ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ им. проф. М. А. БОНЧ-БРУЕВИЧА»

С. С. Владимиров

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Учебное пособие

СПб ГУТ)))

Санкт-Петербург
2016

УДК 621.391(075.8)
ББК 32.88173
В 57

Рецензенты
профессор кафедры СС и ПД, доктор технических наук *О. С. Когновицкий*;
ведущий инженер ЗАО «НПП «ИСТА-Системс», кандидат технических наук *А. А. Березкин*

Утверждено редакционно-издательским советом СПбГУТ в качестве учебного пособия

Владимиров, С. С.

В 57 Математические основы теории помехоустойчивого кодирования : учебное пособие / С. С. Владимиров ; СПбГУТ. — СПб, 2016. — 96 с.

ISBN 978-5-89160-131-4

Настоящее учебное пособие призвано ознакомить студентов старших курсов с математическими основами теории помехоустойчивого кодирования.

Предназначено для студентов, обучающихся по направлениям 11.03.02 «Инфокоммуникационные технологии и системы связи» и 09.03.01 «Информатика и вычислительная техника».

УДК 621.391(075.8)
ББК 32.88173

ISBN 978-5-89160-131-4

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

При разработке систем передачи данных одним из важнейших этапов является выбор методов повышения достоверности при передаче информационных сообщений по каналам связи. Использование избыточных помехоустойчивых кодов является одним из наиболее эффективных методов борьбы с ошибками при передаче дискретных сообщений по каналам связи. Соответственно, вопросам изучения теории помехоустойчивого кодирования придается большое внимание в программах подготовки бакалавров и магистров, обучающихся по специальностям из области телекоммуникаций.

В настоящем пособии приведены основы математического аппарата, который используется при изучении теории помехоустойчивого кодирования, исследовании алгоритмов кодирования и декодирования, разработке и построении кодеров и декодеров приемопередающих устройств.

Пособие состоит из девяти разделов. В разд. 1 рассмотрены основные понятия и классификация помехоустойчивых кодов. Разд. 2 посвящен основам двоичной алгебры и реализации основных операций над двоичными числами. В разд. 3 описаны матрицы и основные действия над ними. В разд. 4 приводятся основные понятия комбинаторики. В разд. 5 — полиномы и основные операции с полиномами. Разд. 6 посвящен понятиям группы, кольца и поля, а в разд. 7 описан основной математический аппарат блочных помехоустойчивых кодов (Боуза–Чоудхури–Хоквингема и Рида–Соломона) — двоичные поля Галуа. В разд. 8 приводятся основы теории графов. В разд. 9 рассмотрены основные модели каналов передачи данных.

1. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Помехоустойчивое кодирование (англ. *Error Correcting Coding*, ECC) — процесс преобразования информации, предоставляющий возможность обнаружить и исправить ошибки, возникающие при передаче информации по каналам передачи данных.

Под *ошибкой* при этом понимают ситуацию, когда в результате действия помех и искажений в канале передачи данных приемник принимает неверное решение, отождествляя принятый сигнал не с фактически переданным символом, а с каким-либо другим [1].

Процесс помехоустойчивого кодирования заключается во введении *избыточности*, т. е. для передачи информации используется код, у которого используются не все возможные комбинации, а только некоторые из них. Такие коды называют избыточными или корректирующими.

Соответственно, процесс введения избыточности (преобразование информационных символов в кодовое слово) называется *кодированием*, а обратный процесс восстановления информации из кодового слова, возможно содержащего ошибки, — *декодированием*.

В рамках цифровой системы передачи данных задачи кодирования и декодирования возложены на *кодер* и *декодер* соответственно. Структура цифровой системы передачи данных показана на рис. 1.1 [2].

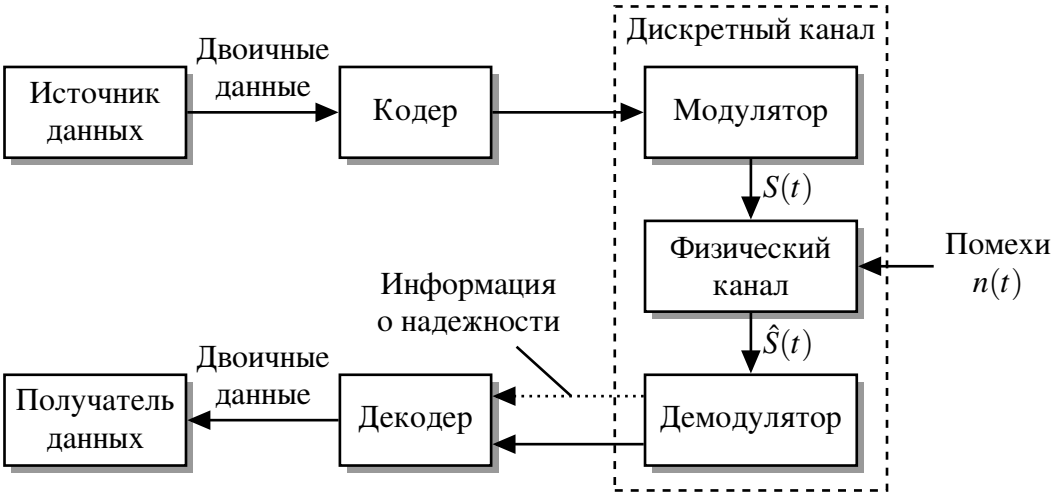


Рис. 1.1. Структура цифровой системы передачи данных

Часто декодеру доступна информация, указывающая на надежность решений, принимаемых о различных символах кодового слова. Такая информация может быть использована для упрощения процесса декодирования, либо для улучшения его характеристик [2].

В целом, способность помехоустойчивых кодов определять и исправлять ошибки — их корректирующие свойства — зависит от правил постро-

ения этих кодов и параметров кода (числа разрядов, избыточности и др.), а также от используемых алгоритмов декодирования.

1.1. Основные параметры помехоустойчивых кодов

Основными параметрами, характеризующими корректирующие свойства кодов являются:

1) избыточность кода;
2) кодовое расстояние;
3) кратность гарантированно обнаруживаемых ошибок;
4) кратность гарантированно исправляемых ошибок.

1.1.1. Избыточность корректирующего кода

Избыточность корректирующего кода может быть абсолютной и относительной. Под *абсолютной избыточностью* понимают число вводимых дополнительных разрядов

$$r = n - k,$$

где n — число кодовых символов на выходе кодера, соответствующих k информационным символам на его входе.

Относительной избыточностью корректирующего кода называют величину

$$R_{\text{отн}} = \frac{r}{n} = \frac{n-k}{n} = 1 - \frac{k}{n}.$$

С ней связана так называемая *относительная скорость передачи информации* или *скорость кода*, которая показывает, какую часть общего числа символов кодовой комбинации составляют информационные символы.

$$\frac{k}{n} = 1 - R_{\text{отн}}.$$

Если производительность источника равна H символов в секунду, то скорость передачи после кодирования этой информации будет равна

$$R = H \cdot \frac{k}{n}.$$

1.1.2. Кодовое расстояние

Кодовое расстояние d или *расстояние Хемминга* характеризует степень различия любых двух кодовых комбинаций. Оно выражается числом разрядов, в которых комбинации отличаются одна от другой.

Чтобы получить кодовое расстояние между двумя комбинациями двоичного кода, достаточно подсчитать число единиц в поразрядной сумме этих

комбинаций по модулю 2:

$$10011 \oplus 11001 = 01010 \quad \Rightarrow \quad d = 2.$$

Кодовое расстояние может быть различным. Так, в первичном натуральном безызбыточном коде это расстояние для различных комбинаций может различаться от единицы до n, где n — длина (значность) кода.

Для помехоустойчивого кода наиболее важным является *минимальное кодовое расстояние* $d_{\min}$ — наименьшее кодовое расстояние из всех между всеми парами кодовых комбинаций.

В безызбыточном коде все комбинации являются разрешенными, $d_{\min} = 1$. Поэтому искажение хотя бы одного символа в комбинации будет приводить к получению ошибочного сообщения.

1.1.3. Кратности гарантированно обнаруживаемых и гарантированно исправляемых ошибок

Эти параметры напрямую зависят от минимального кодового расстояния. Под *кратностью* понимается количество поражённых ошибками символов кодовой комбинации.

В общем случае при необходимости обнаруживать ошибки кратности $t_{\text{обн}}$ минимальное кодовое расстояние должно быть, по крайней мере, на единицу больше $t_{\text{обн}}$, т. е.

$$d_{\min} \geq t_{\text{обн}} + 1.$$

Соответственно, *кратность гарантированно обнаруживаемых кодом ошибок* равна

$$t_{\text{обн}} \leq d_{\min} - 1.$$

Кратность гарантированно исправляемых кодом ошибок вычисляется по формуле

$$t \leq \frac{d_{\min} - 1}{2}.$$

Таким образом, код, имеющий минимальное кодовое расстояние $d_{\min} = 3$, позволяет гарантированно обнаружить $t_{\text{обн}} = 2$ и менее ошибок и гарантированно исправить $t = 1$ ошибку.

1.2. Классификация помехоустойчивых кодов

Помехоустойчивые коды классифицируются по различным признакам. Одной из основных классификаций является деление кодов на *блочные* и *непрерывные*.

Блочный (блоковый) код является *кодом без памяти*. Кодер блочного кода отображает подающийся на вход блок информационных символов длиной k в кодовую последовательность из n выходных символов. Термин «без памяти» указывает, что каждый блок из n символов зависит только от соответствующего блока из k символов и не зависит от других блоков [2].

Основными параметрами блочных кодов являются длина информационного блока k, длина кодового слова n, скорость кода $\frac{k}{n}$ и минимальное кодовое расстояние $d_{\min}$.

Непрерывные или *древовидные* коды — это помехоустойчивые коды использующие непрерывную, или последовательную, обработку информации короткими фрагментами (блоками). Кодер древовидного кода является устройством *с памятью*. На его вход поступают наборы из k входных информационных символов, а на выходе появляются наборы из n кодовых символов. Каждый набор n кодовых символов зависит от текущего входного набора и от v предыдущих входных символов. Следовательно кодер должен содержать устройство памяти на $m = k + v$ входных символов. Параметр m часто называют *длиной кодового ограничения* кода [2].

Также *непрерывные* коды характеризуются скоростью кода $\frac{k}{n}$ и свободным расстоянием $d_{\text{св}}$ [2].

Чаще всего используются линейные древовидные коды, называемые *сверточными*.

Особое место в классификации помехоустойчивых кодов занимают каскадные коды и турбо коды, представляющие из себя комбинации блочных и/или непрерывных кодов [3].

Другой подход к классификации делит коды на *линейные* и *нелинейные*. Линейные коды образуют векторное пространство, в котором два кодовых слова при сложении по определенному правилу дают в результате третье кодовое слово [2].

Практически все применяемые на практике схемы кодирования основаны на использовании линейных кодов. Двоичные линейные блоковые коды часто называют *групповыми* кодами, так как их кодовые слова образуют математическую структуру, называемую *группа* [2].

Нелинейные коды применяются гораздо реже линейных. К нелинейным кодам относится *код с контрольным суммированием*, в котором проверочные разряды являются записью суммы единиц в кодовой комбинации [1].

По способу кодирования коды делятся на *систематические* и *несистематические*. В первом случае информационные символы передаются на выход декодера без изменения и к ним добавляются проверочные символы. В

случае несистематического кодирования информационные символы в явном виде в кодовом слове отсутствуют.

Большинство помехоустойчивых кодов может быть использовано как для обнаружения, так и для исправления ошибок, хотя есть коды, которые позволяют лишь обнаруживать ошибки. Поскольку избыточность, требуемая для обнаружения ошибок, меньше избыточности для исправления ошибок, то коды с обнаружением ошибок часто используют в системах с обратной связью [1].

Ещё одним вариантом деления помехоустойчивых кодов является разделение их на *коды, исправляющие случайные ошибки*, и *коды, исправляющие пакеты (пачки) ошибок*. Хотя для исправления пачек ошибок было разработано большое количество кодов с хорошими характеристиками, часто оказывается выгодным использовать коды, исправляющие случайные ошибки, совместно с устройствами перемежения/деперемежения [2]. Также стоит отметить, что существуют алгоритмы декодирования, позволяющие использовать коды, рассчитанные на исправление случайных ошибок, для исправления пачек ошибок без использования перемежителей. К таким алгоритмам относится, например, мажоритарное декодирование на основе двойственного базиса [4].

Контрольные вопросы

1. Что такое помехоустойчивое кодирование?
2. Опишите структуру цифровой системы передачи данных.
3. Дайте понятие избыточности корректирующего кода. Что такое абсолютная и относительная избыточности? Как определяется скорость кода?
4. Что такое кодовое расстояние? Как оно определяется?
5. Как рассчитываются кратности гарантированно обнаруживаемых и гарантированно исправляемых ошибок?
6. Приведите классификацию помехоустойчивых кодов.

2. ЭЛЕМЕНТЫ ДВОИЧНОЙ АЛГЕБРЫ

2.1. Понятие системы счисления. Основные системы счисления

Под *системой счисления* как правило понимают совокупность приемов записи и наименования чисел [5].

Системы счисления разделяют на *непозиционные*, в которых значение цифры не зависит от ее положения в записи числа, и *позиционные*, где значение каждой цифры изменяется с изменением ее позиции в числе [5].

На сегодняшний день в основном используются позиционные системы счисления. Главной характеристикой позиционной системы можно считать *основание системы счисления* p, определяющее количество цифр/знаков (от 0 до $p-1$), использующихся для записи чисел. Сами цифры $0 \dots p-1$ называются *базисными числами* [5, 6].

Любое число N в системе счисления представляется в виде комбинации степеней основания p с коэффициентами (цифрами) a_i, относящимися к множеству базисных чисел $0 \dots p-1$, как показано в формуле

$$a_k p^k + a_{k-1} p^{k-1} + \ldots + a_1 p + a_0,$$

и сокращенно записывается в виде $(a_k a_{k-1} \dots a_1 a_0)_p$ [6].

В настоящее время широко используются четыре системы счисления.

1. *Десятичная система счисления* — позиционная система счисления с основанием 10. Для записи чисел в десятичной системе используются цифры: $0, 1, \dots, 9$. Это основная система счисления, использующаяся повсеместно [5].

2. *Двоичная система счисления* — позиционная система счисления с основанием 2. Для записи чисел в двоичной системе используются две цифры: 0 и 1 [5]. Каждая цифра двоичного числа соответствует в десятичной системе счисления двойке в степени, равной номеру позиции цифры слева. В табл. 2.1 приведены значения первых двенадцати степеней.

Таблица 2.1

Степенной ряд двойки (до 12-й степени включительно)

Показатель степени	1	2	3	4	5	6	7	8	9	10	11	12
Значение	2	4	8	16	32	64	128	256	512	1024	2048	4096

3. *Восьмеричная система счисления* — позиционная система счисления с основанием 8. Для записи чисел используются цифры: $0, 1, \dots, 7$ [5]. Используется, например, в некоторых справочниках для представления полиномов.

4. *Шестнадцатеричная система счисления* — позиционная система счисления с основанием 16. Для записи чисел используются цифры:

$0, 1, \ldots, 9, A, B, C, D, E, F$ [5]. Широко используется в программировании и телекоммуникациях, так как позволяет компактно и удобно представлять длинные двоичные последовательности.

Также стоит отметить так называемую *двоично-десятичную* систему счисления, в которой каждая цифра десятичного числа записывается соответствущим ей двоичным четырехразрядным числом. Эта система счисления используется в ЭВМ как промежуточная при преобразовании десятичных чисел в двоичные [5].

Для того, чтобы отличать числа в разных системах счисления (если надо записывать их вперемешку), используется несколько способов.

1. В конце числа указывается нижний индекс со значением основания системы счисления.
2. В конце числа указывается нижний индекс с названием системы счисления. Обычно используются первые три буквы латинского названия.
3. После числа указывается буквенный постфикс. Обычно используют первую букву латинского названия.
4. Перед числом указывается определенный префикс.

Последние два способа записи широко применяются в информатике, в частности в программировании, где невозможно использовать верхние и нижние индексы.

В табл. 2.2 показаны примеры различной записи чисел в разных системах счисления. Приведены только некоторые формы записи. В различных языках программирования и языках разметки могут использоваться другие префиксы и постфиксы. Также необходимо отметить, что при записи десятичных чисел постфиксы и префиксы не используются. Число без них по умолчанию считается десятичным, если иное не следует из контекста.

Таблица 2.2

Формы записи чисел в различных системах счисления

Основание системы	**Ниж. индекс со значением**	**Ниж. индекс с названием**	**Постфикс**	**Префикс**
2	1010_2	1010_{bin}	$1010b$	$0b1010$
8	1724_8	1724_{oct}	$1724o$	$0o1724$
10	1942_{10}	1942_{dec}	—	—
16	$3AF5_{16}$	$3AF5_{hex}$	$3AF5h$	$0x3AF5$

2.2. Перевод чисел между системами счисления

2.2.1. Перевод двоичного числа в десятичное

Перевод двоичного числа в десятичное производится по классической схеме сложения степеней двойки с коэффициентами, которыми являются соответствующие цифры этого двоичного числа.

Процесс перевода двоичного числа в десятичное рассмотрим на примере двоичного числа

$$100111001011_2.$$

Для простоты распишем показатели степени

$$\begin{matrix}11&10&9&8&7&6&5&4&3&2&1&0\\1&0&0&1&1&1&0&0&1&0&1&1\end{matrix}_2.$$

Таким образом, для получения десятичного числа можно написать

$$1\cdot2^{11}+0\cdot2^{10}+0\cdot2^9+1\cdot2^8+1\cdot2^7+1\cdot2^6+0\cdot2^5+0\cdot2^4+1\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0.$$

Убрав степени с коэффициентом 0, получим

$$1\cdot2^{11}+1\cdot2^8+1\cdot2^7+1\cdot2^6+1\cdot2^3+1\cdot2^1+1\cdot2^0.$$

Уберем коэффициенты и раскроем степени

$$2^{11}+2^8+2^7+2^6+2^3+2^1+2^0=2048+256+128+64+8+2+1=2507.$$

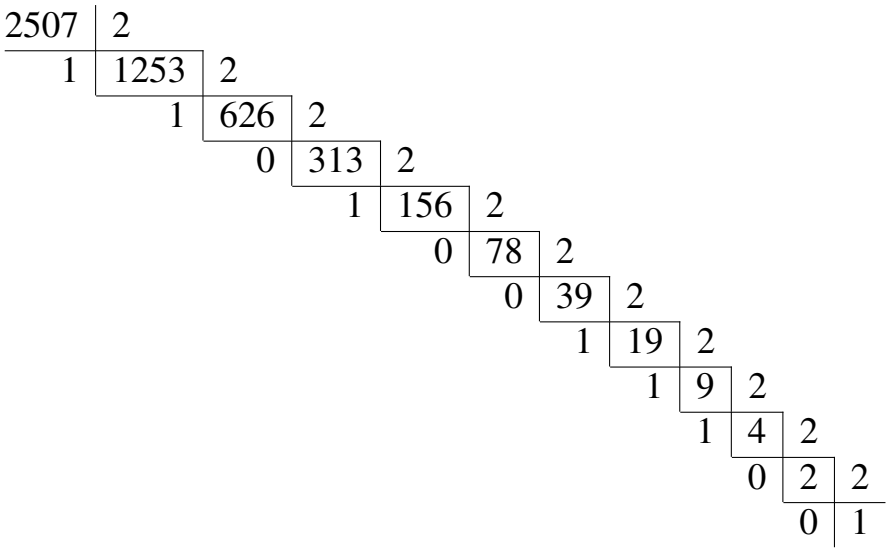
Таким образом

$$100111001011_2=2507_{10}.$$

2.2.2. Перевод десятичного числа в двоичное

Для перевода десятичного числа в двоичное используется процедура последовательного деления десятичного числа на два с накоплением остатка.

Для примера рассмотрим обратный перевод числа 2507_{10}, полученного в примере, приведенном в предыдущем подпункте.



Результат записывается, начиная с конца. Таким образом

$$2507_{10} = 100111001011_2.$$

При переводе небольших десятичных чисел иногда удобнее использовать разложение в ряд степеней двойки (табл. 2.1), обратно рассмотренной в пп. 2.2.1 процедуре.

Например, число 292_{10} можно представить как

$$292_{10} = 256 + 32 + 4 = 2^8 + 2^5 + 2^2 = 100100100_2.$$

Вначале берем ближайшую к заданному десятичному числу степень двойки — в примере это число $256 = 2^8$. Теперь считаем разность $292 - 256 = 36$. Далее снова берем ближайшую степень двойки $32 = 2^5$. Остается $4 = 2^2$. В итоге получаем искомое число в двоичном виде.

2.2.3. Перевод между двоичной и восьмеричной системами счисления

Перевод целых чисел из двоичной системы в восьмеричную и обратно основан на том, что каждому восьмеричному числу ставится в соответствие его значение в двоичной системе счисления, выраженное в виде блока из трех двоичных цифр — *двоичная триада* [5]. Соответствия между восьмеричными цифрами и двоичными числами показаны в табл. 2.3

Таблица 2.3

Соответствия между цифрами восьмеричной системы счисления и двоичными числами (в виде триад)

Восьмеричные цифры	**Двоичные триады**	**Восьмеричные цифры**	**Двоичные триады**
0	000	4	100

Окончание табл. 2.3

Соответствия между цифрами восьмеричной системы счисления и двоичными числами (в виде триад)

Восьмеричные цифры	**Двоичные триады**	**Восьмеричные цифры**	**Двоичные триады**
1	001	5	101
2	010	6	110
3	011	7	111

При переводе двоичного целого числа в восьмеричную систему счисления его необходимо разбить на триады, начиная с младшего разряда, а затем вместо каждой триады записать соответствующую ей восьмеричную цифру [5]. Для примера рассмотрим перевод двоичного числа 1001001000011_2:

$$1001001000011_2 \Rightarrow 001.001.001.000.011 \Rightarrow 1.1.1.0.3 \Rightarrow 11103_8.$$

Заметим, что при разбиении двоичного числа на триады пришлось дополнить его нулями слева.

При переводе целого восьмеричного числа в двоичную систему счисления достаточно вместо каждой восьмеричной цифры записать соответствующую ей двоичную триаду [5]. Для примера рассмотрим обратный перевод восьмеричного числа 11103_8:

$$11103_8 \Rightarrow 001.001.001.000.011 \Rightarrow 1001001000011_2.$$

Незначащие нули в левой части полученного двоичного числа были удалены.

2.2.4. Перевод между двоичной и шестнадцатеричной системами счисления

Перевод целых чисел из двоичной системы в шестнадцатеричную и обратно аналогичен переводу между двоичной и восьмеричной системами. Каждой шестнадцатеричной цифре ставится в соответствие ее значение в двоичной системе счисления, выраженное в виде блока из четырех двоичных цифр — *двоичная тетрада* [5]. Соответствия между восьмеричными цифрами и двоичными числами показаны в табл. 2.4

Таблица 2.4

Соответствия между цифрами шестнадцатеричной системы счисления и двоичными числами (в виде тетрад)

Шестнадцатеричные цифры	**Двоичные тетрады**	**Шестнадцатеричные цифры**	**Двоичные тетрады**
0	0000	8	1000
1	0001	9	1001
2	0010	*A*	1010

Окончание табл. 2.4

Соответствия между цифрами шестнадцатеричной системы счисления и двоичными числами (в виде тетрад)

Шестнадцатеричные цифры	**Двоичные тетрады**	**Шестнадцатеричные цифры**	**Двоичные тетрады**
3	0011	*B*	1011
4	0100	*C*	1100
5	0101	*D*	1101
6	0110	*E*	1110
7	0111	*F*	1111

При переводе двоичного целого числа в шестнадцатеричную систему счисления его необходимо разбить на тетрады, начиная с младшего разряда, а затем вместо каждой тетрады записать соответствующую ей шестнадцатеричную цифру [5]. Для примера рассмотрим перевод двоичного числа 1001001001011_2:

$$1001001001011_2 \Rightarrow 0001.0010.0100.1011 \Rightarrow 1.2.4.B \Rightarrow 124B_{16}.$$

Заметим, что при разбиении двоичного числа на тетрады пришлось дополнить его нулями слева.

При переводе целого шестнадцатеричного числа в двоичную систему счисления достаточно вместо каждой шестнадцатеричной цифры записать соответствующую ей двоичную тетраду [5]. Для примера рассмотрим обратный перевод шестнадцатеричного числа $124B_{16}$:

$$124B_{16} \Rightarrow 0001.0010.0100.1011 \Rightarrow 1001001001011_2.$$

Незначащие нули в левой части полученного двоичного числа были удалены.

2.2.5. Перевод между десятичной и двоично-десятичной системами счисления

Перевод целых чисел между десятичной и двоично-десятичной системами счисления аналогичен переводу между двоичной и шестнадцатеричной системами. Каждой десятичной цифре ставится в соответствие ее значение в двоичной системе счисления, выраженное в виде двоичной тетрады [5]. Соответствия между десятичными цифрами и двоичными тетрадами показаны в табл. 2.5

Таблица 2.5

Соответствия между цифрами десятичной системы счисления и двоичными тетрадами

Десятичные цифры	**Двоичные тетрады**	**Десятичные цифры**	**Двоичные тетрады**
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

При переводе целого десятичного числа в двоично-десятичную систему счисления достаточно вместо каждой десятичной цифры записать соответствующую ей двоичную тетраду [5]. Для примера рассмотрим перевод десятичного числа 124_{10}:

$$124_{10} \Rightarrow 0001.0010.0100 \Rightarrow 000100100100_{2-10}.$$

Обратный перевод из двоично-десятичной системы в десятичную полностью аналогичен переводу из двоичной в шестнадцатеричную систему.

2.3. Операции над двоичными числами

К основным операциям над двоичными числами относятся логические операции *инверсия*, *конъюнкция* и *дизъюнкция* и арифметические операции *сложение* и *умножение*, а также *сложение по модулю 2*, соответствующее логической операции «Исключающее ИЛИ».

Инверсия является *унарной* операцией, т. е. осуществляется над одним операндом. Прочие операции являются *бинарными*, т. е. осуществляются над двумя операндами.

Инверсия, конъюнкция и дизъюнкция являются основными логическими операциями. Этим операциям соответствуют простейшие логические элементы «НЕ» (*инвертор*), «И» и «ИЛИ», на основе которых строятся любые цифровые микросхемы. Для описания логических элементов используются *таблицы истинности*, содержащие совокупность всех возможных комбинаций логических сигналов на входе цифрового устройства и значений выходных сигналов для каждой комбинации. К основным логическим элементам также относят элемент «Исключающее ИЛИ», буфер памяти, а также обратные элементы «И–НЕ», «ИЛИ–НЕ», «Исключающее ИЛИ–НЕ».

