Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Амурский государственный университет»

Акилова И.М., С.Г. Самохвалова

ТЕОРИЯ ИНФОРМАЦИИ

Методические указания к лабораторным занятиям
для студентов очной формы обучения

Благовещенск
2020

Теория информации. Методические указания к лабораторным занятиям для студентов очной формы обучения. / Составитель И.М., Акилова, С.Г. Самохвалова – Благовещенск.: ФГБОУ ВО «АмГУ», 2020 г. – 45 с.

Основу методических указаний составляют краткие теоретические сведения из теории кодирования, примеры, задания для лабораторных работ, контрольные вопросы. Лабораторные занятия призваны обеспечить закрепление полученных теоретических знаний по основам теории кодирования, выработать необходимые навыки кодирования информации методами оптимального и помехоустойчивого кодирования.

Методические указания рекомендуется студентам направления подготовки 09.03.01 – Информатика и вычислительная техника и 09.03.02 - Информационные системы и технологии, изучающим дисциплину «Теория информации», а также могут быть полезны для преподавателей и студентов, преподающих и осваивающих эту дисциплину в рамках других направлений подготовки, где лабораторные занятия по дисциплине предусмотрены учебными планами.

Рецензент:

Чалкина Н.А. доцент, к.п.н. доцент кафедры общей математики и информатики ФГБОУ ВО АмГУ

СОДЕРЖАНИЕ

Введение

Методы кодирования давно и широко используются в практической деятельности человека. Цели, которые может преследовать кодирование информации:

представление информации в цифровом виде для обеспечения простоты, надежности и эффективности информационных устройств;

уменьшение размера сообщений (сжатие);

повышение помехоустойчивости и достоверности при передаче или хранении информации;

криптографическая защита информации от несанкционированного доступа;

кодовое разделение сигналов для передачи нескольких сообщений по одному каналу связи;

согласование сигнала и канала связи.

С точки зрения теории информации кодирование — это преобразование исходного сообщения в совокупность или последовательность кодовых символов, отображающих сообщение, передаваемое по каналу связи.

Задачи кодирования при отсутствии помех и при их наличии существенно различаются. Поэтому различают эффективное (оптимальное) кодирование и корректирующее (помехоустойчивое) кодирование.

При эффективном кодировании ставится задача добиться представления символов алфавита источника сообщений минимальным числом элементов кодовых символов в среднем на один символ алфавита источника сообщений за счет уменьшения избыточности кода, что ведет к повышению скорости передачи сообщения.

При корректирующем (помехоустойчивом) кодировании ставится задача снижения вероятности ошибок в передаче символов исходного алфавита путем обнаружения и исправления ошибок за счет введения дополнительной избыточности кода.

В отдельный класс выделяют методы кодирования, которые позволяют построить (без потери информации) коды сообщений, имеющие меньшую длину по сравнению с исходным сообщением. Такие методы кодирования называют методами сжатия. Классификация помехоустойчивых кодов приведена на рисунке 1.

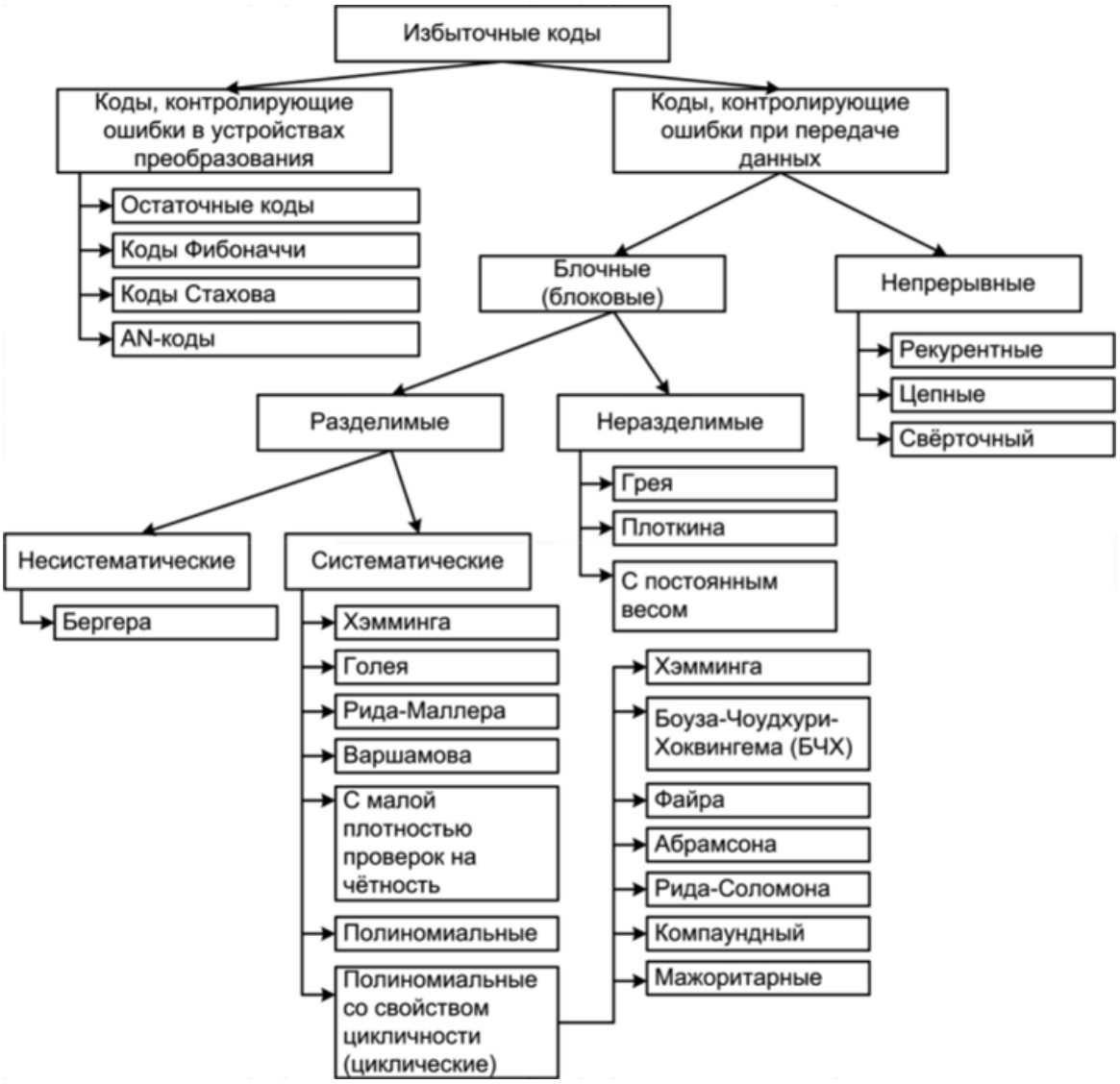


Рисунок 1. Классификация помехоустойчивых кодов

Основные понятия

Процесс преобразования сообщений в комбинации из дискретных сигналов называется **кодированием**, совокупность правил, в соответствии, с которыми производятся данные преобразования – **кодом**. Каждая комбинация записывается в виде последовательности, составленной из некоторых условных символов-элементов кодовой комбинации. В качестве ее элементов могут использоваться буквы и цифры.

Каждому сообщению однозначно соответствует определенная кодовая комбинация. Код позволяет записывать все сообщения на некотором общем для данного набора сообщений языке. С этой точки зрения набор элементов данного кода рассматривают как **алфавит**, число элементов — **объемом алфавита**. Каждое сообщение передается собственным кодовым словом.

Декодирование — операция восстановления исходного сообщения.

Правила составления кодовых комбинаций (коды) и сами кодовые комбинации могут иметь различные характеристики:

число кодовых признаков, используемых для комбинирования,

количество разрядов кодовой комбинации,

способ комбинирования (закон, согласно которому из единичных элементов образуются кодовые комбинации).

Эти три свойства относятся к структурным характеристикам кода.

По числу кодовых признаков (символов) коды подразделяются на *единичные, двоичные, многопозиционные*. В единичном коде используется только один символ, и кодовые комбинации отличаются друг от друга лишь количеством символов. Кодовые комбинации двоичных кодов содержат два символа (0 или 1), многопозиционные – более двух.

По количеству разрядов кодовые комбинации разделяет коды на *равномерные и неравномерные.* Равномерные – это коды, все кодовые комбинации которых содержат постоянное количество разрядов; неравномерные содержат кодовые комбинации с различным числом разрядов.

