Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра радиотехнических систем

С.Б. Саломатин

КОДИРОВАНИЕ ИНФОРМАЦИИ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ

УЧЕБНОЕ ПОСОБИЕ

по курсу
«Кодирование и защита информации»
для студентов специальностей «Радиоэлектронные системы»,
«Радиоинформатика» дневной формы обучения

Минск 2005

УДК 621.391.25 (075.8)
ББК 32. 811 я 73
С 16

Р е ц е н з е н т:
зав. кафедрой СиУТ БГУИР, д-р техн. наук, проф. В.К. Конопелько

Саломатин С.Б.

С 16 Кодирование информации в радиоэлектронных системах: Учеб. пособие по курсу «Кодирование и защита информации» для студ. спец. «Радиоэлектронные системы», «Радиоинформатика» дневной формы обуч./ С.Б.Саломатин.- Мн.: БГУИР, 2005.- 96 с.: ил.
ISBN 985 – 444 – 708 - 1

Данное учебное пособие посвящено теории алгебраических кодов, позволяющих решать различные задачи кодирования в радиоэлектронных системах. В учебном пособии рассмотрены методы помехоустойчивого кодирования информации блочными алгебраическими кодами с использованием теории конечных полей. Конструкции сверточных кодов показаны с позиции теории линейных динамических систем и метода пространства состояний. Также приведены методы и алгоритмы декодирования, позволяющие обнаружить, исправить и восстановить после стирания ошибки канала.

УДК 681.327 (075.8)
ББК 32. 811 я 7

ISBN 985 – 444 – 708 - 1

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Теория кодирования возникла в конце 40-х годов и получила широкое распространение в настоящее время. Её задачи связаны с техникой передачи, обработки и защиты информации, повышением эффективности работы систем радиолокации, навигации и управления.

Основополагающими в теории кодирования были работы К. Шеннона [1,2], в которых формулируются задачи помехоустойчивой передачи информации с любой наперед заданной точностью и секретной передачи информации со сколь угодно высокой степенью защиты. Сделанные Шенноном постановки задачи и доказательство фундаментальных теорем теории информации дали толчок для поиска решения задач с использованием детерминированных (неслучайных) сигналов и алгебраических методов помехоустойчивого кодирования защиты от помех и шифрования для обеспечения секретности информации.

В последующие годы было разработано большое количество алгебраических кодов с исправлением ошибок, среди которых следует назвать коды Хэмминга, Голея, Боуза-Чоудхури-Хоквингема (БЧХ), Рида-Соломона (РС), Рида-Малера (РМ), Адамара, Юстенсена, Гоппы, Сривэставы, циклические коды, сверточные коды с разными алгоритмами декодирования, арифметические коды.

В технике передачи информации широко применяются коды Боуза-Чоудхури-Хоквингема, коды Рида-Соломона и сверточные коды. Высокоскоростные коды с обнаружением ошибок используются в протоколах взаимодействия информационных систем. Коды с исправлением ошибок находят применение в каналах радиосвязи, спутниковой связи, телеметрических системах. Стандартом стало многоуровневое кодирование информации с использованием кодов РС, перестановочных и сверточных кодов.

Низкоскоростные коды с хорошими корреляционными свойствами применяются в навигационных (Navstar) и локационных системах, позволяют эффективно решать задачи энергетической скрытности. Коды РМ нашли широкое применение в сотовых системах связи с кодовым разделением (CDMA) и специальных радиосистемах, работающих в условиях действия сильных помех или шумов.

Представление информации в конечных полях позволяет определить криптографические преобразования, использующиеся для защиты информации и криптоанализа.

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ КОДИРОВАНИЯ

1.1. Модель системы кодирования

Комбинаторно-логическое описание модели. В широком смысле термин *информация* означает запись (расположение элементарных символов, букв) вместе с семантикой – некоторым правилом определения смысла записи. *Кодирование* – это переход от одного способа представления информации к другому. Из элементов семантики существенным при кодировании может быть *отношении синонимии* – бинарного отношения, в котором находятся два равнозначных, но не тождественных выражения. Основные факторы, характеризующие кодирование, – это *реализуемость* и *надежность*. Обеспечение надежности связано с введением в записи избыточности, но рост избыточности ухудшает шансы на реализуемость. Большое значение имеют проблемы сжатия информации, т.е. устранения естественной избыточности с целью замены неуправляемой избыточности на управляемую, повышающую надежность.

Формулировка задачи кодирования. Даны два языка $L_1=<\Lambda_1, \Theta_1>$ и $L_2=<\Lambda_2, \Theta_2>$, где Λ_i – собственно язык, т.е. множество сообщений, а Θ_i – отношение синонимии в нем. Язык называется простым, если Θ есть равенство, тогда $L=\Lambda$. Пусть задано ограничение φ, отображение Λ_1 в q^{Λ_2} и имеется функция Σ, сопоставляющая элементам в Λ_2 действительные числа (стоимости). Требуется найти в некотором классе отображений *кодирование* $f:\Lambda_1 \to \Lambda_2$, т.е. такое отображение, что

- $f(\xi) \in \varphi(\xi)$ для любого сообщения ξ из Λ_1;
- если $<f(\xi), f(\zeta)> \in \Theta_2$, то сообщения ξ, ζ принадлежат Θ_1.

Если задать критерий качества $J \Rightarrow \min \Sigma$ и потребовать, чтобы выполнялось ограничение $\Sigma(f(\xi))=J(\xi)$, тогда кодирование f называется *оптимальным*. При этом говорят – для кодирования достигается *граница*.

Рассмотрим модель кодирования для системы передачи информации (рис. 1.1).

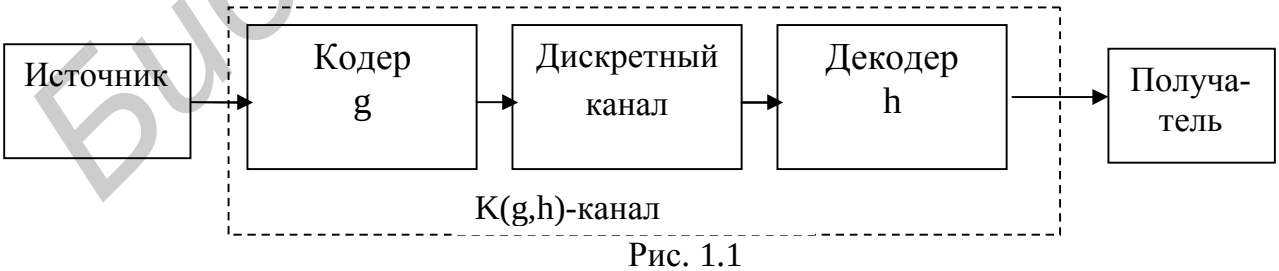


Рис. 1.1

Система передает информацию от источника к адресату и содержит: источник сообщения (детерминированный или случайный) и его получателя, схемы кодирования и декодирования, дискретный канал. Вход канала кодирования есть вход схемы кодирования g, выход канала – выход схемы декодирования h, $K_u(g,h)$ – канал с

данными g, h из некоторого класса управляющих систем U. Дискретный канал включает модели средств передачи и извлечения информации, среды распространения. В дискретном канале закодированное сообщение может подвергаться искажением, если имеется некоторый источник помех – определенным образом организованных или случайных.

Функция декодера состоит в выдаче получателю предполагаемого сообщения в требуемой форме. Для помехоустойчивых кодов декодер *исправляет* возможные ошибки, а также выполняет другие функции. Например, можно спроектировать декодер так, чтобы в «доверенных» случаях он выдавал получателю предполагаемое сообщение, а в «сомнительных» - специальный *сигнал отказа от декодирования.* Если между получателем и источником сообщений существует канал обратной связи, то декодер вместо исправления ошибок выполняет более простую операцию — *обнаружение ошибок.* Если ошибки обнаружены, то по каналу обратной связи передается, например, сигнал о повторной передаче сообщения.

В функционировании схемы декодирования можно выделить два этапа: *локализация информации* в закодированном сообщении (т.е. выделение в нем информационно законченных фрагментов) и собственно декодирование – *восстановление информации.*

В процессе кодирования и декодирования должны выполняться условия:

- согласования языков, которое означает, что в канале данных K(g,h) при общении источника с адресатом не происходит потерь информации;

- согласование процесса кодирования и декодирования с характеристиками дискретного канала, что позволяет минимизировать влияние источника помех.

При построении математических моделей каналов передачи информации рассмотрение языка адресата можно заменить рассмотрением языка канала, который включает в себя алфавит канала и полную синонимию канала. Такой моделью могут быть описаны разнообразные случаи кодирования, возникающие в технике связи, преобразования информации в памяти вычислительных систем, криптографии. К проблемам, которые составляют основу углубленного исследования, относятся проблемы взаимной однозначности кодирования, и проблема сложности реализации канала передачи информации или его элементов при заданных условиях.

Кодирование источника. Функции кодера могут быть различными. Он может просто изменять форму представления входного сообщения, используя на выходе другой алфавит. Если источник сообщений обладает статистической избыточностью (например, одни символы встречаются в среднем чаще других), то функцией кодера может быть устранение этой избыточности, т.е. наиболее экономное представление входных данных. Эту операцию называют *кодированием источника или кодированием для канала без шума.*

Источники информации удобно описывать вероятностной моделью, выходом которой служат события или случайные величины, заданные на некотором вероятностном распределении. Предположим, что U представляет собой конечное множество и D – вероятностное распределение U, т.е. $D{:}U\to[0,1]$ с $\sum_{x\in U} D(x)=1$. Пусть X – случайная переменная, заданная распределением D. Энтропия D определяется как

$$H(D)=\sum_{x\in X} D(x)\log\frac{1}{D(x)}.$$

Для конечного множества U и для вероятностного распределения $D{:}U\to[0,1]$ известна теорема Шеннона, согласно которой существуют функции кодирования $En{:}U\to\{0,1\}^*$ и декодирования Dec: $\{0,1\}^*\to U$, такие, что для каждого $x\in U$, $Dec(En(x)) = x$ и $M_{x\leftarrow D}[|En(x)|]\in[H(D),H(D)+1]$, где $M_{x\leftarrow D}[f(x)\cdot]$ - математическое ожидание функции $f(x)$, когда величина x выбирается из распределения D; $|\mathbf{s}|$ - обозначает длину вектора $\mathbf{s}$.

Теорема показывает, что возможно округление вероятностных характеристик до степени двойки т.е. создание $D`$ для каждого $x$, таких, что $D`(x)=2^{-i}$ для некоторых целых i и $D(x)/2 < D`(x) \le D(x)$. Существует алгоритм кодирования $En$ элемента $x$ (с новой вероятностью появления $D`(x)$, равной 2^{-i}) кодом из i бит. Математическое ожидание длины кодового слова находится в пределах желаемого диапазона.

Кодирование канала. Вопросы помехоустойчивости рассматриваются в двух постановках:

- замещение символов в некоторых позициях кодовых комбинаций, их называют аддитивными ошибками, так как результат их действия можно изобразить прибавлением некоторого вектора-ошибки (поразрядно) к кодовой комбинации;
- второй тип – синхронизационные помехи, результатом которых является рассогласование схемы кодирования и схемы декодирования.

Основными формами помехоустойчивости канала передачи информации является способность обнаруживать некоторое количество ошибок либо более сильная способность – исправлять ошибки. Эти эффекты и методы, приводящие к ним, прослеживаются на модели равномерного блочного кодирования. Равномерное кодирование с параметрами k и n определяется следующим образом. Сообщение $\boldsymbol{a}$ разбивается на блоки длиной k: $\boldsymbol{a} = (a_0,\ldots,a_{k-1})(a_k,\ldots,a_{2k-1})\ldots(a_{sk},\ldots,a_{sk-1})$. Блоки длиной k рассматриваются теперь как буквы алфавита и кодируются словами длиной n в соответствии со схемой кодирования.

Избыточностью на символ сообщения называется величина $(n-k)/k$. Проблема регулирования избыточности состоит не в том, чтобы просто повысить надежность за счет введения большой избыточности, а в том, как с помощью по возможности меньшей, специальным образом вводимой избыточности достичь нужной степени

надежности. При статистическом подходе характеристикой надежности канала является вероятность неправильного декодирования.

1.2. Модели источников ошибок

Источники ошибок можно разделить на два класса: детерминированные и стохастические.

Источник детерминированных аддитивных ошибок описывается множеством $P_a(N,t)$ q-ичных векторов ошибок, состоящим из всех q-ичных последовательностей $x_0,x_1,\ldots,x_s$, у которых вес любого фрагмента $x_i,x_{i+1},\ldots,x_{s+3+i}$ не превосходит t, если длина фрагмента $p \le t$. Это означает, что если по каналу передано сообщение ξ, то на выходе схемы декодирования может быть получено любое слово из множества

$$\{\xi+\nu \bmod q \mid \nu \in P_a(N,t),\ |\xi|=|\nu|\}.$$

Источник синхронизационных ошибок. Описывается подмножеством $P_c(n)$ множества $\{1, 2, \ldots, n\text{-}1\}$. При этом подразумевается, что если было передано сообщение $x_0, \ldots, x_i, \ldots$, то на вход схемы декодирования может поступить любое слово из $\{x_{sn+i}, x_{sn+i+1}, \ldots \mid s \ge 0,\ i \in P_c(n)\}$. В случае синхронизационной ошибки исправление с задержкой T состоит в нахождении правильного способа локализации, т.е. в определении значения t по модулю n на основе отрезка полученного сообщения.

Модели стохастических каналов. Пусть $c_1,\ldots,c_2,\ldots$ — последовательность символов на входе дискретного канала, а $y_1, y_2,\ldots$– последовательность символов на выходе дискретного канала. Если $c_i \neq y_i$, то говорят, что в i-м символе произошла ошибка. Статистические свойства дискретного канала описываются набором условных вероятностей, т.е. вероятностей получить на выходе канала одну последовательность при условии, что на входе была другая последовательность. Графически модель канала изображается в виде графа, в узлах которого размещают элементы множества входного и выходного алфавита, а ребрам присваивают веса вероятностных переходов из входного узла в выходной.

Наиболее простыми являются *каналы без памяти* (ДКБП). Для этих каналов символы искажаются независимо: ошибка в любом символе не влияет на вероятность ошибки в остальных символах. Каналы без памяти полностью описываются одномерными *условными вероятностями:* $w(b \mid a) = P(y_i = b \mid c_i = a)$, $a_i, b_i \in A$. Вероятность искажения символа определяется как

$$P(a) = P(y_i \neq a \mid c_i = a) = \sum_{b \in A, b \neq a} w(b \mid a).$$

Если вероятность искажения символа $P(a) = P = \text{const}$ для всех a, то дискретный канал называется *симметрично искажающим*. Примером такого канала является

двоичный симметричный канал (ДСК), показанный на рис. 1.2. Алфавит этого канала состоит из символов 0 и 1, а матрица условных вероятностей

$$W = \begin{bmatrix} w(0|0) & w(0|1) \\ w(1|0) & w(1|1) \end{bmatrix} = \begin{bmatrix} 1-P & P \\ P & 1-P \end{bmatrix}.$$

Рассмотрим блок из n символов на выходе симметрично искажающего канала. Можно подсчитать вероятность s-кратных ошибок, т. е. того, что в каких-либо s фиксированных позициях произойдут ошибки, а в остальных позициях их не будет. Эта вероятность не зависит от того, какой n - блок был на входе: $P_S = P^S(1-P)^{n-s}$. Если P < 0,5, то P_s, уменьшается с ростом s. При передаче по симметрично искажающему каналу наибольшую вероятность имеют одиночные ошибки, затем двойные, тройные и т.д.

Для ДСК с вероятностью ошибки p известна *теорема Шеннона*, согласно которой для каждой p < 1/2 существует константа c<∞ и пара функций кодирования En:$\{0,1\}^k \to \{0,1\}^{kc}$ и декодирования Dec:$\{0,1\}^{kc} \to \{0,1\}^k$ со следующими свойствами: если сообщение выбирается из случайного множества $\{0,1\}^k$ с равномерным распределением вероятностей, кодируется функцией En, передается по дискретному каналу с шумами, то декодирование функцией Dec восстанавливает сообщение с вероятностью (1 - o(1)), где величина o(1) зависит только от k.

Канал со стиранием. В канале принимаемые символы разделяются на две категории: надежные и ненадежные. Надежные символы с вероятностью (1-ε) принимаются и поступают в декодер. Ненадежные символы с вероятностью ε стираются и на вход декодера поступает сигнал стирания “?” с соответствующим номером позиции стертого символа. Граф модели стирающего канала показан на рис. 1.3.

Канал без памяти с дискретным входом и аддитивным гауссовским шумом представляет собой канал без памяти с дискретными входным $X = \{x_1,\ldots x_Q\}$ и выходным $Y = \{-\infty,\ldots y_i,\ldots \infty\}$ алфавитами и условной плотностью вероятностей, описывающейся законом Гаусса.

Канал с аддитивным шумом. Определяется как канал, для которого входное пространство – множество действительных чисел (или действительных векторов) и выход представляется как сумма входа и статистически независимой случайной величины (или вектора), называемой шумом. Широко используется модель белого гауссовского шума (БГШ). Предположим, что шум n имеет плотность распределения вероятности $p(n)$. Для заданного входа x выход принимает значение y тогда и только тогда, когда $n = y – x$. Так как n не зависит от x, то переходная плотность вероятности задается равенством $p_{Y/X}(y|x) = p(y\text{-}x)$.

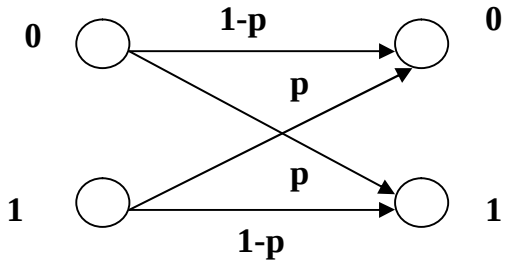


Рис. 1. 2

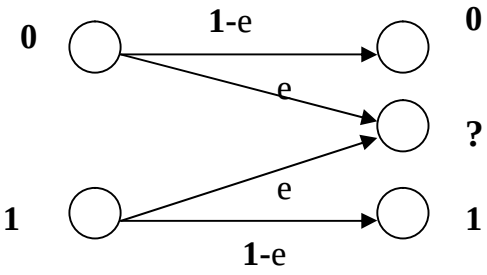
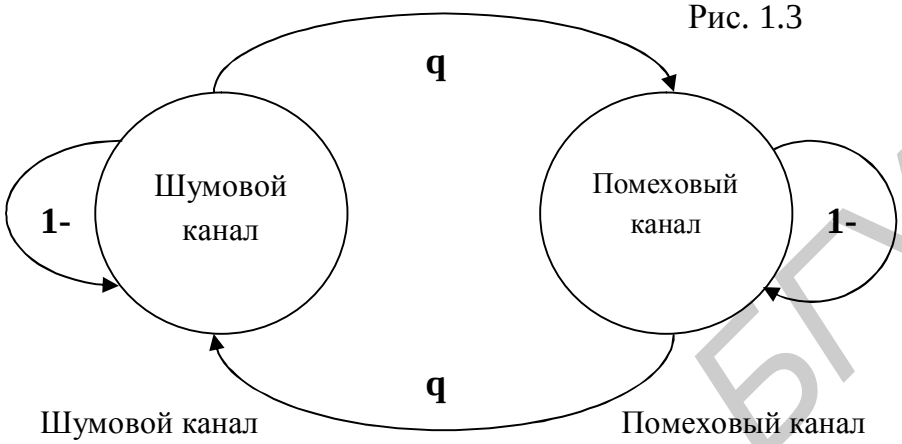


Рис. 1.3



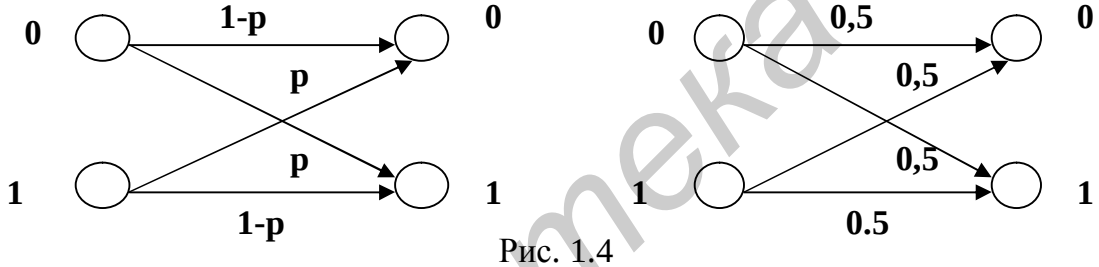


Рис. 1.4

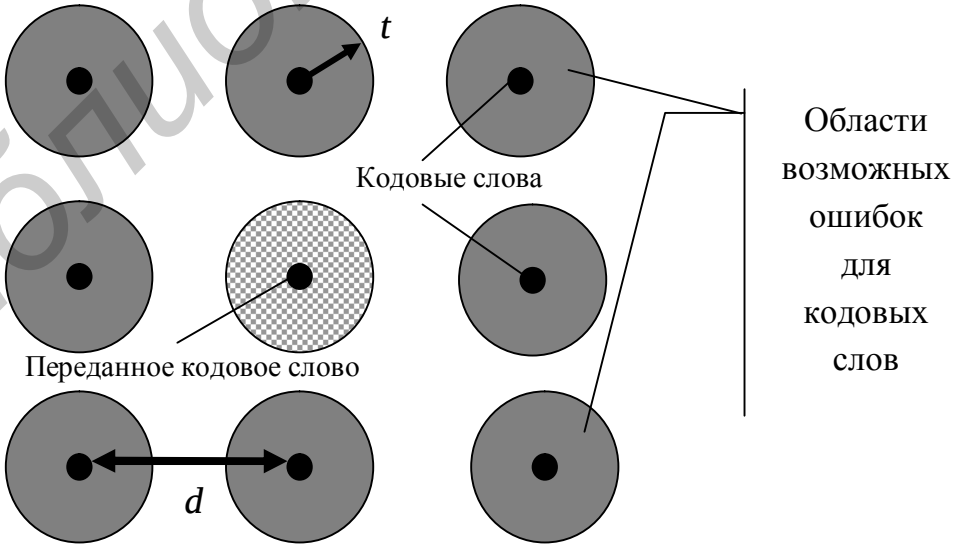


Рис. 1.5. Области декодирования

Дискретные каналы с памятью. Для дискретного канала с памятью каждая буква выходной последовательности статистически зависит как от соответствующего входа, так и от прошлых входов и выходов. В качестве памяти рассматриваются параметры, имеющие физический смысл (например, уровень замирания сигнала). Дискретный канал с конечным множеством состояний имеет на входе последовательность **x** , на выходе последовательность **y** и последовательность состояний **s.** Статистически канал описывается условной вероятностью $p(y_n, s_n | x_n, s_{n-1})$. Для квазистационарного случая модель описывается цепью Маркова с конечным числом состояний, причем каждое состояние представляется стационарным ДСК. Другим примером канала с памятью и аддитивным шумом является канал с *пакетами ошибок*. В таком канале наиболее вероятными конфигурациями ошибок являются пакеты заданной длины. Используется модель пакетированных ошибок с привлечением ключевого канала, имеющего два состояния – хорошее («шумовой канал») и плохое (помеховый канал) (рис. 1.4). В первом состоянии канал искажает сигнал в соответствии с моделью ДСК. В состоянии «помеховый канал» возможно искажение каждого передаваемого бита с вероятность 1/2.

1.3. Блоковые и неблоковые коды

При кодировании могут использоваться различные стратегии, которые обычно разбивают на два класса — блоковое и неблоковое кодирование.

Блоковое кодирование состоит в том, что последовательность символов источника сообщений $(a_1, a_2, \ldots)$ разбивается на блоки, например, по k символов в каждом: $\mathbf{a}_1=(a_1, a_2, \ldots, a_k)$, $\mathbf{a}_2=(a_{k+1}, a_{k+2}, \ldots, a_{2k})$. Кодер преобразует каждый входной k-блок $\mathbf{a_i}$ в выходной n-блок $\mathbf{c}_i = c(\mathbf{a}_i) = (c_1(\mathbf{a}_i), \ldots, c_n(\mathbf{a}_i))$, $n \geq k$ таким образом, чтобы различным входным блокам соответствовали различные выходные.

Блоковое кодирование можно интерпретировать следующим образом. Будем рассматривать входные k-блоки $\mathbf{a} = (a_1, a_2, \ldots, a_k)$ как буквы «укрупненного» алфавита A_k. Мощность этого алфавита $|A_k|=q^k$. Аналогично выходные n-блоки $\mathbf{c} = (c_1, \ldots, c_n)$ будем считать буквами «укрупненного» алфавита A_n. Мощность этого алфавита $|A_n| = q^n$. Кодирование ставит в соответствие каждой входной букве a некоторую выходную букву $\mathbf{c}(\mathbf{a})$. Совокупность C всех таких различных букв $\mathbf{c}(\mathbf{a})$ называется *блоковым кодом* длиной n и мощностью (число различных букв или объема) $M = q^k$. *Скорость кода* в q-ичных единицах измерения $R = [\log_q M]/n = k/n$.

В более общем случае код C определяется как произвольное подмножество букв алфавита A_n объемом M, причем M необязательно равно q^k, а может быть любым целым числом от 1 до q^n. Эти определения удобнее, когда алфавиты источника сообщений и кодера не совпадают. Если рассматривать процесс кодирования в

динамике, то можно полагать, что в дискретные моменты времени $i = 1,2,\ldots$ на вход кодера поступают *информационные* k- блоки a_i, которые преобразуются в *кодовые* n-блоки $c_i = \varphi(\mathbf{a}_i)$, причем i-й выходной блок зависит только от i-го входного блока, но не от более ранних или более поздних блоков. Говорят также, что блоковое кодирование является *кодированием без памяти.*

Важным частным случаем блокового кодирования является *безызбыточное кодирование*, когда размеры кодовых и информационных символов совпадают. Характеристики такой системы определяют целесообразность применения того или иного способа кодирования.

Для *неблокового кодирования* характерно наличие *памяти.* Существует много способов введения памяти, что отражается и в названиях соответствующих методов (цепные коды, коды с зацеплением, древовидные коды, сверточные коды и др.).

Пусть на входе кодера в момент i доступен не только информационный блок $\mathbf{a}_i$ из k_0 символов, но и m предыдущих информационных блоков $\mathbf{a}_{i-1}$, $\mathbf{a}_{i-2}$, … , $\mathbf{a}_{i-m}$. Все эти блоки преобразуются в текущий кодовый блок $\mathbf{c}_i = \varphi(\mathbf{a}_{i-1}, \mathbf{a}_{i-2}, \ldots, \mathbf{a}_{i-m})$ из n_0 символов. Говорят, что кодер имеет память m. Функция кодирования φ должна быть такой, чтобы при фиксированных значениях информационных блоков различным информационным блокам соответствовали различные кодовые блоки. Однако допустимо, чтобы двум наборам информационных блоков $(\mathbf{a}_i, \mathbf{a}_{i-1}, \ldots, \mathbf{a}_{i-m})$, и $(\mathbf{b}_i, \mathbf{b}_{i-1}, \ldots, \mathbf{b}_{i-m})$ соответствовал один и тот же кодовый блок. Из определения следует, что при неблоковом кодировании с памятью m информационный блок $\mathbf{a}_i$ влияет на $m + 1$ кодовых блоков $\mathbf{c}_i, \ldots, \mathbf{c}_{i+m-1}$. Таким образом: $n' = n_0 (m+1)$ кодовых символов явно зависят от блока $\mathbf{a}_i$. Поэтому величину n' называют *длиной кодового ограничения.* Для неблокового кодирования понятие «объем кода» не имеет смысла из-за этой зависимости. Рассмотрим s последовательных информационных блоков $a_o, a_ь \ldots, a_{s-i}$, которые определяют $M_s = (q^k)^s$ различных сообщений. Общая длина кодовых блоков равна $n_s = n_0 (s + m - 1)$. Частичная скорость s-блока кода определяется как $R_s = \dfrac{\log_q M_s}{n_s} = \dfrac{k_0}{n_0}\dfrac{s}{s+m}$, а скорость неблокового кодирования $R_{св} = \lim\limits_{s\to\infty} R_s = {k_0}/{n_0}$.

Нормы, метрики и кодовые расстояния блочных кодов. На выходе дискретного канала из-за ошибок может появиться любой блок из общего числа q^n различных блоков. Назначение декодера состоит в том, чтобы по принятому n-блоку с максимальной достоверностью указать, какой из кодовых блоков передавался. Одним из критериев качества кода является *кодовое расстояние.*

Определим предварительно операции над буквами конечного алфавита *A*. Будем считать, что алфавит *A* наделен структурой конечного поля. Это означает, что его объем $q = p^s$, где p — простое, а s - положительное целое число. Поле из q элементов обозначается через $GF(q)$. Блоки длиной n с координатами из $GF(q)$ можно складывать: $\mathbf{x} + \mathbf{y} = (x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$ и умножать на элементы из GF(q): $a\mathbf{x} = a(x_ь, \ldots, x_n) = (ax, \ldots, ax_n)$. Сами блоки называют векторами, а множество всех векторов - линейным векторным пространством над $GF(q)$. Рассмотрим произвольный вектор $\mathbf{x} = (x_1\, x_2, \ldots, x_n)$.

Нормой (или весом) Хэмминга $wt(\mathbf{x})$ вектора **x** называется число его ненулевых координат. Норма вектора **x** обозначается через $|\mathbf{x}|$.

Расстоянием (метрикой) Хэмминга $d(\mathbf{x}, \mathbf{y})$ между векторами **x** и **y** называется норма их разности: $d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$. Расстояние Хэмминга между двумя векторами равно числу несовпадающих координат.

Можно показать, что $||\mathbf{x}| - |\mathbf{y}|| < d(\mathbf{x}, \mathbf{y}) < |\mathbf{x}| + |\mathbf{y}|$.

Для вектора $\mathbf{x} \in R^n$ его вес wt(**x**)через расстояние Хэмминга представляется как $wt(\mathbf{x}) = d(\mathbf{x}, 0)$. Так, для слова $\mathbf{x} = (0,0,1,1)$ вес равен $wt(\mathbf{x}) = d(\mathbf{x}, 0) = 2$.

Кодовым расстоянием в метрике Хэмминга $d(C)$ кода $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\}$ называют минимальное из всех парных расстояний между его векторами:

$$d(C) = \min_{i \neq j} d(\mathbf{c}_i, \mathbf{c}_j) = \min_{i \neq j} |\mathbf{c}_i - \mathbf{c}_j|.$$

Рассмотрим декодер, который вычисляет последовательно расстояние от принятого слова **y** до каждого из возможных кодовых слов $\mathbf{c}_j$- j=1,...,*M,* и выбирает в качестве решения то кодовое слово, которое лежит ближе всего к **y**. Для слова $\mathbf{c}_j$ декодер определит, что $|\mathbf{y} - \mathbf{c}_j| = |\mathbf{e}| = t$. Для любого другого слова $\mathbf{c}_i$, $i \neq j$ декодер найдет, что $|\mathbf{y} - \mathbf{c}_i| = |\mathbf{c}_j - \mathbf{c}_i + \mathbf{e}| > ||\mathbf{c}_j - \mathbf{c}_i| - |\mathbf{e}|| > |d - t|$. Если выполняется условие $2t \leq d - 1$, то $d - t \geq t+1$, так что $|\mathbf{y}-\mathbf{c}_i| \geq t+1$. Следовательно, в качестве решения декодер выберет правильное кодовое слово $\mathbf{c}_j$. Говорят также, что декодер исправляет любые ошибки кратностью *t,* удовлетворяющей условию $2t \leq d - 1$.

Пример. Пусть заданы векторы $\mathbf{x} \in R^n$ и $\mathbf{y} \in R^n$. Рассмотрим два слова в поле $GF(2)$: $\mathbf{x} = (0,0,1,1)$ и $\mathbf{y} = (1,0,1,0)$. Расстояние Хэмминга равно $d(\mathbf{x}, \mathbf{y}) = 2$.

