1. 1. Понятие и свойства информации. Носитель информации

Информация - любые материально зафиксированные *следы*, образованные взаимодействием предметов или сил и поддающиеся *пониманию*.

Основные свойства:

- приносит сведения, которых не было;

- нематериальна, но проявляется в виде материальных носителей;

- воспринимается только тем получателем, который может распознать знаки.

Носитель информации - *знак*

1. 2. Понятие знака. Классификация знаков по Ч.Пирсу (с примерами). Классификация знаков по способу восприятия.

Формула:

X *понимает и использует* ***Y*** *в качестве представителя* ***Z***

где:

X - пользователь знака;

Y - знак (форма, материальный носитель, представитель Z);

Z - значение (содержание, смысл).

Группы знаков (Ч.С.Пирс):

- иконические знаки (иконы): форма ~ содержание;

- индексальные знаки (индексы): причинно-следственная связь между формой и содержанием;

- символические знаки (символы): связь между формой и содержанием произвольна, по соглашению.

По способу восприятия:

1) зрительные;

2) слуховые;

3) осязательные;

4) обонятельные;

5) вкусовые.

1. 3. Предмет изучения математической теории информации и теории кодирования.

Основное содержание "математической теории информации" - исследование методов кодирования:

а) для экономного формирования сообщений от различных источников;

б) и для надежной передачи сообщений по каналам связи;

в) с защитой от несанкционированного доступа.

В основе классической ТИК - измерение количества информации, содержащейся в сообщениях, на базе статистического описания источников сообщений и каналов связи.

2. 1. Основные отличия цифровых и аналоговых сигналов. Способы передачи сигнала по каналу связи. Понятие о несущей частоте.

Цифровой сигнал

- конечное множество состояний;
- изменение – в определенные моменты времени, кратные интервалу времени *T*.

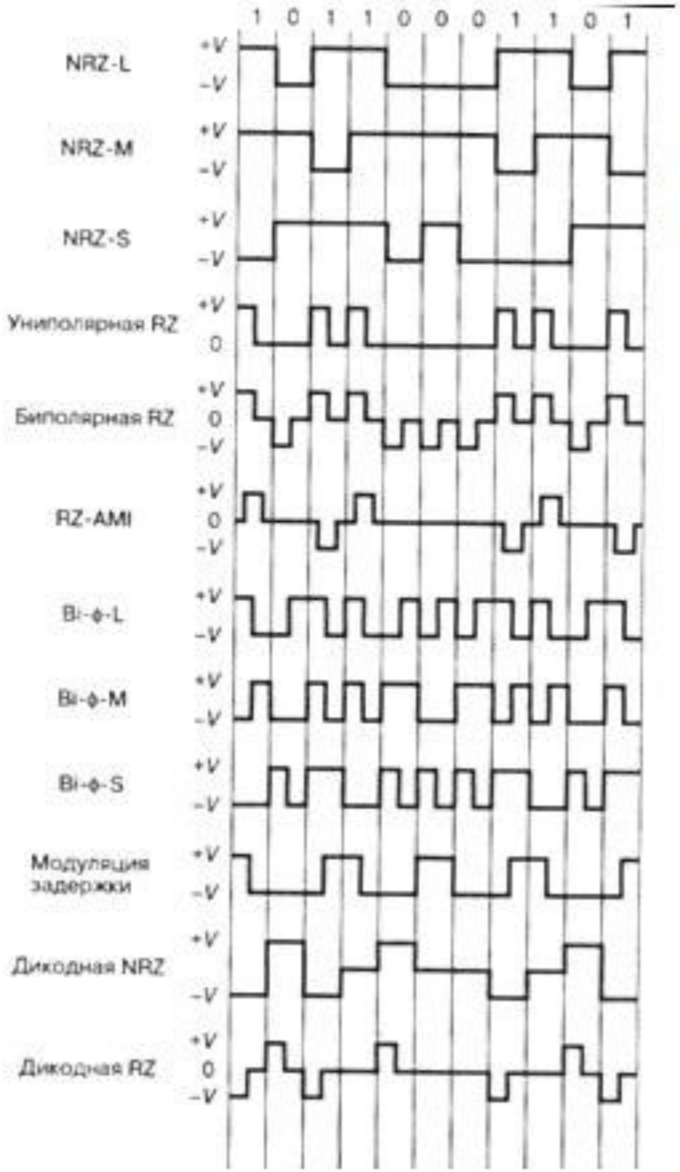
Аналоговый сигнал

- непрерывная функция времени с бесконечным множеством состояний;
- представление в виде набора простейших синусоидальных колебаний (гармоник) с различными частотами *f*.

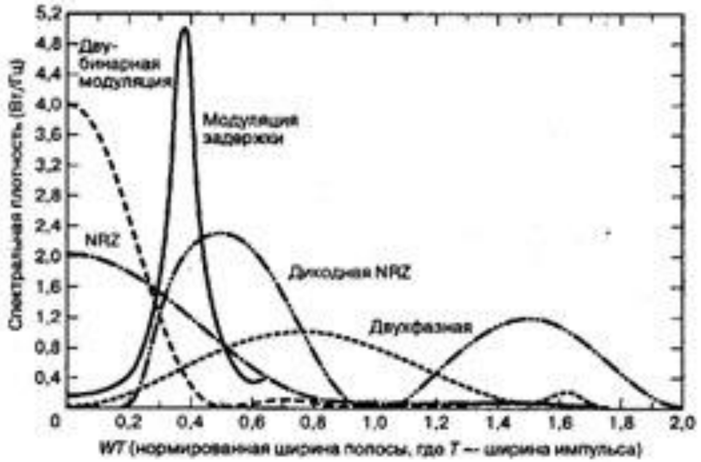
Способы передачи сигнала по каналу связи:

- в виде изменения какого-либо параметра периодического сигнала (частоты, амплитуды, фазы синусоиды) – в этом случае говорят об **аналоговом канале**, а периодический сигнал, параметры которого меняются, называется несущим сигналом или *несущей частотой*;
- в виде изменения знака потенциала последовательности прямоугольных импульсов – в этом случае имеет место **цифровой канал связи**.

Примеры способов цифровой модуляции



Сигналы импульсной модуляции делятся на 4 группы:
- без возврата к нулю (NRZ);
- с возвратом к нулю (RZ);
- фазовое кодирование;
- многоуровневое бинарное кодирование.



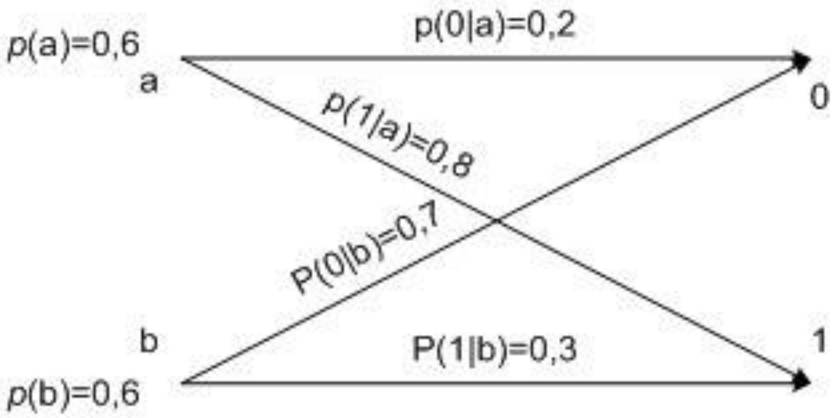
2. 2. Пропускная способность канала и единицы измерения. Скорость манипуляции и единицы измерения.

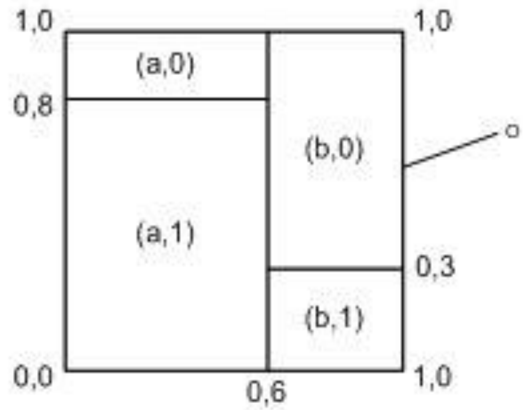
Пропускная способность (throughput) линии характеризует максимально возможную скорость передачи данных по линии связи, измеряемую в битах в секунду – бит/сек, bps (Кбит/с, Мбит/с).

Количество изменений информационного параметра несущего периодического сигнала в секунду называется *скоростью манипуляции* (*B*) и измеряется в *бодах*.

Вероятностная модель дискретного канала связи

Пример





$$p(a,0) = p(a)\cdot p(0\,|\,a) = 0{,}6\cdot 0{,}2 = 0{,}12$$

$$p(b,0) = p(b)\cdot p(0\,|\,b) = 0{,}4\cdot 0{,}7 = 0{,}28$$

$$p(a,1) = p(a)\cdot p(1\,|\,a) = 0{,}6\cdot 0{,}8 = 0{,}48$$

$$p(b,1) = p(b)\cdot p(1\,|\,b) = 0{,}4\cdot 0{,}3 = 0{,}12$$

Поскольку $p(b,0)>p(a,0)$, $p(a,1)>p(b,1)$, то оптимальный приемник определяется при помощи отображения: $m'_0=b$, $m'_1=a$.