Кодовая комбинация (кодовое слово) — последовательность символов кода, соответствующая букве или группе букв исходного алфавита. Для краткости словом «код» часто называют сами кодовые комбинации.

Число символов в кодовой комбинации называется **длиной кодовой комбинации.**

Вес кодовой комбинации w (вес Хэминга) — количество единиц в кодовой комбинации. Например, кодовая комбинация 0110010 имеет вес $w = 3$.

Весовая характеристика кода — число кодовых комбинаций определенного веса. Различие между двумя кодовыми комбинациями характеризуется расстоянием Хэмминга d (кодовым расстоянием). Оно равно числу разрядов, в которых комбинации отличаются одна от другой.

Для вычисления кодового расстояния надо найти вес суммы этих комбинаций по модулю два. Сложение по модулю два (записывается в виде a + b (mod 2) или a $\oplus$ b) наиболее распространенная при кодировании и декодировании операция в двоичной системе счисления, определяемая равенствами:

$0 \oplus 0 = 0;$ $0 \oplus 1 = 1;$ $1 \oplus 0 = 1;$ $1 \oplus 1 = 0.$

Например: кодовое расстояние d_0 между комбинациями 10010111 и 00100110, находится путем суммирования их по модулю два:

$$\begin{array}{r} 10010111 \\ +\ 00100110 \\ \hline 10110001 \end{array}$$

Полученная в результате сложения ванная новая кодовая комбинация характеризуется весом w = но, кодовое расстояние между исходными комбинациями $d_0 = 4$.

Основные характеристики кодов

Оценка кодов обычно производится по их основным характеристикам, выражающим различные количественные и качественные показатели. Эти характеристики используются при выборе кодов, предназначенных для передачи, хранения и обработки информации: длина кода; основание кода; мощность кода; полное число кодовых комбинаций; число информационных символов; число проверочных символов; избыточность кода; скорость передачи; вес кодовой комбинации; кодовое расстояние d_0; весовая характеристика кода; веро-

ятность необнаруженной ошибки; оптимальность кода; коэффициент ложных переходов.

Длина кода n – число разрядов, составляющих кодовую комбинацию.

Основание кода m – количество отличающихся друг от друга значений импульсных признаков, используемых в кодовых комбинациях. Для случая двоичных кодов $m=2$. В качестве значений импульсных признаков используются цифры 0 и 1.

Число информационных символов ($n_и$) – количество символов (разрядов) кодовой комбинации, предназначенных для передачи сообщения, связь между информационными разрядами и первичным алфавитом можно записать в виде:

$$N = 2^{n_и}$$

Число проверочных символов ($n_к$) – количество символов (разрядов) кодовой комбинации, необходимых для коррекции ошибок. Это число характеризует абсолютную избыточность кода.

Формулы, по которым определяется связь между n, $n_и$ и $n_к$.

$$n = n_и + n_к,$$

Если известно количество информационных разрядов $n_и$, то $n_к$ вычисляется по формуле:

$$n_k = [\log((n_и + 1) + [\log(n_и + 1)])]$$

Квадратные скобки означают округление полученного числа до целого. Если известно количество разрядов в коде, т. е. n, то количество корректирующих разрядов равно:

$$n_k = [\log((n + 1)]$$

Корректирующая способность кода зависит от кодового расстояния:

а) при $d=1$ ошибка не обнаруживается;

б) при $d=2$ обнаруживаются одиночные ошибки;

в) при $d=3$ исправляются одиночные ошибки или обнаруживаются двойные ошибки.

В общем случае

$$d = r + s + 1, \qquad d = 2s + 1.$$

где d – минимальное кодовое расстояние,

r – число обнаруживаемых ошибок,

s – число исправляемых ошибок.

При этом обязательным условием является $r \geq s$.

Помехоустойчивое кодирование

Высокие требования к достоверности и надежности передачи, обработки и хранения информации в системах передачи данных, в вычислительных системах и сетях, в региональных системах управления и различного рода информационных системах требуют такого кодирования информации, которое обеспечивало бы безошибочную ее передачу, а в случае появления ошибок - их обнаружение и исправление.

Коды, обладающие такой способностью, называют *помехоустойчивыми или корректирующими*. Подавляющее большинство существующих в настоящее время помехоустойчивых кодов обладают требуемыми свойствами благодаря их алгебраической структуре. Поэтому их еще называют алгебраическими кодами.

Кодовые комбинации (кодовые символы) алгебраических кодов включают в себя две группы элементов кодовых символов: информационные элементы и проверочные элементы. Совокупность информационных элементов кодового символа соответствуют символу кодируемого сообщения, а проверочные (избыточные) элементы добавляются к информационным элементам и служат для обнаружения и исправления ошибок.

Все алгебраические коды можно разделить на два больших класса: *блочные (блоковые) и непрерывные*. Блочные коды представляют собой совокупность кодовых символов, состоящих из отдельных комбинаций (блоков) элементов символов кода, которые кодируются и декодируются независимо. При этом каждому символу кодируемого исходного сообщения ставится в соответствие блок (комбинация) из n элементов символов кода, куда включаются информационные и проверочные элементы.

Блочный код называют равномерным, если n для всех блоков одинаково. *Непрерывные (древовидные) коды* представляют собой непрерывную последовательность кодовых символов, причем введение проверочных элементов производится непрерывно, без разделения ее на независимые блоки.

Как блочные коды, так и непрерывные могут быть разделимыми и неразделимыми. В разделимых кодах информационные и проверочные элементы символов кода отчетливо разграничены и всегда занимают одни и те же определенные позиции (разряды). Такие коды часто называют (n,k) коды, где n — длина кодового символа, k — число информационных элементов в нем.

При кодировании неразделимыми кодами разделение кодового символа на информационные элементы и проверочные невозможно. Среди разделимых кодов выделяют систематические (линейные) и несистематические.

Систематическими кодами называют коды, в которых проверочные элементы являются линейными комбинациями информационных. Эти коды наиболее распространены, т.к. их использование существенно упрощает техническую реализацию кодирующих и декодирующих устройств.

Контрольные вопросы:

1. Что понимают под кодированием сообщения?

2. Какие коды называются равномерными?

3. Что называется кодовым расстоянием?

4. Что называется декодирование сообщения?

5. Если известно число информационных разрядов, как можно определить проверочные разряды?

6. Какие коды называются единичными?

7. По какой формуле определяется связь между информационными и проверочными разрядами?

8. На какие классы делятся блочные коды?

9. Какие коды называются систематическими?

Лабораторная работа. Оптимальное кодирование

Эффективное кодирование используется в каналах без шума, т.е. в таких каналах, где помехи отсутствуют, либо ими можно пренебречь. Основной задачей кодирования в таком канале является обеспечение максимальной скорости передачи информации, близкой к пропускной способности канала передачи.

Оптимальным кодированием называется процедура преобразования символов первичного алфавита N в кодовые слова во вторичном алфавите m, при которой средняя длина сообщений во вторичном алфавите имеет минимально возможную для данного m длину.

Оптимальными именуются коды, представляющие кодируемые понятия кодовыми словами минимальной средней длины. Оптимальные коды относятся к классу *префиксных кодов*, т.е. каждая кодовая комбинация имеет свою длину и ни одна не является началом другой, более длиной.

В сообщениях, составленных из кодовых слов оптимального кода, статистическая избыточность сведена к минимуму, в идеальном случае – к нулю.

Основная теорема кодирования для каналов связи без шумов доказывает принципиальную возможность построения оптимальных кодов. Из нее однозначно вытекают методика построения и свойства оптимальных кодов.

Одно из основных положений этой теории заключается в том, что при кодировании сообщения, разбитого на N - буквенные блоки, можно, выбрав N достаточно большим, добиться, чтобы среднее число двоичных элементарных сигналов, приходящихся на одну букву исходного сообщения, было сколь угодно близким к $H/\log m$. Разность $L - \frac{H}{\log m}$ будет тем меньше, чем больше H, а H достигает максимума при равновероятных и взаимонезависимых символах. Отсюда вытекают основные свойства оптимальных кодов:

минимальная средняя длина кодового слова оптимального кода обеспечивается в случае, когда избыточность каждого кодового слова сведена к минимуму (в идеальном случае - к нулю);

кодовые слова оптимального кода должны строиться из равновероятных и взаимонезависимых символов.