Геометрическая интерпретация кодового расстояния. Для некоторого $r > 0$ и вектора $\mathbf{x} \in R^n$ определим множество $S(\mathbf{x}, r) = \{\mathbf{y} \in R^n \mid d(\mathbf{x}, \mathbf{y}) \leq r\}$, которое описывает геометрическую фигуру шара радиусом r и центром в **x** (см. рис.1.5). Сфера шара описывается множеством $S_=(\mathbf{x}, r) = \{\mathbf{y} \in R^n \mid d(\mathbf{x}, \mathbf{y}) = r\}$. Количество

элементов шара или мощность шара

$$Vol(\mathbf{x},\rho) = \left|S(\mathbf{x}, r)\right| = \sum_{i=0}^{r} (q-1)^i \binom{i}{n}, \qquad \text{где } \binom{i}{n} = C_n^i = \frac{n!}{i!(n-i)!}.$$

Рассмотрим подмножества C множества R^n, обладающие тем свойством, что расстояние между двумя любыми различными словами из C не менее чем $(2e+1)$. Если взять произвольное слово $\mathbf{x} \in C$ и изменить у него t координат, где $t \le e$, т.е. внести t ошибок, то полученное слово, тем не менее, ближе к исходному, нежели любое другое из множества C — оно имеет меньшее расстояние до $\mathbf{x}$, чем другие слова из C. Код, корректирующий ошибки, можно определить как подмножество C множества R^n, обладающее следующим свойством:

$$\forall \mathbf{x} \in C, \ \forall \mathbf{y} \in C \quad \left[\mathbf{x} \neq \mathbf{y} \to d(\mathbf{x},\mathbf{y}) = 2e+1\right] \text{ или } \left[\mathbf{x} \neq \mathbf{y} \to S(\mathbf{x},e) \cap S(\mathbf{y},e) = 0\right].$$

Связь корректирующей способности кода с кодовым расстоянием. Предположим, что слова корректирующего кода задают координаты центров пространственно расположенных шаров, имеющих радиус t. Внутри шара располагаются слова, получающие из кодового слова в результате искажения t символов (так называемые запрещенные комбинации).

1. *Обнаружение* $t_{обн}$ *ошибок*. Потребуем, чтобы сферы не захватывали соседние центры. В этом случае центры шаров однозначно различимы, но при этом кодовое расстояние должно удовлетворять соотношению

$$d(C) = d_{\min} \ge t_{обн} + 1 \ \Rightarrow \ t_{обн} = d_{\min} - 1.$$

2. *Исправление* $t_{испр}$ *ошибок*. Возьмём кодовые векторы $\mathbf{x}_i$ и $\mathbf{x}_j$, образуем вокруг них сферы радиусом t и потребуем, чтобы эти сферы не пересекались, что гарантирует исправление ошибок. Тогда получаем условия для требуемого кодового расстояния: $d_{\min} = 2t_{испр} + 1$, $\quad t_{испр} = \dfrac{d_{\min} - 1}{2}$.

3. *Обнаружение и исправление ошибок*. Это случай является композицией первых двух: $d_{\min} \ge t_{испр} + t_{обн} + 1$.

Пример. Пусть $n = 15, d_{\min} = 5$, то $t_{испр} = 2$. Если следует исправить 2 ошибки и обнаружить 4, то требуется $d_{\min} = 7$.

Код $C \subset R^n$, исправляющий ошибки, называются совершенным, если $\bigcap_{\mathbf{x} \in C} S(\mathbf{x},e) = R^n$.

2. КОНЕЧНЫЕ ПОЛЯ

2.1. Алгебраические структуры

В этом разделе дается определение таких алгебраических структур, как группы, кольца и поля, а также излагаются полиномиальная и другие концепции представления в конечных полях. Будем использовать следующие обозначения числовых множеств: N –натуральные ; Z –целые; Q - рациональные; R – действительные; C – комплексные числа.

Бинарные операции. Предположим, имеется множество S и пусть S×S обозначает множество всех упорядоченных пар (s,t), где s∈ S, t∈ S. Отображение из S×S в S называется бинарной операцией на S.

Алгебраическая структура. Множество S вместе с одной или более операцией на S называется алгебраической структурой.

Группа. Группой называется такое множество G, определенное совместно с бинарной операцией * на G , для которого справедливы следующие аксиомы:

- ассоциативности, согласно которой для любых элементов $a,b,c \in G$ выполняется равенство $a*(b*c)=(a*b)*c$;
- замкнутости, для любых $a, b \in G$ операция $a*b = c \in G$;
- существования нейтрального элемента e в G такого, что для всех $a \; \hat{I} \; G$ $a*e=e*a=a$;
- для каждого $a \; \hat{I} \; G$ существует обратный элемент $a^{-1} \; \hat{I} \; G$, такой, что $a* a^{-1}=e$.

Если для всех $a,b \; \hat{I} \; G$ выполняется соотношение $a*b=b*a$, тогда группа называется абелевой, или коммутативной. Часто группа обозначается в виде тройки $<G,*, e>$.

Примеры. $<\mathbf{Z}_n,+,0>$, <R,+,0>, и <R,×,1> –являются группами. Здесь $\mathbf{Z}_n$ обозначает множество остатков от деления всех целых чисел на n, т.е. $\mathbf{Z}_n = \{0,1,...,n-1\}$. Обозначим множество, содержащее ненулевые элементы множества $\mathbf{Z}_n$ как $\mathbf{Z}_n^* = \{1,...,n-1\}$. Тогда $<\mathbf{Z}_5^*,\times,1>$ является группой относительно операции умножения.

Заметим, что для любого простого числа p $<\mathbf{Z}_p^*,\times,1>$ формирует группу, которую называют мультипликативной группой целых чисел по модулю p. Для любого положительного целого числа n $(\mathbf{Z}_n,+,1)$ формирует группу, которую называют аддитивной группой целых чисел по модулю n.

Определим некоторое простое число p. Для любого целого числа a: $0 < a < p$, наибольший общий делитель НОД(a, p) = 1. Более того, существуют два целых числа u и v таких, что $au + pv = 1$, где $0 < u < p$. Кроме того, можно записать $au + pv \pmod p =$

$=au \pmod p$. Следовательно, получаем, что $au(\mathrm{mod}\, p) = 1 \Rightarrow a^{-1} = u \in \mathbf{Z}_p^*$. Иными словами, u является инверсией числа a в $\mathbf{Z}_p^*$.

Мультипликативная группа $\boldsymbol{G}$ называется *циклической*, если существует элемент $a \in \boldsymbol{G}$, такой, что для любого $b \in \boldsymbol{G}$ существует такое целое число i, что $b = a^i$. Этот элемент a называется генераторным элементом группы и группа представляется как $\boldsymbol{G} = \langle a \rangle$.

Примеры. Для аддитивной группы целых чисел $<\mathbf{Z}_n, +, 0>$ генераторными являются числа 1 и –1.

Для аддитивной группы $<\mathbf{Z}_6, +, 0>$ – генераторными являются числа 1 и 5.

Для мультипликативной группы $<\mathbf{Z}_5^*, \times, 1>$ чисел по модулю 5 генераторными являются числа 2 и 3.

Группа называется конечной, если она состоит из конечного числа элементов. Число элементов в группе называется ее порядком и обозначается как $|\boldsymbol{G}|$.

Кольца и поля. Наиболее часто применяемые числовые системы используют две бинарные операции: сложение и умножение. Основные свойства таких числовых систем описывают алгебраические структуры типа кольцо и поля.

Кольцо $<R, +, \times>$ определяется как множество R, на котором определены две бинарные операции, обозначаемые как + и $\times$, такие, что:

- R является абелевой группой относительно операции +;
- $<R, \times>$ – подгруппа;
- операция $\times$ является ассоциативной, т.е. $(a \times b) \times c = a \times (b \times c)$ для всех $a, b, c \in R$;
- выполняется дистрибутивный закон $a(b+c) = ab+ac$ и $(b+c)\, a = ba+ca$ для всех $a, b, c \in R$.

Примеры. Множества $<\mathbf{Z}, +, \times>$, $<Q, +, \times>$, $<R, +, \times>$ и $<C, +, \times>$ являются кольцами. Множество $<\mathbf{Z}_4, +, \times>$ формирует кольцо из 4 элементов, которые образуют алгебраическую структуру кода Z_4. Множество $<\mathbf{Z}_n, +, \times>$ образует кольцо, которое называется кольцом класса вычетов по модулю n.

Пусть $<F, +, \times>$ представляет собой кольцо. Предположим, что существует множество отличных от нуля элементов $F^* = \{a \in R \mid a \neq 0\}$.

Поле можно определить как кольцо $<F, +, \times>$, такое, что множество F^* совместно с операцией умножения образует коммутативную группу. Согласно определению поле это множество F, на котором определены две бинарные операции, называемые сложением и умножением, и которое содержит два отличающихся друг от друга элемента 0 и 1. Более того, множество $<F, +, 0>$ является абелевой группой относительно сложения, при этом элемент 0 рассматривается как нейтральный.

Множество $< F^*, \times, 1 >$ представляет собой абелеву группу относительно операции умножения, принимая 1 как нейтральный элемент. Две операции сложения и умножения подчиняются дистрибутивному закону. Элемент 0 называется нулевым элементом, а 1 – мультипликативным единичным элементом или единицей.

Конечным называется поле, содержащее конечное число элементов. Количество этих элементов определяет порядок поля. Конечные поля часто называют полями Галуа и обозначают как *GF*.

Примеры.

<*Q*,+, ×>, <*R*,+, ×> и <C,+, ×> являются полями.

$< \mathbf{Z}_5, +, \times >$ формирует конечное поле.

$< \mathbf{Z}_2, +, \times >$ формирует бинарное конечное поле порядка два. Элементами этого поля являются 0 и 1.

Если p – простое число, то множество $< \mathbf{Z}_p, +, \times >$ всегда образует поле класса вычетов по модулю p, которое обозначают как $\mathbf{Z}_p$ или *GF*(p).

Полиномы. Полином над произвольным кольцом R записывается в виде

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + \ldots + a_n x^n,$$

где положительное целое число n и i ($0 \le i \le n$) являются элементами R, символ x не принадлежит R, или говорят, что он неопределен над R. Полином $f(x)$ всегда можно представить в эквивалентной форме

$$f(x) = a_0 + a_1 x + \ldots + a_n x^n + 0x^{n+1} + \ldots + 0x^{n+h},$$

где h – любое положительное число. Поэтому, если сравнивать два различных полинома, определенных над кольцом R, можно предполагать, что они имеют одинаковую степень.

Полиномы $f(x) = \sum_{i=0}^{n} a_i x^i$ и $g(x) = \sum_{i=0}^{n} b_i x^i$, определенные над кольцом R, считаются равными только в том случае, если $a_i = b_i$ для всех $0 \le i \le n$.

Сумма двух полиномов определяется как

$$f(x) + g(x) = \sum_{i=0}^{n} (a_i + b_i) x^i.$$

Произведение двух полиномов

$$f(x) = \sum_{i=0}^{n} a_i x^i \text{ и } g(x) = \sum_{j=0}^{m} b_j x^j$$

над кольцом R определяется как

$$f(x)g(x) = \sum_{k=0}^{n+m} c_k x^k ,$$

где $c_k = \sum_{i+j=k} a_i b_j$, а индексы i и j пробегают все значения $0 \le i \le n, 0 \le j \le m$.

Просто показать, что описанные операции сложения и умножения полиномов в свою очередь позволяют сформировать кольцо, которое называется полиномиальным кольцом над R и обозначается как $R[x]$, т.е.

$$R[x] = \{ \sum_{i=0}^{n} a_i x^i \mid a_i \in R, n \ge 0 \}.$$

Нулевым элементом в $R[x]$ является полином, имеющий все нулевые коэффициенты a_i. Степень n при старшем коэффициенте a_n ненулевого полинома $f(x)$ называется степенью полинома и обозначается как $n = \deg(f(x))$. Постоянным полиномом называется полином, имеющий степень, меньшую или равную нулю.

Если R является полем, то над этим полем можно определять полиномы. Полином $f(x)$, принадлежащий полю $F[x]$, делится полиномом $g(x) \in F[x]$, если существует такой полином $h(x) \in F[x]$, что $f(x) = g(x)h(x)$. Полином $g(x)$ называют делителем (дивизором) $f(x)$.

Полином $f(x) \in F[x]$ называется неприводимым над полем F, если $f(x)$ имеет положительную степень и $f(x) = b(x)c(x)$, где $b(x), c(x) \in F[x]$, и или $b(x)$, или $c(x)$ являются постоянным полиномом. В противном случае полином называется приводимым.

Элемент $b \in F$ является корнем полинома (или нулем) $f(x) \in F[x]$, если $f(b) = 0$.

2.2. Конструкция расширенного поля $GF(p^n)$

Положим, что n — это положительное целое число. Для построения расширенного конечного поля $GF(p^n)$ порядка p^n необходимо использовать неприводимый над $GF(p)$ полином $f(x)$ степени n. Кроме того, положим, что элемент α является корнем полинома $f(x) = 0$.

Пусть $GF(p^n) = \{a_0 + a_1 a + ... + a_{n-1} a^{n-1} \mid a_i \in GF(p)\}$. Определим две операции сложения (+) и умножения (·) следующим образом. Для $g(a), h(a) \in GF(p^n)$:

$$g(a) = \sum_{i=0}^{n-1} a_i a^i \quad \text{и} \quad h(a) = \sum_{j=0}^{n-1} b_j a^j ,$$

$$g(a)+h(a)=\sum_{i=0}^{n-1}(a_i+b_i)a^i\in GF(p^n),\ g(a)h(a)=r(a),$$

где $r(a)$ вычисляется по алгоритму:

а) выполняется умножение $g(a)$ и $h(a)$ по правилу умножения полиномов, т.е.

$$g(a)h(a)=\sum_{i=0}^{n-1}a_i a^i\sum_{j=0}^{n-1}b_j a^j=\sum_{k=0}^{n+m}s_k x^k=c(a),$$

где $c_k=\sum_{i+j=k;0\le i\le n,0\le j\le m}a_i b_j$;

б) применяется алгоритм деления $c(a)$ на $f(a)$:

$$c(a)=g(a)f(a)+r(a),\ \deg(r(a))<n,$$

при этом, учитывая, что $f(a)=0$, получаем окончательно $c(a)=r(a)\in GF(p^n)$.

Множество $GF(p^n)$ совместно с двумя определенными выше операциями формирует конечное поле порядка p. Можно говорить, что поле $GF(p^n)$ получается в результате присоединения к полю $GF(p)$ нуля $f(a)$, т.е. осуществляется расширение поля $GF(p)$.

Пример. Пусть p=2, $f(x)=x^3+x+1$ - неприводимый над $GF(2)$ полином. Положим, что $f(a)=0$, a - корень неприводимого полинома. Конечное поле $GF(2^3)$ формируется как

$$GF(2^3)=\{a_0+a_1a+a_2a^2\mid a_i\in GF(2)\}.$$

Элементы поля могут быть представлены в виде двоичного кода, полинома и степени корня a (табл. 2.1).

Таблица 2.1

Двоичный код	Полином	Степень корня a
000	0	0
001	1	1
010	a	a
100	a^2	a^2
110	$1+a$	a^3
011	$a+a^2$	a^4
111	$1+a+a^2$	a^5
101	$1+a^2$	a^6
$a^7=1$		

Заметим, что $GF(2^3)^* = \langle a \rangle$, иными словами, ненулевые элементы поля $GF(2^3)$ формируют циклическую группу порядка 7. Генераторным, или порождающим, является элемент a, $a^7 = 1$.

Пример. Пусть p=2, $f(x) = x^4 + x + 1$ - неприводимый над $GF(2)$ полином. Положим, что $f(a) = 0$, a - корень неприводимого полинома. Конечное поле $GF(2^4)$ формируется как $GF(2^4) = \{a_0 + a_1 a + a_2 a^2 + a_3 a^3 \mid a_i \in GF(2)\}$.

Элементы поля можно представить в виде табл. 2.2.

Таблица 2.2

Представление в виде коэффициентов полинома				Представление в виде степени корня неприводимого полинома
a_0	a_1	a_2	a_3	
0	0	0	0	$0 = a^{\infty}$
1	0	0	0	$a^0 = 1$
0	1	0	0	a
0	0	1	0	a^2
0	0	0	1	a^3
1	1	0	0	a^4
0	1	1	0	a^5
0	0	1	1	a^6
1	1	0	1	a^7
1	0	1	0	a^8
0	1	0	1	a^9
1	1	1	0	a^{10}
0	1	1	1	a^{11}
1	1	1	1	a^{12}
1	0	1	1	a^{13}
1	0	0	1	a^{14}
$a^{15} = 1$, $a^{16} = a$ - циклическое повторение форм представлений				

Сложим и умножим два элемента поля $(1+a)$ и $(a+a^3)$. Получим следующие результаты: $(1+a)+(a+a^3) = 1+a^3$ и $(1+a)(a+a^3) = a^4 a^9 = a^{13} = 1+a+a^3$.

Для рассматриваемого примера $GF(2^4)^* = \langle a^i, i = 0,...14 \rangle$ является циклической группой порядка 15 с генераторным элементом a, $a^{15} = 1$.

2.3. Основные свойства конечных полей

Пусть F является конечным полем и существует положительное целое m, такое, что $mb = 0$ для любого $b \in F$, тогда положительное наименьшее число m называется характеристикой F или, другими словами, что F имеет характеристику m. Если подмножество K множества F само является полем относительно операций в F, то оно называется подполем F. Поле F называется расширением K.

Анализ конечных полей показывает, что если F является конечным полем порядка q и характеристики p, тогда $q = p$ или $q = p^n$ (n ³ 1). Два конечных поля F и G называются изоморфными, если существует однозначное отображение из F в G с сохранением операций сложения и умножения. Все конечные поля порядка p^n являются изоморфными. Для конечного поля F можно определить мультипликативную группу F^* ненулевого элемента F. Если для некоторого $a \in F$ существует положительное наименьшее целое число r, такое, что $a^r = 1$, то r – называется порядком a и записывается как $ord(a) = r$.

Генераторный элемент циклической группы $GF(p^n)^*$ называется примитивным элементом поля $GF(p^n)$. Полином, для которого примитивный элемент является корнем, называется примитивным полиномом. Заметим, что не все неприводимые полиномы являются и примитивными. Так, полином $x^4 + x^3 + x^2 + x + 1$ является неприводимым, но не примитивным. Соответственно, если F –поле порядка p^n, элемент α является примитивным, если он имеет порядок p^n-1. Можно показать, что любое поля содержит примитивный элемент.

Теорема Ферма. Каждый элемент β поля порядка p^n удовлетворяет тождеству $b^{p^n} = b$, или, что эквивалентно, является корнем уравнения $x^{p^n} = x$. Откуда следует, что $x^{p^n} - x = \prod_{b \in F} (x - b)$.

2.4. Представление элементов конечного поля

Конечное поле $GF(p^n)$ можно рассматривать как векторное пространство размером n, построенное над $GF(p)$.Следовательно, любое множество n линейно независимых элементов может образовывать базис векторного пространства.

Известны два основных базиса, которые используются для конструирования конечного поля.

Полиномиальный базис $1, a, ..., a^{n-1}$ используется для формирования поля из неприводимого полинома $f(x)$ над $GF(p)$ степени n, $f(a) = 0$.

Нормальный базис образуется элементами $a, a^p, ..., a^{p^{n-1}}$, если они линейно независимы над $GF(p)$. Нормальный базис всегда существует для $GF(p^n)$.

Положим, что α является примитивным элементом $GF(p^n)$ и что $\{b_0, b_1, ..., b_{n-1}\}$ образует базис $GF(p^n)$. Тогда элементы поля можно записать как

- $GF(p^n) = \{a_0 b_0 + a_1 b_1 + ... + a_{n-1} b_{n-1} \mid a_i \in GF(p)\}$ - векторное представление;
- $GF(p^n) = \{a^i \mid 0 \le i \le p^n - 2 \text{ или } i = \infty\}$ - экспоненциальное представление.

Пример. Элементы конечного поля $GF(2^3)$, $f(x) = x^3 + x + 1$ представляются в различных базисах следующим образом, табл. 2.3.

Таблица 2.3

Полиномиальный базис $b_0=1,\ b_1=a,\ b_2=a^2$			*Нормальный базис* $b_0=a^3, b_1=a^6, b_2=a^5$			*Экспоненциальный базис*
a_0	a_1	a_2	a_0	a_1	a_2	
0	0	0	0	0	0	$0 = a^\infty$
1	0	0	1	1	1	$1 = a^0$
0	1	0	0	1	1	a
0	1	0	0	1	1	a^2
1	1	0	1	0	0	a^3
0	1	1	1	1	0	a^4
1	1	1	0	0	1	a^5
1	0	1	0	1	0	a^6

Методы вычисления в $GF(p^n)$. Пусть α - примитивный элемент $GF(p^n)$. Тогда любой элемент поля $0 \ne \gamma \in GF(p^n)$ представляется в виде степени $\gamma = \alpha^i$ для $i \ge 0$. Положительное целое i называется логарифмом γ по основанию α.

Сложение использует таблицы векторного представления элементов $GF(p^n)$. Умножение использует таблицы экспоненциального представления $GF(p^n)$. Таблицы представлений пересчитываются в таблицы логарифмов и антилогарифмов.

Преобразование Фурье в поле Галуа. Пусть $\mathbf{v} = \{v_i, i = 0,...,n-1\}$ — вектор над $GF(p)$, где n делит $p^m - 1$ при некотором m, и пусть α — элемент порядка n в поле $GF(p^m)$. Преобразование Фурье в поле Галуа вектора $\mathbf{v}$ определяется как вектор $\mathbf{V} = \{V_j, j = 0,...,n-1\}$, задаваемый равенствами

$$V_j = \sum_{i=0}^{n-1} a^{ij} v_i, \quad j = 0,...,n-1.$$

Дискретный индекс i можно назвать временем, а $\mathbf{v}$ – временной функцией. Индексу j в этом случае отводится роль частоты, а $\mathbf{V}$ – спектра или частотной функции.

Временной функции $\mathbf{v}$, значения которой принадлежат полю $GF(p)$, имеет спектр $\mathbf{V}$, который лежит в расширении поля $GF(p^m)$.

2.5. Минимальные полиномы и циклотомические классы

Минимальные полиномы. Теорема Ферма утверждает, что каждый элемент α конечного поля $GF(q)$, $q = p^n$ удовлетворяет уравнению $x^q - x = 0$. Однако элемент α может быть описан уравнением с меньшей степенью. Минимальным полиномом $m(x) \in GF(p)[x]$ называют полином наименьшей степени над $GF(p)$ для элемента α, такой, что $m(a) = 0$.

Пример. В поле $GF(2^4)$ выполняется равенство $a^4 + a + 1 = 0$, и элементам поля соответствуют минимальные полиномы с коэффициентами, равными 0 или 1, табл.2.4.

Таблица 2.4

Элемент поля	*Минимальный полином*
0	x
1	$x+1$
a	$x^4 + x + 1$
a^{-1}	$x^4 + x^3 + 1$
a^3	$x^4 + x^3 + x^2 + x + 1$
a^5	$x^2 + x + 1$

Свойства минимального полинома

1. Минимальный полином $m(x)$ элемента $a \in GF(p^n)$ является неприводимым.

2. Любой полином $f(x) \in GF(p)[x]$, для которого $f(a) = 0$, делится на $m_a(x)$, $m_a(x) | f(x)$.

3. Минимальный полином $m(x) | x^{p^n} - x$.

4. Степень минимального полинома $deg(m(x)) \le n$.

5. Минимальный полином примитивного элемента a поля $GF(p^n)$ имеет степень, равную n.
6. Элементы a и a^p имеют одинаковые минимальные многочлены.

Сопряженные элементы и циклотомические классы. Пусть $a \in GF(p^n)$, тогда элементы $a, a^p, ..., a^{p^{n-1}}$ называются сопряженными относительно поля $GF(p)$. Все сопряженные элементы имеют одинаковые минимальные полиномы. Операция умножения на p по модулю $\bmod p^n - 1$ делит все множество чисел на множества, называемые циклотомическими классами по $\bmod p^n - 1$ элемента s:

$$\{s, sp, sp^2, ..., sp^{n_s - 1}\},$$

где n_s - наименьшее положительное число, такое, что $p^{n_s} s \equiv s (\bmod p^n - 1)$.

Пример. Циклотомические классы по mod 15 (p = 2) запишутся как

$K_0 = \{0\}$, $K_1 = \{1,2,4,8\}$, $K_3 = \{3,6,12,9\}$, $K_5 = \{5,10\}$, $K_7 = \{7,14,13,11\}$.

Здесь символ s обозначает наименьшее число в множестве C_s и называется лидером множества по $\bmod p^n - 1$. Все элементы a^j, у которых степень j пробегает значения циклотомических классов, имеют одинаковые минимальные полиномы. Это свойство позволяет определить алгоритм вычисления минимальных полиномов.

Пример. Для поля $GF(2^4)$, $a^4 + a + 1 = 0$, получаем следующий результат (табл.2.5), где Г={1,3,5,7}.

Таблица 2.5

Циклотомическое множество	*Элемент поля*	*Минимальный полином*
	0	x
$K_0 = \{0\}$	1	$m_0(x) = x + 1$
$K_1 = \{1,2,4,8\}$	a, a^2, a^4, a^8	$m_1(x) = x^4 + x + 1$
$K_3 = \{3,6,12,9\}$	a^3, a^6, a^{12}, a^9	$m_1(x) = x^4 + x^3 + x^2 + x + 1$
$K_5 = \{5,10\}$	a^5, a^{10}	$m_1(x) = x^2 + x + 1$
$K_7 = \{7,14,13,11\}$	$a^7, a^{14}, a^{13}, a^{11}$	$m_1(x) = x^4 + x^3 + 1$

Алгоритм вычисления минимальных многочленов

Входные данные: $f(x)$ – минимальный полином степени n над полем $GF(p)$.

Выходные данные : все неприводимые полиномы над $GF(p)$, степени которых делят n.

1. Формируется конечное поле $GF(p^n)$, для которого $f(\alpha) = 0$.

2. Вычисляются все циклотомические классы по $\bmod p^n - 1$: $K_1,\ldots,K_r$ и множество Г, содержащее все лидеры модуля $\bmod p^n - 1$.

3. Вычисляются минимальные полиномы по формуле $m_s(x) = \prod_{i \in K_s} (x - a^i)$ для всех $s \in$ Г. Вычисленный полином $m_s(x)$ является минимальным для элемента a^s и всех его сопряжений.

2.6. Функции следа

Пусть q будет простым числом или степенью простого числа. $q = p^m$. Для $a \in F = GF(q^n)$ и $K = GF(q)$ функция следа $Tr_{F/K}(x)$ определяется как

$$Tr_{F/K}(x) = x + x^q + \ldots + x^{q^{n-1}}, \quad x \in F.$$

Если $a \in F$, $Tr_{F/K}(a)$ называется следом элемента α над *K*. Если $q = p$ простые числа, то используется упрощенная запись $Tr_F(a)$ или $Tr(a)$. Так как

$$Tr_{F/K}(x^q) = \sum_{i=0}^{n-1} x^{q^{i+1}} = Tr_{F/K}(x),$$

то можно говорить, что $Tr_{F/K}(x)$ представляет собой отображение из *F* в *K*. Если q=2, то след $Tr_{F/K}(x)$ равен 0 или 1, при этом

$$Tr_F(x) = x + x^2 + \ldots + x^{2^{n-1}} \in GF(2) \text{ для всех } x \in GF(2^n).$$

Свойства функции следа. Пусть $F = GF(q^n)$ и $K = GF(q)$. Тогда для функции следа $Tr_{F/K}(x)$ справедливы свойства:

1. $Tr_{F/K}(a + b) = Tr_{F/K}(a) + Tr_{F/K}(b)$ для всех $a, b \in F$.
2. $Tr_{F/K}(ca) = cTr_{F/K}(a)$ для всех $a \in F$, $c \in K$.
3. След $Tr_{F/K}(x)$ является линейным отображением из поля *F* в поле *K*.
4. $Tr_{F/K}(c) = nc$ для всех $c \in K$.
5. $Tr_{F/K}(a^q) = Tr_{F/K}(a)$ для всех $a \in F$.

Функция следа определяет такое понятие, как *аддитивный характер* конечного поля. Предположим, что p – простое число, а *r*- положительное целое. Аддитивный характер поля $GF(p^r)$ представляется как $c(x) = \exp(\frac{2pTr_p^r(x)}{p})$, $x \in GF(p^r)$, где

$$Tr_p^r(x) = Tr_{GF(p^r)/GF(p)} = x + x^p + \ldots + x^{p^{r-1}}, \quad x \in GF(p^r).$$

3. ЛИНЕЙНЫЕ КОДЫ

3.1. Векторные пространства

Совокупность всех векторов длиной n с координатами из конечного поля $GF(q)$ называют *векторным пространством* V над $GF(q)$, если определены операции суммирования любых двух векторов и умножение вектора на любой элемент из $GF(q)$. В общем случае кодом называется любое подмножество векторов. Среди этих подмножеств важную роль играют линейные подпространства.

Множество C векторов называется *линейным подпространством,* если:

1) для любых двух векторов этого множества $\mathbf{x}$, $\mathbf{y} \in C$, их сумма также принадлежит множеству C: $(\mathbf{x} + \mathbf{y}) \in C$;

2) для любого вектора $\mathbf{x} \in C$ и любого элемента $a \in GF(q)$ вектор $a\mathbf{x}$ также принадлежит множеству C: $a\mathbf{x} \in C$.

Векторное пространство V можно задать с помощью базиса из n векторов, например, с помощью векторов $d_1 = (1, 0, ..., 0)$, $d_2 = (0, 1, 0,..., 0)$,..., $d_n = (0, 0, ...,1)$. Это означает, что любой вектор из V может быть однозначно представлен как линейная комбинация этих векторов с коэффициентами из $GF(q)$.

Аналогично любое линейное подпространство C можно задать с помощью базиса из k линейно независимых векторов $(1 \le k \le n)$:

$$\mathbf{g}_1 = (g_{11}, \ldots, g_{1n}), \mathbf{g}_2 = (g_{21}, \ldots, g_{2n}), \ldots, \mathbf{g}_k = (g_{k1}, \ldots, g_{kn}). \quad (3.1)$$

Число k называют *размерностью подпространства.* Очевидно, что число различных векторов в k-мерном подпространстве равно q^k. Линейным (n, k)-кодом называют k-мерное линейное подпространство. Линейные коды играют важную роль прежде всего потому, что для них проста процедура кодирования. Для нелинейного кода объемом $M = q^h$ необходимо помнить все q^k кодовых последовательностей. Для линейного кода можно ограничиться хранением лишь k базисных векторов $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$ или эквивалентной им *порождающей матрицы кода*

$$\mathbf{G} = \begin{bmatrix} \mathbf{g_1} \\ \mathbf{g_2} \\ \mathbb{M} \\ \mathbf{g_k} \end{bmatrix} = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ g_{21} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots \\ g_{k1} & \cdots & g_{kn} \end{bmatrix}.$$

Для кодирования достаточно задать блок информационных символов $\mathbf{a} = [a_1, a_2, ..., a_k]$. Тогда кодовый вектор получается путем умножения этого вектора на порождающую матрицу **G**:

$$\mathbf{c} = \mathbf{aG} = (c_1, \ldots, c_n). \quad (3.2)$$

Ортогональная проекция кода. Пусть **H** — матрица размером r×n *с* линейно независимыми строками, где $r=n—k$, такая, что $\mathbf{GH}^T = \mathbf{0}$. Тогда любое кодовое слово **c** удовлетворяет уравнению

$$\mathbf{cH}^T = 0. \qquad (3.3)$$

Выражение (3.3) определяет проекцию кода в ортогональное пространство. Матрица **H** называется аннулирующей или *проверочной матрицей кода.*

3.2. Способы задания линейного кода

Рассмотрим более подробно способы задания линейного (n, k)-кода.

1. *Перечисление кодовых слов.* Код задается в виде списка всех кодовых слов кода.