Основные логические элементы имеют стандартные условно-графические обозначения. Существует несколько стандартов обозначения.

1. Стандарт Международной электротехнической комиссии (МЭК, International Electrotechnical Commission, IEC), аналогичный британскому стандарту BS3939.

2. Американский стандарт MIL/ANSI.

3. Обозначения, согласно книге A. Croft, R. Davidson and M. Hargreaves *Engineering Mathematics*, 1992. Эти обозначения часто называют CDH. Эти обозначения аналогичны стандарту MIL/ANSI за исключением обозначений для элементов «И»/«И–НЕ».

4. ГОСТ 2.743-91 «Единая система конструкторской документации. Обозначения условные графические в схемах. Элементы цифровой техники» (ЕСКД). Этот стандарт во многом аналогичен стандарту IEC.

Для основных логических элементов в пособии будут приведены все варианты обозначений. Основным вариантом, который будет использоваться для более сложных схем, как наиболее употребимый на практике, выбран стандарт MIL/ANSI.

2.3.1. Инверсия

Инверсия (*отрицание*, *дополнение*, *операция «НЕ»*, NOT) осуществляет смену значения операнда на противоположное. В том случае, если речь идет об инверсии двоичного числа из нескольких разрядов, эта операция осуществляется *поразрядно* (побитово), т. е. инвертируется каждый разряд (бит) числа.

В тексте операция инверсии может обозначаться различными способами, как показано в формуле:

$$\overline{a}; \qquad !a; \qquad \neg a. \tag{2.1}$$

В настоящем пособии будем использовать первое обозначение. Например

$$\overline{01110010} = 10001101.$$

Таблица истинности инвертора показана в табл. 2.6.

Таблица 2.6

Таблица истинности инвертора

a	$\overline{a}$
0	1
1	0

На рис. 2.1 представлены основные условно-графические обозначения инвертора.

(а) *(б)*

Рис. 2.1. Условно-графические обозначения инвертора:
(а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

2.3.2. Конъюнкция

Конъюнкция (*логическое умножение*, *операция «И»*, AND) является бинарной операцией, возвращающей 1, только когда оба операнда равны 1. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*. Поразрядные операции как правило осуществляются с операндами равной длины. В случае разной длины операндов, тот, который имеет меньшую длину, дополняется нулями в старших степенях. Например, при необходимости осуществить поразрядное логическое умножение чисел $c = 110010_2$ и $d = 1010_2$, число d следует представить как $d = 001010_2$.

В записи операция логического умножения может обозначаться различными способами, как показано в формуле:

$$a \wedge b; \qquad a \cap b; \qquad a \& b. \tag{2.2}$$

В настоящем пособии будем использовать первое обозначение.

Далее приведем таблицу логического умножения (табл. 2.7) и таблицу истинности соответствующего ему логического элемента «И» (табл. 2.8).

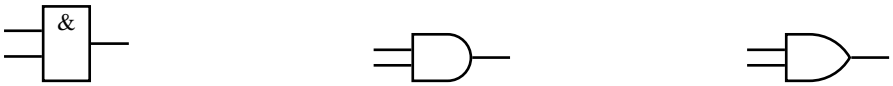
Таблица 2.7
Таблица логического умножения

$\wedge$	0	1
0	0	0
1	0	1

Таблица 2.8
Таблица истинности элемента «И»

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

На рис. 2.2 представлены основные условно-графические обозначения логического элемента «И».



(а) *(б)* *(в)*

Рис. 2.2. Условно-графические обозначения логического элемента «И»:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI; (в) по стандарту CDH

Рассмотрим операцию поразрядного логического умножения на примере двух двоичных чисел a и b:

$$a = 101101_2; \qquad b = 100110_2.$$

$$\begin{array}{cccccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \wedge & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

Таким образом, $a \wedge b = 101101_2 \wedge 100110_2 = 100100_2$.

2.3.3. Дизъюнкция

Дизъюнкция (*логическое сложение*, *операция «ИЛИ»*, OR) является бинарной операцией, возвращающей 1, когда хотя бы один операнд равен 1. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*.

В записи операция логического сложения может обозначаться различными способами:

$$a \vee b; \qquad a \cup b; \qquad a|b. \tag{2.3}$$

В настоящем пособии будем использовать первое обозначение.

Далее приведем таблицу логического сложения (табл. 2.9) и таблицу истинности соответствующего ему логического элемента «ИЛИ» (табл. 2.10).

Таблица 2.9

Таблица логического сложения

$\vee$	0	1
0	0	1
1	1	1

Таблица 2.10

Таблица истинности элемента «ИЛИ»

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

На рис. 2.3 представлены основные условно-графические обозначения логического элемента «ИЛИ».

(а) *(б)*

Рис. 2.3. Условно-графические обозначения логического элемента «ИЛИ»: (а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

Рассмотрим операцию поразрядного логического сложения на примере двух двоичных чисел a и b:

$$a \vee b = 101101_2 \vee 100110_2 = 101111_2.$$

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \vee & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline & 1 & 0 & 1 & 1 & 1 & 1 \end{array}$$

2.3.4. *Сложение по модулю 2*

Сложение по модулю 2 (*операция «Исключающее ИЛИ»*, XOR) является бинарной операцией, возвращающей 1, когда один операнд равен 0, а второй 1, и возвращающей 0 при равенстве операндов. Фактически, этот сумматор выполняет суммирование без учета переноса. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*.

В записи операция сложения по модулю 2 обычно обозначается способом, показанным в формуле:

$$a \oplus b. \tag{2.4}$$

Далее приведем таблицу сложения по модулю 2 (табл. 2.11) и таблицу истинности сумматора по модулю 2 (табл. 2.12).

Таблица 2.11

Таблица сложения по модулю 2

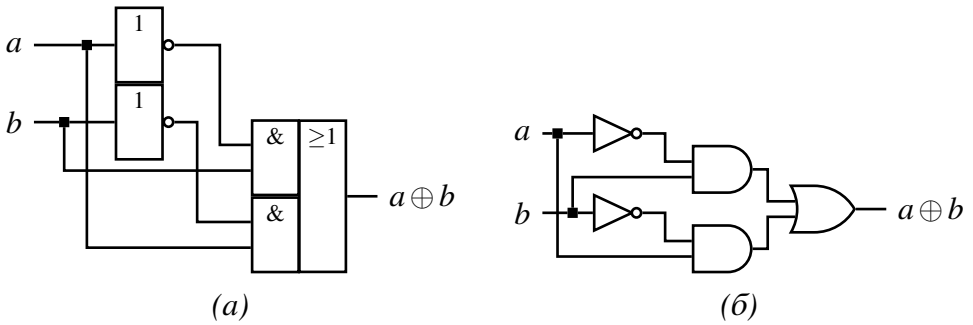
$\oplus$	0	1
0	0	1
1	1	0

Таблица 2.12

Таблица истинности сумматора по модулю 2

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Принципиальная схема сумматора по модулю 2 показана на рис. 2.4.



(а) *(б)*

Рис. 2.4. Принципиальная схема сумматора по модулю 2: (а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.5 представлены основные условно-графические обозначения сумматора по модулю 2.

(а) *(б)*

Рис. 2.5. Условно-графические обозначения сумматора по модулю 2: (а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

Рассмотрим операцию поразрядного сложения по модулю 2 на примере двух двоичных чисел a и b:

$$a \oplus b = 101101_2 \oplus 100110_2 = 001011_2.$$

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 1 \end{array}$$

2.3.5. Сложение

В отличие от сложения по модулю 2 операция двоичного арифметического сложения производится с переносом в следующий двоичный разряд. То есть, $1_2 + 1_2 = 10_2$. В записи для обозначения двоичного арифметического сложения используется обычный символ сложения «+».

Поскольку необходимо учитывать перенос, схема построения двоичного сумматора усложняется по сравнению с сумматором по модулю 2. Выделяют два типа схем:

1) *полусумматор*, который формирует перенос в следующий разряд, но не может учитывать перенос из предыдущего разряда;

2) *полный сумматор*, который формирует перенос в следующий разряд, и учитывает перенос из предыдущего разряда.

Далее приведем таблицу двоичного сложения (табл. 2.13) и таблицу истинности одноразрядного полусумматора (табл. 2.14), в которой S обозначает сумму, а PO — перенос.

Таблица 2.13

Таблица двоичного сложения

+	0	1
0	0	1
1	1	10

Таблица 2.14

Таблица истинности одноразрядного полусумматора

A	B	S	PO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Принципиальная схема, реализующая таблицу истинности одноразрядного полусумматора, показана на рис. 2.6.

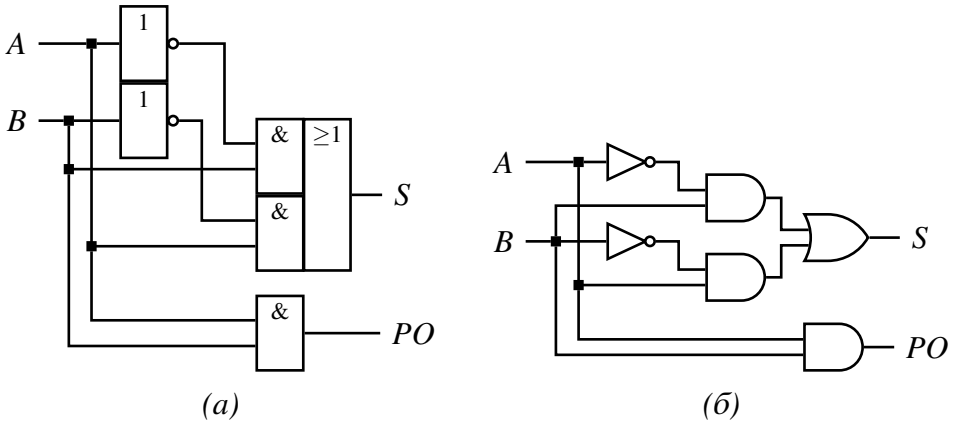


Рис. 2.6. Принципиальная схема одноразрядного полусумматора: (а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

Анализируя схему полусумматора на рис. 2.6, можно увидеть, что часть схемы, отвечающая за получение суммы, соответствует схеме сумматора по модулю 2, следовательно, схему полусумматора можно преобразовать так, как показано на рис. 2.7.

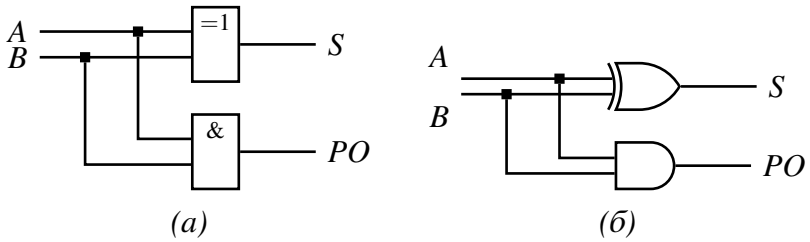


Рис. 2.7. Принципиальная схема полусумматора на основе сумматора по модулю 2: (а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.8 приведено условно-графическое обозначение одноразрядного полусумматора.

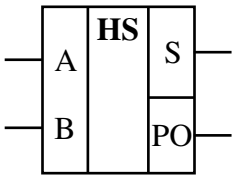


Рис. 2.8. Условно-графическое обозначение одноразрядного полусумматора

Таблица истинности полного двоичного одноразрядного сумматора показана в табл. 2.15, где *S* обозначает сумму, *PI* — перенос на входе, а *PO* — перенос на выходе.

Таблица 2.15

Таблица истинности полного двоичного одноразрядного сумматора

PI	*A*	*B*	*S*	*PO*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Принципиальная схема полного двоичного одноразрядного сумматора на основе сумматоров по модулю 2 показана на рис. 2.9. Можно видеть, что эта схема состоит из двух полусумматоров и схемы ИЛИ, объединяющей сигналы переносов от полусумматоров.

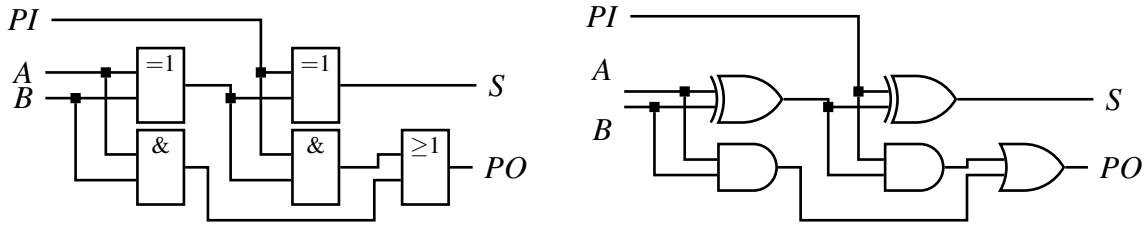


Рис. 2.9. Принципиальная схема полного двоичного одноразрядного сумматора на основе сумматоров по модулю 2:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.10 приведено условно-графическое обозначение полного двоичного одноразрядного сумматора.

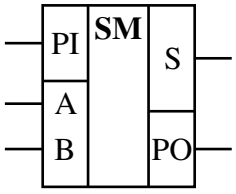


Рис. 2.10. Условно-графическое обозначение полного двоичного одноразрядного сумматора

Для того чтобы получить многоразрядный сумматор, достаточно соединить входы и выходы переносов соответствующих двоичных разрядов. На практике одноразрядные сумматоры практически никогда не использовались, так как почти сразу же были выпущены микросхемы многоразрядных сумматоров. На рис. 2.11 приведены принципиальная схема и условно-графическое обозначение полного двоичного четырехразрядного сумматора.

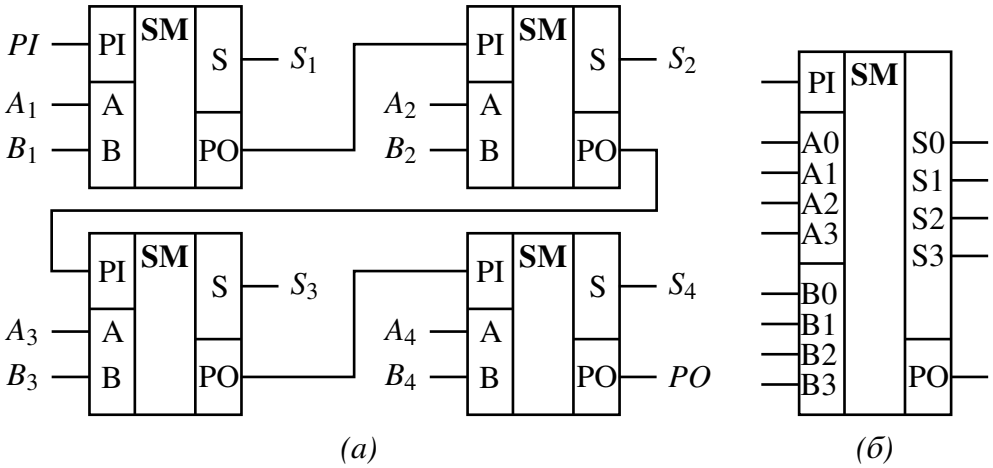


Рис. 2.11. Полный двоичный четырехразрядный сумматор: (а) принципиальная схема; (б) условно-графическое обозначение

При ручных расчетах арифметическое двоичное сложение a и b легко осуществляется методом сложения «в столбик»:

$$\begin{array}{rcccccc} & & {\scriptstyle 1} & {\scriptstyle 1} & & & \\ & 1 & 0 & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array} \begin{array}{l} = 45_{10} = a \\ = 38_{10} = b \\ = 83_{10} \end{array}$$

2.3.6. Умножение

Операция арифметического умножения обычно рассматривается как многоразрядная, поскольку одноразрядное умножение полностью соответствует логическому умножению (конъюнкции), а одноразрядный умножитель, соответственно, схеме «И».

При ручных расчетах умножение, как и сложение, удобно вычисляется «в столбик». Рассмотрим на примере $b \cdot a$:

$$\begin{array}{rcccccccccc} & & & & & \times & 1 & 0 & 0 & 1 & 1 & 0 \\ & & & & & & 1 & 0 & 1 & 1 & 0 & 1 \\ \hline & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & 1 & 0 & 1 & 1 & 0 & 1 & \\ & & & & 1 & 0 & 1 & 1 & 0 & 1 & & \\ & & & 1 & 0 & 1 & 1 & 0 & 1 & & & \\ \hline 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & \end{array} \begin{array}{l} = 38_{10} = b \\ = 45_{10} = a \\ \\ \\ \\ \\ = 1710_{10} \end{array}$$

Из этого расчета видно, что умножение двух двоичных чисел представляет собой последовательное суммирование первого множителя с ним же, но сдвинутым согласно позициям единиц во втором множителе.

Контрольные вопросы

1. Что такое система счисления? Какие системы счисления сейчас применяются?
2. Как осуществляется перевод из десятичной системы счисления в двоичную и обратно?
3. Как осуществляется перевод между двоичной и шестнадцатеричной системами счисления?
4. Что такое двоично-десятичная система счисления?
5. Что такое конъюнкция?
6. Что такое дизъюнкция?
7. Нарисуйте схему сумматора по модулю 2.
8. Как работает полный двоичный сумматор? Нарисуйте схему.
9. Как строится полный двоичный многоразрядный сумматор?
10. Как производится перемножение многоразрядных двоичных чисел?

3. МАТРИЦЫ И ДЕЙСТВИЯ НАД НИМИ

3.1. Понятие матрицы

Матрицей называется прямоугольная таблица чисел из некоторого числового поля, имеющая m строк и n столбцов [7, 8]:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

В общем случае такая матрица называется *прямоугольной* размера $m \times n$ или $m \times n$-матрицей. Если $m = n$, то матрица называется *квадратной порядка n*. Числа, составляющие матрицу, называются ее *элементами*. При двухиндексном обозначении элементов, например a_{12}, первый индекс всегда указывает номер строки, а второй индекс — номер столбца, на пересечении которых стоит данный элемент [7, 8].

Каждой $m \times n$-матрице $\mathbf{A}$ с элементами a_{ij} соответствует $n \times m$-матрица с элементами a_{ji}. Она называется *транспонированной* к $\mathbf{A}$ и обозначается через $\mathbf{A}^T$. $(\mathbf{A}^T)^T = \mathbf{A}$. Строки матрицы $\mathbf{A}$ становятся столбцами в $\mathbf{A}^T$ и столбцы матрицы $\mathbf{A}$ становятся строками в $\mathbf{A}^T$ [8]:

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 & -3 \\ 3 & 5 & 12 & 6 \\ 7 & -4 & -8 & 2 \end{bmatrix} \quad \longrightarrow \quad \mathbf{A}^T = \begin{bmatrix} 1 & 3 & 7 \\ 3 & 5 & -4 \\ 5 & 12 & -8 \\ -3 & 6 & 2 \end{bmatrix}.$$

Прямоугольная матрица размера $m \times 1$, т. е. состоящая из одного столбца, называется *вектор-столбцом* или *столбцевой матрицей*. Прямоугольная матрица размера $1 \times n$, т. е. состоящая из одной стороки, называется *вектор-строкой* или *строчной матрицей* [7, 8].

Квадратная матрица, в которой все элементы ниже или выше главной диагонали равны нулю, называется *треугольной*. Соответственно, если все элементы ниже главной диагонали равны нулю, то матрица является *верхнетреугольной*, а если все элементы выше главной диагонали равны нулю, то *нижнетреугольной*. Определитель треугольной матрицы равен произведению элементов на её главной диагонали. Треугольная матрица, в которой все элементы на главной диагонали равны единице, получила название верхней или нижней *унитреугольной*. Соответственно, определитель такой матрицы

равен единице [7, 8].

Верхнетреугольная матрица :
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}.$$

Нижнетреугольная матрица :
$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}.$$

Квадратную матрицу, у которой все элементы, расположенные вне главной диагонали, равны нулю, называют *диагональной*. Эта матрица является частным случаем как верхнетреугольной, так и нижнетреугольной матриц [7, 8]:
$$\begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n \end{bmatrix}.$$

Диагональная матрица, в которой все элементы главной диагонали равны 1, называется *единичной* [8]:
$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

Еще одним частным случаем квадратной матрицы является *диагонально-постоянная матрица* или *матрица Тёплица*, названная в честь немецкого математика Отто Тёплица (1881–1940). Это матрица, в которой на всех диагоналях, параллельных главной, стоят равные элементы, т. е. выполняется соотношение $a_{ij} = a_{i-1,j-1}$ [9].

В общем виде $n \times n$ матрица Тёплица имеет вид:
$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \dots & \dots & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \dots & \dots & a_2 & a_1 & a_0 \end{bmatrix}.$$

По аналогии с треугольными матрицами выделяют верхнетреугольные и нижнетреугольные матрицы Тёплица.

3.2. Операции с матрицами

3.2.1. Сложение матриц

Складываются только матрицы одного размера. Сложение матриц производится поэлементно [8]:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

где $c_{ij} = a_{ij} + b_{ij}$.

Например,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \\ a_{41}+b_{41} & a_{42}+b_{42} & a_{43}+b_{43} \end{bmatrix}$$

или

$$\begin{bmatrix} 1 & 3 & 5 & -3 \\ 3 & 5 & 12 & 6 \\ 7 & -4 & -8 & 2 \end{bmatrix} + \begin{bmatrix} 4 & -2 & 3 & 1 \\ 8 & -6 & 1 & 4 \\ 2 & 4 & 13 & 5 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 8 & -2 \\ 11 & -1 & 13 & 10 \\ 9 & 0 & 5 & 7 \end{bmatrix}.$$

3.2.2. Умножение матрицы на число

При умножении матрицы на число, каждый элемент матрицы умножается на это число [8]:

$$b \cdot \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

где $c_{ij} = b \cdot a_{ij}$.

Например,

$$3 \times \begin{bmatrix} 1 & 3 & 5 & -3 \\ 3 & 5 & 12 & 6 \\ 7 & -4 & -8 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 9 & 15 & -9 \\ 9 & 15 & 36 & 18 \\ 21 & -12 & -24 & 6 \end{bmatrix}.$$

3.2.3. Произведение матриц

Матрицы умножаются по правилу «строка-на-столбец». Для того, чтобы матрицу **A** можно было умножить на матрицу **B**, количество столбцов в **A**

должно быть равно количеству строк в **B**. Таким образом, результатом произведения $m \times l$-матрицы **A** на $l \times n$-матрицу **B** будет $m \times n$-матрица **C** [8]:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1l} \\ a_{21} & a_{22} & \dots & a_{2l} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{ml} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{l1} & b_{l2} & \dots & b_{ln} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

где $c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \ldots + a_{il} \cdot b_{lj}$.

Например,

$$\begin{bmatrix} 1 & 3 & 5 & -3 \\ 3 & 5 & 12 & 6 \\ 7 & -4 & -8 & 2 \end{bmatrix} \times \begin{bmatrix} 4 & -2 & 3 & 1 \\ 8 & -6 & 1 & 4 \\ 2 & 4 & 13 & 5 \\ 4 & 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 26 & -3 & 71 & 32 \\ 100 & 18 & 170 & 95 \\ -12 & -20 & -87 & -45 \end{bmatrix}.$$

Для примера рассмотрим умножение первой строки матрицы **A** на первый столбец матрицы **B**:

$$(1 \cdot 4) + (3 \cdot 8) + (5 \cdot 2) + (-3 \cdot 4) = 26.$$

3.2.4. Сложение двоичных матриц по модулю 2

Сложение *двоичных матриц* (т. е. матриц, элементы которых являются двоичными цифрами 0 или 1) по модулю 2, производится поэлементно и, как и обычное сложение, осуществляется над матрицами одного размера:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \oplus \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

где $c_{ij} = a_{ij} \oplus b_{ij}$.

Например,

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

3.2.5. Произведение двоичных матриц

Произведение двоичных матриц **A** и **B** осуществляется по обычному правилу «строка-на-столбец». Соответственно, как и при умножении обычных матриц, количество столбцов в **A** должно быть равно количеству строк

в **B**. Отличием является то, что при вычислении элементов результирующей матрицы **C** используется не обычное сложение, а сложение по модулю 2:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1l} \\ a_{21} & a_{22} & \dots & a_{2l} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{ml} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{l1} & b_{l2} & \dots & b_{ln} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

где $c_{ij} = a_{i1} \cdot b_{1j} \oplus a_{i2} \cdot b_{2j} \oplus \ldots \oplus a_{il} \cdot b_{lj}$.

Например,

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

Контрольные вопросы

1. Что такое матрица? Какие виды матриц бывают?
2. Что такое транспонированная матрица?
3. Как производится умножение матриц?

4. ЭЛЕМЕНТЫ КОМБИНАТОРИКИ

Любая совокупность элементов произвольного рода образует *множество*. Множество, состоящее из конечного числа элементов, называется *конечным множеством* [10].

Если существует два множества A и B, и при этом каждый элемент множества B принадлежит множеству A, то B называется *подмножеством* множества A. Все возможные k-элементные комбинации из элементов n-элементного множества представляют собой подмножества n-элементного множества, которые называют *сочетаниями* из n по k элементов. Иногда вместо слова «сочетание» употребляют термин — *комбинация* из n элементов по k [10]. Число сочетаний (комбинаций) из n по k обозначают C_n^k [10] или $\binom{n}{k}$ [11]. В пособии будем использовать первое обозначение. Число сочетаний рассчитывается по формуле (4.1) [10]:

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{4.1}$$

Множество называется *упорядоченным*, если каждому элементу этого множества поставлено в соответствие некоторое число (номер элемента) от 1 до n, где n — число элементов множества, так что различным элементам соответствуют различные числа. Упорядоченные множества считаются различными, если они отличаются либо своими элементами, либо их порядком. Различные упорядоченные множества, которые отличаются лишь порядком элементов (т. е. могут быть получены из того же самого множества), называются *перестановками* этого множества. Число перестановок обозначается P_n и рассчитывается по формуле (4.2) [10]:

$$P_n = n!. \tag{4.2}$$

Упорядоченные k-элементные подмножества множества из n элементов называются *размещениями* из n элементов по k. Различные размещения из n по k отличаются количеством элементов либо их порядком [10]. Число размещений из n по k обозначают A_n^k и рассчитывают по формуле (4.3) [10]:

$$A_n^k = k! \cdot C_n^k = \frac{n!}{(n-k)!} = n(n-1)\dots(n-k+1). \tag{4.3}$$

Контрольные вопросы

1. Что такое сочетания? Как рассчитывается число сочетаний?
2. Что такое упорядоченное множество?
3. Что такое перестановки? Как рассчитывается число перестановок?
4. Что такое размещения? Как рассчитывается число размещений?

5. ПОЛИНОМЫ И ДЕЙСТВИЯ НАД НИМИ

При изучении помехоустойчивых кодов под *полиномом (многочленом)* будем понимать многочлен от одной переменной, т. е. конечную сумму вида

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n,$$

где a_k — *коэффициент* многочлена $f(x)$ при x^k [12]. Каждый из элементов многочлена вида $a_k x^k$ называется *одночленом* степени k.

