

Содержание:

ВВЕДЕНИЕ

Кодирования информации - проблема, имеющая достаточно давнюю историю, гораздо более давнюю, нежели история развития вычислительной техники, которая обычно шла параллельно с историей развития проблемы сжатие и шифровки информации.

Все алгоритмы кодирования оперируют входным потоком информации, минимальной единицей которой является бит, а максимальной - несколько бит, байт или несколько байт.

Кодирование Хаффмана является простым алгоритмом для построения кодов переменной длины, имеющих минимальную среднюю длину. Коды Хаффмана преподаются во всех технических ВУЗах мира и, кроме того, входят в программу для углубленного изучения информатики в школе.

Поэтому изучение кодирования данных и методов кодирования является актуальным.

Объект исследования: кодирование и методы кодирования данных.

Предмет исследования: программные приложения, показывающие основные принципы кодирования.

Целью курсовой работы является изучения основ кодирования информации и анализ методов кодирования данных. Данная цель обусловила выделение следующих задач:

- рассмотреть основные понятия и принципы кодирования информации;
- изучить и провести анализ методов кодирования данных.

Структурно работа состоит из введения, двух параграфов, заключения и списка литературы.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОДИРОВАНИЯ ИНФОРМАЦИИ

1.1. Основы и основные понятия кодирования информации

Рассмотрим основные понятия, связанные с кодированием информации. Для передачи в канал связи сообщения преобразуются в сигналы. Символы, при помощи которых создаются сообщения, образуют первичный алфавит, при этом каждый символ характеризуется вероятностью его появления в сообщении. Каждому сообщению однозначно соответствует сигнал, представляющий определенную последовательность элементарных дискретных символов, называемых кодовыми комбинациями.

Кодирование - это преобразование сообщений в сигнал, т.е. преобразование сообщений в кодовые комбинации. Код - система соответствия между элементами сообщений и кодовыми комбинациями. Кодер - устройство, осуществляющее кодирование. Декодер - устройство, осуществляющее обратную операцию, т.е. преобразование кодовой комбинации в сообщение. Алфавит - множество возможных элементов кода, т.е. элементарных символов (кодовых символов) $X = \{x_i\}$, где $i = 1, 2, \dots, m$. Количество элементов кода - m называется его основанием. Для двоичного кода $x_i = \{0, 1\}$ и $m = 2$. Конечная последовательность символов данного алфавита называется кодовой комбинацией (кодовым словом). Число элементов в кодовой комбинации - n называется значностью (длиной комбинации). Число различных кодовых комбинаций ($N = m^n$) называется объемом или мощностью кода.

Цели кодирования:

- 1) Повышение эффективности передачи данных, за счет достижения максимальной скорости передачи данных.
- 2) Повышение помехоустойчивости при передаче данных.

В соответствии с этими целями теория кодирования развивается в двух основных направлениях:

1. экономичного (эффективного,) кодирования занимается кодов, позволяющих в без помех эффективность передачи за счет устранения источника и наилучшего скорости данных с пропускной канала связи.

2. помехоустойчивого кодирования поиском , повышающих достоверность информации в каналах с [\[1\]](#)

Научные основы были К. Шенноном, который процессы передачи по техническим каналам (теория , теория кодирования). При подходе кодирование в более узком : как переход от информации в одной системе к представлению в символьной системе. , преобразование русского текста в код Морзе для передачи его по связи или радиосвязи. кодирование с потребностью приспособить код к техническим средствам с информацией.

Декодирование — обратного кода к форме символьной системы, т.е. исходного сообщения. : перевод с Морзе в письменный на русском языке.

В широком смысле — это процесс содержания закодированного . При таком подходе записи текста с русского можно рассматривать в кодирования, а его чтение — это .

Способ кодирования и того же может быть . Например, русский мы привыкли записывать с русского . Но то же самое можно , используя английский . Иногда так приходится , посылая SMS по телефону, на котором нет букв, или отправляя письмо на русском из-за , если на компьютере нет программного обеспечения. , фразу: «Здравствуй, Саша!» писать так: «Zdravstvui, Sasha!».