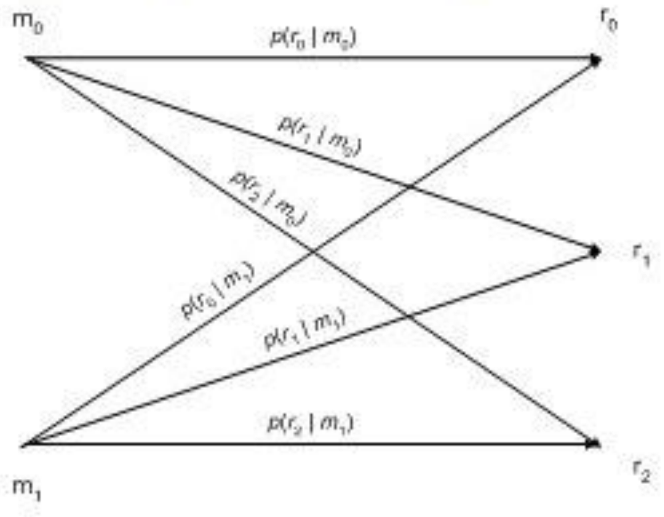
$p(\xi)=p(b,0)+p(a,1)=0{,}76$, и, следовательно, вероятность ошибки равна $1-p(\xi)=0{,}24$.

Вероятностная модель дискретного канала связи

Пусть канал имеет M возможных сообщений на входе $\{m_i\}$, $0 \le i \le M$-1 и N возможных сообщений на выходе $\{r_j\}$, $0 \le j \le N$-1.

Математическая модель канала определяется совокупностью MxN условных вероятностей $\{p(r_j \mid m_i)\}$, задающих вероятность появления каждого символа на выходе при поступлении любого сообщения на вход.

Диаграмма условных вероятностей



2. 3. Связь между пропускной способностью канала связи и шириной полосы пропускания (по Шеннону)

Связь между полосой пропускания и ее максимально возможной пропускной способностью, вне зависимости от принятого способа физического кодирования (К. Шеннон):

$$C = F\log_2(1 + P_c/P_ш),$$

где*C* – максимальная пропускная способность линии (бит/сек), *F* – ширина полосы пропускания линии (Гц), *P*с – мощность сигнала, *P*ш – мощность шума.

2. 4. Найти пропускную способность канала связи, обеспечивающего передачу М-позиционного сигнала со скоростью манипуляции В бод

$$C = F\log_2(1 + P_c/P_ш),$$

где*C* – максимальная пропускная способность линии (бит/сек), *F* – ширина полосы пропускания линии (Гц), *P*с – мощность сигнала, *P*ш – мощность шума.

2. 5. Модуляция. Назначение узкополосной (импульсной) и полосовой (аналоговой) модуляции.

Модуляция – это процесс, посредством которого символы сообщений преобразуются в сигналы, совместимые с требованиями, налагаемыми каналом передачи данных.

Способ физического кодирования дискретных данных на основе синусоидального несущего сигнала называется *аналоговой (полосовой) модуляцией*, а на основе последовательности прямоугольных импульсов – *цифровой (узкополосной, импульсной) модуляцией*.

2. 6. Требования к методам импульсной модуляции и способы их достижения.

Требования к способу модуляции:

- при одной и той же битовой скорости имеет наименьшую ширину спектра результирующего сигнала;

- обеспечивает синхронизацию между передатчиком и приемником (обычно для синхронизации используются фронты сигналов);

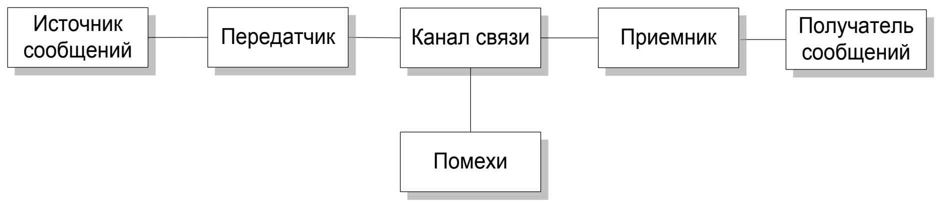
- обладает способностью распознавать ошибки.

2. 7. Виды аналоговой модуляции

Амплитудная, фазовая, частотная и смешанная

2. 8. Общая модель системы связи по К.Шеннону. Назначение блоков модели.

модель связи по Шеннону



Источник информации (ИИ) выдает ее в виде первичного сообщения, представленного последовательностью первичных сигналов. Для дальнейшей передачи эти сигналы преобразуются в сигналы такой физической природы, которые могут распространяться в заданном материальном носителе – формируется вторичное сообщение. Примерами преобразователей являются: мегафон или телефонный аппарат, преобразующие голосовые сигналы в электрические; радиопередатчик, преобразующие голосовые сигналы в радиоволны; телекамера, преобразующая изображение в последовательность электрических импульсов; модем, переводящий высокочастотные компьютерные сигналы в аналоговые низкочастотные и обратно, и пр.

При необходимости перед преобразованием или в процессе его может осуществляться кодирование первичного сообщения кодером (К). Кодирование (точнее, первичное кодирование) может осуществляться непосредственно источником информации, например, человеком при работе на передатчике с использованием азбуки Морзе.

Непосредственная передача осуществляется передатчиком вторичного сообщения (ПрдС). Он инициирует некоторый нестационарный процесс, обеспечивающий распространение сигналов в канале связи.

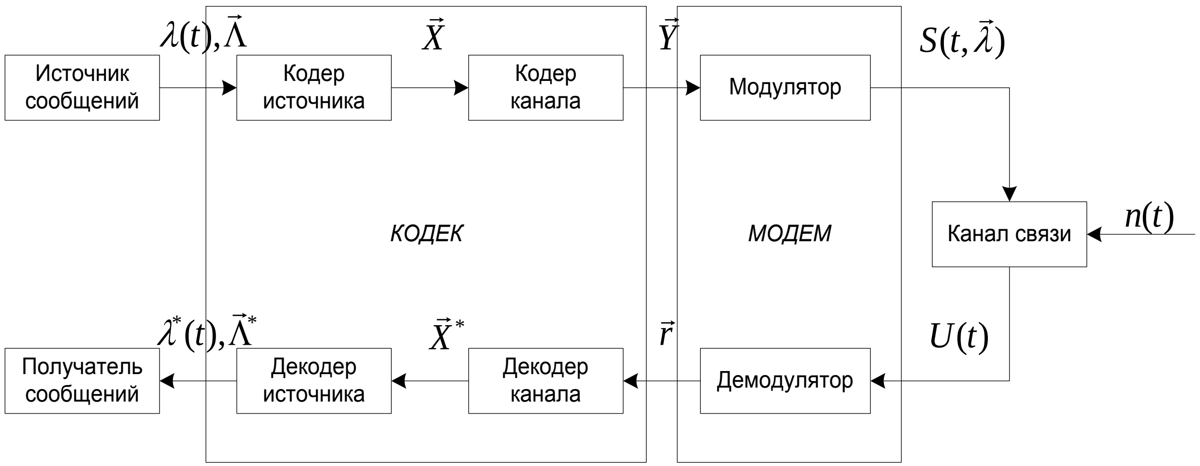
Канал связи – это материальная среда, а также физический или иной процесс, посредством которого осуществляется передача сообщения, т.е. распространение сигналов в пространстве с течением времени. Каналы связи в зависимости от характера сигналов, передаваемых по ним подразделяются на дискретные и аналоговые. Примером дискретного канала является компьютерная сеть; аналогового – телефонная линия и радиоканал.

Любой реальный канал связи подвержен внешним воздействиям, а также в нем могут происходить внутренние процессы, в результате которых искажаются передаваемые сигналы и, следовательно, связанная с ними информация. Такие воздействия называются шумами (помехами).

После прохождения вторичного сообщения по каналу связи оно попадает в приемное устройство, где одновременно преобразуется в форму, необходимую для дальнейшей интерпретации. Если перед передачей применялось кодирование, после приема вторичное сообщение направляется в декодер и лишь затем – к получателю (потребителю) информации.

2. 9. Детализированная модель системы связи. Назначение блоков модели.

Детализированная модель связи



Источник информации (ИИ) выдает ее в виде первичного сообщения, представленного последовательностью первичных сигналов. Для дальнейшей передачи эти сигналы преобразуются в сигналы такой физической природы, которые могут распространяться в заданном материальном носителе – формируется вторичное сообщение. Примерами преобразователей являются: мегафон или телефонный аппарат, преобразующие голосовые сигналы в электрические; радиопередатчик, преобразующие голосовые сигналы в радиоволны; телекамера, преобразующая изображение в последовательность электрических импульсов; модем, переводящий высокочастотные компьютерные сигналы в аналоговые низкочастотные и обратно, и пр.

Кодер источника служит для преобразования сообщений в кодовые символы с целью уменьшения избыточности источника сообщения, т.е. обеспечении минимума среднего числа символов на одно сообщение и представления в удобной форме (например, в виде двоичных чисел).

Кодер канала, предназначен для введения избыточности, позволяющей обнаруживать и исправлять ошибки в канальном декодере, с целью повышения достоверности передачи.

Декодер канала обеспечивает проверку избыточного (помехоустойчивого) кода и преобразование его в последовательность первичного электрического сигнала безызбыточного кода.

Декодер источника (ДИ) – это устройство для преобразования последовательности ПЭС безизбыточного кода в сообщение.