Многочлен, все коэффициенты которого равны нулю, называется *нулевым многочленом*. Если многочлен не является нулевым, то наибольшее из таких чисел k, что $a_k \neq 0$, называется *степенью* этого многочлена [12].

При изучении помехоустойчивого кодирования мы в основном будем сталкиваться с полиномами следующих трех видов.

1. Полином с десятичными коэффициентами.
2. Полином с коэффициентами 0 или 1 (простое поле $\mathrm{GF}(2)$).
3. Полином с коэффициентами, принадлежащими конечному полю $\mathrm{GF}(2^p)$.

В этом разделе рассмотрим операции с полиномами первых двух видов. Полиномы с коэффициентами, принадлежащими конечному полю $\mathrm{GF}(2^p)$, затронем при рассмотрении математики конечных полей Галуа.

Полиномы можно записывать в виде вектор-строки коэффициентов. Например, полином

$$f(x) = 2 + 4x + 5x^2 + 3x^4 + 2x^6$$

можно записать как

$$f(x) = [2\ 4\ 5\ \overrightarrow{0}\ 3\ 0\ 2]$$

или

$$f(x) = [2\ 0\ 3\ \overleftarrow{0}\ 5\ 4\ 2],$$

в зависимости от того, записывать его по возрастанию или по убыванию степеней. Часто такая форма записи применяется для полиномов с коэффициентами, принадлежащими простому полю $\mathrm{GF}(2)$. Такая запись внешне совпадает с записью двоичного числа. Также при работе с литературой необходимо уточнять используется запись по возрастанию или по убыванию степеней. Полином

$$f(x) = x^4 + x + 1$$

может быть записан как

$$f(x) = 1\ 1\ \overrightarrow{0}\ 0\ 1$$

или

$$f(x) = 1\ 0\ \overleftarrow{0}\ 1\ 1.$$

5.1. Операции с полиномами

5.1.1. Сумма полиномов

При суммировании полиномов складываются коэффициенты одночленов одноименной степени.

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n,$$
$$g(x) = b_0 + b_1x + b_2x^2 + \ldots + b_mx^m,$$
$$m > n,$$
$$h(x) = f(x) + g(x) = (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + \ldots +$$
$$+(a_n + b_n)x^n + \ldots + (a_m + b_m)x^m.$$

При рассмотрении полиномов с коэффициентами 0 и 1, принадлежащими простому полю $\mathrm{GF}(2)$ необходимо учитывать, что сложение в этом случае производится по модулю 2. Таким образом, при сложении двух одночленов одной степени с коэффициентом 1, результирующий одночлен будет иметь коэффициент 0:

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

То есть, происходит сокращение пар одночленов одинаковой степени.

5.1.2. Произведение полиномов

Умножение полиномов выполняется по следующему принципу

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n,$$
$$g(x) = b_0 + b_1x + b_2x^2 + \ldots + b_mx^m,$$
$$h(x) = f(x)g(x) = f(x)b_0 + f(x)b_1x + f(x)b_2x^2 + \ldots + f(x)b_mx^m =$$
$$= (b_0a_0 + b_0a_1x + \ldots + b_0a_nx^n) + (b_1a_0x + b_1a_1x^2 + \ldots + b_1a_nx^{n+1}) + \ldots +$$
$$+(b_ma_0x^m + b_ma_1x^{1+m} + \ldots + b_ma_nx^{n+m}).$$

Далее полученные полиномы складываются по обычному правилу. Таким образом, в результате произведения полинома степени n на полином степени m получается полином степени $n + m$.

Умножение полиномов с коэффициентами из простого поля $\mathrm{GF}(2)$ производится аналогично с учетом того, что операция сложения осуществляется по модулю 2:

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

5.1.3. Деление полиномов

Для деления полиномов удобно использовать процедуру деления в столбик:

$$\begin{array}{r|l} 4x^3+10x^2+6x+1 & 2x^2+4x+1 \\ \underline{-\,4x^3+\ 8x^2+\ 2x} & \hline 2x+1 \\ \quad 2x^2+\ 4x+1 & \\ \underline{\qquad\qquad 0} & \end{array}$$

Таким образом

$$\frac{4x^3+10x^2+6x+1}{2x^2+4x+1} = 2x+1.$$

Аналогично производится деление полиномов с коэффициентами из простого поля $\mathrm{GF}(2)$. Вместо операции вычитания используется сложение по модулю 2:

$$\begin{array}{r|l} \oplus\, x^5+x^4+x^3+x+1 & x^2+1 \\ \underline{\quad x^5+x^3} & \hline x^3+x^2+1 \\ \quad \oplus\, x^4+\ x+1 & \\ \quad\ \underline{x^4+x^2} & \\ \quad\quad x^2+x+1 & \\ \quad\quad \oplus\, \underline{x^2+1} & \\ \quad\quad\quad\ x & \end{array}$$

Таким образом

$$\frac{x^5+x^4+x^3+x+1}{x^2+1} = (x^3+x^2+1) + \frac{x}{x^2+1}.$$

Контрольные вопросы

1. Что такое полином? Какие способы записи полиномов существуют?
2. Как умножаются полиномы?
3. Как осуществляется деление полиномов?

6. ПОНЯТИЕ ГРУППЫ, КОЛЬЦА И ПОЛЯ

6.1. Группа

Группой называется множество элементов, для которых определена некоторая операция «$\bullet$» (сложение или умножение) и выполняется ряд приведенных ниже аксиом G.1–G.4 [4, 13].

Аксиома G.1. Операция «$\bullet$» может быть применена к любым двум элементам группы, в результате чего получается третий элемент группы.

$$\text{Если} \quad a \in \mathbb{G} \quad \text{и} \quad b \in \mathbb{G}, \quad \text{то} \quad a \bullet b \in \mathbb{G}.$$

Аксиома G.1 определяет *замкнутость операции в группе*. Как правило операции над элементами называют *сложением* («+») или *умножением* («·»; «×»), даже если они не являются обычными сложением и умножением. В соответствии с двумя записями операций различают *аддитивную* и *мультипликативную* группы [4, 13].

Аксиома G.2. *Свойство ассоциативности.* Для любых трех элементов a, b и c из группы $\mathbb{G}$ верно

$$a \bullet (b \bullet c) = (a \bullet b) \bullet c.$$

То есть порядок выполнения операций несущественен.

Аксиома G.3. В группе $\mathbb{G}$ всегда существует единичный элемент e, такой, что

$$a \bullet e = e \bullet a = a \ \text{ для любого } \ a \in \mathbb{G}.$$

Для аддитивной группы единичный элемент называют нулем, обозначают 0 и определяют из уравнения

$$a + 0 = 0 + a = a.$$

Для мультипликативной группы единичный элемент называют единицей и определяют из уравнения

$$a \cdot 1 = 1 \cdot a = a.$$

Аксиома G.4. Для любого элемента $a \in \mathbb{G}$ существует обратный элемент a^{-1} такой, что

$$a \bullet a^{-1} = a^{-1} \bullet a = e.$$

Для аддитивной группы обратный к a элемент обозначается $-a$ и находится из уравнения

$$a + (-a) = (-a) + a = 0.$$

Для мультипликативной группы обратный к a элемент обозначается a^{-1} и находится из уравнения

$$a \cdot a^{-1} = a^{-1} \cdot a = 1.$$

Кроме того, для любого элемента $a \in \mathbb{G}$ и любого целого положительного n

$$(a^n)^{-1} = (a^{-1})^n.$$

Можно ввести степени элемента a с целыми отрицательными показателями так что $a^{-n} = (a^n)^{-1}$ [14, 15]. Также можно обозначить

$$a^0 = e.$$

Все обычные правила действий со степенями остаются справедливыми в любой группе [14, 15].

Рассмотрим все возможные степени произвольного элемента g группы $\mathbb{G}$

$$\ldots, g^{-2}, g^{-1}, g^0 = e, g^1 = g, g^2, \ldots .$$

Если все эти степени различны, то элемент g называется *элементом бесконечного порядка*; иначе он называется *элементом конечного порядка*.

Для любого элемента конечного порядка существуют такие числа N, что $g^N = e$. Наименьшее из этих чисел называется порядком n элемента g [14, 15].

Также верно утверждение, что для любого элемента g порядка n равенство $g^m = e$ имеет место тогда и только тогда, когда m делится на n [14, 15].

В группе $\mathbb{G}$ порядок 1 имеет только единичный элемент e [14, 15].

Соответственно, выделяют *конечные* группы, состоящие из конечного числа элементов, и *бесконечные*. Количество элементов в конечной группе, называется ее *порядком* [14].

Аксиома G.5. *Аксиома коммутативности.* Для двух произвольных элементов a и b из $\mathbb{G}$ справедливо[13, 14, 15]

$$a \bullet b = b \bullet a.$$

Если кроме аксиом G.1–G.4 выполняется *аксиома коммутативности* G.5, то группа называется *коммутативной* или *абелевой* [4, 13, 14].

В качестве примера аддитивной группы можно привести совокупность действительных чисел. Единичным элементом при этом является ноль. Множество всех действительных чисел без нуля образует мультипликативную группу. Единичным элементом является 1, а обратным $\frac{1}{a}$ [13].

Другим примером группы является совокупность двоичных n-символьных комбинаций, которая образует группу из 2^n элементов вокруг операции

сложения по модулю 2. Единичным является элемент, состоящий из нулей (например, 0000), а обратный элемент равен самому элементу ($0101 \oplus 0101 = 0000$) [13].

6.2. Подгруппы и смежные классы

Подмножество элементов группы $\mathbb{G}$ называется *подгруппой* $\mathbb{H}$, если оно удовлетворяет всем аксиомам группы. Для того чтобы определить, является ли $\mathbb{H}$ подгруппой $\mathbb{G}$, надо проверить только замкнутость операции (G.1) и наличие обратных элементов (G.4) в подмножестве $\mathbb{H}$. Например, множество целых чисел является подгруппой группы из множества действительных чисел [13, 14, 15].

Таким образом, любая подгруппа автоматически является группой [14].

Подмножество группы $\mathbb{G}$, состоящее из ее единицы e, а также сама группа $\mathbb{G}$ тоже являются подгруппами. Они получили название *тривиальных подгрупп* [14].

Важным классом подгрупп являются *циклические подгруппы* [1]. *Циклической подгруппой* группы $\mathbb{G}$ называется подгруппа $\mathbb{H}$ порядка m, состоящая из элементов $(h, h^2, h^3, \ldots, h^{m-1}, h^m = e) = (e, h, h^2, h^3, \ldots, h^{m-1})$ [13].

Для произвольной группы $\mathbb{G}$ и ее подгруппы $\mathbb{H}$ подмножество группы $\mathbb{G}$, состоящее из всех элементов вида $h \bullet g$ (или $g \bullet h$), где h — прозвольный элемент подгруппы $\mathbb{H}$, а g — некоторый фиксированный элемент группы $\mathbb{G}$, называется *смежным классом* элемента g по подгруппе $\mathbb{H}$ и обозначается через $\mathbb{H}g$ (или $g\mathbb{H}$) [14].

Смежные классы, образованные операцией $h \bullet g$, получили название *правых смежных классов* ($\mathbb{H}g$), а классы, образованные операцией $g \bullet h$ — *левых смежных классов* ($g\mathbb{H}$) [13]. Для абелевых групп правые и левые смежные классы совпадают [13, 1].

Отдельный элемент $g \in g\mathbb{H}$ называется *представителем смежного класса* $g\mathbb{H}$ [1].

Для смежных классов верен ряд **теорем** [14].

1. *Смежный класс* $\mathbb{H}g'$ *любого элемента* g' *из смежного класса* $\mathbb{H}g$ *совпадает с классом* $\mathbb{H}g$. *То есть, если два смежных класса пересекаются, то они совпадают.*

2. *Два элемента* g_1 *и* g_2 *группы* $\mathbb{G}$ *тогда и только тогда принадлежат одному смежному классу по подгруппе* $\mathbb{H}$, *когда* $g_1 \bullet g_2^{-1} \in \mathbb{H}$.

3. *Смежный класс* $\mathbb{H}g$ *тогда и только тогда совпадает с подгруппой* $\mathbb{H}$, *когда* $g \in \mathbb{H}$.

Для подгруппы $\mathbb{H}$ конечной группы $\mathbb{G}$ верна *теорема Лагранжа*. *Порядок конечной группы делится на порядок любой ее подгруппы. Соответ-*

ствующее частное равно индексу подгруппы. При этом, под индексом подгруппы понимается число смежных классов по подгруппе $\mathbb{H}$ [14].

Подгруппа $\mathbb{H}$ группы $\mathbb{G}$ называется *нормальным делителем*, если для любого элемента $h \in \mathbb{H}$ и любого элемента $g \in \mathbb{G}$ элемент $g \bullet h \bullet g^{-1}$ принадлежит $\mathbb{H}$. Любая подгруппа абелевой группы является конечным делителем [14].

Для примера возьмём бесконечную группу целых чисел $\mathbb{Z}$. В ней возьмем подгруппу $5\mathbb{Z}$ из чисел, кратных 5. Индекс подгруппы равен 5. Получим следующие смежные классы

$$\begin{aligned}
0+5\mathbb{Z} &= \{\ldots,-15,-10,-5,0,5,10,15,\ldots\};\\
1+5\mathbb{Z} &= \{\ldots,-14,-9,-4,1,6,11,16,\ldots\};\\
2+5\mathbb{Z} &= \{\ldots,-13,-8,-3,2,7,12,17,\ldots\};\\
3+5\mathbb{Z} &= \{\ldots,-12,-7,-2,3,8,13,18,\ldots\};\\
4+5\mathbb{Z} &= \{\ldots,-11,-6,-1,4,9,14,19,\ldots\}.
\end{aligned}$$

В качестве другого примера возьмем конечную циклическую группу $\mathbb{G}$ поворотов на углы, кратные $\frac{360^\circ}{9}$

$$\mathbb{G} = \{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\},$$

где g_i — поворот на $\frac{i \cdot 360^\circ}{9}$.

Группа $\mathbb{H} = \{g_0, g_3, g_6\}$ является подгруппой группы $\mathbb{G}$, а подгруппы

$$\begin{aligned}
\mathrm{g}_0\mathbb{H} &= \{g_0, g_3, g_6\},\\
\mathrm{g}_1\mathbb{H} &= \{g_1, g_4, g_7\},\\
\mathrm{g}_2\mathbb{H} &= \{g_2, g_5, g_8\}
\end{aligned}$$

являются левыми смежными классами группы $\mathbb{G}$ по подгруппе $\mathbb{H}$. При этом $\mathbb{G} = \mathrm{g}_0\mathbb{H} \cup \mathrm{g}_1\mathbb{H} \cup \mathrm{g}_2\mathbb{H}$.

6.3. Кольцо

Кольцом $\mathbb{R}$ называется множество элементов, на котором определены две операции — сложение и умножение, и выполняется ряд аксиом [13, 1].

Аксиома R.1. Множество $\mathbb{R}$ является *аддитивной абелевой группой.*

Аксиома R.2. *Замкнутость операции умножения.* Для любых двух элементов $a, b \in \mathbb{R}$ определено их произведение

$$a \cdot b = c \in \mathbb{R}.$$

Аксиома R.3. Для любых трех элементов $a, b, c \in \mathbb{R}$ выполняется *ассоциативный закон*

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c, \qquad a + (b + c) = (a + b) + c.$$

Аксиома R.4. Для любых трех элементов $a, b, c \in \mathbb{R}$ выполняется *дистрибутивный закон*

$$a \cdot (b + c) = a \cdot b + a \cdot c, \qquad (b + c) \cdot a = b \cdot a + c \cdot a.$$

Аксиома R.5. В кольце существует элемент e, называемый *единицей кольца*, который является нейтральным элементом относительно умножения, т. е. $e \cdot a = a \cdot e = a$ для любого $a \in \mathbb{R}$ [1].

Элемент $a \neq 0$ кольца R называется *делителем нуля*, если в $\mathbb{R}$ существует такой элемент $b \neq 0$, что $a \cdot b = 0$ [1].

В кольце для операции умножения аксиомы G.3, G.4 и G.5 (п. 6.1) могут не выполняться. Если же операция умножения коммутативна в кольце, то такое кольцо называется *коммутативным*. Если в кольце существует единичный элемент относительно операции умножения (выполняется аксиома G.3), то это кольцо называется *кольцом с единицей* [13].

Примером кольца являются все целые положительные и отрицательные числа и нуль, образующие коммутативное кольцо с единицей относительно обычных операций сложения и умножения [13].

6.4. Поле

Полем $\mathbb{F}$ называют коммутативное кольцо с единицей, в котором каждый ненулевой элемент имеет мультипликативный обратный элемент (т. е. обратный по умножению) [13, 1].

Другими словами, полем называют множество, которое является аддитивной абелевой группой; ненулевые же элементы этого множества образуют мультипликативную абелевую группу, и выполняется закон дистрибутивности [13].

По аналогии с группами число элементов поля называется *порядком* поля. Поля, порядки которых конечны, называются *конечными полями* или *полями Галуа*. Конечные поля имеют наибольшее значение в теории кодирования [13]. Подробнее поля Галуа рассмотрены в разд. 7

Поле имеет ряд свойств, вытекающих из его определения [13].

F.1. Для любого элемента поля $a \cdot 0 = 0 \cdot a = 0$.
F.2. Для элементов $a, b \in \mathbb{F}$, не равных нулю, $a \cdot b \neq 0$.
F.3. Для любых элементов $a, b \in \mathbb{F}$ $a + b \neq 0$.
F.4. Если $a \cdot b = a \cdot c$ и $a \neq 0$, то $b = c$.

Примером поля является множество чисел $(0, 1, 2, \ldots, p-1)$, где p — простое число, образующее конечное поле, в котором сложение и умножение производятся по модулю p [13].

Контрольные вопросы

1. Что такое группа?
2. В чем заключается аксиома коммутативности?
3. Дайте понятие подгруппы.
4. Что такое смежные классы? Приведите пример.
5. Сформулируйте теорему Лагранжа.
6. Что такое нормальный делитель группы?
7. Что такое кольцо?
8. Какое кольцо называется кольцом с единицей?
9. Что такое поле?

7. МАТЕМАТИКА ПОЛЕЙ ГАЛУА

7.1. Поле Галуа и его свойства

В теории помехоустойчивого кодирования широко используются конечные поля, называемые *полями Галуа* в честь французского математика Эвариста Галуа (1811–1832), чьи работы легли в основу теории групп.

Теория полей Галуа подробно освещены во многих работах, как относительно давно написанных трудах Н. Г. Чеботарёва [16, 17, 18], М. М. Постникова [14, 15], Р. Лиддла и Г. Нидеррайтера [19] и других авторов, так и недавно вышедших трудах, таких как монография О. С. Когновицкого [4].

Поле Галуа, обозначаемое $\mathrm{GF}(q)$, представляет собой конечное множество, состоящее из q элементов, обладающих свойствами поля. Число элементов поля q является простым числом или степенью простого числа. Если q — простое число, то элементами поля $\mathrm{GF}(q)$ с *характеристикой* q являются числа 0, 1, 2, …, $(q-1)$. При этом в соответствии со свойствами поля сложение и умножение элементов такого поля осуществляется с приведением по модулю q. Такое поле Галуа называется *простым*.

Если характеристика q является степенью простого числа p, т. е. $q = p^m$, где m — целое, то элементами поля $\mathrm{GF}(p^m)$ будут многочлены степени $(m-1)$ вида

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_{m-1} x^{m-1}, \tag{7.1}$$

где все коэффициенты a_i пробегают полную систему вычетов по модулю p, т. е. принадлежат простому полю $\mathrm{GF}(p)$. В этом случае p называется *основанием* поля, а m — *степенью* поля. Само поле Галуа называется *расширенным*.

В дальнейшем будем рассматривать только поля Галуа по основанию 2 — двоичные поля Галуа $\mathrm{GF}(2^m)$. Соответственно, коэффициенты a_i из формулы (7.1) равны 0 или 1.

7.1.1. Образующий полином поля Галуа

Поле Галуа $\mathrm{GF}(p^m)$ строится на основе так называемого *образующего* (*порождающего*) многочлена $p(x)$, который является неприводимой примитивной функцией степени m.

Важно: Кроме обозначения $p(x)$ для образующего многочлена поля в некоторых источниках используется обозначение $g(x)$ (также используется для обозначения порождающего многочлена кода). Здесь, чтобы избежать путаницы, мы будем использовать $p(x)$ для образующего многочлена поля, а $g(x)$ для порождающего многочлена кода.

Как уже было сказано выше, существуют готовые таблицы с образующими полиномами. Их можно как найти в литературе [4], так и воспользоваться математическими пакетами, например Matlab или Octave. Если же нет

возможности воспользоваться таблицами, то можно проверить, является ли многочлен неприводимым и примитивным путем построения поля, которое в ином случае построено быть не может. Также возможность построения поля можно оценить, попытавшись определить значение полинома $x^{p^m-1} \bmod p(x)$. Так как поле Галуа является циклическим, то, в случае, если поле может быть построено, остаток от деления x^{p^m-1} на $p(x)$ должен быть равен единице.

В табл. 7.1 приведены примеры образующих полиномов $p(x)$ для некоторых степеней поля $\mathrm{GF}(2^m)$.

Таблица 7.1

Примеры образующих полиномов $p(x)$ поля $\mathrm{GF}(2^m)$

Степень	**Полином**	**Степень**	**Полином**
1	$x+1$	2	x^2+x+1
3	x^3+x+1 x^3+x^2+1	4	x^4+x+1 x^4+x^3+1
5	x^5+x^2+1 x^5+x^3+1 $x^5+x^3+x^2+x+1$ $x^5+x^4+x^2+x+1$ $x^5+x^4+x^3+x+1$ $x^5+x^4+x^3+x^2+1$	6	x^6+x+1 $x^6+x^4+x^3+x+1$ x^6+x^5+1 $x^6+x^5+x^2+x+1$ $x^6+x^5+x^3+x^2+1$ $x^6+x^5+x^4+x+1$

7.1.2. Левый степенной базис и представление элементов поля

На практике элементы поля Галуа представляются в трех видах.

1. Степень первого элемента поля.
2. «Двоичное» (полиномиальное) представление элемента поля через левый степенной базис.
3. Десятичное представление элемента поля.
4. Представление элемента поля через двойственный базис.

Первый элемент поля Галуа будем обозначать как ε, тогда остальные элементы поля будут рассматриваться как его степени.

$$\varepsilon, \varepsilon^2, \varepsilon^3, \ldots, \varepsilon^{p^m-1}.$$

Так как поле Галуа является циклическим, $\varepsilon^{p^m} = \varepsilon$.

Одним из свойств расширенного поля Галуа $\mathrm{GF}(p^m)$ является то, что любой элемент поля может быть выражен суммой из m элементов поля. Как правило, для выражения элементов используются первые m элементов

$$[1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{m-1}], \tag{7.2}$$

получившие название *левый степенной базис* (*ЛСБ*). В различных источниках применяются также термины *полиномиальный базис* и *стандартный базис*.

Полиномиальное представление элемента поля через левый степенной базис выражается по формуле

$$a_0 + a_1\varepsilon + a_2\varepsilon^2 + \ldots + a_{m-1}\varepsilon^{m-1},$$

где коэффициенты a_i принадлежат простому полю $\mathrm{GF}(p)$.

По аналогии с полиномом, это выражение может быть представлено в виде вектор-строки коэффициентов

$$[a_0, a_1, a_2, \ldots, a_{m-1}].$$

Соответственно, элемент поля представляется также и в виде многочлена

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{m-1}x^{m-1}. \tag{7.3}$$

В случае рассматриваемых нами двоичных полей, каждый из коэффициентов a_i может быть равен 0 или 1. И вектор-строка может быть записана как двоичное число («двоичный» полином), которое может быть преобразовано в десятичное по стандартному правилу преобразования двоичного числа в десятичное (пп. 2.2.1).

Следует отметить, что в математических системах Matlab/Octave для двоичных чисел по умолчанию используется обратная запись, от младшей степени к старшей (слева-направо). Соответственно, преобразование в десятичный вид и обратно будет иным.

Для примера приведем поле Галуа $\mathrm{GF}(2^3)$, построенное на основе полинома $p(x) = x^3 + x + 1$ (табл. 7.2).

Таблица 7.2

Поле Галуа $\mathrm{GF}(2^3)$. $p(x) = x^3 + x + 1$.

Элемент поля	**Полином** $a_0 + a_1 x + a_2 x^2$	**Двоичный вид** $\{a_0 a_1 a_2\}$	**Десятичный вид обычный**	**Десятичный вид (Matlab/Octave)**
$\varepsilon^0 = 1$	1	`100`	4	1
ε	x	`010`	2	2
ε^2	x^2	`001`	1	4
ε^3	$1 + x$	`110`	6	3
ε^4	$x + x^2$	`011`	3	6
ε^5	$1 + x + x^2$	`111`	7	7
ε^6	$1 + x^2$	`101`	5	5

7.1.3. Пример построения поля Галуа

Для примера рассмотрим процесс построения поля Галуа $\mathrm{GF}(2^3)$ с образующим полиномом $p(x) = x^3 + x + 1$, показанного в табл. 7.2.

Каждый элемент поля ε^i можно представить в виде соответствующего ему полинома $x^i \bmod p(x)$. То есть, каждому элементу поля ε^i соответствует полином, равный остатку от деления x^i на $p(x)$.

Первые элементы поля $\varepsilon^0 = 1$, ε, ε^2 образуют левый степенной базис поля. Им соответствуют двоичные полиномы с тем же значением показателя степени.

Для определения оставшихся элементов поля просто делим соответствующие им полиномы на $p(x)$ и берем остаток от деления. Например,

$$\frac{x^3}{x^3+x+1} = 1 + \frac{x+1}{x^3+x+1} \quad \Rightarrow \quad \varepsilon^3 = 1+x.$$

Заметим, что элемент поля, следующий за левым степенным базисом, равен образующему полиному поля за вычетом одночлена старшей степени x^m, который для рассматриваемого примера равен x^3.

