

Хмельник С. И.

Кодирование
КОМПЛЕКСНЫХ ЧИСЕЛ И
векторов

Израиль Россия
2004

Solomon I. Khmelnik

Coding of Complex Numbers and Vectors

(in Russian)

Copyright © 2004 by Solomon I. Khmelnik

All right reserved. No portion of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, without written permission of the author.

Published by “MiC” - Mathematics in Computer Comp.

BOX 15302, Beney-Ayish, Israel, 79845

Fax: ++972-3-9471301

<http://comarithmetic.com>

Printed in United States of America, Lulu Inc.

Israel Russia
2004

Аннотация

В книге рассматриваются малоизвестная теория и кодирования комплексных чисел и векторов, которая может быть применена для разработки специализированных процессоров. Описывается структура кодов, алгоритмы кодирования, декодирования и арифметических операций. Теория дополняется многочисленными примерами.

Книга ориентирована на студентов, инженеров и разработчиков специализированных процессоров.

Предлагаемые в книге алгоритмы и устройства разрабатываются в виде моделей на VHDL и FPGA. Любые предложения о сотрудничестве посылайте по адресу: solik@netvision.net.il

Содержание

Подробное оглавление \	5
Предисловие \	7
Глава 1. Позиционные коды комплексных чисел и векторов \	10
Глава 2. Точность кодирования \	38
Глава 3. Поразрядные арифметические операции \	61
Глава 4. Алгоритмы кодирования и декодирования комплексных чисел \	83
Глава 5. Умножение \	87
Глава 6. Деление \	107
История вопроса \	116
Литература \	118

Подробное оглавление

Предисловие \ 7

Глава 1. Позиционные коды комплексных чисел и векторов \ 10

1. О методе позиционного кодирования \ 10
2. Два способа синтеза кодов комплексных чисел \ 12
3. Метод кодирования точек многомерного пространства \ 14
4. Арифметические системы кодирования \ 20
5. Коды действительных чисел \ 23
6. Коды комплексных чисел \ 25
7. Коды многомерных векторов \ 34

Глава 2. Точность кодирования \ 38

1. Область представимых чисел \ 38
2. Область представимых модулей \ 46
3. Погрешности конечных кодов \ 53
4. Коды с плавающей точкой \ 56

Глава 3. Поразрядные арифметические операции \ 61

1. Поразрядные операции \ 61
2. Первый алгоритм поразрядных операций \ 63
3. Второй алгоритм поразрядных операций \ 70
4. Поразрядные операции при отрицательном основании \ 81

Глава 4. Алгоритмы кодирования и декодирования комплексных чисел \ 83

Глава 5. Умножение \ 87

1. Специальная алгебра в векторном пространстве \ 88
 - 1.1. Алгебра в трехмерном векторном пространстве \ 88
 - 1.2. Покомпонентное умножение \ 89
 - 1.3. Векторное произведение \ 89
 - 1.4. Скалярное произведение \ 89
 - 1.5. Поворот вектора \ 89
 - 1.6. Центроаффинное преобразование \ 90
 - 1.7. Многомерное пространство \ 91
2. Умножение многомерных векторов \ 92

- 2.1. Метод умножения комплексных чисел и многомерных векторов \ 92
- 2.2. Умножение векторов по основанию (1.31) \ 93
- 2.3. Умножение векторов по основанию (1.30) \ 93
- 2.4. Последовательное и матричное умножение \ 94
- 2.5. Умножение целых кодов векторов по основанию (1.31) \ 97
- 2.6. Умножение целых кодов векторов по основанию (1.30) \ 97
- 2.7. Покомпонентное умножение многомерных векторов \ 98

3. Скалярное и векторное умножения \ 100

- 3.1. Скалярное произведение \ 100
- 3.2. Векторное произведение \ 101
- 3.3. Переносы при скалярном умножении \ 102
- 3.4. Переносы при векторном умножении \ 104

Глава 6. Деление \ 107

- 1. Метод деления \ 107**
- 2. Деление комплексных чисел по основанию ρ_2 \ 110**
- 3. Деление трехмерных векторов \ 114**

История вопроса \ 116

Литература \ 118

Предисловие

В книге рассматривается теория позиционного кодирования комплексных чисел и векторов. Предпочтение, отдаваемое именно позиционным кодам, объясняется тем, что с ними очень просто выполняются арифметические операции. Так, вне зависимости от объекта кодирования сложение позиционных кодов связано с распространением переносов от младших разрядов к старшим, а умножение и деление состоит из сдвигов и алгебраических сложений. Метод «цифра-за-цифрой» вообще применим только в сочетании с позиционной системой кодирования.

Далее показано, что с позиционными кодами комплексных чисел и векторов выполнимы операции алгебраического сложения, векторного, скалярного и специального умножения, деления комплексных чисел и векторов. Это может быть использовано при построении процессоров, оперирующих с комплексными числами и векторами в целом. Такой процессор требует более простого алгоритма для решения задач, а при данном алгоритме работает по более короткой программе и обладает повышенным быстродействием. Для оценки этих величин можно указать, например, что программа векторного умножения векторов, заданных тремя числами, содержит 6 операций умножения и 3 операции вычитания.

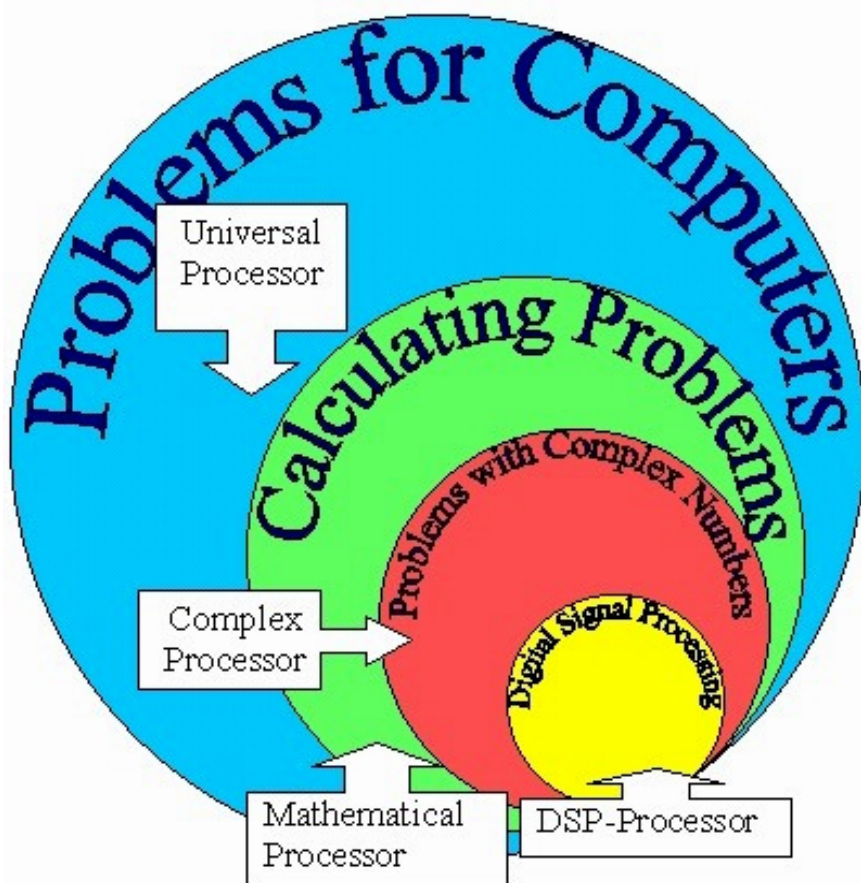
Укажем некоторые области применения комплексных чисел: телекоммуникация, цифровая фильтрация, обработка изображений, обработка сигналов в радарх/сонарах, распознавание сигналов, спектральный анализ, инженерная графика в системах автоматизированного проектирования, навигационные системы, компьютерные игры, медицинская диагностика, проектирование электромеханических систем, управление энергосистемами, многоканальные цифровые системы автоматического регулирования. Вообще, вычислительная математика в большей своей части является математикой комплексных чисел. Поэтому универсальный компьютер для вычислений – это компьютер, оперирующий с комплексными числами. Этот факт не учитывается практикой по нескольким причинам:

- Программы позволяют завуалировать базовые возможности компьютера (на самом деле даже

арифметические операции можно было бы заменить на более простые – счетчик и сравнение с нулем),

- Существуют специализированные компьютеры для решения вычислительных задач, требующих значительного быстродействия,
- Не было предложено комплексной компьютерной арифметики.

Арифметическое устройство для операций с комплексными числами САУ превращает компьютер, оснащенный им, в универсальный математический компьютер. Место математического компьютера в общем круге компьютерных задач иллюстрируется следующей диаграммой.



Книга содержит 6 глав. В главе 1 рассматривается теория

позиционного кодирования комплексных чисел и векторов. Описывается метод поиска оснований кодирования и предлагаются различные основания кодирования. Выделяется несколько вариантов построения двоичных кодов комплексных чисел и векторов. В **главе 2** исследуются вопросы точности представления комплексных чисел кодами, построенными выше. В **главе 3** рассматриваются операции, которые можно выполнять на схемах с распространением переноса. Формулируются два алгоритма таких операций и рассматриваются многочисленные частные случаи операций с различными кодами. В **главе 4** предлагаются алгоритмы для преобразования чисел из одной системы кодирования в другую. В **главе 5** рассматриваются методы, алгоритмы и схемы умножения чисел и векторов, в т. ч. скалярного, векторного и специального умножения векторов. Для последнего случая строится специальная алгебра. В **главе 6** рассматриваются методы, алгоритмы и схемы деления комплексных чисел, а также деления векторов в рассмотренной при умножении специальной алгебре.

В книге принята самостоятельная нумерация рисунков, таблиц, формул, алгоритмов внутри каждого раздела. При ссылке на элемент другого раздела номер элемента дополняется префиксом другого раздела.

Глава 1. Позиционные коды комплексных чисел и векторов

1. О методе позиционного кодирования \ 10
2. Два способа синтеза кодов комплексных чисел \ 12
3. Метод кодирования точек многомерного пространства \ 14
4. Арифметические системы кодирования \ 20
5. Коды действительных чисел \ 23
6. Коды комплексных чисел \ 25
7. Коды многомерных векторов \ 34

1. О методе позиционного кодирования

В этом разделе будут рассмотрены позиционные коды многомерных векторов Z , основанные на их представлении в виде разложения

$$Z = \sum_m r_m f(\rho, m), \quad (1)$$

где

m - номер разряда,

ρ - основание кодирования, число или вектор,

$f(\rho, m)$ - базовая функция номера и основания,

r - разряд разложения, число или вектор, принимающий значения из ограниченного множества

$A_R = \{a_0, a_1, a_2, \dots, a_j, \dots, a_{R-1}\}$, содержащего R

различных величин a_j .

Позиционный код вектора Z , соответствующий этому разложению, имеет вид

$$K(Z) = \dots \sigma_m \dots,$$

где σ_m - цифра, обозначающая величину r_m . Формула (1) включает операции сложения и умножения. Для существования алгоритмов операций с такими разложениями (или, что одно и то же, с позиционными кодами) сложение и умножение должны быть ассоциативными и коммутативными, а также подчиняться дистрибутивному закону. Следовательно, для возможности позиционного кодирования некоторого множества объектов это множество должно составлять кольцо. Такому требованию удовлетворяет множество действительных чисел и множество многомерных векторов, в котором определены операции сложения и умножения на число. Для действительных чисел позиционные системы известны. Для указанного множества векторов ниже будет построена позиционная система счисления с действительным основанием.

Множество комплексных чисел составляет кольцо и для него также будут построены позиционные системы счисления по действительному и комплексному основаниям.

Для построения позиционной системы счисления многомерных векторов по векторному основанию должна быть определена операция умножения векторов, подчиняющаяся вышеперечисленным законам. Другими словами, должна быть определена алгебра в многомерном векторном пространстве. Это сделано ниже.

Вначале рассмотрим два способа кодирования, а затем перейдем к более общему и строгому описанию метода позиционного кодирования.

2. Два способа синтеза кодов КОМПЛЕКСНЫХ ЧИСЕЛ

Позиционные коды комплексных чисел могут быть получены некоторой композицией кодов действительных чисел по отрицательному основанию. Здесь и далее j - мнимая единица.

Пусть X_α и X_β - действительные числа, заданные двоичными разложениями по основанию $\rho = -2$, то есть

$$X_\alpha = \sum_{(m)} \alpha_m \rho^m, \quad X_\beta = \sum_{(m)} \beta_m \rho^m.$$

Этим разложениям соответствуют коды

$$K(X_\alpha) = \alpha_m, \quad K(X_\beta) = \beta_m.$$

Существуют два способа объединения этих двух кодов в единый код комплексного числа. **Первый** из них заключается в том, что пара разрядов α_m и β_m обозначается одной цифрой σ_m . При этом образуется код

$$K(Z) = \dots \sigma_m \dots$$

комплексного числа $Z = X_\alpha + j \cdot X_\beta$ по основанию $\rho = -2$ с разрядами, принимающими одно из четырех значений:

$$\sigma_m \in \{0, 1, j, 1+j\}.$$

Рассмотрим комплексную функцию от действительного целого аргумента m :

$$\rho_2 = \left\{ \begin{array}{l} (-2)^{m/2} \text{ if } m - \text{even} \\ j(-2)^{m-1/2} \text{ if } m - \text{odd} \end{array} \right\} \quad (2)$$

При этом рассматриваемый код комплексного числа по основанию (-2) с комплексными значениями разрядов может рассматриваться

как код комплексного числа по основанию ρ_2 с двоичными разрядами. Этому коду соответствует разложение комплексного числа в виде $Z = \sum_m (\sigma_m \rho_2)$. где двоичные разряды

$\sigma_m = \left\{ \begin{array}{l} \alpha_m \text{ if } m - \text{even} \\ j \cdot \beta_m \text{ if } m - \text{odd} \end{array} \right\}$. Для иллюстрации запишем коды

некоторых характерных чисел в этой этой системе: $K(2) = 10100$, $K(-2) = 100$, $K(-1) = 101$, $K(j) = 10$, $K(-j) = 1010$, $K(2j) = 101000$.

Второй способ объединения этих двух кодов в единый код комплексного числа состоит в построении последовательности чередующихся разрядов α_m и β_m :

$$\dots \beta_{m+1} \alpha_{m+1} \beta_m \alpha_m \beta_{m-1} \alpha_{m-1} \dots$$

Обозначим $\alpha_m = \sigma_{2m}$, $\beta_m = \sigma_{2m+1}$ и перепишем указанную последовательность в ином виде:

$$\dots \sigma_{k+3} \sigma_{k+2} \sigma_{k+1} \sigma_k \sigma_{k-1} \sigma_{k-2}$$

где $k=2m$. Эта последовательность является двоичным кодом

$$K(Z) = \dots \sigma_m \dots$$

некоторого комплексного числа Z . Можно показать (и это будет сделано ниже), что код, полученный таким способом, является двоичным кодом по основанию $\rho = \pm j\sqrt{2}$, а закодированное число $Z = X_\alpha + \rho \cdot X_\beta$.

Таким образом, некоторые композиции двоичных кодов действительных чисел по основанию $\rho = -2$ образуют коды комплексных чисел. При выполнении алгебраического сложения комплексных чисел такие коды можно рассматривать как простую совокупность кодов действительных чисел и выполнять одноименную операцию с каждой парой действительных чисел независимо. В то же время с такими кодами выполнимы операции умножения и деления. При этом операции умножения и деления состоят, как обычно, из циклов ‘сдвиг-сложение’.

3. Метод кодирования точек многомерного пространства

Метод кодирования точек многомерного евклидова пространства должен устанавливать некоторое соответствие между этими точками и кодами из некоторого множества. Это соответствие, вообще говоря, может быть не взаимно-однозначным. Но для возможности однозначного декодирования каждому коду должна соответствовать только одна точка кодируемого пространства. В то же время даже ограниченная область пространства содержит несчетное множество точек. Следовательно, множество соответствующих кодов также несчетно и среди них должны быть коды с бесконечным числом разрядов (**бесконечные коды**). Однако в практике вычислений могут использоваться только **конечные коды**, а множество конечных кодов ограничено.

Для того, чтобы в этих условиях сохранить соответствие между кодами и точками пространства, естественно ограниченную кодируемую область G разбить на ограниченное множество областей δ определенного размера и конфигурации так, чтобы каждая точка области G находилась в одной из областей δ . Тогда между множеством конечных кодов и множеством областей δ можно установить взаимно-однозначное соответствие.

Такой способ кодирования точек многомерного пространства является приближенным. Действительно, всем точкам $Z_j \in \delta_i$ соответствует единственный код K_i . Однако при декодировании кода K_i образуется единственная точка Z_i . Обозначим радиус-вектор точки Z символом \bar{Z} . Разность $\Delta Z_j = |\bar{Z}_j - \bar{Z}_i|$ определяет абсолютную погрешность кодирования точки Z_j .

В качестве иллюстрации рассмотрим рис. 1, где изображена область Z_j двумерного пространства, разбитая на области δ .

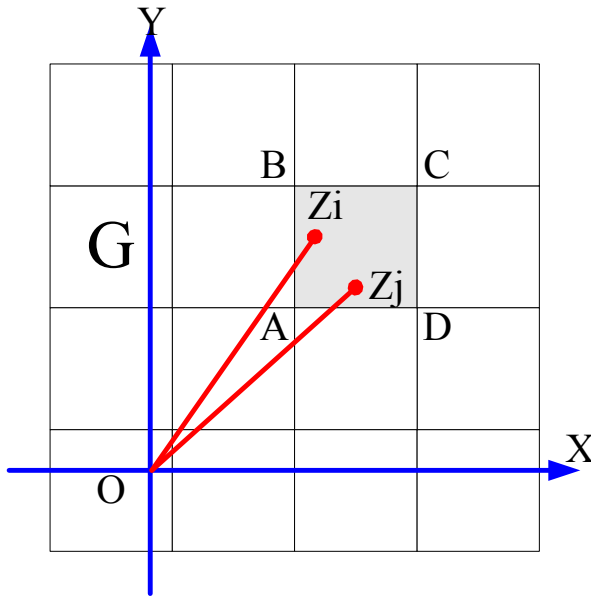


Рис. 1. Кодирование двумерной области

На этом рисунке выделена область $\delta_i = ABCD$, причем области δ_i принадлежит также ее нижняя (AD) и правая (CD) границы. В области δ_i выделена базисная точка Z_i и некоторая точка $Z_j \in \delta_i$. Длина отрезка ΔZ_j характеризует абсолютную погрешность кодирования точки Z_j .

Итак, изложенный принцип кодирования точек многомерного пространства заключается в следующем:

- ограниченная область G кодируемого пространства разделяется на ограниченное множество равных областей δ_i ($i=1, 2, \dots, N$), причем $G = \bigcup \delta_i$ и $\delta_i \cap \delta_j = \emptyset$ при $i \neq j$;
- определяется множество конечных кодов K_i ($i=1, 2, \dots, N$);
- между областями и кодами устанавливается взаимно-однозначное соответствие.

При соблюдении этих условий будем говорить, что **система кодирования** области G многомерного пространства удовлетворяет **принципу кодирования** и область G кодируется с **дискретностью** δ . Следующие две леммы очевидны.

Лемма 1. Система кодирования области G удовлетворяет принципу кодирования, если $V=NU$, и обратно, где

U - объем области δ ,

V - объем области G ,

N - мощность множества конечных кодов.

Лемма 2. Система кодирования области G , удовлетворяющая принципу кодирования, является **полной** (то есть любой точке соответствует конечный код), **неизбыточной** (то есть каждой точке соответствует единственный конечный код) и **приближенной** (то есть подмножеству точек-векторов, модуль разности которых не превышает некоторой величины, соответствует один конечный код).

Рассмотрим множество n -разрядных кодов вида

$$K = \alpha_{n-1} \dots \alpha_k \dots \alpha_1 \alpha_0, \quad (3)$$

где α_k - цифра, принимающая одно из R_k значений, причем $R_k > 1$ и целое число.

Лемма 3. Если система кодирования удовлетворяет принципу кодирования, то при увеличении разрядности конечных кодов и сохранении дискретности кодирования объем кодируемой области увеличивается также, как мощность множества конечных кодов, и обратно.

Доказательство. Мощность множества конечных кодов

$$N_n = \prod_{k=1}^n R_k. \quad (4)$$

Пусть это множество кодов удовлетворяет принципу кодирования и кодирует область G_n с дискретностью δ . В соответствии с леммой 1 количество областей δ , содержащихся в области G_n , также равно N_n , а область G_n имеет объем

$$V_n = N_n U. \quad (5)$$

Увеличим теперь разрядность кодов на единицу, то-есть добавим разряд α_n , принимающий одно из R_n значений. Очевидно

$$N_{n+1} = R_n N_n. \quad (6)$$

Пусть новое множество кодов также удовлетворяет принципу кодирования и кодирует область G_{n+1} с той же дискретностью δ . Количество областей δ , содержащихся в области G_{n+1} , равно N_{n+1} , то-есть область G_{n+1} имеет объем

$$V_{n+1} = N_{n+1}U. \quad (7)$$

Совмещая три последние формулы, находим, что

$$V_{n+1} = R_{n+1}V_n, \quad (8)$$

то-есть прямая часть леммы доказана.

По условию обратной части леммы справедливы формулы (5), (6), (8). Из них следует (7), откуда в соответствии с леммой 1 получаем доказательство обратной части данной леммы.

Рассмотрим теперь позиционную систему кодирования. В этой системе каждому позиционному коду

$$K(Z) = \alpha_n \dots \alpha_k \dots \alpha_m$$

соответствует точка Z кодируемого многомерного пространства, имеющая разложение следующего вида:

$$Z = \sum_{k=m}^n \alpha_k \rho^k, \quad (9)$$

где ρ - основание кодирования,

k - номер разряда,

α_k - k -ый разряд кода (цифра или количественный эквивалент, сопоставляемый ей в разложении), принимающий одно из R_k значений.