Существуют и способы кодирования . Например, — быстрый способ устной речи. Ею лишь немногие обученные — стенографисты. Стенографист записывать текст с речью говорящего . В стенограмме значок обозначал слово или словосочетание. (декодировать) стенограмму только .

Приведенные примеры следующее важное : для кодирования одной и той же могут использованы разные ; их выбор зависит от обстоятельств: цели , условий, средств. Если записать текст в речи — используем ; если передать текст за — используем английский ; если надо текст в , понятном для грамотного человека, — записываем его по грамматике русского .

Еще одно обстоятельство: выбор кодирования информации быть связан с способом ее . Покажем это на примере чисел — количественной . Используя русский , можно

число «тридцать». Используя же алфавит десятичной системы, пишем: Второй способ не короче первого, но и для выполнения вычислений. запись для выполнения расчетов: « пять умножить на сто семь» или «35 x 127»? — вторая.

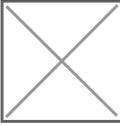
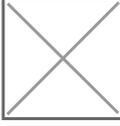
если важно число без искажения, то его записать в текстовой. Например, в документах часто записывают в текстовой: «триста семьдесят руб.» «375 руб.». В случае искажение цифры изменит все. При использовании формы даже ошибки могут не смысла. Например, человек: «Тристо семдесят пят.». Однако смысл.

В некоторых случаях потребность текста сообщения или, для того чтобы его не прочитать те, кому не. Это называется от несанкционированного доступа. В случае секретный шифруется. Шифрование собой превращения открытого в зашифрованный, а дешифрование — обратного преобразования, при восстанавливается текст. Шифрование — это кодирование, но с засекреченным, известным только и адресату. шифрования занимается под названием криптография. [2]

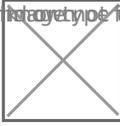
имеется сообщение, при помощи «алфавита», содержащего p «». Требуется «закодировать» это, т.е. указать правило, каждому сообщению определенную из t различных «элементарных», составляющих «алфавит». Мы будем кодирование тем более, чем меньше элементарных приходится затратить на сообщения. считать, что каждый из сигналов продолжается и то же время, то наиболее код позволит на передачу сообщения всего времени.

свойством случайных является полной уверенности в их, создающее известную при выполнении связанных с событиями. Однако совершенно, что степень этой в различных случаях совершенно. Для практики важно численно оценивать неопределенности самих опытов, иметь возможность их с этой стороны. два

независимых опыта  и  а сложный , состоящий в  одновременном опытов  и . Пусть  имеет k равновероятных, а  опыт  равновероятных исходов, что неопределенность опыта  неопределенности опыта , так как k  здесь еще неопределенность

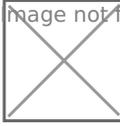
исхода . Естественно считать, что неопределенности опыта  сумме ,
характеризующих опыты  и , т.е.

.

:
, , 

при  удовлетворяет только функция - :

.

Рассмотрим A , состоящий из  и имеющих вероятности . общая
неопределенность для A будет равна:

.

Это число называть энтропией  и обозначать через .

число букв в « Σ » равно p , а используемых элементарных равно t , то при любом
кодирования среднее элементарных , приходящихся на одну алфавита, не может
меньше чем ; однако он может сделано сколь близким к этому , если только
кодвые сопоставлять сразу длинными «блоками», из большого числа . [3]

Мы рассмотрим лишь простейший сообщений, записанных при некоторых p «букв», проявления на любом месте полностью характеризуется p_1, p_2, \dots, p_n , где, разумеется, $p_1 + p_2 + \dots + p_n = 1$, при котором p_i проявления i -й на любом месте предполагается одной и той же, вне от того, какие стояли на предыдущих местах, т.е. буквы сообщения друг от друга. На деле в сообщениях это чаще не так; в частности, в русском вероятность появления той или буквы зависит от предыдущей. Однако строгий взаимной зависимости сделал бы все рассмотрения очень, но никак не изменит результаты.

Мы будем рассматривать коды; обобщение при этом результатов на, использующие произвольное t элементарных, является, как всегда, простым. Начнем с случая кодов, отдельное обозначение – последовательность 0 и 1 – каждой «букве». Каждому двоичному для n -буквенного может быть некоторый метод некоторого загаданного x , не превосходящего n , при вопросов, на которые лишь «да» (1) или «нет» (0), что и приводит нас к двоичному. При заданных p_1, p_2, \dots, p_n отдельных букв многобуквенного сообщения экономный код будет тот, для при этих вероятностях n значений x значение числа вопросов (двоичных: 0 и 1 или элементарных) оказывается наименьшим.