Также, для определения элементов поля можно использовать и другой способ. Зная элемент ε^3, дальнейшие элементы поля получаем путем умножения элемента предыдущей степени на элемент ε и приведением получившегося результата к левому степенному базису. По другому этот процесс можно описать как побитовый сдвиг значения элемента поля влево (вправо, если говорить о форме представления в Octave) со сложением по модулю 2 с ε^3 при переносе единицы за границу элемента поля:

$$\varepsilon^4 = \varepsilon^3 \cdot \varepsilon = (1+x) \cdot x = x + x^2 = 110_2;$$
$$\varepsilon^5 = \varepsilon^4 \cdot \varepsilon = (x+x^2) \cdot x = x^2 + x^3 = (1+x) + x^2 = 1 + x + x^2 = 111_2;$$
$$\varepsilon^6 = \varepsilon^5 \cdot \varepsilon = (1+x+x^2) \cdot x = x + x^2 + x^3 = x + x^2 + 1 + x = 1 + x^2 = 101_2;$$
$$\varepsilon^7 = \varepsilon^6 \cdot \varepsilon = (1+x^2) \cdot x = x + x^3 = x + 1 + x = 1 = \varepsilon^0 = 001_2.$$

Можно видеть, что поле циклично. Элемент поля ε^7 равен элементу поля ε^0.

7.1.4. Характеристическая матрица

Каждому элементу поля ε^k взаимнооднозначно соответствует многочлен:

$$\mathbf{F}^k = a_0\mathbf{E} + a_1\mathbf{F} + a_2\mathbf{F}^2 + \ldots + a_{m-1}\mathbf{F}^{m-1}, \tag{7.4}$$

где $\mathbf{F}$ — *характеристическая матрица*, а $\mathbf{E} = \mathbf{F}^0$ — *единичная матрица*.

Характеристическая матрица $\mathbf{F}$ для образующего полинома $p(x) = x^3 + x + 1$ равна

$$\mathbf{F} = \begin{pmatrix} \varepsilon \\ \varepsilon^2 \\ \varepsilon^3 \end{pmatrix} = \begin{pmatrix} 010 \\ 100 \\ 011 \end{pmatrix}.$$

Матрицу $\mathbf{F}^i$ для поля $\mathrm{GF}(2^m)$ можно представить в общем виде:

$$\mathbf{F}^i = \begin{pmatrix} \varepsilon^i \\ \varepsilon^{i+1} \\ \vdots \\ \varepsilon^{i+m-1} \end{pmatrix}. \tag{7.5}$$

Для рассмотренного выше в табл. 7.2 поля матрица F^3 равна

$$\mathbf{F}^3 = \begin{pmatrix} \varepsilon^3 \\ \varepsilon^4 \\ \varepsilon^5 \end{pmatrix} = \begin{pmatrix} 011 \\ 110 \\ 111 \end{pmatrix}.$$

7.1.5. Двойственный базис поля

В работах [20] и [21] показано, что для любого степенного базиса $(\gamma_1, \gamma_2, \ldots, \gamma_m)$ поля Галуа $\mathrm{GF}(p^m)$ существует *двойственный* ему базис $(\omega_1, \omega_2, \ldots, \omega_m)$, который также позволяет выразить все элементы поля Галуа.

В [20] доказано, что базис $(\omega_1, \omega_2, \ldots, \omega_m)$ двойственный степенному базису $(\gamma_1, \gamma_2, \ldots, \gamma_m)$, равному $(\varepsilon^n, \varepsilon^{n+1}, \ldots, \varepsilon^{n+m-1})$, рассчитывается по формуле:

$$\omega_i = \varepsilon^{-n}\alpha_i; \quad i = 1, 2, \ldots, m, \tag{7.6}$$

где коэффициенты α_i равны [20]:

$$\alpha_i = \frac{\sum_{l=0}^{m-j} p_{m-j-l}\varepsilon_i^l}{p'(\varepsilon_i)}; \quad \mathrm{GF}(2^m), \tag{7.7}$$

где p_j — коэффициенты характеристического многочлена $p(x)$.

При использовании левого степенного базиса $(1, \varepsilon, \ldots, \varepsilon^{m-1})$, элементы ω_i двойственного ему базиса совпадают с коэффициентами α_i.

В некоторых работах, например [21] и [22], двойственный базис называют *дополняющим* или *взаимным* базисом.

7.1.6. Свойства полей Галуа

1. Все отличные от нуля элементы поля $\mathrm{GF}(p^m)$ образуют мультипликативную группу порядка $p^m - 1$. Тогда для любого элемента поля ε имеет место равенство

$$\varepsilon^{p^m-1} = 1. \tag{7.8}$$

2. В поле $\mathrm{GF}(p^m)$ всегда существует первообразный элемент ε, т. е. элемент порядка $p^m - 1$. Каждый ненулевой элемент поля может быть представлен как некоторая степень одного и того же первообразного элемента ε. Иными словами: мультипликативная группа поля Галуа циклична.

3. Любой элемент поля $\mathrm{GF}(p^m)$ можно представить в виде

$$a_0 + a_1\varepsilon + a_2\varepsilon^2 + \ldots + a_{m-1}\varepsilon^{m-1}. \tag{7.9}$$

4. В поле $\mathrm{GF}(p^m)$ имеет место равенство

$$(a+b)^p = a^p + b^p \pmod{p}. \tag{7.10}$$

5. Если элемент ε поля $\mathrm{GF}(p^m)$ есть корень неприводимого по модулю p многочлена $P(x)$ степени m, то остальными корнями $P(x)$ будут элементы $\varepsilon^p, \varepsilon^{p^2}, \ldots, \varepsilon^{p^{m-1}}$.

6. Для любого простого числа p и любого примитивного многочлена $P(x)$ степени m, т. е. минимального многочлена примитивного элемента поля $\mathrm{GF}(p^m)$ [19], существует только одно поле Галуа $\mathrm{GF}(p^m)$, иными словами, поля Галуа $\mathrm{GF}(p^m)$, образованные различными неприводимыми примитивными многочленами степени m, изоморфны.

7.2. Основные действия над элементами поля

7.2.1. Логарифмирование и антилогарифмирование

Как было показано в пп. 7.1.2, любой элемент поля ε^i может быть представлен двоичным вектором или многочленом вида (7.3).

Возьмем логарифм по основанию ε от элемента поля, представленного в виде ε^i. Получим равенство:

$$\log_\varepsilon \varepsilon^i = i. \tag{7.11}$$

Операция, реализованная по такому правилу, называется *логарифмированием*. Для целочисленных значений i данный вид логарифма принято называть *дискретным*.

Смысл операции (7.11) состоит в поиске показателя степени i при ε^i, т. е. для некоторого элемента поля, заданного в виде вектора. Показатель степени может принимать значения $i = 0, 1, 2, \ldots, 2^m - 2$, где 2^m — порядок поля.

При выполнении операции логарифмирования, необходимо учитывать одну особенность. В любом поле Галуа присутствует нулевой элемент, логарифм которого не определен. Следовательно, устройство или алгоритм, реализующие эту операцию должны обрабатывать это исключение. Например, при программной реализации, учитывая, что степени любых элементов поля можно свести к диапазону $0, 1, \ldots, 2^m - 2$, для логарифма нуля можно использовать условное обозначение «-1».

Операция, противоположная операции (7.11), называется *антилогарифмированием*. В этом случае необходимо найти вектор при известном показателе степени элемента. То есть при заданном i элемента поля ε^i определяется соответствующий ему вектор.

Для примера возьмем элемент $\varepsilon^4 = 011$ поля $\mathrm{GF}(2^3)$, представленного в табл. 7.2. Из формулы (7.11) следует, что логарифм ε^4 равен $\log_\varepsilon 011 = \log_\varepsilon \varepsilon^4 = 4$, а антилогарифм 4 равен $\mathrm{antilog}_\varepsilon 4 = \varepsilon^4 = 011$.

Логарифм и антилогарифм применяются для проведения базовых операций над элементами поля.

Различные варианты реализации операций логарифмирования и антилогарифмирования рассмотрены в [20].

На практике при написании программ применяют два основных способа логарифмирования/антилогарифмирования. Первый способ — *табличный*. В этом случае заранее создаются две таблицы. В одной содержится зависимость значения элемента от его степени, в другой — зависимость степени элемента от его значения. Операции реализуются простой выборкой по индексу из соответствующей таблицы. Для больших полей способ требует большого объема памяти. Второй способ заключается в последовательной генерации элементов поля до получения необходимого значения. Для больших полей требуются значительные временные затраты. Сотрудником СПбГУТ Дмитрием Кукуниным был предложен способ логарифмирования, сочетающий в себе эти два метода. При этом создается таблица с контрольными точками и производится динамическая генерация элементов до получения необходимого. Этот способ является оптимальным для больших полей как по быстродействию, так и по требуемой для хранения таблиц памяти [23].

7.2.2. Сложение элементов поля

Для того, чтобы произвести сложение элементов поля ε^i и ε^j, необходимо выразить эти элементы через левый степенной базис:

$$\begin{aligned} \varepsilon^i &= a_0 + a_1\varepsilon + a_2\varepsilon^2 + \ldots + a_{m-1}\varepsilon^{m-1} \Leftrightarrow \{a_0 a_1 a_2 \ldots a_{m-1}\}; \\ \varepsilon^j &= b_0 + b_1\varepsilon + b_2\varepsilon^2 + \ldots + b_{m-1}\varepsilon^{m-1} \Leftrightarrow \{b_0 b_1 b_2 \ldots b_{m-1}\}. \end{aligned} \tag{7.12}$$

Результатом сложения элементов (7.12) будет элемент

$$\varepsilon^k = \varepsilon^i + \varepsilon^j = c_0 + c_1\varepsilon + c_2\varepsilon^2 + \ldots + c_{m-1}\varepsilon^{m-1} \Leftrightarrow \{c_0 c_1 c_2 \ldots c_{m-1}\}, \tag{7.13}$$

где $c_i = a_i + b_i (\mathrm{mod}\ p)$.

Схема устройства, реализующего сложение элементов, представленных в виде вектора, состоит из сумматора, поэлементно складывающего эти векторы по mod 2.

Сложение элементов (7.12), в случае, когда они представлены их степенями i и j, требует преобразований и, соответственно, наличия в схеме блоков, производящих операции логарифмирования и антилогарифмирования (рис. 7.1) [20].

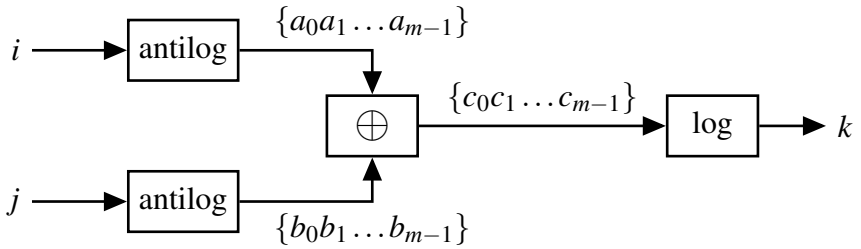


Рис. 7.1. Схема сложения элементов поля, представленных индексами (степенями)

Нужно отметить, что в том случае, когда складываемые элементы равны, т. е. имеют одинаковую по модулю (p^m-1) степень, их сумма будет равна нулю. В этом случае, как было отмечено в пп. 7.2.1, блок логарифмирования (рис. 7.1) должен обработать это исключение и передать на выход особый сигнал, соответствующий нулевому элементу.

7.2.3. Перемножение элементов поля

Умножение элементов поля (7.12), выраженных через левый степенной базис, производится с учётом выражений (7.9) и (7.4):

$$\begin{aligned}\varepsilon^k &= \varepsilon^i\cdot\varepsilon^j = \{a_0a_1\ldots a_{m-1}\}\cdot F^j = \\ &= \{a_0a_1\ldots a_{m-1}\}\cdot(b_0E+b_1F+\ldots+b_{m-1}F^{m-1}) = \\ &= b_0(\{a_0a_1\ldots a_{m-1}\}E)+\ldots+b_{m-1}(\{a_0a_1\ldots a_{m-1}\}F^{m-1}) = \\ &= \{c_0c_1\ldots c_{m-1}\}.\end{aligned} \tag{7.14}$$

На рис. 7.2 показана схема, реализующая умножение двух элементов поля $\mathrm{GF}(2^m)$, согласно формуле (7.14) [20].

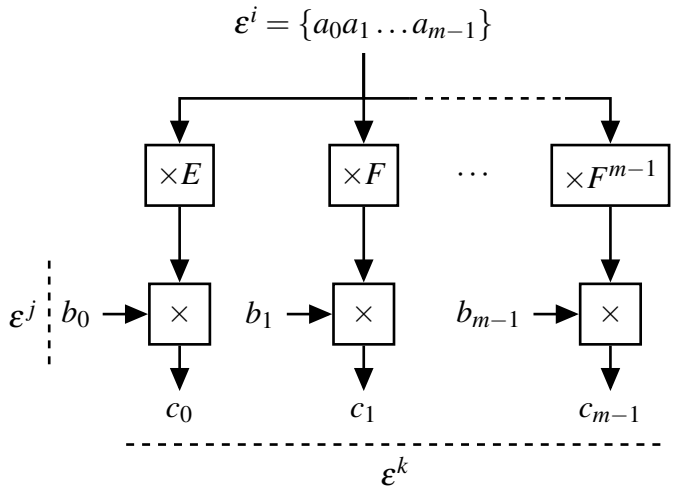


Рис. 7.2. Структурная схема прямого умножителя элементов в поле $\mathrm{GF}(2^m)$

В случае, когда элементы поля (7.12) выражены их показателями степени i и j, их произведение определяется путём сложения этих показателей степеней элементов:

$$\varepsilon^k = \varepsilon^i\cdot\varepsilon^j = \varepsilon^{i+j}, \tag{7.15}$$

где $i + j$ приводится по mod $(2^m - 1)$.

Для того, чтобы произвести умножение элементов поля, выраженных через левый степенной базис, согласно формуле (7.15), необходимо сначала произвести логарифмирование элементов поля (7.12), затем сложить показатели степеней и результат перевести в вектор с помощью операции антилогарифмирования. На рис. 7.3 изображена структурная схема логарифмического умножителя элементов, реализующего данный принцип [20].

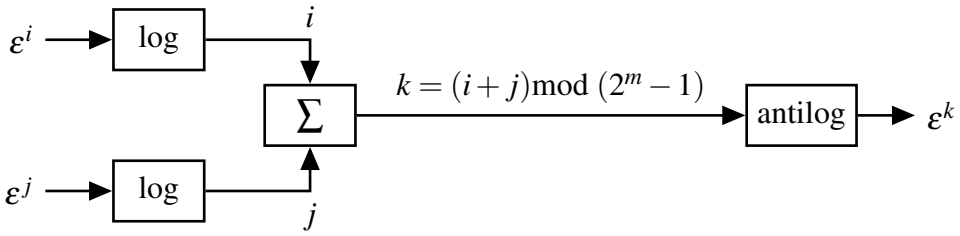


Рис. 7.3. Структурная схема умножения элементов поля через операции логарифмирования и антилогарифмирования

Необходимо учесть, что в том случае, когда один из множителей равен нулю, результат также равен нулю, а проведение операций логарифмирования невозможно. Следовательно, устройство или программный алгоритм, реализующие схему умножения, приведенную на рис. 7.3, должны обрабатывать этот вариант.

7.2.4. Возведение элемента поля в степень

Возведение элемента поля ε^i в степень j осуществляется по формуле:

$$(\varepsilon^i)^j = \varepsilon^{i \cdot j}, \tag{7.16}$$

где произведение $i \cdot j$ приводится по модулю $2^m - 1$.

Схема реализации операции возведения в степень с использованием операций логарифмирования и антилогарифмирования приведена на рис. 7.4 [20].

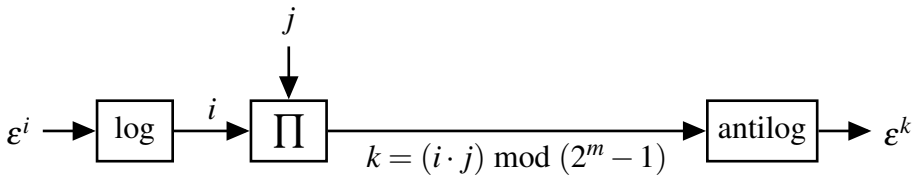


Рис. 7.4. Структурная схема возведения элемента поля в степень

Так же, как и в случае логарифмического умножителя, устройство или программный алгоритм, реализующие логарифмическую схему операции возведения в степень, должны учитывать случай, когда элемент поля ε^i будет равен нулю. Результат при этом также будет равен нулю.

Отдельного внимания заслуживает операция возведения в квадрат. Возведение элемента поля ε^i, представление которого через левый степенной базис показано в (7.12), в квадрат можно представить в соответствии с (7.10) как [20]:

$$\begin{aligned}(\varepsilon^i)^2 &= (a_0 + a_1\varepsilon + a_2\varepsilon^2 + \ldots + a_{m-1}\varepsilon^{m-1})^2 = \\ &= a_0 + a_1\varepsilon^2 + a_2(\varepsilon^2)^2 + \ldots + a_{m-1}(\varepsilon^{m-1})^2.\end{aligned} \tag{7.17}$$

Соответственно, формулу (7.17) можно представить в векторном виде как

$$\{a_0 a_1 a_2 \ldots a_{m-1}\}\mathbf{X}_2 = \{c_0 c_1 c_2 \ldots c_{m-1}\}, \tag{7.18}$$

где $\mathbf{X}_2$ — матрица порядка m равная

$$\mathbf{X}_2 = \begin{pmatrix} \varepsilon^0 \\ \varepsilon^2 \\ \varepsilon^4 \\ \vdots \\ \varepsilon^{2(m-1)} \end{pmatrix}.$$

Аналогично реализуются операции возведения в четвёртую, восьмую и более высокие степени, показатель которых равен степени двойки [20]. Соответствующие матрицы равны

$$\mathbf{X}_4 = \begin{pmatrix} \varepsilon^0 \\ \varepsilon^4 \\ \varepsilon^8 \\ \vdots \\ \varepsilon^{4(m-1)} \end{pmatrix}, \qquad \mathbf{X}_8 = \begin{pmatrix} \varepsilon^0 \\ \varepsilon^8 \\ \varepsilon^{16} \\ \vdots \\ \varepsilon^{8(m-1)} \end{pmatrix}.$$

7.2.5. Обращение элемента поля

Как показано в п. 7.1, поле Галуа представляет собой конечное множество, состоящее из элементов, обладающих свойствами поля, а это значит, что каждому элементу поля ε^i соответствует обратный ему элемент ε^{-i}.

Как показано в пп. 7.1.6 ненулевые элементы конечного поля $\mathrm{GF}(2^m)$ образуют циклическую мультипликативную группу порядка $2^m - 1$, следовательно для любого ненулевого элемента ε^i, согласно малой теореме Ферма [24], справедливо равенство:

$$(\varepsilon^i)^{2^m-1} = 1. \tag{7.19}$$

Как следует из (7.19), для нахождения ε^{-i} нужно элемент ε^i возвести в степень $2^m - 2$:

$$(\varepsilon^i)^{2^m-2} = \varepsilon^{-i}. \tag{7.20}$$

7.2.6. *Деление элементов поля*

Операция деления элементов поля (7.12) может быть представлена следующим образом:

$$\varepsilon^k = \frac{\varepsilon^i}{\varepsilon^j} = \varepsilon^i \cdot \varepsilon^{-j}, \tag{7.21}$$

где ε^{-j} — элемент поля, обратный элементу ε^j.

Таким образом, операция деления элементов поля может быть реализована через операцию умножения с обращением делителя. На рис. 7.5 представлена схема, реализующая данный принцип.

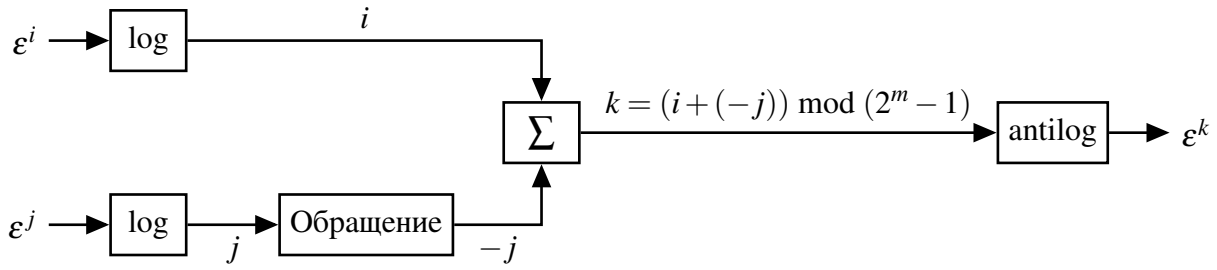


Рис. 7.5. Структурная схема деления элементов поля

Так же, как и в случае логарифмического умножителя, устройство или программный алгоритм, реализующие логарифмическую схему операции возведения в степень должны учитывать случай, когда элемент поля ε^i будет равен нулю. Результат при этом также будет равен нулю. Ещё одним исключением является равенство нулю делителя ε^j. В этом случае алгоритм или схема деления должны выдавать ошибку.

7.2.7. *Логарифмы Зеча*

В некоторых случаях неудобно производить последовательные преобразования элементов поля путем операций логарифмирования и антилогарифмирования при осуществлении операций над элементами поля. В таком случае может применяться подход, состоящий в использовании логарифмов Зеча [2, 22].

Логарифм Зеча $Z(n)$ задаётся равенством

$$\varepsilon^{Z(n)} = 1 + \varepsilon^n. \tag{7.22}$$

При использовании этого метода, можно выполнять арифметические операции в конечном поле, работая только с логарифмами и не обращаясь к антилогарифмам.

Вычисление суммы элементов поля ε^m и ε^n при использовании логарифмов Зеча производится по формуле

$$\varepsilon^m + \varepsilon^n = \varepsilon^m \cdot (1 + \varepsilon^{n-m}) = \varepsilon^{m+Z(n-m)}. \tag{7.23}$$

В табл. 7.3 приведен логарифм Зеча в поле $\mathrm{GF}(2^4)$ [22].

Таблица 7.3

Логарифм Зеча в поле $\mathrm{GF}(2^4)$, *образованном полиномом* $p(x) = x^4 + x + 1$.

Степень элемента n	Логарифм Зеча $Z(n)$	Степень элемента n	Логарифм Зеча $Z(n)$
$-\infty$	0	7	9
0	$-\infty$	8	2
1	4	9	7
2	8	10	5
3	14	11	12
4	1	12	11
5	10	13	6
6	13	14	3

Для примера рассмотрим сложение двух элементов поля $\mathrm{GF}(2^4)$, образованного полиномом $p(x) = x^4 + x + 1$:

$$a = \varepsilon^4 = 1100 \qquad \text{и} \qquad b = \varepsilon^8 = 1010.$$

В сумме элементы a и b дают элемент c, равный

$$c = a + b = \varepsilon^4 + \varepsilon^8 = 1100 + 1010 = 0110 = \varepsilon^5.$$

Теперь рассчитаем элемент c через логарифм Зеча (табл. 7.3) по формуле (7.23):

$$c = \varepsilon^4 + \varepsilon^8 = \varepsilon^{4+Z(8-4)} = \varepsilon^{4+Z(4)} = \varepsilon^{4+Z(4)} = \varepsilon^{4+1} = \varepsilon^5.$$

Можно видеть, что результаты, полученные с помощью сложения посредством логарифмирования и антилогарифмирования, и результаты расчета через логарифм Зеча совпадают.

7.3. Алгоритмы для проведения расчетов в двоичных полях Галуа и их реализации

7.3.1. Алгоритм обращения элемента поля на основе расширенного алгоритма Евклида

Алгоритм Евклида для двух полиномов предназначен для определения их наибольшего общего делителя (НОД), а расширенный алгоритм Евклида позволяет найти два полинома $u(x)$ и $v(x)$, удовлетворяющих формуле (7.24) [25]:

$$\text{НОД}(a(x), b(x)) = u(x)a(x) + v(x)b(x). \tag{7.24}$$

Алгоритм обращения элемента поля, представленного полиномом $a(x)$, показан в алг. 7.1 [25].

Алгоритм 7.1. Алгоритм обращения элемента поля на основе расширенного алгоритма Евклида

$s(x) = p(x)$; // $p(x)$ — образующий полином поля
$r(x) = a(x)$;
$v(x) = 0$;
$u(x) = 1$;
while $\deg(r(x)) \neq 0$ **do**
    $\delta = \deg(s(x)) - \deg(r(x))$;
    if $\delta < 0$ **then**
        $s(x) \rightleftarrows r(x)$;
        $v(x) \rightleftarrows u(x)$;
        $\delta = -\delta$;
    end
    $s(x) = s(x) + x^{\delta} r(x)$;
    $v(x) = v(x) + x^{\delta} u(x)$;
end
Result: $u(x) = a^{-1}(x)$

Утверждается [26], что данный алгоритм быстрее прочих алгоритмов обращения элементов поля [25].

В качестве примера рассмотрим процедуру нахождения обратного элемента для элемента поля Галуа $\text{GF}(2^3)$, образованного порождающим полиномом $p(x) = x^3 + x + 1$. Выберем элемент поля $a(x) = \varepsilon^4 = x^2 + x$ (табл. 7.2).

Начальные значения:
$s(x) = p(x) = x^3 + x + 1$; $v(x) = 0$;
$r(x) = a(x) = x^2 + x$; $u(x) = 1$;

Шаг 1:
$\deg(r(x)) > 0$;
$\delta = \deg(s(x)) - \deg(r(x)) = 3 - 2 = 1$;
$s(x) = s(x) + x^{\delta} r(x) = (x^3 + x + 1) + x(x^2 + x) = x^2 + x + 1$;
$v(x) = v(x) + x^{\delta} u(x) = 0 + x \cdot 1 = x$;

Шаг 2:
$\delta = \deg(s(x)) - \deg(r(x)) = 2-2 = 0;$
$s(x) = s(x) + x^{\delta} r(x) = (x^2+x+1) + (x^2+x) = 1;$
$v(x) = v(x) + x^{\delta} u(x) = x+1;$
Шаг 3:
$\delta = \deg(s(x)) - \deg(r(x)) = 0-2 = -2 < 0 \Rightarrow s(x) \rightleftarrows r(x), v(x) \rightleftarrows u(x);$
$s(x) = x^2+x; \qquad r(x) = 1; \Rightarrow \deg(r(x)) = 0; //$ завершаем работу
$v(x) = 1; \qquad u(x) = x+1;$
$\delta = -\delta = 2;$
$s(x) = s(x) + x^{\delta} r(x) = (x^2+x) + x^2 \cdot 1 = x;$
$v(x) = v(x) + x^{\delta} u(x) = 1 + x^2(x+1);$
Результат: $a^{-1}(x) = u(x) = x+1 = \varepsilon^3.$

7.3.2. Алгоритм умножения двух элементов поля «сдвиг-со-сложением, справа-налево»

Данный алгоритм реализует умножение двух элементов поля, представленных в виде полиномов, в поле Галуа $\mathrm{GF}(2^m)$, образованном полиномом $p(x)$. Приведенное название является калькой с англоязычного обозначения данного алгоритма: «right-to-left shift-and-add field multiplication». Этот алгоритм основан на том, что

$$a \cdot b = a_{m-1} x^{m-1} b + \ldots + a_2 x^2 b + a_1 x b + a_0 b.$$

На каждом шаге i в этом алгоритме вычисляется произведение $x^i b \mod p(x)$ и полученное складывается с произведением c при условии, что $a_i = 1$, как показано в алг. 7.2 [26].