Заметим, что ρ и α_k также являются точками кодируемого многомерного пространства. Позиционный код называется **бесконечным**, если $m = -\infty$, и **конечным**, если m ограничено. Число n называется **длиной** позиционного кода. Если $R_k = R$, то разложение и код называются R -ми. Итак, будем рассматривать величины α , принимающие значение из множества

$$A_R = \{a_0, a_1, a_2, \dots, a_j, \dots, a_{R-1}\}, \quad (10)$$

содержащего R различных величин a_j . В практике позиционного кодирования существенно то, что R ограничено и не превышает нескольких единиц.

Позиционный код точки Z по основанию ρ будем обозначать и записывать также следующим образом

$$\langle Z \rangle_\rho = \alpha_n \dots \alpha_k \dots \alpha_1 \alpha_0, \alpha_{-1} \alpha_{-2} \dots \alpha_m, \quad (11)$$

размещая запятую между нулевым и (-1)-разрядом (индекс - основание не будет указываться, если значение основания ясно из контекста). Вектор (точка) Z , в коде которого $m \geq 0$, будем называть

ρ -целым. Соответственно определяются ρ -дробные (правильные и неправильные) векторы Z . В частности,

$$\langle \rho \rangle_{\rho} = 10. \quad (12)$$

Совокупность $\langle \rho, A_R \rangle$ основания кодирования ρ и множества

A_R будем называть **системой позиционного кодирования**. Будем говорить, что точка многомерного евклидова пространства **представима** в данной системе позиционного кодирования, если ей соответствует разложение вида (9) и позиционный код вида (11), в котором разряды принимают значения из множества (10).

Задача заключается в построении таких позиционных систем кодирования, в которых представима любая точка данного пространства и при этом выполняются условия полноты, неизбыточности и приближенности, определенные в лемме 2.

Смысл построения позиционных систем заключается в упрощении выполнения арифметических операций с точками (векторами) многомерного пространства. С другой стороны, существование позиционных кодов, основанных на разложении (9), возможно лишь в том случае, если в данном пространстве определены операции суммирования векторов и умножения вектора на основание ρ (которое также может быть вектором).

В одномерном и двумерном пространствах умножение на основание ρ (умножение на действительное или комплексное число) соответствует увеличению модуля вектора-множимого в $|\rho|$ раз, то-есть

$$\text{если } Z_2 = Z_1 \rho, \text{ то } |Z_2| = |Z_1| \cdot |\rho|. \quad (13)$$

Следует еще раз отметить, что умножению $Z_1 \rho$ соответствует сдвиг кода $\langle Z_1 \rangle$ на один разряд влево *в любом пространстве*. Мы потребуем, чтобы условие (13) выполнялось также *для любого кодируемого пространства* и докажем некоторое условие существования позиционной системы счисления, используя эти два факта.

Теорема 1. Необходимым и достаточным условием того, что любая точка h -мерного евклидова пространства, в котором выполняется условие (13), представима в данной системе позиционного кодирования, является условие

$$|\rho|^h = R. \quad (14)$$

Доказательство. Каждый код $\langle Z_2 \rangle_\rho$ длины $(n+h)$ при $m = -\infty$ может быть получен сдвигом на h разрядов влево некоторого кода $\langle Z_1 \rangle_\rho$ длины n . Но в соответствии с (12) такой сдвиг эквивалентен умножению на основание, то-есть $Z_2 = Z_1 \rho^h$. При этом из соотношения (13) следует, что $|Z_2| = |Z_1| \cdot |\rho|^h$. Следовательно, линейные размеры кодируемой области увеличиваются в $|\rho|^h$ раз (кроме того, кодируемая область, вообще говоря, поворачивается относительно предыдущего положения). Таким образом, объемы областей G_n и G_{n+h} связаны соотношением

$$V_{n+h} = |\rho|^h V_n. \quad (15)$$

Очевидно, ограничение m не изменяет объем кодируемой области. Появляется лишь дискретность кодирования $\delta = G_{m-1}$. Учитывая (14), из (15) получаем

$$V_{n+h} = R V_n. \quad (16)$$

Сравнивая (16) и (8), из леммы 3 находим, что система позиционного кодирования при $m < -\infty$ удовлетворяет принципу кодирования, то есть, вследствие леммы 2, является полной, избыточной и приближенной. Теорема доказана.

4. Арифметические системы кодирования

Среди позиционных систем кодирования наибольший интерес представляют такие, к которым применимы простые алгоритмы сложения и умножения. Именно такие системы мы и будем рассматривать в дальнейшем, но предварительно определим их более строго.

Определение 1. Система $\langle \rho, A_R \rangle$ позиционного кодирования называется **арифметической**, если выполнены следующие условия

- число (-1) является ρ -целым,
- сумма и произведение любых пар векторов, принадлежащих множеству A_R , являются ρ -целыми.

Заметим, что условие (13) может выполняться и для неарифметической системы.

Лемма 4. Если в арифметической позиционной системе представимы векторы Z_1 и Z_2 , то в этой системе представимы и векторы $-Z_1, -Z_2, Z_1 + Z_2, Z_1 \cdot Z_2$.

Справедливость леммы вытекает из того, что, как будет показано ниже, для арифметических позиционных систем существуют алгоритмы арифметических операций.

Определение 2. Позиционная система $\langle \rho, A_R \rangle$ называется **нормальной**, если $A_R = B_R$, где $B_R = \{0, 1, 2, \dots, R-1\}$.

Лемма 5. Нормальная система счисления, в которой

$$R = \sum_{k=1}^n \alpha_k \rho^k, \quad (17)$$

$$-R = \sum_{k=1}^w \beta_k \rho^k, \quad (18)$$

то есть коды чисел R и $-R$ являются ρ -целыми и имеют нулевое значение нулевого разряда, является арифметической.

Доказательство. Любое число из множества B_R находится в пределах $0 \leq a_j \leq (R-1)$. Следовательно, для чисел из этого множества выполняются соотношения $-a_j = a_k - R$ и

$a_j + a_k = a_m + R$, если $a_j + a_k \geq R$. Учитывая условия леммы, заключаем, что числа $(-a_j)$ и $(a_j + a_k)$ являются ρ -целыми. Очевидно, произведение $a_j a_k$ можно представить суммой чисел из множества B_R . По индукции в силу существования алгоритма сложения заключаем, что такая сумма также является ρ -целой. Таким образом, условия определения 1 выполняются. Следовательно, рассматриваемая система является арифметической.

Лемма 6. Нормальная система счисления, в которой число R имеет разложение вида (17) и

$$R = \sum_{k=1}^m \alpha_k, \tag{19}$$

является арифметической.

Доказательство. Как следует из (17) и (19), в лемме рассматриваются системы, в которых

$$R = \sum_{k=1}^n \alpha_k \rho^k = \sum_{k=1}^n \alpha_k \cdot$$

Рассмотрим следующий алгоритм:

α_3	α_2	α_1	0				переносы
	α_3	α_2	α_1	0			переносы
		α_3	α_2	α_1	0		переносы
			α_3	α_2	α_1	$0 < R >_\rho$	слагаемое 1
				β_2	β_1	$0 < X >_\rho$	слагаемое 2
0	0	0	0	0	0	0	сумма

Здесь код числа R складывается с кодом некоторого числа X , разряды которого образуются таким образом, чтобы

$$\alpha_1 + \beta_1 = R \text{ и } \alpha_1 + \alpha_2 + \beta_2 = R.$$

При этом и вследствие (19) сложение цифр каждого столбца образует число R , которое формирует перенос и нулевой разряд суммы. В результате образуются бесконечные переносы и нулевая сумма. Следовательно, $X=-R$. Очевидно, такой алгоритм образования кода числа $-R$ осуществим при любом R , соответствующем разложению (17) или, что одно и то же, при любом коде числа R вида

$$< R >_\rho = \alpha_m \dots \alpha_2 \alpha_1 0.$$

Результатом этого алгоритма является код числа $-R$ вида

$$\langle -R \rangle_{\rho} = \beta_w \dots \beta_2 \beta_1 0.$$

Этот код соответствует разложению (18). Тем самым лемма доказана.

Заметим, что разложения (17) и (18) можно рассматривать как систему двух степенных уравнений относительно неизвестного ρ .

Решая ее, можно, вообще говоря, определить некоторую систему кодирования. Однако такой прием далеко не всегда приводит к положительному результату потому, что данная система либо не разрешима аналитически, либо не совместима, либо дает в качестве решения результат, не удовлетворяющий условию теоремы 1, либо дает в качестве решения действительное число.

Леммы 4, 5, 6 будут использованы далее при поиске нормальных позиционных систем кодирования.

5. Коды действительных чисел

Для действительных чисел размерность кодируемого пространства $h=1$. Следовательно, для позиционных кодов действительных чисел необходимо соблюдать условие $|\rho| = R$.

Широко известны и распространены позиционные коды действительных чисел, в которых $\rho = R$ и разряды принимают значения из множества B_R . Сюда относятся обычные десятичные ($R=10$) и двоичные ($R=2$) коды. Однако такими кодами нельзя изобразить отрицательные действительные числа, для представления которых приходится применять искусственные приемы, в частности, использовать обратные и дополнительные коды, что вызывает ряд неудобств.

В тоже время существуют два способа построения позиционных кодов, пригодных для изображения действительных - положительных и отрицательных чисел. Первый из них заключается в том, что разрядам придают положительные и отрицательные значения из множества

$$D_R = \{-r_1, -r_1 + 1, \dots, -1, 0, 1, \dots, r_2 - 1, r_2\}$$

причем $R = r_1 + r_2 + 1$, $r_1 \neq 0$, $r_2 \neq 0$, а основание, по-прежнему, оставляют равным R (при $r_1 = 0$ множество D_R превращается в множество B_R). Другой способ основан на применении отрицательного основания $\rho = -R$. Величины разрядов при этом могут принимать значения либо из множества B_R , либо из множества D_R . Итак, известные результаты, относящиеся к позиционному кодированию действительных чисел, формулируются следующим образом.

Теорема 2. Любое действительное число представимо в системах

$$\langle R, B_R \rangle, \langle R, D_R \rangle, \langle -R, B_R \rangle, \langle -R, D_R \rangle.$$

Итак, существует четыре системы кодирования действительных чисел:

система $\langle R, B_R \rangle$, например $\langle 5, \{0, 1, 2, 3, 4\} \rangle$;

система $\langle R, D_R \rangle$, например $\langle 5, \{-2, -1, 0, 1, 2\} \rangle$;

система $\langle -R, B_R \rangle$, например $\langle -5, \{0, 1, 2, 3, 4\} \rangle$;

система $\langle -R, D_R \rangle$, например $\langle -5, \{-2, -1, 0, 1, 2\} \rangle$.

Приведем примеры пятиричных кодов некоторых чисел в рассмотренных системах, обозначая величины -1 и -2 цифрами $\bar{1}$ и $\bar{2}$:

1. $K(16) = +31$, $K(-13) = -23$,
2. $K(16) = 1\bar{2}1$, $K(-13) = \bar{1}22$,
3. $K(16) = 121$, $K(-13) = 32$,
4. $K(16) = 121$, $K(-13) = \bar{1}\bar{2}2$.

Здесь следует обратить внимание на то, что коды в первой из этих систем сопровождаются знаками '+' и '-', которые отсутствуют во всех остальных системах, поскольку в них знак числа вместе с модулем определяются значениями разрядов кода.

Важно отметить, что среди указанных систем существуют лишь две системы двоичного кодирования, а именно системы с цифрами $\{0, 1\}$ и основаниями '2' и '-2'.

6. Коды комплексных чисел

Докажем вначале несколько теорем существования нормальных арифметических систем кодирования с комплексным основанием, обозначая через j мнимую единицу.

Теорема 3. Любое комплексное число представимо в нормальной системе кодирования по комплексному основанию ρ и эта система является арифметической, если

$$|\rho| = \sqrt{R} \quad (20)$$

и выполняются условия (17), (19).

Доказательство. Для комплексных чисел размерность кодируемого пространства $h=2$ и при любом ρ выполняется условие (13). Отсюда и из (20) следует, что выполняются условия теоремы 1. Следовательно, любое комплексное число представимо в данной системе кодирования. Далее, условия (17) и (19) являются условиями леммы 6. Следовательно, данная система является арифметической.

Теорема 3 дает возможность свести доказательство теорем о представимости любого комплексного числа в нормальной системе кодирования и арифметичности этой системы к доказательству того, что выполняется условие (19) и ρ является комплексным корнем уравнения (17). Именно этим методом доказательства мы и воспользуемся в дальнейшем.

Теорема 4. Любое комплексное число представимо в нормальной системе кодирования по комплексному основанию

$$\langle \rho = \sqrt{2}e^{\pm j\pi/2}; B_2 \rangle \text{ или } \langle \rho = -1 \pm j; \{0, 1\} \rangle$$

и эта система является арифметической.

Доказательство. Предположим, что $\langle 2 \rangle_\rho = 1100$, Это условие равносильно уравнению $\rho^3 + \rho^2 = 2$. Его решение совпадает с условием данной теоремы. Следовательно, выполняется условие (17). Очевидно, что условие (19) также выполняется, поскольку $R=2$. В силу теоремы 3 данная теорема доказана. Для иллюстрации запишем коды некоторых характерных чисел в системе с основанием $\rho = (j-1)$, обозначив через $\bar{\rho}$ число, сопряженное числу ρ : $K(2) = 1100$, $K(-2) = 11100$, $K(-1) = 11101$, $K(j) = 11$, $K(-j) = 111$, $K(\bar{\rho}) = 110$.

Теорема 5. Любое комплексное число представимо в нормальной системе кодирования по комплексному основанию ρ и эта система является арифметической, если

$\rho = \sqrt{R}e^{j\varphi}$, $\varphi = \pm \arccos(-\beta / 2\sqrt{R})$, $\beta < (R, 2\sqrt{R})_{\min}$
и β - целое положительное число.

Доказательство. Предположим, что $\langle R \rangle_{\rho} = 1\alpha_2\alpha_10$, где $1 + \alpha_2 + \alpha_1 = R$, $\alpha_2 = \beta - 1$. Это условие равносильно уравнению

$$\rho^3 + (\beta - 1)\rho^2 + (R - \beta)\rho = R.$$

Его решение дает значение, приведенное в условиях теоремы. В силу теоремы 3 данная теорема доказана.

Для иллюстрации запишем коды некоторых характерных чисел в этой системе, обозначив через $\bar{\rho}$ число, сопряженное числу ρ :

$$K(R) = 1 (\beta - 1) (R - \beta) 0,$$

$$K(-R) = 1 \beta 0,$$

$$K(-1) = 1 \beta (R - 1),$$

$$K(\bar{\rho}) = 1 (\beta - 1) (R - \beta),$$

$$K(-\bar{\rho}) = 1 \beta,$$

$$K(-\rho) = 1 \beta (R - 1) 0,$$

$$K(\rho - \bar{\rho}) = 2 \beta,$$

$$K(\rho + \bar{\rho}) = 1 \beta (R - \beta).$$

В связи с тем, что β может принимать несколько значений при постоянном R , существует несколько типов позиционных кодов в системах рассматриваемого вида. В качестве примера в табл. 1 приведены возможные коды числа R при различных R и β .

Таблица 1. Коды числа R

$R \setminus \beta$	1	2	3	4	5
2	1010				
3	1020	1110			
4	1030	1120	1210		
5	1040	1130	1220	1310	
6	1050	1140	1230	1320	
7	1060	1150	1240	1330	1420
8	1070	1160	1250	1340	1430
9	1080	1170	1260	1350	1440

Для иллюстрации запишем коды некоторых характерных чисел в системе с основанием $\rho = \frac{1}{2}(-1 + j\sqrt{7})$, обозначив через $\bar{\rho}$ число, сопряженное числу ρ : $K(2)=1010$, $K(-2)=110$, $K(-1)=111$, $K(\bar{\rho})=101$, $K(-\rho)=1110$, $K(-\bar{\rho})=11$, $K(j\sqrt{7})=10101$, $K(-j\sqrt{7})=1110011$.

Из систем теоремы 5 можно выделить группы с фиксированным значением аргумента основания, например

$$\varphi = \pm 2\pi/3, \text{ если } \beta = \sqrt{R}, \text{ то-есть при } R=4, 9, 16, 25, \dots;$$

$$\varphi = \pm 3\pi/4, \text{ если } \beta = \sqrt{2R}, \text{ то-есть при } R=8, 18, 32, 50, \dots;$$

$$\varphi = \pm 5\pi/6, \text{ если } \beta = \sqrt{3R}, \text{ то-есть при } R=12, 27, 48, 75, \dots$$

Рассмотрим теперь позиционную систему более общего вида.

Теорема 6. Любое комплексное число представимо в системе кодирования $\langle \rho = 2e^{j\pi/3}, A_4 \rangle$, $A_4 = \{0, 1, e^{2j\pi/3}, e^{-2j\pi/3}\}$ и эта система является арифметической.

Доказательство. Заметим, что $(-2)^k = l_k \rho^k$, где

$$l_k = \left\{ 1, e^{2j\pi/3}, e^{-2j\pi/3} \right\}$$

соответственно при $k = (3m, 3m+1, 3m+2)$, где m - целое. Очевидно, $l_k \in A_4$. Следовательно, любая степень числа '-2' представима в указанной системе кодирования одним разрядом. В силу теоремы 2

любое действительное число X представимо в виде разложения по основанию $'-2'$. Но каждый разряд такого разложения, представляющий степень числа $'-2'$ или 0 , может быть заменен разрядом разложения в указанной системе кодирования, то есть любое действительное число представимо в этой системе кодирования.

Любое комплексное число Z может быть представлено как $Z = u_1 + u_2 e^{2j\pi/3} + u_3 e^{-2j\pi/3}$, где u_1, u_2, u_3 - некоторые действительные числа. В этой сумме все составляющие представимы в указанной системе счисления поскольку сомножители действительных чисел u_1, u_2, u_3 принадлежат множеству A_4 .

Если эта система является арифметической, то в ней представима и данная сумма, то есть любое комплексное число. Остается показать, что указанная система является арифметической. Для этого составим таблицы попарного умножения, суммирования и таблицу инвертирования (умножения на $'-1'$) цифр из множества A_4 - см. таблицы 2, 3, 4. Для удобства записи эти цифры обозначены символами $0, 1, c, d$. Как видно из этих таблиц, в рассматриваемой системе кодирования выполнены все условия определения 1. Следовательно, эта система является арифметической, что и требовалось показать.

Таблица 2. Одноразрядное умножение

*	0	1	c	d
0	0	0	0	0
1	0	1	c	d
c	0	c	d	1
d	0	d	1	c

Таблица 3. Одноразрядное сложение

+	0	1	c	d
0	0	1	c	d
1	1	dc0	1d	dc
c	c	1d	d10	c1
d	d	dc	c1	c10

Таблица 4. Инвертирование разряда

x	0	1	c	d
-x	0	c1	dc	1d

Заметим, что в этой системе очень простой вид имеют коды комплексных чисел вида e^{jk60° , где k - целое число - см. табл. 4а. Кроме того, для этой системы в табл. 4в представлены коды чисел 2^k и $(-2)^k$, где k - целое число.

Таблица 4а. Коды чисел e^{jk60° .

φ	0	60	120	180	240	300
код	00	1d	0c	c1	0d	dc

Таблица 4в. Коды чисел 2^k и $(-2)^k$.

k	$(-2)^k$	2^k
-4	0.000d	0.000d
-3	0.001	0.0c1
-2	0.0c	0.0c
-1	0.d	1.d
0	1	1
1	c0	dc0
2	d00	d00
3	1000	c1000
4	c000	c000

Далее мы только более строго изложим результаты, полученные в разделе 2.

Теорема 7. Любое комплексное число Z представимо в позиционной системе счисления $\langle \rho = -R, A_{R^2} \rangle$, где множество A_{R^h} состоит из комплексных чисел $r_m = \alpha_m^1 + j\alpha_m^2$, а числа $\alpha_m \in B_R$.

В частности, существует система $\langle -2, \{0, 1, j, 1+j\} \rangle$.

Теорема 8. Любое комплексное число Z представимо в нормальной позиционной системе $\langle \pm j\sqrt{R}, B_R \rangle$.

Например, существует система $\langle \pm j\sqrt{2}, \{0,1\} \rangle$. Для иллюстрации запишем коды некоторых характерных чисел в системе $\rho = j\sqrt{2}$: $K(2)=10100$, $K(-2)=100$, $K(-1)=101$, $K(j\sqrt{2})=10$, $K(-j\sqrt{2})=1010$.

Очевидно, для систем из теорем 7 и 8 выполняется условие (14), Доказательство этих теорем основано на рассуждениях раздела 2.

Приведем для иллюстрации и сравнения двоичные коды чисел во всех указанных системах кодирования, включая системы кодирования по действительному (положительному и отрицательному) и комплексному основаниям - см. табл. 5.

Таблица 5. Двоичные системы кодирования.

Выделенная система счисления	ρ	$\langle 2 \rangle$	$\langle -2 \rangle$	$\langle -1 \rangle$	Теорема	Рис.
Система 1	ρ_2	10100	100	101	Формула (2)	1
Система 2	$j\sqrt{2}$	10100	100	101	Теорема 8	2
Система 3	$-1 + j$	1100	11100	11101	Теорема 4	3
Система 4	$\frac{1}{2}(-1 + j\sqrt{7})$	1010	110	111	Теорема 5	4
	-2	110	10	11	Теорема 2	
	2	10			Теорема 2	

В дальнейшем мы более подробно остановимся на четырех двоичных системах счисления комплексных чисел – см. столбец «Выделенная система счисления» в табл. 5. На рисунках изображены первые 4 значения базовой функции для выделенных систем счисления.