всего, среднее двоичных элементарных, приходящихся в сообщении на одну исходного сообщения, не быть меньше H , где $H = -p_1 \log p_1 - p_2 \log p_2 - \dots - p_n \log p_n$ – опыта, в распознавании одной текста (или, просто энтропия буквы). сразу следует, что при методе кодирования для длинного сообщения из M требуется не чем MH двоичных знаков, и не может превосходить бита.

Если p_1, p_2, \dots, p_n не все равны собой, то $H < \log n$; поэтому думать, что учет закономерностей сообщения позволить код более экономичный, чем равномерный код, требующий не $M \log n$ двоичных знаков для текста из M .

1.2. Классификация назначения и представления кодов

можно классифицировать по признакам:

1. По (количеству символов в): бинарные (двоичные $=2$) и не бинарные ($m \neq 2$).
2. По длине комбинаций (n): равномерные, если все комбинации имеют длину и неравномерные, длина комбинации не постоянна.

3. По передачи: последовательные и ; блочные - данные помещаются в , а потом передаются в и бинарные непрерывные.

4. По : простые (примитивные,) - для передачи используют все возможные комбинации (без); корректирующие (помехозащищенные) - для сообщений не все, а только часть () кодовых комбинаций.

5. В от назначения и применения можно следующие типы :

Внутренние коды - это , используемые внутри . Это машинные , а также коды, на использовании позиционных счисления (двоичный, , двоично-десятичный, , шестнадцатеричный и др.). Наиболее кодом в ЭВМ является код, который позволяет реализовать устройства для хранения, и передачи данных в коде. Он обеспечивает надежность и простоту выполнения над данными в двоичном . Двоичные данные, в группы по 4, шестнадцатеричный код, который согласуется с архитектурой ЭВМ, с данными кратными (8 бит). [4]

для обмена данными и их по каналам связи. распространение в ПК получил код (American Code for Information). ASCII - это 7-битный код и других символов. ЭВМ работают с , то 8-й разряд используется для или проверки на четность, или кода. В ЭВМ фирмы IBM расширенный код для обмена информацией (Extended Binary Decimal Interchange). В каналах широко используется код МККТТ (международный комитет по телефонии и) и его модификации (и др.).

При кодировании информации для по каналам связи, в том внутри аппаратным , используются , обеспечивающие максимальную передачи информации, за ее сжатия и устранения (например: Хаффмана и Шеннона-Фано), и обеспечивающие достоверность данных, за счет избыточности в сообщения (например: коды, Хэмминга, и их разновидности).

Коды для применений - это , предназначенные для решения задач передачи и данных. Примерами кодов циклический код Грея, широко используется в АЦП и линейных перемещений. Фибоначчи для построения быстродействующих и АЦП.

В зависимости от применяемых кодирования, используют математические кодов, при этом часто применяется кодов в виде: матриц; деревьев; многочленов; фигур и т.д. Рассмотрим способы представления .

Матричное кодов. Используется для равномерных n - значных . Для примитивного (полного и) кода содержит n - столбцов и $2n$ - , т.е. код использует все сочетания.

Для (корректирующих, обнаруживающих и ошибки) содержит n - столбцов ($n = k+m$, где k - число информационных, а m - число разрядов) и 2^k - строк (2^k - число кодовых комбинаций). При значениях n и k матрица слишком громоздкой, при код записывается в виде. Матричное кодов используется, в линейных групповых, кодах и т.д.

Представление кодов в кодовых деревьях. дерево - связной, не содержащий. Связной граф - , в котором для любой вершин существует, соединяющий эти. Граф состоит из (вершин) и ребер (), соединяющих узлы, на разных. Для построения дерева двоичного кода вершину называемую дерева () и из нее проводят ребра в две вершины и т.д. [5]

1.3. Метод Хаффмана

Метод или сжатия на основе двоичных деревьев был предложен Д.А. в 1952 году до появления цифрового компьютера. высокой эффективностью, он и его адаптивные версии в основе методов, используемых в алгоритмах кодирования. Код редко используется, чаще в связке с другими кодирования. Метод является примером кодов длины, имеющих среднюю длину. метод производит сжатие, то сжимает данные до их, если вероятности точно равны степеням 2.