Алгоритм 7.2. Алгоритм умножения двух элементов поля «сдвиг-со-сложением, справа-налево»

Data: Двоичные полиномы $a(x)$ и $b(x)$ степени $\leq m-1$
if $a_0 = 1$ **then**
  $c(x) = b(x);$
else
  $c(x) = 0;$
end
for $i = 1 \ldots m-1$ **do**
  $b(x) = x \cdot b(x) \mod p(x);$
  if $a_i = 1$ **then**
    $c(x) = c(x) + b(x);$
  end
end
Result: $c(x) = a(x)b(x) \mod p(x)$

Для примера рассмотрим перемножение двух элементов поля, представленных в виде полиномов $a(x)$ и $b(x)$, над полем $\mathrm{GF}(2^4)$, образованного полиномом $p(x) = x^4 + x + 1$:

$$a(x) = x^2 + 1 = 1010 = \varepsilon^8;$$
$$b(x) = x^2 + x = 0110 = \varepsilon^5.$$

Поскольку $a_0 = 1$, $c(x) = b(x) = x^2 + x$.
Шаг 1:
$b(x) = x \cdot b(x) \mod p(x) = x^3 + x^2$;
$a_1 = 0 \Rightarrow$ переход на следующий шаг
Шаг 2:
$b(x) = x \cdot b(x) \mod p(x) = (x \cdot (x^3 + x^2)) \mod p(x) = x^3 + x + 1$;
$a_1 = 1 \Rightarrow c(x) = c(x) + b(x) = (x^2 + x) + (x^3 + x + 1) = x^3 + x^2 + 1$;
Шаг 3:
$b(x) = (x \cdot (x^3 + x + 1)) \mod p(x) = x^2 + 1$;
$a_1 = 0 \Rightarrow$ завершаем
Результат: $c(x) = x^3 + x^2 + 1 = 1011 = \varepsilon^{13}$.

Считается, что этот алгоритм хорошо подходит для аппаратной реализации на логических элементах, где сдвиг вектора b реализуется за один такт, и менее применим для программной реализации из-за значительного числа таких сдвигов [26].

7.3.3. Алгоритм Карацубы–Офмана умножения двух элементов поля

Этот алгоритм изначально предназначен для умножения длинных чисел. Он был предложен в 1960 г. А. А. Карацубой и опубликован в 1962 г. в сборнике докладов АН СССР [27]. Этот алгоритм позволил уменьшить оценку сложности умножения n-значных чисел с $O(n^2)$ до $O(n^{\log_2 3})$. Позднее данный алгоритм было предложено использовать для умножения элементов конечного поля, представленных в виде полиномов [28]. Далее будем говорить именно об алгоритме умножения двух элементов поля.

В разных источниках этот алгоритм может называться и как алгоритм Карацубы–Офмана, и как просто алгоритм Карацубы [28, 29].

Обычно выделяют два варианта алгоритма Карацубы–Офмана. Первый вариант алгоритма — это так называемый $2^k n$-битный умножитель. Он предназначен для работы с элементами поля $\mathrm{GF}(2^m)$, где $m = rn$, $r = 2^k$ [28].

Множители A и B, принадлежащие полю $\mathrm{GF}(2^m)$, представляются в виде полиномов через левый степенной базис формулой [28, 29]:

$$\begin{aligned} A(x) &= \sum_{i=0}^{m-1} a_i x^i = \sum_{i=\frac{m}{2}}^{m-1} a_i x^i + \sum_{i=0}^{\frac{m}{2}-1} a_i x^i = \\ &= x^{\frac{m}{2}} \sum_{i=0}^{\frac{m}{2}-1} a_{i|\frac{m}{2}} x^i + \sum_{i=0}^{\frac{m}{2}-1} a_i x^i = x^{\frac{m}{2}} A^H + A^L, \\ B(x) &= \sum_{i=0}^{m-1} b_i x^i = \sum_{i=\frac{m}{2}}^{m-1} b_i x^i + \sum_{i=0}^{\frac{m}{2}-1} b_i x^i = \\ &= x^{\frac{m}{2}} \sum_{i=0}^{\frac{m}{2}-1} b_{i|\frac{m}{2}} x^i + \sum_{i=0}^{\frac{m}{2}-1} b_i x^i = x^{\frac{m}{2}} B^H + B^L. \end{aligned} \tag{7.25}$$

Каждый из полученных полиномов $A^H(x), A^L(x), B^H(x), B^L(x)$ имеет степень в два раза меньшую нежели исходные множители $A(x)$ и $B(x)$.

Исходя из формулы (7.25), произведение $C = A \cdot B$ можно представить в виде формулы [28, 29]:

$$\begin{aligned} C(x) &= (x^{\frac{m}{2}} A^H + A^L)(x^{\frac{m}{2}} B^H + B^L) = \\ &= x^m A^H B^H + (A^H B^L + A^L B^H) x^{\frac{m}{2}} + A^L B^L = \\ &= x^m A^H B^H + A^L B^L + (A^H B^H + A^L B^L + \\ &+ (A^H + A^L)(B^H + B^L)) x^{\frac{m}{2}} = x^m C^H + C^L. \end{aligned} \tag{7.26}$$

Необходимо сразу отметить, что полином-произведение $C(x)$, полученный в результате вычисления по алгоритму Карацубы–Офмана, имеет длину до $2m-1$ двоичных элементов. Следовательно, чтобы определить элемент поля $C \in \mathrm{GF}(2^m)$, равный $A \cdot B$, необходимо привести полином $C(x)$ по модулю образующего многочлена поля $p(x)$ [28, 29].

Далее введем две переменные, как показано в формуле [28, 29]:

$$\begin{aligned} M_A &= A^H + A^L, \\ M_B &= B^H + B^L. \end{aligned} \tag{7.27}$$

В соответствии с формулой (7.26), для вычисления произведения C необходимо провести четыре операции сложения

$$\begin{aligned} &M_A = A^H + A^L, \\ &M_B = B^H + B^L, \\ &R = A^H B^H + A^L B^L, \\ &R + M_A M_B \end{aligned}$$

и три операции умножения

$$\begin{aligned} &A^H B^H, \\ &A^L B^L, \\ &M_A M_B, \end{aligned}$$

каждую из которых также можно разбить вышеописанным способом [28, 29].

Процедура умножения повторяется рекурсивно пока не будут получены однобитные множители, произведение которых вычисляется за одну операцию умножения (конъюнкция). Количество итераций не превышает $\lceil \log_2(m) \rceil$. На практике для умножения элементов поля большой степени часто применяют схему, в которой алгоритм Карацубы–Офмана используется для уменьшения длины операндов до значения, при котором удобно применить какой-либо из быстрых алгоритмов умножения, который был бы слишком сложен для реализации при применении его к операндам большой длины [28].

Алгоритм $2^k n$-битного умножителя Карацубы–Офмана представлен в алг. 7.3 [28, 29].

Алгоритм 7.3. Алгоритм $2^k n$-битного умножителя Карацубы–Офмана

Data: Элементы поля $A, B \in \mathrm{GF}(2^m)$, где $m = rn = 2^k n$, и множители A и B могут быть представлены в виде $A = x^{\frac{m}{2}} A^H + A^L$ и $B = x^{\frac{m}{2}} B^H + B^L$ соответственно.

Procedure $Kmul2^k(C, A, B)$
- **if** $r = 1$ **then**
  - $C = A \cdot B$;
  - return;
- **end**
- **for** $i = 0 \ldots (\frac{r}{2} - 1)$ **do**
  - $M_{Ai} = A_i^L + A_i^H$;
  - $M_{Bi} = B_i^L + B_i^H$;
- **end**
- $Kmul2^k(C^L, A^L, B^L)$;
- $Kmul2^k(M, M_A, M_B)$;
- $Kmul2^k(C^H, A^H, B^H)$;
- **for** $i = 0 \ldots (r - 1)$ **do**
  - $M_i = M_i + C_i^L + C_i^H$;
- **end**
- **for** $i = 0 \ldots (r - 1)$ **do**
  - $C_{\frac{r}{2}+i} = C_{\frac{r}{2}+i} + M_i$;
- **end**

end

Result: Полином $C(x) = A(x) \cdot B(x)$ длиной до $2m - 1$ элементов, где $C(x) = x^m C^H + C^L$.

Вторым вариантом использования алгоритма Карацубы–Офмана является так называемый двоичный умножитель Карацубы, предназначенный для работы с полями Галуа $\mathrm{GF}(2^m)$ произвольной степени m [28, 29].

Алгоритм двоичного умножителя Карацубы представлен в алг. 7.4. Необходимо заметить, что двоичный умножитель Карацубы использует внутри себя $2^k n$-битный умножитель (алг. 7.3) [28, 29].

Алгоритм 7.4. Алгоритм двоичного умножителя Карацубы

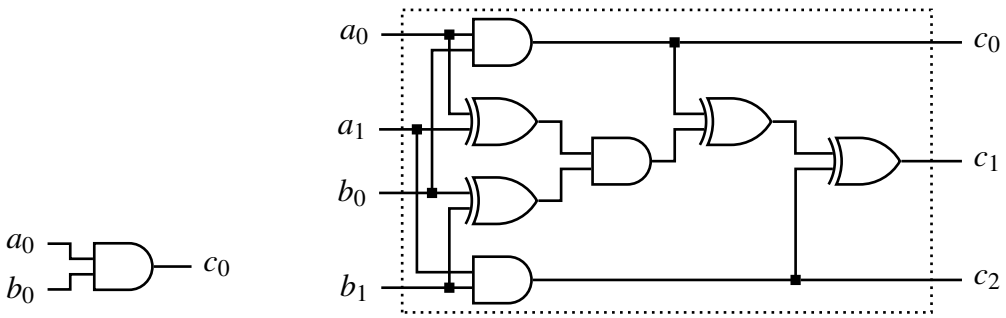
Data: Элементы поля $A, B \in \mathrm{GF}(2^m)$, где m — любое целое, и множители A и B могут быть представлены в виде $A(x) = x^{\frac{m}{2}} A^H + A^L$ и $B(x) = x^{\frac{m}{2}} B^H + B^L$ соответственно.

Procedure $BK(C, A, B)$
- $k = \lfloor \log_2(m) \rfloor$;
- $d = m - 2^k$;
- **if** $d = 0$ **then**
  - $Kmul2^k(C, A, B)$;
  - return;
- **end**
- **for** $i = 0 \ldots (d-1)$ **do**
  - $M_{Ai} = A_i^L + A_i^H$;
  - $M_{Bi} = B_i^L + B_i^H$;
- **end**
- $Kmul2^k(C^L, A^L, B^L)$;
- $Kmul2^k(M, M_A, M_B)$;
- $BK(C^H, A^H, B^H)$;
- **for** $i = 0 \ldots (2k-2)$ **do**
  - $M_i = M_i + C_i^L + C_i^H$;
- **end**
- **for** $i = 0 \ldots (2k-2)$ **do**
  - $C_{k+i} = C_{k+i} + M_i$;
- **end**

end

Result: Полином $C(x) = A(x) \cdot B(x)$ длиной до $2m-1$ элементов, где $C(x) = x^m C^H + C^L$.

Аппаратная реализация умножителя Карацубы, как можно видеть из формул и алгоритмов, требует использования блоков умножения (операция «И») и сложения по модулю два («Исключающее ИЛИ»). На рис. 7.6 приведены схемы одноразрядного и двухразрядного умножителей Карацубы. Можно видеть, что однобитный умножитель представляет из себя обычный блок «И» с двумя входами [30].



(а) *(б)*

Рис. 7.6. Схемы умножителей Карацубы:
(а) одноразрядный умножитель; (б) двухразрядный умножитель

На рис. 7.7 приведена схема четырехразрядного умножителя Карацубы, в состав которого входят три двухразрядных умножителя, которые на схеме обозначены как KM2 [30].

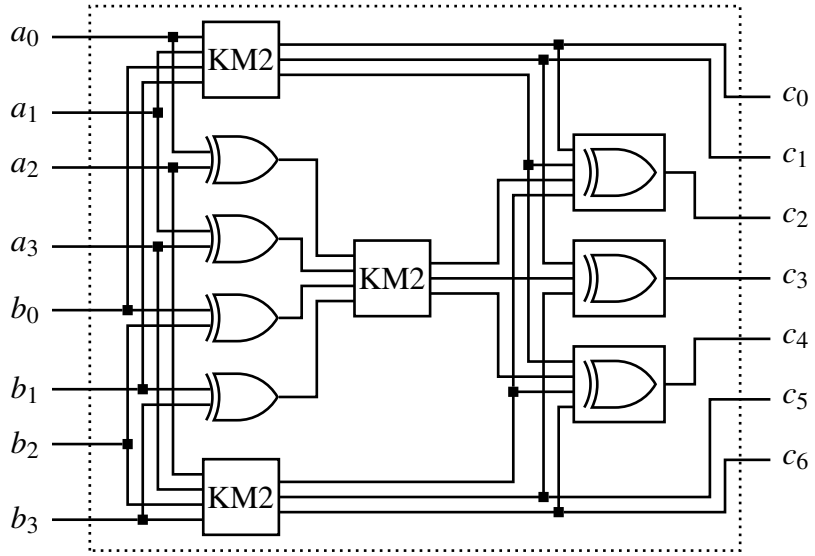


Рис. 7.7. Схема четырехразрядного умножителя Карацубы

7.3.4. Умножитель двух элементов поля по схеме Рейхани-Мазолеха

Данная реализация умножителя была предложена в начале 2000-х членами IEEE А. Рейхани-Мазолехом и М. А. Хасаном. Являясь модификацией умножителя Мастровито, эта схема позволяет построить эффективный умножитель с параллельной структурой для двоичных полей Галуа, построенных с использованием порождающего многочлена определённого вида [31].

Для рассмотрения умножителя вводится ряд обозначений.

Элементы поля $\mathrm{GF}(2^m)$, являющиеся множителями, обозначаются как A и B. Множитель A можно представить как сумму

$$A = \sum_{i=0}^{m-1} a_i \varepsilon^i,\ a_i \in \{0,1\},$$

где ε^i — элементы левого степенного базиса поля $\mathrm{GF}(2^m)$.

Элементы a_i можно представить в виде вектора $\mathbf{a} = [a_0, a_1, \ldots, a_{m-1}]^T$. В таком случае, множитель A можно представить как произведение векторов $A = \boldsymbol{\varepsilon}^T \mathbf{a}$, где $\boldsymbol{\varepsilon} = [1, \varepsilon, \ldots, \varepsilon^{m-1}]^T$. Аналогично можно представить и множитель B.

$$B = \sum_{j=0}^{m-1} b_j \varepsilon^j = \boldsymbol{\varepsilon}^T \mathbf{b},$$

где $\mathbf{b} = [b_0, b_1, \ldots, b_{m-1}]^T$.

Вводится понятие *приведенной матрицы* $\mathbf{Q}$ размера $m-1 \times m$, которая является двоичной матрицей, получаемой из тождества

$$\boldsymbol{\varepsilon}^{\uparrow} \equiv \mathbf{Q}\boldsymbol{\varepsilon}(\mathrm{mod}(p(\varepsilon))), \tag{7.28}$$

где $\boldsymbol{\varepsilon}^{\uparrow} = [\varepsilon^m, \varepsilon^{m-1}, \ldots, \varepsilon^{2m-2}]^T$.

Каждому неприводимому полиному $p(x)$ соответствует одна и только одна приведенная матрица $\mathbf{Q}$ [31].

Также вводятся два вектора $\mathbf{d}$ и $\mathbf{e}$, являющиеся функциями от A и B, соответственно [31]:

$$\mathbf{d} = \mathbf{L}\mathbf{b} \quad \text{и} \quad \mathbf{e} = \mathbf{U}\mathbf{b}, \tag{7.29}$$

где $\mathbf{L}$ — нижнетреугольная матрица Теплица размера $m \times m$, а $\mathbf{U}$ — верхнетреугольная матрица Теплица размера $(m-1) \times m$, которые представлены в формуле:

$$\begin{aligned}
\mathbf{L} &\triangleq \begin{bmatrix}
a_0 & 0 & 0 & 0 & \cdots & 0 \\
a_1 & a_0 & 0 & 0 & \cdots & 0 \\
a_2 & a_1 & a_0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
a_{m-2} & a_{m-3} & \cdots & a_1 & a_0 & 0 \\
a_{m-1} & a_{m-2} & \cdots & a_2 & a_1 & a_0
\end{bmatrix}, \\
\mathbf{U} &\triangleq \begin{bmatrix}
0 & a_{m-1} & a_{m-2} & \cdots & a_2 & a_1 \\
0 & 0 & a_{m-1} & \cdots & a_3 & a_2 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & a_{m-1} & a_{m-2} \\
0 & 0 & \cdots & 0 & 0 & a_{m-1}
\end{bmatrix}.
\end{aligned} \tag{7.30}$$

Результат C произведения элементов поля A и B получается из следующей формулы:

$$\mathbf{c} = \mathbf{d} + \mathbf{Q}^T\mathbf{e}, \tag{7.31}$$

где $\mathbf{c} = [c_0, c_1, \ldots, c_{m-1}]^T$.

Используя формулы (7.29), (7.30) и (7.31), можно построить эффективный умножитель двух элементов двоичного поля Галуа $\mathrm{GF}(2^m)$, общая архитектура которого представлена на рис. 7.8 [31].

Можно видеть, что умножитель состоит из двух крупных блоков: IP-сети[1] и Q-сети. В IP-сети производится расчет элементов векторов $\mathbf{d}$ и $\mathbf{e}$ согласно формуле (7.29). IP-сеть состоит из блоков циклического сдвига (рис. 7.9(а)) и m блоков I_i, каждый из которых, кроме последнего блока I_{m-1}, состоит из двух ячеек IP. Блок I_{m-1} состоит из одной ячейки $IP(m)$. Структура ячеек IP показана на рис. 7.9(б). В Q-сети производится вычисление по

[1] IP-сеть — от англ. Inner Product — внутреннее (скалярное) произведение

формуле (7.31). Q-сеть состоит из m блоков BTX[2], структура которых показана на рис. 7.9(б) [31].

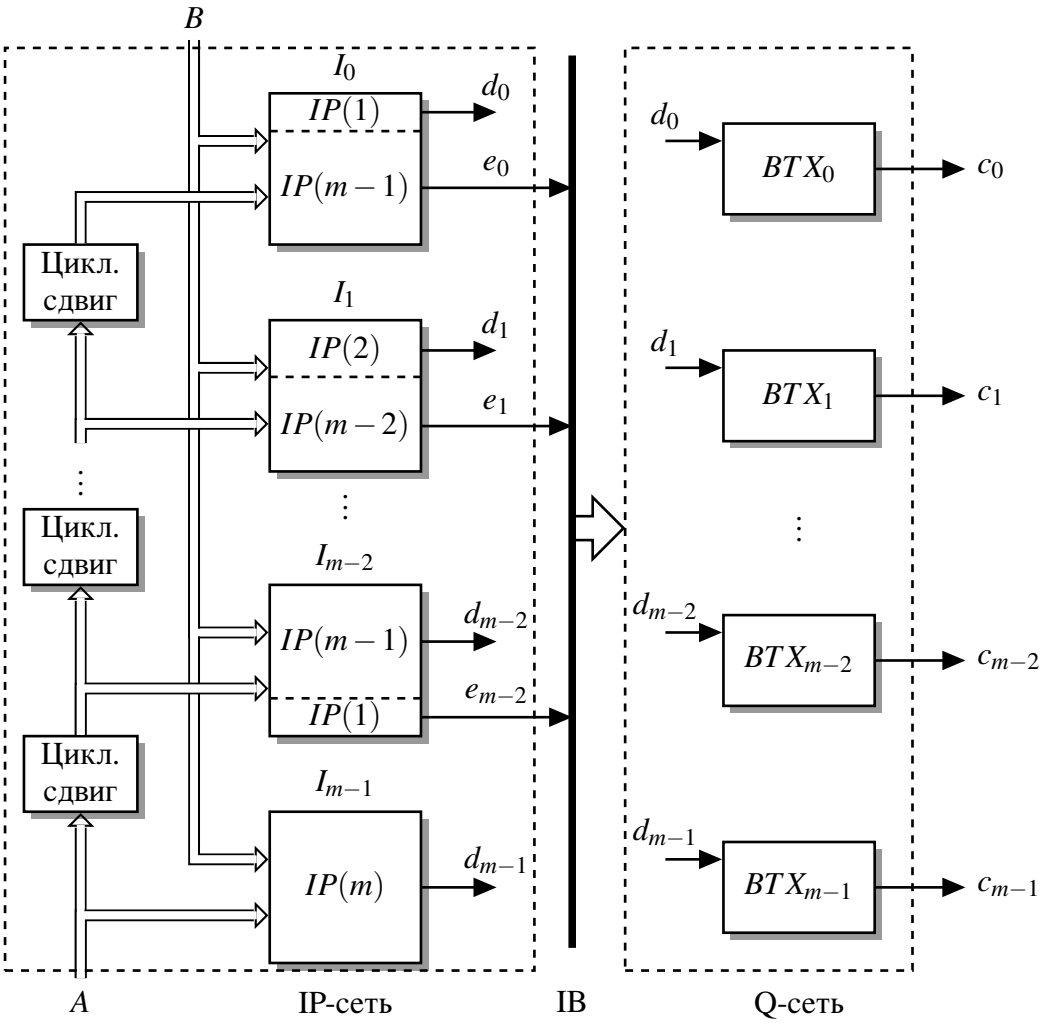


Рис. 7.8. Архитектура умножителя двух элементов двоичного поля Галуа $\mathrm{GF}(2^m)$ *по схеме Рейхани-Мазолеха*

Для некоторых типов образующих поле Галуа полиномов $p(x)$ существуют определенные формы матрицы $\mathbf{Q}$, которые позволяют построить ее, не прибегая к сложным расчетам [31].

В качестве примера можно привести *равномерно распределенные полиномы*[3], под которыми понимаются неприводимые полиномы вида

$$p(x) = x^{ns} + x^{(n-1)s} + \ldots + x^s + 1, \tag{7.32}$$

образующие поле Галуа $\mathrm{GF}(2^m)$ с $m = ns$ [31].

[2]BTX — от англ. Binary Tree of XOR's — двоичное дерево элементов XOR

[3]В англ. литературе используется термин Equally Spaced Polynomial (s-ESP)

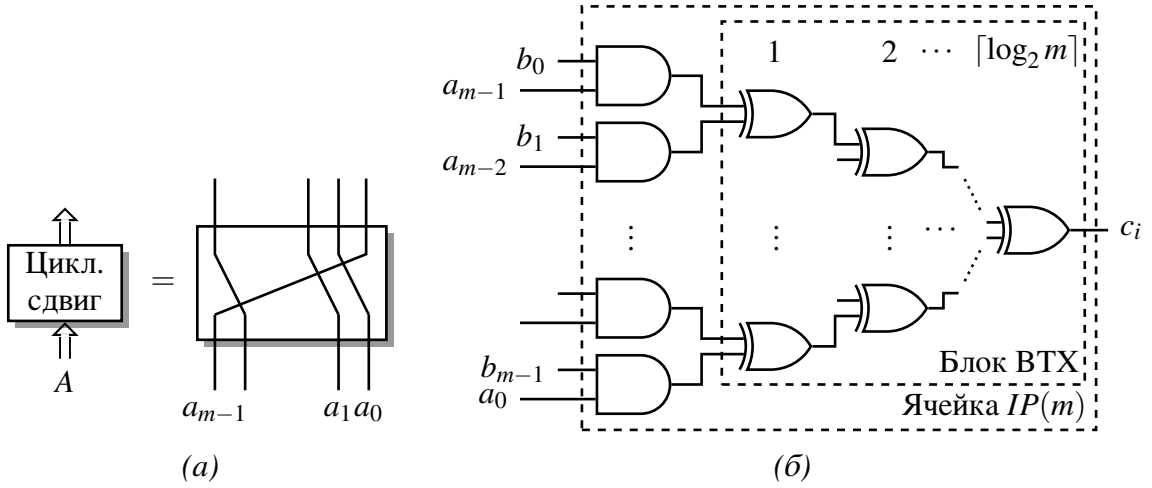


Рис. 7.9. Элементы умножителя по схеме Рейхани-Мазолеха: (а) блок циклического сдвига; (б) пример ячейки $IP(m)$ и блока ВТХ

На рис. 7.10 в графическом виде показаны матрицы $\mathbf{Q}$ для трех типов равномерно распределенных полиномов. Черными квадратами показаны позиции ненулевых элементов в матрице $\mathbf{Q}$ [31].

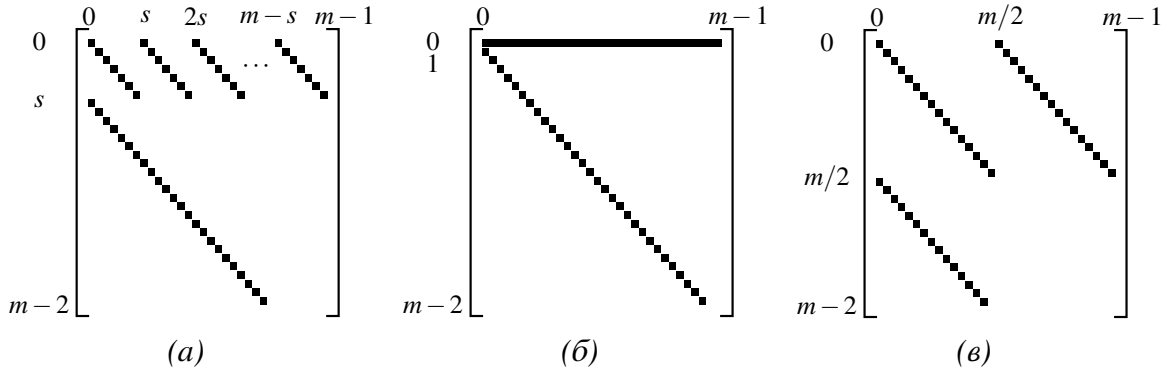


Рис. 7.10. Графическое представление матрицы $\mathbf{Q}$ для трех типов равномерно распределенных полиномов $p(x) = x^{ns} + x^{(n-1)s} + \ldots + x^s + 1$, $m = ns$: (а) $1 < s < \frac{m}{2}$; (б) $s = 1$ (все коэфф. равны 1); (в) $s = \frac{m}{2}$ (трином)

Другими образующими полиномами, имеющими характерную форму матрицы $\mathbf{Q}$, являются полиномы из трех элементов — триномы:

$$p(x) = x^m + x^k + 1, \qquad k < m, \tag{7.33}$$

включающие в себя и равномерно распределенный триномс $k = \frac{m}{2}$, показанный на рис. 7.10(в). Прочие типы триномов представлены на рис. 7.11 [31].