Пример. В табл. 3.1 представлены все кодовые слова (5,3) - кода $\mathbf{C} = \{\mathbf{c}_i\}$,где $a_{i,j}$ - информационные, $b_{i,j}$ - проверочные $c_{i,j}$ - кодовые символы. В дальнейшем второй индекс j может опускаться, если его значение ясно из контекста.

Таблица 3.1

	$c_{0,j}=a_{1,j}$	$c_{1,j}=a_{2,j}$	$c_{2,j}=a_{3,j}$	$c_{3,j}=b_{1,j}$	$c_{4,j}=b_{2,j}$
$\mathbf{c}_1$	0	0	1	1	0
$\mathbf{c}_2$	0	1	0	1	1
$\mathbf{c}_3$	0	1	1	0	1
$\mathbf{c}_4$	1	0	0	0	1
$\mathbf{c}_5$	1	0	1	1	1
$\mathbf{c}_6$	1	1	0	1	0
$\mathbf{c}_7$	1	1	1	0	0
$\mathbf{c}_8$	0	0	0	0	0

2. Система проверочных уравнений. Задаются правила формирования проверочных символов по информационным:

$$b_i = \sum_{l=1}^{k} a_l h_{il} \ ,$$

где i - номер проверочного символа; l - номер информационного символа; h_{il} - коэффициенты, определяемые правилами формирования кодов.

Пример. Для кода (5,3) проверочные уравнения имеют вид

$$b_1 = a_2 + a_3;\ b_2 = a_1 + a_2.$$

3. Задание с помощью порождающей матрицы **G.** Векторное пространство R^n над $GF(q)$ включает в себя q^n векторов (n-последовательностей) и задает так называемый «полный код.». Подпространство С пространства R^n представляет собой множество (код) из q^k кодовых слов длиной n, которое однозначно определяется порождающей

матрицей **G**. Строки матрицы **G** являются базисными векторами $\mathbf{B} = \{\mathbf{v}_1,...,\mathbf{v}_k\}$ линейного подпространства *C*:

$$\mathbf{G} = \begin{bmatrix} \mathbf{v}_1 \\ ... \\ \mathbf{v}_k \end{bmatrix}.$$

Если задан информационный вектор-строка $\mathbf{a} = [a_1,...,a_k]$, то код образуется в результате умножения информационного вектора на порождающую матрицу

$$C = \mathbf{aG} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + ... + a_k\mathbf{v}_k \ . \qquad (3.4)$$

Линейный код может иметь несколько базисов. Коды считаются *эквивалентными*, если они задаются эквивалентными матрицами **G**, которые получаются из исходной матрицы с помощью элементарных операций. Стандартная форма записи матрицы *G* имеет следующий вид:

$$\mathbf{G} = [I_k | P], \qquad (3.5)$$

где $I_k = diag(1,...,1) \leftarrow (k \times k)$; $P \leftarrow (k \times (n-k))$ - это матрица, содержащая в себе информацию о проверочных соотношениях для данного кода.

Код, заданный порождающей матрицей **G** в стандартной форме, называется *систематическим*. В остальных случаях коды называются *несистематическими*. У систематического кода первые *k* символов повторяют информационные символы.

Теорема. В линейном коде наименьшее расстояние равно наименьшему весу среди всех ненулевых кодовых слов.

Доказательство. Если $\mathbf{x} \in C$ и $\mathbf{y} \in C$, то для линейного кода $\mathbf{x} - \mathbf{y} \in C$ и $d(\mathbf{x},\mathbf{y}) = d(\mathbf{x}-\mathbf{y},\mathbf{0}) = wt(\mathbf{x}-\mathbf{y})$.

4. Задание линейного кода с помощью проверочной матрицы **H**.

Если *C* – линейный код размером *k*, то *ортогональным или двойственным* ему будет код $C^{\perp}$, определяемый как

$$C^{\perp} = \left\{ \mathbf{x} \in R^n \middle| \forall \mathbf{y} \in C, (\mathbf{x},\mathbf{y}) = 0 \right\}, \qquad (3.6)$$

где $(\mathbf{x},\mathbf{y}) = \sum_i x_i \cdot y_i$ - скалярное произведение двух векторов.

Двойственный код $C^{\perp}$ является (*n*, *n*-*k*)-кодом. Если **H** – это порождающая матрица двойственного кода, то матрицу **H** называют проверочной матрицей кода *C*. Всякая матрица **H** называется проверочной, если её строки порождают пространство

$C^{\perp}$; следовательно, код C можно определить через проверочную матрицу следующим образом:

$$C = \left\{ \mathbf{x} \in R^n \mid \mathbf{x} \cdot \mathbf{H} = 0 \right\}. \tag{3.7}$$

Если $\mathbf{G} = \left[I_k \middle| P \right]$, тогда $\mathbf{H} = \left[-P^T \middle| I_{n-k} \right]$, причём $\mathbf{G} \cdot \mathbf{H}^T = -P + P = \mathbf{0}$.

Пример. Порождающая матрица **G** кода Хемминга $(n,k) = (7,4)$ в поле $GF(2)$, $c_i \in \{0,1\}$ задаётся как

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \end{bmatrix} = \left[I_4 \middle| P \right].$$

Требуется закодировать слово $\mathbf{a} = (1,0,1,0)$ с помощью порождающей матрицы и построить проверочную матрицу.

Решение. Кодовое слово определяется как

$$\mathbf{c} = \mathbf{a} \cdot \mathbf{G} = 1 \cdot \mathbf{v}_1 + 0 \cdot \mathbf{v}_2 + 1 \cdot \mathbf{v}_3 + 0 \cdot \mathbf{v}_4 = (1010\,011).$$

Проверочная матрица **H** имеет вид

$$\mathbf{H} = \left[\begin{array}{c} -\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \end{array}\middle| \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] = \left[-P^T \middle| I_3 \right].$$

Проверим, выполняется ли условие ортогональности порождающей и проверочной матриц $\mathbf{G} \cdot \mathbf{H}^T = 0$:

$$\mathbf{G} \cdot \mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{0}.$$

Параметры кода записывается в виде тройки: $(n, k, d_{\min})$. Для приведённого выше примера $d_{\min} = 3$,что дает линейный код (7,4,3).

Основные свойства линейных кодов

1. Произведение любого кодового слова на транспонированную проверочную матрицу дает нулевой вектор размерностью (n - k) :

$$\mathbf{c}_i \, \mathbf{H}^T = \mathbf{0} \, .$$

2. Произведение некоторого принимаемого слова $\mathbf{y}_i$, (возможно с ошибкой **e**) на транспонированную проверочную матрицу называется синдромом, обозначается $\mathbf{s}_i$ и определяет проекцию принимаемого сигнала в ортогональное пространство

$$\mathbf{y}_i \mathbf{H}^T = \mathbf{s}_i \, .$$

3. Между порождающей и проверочной матрицами в систематическом виде существует однозначное соответствие, а именно:

$$\mathbf{G}\mathbf{H}^T = \mathbf{0} \, .$$

5. Минимальный вес линейного (n, k)-кода C равен d тогда и только тогда, когда любые $(d - 1)$ столбцов проверочной матрицы этого кода линейно независимы, но некоторые d столбцов проверочной матрицы зависимы.

6. Линейный q-ичный $(n+1, k)$-код C_p называется *расширенным*, если он образуется из исходного линейного q-ичного (n, k)-кода C в результате преобразования

$$C_p = \left\{ (-\sum_{i=1}^{n} c_i, \; c_1,...,c_n \mid (c_1,...,c_n) \in C \right\}.$$

7. Два кода называются эквивалентными, если их порождающие матрицы отличаются перестановкой координат, т.е. порождающие матрицы получаются одна за другой после перестановки столбцов и элементарных операций над строками.

3.3. Декодирование линейного кода

Методы декодирования помехоустойчивых кодов зависят от статистических характеристик каналов, искажающих сигнал. При выбранной процедуре кодирования статистические особенности канала определяются стохастической матрицей трансформации сигналов $\lfloor p_{ij} \rfloor$. Задача оптимального декодирования заключается в том, чтобы найти процедуру декодирования с минимальным средним риском. Другими словами, необходимо указать алгоритм разбиения множества выходных «сигналов» канала на ряд M непустых, непересекающихся подмножеств.

Схема декодирования по критерию максимального правдоподобия. Применяется в системах, в которых элементы обобщенной матрицы потерь удовлетворяют условию $p_{ij} = 1/M$. Если использовать критерий максимального правдоподобия, то схему многоканального декодера можно построить так, как это показано на рис 3.1. Оценка

информационного слова определяется из алгоритма

$$\hat{\mathbf{a}} = \arg\max l(\mathbf{Y}) , \qquad (3.8)$$

где $l(\mathbf{Y})$ - отношение правдоподобия [3]. Максимум $l(\mathbf{Y})$ оценивается по минимуму кодовых расстояний между принимаемым сигналом и опорными копиями всех кодовых слов. Данная схема декодирования позволяет получить минимальную вероятность ошибки для ДСК и гауссовского канала. При большой мощности кода и высокой скорости передачи информации схема становится сложной, что является главным её недостатком.

Линейный код позволяет упростить процесс декодирования, если используются свойства группы и *смежного класса*. Пусть $\mathbf{g}$ – подгруппа, $\mathbf{g} \in \mathbf{G}$, элемент $h \in \mathbf{G}$, но $h \notin \mathbf{g}$, тогда $h * \mathbf{g}$ - смежный класс относительно операции *. Элемент минимального веса в смежном классе называется лидером смежного класса.

Табличная схема декодирования по смежным классам. Пусть над полем $GF(q)$ задан линейный (n,k,d)-код, который содержит кодовые слова $C = \left\{ \mathbf{0}, \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{q^k-1} \right\}$ и является подгруппой множества R^n, а также $\mathbf{e}$ - вектор ошибки, $\mathbf{e} \in R^n$, $\mathbf{e} \neq \mathbf{c_j}$. Если принять вектор $\mathbf{y}_j = \mathbf{c}_j + \mathbf{e}_j$, то возможные значения вектора $\mathbf{c}_j$ лежат в одном смежном классе, лидером которого является вектор ошибок $\mathbf{e}_j$. Наиболее вероятным вектором ошибок является вектор $\mathbf{e}_j$, имеющий минимальный вес среди всех векторов смежного класса относительно $\mathbf{y}_j$. Тогда кодовое слово декодируется как $\mathbf{c}_j = \mathbf{y}_j - \mathbf{e}_j$. Таблица декодирования через смежный класс $\mathbf{e}_j * \mathbf{c}_j$ строится по форме, показанной на рис.3.2. В таблице черта разделяет две области декодирования. Выше черты лежит область, в которой можно гарантированно исправить заданное количество ошибок. Каждая строка таблицы образует смежный класс, при этом векторы $\mathbf{e}_1, ..., \mathbf{e}_M$ являются лидерами смежных классов. Элементы столбцов (кроме первого) таблицы до черты образуют сферы гарантированного декодирования. Элементы столбцов после черты соответствуют областям между сферами гарантированного декодирования, тяготеющими к некоторому кодовому слову $\mathbf{c}_i$. Все элементы тех или иных столбцов определяют сферы полного декодирования. Соответственно различают полный и неполный декодеры. *Полный декодер* работает со сферами полного декодирования и ставит в соответствие принятому вектору $\mathbf{y}$ ближайшее кодовое слово из области декодирования. *Неполный декодер* работает со сферами гарантированного декодирования и ставит в соответствие принятому вектору $\mathbf{y}$ ближайшее кодовое слово из сферы гарантированного декодирования.

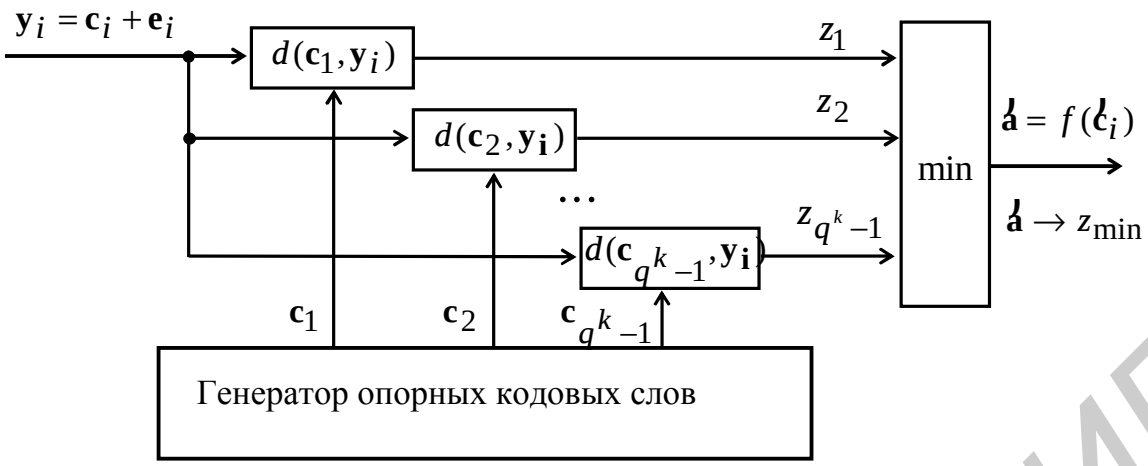
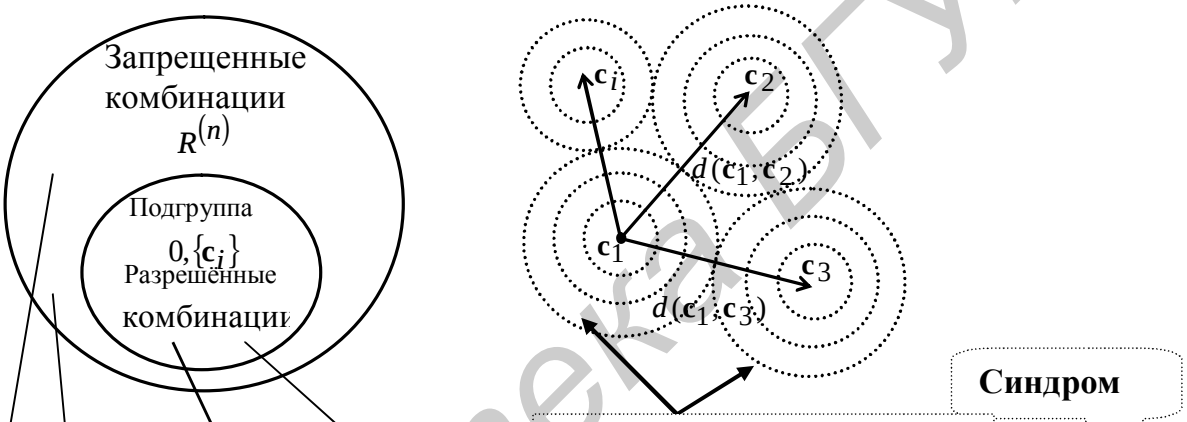


Рис. 3.1



	0	$\mathbf{c}_1$	$\mathbf{c}_2$	…	$\mathbf{c}_{q^k-1}$	**S**
$\mathbf{e}_1$	$\mathbf{e}_1 + \mathbf{0}$	$\mathbf{e}_1 + \mathbf{c}_1$	$\mathbf{e}_1 + \mathbf{c}_2$	…	$\mathbf{e}_1 + \mathbf{c}_{q^k-1}$	$\mathbf{e}_1 \cdot \mathbf{H}^T$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{0}$	$\mathbf{e}_2 + \mathbf{c}_1$	$\mathbf{e}_2 + \mathbf{c}_2$	…	$\mathbf{e}_2 + \mathbf{c}_{q^k-1}$	$\mathbf{e}_2 \cdot \mathbf{H}^T$
…	…	…	…	**Смежный класс**		…
$\mathbf{e}_j$	$\mathbf{e}_j + \mathbf{0}$	$\mathbf{e}_j + \mathbf{c}_1$	$\mathbf{e}_1 + \mathbf{c}_2$	…	$\mathbf{e}_j + \mathbf{c}_{q^k-1}$	$\mathbf{e}_j \cdot \mathbf{H}^T$
$\mathbf{e}_{j+1}$	$\mathbf{e}_{j+1} + \mathbf{0}$	$\mathbf{e}_{j+1} + \mathbf{c}_1$	$\mathbf{e}_{j+1} + \mathbf{c}_2$	…	$\mathbf{e}_{j+1} + \mathbf{c}_{q^k-1}$	$\mathbf{e}_{j+1} \cdot \mathbf{H}^T$

Рис. 3.2

Синдромное декодирование по лидеру смежного класса. Параметры смежного класса, соответствующего вектору **y**, можно определить с помощью синдрома, который вычисляется как

$$\mathbf{s}_j = \mathbf{y}_j \cdot \mathbf{H}^T = (s_{0,j},...,s_{n-k-1,j}), \tag{3.9}$$

где $\mathbf{y}_j = \mathbf{c} + \mathbf{e}_j$ вектор-строка принятого сигнала, представляющего собой аддитивную смесь слова **c** корректирующего кода и вектора ошибки $\mathbf{e}_j$.

Основные свойства синдрома:

- Все векторы из одного смежного класса имеют одинаковый синдром.

Доказательство: если $\mathbf{y}_{i,j} = \mathbf{c}_i + \mathbf{e}_j$, тогда $\mathbf{y}_{i,j}\mathbf{H}^T = \mathbf{c}_i\mathbf{H}^T + \mathbf{e}_j\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T = \mathbf{s}_j$.

- Синдром зависит от характера канала (вектора ошибки) и фильтрующих свойств кода (проверочной матрицы).

- Принятый вектор не содержит ошибок, если синдром равен нулю.

Алгоритм декодирования по лидеру смежного класса (рис.3.3).

Входные данные: линейный (n,k) код $C \subseteq \mathrm{F}_q^n$, **y** –принимаемый вектор.

Выходные данные: оценка принимаемого кодового слова.

1. Вычисляется синдром **s**($\mathbf{y}_j$) и по его форме определяется соответствующий ему лидер смежного класса $\mathbf{e}_j$.

2. Оценка кодового слова вычисляется как $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}_j$, где $\hat{\mathbf{c}}$ – кодовое слово, находящееся на минимальном расстоянии от **y**.

Пример. Рассмотрим линейный групповой код (5,2):

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \; d_{\min} = 3, \text{ т.е. } t_{испр} = 1.$$

В канале возможно возникновение не более двух ошибок. Таблица декодирования возможных комбинаций **y = c** + **e** имеет следующий вид (табл.3.1).

Пусть $\mathbf{c} = \mathbf{c}_2 = 10101$, $\mathbf{e}_2 = 00010$. Принимаемое слово имеет вид

$$\mathbf{y} = \mathbf{c}_2 + \mathbf{e}_2 = \oplus \begin{matrix} 10101 \\ 00010 \end{matrix} = 10111.$$

Вычисленный синдром равен $\mathbf{s}_2$ =(0,1,0). Из таблицы декодирования находим, что этому коду синдрома соответствует оценка вектора ошибки $\hat{\mathbf{e}} = (0,0,0,1,0)$. Тогда принятое кодовое слово находится как $\hat{\mathbf{c}} = \mathbf{y} + \hat{\mathbf{e}} = (1,0,1,0,1)$.

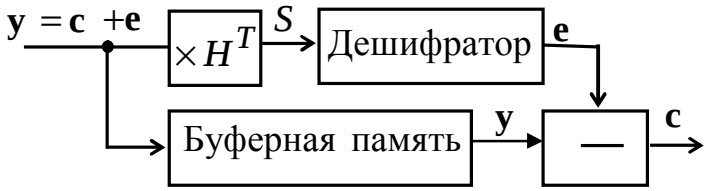


Рис. 3.3

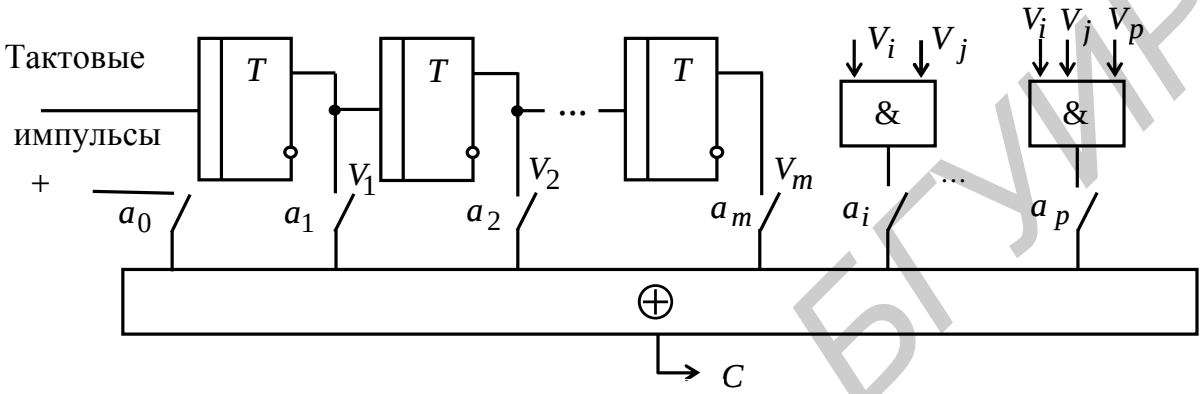


Рис. 3.4

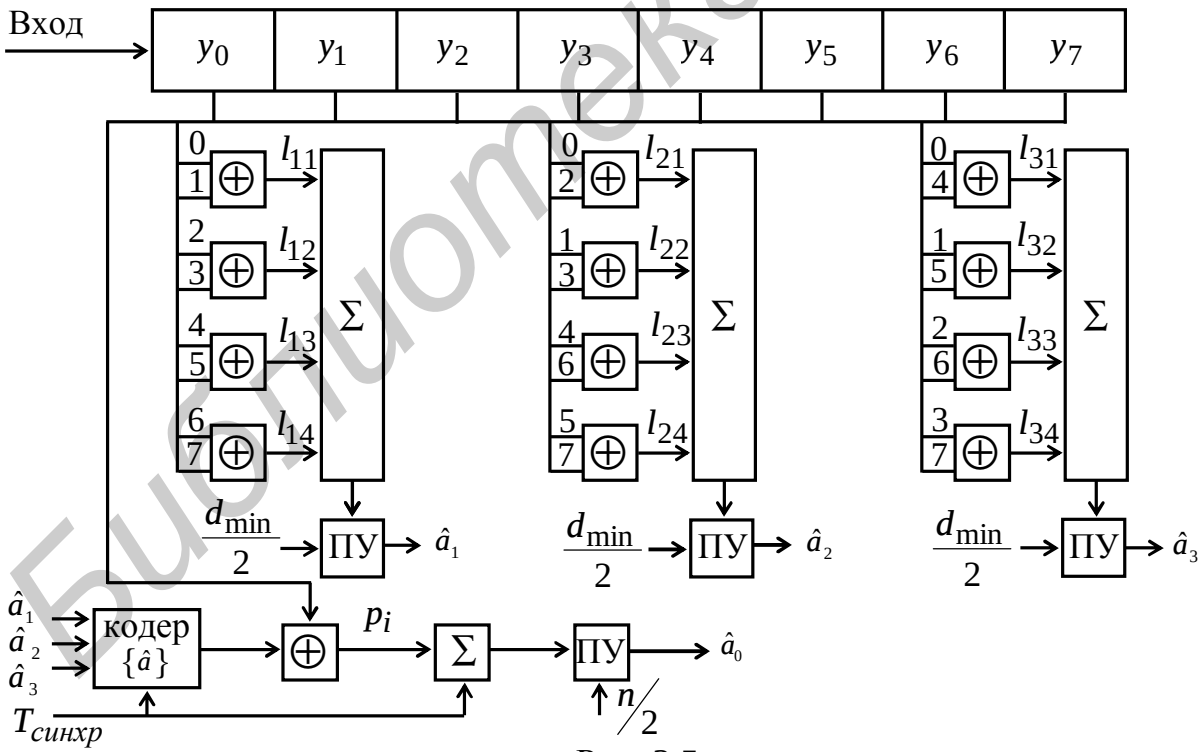


Рис. 3.5

Таблица 3.1

	$\mathbf{0} = 00000$	$\mathbf{c}_1 = 01011$	$\mathbf{c}_2 = 10101$	$\mathbf{c}_3 = 11110$	$\mathbf{s_0} = 000$
$\mathbf{e}_1 = 00001$	00001	01010	10100	11111	$\mathbf{s}_1 = 001$
$\mathbf{e}_2 = 00010$	00010	01001	10111	11100	$\mathbf{s}_2 = 010$
$\mathbf{e}_3 = 00100$	00100	01111	10001	11010	$\mathbf{s}_3 = 100$
$\mathbf{e}_4 = 01000$	01000	00011	11101	10110	$\mathbf{s}_4 = 011$
$\mathbf{e}_5 = 10000$	10000	11011	00101	01110	$\mathbf{s}_5 = 101$
$\mathbf{e}_6 = 01100$	01100	00111	11001	10010	$\mathbf{s}_6 = 111$
$\mathbf{e}_7 = 11000$	11000	10011	01101	00110	$\mathbf{s}_7 = 110$

Заметим, что если декодер работает ниже гарантированной области, то может возникнуть неоднозначность соответствия синдрома разным смежным классам, что приведёт к необратимой ошибки.

В случае линейных кодов большой мощности количество смежных классов резко возрастает. Так, например, линейный (50,20)-код над полем F_2 имеет около 10^9 смежных классов. Возникает задача минимизации вычислительной сложности декодирования.

3.4. Распределения весов линейного кода

Пусть задан линейный (n,k)-код, символы которого принадлежат q-ичному множеству из $GF(q)$, и пусть этот код содержит A_i векторов веса i.

Тогда совокупность чисел $A_0,...,A_i,...,A_n$ называют распределением весов, а многочлен от двух переменных x, y над полем комплексных чисел $A(x,y) = \sum_{i=0}^{n} A_i \cdot x^i y^{n-i}$ называется *нумератором весов*, или *весовой функцией кода*. Иногда в весовых функциях делают подстановку $x = z$, $y = 1$ и используют многочлены типа $A(z) = \sum_{i=0}^{n} A_i \cdot z^i$.

Предположим, что код C характеризуется весом d и в ДСК символ искажается с

вероятностью p_0. Определим отказ от декодирования в гарантированной области как состояние декодера с вероятность появления этого состояния P_{df}. Примем, что декодер с вероятностью P_{dS} неправильно исправляет ошибки, а вероятность правильного декодирования равна P_{dC}. Очевидно, что $P_{df} + P_{dS} + P_{dC} = 1$.

Используя понятие нумератора веса, вероятность P_{df} можно трактовать как вероятность того, что вектор ошибки является одним из ненулевых кодовых векторов. Тогда

$$P_{df} = 1 - (1 - p_0^n) \cdot A\left(\frac{p_0}{1 - p_0}\right); \quad P_{dC} = \sum_{i=0}^{t} C_n^i \cdot p_0^i (1 - p_0)^{n-i}, \qquad (3.10)$$

где $A\left(\frac{p_0}{1 - p_0}\right)$ нумератор веса.

Определим $N_{i,j}^{(i)}$ как число векторов **u** веса w, находящихся на расстоянии j от другого вектора **v**, имеющего вес i. Тогда вероятность того, что вектор ошибки принадлежит смежному классу с минимальным весом t и менее, равна

$$P_{см.кл} = \sum_{w=0}^{n} \sum_{j=0}^{t} \sum_{i=0}^{n} A_i \cdot N_{j,w}^{(t)} \cdot p_0^w (1 - p_0)^{n-w} . \qquad (3.11)$$

Нумератор веса является важнейшей характеристикой кода. Он позволяет более точно оценить вероятность ошибки и корректирующую способность кода.

Оценку нумератора веса кода можно получить через нумератор веса двойственного ему кода, которую в ряде случаев проще вычислить. Если C – линейный q-ичный код (n, k), а $C^{\perp}$ - это двойственный к линейному код, причём эти коды имеют следующие нумераторы веса: $C \to A(z)$ и $C^{\perp} \to B(z)$, тогда

$$A(z) = q^{k-n}(1 + (q-1)z)^n \cdot B\left(\frac{1 - z}{1 + (q-1)z}\right). \qquad (3.12)$$

3.5. Коды Хемминга

Кодом Хемминга называется (n, k)-код, проверочная матрица которого имеет r = =(n – k) строк и (2^r – 1) столбцов, причем столбцами являются все различные ненулевые последовательности, представляющие собой двоичную запись чисел 1,2,…,2^r – 1.

Пример. Для (7,4)-кода Хэмминга

$$\mathbf{H}_{(7,4)} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \qquad (3.13)$$

Распределение и нумератор весов кода соответственно равны

$$[1, 0, 0. 7, 7, 0, 0, 1] \text{ и } y^7 + 7x^3y^4 + 7x^4y^3 + x^7.$$

Проверочная матрица любого кода Хэмминга всегда содержит минимум три линейно зависимых столбца, поэтому кодовое расстояние кода равно трем. Если столбцы проверочной матрицы представляют упорядоченную запись десятичных чисел, т.е. 1,2,3... в двоичной форме, то вычисленный синдром

$$\mathbf{s}_i = [s_0, \ldots, s_{r-1}] = \mathbf{y}_i \mathbf{H}^T \qquad (3.14)$$

однозначно указывает на номер позиции искаженного символа.

Пример. Для (7,4)-кода Хемминга проверочная матрица в упорядоченном виде имеет вид

$$\mathbf{H}_{(7,4)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (3.15)$$

Пусть переданное кодовое слово **c** = [1,1,0,1,0,0,1], а принятое слово – $\mathbf{y}_5$ = =[1,1,0,1,1,0,1]. Синдром, соответствующий принятому сигналу, равен

$$\mathbf{s}_5 = [1101101] \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}^T = [101].$$

Вычисленный синдром указывает на ошибку в пятой позиции.

Корректирующая способность кода Хемминга может быть увеличена введением дополнительной проверки на четность. Так, проверочная матрица для расширенного (8,4)-кода будет иметь вид

$$\mathbf{H}_{(8,4)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad (3.16)$$

при этом кодовое расстояние стало равным d = 4.

Коды Хемминга можно также определить и для не бинарного случая, т.е. над произвольным конечным полем. В этом случае проверочная матрица **H** имеет размер $(m \times [q^m - 1)/(q - 1)]$ и столбцы матрицы попарно линейно независимы. Такая матрица

определяет линейный код ((q^m – 1)/(q – 1), (q^m – 1)/(q – 1) – m) – код с минимальным расстоянием, равным 3.

Пример. Код Хемминга (13, 10), q = 3 имеет следующую проверочную матрицу:

$$\mathbf{H} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{1} & \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{1} & \mathbf{2} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

3.6. Коды Рида - Маллера

Коды Рида – Маллера (РМ) представляют собой класс линейных кодов над $GF(2)$ с простым описанием и декодированием. Для любых целых m и l, $l < m$ существует код РМ длиной 2^m, который называется кодом РМ l–го порядка. Определим этот код через порождающую матрицу, которую будем строить в удобной для декодирования несистематической форме.

Порождающая матрица кода РМ l-го порядка длиной 2^m определяется как совокупность блоков $\mathbf{G} = [G_0 \quad G_1 \quad \mathbf{L} \quad G_l]^T$, где $\mathbf{G}_0$ – вектор размерностью $n = 2^m$, состоящий из одних единиц; $\mathbf{G}_1$- матрица размером ($m \times 2^m$), содержащая в качестве столбцов все двоичные коды из m бит; строки матрицы $\mathbf{G}_l$ получаются как все возможные произведения l строк матрицы $\mathbf{G}_1$. Для определенности будем считать, что первый столбец в $\mathbf{G}_1$ состоит из одних нулей, последний – из одних единиц, а остальные – коды чисел 1,2,…, упорядоченных по возрастанию, считая, что младший бит расположен в нижней строке.

Поскольку существует всего $\binom{m}{j}$ способов выбора j строк, входящих в произведение, то матрица $\mathbf{G}_j$ имеет размер $\binom{m}{j} \times 2^m$. Ясно, что для кода РМ - l

$$k = 1 + \binom{m}{1} + \ldots + \binom{m}{l}, \qquad n - k = 1 + \binom{m}{1} + \ldots + \binom{m}{m-l-1}, \qquad (3.17)$$

что обеспечивается линейной независимостью строк в матрице $\mathbf{G}$.