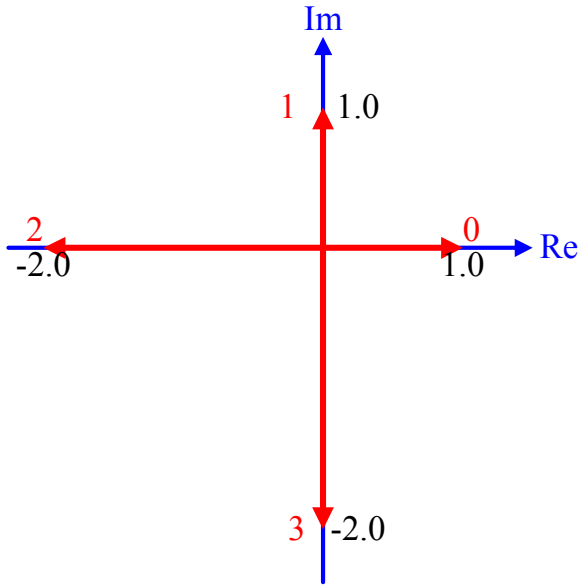


Рис. 1

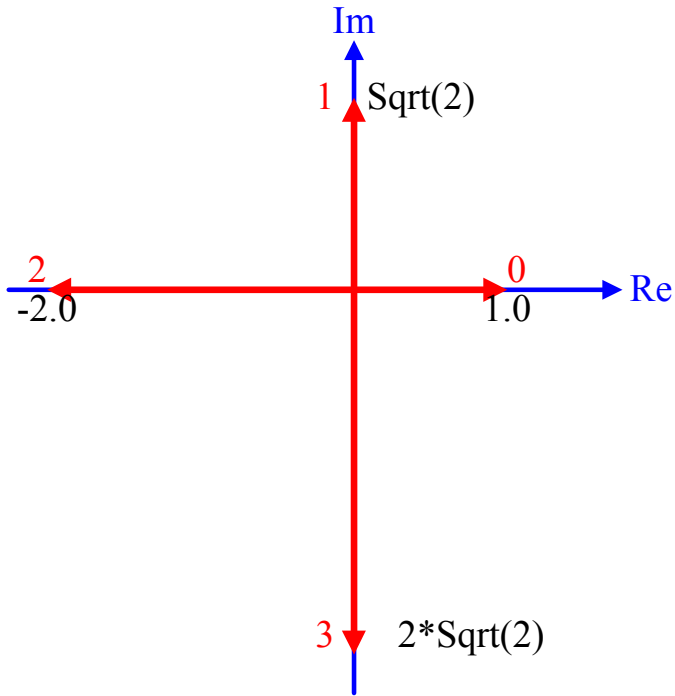


Рис. 2.

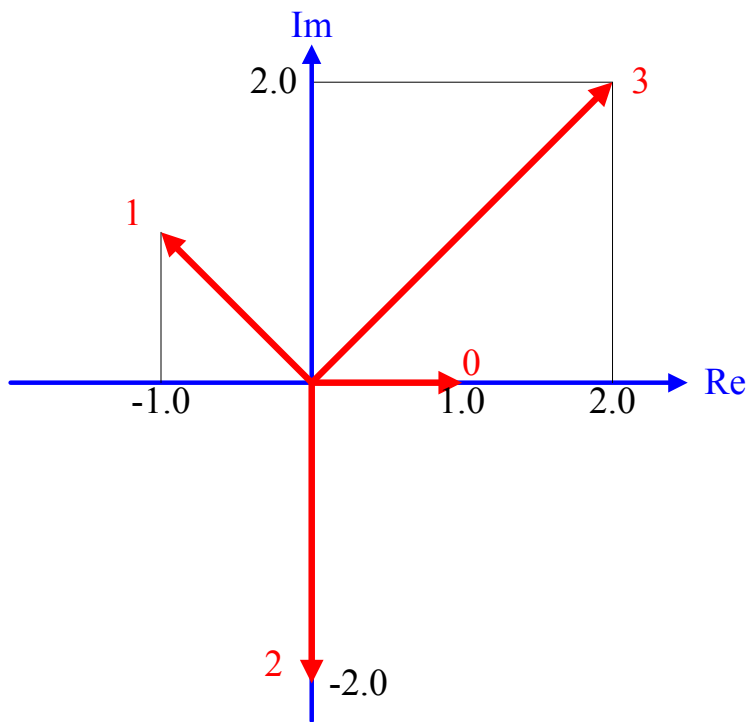


Рис. 3.

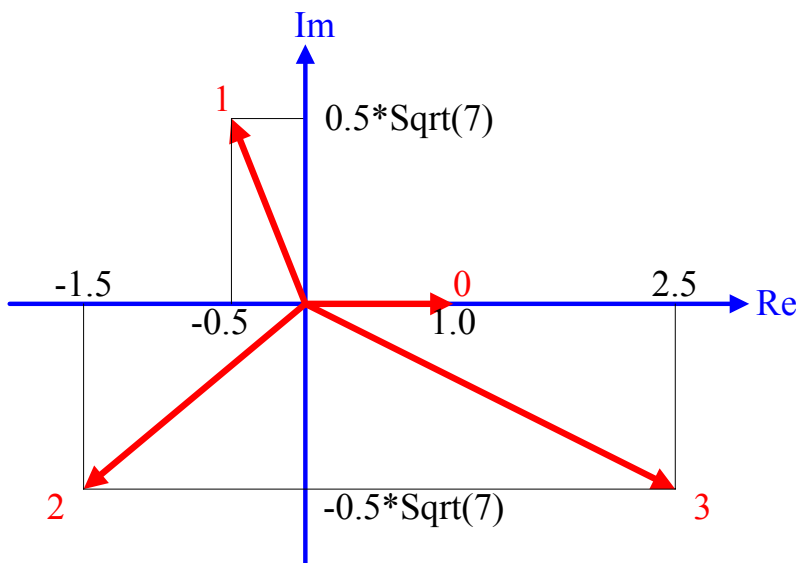


Рис. 4.

Приведем еще табл. 6 четверичных кодов чисел '4' и '-4' во всех рассмотренных выше системах кодирования (в этой таблице '-1' обозначена 'цифрой' 'h').

Таблица 6. Четверичные системы кодирования.

ρ	A_4	$\langle 4 \rangle$	$\langle -4 \rangle$	Теорема
4	{0,1,2,3}	10		2
4	{-1,0,1,2}	10	h0	2
4	{-2,-1,0,1}	10	h0	2
-4	{0,1,2,3}	130	10	2
-4	{-1,0,1,2}	h0	10	2
-4	{-2,-1,0,1}	h0	10	2
$2e^{2j\pi/3}$	{0,1,2,3}	1120	120	5
$2e^{j\pi/3}$	{0,1,c,d}	d00	1d00	6
-2	{0,1,j,1+j}	100	1100	7
ρ_4	{0,1,2,3}	10300	100	8
$\pm 2j$	{0,1,2,3}	10300	100	8

7. Коды многомерных векторов

7.1. Двоичные коды векторов- способ 1.

Изложенный в разделе 2 метод построения кодов комплексных чисел может быть обобщен и использован для кодирования многомерных векторов. Для этого рассмотрим множество действительных чисел $\{X_i\}$, каждое из которых задано двоичным разложением по основанию $\rho = -2$, то есть

$$X_i = \sum_{(m)} \alpha_m^i \rho^m \quad (i=1, 2, \dots, n).$$

Каждому такому разложению соответствует код

$$K(X_i) = \dots \alpha_m^i \dots$$

Рассмотрим теперь n -мерный вектор

$$Z = E_1 X_1 + E_2 X_2 + \dots + E_n X_n, \quad (21)$$

где $\{E_i\}$ - база n -мерного векторного пространства. Множество кодов $\{K(X_i)\}$ можно при этом трактовать как единый код вектора Z по основанию '-2'. Каждый m -ый разряд этого кода изображается множеством $\{\alpha_m^i\}$ двоичных разрядов. Обозначив эти множества цифрами σ_m , получаем код вектора

$$K(Z) = \dots \sigma_m \dots,$$

соответствующий разложению (21), где вектор

$$r_m = E_1 \alpha_m^1 + E_2 \alpha_m^2 + \dots + E_i \alpha_m^i + \dots + E_n \alpha_m^n \quad (22)$$

изображается цифрой σ_m .

В частности, при $n=2$ образуются коды комплексных чисел по основанию ρ_2 , рассмотренные выше – см. формулу (2). При $n=3$ образуются коды трехмерных векторов, в которых разряды принимают одно из восьми значений:

$$r_m \in \{0, \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{i+j}, \mathbf{i+k}, \mathbf{j+k}, \mathbf{i+j+k}\}, \quad (23)$$

где $\mathbf{i}, \mathbf{j}, \mathbf{k}$ - орты прямоугольных координатных осей. Аналогично предыдущему для кодирования трехмерных векторов может быть введена векторная функция действительного целого аргумента m :

$$\mathcal{G}_2 = \left\{ \begin{array}{l} i(-2)^m \text{ if } m = 3k \\ j(-2)^{m-1} \text{ if } m = 3k + 1 \\ k(-2)^{m-2} \text{ if } m = 3k + 2 \end{array} \right\}, \quad (24)$$

При этом рассматриваемый код трехмерного вектора по основанию \mathcal{G}_2 с векторными значениями разрядов (23) может рассматриваться как код трехмерного вектора по основанию \mathcal{G}_2 с двоичными разрядами. Этому коду соответствует разложение вектора в виде $Z = \sum_m (\alpha_m \mathcal{G}_2)$.

Аналогично, для кодирования n -мерных векторов может быть введена векторная функция действительного целого аргумента m :

$$\mathcal{G}_2^n = \left\{ \begin{array}{l} i(-2)^m \text{ if } m = nk \\ j(-2)^{m-1} \text{ if } m = nk + 1 \\ \dots \\ k(-2)^{m-n+1} \text{ if } m = nk + n - 1 \end{array} \right\}, \quad (25)$$

Очевидно, $\rho_2 = \mathcal{G}_2^2$, $\mathcal{G}_2 = \mathcal{G}_2^3$.

7.2. Двоичные коды векторов- способ 2.

Построим теперь, как ранее для комплексных чисел, последовательность чередующихся двоичных разрядов α_m^i :

$$\dots \alpha_{m+1}^2 \alpha_{m+1}^1 \alpha_m^n \alpha_m^{n-1} \dots \alpha_m^2 \alpha_m^1 \alpha_{m-1}^n \alpha_{m-1}^{n-1} \dots$$

В других обозначениях эта последовательность является двоичным кодом

$$K(Z) = \dots \alpha_k \dots$$

некоторого вектора Z . При этом основание кодирования также является вектором

$$\rho = E_2 \sqrt[n]{2}, \quad (26)$$

где E_2 - второй орт базы $\{E_i\}$ n -мерного векторного пространства. Закодированный вектор Z определяется в данном случае по формуле

$$Z = X_1 + \rho X_2 + \dots + \rho^{i-1} X_i + \dots + \rho^{n-1} X_n. \quad (27)$$

7.3. Мнозначные коды векторов- способ 2.

Совершенно аналогично строятся позиционные коды векторов (в том числе комплексных чисел и многомерных векторов) на основе объединения позиционных кодов чисел-проекции векторов по основанию $\rho = -R$, где R - целое число. В этом случае, например, вместо функции ρ_2 в качестве основания кодирования комплексных чисел должна рассматриваться функция

$$\rho_R = \left\{ \begin{array}{l} (-R)^{m/2} \text{ if } m - \text{even} \\ j(-R)^{m-1/2} \text{ if } m - \text{odd} \end{array} \right\}, \quad (28)$$

вместо функции ρ_2 в качестве основания кодирования трехмерных векторов должна рассматриваться функция

$$\rho_R = \left\{ \begin{array}{l} i(-R)^{m/3} \text{ if } m = 3k \\ j(-R)^{m-1/3} \text{ if } m = 3k + 1 \\ k(-R)^{m-2/3} \text{ if } m = 3k + 2 \end{array} \right\}, \quad (29)$$

и, вообще, вместо функции ρ_2^n для кодирования n -мерных векторов должна рассматриваться функция

$$\rho_R^n = \left\{ \begin{array}{l} i(-R)^m \text{ if } m = nk \\ j(-R)^{m-1} \text{ if } m = nk + 1 \\ \dots \\ k(-R)^{m-n+1} \text{ if } m = nk + n - 1 \end{array} \right\}. \quad (30)$$

Итак, справедливы следующие теоремы.

Теорема 7а. Если в n -мерном евклидовом пространстве с базой $\{E_i\}$ определена алгебра, то любая точка Z этого пространства представима в позиционной системе счисления $\langle \rho = -R, A_{R^n} \rangle$, где множество A состоит из векторов (22), а числа $\alpha_m \in B_R$.

В частности, для комплексных чисел существует система $\langle \rho = -R, A_{R^2} \rangle$, например, четверичная система $\langle 2, \{0, 1, j, 1+j\} \rangle$, а для трехмерных векторов с ортами $\mathbf{i}, \mathbf{j}, \mathbf{k}$ – восьмиричная система, где каждый разряд принимает значения (23).

Теорема 8а. Если в n -мерном евклидовом пространстве с базой $\{E_i\}$ определена алгебра, то любая точка Z представима в нормальной позиционной системе

$$\langle \rho = \pm E_2 \sqrt[n]{R}, B_R \rangle. \quad (31)$$

В частности, при $R=2$ имеем двоичную систему кодирования векторов по основанию (26). Для комплексных чисел существует система $\langle \pm j\sqrt{R}, B_R \rangle$, например, двоичная система $\langle \pm j\sqrt{2}, \{0, 1\} \rangle$, а для трехмерных векторов с ортами $\mathbf{i}, \mathbf{j}, \mathbf{k}$ – двоичная система $\langle \pm j\sqrt[3]{2}, \{0, 1\} \rangle$. В последней системе имеем:

$$\begin{aligned} \langle i \rangle &= 1, \langle -i \rangle = 1001, \langle 2i \rangle = 1001000, \langle -2i \rangle = 1000; \\ \langle j \rangle &= 10, \langle -j \rangle = 10010, \langle 2j \rangle = 10010000, \langle -2j \rangle = 10000; \\ \langle k \rangle &= 100, \langle -k \rangle = 100100, \langle 2k \rangle = 100100000, \langle -2k \rangle = 100000. \end{aligned}$$

Для трехмерных векторов с ортами $\mathbf{i}, \mathbf{j}, \mathbf{k}$ существует также четверичная система $\langle \pm j\sqrt[4]{4}, \{0, 1, 2, 3\} \rangle$, где

$$\langle 4i \rangle = 1003000 \text{ и } \langle -4i \rangle = 1000.$$

Очевидно, для систем из теорем 7а и 8а выполняется условие (14).

Глава 2. Точность кодирования

1. Область представимых чисел \ 38
2. Область представимых модулей \ 46
3. Погрешности конечных кодов \ 53
4. Коды с плавающей точкой \ 56

1. Область представимых чисел

Рассмотрим числа Z , имеющие в данной системе $\langle \rho; D_1 = \{0, 1, \dots, R-1\} \rangle$ бесконечные разложения вида:

$$Z = \sum_{K=-\infty}^M \alpha_k \rho^k. \quad (1)$$

Эти числа могут быть представлены в виде суммы

$$Z = \sum_{j=0}^{a-1} Z^{(j)}, \quad (2)$$

где $Z^{(j)} = \sum_{m=-\infty}^{\left[\frac{M+j}{a} \right]} \alpha_{am-j} \rho^{am-j}$ или

$$Z^{(j)} = \rho^{-j} \sum_{m=-\infty}^{\left[\frac{M+j}{a} \right]} \alpha_{am-j} \rho^{am}. \quad (3)$$

(здесь $[\lambda]$ - целая часть от λ).

Пусть a - наименьшее целое число ($\neq 0$), при котором $\arg(\rho^a) = 0$. Тогда, очевидно, $\arg(\rho^{am}) = 0$ и $\arg(Z^{(j)}) = \arg(\rho^{-j})$
или

$$\arg(Z^{(j)}) = -j\varphi, \quad (4)$$

где $\varphi = \arg(\rho)$. В частности, при $j=0$ и $\arg(Z^{(j)})=0$, т.е. число $Z^{(0)}$ является действительным положительным. Определим область изменения числа $Z^{(0)}$ при $M=0$. Согласно (3) имеем:

$$Z^{(0)} = \sum_{m=-\infty}^0 \alpha_{am} \rho^{am}. \quad (5)$$

Очевидно, $Z_{\min}^{(0)} = 0$ и $Z_{\max}^{(0)} = \sum_{m=-\infty}^0 (R-1) \rho^{am}$ или $Z_{\max}^{(0)} = (R-1)L$, где

$$L = \frac{\rho^a}{\rho^a - 1}. \quad (6)$$

Таким образом, при $M=0$

$$0 \leq Z^{(0)} \leq (R-1)L. \quad (7)$$

Далее, при $M=0$ имеем: $Z^{(j)} = \rho^{-j} \sum_{m=-\infty}^0 \alpha_{am-j} \rho^{am}$. Область

изменения числа $\sum_{m=-\infty}^0 \alpha_{am-j} \rho^{am}$ совпадает с областью изменения числа $Z^{(0)}$ - см. (5). Следовательно,

$$0 \leq |Z^{(j)}| \leq |\rho|^{-j} (R-1)L. \quad (8)$$

Формулы (8) и (4) определяют область изменения числа $Z^{(j)}$. В геометрической интерпретации эта область является отрезком $OA = |\rho|^{-j} (R-1)L$ прямой, расположенной под углом $(-j\varphi)$ на комплексной плоскости - см. рис. 1. Иначе говоря, областью изменения числа $Z^{(j)}$ при $M=0$ является радиус-вектор

$$r_j = (R-1)L\rho^{-j}. \quad (9)$$

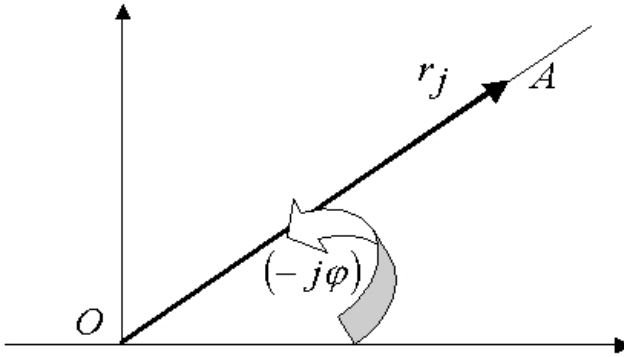


Рис. 1.

Заметим, что

$$r_{j+1} = r_j / \rho. \tag{10}$$

В табл. 1 приведены величины φ , a , L для различных оснований ρ , необходимые для определения векторов r_j . Векторы r_j используются, в свою очередь, для построения области изменения числа Z при $M=0$, которую будем обозначать символом G_0 . В простейшем случае при $\rho=R$, $Z=Z^{(0)}$ область G_0 есть вектор $r_0=R$ (см. (9) и табл. 1.), т.е. отрезок $[0, R]$ действительной оси - см. рис. 2.

Таблица 1.

ρ	R	$-R$	$\sqrt{R} \cdot e^{i\varphi}$			
φ	0	π	$\pm \frac{1}{2}\pi$	$\pm \frac{2}{3}\pi$	$\pm \frac{3}{4}\pi$	$\pm \frac{5}{6}\pi$
a	1	2	4	3	8	12
L	$\frac{R}{R-1}$	$\frac{R^2}{R^2-1}$	$\frac{R^2}{R^2-1}$	$\frac{R\sqrt{R}}{R\sqrt{R}-1}$	$\frac{R^4}{R^4-1}$	$\frac{R^6}{R^6-1}$

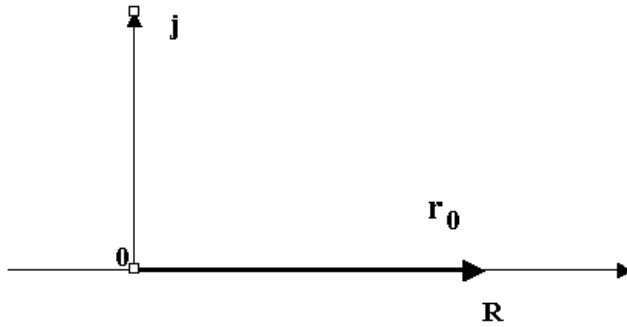


Рис. 2.

При $\rho = -R$ область G_0 для числа $Z = Z^{(0)} + Z^{(1)}$ есть совокупность векторов $r_0 = \frac{R^2}{R+1}$ (см. (9) и табл. 1) и $r_1 = -\frac{R}{R+1}$ (см. (10)), т.е. отрезок $\left[-\frac{R}{R+1}; \frac{R^2}{R+1}\right]$ действительной оси - см. рис. 3.

Для определения области G_0 при $\rho = (\sqrt{R}) \cdot e^{\pm\pi/2}$ необходимо определить предварительно области изменения действительной и мнимой частей числа Z .



Рис. 3.

Очевидно, $\operatorname{Re} Z = Z^{(0)} + Z^{(2)}$ и $\operatorname{Im} Z = Z^{(1)} + Z^{(3)}$. Поэтому область изменения числа $\operatorname{Re} Z$ есть совокупность векторов $r_0 = \frac{R^2}{R+1}$ и $r_2 = -\frac{R}{R+1}$, а область изменения числа $\operatorname{Im} Z$ есть совокупность

векторов $r_1 = \frac{\mu i R \sqrt{R}}{R+1}$ и $r_3 = \frac{\mu i \sqrt{R}}{R+1}$ - здесь векторы

r_0, r_1, r_2, r_3 определены, по-прежнему, при помощи формул (9), (10) и табл. (1). Таким образом, числа $\text{Re}Z$ и $\text{Im}Z$ определены соответственно на отрезке AB действительной оси и на отрезке CD мнимой оси - см. рис. 4. Очевидно, число Z может принимать лишь те значения, для которых $\text{Re}Z$ и $\text{Im}Z$ не выходят из пределов отрезков AB и CD . Следовательно, область G_0 в данном случае представляет собой прямоугольник $EKLM$, стороны которого проходят через точки A, B, C, D перпендикулярно осям.