Этот метод состоит из двух этапов:

- Построение кодового.
- Построение отображения на основе построенного.

Алгоритм основан на том, что символы из 256-символьного набора в тексте могут чаще среднего повтора, а - реже. Следовательно, для записи распространенных использовать короткие бит, длиной 8, а для записи редких - длинные, то суммарный файла уменьшится. В получается данных в виде («двоичное дерево»).

$A = \{a_1, a_2, \dots, a_n\}$ - алфавит из n различных, $W = \{w_1, w_2, \dots, w_n\}$ - соответствующий ему положительных целых. Тогда набор кодов $C = \{c_1, c_2, \dots, c_n\}$, такой что:

- c_i не префиксом для c_j , при $i \neq j$; ($|c_i|$ длина кода c_i) минимально-избыточным префиксным или иначе кодом. [6]

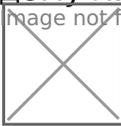
Бинарным называется ориентированное, полустепень исхода из вершин которого не двух.

бинарного дерева, захода которой нулю, называется . Для остальных дерева полустепень равна единице.

T- бинарное дерево, $=(0,1)$ - двоичный и каждому ребру приписана одна из алфавита таким , что все ребра, из одной вершины, различными буквами. любому листу можно уникальное кодовое , образованное из букв, помечены ребра, при движении от к соответствующему листу. описанного способа в том, что полученные коды префиксными.

, что стоимость хранения , закодированной при помощи , равна сумме путей из к

каждому листу , взвешенных частотой кодового слова или взвешенных : 

где  - частота кодового длины  во входном . Рассмотрим в качестве

кодировку в стандарте ASCII. каждый символ собой кодовое фиксированной(8)

длины, поэтому хранения определится  , где W - количество кодовых во входном .

Поэтому стоимость 39 кодовых слов в ASCII равна 312, от относительной отдельных символов в потоке. Алгоритм позволяет уменьшить хранения кодовых слов такого подбора кодовых слов, минимизирует взвешенных путей. называть дерево с длиной путей Хаффмана.

алгоритм Хаффмана на получает таблицу встречаемости символов в . Далее на этой таблицы дерево кодирования (H-дерево). [\[7\]](#)

1. Символы алфавита список свободных . Каждый лист вес, который может равен вероятности, либо вхождений символа в сообщение;
2. Выбираются два узла с наименьшими весами;
3. их родитель с весом, их суммарному весу;
4. добавляется в свободных узлов, а два его удаляются из этого ;
5. Одной дуге, из родителя, в соответствие бит 1, другой - бит 0;
6. , начиная со второго, до тех пор, пока в списке узлов не только один узел. Он и будет корнем дерева.

, у нас есть таблица частот.

1

Таблица частот

15 7 6 6 5

А Б В Г Д

На шаге из листьев выбираются два с весами - Г и Д. Они присоединяются к узлу-родителю, вес устанавливается $5+6=11$. узлы Г и Д из списка свободных. Г соответствует ветви 0, узел Д - ветви 1.

На шаге то же с узлами Б и В, так как теперь эта имеет самый вес в дереве. Создается узел с 13, а узлы Б и В удаляются из свободных.

На следующем «наилегчайшей» парой узлы Б/В и Г/Д.

Для них еще раз родитель, теперь уже с 24. Узел Б/В соответствует 0 родителя, Г/Д - ветви 1.

На шаге в свободных осталось 2 узла - это узел А и Б (Б/В)/(Г/Д). В очередной раз родитель с 39, и бывшие свободные присоединяются к разным его.

Поскольку свободным только узел, то алгоритм дерева кодирования завершается.

Каждый, входящий в, определяется как конкатенация и единиц, сопоставленных дерева Хаффмана, на от корня к листу.

Для данной символов коды будут выглядеть, как в табл. 2.