Код Рида – Маллера l-го порядка длиной $n = 2^m$ представляет собой бинарный код с параметрами $(n,k) = (2^m, \sum_{i=0}^{l} C_m^i)$. Код РМ-1 является двойственным коду Хемминга, для него $d_{\min} = 2^{m-1}$. Код РМ-2 имеет $d_{\min} = 2^{m-1} \pm 2^{m-1-n}$, где $1 \le h \le [m/2]$.

Пример. Для кода РМ-1 $m=3$ порождающая матрица **G** имеет вид

$$\mathbf{G}=\begin{bmatrix}11111111\\01010101\\00110011\\00001111\end{bmatrix}.$$

Исходя из понятия базисного вектора, можно записать следующее определение. Кодом РМ порядка l называют код, базисом которого являются все векторы $\mathbf{v}_0,\mathbf{v}_1,...,\mathbf{v}_m$ и все их произведения из l и меньшего числа векторов. Так, для $l=1$ $\mathbf{G}=[\mathbf{v}_0,\mathbf{v}_1,...,\mathbf{v}_m]^T$ и для $l=2$ $\mathbf{G}=[\mathbf{v}_0,\mathbf{v}_1,...,\mathbf{v}_m,\mathbf{v}_1\mathbf{v}_2,\mathbf{v}_1\mathbf{v}_3,...,\mathbf{v}_i\mathbf{v}_j]^T$

*Пример***.** Построим матрицу **G** для $n=8$ и $l=2$.

$$\mathbf{G}=\begin{bmatrix}\mathbf{v}_0=11111111\\\mathbf{v}_1=01010101\\\mathbf{v}_2=00110011\\\mathbf{v}_3=00001111\\\mathbf{v}_1\cdot\mathbf{v}_2=00010001\\\mathbf{v}_1\cdot\mathbf{v}_3=00000101\\\mathbf{v}_2\cdot\mathbf{v}_3=00000011\end{bmatrix}\Rightarrow\quad\text{код}\,(8,7).$$

Схема кодера, осуществляющего кодирование кодом РМ-2, приведена на рис. 3.4.

Код РМ позволяет просто вычислить вес кодовых слов, и, следовательно, определить кодовое расстояние. Общая формула распределения веса имеет вид

$$A_{2^{m-1}}=2^{\left(m^2+m+2\right)}-2-2\cdot\sum_{h=1}^{\left\lfloor\frac{m}{2}\right\rfloor}2^{h(h+1)}\cdot\frac{\prod_{i=m-2h+1}^{m}(2^i-1)}{\prod_{i=1}^{h}(2^{2i}-1)}\ . \qquad (3.18)$$

Пример. Для кода РМ-1: $m=4$, $l=1$. $n=2^4=16$; $d_{\min}=2^{4-1}=8$; $k=\sum_{i=0}^{l}C_m^i=5$, скорость кода $R=\frac{k}{n}=\frac{5}{16}<0{,}5$.

Алгоритм мажоритарного декодирование кода РМ. Рассмотрим метод мажоритарного декодирования кода РМ по большинству голосов на конкретном примере. Пусть $n=2^3=8$, $m=3$, $l=1$, $k=4$. Информационный вектор-строка $\mathbf{a}=[a_0,a_1,...,a_{k-1}]$ и порождающая матрица $\mathbf{G}_{PM}$ задают код

$$C = \mathbf{a} \cdot \mathbf{G}_{PM} = a_0\mathbf{v}_0 + a_1\mathbf{v}_1 + ... + a_{k-1}\mathbf{v}_m . \qquad (3.19)$$

$$\mathbf{G} = \begin{bmatrix} 11111111 \\ 01010101 \\ 00110011 \\ 00001111 \end{bmatrix} \Rightarrow C = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 11111111 \\ 01010101 \\ 00110011 \\ 00001111 \end{bmatrix} = \begin{bmatrix} c_0, c_1, c_2, ..., c_7 \end{bmatrix}.$$

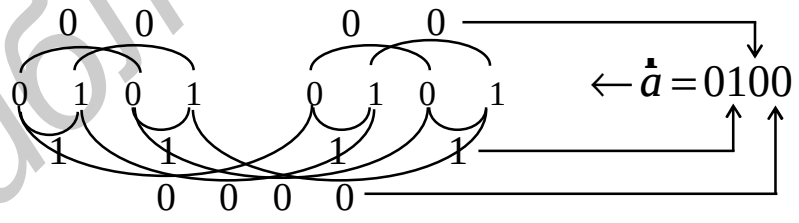
Можно построить проверочные соотношения, связывающие информационный символ с символами кодового слова. Эти соотношения имеют вид:

$c_0 = a_0$	$a_1 = c_1 + c_0 = l_{11}$	$a_3 = c_0 + c_4 = l_{31}$
$c_1 = a_0 + a_1$	$a_1 = c_2 + c_3 = l_{12}$	$a_3 = c_1 + c_5 = l_{32}$
$c_2 = a_0 + a_2$	$a_1 = c_4 + c_3 = l_{13}$	$a_3 = c_2 + c_6 = l_{33}$
$c_3 = a_0 + a_1 + a_2$	$a_1 = c_6 + c_7 = l_{14}$	$a_3 = c_3 + c_7 = l_{34}$
$c_4 = a_0 + a_3$	$a_2 = c_0 + c_4 = l_{21}$	
$c_5 = a_0 + a_1 + a_3$	$a_2 = c_1 + c_3 = l_{22}$	
$c_6 = a_0 + a_2 + a_3$	$a_2 = c_4 + c_6 = l_{23}$	
$c_7 = a_0 + a_1 + a_2 + a_3$	$a_2 = c_5 + c_7 = l_{34}$	

Правило принятия решений для декодирования по большинству голосов:

Если $0 < \sum_{j=1}^{n/2} l_{ij} \le \frac{d_{\min}}{2}$, то $a_i = 0$. Если $\frac{d_{\min}}{2} < \sum_{j=1}^{n/2} l_{ij} \le d_{\min}$, то $a_i = 1$.

Пример. Принят вектор **y**=[0,1,0,1,0,10,1]. Процедура декодирования дает следующий результат:



Рассмотрим декодирование нулевого информационного символа. Пусть $\vec{V}_0 = 1,...1$, тогда $C = a_0\vec{V}_0 + a_1\vec{V}_1 + ... + a_m\vec{V}_m$, $a_0\vec{V}_0 = C + a_1\vec{V}_1 + ... + a_m\vec{V}_m$.

Определяя проверочное соотношение как **p** = (p_1, …, p_n) = $C + \hat{a}_1\vec{V}_1 + ... + \hat{a}_m\vec{V}_m$, получаем правило декодирования нулевого символа:

$$\text{если } \sum_{i=1}^{n} p_i \geq \frac{n}{2}, \text{ то } a_0 = 1; \text{ если } \sum_{i=1}^{n} p_i < \frac{n}{2}, \text{ то } a_0 = 0. \qquad (3.20)$$

Схема декодирования кода РМ -1 приведена на рис. 3.5.

Схему декодирования кодов РМ высших порядков можно получить по индукции. Предположим, что информационная последовательность разбита на сегменты следующим образом: каждому сегменту соответствует один из l блоков порождающей матрицы, который при кодировании умножается на этот сегмент. Если мы сможем восстановить информационные биты в l-м сегменте, то затем сможем вычислить их вклад в принятое слово и вычесть его из принятого слова. Тогда задача сводится к декодированию кода РМ меньшего (l - 1) - го порядка. Процедура декодирования представляет собой последовательность мажоритарных процедур и начинается с нахождения мажоритарным методом информационных символов в сегменте с номером l.

3.7. Границы линейных кодов

Для системного анализа используются граничные соотношения, позволяющие оценить корректирующие и потенциальные корректирующие способности кода.

Граница случайного кодирования. Определяется *теоремой Шеннона – Галлагера*, в которой утверждается,что если задан дискретный канал без памяти с входным алфавитом из k символов $(u_1, u_2, ..., u_k)$ и выходным алфавитом из j символов $(v_1, v_2, ..., v_k)$ и переходными вероятностями $P_{jk} = P_r(v_j | u_k)$, тогда для любой длины кода n любого числа кодовых слов $M = e^{nR}$ и любого распределении вероятностей на множестве этих кодовых слов существует код длиной n, вероятность ошибки для которого в указаном выше канале оценивается сверху следующим образом:

$$P_e \leq \exp\left[-n\left\{-nR + E_0(n, \vec{Q})\right\}\right], \qquad (3.21)$$

$$E_0(n, \vec{Q}) = -\ln \sum_{j=1}^{J} \left(\sum_{k=1}^{k} q_k \cdot p_{jk}^{\frac{1}{1+n}} \right)^{1+n}. \qquad (3.22)$$

Здесь n - произвольное число из интервала $0 \leq n \leq 1$; $\vec{Q} = [q_1, ..., q_k]$ - произвольный вероятностный вектор, для которого $q_i > 0$ и $\sum_{i=1}^{k} q_i = 1$.

Нижняя граница Варшамова – Гильберта гарантирует существование линейного

кода с заданным числом $r = n - k$ проверочных символов и заданным $d_{\min}$, число информационных символов которого больше определённой величины.

Пусть $q = p^m$ - степень простого числа p, а r и d – некоторые целые положительные числа, тогда существует q-ичный линейный код длиной n с r проверочными символами и кодовым расстоянием не меньше d, параметры которого удовлетворяют следующему неравенству:

$$q^r = \sum_{i=0}^{d-2} C_n^i \cdot (q-i)^i \ . \tag{3.23}$$

Используя функцию Чернова, можно записать следующее неравенство:

$$\frac{r}{n} \le j\left(\frac{d_{\min} - 2}{n}\right), \tag{3.24}$$

где $j(x) = x \cdot \log_q(q-1) - x \cdot \log_q x - (1-x) \cdot \log_q(1-x)$ - функция Чернова.

Верхняя граница Хемминга показывает, что если существует q-ичный линейный код длиной n со скоростью передачи информации $R = \frac{n}{k}$ и минимальным расстоянием Хемминга $d_{\min} = 2t + 1$ или более, то

$$\sum_{i=0}^{t} C_n^i (q-1)^i \le q^{n(1-R)} \ . \tag{3.25}$$

Двоичный код имеет $q = 2$ и для него справедливо неравенство $\sum_{i=0}^{t} C_n^i \le 2^{n-k}$.

Если граница Хемминга выполняется со знаком равенства, то код называется *совершенным*, т.е. сферы радиусами t, центрами которых являются кодовые слова, попарно не пересекаются и наиболее плотно заполняют всё кодовое пространство. Для совершенного кода, рассматриваемого как смежный класс, в качестве лидеров остальных смежных классов (количеством $(q^{n-k} - 1)$) удаётся взять все векторы весом t и только их.

Пример. Пусть $n = 12, d = 5, t = 2, q = 2$. Граница Хемминга дает следующий результат: $\sum_{i=0}^{2} C_{12}^i \le 2^{12-k} \Rightarrow 2^k \le \frac{2^n}{\sum_{i=0}^{2} C_{12}^i} \Rightarrow k = 5$. Таким образом, возможно

построение кода с параметрами (n, k) = (12,5).

Верхнюю границу Хемминга, используя функцию Чернова, можно записать в следующем виде:

$$R \le 1 - j\left(\frac{t}{n}\right);\ R \le 1 - \frac{1}{n}\log_q\left[\sum_{i=0}^{t} C_n^i (q-1)^i\right]. \qquad (3.26)$$

Для низкоскоростных кодов более точной является *граница Плоткина*, суть которой в том, что если существует q-ичный блоковый код длиной n с общим числом кодовых слов k и минимальным расстоянием $d_{\min}$, то

$$d_{\min} \le \frac{(q-1)\cdot n \cdot k}{q\cdot(k-1)}. \qquad (3.27)$$

Для $q = 2$ получаем $R \le 1 - \frac{2d_{\min}}{n} + \frac{2}{n} + \frac{1}{n}\log_2 d_{\min}$.

Графическое изображение границ показано на рис. 3.6.

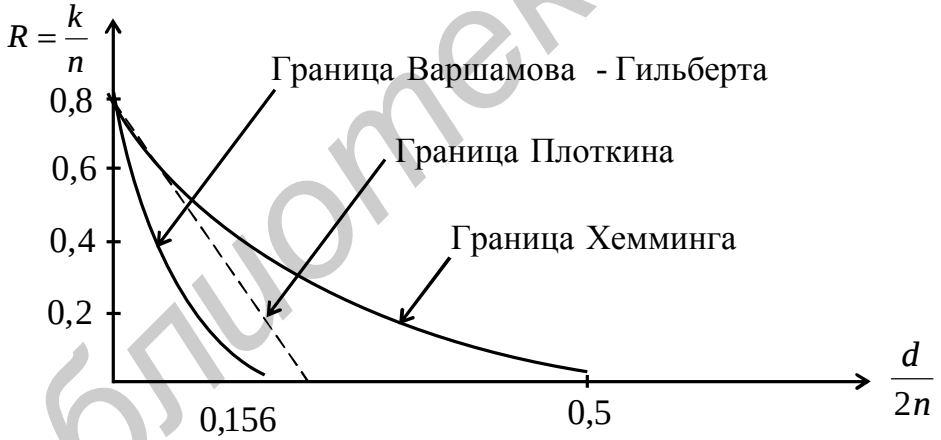


Рис.3.6

Пример. Пусть дан код $(n,k) = (63,51)$. Находим границу Хемминга:

$$\sum_{i=0}^{2} C_{63}^i \le 2^{n-k};\ 2017 \le 2^{n-k};\ r = n - k = 11.$$

Граница Варшамова – Гильберта равна $\sum_{i=0}^{3} C_{62}^i \le 2^{n-k};\ 39775 > 2^{n-k};$

$r = n - k = 16$. Таким образом, $r = n - k = 12;\ 11 < 12 < 16$; т.е. данный код близок к границе Хемминга.

Граница Синглтона связывает кодовое расстояние и параметры кода. Если C является $(n, k, d)_q$-кодом, тогда $d \le n - k + 1$. Коды, которые лежат на границе, т.е. для которых $k=n-d+1$, называются кодами с максимальным расстоянием (MDS-код). Граница справедлива для любого алфавита и показывает, что для исправления t ошибок код должен иметь не менее $2t$ проверочных символа.

Контрольные вопросы и задачи

1. Описать таблицу декодирования в случае, если часть ошибок декодируется (исправляется), а другая часть обнаруживается.

2. Покажите путем расчетов, что минимальный вес большинства линейных кодов близок к величине, гарантируемой нижней границей Варшамова — Гильберта.

3. Порождающая матрица кода над GF(2) задается как

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \qquad (3.28)$$

Сформировать все кодовые слова и найти минимальное кодовое расстояние.

4. Найти порождающую и проверочную матрицу двойственного кода, если порождающая матрица исходного кода задается (3.28).

5. Код задается порождающей матрицей (3.28). Декодировать принятое слово 111001 по алгоритму максимального правдоподобия.

6. Сделав необходимый расчет, доказать возможность существования совершенного $(q + 1, q – 1)$-кода над GF(q), исправляющего одиночные ошибки.

7. Найти порождающую матрицу кода Хемминга (13, 10), *GF*(3). Сформировать кодовое слово, соответствующее информационному вектору **a** = (1000121000).

4. ЦИКЛИЧЕСКИЕ КОДЫ

4.1. Основные понятия циклического кода

Циклическим является линейный код, который не изменяет своих свойств при циклической перестановке компонентов кодовых слов. Почти все важнейшие коды являются циклическими.

Пусть над полем $GF(q)$ задан код $\mathbf{C}$ с параметрами (n,k), кодовое слово которого задается в виде вектора-строки $\mathbf{c}=(c_0,c_1...,c_{n-1})$.

Если коду $\mathbf{C}$ вместе с кодовым словом $\mathbf{c}$ принадлежат его циклические сдвиги $(c_{n-1},c_0,c_1...,c_{n-2})$, то код называется циклическим.

Циклические коды удобно представлять в виде полинома

$$c(x)=c_0+c_1x+c_2x^2+...+c_{n-1}x^{n-1},$$

где x - формальная переменная, а полином $c(x)$ имеет коэффициенты, которые принадлежат полю $c_i\in GF(q)$.

Определим кольцо полиномов F[x] над полем F по модулю полинома $f(x)$. Зададим в этом кольце два полинома $g(x)$ и $h(x)$, сравнимых по модулю $f(x)$. В этом случае кольцо F[x] разбивается на классы эквивалентности:

$$[g(x)] = \{h(x) \mid h(x) = (g(x) \bmod f(x)) = R_{f(x)}(g(x))\}.$$

Множество классов эквивалентности обозначим как

$$F[x] / (f(x)=\{[g(x)] \mid g(x) \in F[x]\}.$$

Операции сложения и умножения задаются в соответствии с правилами

$$[g(x)] + [h(x)]=[g(x) + h(x)],\ [g(x)]\cdot[h(x)]=[g(x) \cdot h(x)].$$

Классы эквивалентности описывают все полиномы в F[x] степени, меньшей $f(x)$, соответствующие всевозможным остаткам $R_{f(x)}(\ )$от деления на f(x).

Пример. Кольцо $Z_2[x] / f(x)$, $f(x) = x^3 + 1$ включает полиномы

$$Z_2[x] / f(x)=\{[0], [1], [x], [1 + x], [x^2], [1 + x^2], [x + x^2], [1 + x + x^2]\}.$$

Операция сложения:

$$[x] + [1 + x + x^2]=[1 + x^2].$$

Операция умножения:

$$[1 + x^2] + [1 + x + x^2]=[1 + x + x^3 + x^4]=[0],\ \text{т.к. } 1 + x + x^3 + x^4 = 0 \bmod f(\mathbf{x}).$$

Идеалом кольца <R, +, ·> называется непустое подмножество I$\in$ R, такое, что:

- множество <I, +> – является группой;
- $i\cdot r \in$ I для любых $i\in$ I и всех $r\in$ R.

Конструкцию идеала просто получить следующим образом:

$$I=\{g(x) \cdot r(x) \mid r(x) \in R\}.$$

Пример. Зададим кольцо $Z_2[x]$ /$f(x)$, $f(x) = x^6 + 1$. Определим $g(x) = 1 + x^2 + x^4$. Идеал имеет вид

$$I = \{0,\ 1 + x^2 + x^4,\ x + x^3 + x^5,\ 1 + x + x^2 + x^3 + x^4 + x^5\}.$$

Определим векторное пространство V(n,q). Вектор (a_0,a_1,…,a_{n-1}) в пространстве задается в виде полинома

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} \in F[x]/(x^n - 1),$$

где F=GF(q). Выбор модуля в виде полинома (x^n - 1) позволяет поставить в соответствии операции умножения на x циклический сдвиг вектора.

Определение. Код **C** является циклическим, если и только если **C** – идеал.

Если **C** является циклическим кодом и $g(x)$ представляет собой нормированный многочлен наименьшей степени в **C**, тогда $g(x)$ уникален и каждое кодовое слово формируется в результате умножения на g(x).

Полином $g(x)$ называется генераторным или порождающим полиномом кода **C**.

Генераторный полином циклического кода делит (x^n - 1) и любой делитель (x^n - 1) является генераторным полиномом циклического кода.

Пример. Пусть задано пространство V(7, 2) и $f(x) = x^7 - 1$. Полином $f(x)$ факторизуется над полем GF(2) следующим образом:

$$x^7 - 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1).$$

Нормированные делители $f(x)$ имеют вид

$$g_1(x) = 1;\ g_2(x) = x + 1;\ g_3(x) = x^3 + x^2 + 1;\ g_4(x) = x^3 + x + 1\ g_5(x) = (x + 1)(x^3 + x^2 + 1);$$

$$g_6(x) = (x + 1)(x^3 + x + 1);\ g_7(x) = (x^3 + x^2 + 1)(x^3 + x + 1);\ g_8(x) = f(x).$$

Полином $g_1(x)$ генерирует все элементы пространства V(7, 2), в то время как полином $g_8(x)$ – тривиальное циклическое подпространство {(0000000)}. Полином $g_6(x)$ генерирует циклический код
{(0000000),(1011100),(0101110),(0010111),(1001011),(1100101),(1110010),(0111001)}.

Полином $g_7(x)$ формирует циклический код {(0000000), (1111111)}. Пространство V(7,2) состоит из 8 циклических кодов.

4.2. Порождающие и проверочные матрицы циклического кода

По аналогии с представлением линейного кода циклический код можно описать с помощью порождающей матрицы.

Теорема. Если генераторный полином $g(x)$ кода **C** имеет степень (n - k), тогда код **C** является циклическим (n, k)- кодом. Если $g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_{n-k} x^{n-k}$, тогда генераторная матрица кода представляется как матрица размером ($k \times n$) следующего вида:

$$\mathbf{G} = \begin{bmatrix} g(x) \\ x \cdot g(x) \\ x^2 \cdot g(x) \\ \mathbf{L\,L\,L} \\ x^{k-1} \cdot g(x) \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & \dots & \dots & g_{n-k} & \dots & 0 \\ 0 & 0 & g_0 & g_1 & \dots & g_{n-k} & \dots \\ \mathbf{L} & \mathbf{L} & \mathbf{L} & \mathbf{L} & \mathbf{L} & \mathbf{L} & \mathbf{L} \\ 0 & 0 & 0 & \dots & g_0 & \dots & g_{n-k} \end{bmatrix}.$$

Множество $\{x^i g(x);\ 0 \le i \le k - 1\}$ можно рассматривать как базис размерностью k циклического кода **C**.

Предположим, что **C** линейный (n,k,d)-код с генераторным полиномом $g(x)$, тогда существует такой полином $h(x)$ степени k ($\deg(h(x)) = k$), что

$$g(x) \cdot h(x) = x^n - 1 \equiv 0 \bmod (x^n - 1).$$

Многочлен $h(x)$ называется проверочным полиномом, он определяется как

$$h(x) = \frac{x^n - 1}{g(x)}.$$

Полином $h(x)$ формирует циклический код $\mathbf{C}_э$, размерностью $(n - k)$. Определим два кодовых слова. Первое $c_1(x) = a_1(x)\, g(x)$ принадлежит коду **C**, а $c_2(x) = a_2(x)\, h(x)$ – коду $\mathbf{C}_э$. Тогда их произведение равно

$$c_1(x)\, c_2(x) = a_1(x)\, g(x)\, a_2(x)\, h(x) = a_1(x)\, a_2(x)\, f(x) = 0 \ (\mathrm{mod}\ f(x)),$$

где $f(x) = x^n - 1$.

Коды **C** и $\mathbf{C}_э$ называются эквивалентными.

Найдем условие существования двойственного кода. Определим два вектора

$$\mathbf{a} = (a_0, a_1, \dots, a_{n-1}) \in \mathbf{C},$$

$$\mathbf{b} = (b_0, b_1, \dots, b_{n-1}) \in \mathbf{C}_э.$$

Умножим два полинома $a(x)$ и $b(x)$ друг на друга:

$$a(x)b(x) = \Big(\sum_{i=0}^{n-1} a_i x^i\Big)\Big(\sum_{i=0}^{n-1} b_i x^i\Big) = \sum_{i=0}^{n-1} c_i x^i \ (\mathrm{mod}\, x^n - 1).$$

Коэффициенты произведения образуют вектор $\mathbf{c} = (c_0, \dots, c_{n-1})$.

Коэффициент c_0 равен

$$c_0 = a_0 b_0 + a_1 b_{n-1} + a_2 b_{n-1} + \dots + a_{n-1} b_1.$$

Этот результат можно получить перемножением двух векторов

$$c_0 = \mathbf{a}\, \mathbf{b}(1),$$

где вектор $\mathbf{b}(1)$ получается из **b** путем циклического его сдвига влево на одну позицию и прочтения элементов в обратном порядке. Аналогично любой другой коэффициент получается как

$$c_t = \mathbf{a}\, \mathbf{b}(t),$$

где вектор **b**(t) получается из **b** путем циклического сдвига его влево на t позиций и прочтения элементов в обратном порядке.

Пример. Пусть n = 3, **a**=(a_0, a_2 , a_3), **b**=(b_0, b_1, b_3), Модуль равен (x^3 – 1). Тогда

$$a(x)b(x) = (a_0 + a_1 x + a_2 x^2)\,(b_0 + b_1 x + b_2 x^2) =$$

$$= (a_0 b_0 + a_1 b_2 + a_2 b_1) + (a_0 b_1 + a_1 b_0 + a_3 b_2)\, x + (a_0\, b_2 + a_1 b_1 + a_3 b_0)\, x^2 .$$

Так как $a(x)\ b(x) = 0 \bmod x^n - 1$, то в произведении коэффициенты при каждой степени x должны быть равны нулю. Но это означает, что **ac** = 0. Иными словами, векторы **a** и **c** – ортогональны, при этом вектор **c** получается из вектора **b** в результате циклического сдвига и инверсного упорядочения элементов. Код $\mathbf{C}^{\perp}$, образованный векторами **c,** ортогонален относительно скалярного произведения и является двойственным коду **C**.

Вектор **b** формируется с помощью порождающей матрицы

$$\mathbf{G}_э = [h(x),\ x\,h(x),\ x^2 h(x),\ \ldots,\ x^{n-k-1} h(x)]^{\mathrm{T}}.$$

Если в матрице $\mathbf{G}_э$ переупорядочить столбцы в обратном, инверсном порядке, то получим порождающую матрицу кода $\mathbf{C}^{\perp}$, которая является в то же время проверочной матрицей для кода **C**.

Пример. Предположим, что требуется сформировать порождающую и проверочную матрицы для (7,4) двоичного циклического кода. В качестве порождающего возьмем полином $g(x)$=1 + x + x^3, $f(x)$ = x^7 - 1. Полином $g(x)$ делит $f(x)$ и, следовательно, формирует циклический код **C** (7,4). Проверочный полином равен

$$h(x) = f(x)/g(x) = 1 + x + x^2 + x^4.$$

Полином $h(x)$ формирует циклический код (7,3) $\mathbf{G}_э$. Порождающая матрица кода **C** равна

$$\mathbf{G} = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \\ x^3 g(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Порождающая матрица кода $\mathbf{C}_э$ равна

$$\mathbf{G}_э = \begin{bmatrix} h(x) \\ xh(x) \\ x^2 h(x) \\ x^3 h(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Прочитав столбцы матрицы $\mathbf{G}_э$ в обратном порядке, получаем генераторную матрицу двойственного кода $\mathbf{C}^{\perp}$:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Простой проверкой можно убедиться, что $\mathbf{GH}^{\mathrm{T}} = 0$. Поскольку коды $\mathbf{C}$` и $\mathbf{C}^{\perp}$ получаются один из другого простой перестановкой компонентов, то они эквивалентны.

Введем понятие *обратного полинома*. Пусть задан полином $h(x)=a_0 + a_1 x + \ldots + a_k x^k$. Обратный к нему полином запишется как

$$h_r(x) = \sum_{i=0}^{k} a_{k-i} x^i .$$

Можно заметить, что $h_r(x) = x^k\, h(1/x)$, где $k = \deg(h(x))$. Таким образом, если полином $g(x)$ степени $(n - k)$ делит полином $f(x) = x^n - 1$ и, следовательно, формирует циклический (n, k) код $\mathbf{C}$ с проверочным полиномом $h(x)$, то обратный полином $h_r(x)$ генерирует двойственный код $\mathbf{C}^{\perp}$.

Порождающая матрица систематического кода. Построим порождающую матрицу $\mathbf{G}$ циклического кода, в структуре которого явно выделены информационные символы. Форма матрицы в этом случае имеет вид $\mathbf{G} = [\, P \mid I_k \,]$. Положим, что код задается порождающим полиномом $g(x)$. Разделим x^{n-k+i} на $g(x)$ для $i = 0, 1, \ldots, k - 1$. Результат деления можно записать в виде

$$x^{n-k+i} = q_i(x)\, g(x) + r_i(x),$$

где $q_i(x)$ – целая часть, а $r_i(x)$ – остаток от деления на полином.

Произведение $q_i(x)g(x)$ представляет собой кодовое слово. Тогда разность

$$x^{n-k+i} - r_i(x) = q_i(x)\, g(x) \in \mathbf{C}$$

образует множество k линейно независимых кодовых слов. Матрица, составленная из этих кодовых слов, является порождающей.

Пример. Предположим, что бинарный циклический код (7,4) формируется с помощью порождающего полинома $g(x) = 1 + x + x^3$. Алгоритм деления полиномов дает следующие выражения:

$$x^3 + (1 + x) = (1)(x^3 + x + 1),$$
$$x^4 + (x + x^2) = (x)(x^3 + x + 1),$$
$$x^5 + (1 + x + x^2) = (x^2 + 1)(x^3 + x + 1),$$
$$x^6 + (1 + x^2) = (x^3 + x + 1)(x^3 + x + 1).$$

Порождающая матрица систематического кода имеет вид

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = [P \mid I_4].$$

Соответствующая проверочная матрица систематического кода получается из проверочной после преобразования

$$\mathbf{H} = \left\lfloor I_{n-k} \mathbb{M} - P^T \right\rfloor .$$

4.3. Свойства синдрома циклического кода

Пусть $y(x) = y_0 + y_1 x + y_2 x^2 + \ldots + y_{n-1} x^{n-1}$ и $s(x)$ будут соответственно полиномы принимаемого вектора-строки **y** и синдрома $\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathrm{T}}$. Покажем, что синдром можно получить в результате деления $y(x)$ на порождающий полином $g(x)$.

Заметим, что в матрице **H** первые $(n - k - 1)$ столбцов соответствуют полиному x^i , а остальные (n - k) столбцов – остаткам r_{j-n+k}, $j = (n - k), \ldots (n - 1)$. Таким образом, можно записать

$$s(x) = y_0 + y_1 x + \ldots + y_{n-k-1} x^{n-k-1} + y_{n-k} r(x) + \ldots + y_{n-1} r_{k-1}(x) =$$
$$= y_0 + y_1 x + \ldots + y_{n-k-1} x^{n-k-1} + y_{n-k} (x^{n-k} - q_0 g(x)) + \ldots + y_{n-1} (x^{n-1} - q_{k-1} g(x)) =$$
$$= y(x) - (y_{n-k} q_0 g(x) + \ldots + y_{n-1} q_{k-1} g(x)) = y(x) - p(x) g(x),$$

где $p(x) = y_{n-k} q_0 + y_{n-1} q_{k-1}$.

Следовательно,

$$y(x) = p(x) g(x) + s(x).$$

Поскольку $\deg(r_i(x)) \le n - k - 1$, то $\deg(s(x)) \le n - k - 1$. Учитывая единственность представления целой части и остатка в алгоритме деления полиномов, можно сделать вывод, что синдром $s(x)$ представляет собой остаток от деления принимаемого вектора $y(x)$ на порождающий полином $g(x)$.

Структура синдрома вектора зависит от его циклических сдвигов. Принимаемый вектор можно записать в виде

$$y(x) = q(x) g(x) + s(x),$$

где $\deg(s(x))$ больше, чем $(n - k - 1)$.

Циклический сдвиг вектора равен

$$x\, y(x) = x\, q(x) g(x) + x\, s(x).$$

Если $(x\, s(x))$ имеет степень более $(n - k - 1)$, тогда он является синдромом $(x\, y(x))$. В противном случае синдром полинома $(x\, y(x))$ получается как остаток от деления $(xs(x))$ на $g(x)$. Представим синдром в виде

$$s(x) = \sum_{i=0}^{n-k-1} s_i x^i = s`(x) + s_{n-k-1} x^{n-k-1}, \ \deg(s`(x)) \le n - k - 2 .$$

Нормированный порождающий полином представим в виде

$$g(x) = \sum_{i=0}^{n-k-1} g_i x^i = g`(x) + x^{n-k}, \ \deg(g`(x)) \le n - k - 1 .$$

Тогда

$$xs(x) = xs`(x) + s_{n-k-1}(g(x) - g`(x)) = s_{n-k-1} g(x) + (xs`(x) - s_{n-k-1} g`(x)),$$

где $\deg(xs`(x) - s_{n-k-1}g`(x)) \le n-k-1$. Учитывая единственность представления остатка в алгоритме деления, можно заключить, что синдром полинома ($x\ s(x)$) равен ($x\ s(x)$) – ($x\ s_{n-k-1}$) g(x).