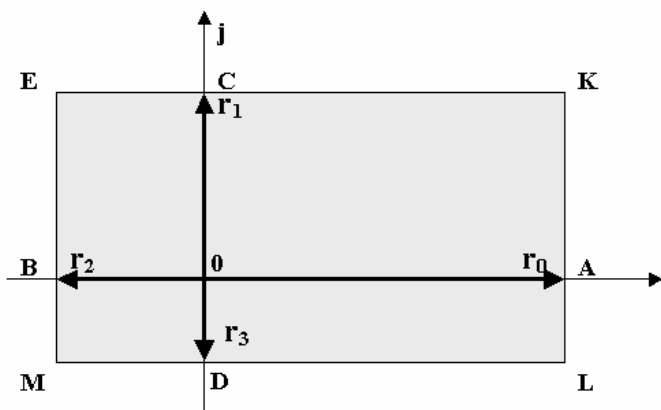


Рис. 4.

Для построения области G_0 при $\rho = (\sqrt{R}) \cdot e^{\pm \frac{2}{3}\pi \cdot i}$, также, как и в предыдущем случае, предварительно вычисляются и строятся векторы r_0, r_1 и r_2 - см. рис. 5. Число Z всегда может быть представлено проекциями на оси OA, OC, OB . При этом оно может принимать лишь те значения, при которых указанные проекции не превышают величины отрезков OA, OC, OB . Поэтому область G_0 представляет собой сумму трех параллелограммов, построенных на отрезках OA, OC, OB .

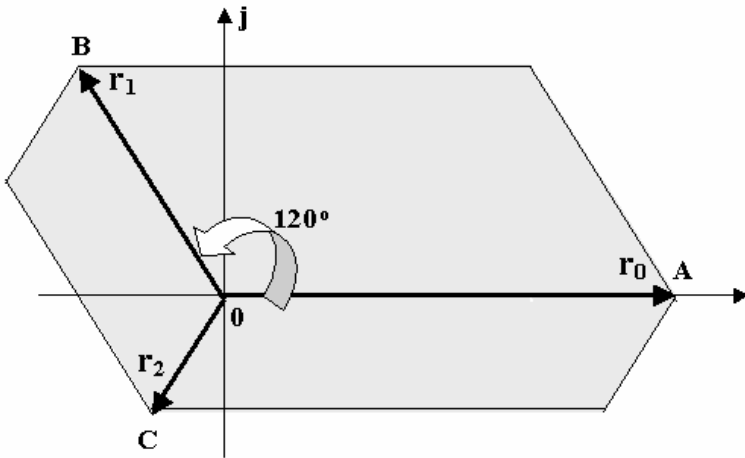


Рис. 5.

Более сложные построения приходится производить для определения области G_0 при основаниях, где j принимает 8 и 12 значений - см. табл. 1. В качестве примера произведем эти построения для $\rho = i - 1$ ($\rho = \sqrt{2} e^{i\varphi}$; $R = 2$; $\varphi = \frac{3}{4}\pi$; $a = 8$; $L = \frac{16}{15}$).

По формулам (9) и (10) находим: $r_0 = \frac{16}{15}$; $r_{j+1} = \frac{r_j}{\sqrt{2}} e^{-i\varphi}$.

Используя полученные формулы, строим векторы r_j на комплексной плоскости - см. рис. 6. Заметим, что модуль вектора r_{j+1} может быть определен как сторона квадрата, диагональю которого является модуль вектора r_j .

Эти векторы определяют две пары взаимно-перпендикулярных осей координат: (15) $_|_$ (37) и (26) $_|_$ (48). В каждой из полученных осей координат построим, аналогично рис. 4, области, определяемые тетрадами векторов: в системе осей (15) и (37) - область G_0^I (прямоугольник ABCD) изменения числа $Z^{(0)} + Z^{(2)} + Z^{(4)} + Z^{(6)}$; в системе осей (26) и (48) - область G_0^{II} (прямоугольник EKLМ) изменения числа $Z^{(1)} + Z^{(3)} + Z^{(5)} + Z^{(7)}$. Очевидно, область G_0 определяется как совокупность областей G_0^I и G_0^{II} .

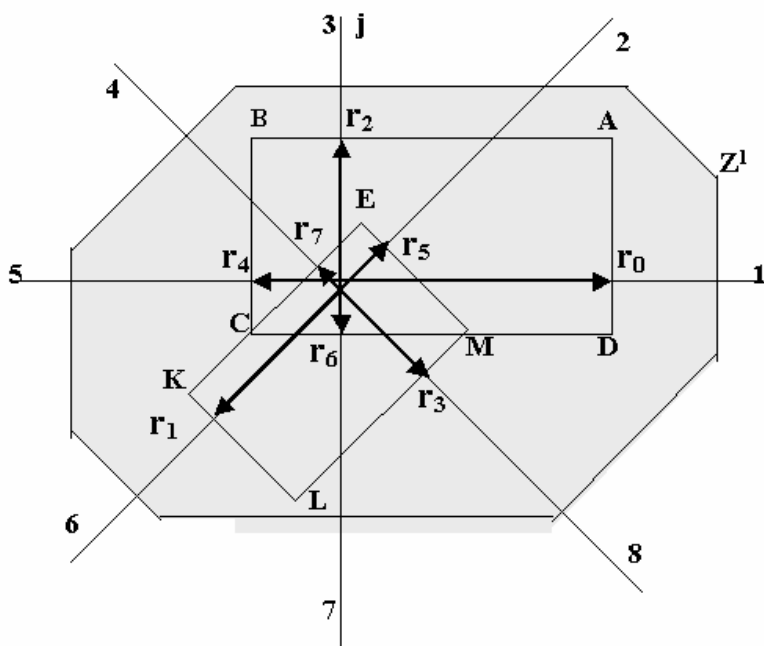


Рис. 6.

Для определения границ области G_0 необходимо точку O , принадлежащую системе координат (26) и (48), поместить на границу области G_0^I и перемещать по этой границе без поворота осей (26) и (48). Площадь, закрываемая прямоугольником $EKLM$ при перемещении, в сумме с площадью $ABCD$ и будет составлять искомую область G_0 . Таким образом, область G_0 принимает вид фигуры, заштрихованной на рис. 6.

Аналогично строится область G_0 при других основаниях, где $a=(8, 12)$, за исключением того, что при $a=12$ область G_0 определяется как совокупность трех прямоугольных областей.

Итак, область G_0 может быть построена для любых ρ и Z при $M=0$. Нетрудно заметить, что при $M \neq 0$ область G_M изменения числа Z может быть получена подобным преобразованием и поворотом области G_0 , причем коэффициент подобия равен $|\rho|^M$, а угол поворота равен $\arg \rho^M$. Построенная таким образом

область G_M является областью определения чисел Z , представимых бесконечными кодами вида:

$$K(Z) = \alpha_M \dots \alpha_K \dots, \quad (11)$$

где M фиксировано. В частности, в эту область входят и числа, имеющие конечные коды, а также коды, у которых ограниченное число старших разрядов равно нулю.

Область G_M позволяет определить область изменения представимого бесконечным кодом числа Z с фиксированным аргументом. Для этого достаточно из начала координат провести луч под углом $\arg Z$. Отрезок этого луча, заключенный в области G_M , и является искомой областью изменения. В частности, *область изменения действительной правильной ρ -ой дроби есть отрезок действительной оси, заключенный в области G_{-1} .*

2. Область представимых модулей

Граница области G_M изображает функцию максимального модуля, представимого кодом (11) числа, имеющего данный аргумент. Эта функция интересна тем, что по ней нетрудно найти число Z' с максимально возможным модулем. Точка, изображающая это число, имеет наибольший радиус-вектор - см., например, рис. 6. Модуль числа Z' при данном ρ является функцией только от M . Поэтому обозначим:

$$|Z'| = \psi(M)$$

Очевидно,

$$\psi(M) = |\rho|^M \cdot \psi(0). \quad (12)$$

В частности, максимальный модуль правильной ρ -ой дроби

$$\psi(-1) = \frac{1}{|\rho|} \cdot \psi(0) \quad (13)$$

Пример 1.

При $\rho = R$ из рис. 2 находим: $\psi(0) = R$; $\psi(-1) = 1$.

При $\rho = -R$ из рис. 3 находим: $\psi(0) = \frac{R^2}{R+1}$; $\psi(-1) = \frac{R}{R+1}$.

При $\rho = \pm i\sqrt{R}$ из рис. 4 находим:

$$\psi(0) = \sqrt{|r_0|^2 + |r_1|^2} = \frac{R\sqrt{R(R+1)}}{R+1}; \quad \psi(-1) = \frac{R\sqrt{R+1}}{R+1}.$$

При $\rho = i - 1$ из рис. 6 находим: $\psi(0) \approx \frac{72}{50} \cdot r_0 = 1,53$; $\psi(-1) \approx 1,1$.

Рассмотрим код, в котором старший разряд $\alpha_M = 1$:

$$K(Z) = 1\alpha_{M-1}\dots\alpha_k\dots$$

Будем называть такой код *приведенным*. Пусть номер старшего разряда приведенного бесконечного кода $M=1$; область Q_1 чисел, представленных таким кодом, образуется из области G_0 смещением

начала координат из точки 0 в точку 0_1 такую, что $\overline{O_1 O} = \rho$ - см. рис. 7. Область G_0 , построенная в осях (11) и (22), является областью Q_1 в осях (33) и (44). Область Q_M чисел, представленных приведенным кодом, в котором $M \neq 1$, образуется аналогично области G_M подобным преобразованием и поворотом области Q_1 . При этом коэффициент подобия равен $|\rho|^{M-1}$, а угол поворота равен $\arg \rho^{M-1}$.

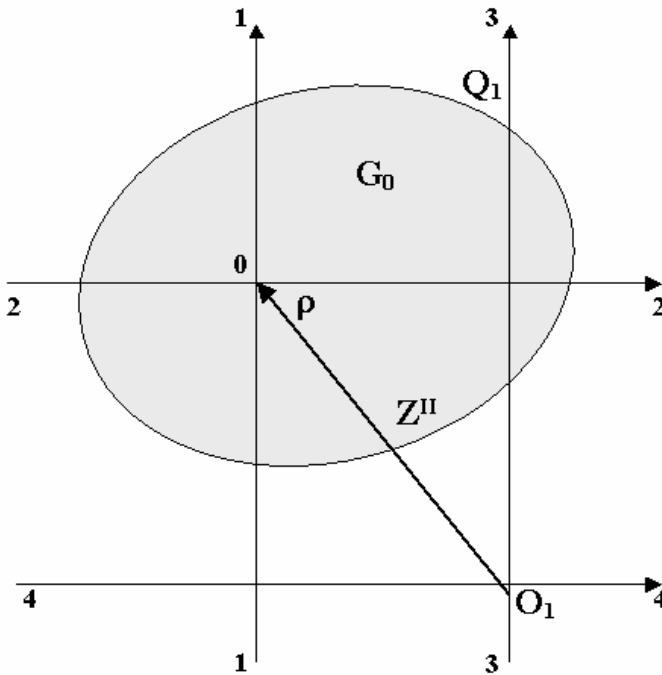


Рис. 7.

В области Q_M нетрудно найти точку, изображающую число Z'' с минимально возможным модулем. Эта точка имеет наименьший радиус-вектор – см. например, рис. 7. Модуль числа Z'' при данном ρ является функцией только от M . Поэтому обозначим:

$$|Z''| = \phi(M).$$

Очевидно,

$$\phi(M) = |\rho|^{M-1} \cdot \phi(1). \quad (14)$$

В частности, минимальный модуль правильной ρ -ой дроби

$$\phi(-1) = \frac{1}{|\rho|^2} \cdot \phi(1) \quad (15)$$

Пример 2.

Производя построения, аналогичные рис. 7, находим:

$$\text{при } \rho = R: \phi(1) = R; \phi(-1) = \frac{1}{R}$$

$$\text{при } \rho = -R: \phi(1) = \frac{R}{R+1}; \phi(-1) = \frac{1}{R(R+1)}$$

$$\text{при } \rho = \pm i\sqrt{R}: \phi(1) = \frac{\sqrt{R}}{R+1}; \phi(-1) = \frac{1}{\sqrt{R}(R+1)}$$

$$\text{при } \rho = i-1: \phi(1) \approx 0,28; \phi(-1) \approx 0,14$$

Рассмотрим код, в котором старший разряд $\alpha_M \neq 0$:

$$K(Z_M) = \alpha_M \dots \alpha_K \dots$$

Будем называть такой код *нормализованным*. Заметим, что частным случаем нормализованного кода является приведенный код, а для двоичных кодов эти понятия совпадают. Модуль числа Z_M представленного нормализованным кодом, заключен в пределах:

$$\phi(M) \leq |Z_M| \leq \psi(M) \quad (16)$$

Последняя формула показывает область изменения модуля числа, представленного кодом с фиксированным номером старшего значащего разряда. Будем обозначать его как НСЗР или символом Ω (в формулах). Из формулы (16) следует, что между НСЗР и модулем $|Z_M|$ в общем случае не существует однозначной зависимости. Область изменения модуля $|Z_M|$ изображена на рис. 8 заштрихованной полосой, ограниченной функциями $\phi(M)$ и $\psi(M)$. Эти функции являются показательными – см. рис. (12) и (14).

Область изменения модуля при конкретном значении НСЗР = M_1 , изображается здесь вертикальным отрезком ab .

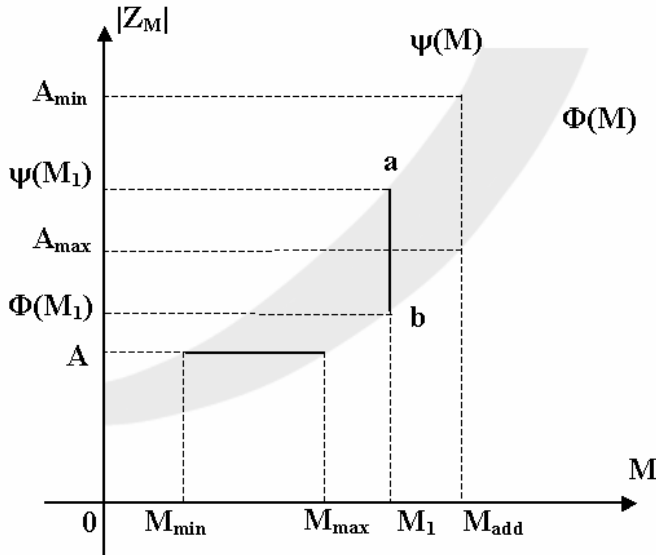


Рис. 8.

Вид функций $\psi(M)$ и $\phi(M)$ для двоичных кодов определяется в следующем примере.

Пример 3.

Используя результаты, полученные в примерах 1 и 2, и формулы (12) и (14) для двоичных кодов находим:

$$\text{при } \rho = 2: \quad 2^M \leq |Z_M| \leq 2 \cdot 2^M;$$

$$\text{при } \rho = -2: \quad \frac{1}{3} \cdot 2^M \leq |Z_M| \leq \frac{4}{3} \cdot 2^M;$$

$$\text{при } \rho = i\sqrt{2}: \quad \frac{1}{3} \cdot 2^{\frac{M}{2}} \leq |Z_M| \leq \frac{2\sqrt{6}}{3} \cdot 2^{\frac{M}{2}};$$

$$\text{при } \rho = i-1: \quad 0,2 \cdot 2^{\frac{M}{2}} \leq |Z_M| \leq 1,53 \cdot 2^{\frac{M}{2}}.$$

Числа, имеющие равный модуль, в общем случае могут быть представлены кодами с разными НСЗР. Найдем пределы изменения кодов чисел с данным модулем A - см. рис. 8. Обозначим:

M_{\min} - минимальный НСЗР кода числа с данным модулем A ;

M_{\max} - максимальный НСЗР кода числа с данным модулем A .

Имеем:

$$\psi(M_{\min}) \leq A \leq \phi(M_{\max}). \quad (17)$$

Точное равенство в общем случае не выполняется в связи с тем, что величина M и, следовательно, функции $\psi(M)$ и $\phi(M)$ принимают дискретные значения. Из формулы (17) нетрудно определить M_{\min} и M_{\max} как предельные значения аргумента M в функциях $\psi(M)$ и $\phi(M)$, при которых неравенства еще выполняются.

Пример 4.

Определим пределы изменения НСЗР (Ω) двоичных кодов чисел с модулем $A=2^C$. Пользуясь результатами примера 3, находим:

- при $\rho = 2$: $2 \cdot 2^{M_{\min}} \leq 2^C$, т.е. $M_{\min} \leq C-1$,
 $2^{M_{\max}} \geq 2^C$, т.е. $M_{\max} \geq C$; таким образом, $C-1 \leq \Omega \leq C$;
- при $\rho = -2$: $\frac{4}{3} \cdot 2^{M_{\min}} \leq 2^C$, т.е. $M_{\min} \leq C$, $\frac{1}{3} \cdot 2^{M_{\max}} \geq 2^C$,
 т.е. $M_{\max} \geq C+2$; таким образом, $C \leq \Omega \leq C+2$;
- при $\rho = i\sqrt{2}$: $\frac{2\sqrt{6}}{3} \cdot 2^{\frac{1}{2}M_{\min}} \leq 2^C$, т.е. $M_{\min} \leq 2C-1$,
 $\frac{1}{3} \cdot 2^{\frac{1}{2}M_{\max}} \geq 2^C$, т.е. $M_{\max} \geq 2C+4$, таким образом,
 $2C-1 \leq \Omega \leq 2C+4$;
- при $\rho = i-1$: $1,53 \cdot 2^{\frac{1}{2}M_{\min}} \leq 2^C$, т.е. $M_{\min} \leq 2C-1$,
 $0,2 \cdot 2^{\frac{1}{2}M_{\max}} \geq 2^C$, т.е. $M_{\max} \geq 2C+4$; таким образом,
 $2C-1 \leq \Omega \leq 2C+4$.

Полагая $A = \psi(M_{\min}) = \phi(M_{\max})$, нетрудно найти

$$M_{\max} - M_{\min} = 1 + \lg_{|\rho|} \frac{\psi(0)}{\phi(1)}.$$

Из этой формулы следует, что разность $M_{\max} - M_{\min}$ не зависит от величины модуля A . В частности, это означает, что горизонтальные отрезки, заключенные в заштрихованной области рис. 1, имеют постоянную длину. Расчет по этой формуле дает результаты, приведенные в примере 4.

Аналогично можно определить максимальный и минимальный НСЗР кода числа, модуль которого заключен в некоторых пределах. Приведенные формулы дают также возможность определить максимальный модуль числа, при котором НСЗР не превышает заданной величины.

Пусть допустимый максимальный НСЗР равен M_{\lim} - см. рис. 8. *Допустимый максимальный модуль* $A_{\lim \max}$ числа, код которого имеет $\Omega \leq M_{\lim}$, определяется из неравенства:

$$A_{\lim \max} \leq \phi(M_{\lim}). \quad (18)$$

Например, допустимый максимальный модуль правильной ρ -ой дроби равен $\phi(-1)$. Однако это не означает, что правильная ρ -ая дробь не может иметь модуль $\geq \phi(-1)$: допустимый максимальный модуль *гарантирует* соблюдение условия $\Omega \leq M_{\lim}$, но *не ограничивает* величину возможного модуля, при котором это условие соблюдается.

Величину $A_{g.\max}$ удобно применять при выборе масштабных коэффициентов. В этом случае выбор может оказаться намного проще, чем при использовании области G_M .

Наконец, *допустимый минимальный модуль* $A_{g.\max}$ числа, код которого имеет $\Omega \geq M_{\lim}$, определяется из неравенства (см. рис. 8):

$$A_{\lim \min} \geq \psi(M_{\lim}) \quad (19)$$

В частности, допустимый минимальный модуль правильной ρ -ой дроби равен $\psi(-1)$. Здесь также, как и в предыдущем случае, допустимый минимальный модуль *гарантирует* соблюдение условия $\Omega \geq M_{\text{lim}}$, но *не ограничивает* величину возможного модуля, при котором это условие соблюдается.

Из (18) и (19) следует, что соблюдение условия $M' \leq \Omega \leq M''$ для кода некоторого числа Z гарантируется в том случае, если

$$\psi(M') \leq |Z| \leq \phi(M''). \quad (20)$$

3. Погрешности конечных кодов

В общем случае для точного представления числа Z необходимо использовать бесконечный код

$$K(Z) = \alpha_M \dots \alpha_{N+1} \alpha_N \dots \quad (21)$$

Однако практически всегда достаточно иметь приближенное значение \tilde{Z} числа Z , отличающееся от него на величину погрешности Δ :

$$\tilde{Z} = Z - \Delta. \quad (22)$$

Пусть \tilde{Z} имеет конечный код

$$K(\tilde{Z}) = \alpha_M \dots \alpha_{N+1}; \quad (23)$$

тогда $K(\Delta) = \alpha_N \dots$ (24)

По определению, абсолютная погрешность представления числа Z приближенным значением \tilde{Z} равна модулю погрешности Δ : $|\Delta| = |Z - \tilde{Z}|$. Максимальная абсолютная погрешность $\varepsilon = |\Delta|_{\max}$. Следовательно,

$$\varepsilon = \psi(N). \quad (25)$$

Область изменения числа, представленного конечным кодом (23), отличается от области изменения числа, представленного бесконечным кодом (21) с тем же номером старшего разряда, на область изменения числа Δ , представленного кодом (24). Заметим, что область изменения числа, представленного конечным кодом, следует отличать от диапазона представимых чисел, т.е. от количества различных конечных кодов с данным числом разрядов. Понятия области и диапазона совпадают для целых кодов по основанию $\rho = R$ с нулевым номером младшего разряда.

Относительная погрешность представления числа Z конечным кодом (23) по определению равна отношению $|\Delta| : |Z_M|$, где Z_M - число, представленное нормализованным кодом (21). Это отношение принимает максимальное значение при $|\Delta| = |\Delta|_{\max}$ и $|Z_M| = |Z_M|_{\min}$. Следовательно, максимальная относительная погрешность

$$\delta = \frac{\psi(N)}{\phi(M)}. \quad (26)$$

По этой формуле подсчитывается максимальная относительная погрешность представления числа Z_M нормализованным конечным кодом (23). Для решения обратной задачи, т.е. для определения *длины* (числа разрядов) конечного кода, обеспечивающей необходимую максимальную относительную погрешность, необходимо из формулы (26) выразить длину кода через δ . Совмещая формул (12), (14) и (26), получаем:

$$\delta = \frac{|\rho|^N \cdot \psi(0)}{|\rho|^{M-1} \cdot \phi(1)}. \quad (27)$$

Длина кода

$$l = M - N + 1. \quad (28)$$

Подставляя (28) в (27), находим:

$$\delta = \gamma \cdot |\rho|^{-l} \quad (29)$$

или

$$l = \lg_{|\rho|} \frac{\gamma}{\delta}, \quad (30)$$

где

$$\gamma = \frac{|\rho|^2 \cdot \psi(0)}{\phi(1)}. \quad (31)$$

Пример 5.