Каналом связи называется совокупность средств, обеспечивающих передачу сигнала от некоторой точки А системы до точки В. Точки А и В могут быть выбраны различным образом в зависимости от решаемой задачи построения модели, проектирования или анализа СЭС. В зависимости от вида входных и выходных символов канал связи может быть непрерывным, дискретным и полунепрерывным. В одной и той же схеме можно выделить как дискретный так и непрерывный канал, в зависимости от выбора рассматриваемых точек.

Модулятор – осуществляет преобразование первичного сигнала во вторичный сигнал, удобный для передачи в среде распространения в условиях действия помех.

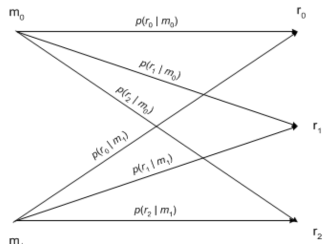
Демодулятор это устройство, в котором из принятого сигнала выделяется первичный электрический сигнал, который из-за действия помех может значительно отличаться от переданного.

2. 10. Способ задания математической модели дискретного канала связи. Используемые вероятности. Вид диаграммы условных вероятностей.

Пусть канал имеет M возможных сообщений на входе $\{m_i\}$, $0 \le i \le M-1$ и N возможных сообщений на выходе $\{r_j\}$, $0 \le j \le N-1$.

Математическая модель канала определяется совокупностью MxN условных вероятностей $\{p(r_j \mid m_i)\}$, задающих вероятность появления каждого символа на выходе при поступлении любого сообщения на вход.

Диаграмма условных вероятностей



3. 1. Этапы аналого-цифрового преобразования. Параметры АЦП. Как определить необходимую частоту дискретизации сигнала при АЦП?

Этапы аналого-цифрового преобразования:

- дискретизация сигнала по времени;

- квантование сигнала по уровню.

Параметры АЦП:

- интервал дискретизации;

- 0-уровень (уровень отсчета);

- диапазон квантования;

- размер шага квантования.

Чаще всего частота дискретизации берётся равной $\Delta t=1/2Fm$
где Fm - максимальная частота спектра преобразуемого сигнала.
Также выбор зависит от допустимого для данной системы уровня погрешностей, возникающих при восстановлении исходного сигнала по его отсчетам.

Мощность сигнала

Величина, выраженная в децибелах, численно равна десятичному логарифму безразмерного отношения физической величины к одноимённой физической величине, принимаемой за исходную, умноженному на десять:

$$A_{dB} = 10\lg \frac{A}{A_0}$$

Отношение сигнал/шум (ОСШ; англ. *signal-to-noise ratio*, сокр. *SNR*) — безразмерная величина, равная отношению мощности полезного сигнала к мощности шума.

$$\mathrm{SNR} = \frac{P_\mathrm{signal}}{P_\mathrm{noise}} = \left(\frac{A_\mathrm{signal}}{A_\mathrm{noise}}\right)^2$$

где *P* — средняя мощность, а *A* — среднеквадратичное значение амплитуды. Оба сигнала измеряются в полосе пропускания системы.

Обычно отношение сигнал/шум выражается в децибелах (дБ). Чем больше это отношение, тем меньше шум влияет на характеристики системы.

$$\mathrm{SNR(dB)} = 10\log_{10}\left(\frac{P_\mathrm{signal}}{P_\mathrm{noise}}\right) = 20\log_{10}\left(\frac{A_\mathrm{signal}}{A_\mathrm{noise}}\right)$$

3.2. Понятие диапазона квантования, интервалов квантования, пороговых уровней, уровней квантования. Постановка задачи квантования

Процесс преобразования непрерывного аналогового сигнала в последовательность его мгновенных значений (выборок) называется дискретизацией. Определение численного значения величины выборки (отсчета) называется квантованием. Для этого весь диапазон возможных изменений амплитуды преобразуемого сигнала делится на множество уровней квантования, количество которых определяется разрядностью используемого при этом двоичного числа. Чем больше число разрядов квантования, тем меньше расстояние между уровнями квантования (шаг квантования) и тем выше получается точность преобразования. Посередине между уровнями квантования расположены пороговые уровни. Когда исходный непрерывный сигнал проходит через какой-либо пороговый уровень, расположенный между соседними уровнями квантования, происходит скачкообразное изменение значения квантованного сигнала.

Задача квантования заключается в представлении текущих значений непрерывно изменяющегося сигнала значениями уровней из конечного множества. В результате чего непрерывно изменяющийся сигнал F(t) заменяется ступенчатой функцией Y(t), зависящей только от характера изменения квантованного сигнала;

3.4. Различия между равномерной и логарифмической шкалой квантования. Влияние параметра компандирования на характеристику квантователя

Линейное (однородное) квантование амплитуды

Отведём для записи одного значения амплитуды сигнала в памяти компьютера N бит. Значит, с помощью одного N -битного слова можно описать 2N разных положений. Пусть амплитуда оцифровываемого сигнала колеблется в пределах от -1 до 1 некоторых условных единиц. Представим этот диапазон изменения амплитуды - динамический диапазон сигнала - в виде 2N -1 равных промежутков, разделив его на 2N уровней - квантов. Теперь, для записи каждого отдельного значения амплитуды, его необходимо округлить до ближайшего уровня квантования. Этот процесс носит название квантования по амплитуде. Квантование по амплитуде – процесс замены реальных значений

амплитуды сигнала значениями, приближенными с некоторой точностью. Каждый из 2N возможных уровней называется уровнем квантования, а расстояние между двумя ближайшими уровнями квантования называется шагом квантования. Если амплитудная шкала разбита на уровни линейно, квантование называют линейным (однородным).
Точность округления зависит от выбранного количества (2N) уровней квантования, которое, в свою очередь, зависит от количества бит (N), отведенных для записи значения амплитуды.

Способ неоднородного квантования предусматривает разбиение амплитудной шкалы на уровни по логарифмическому закону. Такой способ квантования называют логарифмическим квантованием. При использовании логарифмической амплитудной шкалы, в области слабой амплитуды оказывается большее число уровней квантования, чем в области сильной амплитуды (при этом, общее число уровней квантования остается таким же, как и в случае однородного квантования).

3. 6. Причины возникновения эффекта "ложных контуров" при квантовании изображений и способы борьбы с ним (перечислить основные подходы)

При равномерном квантовании при существенном уменьшении числа бит на пиксел появляются большие области с одинаковым цветом. Вследствие этого могут возникнуть заметные ложные контуры.

- Если выходное изображение квантовать с большим числом уровней, чем входное, то можно получить равномерное размещение выходных уровней и благодаря этому уменьшить эффект появления ложных контуров.

- Добавление шума к изображению (псевдошумовое квантование) позволяет ослабить эффект ложных контуров. Для разрушения ложных контуров Робертс предложил перед равномерным квантованием к отсчетам яркости добавлять шум с равномерной плотностью распределения вероятностей. Добавленный шум переводит одни отсчеты изображения на уровень выше, а другие на уровень ниже. Тем самым разрушаются ложные контуры.

- Квантование с улучшенной передачей градаций яркости.

- Квантование с диффузией ошибки по Флойду-Стейнбергу. Ошибка между истинным цветом и цветом после квантования не исчезает, а распределяется на соседние пикселы по коэффициентам, сумма которых должна быть равна единице.

3. 7. Принципы и ключевые особенности ДИКМ. Математическое представление ДИКМ

Эффективность основана на том, что разность между близлежащими сигналами в большинстве случаев меньше по модулю, нежели амплитуды отсчётов. Это позволяет использовать для кодирования разностей кодовые слова с меньшим количеством битов, чем при кодировании собственно самих амплитуд.

Особенности:

- Кодирование не амплитуд, а разности.

- Наличие схемы предсказания, в которой алгоритм ДИКМ предсказывает не следующий отсчёт, а разность между следующим истинным отсчётом и предсказанным отсчётом.

- Обратная связь на стороне кодера.

Кодер:

$$d(n) = x(n) - x'(n)$$
$$d_q(n) = Q[d(n)]$$
$$x_q(n) = x'(n) + d_q(n)$$
$$x'(n+1) = F(x_q(n), x_q(n-1), \dots, d_q(n), d_q(n-1), \dots)$$

Декодер:

$$x_q(n) = x'(n) + d_q(n)$$
$$x'(n+1) = F(x_q(n), x_q(n-1), \dots, d_q(n), d_q(n-1), \dots)$$

3. 8. Параметры ДИКМ. Понятие об адаптивной ДИКМ

Параметры:
- Нулевой уровень отсчёта
- Диапазон квантования
- Размер шага квантования
    - адаптация частоты дискретизации сигнала;

    - адаптация коэффициентов предсказания;

- адаптация размера шага квантования:

$$\Delta_{n+1} = \Delta_n M(n)$$

Адапти́вная дифференциа́льная и́мпульсно-ко́довая модуля́ция (АДИКМ)— разновидность дифференциальной импульсно-кодовой модуляции, алгоритм которой подразумевает изменение шага квантования, что позволяет снизить требуемую полосу пропускания для заданного отношения сигнал/шум. Обычно адаптация основывается на адаптивном коэффициенте масштабирования[1].

3. 9. Отличие дельта-модуляции от ДИКМ. Виды искажений, типичные для кодера ДМ. В чем заключается сложность борьбы с этими видами искажений?

Дельта-модуляцию можно рассматривать как простейшую форму ДИКМ, в которой используется двухуровневый (однобитный) квантователь в сочетании с фиксированным предсказателем первого порядка. Простейшей формой квантования является компаратор, который обнаруживает и сообщает знак разности сигнала.