Еще более сложную форму матрицы $\mathbf{Q}$ имеют образующие полиномы, состоящие из пяти элементов — пентаномы:

$$p(x) = x^m + x^{k_3} + x^{k_2} + x^{k_1} + 1, \qquad 1 \le k_1 < k_2 < k_3 \le m-1. \tag{7.34}$$

Матрицы $\mathbf{Q}$ для некоторых типов пентаномов можно найти в [31].

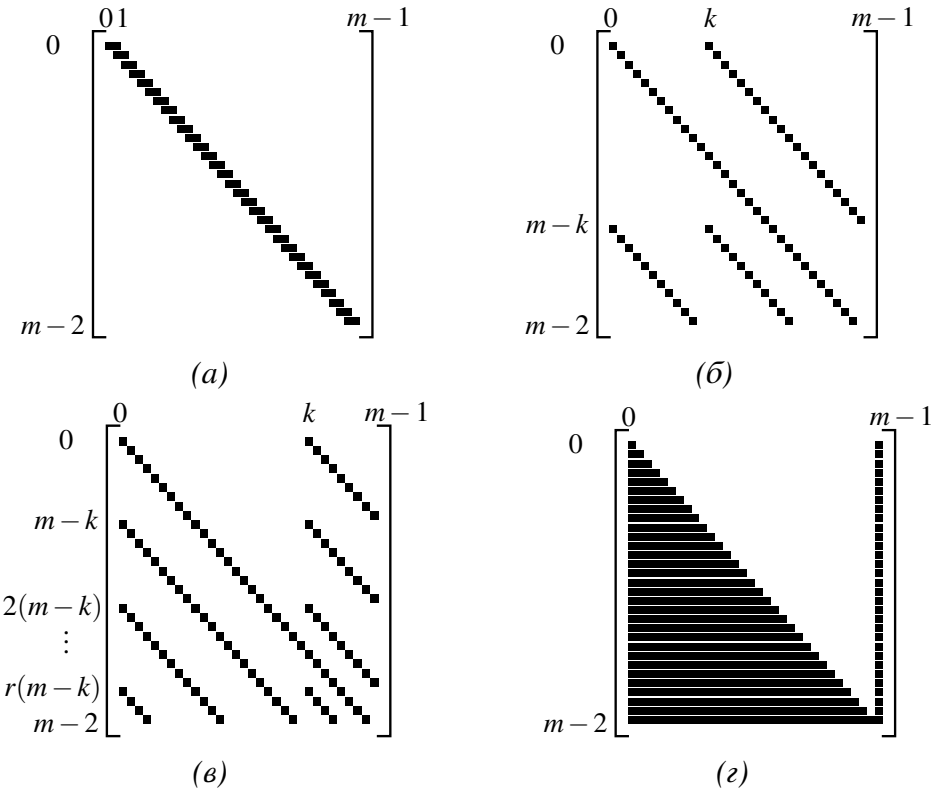


Рис. 7.11. Графическое представление матрицы $\mathbf{Q}$ *для различных типов триномов* $p(x) = x^m + x^k + 1$*:*
(а) $k = 1$*; (б)* $1 < k < \frac{m}{2}$*; (в)* $\frac{m}{2} < k < m-1$*,* $r = \lfloor \frac{m-2}{m-k} \rfloor$*; (г)* $k = m-1$

В качестве примера рассмотрим построение умножителя для поля Галуа $\mathrm{GF}(2^4)$, образованного полиномом

$$p(x) = x^4 + x + 1.$$

Этот полином является триномом (7.33) с $k = 1$. Следовательно, его матрица $\mathbf{Q}$ строится по схеме, представленной на рис. 7.11(а), и имеет вид

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

Множители A и B представляются векторами

$$\mathbf{a} = [a_0, a_1, a_2, a_3]^T, \qquad \mathbf{b} = [b_0, b_1, b_2, b_3]^T.$$

Матрицы $\mathbf{L}$ и $\mathbf{U}$ согласно формуле (7.30) равны

$$\mathbf{L} = \begin{bmatrix} a_0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 \\ a_2 & a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}, \qquad \mathbf{U} = \begin{bmatrix} 0 & a_3 & a_2 & a_1 \\ 0 & 0 & a_3 & a_2 \\ 0 & 0 & 0 & a_3 \end{bmatrix}.$$

Таким образом, промежуточные векторы $\mathbf{d}$ и $\mathbf{e}$ согласно формуле (7.29) равны

$$\mathbf{d} = \mathbf{L}\mathbf{b} = \begin{bmatrix} a_0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 \\ a_2 & a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_1 b_0 + a_0 b_1 \\ a_2 b_0 + a_1 b_1 + a_0 b_2 \\ a_3 b_0 + a_2 b_1 + a_1 b_2 + a_0 b_3 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix},$$

$$\mathbf{e} = \mathbf{U}\mathbf{b} = \begin{bmatrix} 0 & a_3 & a_2 & a_1 \\ 0 & 0 & a_3 & a_2 \\ 0 & 0 & 0 & a_3 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_3 b_1 + a_2 b_2 + a_1 b_3 \\ a_3 b_2 + a_2 b_3 \\ a_3 b_3 \end{bmatrix} = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \end{bmatrix}.$$

Подставляя полученную матрицу $\mathbf{Q}$ и векторы $\mathbf{d}$ и $\mathbf{e}$ в формулу (7.31), получим результат $\mathbf{c}$:

$$\mathbf{c} = \mathbf{d} + \mathbf{Q}^T \mathbf{e} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_0 \\ e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} d_0 + e_0 \\ d_1 + e_0 + e_1 \\ d_2 + e_1 + e_2 \\ d_3 + e_2 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

Схема умножителя для рассмотренного поля показана на рис. 7.12.

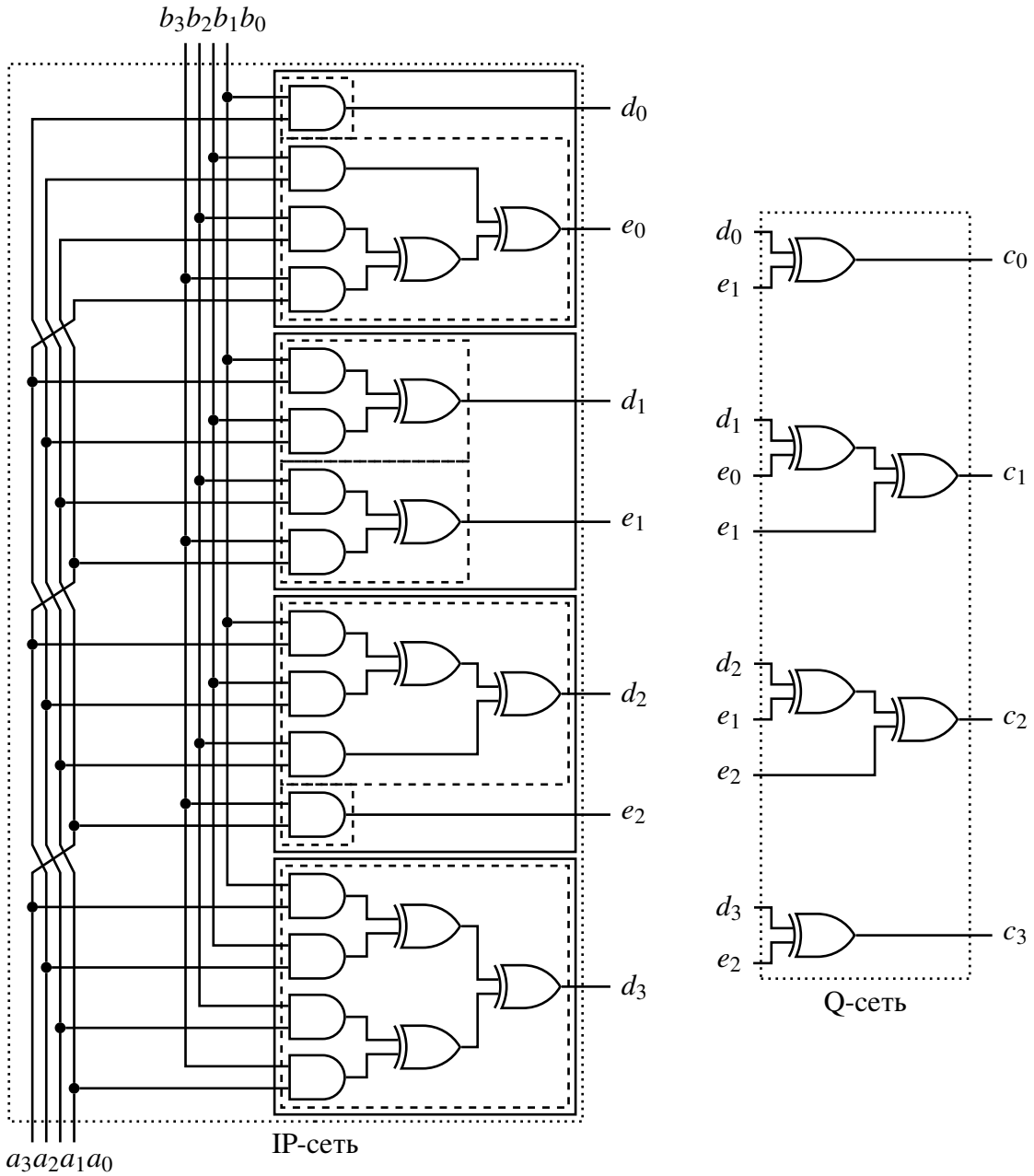


Рис. 7.12. Умножитель двух элементов двоичного поля Галуа $\mathrm{GF}(2^m)$ $p(x) = x^4 + x + 1$ *по схеме Рейхани-Мазолеха*

Контрольные вопросы

1. Что такое левый степенной базис поля Галуа?
2. Как строится поле Галуа?
3. Дайте понятие двойственного базиса поля.
4. Как производится логарифмирование и антилогарифмирование в поле Галуа?
5. Опишите принцип работы алгоритма умножения Карацубы–Офмана.

8. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

Теория графов является разделом дискретной математики, изучающим свойства графов. Эта область науки крайне обширна и столь же интересна, сколь и сложна. Этот математический инструмент находит применение во многих отраслях науки и промышленности. В качестве примеров использования теории графов можно привести поиск кратчайшего маршрута на карте, построение блок-схем алгоритмов, схемы размещения элементов на печатных платах, принципиальные и структурные схемы устройств (например рис. 2.4 и 7.1) и многое другое. В данном пособии даны только основные понятия теории и те элементы, которые могут пригодиться при рассмотрении теории помехоустойчивого кодирования.

Основоположником теории графов является Леонард Эйлер (1707–1782), который решил в 1736 г. известную в то время задачу о семи кенигсбергских мостах. Впоследствии эта задача стала одной из классических задач теории графов. Впоследствии методы теории графов применялись и другими известными учеными в самых различных областях науки: в 1847 г. Кирхгоф использовал теорию деревьев для расчета силы тока в электрической цепи; в 1857 г. Кэли использовал деревья в решении задач органической химии. И таких примеров можно привести ещё много [32].

8.1. Основные понятия

В общем смысле *граф* G задается множеством точек или *вершин* (*узлов*) $x_1, x_2, \ldots, x_n$ (обозначается символом X) и множеством линий или *ребер* $a_1, a_2, \ldots, a_m$ (обозначается A), которые соединяют между собой все или часть этих вершин. То есть, граф полностью задается и обозначается парой (X, A) [33, 34]. Существует и другое обозначение. Граф, содержащий n вершин и m ребер, называется (n, m)-*графом*, а $(1, 0)$-граф называется *тривиальным* [32]. Пример графа показан на рис. 8.1.

Две вершины x_i и x_j, соединенные ребром a_k, называются *смежными*. Их иногда обозначают как x_i adj x_j. При этом говорят, что вершина x_i и ребро a_k *инцидентны*, как и x_j и a_k. Два различных ребра, инцидентных одной и той же вершине, также называются *смежными* [32]. Так, на рис. 8.1 вершины x_1 и x_2 смежные, также смежными являются ребра a_1 и a_3, которые инцидентны вершине x_2.

Если в графе $G = (X, A)$ ребра из множества A ориентированы (обычно показывается стрелкой), то такой граф называется *ориентированным графом* или *орграфом*, а сами ребра называются *дугами* [32, 34]. Граф, ребра которого не имеют ориентации, называется *неориентированным*, а содержащий и ориентированные, и неориентированные ребра — *смешанным*. Неориентированный граф, соответствующий орграфу $G = (X, A)$, обозначается как

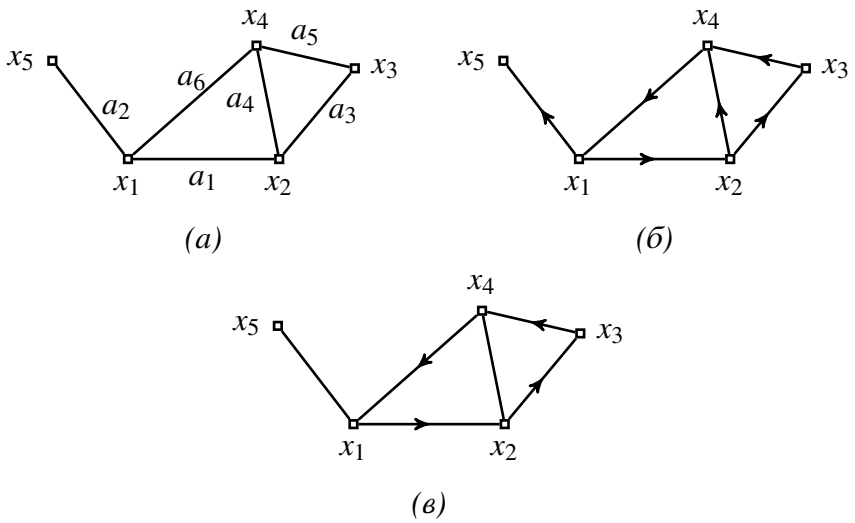


Рис. 8.1. Примеры графа: (а) неориентированный граф; (б) ориентированный граф; (в) смешанный граф.

$\bar{G} = (X, \bar{A})$ и называется *неориентированным дубликатом* или *неориентированным двойником* графа G [34]. Примеры всех трех видов графа показаны на рис. 8.1. При этом, граф на рис. 8.1(а) является неориентированным дубликатом графа на рис. 8.1(б).

Дуги орграфа удобно обозначать парами вершин вида (x_i, x_j), указывая от какой вершины к какой направлена дуга [32, 34]. Например, на рис. 8.1(б) можно выделить дугу (x_1, x_2), соответствующую ребру a_1 неориентированного графа на рис. 8.1(а).

Отдельно выделяют два вида графов. *Мультиграф* (см. рис. 8.2(а)), в котором нет петель, но две вершины могут быть соединены более чем одним ребром — такие ребра называются *кратными*. *Петлей* называется ребро (дуга) начальная и конечная вершины которой совпадают. Граф, в котором допускаются и кратные ребра и петли, называется *псевдографом* (рис. 8.2(б)) [32, 34].

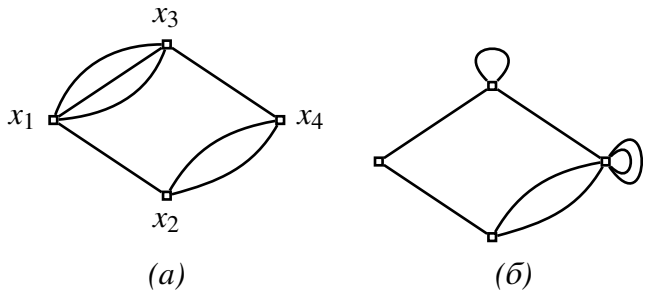
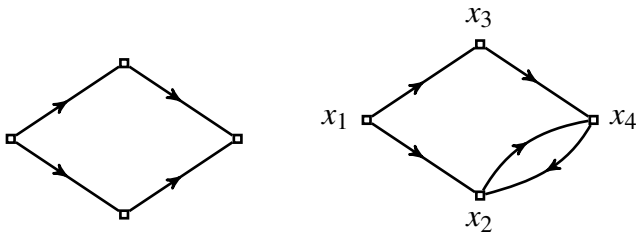


Рис. 8.2. Мультиграф (а) и псевдограф (б)

Орграф, не имеющий симметричных пар дуг, называется *направленным графом*. При этом *симметричными* называют дуги, соединяющие одну и ту же пару вершин, но ориентированные в противоположных направлениях, на-

пример (x_i, x_j) и (x_j, x_i) [32]. На рис. 8.3(а) показан пример направленного графа. Дуги (x_2, x_4) и (x_4, x_2) ненаправленного графа на рис. 8.3(б) являются симметричными.

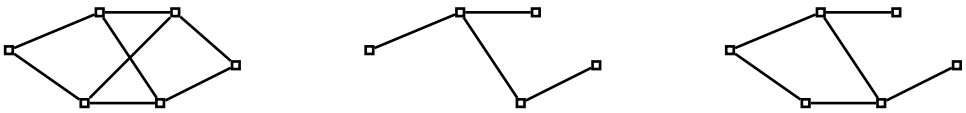


(а) *(б)*

Рис. 8.3. Направленный (а) и ненаправленный (б) графы

Если вершины графа имеют какие-либо метки, отличающие их друг от друга, то такой граф называется *помеченным* или *перенумерованным* [32]. Например, графы на рис. 8.2(а) и рис. 8.3(б) являются помеченными, а графы на рис. 8.2(б) и 8.3(а) — нет.

Граф G_1, все вершины и ребра которого принадлежат графу G, называется *подграфом* графа G, а G, в свою очередь, является *надграфом* для G_1. Подграф графа G, содержащий все его вершины, называется *остовным подграфом* или *частичным графом* [35, 32]. Графы на рис. 8.4(б) и 8.4(в) являются подграфами графа 8.4(а), который является для них надграфом. При этом, граф 8.4(в) является остовным подграфом графа 8.4(а).



(а) *(б)* *(в)*

Рис. 8.4. Граф (а) и его подграфы (б) и (в)

Маршрутом в графе G называют чередующуюся последовательность вершин и ребер $x_1, a_1, x_2, a_2, \ldots, x_{n-1}, a_{n-1}, x_n$, которая начинается и заканчивается вершиной, а каждое ребро инцидентно двум вершинам — непосредственно предшествующей и непосредственно следующей за ним [32]. Фактически, достаточно говорить просто про последовательность вершин или про последовательность ребер — одно подразумевает другое [32, 34]. Такой маршрут, соединяющий вершины x_1 и x_n, иногда называют *$(x_1$–$x_n)$-маршрутом* [32]. Если $x_1 = x_n$, то маршрут называется *замкнутым*, в ином случае — *открытым*. Маршрут, все ребра которого различны, называется *цепью*. Если при этом различны все вершины, то маршрут называется *простой цепью*. Замкнутая цепь называется *циклом*, а если все n вершин замкнутого маршрута различны и $n \geq 3$, то он называется *простым циклом* [32].

В случае, если речь идет об ориентированных графах, используют понятие *ориентированный маршрут* или *путь*, под которым понимают последовательность дуг (и, соответственно, вершин), в которой конечная вершина всякой дуги, отличной от последней, является начальной вершиной следующей. Аналогично понятиям цепи и простой цепи выделяют *ориентированную цепь* (*орцепь*) и *простую орцепь* [34].

Длина (*мощность*) *маршрута* определяется количеством ребер в нем. При этом, каждое ребро считается столько раз, сколько оно встречается в этом маршруте. *Расстоянием* $d(x_i, x_j)$ между вершинами x_i и x_j графа называется длина кратчайшей простой цепи, соединяющей их. Для несоединенных x_i и x_j полагают $d(x_i, x_j) = \infty$ [32, 34].

Иногда дугам (x_i, x_j) графа G ставится в соответствие некоторое число c_{ij}, называемое *весом* (*длиной, стоимостью*) дуги. Такой граф G называется *графом со взвешенными дугами*. Если же некоторые веса v_i приписаны вершинам x_i, то такой граф называется *графом со взвешенными вершинами* или просто *взвешенным*. Иногда *взвешенным* называют граф, дуги и вершины которого имеют соответствующие им веса [34].

В том случае, если рассматривается некоторый путь μ, представленный последовательностью дуг, то за его *вес* принимается число $l(\mu)$, равное сумме весов всех дуг, входящих в μ [34]:

$$l(\mu) = \sum_{(x_i, x_j) \in \mu} c_{ij}.$$

8.2. Матричное представление графа

Для алгебраического задания граф удобно представлять в виде матриц [34].

Для примера возьмем граф G, представленный на рис. 8.5.

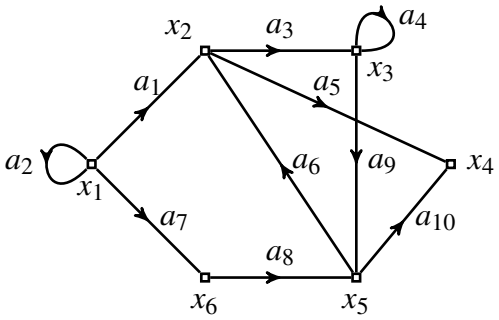


Рис. 8.5. Пример графа для матричного представления

8.2.1. Матрица смежности

Матрица смежности графа G полностью определяет структуру графа [34]. Она обозначается как $\mathbf{A} = [a_{ij}]$, где

$$a_{ij} = 1, \text{ если в } G \text{ существует дуга } (x_i, x_j),$$
$$a_{ij} = 0, \text{ если в } G \text{ отсутствует дуга } (x_i, x_j).$$

Таким образом, для графа G на рис. 8.5 матрица смежности будет равна

$$\mathbf{A} = \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1 & 1 & 0 & 0 & 0 & 1 \\ x_2 & 0 & 0 & 1 & 1 & 0 & 0 \\ x_3 & 0 & 0 & 1 & 0 & 1 & 0 \\ x_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 1 & 0 & 1 & 0 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}.$$

8.2.2. Матрица инциденций

Матрица инциденций графа G с n вершинами и m дугами, обозначается как $\mathbf{B} = [b_{ij}]$ [34]. Она имеет размерность $n \times m$ и определяется как

$$b_{ij} = 1, \quad \text{если } x_i \text{ — начальная вершина дуги } a_j,$$
$$b_{ij} = -1, \quad \text{если } x_i \text{ — конечная вершина дуги } a_j,$$
$$b_{ij} = 0, \quad \text{если } x_i \text{ не инцидентна } a_j \text{ или } a_j \text{ — петля.}$$

Таким образом, для графа G на рис. 8.5 матрица инциденций будет равна

$$\mathbf{B} = \begin{array}{c|cccccccccc} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \\ \hline x_1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_2 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ x_4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ x_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 \\ x_6 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}.$$

Так как каждая дуга инцидентна двум различным вершинам, кроме случая, когда она образует петлю, каждый столбец матрицы инциденций либо содержит и «1» и «-1», либо все его элементы равны «0» [34].

Для неориентированного графа матрица инциденций определяется аналогично, за исключением того, что «-1» заменяется на «1» [34].

8.2.3. *Матрицы достижимостей и контрадостижимостей*

Матрица достижимостей $\mathbf{R} = [r_{ij}]$ определяется как

$$\begin{aligned} r_{ij} &= 1, \quad \text{если вершина } x_j \text{ достижима из } x_i, \\ r_{ij} &= 0, \quad \text{в противном случае.} \end{aligned}$$

Множество вершин $R(x_i)$ графа G, достижимых из заданной вершины x_i, состоит из таких x_j, для которых элемент r_{ij} в матрице $\mathbf{R}$ равен «1». Все диагональные элементы $\mathbf{R}$ равны «1», так как каждая вершина достижима из себя самой с помощью пути длины 0 [34].

Матрица достижимостей $\mathbf{R}$ для графа G на рис. 8.5 равна

$$\mathbf{R} = \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_2 & 0 & 1 & 1 & 1 & 1 & 0 \\ x_3 & 0 & 1 & 1 & 1 & 1 & 0 \\ x_4 & 0 & 0 & 0 & 1 & 0 & 0 \\ x_5 & 0 & 1 & 1 & 1 & 1 & 0 \\ x_6 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}.$$

Матрица контрадостижимостей (или *матрица обратных достижимостей*) $\mathbf{Q} = [q_{ij}]$ определяется как

$$\begin{aligned} r_{ij} &= 1, \quad \text{если вершина } x_i \text{ достижима из } x_j, \\ r_{ij} &= 0, \quad \text{в противном случае.} \end{aligned}$$

Таким образом, контрадостижимое множество $Q(x_i)$ графа G — это множество таких вершин, что из любой вершины этого множества можно достигнуть вершины x_i [34].

Из определений матриц $\mathbf{Q}$ и $\mathbf{R}$ следует, что матрица контрадостижимостей $\mathbf{Q}$ равна транспонированной матрице достижимостей $\mathbf{R}^T$ [34]:

$$\mathbf{Q} = \mathbf{R}^T.$$

Матрица контрадостижимостей $\mathbf{Q}$ для графа G на рис. 8.5 равна

$$\mathbf{Q} = \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 1 & 1 & 0 & 1 & 1 \\ x_3 & 1 & 1 & 1 & 0 & 1 & 1 \\ x_4 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_5 & 1 & 1 & 1 & 0 & 1 & 1 \\ x_6 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}.$$

8.3. Линейные графы сигналов и передача графа

Граф сигналов — это графическое представление соотношений между несколькими переменными. В случае линейности этих соотношений, граф выражает систему линейных алгебраических уравнений и называется *линейным графом сигналов*. Такой способ представления позволяет наглядно выразить систему уравнений и решить ее непосредственно путем анализа графа [36].

Граф сигналов представляет из себя ориентированную цепь, в которой каждая дуга[4] $(x_j x_k)$ связана с числом t_{jk}, называемым *передачей дуги*, а каждому узлу x_j соответствует так называемый *узловой сигнал* X_j [36].

В дальнейшем, говоря в этом разделе о графах, будем иметь в виду именно линейные графы сигналов.