Из свойств оптимальных кодов вытекают принципы их построения:

1 - выбор каждого кодового слова необходимо производить так, чтобы содержащееся в нем количество информации было максимальным;

2 - буквам первичного алфавита, имеющим большую вероятность, присваивается более короткие кодовые слова во вторичном алфавите.

Принципы оптимального кодирования определяют методику построения оптимальных кодов. *Построение оптимального кода по методу* ***Шеннона-Фано*** для ансамбля из *M* сообщений сводится к следующей процедуре:

1) множество из *M* сообщений располагают в порядке убывания вероятностей;

2) первоначальный ансамбль кодируемых сигналов разбивают на две группы таким образом, чтобы суммарные вероятности сообщений обеих групп были по возможности равны;

3) первой группе присваивают символ *0*, второй - символ *1*;

4) каждую из групп делят на две подгруппы так, чтобы их суммарные вероятности были по возможности равны;

5) первым подгруппам каждой из групп вновь присваивают *0*, а вторым – *1*, в результате получают вторые цифры кода. Затем каждую из четырех подгрупп вновь делят на равные (с точки зрения суммарной вероятности) части и т.д. - до тех пор, пока в каждой из них останется одна буква.

Пример. Построим оптимальный код для передачи сообщений, в которых вероятности появления букв первичного алфавиты равны: $A_1=1/4$, $A_2=1/4$, $A_3=1/8$, $A_4=1/8$, $A_5=1/16$, $A_6=1/16$, $A_7=1/16$, $A_8=1/16$.

Решение. Построение ведем по общей методике. Оптимальный код для данных условий представлен в табл. 1.

Таблица 1

Буква	Вероятность появления буквы	Кодовое слово после разбиения				Число знаков в кодовом слове	L(i) p_i
		1-го	2-го	3-го	4-го		

A_1	1/4	0	0			2	0,5
A_2	1/4	0	1			2	0,5
A_3	1/8	1	0	0		3	0,375
A_4	1/8	1	0	1		3	0,375
A_5	1/16	1	1	0	0	4	0,25
A_6	1/16	1	1	0	1	4	0,25
A_7	1/16	1	1	1	0	4	0,25
A_8	1/16	1	1	1	1	4	0,25

Проверка оптимальности кода осуществляется путем сравнения энтропии кодируемого (первичного) алфавита со средней длиной кодового слова во вторичном алфавите.

Для рассматриваемого примера энтропия источника сообщений

$$H= - \sum_{i=1}^{N} p_i \log p_i = 2{,}75 \text{ бит/символ.}$$

Среднее число двоичных знаков на букву кода

$$L= \sum_{i}^{N} l(i) * p_i = 2*0.5+2*0.375+4*0.25=2{,}75 \quad \text{бит/символ,}$$

где $l(i)$ – длина i-й кодовой комбинации;

p_i - вероятность появления *i-го* символа комбинации длиной в $l(i)$.

Таким образом, $H=L$, т.е. код, оптимален для данного ансамбля сообщений.

Коды, представляющие первичные алфавиты с неравномерным распределением символов, имеющие минимальную среднюю длину кодового слова во вторичном алфавите, называются оптимальными неравномерными кодами (ОНК).

Максимально эффективными будут те ОНК, у которых

$$log_2 m \sum_{i=1}^{N} l(i) p_i = l_{cp} = H,$$

где m и N – символы соответственно вторичного и первичного алфавитов.

Эффективность ОНК оценивают при помощи *коэффициента статистического сжатия*

$$K_{c.c} = \frac{H_{\max}}{l_{cp}} = \frac{\log_2 N}{\log_2 m \sum_{i=1}^{N} l(i) p_i},$$

характеризующего уменьшение количества двоичных знаков на символ сообщения при применении ОНК по сравнению с применением методов нестатистического кодирования, и *коэффициента относительной эффективности*

$$K_{о.э} = \frac{H}{l_{ср}} = \frac{-\sum_{i=1}^{N} p_i \log_2 p_i}{\log_2 m \sum_{i=1}^{N} l(i) p_i},$$

показывающего, насколько используется статистическая избыточность передаваемого сообщения.

Средняя длина кодового слова передаваемого сообщения по мере укрупнения кодируемых блоков будет уменьшаться, а код – приближаться к оптимальному.

Рассмотрим пример, иллюстрирующий преимущества укрупнения символов.

Пример. Даны символы *a* и *b* с частотами, соответственно, 0,9 и 0,1. Построить эффективный код методом Шеннона-Фано , при кодировании по одному (1 случай), два (2 случай) и три (3 случай) символов в блоке.

Решение. Представим кодирование символов в таблице

Случай кодирования	Блок	Вероятность появления блока	Кодовые слова после разбиения					Число знаков в кодовом слове	$l(i)p_i$
1	***а***	0,9	0					1	0,9
	в	0,1	1					1	0,1
2	***аа***	0,81	0	-	-			1	0,81
	ав	0,09	1	0	-			2	0,18
	ва	0,09	1	1	0			3	0,27
	вв	0,01	1	1	1			3	0,03
3	***ааа***	0,729	0	-	-	-	-	1	0,729
	аав	0,081	1	0	0	-	-	3	0,243
	ава	0,081	1	0	1	-	-	3	0,243
	ваа	0,081	1	1	0	-	-	3	0,243
	авв	0,009	1	1	1	0	0	5	0,045
	вав	0,009	1	1	1	0	1	5	0,045
	вва	0,009	1	1	1	1	0	5	0,045
	ввв	0,001	1	1	1	1	1	5	0,005

Определим средние числа двоичных знаков на букву кода и энтропии источников сообщений для каждого случая.

1.
$$L_1 = \sum_i l(i)\, p_i = 0{,}9 + 0{,}1 = 1$$
$$H_1 = -\sum_i p_i \log_2 p_i = -(0{,}9 \log 0{,}9 + 0{,}1 \log 0{,}1) = 0{,}468$$

2.
$$L_1 = \sum_i l(i)\, p_i = 1{,}29$$
$$H_1 = -\sum_i p_i \log_2 p_i = 0{,}932$$

3.
$$L_1 = \sum_i l(i)\, p_i = 1{,}598$$
$$H_1 = -\sum_i p_i \log_2 p_i = 1{,}405$$

Сравнивая полученные данные, убеждаемся, что с укрупнением кодируемых блоков разница значений L и H быстро уменьшается, а полученный код приближается к оптимальному.

Эффективность блочного кодирования тем выше, чем больше символов включается в блок.

Метод Шеннона-Фано не единственный способ построения оптимальных кодов.

В 1952 году Давид Хаффман показал, что предложенный им метод кодирования является оптимальным префиксным кодом для дискретных источников без памяти (заметим, что для таких источников все генерируемые сообщения независимы друг от друга).

Д. Хаффмен показал, что для получения минимально возможной длины кода основания m с числом взаимонезависимых букв первичного алфавита N

$$l_{cp} = \sum_{i=1}^{N} l(i)\, p_i$$

необходимо и достаточно выполнение следующих условий:

1) если выписать символы в порядке убывания вероятностей $p_i > p_j$, то при $i < j$, $l(i) < l(j)$;

2) два последних, но не больше чем m кодовых слова равны по длитель-

ности и различаются лишь значениями последнего символа, при этом

$$2 \le n_0 \le m,$$

где m – число качественных признаков вторичного алфавита, а n_0 – число наименее вероятных сообщений, объединяемых на первом этапе построения кодового дерева; кроме того

$$\frac{N - n_0}{m - 1} = a$$

где a – целое положительное число;

3) любая возможная последовательность l_N-1 кодовых слов должна сама быть кодовой комбинацией.

Исходя из данных условий, Хаффмен предложил следующий метод построения ОНК. Символы первичного алфавита выписываются в порядке убывания вероятностей. Последние n_0 символов, где $2 \le n_0 \le m$ и $N - n_0 / m - 1$ - целое число, объединяют в некоторый новый символ с вероятностью, равной сумме вероятностей объединяемых символов. Последние символы с учетом образованного символа вновь объединяют и получают новый, вспомогательный символ. Опять выписывают символы в порядке убывания вероятностей с учетом вспомогательного – и так до тех пор, пока вероятности m оставшихся символов после N-n_0/m-1-го выписывания не дадут в сумме 1.