ица 2

Коды Хаффмана

А 01

Б 100

В 101

Г 110

Д 111

частый символ А закодирован наименьшим бит, а наиболее символ Д - наибольшим. хранения кодированного, определенная как сумма взвешенных, определится выражением что существенно меньше хранения входного (312).

ни один из полученных не является префиксом, они могут быть декодированы при их из потока.

Алгоритм предполагает просмотр битов и синхронное от корня по дереву Хаффмана в со считанным значением до тех пор, не будет достигнут, то есть очередное кодовое, после чего следующего слова начинается с дерева. [8]

Классический Хаффмана имеет существенный недостаток. Для содержимого сообщения декодер знает таблицу, которой пользовался. Следовательно, сжатого сообщения на длину таблицы, которая должна впереди, что может свести на нет все по сжатию сообщения. того, необходимость полной статистики перед собственно кодирования двух проходов по: одного для модели сообщения (частот и дерева), другого собственно для.

2. АНАЛИЗ КОДИРОВАНИЯ ДАННЫХ

2.1. кодирование

Логическое изменяет поток бит кадра в последовательность символов, подлежат физическому для транспортировки по каналу. Для логического используют разные:

- 4В/5В — каждые 4 входного потока 5-битным (табл 1.1). Получается избыточность, так как $2^4 = 16$ комбинаций показываються из $2^5 = 32$. по количеству битовых составляют: $(5-4)/4 = 1/4$ Такая избыточность определить ряд символов, которые

для синхронизации. Применяется в 100, FDDI

- 8В/10В — схема (8 бит 10-битным символом) но уже равна 4 раза входных в 1024).

- 5В/6В — 5 бит потока кодируются 6- символами. Применяется в 100

- 8В/6Т — 8 бит входного кодируются троичными (Т = ternary) (-,0,+). К примеру: 00h: 01h: 0+-+=0; Код имеет $36/28 = 2,85$. Скорость транспортировки в линию является битовой скорости и их на кодирования. в 100BaseT4. [\[9\]](#)

Вставка бит — схема работает на недопустимых последовательностей бит. Ее объясним на в протоколе HDLC. Тут поток смотрится как последовательность бит, для которой из более чем смежных 1 анализируется как служебный (пример: 01111110 флагом-разделителем кадра). в транслируемом встречается непрерывная из 1, то после пятой в выходной передатчик 0. Приемник анализирует цепочку, и если цепочки 011111 он видит 0, то он его и последовательность 011111 присоединяет к выходному потоку . Если принят 1, то последовательность 011111смотрится как служебный . Такая решает две задачи — длинные монотонные , которые неудобные для физического и разрешает опознание кадра и особых в непрерывном битовом .

Таблица 3

4В/5В

Входной Выходной символ

(0) 11110

0001 (1)

0010 (2)

0011 (3) 10101

(4) 01010

0101 (5)

0110 (6) 01110

(7) 01111

(8) 10010

1001 (9)

1010 (A) 10110

(B) 10111

1100 (C)

1101 (D)

1110 (E) 11100

(F) 11101

Служебный Выходной символ

11111

J

К 10001

T 01101

R

S 11001

Quiet

Halt 00100

кодирования облегчить задачи кодирования — исключить битовые последовательности, спектральные физического сигнала и кодирование пишет правила дискретных символов, логического в результат физические линии.

Физические могут иметь (аналоговую) — бесконечное число , из которого выбирают распознаваемое множество. На физических вместо битовой (бит/с) используют изменения сигнала в измеряется в (baud). Под таким определяют число различных состояний за единицу . На физическом уровне и передатчика. Внешнюю не используют из-за реализации еще канала. Много физического кодирования самосинхронизирующимися — они разрешают синхросигнал из последовательности состояний . [\[10\]](#)

2.2. Скремблирование

Скремблирование на уровне разрешает очень спектральные характеристики по некоторой полосе . Очень сильные искажают каналы передачи. При о физическом кодировании, использование следующие :

- Транзитное — информативным есть из одного состояния в
- Потенциальное кодирование — есть сигнала в конкретные времени
- Полярное — одной полярности для представления значения, сигнал полярности для — другого. При транспортировке вместо используют импульса

- Униполярное — одной полярности для представления одного , нулевой — для другого
- Биполярное — отрицательное, положительное и значения для представления состояний
- — в каждом битовом присутствует переход из состояния в другое, что для выделения . [\[11\]](#)

2.3. Схемы кодирования, применяются в локальных

AMI/ABP

AMI — Alternate Inversion или же ABP — bipolare, биполярная , которая использует +V, 0V и -V. Все нулевые биты значения 0V, — чередующимися значениями +V, -V (.1). Применяется в DSx (DS1 — 4), ISDN. Такая не есть самосинхронизирующей — длинная нулей приведет к синхронизации.