Узловые сигналы определяются уравнениями вида [36]:

$$X_k = \sum_j X_j t_{jk}, \quad k = 1, 2, 3, \ldots . \tag{8.1}$$

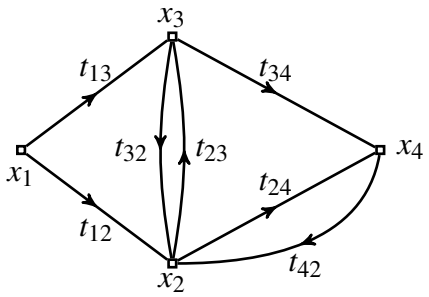


Рис. 8.6. Пример графа сигналов

Для примера рассмотрим граф сигналов на рис. 8.6. Согласно формуле (8.1) его узловые сигналы будут равны

$$\begin{aligned} X_2 &= X_1 t_{12} + X_3 t_{32} + X_4 t_{42}; \\ X_3 &= X_1 t_{13} + X_2 t_{23}; \\ X_4 &= X_2 t_{24} + X_3 t_{34}. \end{aligned}$$

Таким образом, можно сказать, что линейный граф сигналов представляет из себя систему линейных уравнений, представленную особым образом — вместо символов «+», «=» используются направленные дуги и узлы. Каждое уравнение в этой системе имеет форму «причина — следствие». При этом каждый зависимый узловой сигнал выражен один раз в виде явного *следствия*, вызванного другими узловыми сигналами, действующими в качестве *причин*. Совокупность этих свойств приводит к тому, что уравнения вида (8.1) могут быть решены непосредственно путем вычисления графа [36].

[4] В источниках (например, [36] и [37]) также используют термины «ребро» и «ветвь».

8.3.1. Эквивалентные преобразования графов

При решении графов сигналов используются эквивалентные преобразования, позволяющие упростить структуру графа и уменьшить сложность расчетов. На рис. 8.7 приведены элементарные эквивалентные схемы графов [36].

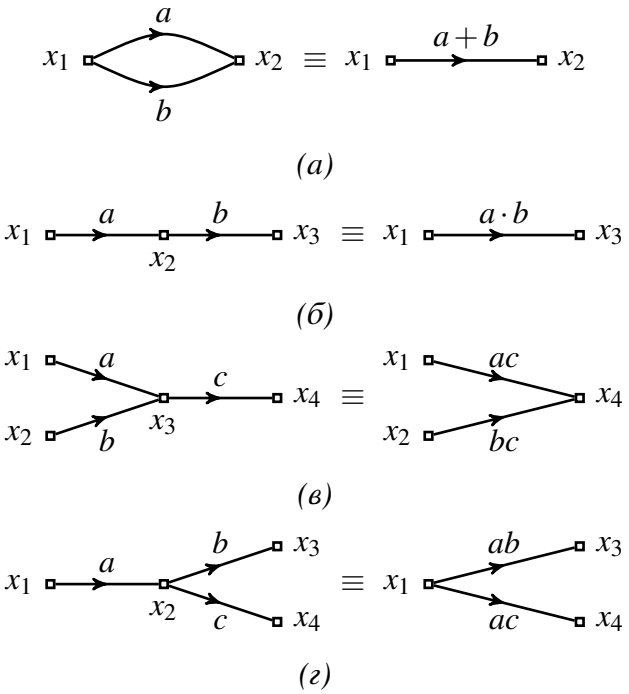


Рис. 8.7. Элементарные эквивалентные схемы графов:
(а) сложение; (б) умножение; (в) распределение (разложение) на множители справа;
(г) распределение (разложение) на множители слева

Преобразования на рис. 8.7 являются обратимыми [36].

На рис. 8.8 приведено исключение узла в звезде. Такое преобразование в общем случае не является обратным. То есть, если задан граф в виде квадрата (рис. 8.8(б)), то не следует ожидать, что эквивалентный ему граф будет иметь форму звезды (рис. 8.8(а)) [36].

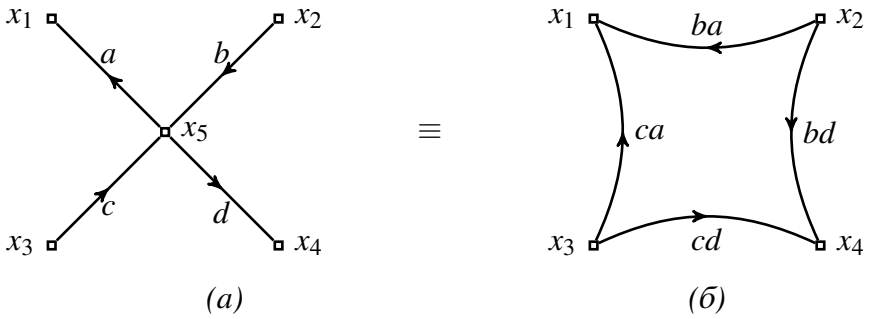
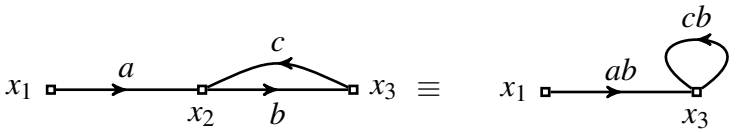


Рис. 8.8. Исключение узла в звезде:
(а) исходная звезда; (б) итоговый квадрат

Если преобразуемый граф содержит любые замкнутые цепи, то при его преобразовании появляются одна или больше петель, как показано на рис. 8.9 [36].



$X_2 b = (X_1 a + X_3 c)b = X_3$ $\qquad$ $X_1 ab + X_3 cb = X_3$

(а) $\qquad$ *(б)*

Рис. 8.9. Пример образования петли при разделении дуги: (а) исходный граф; (б) итоговый граф с петлей

Уравнение для узлового сигнала, приведенное на рис. 8.9(б) можно преобразовать к виду

$$X_1 ab = X_3(1 - cb) \quad \Rightarrow X_3 = \frac{X_1 ab}{1 - cb}. \tag{8.2}$$

Таким образом, граф на рис. 8.9(б) можно свести к одной дуге $(x_1 x_3)$ с передачей, равной $\frac{X_1 ab}{1-cb}$.

8.3.2. Передача графа

Передача T графа равна сигналу, возникающему в некотором зависимом узле, на единицу сигнала в некотором заданном узле-источнике. Если в графе содержится только один узел-сток x_k, т. е. узел, имеющий только входящие ветви, и один узел-источник x_j только с исходящими ветвями, то передача графа определяется однозначно по формуле [36]

$$T = \frac{X_k}{X_j}. \tag{8.3}$$

Отметим, что индексы у передачи графа в этом случае не указываются.

Если же в графе есть источник, но нет стока или стоков несколько, а также в случае произвольной структуры графа при отсутствии источников и стоков, необходимо указывать между какими узлами считается передача графа. В этом случае она обозначается как T_{jk} — передача графа между узлами x_j и x_k [36].

При определении передачи графа используются его топологические свойства. Важными топологическими параметрами графа являются его пути и контуры [36].

Путь — это непрерывная последовательность дуг (в указанном направлении), вдоль которой каждый узел встречается не более одного раза. Произведение передач дуг вдоль этого пути образует *передачу пути P* [36].

Контур (или контур обратной связи) — это простой замкнутый путь, вдоль которого каждый узел встречается не более одного раза за один обход контура. *Передача контура L* равна произведению передач ветвей в этом контуре [36].

Для примера рассмотрим граф на рис. 8.6. Он содержит четыре различных пути от узла x_1 до узла x_4: $P_1 = t_{13}t_{34}$, $P_2 = t_{13}t_{32}t_{24}$, $P_3 = t_{12}t_{23}t_{34}$ и $P_4 = t_{12}t_{24}$. Также в нем присутствует 3 контура: $L_1 = t_{23}t_{32}$, $L_2 = t_{24}t_{42}$ и $L_3 = t_{23}t_{34}t_{42}$.

Зная все пути между узлами x_j и x_k и все контуры в графе можно полностью выразить передачу графа T_{jk} [36].

Для расчета передачи любого графа T_{jk} через p путей и m контуров используется общее правило (8.4), которое часто называют формулой Мэзона–Циммермана [36, 37, 38].

$$T_{jk} = \frac{[(P_1 + P_2 + \ldots + P_p)(1 - L_1)(1 - L_2)\ldots(1 - L_m)]^*}{[(1 - L_1)(1 - L_2)\ldots(1 - L_m)]^*}. \tag{8.4}$$

Звездочка означает, что при умножении коэффициентов внутри скобок приравнивается к нулю любой член, который содержит произведение передач двух контуров или произведение передач пути и контура, которые касаются друг друга в графе [36].

Формулу (8.4) можно переписать в виде

$$T_{jk} = \frac{\left[\sum\limits_i P_i \prod\limits_v (1 - L_v)\right]^*}{\left[\prod\limits_{w=1}^{m} (1 - L_w)\right]^*} = \frac{\left[\sum\limits_i P_i \prod\limits_v (1 - L_v)\right]^*}{1 - \sum\limits_{w=1}^{m} L_w + \left[\sum\limits_{s,t} L_s L_t\right]^* - \left[\sum\limits_{s,t,u} L_s L_t L_u\right]^* + \ldots}, \tag{8.5}$$

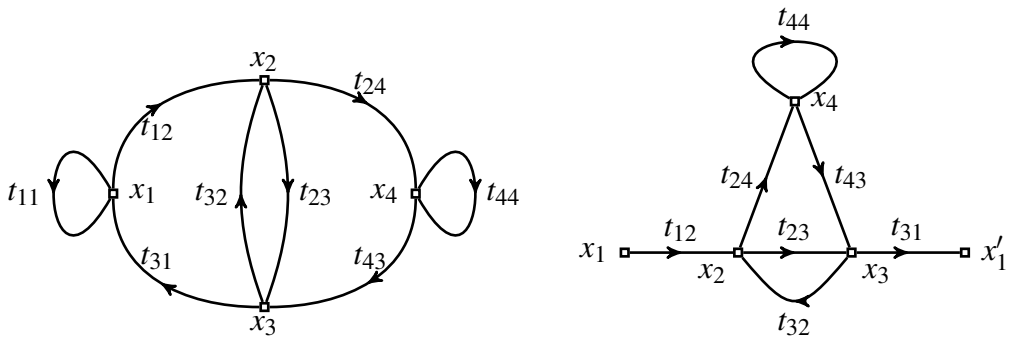
где P_i — передача i-го прямого пути (без замкнутых циклов) между узлами x_j и x_k ($i = 1..p$); L_v — передача v-го контура [38].

В числителе формулы (8.5) должны находиться только те сомножители $(1 - L_v)$, для которых контур L_v не соприкасается с путем P_i. Звездочки в знаменателе показывают, что среди членов соответствующих сумм в произведениях должны отсутствовать соприкасающиеся или пересекающиеся контуры [38].

Для примера рассмотрим нахождение передачи графа для диаграммы состояний сверточного кода, которая используется для решения задачи определения весового спектра сверточного кода [37, 38].

На рис. 8.10 приведены диаграмма состояний сверточного кода (рис. 8.10(а)) и модифицированная диаграмма состояний (рис. 8.10(б)), в которой узел x_1 разделен на начальный узел x_1 и конечный узел x_1'. При этом

рассматриваются только те пути, которые выходят из узла x_1 и в него же возвращаются, не попадая в это состояние в промежуточные моменты — именно поэтому на модифицированной диаграмме (рис. 8.10(б)) отсутствует петля t_{11} [37, 38].



(а) *(б)*

Рис. 8.10. Диаграмма состояний сверточного кода:
(а) исходная; (б) модифицированная

Модифицированная диаграмма на рис. 8.10(б) содержит два пути между узлами x_1 и x_1'

$$P_1 = (x_1x_2x_3x_1') = t_{12}t_{23}t_{31};$$
$$P_2 = (x_1x_2x_4x_3x_1') = t_{12}t_{24}t_{43}t_{31}.$$

Также она содержит три контура

$$L_1 = (x_4x_4) = t_{44};$$
$$L_2 = (x_2x_3x_2) = t_{23}t_{32};$$
$$L_3 = (x_2x_4x_3x_2) = t_{24}t_{43}t_{32}.$$

Таким образом, согласно формуле (8.5), передача графа для диаграммы на рис. 8.10(б) между узлами x_1 и x_1' равна

$$T_{11'} = \frac{P_1(1-L_1)+P_2}{1-(L_1+L_2+L_3)+L_1L_2}.$$

Контрольные вопросы

1. Что такое граф и орграф?
2. Дайте понятие матрицы смежности. Приведите пример.
3. Как строятся матрицы достижимостей и контрадостижимостей?
4. Что такое граф сигналов? Как рассчитываются узловые сигналы в таком графе?
5. Что такое передача графа? Как она рассчитывается?

9. МОДЕЛИ КАНАЛОВ ПЕРЕДАЧИ ДАННЫХ

Точное математическое описание любого реального канала передачи данных обычно весьма сложное [13]. Вместо этого используют упрощенные математические модели, которые позволяют выявить важнейшие закономерности реального канала.

В физическом канале сигнал $S(t)$ подвергается воздействию шума $n(t)$ [39]. Схема этого явления показана на рис. 9.1.

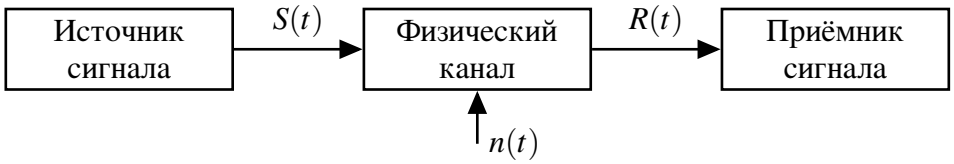


Рис. 9.1. Структурная схема физического канала в общем виде

Для количественной оценки степени влияния шума $n(t)$ на сигнал $S(t)$ обычно используют *отношение сигнал-шум* (SNR), определяемое как отношение мощности сигнала к мощности шума, как показано в формуле

$$\mathrm{SNR} = \frac{P_\mathrm{c}}{P_\mathrm{ш}} = \left(\frac{A_\mathrm{c}}{A_\mathrm{ш}}\right)^2, \tag{9.1}$$

где P — средняя мощность, а A — среднеквадратичное значение амплитуды. Параметры сигнала и шума измеряются в полосе пропускания системы передачи данных.

Как правило отношение сигнал-шум выражается в децибелах и рассчитывается по формуле

$$\mathrm{SNR}_\mathrm{dB} = 10\log_{10}\left(\frac{P_\mathrm{c}}{P_\mathrm{ш}}\right) = 20\log_{10}\left(\frac{A_\mathrm{c}}{A_\mathrm{ш}}\right). \tag{9.2}$$

В цифровой связи основным критерием качества канала связи и системы передачи данных является *отношение сигнал-шум, нормированное на ширину полосы пропускания и битовую скорость передачи данных*. Нормированное отношение сигнал-шум обозначается как $\frac{E_b}{N_0}$ и рассчитывается по формуле (9.3). E_b — это энергия бита, которая представляет из себя мощность сигнала P_c, умноженную на время передачи одного бита T_b. N_0 — это спектральная плотность мощности шума, которая выражается как отношение мощности шума $P_\mathrm{ш}$ на ширину полосы пропускания канала W [40]:

$$\frac{E_b}{N_0} = \frac{P_\mathrm{c}T_b}{P_\mathrm{ш}/W} = \frac{P_\mathrm{c}/R}{P_\mathrm{ш}/W} = \frac{P_\mathrm{c}}{P_\mathrm{ш}} \cdot \frac{W}{R} = \mathrm{SNR} \cdot \frac{W}{R}, \tag{9.3}$$

где R — битовая скорость передачи данных.

Выделяют два основных вида моделей каналов передачи данных. Непрерывные (аналоговые) каналы и дискретные (цифровые) каналы.

Непрерывные каналы имеют непрерывный сигнал $S(t)$ на входе и непрерывный сигнал $R(t)$ на выходе. Эти сигналы являются непрерывной функцией от времени.

Дискретные каналы имеют на входе дискретные кодовые символы x_j, а на выходе — дискретные кодовые символы y_i, в общем случае не совпадающие с x_i [41].

Почти во всех реальных линиях связи дискретный канал содержит внутри себя непрерывный канал, на вход которого подаются сигналы $S(t)$, а с выхода снимаются искаженные помехами сигналы $R(t)$ [41]. Свойства этого непрерывного канала наряду с характеристиками модулятора и демодулятора однозначно определяют все параметры дискретного канала. Поэтому иногда дискретный канал называют дискретным отображением непрерывного канала. Однако при математическом исследовании дискретного канала обычно отвлекаются от непрерывного канала и действующих в нем помех и определяют дискретный канал через алфавит источника $\{x_0, x_1, \ldots, x_{q-1}\}$, вероятности появления символов алфавита, скорость передачи символов, алфавит получателя $\{y_0, y_1, \ldots, y_{Q-1}\}$ и значения переходных вероятностей $P(y_i|x_j)$, где $i = 0, 1, \ldots, Q$, $j = 0, 1, \ldots, q$ [41, 13].

Переходные вероятности $P(y_i|x_j)$ являются вероятностями того, что при отправке в канал символа x_j на выходе будет получен символ y_i.

Если переходные вероятности для каждой пары i, j остаются постоянными и не зависят от того, какие символы передавались и принимались ранее, то дискретный канал называется постоянным или однородным. Иногда применяют также другие названия: канал без памяти или канал с независимыми ошибками. Если же вероятности перехода зависят от времени или от имевших место ранее переходов, то канал называют неоднородным или каналом с памятью [41].

Также выделяют дискретно-непрерывные каналы, которые имеют дискретный вход и непрерывный выход.

9.1. Параметры моделей каналов ПД

Одним из параметров, использующихся для оценки и сравнения моделей каналов ПД является *вероятность безошибочного участка*, определяемая, как вероятность появления последовательности m (или более) безошибочных бит, за которыми следует бит с ошибкой. Она обозначается как $\mathrm{EFR}(m)$[5] [42] либо как $P(0^m|1)$ [43].

[5] От англ. Error Free Run

По аналогии с вероятностью безошибочного участка вводится и *вероятность пачки ошибок*, определяемая, как вероятность появления последовательности из m (или более) ошибок, за которыми следует безошибочный бит. Обозначается как $P(1^m|0)$ [43].

Важным параметром канала ПД является его *пропускная способность*, т. е. максимальная скорость передачи информации по всем допустимым распределениям вероятностей входных сигналов. Пропускная способность канала обозначается как C [41].

С понятием пропускной способности канала связана основная теорема теории информации — теорема кодирования. Она вперные была сформулирована К. Шенноном [44] и заключается в том, что сообщения всякого дискретного источника могут быть закодированы сигналами канала $x(t)$ и восстановлены по сигналам на выходе канала $y(t)$ с вероятностью ошибки, сколь угодно близкой к нулю, если производительность источника с фиксированной скоростью (либо производительность передающего устройства для источника с управляемой скоростью) $H'(x)$ меньше C. Если же $H'(x) > C$, то такое кодирование невозможно [41].

Для источника с управляемой скоростью эта теорема формулируется иначе: сообщения источника с управляемой скоростью можно закодировать сигналами $x(t)$ и восстановить по сигналам $y(t)$ на выходе канала так, чтобы вероятность ошибки была сколь угодно близка к нулю, а средняя скорость передачи — сколь угодно близка к $\frac{C}{H(x)}$ сообщений в секунду, где $H(x)$ — *энтропия источника* на одно сообщение, т. е. средняя собственная информация на символ источника [41, 45].

9.2. Двоичный симметричный канал

Модель двоичного симметричного канала[6] (ДСК) является самой простой моделью дискретного канала [39]. Модель ДСК соответствует случаю использования двоичной модуляции в канале с аддитивным шумом (в котором выходной сигнал $R(t)$ равен сумме входного сигнала $S(t)$ и шума $n(t)$) и жёсткого решения демодулятора. Таким образом, модель ДСК является дискретной двоичной моделью передачи информации по каналу с абсолютно белым гауссовским шумом (АБГШ), описанному в п. 9.7 [3]. Граф, описывающий модель ДСК, представлен на рис. 9.2.

Входом и выходом данного канала являются наборы $X = \{0,1\}$ и $Y = \{0,1\}$ из двух возможных двоичных символов. Также ДСК характеризуется набором переходных вероятностей $P(Y|X)$, определяющих вероятность приёма из канала символа Y при передаче символа X. Переходные вероятно-

[6] В зарубежной литературе используется англоязычное наименование Binary Symmetric Channel (BSC).

сти для ДСК задаются выражениями [3, 46]

$$\begin{aligned} P(0|0) &= P(1|1) = 1 - p_0; \\ P(0|1) &= P(1|0) = p_0, \end{aligned} \tag{9.4}$$

где p_0 — вероятность битовой ошибки в канале.

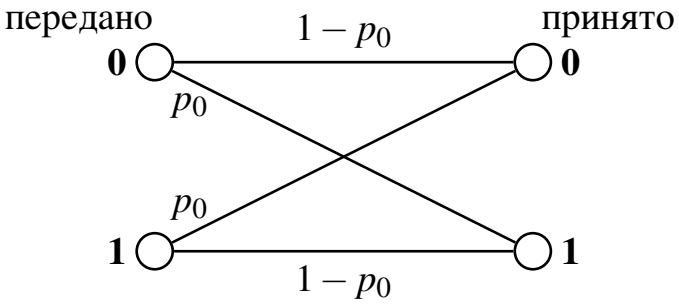


Рис. 9.2. Граф модели двоичного симметричного канала

Для случая использования двух противоположных сигналов $s_0(t) = -s_1(t)$ вероятность битовой ошибки p_0 связана с отношением сигнал-шум выражением [39, 40]

$$p_0 = Q\left(\sqrt{2 \cdot \frac{E_b}{N_0}}\right), \tag{9.5}$$

где $Q(x)$ — функция, определяемая по формуле:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt. \tag{9.6}$$

Переходные вероятности в канале ДСК не зависят от того, какие символы передавались и принимались ранее, и следовательно канал ДСК является каналом без памяти [13].

Пропускная способность канала ДСК рассчитывается как [13, 47]

$$C_{\text{ДСК}} = 1 + p_0 \log_2 p_0 + (1 - p_0) \log_2 (1 - p_0). \tag{9.7}$$

Из формулы (9.7) видно, что при $p_0 = 0,5$ пропускная способность канала C равна нулю. Этот случай называют обрывом канала [13].

Канал ДСК является частным случаем дикретного канала без памяти (ДКБП) [39]. Канал ДКБП имеет на входе набор $\{x_0, x_1, \ldots, x_{q-1}\}$ из q символов, а на выходе — набор $\{y_0, y_1, \ldots, y_{Q-1}\}$ из Q символов, и характеризуется набором из $q \cdot Q$ переходных вероятностей $P(y_i|x_j)$, где $i = 0, 1, \ldots, Q$, $j = 0, 1, \ldots, q$. Эти переходные вероятности постоянны во времени, и переходы различных символов независимы.

9.3. Двоичный симметричный канал со стираниями

Двоичный симметричный канал со стираниями[7] (ДСКС) является важным частным случаем канала ДСК. Как и ДСК, двоичный симметричный канал со стираниями может служить упрощённой моделью передачи данных по каналу АБГШ. Граф, описывающий модель канала ДСКС, представлен на рис. 9.3 [48].

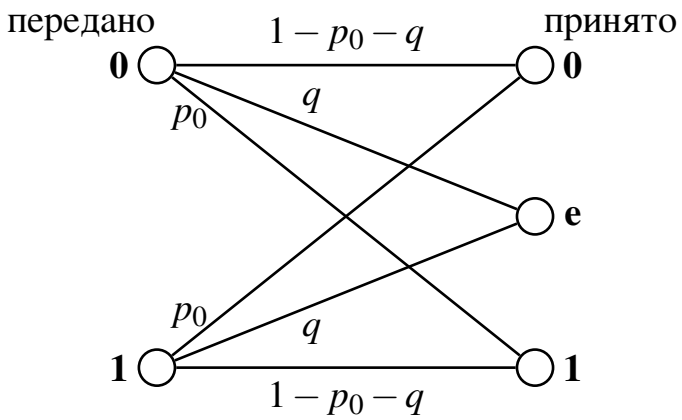


Рис. 9.3. Граф модели двоичного симметричного канала со стираниями

Можно видеть, что по сравнению с моделью ДСК в ДСКС добавляется третье состояние на выходе — «стирание», вероятность которого обозначается q. С точки зрения аналогового канала стирание происходит в случае, когда продетектированный аналоговый сигнал V попадает в зону, в которой значения условных функций плотности распределения вероятностей $f(V/0)$ и $f(V/1)$ оказываются близки к нулю, т. е., когда демодулятор не может надежно опознать переданный символ. Пример подобной ситуации представлен на рис. 9.4 [48].

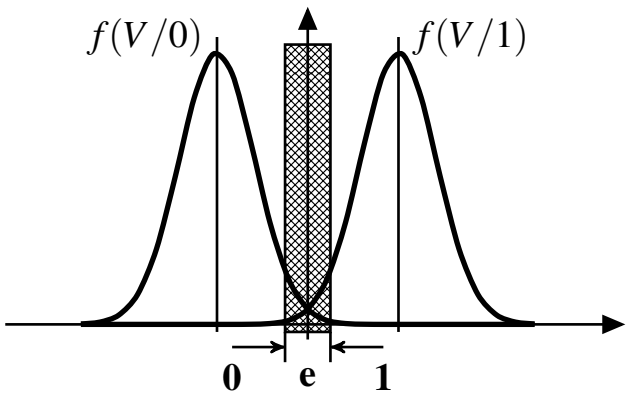


Рис. 9.4. Пример области принятия решений о стирании

[7] В зарубежной литературе используется англоязычное наименование Binary Erasure Channel (BEC).

Матрица переходных вероятностей канала ДСКС равна [48]

$$P_{\text{ДСКС}} = \begin{pmatrix} 1-p_0-q & q & p_0 \\ p_0 & q & 1-p_0-q \end{pmatrix}. \tag{9.8}$$

Пропускная способность канала ДСКС рассчитывается по формуле (9.9) и зависит только от вероятностей p_0 и q, т. е., является функцией $C_{\text{ДСКС}} = f(p_0, q)$ [48]:

$$C_{\text{ДСКС}} = 1 - q + (1-p_0-q)\log_2 \frac{1-p_0-q}{1-q} + p_0 \log_2 \frac{p_0}{1-q}. \tag{9.9}$$

Важным частным случаем канала ДСКС является канал, содержащий только стирания. В таком канале $p_0 = 0$ — т. е. ошибок либо нет, либо мы ими пренебрегаем. На практике такой канал реализуется оптимальным подбором области стирания, показанной на рис. 9.4. Этот вариант канала ДСКС интересен тем, что он позволяет достичь большей пропускной способности, нежели обычный канал ДСК. Пропускная способность такого канала определяется формулой [48]

$$C_{\text{ДСКС}} = 1 - q. \tag{9.10}$$

Граф такой модели ДСКС показан на рис. 9.5.