Два вида искажений:

- перегрузка по крутизне (шаг слишком мал);

- гранулярный шум (шаг слишком велик).

3. 10. Математическое представление ДМ первого порядка и ДМ второго порядка

Первого порядка

Модуляция:

$z_i = Y_i - y_i$;

$\Delta_{i+1} = -\mathrm{sign}(z_i)$;

Демодуляция:

$\nabla Y_{i+1} = c^*\Delta_{i+1}$;

$Y_{i+1} = Y_i + \nabla Y_{i+1}$; $c^*>0$

Второго порядка

Модуляция:

$z_i = Y_i - y_i;$

$\nabla z_i = z_i - z_{i-1};$

$F_i = z_i + 1.5\nabla z_i + (0.5\nabla z_i^2/c - 0.125c)\mathrm{sign}(\nabla z_i);$

$\Delta_{i+1} = -\mathrm{sign}(F_i);$

Демодуляция:

$\nabla 2Y_{i+1} = c^*\Delta_{i+1};$

$\nabla Y_{i+1} = \nabla Y_i + \nabla 2Y_{i+1};$

$Y_{i+1} = Y_i + \nabla Y_{i+1};$ $\quad$ $c^* \geq c;\ c>0$

3. 11. Понятие мгновенного и слогового компандирования. Цель использования компандирования в алгоритмах ДМ

При *мгновенном компандировании* абсолютная величина размера шага квантования определяется значениями нескольких знаков квантов модуляции.

При *инерционном (слоговом) компандировании* размер шага квантования на следующем шаге вычисляется с коэффициентом увеличения/уменьшения относительно размера шага квантования на предыдущем шаге.

Цель использования компандирования – сделать ОСШК одинаковой для всех амплитуд.

3. 12. Понятие о векторном квантовании. Что такое кодовая книга? Преимущества и недостатки векторного квантования по сравнения со скалярным

Вид квантования, при котором выполняется одновременное квантование блока отсчетов, называется векторным квантованием.

Каждая ячейка в многомерном пространстве, в которую может попасть исходный вектор ***X***, характеризуется центроидом, минимизирующем ошибку квантования – значением ***X***'. Обычно ***X***' выбирается из конечного множества значений – кодовой книги. Размер кодовой книги можно считать равным числу уровней скалярных квантователей.

Недостатки по сравнению со скалярным квантованием:

- необходимость формирования оптимальной кодовой книги и ее хранения/передачи;
- высокая трудоемкость.

Преимущества :

- теоретически более высокая эффективность, чем у скалярного квантователя.

4.1. Комбинаторный, вероятностный, марковский и бернуллиевский источники сообщений. Определение дискретного ансамбля

И. с. делятся на различные классы в зависимости от типа сообщения - случайного процесса x(t), вырабатываемого источником сообщений. Например, если x(t) - случайный процесс с независимыми одинаково распределенными значениями или стационарный, эргодический, марковский, гауссовский и т. д. процесс, то источник сообщений. наз. соответственно источником сообщений без памяти, стационарным, эргодическим, марковским, гауссовскими т. д.

Комбинаторный – вероятность появления символов не учитывается.

Вероятностный – вероятность появления символов учитывается.

Марковский – когда следующий символ зависит от предыдущего (источник с памятью).

Бернуллиевский – когда следующий символ не зависит от предыдущего (источник без памяти).

Дискретный ансамбль – таблица, в которой в первой строке записаны символы, а во второй – вероятности их появления. Сумма вероятностей должна равняться единице. Описывает источник сообщений.

4.1. Найти энтропию источника, статистические характеристики которого описываются сообщением "abcbcaabacabd". Найти избыточность источника

При разных вероятностях сообщения несут различное количество информации $I(x_i)$. При решении большинства практических задач необходимо знать среднее количество информации, приходящееся на один элемент сообщения. Это среднее количество информации при общем числе элементов сообщения источника n и числе символов алфавита m равно:

$$H(X) = \frac{I(x,n)}{n} = -\sum_{i=1}^{m} p(x_i) \cdot \log_2 p(x_i) \quad (4.2)$$

(бит/сообщение).

Величину $H(X)$ называют энтропией источника сообщений.

Избыточными в источнике являются сообщения, которые несут малое, иногда нулевое, количество информации. Наличие избыточности означает, что часть сообщений можно и не передавать по каналу связи, а восстановить на приеме по известным статистическим связям. Так и поступают при передаче телеграмм, исключая из текста союзы, предлоги, знаки препинания, поскольку они легко восстанавливаются по смыслу телеграммы на основании известных правил построения фраз.

Количественно избыточность оценивается коэффициентом избыточности:

$$\chi = \frac{H_{\max}(X) - H(X)}{H_{\max}(X)} = 1 - \frac{H(X)}{H_{\max}(X)}$$

где $H(X)$ – энтропия источника; $H_{\max}(X) = \log_2 m$ – максимальная энтропия источника с алфавитом из m сообщений.

4. 2. Требования к мере количества информации в сообщении. Известные меры к определению количества информации (комбинаторный подход, вероятностный подход, алгоритмический подход)

Требования к вводимой мере оценки количества информации:

1) Чем больше число возможных сообщений (возможных значений сигнала), тем больше априорная неопределенность и тем большее количество информации получает адресат, когда эта неопределенность снимается. Если же выбор сообщения заранее предопределен, то количество информации в этом сообщении равно нулю.

2) Вводимая мера должна обладать свойством аддитивности, в соответствии с которым неопределенность объединенного источника равна сумме неопределенностей исходных источников.

Комбинаторный подход к оценке количества информации (Р.Хартли, 1928г.).

Степень неопределенности опыта X с N различными исходами характеризуется числом

$H(X) = \log N.$

Не учитываются вероятности различных исходов.

Вероятностный подход к оценке количества информации (К.Шеннон, 1949г.).

Степень неопределенности конкретного состояния зависит не только от объема алфавита источника, но и от вероятности этого состояния.

Количество информации, содержащееся в одном элементарном дискретном сообщении x_k целесообразно определить как функцию вероятности появления этого сообщения $p(x_k)$ и характеризовать величиной

Величина $i(x_k)$ называется количеством собственной информации в сообщении $x_k \in X$.

Алгоритмический подход к оценке количества информации (А.Н.Колмогоров, 1965г.).

Энтропия $H(X, Y)$ ("колмогоровская сложность" объекта Y при заданном X) есть мнимая длина, записанная в виде последовательности нулей и единиц, программы, которая позволяет построить объект Y, имея в своем распоряжении объект X.

Колмогоровская сложность обычно невычислима.

4. 3. Количественные информационные оценки для дискретных источников с памятью. Понятие условной собственной информации, совместной и взаимной информации пары событий ансамбля XY

Пусть $\{XY, p(x_i,y_j)\}$ – два совместно заданных ансамбля $\{X, p(x_i)\}$ и $\{Y, p(y_j)\}$.

Зафиксируем некоторое сообщение y_j и рассмотрим условное распределение на *X.*

Апостериорная вероятность $p(x_i|y_j)$ - неопределенность, остающаяся о сообщении x_i после того, как было принято сообщение y_j.

Условная собственная информация:

$$i(x_i \mid y_j) = -\log p(x_i \mid y_j)$$

Совместная информация пары событий:

$$i(x_i, y_j) = i(x_i) + i(y_j) - \log \frac{p(x_i \mid y_j)}{p(x_i)} = i(y_j) + i(x_i) - \log \frac{p(y_j \mid x_i)}{p(y_j)}$$

Взаимная информация пары событий:

$$i(x_i; y_j) = \log \frac{p(x_i \mid y_j)}{p(x_i)} = \log \frac{p(y_j \mid x_i)}{p(y_j)}$$

4. 4. Количественные информационные оценки для дискретных источников с памятью. Понятие совместной энтропии, условной энтропии и средней взаимной информации ансамбля XY

Пусть $\{XY, p(x_i,y_j)\}$ – два совместно заданных ансамбля $\{X, p(x_i)\}$ и $\{Y, p(y_j)\}$.

Зафиксируем некоторое сообщение y_j и рассмотрим условное распределение на *X.*

Апостериорная вероятность $p(x_i|y_j)$ - неопределенность, остающаяся о сообщении x_i после того, как было принято сообщение y_j.

Энтропия (совместная энтропия) ансамбля *XY*:

$$H(X,Y) = -\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i, y_j)\log p(x_i, y_j)$$

$$H(X,Y) = -\underline{\sum_{j=1}^{M} p(y_j)\log p(y_j)} - \sum_{j=1}^{M} p(y_j)\underline{\sum_{i=1}^{N} p(x_i \mid y_j)\log p(x_i \mid y_j)}$$

$$H(X,Y) = H(Y) + \left(-\sum_{j=1}^{M} p(y_j) H(X \mid y_j)\right)$$

Математическое ожидание случайной величины $i(x_i;y_j)$ - *средняя взаимная информация* между источниками *X* и *Y*:

$$I(X;Y) = \sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i, y_j) i(x_i; y_j) = \sum_{i=1}^{N}\sum_{j=1}^{N} p(x_i, y_j)\log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

можно записать:

$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

5. 1. **Постановка задачи кодирования источника. Типы кодирования (понятие кодов фиксированной и переменной длины). Цель эффективного кодирования.**

При кодировании, в соответствии с определенным правилом (кодом) f последовательность u_i преобразуется в конечную последовательность (кодовое слово) $x_i = (x_1, x_2, ..., x_k)$, формируемую из букв алфавита $D=(d_1, ..., d_m)$ кодового словаря X. Если множество конечных последовательностей источника обозначить как U^*, а множество конечных кодовых слов – как X^*, то кодирование – это отображение

$$f: U^* \to X^*,$$

а код последовательности u_i или кодовое слово x_i – как $x_i = f(u_i)$.