Таким образом, в случае применения циклического (n, k)-кода **C**, задаваемого полиномом $g(x)$, синдром $s(x) = (s_0 + s_1\ x + s_2\ x^2 + \ldots + s_m\ x^m)$ сдвига ($x\ y(x)$) принимаемого вектора $y(x)$ определяется как:

1) ($x\ s(x)$), если $\deg(s(x)) < n = k - 1$;

2) ($x\ s(x) - x\ s_{n-k-1}\ g(x)$), если $\deg(s(x)) = n - k - 1$.

Пример. Возьмем исходные данные из предыдущего примера, $g(x) = 1 + x + x^3$. Проверочная матрица имеет вид

$$\mathbf{H} = \left[I_k \mathbb{M} - P^T \right] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Предположим, что принимается вектор **y**=(1011011). Синдром равен

$$\mathbf{s} = \mathbf{y}\mathbf{H}^T = (001).$$

Полиномиальная форма вектора **y** имеет вид

$$y(x) = 1 + x^2 + x^3 + x^5 + x^6.$$

Разделив $y(x)$ на $g(x)$, получим

$$y(x) = (x^3 + x^2 + x + 1)\ g(x) + x^2.$$

Полином остатка $s(x) = x^2$ соответствует вектору **s** =(001).

Рассмотренное свойство синдрома позволяет построить эффективные методы декодирования циклических кодов.

4.4. Методы синдромного декодирования циклического кода

Декодер Меггитта. Наиболее сложной частью простого синдромного декодирования является табулирование соответствия между синдромными многочленами и соответствующих им многочленов ошибок. Декодер Меггитта использует соответствие для некоторых типичных синдромных многочленов и проверяет синдромы только тех конфигураций ошибок, которые расположены в старших позициях. Декодирование ошибок в остальных позициях основано на циклической структуре кода и осуществляется позже. Таблица синдромов содержит только те синдромы, которые соответствуют многочленам ошибок с ненулевыми коэффициентами. Если вычисленный синдром находится в этой таблице, то ошибка исправляется. Затем принятое слово циклически сдвигается и повторяется процесс нахождения возможной ошибки в предшествующей по старшинству позиции. Этот

процесс повторяется последовательно для каждой компоненты; каждая компонента проверяется на наличие ошибки, и если ошибка найдена, то она исправляется.

В действительности нет необходимости вычислять синдромы для всех циклических сдвигов принятого слова. Новый синдром можно легко вычислить по уже вычисленному. Основная взаимосвязь описывается следующей теоремой.

Теорема. Предположим, что $g(x)\,h(x) = x^n - 1$ и $R_{g(x)}[y(x)] = s(x)$. Тогда

$$R_{g(x)}[x\,y(x) \bmod (x^n - 1)] = R_{g(x)}[x\,s(x)].$$

Теорема показывает, как вычислить синдром произвольного циклического сдвига конфигурации ошибки. Такое вычисление можно реализовать на простой цепи с регистром сдвига. Схема декодера Меггитта показана на рис. 4.1.

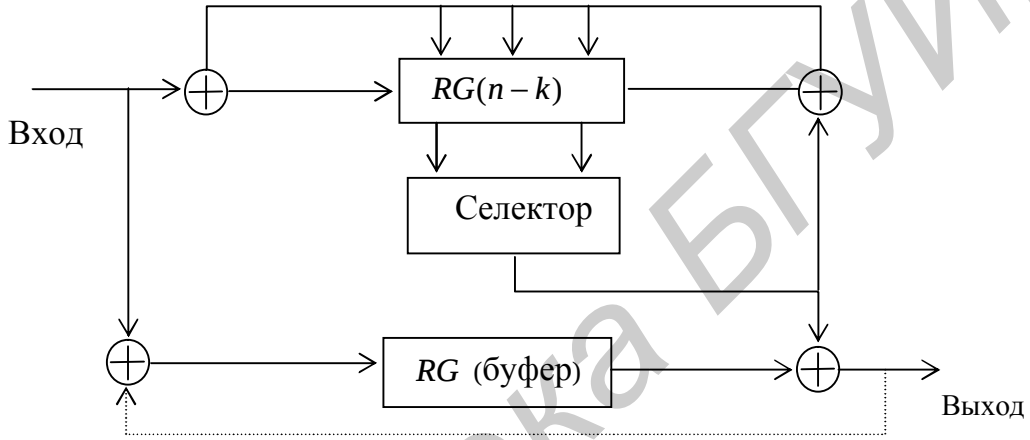


Рис. 4.1

Декодирование циклического кода путем вылавливания ошибок. Рассмотрим случай декодирования С кода с параметрами (n, k), $d = 2t+1$ и проверочной матрицей $\mathbf{H}=[I_{n-k}A]$. Передается кодовое слово **c**, а принимается вектор

$$\mathbf{y} = \mathbf{c} + \mathbf{e},$$

где **e –** вектор ошибки.

Синдром вектора **y** вычисляется как

$$\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathrm{T}} = (\mathbf{c} + \mathbf{e})\,\mathbf{H}^{\mathrm{T}} = \mathbf{e}\,\mathbf{H}^{\mathrm{T}}.$$

Образуем вектор $\mathbf{e^*}=(\mathbf{s}^{\mathrm{T}}, \mathbf{0})$, где **0** нулевой вектор, состоящий из k нулей. Нетрудно показать, что выполняется следующее соотношение:

$$\mathbf{e^*}\,\mathbf{H}^{\mathrm{T}} = \mathbf{s}\,.$$

Векторы **e** и **e*** имеют один и тот же синдром и соответствуют одному и тому же подмножеству кода **C**. Предположим, что вес синдрома $wt(\mathbf{s}) \le t$. Тогда $wt(\mathbf{e^*}) \le t$ и, следовательно, **e** = **e*,** так как соответствующее подмножество кода **C** может содержать только один вектор заданного веса. Таким образом, вектор ошибки можно записать как **e** = (**s**, **0**). Теперь предположим, что структура вектора ошибки весом не менее t может иметь в своем составе циклический сдвиг пачки из k нулей. На определенном i–м циклическом сдвиге в структуре вектора ошибки отличные от нуля

символы будут располагаться на первых (n - k) позициях. Для этого значения i вес соответствующего синдрома $s_i(x)$ будет удовлетворять неравенству $wt(s_i(x)) \le t$. Если синдром $s_i(x)$ вычислять как остаток от деления $x^i y(x)$ на порождающий полином $g(x)$, тогда $wt(s_i(x)) \le t$ для значений i, соответствующих соотношению $x^i e(x) = (s_i(x) \mathbb{M} \mathbf{0}(x))$.

Здесь $\mathbf{0}(x) = \sum_{j=n-k}^{n-1} 0 \cdot x^j$ полином нулевого вектора, $\mathbb{M}$ - операция конкатенации. Вектору ошибки $e(x)$ в этом случае соответствует циклический сдвиг $e(x) = x^i (s_i(x) \mathbb{M} \mathbf{0}(x))$. Свойство синдрома позволяет построить следующий алгоритм декодирования.

Алгоритм I

1. Вычисляется синдром $s(x)$ для принимаемого сигнала $y(x)$ с использованием алгоритма деления на порождающий полином.
2. Установка $i := 0$.
3. Если $wt(s_i(x),) \le t$, тогда полагаем $e(x) = x^i (s_i(x) \mathbb{M} \mathbf{0}(x))$ и корректируем ошибку, вычисляя $y(x) - e(x)$.
4. Устанавливаем $i := i + 1$;
5. Если $i = n$, тогда алгоритм останавливается и ошибка считается невыловленной.
6. Если $\deg(s_{i-1}(x) < n - k - 1$, тогда $s_i(x) = x\, s_{i-1}(x)$; в противном случае получаем $s_i(x) = x\, s_{i-1}(x) - g(x)$.
7. Перейти к шагу 3.

Пример. Пусть $g(x) = 1 + x^2 + x^3$ генерирует бинарный циклический код (7, 4, 3), позволяющий исправлять одну ошибки. Предположим, что передается кодовое слово $c(x) = 1 + x + x^5 = (1 + x + x^2)\, g(x)$. Принимается вектор $y(x) = 1 + x + x^5 + x^6$.

Разделим вектор $y(x)$ на порождающий полином $g(x)$:

$$y(x) = (x^3 + 1)g(x) + (x + x^2),\ \ s(x) = (x + x^2).$$

Так как вес синдрома больше 1, то вычислим синдром циклического сдвига $s_1(x)$ для $x\, y(x)$. Поскольку степень синдрома $s(x)$ равна $2 = n - k - 1$, то

$$s_1(x) = x\, s(x) \mod g(x) = 1.$$

Вес синдрома равен единице и соответствует корректирующей способности кода. Следовательно, вектор ошибки равен

$$e(x) = x^{7-1}(s_1(x) \mathbb{M} \mathbf{0}(x)) = x^6(1 + 0\, x^3 + 0\, x^4 + \ 0\, x^5 + 0\, x^6) = x^6.$$

Пример. Пусть $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ генерирует бинарный циклический код (15,7,5), позволяющий исправлять две ошибки. Любая ошибка весом 2, содержащая в своей структуре пачку из 7 нулей, может быть выловлена. Предположим, что принимается вектор **y**= (1100 1110 1100 010).

Вычислим синдром $y(x) = (x + x^2 + x^4 + x^5)\, g(x) + (1 + x^2 + x^5 + x^7)$. Далее будем вычислять синдромы $s_i(x)$ для циклических сдвигов $x^i\, y(x)$ до тех пор, пока вес синдрома не станет не более двух $wt(s_i(x)) \le 2$. Вычисления сведем в таблицу, табл. 4.1.

Ошибка представляется как

$$e(x) = x^{15-7}(s_7(x) \mathbb{M} \mathbf{0}(x)) = x^8(1 + x^5 + 0\, x^8 + \ldots + 0\, x^{14}) = (x^8 + x^{13}),$$

что соответствует вектору **c** = (0000 0000 1000 010).

Таблица 4.1

i	$s_i(x)$
0	10100101
1	11011001
2	11100111
3	11111000
4	01111100
5	00111111
6	00011111
7	10000100

Декодируем кодовое слово как

$$\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e} = (1100\ 1110\ 0100\ 000).$$

Коррекция пачки ошибок. Циклическая пачка из t ошибок может быть представлена в виде вектора, ненулевые компоненты которого циклически группируются на отрезке длиной t, причем в начале и конце отрезка располагаются отличные от нуля элементы вектора.

Пример

1. $\mathbf{e}_1$ = (01010110000) – вектор пачки ошибок длиной 6 в пространстве $V_{11}(2)$.

2. $\mathbf{e}_2$ = (00000010001) и $\mathbf{e}_3$ = (01000000100) – векторы пачек ошибок длиной 5 в пространстве $V_{11}(2)$.

В полиномиальной форме пачка ошибок длиной t может быть записана как

$$e(x) = x^i\, b(x) \pmod{x^n - 1},$$

где $b(x)$ – полином степени (t - 1), который описывает расположение ошибок; значение индекса i определяет начало пачки.

Для рассмотренного выше примера пачки ошибок в полиномиальной форме имеют вид $e_1(x) = x\,(1 + x^2 + x^4 + x^5)$, $e_2(x) = x^6(1 + x^4)$, $e_3(x) = x^8\,(1 + x^4)$.

Рассмотрим случай применения линейного кода. Если все пачки ошибок длиной t или меньше соотносятся с разными смежными классами кода **C**, тогда каждая такая пачка ошибок идентифицируется уникальным синдромом и все ошибки могут быть исправлены. Более того, если **C** – линейный код, способный корректировать пачки

ошибок длиной t, тогда все эти ошибки должны появляться в разных смежных классах.

Декодирование пачки ошибок методом вылавливания. Параметры корректирующего кода (n, k), исправляющего пачки ошибок длиной t, должны удовлетворять условию $(n - k) \geq 2t$. Предполагается, что структура вектора пачки ошибок длиной t имеет отрезок из $(n - t)$ нулевых элементов. Если вектор **e** представляет собой пачку ошибок длиной t и ошибки располагаются на первых $(n - k)$ позициях вектора, тогда синдром $\mathbf{eH}^{\mathrm{T}} = \mathbf{s}$ характеризует структуру (нециклической) пачки ошибок длиной не более t. Если ошибки располагаются не на первых $(n - k)$ позициях вектора, то для вычисления синдрома используется его свойство (см. алгоритм I).

Алгоритм II

1. Вычисляется синдром $s(x)$ для $y(x)$.
2. Устанавливается $i := 0$.
3. Контролируется $(n - k)$ первых позиций синдрома. Если конфигурация компонент синдрома $s_i(x)$ соответствует нециклической пачке ошибок длиной t или менее, то вектор ошибок $e(x) = x^{n-i}(s_i(x) \mathbb{M} \mathbf{0}(x))$.
4. Устанавливается $i := i + 1$.
5. Если $i = n$, то алгоритм останавливается и считается, что ошибка не вылавливается .
6. Вычисляется синдром $s_i(x)$ циклического сдвига по аналогии с алгоритмом I.
7. Переход к шагу 3.

Пример. Пусть $g(x) = 1 + x + x^2 + x^3 + x^6$ генерирует бинарный циклический код (15, 9), позволяющий исправлять пачку ошибок длиной $t = 3$. Принимается вектор

$$\mathbf{y} = (1110\ 1110\ 1100\ 000).$$

Вычислим синдром $y(x) = (x^2 + x^3)\, g(x) + (1 + x + x^4 + x^5)$, $s(x) = (1 + x + x^4 + x^5)$.

Конфигурация первых символов (n - k) =15 – 9 = 6 синдрома не соответствует пачке ошибки длиной 3. Значения синдрома для других циклических сдвигов принимаемого сигнала приведены в таблице, табл.4.2.

Таблица 4.2

i	$s_i(x)$
0	110011
1	100101
2	101110
3	010111
4	110111
5	100111
6	101111

7	101011
8	101001
9	101000 – пачка ошибок $t = 3$

Вектор ошибок вычисляется как

$$e(x) = x^{n-i}\,(s_9(x)\ \mathbb{M}\ \mathbf{0}(x)) = x^6\,(1 + x^2 + 0\,x^6 + \ldots + 0\,x^{14}) \bmod(x^{15}-1)\ \ \mathbf{a}\ \ (000000\ 101000000).$$

Переданное кодовое слово восстанавливается как

$$c(x) = y(x) - e(x) = (1 + x + x^2 + x^4 + x^5 + x^9)\ \ \mathbf{a}\ \ (1110\ 1100\ 0100\ 000).$$

Заметим, что в рассматриваемом примере синдром $s_8(x)$ имеет вес, равный 3, но конфигурация структуры не соответствует пачки ошибок длиной 3.

В табл. 4.3 приводятся сведения о корректирующей способности пачки ошибок некоторых циклических кодов.

Таблица 4.3

$g(x)$	(n, k)	Длина исправляемой пачки ошибок t
$1 + x^2 + x^3 + x^4$	(7, 3)	2
$1 + x^2 + x^4 + x^5$	(15, 10)	2
$1 + x^4 + x^5 + x^6$	(31, 25)	3
$1 + x^3 + x^4 + x^5 + x^6$	(15, 9)	3
$1 + x + x^2 + x^3 + x^6$	(15, 9)	3

Граница Рейджера. Для любого линейного (n, k) –кода, исправляющего пачки ошибок длиной b и меньше, должно выполняться следующее соотношение: $n - k \geq 2b$.

Теорема Файра. Пусть **C** – циклический код длиной n_0 с порождающим многочленом $g_0(x)$, исправляющий пачки ошибок длиной b и менее, и пусть $g_1(x)$ – неприводимый взаимно простой с $g_0(x)$ многочлен с периодом n_1, степень которого не меньше b. Тогда циклический код длиной $n = (n_0\ n_1 / \text{НОД}(n_0, n_1))$ с порождающим многочленом $g(x) = g_0(x)\ g_1(x)$ исправляет пачки ошибок длиной b и менее.

Из теоремы следует, что если $g_1(x)$ – неприводимый многочлен с периодом n_1, степень которого не меньше b, взаимно простой с полиномом $(x^{2b} - 1)$, тогда циклический код длиной $(2b - 1)\ n_1 / \text{НОД}(2b - 1, n_1)$ с порождающим многочленом $(x^{2b-1} - 1)\ g_1(x)$ исправляет пачки ошибок длиной b и менее. Такой код называется *кодом Файра*, он имеет более чем $3b - 1$ проверочных символов, что на $b - 1$ больше нижней границы Рейджера, равной $2b$.

4.5. Циклические коды Хэмминга

Для каждого целого m над полем $GF(q)$ существует код Хэмминга с параметрами $(n = \frac{q^m - 1}{q-1},\ k = \frac{q^m - 1}{q-1} - m)$, где n - длина кода.

Пусть a - примитивный элемент поля $GF(q^m)$. Определим $b = a^{q-1}$. Согласно теореме Эйлера

$$b^{(\frac{q^m-1}{q-1})} = 1 \bmod f(x). \qquad (4.1)$$

Поскольку b - это корень многочлена, то справедливо соотношение

$$x^{(\frac{q^m-1}{q-1})} - 1 = f(x).$$

Следовательно, b является делителем многочлена $f(x)$ и может быть выбрано для формирования минимального многочлена $m_b(x)$:

$$m_b(x) \mid (x^{\left(\frac{q^m-1}{q-1}\right)} - 1), \qquad (4.2)$$

который в свою очередь определяет порождающий многочлен $m_b(x) = g(x)$. Проверочная матрица задается как $\mathbf{H} = \left\lfloor b^0, b^1, ..., b^{n-1} \right\rfloor$. Привязка к элементам поля позволяет осуществить маркировку позиции ошибки (локализовать ошибку при приеме) через элемент поля.

Пример. Примем $q = 2$ и $b = a$. Кодовое слово имеет вид $c(x) = a(x) \cdot g(x)$. Декодируется полином $y(x) = c(x) + e(x)$, где $e(x)$ - полином ошибки.

Если код Хэмминга исправляет одну ошибку, то положение этой ошибки может быть определено как $e(x) = x^i$. Если учесть, что при декодировании $c(x) = \left| x = a^j \right| = c(a^j) = 0$, то $y(a^j) = e(a^j) = x^i$, т.е. значение индекса i определяет местоположение ошибки.

Так как $g(a) = 0$ и все значения a от 0 до $2^m - 2$ различны, то по величине a^i можно определить позицию ошибки и минимальное кодовое расстояние.

Минимальное кодовое расстояние кода с проверочной матрицей $\mathbf{H} = \left\lfloor b^0, b^1, .., b^{n-1} \right\rfloor$, где $b = a^{q-1}$, $n = \frac{q^m - 1}{q-1}$, равно, по меньшей мере, 3 тогда и

только тогда, когда числа n и $q-1$ взаимно просты.

Примеры циклических кодов Хэмминга

1. Троичный код над полем $GF(3)$. Пусть $m=3$, тогда $n=\frac{q^m-1}{q-1}=\frac{3^3-1}{3-1}=13$, $k=13-m=10$.

2. Четверичный код q = 4 с параметрами $(n,k)=(85,81)$. Зададим $m=4$, тогда $n=\frac{q^m-1}{q-1}=\frac{4^4-1}{4-1}=85$, $k=85-4$. Элементы кода принадлежат расширенному полю $GF(4)=GF(2^2)$, $f(x)=x^2+x+1$. Структура кода определяется другим конечным полем $GF(q^m)=GF(4^4)$, в котором определены элемент $b=a^{q-1}=a^3$ и минимальный многочлен. Многочлен $m(x)$ будет минимальным, если $b, b^q, b^{q^2},...,b^{q^{m-1}}$ будут корнями многочлена $m(x)$. Для нашего случая $\left\{b^{q^i}\right\}=\left\{b, b^4, b^{16}, b^{64}\right\}$. Порождающий полином $g(x)$ кода Хэмминга равен минимальному многочлену:

$$m_b(x)=\prod_i (x-b^{q^i})=(x-b)\cdot(x-b^4)\cdot(x-b^{16})\cdot(x-b^{64})=g(x).$$

Расширенный код Хэмминга $(n,k)=(2^m, 2^m-m)$ строится, если использовать матрицу следующего вида:

$$\mathbf{H}_p=\begin{bmatrix} 1 & 1 & ... & 1 \\ 0 & & & \\ ... & & \mathbf{H}_x & \\ 0 & & & \end{bmatrix}. \qquad (4.3)$$

4.6. Задание циклического кода с помощью элементов поля

Пусть $c(x)=c_0+c_1\cdot x+...+c_{n-1}\cdot x^{n-1}$ - произвольное слово циклического кода, который образован с помощью генераторного полинома $g(x)$. Предположим, что $a_1, a_2, ..., a_r$ - это неповторяющиеся корни $g(x)$, т.е. $g(a_i)=0$. Так как $c(x)$ делится на $g(x)$ без остатка, то все корни $g(x)$ должны быть также корнями любого слова $c(x)$:

$$c_0+c_1\cdot a_j+c_2\cdot a_j^2+...+c_{n-1}\cdot a_j^{n-1}=0, \quad j=1,...,r\ . \qquad (4.4)$$

В матричной форме эта запись будет иметь следующий вид:

$$(c_0, c_1, ..., c_{n-1}) \cdot \begin{bmatrix} 1 & a_1 & a_1^2 & ... & a_1^{n-1} \\ 1 & a_2 & a_2^2 & ... & a_2^{n-1} \\ ... & ... & ... & ... & ... \\ 1 & a_r & a_r^2 & ... & a_r^{n-1} \end{bmatrix}^T = 0. \qquad (4.5)$$

Матрица элементов поля удовлетворяет условию $\mathbf{c} \cdot \mathbf{H}^T = 0$ и является проверочной для кода **C**. Справедливо и обратное. Если задано r неповторяющихся величин $(a_1, a_2, ..., a_r)$, то матрица **C**, строки **c** которой удовлетворяют условию $\mathbf{c} \cdot \left[a_i^j \right]^T = 0$, образует линейный код, при этом длина кода должна быть равна

$$n = НОК(порядков\ a_1, a_2, ..., a_r). \qquad (4.6)$$

Порождающий полином $g(x)$ является произведением всех различных неприводимых над полем $GF(q)$ полиномов, корнями которых служат величины $(a_1, a_2, ..., a_r)$, т.е.

$$g(x) = НОК(g_1(x), g_2(x), ..., g_r(x)), \qquad (4.7)$$

где $g_i(a_i) = 0$ и $g_i(x)$ - неприводимый полином.

Пример. Элементы поля $GF(2^3)$, построенного по неприводимому полиному $f(x) = x^3 + x + 1$, равны

$$a^0 = 100 \quad a^1 = 010 \quad a^2 = 001 \quad a^3 = 110 \quad a^4 = 011 \quad a^5 = 111 \quad a^6 = 101.$$

Введём обозначения. Пусть $a_1 = a$; $a_2 = a^2$; $a_3 = a^4$ … $a_i = a^{2^{i-1}}$. Тогда для $N = 7$ проверочная матрица кода будет иметь вид

$$\mathbf{H} = \begin{bmatrix} 1 & a_1 & a_1^2 & a_1^3 & a_1^4 & a_1^5 & a_1^6 \\ 1 & a_2 & a_2^2 & a_2^3 & a_2^4 & a_2^5 & a_2^6 \\ 1 & a_3 & a_3^2 & a_3^3 & a_3^4 & a_3^5 & a_3^6 \end{bmatrix} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

По своей структуре данная матрица избыточна, т.к. некоторые строки совпадают. Например, строки 6 и 8 являются суммой строк 2 и 3. После удаления избыточности матрица приобретает следующий вид:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Структура данной проверочной матрицы совпадает со структурой проверочной матрицы кода Хэмминга.

5. ВАЖНЕЙШИЕ КОДЫ

5.1. БЧХ-коды

Определение примитивного БЧХ-кода. Пусть m и $t < q^m - 1$ - целые числа, тогда q-ичный БЧХ-код имеет параметры $n = q^m - 1;\ k \geq n - m \cdot t;\ d_{\min} \geq 2t + 1$. Код формируется генераторным полиномом $g(x)$, имеющим корнями примитивный элемент поля $GF(q^m)$ a^i, $i = 1,..., 2t$,. Соответственно

$$g(x) = НОК(m_a(x), m_{a^2}(x),..., m_{a^{2t}}(x)), \tag{5.1}$$

где m_{a^i} - минимальный полином, соответствующий элементу поля a^i.

Для $q = 2$ в поле $GF(2^m)$ элементы a и a^2 имеют одинаковые минимальные полиномы, поэтому для двоичного БЧХ-кода

$$g(x) = НОК(m_a(x), m_{a^3}(x),..., m_{a^{2t-1}}(x)). \tag{5.2}$$

Расширенные БЧХ-коды формируются по следующему правилу:

$$g(x) = НОК(m_{a^b}(x), m_{a^{b+1}}(x),..., m_{a^{b+2t-1}}(x)). \tag{5.3}$$

Пусть необходимо исправить t ошибок. Предположим, что

$$g(x) = НОК(m_1(x), m_2(x),..., m_r(x)), \tag{5.4}$$

где $m_i(x) = m_{a^i}$. Код принятого сигнала можно записать как

$$y(x) = c(x) + e(x), \quad e(x) = e_0 + e_1 \cdot x + e_2 \cdot x^2 + ... + e_{n-1} \cdot x^{n-1},$$

где $e(x)$ - полином ошибки.

Если примем, что h_i - корни генераторного полинома $g(x)$, тогда

$$y(h_i) = e(h_i) = S_j, \tag{5.5}$$

где S_j - это j-я компонента синдрома, равная $S_j = \sum_{i=0}^{n-1} e_i \cdot h_j^i$. Выберем множество $\{h_j\}$ в виде $(h_1,...,h_j) = (a, a^2, a^3,..., a^{2t})$, где a - примитивный элемент поля $GF(q^m)$. Такая конструкция позволяет по множеству $\{S_j\}$ исправить t ошибок.

Алгоритм формирования кода имеет следующий вид:

1. Строится поле $GF(q^m)$, для которого a - примитивный элемент поля.
2. Определяются примитивные полиномы $m_i(x) = m_{a^i}(x)$ для a^i, $i = 1,...,2t$.
3. Находится $g(x)$ как $g(x) = HOK(m_1(x), m_2(x),..., m_{2t}(x))$.

Пример. Пусть в поле $GF(2^4)$ $m = 4$, $q = 2$. Положим $t = 2$, $n = 2^4 - 1 = 15$, $k \geq n - m \cdot t = 15 - 4 \cdot 2 = 7$, тогда

$$g(x) = HOK(m_1(x), m_2(x), m_3(x), m_4(x)) =$$
$$= HOK\left[x^4 + x + 1, x^4 + x + 1, x^4 + x^3 + x^2 + x + 1, x^4 + x + 1\right] =$$
$$= (x^4 + x + 1) \cdot (x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1.$$

Соответственно $d_{\min} \geq 2 \cdot 2 + 1 = 5$, $\deg(g(x)) = 8$, что дает $r = n - k = 8$, $k = 7$. В результате получается БЧХ-код с параметрами $(15,7,5)$.

Структура БЧХ-кодов тесно связана с понятием циклотомических классов. Рассмотрим возможность построения БЧХ-кода, корректирующего три ошибки. Определим конечное поле $GF(2^4)$, для которого существует четыре циклотомических класса $K_1 = \{1, 2, 4, 8\}$; $K_2 = \{3, 6, 12, 9\}$; $K_5 = \{5, 10\}$; $K_7 = \{7, 14, 13, 11\}$.

Возьмем элемент поля $a^3 \in GF(2^4)$, которому соответствует минимальный неприводимый в $\mathbf{Z}_2[x]$ полином $m_3(x) = x^4 + x^3 + x^2 + x + 1$. Элемент α^3 является корнем полинома $(x^5 - 1)$. Элементы α и α^3 являются корнями полинома

$$m_{13}(x) = m_1(x) m_3(x) = (x^4 + x + 1) m_3(x) = x^8 + x^7 + x^6 + x^4 + 1.$$

Степень примитивного элемента поля $(a^5)^3 = a^{15} = 1$ и, следовательно, a^5 - является корнем многочлена $x^3 - 1 = x^3 + 1 = (x + 1)(x^2 + x + 1)$ и кодовые слова должны быть кратны полиному

$$g(x) = m_{135}(x) = m_{13}(x)\, m_5(x) = m_{13}(x)\,(x^2 + x + 1) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1.$$

В результате получается БЧХ-код с параметрами $(15,5,7)$.

Возможности поля $GF(2^4)$ позволяют построить код, корректирующий четыре ошибки. Действительно, структура циклотомических классов показывает, что может быть определен минимальный полином $m_7(x)$, корнями которого являются степени $a^7, a^{14}, a^{13}, a^{11}$. Требуемый полином $m(x)$, на основе которого формируется код, определяется через произведение $g(x) = m_{1357}(x) = m_1(x) m_3(x) m_5(x) m_7(x)$ и имеет степень, равную 14. Получается БЧХ-код (15, 1) с одним информационным символом.

Проверочные частоты. В несистематическом виде циклический код записывается в виде произведения полиномов $c(x) = a(x)g(x)$. Дискретное преобразование Фурье в конечном поле задается как $E_k = \sum_{l=0}^{n-1} e_l w^{lk}$, $k = 1,\ldots, 2t$, при $w^i = a^i$. В частотной области можно определить образы Фурье для кодового слова $C_w = \{C_{w,i}\}$ и порождающего полинома $G_w = \{G_{w,i}\}$. Роль спектральных компонент образа $G_{w,i}$ состоит в том, чтобы задать частоты, в которых стоят нулевые спектральные компоненты $C_{w,i}$. Циклический код можно определить как множество таких кодовых слов над полем $GF(p)$, у которых все спектральные компоненты, принадлежащие заданному множеству так называемых проверочных частот $j_1,\ldots, j_{n-k}$, равны нулю. В этом случае элемент a^j является корнем многочлена $g(x)$. При этом обратные преобразования Фурье спектральных векторов кода должны принадлежать полю $GF(p)$. Это условие выполняется, если соблюдены ограничения сопряженности. Если **V** есть n-мерный вектор с компонентами из $GF(p^m)$, где n делит $(p^m - 1)$, тогда обратное преобразование Фурье является вектором с компонентами из $GF(p)$, когда выполняются равенства $V_j^p = V_{((pj))}$, $j = 0, \ldots, n-1$.

Коды БЧХ являются циклическими кодами, в которых проверочные частоты выбираются последовательно. Исправляющий t ошибок код БЧХ длиной $(p^m - 1)$ определяется как множество всех кодовых слов над $GF(p)$, спектр которых равен нулю в заданном блоке из $2t$ последовательных частот.

5.2. Декодирование БЧХ-кодов

Предположим, что задан БЧХ-код, корректирующий t ошибок, и при приеме произошло n ошибок, причём $0 \le n \le t$. Ошибки располагаются на позициях $(i_1, i_2, \ldots, i_n)$, вектор ошибок $e(x) = e_{i_1} \cdot x^{i_1} + e_{i_2} \cdot x^{i_2} + \ldots + e_{i_n} \cdot x^{i_n}$, $i_j \to a^{i_j}$. Декодер решает две задачи. Первая – определяет координаты ошибок, т.е. значения $\{i_j\}$. Вторая - оценивает величины ошибок. Свойства кода позволяют рассматривать структуру синдрома с двух точек зрения. Во-первых, компонента синдрома может быть вычислена как

$$S_j = y(a^j) = c(a^j) + e(a^j) = e(a^j) \qquad (5.6)$$

Во-вторых, если для функции ошибки определить дискретное преобразование Фурье в конечном поле, то компоненты синдрома можно трактовать как коэффициенты этого преобразования $S_j = E_j$, $j = 1,\ldots,2t$.