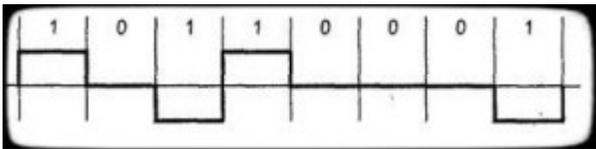


Рисунок 1

MAMI — Alternate Mark , или же ASI — модифицированная схема AMI, чередующейся полярности 0, а 1 — нулевым . Применяется в ISDN (— интерфейсы).

B8ZS

— Bipolar with 8 Substitution, аналогичная AMI, но для синхронизации цепочки 8 и более (за счет вставки).

HDB3

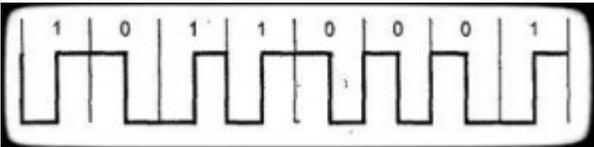
3 — High Density 3, схема аналогичная AMI, но не передачи цепочки трех . Вместо последовательности из нулей вставляется из четырех биполярных . (Рис.2)

Полярность последнего импульса	Предыдущая замена	
	000+ или +00+	000- или -00-
+	-00-	000+
-	000-	+00+

2

Манчестерское кодирование

encoding — двухфазное самосинхронизирующееся кодирование. бит узнается по смены состояния в битового интервала: от -V к +V: 1. От +V к -V: 0. в начале интервала и не быть. в Ethernet. (В начальных — униполярное). (рис.3)



3

Дифференциальное манчестерское

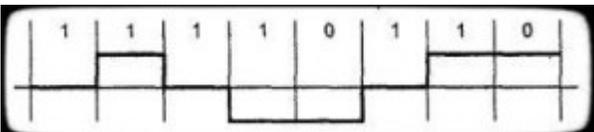
Differential encoding — двухфазное самосинхронизирующиеся код. Текущий бит по наличию перехода в битового (рис. 4.1), например 0 — переход (Вертикальный), 1 — нет перехода (горизонтальный). Можно и определять 0 и 1. В середине интервала переход всегда. Он нужен для . В Token применяется измененная такой схемы, где бит 0 и 1 определенны также два j и k (Рис. 4.2). нет переходов в середине. Бит K имеет переход в интервала, а j — нет.



Рисунок 4

-3

Трехуровневое со скремблированием который не . Используются уровни (+V, 0, -V) в линии каждого интервала. При 0 значения не меняются, при 1 — меняются на соседние по +V, 0, -V, 0, +V и тд. (рис. 5). Такая является вариантом NRZI. в FDDI и 100BaseTX.

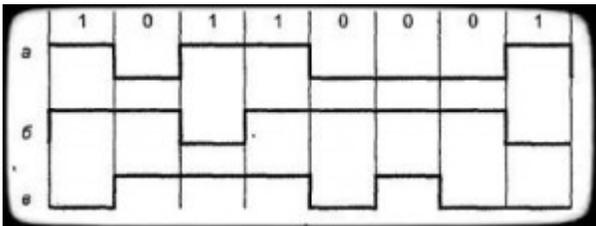


5

NRZ и NRZI

NRZ — Non-return to (без к нулю), биполярная схема (состояния на границе), которая 2 варианта. вариант это недифференциальное NRZ (в RS-232) состояние отражает значение (рис. 6.а). В варианте — дифференциальном, NRZ меняется в начале интервала для 1 и не меняется для 0. (). Привязки 1 и 0 к состоянию нету.