Типы кодирования источника:

Коды фиксированной длины

- слова источника u_i различной длины n_i -> кодовые слова x_i одинаковой длины $k_i=k=\text{const}$;
- слова источника u_i одинаковой длины $n_i=n=\text{const}$ -> кодовые слова x_i одинаковой длины $k_i=k=\text{const}$;

Коды переменной длины

- слова источника u_i одинаковой длины $n_i=n=\text{const}$ -> кодовые слова x_i различной длины k_i;
- слова источника переменной длины u_i -> кодовые слова x_i переменной длины k_i.

Эффективное кодирование обеспечивает увеличение средней информационной нагрузки на кодовое слово (символ кодового словаря).

5.1. Неравенство Крафта. Можно ли построить префиксно-свободный код с кодовыми словами длиной 1, 2, 3, 4, 5, 6, 7, 8, 9 битов?

Неравенство Крафта

Теорема:

Для любого префиксного кода C, отображающего произвольный алфавит A на двоичный алфавит $\{0, 1\}$, длины кодовых слов должны удовлетворять неравенству:

$$\sum_{i=1}^{I} 2^{-l_i} \leq 1,$$

где $|A| = I$, а l_i — длины кодовых слов.

Как я понял, нужно проверить, существует ли такое целое положительное число I, для которого при подстановке наших значений вместо l_i данное неравенство выполняется.

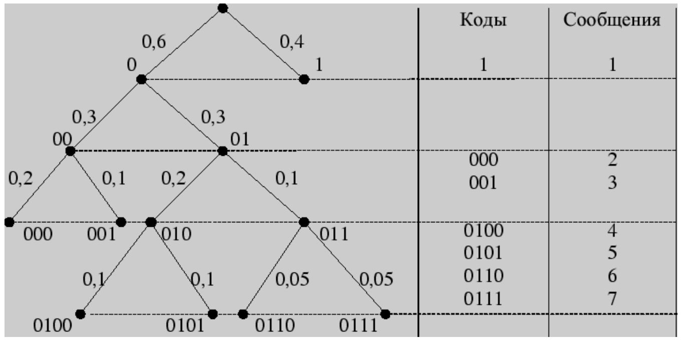
5.2. Построить дерево Фано и дерево Хаффмана для кодирования сообщения "abacdfdeeaaab"

При кодировании по Фано все сообщения записываются в таблицу по степени убывания вероятности и разбиваются на две группы примерно (насколько это возможно) равной вероятности. Соответственно этой процедуре из корня кодового дерева исходят два ребра, которым в качестве весов присваиваются полученные вероятности. Двум образовавшимся вершинам приписывают кодовые символы 0 и 1. Затем каждая из групп вероятностей вновь делится на две подгруппы примерно равной вероятности. В соответствии с этим из каждой вершины 0 и 1 исходят по два ребра с весами, равными вероятностям подгрупп, а вновь образованным вершинам приписывают символы 00 и 01, 10 и 11. В результате многократного повторения процедуры разделения вероятностей и образования вершин приходим к ситуации, когда в качестве веса, приписанного ребру бинарного дерева, выступает вероятность одного из данных сообщений. В этом случае вновь образованная вершина оказывается листом дерева, т.к. процесс деления вероятностей для нее завершен. Задача кодирования считается решенной, когда на всех ветвях кодового бинарного дерева образуются листья.

Пример. Закодировать по Фано сообщения, имеющие следующие вероятности:

сообщение	1	2	3	4	5	6	7
вероятность	0,4	0,2	0,1	0,1	0,1	0,05	0,05

Решение 1 (с использованием кодового дерева)



Методика кодирования Хаффмана

Кодируемые знаки располагают в порядке убывания их вероятностей (таблица 9.3). Далее на каждом этапе две последние позиции списка заменяются одной и ей приписывают вероятность, равную сумме вероятностей заменяемых позиций. После этого производится пересортировка списка по убыванию вероятностей, с сохранением информации о том, какие именно знаки объединялись на каждом этапе. Процесс продолжается до тех пор, пока не останется единственная позиция с вероятностью, равной 1.

После этого строится кодовое дерево. Корню дерева ставится в соответствие узел с вероятностью, равной 1. Далее каждому узлу приписываются два потомка с вероятностями, которые участвовали в формировании значения вероятности обрабатываемого узла. Так продолжают до достижения узлов, соответствующих вероятностям исходных знаков.

Процесс кодирования по кодовому дереву осуществляется следующим образом. Одной из ветвей, выходящей из каждого узла, например, с более высокой вероятностью, ставится в соответствие символ 1, а с меньшей – 0. Спуск от корня к нужному знаку дает код этого знака. Правило кодирования в случае равных вероятностей оговаривается особо.

Замечательным свойством кодов, построенных с применением методик Хаффмана, является их префиксность. Оно заключается в том, что ни одна комбинация кода не является началом другой, более длинной комбинации. Это позволяет при отсутствии ошибок осуществлять однозначное декодирование ряда следующих друг за другом кодовых комбинаций, между которыми отсутствуют разделительные символы.

Таблица 9.3

Знаки	p_i	Вспомогательные столбцы						
		1	2	3	4	5	6	7
z_1	0,22	0,22	0,22	0,26	0,32	0,42	**0,58**	**0,1**
z_2	0,2	0,2	0,2	0,22	0,26	**0,32**	**0,42**	
z_3	0,16	0,16	0,16	0,2	**0,22**	**0,26**		
z_4	0,16	0,16	0,16	**0,16**	**0,2**			
z_5	0,1	0,1	**0,16**	**0,16**				
z_6	0,1	**0,1**	**0,1**					
z_7	**0,04**	**0,06**						
z_8	**0,02**							

Таблица 9.4

Знаки	Коды
z_1	01
z_2	00
z_3	111
z_4	110
z_5	100
z_6	1011
z_7	10101
z_8	10100

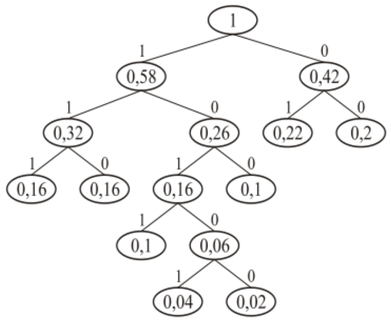


Рис. 9.1. Кодовое дерево

5. 2. Теорема кодирования источника кодами фиксированной длины и ее смысл. Обобщенная теорема кодирования источника и ее смысл

Теорема:

При кодировании источника m-кратными кодами фиксированной длины выбором значения J(увеличения значения) сегда можно добиться того, чтобы среднее количество элементарных символов затратив на передачу одной буквы сообщения стало сколь угодно близким к logL/lgm. L-количество символов в исходном сообщении, m-какой код используется.

Обобщенная теорема кодирования источника:

Дискретный источник без памяти U имеет алфавит из L букв $A=(a_1,...,a_L)$ с вероятностями $p(a_1)$, ..., $p(a_L)$.

Обозначим через m количество возможных символов в кодовом алфавите, а k_i – число букв в кодовом слове, соответствующем a_i. Тогда среднее число букв в кодовом слове на одну букву источника будет определяться как

$$\bar{k} = \sum_{i=1}^{L} p(a_i) k_i$$

При заданном конечном ансамбле источника U с энтропией $H(U)$ и кодовом алфавите из m символов существует возможность создать код, который отвечает префиксному требованию условию и имеет среднюю длину, удовлетворяющую условию

$$\frac{H(U)}{\log m} \le \bar{k} < \frac{H(U)}{\log m} + 1$$

5. 3. Базовые стратегии компрессии данных.

1. Статистическое кодирование:

- *блочное*, когда статистика хранится с данными

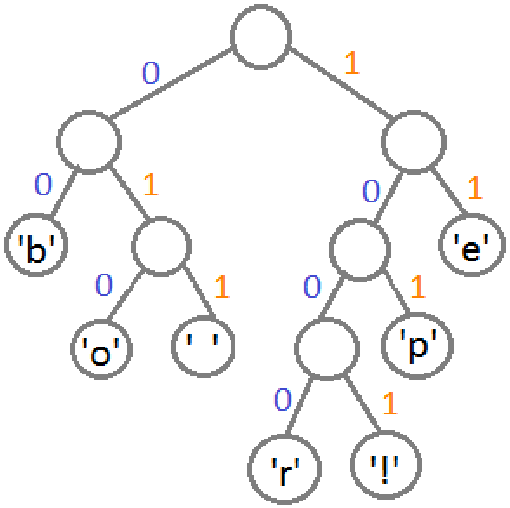
- *поточное*, когда статистика постоянно обновляется

2. Трансформация потока (словарные методы)

3. Трансформация блока

5.3. Дано дерево Хаффмана и кодовая последовательность битов. Декодировать сообщение

Коды для символов:



Символ	Код
'b'	00
'e'	11
'p'	101
' '	011
'o'	010
'r'	1000
'!'	1001

Чтобы расшифровать закодированную строку, нам надо, соответственно, просто идти по дереву, сворачивая в соответствующую каждому биту сторону до тех пор, пока мы не достигнем листа. Например, если есть строка «101 11 101 11» и наше дерево, то мы получим строку «pepe».