Введем следующие обозначения: $X_e = a^{i_e}$ - локатор ошибки; $e_{i_e} = Q_e$ - величина ошибки. Для всех компонентов синдрома можно составить следующие уравнения:

$$\begin{aligned} &S_1 = Q_1 \cdot x_1 + Q_2 \cdot x_2 + \ldots + Q_n \cdot x_n \ , \\ &S_2 = Q_1 \cdot x_1^2 + Q_2 \cdot x_2^2 + \ldots + Q_n \cdot x_n^2 \ , \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &S_{2t} = Q_1 \cdot x_1^{2t} + Q_2 \cdot x_2^{2t} + \ldots + Q_n \cdot x_n^{2t} \ , \end{aligned} \tag{5.7}$$

$$S_j = \sum_{i=1}^{n} Q_i \cdot x_i^j \ . \tag{5.8}$$

Оценки параметров ошибки можно получить путем решения нелинейной системы уравнений (5.7 – 5.8), что достаточно сложно. Используем искусственный приём. Введем так называемый *полином локаторов ошибки* $s(x)$.

$$s(x) = s_n \cdot x^n + s_{n-1} \cdot x^{n-1} + \ldots + s_1 \cdot x + s_0 . \tag{5.9}$$

Полином $s(x)$ можно представить в виде

$$s(x) = (1 - x \cdot X_1) \cdot (1 - x \cdot X_2) \cdot \ldots \cdot (1 - x \cdot X_n) = \prod_i (1 - x \cdot X_i) . \tag{5.10}$$

Корнями полинома $s(x)$ являются величины $x = X_i^{-1}$. Если коэффициенты $s(x)$ известны, то соответственно для определения локаторов ошибок X_i нужно найти корни $s(x)$, после чего вычислить обратные к ним величины. При этом решение системы нелинейных уравнений сводится к решению системы линейных уравнений относительно коэффициентов полинома $s(x)$. Составим систему линейных уравнений, воспользовавшись записью

$$s(x) = s_n \cdot x^n + s_{n-1} \cdot x^{n-1} + \ldots + s_1 \cdot x + s_0 .$$

Умножим левую и правую части уравнения на $Q_i \cdot x_i^{j+n}$, а также положим, что $x = X_e^{-1}$ и $s(X_e^{-1}) = 0$,тогда справедлива система уравнений

$$0 = Q_e \cdot X_e^{j+n} \cdot (1 + s_1 \cdot X_e^{-1} + s_2 \cdot X_e^{-2} + \ldots + s_n \cdot X_e^{-n}), e = 1,\ldots,n \ .$$

Просуммировав уравнения по индексу e от 1 до n , для каждого индекса j получим:

$$\sum_{e=1}^{n} Q_e \cdot X_e^{j+n} + s_1 \cdot \sum_{e=1}^{n} Q_e \cdot X_e^{j+n-1} + \ldots + s_n \cdot \sum_{e=1}^{n} Q_e \cdot X_e^{j} = 0. \qquad (5.11)$$

Учитывая, что $S_i = \sum_{e=1}^{n} Q_e \cdot X_e^{j}$ получаем алгебраическую систему уравнений

$$s_1 \cdot S_{j+n-1} + s_2 \cdot S_{j+n-1} + \ldots + s_n \cdot S_j = -S_{j+n},\ j = 1,\ldots,n,$$

или в матричной форме

$$\begin{bmatrix} S_1 & S_2 & S_3 & \ldots & S_{n-1} & S_n \\ S_2 & S_3 & S_4 & \ldots & S_n & S_{n+1} \\ S_3 & S_4 & S_5 & \ldots & S_{n+1} & S_{n+2} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ S_n & S_{n+1} & S_{n+2} & \ldots & S_{2n-2} & S_{2n-1} \end{bmatrix} \cdot \begin{bmatrix} s_n \\ s_{n-1} \\ s_{n-2} \\ \ldots \\ s_1 \end{bmatrix} = \begin{bmatrix} -S_{n+1} \\ -S_{n+2} \\ -S_{n+3} \\ \ldots \\ -S_{2n} \end{bmatrix}. \qquad (5.12)$$

Система связывает значения компонент синдрома S_i с коэффициентами полинома локаторов ошибок, задача вычисления которых решается достаточно просто, если можно найти обратную матрицу синдромов.

Алгоритм декодирования Питерсона-Горенштейна-Цирлера (ПГЦ)

1. Вычисляются компоненты синдрома: $S_i = y(a^j),\ \ j = 1,2,\ldots,2t$. Задается величина $\nu = t$.

2. Составляется матрица синдромов $M_n = [S_{l+k+1}]$, $l,k = 0,1,\ldots, \nu\text{-}1$.

3. Вычисляется определитель матрицы $\Delta = \det M_n$. Если $\Delta = 0$, то матрица является сингулярной, и обратной матрицы не существует. В этом случае изменяется размер матрицы $\nu := (\nu - 1)$ до тех пор, пока определитель не станет отличным от нуля. После чего повторяют второй шаг.

4. Определяются коэффициенты полинома локаторов ошибок:

$$\begin{bmatrix} s_n \\ s_{n-1} \\ s_{n-2} \\ \ldots \\ s_1 \end{bmatrix} = M_n^{-1} \cdot \begin{bmatrix} -S_{n+1} \\ -S_{n+2} \\ -S_{n+3} \\ \ldots \\ -S_{2n} \end{bmatrix}. \qquad (5.13)$$

5. Находятся корни $s(x) = 0$ и через инверсию корней вычисляются значения X_i.

6. Оцениваются величины ошибок $\{Q_i\}$ при решении системы

$$\begin{bmatrix} Q_1 \\ Q2 \\ Q_3 \\ \dots \\ Q_m \end{bmatrix} = \begin{bmatrix} X_1 & \dots & X_m \\ X_1^2 & \dots & X_m^2 \\ X_1^3 & \dots & X_m^3 \\ \dots & \dots & \dots \\ X_1^m & \dots & X_m^m \end{bmatrix}^{-1} \cdot \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \dots \\ S_m \end{bmatrix}. \qquad (5.14)$$

7. Корректируется принятый вектор, подставляя на позиции X_i вычисленные значения Q_i .

Для исправления одиночной ошибки можно использовать выражения

$$S_1 \cdot s_1 = -S_2 ; \quad s_1 = -\frac{S_2}{S_1}.$$

Если код бинарный, то $s_1 = -S_1$, следовательно, $s(x) = 1 - S_1 \cdot x$. Полином $s(x)$ имеет 0 в точке $1/S_1$. Следовательно, $S_1 = a^i$ и ошибка локализуется в позиции $2^m - 1 - i$.

Рассмотрим случай, когда код исправляет 2 ошибки. Тогда имеем

$$[S_{i+j}] = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix}.$$

Если определитель матрицы $\Delta \neq 0$, то получаем:

$$\begin{bmatrix} s_2 \\ s_1 \end{bmatrix} = \frac{1}{S_1 \cdot S_2 - S_2^2} \cdot \begin{bmatrix} S_3 & -S_2 \\ -S_2 & S_1 \end{bmatrix} \cdot \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix},$$

$$s_1 = \frac{S_2 \cdot S_3 - S_1 \cdot S_4}{S_1 \cdot S_3 - S_2^2}; \quad s_2 = \frac{-S_3^2 + S_2 \cdot S_4}{S_1 \cdot S_3 - S_2^2}.$$

Для бинарного кода $s_1 = -S_1, \quad s_2 = -\frac{S_3 + S_1^2}{S_1}$.

При большом количестве ошибок алгоритм ПГЦ характеризуется значительной вычислительной сложностью. В таких случаях для вычисления полинома локаторов ошибок используют алгоритмы Сигуяма, Берлекемпа–Месси, а для оценки значений ошибок для q-ичных кодов – алгоритм Форни .

Алгоритм Сигуямы. Позволяет решить систему уравнений в поле *F*:

$$\begin{bmatrix} E_{t-1} & E_{t-2} & \mathbf{L} & E_0 \\ E_t & E_{t-1} & \mathbf{L} & E_1 \\ \mathbb{M} & \mathbb{M} & \mathbf{O} & \mathbb{M} \\ E_{2t-2} & E_{2t-3} & \mathbf{L} & E_{t-1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \mathbb{M} \\ s_t \end{bmatrix} = -\begin{bmatrix} E_t \\ E_{t+1} \\ \mathbb{M} \\ E_{2t-1} \end{bmatrix},$$

что разрешает центральную проблему декодирования локаторов ошибок.

Предполагаем, что $s_0 = 1$, полином локаторов ошибок $s(x) = \sum_{j=0}^{t} s_j x^j$ и полином спектральных коэффициентов $\langle E_j \rangle$ функции ошибок $E(x) = \sum_{j=0}^{2t-1} E_j x^j$. Считается, что коэффициенты полинома произведения $s(x)E(x)$ равны нулю для $j = t,\ldots,2t\text{-}1$.

Решение матричной системы уравнений эквивалентно решению полиномиального уравнения $s(x)E(x) = \Gamma(x) \bmod x^{2t}$ для степени полинома $s(x)$ не больше, чем t и степени $\Gamma(x)$ не больше, чем $t-1$.

Рассматриваемый алгоритм решает полиномиальное уравнение с использованием алгоритма Евклида следующим образом. Обозначим начальные условия как $a^{(0)}(x) = x^{2t}$ и $b^{(0)} = E(x)$. Для (r)-й итерации вычисления алгоритма Евклида можно записать

$$a^{(r-1)}(x) = \Theta^{(r)}(x) b^{(r-1)}(x) + b^{(r)} \text{ при } \deg(b^{(r)}(x) < \deg(b^{(r-1)}(x)).$$

Определим условие выполнения равенства $a^{(r)}(x) = b^{(r-1)}(x)$:

$$\begin{bmatrix} a^{(r)}(x) \\ b^{(r)}(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -\Theta^{(r)}(x) \end{bmatrix} \begin{bmatrix} a^{(r-1)}(x) \\ b^{(r-1)}(x) \end{bmatrix}.$$

Матрицу $A^{(r)}(x)$ (r)-й итерации по индукции можно определить как

$$A^{(r)}(x) = \begin{bmatrix} 0 & 1 \\ 1 & -\Theta^{(r)}(x) \end{bmatrix} A^{(r-1)}(x) \quad \text{и} \quad A^{(0)}(x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Для $r \geq 0$ имеем

$$\begin{bmatrix} a^{(r)}(x) \\ b^{(r)}(x) \end{bmatrix} = \begin{bmatrix} A_{11}^{(r)}(x) & A_{12}^{(r)}(x) \\ A_{21}^{(r)}(x) & A_{22}^{(r)}(x) \end{bmatrix} \begin{bmatrix} x^{2t} \\ E(x) \end{bmatrix}.$$

Откуда получаем желаемую форму уравнения

$$b^{(r)}(x) = A_{22}^{(r)} E(x) \bmod x^{2t},\ s(x) \leftarrow A_{22}^{(r)}.$$

Для решения задачи декодирования требуется найти такое значение r,при котором $\deg(A_{22}^{(r)}(x)) \leq t$ и $\deg(b^{(r)}(x) \leq t-1$. Это требование удобно выполнить, выбирая r^t как значения r, удовлетворяющие условиям $\deg(b^{(r^t-1)}) \geq t$ и $\deg(b^{(r^t)}) \leq t-1$.

Алгоритм Берлекемпа-Месси. Находит полином локаторов ошибок

$$s(x) = s_n \cdot x^n + s_{n-1} \cdot x^{n-1} + \ldots + s_1 \cdot x + 1$$

по заданному синдрому. В основе алгоритма лежит адаптивная модель авторегрессионного фильтра, который генерирует компоненты синдрома по правилу

$$S_j = -\sum_{i=1}^{n} s_i \cdot S_{j-1},$$

где n - количество произошедших ошибок; $j = n+1,...,2n$. Фильтр перестраивает свою структуру в зависимости от возможной конфигурации ошибок.

Для нахождения полинома локатора ошибок используется итерационная процедура. На каждой итерации вычисляется модель регистра с обратными связями, генерирующего первые r компонент синдрома, где r – номер этапа. Длина регистра на r-м этапе обозначается L_r. На r-м этапе оценка $\hat{S}_r$ r-й компоненты синдрома равна

$$\hat{S}_r = -\sum_{j=1}^{L_r} s_j^{(r-1)} \cdot S_{r-j}, \qquad (5.15)$$

где s_j^{r-1}- весовые коэффициенты на $(r-1)$-м предыдущем этапе.

Ошибка (невязка) вычислений Δ_r определяется как

$$D_r = S_r - \hat{S}_r = \sum_{j=0}^{L_r} s_j^{(r-1)} \cdot S_{r-j} \quad . \qquad (5.16)$$

Если $\Delta_r = 0$, то параметры модели не меняются:

$$(L_r, s^{(r)}(x)) := (L_{r-1}, s^{(r-1)}(x)). \qquad (5.17)$$

Если $\Delta_r \neq 0$, то параметры модели меняются по правилу

$$s^{(r)}(x) = s^{(r-1)}(x) + A \cdot x^l \cdot s^{(m-1)}(x), \qquad (5.18)$$

где $s(x)$ - веса; x^l - некоторое целое число; A - элемент поля.

Новая невязка может быть вычислена как

$$\Delta'_r = \sum_{j=0}^{L_r} s_j^{(r)} \cdot S_{r-j} = \sum_{j=0}^{L_r} s_j^{(r-1)} \cdot S_{r-j} + A \cdot \sum_{j=0}^{L_r} s_j^{(m-1)} \cdot S_{r-j-1} \qquad (5.19)$$

Определим величины A, m и r так, чтобы новая невязка Δ'_r была равна нулю. Выберем следующие значения:

$$m < r \rightarrow \Delta_m \neq 0; \quad l = r - m; \quad A = -\Delta_m^{-1} \cdot \Delta_r. \qquad (5.20)$$

После подстановки получаем

$$\Delta'_r = \Delta_r - \frac{\Delta_r}{\Delta_m} \cdot \Delta_m = 0$$

В этом случае новая модель регистра сдвига будет генерировать последовательность $S_1,...,S_{r-1},S_r$. Если выберем $L_m > L_{m-1}$, то получим регистр с обратной связью минимальной длины.

Теорема Берлекемпа–Месси. Пусть заданы компоненты синдрома $S_1,...,S_{2t}$ из некоторого поля, тогда при начальных условиях $s^{(0)}(x)=1,\ B^{(0)}(x)=1,\ L_0=1$ выполняются следующие рекуррентные равенства, использующиеся для вычисления $s(x)$:

1. $\Delta_r = \sum_{j=0}^{n-1} s_j^{(r-1)} \cdot S_{r-j}$
2. $L_r = d_r \cdot (r - L_{r-1}) + (1-d_r) \cdot L_{r-1}$
3. $\begin{bmatrix} s^{(r)}(x) \\ B^{(r)}(x) \end{bmatrix} = \begin{bmatrix} 1 & -\Delta_r \cdot x \\ \Delta_r^{-1} \cdot d_r & (1-d_r) \cdot x \end{bmatrix} \cdot \begin{bmatrix} s^{(r-1)}(x) \\ B^{(r-1)}(x) \end{bmatrix}$, где $r = 1,2,...,2t$.

$$\begin{cases} d_r = 1, \text{ если одновременно } \begin{cases} \Delta_r \neq 0, \\ 2L_{r-1} \leq r-1. \end{cases} \\ d_r = 0 \quad \text{в остальных случаях.} \end{cases}$$

При выполнении таких условий полином $s(x)$ является многочленом наименьшей степени, коэффициенты которого удовлетворяют равенству

$$S_j + \sum_{j=1}^{n-1} s_j^{(2e} \cdot S_{r-j} = 0, \text{ где } \quad r = L_{2t} + 1,...,2t \tag{5.21}$$

Граф-схема алгоритма Берлекемпа–Месси приведена на рис. 5.1.

Пример. Рассмотрим БЧХ-код (15,5) над полем $GF(2^4)$, $f(x) = x^4 + x + 1$, количество ошибок $t = 3$, $g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$.

Пусть $e(x) = x^2 + x^7$, $S_1 = a^{12}, S_2 = a^9, S_3 = 0, S_4 = a^3, S^5 = 1, S_6 = 0$.

Необходимо вычислить полином $s(x)$ по алгоритму Берлекемпа–Месси.

1. Начальные условия: $s(x) = 1, r = 0, L = 0, B(x) = 1$;
2. *Первая итерация.* Установка индекса r = 1.

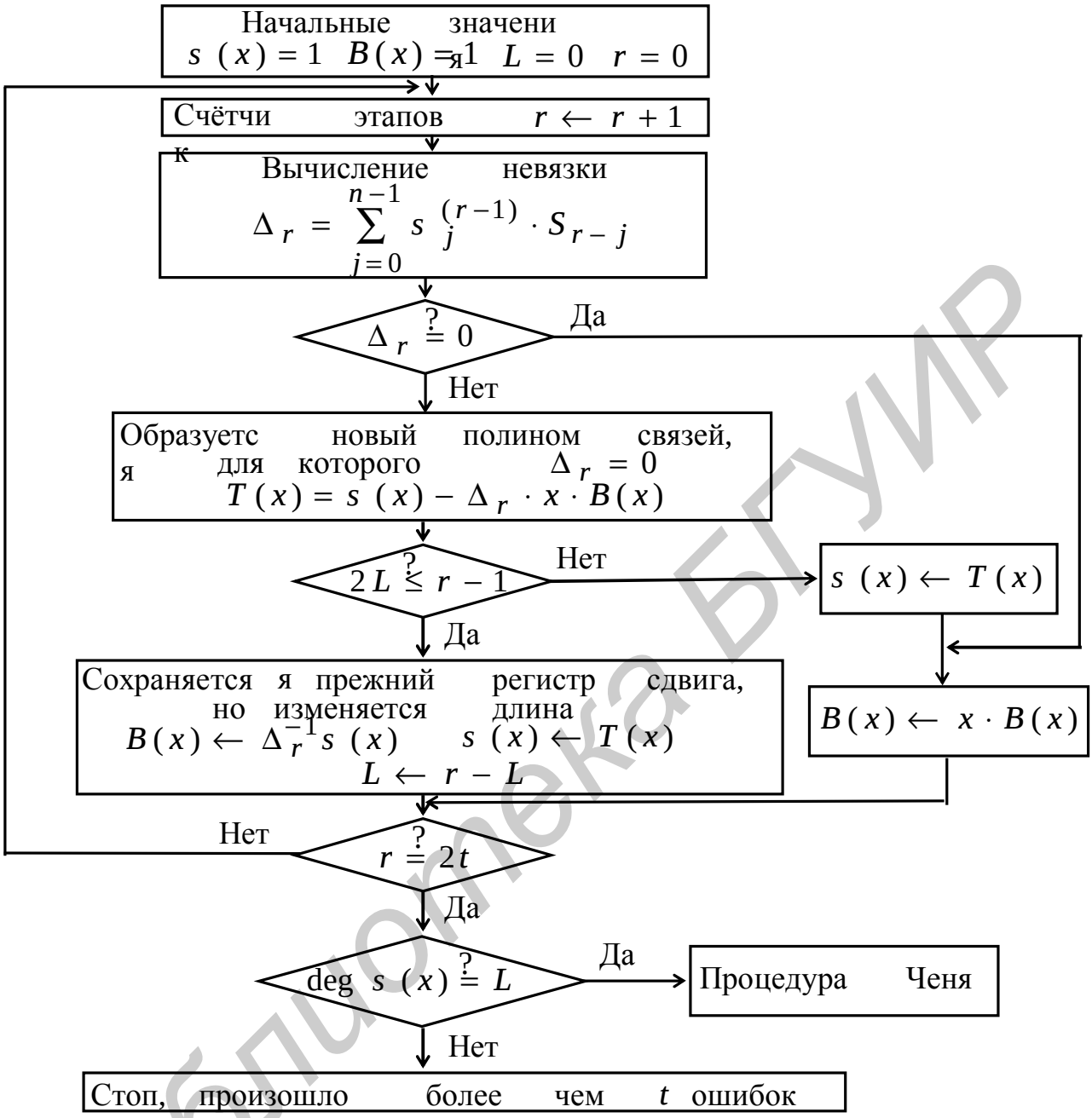


Рис. 5.1

3. Вычисление невязки: $\Delta_{r=1} = \sum_{j=0}^{0} s_j \cdot S_{r-j} = s_0 \cdot S_1 = 1 \cdot a^{12} = a^{12} \neq 0$.

4. Изменение обратных связей полинома, поскольку $\Delta_{r=1} \neq 0$;

$$T(x) = s(x) - \Delta_r \cdot xB(x) = 1 + a^{12} \cdot x \cdot 1 = 1 + a^{12} \cdot x \; ; 2L \leq r - 1 \Rightarrow (0 = 0).$$

5. Проверка условия $2L \leq r - 1? \rightarrow (0 = 0)$.

6. Вычисление полинома $B(x)$ и сохранение результата $s(x)$:

$$B(x) = \Delta_r^{-1} \cdot s(x) = a^{-12} \cdot 1 = a^3; \qquad s(x) \leftarrow T(x) = 1 + a^{12} \cdot x.$$

7. Проверка условий: $L = r - L? \rightarrow 1 - 0 = 1;$. $r = 2t? \rightarrow 1 \neq 6$.

8. *Вторая итерация*. Изменение индекса $r \leftarrow r + 1 = 2$.

9. Вычисление невязки

$$\Delta_{r=2} = \sum_{j=0}^{1} s_j \cdot S_{r-j} = s_0 S_2 + s_1 \cdot S_1 = a^9 + a^{12} \cdot a^{12} == a^9 + a^9 = 0.$$

10. Изменение полинома $B(x) \leftarrow x \cdot B(x) = x \cdot a^3$, т. к. $\Delta_2 = 0$.

11. Проверка условия: $r = 2t? \rightarrow 2 \neq 6$.

12. *Третья итерация*. Изменение индекса $r \leftarrow r + 1 = 3$;

13. Вычисление невязки

$$\Delta_{r=3} = \sum_{j=0}^{1} s_j \cdot S_{r-j} = s_0 \cdot S_3 + s_1 \cdot S_2 = 0 + a^{12} \cdot a^9 = a^6 \neq 0.$$

14. Формирование нового полинома

$$T(x) = s(x) - \Delta_2 \cdot x \cdot B(x) = 1 + a^{12} \cdot x - a^6 \cdot x \cdot (x \cdot a^3) = 1 + a^{12} \cdot x + a^9 \cdot x^2.$$

15. Проверка условия: $2L \leq r - 1? \rightarrow 2 = 2$.

16. Изменение полинома $B(x) = \Delta_r^{-1} \cdot s(x) = a^{-6} \cdot (1 + a^{12} \cdot x) = a^9 + a^6 \cdot x$.

17. Сохранение результата $s(x) \leftarrow T(x) = 1 + a^{12} \cdot x + a^9 \cdot x^2$.

18. Увеличение длины регистра $L \leftarrow r - L = 2$.

19. Проверка выполнение условия: $r = 2t? \rightarrow 3 \neq 6$.

20. *Четвертая итерация*. Изменение индекса $r = 4$.

21. Вычисление невязки

$$\Delta_{r=4} = \sum_{j=0}^{2} s_j \cdot S_{4-j} = s_0 \cdot S_4 + s_1 \cdot S_3 + s_2 \cdot S_2 = a^3 + a^{12} \cdot 0 + a^9 \cdot a^9 = 0 \; .$$

22. Изменение полинома $B(x) \leftarrow x \cdot B(x) = x \cdot a^9 + a^6 \cdot x^2$.

23. Проверка условия: $r = 2t$? $\rightarrow 4 \neq 6$.

24. На следующих итерациях для $r = 5$ и $r = 6$ получаем $\Delta_{r=5} = \Delta_{r=6} = 0$.

Ответ: $s(x) = 1 + a^{12} \cdot x + a^{9} \cdot x^2$.

Корнями данного полинома будут $l_1 = a^{13}$, $l_2 = a^{8}$. Ошибки будут находиться на второй $e_1 \rightarrow l_1^{-1} = a^{-13} = a^{2}$ и седьмой $e_2 \rightarrow l_2^{-1} = a^{-8} = a^{7}$ позициях.

Процедура Ченя. Для вычисления инверсии корней полинома ошибок удобно использовать специальный алгоритм. Воспользуемся тем свойством, что сумма $\sum_{k=0}^{n} s_k \cdot a^{-ik} = 0$ только в том случае, если символ, располагающийся на $(n-i)$-й позиции, оказывается ошибочным. Если определить $s_k^{(i)} = s_k \cdot a^{-ik}$, тогда $s_k^{(i-1)} = s_k^{(i)} \cdot a^{k}$, а $s(a^{-i}) = \sum_{k=0}^{n} s_k^{(i)}$. В итоге получается простая переборная схема вычислителя корней:

Алгоритм Форни. Позволяет оценить величины ошибок Q_i. Известно, что $S(x) = \sum_{j=1}^{n=2t} S_j \cdot x^j = \sum_{j=1}^{2t} \sum_{i=1}^{n} Q_i \cdot X_i^j \cdot x^j$. Полином локатора ошибок имеет вид $s(x) = \sum_{j=0}^{n} s_j \cdot x^j = \prod_{l=1}^{n} (1 - x \cdot X_l)$, где X_l^{-1} - корни полинома. Вычисления по алгоритму ведутся в следующем порядке.

1. Составляется ключевое уравнение:

$$w(x) = (1 + S(x)) \cdot s(x) \bmod x^{2t+1}$$

2. Вычисляется оценка ошибки, после подстановки в ключевое уравнение выражения для S(x), σ(x) и решение его относительно Q_i,

$$Q_l = \frac{-X_l \cdot w(X_l^{-1})}{s'(X_l^{-1})} \bmod x^{2t+1},$$ где s' - формальная производная полинома.

Пример. Троичный БЧХ-код (7,3) формируется над полем $GF(8)=GF(2^3)$, $f(x)=x^3+x+1$, для которого $a^0\to 100$; $a^1\to 010$; $a^2\to 001$; $a^3\to 110$; $a^4\to 011$; $a^5\to 111$; $a^0\to 101$.

Порождающий полином имеет вид

$$g(x)=(x-\underbrace{a+x^2+a\cdot x}_{m_a})+a^3\cdot(x-\underbrace{a^2}_{m_{a^2}})\cdot(x-\underbrace{a^3}_{m_{a^3}})\cdot(x-\underbrace{a^4}_{m_{a^4}})=$$

$$=x^4+a^3\cdot x^3+x^2+a\cdot x+a^3.$$

Предположим, что на вход декодера поступает следующее слово:

$$y(x)=a^2\cdot x^6+a^2\cdot x^4+x^3+a^5\cdot x^2,$$

тогда $S_1=a^6$; $S_2=a^3$; $S_3=a^4$; $S_4=a^3$.

Алгоритм Берлекемпа–Месси дает полином локатора ошибок (табл.5.1).

Таблица 5.1

k	S_k	$s'(x)$	Δ_k	L	$B(x)$
0	-	1	-	0	x
1	a^6	$1+a^6$	$S_1-0=a^6$	1	$a^2\cdot x$
2	a^3	$1+a^4\cdot x$	$S_2-a^5=a^2$	1	$a\cdot x^2$
3	a^4	$1+a^4\cdot x+a^6\cdot x^2$	$S_3-1=a^5$	2	$a^2\cdot x+a^6\cdot x^2$
4	a^3	$1+a^2\cdot x+a\cdot x^2$	$S_4-a^4=a^6$	-	-

Из таблицы получаем следующие уравнения:

$$s(x)=1+a^2\cdot x+a\cdot x^2,\quad s'(x)=a^2+2a\cdot x=a^2. \qquad (5.22)$$

Решение уравнения (5.22) даёт корни: $x_1=a^3$, $x_2=a^5$, где 3 и 5 – это номера позиций ошибок. Ключевое уравнение запишется как

$$w(x)=(1+a^2\cdot x+a\cdot x^2)\cdot(1+a^6\cdot x+a^3\cdot x^2+a^4\cdot x^3+a^3\cdot x^4)\bmod x^{(2t+1)=5}=$$

$$=1+x+a^3\cdot x^2;$$

Откуда

$$Q_l=\frac{-X_l\cdot w(X_l^{-1})}{s'(X_l^{-1})}=\frac{-X_l\cdot(1+X_l^{-1}+a^3\cdot X_l^{-2})}{a^2}=X_l\cdot a^5+a^5+a\cdot X_l^{-1};$$

$$Q_3=Q_3(a^3)=a^3\cdot a^5+a^5+a\cdot a^4=a+a^5+a^5=a;$$

$$Q_5=Q_3(a^5)=a^5\cdot a^5+a^5+a\cdot a^{-5}=a^3+a^5+a^3=a^5.$$

Оценка вектора ошибки равна $e(x) = a \cdot x^3 + a^5 \cdot x^5$. Кодовое слово $\hat{c}(x)$ декодируется как

$$y(x) - \hat{e}(x) = a^2 \cdot x^6 + a^2 \cdot x^4 + x^3 + a^5 \cdot x^2 - a \cdot x^3 - a^5 \cdot x^5 =$$
$$= a^2 \cdot x^6 + a^5 \cdot x^5 + a^2 \cdot x^4 + \underbrace{a^3 \cdot x^3 + a^5 \cdot x^2}_{1-a} = \hat{c}(x)$$

Режим исправления ошибок и стирания. В режиме стирания сомнительных случаев фиксируются не сами оценки принятых символов, а их местоположение и им присваивается статус стёртого символа. Суть исправления и стирания состоит в том, что после декодирования можно провести восстановление стёртых символов, используя алгоритмы интерполяции. Если при декодировании использовать l стёртых символов, тогда два кода будут отличаться друг от друга по меньшей мере на $(d-l)$ позиций, где d – кодовое расстояние. Тогда в дополнение к стиранию можно будет исправлять $t_m = \lfloor (d-l-1)/2 \rfloor$ ошибок, где $\lfloor x \rfloor$ - это целая часть числа *x*. Код может исправлять все комбинации из n ошибок и l стираний в канале, для которого $2n + l < d$.

Алгоритм исправления стираний. Предположим, что выполнено f стираний при приеме кодового слова, в котором имеется *n* ошибок.

Обозначим локаторы ошибок $X_1 = a^{i_1}, X_2 = a^{i_2}, ..., X_n = a^{i_n}$, а стираний – как $Y_{c,1} = a^{j_1}, Y_{c,2} = a^{j_2}, ..., Y_{c,f} = a^{j_f}$. Декодирование ведется в следующем порядке.

1. Вычисляется полином локаторов стираний:

$$\Gamma(x) = \prod_{l=1}^{f} (1 - Y_{c,l} \cdot x).$$

2. В декодируемом векторе заменяются символы с координатами стираний на нулевые символы. Для нового вектора находится полином синдрома стираний s(x).

3. Определяется модифицированный полином синдрома:

$$SE(x) = (\Gamma(x)[1 + s(x)] - 1) \bmod x^{2t+1}.$$

4. Вычисляется полином локаторов ошибок $s(x)$ с применением алгоритма Берлекемпа–Месси и значения модифицированного полинома $SE_i, i = f+1, ..., 2t$.

5. Определяются корни уравнения $s(x)=0$ и координаты ошибок

6. Составляется ключевое уравнение

$$w(x) = s(x)[1 + SE(x)] \bmod x^{2t+1}$$

и определяется полином локаторов ошибок-стираний $y(x) = s(x)\Gamma(x)$.

7. Оцениваются значения ошибок и стираний. Значения ошибок вычисляются по формуле

$$Q_{i_k} = \frac{-X_k w(X_k^{-1})}{y'(X_k^{-1})}$$, где y' - формальная производная.

Значения стираний вычисляются по формуле $F_{i_k} = \frac{-Y_k w(Y_k^{-1})}{y'(Y_k^{-1})}$.