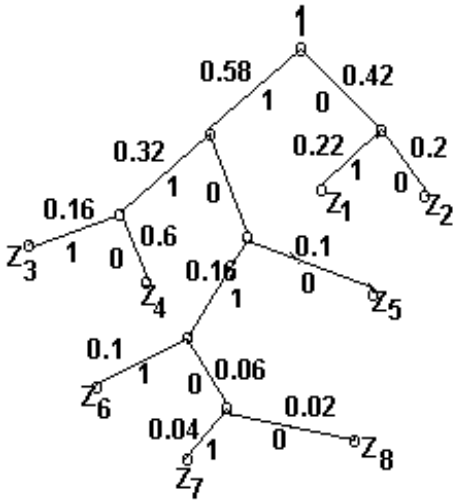
На практике обычно не производят многократного выписывания вероятностей символов, а обходятся геометрическими построениями, суть которых для кодов с числом качественных признаков $m=2$ сводится к тому, что символы кодируемого алфавита попарно объединяются в новые, начиная с символов, имеющих наименьшую вероятность, а затем, с учетом вероятностей вновь образованных символов, опять производят попарное объединение символов с наименьшими вероятностями и таким образом строят двоичное кодовое дерево, в вершине которого стоит символ с вероятностью *1*.

Пример. Используя методику Хаффмена, осуществить эффективное кодирование ансамбля знаков с вероятностями соответственно: $z_1=0,22$; $z_2=0,20$; $z_3=0,16$; $z_4=0,16$; $z_5=0,10$; $z_6=0,10$; $z_7=0,04$; $z_8=0,02$.

Решение.

Знаки	**Вероятности**	**Вспомогательные столбцы**						
		I	II	III	IV	V	VI	VII
Z_1	0.22	0.22	0.22	**0.26**	**0.32**	**0.42**	**0.58**	**1**
Z_2	0.20	0.20	0.20	0.22	0.26	0.32	0.42	
Z_3	0.16	0.16	0.16	0.20	0.22	0.26		
Z_4	0.16	0.16	0.16	0.16	0.20			
Z_5	0.10	0.10	**0.16**	0.16				
Z_6	0.10	0.10	0.10					
Z_7	0.04	**0.06**						
Z_8	0.02							

Для наглядности построим кодовое дерево. Из точки соответствующей вероятности *1* направляем две ветви, причем ветви с большей вероятностью присваиваем символ *1*, а с меньшей *0*. Такое последовательное ветвление продолжаем до тех пор, пока не дойдем до вероятности каждой буквы.



Теперь, двигаясь по кодовому дереву сверху вниз, можно записать для каждой буквы соответствующую ей кодовую комбинацию:

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
01	*00*	*111*	*110*	*100*	*1011*	*10101*	*10100*

При построении ОНК для вторичных алфавитов с $m=2$ методы Шеннона-Фано и Хаффмена дают в большинстве случаев одинаковые результаты.

Достоинства оптимальных эффективных кодов:

- на передачу сообщения затрачивается минимальное количество символов;
- обеспечивается преобразование сообщения в сигнал с меньшей, чем у сообщения избыточностью (в пределе – без избыточности);

• при эффективном кодировании, учитывающем вероятности появления букв алфавита источника сообщений, удается построить коды с максимальной удельной энтропией на символ;

• решается задача согласования источника сообщений с каналом связи, в результате чего скорость передачи информации может быть приближена к пропускной способности канала;

• не требуется введения специальных разделительных символов (маркеров), как, например, в коде Морзе, для отделения одной кодовой комбинации от другой, так как ни одна комбинация эффективного кода не совпадает с началом другой, более длинной. Такое свойство кода называется «неприводимостью», и коды называются префиксными, или кодами без запятой.

Недостатки эффективных кодов:

• эффективные коды являются неравномерными, т.е. кодовые комбинации имеют различное количество символов;

• наибольший эффект оптимальные коды дают при кодировании исходного сообщения длинными блоками, поскольку при этом достигается равновероятность и статистическая независимость блоков. Однако блочное кодирование вызывает необходимость накапливать слова алфавита источника, прежде чем поставить им в соответствие определенную кодовую группу эффективного кода. Это приводит к задержкам при передаче и приёме сообщений, что затрудняет (или исключает) применение эффективных кодов в системах, работающих в реальном масштабе времени. В настоящее время эффективное кодирование (кодом Хаффмана) применяется при записи информации на магнитные носители (системы архивации) и в системах факсимильной связи;

• существенным недостатком эффективных кодов является то, что они непомехозащищённые. Любая одиночная ошибка при приёме переводит передаваемую комбинацию в другую, не равную ей по длительности, что влечет за собой неправильное декодирование целого ряда последующих кодовых групп. Такое специфическое влияние помех называется «треком ошибок». В чистом

виде эффективное кодирование можно применять только для каналов без помех.

Таким образом, непосредственная передача сообщений при применении эффективных кодов по каналу связи с шумами приводит к недопустимо большим искажениям (потере информации). Однако, эффективное кодирование, устраняющее статистическую избыточность в передаваемом сообщении, наилучшим образом подготавливает непрерывную кодовую последовательность, полученную после первичного кодирования сообщений источника, к последующему помехоустойчивому кодированию с помощью корректирующих кодов в кодере канала. Целенаправленное введение избыточности при помехоустойчивом кодировании путём добавления дополнительных проверочных символов в кодовые информативные группы позволяет при декодировании обнаруживать и исправлять ошибки, вызванные помехами.

В чистом виде оптимальные коды можно применять только в каналах без помех, и на практике такое кодирование является предварительной ступенью для последующего помехоустойчивого кодирования

Задание.

1. Построить код по методу Шеннона-Фано и проверить его оптимальность.

2. Построить код по методу Хаффмена и кодовое дерево.

3. Провести программный контроль выполнения 1,2 пунктов на примере случайных сообщений.

4. Подготовить отчет и сдать работу.

Контрольные вопросы.

1. Запишите выражение для средней длины кодового слова.
2. Поясните принцип кодирования сообщений в коде Шеннона-Фано.
3. Поясните принцип кодирования сообщений в коде Хаффмана.
4. В чем преимущество кодирования групп сообщений?
5. Поясните принцип кодирования сообщений оптимальными кодами.
6. Какой код называется оптимальным?

7. Способы построения оптимальных кодов.
8. Как оценивается эффективность ОНК?
9. Какие коды называются оптимальными неравномерными кодами?

Лабораторная работа. Код Хэмминга

Код Хэмминга – один из наиболее распространенных систематических кодов, имеющих простой и удобный для технической реализации алгоритм обнаружения и исправления одиночной ошибки.

Для вычисления основных параметров кода задается количество либо информационных символов, либо информационных комбинаций – $N = 2^{n_u}$. При помощи следующих формул вычисляются n и n_k:

$$2^{n_k} \geq n+1 \qquad 2^n = 2^{n_k} * 2^{n_u}$$

Соотношение между n, n_k и n_u для кода Хэмминга представлены в таблице :

N	**$n_и$**	**n_k**	**n**	**$n_и$**	**n_k**
1	0	1	**9**	5	4
2	0	2	**10**	6	4
3	1	2	**11**	7	4
4	1	3	**12**	8	4
5	2	3	**13**	9	4
6	3	3	**14**	10	4
7	4	3	**15**	11	4
8	4	4	**16**	11	5

Зная основные параметры корректирующего кода, определяют, какие позиции сигналов будут рабочими, а какие – контрольными.

Практика показала, что номера контрольных символов удобно выбирать по закону 2^i, где $i=0,1, 2, 3\ldots$ - натуральный ряд чисел. Номера контрольных символов в этом случае равны *1, 2, 4, 8, 16, 32…* Затем определяют значения контрольных коэффициентов (0 или 1), руководствуясь следующим правилом: *сумма единиц на проверочных позициях должна быть четной.* Если эта сумма четна – значение контрольного коэффициента 0, в противном случае – 1.

Проверочные позиции выбирают следующим образом. Составляют табличку для ряда натуральных чисел в двоичном коде. Число ее строк равно $n=n_u+n_k$. Первой строке соответствует проверочный коэффициент a_1, второй

a_2 и т.д.

0001	a_1	0101	a_5	1001	a_9
0010	a_2	0110	a_6	1010	a_{10}
0011	a_3	0111	a_7	1011	a_{11}
0100	a_4	1000	a_8		

Затем выявляют проверочные позиции, выписывая коэффициенты по следующему принципу: в первую проверку входят коэффициенты, которые содержат 1 в младшем разряде, т.е. $a_1, a_3, a_5, a_7, a_9, a_{11}$ и т.д.; во вторую – содержащие 1 во втором разряде, т.е. $a_2, a_3, a_6, a_7, a_{10}$ и т.д.; в третью – содержащие 1 в третьем разряде, и т.д. Номера проверочных коэффициентов соответствуют номерам проверочных позиций, что позволяет составить общую таблицу проверок (таблица 1).