Важно иметь в виду, что каждый код не является префиксом для кода другого символа. В нашем примере, если 00 — это код для 'b', то 000 не может оказаться чьим-либо кодом, потому что иначе мы получим конфликт. Мы никогда не достигли бы этого символа в дереве, так как останавливались бы ещё на 'b'

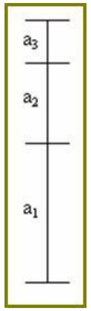
5.4. Задан ансамбль источника символов. Дано число с плавающей запятой (в десятичной системе счисления). Декодировать сообщение с использованием арифметического декодера

Как и в алгоритме Хаффмана, все начинается с таблицы элементов и соответствующих вероятностей. Допустим, входной алфавит состоит всего из трех элементов: $\boldsymbol{a_1}$, $\boldsymbol{a_2}$ и $\boldsymbol{a_3}$, и при этом

$$\mathbf{P}(a_1) = \mathbf{1/2}$$
$$\mathbf{P}(a_2) = \mathbf{1/3}$$
$$\mathbf{P}(a_3) = \mathbf{1/6}$$

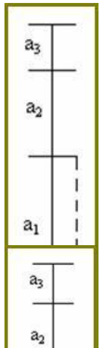
Предположим также, что нам надо закодировать последовательность $\boldsymbol{a_1}$, $\boldsymbol{a_1}$, $\boldsymbol{a_2}$, $\boldsymbol{a_3}$.

Разобьем полуинтервал [0, 1) на три части в соответствии с вероятностями элементов:

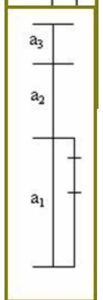


Элементу $\boldsymbol{a_1}$ будет соответствовать диапазон [0, 1/2); элементу $\boldsymbol{a_2}$ диапазон [1/2, 1/2 + 1/3), а элементу $\boldsymbol{a_3}$ - диапазон [1/2 +1/3 , 1).

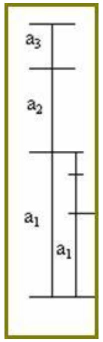
Первый элемент сжимаемого файла равен $\boldsymbol{a_1}$. Выберем полуинтервал, соответствующий ему [0, 1/2):



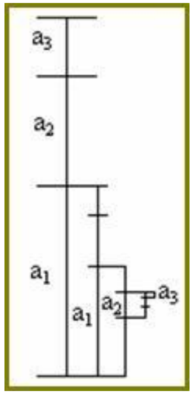
Разобьем его таким же образом на три части:



Поскольку второй элемент сжимаемой последовательности тоже равен ***a*₁**, снова выбираем самый нижний диапазон (точнее, теперь уже поддиапазон [0, 1/4)):



Аналогичным образом, продолжая процесс для оставшихся элементов входного файла ***a*₂** и ***a*₃**, получаем такой рисунок:



Последним полученным диапазоном будет [5/24, 1/4). Теперь -внимание! ***Любое*** число из этого диапазона (например, 0,22) однозначно описывает ***весь*** входной файл. Зная вероятности появления элементов, декодер может правильно разбить диапазон [0, 1) на три части и определить, что число 0,22 находится в поддиапазоне [0, 1/2). В свою очередь, это означает, что первый элемент декодированного файла равен $\boldsymbol{a_1}$. Затем декодер разобьет диапазон [0, 1/2) на три части и определит, что число 0,22 находится в поддиапазоне [0, 1/4), следовательно, следующий элемент выходного файла — снова $\boldsymbol{a_1}$ и т. д.

Поскольку декодер не может точно **определить,** когда следует закончить работу (процесс при желании можно продолжать бесконечно), придется либо явно прописать во входном файле количество элементов, подлежащих декодированию, либо ввести специальный символ «конец файла», встретив который, распаковщик остановится.

Хотя для описания сжимаемого файла годится любое число конечного диапазона (0.22, 0.222, 0.222222222222,...), выбирать надо, разумеется, то, которое содержит меньше цифр, поскольку каждая лишняя цифра увеличивает размер выходного файла.

5. 4. Коды неравномерной длины: понятие однозначности декодирования (примеры кодов с однозначным и неоднозначным декодированием); понятие мгновенного кода и его преимущества (примеры мгновенных и "немгновенных" кодов)

Однозначно декодируемый код — код, в котором любое слово составленное из кодовых слов можно декодировать только единственным способом.

Свойство мгновенного декодирования – свойство отсутствия префикса

5.4. Коды неравномерной длины: понятие однозначности декодирования (примеры кодов с однозначным и неоднозначным декодированием); понятие мгновенного кода и его преимущества (примеры мгновенных и "немгновенных" кодов)

Первым необходимым свойством кода является *однозначность декодирования*: принятое сообщение должно иметь единственно возможное толкование. Рассмотрим код, в котором алфавит источника S состоит из четырех символов, и они следующим образом кодируются в двоичном алфавите: $s_1=0$, $s_2=01$, $s_3=11$, $s_4=00$.

Принятое сообщение 0011 может означать одно из двух:

$$0011=\begin{cases} s_4,\ s_3 \\ s_1,\ s_1,\ s_3. \end{cases}$$

Таким образом, код не является однозначно декодируемым. Хотя однозначность декодирования не всегда абсолютно необходима, она обычно весьма желательна.

Следующий код обладает однозначным декодированием, но не является мгновенным, поскольку нельзя решить, когда кончается кодовое слово, не иследуя последующие биты: $s_1=0$, $s_2=01$, $s_3=011$, $s_4=111$. Это предыдущий код с перевернутыми кодовыми словами. Причина неприятностей состоит в том, что некоторые кодовые слова являются *префиксами* (иначе говоря, совпадают с начальными частями) других кодовых слов.

Рассмотрим последовательность $0111\underbrace{\ldots}_{s_4}\underbrace{111}_{s_4}$. Ее *можно декодировать только* начиная с конца и сопоставлять каждым трем единицам символ s_4 до тех пор, пока не будет достигнуто первое кодовое слово. Только после этого ему можно сопоставить один из символов: s_1, s_2 или s_3. Таким образом, приемник не может решить, получил ли он уже кодовое слово или ему нужно ждать дополнительных битов для завершения кодового слова. В приведенном частном случае простейший способ декодирования сообщения состоит в том, чтобы начинать с конца принятой последовательности. Этот способ требует значительной памяти, а также обусловливает задержку при декодировании.

Для получения мгновенного кода необходимо и достаточно, чтобы никакое кодовое слово, соответствующее s_i, не было префиксом никакого другого кодового слова, соответствующего s_j. Дерево решений, соответствующее немгновенному коду, имеет такую структуру, при которой в момент принятия окончательного решения о том, что полное кодовое слово уже пришло некоторое время тому назад, происходит выдача соответствующего символа источника и переход в некоторое не обязательно начальное, как для мгновенных кодов, состояние. Как показывает приведенный пример, принятие решения иногда требует, вообще говоря, неограниченного объема памяти.

В этом примере декодирование является *мгновенным*, поскольку при получении всего кодового слова приемник сразу же об этом узнает и ему не нужно исследовать последующие биты, чтобы решить, какой символ источника принят. Никакое кодовое слово этого кода не является префиксом никакого кодового слова.

Предположим, что требуется построить код для источника с алфавитом S, состоящим из пяти символов s_i, а кодовый алфавит должен быть двоичным. Для получения мгновенного кода (с запятой) можно символам источника сопоставить следующие слова: $s_1=0$, $s_2=10$, $s_3=110$, $s_4=1110$, $s_5=1111$ (рис. 4.4.1).

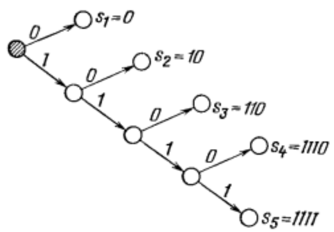


Рис. 4.4.1. Дерево декодирования

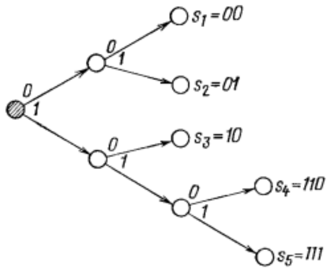


Рис. 4.4.2. Дерево декодирования

5. 5. Особенности оптимального кода, построенного по статическому алгоритму Хаффмана

Статический метод Хаффмана

Лемма. Пусть вероятности букв (символов) алфавита источника упорядочены по убыванию

$$p(a_1) \geq p(a_2) \geq \ldots \geq p(a_L).$$

Тогда существует оптимальный код, длина слов которого не убывают, а две наименее вероятные буквы имеют коды одинаковой длины:

$$|f(a_1)| \leq |f(a_2)| \leq \ldots \leq |f(a_{L-1})| = |f(a_L)|.$$

5.6. **Структура кодера и декодера адаптивного кодирования. Преимущества и проблемы адаптивного кодирования (на примере адаптивного метода Хаффмана)**

Адаптивный метод Хаффмана

КОДЕР	ДЕКОДЕР
ИнициализироватьМодель();	ИнициализироватьМодель();
Пока не конец Сообщения	Пока не конец БитовогоПотока
Символ = ВзятьСледующийСимвол();	Символ = РаскодироватьСледующийСимвол();
Закодировать(Символ);	ВыдатьСимвол(Символ);
ОбновитьМодельСимволом(Символ);	ОбновитьМодельСимволом(Символ);
Конец Пока	Конец Пока

Проблемы адаптивного кодирования:

- инициализация модели (специальные символы EOF и ESCAPE);

- переполнение.