Определим коэффициенты γ для различных оснований, пользуясь результатами примеров 1 и 2.

- при $\rho = R$: $\gamma = R^2$; в частности, при $\rho = 2$: $\gamma = 4$.
- при $\rho = -R$: $\gamma = R^3$; в частности, при $\rho = -2$: $\gamma = 8$.
- при $\rho = i\sqrt{R}$: $\gamma = R^2\sqrt{R+1}$; в частности, при $\rho = i\sqrt{2}$: $\gamma = 6,9$.
- при $\rho = i-1$: $\gamma = 11$.

При $|\rho| = R$ имеем:

$$l = \lg_R \gamma - \lg_R \delta. \quad (32)$$

При $|\rho| = \sqrt{R}$ получаем:

$$l = 2[\lg_R \gamma - \lg_R \delta]. \quad (33)$$

Полученные формулы позволяют вычислить длину кода по известной величине δ . Сравнение формул (32) и (33) показывает, что при фиксированном δ длина кода по комплексному основанию вдвое больше длины кода по действительному основанию. Это утверждение не совсем точно, если учесть различие в величинах $\lg_R \gamma$. Однако это различие не превышает единицы (см. пример 5) и ей можно пренебречь. Этому результату и следовало ожидать, поскольку код комплексного числа несет информацию одновременно о действительной и мнимой частях этого числа.

Пример 6.

Пусть $\delta = 2^{-b}$, где b - положительное число. Определим длину двоичных кодов по данному δ для различных оснований.

При $\rho = 2$: $l_1 = b + 2$.

При $\rho = -2$: $l_2 = b + 3$.

При $\rho = i\sqrt{2}$: $l_3 = 2b + 6$.

При $\rho = i - 1$: $l_4 = 2b + 7$.

Следует заметить, что действительная длина двоичного кода по основанию $\rho = 2$ за счет добавления знакового разряда равна $l_1^I = l_1 + 1$. Таким образом, $l_1^I = l_2$ и $l_3 = 2l_2$. Следовательно, применение кодов комплексных чисел удваивает разрядность регистров.

4. Коды с плавающей точкой

Формула (29) выражает максимальную относительную погрешность числа, представленного нормализованным кодом длины l . Очевидно, если старшие разряды $\alpha_M, \alpha_{M-1}, \alpha_{M-2}, \dots$ принимают нулевые значения, то длина кода уменьшается и максимальная относительная погрешность возрастает. Для того, чтобы сохранять длину кода, применяют коды с плавающей точкой. Этот прием можно распространить и на коды комплексных чисел.

Любое комплексное число Z может быть представлено в следующей форме:

$$Z = T \cdot \rho^\pi, \quad (34)$$

где

π - целое действительное число, *экспонента* (порядок) числа Z ;

T - комплексное число, правильная ρ -ая дробь; *мантисса* числа Z .

Итак,

$$K_\rho(T) = 0, \alpha_{-1} \alpha_{-2} \dots \alpha_k \dots \alpha_{-n}. \quad (35)$$

Пусть

$$K_{\rho_1}(\pi) = \beta_{Q-1} \dots \beta_q \dots \beta_0. \quad (36)$$

это Q -разрядный код числа π по действительному основанию $\rho_1 = \pm R$, представляющий R^Q различных значений числа π , заключенного в пределах: $\pi_{\min} \leq \pi \leq \pi_{\max}$, причем

$$\pi_{\max} - \pi_{\min} + 1 = R^Q. \quad (37)$$

Кодом с плавающей точкой числа Z называется выражение

$$\tilde{K}(Z) = \beta_{Q-1} \dots \beta_q \dots \beta_0; \alpha_{-1} \alpha_{-2} \dots \alpha_k \dots \alpha_{-n}. \quad (38)$$

Код $\tilde{K}(Z)$ называется *нормализованным*, если $\alpha_{-1} \neq 0$. Максимальная относительная погрешность нормализованного кода с плавающей запятой определяется согласно (29):

$$\delta = \gamma \cdot |\rho|^{-n}. \quad (39)$$

Учитывая (18), (19) и (34), находим, что допустимый максимальный модуль числа Z , представленного нормализованным кодом $\tilde{K}(Z)$,

$$|Z|_{\lim \max} = |\rho|^{\tau_{\max}} \cdot \phi(-1), \quad (40)$$

а допустимый минимальный модуль этого же числа

$$|Z|_{\lim \min} = |\rho|^{\tau_{\min}} \cdot \psi(-1). \quad (41)$$

Отношение

$$\lambda = \frac{|Z|_{\lim \max}}{|Z|_{\lim \min}} \quad (42)$$

характеризует область изменения модуля числа Z , представленного нормализованным кодом $\tilde{K}(Z)$ с максимальной относительной погрешностью δ . Из (13), (15), (31), (37), (40), (41), (42) находим:

$$\lambda = \frac{1}{\gamma} \cdot |\rho|^{RQ}. \quad (43)$$

Определим для сравнения коэффициент λ^I для обычного кода, имеющего ту же разрядность, что и код $\tilde{K}(Z)$. Пусть

$$K(Z^I) = \alpha_{Q-1} \dots \alpha_q \dots \alpha_0, \alpha_{-1} \alpha_{-2} \dots \alpha_{-n}. \quad (44)$$

Для того, чтобы максимальная относительная погрешность этого кода определялась формулой (39), необходимо, чтобы его НСЗР находился в пределах: $-1 \leq \text{НСЗР} \leq Q-1$. Тогда согласно (20)

находим $\psi(-1) \leq |Z'| \leq \phi(Q-1)$, откуда получаем $\lambda' = \frac{\phi(Q-1)}{\psi(-1)}$ или, согласно (13), (14), (31),

$$\lambda' = \frac{1}{\gamma} \cdot |\rho|^{Q+1}. \quad (45)$$

Очевидно, коэффициент λ' не зависит от положения точки в коде (44).

Отношение коэффициентов λ и λ' характеризует эффективность применения кодов с плавающей точкой вместо

кодов с фиксированной точкой при заданной максимальной относительной погрешности:

$$\lambda : \lambda' = |\rho|^{R^Q - Q - 1}. \quad (46)$$

Пример 7

Определим эффективность применения плавающей точки для двоичных 40-разрядных кодов при $\rho = i\sqrt{2}$ и $\rho_1 = -2$. Здесь $\gamma = 6,9$ - см. пример 5.

1. Пусть $\delta = 2^{-15}$. Из примера 6 находим $n = 36$. Далее имеем:

$$Q = 4; \quad \lambda = 0,15 \cdot 2^{2^3}; \quad \lambda' = 0,15 \cdot 2^{2^{2,5}}. \quad \text{Следовательно,}$$

$$\text{отношение } \lambda : \lambda' = 2^{5,5}.$$

2. Пусть $\delta = 2^{-14}$. Тогда $n = 34$; $Q = 6$; $\lambda = 0,15 \cdot 2^{2^5}$; $\lambda' = 0,15 \cdot 2^{3,5}$ и $\lambda : \lambda' = 2^{28,5}$.

3. Пусть $\delta = 2^{-13}$. Тогда $n = 32$; $Q = 8$; $\lambda = 0,15 \cdot 2^{2^7}$; $\lambda' = 0,15 \cdot 2^{4,5}$ и $\lambda : \lambda' = 2^{123,5}$.

4. Пусть $\delta = 2^{-12}$. Тогда $n = 30$; $Q = 10$; $\lambda = 0,15 \cdot 2^{2^9}$; $\lambda' = 0,15 \cdot 2^{5,5}$ и $\lambda : \lambda' = 2^{506,5}$.

Таким образом, эффективность применения плавающей точки резко возрастает с уменьшением допустимой относительной погрешности.

Пример 8. Точность кодирования по основанию $\rho = j\sqrt{2}$.

Рассмотрим пределы изменения нормализованного числа с 10-разрядной экспонентой. Пределы изменения экспоненты и мантиссы таковы:

- экспонента (степень числа «-2») – от (-682) до (341)
- величина положительной экспоненты – $(4^{-341}) \div (4^{170})$,

- величина отрицательной экспоненты –

$$\left(-2 \cdot 4^{-340}\right) \div \left(-2 \cdot 4^{170}\right),$$
- действительная часть мантиссы – от $\left(-\frac{2}{3}\right) \approx (-0.6667)$ до

$$\left(\frac{1}{3}\right) \approx (0.3333),$$
- мнимая часть мантиссы – от $\left(-\frac{2\sqrt{2}}{3}\right) \approx (-0.9428)$ до

$$\left(\frac{\sqrt{2}}{3}\right) \approx (0.4714).$$

Пример 9. Точность кодирования по основанию ρ_2 :

Рассмотрим пределы изменения мантиссы комплексного числа при кодирования по основанию ρ_2 . Код нормализованной мантиссы X имеет вид

$$0.xu*** \dots\dots,$$

где по крайней мере один из разрядов x, u имеет значение «1». Величина соответствующей части мантиссы (x относится к Im , u относится к Re) находится в пределах

$$-\frac{2}{3} < \bullet < -\frac{1}{6}.$$

Величина другой (a , следовательно, любой) части мантиссы находится в пределах

$$-\frac{2}{3} < \bullet < \frac{1}{3}.$$

Пределы изменения мантиссы комплексного числа таковы:

- действительная или мнимая нормализованная часть мантиссы – от $\left(-\frac{2}{3}\right) \approx (-0.6667)$ до $\left(-\frac{1}{6}\right) \approx (-0.16667),$

- действительная или мнимая ненормализованная часть мантиссы – от $\left(-\frac{2}{3}\right) \approx (-0.6667)$ до $\left(\frac{1}{3}\right) \approx (0.3333)$.

Глава 3. Поразрядные арифметические операции

1. Поразрядные операции \ 61
2. Первый алгоритм поразрядных операций \ 63
3. Второй алгоритм поразрядных операций \ 70
4. Поразрядные операции при отрицательном основании \ 81

1. Поразрядные операции

Пусть Z_1 и Z_2 - некоторые векторы,

$$Z = Z_1 \Delta Z_2 \quad (1)$$

- результат операции Δ с ними, а позиционные коды этих векторов в арифметической системе $\langle \rho, A_R \rangle$ имеют вид $K(Z_1) = \dots \alpha_k \dots$, $K(Z_2) = \dots \beta_k \dots$, $K(Z) = \dots \sigma_k \dots$

Рассмотрим класс поразрядных операций, то есть операций сложения, вычитания, умножения на постоянный коэффициент и, в том числе, инвертирования, то есть умножения на $'-1'$. Поразрядная Δ операция производится последовательно над каждой парой чисел α_k и β_k (k -ых разрядов исходных кодов) с учетом переноса из младших разрядов по формуле

$$S_k = Q_k + P_k, \quad (2)$$

где

$Q_k = \alpha_k \Delta \beta_k$ - результат операции Δ над k -ми разрядами,

P_k - перенос из младших разрядов в k -ый разряд,

S_k - разрядный результат.

Разрядный результат всегда можно представить в виде

$$S_k = \sigma_k + \rho P_{k+1}, \quad (3)$$

где

σ_k - k -ый разряд результирующего кода,

P_{k+1} - перенос из k -го разряда в $(k+1)$ -ый разряд.

Очевидно, для того, чтобы не возникало переноса из старших разрядов в младшие, значение P_{k+1} должно быть ρ -целым при любом k . В частности, P_k должно быть ρ -целым. Отсюда следует, что значения S_k и Q_k также должны быть ρ -целыми. Для соблюдения всех этих условий необходимо и достаточно, чтобы лишь значение Q_k было ρ -целым.

Таким образом, операция с позиционными кодами является поразрядной, если выполняются два условия:

- $(\alpha_k \Delta \beta_k) \rho^k = \alpha_k \rho^k \Delta \beta_k \rho^k$,
- Q_k - ρ -целое.

Из определения 1.1 следует, что в арифметической системе операции инвертирования, сложения и умножения на число из множества A_R являются поразрядными операциями. Следовательно, операции алгебраического сложения и умножения на ρ -целый множитель в арифметической системе являются поразрядными операциями.

Метод выполнения поразрядной операции и сложность ее реализации зависит, главным образом, от метода и сложности вычисления значений переноса из младших разрядов в старшие. Например, при большой разрядности ρ -целого множителя задача вычисления переносов практически неразрешима. При алгебраическом сложении сложность вычисления переносов зависит, в основном, от величины R и числа разрядов в кодах чисел ' R ' и ' $-R$ '.

Существует два способа определения значений переноса и поэтому поразрядная операция может выполняться по двум различным алгоритмам.

2. Первый алгоритм поразрядных операций

Пусть $\langle S_k \rangle = \gamma_m \dots \gamma_j \dots \gamma_1 \gamma_0$, где γ_j - цифры из множества A_R . Тогда, согласно (3),

$$\sigma_k = \gamma_0, \quad (4)$$

$$\langle P_{k+1} \rangle = \mu_{m-1} \dots \mu_j \dots \mu_0, \quad (5)$$

где $\mu_{j-1} = \gamma_j$. С использованием полученных соотношений составляем последовательность элементарных операций для вычислений в k -ом разряде:

Алгоритм 1. Известны α_k, β_k, P_k .

1. Определяется $Q_k = \alpha_k \Delta \beta_k$.
2. Определяется S_k по (2).
3. Определяется код $\langle S_k \rangle$.
4. Определяется σ_k по (4).
5. Определяется код $\langle P_{k+1} \rangle$ вида (5).
6. Определяется P_{k+1} по коду $\langle P_{k+1} \rangle$.
7. Выполняются операции 1-6 для $(k+1)$ -го разряда.

Рис. 1 иллюстрирует схему вычислений в k -разряде, а рис. 2 иллюстрирует схему распространения переносов по алгоритму 1.

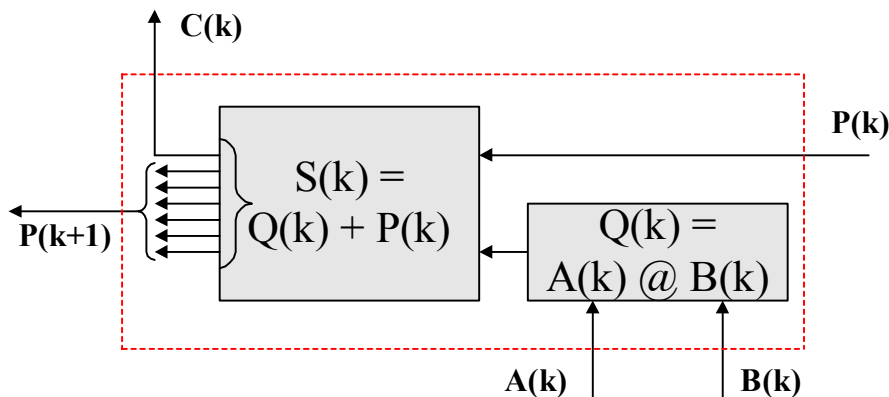


Рис. 1. Схема вычислений в k -разряде (к алгоритму 1)

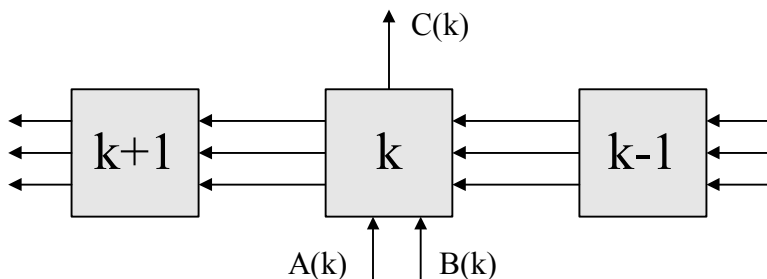


Рис. 2. Схема распространения переносов (к алгоритму 1)

Полученная форма записи алгоритма не дает наглядного представления о его сложности и особенностях и неудобна для практического применения. Поэтому она используется только для получения другой формы записи алгоритма - построения таблицы, описывающей поразрядную операцию в одном разряде. Эта таблица является законченной только в том случае, если удовлетворяет двум следующим условиям:

- если перенос P_{k+1} принимает некоторое значение, то и перенос P_k принимает такое же значение;
- в таблице присутствует любая возможная комбинация значений α_k, β_k, P_k .

Эти условия в дальнейшем будем называть **условиями полноты** таблицы поразрядных операций. Рассмотрим некоторые примеры синтеза таблиц поразрядных операций по алгоритму 1 и вычислений с помощью полученных таблиц.

Пример 1. Инвертирование кодов в системе $\langle \rho = -1 \pm j; \{0, 1\} \rangle$ (см. теорему 1.4). Имеется код $K(Z) = \dots\alpha_k\dots$. Необходимо определить код $K(-Z) = \dots\sigma_k\dots$

Очевидно, $Q_k = \alpha_k$. Применяя алгоритм 1, строим таблицу 1, удовлетворяющую условиям полноты: переносы P_k и P_{k+1} принимают значения из общего пятиэлементного множества $P = \{0, 1, j, -1, 1+j\}$ и в таблице присутствуют все возможные пары чисел, одно из которых принадлежит множеству $\{0, 1\}$, а другое - множеству P .

Таблица 1. Инвертирование кодов.

α_k	P_k	Q_k	S_k	$\langle P_{k+1} \rangle$	σ_k	P_{k+1}
0	0	0	0	0	0	0
1	0	-1	-1	1110	1	1+j
0	1+j	0	1+j	111	0	-j
1	1+j	-1	j	1	1	1
0	-j	0	-j	11	1	j
1	-j	-1	-1-j	11	0	j
0	j	0	j	1	1	1
1	j	-1	-1+j	1	0	1
0	1	0	1	0	1	0
1	1	-1	0	1110	1	1+j

Применяя таблицу 1, легко выполнить инвертирование любого кода данной системы. Приведем два примера:

переносы	1	j	-j	1+j	
данный код			1	0	1
результат	1	1	0	0	1

переносы	1	1+j	0	0	1	j	-j	1+j	
данный код	1	1	1	0	0	1	0	0	1
результат		1	1	0	1	0	1	0	1

Пример 2. Сложение кодов в системе $\langle \rho = \sqrt{2}e^{j\varphi}, \{0, 1\} \rangle$, где $\cos\varphi = -1/2\sqrt{2}$ (см. теорему 1.5). В этом случае $Q_k = \alpha_k + \beta_k$ принимает три значения: $\{0, 1, 2\}$. Применяя алгоритм 1, строим таблицу 2, описывающую процесс сложения. В ней для сокращения объема

- не записаны комбинации разрядов α_k и β_k , дающие в сумме Q_k ;
- величина S_k не приводится, а записывается только код $\langle S_k \rangle$;
- код $\langle S_k \rangle$ записывается в клетке, находящейся на пересечении столбца Q_k и строки P_k (например, при $Q_k = 1$ и $P_k = \rho$ код $\langle S_k = 1 + \rho \rangle = 101$).

Эта таблица удовлетворяет условиям полноты, так как в ней присутствуют все возможные комбинации значений Q_k и P_k , а код $\langle P_k \rangle$ принимает те же значения, что и код $\langle P_{k+1} \rangle$, состоящий из старших разрядов кода $\langle S_k \rangle$ (например, при $\langle S_k \rangle = 1110$ код $\langle P_{k+1} \rangle = 111$, но код $\langle P_k \rangle = 111$ также присутствует в таблице).

Таблица 2. Сложение кодов.

P_k	$\langle P_k \rangle$	код $\langle S_k \rangle$ при		
		$Q_k=0$	$Q_k=1$	$Q_k=2$
0	0	0	1	1010
1	1	1	1010	1011
ρ	10	10	11	11100
$-\bar{\rho}$	11	11	11100	11101
ρ^2	100	100	101	1110
$\bar{\rho}$	101	101	1110	1111
-1	111	111	0	1
$-\rho$	1110	1110	1111	1000

Приведем два примера использования построенной таблицы 2:

переносы	-1	$\bar{\rho}$	0	-1	$\bar{\rho}$	0	
слагаемое 1	1	0	1	1	1	1	1
слагаемое 2		1	1	0	1	1	0
сумма					1	0	1

переносы				1	$-\bar{\rho}$	-1	$-\rho$	ρ	$\bar{\rho}$	
слагаемое 1			1	0	0	0	0	1	0	1
слагаемое 2		1	0	0	0	0	1	1	0	1
сумма	1	1	1	1	1	1	1	0	1	0

Пример 3. Вычитание кодов в системе $\langle \rho = \sqrt{2}e^{j\varphi}, \{0, 1\} \rangle$, где $\cos \varphi = -1/2\sqrt{2}$ (см. теорему 1.5). В этом случае $Q_k = \alpha_k - \beta_k$ принимает значения: $\{0, 1, -1\}$. Применяя алгоритм 1, строим таблицу 3, описывающую процесс вычитания.

Таблица 3. Вычитание кодов.

P_k	$\langle P_k \rangle$	код $\langle S_k \rangle$ при		
		$Q_k=0$	$Q_k=-1$	$Q_k=1$
0	0	0	111	1
1	1	1	0	1010
ρ	10	10	111001	11
$-\bar{\rho}$	11	11	10	11100
$-\rho^2$	11100	11100	11	11101
$\bar{\rho}$	101	101	100	1110
-1	111	111	110	0
$-\rho$	1110	1110	101	1111

Пример 4. Инвертирование кодов в системе $\langle \rho = \sqrt{2}e^{j\varphi}, \{0, 1\} \rangle$, где $\text{Cos } \varphi = -1/2\sqrt{2}$ (см. теорему 1.5). В этом случае $Q_k = \alpha_k - \beta_k$ принимает значения: $\{0, -1, -2\}$. Применяя алгоритм 1, строим таблицу 4, описывающую процесс инвертирования. В ней (также, как и в таблице 2) для сокращения объема

- величина S_k не приводится, а записывается только код $\langle S_k \rangle$;
- код $\langle S_k \rangle$ записывается в клетке, находящейся на пересечении столбца Q_k и строки P_k .