Таблица 1

№ проверки	Проверочные позиции (П)	№ контрольного символа
1	1, 3, 5, 7, 9, 11,. . .	1
2	2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 24,…	2
3	4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22,23,…	4
4	8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31, 40, 41, 42,…	8

Пример. Требуется исправить любую одиночную ошибку при передаче комбинации 0101, т.е. $n_u=4$.

Решение. Согласно табл.1 минимальное число контрольных символов $n_k=3$, при этом $n=7$. Контрольные коэффициенты будут расположены на позициях 1, 2, 4. Составляем макет корректирующего кода и записываем его во вторую колонку в табл. 3. Пользуясь табл.2, определим значения коэффициентов $К_1$, $К_2$, $К_3$.

Первая проверка: сумма $П_1+П_3+П_5+П_7$ должна быть четной, а сумма $К_1+0+1+1$ будет четной при $К_1=0$.

Вторая проверка: сумма $П_2+П_3+П_6+П_7$ должна быть четной, а сумма $К_2+0+0+1$ будет четной при $К_2=1$.

Третья проверка: сумма $П_4+П_5+П_6+П_7$ должна быть четной, а сумма $К_3+1+0+1$ будет четной при $К_3=0$.

Окончательное значение искомой комбинации корректирующего кода записываем в третью колонку в таблице:

Позиция символов корректирующего кода	**Кодовое слово**	
	без значений контрольных коэффициентов	со значениями контрольных коэффициентов
1	$К_1$	0
2	$К_2$	1
3	0	0
4	$К_3$	0
5	1	1
6	0	0
7	1	1

Предположим, что в канале связи под действием помех произошло искажение и вместо 0100101 было принято 0100111. Для обнаружения ошибки производят проверки на четность.

Первая проверка: сумма $П_1+П_3+П_5+П_7=0+0+1+1$ четна. В младший разряд номера ошибочной позиции записываем 0.

Вторая проверка: сумма $П_2+П_3+П_6+П_7=1+0+1+1$ нечетна. Во второй разряд номера ошибочной позиции записываем 1.

Третья проверка: сумма $П_4+П_5+П_6+П_7=0+1+1+1$ нечетна. В третий разряд номера ошибочной позиции записываем 1.

Номер ошибочной позиции 011=6. Следовательно, символ шестой позиции следует изменить на обратный, и мы получим правильную кодовую комбинацию.

Задание.

1. Ознакомиться с теоретической частью, используя дополнительную литературу.

2. Построить код Хэмминга по заданным исходным данным (число информационных разрядов k).

3. Составить систему уравнений кодирования для определения проверочных разрядов для кода Хэмминга по пункту 2.

4. Провести программный контроль выполнения 2 и 3 пунктов на примере случайных кодовых комбинаций.

5. Подготовить отчет.

Контрольные вопросы.

1. На каких позициях располагаются проверочные символы в коде Хэмминга?

2. Что такое информационные и проверочные символы?

3. Какими графическими и геометрическими способами можно представить коды? Приведите пример.

4. Что такое кодовое расстояние, как оно определяется между двумя комбинациями двоичного кода?

5. Каким соотношением связаны информационные, проверочные символы и минимальное кодовое дерево?

Лабораторная работа. Линейные групповые коды

Систематический код – групповой n-значный код, в котором из n символов, образующих кодовую комбинацию, $n_и$ символов информационные, а $n_k = n - n_u$ –избыточные, предназначенные для проверки.

Систематические коды удобно задавать при помощи производящей матрицы. Число строк матрицы равно $n_и$, число столбцов равно n.

$$C=\begin{Vmatrix} a_{11}\ a_{12}\dots\ a_{1n_u} & P_{11}\ P_{12}\dots\ P_{1n_k} \\ a_{21}\ a_{22}\dots\ a_{2n_u} & P_{21}\ P_{22}\dots\ P_{2n_k} \\ \dots\ \dots\ \dots\ \dots & \dots\ \dots\ \dots\ \dots \\ a_{n_u1}\ a_{n_u2}\dots a_{n_un_u} & P_{n_u1}\ P_{n_u2}\dots P_{n_un_k} \end{Vmatrix}$$

Производящая матрица С может быть представлена при помощи двух матриц $И$ и $П$ (информационной и проверочной). Число столбцов матрицы $П$ равно n_k, число столбцов матрицы $И$ равно n_u.

$$C=\begin{Vmatrix} a_{11}\ a_{12}\dots\ a_{1n_u} \end{Vmatrix} \begin{Vmatrix} P_{11}\ P_{12}\ \dots\ P_{1n_k} \\ P_{21}\ P_{22}\ \dots\ P_{2n_k} \\ \dots\ \dots\ \dots\ \dots \\ P_{n_u1}\ P_{n_u2}\dots P_{n_un_k} \end{Vmatrix} = \begin{Vmatrix} a_{11}\ a_{12}\dots\ a_{1n_u} \\ a_{21}\ a_{22}\dots\ a_{2n_u} \\ \dots\ \dots\ \dots\ \dots \\ a_{n_u1}\ a_{n_u2}\dots a_{n_un_u} \end{Vmatrix} \begin{Vmatrix} P_{11}\ P_{12}\ \dots\ P_{1n_k} \\ P_{21}\ P_{22}\ \dots\ P_{2n_k} \\ \dots\ \dots\ \dots\ \dots \\ P_{n_u1}\ P_{n_u2}\dots P_{n_un_k} \end{Vmatrix} = \begin{Vmatrix} И \end{Vmatrix} \begin{Vmatrix} П \end{Vmatrix}$$

Теорией и практикой установлено, что в качестве матрицы *И* удобно брать единичную матрицу

$$\begin{Vmatrix} 10\ldots\ 000 \end{Vmatrix}$$
$$\begin{Vmatrix} 01\ldots\ 000 \end{Vmatrix}$$
$$\begin{Vmatrix} \ldots\ \ldots\ \ldots \end{Vmatrix}$$
$$\begin{Vmatrix} 00\ \ldots\ 001 \end{Vmatrix}$$

При выборе матрицы *П* исходят из следующих рассуждений: чем больше единиц в разрядах проверочной матрицы *П*, тем ближе соответствующий порождаемый код к оптимальному.

Критерием оптимальности таких кодов является соблюдение условия

$$2^{n-n_u} - 1 \geq \sum_{i=1}^{r} C_n^i$$

где r – число ошибок.

С другой стороны, число единиц в матрице *П* определяет число сумматоров по модулю 2 в шифраторе и дешифраторе, т.е. чем больше единиц в матрице *П*, тем сложнее аппаратура. Но даже если основным требованием к аппаратуре будет ее простота, вес каждой строки матрицы *П* должен быть не менее $W_П \geq d_0 - W_И$, где $W_И$ – вес соответствующей строки матрицы *И*. Если матрица *И* – единичная, то $W_И = 1$ (при $W_И > 1$ усложнилось бы как построение кодов, так и их техническая реализация).

Производящая матрица позволяет получить все возможные комбинации кода суммированием по модулю 2 всех возможных сочетаний строк матрицы.

Пример. Построить матрицу для группового кода, способного исправлять одиночную ошибку при передаче 16 символов первичного алфавита.

Кодовое расстояние $d_0 = 3$. Так как число информационных разрядов кода $n_u = 4$ $(16 = 2^4 = 2^{n_u})$, то число строк производящей матрицы *С* должно быть равно 4. Число столбцов матрицы *С* равно n; n – длина кода, в свою очередь, равна $n_u + n_k$; n_k – число корректирующих разрядов, равное

$$n_k = \log_2\{5 + [\log_2 5]\} = \log_2 8 = 3.$$

Следовательно, число столбцов, содержащих контрольные разряды, должно быть равно 3, а общее число столбцов матрицы C равно $n_u+n_k=4+3=7$.