5.3. Коды Рида–Соломона

Пусть имеется поле F размерностью q, $n < q$, $(g_0, \ldots, g_{n-1})$ элементов поля, а также информационное сообщение $a = (a_0,\ldots,a_{k-1})$, $a(x) = \sum_{i=0}^{k-1} a_i x^i$. Тогда для $k \le n$ код Рида–Соломона (РС) $(n, k)_q$ определяется как

$$\mathbf{C}(a) = C_0, \ldots, C_{n-1}, \text{ где } C_j = a(g_j).$$

Если $n = q - 1$, $g_i = \alpha^i$, то $c_j = \sum_{j=0}^{k-1} a_j (a^i)^j$. Можно дать и другое определение кода (РС). Пусть имеется поле F размерностью q с примитивным элементом α, $n = q - 1$, $k \le n$ и полином $g(x) = (x - \alpha)(x - \alpha^2)\ldots(x - \alpha^{n-k})$. Тогда для информационного сообщения $\mathbf{a} = (a_0, \ldots, a_{k-1})$, $a(x) = \sum_{i=0}^{k-1} a_i x^i$ код РС задается как последовательность $\mathbf{c} = (c_0, \ldots, c_{n-1})$, элементы c_j которой определяются как коэффициенты при степенях x^j полинома $g(x)\,a(x)$.

Поле символов совпадает с полем локаторов ошибок. Если a - примитивный элемент поля и длина кода $n = q^m - 1 = q - 1$, то минимальный многочлен над полем $GF(q)$ элемента $b \in GF(q)$ равен

$$m_b(x) = x - b. \qquad (5.23)$$

Порождающий полином кода РС равен

$$g(x) = (x - a)\cdot(x - a^2)\cdot\ldots\cdot(x - a^{2t}), \qquad (5.24)$$

где t – число исправляемых ошибок, $\deg d(x) = 2t$.

Если это так, то $r = n - k = 2t$, где r - число проверочных разрядов.

В обобщенном случае

$$g(x) = (x - a^{j_0})\cdot(x - a^{j_0+1})\cdot\ldots\cdot(x - a^{j_0+2t-1}), \qquad (5.25)$$

где $j_0 = 0,1,2,\ldots$ - целое число.

Пример. Найдём $g(x)$ для кода РС (15,11) над полем $GF(16)$ для $j_0 = 1$, $n - k = 4 = 2t$ при $t = 2$:

$$g(x) = (x - a)\cdot(x - a^2)\cdot(x - a^3)\cdot(x - a^4) = x^4 + a^{13}\cdot x^3 + a^6\cdot x^2 + a^3\cdot x + x^{10}.$$

Информационный многочлен представляет собой 11 символов из поля $GF(16)$. Каждый символ несёт информацию в 4 бита, т.е. эквивалентная ёмкость информационного сообщения будет 44 бита.

Пример. Построим порождающий полином для РС кода (7,3) для $t = 2$ в поле $GF(8)$ при $j_0 = 4$. Тогда

$$g(x) = (x - a^4)\cdot(x - a^5)\cdot(x - a^6)\cdot(x - a^0) =$$
$$= x^4 + (z^2 + 1)\cdot x^3 + (z^2 + 1)\cdot x^2 + (z + 1)\cdot x + z.$$
$$\underset{a^6}{123} \qquad \underset{a^6}{123} \qquad \underset{a^3}{123}$$

Здесь $(z^2 + 1)$, $(z + 1)$ и z - формальная запись элементов поля. Зададим информационный полином в виде $a(x) = (z^2 + z)\cdot x^2 + x + (z + 1)$. Информационная
$$\underset{a^4}{123} \qquad \underset{a^3}{123}$$
ёмкость полинома $a(x)$ равна 9 бит. Полином кода кодируется как

$$c(x) = a(x)\cdot g(x) = (a^4\cdot x^2 + x + a^3)\cdot(x^4 + a^6\cdot x^3 + a^6\cdot x^2 + a^3\cdot x + a) =$$
$$= a^4\cdot x^6 + a\cdot x^5 + a^6\cdot x^4 + 0\cdot x^3 + 0\cdot x^2 + a^5\cdot x + a^4.$$

Информационная ёмкость $c(x)$ - 21 бит.

Код РС имеет минимальное расстояние $d_{\min} = n - k + 1$ и лежит на границе Синглтона (MDS-код). Для каждой простой степени q и каждой пары положительных чисел k, n таких, что $k \le n \le q$, существует код РС с параметрами (n, k, $n - k + 1$).

Декодирование кода РС. Реализация декодера максимального правдоподобия кода РС большой значности представляет собой сложную задачу. В теории вычислительной сложности считается, что такое декодирование относится к классу задач с NP-сложностью.

Декодирование кода РС можно вести по алгоритмам декодирования кода БЧХ, если рассматривать код РС как частный случай кода БЧХ. В качестве иллюстрации рассмотрим пример декодирования со стиранием кода РС.

Пример. Декодируется код РС (7,3), построенный над полем $GF(8)$.

Принимаемый вектор равен

$$y(x) = \alpha^4 x^6 + \alpha^5 x^5 + \alpha^2 x^4 + x^3 + \alpha^6 x^2 + \alpha^5 x + \alpha^6 =$$
$$= y_6 x^6 + y_5 x^5 + y_4 x^4 + y_3 x^3 + y_2 x^2 + y_1 x + y_0 .$$

Приемник выдал стирания на позициях $j_1 = 1$, $j_2 = 6$, что позволяет записать полином стираний как $\mathrm{er}(x) = \mathrm{er}_1 x + \mathrm{er}_2 x^2$. Алгоритм декодирования:

1. $\Gamma(x) = (1 - \alpha x)(1 - \alpha^6 x) = 1 + \alpha^5 x + x^2$;

2. Заменяем в y(x) символы на 1 и 6 позициях нулями:

$y_c(x) = \alpha^5 x^5 + \alpha^2 x^4 + x^3 + +\alpha^6 x^2 + \alpha^6$, после чего вычисляем синдром

$$S_1 = y_c(\alpha) = \alpha;\ S_2 = y_c(\alpha^2) = \alpha;\ S_3 = y_c(\alpha^3) = = \alpha;\ S_4 = y_c(\alpha^4) = \alpha^3;$$

$$s(x) = \alpha\, x + \alpha\, x^2 + \alpha\, x^3 + \alpha^3\, x^4.$$

3. Модифицируем полином синдрома

$$\mathrm{SE}(x) = [(1+ \alpha^5\, x + x^2)(\, \alpha\, x + \alpha\, x^2 + \alpha\, x^3 + \alpha^3 + \; + x^4) - 1] \bmod x^5 =$$
$$= \alpha^6\, x + \alpha^4\, x^2 + \alpha^6\, x^3 + \alpha^2\, x^4.$$

4. Применяя алгоритм Берлекемпа–Месси, получим выражение для полинома локаторов ошибок $s(x) = 1 + \alpha^3\, x$. Локатор ошибки $X_1 = \alpha^3$. Ошибка расположена на третьей позиции.

5. Ключевое уравнение равно

$$w(x) = s(x)[1 + \mathrm{SE}(x)] \bmod x^5 =$$
$$= (1 + \alpha^3\, x)\,(1 + \alpha^6\, x + \alpha^4\, x^2 + \alpha^6\, x^3 + \alpha^2\, x^4) \bmod x^5 = 1 + \alpha^4\, x + \alpha\, x^2 + \alpha^2\, x^3.$$

Полином $y(x) = (1 + \alpha^2\, x + \alpha^3\, x^2 + \alpha^3\, x^3)$.

Формальная производная $y'(x) = (\, \alpha^2 + \alpha^3\, x^2\,)$.

6. Вычисляем значения ошибок и стираний

$$Q_3 = \frac{a^3 w(a^4)}{y'(a^4)} = a^6, \quad F_1 = \frac{a w(a^6)}{y'(a^6)} = a^5, \; F_6 = \frac{a^6 w(a)}{y'(a)} = a^4.$$

5.4. Альтернантные коды

Альтернантные коды представляют собой большое и мощное семейство кодов, которые строятся из кодов БЧХ (РС) таким образом, чтобы при фиксированной скорости получить большое минимальное кодовое расстояние.

Выберем код Рида–Соломона на полем $GF(q^m)$ с конструктивным расстоянием $d = 2t + 1$, $n = q^m - 1$. Выберем и зафиксируем n–мерный вектор **h** с ненулевыми компонентами и назовем его шаблоном. Альтернантный код состоит из всех $GF(q)$–значных векторов **c**, таких, что вектор с компонентами $h_i\, c_i$, $i = 0, \ldots, n - 1$ является словом кода Рида –Соломона.

Другое понятие альтернантного кода можно получить из определения обобщенного кода Рида–Соломона (ОРС). В поле $GF(q^m)$ зададим вектор $\mathbf{a} = (\alpha_1, \ldots, \alpha_n)$, где α_i – различные элементы поля $GF(q^m)$, и $\mathbf{v} = (v_1, \ldots, v_n)$, где v_i – ненулевые (но необязательно различные) элементы из $GF(q^m)$. Тогда обобщенный код РС $RS_{k_0}(a, \mathbf{v})$ состоит из всех векторов вида

$$(v_1 f(\alpha_1), v_2 f(\alpha_2), \ldots, v_n f(\alpha_n)),$$

где $f(z)$ – любой многочлен с коэффициентами из $GF(q^m)$, степень которого не превосходит k_0. Код ОРС является кодом с параметрами $(n, k_0, r + 1)$, его проверочная матрица равна

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \dots & \dots & \dots & \dots \\ a_1^{r-1} & a_2^{r-1} & \dots & a_n^{r-1} \end{bmatrix} \begin{bmatrix} g_1 & & & \mathbf{0} \\ & g_2 & & \\ & & \mathbf{O} & \\ \mathbf{0} & & & g_n \end{bmatrix} = \mathrm{AY}, \qquad (5.26)$$

где вектор g = (g_1, …, γ_n) такой, что $g \in GF(q^m)$; $g_i \neq 0$, причем двойственный ОРС код с k_0 информационными символами $RS_{k_0}^{\perp}(a, \mathbf{v})$ равен ОРС коду $RS_{n-k_0}(a, g)$.

Альтернантный код $A(a, g)$ состоит из всех слов кода $RS_{k_0}^{\perp}(a, \mathbf{v})$ таких, что их компоненты лежат в поле $GF(q)$; иными словами $A(a, g)$ равен ограничению кода $RS_{k_0}(a, \mathbf{v})$ на подполе $GF(q)$. Код $A(a, g)$ состоит из всех векторов **x** над $GF(q)$, удовлетворяющих равенству $\mathbf{xH}^{\mathrm{T}} = \mathbf{0}$ для матрицы **H**, задаваемой условием (5.26).

$A(a, g)$ - линейный код над $GF(q)$ с параметрами:

- длина n, размерность $k \geq n - m\,r$;
- минимальное расстояние $d \geq r + 1$.
- двойственным альтернантному коду $A(a, g)$ является код

$$Tr_m(RS_{n-k_0}(a, g)) = Tr_m(RS_{k_0}^{\perp}(a, \mathbf{v})).$$

Здесь $Tr_m(z)$ - след произвольного кода z над $GF(q^m)$. Код состоит из всех векторов вида ($Tr_m(c_1)$, $Tr_m(c_2)$, …, $Tr_m(c_n)$), где $(c_1,\dots,c_n) \in z$.

Существуют длинные альтернантные коды, лежащие на границе Варшамова–Гильберта. Важнейшими подклассами альтернантных кодов являются коды БЧХ, коды Гоппы, коды Сривэставы.

Пример. Выберем вектор a = (1, α, α^2, … , α^6), где α - примитивный элемент поля $GF(2^3)$. Матрица элементов поля равна

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & a & a^2 & a^3 & a^4 & a^5 & a^6 \end{bmatrix}.$$

Зададим g = (1, α, α^2, … , α^6). Тогда проверочная матрица кода $A(a, g)$ равна

$$\mathbf{H} = \begin{bmatrix} 1 & a & a^2 & a^3 & a^4 & a^5 & a^6 \\ 1 & a^2 & a^4 & a^6 & a & a^3 & a^5 \end{bmatrix}.$$

Заменяя каждый элемент матрицы соответствующим двоичным вектором длины 3 и устраняя избыточность строк, получим

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

В этом случае $A(a, g)$ представляет собой (7, 4, 3) –код Хэмминга. Роль вектора g сводится к уменьшению минимального расстояния и увеличению информационных символов.

Другим примером альтернантных кодов являются коды БЧХ. Действительно, проверочная матрица кода БЧХ в общем виде задается равенством

$$\mathbf{H} = \begin{bmatrix} 1 & a^{b} & a^{2b} & \mathbf{L} & a^{(n-1)b} \\ 1 & a^{b+1} & a^{2(b+1)} & \mathbf{L} & a^{(n-1)(b+1)} \\ \text{M} & \text{M} & \text{M} & \mathbf{O} & \text{M} \\ 1 & a^{b+d-2} & \mathbf{L} & \mathbf{L} & a^{(b+d-2)(n-1)} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & a & a^{2} & \ldots & a^{n-1} \\ 1 & a^{2} & a^{4} & \ldots & a^{2(n-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & a^{d-2} & \ldots & \ldots & a^{(d-2)(n-1)} \end{bmatrix} \begin{bmatrix} 1 & & & & \mathbf{0} \\ & a^{b} & & & \\ & & a^{2b} & & \\ & & & \mathbf{O} & \\ \mathbf{0} & & & & a^{(n-1)b} \end{bmatrix},$$

и, следовательно, код является альтернантным.

Коды Гоппы. Коды Гоппы являются наиболее интересным подклассом альтернантных кодов. Они определяются в терминах многочленов Гоппы $G(z)$. Коды Гоппы обладают тем свойством, что кодовое расстояние $d \geq \deg(G(z) + 1$.

При определении кода Гоппы длиной n из поля $GF(q)$ использует два понятия. Первое — это многочлен с коэффициентами из поля $GF(q^m)$ для некоторого фиксированного m, называемый многочленом Гоппы. Второе – подмножество L = $= \{\alpha_1, \ldots, \alpha_n\}$ элементов из $GF(q^m)$ таких, что $G(\alpha_i) \neq 0$ для всех $\alpha_i \in$ L. Обычно в качестве L выбирается подмножество всех элементов поля $GF(q^m)$, которые не являются корнями многочлена $G(z)$.

Каждому вектору **c** = $(c_1, \ldots, c_n)$ над $GF(q)$ поставим в соответствие рациональную функцию

$$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z - a_i}.$$

Код Гоппы Г(LG) состоит из всех векторов **c,** таких, что $R_c(z) \equiv 0 \bmod G(z)$ или, что эквивалентно, таких, что $R_c(z) = 0$ в кольце многочленов $GF(q^m)[z]/G(z)$.

Свойства кода

Код Г(LG) линейный с параметрами:

- длина $n = |L|$;
- размерность $k \geq n - m\,r$, $r = \deg G(z)$;
- минимальное расстояние $d \geq r + 1$; в двоичном случае, если у многочлена $G(z)$ нет кратных корней $d \geq 2\,r + 1$.

Код Г(LG) – альтернантный код $A(a, g)$, где $g_i = G^{-1}(\alpha_i)$. Двойственный код $\Gamma(L G)^{\perp} = Tr_m\,(RS_r(\alpha,\ g))$. Существуют длинные коды Гоппа, лежащие на границе Варшамова-Гильберта.

Проверочная матрица Г(LG)- кода равна

$$\mathbf{H} = \begin{bmatrix} G^{-1}(a_1) & \ldots & G^{-1}(a_n) \\ a_1 G^{-1}(a_1) & \ldots & a_n G^{-1}(a_n) \\ \ldots & \ldots & \ldots \\ a_1^{r-1} G^{-1}(a_1) & \ldots & a_n^{r-1} G^{-1}(a_n) \end{bmatrix}.$$

Пример. Выберем $G(z) = z^2 + z + 1$; $L = GF(2^3) = \{0, 1, \alpha, \ldots, \alpha^6\}$, где α - примитивный элемент поля, $q = 2$, $q^m = 8$. Для всех $b \in GF(2^3)$ выполняется условие $G(b) \neq 0$, так как корни многочлена $z^2 + z + 1$ лежат в полях $GF(2^2)$, $GF(2^4)$, $GF(2^6)$,… и не лежат в поле $GF(2^3)$. Получаем неприводимый код Гоппы для $n = |L| = 8$, размерностью $k \geq 8 - 2\ 3 = 2$ и минимальным расстоянием $d \geq 2\ 2 + 1 = 5$. Согласно определению проверочная матрица равна

$$\mathbf{H} = \begin{bmatrix} G^{-1}(0) & G^{-1}(1) & \ldots & G^{-1}(a^6) \\ 0\,G^{-1}(0) & G^{-1}(1) & \ldots & a^6 G^{-1}(a^6) \end{bmatrix} = \begin{bmatrix} 1 & 1 & a^2 & a^4 & a^2 & a & a & a^4 \\ 0 & 1 & a^3 & a^6 & a^5 & a^5 & a^6 & a^3 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Кодовыми словами (8, 2, 5) кода Гоппы являются

$$\left\langle \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{matrix} \right\rangle.$$

Примитивные коды БЧХ в узком смысле являются частным случаем кодов Гоппы. Действительно, выберем G(z) = z^r и L ={1, α, … , α^{n-1}}, где $n = q^m - 1$, а α - примитивный элемент поля GF(q^m). Тогда получаем

$$\mathbf{H} = \begin{bmatrix} 1 & a^{-r} & a^{-2r} & \dots & a^{-(n-1)r} \\ 1 & a^{-(r-1)} & a^{-2(r-1)} & \dots & a^{-(n-1)(r-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & a^{-1} & a^{-2} & \dots & a^{-(n-1)} \end{bmatrix},$$

что дает проверочную матрицу кода БЧХ.

Заметим, что для построения двоичных кодов, исправляющих t ошибок, надо выбрать G(z) = z^{2t} и L = $GF(2^m)$.

Декодирование кодов Гоппы. Определим множество индексов $\{i_1, i_2,\dots,i_l\}$ компоненты c_i вектора **c**, равных единицы. Зададим функцию

$$f_c(z) = (z - a_{i_1})(z - a_{i_2})\dots(z - a_{i_l}).$$

Тогда рациональную функцию можно представить в виде

$$R_c(z) = \frac{f_c'(z)}{f_c(z)},$$

где $f_c'(z)$ - формальная производная f_c.

Пусть **c** и **e** – соответственно переданный кодовый вектор и вектор ошибок. Принятый вектор **y** = **c** + **e** = $(y_1,\dots,y_n)$ и для него выполняются следующие соотношения:

$$\frac{f_y'(z)}{f_y(z)} \equiv \frac{f_e'(z)}{f_e(z)} \bmod G(z);$$

синдром $s(z)$ может быть найден по принятому вектору

$$s(z) = \sum_{i=1}^{n} y_i \frac{G(z) - G(a_i)}{z - a_i} G^{-1}(a_i) \equiv \frac{f_y'(z)}{f_y(z)} \equiv \frac{f_e'(z)}{f_e(z)} \bmod G(z).$$

Задача заключается в том, чтобы найти по заданному многочлену $s(z) \neq 0$ такой многочлен $f_e(z)$, что $f_e'(z) \equiv f_e(z)s(z) \bmod G(z)$. Решение будем искать в виде

$$f(z) = a_0 + z j(z), \quad \deg j < t, \quad a_0 \in GF(2),$$

где t – определяет количество исправляемых ошибок, причем deg(G(z)) = 2t.

Алгоритм декодирования

1. По принятому вектору находится синдром s(z). Если синдром равен нулю, то ошибок нет, в противном случае переходят к шагу 2.

2. Для каждого $0 \le i < t$ находится остаток $\sum_{j=0}^{t-1} h_{ij} z^j$ от деления $(1+zs(z))z^i$ на G(z).

3. Строится матрица $\mathbf{D} + \mathbf{T}_s$, где $\mathbf{D}$ – матрица размером (t × t), у которой диагональные элементы с четными номерами равны 1, а остальные элементы равны 0; $\mathbf{T}_s$ – матрица размером (t × t), (i, j)-й элемент который равен $h_{j-1,i-1}$. Если полученная матрица является вырожденной, то к n-й компоненте прибавляется единица и осуществляется переход к шагу 1. Если полученная матрица оказывается невырожденной, то находится решение j уравнения $(\mathbf{D} + \mathbf{T}_s)\ \mathrm{j}^{\mathrm{T}} = \mathbf{s}^{\mathrm{T}}$ и далее определяется многочлен φ(z).

4. Подставляя в ϕ(z) = 1 + zφ(z) последовательно элементы последовательности < L - {0} >, находят корни $a_{i_1}, a_{i_2}, ..., a_{i_v}$ уравнения ϕ(z) = 0. Компоненты принятого вектора $i_1, i_2, \ldots, i_v$ заменяются на противоположные.

Пример. Пусть m = 4 и L = $GF(2^4)$. Так как в этом случае многочлен G(z) не должен иметь корней среди элементов $GF(2^4)$, то он должен выбираться из многочленов, неприводимых над $GF(2^4)$, и их произведений. Пусть α - примитивный элемент $GF(2^4)$, а $x^4 + x + 1$ – его минимальный многочлен. Упорядочим элементы поля как L = $\{\alpha^1, \alpha^2, \ldots, \alpha^{15}, 0\}$. Так как элемент α^3 имеет порядок 5 и его минимальный многочлен равен $x^4 + x^3 + x^2 + x + 1$, то сумма корней $Tr_2(\alpha^3)$ равна 1. Следовательно, многочлен $z^2 + z + \alpha^3$ является неприводимым над $GF(2^4)$ и его можно использовать в качестве G(z). С помощью таблиц поля определим $G(\alpha) = \alpha^2 + \alpha + \alpha^3 = \alpha^{11}$, $G^{-1}(\alpha) = \alpha^4$, $G(\alpha^2) = \alpha^4 + \alpha^2 + \alpha^3 = \alpha^{12}$, $G^{-1}(\alpha^2) = \alpha^3$,….., $(1 + \alpha)G^{-1}(\alpha) = \alpha^8$, $(1 + \alpha^2)G^{-1}(\alpha^2) = \alpha^{11}$, …. . В этом случае проверочная матрица имеет вид

$$\mathbf{H} = \begin{bmatrix} a^4 & a^3 & a^9 & a^4 & a & a^8 & a^6 & a^3 & a^6 & a & a^2 & a^2 & a^8 & a^9 & a^{12} & a^{12} \\ a^8 & a^{11} & a^8 & a^5 & a^{11} & a^6 & 1 & a^5 & a^{13} & a^6 & a^{14} & a^{13} & a^{14} & a^{12} & 0 & a^{12} \end{bmatrix}.$$

В двоичном виде проверочная матрица запишется как

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Так как ранг этой матрицы равен 8, то код имеет 8 проверочных и 8 информационных символов. Предположим, что ошибки появились на 1-й и 5-й позициях. Тогда

$$s(z) = (\alpha + \alpha^4)\, z + \alpha^8 + \alpha^{11} = \alpha^7 + z,$$

$$1 + z\, s(z) = z^2 + \alpha^7 z + 1 \equiv (\alpha^7 + 1)\, z + \alpha^3 + 1 \pmod{G(z)} \equiv \alpha^{14} + \alpha^9 z \pmod{G(z)},$$

$$(1 + zs(z))z \equiv \alpha^9 z^2 + \alpha^{14} z \equiv \alpha^9 (z + \alpha^3) + \alpha^{14} z \equiv \equiv (\alpha^9 + \alpha^{14}) z + \alpha^{12} \equiv \alpha^{12} + \alpha^4 z \pmod{G(z)},$$

$$\mathbf{D} + \mathbf{T}_S = \begin{bmatrix} a^{14} & a^{12} \\ a^9 & 1 + a^4 \end{bmatrix} = \begin{bmatrix} a^{14} & a^{12} \\ a^9 & a \end{bmatrix},$$

вектор синдрома ошибок **s** = (α^7, 1), вектор j = (φ_0, φ_1). Уравнения декодирования имеют вид

$$\alpha^{14}\varphi_0 + \alpha^{12}\varphi_1 = \alpha^7,\ \alpha^9\varphi_0 + \alpha\varphi_1 = 1,$$

$$\varphi_0 = (\alpha^8 + \alpha^{12})/(1 + \alpha^6) = \alpha^{11},\ \varphi_1 = (\alpha^{14} + \alpha)/(1 + \alpha^6) = \alpha^9.$$

Следовательно, $\phi(z) = 1 + \alpha^{11} z + \alpha^9 z^2$. Легко проверить, что $\phi(\alpha) = \phi(\alpha^5) = 0$.

5.5. Асимптотические кодовые конструкции

Концепция поиска «хороших» кодов предполагает изучение асимптотического поведения их параметров. Для этого необходимо принять, что имеется бесконечное множество кодов $C = \{(n_i, k_i, d_i)\ ;\ i = 1, \ldots \infty\}$ и $\lim_{i \to \infty} \{n_i\} = \infty$.

Определим для бесконечного множества кодов понятия скорости передачи информации R(C) и относительного кодового расстояния δ(C) как

$$R(C) = \lim_{i \to \infty} \left\{ \frac{k_i}{n_i} \right\}, \qquad d(C) = \lim_{i \to \infty} \left\{ \frac{d_i}{n_i} \right\}.$$

Семейство кодов считается асимптотически хорошими, если $R(C)$, $\delta(C) > 0$.

Обратимся к асимптотической интерпретации граничных соотношений. Граница Хэмминга свидетельствует, что для бинарных кодов $2^k Vol(\frac{d-1}{2}, n) \le 2^n$. Используя аппроксимации $\frac{d-1}{2} \approx \frac{d(C)n}{2}$ и для объема сферы $Vol(pn, n) \approx 2^{H(p)n}$, получаем

$$R(C) + H(\frac{d(C)}{2}) \le 1 \text{ для q=2.}$$

Граница Варшамова–Гильберта показывает, что существует бесконечное семейство кодов C, удовлетворяющих соотношению R(C) ≥ 1 – H(δ(C)). Подтверждение этого факта базируется на доказательствах Гильберта, Варшамова и Возенкрафта.

Коды Гильберта. Для построения кодов Гильберт использовался «жадный» алгоритм с выбрасыванием кодовых слов, не удовлетворяющих границе

$$R(\mathrm{C}) \ge 1 - H(d(\mathrm{C})).$$

Алгоритм с выбрасыванием

Исходные параметры (n, d).

1. Инициализация множества S←$\{0,1\}^n$ и начального значения кода C← **0**.
2. Выполнение следующих итераций до тех пор, пока $S = 0$:

- выбор вектора **x** Î S и перемещение его в C;
- выбрасывание множества в виде точек внутри сферы $B(\mathbf{x},d)$ с радиусом d относительно **x** из S.

Если зафиксировать значения $0 < d(\mathrm{C}) < 1/2$, $\varepsilon > 0$ и $R \geq 1 - H(d) - e$, то для всех достаточно больших n алгоритм с выбрасыванием дает коды из не менее 2^{Rn} кодовых слов. Действительно, с каждой итерацией из множества S выбрасывается $Vol(d, n)$ элементов. Для достаточно больших n $Vol(d, n) \leq 2^{(H(d)+e)n}$. При поиске K кодовых слов, при условии, что алгоритм начинает работу с 2^n элементами, получаем

$$K \geq 2^n / Vol(d, n) \geq 2^{(1-H(d)-e)n} = 2^{Rn}.$$

Рандомизация алгоритма относительно размера 2^k случайного подмножества приводит с высокой достоверностью к аналогичному результату.

Линейный код Варшамова. Для построения кода используется следующая вероятностная процедура. Выбираются из поля $F_2^{k \times n}$ элементы порождающей матрицы **G** размером (k × n) как независимые случайные объекты с равномерным распределением вероятностей. Код образуется в результате умножения вектора **y** на порождающую матрицу $C = \{\mathbf{yG} : \mathbf{y} \in F_2^k\}$. Если задать

$$0 < d(\mathbf{C}) < 1/2 \text{ , } e > 0 \text{ и } R = 1 - H(d) - e,$$

то для больших значений n и $k = \lceil R\ n \rceil$ процедура формирования случайного линейного кода с высокой вероятностью формирует код, состоящий из 2^k кодовых слов с кодовым расстоянием не менее $(d\ n)$.

Предположим, что n – достаточно большое число и что объем сферы $Vol(dn, n) \leq 2^{(H(d)+e/2)n}$, $d = d\ n$. Для каждого фиксированного отличного от нуля вектора **y** кодовое слово **yG** будет случайным вектором в $\{0,1\}^n$ с равномерным законом распределением. Вероятность того, что вес кодового слова меньше кодового расстояния, будет равна вероятности того, что кодовое слово не находится внутри сферы B(**0**, δ n) и следовательно,

$$P(wt(\mathbf{yG} \leq d) = P(\mathbf{yG} \in B(\mathbf{0}, dn) = \frac{Vol(d,n)}{2^n} \leq 2^{(H(d)+e/2-1)n}.$$

Если $R = k/n = 1 - H(d) - e$, то эта вероятность оценивается величиной $2^{-(e/2)n}$, которая стремится к нулю при увеличении n.

Конструкция Возенкрафта. Алгоритм Варшамова требует не менее 2^{kn} шагов для формирования кода. Возенкрафт использовал простую идею для уменьшения затрат до величины 2^n, проводя поиск в семействе таких раздельных множеств $(S_1,\ldots,S_t) \subseteq \{0,1\}^n$ - **0**, что для каждого i множество $S_i \vee \{$ **0** $\}$ есть линейное

подпространство $\{0, 1\}^n$. Если такое семейство существует и $t \geq Vol(d,n)$, тогда имеется такое i ,что $S_i \vee \{ \mathbf{0} \}$ является линейным кодом с кодовым расстоянием не менее t. Заметим, что если наложить ограничение в виде равенства мощностей всех раздельных множеств, то получим линейный код размером $2^n / t$.

5.6. Многомерные полиномиальные коды

В данном подразделе рассматриваются коды, которые используют небольшие алфавиты. Основная идея состоит в применении многомерных полиномов для описания кодов.

Двухмерные полиномы. Для степени простого числа q и целого $l < q$ двухмерный полиномиальный код $B_{q,l}$ определяется следующим образом.

1. Сообщение источника состоит из $(l + 1)^2$ элементов поля, которые представляются в виде матрицы размером $(l + 1) \times (l + 1)$ с коэффициентами $<a_{ij}>$; i, j = =0,…,l. Сообщение такого вида можно описать в виде полинома

$$a(x, y) = \sum_{i=0}^{l} \sum_{j=0}^{l} a_{ij} x^i y^j .$$

2. Кодирование сообщения заключается в подстановку в полином $a(x,y)$ всех элементов поля, т.е. в получении $\left\langle c = a(a, b); a \in \mathbf{F}_q, b \in \mathbf{F}_q \right\rangle$.

В результате получается $(n, k, d)_q$ – код с параметрами $n = q^2$ и $k = (l + 1)^2$. Минимальное расстояние кода равно $d = (q - 1)^2$. Для сравнения граница Синглтона равна $d = q^2 – (l + 1)^2$.

Многомерные полиномиальные коды Рида–Маллера. Для положительных целых чисел m, l и степени простого числа q, $l < q$ код Рида–Маллера $\mathrm{PM}_{m,l,q}$ определяется следующим образом.

1. Сообщение источника, заданное в виде последовательности $\left\langle a_{i_1,\ldots,i_m \leq l} \right\rangle$, представляется в виде полинома $a(x_1,\ldots,x_m) = \sum a_{i_1,\ldots,i_m} x_1^{i_1} \ldots x_m^{i_m}$.

2. В полученный полином $A(\boldsymbol{x})$ подставляются элементы поля $\left\langle a(a); a \in F_q^m \right\rangle$.

Под подстановкой элементов поля понимается следующее. Пусть α -один из примитивных элементов поля GF(q^m),

$$a^i = \sum_{j=0}^{m-1} g_{ij} a^j , \quad 0 \leq i \leq q^m - 1, g_{ij} \in GF(q).$$

Под $f(\alpha^i)$ будем понимать значение $f(g_{i,0},g_{i,1},\ldots,g_{i,m-1})$ многочлена $f(x_1,\ldots,x_m)$, принадлежащего множеству всех многочленов степени l и меньше от m переменных. Таким образом, код РМ представляет собой множество

$$v(f) = \{ f(0), f(a^0), \ldots f(a^{q^m-2}) \},$$

где $f(0,\ldots,0) = f(0)$ - нулевая компонента.