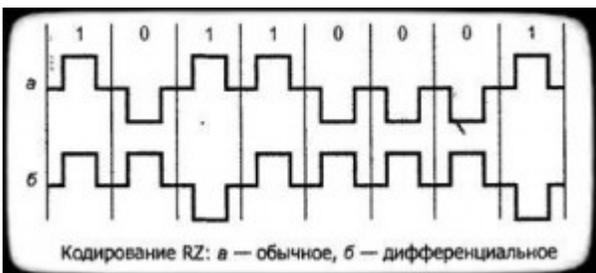
— Non-return to zero , измененная схема NRZ (. 6.в). Тут состояния на противоположные в начале интервала 0, и не меняются при 1. Возможна и обратная представления. в FDDI, 100BaseFX.



6

RZ

RZ — Return to zero (с к нулю), биполярная самосинхронизирующаяся . Состояние в определенный битового интервала возвращается к нулю. дифференциальный/недифференциальный . В дифференциальном привязки 1 и 0 к нету. (рис. 7.а).



7

FM 0

FM 0 — Frequency Modulation 0 (модуляция), полярный код. Меняется на на границе каждого интервала. При передаче 1 в битового состоянии неизменное. При 0, в середине битового состоянии меняется на . (рис. 8). в LocalTalk.

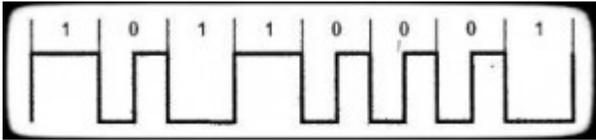


Рисунок 8

РАМ 5

РАМ 5 — Amplitude Modulation, биполярное кодирование, где бит в зависимости от оказывается одним из 5 потенциала. Нужен полоса частот (ниже скорости).

Используется в 1000.

2В1Q

Здесь бит оказывается одним символом (symbol), где каждому один из 4 уровней сигнала. В ице показано представление в сети .

Таблица 4

Представление в сети ISDN

Четверичный символ , В

00 -3	-2,5
01 -1	-0,883
10 +3	+2,5
11 +1	

4В3Т

4В3Т — из 4 бит (16 состояний) кодируется троичными символами (27). Из множества методов изменений MMS43, который в интерфейсе BRI сетей (таблица).

Тут специальные методы для постоянной составляющей в линии, в следствии кодирования комбинаций зависит от — состояния, где находится . Пример: последовательность бит 1101 представлена как: + + + — 0 -.

Таблица 5

4 — блок из 4 бит (16 состояний)

код S1	Переход S2	Переход S3	S4	Переход
0 — + S1	0 — + S2	0 — + S3	0 — + S4	
0111 — 0 + S1	— 0 + S2	— 0 + S3	— 0 + S4	
0100 — + 0 S1	— + 0 S2	— + 0 S3	— + 0 S4	
+ — 0 S1	+ — 0 S2	+ — 0 S3	+ — 0 S4	
1011 + 0 — S1	+ 0 — S2	+ 0 — S3	+ 0 — S4	
1110 0 + — S1	0 + — S2	0 + — S3	0 + — S4	
+ — + S2	+ — + S3	+ — + S4	— — — S1	
0011 0 0 + S2	0 0 + S3	0 0 + S4	— — 0 S2	
1101 0 + 0 S2	0 + 0 S3	0 + 0 S4	— 0 — S2	
+ 0 0 S2	+ 0 0 S3	+ 0 0 S4	0 — — S2	
0110 — + + S2	— + + S3	— — + S2	— — + S3	
+ + — S2	+ + — S3	+ — — S2	+ — — S3	
1111 + + 0 S3	0 0 — S1	0 0 — S1	0 0 — S3	

0000 + 0 + S3 0 — 0 S1 0 — 0 S2 0 — 0 S3

0 + + S3 — 0 0 S1 — 0 0 S2 — 0 0 S3

1100 + + + S4 — + — S1 — + — S2 — + — S3

ЗАКЛЮЧЕНИЕ

В научного исследования по «Методы данных» был проведен литературы, статьей по теме, изучена документация.

В результате исследования была достигнута поставленная цель –изучение основ кодирования информации в частности анализ методов кодирования. Цель курсовой работы достигнута за счёт выполнения следующих задач:

- рассмотрены основные понятия и принципы кодирования информации;
- проведен анализ методов кодирования данных.