Так как вес каждой строки проверочной матрицы $П$ должен быть

$$W_{П} \geq d_0 - W_{И},$$

то в качестве ее строк могут быть выбраны трехзначные двоичные комбинации с числом единиц ≥ 2: 111; 110; 101; 011.

$$C = \begin{pmatrix} 1000111 \\ 0100101 \\ 0010011 \\ 0001110 \end{pmatrix} \qquad C = \begin{pmatrix} 1000011 \\ 0100101 \\ 0010111 \\ 0001110 \end{pmatrix} \qquad C = \begin{pmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{pmatrix}$$

Как видно из примера, основным требованиям могут удовлетворять несколько матриц. Выбор той или иной из матриц, возможных для данного n_u, n_k, и d_0, определяется по дополнительным требованиям: минимум корректирующих разрядов или максимальная простота аппаратуры.

В процессе декодирования систематического кода осуществляются проверки, идея которых в общем виде может быть представлена следующим образом:

$$P_j \oplus \sum_{i=1}^{n_u} P_{ij} a_i = S_j, \qquad j=1, 2, \ldots n_k . \qquad (2)$$

Для каждой конкретной матрицы существует своя, одна - единственная система проверок. Проверки производятся по следующему правилу: в первую вместе с проверочным разрядом p_1 входят информационные разряды, соответствующие единицам первого столбца проверочной матрицы $П$, во вторую - второй проверочный разряд p_2 и информационные разряды, соответствующие единицам второго столбца проверочной матрицы и т.д. Число проверок равно числу проверочных разрядов корректирующего кода n_k. В результате проверок образуется проверочный вектор $S_1, S_2, \ldots S_{n_k}$, который называют *синдромом.* Если вес синдрома равен нулю, то принятая комбинация считается безошибочной. Но если хотя бы один разряд проверочного вектора содержит единицу, принятая комбинация содержит ошибку.

Исправление ошибки производится по виду синдрома, так как каждому ошибочному разряду соответствует один – единственный проверочный вектор.

Вид синдрома для каждой конкретной матрицы может быть определен при помощи контрольной матрицы *H*, представляющей собой транспонированную матрицу *П*, дополненную единичной матрицей *I*, число столбцов которой равно числу проверочных разрядов кода

$$H = \left\| П^T I_{n_k} \right\|$$

Столбцы такой матрицы – значение синдрома для разряда, соответствующего номеру столбца матрицы *H*.

Пример. Групповой код построен по матрице

$$C = \left\| \begin{matrix} 1\,0\,0\,0\,0\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1 \end{matrix} \right\|$$

Показать процесс исправления ошибки в произвольном разряде корректирующего кода, информационная часть которого – четырехразрядные комбинации натурального двоичного кода.

Решение. Кодовое расстояние $d_0 = 3$. Число проверочных символов $n_к=3$; $n_и=4$.

Производящая матрица *C* в виде информационной матрицы *И* и проверочной матрицы *П* может быть представлена следующим образом

$$C = \left\| \begin{matrix} 1\,0\,0\,0\,0\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1 \end{matrix} \right\| = \underbrace{\left\| \begin{matrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{matrix} \right\|}_{И} = \underbrace{\left\| \begin{matrix} 0\,1\,1 \\ 1\,0\,1 \\ 1\,1\,0 \\ 1\,1\,1 \end{matrix} \right\|}_{П}$$

Согласно принципу построения системы проверки (2) система проверок для кодов, построенных по матрице *C*, будет иметь вид

$$P_1 \oplus a_2 \oplus a_3 \oplus a_4 = S_1$$

$P_2 \oplus a_1 \oplus a_3 \oplus a_4 = S_2$

$P_3 \oplus a_1 \oplus a_2 \oplus a_4 = S_3$.

Чтобы знать, какая комбинация значений разрядов синдрома S_1, S_2, S_3 будет соответствовать ошибке в определенном разряде принятой комбинации, строим контрольную матрицу H, ее строками являются столбцы матрицы $П$, дополненные единичной транспонированной матрицей I, размерность которой определяется числом избыточных разрядов кода, т.е. в нашем случае равная 3.

$$H = \begin{Vmatrix} a_1 a_2 a_3 a_4 & P_1 P_2 P_3 \\ 0\ 1\ 1\ 1 & 1\ 0\ 0 \\ 1\ 0\ 1\ 1 & 0\ 1\ 0 \\ 1\ 1\ 0\ 1 & 0\ 0\ 1 \end{Vmatrix}$$

Если разряды синдрома соответствуют первому столбцу матрицы H, т.е. $S_1=0$, $S_2=1$, $S_3=1$, то ошибка в первом разряде принятой комбинации; если синдром имеет вид *101*, что соответствует второму столбцу матрицы H, то ошибка во втором разряде; синдром *001* соответствует ошибке в третьем проверочном разряде кода.

В качестве примера проверки корректирующих свойств кода используем комбинации кода вида

0 1 0 0 1 1 0

Находим проверочные векторы согласно системе проверок.

Для первой комбинации: $P_1=1$, $P_2=1$, $P_3=0$.

$P_1 \oplus a_2 \oplus a_3 \oplus a_4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$

$P_2 \oplus a_1 \oplus a_3 \oplus a_4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$

$P_3 \oplus a_1 \oplus a_2 \oplus a_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$

Синдром – *0 1 1* показывает, что в первом разряде символ следует заменить на обратный.

Задание.

1. Ознакомиться с теоретической частью, используя дополнительную литературу.

2. Исходя из полученных у преподавателя исходных данных (количества передаваемых сообщений N), рассчитать необходимое число информационных и контрольных разрядов для систематического кода, обнаруживающего и исправляющего одиночные ошибки.

3. Составить порождающую и проверочную матрицы, а также уравнения проверки по пункту 2, исходя из количества информационных разрядов.

4. Провести программный контроль выполнения 2 и 3 пунктов на примере некоторых случайных кодовых комбинаций рассчитанной ранее разрядности.

5. Отчет.

Контрольные вопросы.

1. Приведите классификацию корректирующих кодов по способу введения и использования избыточности, по структуре кода.

2. Какие среди систематических кодов имеют наибольшую практическую значимость и почему?

3. Что такое синдром ошибки?

4. Как получают проверочную матрицу при формировании систематических кодов и чем объясняется такое требование ее построения?

5. Какими выражениями удобно пользоваться для практических расчетов при определении числа контрольных разрядов с d=3? d=4?

Лабораторная работа. Циклические коды

Циклические коды получили такое название потому, что в них часть комбинаций кода или все комбинации могут быть получены путем циклического сдвига одной или нескольких комбинаций кода.

Циклические коды относятся к систематическим (строятся по строго определенному правилу). Кроме того, циклические коды относятся к числу блочных кодов, каждый блок является частным случаем буквы, кодируется самостоятельно.

Идея построения циклических кодов базируется на использовании неприводимых в поле двоичных чисел многочленов. Неприводимые называются

многочлены, которые могут быть представлены в виде произведения многочленов низших степеней с коэффициентом из того же поля, то есть неприводимые многочлены делятся без остатка только на себя или на 1. Идея коррекции ошибок в циклических кодах базируется на том, что разрешенные комбинации кода делятся без остатка на некоторый образующий многочлен, который выбирается из числа неприводимых многочленов.

В теории циклических кодов принято записывать кодовые комбинации в виде полинома некоторой фиктивной переменной x:

$$C_1a_1 \oplus C_2a_2 \oplus C_3a_3 \oplus \ldots \oplus C_ia_i \oplus \ldots \oplus C_qa_q \neq 0$$

где a_i – значение символа кодовой комбинации на позиции i при нумерации справа налево.

Пример: Представить в виде полинома кодовую комбинацию 1011101.

$$\begin{aligned} a(x) &= a^7x^{7-1} + a^6x^{6-1} + a^5x^{5-1} + a^4x^{4-1} + a^3x^{3-1} + a^2x^{2-1} + a^1x^{1-1} = \\ &= a^7x^6 + a^6x^5 + a^5x^4 + a^4x^3 + a^3x^2 + a^2x + a^1 = \\ &= 1\cdot x^6 + 0\cdot x^5 + 1\cdot x^4 + 1\cdot x^3 + 1\cdot x^2 + 0\cdot x + 1 = \\ &= x^6 + x^4 + x^3 + x^2 + 1. \end{aligned}$$

Максимальная степень x в полиноме на единицу меньше числа элементов в кодовой комбинации.

Чтобы обнаружить ошибку при делении на выбранный многочлен, надо чтобы все комбинации кода не делились не какой другой многочлен, а для этого необходимо, чтобы выбранный многочлен не разлагался на другие многочлены, был простым не приводимым многочленом.