Если в множестве $v(f)$ исключить нулевую компоненту, то получим так называемый обобщенный код РМ. Конструкция позволяет строить коды длиной n = $=q^m$. Длина информационной части определяется количеством последовательностей m-значных чисел, сумма которых не менее l. Число таких последовательностей оценивается, как $\binom{m+l}{m}$. Код имеет минимальное расстояние не менее $d = (1 - l/q)\, n$.

Выбор параметров кода. Предположим, требуется закодировать k букв алфавита. Спрашивается: для какого кода РМ длиной n = poly(k) и минимальным кодовым расстоянием (например $n/2$) достигается минимальный размер алфавита? Выберем $m = \frac{\log k}{\log\log k}$ и $q = \log^2 k$, а также такое значение l, чтобы $\binom{m+l}{l} = k$. В этом случае код РМ имеет длину, равную $n = q^m = k^2$. Для оценки кодового расстояния заметим, что $\binom{m+l}{l} \geq (l/m)^m$ и $l \leq mk^{1/m} = m\log k = \log^2 k / \log\log k = o(q)$ при $k \to \infty$. Следовательно, для данного семейства кодов минимальное расстояние равно (1- o(1)) n.

Рассмотрим случай, когда l > q. В этом случае сообщение задается полиномом общей степени большей чем l, а отдельные переменные имеют степени большие чем (q - 1). Пусть $K(m,l,q) = |S(m,l,q)|$. Код РМ определяется следующим образом.

1. Сообщение представляет собой последовательность $\langle a_i; i \in S(m,l,q) \rangle$, состоящую из $K(m, l, q)$ элементов поля F_q . Этой последовательности ставится в соответствие полином $a(x_1,\ldots,x_m) = \sum_{i \in S(m,l,q)} a_i x_1^{i1} \ldots x_m^{im}$.

2. Кодирование заключается в подстановке в полученный полином элементов поля $a \in F_q^m$, т.е. в получении последовательности $\left\langle a(a); a \in F_q^m \right\rangle$.

Алгоритм позволяет получить блочный код длиной $n = q^m$ и $k = K(m, l, q)$. Оценим кодовое расстояние. Заметим, что число l можно представить в виде l = e (q-1) + b, где $b < (q - 1)$. Минимальное кодовое расстояние в данном случае будет не меньше величины $q^{m-e}(1 - b/q)$.

Параметры кода. Пусть $q = 2$ и l < m. Тогда $K(m, l, 2)$ = Vol(l, m), т.е. совпадает с величиной радиуса l шара в пространстве $\{0,1\}^n$. Нижняя граница в этом случае

может быть оценена как $\binom{m}{l}$. Минимальное кодовое расстояние равно $d = 2^{m-l}$. Следовательно, получили код с параметрами $(2^m, \binom{m}{l}, 2^{m-l})$.

Порождающий полином. Зададим q-ичное разложение целого числа j как

$$j = j_0 + j_1 q + j_2 q^2 + \ldots + j_{m-1} q^{m-1}.$$

Весом j в q-ичном разложении называется сумма

$$w_q(j) = j_0 + j_1 + j_2 + \ldots + j_{m-1}.$$

Циклическим обобщенным кодом РМ порядка l и длиной $n = q^m - 1$ над полем $GF(q)$ называется циклический код, порождающий многочлен $g(x)$ которого имеет корни α^j при всех $j = 1, \ldots, q^m - 1$, таких, что $0 < w_q(j) \le (q - 1)\, m - l - 1$.

Спектральное представление. Обобщенные коды РМ допускают простое описание в частотной области. Проверочные частоты кода $w_q(j)$ маркируются всеми отличными от нуля индексами j, для которых выполняется неравенство $w_q(j) \le b$, где b некоторое число.

Пример. Выберем $m = 5$, и $l = 2$. Длина кода равна 31. Проверочные частоты кода удовлетворяют неравенству $w_2(j) \le 2$, т. е. теми j, двоичное представление которых содержит не более двух единиц. Проверочные частоты появляются при $j = 1, 3, 5$.

Циклический код РМ первого порядка удобно задавать с помощью функции следа. Пусть $f(x)$ будет неприводимый полином степени n над полем $F = GF(q^n)$ и α - корень $f(x)$ в поле $GF(q^n)$. Определим последовательность $\mathbf{c} = (c_0, \ldots, c_{n-1})$, элементы которой определяются как

$$c_i = Tr(b\, a^i),\ i = 0,1,\ldots,\ 0 \ne b \in GF(q^n).$$

Циклический код РМ первого порядка состоит из вектора **a** и всех его циклических сдвигов.

Пример. Пусть $q = 2$ и $f(x) = x^3 + x + 1$ задает поле $GF(2^3)$. Выберем α, которое удовлетворяет условию $f(\alpha) = 0$. Вычислим элементы $c_i = Tr(a^i)$, $i = 0,1,\ldots$. Получим вектор-строку $\mathbf{c} = (1, 0, 0, 1, 0, 1, 1)$. Все циклические сдвиги вектора образуют код.

Контрольные вопросы и задачи

1. Найти порождающий многочлен $g(x)$ для исправляющего двойные ошибки двоичного кода $n = 31$. Использовать примитивный элемент и примитивный полином $f(x) = x^5 + x^2 + 1$.

6. СВЕРТОЧНЫЕ КОДЫ

Сверточное кодирование разбивает поток данных на блоки длиной k_0, которые иногда называют кадрами информационных символов. Кадры информационных символов кодируются кадрами кодового слова длиной n_0 каждый. При этом кодирование производится с учетом предыдущих m кадров информационных символов. Процессы преобразования информации такого типа удобно описывать с помощью математического аппарата пространства состояний линейной динамической системы (рис. 6.1):

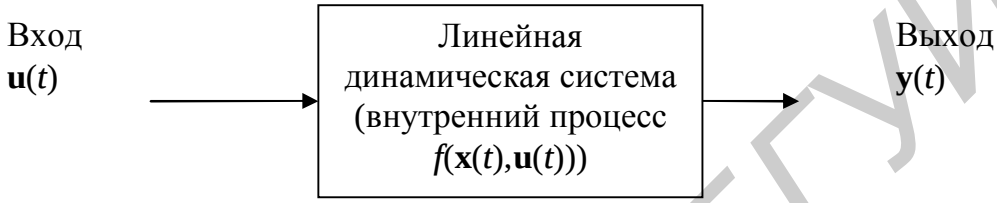


Рис. 6.1

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t),\mathbf{u}(t));\ \mathbf{y}(t) = h(\mathbf{x}(t),\mathbf{u}(t)),$$

где $f: R^{m+n} \to R^n$ - функция, моделирующая внутренний процесс (пространство состояний $\mathbf{x}(t)$) системы; $h: R^{m+n} \to R^p$ - функция, описывающая выходные сигналы.

В матричном виде уравнения пространства состояний часто записывают как

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t);\ \mathbf{y}(t) = C\mathbf{x}(t) + D(t),$$

где A, B, C и D скалярные матрицы.

Поведение системы можно охарактеризовать тройкой операторов $\Sigma = (T, W, B)$. Оператор $T \subseteq R$ – определяет временную ось, W – матричный оператор формальной переменной, называемый алфавитом сигналов, $B \subseteq W^T$ – оператор, описывающий поведение системы, элементы которого называются траекториями.

Пример. Предположим $T = Z_+$ - множество целых положительных чисел, W – конечное поле, состоящее из двух элементов, $B \subseteq W^T$ такой, что его элементы **щ**$\in B$ удовлетворяют уравнению $w_{t+2} = w_{t+1} + w_t$. Траектории пространства состояния системы имеют вид

$$B = \begin{Bmatrix} (0, & 0, & 0, & 0, & 0, & 0, & \mathbf{L}) \\ (1, & 0, & 1, & 1, & 0, & 1, & \mathbf{L}) \\ (1, & 1, & 0, & 1, & 1, & 0, & \mathbf{L}) \\ (0, & 1, & 1, & 0, & 1, & 1, & \mathbf{K}) \end{Bmatrix}.$$

Дискретная система, работающая в конечном поле $F(q)$ определяется как $T:= Z_+$, $W=F^n$ для $n\in Z_+$. Если определить кольцо $F^n[z]$ элементов степенного ряда, то вектор $w=(\omega_0, \omega_1, \omega_2,\ldots) \in W^T$ ассоциируется с полиномом $\sum_{i=1}^{\infty} w_i z^i$. Операторы дискретного сдвига вправо и влево задаются как

$$\mathbf{Z}: W^T \to W^T,\ (\omega_0, \omega_1, \omega_2,\ldots) \to (0, \omega_0, \omega_1, \omega_2,\ldots)$$

и

$$\mathbf{Z^{-1}}: W^T \to W^T,\ (\omega_0, \omega_1, \omega_2,\ldots) \to (\omega_1, \omega_2, \omega_3,\ldots).$$

Определим линейную динамическую систему как право-(лево) инвариантную относительно сдвига, если и только если $\mathbf{Z}\,B \subset B$ ($\mathbf{Z^{-1}}\,B \subset B$).

Сверточным кодом назовем подмножество $C \subseteq W^T$, если C–линейно, право-инвариантно и имеет средства обеспечения $C \subseteq F^n[z]$.

Сверточный код $C \subseteq F^n[z]$ можно определить как инъективный модуль гомоморфизма $j : F^k[z] \to F^n[z]$, что эквивалентно заданию полиномиальной матрицы $\mathbf{G}(z)$ размером $(n \times k)$,чьи столбцы формируют базис свободного подмодуля C. Такой подмодуль будем называть кодером сверточного кода со скоростью k/n.

Запишем полиномиальную матрицу кодера как

$$\mathbf{G}(z) = \begin{bmatrix} g_{11} & g_{12} & \mathbf{L} & g_{1k} \\ g_{21} & g_{22} & \mathbf{L} & g_{2k} \\ \mathbb{M} & \mathbb{M} & \mathbf{O} & \mathbb{M} \\ g_{n1} & g_{n2} & \mathbf{L} & g_{nk} \end{bmatrix}.$$

Определим степени столбцов кодера $\mathbf{G}(z)$ как множество целых чисел $\nu_1, \nu_2, \ldots, \nu_k$, где $n_i = \max\{\deg(g_{ij}) \mid 1 \le i \le n\},\ j = 1,..,k$. Упорядочим степени столбцов по возрастанию $\nu_1 \ge \nu_2 \ge \ldots \ge \nu_k$ (это всегда можно сделать после перестановки столбцов кодера). Величина ν_1 оценивает память кодера. Сложность кодера $d_{C(z)}$ задается выражением $d_{C(z)} = \sum_{i=1}^{k} n_i$.

Если C код с памятью $\nu_1=0$, то сверточный код совпадает с блочным кодом. В этом случае в момент времени t состояние выхода кодера зависит только от состояния его входа.

Формой синдрома для сверточного кода является модуль гомоморфизма, задаваемый как $y : F^n[z] \to F^{n-k}[z]$, со свойством $\mathrm{Im}(j) \subseteq \ker(y)$, где $\ker(y)$ и $\mathrm{Im}(\varphi)$- соответственно ядро и образ матричных представлений y и φ.

Синдром может быть определен через полиномиальную матрицу **H**(z) размером (n- k) × n, которая удовлетворяет условию **H**(z)**G**(z) = 0.

Теорема существования. Предположим, что сверточный код C $\subseteq F^n$[z] имеет скорость k/n и характеризуется сложностью d. Тогда существуют матрицы K, L размером (d + n - k) × d и матрица M (все определены над F), такие, что код C описывается множеством

$$C = \{v(z) \in F^n[z] \mid \exists x(z) \in F^d[z] : zKx(z) + Lx(z) + Mc(z) = 0\},$$

при этом удовлетворяются следующие условия:

- матрица K имеет полный ранг системы вектор-столбцов;
- матрица ($K|M$) имеет полный ранг системы вектор-строк;
- ранг $rank(z_0 K + L) = d + n - k, \forall z_0 \in \overline{F}$.

Говорят, что код задается тройкой (***K, L, M***). Можно определить базисную матрицу размером ($d \times k$) $\mathbf{X}(z) = diag((1, z, \ldots z^{n_1 - 1}), \ldots, (1, z, \ldots, z^{n_k - 1}))$.

Матрица **X**(z) имеет полный ранг для всех $z_0 \in \overline{F}$ и обладает таким свойством, что для каждого полиномиального вектора $\boldsymbol{f}(z)=(f_1(z),\ldots,f_k(z)) \in F^k$[z] со степенями $\deg f_i(z) \le n_i - 1$ существует уникальный вектор $\mathbf{v} \in F^d$, такой, что **v**$\mathbf{X}(z)=\boldsymbol{f}(z)$.

Ядро скалярной матрицы $\Lambda : F^{2d+n} \to F^{d+k}$

$$\mathbf{v} \to \mathbf{v} \begin{bmatrix} z\mathbf{X}(x) \\ \mathbf{X}(z) \\ \mathbf{G}(z) \end{bmatrix}$$

позволяет вычислить матрицы K, L, M.

Любую тройку (K, L, M) после элементарных преобразований записать в виде

$$K = \begin{bmatrix} -I \\ \mathbf{0} \end{bmatrix}, \quad L = \begin{bmatrix} A \\ C \end{bmatrix}, \quad M = \begin{bmatrix} \mathbf{0} & B \\ -I & D \end{bmatrix}.$$

Записывая кодовый вектор и вектор состояний в виде

$$\mathbf{v}(z) = \mathbf{v}_g z^g + \mathbf{v}_{g-1} z^{g-1} + \ldots + \mathbf{v}_0, \quad \mathbf{x}(z) = \mathbf{x}_g z^g + \mathbf{x}_{g-1} z^{g-1} + \ldots + \mathbf{x}_0,$$

сверточный код может быть описан системой уравнений

$$\mathbf{x}_{t-1} = A\mathbf{x}_t + B\mathbf{u}_t; \quad \mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t,$$

$$\mathbf{v}_t = \begin{bmatrix} \mathbf{y}_t \\ \mathbf{u}_t \end{bmatrix}, \quad \mathbf{x}_g = 0, \quad (zK + L)\mathbf{x}(z) + M\mathbf{v}(z) = 0.$$

Пример. Зададим базисную матрицу в виде $\mathbf{X}(z) = \begin{bmatrix} 1 & z & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$. Матрица Λ будет иметь вид

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}^T.$$

Ядро матрицы Λ равно

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Тройка (*K*, *L*, *M*) определится как

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad L = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Используя элементарные преобразования, матрицы (*K*, *L*, *M*) можно привести к виду

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Система кодирования запишется как

$$\mathbf{x}_{t-1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}\mathbf{x}_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{u}_t; \qquad \mathbf{y}_t = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\mathbf{x}_t + \begin{bmatrix} 1 & 1 \end{bmatrix}\mathbf{u}_t.$$

Пример. Сверточный код со скоростью 1/2 задается матрицей $\mathbf{G}(z) = \begin{bmatrix} z^2 + z + 1 \\ z^2 + 1 \end{bmatrix}$.

Структурная схема кодера имеет вид (рис. 6.2). Работа кодера описывается следующей системой уравнений:

$$\mathbf{x}_{t+1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\mathbf{x}_t + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u_t; \quad \mathbf{v}_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\mathbf{x}_t + \begin{bmatrix} 1 \\ 1 \end{bmatrix}u_t,$$

где $\mathbf{x}_t = \begin{bmatrix} x_t^{(1)}, & x_t^{(2)} \end{bmatrix}^T$, $\mathbf{v}_t = \begin{bmatrix} v_t^{(1)}, & v_t^{(2)} \end{bmatrix}^T$.

Откуда получаем уравнения, описывающие выходные сигналы кодера

$$v_t^{(1)} = x_t^{(1)} + x_t^{(2)} + u_t, \quad v_t^{(2)} = x_t^{(2)} + u_t.$$

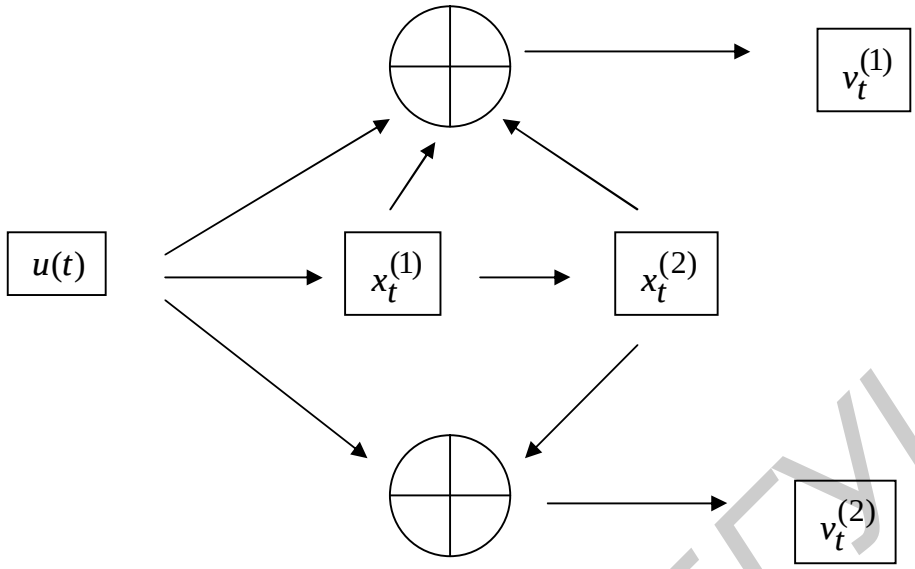


Рис. 6.2

Алгоритм синдромного декодирования. Точно так же, как и в случае блочных кодов для декодирования сверточных кодов можно вычислить синдром, используя проверочную матрицу **H**. Однако в этом случае синдром имеет бесконечную длину. Декодер не просматривает весь синдром однократно. Он работает с конца, вычисляя компоненты синдрома по мере их поступления, исправляя ошибки и исключая те компоненты синдрома, которые вычислены давно. Декодер содержит таблицу сегментов синдромов и сегментов конфигураций шума, которые приводят к данным сегментам синдрома. Когда декодер находит в таблице полученный сегмент синдрома, он исправляет начальный сегмент кодового слова.

Алгоритм декодирования Витерби. Алгоритм является полной процедурой декодирования сверточных кодов. Вероятность отказа от декодирования равна нулю. Однако при фиксированном коде вероятность ошибки декодирования будет больше, чем у неполного декодера. Декодер Витерби итеративно обрабатывает кадр за кадром, двигаясь по условной решетке, которая описывает пространство состояний кодера. В любой момент декодер не знает, в каком узле (состоянии) находится кодер и поэтому не пытается декодировать этот узел. Вместо этого декодер по принятой последовательности определяет наиболее правдоподобный путь к каждому узлу и определяет расстояние между каждым таким путем и принятой последовательностью. Это расстояние называется мерой расходимости пути. Декодер, как правило, знает начало пути пройденного кодером. В следующем кадре декодер определяет наиболее правдоподобный путь к каждому из новых узлов этого кадра. Но путь в каждый новый узел должен пройти через один из старых узлов. Возможные пути к новому узлу можно получить из старых, выбирая наиболее правдоподобный по наименьшей мере расходимости.

7. КОРРЕЛЯЦИОННЫЕ КОДЫ

Бинарные псевдослучайные кодовые структуры позволяют создать сигналы с минимальными корреляционными взаимодействиями. Наиболее известными примерами служат кодовые структуры, имеющие двух- и трехуровневые периодические функции авто- и взаимной корреляции [9,13].

Бинарные псевдослучайные последовательности $S = \{s_i, i = 0,..., N-1\}$, $s_i \in \{\pm 1\}$ можно получить с помощью гомоморфного отображения L элементов расширенного конечного поля $GF(2^n)$ в двоичное множество [12]

$$L : GF(2^n) \to GF(2)\ ,\ S = \{(-1)^{L(x)}\},\ x \in GF(2^n).$$

Для псевдослучайной m-последовательности оператор отображение представляет собой функцию следа $Tr(a^i)$, $Tr(a^{2^i+1})$, где a - примитивный элемент поля.

Корреляционные функции кодовых последовательностей можно оценить через преобразование Адамара функции отображения, которое запишется как

$$H(k) = \sum_{x \in GF(2^n)} (-1)^{Tr(kx)+L(x)},\ k \in GF(2^n)\ .$$

Для класса последовательностей с «хорошими» корреляционными функциями коэффициенты преобразования Адамара принимают 2^{n-1}раз нулевые значения и 2^{n-1}раз значения, равные ($\pm 2^{\frac{n+1}{2}}$).

Определим матрицу-циркулянт и ассоциированный с ней в модуле $x^n + 1$ циклический код $c(x)$ как

$$C = \begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \dots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \dots & c_{n-3} \\ \dots & \dots & \dots & \dots & \dots \\ c_1 & c_2 & c_3 & \dots & c_0 \end{bmatrix} = \left[c_{j-i \bmod n}\right]$$

где $c_i \in \{0,1\}$.

Если определить размерность ядра матрицы C и положить $\dim(\ker(C)) = k$, то можно показать, что $\sum_{x \in \ker C} (-1)^{Tr(kx)+L(x)} = \begin{cases} 2^k \\ 0 \end{cases}$,

и, следовательно, для G-последовательностей $\dim(\ker(C)) = 1$ и $rank(C) = n-1$.

В качестве функции отображения используем преобразование

$$L(x) = \sum_{i=1}^{M} c_i Tr(x^{2^i+1}),$$

где $M = (n-1)/2$, $x \in GF(2^n)$, n – нечетное число.

Для циклического кода справедливо соотношение $rank(C) = n - \deg(g(x))$, где $g(x)$ генераторный многочлен кода. Условию $\deg(g(x)) = 1$ удовлетворяет циклический код, для которого

$$g(x) = x + 1 = НОД(c(x), x^n + 1).$$

Требуемый код удобно отбирать из класса, задаваемым выражением

$$c(x) = \sum_{i=1}^{M} c_i (x^i + x^{n-i}). \qquad (7.1)$$

Пример. Пусть n=7 и S_G–последовательность формируется с помощью преобразований: $L(x) = \sum_{i=1}^{3} c_i Tr(x^{2^i+1})$,

$$c(x) = \sum_{i=1}^{3} c_i (x^i + x^{7-i}) = x^6 + x^5 + x^4 + x^3 + x^2 + x,$$

$$НОД(x^6 + x^5 + x^4 + x^3 + x^2 + x, x^7 + 1) == x + 1,\ s_{G,i} = (-1)^{Tr(a^{3i} + a^{5i} + a^{9i})}.$$

На рис. 7. 1 приведен спектр Адамара рассматриваемой последовательности.

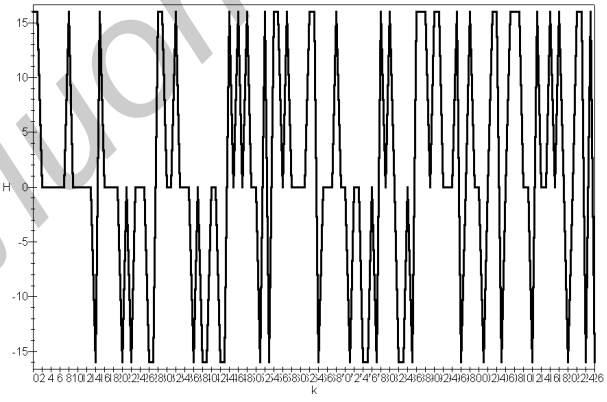


Рис. 7.1

Другой метод, позволяющий удобно формировать псевдослучайные кодовые конструкции с хорошими корреляционными функциями, опирается на понятия характерных спектральных преобразований и функции следа конечных полей.

Если задать кодовые последовательности

$$S_r = [s_{r,0}, s_{r,1}, ..., s_{r,N-1}]^T,\ r=,1,2...$$

периода N над полем F, то их периодические корреляционные функции могут быть записаны следующим образом:

$$R_{n,m}(t) = \sum_{i=0}^{N-1} c(s_{n,i+t})c^*(s_{m,i}),$$

где * – знак комплексного сопряжения, $c(x) = w^x$, $x \in F$ - аддитивный характер над полем F определяется как, w – комплексный примитивный корень N-й степени из единицы.

Идеальные кодовые последовательности имеют двухуровневые автокорреляционные функции и трехзначные взаимно корреляционные функции. Широкий класс кодовых последовательностей описывается с помощью функции следа элементов конечного поля

$$Tr_m^n(x) = x + x^{p^m} + \ldots + x^{p^{m(n/m-1)}}, x \in F_{p^n}$$

где $F_{p^n} = GF(p^n)$-расширенное конечное поле, p – простое число, $m|n$.

Пусть Ψ - множество функций, полученных в результате отображения из поля $GF(p^n)$ в поле $GF(p)$. Согласно [12] любую функцию из множества Ψ можно представить в виде суммы

$$f(x) = \sum_{i=1}^{L} Tr_1^n(A_i x^{t_i}); \quad A_i \in GF(p^{n_i}), \tag{7.2}$$

где $\{t_i; i = 1,\ldots L\}$ - подмножество лидеров смежных классов циклотомических множеств по модулю $(p^{n_i} - 1)$, $n_i|n$. При этом можно определить двухмерное характерное спектральное преобразование

$$H_{g,q,g}(z) = \sum_{y \in K} c\{g(zy)\}(\sum_{x \in K} c\{g(y^g x)\}c^*\{f(x^q)\})^*,$$

где $g(x)$ – ортогональная полиномиальная функция над F, $z \in K$, K - мультипликативная группа.

Для некоторых кодовых последовательностей над $GF(p)$ можно найти функции $f(x)$, которые представляют символы сигнала как

$$s_i = f(a^{l_i}), \quad i = 0,1,\ldots,N-1,$$

где a - примитивный элемент мультипликативной группы K.

Алгоритм формирование псевдослучайных сигналов. Рассмотрим случай формирования бинарных кодовых последовательностей. Примем p = 2, примитивный элемент $a \in GF(2^n)$ и ему соответствует минимальный многочлен $m(x)$. $f(x) = Tr_1^n(x), \; w = (-1)$.

Алгоритм формирования k-го символа кодовой последовательности запишется как

$$s_k = sign(\sum_{i=0}^{N-1}\sum_{l=0}^{N-1}(-1)^{h(a,i,l,g,q)}), \tag{7.3}$$

$$h(a,i,l,g,q)=\begin{bmatrix} Tr_1^n(a^{k+i})+Tr_1^n(a^{ig+l})+ \\ +Tr_1^n(a^{ql}) \end{bmatrix}. \qquad (7.4)$$

Алгоритм позволяет формировать бинарные последовательности различного типа, например m-последовательности, пары последовательностей Голда и др.

Корреляционные свойства сигналов. Структура и корреляционные свойства сигналов зависят от выбора чисел q и g. При соответствующем подборе могут быть получены семейства сбалансированных псевдослучайных последовательностей с двухуровневыми автокорреляционными функциями:

$$R(t)=\begin{cases} N, & t=0 \\ -1, & t\neq 0 \end{cases}.$$

Семейства включают в себя пары сигналов, имеющих друг с другом трехзначные и шестизначные взаимно корреляционные функции с минимальными боковыми лепестками.

Пример. Рассмотрим семейство последовательностей для n = 7, N = 127, $m(x) = x^7 + x + 1$. Задавая $q = g = 3$, получаем

S_m = -1-1-1-1-1-1-11-11-11-1-111-111-1-1-1 1-1-1-1-111111-1-11-11-1-11-1-1-1111-1-1-11-11-11111-1111-11-1-1-1-1111-111-1-1-1-1111-1-1-11-111-11111-11-1-1-1111-1-111-111-1111-111-111-11-111-1-11-1-1.

Выбирая q =3,g =11, формируем сигнал

S_n = -1-1-11-1-111-11-1-1111-1-111-1-11-111-11111-1-1-1-11111-11-111-1-1-11-11-1-111-1111111-1-1-1-1-11-111-11-11-111-111-1-11111-1-1-1-11-1111-1-111-1-1-111-111-1-1111-11-11-11-11-1-1-1-1-1-1-1-1-1 .

На рис. 7.2 и 7.3 показаны взаимно корреляционные функции последовательностей.

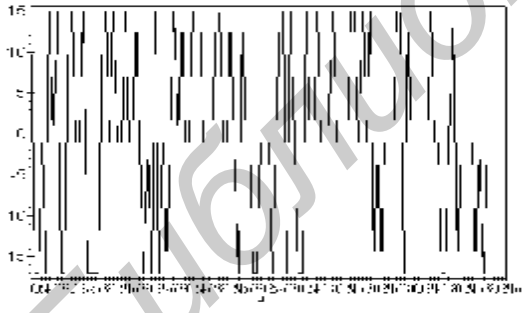


Рис. 7.2

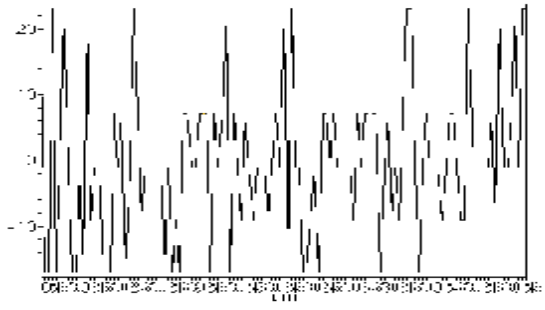


Рис. 7.3

Контрольные вопросы и задачи

1. Используя функцию следа, сформируйте *m*-последовательность для поля GF(32).

2. Как связаны преобразование Адамара и корреляционные функции кодов?

3. Используя преобразования (7.2-7.3), сформируйте пару последовательностей Голда.

ЛИТЕРАТУРА

1. Шеннон К. Математическая теория связи: В сб. "Работы по теории информации и кибернетике". – М.: ИИЛ, 1963.

2. Шеннон К. Теория связи в секретных системах: В сб. "Работы по теории информации и кибернетике" - М.: ИИЛ, 1963.

3. Касами Т., Токура Н., Ивадари Ё., Инагаки Я. Теория кодирования: Пер. с яп.- М.: Мир, 1978.

4. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. - М.: Мир, 1986.

5. Лосев В. В. Помехоустойчивое кодирование в радиотехнических системах передачи информации. Ч.1,2. – Мн.: МРТИ , 1984.

6. Марков А.А. Введение в теорию кодирования. Учеб. пособие.- М.: Наука, 1982.

7. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ.-М.: Радио и связь, 1987.

8. МакВильямс Ф. Дж., Слоэн Н. Дж. Теория кодов, исправляющих ошибки. – М.: Связь, 1979.

9. Габидулин Э.М., Афанасьев В.Б. Кодирование в радиоэлектронике. – М.: Радио и связь, 1986.

10. Конопелько В.К., Липницкий В.А. Теория норм синдромов и перестановочное декодирование помехоустойчивых кодов.- Мн.: БГУИР, 2000.

11. Скляр Б. Цифровая связь. Теоретические основы и практическое применение: Пер. с англ.-М.: Изд. дом «Вильямс», 2003.-1104 с.

12. Gong G., Golomb W. The Decimation-Hadamard Transform of Two-Level Autocorrelation Sequences. IEEE Trans. on Information Theory, v. 48. No. 4. April, 2002. P. 853-865.

13. Khoo K. , Gong G. , Stinson D. , R. A New Family of Gold-like Sequences.- Canada, University of Waterloo, Waterloo.

14. Галлагер Р. Теория информации и надежная связь. – М.: Сов. радио, 1974.

15. Питерсон, Э.Уэлдон. Коды, исправляющие ошибки. - М.: Мир, 1976.

16. Dholakia A. Introduction to Convolution Codes with Applications. Kluwer Academic Publishers,1994.

Св. план 2004, поз. 74

Учебное издание

Саломатин Сергей Борисович

КОДИРОВАНИЕ ИНФОРМАЦИИ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ

УЧЕБНОЕ ПОСОБИЕ
по курсу

«Кодирование и защита информации»
для студентов специальностей
«Радиоэлектронные системы», «Радиоинформатика»
дневной формы обучения

Редактор Е.Н. Батурчик
Компьютерная верстка М.В. Шишло

Подписано в печать 14.12.2004. Формат 60x84 1/16. Бумага офсетная.
Гарнитура «Таймс». Печать ризографическая. Усл. печ. л. 5,81.
Уч.-изд. л. 5,5. Тираж 200 экз. Заказ 592.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0133108 от 30.04.2004.
220013, Минск, П. Бровки, 6