элементов выделяют отдельно информационные элементы и отдельно проверочные элементы;

– из принятой последовательности информационных элементов по известному правилу кодирования формируются новые проверочные элементы;

– каждый сформированный проверочный элемент складывается по модулю 2 с принятым проверочным элементом. При отсутствии искажений результатом суммирования будет 0, в противном случае – 1.

В теории кодирования доказывается, что при шаге сложения ℓ цепной код способен исправлять пачки ошибок длиной $b_s = 2\ell$. Изменяя шаг сложения, можно согласовывать корректирующие способности кода с характеристиками канала связи.

Простейшим цепным кодом является код Финка-Хагельбергера, имеющий $\ell = 1$, т.е. $b_i = a_i \oplus a_{i+1}$. Этот код позволяет исправлять все одиночные ошибки при условии, что между двумя любыми ошибочно принятыми символами имеется по крайней мере три правильно принятых символа. Рассмотрим пример.

Предположим, что в последовательности символов кода Финка-Хагельбергера $a_1\, b_1\, \underline{a_2}\, b_2\, a_3\, b_3\, \underline{a_4}\, b_4\, a_5\, b_5\, a_6\, b_6 \ldots$ подверглись искажению символы a_2 и a_4.

Исправление ошибок осуществляется в следующей последовательности:

– с момента обнаружения ошибки регистрируются все неправильные проверки до появления правильной. Имеем:

1) $b_1 \neq a_1 \oplus a_2$

2) $b_2 \neq a_2 \oplus a_3$

3) $b_3 \neq a_3 \oplus a_4$

4) $b_4 \neq a_4 \oplus a_5$

5) $b_5 \neq a_5 \oplus a_6$. $\rightarrow$ Вывод: a_5 – достоверно.

– в проверке 4) b_4 – достоверно, т.к. в противном случае в проверке 3) было бы равенство (между двумя ошибочными символами должно быть не менее трех правильных символов). Вывод: искажен символ a_4, следовательно, b_3, a_3 и b_2 – достоверны;

– в проверке 2) искаженным является символ a_2, т.к. при искажении b_2 выполнялась бы проверка 1).

2.8. Понятие об адаптивном кодировании.

Все рассмотренные выше корректирующие коды обеспечивают требуемую достоверность передачи данных информации в случаях, когда количество ошибок в канале связи не

превышает то, на которое рассчитан код, т.е. когда канал связи стационарен. Однако большинство реальных каналов связи относится к числу нестационарных, т.е. их качество изменяется в течение времени. Тем не менее, для большинства реальных каналов связи можно выделить некоторые промежутки времени (состояния), в течение которых их свойства можно считать неизменными. Если число таких состояний конечно, а их длительность много больше длительности сообщения, то такие каналы называют кусочно-стационарными или квазистационарными. Именно на свойстве квазистационарности реальных каналов связи и базируется адаптивное кодирование.

Очевидно, что для наилучшего использования канала связи необходимо в зависимости от состояния канала связи менять вносимую в сообщение избыточность, увеличивая ее по мере роста вероятности появления ошибки и уменьшая в противном случае, т.е. менять алгоритм кодирования. Можно также менять алгоритм декодирования, используя, например, режим исправления ошибок при хорошем качестве канала связи и режим обнаружения ошибок при его ухудшении.

Системы, в которых осуществляется целенаправленное изменение их параметров или структуры в зависимости от условий передачи сообщений, называются адаптивными.

Вариантом простейшего адаптивного кодирования является повторение передачи кодовой комбинации при обнаружении в ней ошибки. Системы такого типа наиболее часто используются на практике и относятся к системам с обратной связью. Их принято разделять на два класса: системы с решающей обратной связью и системы с информационной обратной связью. Рассмотрим особенности этих систем.

Системы с решающей обратной связью (РОС).

Упрощенная структурная схема такой системы представлена на рис. 2.10.

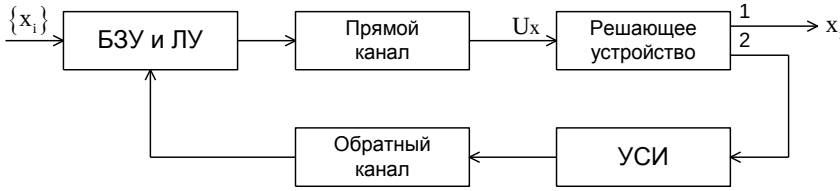


Рис. 2.10. Структурная схема системы с РОС.

На схеме приняты следующие обозначения: БЗУ и ЛУ – буферное запоминающее устройство и логическое устройство; УСИ – устройство выработки служебной информации; U_x – амплитуда сигнала на входе решающего устройства.

Предположим, что сообщение, состоящее из множества символов x_i, передается безизбыточным кодом с использованием $АМ_н$, а прием ведется посимвольно. В данном случае решающее устройство имеет два пороговых уровня – U_1 и U_2 (пусть $U_1 > U_2$). Решение принимается в соответствии с правилом:

$$U_x > U_1 \rightarrow x_i = x_1;$$

$$U_x < U_2 \rightarrow x_i = x_2;$$

$U_2 \le U_x \le U_1 \rightarrow$ решение не принято.