Неприводимые многочлены в теории циклических кодов играют роль образующих многочленов. Так как если заданные кодовые комбинации умножить на выбранный неприводимый многочлен, то получиться циклический код, корректирующие способности которого определяются неприводимым многочленом. Необходимо уметь выделить из него возможные ошибочные разряды, то есть ввести некоторые опознаватели ошибок, которые могли бы выделить ошибочный блок из всех других.

Так как, циклические коды блочные, то каждый блок должен иметь свой опознаватель и тут решающую роль играют свойства образующего многочлена. Методика построения циклического кода такова, что образующий многочлен принимает участие в образовании каждой кодовой комбинации, поэтому любой многочлен циклического кода делится на образующий без остатка.

Без остатка делятся только те многочлены, которые принадлежат данному коду. Если при делении циклического кода на образующий многочлен будет получен остаток, то это значит, что в коде произошла ошибка или эта комбинация какого-то другого кода. Для декодирующего устройства это не имеет принципиальной разности.

По остатку обнаруживается ошибка. Остатки от деления многочленов является опознавателями ошибок циклических кодов. С другой стороны остатки от деления 1 с 0 на образующий многочлен используются для построения для построения образующих матриц.

Построение образующей матрицы сводиться к построению информационной матрицы, которая представляет собой единичную матрицу, но единицы идут по побочной диагонали и проверочной матрицы элементы которой представляют собой остатки от деления 1 с 0 на образующий многочлен. Но не все остатки от деления могут быть использованы в качестве элементов проверочной матрицы. Использовать можно только те остатки вес которых удовлетворяет следующими условию:

$$W \geq d_0 - 1$$

Длина остатка должна быть не меньше nk (числа контрольных разрядов). Число остатка должно быть равно числу информационных разрядов.

Образующая матрица может быть построена также в результате непосредственного умножения элементов единичной матрицы на образующий многочлен. Это часто бывает удобнее, чем нахождение остатков от деления. Полученные коды ни чем не отличаются от кодов построенных по образующим матрицам, в которых дополнительная матрица состоит из остатков от деления.

При построении кодов с минимальным кодовым расстоянием равным 3, n_k=3 и n_k=4 число комбинаций кодов полученным суммированием по модулю 2 всевозможных сочетаний строк образующей матрицы равно числу комбинаций полученных в результате циклического сдвига строки образующей матрицы и зеркальной комбинации. Но этот способ хорош лишь для получения кодов с малым числом n_i >=6, то число комбинаций от суммирования строк образующей матрицы растет гораздо быстрее чем число комбинаций полученных в результате сдвига. В последнем случае коды получаются избыточными, отсюда следует, падает скорость передачи информации. В таких случаях целесообразность применения того или иного метода кодирования может быть определенно из конкретных технических условий .

Построение декодирование циклических кодов сводиться к следующим процедурам:

1) расчеты n_k и n_i

2) выбор образующего многочлена. Образующий многочлен следует выбирать как можно более коротким, но степень его должна быть не меньше n_k, а число ненулевых членов должна быть не меньше минимального кодового расстояния.

3) выбор параметров образующей матрицы.

4) определение элементов проверочной матрицы.

5) обнаружение и исправление ошибок происходит по остаткам от деления принятой комбинации на образующий многочлен.

Если принятая комбинация делиться без остатка, то код принят без ошибки. Остаток от деления свидетельствует об ошибки, но не указывает какой именно. Чтобы найти ошибочный разряд и исправить его в циклических кодах принято осуществлять следующие процедуры:

1. Комбинация делиться на образующий многочлен
2. Подсчитывается количество единиц в остатке. Если вес остатка

$W \le S$, где S-число исправляемых данным кодом ошибок. То принятая комбинация складывается по модулю 2 с полученным остатком. Сумма дает исправленную комбинацию.

3. Если $W>S$. Производим циклический сдвиг влево принятой комбинации и затем делим полученную комбинацию на образующий многочлен.

Если $W<=S$, то складываем делимое с остатком, затем производим циклический сдвиг вправо комбинации, полученные в результате суммирования последнего делимого с остатком, и полученная комбинация уже не содержит ошибок.

Циклический сдвиг вправо производиться на столько разрядов на сколько была сдвинута суммированная с последним остатком комбинация относительно принятой комбинации.

Коды при использовании неприводимых многочленов подобны друг другу и обладают равноценными корректирующими способностями.

Сами многочлены называются обратными или двойственными многочленами

$$x^3 + x^2 + 1 \to 1101$$

$$x^3 + x + 1 \to 1011$$

Пример: Показать процесс исправления одиночной ошибки принятой кодовой комбинации. n_u=4 n_k=3 d_0=3

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

1	0	0	0	1	1	0	1	0	1	1
1	0	1	1							
		1	1	1	1					
		1	0	1	1					
			1	0	0	0				
			1	0	1	1				
					1	1				

S=2, сдвигаем на один символ влево и получим

```
0 0 0 1 1 0 1 | 1 0 1 1
      1 0 1 1 |-------
      -------   1 1 0
        1 1 0
```

Опять сдвигаем

```
0 0 1 1 0 1 0 |
    1 0 1 1   | 1 0 1 1
    -------   |-------
      1 1 0 0 |   1 1 1
      1 0 1 1 |
      -------
        1 1 1
```

Сдвигаем

```
0 1 1 0 1 0 0 |
  1 0 1 1     | 1 0 1 1
  -------     |-------
    1 1 0 0   |   1 0 1
    1 0 1 1   |
    -------
      1 1 1 0
      1 0 1 1
      -------
        1 0 1
```

Сдвигаем

```
1 1 0 1 0 0 0 |
1 0 1 1       | 1 0 1 1
-------       |-------
  1 1 0 0     |   1 0 1
  1 0 1 1     |
  -------
    1 1 1 0
    1 0 1 1
    -------
      1 0 1 0
      1 0 1 1
      -------
            1
```

Кодовую комбинацию складываем с остатком

```
  1 1 0 1 0 0 0
              1
  -------------
  1 1 0 1 0 0 1
```

Полученную комбинацию, сдвигаем вправо на столько разрядов, на сколько была сдвинута суммируемая с последним остатком комбинация относительно принятой комбинации

1 0 0 1 1 1 0

получили искомую комбинацию.

Коды обнаруживающие трёхкратные ошибки

Выбор образующего многочлена производиться исходя из числа контрольных разрядов и минимального кодового расстояния. Выбор числа корректирующих разрядов производиться из соотношения:

$$n_k \geq 1 + \log_2(n+1)$$

$$n_k \geq 1 + \log((n_u + 1) + \log(n_u + 1))$$

Выбор образующего многочлена производят исходя из следующих рассуждений: Для обнаружения трёхкратной ошибки

$d_0=r+1=4$ r- число обнаруживаемых ошибок

Степень образующего многочлена не может быть меньше 4, многочлен 3 степени имеющий число ненулевых членов больше или равным 3 позволяет обнаруживать все двойные ошибки.

Многочлен первой степени $x+1$ обнаруживает любое количество нечётных ошибок. Код, построенный с помощью такого многочлена во всех комбинациях содержит четное число единиц. Нарушение условия четности обнаруживается при делении принятой комбинации на многочлен $x+1$. Остаток будет во всех случаях, когда число ошибок нечётно.

Аналогично обнаруживаются ошибки и в кодах, имеющих более высокую разрядность. Таким образом, многочлен четвёртой степени полученный в результате умножения третей и первой степени обладает их корректирующими свойствами, то есть может обнаруживать 2, а также 3, то есть трехкратные ошибки.

Пример: Выбрать образующий многочлен минимальной возможной длины для построения циклического кода обнаруживающего все трехкратные ошибки.

c(x)=x+1

$$k_1(x) = x^3 + x + 1$$

$$k_2(x) = x^3 + x^2 + 1$$

необходимо перемножить.

$$k_2(x) * c(x) = x^4 + 0 + x + x^2 + 1$$
$$k_1(x) * c(x) = x^4 + x^2 + 0 + x^3 + 1$$

Построение образующей матрицы по данным образующим многочленом производят или с помощью нахождения остатков от деления 1 с нулями на образующий многочлен или умножением строк единичной матрицы на образующий многочлен. Обнаружение ошибок производиться по остаткам от деления принятой комбинации на образующий многочлен. Если остатка нет, то n_k отбрасываются и информационная часть кода используется по назначению.