Эта таблица удовлетворяет условиям полноты, так как в ней присутствуют все возможные комбинации значений Q_k и P_k , а код $\langle P_k \rangle$ принимает те же значения, что и код $\langle P_{k+1} \rangle$, состоящий из старших разрядов кода $\langle S_k \rangle$.

Таблица 4. Инвертирование кодов.

P_k	$\langle P_k \rangle$	код $\langle S_k \rangle$ при	
		$Q_k=0$	$Q_k=-1$
0	0	0	111
1	1	1	0
$-\bar{\rho}$	11	11	10

Пример 5. Инвертирование кодов в системе $\langle \rho = 2e^{j\pi/3}, A_4 = \{0, 1, c, d\} \rangle$, где $c = e^{2j\pi/3}$, $d = e^{-2j\pi/3}$ (см. теорему 1.6).

Таблица 5. Инвертирование кодов.

α_k	Q_k	(P_{k+1}, σ_k) при P_k , равном						
		0	1	c	d	-1	-c	-d
0	0	0,0	0,1	0,c	0,d	c,1	d,c	1,d
1	-1	c,b	0,0	-d,d	1,c	c,d	0,d	0,c
c	-c	d,c	-c,d	0,0	-1,1	0,d	d,0	0,1
d	-d	1,d	-c,c	-d,1	0,0	0,c	0,1	1,0

По алгоритму 1 с помощью таблиц 1.3 и 1.4 одноразрядного сложения и инвертирования строим таблицу 5, удовлетворяющую условиям полноты. В этой таблице на пересечении строк Q_k и столбцов P_k записывается пара чисел (P_{k+1}, σ_k) . Применяя таблицу 5, выполним инвертирование некоторых кодов данной системы.

переносы	1	-d	c	0	-1	d	
данный код			d	1	d	1	c
результат	1	d	1	1	c	c	c

переносы	c	-1	d	0	d	0	0	c	
данный код	c	0	1	c	d	c	0	0	1
результат		1	c	c	0	c	0	c	1

Пример 6. Алгебраическое сложение кодов в системах с базовыми функциями $f(\rho, m) = \begin{cases} \rho^{m/2} & \text{if } m - \text{even} \\ j \cdot \rho^{(m-1)/2} & \text{if } m - \text{odd} \end{cases}$, $\rho = -2$ и

$f(\rho, m) = \rho^m$, $\rho = \pm j\sqrt{2}$. Заметим, что в этих системах $K(2) = 10100$, $K(-2) = 100$, $K(-1) = 101$. Принимая во внимание метод кодирования в этих системах и вид указанных кодов можно заметить, что алгебраическое сложение в этих системах выполняется по правилам алгебраического сложения в системе кодирования действительных чисел по основанию $\rho = -2$, где $K(2) = 110$, $K(-2) = 10$, $K(-1) = 11$.

При обратном сложении $Q_k = -\alpha_k - \beta_k$ принимает значения: $\{0, -1, -2\}$. Применяя алгоритм 1, строим таблицу 6, описывающую процесс обратного сложения.

Таблица 6. Обратное сложение кодов.

P_k	$\langle P_k \rangle$	код $\langle S_k \rangle$ при		
		$Q_k = 0$	$Q_k = -1$	$Q_k = -2$
0	0	0	11	10
1	1	1	0	110
-1	11	11	10	111

В частности при инвертировании $Q_k = -\alpha_k$ принимает два значения: $\{0, -1\}$. Таблица, описывающая процесс инвертирования, является выделенной частью в таблицы 6.

На основе таблицы 6 построена таблица 6а истинности одноразрядных схем для обратного сложения кодов в данной системе. В ней выделена часть, являющаяся таблицей истинности для инвертирования.

Таблица 6а истинности.

β_k	α_k	Q_k	P_k	S_k	$\langle P_{k+1} \rangle$	σ_k	P_{k+1}
0	0	0	0	0	0	0	0
0	1	-1	0	-1	1	1	1
0	0	0	1	1	0	1	0
0	1	-1	1	0	0	0	0
1	0	-1	0	-1	1	1	1
1	1	-2	0	-2	11	0	-1
1	0	-1	1	0	0	0	0
1	1	-2	1	-1	1	1	1

Рис. 3 иллюстрирует схему распространения переносов по одноразрядным схемам комплексного кода в системах кодирования данного примера.

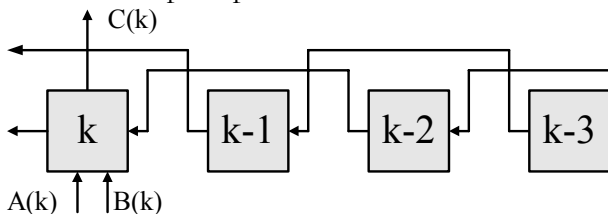


Рис. 3. Схема распространения переносов (к примеру 6).

3. Второй алгоритм поразрядных операций

Будем, в отличие от алгоритма 1, исходить из представления S_k в виде

$$S_k = \rho^m \eta_{k+m} + \dots + \rho^j \eta_{k+j} + \dots + \rho \eta_{k+1} + \sigma_k, \quad (6)$$

где

σ_k - k -ый разряд результирующего кода,

η_{k+j} - **частичный перенос** из k -го разряда в $(k+j)$ -ый,

m - максимальное число частичных переносов.

Очевидно, при такой структуре S_k перенос P_k в k -ый разряд складывается из m частичных переносов μ_{kj} , образующихся в m младших разрядах с номерами от $(k-m)$ до $(k-1)$. Итак,

$$P_k = \sum_{j=1}^m \mu_{kj} \quad (7)$$

Для того, чтобы P_k было β -целым, частичные переносы μ_{kj} также должны быть β -целыми. Других ограничений на числа μ_{kj} и η_{k+j} не накладывается, поэтому они в общем случае могут отличаться от чисел множества A_R .

Назовем выражение вида

$$\langle\langle S_k \rangle\rangle = \eta_{k+m} \cdot \eta_{k+j} \cdot \dots \cdot \eta_{k+1} \sigma_k$$

квазикодом разрядного результата, а выражение вида

$$\langle\langle P_k \rangle\rangle = \eta_{k+m} \cdot \eta_{k+j} \cdot \dots \cdot \eta_{k+1}$$

- **квазикодом** переноса из k -го разряда.

Используя полученные соотношения, составляем последовательность элементарных операций для вычислений в k -ом разряде:

Алгоритм 2. Известны α_k , β_k , μ_{kj} .

1. Определяется $Q_k = \alpha_k \Delta \beta_k$.
2. Определяется P_k по (7).

3. Определяется S_k по (2).
4. Определяется квазикод $\ll S_k \gg$.
5. Определяется σ_k из квазикода $\ll S_k \gg$.
6. Определяется квазикод $\ll P_{k+1} \gg$, то-есть частичные переносы из k -го разряда.
7. Выполняются операции 1-6 для $(k+1)$ -го разряда.

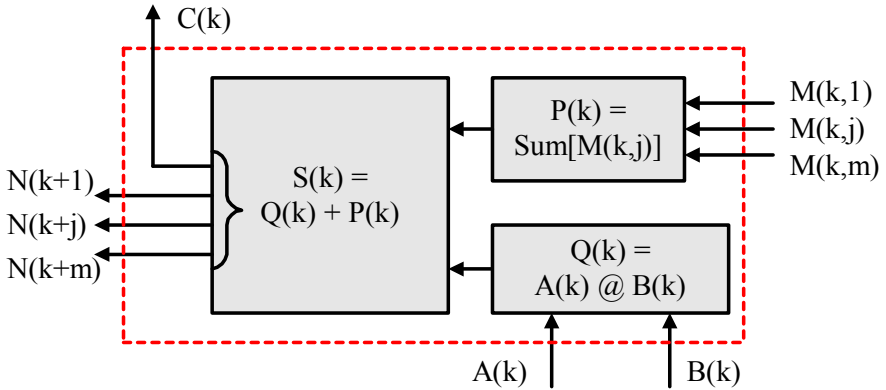


Рис. 4. Схема вычислений в k -разряде (к алгоритму 2)

Рис. 4 иллюстрирует схему вычислений в k -разряде, а рис. 5 иллюстрирует схему распространения переносов по алгоритму 2. На этих рисунках обозначено: $N(k+j) = \eta_{k+j}$, $M(k,j) = \mu_{kj}$.

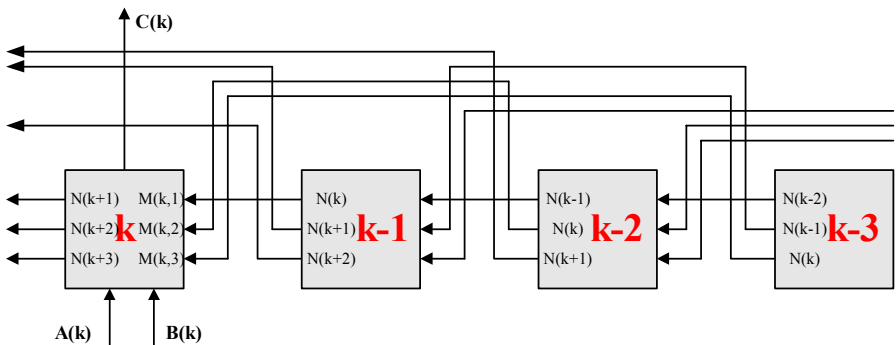


Рис. 5. Схема распространения переносов (к алгоритму 2)

Алгоритм 2 для данной поразрядной операции в данной системе кодирования может иметь несколько вариантов в связи с тем, что существует несколько квазиков для разрядного результата S_k . В частности, квазикод может совпадать с позиционным кодом, если

значения частичных переносов выбраны из множества A_R . В общем случае различные разряды квазикода принимают значения из множеств, отличающихся от A_R не только значениями элементов, но и мощностью (количеством элементов). Например, часть разрядов квазикода может иметь всегда нулевое значение (определяется на пустом множестве), часть разрядов - принимать значения из множества A_R , часть разрядов - из множества $A' \neq A_R$.

Последовательность операций алгоритма 2 используется для синтеза таблиц поразрядных операций, удовлетворяющих условиям полноты, которые в данном случае формулируются несколько иначе:

- перенос P_k принимает все возможные значения, соответствующие любой комбинации η_{k+j} ,
- в таблице присутствует любая возможная комбинация чисел α_k, β_k, P_k .

Очевидно, согласно алгоритму 2, имеющему несколько вариантов, может быть построено несколько типов таблиц для данной поразрядной операции в данной системе кодирования. Рассмотрим некоторые примеры синтеза таблиц поразрядных операций согласно алгоритму 2 и вычислений с помощью полученных таблиц.

Пример 7. Инвертирование кодов в системе $\langle \rho = 2e^{2j\pi/3}, A_4 = \{0,1,2,3\} \rangle$ (см. теорему 1.5). В данном случае $Q_k = -\alpha_k$ принимает четыре значения (0, -1, -2, -3), так как α_k принимает четыре значения (0, 1, 2, 3). Используя алгоритм 2, строим таблицу 7, в которой значения S_k и их квазикоды записываются на пересечении строк Q_k и столбцов P_k в виде пары $(S_k, \langle\langle S_k \rangle\rangle)$. В этой таблице квазикоды значений S_k совпадают с позиционными кодами тех же значений и основаны на их представлении в виде $S_k = \rho^2 \eta_{k+2} + \rho \eta_{k+1} + \sigma_k$.

Таблица 7. Инвертирование кодов

σ_k	Q_k	$(S_k, \langle S_k \rangle)$ при P_k , равном			
		0	1	2	3
0	0	0, 0	1, 1	2, 2	3, 3
1	-1	-1, 123	0, 0	1, 1	2, 2
2	-2	-2, 122	-1, 123	0, 0	1, 1
3	-3	-3, 121	-2, 122	-1, 123	0, 0

Как следует из таблицы 7, частичные переносы принимают значения из множества $\{0, 1, 2\}$. В связи с этим перенос в данный разряд может принимать четыре различных значения, соответствующие всем возможным комбинациям из двух частичных переносов в данный разряд. Таким образом, $P_k \in \{0, 1, 2, 3\}$. Все эти значения P_k , так же как и все значения Q_k , присутствуют в таблице. Следовательно, она удовлетворяет условиям полноты. Ниже приведен пример вычислений с помощью таблицы 7:

переносы	1	2	0			
		1	2	0		
			0	0	0	
				1	2	0
данный код	0	3	1	3	2	2
результат	1	0	3	2	0	2

Пример 8. Сложение кодов в системе $\langle \rho = j-1; \{0, 1\} \rangle$ (см. теорему 1.4). Для описания этой операции построим, согласно алгоритму 2, сокращенную таблицу 8, содержащую только величину S_k и квазикод $\langle\langle S_k \rangle\rangle$, все разряды которого принимают значения 0 или 1. В этом случае квазикод совпадает с позиционным кодом, но два его разряда всегда имеют нулевое значение. Следовательно, в k -ом разряде вырабатывается не более шести частичных переносов, равных 1. Это означает, что и в k -ый разряд может поступить также не более шести частичных переносов, то-есть $0 \leq P_k \leq 6$. Так как $0 \leq Q_k \leq 2$ при сложении данных кодов, то $0 \leq S_k \leq 8$. Все 8 значений S_k присутствуют в таблице 8, следовательно, она удовлетворяет условиям полноты.

Таблица 8. Сложение кодов

S_k	$\langle S_k \rangle$
0	000000000
1	000000001
2	000001100
3	000001101
4	111010000
5	111010001
6	111011100
7	111011101
8	111000000

Для иллюстрации существования нескольких вариантов алгоритма 2 построим еще две таблицы 8а и 8в, описывающие сложение в данной системе. В этих таблицах цифрой a обозначена величина ‘-1’.

Таблица 8а. Сложение кодов

S_k	$\langle S_k \rangle$
-1	11101
0	00000
1	00001
2	01100
3	01101
4	a0000
5	a000

Таблица 8в. Сложение кодов

S_k	$\langle S_k \rangle$
-3	10001
-2	0aa00
-1	0aa01
0	00000
1	00001
2	aaa00
3	aaa01

Как следует из таблицы 8а, в k -ый разряд может поступать одновременно три частичных переноса (частичный перенос, всегда равный нулю, не рассматривается):

$$0 \leq \mu_{k1} \leq 1, 0 \leq \mu_{k2} \leq 1, -1 \leq \mu_{k3} \leq 1,$$

следовательно, $-1 \leq P_k \leq 3$, откуда $-1 \leq S_k \leq 5$. Таким образом, таблица 8а является полной. Аналогично доказывается полнота таблицы 8в. Выбор того или иного варианта алгоритма 2 определяется минимальным числом значений, которые принимает перенос P_k . С этой точки зрения таблица 8 уступает последним таблицам 8а и 8в. Рассмотрим пример сложения по таблице 8в:

квазикоды переносов	1	0	0	0						
			a	a	0					
				a	a	0				
					a	a	a	0		
						a	a	a	0	
слагаемое 1			1	1	0	1	0	0	1	1
слагаемое 2		1	1	1	0	0	1	0	1	1
сумма	1	1	1	0	1	0	1	1	0	0

Пример 9. Обратное сложение кодов в системе $\langle \rho = j - 1; \{0, 1\} \rangle$ (см. теорему 1.4).

Таблица 9. Обратное сложение кодов.

S_k	$\ll S_k \gg$
-2	11100
-1	11101
0	00000
1	00001
2	01100
3	01101

Аналогично предыдущему примеру построим сокращенную таблицу 9, содержащую только величины S_k и квазикод $\ll S_k \gg$, все разряды которого принимают значения 0 или 1. В этом случае квазикод совпадает с позиционным кодом, но один

его разряд всегда имеет нулевое значение. Следовательно, в k -ом разряде вырабатывается не более трех частичных переносов, равных 1. Это означает, что и в k -ый разряд может поступить не более трех частичных переносов, то-есть $0 \leq P_k \leq 3$. Так как $-2 \leq Q_k \leq 0$ при обратном сложении данных кодов, то $-2 \leq S_k \leq 3$. Все 6 значений S_k присутствуют в таблице 9, следовательно, она удовлетворяет условиям полноты. В этой таблице выделена часть, относящаяся к инвертированию кодов в этой системе.

На основе таблицы 9 построена таблица 9а истинности одноразрядных схем для и обратного сложения кодов в данной системе. В ней также выделена часть, являющаяся таблицей истинности для инвертирования. Рис. 6 иллюстрирует схему распространения переносов по одноразрядным схемам в системе кодирования данного примера.

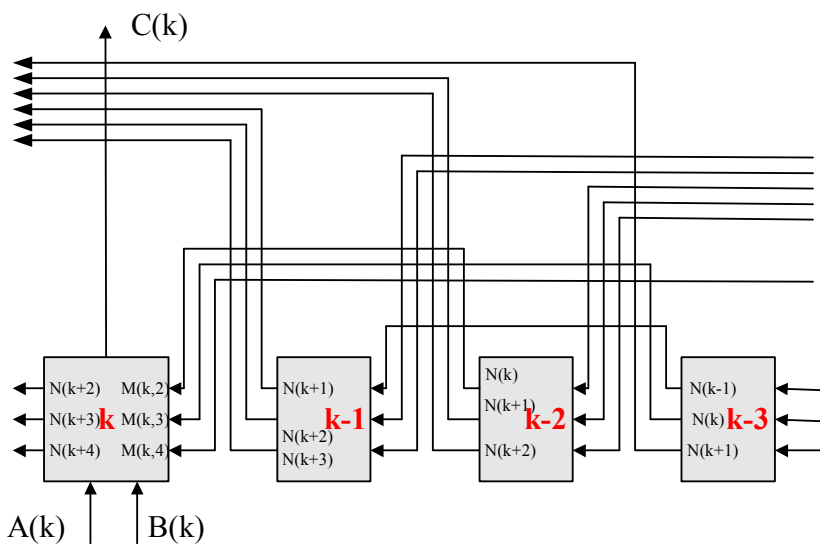


Рис. 6. Схема распространения переносов (k примеру 9)

Таблица 9а истинности.

Input					S_k	$\ll S_k \gg$			
β_k	α_k	μ_k				η_{k+j}			σ_k
		4	3	2		4	3	2	
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	2	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	0	1	0	1	2	0	1	1	0
0	0	1	1	0	2	0	1	1	0
0	0	1	1	1	3	0	1	1	1
0	1	0	0	0	-1	1	1	1	1
0	1	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	1
0	1	1	1	0	1	0	0	0	1
0	1	1	1	1	2	0	1	1	0
1	0	0	0	0	-1	1	1	1	1
1	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	0	0	1
1	0	1	1	0	1	0	0	0	1
1	0	1	1	1	2	0	1	1	0
1	1	0	0	0	-2	1	1	1	0
1	1	0	0	1	-1	1	1	1	1
1	1	0	1	0	-1	1	1	1	1
1	1	0	1	1	0	0	0	0	0
1	1	1	0	0	-1	1	1	1	1
1	1	1	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	1

Пример 10. Обратное сложение кодов в системе $\langle \rho = \sqrt{2}e^{j\varphi}, \{0, 1\} \rangle$, где (см. теорему 1.5).

Таблица 10. Обратное сложение кодов.

S_k	$\langle\langle S_k \rangle\rangle$
-2	0110
-1	0111
0	0000
1	0001
2	1010
3	1011

Аналогично предыдущему примеру построим сокращенную таблицу 10, содержащую только величины S_k и квазикод $\langle\langle S_k \rangle\rangle$, все разряды которого принимают значения 0 или 1. В этом случае квазикод совпадает с позиционным кодом. В k -ом разряде вырабатывается не более трех частичных переносов, равных 1. Это означает, что и в k -ый разряд может поступить также не более трех частичных переносов, то-есть $0 \leq P_k \leq 3$. Так как $-2 \leq Q_k \leq 0$ при обратном сложении данных кодов, то $-2 \leq S_k \leq 3$. Все 6 значений S_k присутствуют в таблице 10, следовательно, она удовлетворяет условиям полноты. В этой таблице выделена часть, относящаяся к инвертированию кодов в этой системе.

На основе таблицы 10 построена таблица 10а истинности одноразрядных схем для обратного сложения кодов в данной системе. В ней также выделена часть, являющаяся таблицей истинности для инвертирования.

Таблица 10а истинности.

Input					S_k	$\ll S_k \gg$			
β_k	α_k	μ_k				α_k			μ_k
		3	2	1		3	2	1	
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	2	1	0	1	0
0	0	1	0	0	1	0	0	0	1
0	0	1	0	1	2	1	0	1	0
0	0	1	1	0	2	1	0	1	0
0	0	1	1	1	3	1	0	1	1
0	1	0	0	0	-1	0	1	1	1
0	1	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	1
0	1	1	1	0	1	0	0	0	1
0	1	1	1	1	2	1	0	1	0
1	0	0	0	0	-1	0	1	1	1
1	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	0	0	1
1	0	1	1	0	1	0	0	0	1
1	0	1	1	1	2	1	0	1	0
1	1	0	0	0	-2	0	1	1	0
1	1	0	0	1	-1	0	1	1	1
1	1	0	1	0	-1	0	1	1	1
1	1	0	1	1	0	0	0	0	0
1	1	1	0	0	-1	0	1	1	1
1	1	1	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	1

Пример 11. Сложение кодов в системе $\langle \rho = -R, A_{R^3} \rangle$ кодирования трехмерных векторов с базой $\mathbf{i}, \mathbf{j}, \mathbf{k}$, где $A_{R^3} = \{0, \mathbf{k}, \mathbf{j}, \mathbf{j}+\mathbf{k}, \mathbf{i}, \mathbf{i}+\mathbf{k}, \mathbf{i}+\mathbf{j}, \mathbf{i}+\mathbf{j}+\mathbf{k}\}$ - см. теорему 1.7. Для обозначения элементов этого множества будем использовать восемь следующих 'цифр': $\mathbf{z} = 0, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}$. Прежде всего заметим, что в этой системе

- $\langle -\mathbf{z} \rangle = \mathbf{zz}$ - инвертирование,
- $\langle \rho \mathbf{z} \rangle = \mathbf{z0}$ - умножение на основание,
- $\langle 2\mathbf{z} \rangle = \mathbf{zz0}$ - удвоение.