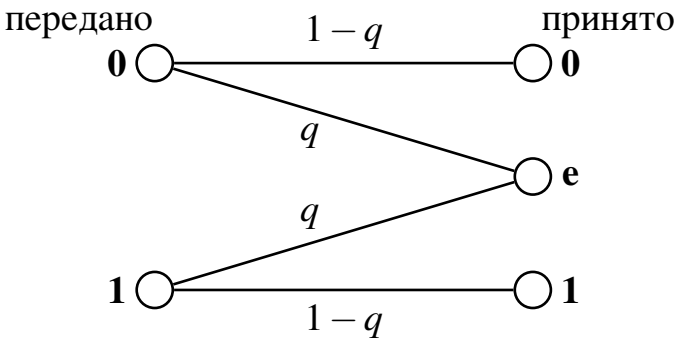


Рис. 9.5. Граф модели двоичного симметричного канала со стираниями для случая $p_0 = 0$

9.4. Двоичный несимметричный канал (Z-канал)

Двоичный несимметричный канал, или Z-канал[8], является каналом без памяти с двоичным входом и двоичным выходом, в котором невозможен переход $0 \longrightarrow 1$. Этот канал получил свое название, благодаря характерному виду своего графа (рис. 9.6) [49, 46]. Поскольку в русскоязычной литературе нет устоявшегося сокращения для канала такого типа, будем использовать англоязычную аббревиатуру ZC.

[8] В зарубежной литературе используется англоязычное наименование Z-Channel (ZC).

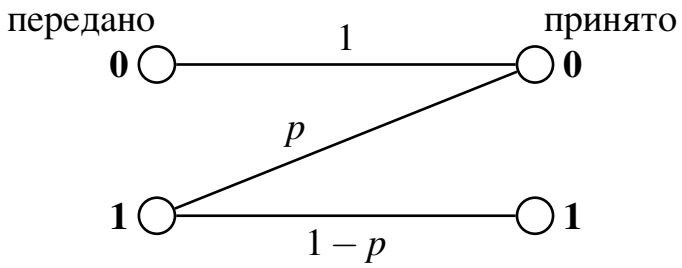


Рис. 9.6. Граф модели двоичного несимметричного канала (Z-канала)

Переходные вероятности для Z-канала задаются выражениями [46]

$$\begin{aligned} P(0|0) &= 1; \quad P(0|1) = p; \\ P(1|0) &= 0; \quad P(1|1) = 1 - p; \end{aligned} \tag{9.11}$$

где p — вероятность ошибки в Z-канале.

Пропускная способность Z-канала рассчитывается по формуле [49]

$$C_{\text{ZC}} = \log_2(1 + (1-p)p^{\frac{p}{1-p}}). \tag{9.12}$$

Модель Z-канала используется при рассмотрении оптической и засекреченной связи [50].

9.5. Канал Гилберта–Эллиотта

Канал Гилберта–Эллиотта[9] (GEC) относится к дискретным каналам с памятью, в которых состояние канала зависит от предыдущего состояния [49, 51]. Эта модель предложена в 1963 г. Эллиоттом [52] и является общим случаем модели Гилберта, представленной в 1960 г. [53].

Канал GEC представляет из себя цепь Маркова первого порядка с двумя состояниями — «хорошим» и «плохим», как показано на рис. 9.7.

Каждое из состояний канала можно описать как канал ДСК с соответствующей вероятностью ошибки [49, 54]. В «хорошем» состоянии вероятность битовой ошибки в канале равна p_G, в «плохом» состоянии — p_B. В любой момент времени канал может перейти из одного состояния в другое. При этом вероятности перехода могут быть отличны друг от друга. Вероятность перехода из «хорошего» состояния в «плохое» обозначим как P_{GB}, а вероятность перехода из «плохого» состояния в «хорошее» обозначим как P_{BG}, что отображено на рис. 9.7. Соответствующая этим вероятностям матрица переходов A имеет вид [49]

$$A = \begin{pmatrix} 1 - P_{GB} & P_{GB} \\ P_{BG} & 1 - P_{BG} \end{pmatrix}. \tag{9.13}$$

[9] В зарубежной и переводной литературе используется англоязычное наименование Gilbert–Elliott Channel (GEC).

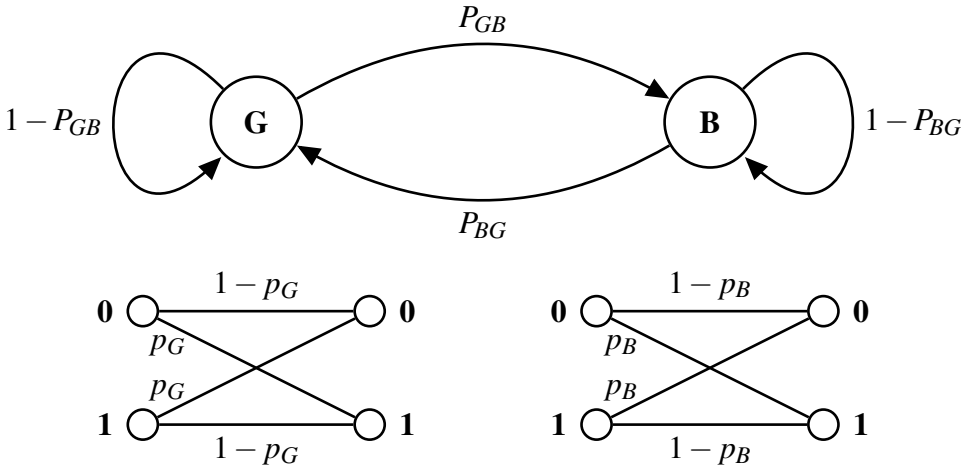


Рис. 9.7. Канал Гилберта–Эллиотта

Из рис. 9.7 следует, что финальные вероятности пребывания канала в состояниях **G** и **B** будут определяться выражениями [51]:

$$\pi_G = \frac{P_{BG}}{P_{GB}+P_{BG}}, \qquad \pi_B = \frac{P_{GB}}{P_{GB}+P_{BG}}. \tag{9.14}$$

Из формул (9.14) следует, что средняя вероятность битовой ошибки в канале может быть вычислена по формуле

$$p_e = p_G \cdot \pi_G + p_B \cdot \pi_B. \tag{9.15}$$

Вероятность того, что в блоке длиной n возникнет m ошибок рассчитывается по формуле

$$P(m,n) = \pi_G \cdot G(m,n) + \pi_B \cdot B(m,n), \tag{9.16}$$

где $G(m,n)$ — вероятность появления m ошибок в блоке длиной n, при условии, что канал во время передачи первого бита находился в состоянии **G**; $B(m,n)$ — вероятность появления m ошибок в блоке длиной n, при условии, что канал во время передачи первого бита находился в состоянии **B**.

Для расчета этих вероятностей Эллиоттом были введены рекуррентные соотношения (9.17), описывающие процесс возникновения ошибок в канале, учитывая, что канал с каждым поступившим новым разрядом может оставаться в прежнем состоянии или переходить в другое [52]:

$$
\begin{aligned}
G(m,n) = \; & G(m,n-1)\cdot(1-P_{GB})\cdot(1-p_G)+ \\
& +B(m,n-1)\cdot P_{BG}\cdot(1-p_G)+ \\
& +G(m-1,n-1)\cdot(1-P_{GB})\cdot p_G+ \\
& +B(m-1,n-1)\cdot P_{BG}\cdot p_G, \\
B(m,n) = \; & G(m,n-1)\cdot P_{GB}\cdot(1-p_B)+ \\
& +B(m,n-1)\cdot(1-P_{BG})\cdot(1-p_B)+ \\
& +G(m-1,n-1)\cdot P_{GB}\cdot p_B+ \\
& +B(m-1,n-1)\cdot(1-P_{BG})\cdot p_B.
\end{aligned}
\tag{9.17}
$$

В формулах (9.18) приведены очевидные начальные значения вероятностей (9.17) при $n = 1$ [52]:

$$
\begin{aligned}
&G(0,1) = (1-p_G), \quad B(0,1) = (1-p_B), \\
&G(1,1) = p_G, \qquad\quad B(1,1) = p_B.
\end{aligned}
\tag{9.18}
$$

Также необходимо учитывать, что:

$$
G(m,n) = B(m,n) = 0, \quad \text{при } m < 0 \text{ или } m > n.
$$

Вероятность безошибочного участка (в случае стационарности) для канала GEC рассчитывается по формуле [42]:

$$
\mathrm{EFR}_{GEC}(m) = \pi_G p_G (1-p_G)^m + \pi_B p_B (1-p_B)^m. \tag{9.19}
$$

Канал GEC широко используется для описания источников ошибок в системах передачи данных, а также при анализе эффективности алгоритмов декодирования помехоустойчивых кодов [51].

Существуют исследования, показывающие, что канал GEC близок по своим свойствам к преобразованному в двоичную форму (квантованному) двухлучевому Релеевскому каналу с замираниями без поворота фазы [42].

Часто при использовании модели GEC для двоичного канала полагают, что вероятность $p_B = 0,5$, т. е. «плохое» состояние рассматривается как полный обрыв связи [13]. Это согласуется с представлением о канале, в котором действуют коммутативные помехи.

9.6. Модель канала Поля

В начале 90-х гг. XX века было определено, что для описания распределения ошибок в коммуникационном канале с памятью может быть использована модель Г. Поля, используемая для моделирования распространения заболеваний [55].

Канал Поля является дискретным двоичным аддитивным каналом связи, в котором сигнал на выходе $y_i \in \{0,1\}$ равен сумме по модулю 2 соответствующих ему сигнала на входе $x_i \in \{0,1\}$ и бита ошибки $z_i \in \{0,1\}$: $y_i = x_i \oplus z_i$, для $i = 1,2,3,\ldots$ [55].

Принцип работы канала Поля заключается в следующем. Имеется урна, в которой изначально содержится T шаров, из которых R красных и S черных ($T = R+S$), при этом $\rho = R/T$ и $\sigma = 1-\rho = S/T$. На каждом шаге i из урны вытаскивается случайный шар, так что

$$Z_i = \begin{cases} 1, & \text{вытащен красный шар;} \\ 0, & \text{вытащен черный шар.} \end{cases} \tag{9.20}$$

После этого в урну возвращается $1+\Delta$ того же цвета, что и вытащенный. Δ — параметр модели канала (целое число). Как правило, предполагают, что $\Delta > 0$ и $\rho < \sigma$. Дополнительно вводится параметр $\delta = \Delta/T$ [55].

Состоянием канала Поля после n шагов, является количество красных шаров, вытащенных за это время:

$$S_n \triangleq Z_1 + Z_2 + \ldots + Z_n = S_{n-1} + Z_n. \tag{9.21}$$

Соответственно, начальное состояние канала: $S_0 = 0$ [55].

Таким образом, состояние канала на шаге n может принимать $n+1$ значение: $\{0,1,\ldots,n\}$, а последовательность состояний $\{S_n\}_{n=1}^{\infty}$ формирует марковскую цепь [55]

$$P(S_N = s_n | S_{n-1} = s_{n-1}, S_{n-2} = s_{n-2}, \ldots, S_1 = s_1) = P(S_N = s_n | S_{n-1} = s_{n-1}).$$

Для заданного блока данных на входе $\mathbf{X} = [X_1, X_2, \ldots, X_n]$ заданного для него блока выходных данных $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_n]$ блочная переходная вероятность равна

$$P(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}) = \frac{\Gamma\left(\frac{1}{\delta}\right)\Gamma\left(\frac{\rho}{\delta}+d\right)\Gamma\left(\frac{\sigma}{\delta}+n-d\right)}{\Gamma\left(\frac{\rho}{\delta}\right)\Gamma\left(\frac{\sigma}{\delta}\right)\Gamma\left(\frac{1}{\delta}+n\right)}, \tag{9.22}$$

где d — расстояние Хэмминга между $\mathbf{x}$ и $\mathbf{y}$; а $\Gamma(x) = \int\limits_0^{\infty} t^{x-1} e^{-t} dt$ для $x > 0$ — гамма-функция [42, 55].

Вероятность безошибочного участка (в случае стационарности) для канала Поля рассчитывается по формуле [42]

$$\mathrm{EFR}_{Polya}(m) = \left(\frac{R}{R+S}\right) \prod_{j=1}^{m} \left(\frac{S+(j-1)\Delta}{R+S+j\Delta}\right). \tag{9.23}$$

Нереалистичность канала Поля заключается в его бесконечной памяти — каждый «вытащенный из урны» красный шар, хоть первый, хоть миллионный, приводит к одному и тому же увеличению количества красных шаров [55]. В связи с этим в работе [55] предлагается модель канала Поля с конечной памятью, в которой влияние каждого вытащенного шара ограничено по времени.

Принцип работы канала Поля с конечной памятью заключается в следующем. В урне изначально содержится T шаров, из которых R красных и S черных, так что $T = R + S$. На каждом j-м шаге ($j = 1, 2, \ldots$) из урны вытаскивается шар и затем заменяется набором из $\Delta + 1$ шаров того же цвета ($\Delta > 0$). Через M шагов, на $(j + M)$-м шаге из урны изымаются те Δ шаров, что были добавлены на j-м шаге. Параметры ρ, σ и δ вычисляются аналогично обычному каналу Поля. Влияние вытащенных шаров на канал также описывается формулой (9.20). В итоге, после периода инициализации канала длиной M шагов, общее число шаров в урне фиксируется и становится равным $(T + M\Delta)$ [55].

9.7. Канал с аддитивным белым гауссовским шумом

Канал с аддитивным белым гауссовским шумом[10] (АБГШ) получается из канала ДКБП при бесконечном уровне квантования выхода детектора ($Q = \infty$) [39]. В этом случае шум является гауссовской случайной величиной с нулевым средним и дисперсией, равной

$$\sigma^2 = \frac{1}{2 \cdot \frac{E_b}{N_0}},$$

где $\frac{E_b}{N_0}$ — нормированное отношение сигнал-шум в канале АБГШ.

Таким образом, канал АБГШ характеризуется дискретным входом $X = \{x_0, \ldots, x_{q-1}\}$, непрерывным выходом $Y = \{-\infty, +\infty\}$ и переходными вероятностями:

$$P(y|x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x_j)^2}{2\sigma^2}}, \quad j = 0, 1, \ldots, q-1. \tag{9.24}$$

Для канала АБГШ зависимость вероятности ошибки p_0 от нормированного отношения сигнал-шум $\frac{E_b}{N_0}$ определяется в соответствии с выражением (9.5).

Модель канала АБГШ широко применяется при расчёте и моделировании многих систем радиосвязи, особенно при моделировании каналов спутниковой и дальней космической связи [56].

[10] В зарубежной литературе используется англоязычный термин Additive White Gaussian Noise (AWGN).

Поскольку в помехоустойчивом кодировании работа производится с дискретными данными, при проведении моделирования с использованием канала АБГШ (и других аналоговых каналов) перед передачей закодированных данных в канал необходимо проводить процедуру манипуляции, а после приема данных из канала — обратную процедуру. При работе с двоичными данными часто используется *двоичная фазовая манипуляция*[11] (ФМн-2).

Контрольные вопросы

1. Дайте понятие двоичного-симметричного канала. Как рассчитывается его пропускная способность?
2. Что такое двоичный симметричный канал со стираниями?
3. Опишите модель канала Гилберта–Эллиотта.
4. Что такое канал Поля?

[11] В зарубежной литературе используется англоязычный термин Phase shift keying modulation (PSK).

ЗАКЛЮЧЕНИЕ

В пособии рассмотрены математические основы теории помехоустойчивых кодов. Приведены и рассмотрены на примерах алгоритмы, использующиеся при построении кодирующих и декодирующих устройств.

Ввиду ограниченного объема пособия в нем не был рассмотрен такой важный вопрос, как оценка эффективности применения помехоустойчивых кодов в системах передачи данных. Тем не менее, изучив математический аппарат помехоустойчивых кодов будущие специалисты в области телекоммуникаций смогут успешно справиться с этой задачей.

СПИСОК ЛИТЕРАТУРЫ

[1] Шварцман, В.О. Теория передачи дискретной информации: Учебник для вузов связи. / В.О. Шварцман, Г.А. Емельянов. — М. : Связь, 1979.

[2] Кларк, Д.К. Кодирование и исправление ошибок в системах цифровой связи / Д.К. Кларк, Д.Б. Кейн. Статистическая теория связи. — М. : «Радио и Связь», 1987.

[3] Гладких, А.А. Основы теории мягкого декодирования избыточных кодов в стирающем канале связи / А.А. Гладких. — Ульяновск : УлГТУ, 2010.

[4] Когновицкий, О.С. Основы циклических кодов. Учебное пособие / О.С. Когновицкий. — Л. : ЛЭИС, 1990.

[5] Ковриженко, Г.А. Системы счисления и двоичная арифметика: От счета на пальцах до ЭВМ / Г.А. Ковриженко. — К. : Рад. шк., 1984.

[6] Фомин, С.В. Системы счисления / С.В. Фомин. — 5-е изд. — М. : Наука, 1987.

[7] Гантмахер, Ф.Р. Теория матриц / Ф.Р. Гантмахер. — 2-е изд. — М. : Наука, 1966.

[8] Ланкастер, П. Теория матриц: Пер. с англ / П. Ланкастер. — 2-е изд. — М. : Наука, 1982.

[9] Gray, R.M. Toeplitz and Circulant Matrices: A Review / R.M. Gray // Foundations and Trends in Communications and Information Theory. — 2006. — Vol. 2, no. 3. — P. 155–239.

[10] Ежов, И.И. Элементы комбинаторики / И.И. Ежов, А.В. Скороход, М.И. Ядренко. — М. : Наука, 1977.

[11] Корн, Г. Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. — М. : Наука, 1973.

[12] Винберг, Э.Б. Алгебра многочленов / Э.Б. Винберг. — М. : Просвещение, 1980.

[13] Теория электрической связи: учебное пособие / К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко ; Под ред. К.К. Васильева. — Ульяновск : УлГТУ, 2008. — ISBN: 978-5-9795-0203-8.

[14] Постников, М.М. Основы теории Галуа / М.М. Постников. — М. : Физматгиз, 1960.

[15] Постников, М.М. Теория Галуа / М.М. Постников. — М. : Физматгиз, 1968.

[16] Чеботарёв, Н.Г. Основы теории Галуа. Часть 1 / Н.Г. Чеботарёв. — М.-Л. : ОНТИ, 1934.

[17] Чеботарёв, Н.Г. Теория Галуа. Книга 1 / Н.Г. Чеботарёв ; Под ред. И.М. Виноградова. — М.-Л. : ОНТИ, 1936.

[18] Чеботарёв, Н.Г. Основы теории Галуа. Часть 2 / Н.Г. Чеботарёв. — М.-Л. : ОНТИ, 1937.

[19] Лидл, Р. Конечные поля: В 2-х т. : Пер. с англ / Р. Лидл, Г. Нидеррайтер ; Под ред. В.И. Нечаева. — М. : Мир, 1988. — ISBN: 978-5-0300-0064-0.

[20] Когновицкий, О.С. Двойственный базис и его применение в телекоммуникациях / О.С. Когновицкий. — СПб. : Линк, 2009. — ISBN: 978-5-98595-020-5.

[21] Теория кодирования / Т. Касами, Н. Токура, Е. Ивадари, Я. Ирагаки. — М. : Мир, 1978.

[22] Мак-Вильямс, Ф.Дж. Теория кодов, исправляющих ошибки / Ф.Дж. Мак-Вильямс, Н.Дж.А. Слоэн. — М. : Связь, 1979.

[23] Кукунин, Д.С. Дискретное логарифмирование в полях Галуа с использованием контрольных точек / Д.С. Кукунин // Научно-технические ведомости СПбГПУ. Серия «Информатика. Телекоммуникации. Управление». — 2009. — Т. 2, № 76. — С. 185–192.

[24] Винберг, Э.Б. Малая теорема Ферма и ее обобщения / Э.Б. Винберг // Математическое просвещение. — Сер. 3 № 12. — М. : МЦНМО, 2008. — С. 43–53.

[25] Kobayashi, K. An Algorithm for Inversion in GF(2^M) Suitable for Implementation Using a Polynomial Multiply Instruction on GF(2) / K. Kobayashi, N. Takagi, K. Takagi // Proceedings of the 18th IEEE Symposium on Computer Arithmetic. — ARITH '07. — Washington, DC, USA : IEEE Computer Society, 2007. — P. 105–112.

[26] Hankerson, D. Software Implementation of Elliptic Curve Cryptography over Binary Fields / D. Hankerson, J. Lopez Hernandez, A. Menezes // Cryptographic Hardware and Embedded Systems — CHES 2000 / Ed. by CetinK. Koc, Christof Paar. — [S. l.] : Springer Berlin Heidelberg, 2000. — Vol. 1965 of Lecture Notes in Computer Science. — P. 1–24.

[27] Карацуба, А. Умножение многозначных чисел на автоматах / А. Карацуба, Ю. Офман // Доклады Академии Наук СССР. — [Б. м. : б. и.], 1962. — Т. 145.

[28] Rodriguez-Henriquez, F. On Fully Parallel Karatsuba Multipliers for GF(2m) / F. Rodriguez-Henriquez, C.K. Koc // Computer Science and Technology. — 2003.

[29] Shohdy, S.M. Hardware Implementation of Efficient Modified Karatsuba Multiplier Used in Elliptic Curves / S.M. Shohdy, A.B. El-Sisi, N. Ismail // International Journal of Network Security. — 2010. — Vol. 11, no. 3. — P. 155–162.

[30] Ajitha, S.S. Efficient Implementation of Bit Parallel Finite Field Multipliers / S.S. Ajitha, D. Retheesh // International Journal of Research in Engineering and Technology. — 2014. — Vol. 3, no. 3. — P. 661–667.

[31] Reyhani-Masoleh, A. Low complexity bit parallel architectures for polynomial basis multiplication over GF(2^m) / A. Reyhani-Masoleh, M.A. Hasan // Computers, IEEE Transactions on. — 2004. — Aug. — Vol. 53, no. 8. — P. 945–959.

[32] Харари, Ф. Теория графов / Ф. Харари ; Под ред. Г.П. Гаврилова. — М. : Мир, 1973.

[33] Басакер, Р. Конечные графы и сети / Р. Басакер, Т. Саати. — М. : Наука, 1974.

[34] Кристофидес, Н. Теория графов. Алгоритмический подход / Н. Кристофидес. — М. : Мир, 1978.

[35] Берж, К. Теория графов и ее применения / К. Берж. — М. : Изд-во Иностранной литературы, 1962.

[36] Мэзон, С. Электронные цепи, сигналы и системы / С. Мэзон, Г. Циммерман ; Под ред. проф. П.А. Ионкина. — М. : Издательство иностранной литературы, 1963.

[37] Деев, В.В. Методы модуляции и кодирования в современных системах связи / В.В. Деев. — Спб. : Наука, 2007. — ISBN: 978-5-02=025182-3.

[38] Когновицкий, О.С. Теория помехоустойчивого кодирования. Часть 2. Сверточные коды. Турбокоды / О.С. Когновицкий, В.М. Охорзин, И.А. Небаев. — СПб. : СПбГУТ, 2015.

[39] Золотарёв, В.В. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник / В.В. Золотарёв, Г.В. Овечкин ; Под ред. чл.-корр. РАН Ю.Б. Зубарева. — М. : Горячая линия–Телеком, 2004.

[40] Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр ; Под ред. А.В. Назаренко. — М. : Издательский дом «Вильямс», 2003. — ISBN: 5-8459-0497-8.

[41] Финк, Л.М. Теория передачи дискретных сообщений / Л.М. Финк. — М. : «Сов. радио», 1970.

[42] Similarity of Discrete Gilbert-Elliot and Polya Channel Models to Continuous Rayleigh Fading Channel Model : Rep. / National Chiao Tung University ; Executor: Pen-Ting Sun. — Taiwan 30050, R.O.C. : 2002. — June.

[43] Jeruchim, M.C. Simulation of Communication Systems: Modeling, Methodology and Techniques / M.C. Jeruchim, P. Balaban, K.S. Shanmugan. Information Technology: Transmission, Processing and Storage. — [S. l.] : Springer US, 2006. — ISBN: 9780306469718.

[44] Шеннон, К. Математическая теория связи / К. Шеннон // Работы по теории информации и кибернетике. — [Б. м.] : Изд-во иностранной литературы, 1963.

[45] Прокис, Дж. Цифровая связь / Дж. Прокис ; Под ред. Д.Д. Кловского. — М. : «Радио и Связь», 2000. — ISBN: 5-256-01434-X.

[46] MacKay, D.J.C. Information Theory, Inference, and Learning Algorithms / D.J.C. MacKay. — Cambridge : Cambridge University Press, 2003. — ISBN: 0-521-64298-1.

[47] Зайдлер, Е. Системы передачи дискретной информации / Е. Зайдлер ; Под ред. Б.Р. Левина. — М. : «Связь», 1977.

[48] Вернер, М. Основы кодирования. Учебник для ВУЗов / М. Вернер. Мир программирования. — М. : Техносфера, 2006. — ISBN: 5-94836-019-9.

[49] Richardson, T. Modern Coding Theory / T. Richardson, R. Urbanke. — Cambridge : Cambridge University Press, 2008. — ISBN: 978-0-5218-5229-6.

[50] Palmieri, P. Secure Two-Party Computation over a Z-Channel / P. Palmieri, O. Pereira // Provable Security / Ed. by Xavier Boyen, Xiaofeng Chen. — [S. l.] : Springer Berlin Heidelberg, 2011. — Vol. 6980 of Lecture Notes in Computer Science. — P. 3–15.

[51] Hasslinger, G. The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet / G. Hasslinger, O. Hohlfeld // Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference. — [S. l. : s. n.], 2008. — P. 1–15.

[52] Elliott, E. O. Estimates of error rates for codes on burst-noise channels / E. O. Elliott // Bell System Technical Journal. — 1963. — Vol. 42. — P. 1977–1997.

[53] Gilbert, E. N. Capacity of a burst-noise channel / E. N. Gilbert // Bell System Technical Journal. — 1960. — September. — Vol. 39. — P. 1253–1265.

[54] Rezaeian, M. Computation of capacity for Gilbert-Elliott channels, using a statistical method / M. Rezaeian // Communications Theory Workshop, 2005. Proceedings. 6th Australian. — [S. l. : s. n.], 2005. — P. 56–61.

[55] A Communication Channel Modeled on Contagion : Rep. : T.R. 93-78r1 / Institute for System Research ; Executor: F. Alajaji, T. Fuja : 1993. — August.

[56] Massey, J.L. Deep-Space Communications and Coding: A Marriage Made in Heaven / J.L. Massey // in Proceedings of the 1992 DLR Seminar Advanced Methods for Satellite and Deep Space Comm. — [S. l.] : Springer, 1992. — P. 1–17.

Владимиров Сергей Сергеевич

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Учебное пособие

Редактор *Л. К. Паршина*

План изданий 2016, п. 45

Подписано к печати 30.06.2016
Объем 6,00 усл.-печ. л. Тираж 26 экз. Заказ 696

Редакционно-издательский отдел СПбГУТ
191186 СПб., наб. р. Мойки, 61
Отпечатано в СПбГУТ