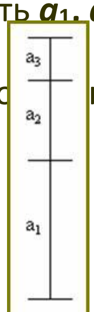
5.7. Принципы арифметического кодирования. Преимущество арифметического кодирования перед кодированием по Хаффману

Как и в алгоритме Хаффмана, все начинается с таблицы элементов и соответствующих вероятностей. Допустим, входной алфавит состоит всего из трех элементов: a_1, a_2 и a_3, и при этом

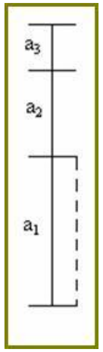
$$P(a_1) = 1/2$$
$$P(a_2) = 1/3$$
$$P(a_3) = 1/6$$

Предположим также, что нам надо закодировать последовательность a_1, a_1, a_2, a_3 .

Разобьем полуинтервал [0, 1) на три части в соответствии с вероятностями элементов:

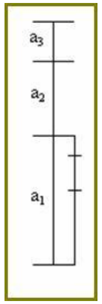


Элементу a_1 будет соответствовать диапазон [0, 1/2); элементу a_2 диапазон [1/2, 1/2 + 1/3), а элементу a_3 - диапазон [1/2 +1/3 , 1).

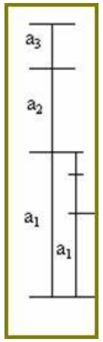


Первый элемент сжимаемого файла равен a_1. Выберем полуинтервал, соответствующий ему [0, 1/2):

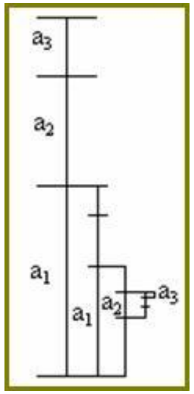
Разобьем его таким же образом на три части:



Поскольку второй элемент сжимаемой последовательности тоже равен a_1, снова выбираем самый нижний диапазон (точнее, теперь уже поддиапазон [0, 1/4)):



Аналогичным образом, продолжая процесс для оставшихся элементов входного файла a_2 и a_3, получаем такой рисунок:



Последним полученным диапазоном будет [5/24, 1/4). Теперь -внимание! ***Любое*** число из этого диапазона (например, 0,22) однозначно описывает ***весь*** входной файл. Зная вероятности появления элементов, декодер может правильно разбить диапазон [0, 1) на три части и определить, что число 0,22 находится в поддиапазоне [0, 1/2). В свою очередь, это означает, что первый элемент декодированного файла равен a_1. Затем декодер разобьет диапазон [0, 1/2) на три части и определит, что число 0,22 находится в поддиапазоне [0, 1/4), следовательно, следующий элемент выходного файла — снова a_1 и т. д.

Поскольку декодер не может точно **определить,** когда следует закончить работу (процесс при желании можно продолжать бесконечно), придется либо явно прописать во входном файле количество элементов, подлежащих декодированию, либо ввести специальный символ «конец файла», встретив который, распаковщик остановится.

Хотя для описания сжимаемого файла годится любое число конечного диапазона (0.22, 0.222, 0.222222222222,...), выбирать надо, разумеется, то, которое содержит меньше цифр, поскольку каждая лишняя цифра увеличивает размер выходного файла.

Формирование одного (длинного) кода для всего потока символов.

Присвоение кода всему передаваемому сообщению, а не отдельным его символам

5.7. Закодировать по методу стопки книг сообщение ABBCBDAAABA. Привести последовательность кодовых слов, выдаваемых на выход кодера и оценить размер закодированного сообщения при условии, что все символы алфавита источника имеются в приведенном сообщении

Метод также известен под названием MTF (Move To Front). Суть его легко понять, если представить процесс перемещения книг в стопке, из которой время от времени достают нужную книгу и кладут сверху. Таким образом, через некоторое время наиболее часто используемые книги оказываются ближе к верхушке стопки.

Для примера произведем преобразование над последовательностью «рдакрааaaбб». Предположим, что мы имеем дело с алфавитом, содержащим только эти 5 символов и в упорядоченном списке символов они расположены в следующем порядке: М = {"а", "б", "д", "к","р"}.

Первый из символов последовательности, "р", находится в списке под номером 4. Это число мы и записываем в выходной блок. Затем мы изменяем список, перенося этот символ в вершину списка, при этом сдвигая все остальные элементы, находившиеся до этого выше.

Следующий символ, "д", после этого сдвига оказывается в списке под номером 3. И т. д. (рис. 5.11).

Таблица Символ...Список упорядоченных символов.

р	абдкр
д	рабдк
а	драбк
к	адрбк
Р	кадрб
а	ркадб
а	аркдб
а	аркдб
а	аркдб
б	аркдб
б	баркд

преобразование строки "рдакраааабб"

Таким образом, в результате преобразования по методу перемещения стопки книг мы получили последовательность "43243200040".

5. 8. Понятие унарного кода. Понятие монотонного кода. Назначение унарного и монотонного кода. Принципы кодирования чисел с разделением мантисс и экспонент.

Унарный код — двоичный префиксный **код** переменной длины для представления натуральных чисел.

Монотонный код – не меняющийся

Применяется в кодах Элайеса??????????

Основная идея - отдельно описывать порядок значения элемента (экспоненту) и отдельно – значащие цифры значения (мантиссу).

Цель – закодировать число произвольной величины, когда верхняя граница чисел потока источника (которую необходимо знать, чтобы выбрать фиксированное количество битов на число) заранее неизвестна.

5. 9. Принципы методов словарного сжатия. Проблемы практической реализации словарных алгоритмов

Словарное сжатие

Входной алфавит $\boldsymbol{A}=\{0,1,\ldots,|\boldsymbol{A}|-1\}$.
x – слово, x^j_i - сегмент $x_i x_{i+1} \ldots x_j$ слова.
Разобьем слово x последовательно на несовпадающие подслова (парсинг):

$$x = x^{n_1}_{n_0+1} x^{n_2}_{n_1+1} \ldots x^{n_{p+1}}_{n_p+1}, \quad n_0 = 0, n_{p+1} = n = |x|$$

таким образом, что все слова, за исключением, возможно, последнего, различны и для каждого $j=1,2,\ldots,p+1$ при $n_j - n_{j-1} > 1$ существует i, $i<j$ такое, что $x^{n_i}_{n_{i-1}+1} = x^{n_j-1}_{n_{j-1}+1}$

Конкатенация кодов всех слов образует слово $f(x)$, для длины которого имеем

$$|f(x)| \le \sum_{j=1}^{p} [\log_j |A|]$$

С учетом того, что максимальное число $C(x)$ слов, на которые разбивается x, может лишь превзойти p ($C(x) \ge p$), имеем

$$|f(x)| \le C(x) \log C(x)(1 + o(1))$$

5. 9. Принципы методов словарного сжатия. Проблемы практической реализации словарных алгоритмов

Наиболее распространенный метод. Используется словарь, состоящий из последовательностей данных или слов. При сжатии эти слова заменяются на их коды из словаря.

Метод сжатия с использованием словаря — разбиение данных на слова и замена их на индексы в словаре. Этот метод является наиболее распространенным подходом для сжатия данных в настоящее время. Является естественным обобщением RLE.

В наиболее распространенном варианте реализации словарь постепенно пополняется словами из исходного блока данных в процессе сжатия.

Основным параметром любого словарного метода является размер словаря. Чем больше словарь, тем больше эффективность. Однако для неоднородных данных чрезмерно большой размер может быть вреден, так как при резком изменении типа данных словарь будет заполнен неактуальными словами. Для эффективной работы данных методов при сжатии требуется дополнительная память. Приблизительно на порядок больше, чем нужно для исходных данных словаря. Существенным преимуществом словарных методов является простая и быстрая процедура распаковки. Дополнительная память при этом не требуется. Такая особенность крайне важна, если необходим оперативный доступ к данным.

5.8. Закодировать число 39 гамма-кодом (дельта-кодом) Элайеса (для представления унарного кода используется код вида 0...0x)

Алгоритм кодирования числа N:

1. Сосчитать L — количество значащих битов в двоичном представлении числа N.
2. Сосчитать M — количество значащих битов в двоичном представлении числа L.
3. Записать $M-1$ нулей и одну единицу.
4. Дописать L_2 = $M-1$ младших битов двоичного представления числа L без старшей единицы.
5. Дописать N_2 = $L-1$ младших битов двоичного представления числа N без старшей единицы.

Пример кодирования числа 10:

1. В двоичном представлении числа $N = 10 = 1010_2$ 4 значащих бита ($L = 4$).
2. В двоичном представлении числа $L = 4 = 100_2$ 3 значащих бита ($M = 3$).
3. Записываем $M - 1 = 2$ нуля и одну единицу → `001`.
4. Дописывем биты числа L без старшей единицы → `00`.
5. Дописывем биты числа N без старшей единицы → `010`.
6. Результат — `00100010`.