Для описания сложения в этой системе построим по алгоритму 2 таблицу 11, содержащую коды слагаемых и суммы.

Таблица 11. Одноразрядное сложение.

+	0	b	c	d	e	f	g	h
0	0	b	c	d	e	f	g	h
b		bb0	d	bbc	f	bbe	h	bbg
c			cc0	ccb	g	h	cce	ccf
d				dd0	h	bbg	ccf	dde
e					ee0	eb	eec	eed
f						ff0	eed	ffc
g							gg0	ggb
h								hh0

Пользуясь этой таблицей, найдем сумму следующих кодов:

квазикоды		b	b			
переносов			e	e		
слагаемое 1				d	f	h
слагаемое 2			c	b	g	0
сумма		b	h	g	d	h

4. Поразрядные операции при отрицательном основании

Система кодирования действительных чисел по отрицательному основанию имеет ряд достоинств по сравнению с традиционной системой кодирования по положительному основанию. Эти достоинства являются следствием того, что положительные и отрицательные числа при отрицательном основании представляются единообразно, и заключаются в следующем:

- не требуется выполнять преобразования из прямого кода в обратный (или дополнительный) и обратно,
- исключаются операции со знаковыми разрядами,
- упрощаются правила определения переполнения при алгебраическом сложении,
- упрощаются алгоритмы выполнения операций с кодами переменной длины.

Вместе с тем алгоритмы арифметических операций при отрицательном основании столь же просты, как и при положительном основании. Покажем это, для чего рассмотрим так называемую операцию **обратного сложения**, выполняемую по формуле $a = -b - c$. В этой операции

$$S_k = -\alpha_k - \beta_k + P_k, \quad S_k = \sigma_k - R \cdot P_{k+1}.$$

Следовательно, $P_k \in \{0, 1\}$, то есть оба алгоритма поразрядных операций сливаются в единственный простейший алгоритм операции с двоичным переносом.

Очевидно, этот алгоритм по сложности эквивалентен алгоритму обычного сложения кодов действительных чисел по положительному основанию. В связи с этим обратное сложение следует использовать как базовую операцию алгебраического сложения при отрицательном основании, а обычное сложение, как более сложную операцию, сводить к трем операциям обратного сложения:

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

Практически операции инвертирования (являющиеся частным случаем обратного сложения) могут выполняться параллельно с основной операцией, что существенно уменьшает время сложения при любых знаках слагаемых и знаках, стоящих перед слагаемыми в формуле алгебраического сложения.

Как указывалось, два способа кодирования комплексных чисел и векторов состоят в построении некоторых композиций из кодов действительных чисел по отрицательному основанию - **композиционных кодов**. При выполнении поразрядных операций такие коды векторов удобно рассматривать как состоящие из n независимых частей - кодов чисел-проекций векторов и выполнять указанные операции с каждой из этих частей параллельно и независимо по известным алгоритмам. Все сказанное выше в данном разделе при этом сохраняет силу, что является определенным достоинством композиционных кодов векторов и комплексных чисел.

Кроме того, эти коды позволяют выполнять разноименные поразрядные операции с каждой проекцией вектора. Например, сложение и действительной и мнимой составляющих комплексных чисел может производиться независимо. В частности, так может быть выполнено инвертирование мнимой составляющей комплексного числа, в результате чего будет получено сопряженное ему число.

Итак, в композиционных системах алгебраическое сложение проекций вектора производится независимо по правилам операций в системе $\langle -R, B_R \rangle$ и только при умножении и делении явно учитывается 'векторность' основания или 'векторность' разрядов кода.

Поразрядные операции с кодами действительных чисел по основанию (-2) подробно описаны в [45]. Поэтому здесь эти операции не рассматриваются.

Глава 4. Алгоритмы кодирования и декодирования комплексных чисел

Кодируемое комплексное число представляется в виде $Z = X_\alpha + jX_\beta$, где X_α , X_β - действительная и мнимая части комплексного числа, являющиеся действительными (положительными или отрицательными) числами. Для этих чисел существуют разложения и позиционные коды по основанию $\rho = -2$. В перечисленных в табл. 1.5 системах кодирования мнимая единица может быть представлена в следующем виде: $j = \mu \cdot \omega$, где μ - действительное число, ω - комплексное число, имеющее малоразрядный код в данной системе. В табл. 1 приведены эти числа и коды для всех вышеуказанных систем кодирования.

Таблица 1.

Система кодирования		μ	ω	$K(\omega)$
№	Базовая функция и основание			
1	$f(\rho, m) = \begin{cases} \rho^{m/2} & \text{if } m - \text{even} \\ j \cdot \rho^{m-1/2} & \text{if } m - \text{odd} \end{cases},$ $\rho = -2$	1	j	10
2.1	$f(\rho, m) = \rho^m, \rho = j\sqrt{2}$	$1/\sqrt{2}$	$j\sqrt{2}$	10
2.2	$f(\rho, m) = \rho^m, \rho = -j\sqrt{2}$	$-1/\sqrt{2}$	$-j\sqrt{2}$	10
3.1	$f(\rho, m) = \rho^m, \rho = (-1 + j)$	1	j	11
3.2	$f(\rho, m) = \rho^m, \rho = (-1 - j)$	-1	$-j$	11
4	$f(\rho, m) = \rho^m, \rho = \frac{1}{2}(-1 + j\sqrt{7})$	$1/\sqrt{7}$	$j\sqrt{7}$	10101

Поэтому кодируемое комплексное число представляется в виде

$$Z = X_\alpha + \omega \cdot \bar{X}_\beta, \text{ where } \bar{X}_\beta = \mu \cdot X_\beta.$$

При этом алгоритм кодирования в общем случае имеет следующий вид:

Алгоритм 1.

1. Вычисление $\bar{X}_\beta = \mu \cdot X_\beta$ в традиционной системе двоичного кодирования.
2. Представление действительных (положительных или отрицательных) чисел X_α, \bar{X}_β в системе кодирования по основанию $\rho = -2$ в виде $X = \sum_{(m)} \alpha_m (-2)^m$, $\alpha_m = \{0,1\}$ (этот алгоритм известен и здесь не описывается).
3. Кодирование действительных чисел X_α, \bar{X}_β в данную систему кодирования комплексных чисел, путем вычисления по предыдущей формуле, где коды чисел $(-2)^m$ определены заранее (например, путем последовательного умножения на (-2)).
4. Вычисление по формуле $Z = X_\alpha + \omega \cdot \bar{X}_\beta$ в данной системе кодирования комплексных чисел, где код $K(\omega)$ известен – см. предыдущую таблицу.

Алгоритм декодирования в общем случае имеет следующий вид:

Алгоритм 2.

1. Выделение комплексных кодов действительной и мнимой частей X_α, X_β из комплексного числа $Z = X_\alpha + jX_\beta$.
2. Декодирование действительных (положительных или отрицательных) чисел X_α, X_β в систему кодирования по основанию $\rho = -2$, а именно, вычисление разрядов $\alpha_m = \{0,1\}$ для разложения вида $X = \sum_{(m)} \alpha_m (-2)^m$. Это вычисление заключается в последовательном вычитании чисел $(-2)^m$ из декодируемого числа. При этом, если очередное вычитание уменьшает модуль остатка, то $\alpha_m = 1$. В противном случае $\alpha_m = 0$, степень m уменьшается на 1 и т.д.

3. Декодирование действительных (положительных или отрицательных) чисел X_α , X_β из системы кодирования по основанию $\rho = -2$ в традиционную систему двоичного кодирования.

Для некоторых систем эти общие алгоритмы существенно упрощаются. Рассмотрим их.

Алгоритм 3. Кодирование в системе 1.

1. Представление действительных (положительных или отрицательных) чисел X_α , X_β в системе кодирования по основанию $\rho = -2$.
2. Формирование кода

$$K(Z) = \dots \beta_m \alpha_m \dots \beta_1 \alpha_1 \beta_0 \alpha_0, \beta_{-1} \alpha_{-1} \beta_{-2} \alpha_{-2} \dots \quad (1)$$

комплексного числа $Z = X_\alpha + jX_\beta$, который в дальнейшем представляется в виде

$$K(Z) = \dots \gamma_m \dots, \quad (2)$$

$$\text{где } \left\{ \begin{array}{l} \gamma_{2m} = \alpha_m \text{ if } m - \text{even} \\ \gamma_{2m+1} = \beta_m \text{ if } m - \text{odd} \end{array} \right\}$$

Алгоритм 4. Декодирование в системе 1.

1. Выделение из кода (2) комплексного числа $Z = X_\alpha + jX_\beta$ четных и нечетных разрядов по правилу

$$\left\{ \begin{array}{l} \alpha_{m/2} = \gamma_m \text{ if } m - \text{even} \\ \beta_{(m-1)/2} = \gamma_m \text{ if } m - \text{odd} \end{array} \right\}$$
2. Формирование из разрядов α_m и β_m по основанию $\rho = -2$ для чисел X_α , X_β соответственно.
3. Декодирование действительных (положительных или отрицательных) чисел X_α , X_β из системы кодирования по основанию $\rho = -2$ в традиционную систему двоичного кодирования.

Алгоритм 5. Кодирование в системе 2 при $\rho = j\sqrt{2}$.

1. Вычисление $\bar{X}_\beta = \mu \cdot X_\beta$ при $\mu = 1/\sqrt{2}$. Это вычисление выполняется в традиционной системе двоичного кодирования.
2. Представление действительных (положительных или отрицательных) чисел X_α, \bar{X}_β в системе кодирования по основанию $\rho = -2$.
3. Формирование кода (1) комплексного числа $Z = X_\alpha + jX_\beta$, который в дальнейшем представляется в виде кода (2).

Алгоритм 6. Декодирование в системе 2 при $\rho = j\sqrt{2}$.

1. Выделение из кода (2) комплексного числа $Z = X_\alpha + jX_\beta$ четных и нечетных разрядов по правилу

$$\left\{ \begin{array}{l} \alpha_{m/2} = \gamma_m \text{ if } m - \text{even} \\ \beta_{(m-1)/2} = \gamma_m \text{ if } m - \text{odd} \end{array} \right\}$$
2. Формирование из разрядов α_m и β_m кодов по основанию $\rho = -2$ для чисел X_α, \bar{X}_β соответственно, где $\bar{X}_\beta = \mu \cdot X_\beta$ при $\mu = 1/\sqrt{2}$.
2. Декодирование действительных (положительных или отрицательных) чисел X_α, \bar{X}_β из системы кодирования по основанию $\rho = -2$ в традиционную систему двоичного кодирования.
3. Вычисление $X_\beta = \bar{X}_\beta \sqrt{2}$. Это вычисление выполняется в традиционной системе двоичного кодирования.

Упомянутые выше алгоритмы кодирования и декодирования кодов по основанию (-2) и соответствующие устройства подробно описаны в [45] и поэтому здесь не рассматриваются.

Глава 5. Умножение

- 1. Специальная алгебра в векторном пространстве \ 88**
 - 1.1. Алгебра в трехмерном векторном пространстве \ 88
 - 1.2. Покомпонентное умножение \ 89
 - 1.3. Векторное произведение \ 89
 - 1.4. Скалярное произведение \ 89
 - 1.5. Поворот вектора \ 89
 - 1.6. Центраффинное преобразование \ 90
 - 1.7. Многомерное пространство \ 91
- 2. Умножение многомерных векторов \ 92**
 - 2.1. Метод умножения комплексных чисел и многомерных векторов \ 92
 - 2.2. Умножение векторов по основанию (1.31) \ 93
 - 2.3. Умножение векторов по основанию (1.30) \ 93
 - 2.4. Последовательное и матричное умножение \ 94
 - 2.5. Умножение целых кодов векторов по основанию (1.31) \ 97
 - 2.6. Умножение целых кодов векторов по основанию (1.30) \ 97
 - 2.7. Покомпонентное умножение многомерных векторов \ 98
- 3. Скалярное и векторное умножения \ 100**
 - 3.1. Скалярное произведение \ 100
 - 3.2. Векторное произведение \ 101
 - 3.3. Переносы при скалярном умножении \ 102
 - 3.4. Переносы при векторном умножении \ 104

1. Специальная алгебра в векторном пространстве

1.1. Алгебра в трехмерном векторном пространстве

Рассмотрим некоторую алгебру в трехмерном векторном пространстве. Пусть \mathbf{i} , \mathbf{j} , \mathbf{k} – база трехмерного векторного пространства. Определим для этой базы табл. 1 умножения ортов. Согласно этой таблице умножение ортов описывается следующим образом: если $U = U_1 * U_2$, где

$$U_m = x_m i + y_m j + z_m k$$

для любых индексов m , то

$$x = x_1 x_2 - y_1 z_2 - z_1 y_2,$$

$$y = x_1 y_2 + y_1 x_2 - z_1 z_2,$$

$$z = x_1 z_2 + y_1 y_2 + z_1 x_2.$$

Таблица 1. Умножение трехмерных векторов

*	\mathbf{i}	\mathbf{j}	\mathbf{k}
\mathbf{i}	i	j	k
\mathbf{j}	j	k	$-i$
\mathbf{k}	k	$-i$	$-j$

Это умножение не имеет ничего общего с векторным или скалярным умножением и, в отличие от них, обозначается далее символом $*$. Нетрудно убедиться, что умножение, определенное табл. 1 для ортов \mathbf{i} , \mathbf{j} , \mathbf{k} будет ассоциативным и коммутативным. Следовательно, умножение, определенное для любых векторов рассматриваемого векторного пространства, также будет ассоциативным и коммутативным и дистрибутивным относительно сложения. Кроме того, выполняется условие

$$(bU_1) * U_2 = U_1 * (bU_2) = b(U_1 * U_2),$$

где b – действительное число. Следовательно, табл. 1 определяет в трехмерном векторном пространстве алгебру без деления над полем действительных чисел.

Сложение в рассматриваемой алгебре соответствует обычному сложению векторов, а умножение – повороту вектора-множимого с одновременным изменением его длины. В общем случае параметры поворота – положение оси поворота, угол поворота, масштабный

множитель зависят от координат обоих сомножителей. Таким образом, геометрическая интерпретация умножения в этой алгебре довольно сложна, однако несколькими операциями умножения и сложения можно описать такие преобразования векторов, которые имеют простой геометрический смысл.

1.2. Покомпонентное умножение

Рассмотрим некоторые операции в этой алгебре, предварительно определив операцию покомпонентного умножения. Этот термин будет использоваться для наименования операции умножения вектора U_1 на упорядоченную тройку векторов U_2, U_3, U_4 . Покомпонентное умножение состоит в вычислении вектора по формуле

$$U = x_1 i * U_2 + y_1 j * U_3 + z_1 k * U_4$$

или

$$x = x_1 x_2 - y_1 z_3 - z_1 y_4,$$

$$y = x_1 y_2 - y_1 x_3 - z_1 z_4,$$

$$z = x_1 z_2 - y_1 y_3 - z_1 x_4.$$

Для обозначения этой операции будем употреблять также запись следующего вида:

$$U = U_1 * [U_2, U_3, U_4].$$

В частности, $U_1 * [U_2, U_2, U_2] = U_1 * U_2$.

1.3. Векторное произведение

$$U_1 \times U_2 = U_1 * [(-jz_2 + ky_2), (-jx_2 - kz_2), (jy_2 - kx_2)].$$

1.4. Скалярное произведение

$$i(U_1 \bullet U_2) = U_1 * [(ix_2), (-ky_2), (-jz_2)]$$

- здесь для удобства вычислений действительное число $U_1 \bullet U_2$ отождествляется с вектором $i(U_1 \bullet U_2)$.

1.5. Поворот вектора

Поворот вектора, когда он перемещается по поверхности некоторого конуса, также может быть описан покомпонентным умножением на тройку векторов, полученных из параметров поворота. Здесь уместно отметить аналогию с алгеброй

комплексных чисел, где умножение эквивалентно повороту плоского вектора.

Рассмотрим прямую с ортом

$$r_0 = i \operatorname{Cos} \alpha + j \operatorname{Cos} \beta + k \operatorname{Cos} \gamma,$$

проходящую через начало координат. Пусть вокруг этой прямой по окружности некоторого радиуса вращается точка. Ее радиус-вектор переходит из положения U_1 в положение U . Если, кроме того, вращение производится против часовой стрелки (при наблюдении с конца вектора r_0) и угол поворота $0 \leq \varphi \leq \pi$, то можно показать, что

$$U = U_1 \operatorname{Cos} \varphi + (r_0 \times U_1) \operatorname{Sin} \varphi - r_0 (r_0 \bullet U_1) (1 - \operatorname{Cos} \varphi)$$

или

$$U = U_1 * [U_2, U_3, U_4].$$

где

$$U_2 = \left\{ \begin{array}{l} i(\operatorname{Cos} \varphi \cdot \operatorname{Sin}^2 \alpha + \operatorname{Cos}^2 \alpha) + \\ j(\operatorname{Cos} \gamma \cdot \operatorname{Sin} \varphi + \operatorname{Cos} \alpha \cdot \operatorname{Cos} \beta \cdot (1 - \operatorname{Cos} \varphi)) + \\ k(-\operatorname{Cos} \beta \cdot \operatorname{Sin} \varphi + \operatorname{Cos} \alpha \cdot \operatorname{Cos} \gamma \cdot (1 - \operatorname{Cos} \varphi)) \end{array} \right\}$$

$$U_3 = \left\{ \begin{array}{l} i(\operatorname{Cos} \varphi \cdot \operatorname{Sin}^2 \beta + \operatorname{Cos}^2 \beta) + \\ j(\operatorname{Cos} \alpha \cdot \operatorname{Sin} \varphi + \operatorname{Cos} \beta \cdot \operatorname{Cos} \lambda \cdot (1 - \operatorname{Cos} \varphi)) + \\ k(\operatorname{Cos} \lambda \cdot \operatorname{Sin} \varphi - \operatorname{Cos} \alpha \cdot \operatorname{Cos} \beta \cdot (1 - \operatorname{Cos} \varphi)) \end{array} \right\}$$

$$U_4 = \left\{ \begin{array}{l} i(\operatorname{Cos} \varphi \cdot \operatorname{Sin}^2 \gamma + \operatorname{Cos}^2 \gamma) + \\ j(-\operatorname{Cos} \beta \cdot \operatorname{Sin} \varphi - \operatorname{Cos} \alpha \cdot \operatorname{Cos} \lambda \cdot (1 - \operatorname{Cos} \varphi)) + \\ k(\operatorname{Cos} \alpha \cdot \operatorname{Cos} \varphi - \operatorname{Cos} \beta \cdot \operatorname{Cos} \lambda \cdot (1 - \operatorname{Cos} \varphi)) \end{array} \right\}$$

1.6. Центроаффинное преобразование

Центроаффинное преобразование эквивалентно покомпонентному умножению на тройку векторов, построенных из элементов матрицы центроаффинного преобразования:

$$U_2 = U_1^T \cdot \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = U_1 * \begin{bmatrix} (ia_{11} - ja_{12} - ka_{13}), \\ (ia_{21} - ja_{22} - ka_{23}), \\ (ia_{31} - ja_{32} - ka_{33}) \end{bmatrix}.$$

1.7. Многомерное пространство

Обобщим теперь полученные результаты на n -мерное пространство. Выберем в нем базу E_1, E_2, \dots, E_n , элементы которой удовлетворяют условию

$$E_a E_b = -E_c \text{ при } d > n+1,$$

$$E_a E_b = E_c \text{ при } d < n+2,$$

где $d=(a+b-1)$, $c = d \bmod n$. (1)

Можно показать, что умножение, определенное для элементов этой базы, является ассоциативным, коммутативным и дистрибутивным относительно сложения, а также удовлетворяет условию

$$(bU_1)*U_2 = U_1*(bU_2) = b(U_1*U_2),$$

где b - действительное число. Следовательно, множество n -мерных векторов составляет алгебру. В частности, при $n=2$ и базе этого пространства $\{1, j\}$ получаем алгебру комплексных чисел - см. также табл. 2 умножения комплексных чисел; при $n=3$ получаем алгебру с табл. 1, описанную выше; при $n=4$ последней формуле соответствует табл. 3 умножения четырех ортов и так далее.

Таблица 2. Умножение комплексных чисел.

*	1	j
1	1	j
j	j	-1

Таблица 3. Умножение четырехмерных векторов

*	i	j	k	m
i	i	j	k	m
j	j	k	m	$-i$
k	k	m	$-i$	$-j$
m	m	$-i$	$-j$	$-k$

2. Умножение многомерных векторов

2.1. Метод умножения комплексных чисел и многомерных векторов

Необходимо найти произведение комплексных чисел или векторов $C=A*B$, где множитель и множимое имеют соответственно разложения

$$A = \sum_h \alpha_h f(\rho, h), \quad B = \sum_k \beta_k f(\rho, k).$$

Код произведения определяется как $C = \sum_h [B\alpha_h f(\rho, h)]$.

Поскольку $\alpha_h = \{0,1\}$, то умножение состоит только из операций умножения на базовую функцию $f(\rho, h)$ и суммирования.

Пример 1. Умножение в системе теоремы 1.6. Умножение двух разрядов в этой системе описывается табл. 1.2. При этом даже не возникают переносы. Рассмотрим, например, поворот плоского вектора - комплексного числа на 60° . Такой поворот соответствует умножению на код **1d** - см. табл. 1.4а. Умножение на код **d** эквивалентно преобразованию всех разрядов множимого в соответствии со столбцом 'd' табл. 1.2. Умножение на код **10**, как всегда, равносильно сдвигу влево. Сложение результатов этих двух действий дает код повернутого на 60° вектора.