Итак, в данном случае имеется зона неопределенности, когда решение не принимается. В литературе эту зону называют также зоной стирания.

Если решение не принято, то на 2-м выходе решающего устройства появляется сигнал, под действием которого УСИ формирует сигнал переспроса, передаваемый по обратному каналу в передающую часть системы. Под действием этого сигнала логическое устройство обеспечивает повторную передачу информационного символа, который хранится в БЗУ. Если снова не будет принято решение, переспрос повторяется и т.д. В случаях, когда сигнал переспроса не поступает в течение заданного времени, хранимый в БЗУ символ стирается и посылается новый.

Системы с РОС особенно эффективны при использовании корректирующих кодов. Все множество кодовых комбинаций при этом разбивается на подмножество разрешенных и запрещенных. Если принятая комбинация отнесена к подмножеству разрешенных, то считается, что такая же комбинация была и передана. В противном случае формируется сигнал переспроса.

Системы с информационной обратной связью (ИОС).

Структурная схема системы ИОС в общих чертах такая же, как и для систем с РОС. Отличие состоит в том, что решение о качестве в данном случае принимает передающая сторона.

В системах с ИОС каждое принятое сообщение передается по обратному каналу в пункт передачи, где оно сравнивается с исходным сообщением, хранимым в БЗУ. Если сообщения совпадают или различаются в допустимых пределах, зависящих от корректирующей способности используемого кода, то на передающей стороне принимается решение, что сообщение принято правильно, и получателю посылается сигнал подтверждения, в соответствии с которым принятое ранее сообщение, хранящееся в запоминающем устройстве, передается по назначению. Если же различие между сообщениями превышает допустимые пределы, передающая сторона посылает сигнал, что принятое сообщение недостоверно и повторяет передачу. Системы с ИОС, в которых по обратному каналу передается вся информация, переданная по прямому, называется системами с ретрансляционной обратной связью.

Существует несколько разновидностей систем с ИОС. В частности, если для передачи применяются корректирующие коды, то по прямому каналу можно передавать только информационные символы, а по обратному – только проверочные. Сравнивая на передающей стороне принятые проверочные символы с хранящимися в запоминающем устройстве, можно сделать вывод о правильности приема сообщения.

Имеется вариант, в котором после проверки принятого по обратному каналу сообщения и обнаружения ошибки передатчик может либо повторить его, либо послать дополнительную информацию, необходимую для исправления (корректирующая информация).

Из принципа действия систем с ИОС следует, что их целесообразно применять в случаях, когда скорость передачи информации не является главным, а требуется обеспечить высокую достоверность передаваемых сообщений (например, при передаче команд).

В системах с ИОС качество обратного канала должно быть не хуже качества прямого во избежание искажений, которые могут увеличить число повторений.

Системы с обратной связью любого типа следует относить к системам с адаптивным кодированием, т.к. реальная скорость передачи информации в них зависит от состояния канала связи – при ухудшении состояния канала увеличивается число повторных передач и наоборот. Это эквивалентно изменению избыточности в передаваемых сообщениях, что является характерным признаком адаптивного кодирования.

Список литературы.

1. Э.М. Габидулин, В.Б. Афанасьев. Кодирование в радиоэлектронике. – М.: «Радио и связь», 1986.

2. Журавлев Ю.П., Забубенков В.Н. Мультитаймеры. – Л.: «Энергия», 1979.

3. В.А. Острейковский. Информатика. – М.: «Высшая школа», 2001.

4. В.И. Першиков, В.М. Савинков. Толковый словарь по информатике. – М.: «Финансы и статистика», 1991.

5. И.В. Ситняковский, О.Н. Порохов, А.Л. Нехаев. Цифровые системы передачи абонентских линий. – М.: «Радио и связь», 1987.

6. Ф.Е. Темников, В.А. Афонин, В.И. Дмитриев. Теоретические основы информационной техники. – М.: «Энергия», 1979.

7. Тутевич В.Н. Телемеханика. – М.: «Высшая школа», 1985.

8. Цымбал В.П. Задачник по теории информации и кодированию. – Киев, изд. «Вища школа», 1976.

9. Н.С. Щербаков. Достоверность работы цифровых устройств. – М.: «Машиностроение», 1989.

10. Ю.Э. Яцкевич. Теоретические основы вычислительной техники. Информационные основы. – Л.: изд. ЛПИ, 1977.

СПИРИДОНОВ Александр Иванович

ОСНОВЫ ТЕОРИИ ИНФОРМАЦИИ И КОДИРОВАНИЯ

Учебное пособие

Технический редактор Спиридонов А.И.
Компьютерная верстка Юткина Т.А.

Лицензия ЛР № 020593 от 07.08.97

Формат 60×84/16. Печать офсетная.
Гарнитура «Times New Roman». Уч.изд. п.л. 5,0.
Тираж 150 экз. Заказ от .
Псковский государственный политехнический институт
Издательство СПбГПУ, член Издательско-полиграфической
ассоциации вузов Санкт-Петербурга.
Адрес университета и издательства:
Россия, 195251, Санкт-Петербург,
ул. Политехническая, 29
Отпечатано с готового оригинал-макета,
предоставленного автором,
в ООО "Литан". 180680, г. Псков, ул. Новаторов, 3.