Если в результате деления получается остаток, то комбинация блокируется. Такие коды могут обнаруживать 75% любого количества ошибок. Так как кроме двойной ошибки обнаруживаются все нечетные ошибки, но гарантировано количество ошибок, которых код никогда не пропустит равно 3, указать на ошибочные разряды при трехкратных искажениях, такие коды не могут.

Циклические коды, исправляющие двукратные ошибки

Методика построения циклических кодов d_0=>5 отличается от методики построения циклических кодов d_0<5, только в части выбора образующего многочлена в литературе эти коды известны как: БЧХ (Боуз, Чоутхури, Хотвинкем)

Построении образующего многочлена зависит от двух параметров:

- длины кодового слова n
- числа исправляемых ошибок.

Остальные параметры, участвующие в построении образующего многочлена в зависимости от заданного S и n, могут быть определены при помощи таблиц вспомогательных соотношений. Для исправления числа ошибок S>=2 ещё не достаточно условия, что между комбинациями кода минимальное кодовое расстояние $d_0=2S+1$ необходимо, чтобы длина кода удовлетворяющее условиям:

$$n = 2^h - 1$$

h – определяет выбор числа контрольных символов.

$$n_k \le hS = [\log_2(n+1)] * S$$

С другой стороны число контрольных символов определяется образующим многочленом и равно его степени. При больших значениях *h* длина кода становиться очень большой, что вызывает определенные трудности при технической реализации кодирующего и декодирующего устройств. При этом ni остается не использованной. В таких случаях определяется:

$$cn = 2^h - 1$$

где *c* является одним из сомножителей, на которые разлагается *h*, на *c* влияет выбор порядкового номера минимальных многочленов, так как индексы первоначально выбранных многочленов умножается на *c*.

Построение обратного многочлена производиться при помощи минимальных многочленов, которые являются простыми неприводимыми многочленами. Образующий многочлен представляет собой произведение нечетных минимальных многочленов и является их наименьшим общим кратным максимума.

Порядок минимальных многочленов равен:

$$\rho = 2S - 1$$

Порядок многочлена используют при определении числа сомножителей. Для построения образующего многочлена используют только нечетные многочлены и число их равно числу исправленных ошибок S.

Старшая степень минимального многочлена равна h. Степень образующего многочлена полученная в результате перемножения выбранных минимальных многочленов будет равна:

$$\beta = h * S$$

Каким бы методом не строились коды, повышение корректирующих способностей ведет к повышению избыточности декодирования кодов.

БЧХ производиться по той же методике, что декодирование циклических кодов когда *d0<5*. Однако в связи с тем, что коды БЧХ представлены комбинациями *n>=15* может возникнуть сложные варианты, когда для обнаружения

и исправления ошибок необходимо производить большое число циклических сдвигов в этом случае можно: комбинацию полученную после *k*-кратного сдвига и суммирования с остатком сдвигать не вправо, а влево на *n*-*k* циклических сдвигов. Это целесообразно делать только тогда, когда $k \geq \frac{n}{2}$

Задание.

1. Вычислить параметры кода *d, m, k, p, l, S*. Найти образующий многочлен, воспользовавшись таблицей неприводимых многочленов.

2. Проверить, имеются ли ошибки в исследуемой комбинации, при наличии ошибок – исправить их.

3. Провести программный контроль выполнения 1, 2 пунктов на примере случайных кодовых комбинаций.

4. Подготовить отчет и сдать работу.

Контрольные вопросы.

1. В чем заключаются основные идеи обнаружения и исправления ошибок циклическим кодом?

2. Что такое кодовое расстояние?

3. Чем отличается представление циклическим кодом для $d = 3$ и $d = 5$? где d - кодовое расстояние?

4. Какие существуют способы формирования комбинаций циклического кода?

5. В чем достоинство циклических кодов?

6. Что такое транспонированная матрица для циклического кода и ее размерность?

Список использованной литературы

1. Решетников М.Т. Методические указания к практическим занятиям по дисциплине «Теория информации» для студентов специальности 230102 – «Автоматизированные системы обработки информации и управления». Томск: ТУСУР, 2012. – 25 с.

2. Зверева Е.Н., Лебедько Е.Г. Сборник примеров и задач по основам теории информации и кодирования сообщений. – СПб: НИУ ИТМО, 2014. – 76 с.

3. Чикрин Д.Е. Теория информации и кодирования: курс лекций / Д.Е.Чикрин.- Казань: Казанский университет, 2013.-116с.

4. Фурсов В. А. Теория информации: учеб. / В.А. Фурсов. - Самара: Изд-во Самар, гос. аэрокосм, ун-та, 2011. - 128 с.

5. Думачев В.Н. Теория информации и кодирования - Воронеж: Воронежский институт МВД России, 2012.–200с.

6. Кавчук С.В. Сборник примеров и задач по теории информации. Руководство для практических занятий на базе Mathcad 6.0 Plus. Таганрог: Изд-во ТРТУ, 2002. 64 с.

7. Мисюткин В. И. Элементы теории информации: пособие по одноим. дисциплине для слушателей специальности 1-40 01 73 «Программное обеспечение информационных систем» заоч. формы обучения / В. И. Мисюткин. – Гомель : ГГТУ им. П. О. Сухого, 2015. – 87 с.

8. Дмитриев В.И. Прикладная теория информации: учебник для студентов вузов по специальности "Автоматизированные системы обработки информации и управления". – М.: Высшая школа, 1989. – 320 с.

9. Кузьмин И.В. Основы теории информации и кодирования / И.В. Кузьмин, В.А. Кедрус; 2-е изд., перераб. и доп. - Киев: Вища школа, 1986.-238 с.

10. Колесник В.Д. Курс теории информации / В.Д. Колесник, Г.Ш. Полтырев. – М.: Наука. Главная редакция физико-математической литературы. 1982. – 416 с.

Приложение

Для упрощения вычислений при решении задач приведена таблица значений величин $-p \log_2 p$ и таблица двоичных логарифмов целых чисел.

p	$-p \log_2 p$	p	$-p \log_2 p$	p	$-p \log_2 p$
0,01	0,0664	0,36	0,5306	0,71	0,3508
0,02	0,1129	0,37	0,5307	0,72	0,3412
0,03	0,1518	0,38	0,5305	0,73	0,3314
0,04	0,1858	0,39	0,5298	0,74	0,3215
0,05	0,2161	0,4	0,5288	0,75	0,3113
0,06	0,2435	0,41	0,5274	0,76	0,3009
0,07	0,2686	0,42	0,5256	0,77	0,2903
0,08	0,2915	0,43	0,5236	0,78	0,2796
0,09	0,3127	0,44	0,5211	0,79	0,2687
0,1	0,3322	0,45	0,5184	0,8	0,2575
0,11	0,3503	0,46	0,5153	0,81	0,2462
0,12	0,3671	0,47	0,5120	0,82	0,2348
0,13	0,3826	0,48	0,5083	0,83	0,2231
0,14	0,3971	0,49	0,5043	0,84	0,2113
0,15	0,4105	0,5	0,5000	0,85	0,1993
0,16	0,4230	0,51	0,4954	0,86	0,1871
0,17	0,4346	0,52	0,4906	0,87	0,1748
0,18	0,4453	0,53	0,4854	0,88	0,1623
0,19	0,4552	0,54	0,4800	0,89	0,1496
0,2	0,4644	0,55	0,4744	0,9	0,1368
0,21	0,4728	0,56	0,4684	0,91	0,1238
0,22	0,4806	0,57	0,4623	0,92	0,1107
0,23	0,4877	0,58	0,4558	0,93	0,0974
0,24	0,4941	0,59	0,4491	0,94	0,0839
0,25	0,5000	0,6	0,4422	0,95	0,0703
0,26	0,5053	0,61	0,4350	0,96	0,0565
0,27	0,5100	0,62	0,4276	0,97	0,0426
0,28	0,5142	0,63	0,4199	0,98	0,0286
0,29	0,5179	0,64	0,4121	0,99	0,0144
0,3	0,5211	0,65	0,4040		
0,31	0,5238	0,66	0,3956		
0,32	0,5260	0,67	0,3871		
0,33	0,5278	0,68	0,3783		
0,34	0,5292	0,69	0,3694		
0,35	0,5301	0,7	0,3602		

n	$\log_2 n$	n	$\log_2 n$	n	$\log_2 n$	n	$\log_2 n$	n	$\log_2 n$
1	0,0000	3	1,5850	5	2,3219	7	2,8074	9	3,1699
2	1,0000	4	2,0000	6	2,5850	8	3,0000	10	3,3219