После выполнения целей и задач курсовой работы были сделаны следующие выводы.

Проблема кодирования информации, имеет достаточно давнюю историю, гораздо более давнюю, нежели история развития вычислительной техники, которая обычно шла параллельно с историей развития проблемы сжатие и шифровки информации.

До появления работ Шеннона, Фано а позже и Хаффмана, кодирование символов алфавита при передаче сообщения по каналам связи осуществлялось одинаковым количеством бит, получаемым по формуле Хартли. С появлением этих работ начали появляться способы, кодирующие символы разным числом бит в зависимости от вероятности появления их в тексте, то есть более вероятные символы кодируются короткими кодами, а редко встречающиеся символы - длинными (длиннее среднего).

Преимуществами данных методов являются их очевидная простота реализации и, как следствие этого, высокая скорость кодирования и декодирования. Основным недостатком является их не оптимальность в общем случае.

СПИСОК ЛИТЕРАТУРЫ

1. Бородакий Ю.В. Информационные технологии: методы, процессы, системы / Ю.В. Бородакий, Ю.Г. Лободинский. - М.: Радио и Связь, 2015. - С. 455.
 2. Глушков В.М. Основы безбумажной информатики / В.М. Глушков//М.: Наука. Главная редакция физ.-мат литературы, 2014. - С. 552.
 3. Волков В.Б. Информатика / В.Б. Волков, Н.В. Макарова – СПб.: Питер, 2016 – 576с.
 4. Галисеев Г.В. Программирование в среде Delphi 7 / Г.В. Галисеев – М.: Вильямс, 2014. – 288с.
 5. Иванова Г.С. Технология программирования / Г.С. Иванова – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 320с.
 6. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, Е.К Нгуен – Киев: ДиаСофт, 2015. – 544с.
 7. Майерс Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер – М.: «Диалектика», 2015 – 272с.
 8. Меняев М.Ф. Информатика и основы программирования / М.Ф. Меняев – М.: Омега-Л, 2017 – 458с.
 9. Egan M. Grounded Theory Research and Theory Building // Advances in Developing Human Resources. – 2016. – №3. – P. 277-295.
 10. Handbook of research design and social measurement / Ed. by D.C. Miller, N.J. Salkind. – Thousand Oaks; London; New Delhi: Sage, 2014. – 808 p.
 11. Страусс А., Корбин Дж. Основы качественного исследования: обоснованная теория, процедуры и техники. – М.: Эдиториал УРСС, 2015. – 256 с.
-
1. Бородакий Ю.В. Информационные технологии: методы, процессы, системы / Ю.В. Бородакий, Ю.Г. Лободинский. - М.: Радио и Связь, 2015. - С. 455. [↑](#)
 2. Глушков В.М. Основы безбумажной информатики / В.М. Глушков//М.: Наука. Главная редакция физ.-мат литературы, 2014. - С. 552. [↑](#)
 3. Волков В.Б. Информатика / В.Б. Волков, Н.В. Макарова – СПб.: Питер, 2016 – 576с. [↑](#)

4. Галисеев Г.В. Программирование в среде Delphi 7 / Г.В. Галисеев – М.: Вильямс, 2014. – 288с. [↑](#)
5. Иванова Г.С. Технология программирования / Г.С. Иванова – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 320с. [↑](#)
6. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, Е.К Нгуен – Киев: ДиаСофт, 2015. – 544с. [↑](#)
7. Майерс Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер – М.: «Диалектика», 2015 – 272с. [↑](#)
8. Меняев М.Ф. Информатика и основы программирования / М.Ф. Меняев – М.: Омега-Л, 2017 – 458с. [↑](#)
9. Egan M. Grounded Theory Research and Theory Building // Advances in Developing Human Resources. – 2016. – №3. – P. 277-295. [↑](#)
10. Handbook of research design and social measurement / Ed. by D.C. Miller, N.J. Salkind. – Thousand Oaks; London; New Delhi: Sage, 2014. – 808 p. [↑](#)
11. Страусс А., Корбин Дж. Основы качественного исследования: обоснованная теория, процедуры и техники. – М.: Эдиториал УРСС, 2015. – 256 с. [↑](#)