5.9. Декодировать бинарный гамма (дельта) код Элайеса (для представления унарного кода используется код вида 0...0x)

Алгоритм декодирования числа из дельта-кода Элиаса:

1. Сосчитать M — количество нулей во входном потоке до первой единицы.
2. За единицей следуют M младших битов числа L, прочитать их и добавить к результату значение 2^M. Если биты L во входном потоке записаны от старших к младшим, то первую единицу после ведущей серии нулей можно читать как часть двоичного представления числа L, в этом случае добавлять 2^M отдельным шагом нет необходимости.
3. Следом идут $L-1$ младших битов числа N, прочитать их и добавить к результату значение 2^{L-1}.

Пример декодирования последовательности битов `001010001`:

1. Прочитать из потока `001` и определить, что в начале 2 ведущих нуля ($M = 2$).
2. Прочитать из потока следующие $M = 2$ бита → `01`; это даёт $L = 2^M + 01_2 = 4 + 1 = 5$.
3. Прочитать из потока следующие $L - 1 = 4$ бита → `0001`; это даёт $N = 2^{L-1} + 0001_2 = 16 + 1 = 17$.

5.10. Закодировать строку "шемшенашемнашем" по алгоритму LZ77 (LZSS, LZ78). Результат представить в виде таблицы. Длина словаря – 30 символов (для LZ78 – 30 слов). Длина lookahead-буфера – 8 символов. Для кодирования каждого поля выбрать минимальное количество битов (для ASCII-символа – 8 битов). Оценить размер сжатой последовательности.

Здесь берётся смещение в словаре символа от начала (началом служит ноль).

Пример. Закодировать по алгоритму LZSS строку "КРАСНАЯ КРАСКА".

СЛОВАРЬ(8)	БУФЕР(5)	КОД	ДЛИНА КОДА
"........"	"КРАСН"	0´К´	9
".......К"	"РАСНА"	0´Р´	9
"......КР"	"АСНАЯ"	0´А´	9
".....КРА"	"СНАЯ "	0´С´	9
"....КРАС"	"НАЯ К"	0´Н´	9
"...КРАСН"	"АЯ КР"	1<5,1>	7
"..КРАСНА"	"Я КРА"	0´Я´	9
".КРАСНАЯ"	" КРАС"	0´ ´	9
"КРАСНАЯ "	"КРАСК"	1<0,4>	7
"НАЯ КРАС"	"КА..."	1<4,1>	7
"АЯ КРАСК"	"А...."	1<0,1>	7

Здесь длина полученного кода равна 7 * 9 + 4 * 7 = 91 бит.

Здесь берётся ещё и следующий символ. Здесь тоже берётся смещение в словаре символа от начала (началом служит ноль).

Пример. Закодировать по алгоритму LZ77 строку "КРАСНАЯ КРАСКА".

СЛОВАРЬ(8)	БУФЕР(5)	КОД
"........"	"КРАСН"	<0,0,´К´>
".......К"	"РАСНА"	<0,0,´Р´>
"......КР"	"АСНАЯ"	<0,0,´А´>
".....КРА"	"СНАЯ "	<0,0,´С´>
"....КРАС"	"НАЯ К"	<0,0,´Н´>
"...КРАСН"	"АЯ КР"	<5,1,´Я´>
".КРАСНАЯ"	" КРАС"	<0,0,´ ´>
"КРАСНАЯ "	"КРАСК"	<0,4,´К´>
"АЯ КРАСК"	"А...."	<0,0,´А´>

В последней строчке, буква "А" берется не из словаря, т. к. она последняя.

Здесь берутся именно указатели на позицию словаря, в которой встретился этот символ и записываем следующий символ.

Пример. Закодировать по алгоритму LZ78 строку "КРАСНАЯ КРАСКА", используя словарь длиной 16 фраз.

ВХОДНАЯ ФРАЗА (В СЛОВАРЬ)	КОД	ПОЗИЦИЯ СЛОВАРЯ
""		0
"К"	<0,'К'>	1
"Р"	<0,'Р'>	2
"А"	<0,'А'>	3
"С"	<0,'С'>	4
"Н"	<0,'Н'>	5
"АЯ"	<3,'Я'>	6
" "	<0,' '>	7
"КР"	<1,'Р'>	8
"АС"	<3,'С'>	9
"КА"	<1,'А'>	10

6. 1. Теорема Шеннона для дискретного канала с шумом

Если производительность источника сообщений $H'(U)$ меньше пропускной способности канала C, т.е. $H'(U)<C$, то существует такая система кодирования, которая обеспечивает возможность передачи сообщений источника со сколь угодно малой вероятностью ошибки (или со сколь угодно малой ненадежностью).

Если $H'(U)>C$, то можно закодировать сообщение таким образом, что потери информации в единицу времени не будут превышать величину $H'(U)-C+\varepsilon$, где ε - сколь угодно мало.

Не существует способа кодирования, обеспечивающего потери в канале, меньшие, чем $H'(U)-C$.

6. 2. Классификация помехоустойчивых кодов. Расстояние Хэмминга. Кратность ошибки. Требования к минимальному расстоянию между кодовыми словами для обнаружения и исправления ошибок

Избыточные блочные коды (длины $n = k+r$):

- разделимые (систематические) - в каждой кодовой комбинации можно отделить информационные (k) и проверочные (r) разряды;

- неразделимые (несистематические) - все разряды равноправные и в кодовой комбинации нельзя отделить информационные и проверочные разряды.

Расстояние между двумя векторами кодового пространства по Хэммингу равно весу разности векторов. Минимальное расстояние между любыми двумя векторами кодового пространства называется кодовым расстоянием набора кодовых векторов (d_{min}). Корректирующий код обозначается либо как (n,k), либо как (n,k,$d_{\min}$).

Для *обнаружения* всех ошибок кратности, не превышающей q_{max}, кодовое расстояние должно быть не менее

$d_{min} = q_{max} + 1$.

Для обеспечения возможности *исправления* ошибок кратности не более q_{max}, кодовое расстояние должно быть не менее

$d_{min} = 2q_{max} + 1$

6. 3. Порождающая и проверочная матрицы систематического блочного кода. Принципы построения и связь между ними. Понятие синдрома ошибки

Пусть $x_1, x_2, \ldots, x_k$ означают слово из k информационных битов на входе кодера, кодируемое в кодовое слово C размерности n битов:

вход кодера: $X=[x1, x2, \ldots, xk]$

выход кодера: $C=[c1, c2, \ldots, cn]$

Пусть задана специальная порождающая матрица $G_{n,k}$, задающая блочный код (n,k).

Строки матрицы $G_{n,k}$ должны быть линейно независимы.

$$G_{n,k} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

Тогда разрешенная кодовая комбинация C, соответствующая кодируемому слову X:

$$C=x_1g_1 + x_2g_2 + \ldots + x_kg_k.$$

Систематическая (каноническая) форма порождающей матрицы G размером kxn :

$$G_{n,k} = [I_k \vdots P] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{11} & p_{12} & \cdots & p_{1r} \\ 0 & 1 & 0 & \cdots & 0 & p_{21} & p_{22} & \cdots & p_{2r} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & p_{k1} & p_{k2} & \cdots & p_{kr} \end{bmatrix}$$

Порождающая матрица систематического кода создает линейный блочный код, в котором первые k битов любого кодового слова идентичны информационным битам, а остальные r=n-k битов любого кодового слова являются линейными комбинациями k информационных битов.

Проверочная матрица $H_{n,k}$ имеет $r x n$ элементов, причем справедливо:

$$C \text{ x } H^T = 0.$$

Это выражение используется для проверки полученной кодовой комбинации. Если равенство нулю не выполняется, то получаем матрицу-строку $||c_1, c_2, ..., c_r||$, называемую синдромом ошибки.

6. 4. Код Хэмминга. Корректирующая и обнаруживающая способности. Правила выбора соотношения между длиной кодового слова и числом информационных битов. Формирование порождающей и проверочной матриц кода Хэмминга. Толкование синдрома ошибки

Код Хэмминга (для $d_{min}=3$)

Число разрешенных кодовых комбинаций для n-разрядных ($n=k+r$) кодовых слов:

$$2^n/(n+1)$$

Число информационных разрядов k и размер кодового слова n

$$k = \log_2(2^n/(n+1)] = n - \log_2(n+1)$$

Целочисленные решения (n,k): (3,1); (7,4); (15,11)....

Два взаимоисключающих режима работы:

- режим обнаружения ошибок кратности $q<=2$;
- режим исправления ошибок кратности $q=1$

Особенность проверочной матрицы кода Хэмминга:

для двоичного (n,k)-кода $n=2^w-1$ столбцов состоят из всех возможных двоичных векторов с $r=n-k$ элементами, исключая вектор со всеми нулевыми элементами.

$$H_{7,4} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Порождающая матрица:

$$G_{7,4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Кодовые вектора дополняются двоичным разрядом так, чтобы число единиц, содержащихся в каждом кодовом слове, было четным.

Преимущества:

- длины кодов = 2^w;

- d_{min}=4 (обнаружение ошибок кратности q=3, коррекция ошибок кратности q=1);

- гибридный режим работы декодера: обнаружение (q=2) и коррекция (q=1) ошибок.

Проверочная матрица *H* (2^w,k)-кода получается из проверочной матрицы (2^w-1,k)-кода:

- к матрице (2^w-1,k)-кода дописывается нулевой столбец;

- полученная матрица дополняется строкой, полностью состоящей из одних единиц.

Синдром ошибки (в гибридном режиме):

При одиночной ошибке s'(r) = 1. По значению синдрома (младшие (r-1) битов) находим и исправляем ошибочный бит.

При двойной ошибке компонента s'(r) = 0, а синдром отличен от нуля. Ситуация обнаруживается, но не исправляется.