Покомпонентное умножение векторов также состоит из циклов 'сдвиг-сложение'. Однако, в отличие от простого умножения действительных и комплексных чисел, в каждом цикле покомпонентного умножения значение множимого, с которым производится сложение, зависит от номера разряда множителя. В частности, если выполняется покомпонентное умножение

$$Z = Y * (X, V, W),$$

то множимое M определяется следующим образом:

$$M = X, \text{ если } m = 3k,$$

$$M = V, \text{ если } m = 3k+1,$$

$$M = W, \text{ если } m = 3k+2,$$

где m - номер разряда множителя Y , k - целое число.

Рассмотрим умножение на базовую функцию для двух важных случаев.

2.2. Умножение векторов по основанию (1.31).

В этом случае умножение множимого на основание равносильно сдвигу на h разрядов. Таким образом, умножение кодов в этой системе сводится к последовательно выполняемым операциям сдвига и сложения.

2.3. Умножение векторов по основанию (1.30).

Разрядность кода множителя $N = n \cdot t$, где n – размерность вектора. Код множителя можно разбить на t групп, а в каждой t -группе рассматривать первый (младший) разряд, второй разряд, ... i -разряд, ... n -разряд. Группы будем нумеровать также, как разряды кода, справа налево от 0 до $(m - 1)$. При этом умножение множимого на базовую функцию (если соответствующий множителя равен 1) состоит из двух действий (которые совмещаются во времени):

1. Сдвиг множимого на $h = n \cdot t$ разрядов, если рассматривается t -группа разрядов множителя.
2. Умножение множимого B на орт в зависимости от номера i разряда в группе:

$$B_i = E_i B, \quad i = \overline{1, n} \quad (2)$$

- см. также (1.21). Например, при $n=2$, имеем:

$$B_1 = B, \quad B_2 = jB;$$

при $n=3$, имеем:

$$B_1 = iB, \quad B_2 = jB, \quad B_3 = kB;$$

при $n=4$, имеем:

$$B_1 = iB, \quad B_2 = jB, \quad B_3 = kB, \quad B_4 = mB$$

и т.д. Заметим, что преобразованные множимые B_i могут быть заготовлены перед умножением.

Вычисление по формуле (2) выполняется в соответствии с таблицей умножения вектора или формулой (1). Пусть в соответствии с (1.21)

$$B = E_1 b_1 + E_2 b_2 + \dots + E_n b_n,$$

В частности, для комплексных чисел используется табл. 2. Имеем:

$$B_1 = B, \quad B_2 = j(b_1 + jb_2) = -b_2 + jb_1.$$

Для трехмерных векторов используется табл. 1. Имеем:

$$B_1 = B,$$

$$B_2 = j(b_1 + jb_2 + kb_3) = -b_3 + jb_1 + kb_2,$$

$$B_{32} = k(b_1 + jb_2 + kb_3) = -b_2 - jb_3 + kb_1.$$

Для четырехмерных векторов используется табл. 3. Имеем:

$$B_1 = B,$$

$$B_2 = j(b_1 + jb_2 + kb_3 + mb_4) = -b_4 + jb_1 + kb_2 + mb_3,$$

$$B_3 = k(b_1 + jb_2 + kb_3 + mb_4) = -b_3 - jb_4 + kb_1 + mb_2,$$

$$B_4 = m(b_1 + jb_2 + kb_3 + mb_4) = -b_2 - jb_3 - kb_4 + mb_1.$$

Отсюда следует, что умножение кода вектора на орт состоит из инвертирования некоторых компонент и перестановки компонент кода вектора.

2.4. Последовательное и матричное умножение векторов.

Для комплексных чисел и многомерных векторов можно (аналогично умножению действительных чисел) предложить последовательные и матричные алгоритмы. Рассмотрим некоторые примеры.

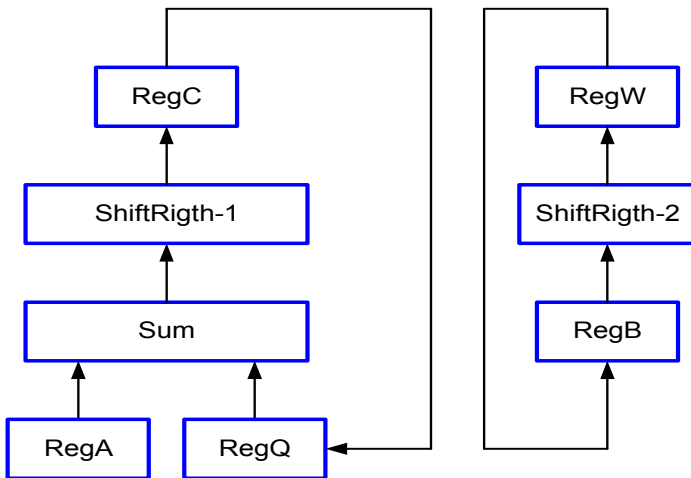


Рис. 1.

На рис. 1 приведена схема последовательного умножения, которая применима и для умножения комплексных чисел $C=A*B$.

В начале умножения множитель A записывается в регистр $RegA$, множимое B - в регистр $RegB$, а в регистр $RegQ$ частичного произведения записывается «0». На каждом шаге

- анализируется m младших разрядов $RegB$; комплексное значение этих разрядов обозначим через M ;
- в сумматоре Sum вычисляется комплексный код числа $S = M * RegA + RegQ$;
- комплексный код S с выхода сумматора Sum через сдвигатель $ShiftRigth-1$ записывается в $RegC$ со сдвигом на m разрядов вправо;
- код из $RegC$ пересылается в $RegQ$;
- комплексный код из регистра $RegB$ через сдвигатель $ShiftRigth-2$ записывается в $RegW$ со сдвигом на m разрядов вправо;
- код из $RegW$ пересылается в $RegB$.

Количество m анализируемых на каждом шаге разрядов определяет быстродействие умножения и сложность сумматора. Например, в системах 1 и 2 при $m = 2$ число $M = \{1, 0, -1, -2\}$ и сумматор должен выполнять сложение, вычитание и вычитание удвоенного числа.

Для умножения комплексных кодов $C = A * B$, кроме рассмотренной выше последовательной схемы, может быть применена матричная схема умножения – см. рис. 2. Такая схема содержит регистры $RegA$, $RegB$, $RegC$ и матричный множитель MM , который состоит из множества сумматоров $Add(K)$. Первые входы всех сумматоров подключены ко выходу регистра $RegA$. К управляющему входу каждого сумматора $Add(K)$ подключен выход K -разряда регистра $RegB$. Выход каждого сумматора (кроме $Add(N)$) соединен со входом следующего сумматора со сдвигом на 1 разряд. Выход сумматора $Add(N)$ соединен со входом регистра $RegC$.

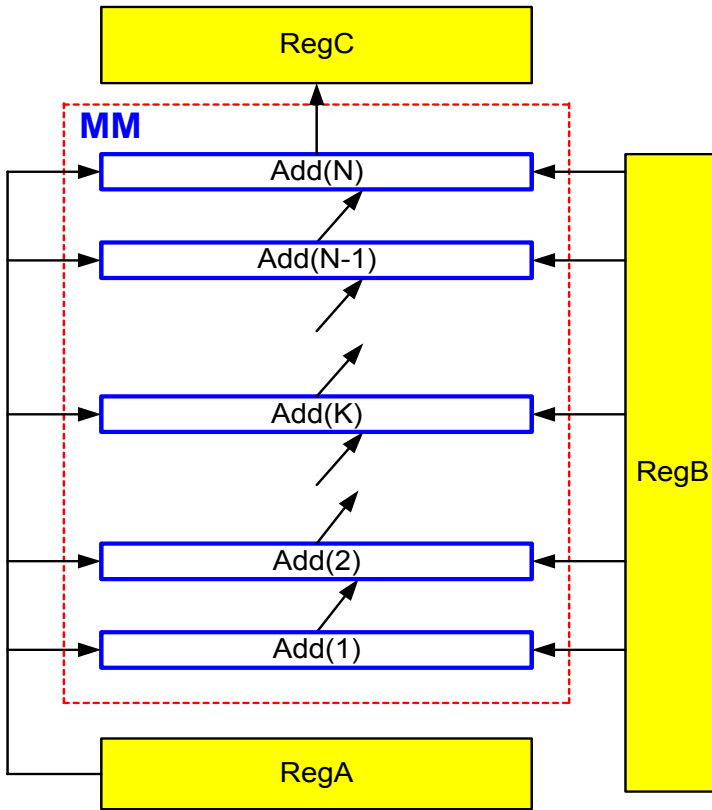


Рис. 2.

В начале умножения множитель A записывается в регистр $RegA$, а множимое B - в регистр $RegB$. Если K -разряд регистра $RegB$ равен «1», то сумматор $Add(K)$ складывает комплексный код A со сдвинутым на 1 разряд кодом выхода предыдущего сумматора. Если K -разряд регистра $RegB$ равен «0», то сумматор $Add(K)$ передает дальше сдвинутый на 1 разряд код выхода предыдущего сумматора. Таким образом на входе сумматора $Add(N)$ образуется комплексный код результата, который записывается в регистр $RegC$.

Подробно различные варианты умножителей рассмотрены в [45].

Далее мы рассмотрим простейшие алгоритмы последовательного умножения векторов. По аналогии с ним можно построить алгоритмы матричного умножения векторов.

2.5. Умножение целых кодов векторов по основанию (1.31).

Рассмотрим систему кодирования n -мерных векторов по основанию (1.31). Будем анализировать разряды множителя, начиная со старшего, и сдвигать множимое вправо. В соответствии с этим алгоритм умножения имеет вид:

1. Вначале частичное произведение равно 0, а множимое B расположено так, что его младший 0 -разряд совмещен со старшим $(N-1)$ -разрядом α_{N-1} множителя A . Номер текущего разряда множителя $t=N-1$, т.е. $\alpha_t = \alpha_{N-1}$.
2. Сложение частичного произведения со множимым B , если $\alpha_t = 1$.
3. Сдвиг множимого B на 1 разряд вправо и уменьшение текущего номера $t := t - 1$.
4. Прекращение вычисления, если $t < 0$, или переход к п. 2.

2.6. Умножение целых кодов векторов по основанию (1.30).

Рассмотрим систему кодирования n -мерных векторов по основанию (1.30). Разрядность кода множителя $N = n \cdot m$. В этом случае алгоритм умножения имеет вид:

1. Подготавливаются n вариантов множимого B по формуле (2).
2. Будем рассматривать группы по n разрядов множителя. Вначале частичное произведение равно 0, а множимые B_i расположены так, что их младшие 0 -разряды совмещены с разрядом множителя A , имеющего номер $N - n = n \cdot (m - 1)$. Номер текущей группы разрядов множителя $t=m$.
3. Рассматривается t -группа разрядов множителя A . В ней
 - 3.1. Рассматривается первый (младший) разряд $\alpha_{n(t-1)}$.
Выполняется сложение частичного произведения со множимым B_1 , если $\alpha_{n(t-1)} = 1$.
 - 3.2. Рассматривается второй разряд $\alpha_{n(t-1)+1}$. Выполняется сложение частичного произведения со множимым B_2 , если $\alpha_{n(t-1)+1} = 1$.

...

- 3.i. Рассматривается i -разряд $\alpha_{n(t-1)+i}$. Выполняется сложение частичного произведения со множимым B_i , если $\alpha_{n(t-1)+i} = 1$.
- ...
- 3.n. Рассматривается n -разряд α_{nt} . Выполняется сложение частичного произведения со множимым B_n , если $\alpha_{nt} = 1$.
4. Сдвиг множимого на n разряд вправо (напомним, что здесь n – размерность вектора) и уменьшение текущего номера $t := t - 1$.
5. Прекращение вычисления, если $t < 0$, или переход к п. 3.

2.7. Покомпонентное умножение многомерных векторов.

В отличие от простого умножения, в каждом цикле покомпонентного умножения значение множимого, с которым производится сложение, зависит от номера m разряда множителя (т.е. от того, к какой компоненте вектора множителя принадлежит этот разряд). Пусть

m - номер разряда множителя A ,
 k - целое число.

Если выполняется покомпонентное умножение *комплексного числа*

$$C = A * (X, Y),$$

то множимое B определяется следующим образом:

$$\begin{aligned} B &= X, \text{ если } m = 3k, \\ B &= Y, \text{ если } m = 3k+1. \end{aligned}$$

Если выполняется *покомпонентное умножение трехмерного вектора*

$$C = A * (X, Y, V),$$

то множимое B определяется следующим образом:

$$\begin{aligned} B &= X, \text{ если } m = 3k, \\ B &= Y, \text{ если } m = 3k+1, \\ B &= V, \text{ если } m = 3k+2. \end{aligned}$$

Если выполняется *покомпонентное умножение четырехмерного вектора*

$$C = A * (X, Y, V, W),$$

то множимое B определяется следующим образом:

$$B = X, \text{ если } m = 3k,$$

$$B = Y, \text{ если } m = 3k+1,$$

$$B = V, \text{ если } m = 3k+2,$$

$$B = W, \text{ если } m = 3k+3.$$

Как показано выше, покомпонентное умножение на заранее определенные тройки векторов эквивалентно скалярному и векторному умножению, умножению на число, центроаффинному преобразованию и т.д.

3. Скалярное и векторное умножения

При выполнении операций *скалярного и векторного* умножения, подчиняющихся каким-либо законам, которые должны выполняться в кольце, реализация умножения усложняется. Выше было показано, что эти операции могут быть заменены покомпонентным умножением. Однако при этом необходимо предварительно формировать сомножители. Поэтому далее рассматриваются другие способы скалярного и векторного умножений трехмерных векторов в системе кодирования трехмерных векторов по основанию (1.30). В этой системе код вектора имеет вид $K(Z) = \dots \sigma_m \dots$, а его разряды принимают значения $r_m \in \{0, i, j, k, i+j, i+k, j+k, i+j+k\}$. Для их обозначения будем использовать 8 следующих «цифр»:

$$\sigma_m = a, b, c, d, e, f, g, h.$$

Ниже разряды векторов-сомножителей V и W представляются векторами v_m и w_m с тремя компонентами - действительными числами, принимающими значения 0 или 1:

$$v_m = \{\alpha', \beta', \gamma'\}, w_m = \{\alpha'', \beta'', \gamma''\}.$$

3.1. Скалярное произведение

По формуле для скалярного произведения

$$v \bullet w = \alpha' \alpha'' + \beta' \beta'' + \gamma' \gamma'' \quad (3)$$

строится табл. 4, в которой для произведений - чисел указаны коды по основанию $\rho = -2$.

Таблица 4. Одноразрядное скалярное умножение.

•	a	b	c	d	e	f	g	h
a	0							
b	0	1						
c	0	0	1					
d	0	1	1	110				
e	0	0	0	0	1			
f	0	1	0	1	1	110		
g	0	0	1	1	1	1	110	
h	0	1	1	110	1	110	110	111

Скалярное произведение $Z = V \bullet W$ может быть вычислено по формуле

$$Z = \sum_k (V \bullet w_k) \rho^k.$$

Следовательно, скалярное умножение многоразрядных кодов векторов состоит из циклов 'сдвиг - скалярное умножение на k -ый разряд множителя - сложение'. В результате образуется код числа Z по основанию $\rho = -2$.

3.2. Векторное произведение

Формула векторного произведения имеет вид: $Z = V \times W$, причем $z_m = \{\alpha, \beta, \gamma\}$, где

$$\begin{aligned} \alpha &= \beta' \gamma'' - \gamma' \beta'', \\ \beta &= \gamma' \alpha'' - \alpha' \gamma'', \\ \gamma &= \alpha' \beta'' - \beta' \alpha''. \end{aligned} \quad (4)$$

Координаты вектора Z , вычисленные по формуле (4), могут принимать значения -1, 0, 1. Следовательно, вектор Z всегда можно представить как разность двух векторов, каждый из которых имеет код по основанию (1.30). Используя, далее, правила алгебраического сложения этих векторов, можно построить табл. 5, описывающую векторное умножение. В отличие от предыдущей таблицы эта таблица должна быть заполнена полностью, так как является несимметричной относительно диагонали (векторное произведение некоммутативно).

Таблица 5. Одноразрядное векторное умножение.

\times	a	b	c	d	e	f	g	h
a	0	0	0	0	0	0	0	0
b	0	0	e	e	cc	cc	cg	cg
c	0	ee	0	ee	b	ef	b	ef
d	0	ee	e	0	cd	gh	ch	cd
e	0	cc	bb	bd	0	c	bb	bd
f	0	cc	bf	bh	cc	0	dh	bf
g	0	eg	bb	fh	b	eh	0	eg
h	0	eg	bf	bd	cd	ef	cg	0

Векторное произведение $Z=V \times W$ может быть вычислено по формуле $Z = \sum_k (V \times w_k) \rho^k$. Следовательно, векторное

умножение многоразрядных кодов векторов состоит из циклов 'сдвиг - векторное умножение на k -ый разряд множителя - сложение'. В результате образуется код вектора Z по основанию (1.30).

3.3. Переносы при скалярном умножении.

При выполнении операций сдвига и алгебраического сложения код вектора удобно рассматривать как состоящий из трех независимых частей - кодов чисел - проекций вектора и выполнять указанные операции с каждой из этих частей независимо. Однако при векторном и скалярном умножении одного кода вектора на разряд другого кода дело усложняется тем, имеется перекрестное воздействие разрядов неоднотипных частей друг на друга. Рассмотрим этот вопрос подробнее сначала для скалярного умножения.

Скалярное умножение описывается формулой (3), но в случае многоразрядного кода необходимо еще учитывать перенос p из младшего разряда. При этом разрядный результат должен вычисляться по формуле

$$S = \alpha' \alpha'' + \beta' \beta'' + \gamma' \gamma'' + p. \quad (5)$$

Величина S должна быть представлена в виде

$$S = \sigma + P \rho, \quad (6)$$

где $\sigma = (0, 1)$ - значение данного разряда результата,
 P - значение переноса в старший разряд.

Нетрудно показать, что перенос P из данного разряда (а, следовательно, и перенос p в данный разряд) может принимать одно из следующих значений: $P = (-1, 0, 1, 2)$. При этом $S = (-1, 0, 1, 2, 3, 4, 5)$ и скалярное умножение описывается табл. 6.

Таблица 6. Переносы при скалярном умножении.

S	-1	0	1	2	3	4	5
σ	1	0	1	0	1	0	1
P	1	0	0	-1	-1	2	2

Распространение переносов можно организовать иначе, а именно, так, чтобы перенос в данный разряд поступал из двух предыдущих (p и q) и передавался из данного разряда в два последующих (P и Q). В этом случае разрядный результат должен вычисляться по формуле

$$S = \alpha'\alpha'' + \beta'\beta'' + \gamma'\gamma'' + p + q. \quad (7)$$

и представляться в виде

$$S = \sigma + P\rho + Q\rho^2. \quad (8)$$

В этом случае переносы могут принимать лишь два значения (0,1) и $S = (0, 1, 2, 3, 4, 5)$. При этом скалярное умножение описывается табл. 7.

Таблица 7. Переносы при скалярном умножении.

S	0	1	2	3	4	5
σ	0	1	0	1	0	1
P	0	0	1	1	0	0
Q	0	0	1	1	1	1

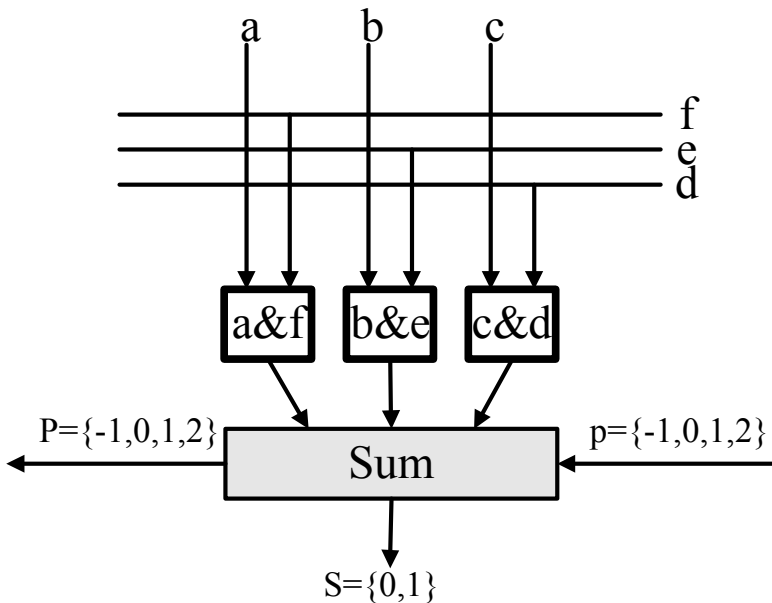


Рис. 3. Сумматор в блоке скалярного умножения

На рис. 3 приведена схема сумматора в блоке скалярного умножения на один разряд множителя. На этой схеме

a, b, c – разряды кода множимого,

d, e, f – разряд кода множителя,

p – входной перенос,

P – выходной перенос,

Sum – одноразрядный сумматор.

3.4. Переносы при векторном умножении.

Векторное умножение описывается формулами (4). С учетом переносов она приобретает следующий вид:

$$\begin{aligned} \rho P_\alpha + \alpha &= p_\alpha + \beta' \gamma'' - \gamma' \beta'', \\ \rho P_\beta + \beta &= p_\beta + \gamma' \alpha'' - \alpha' \gamma'', \\ \rho P_\gamma + \gamma &= p_\gamma + \alpha' \beta'' - \beta' \alpha''. \end{aligned} \quad (9)$$

где p, P – значения переносов в данный и в старший разряды. В формуле (9) переносы принимают только два значения (0,1). Первая (в частности) из этих формул описывается табл. 8.

Таблица 8. Переносы при векторном умножении.

$\beta' \gamma''$	$\gamma' \beta''$	p_α	α	P_α
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	0
0	1	1	0	0
1	0	1	0	-1
1	1	1	1	0
0	0	-1	1	1
0	1	-1	0	1
1	0	-1	0	0
1	1	-1	1	1

На рис. 4 приведена схема сумматора в блоке векторного умножения на один разряд множителя. На этой схеме

a, b, c – разряды кода множимого,

d, e, f – разряд кода множителя,

pG, pH, pM – входные переносы,

PG, PH, PM – выходные переносы,

$SumG, SumH, SumM$ – одноразрядные сумматоры.

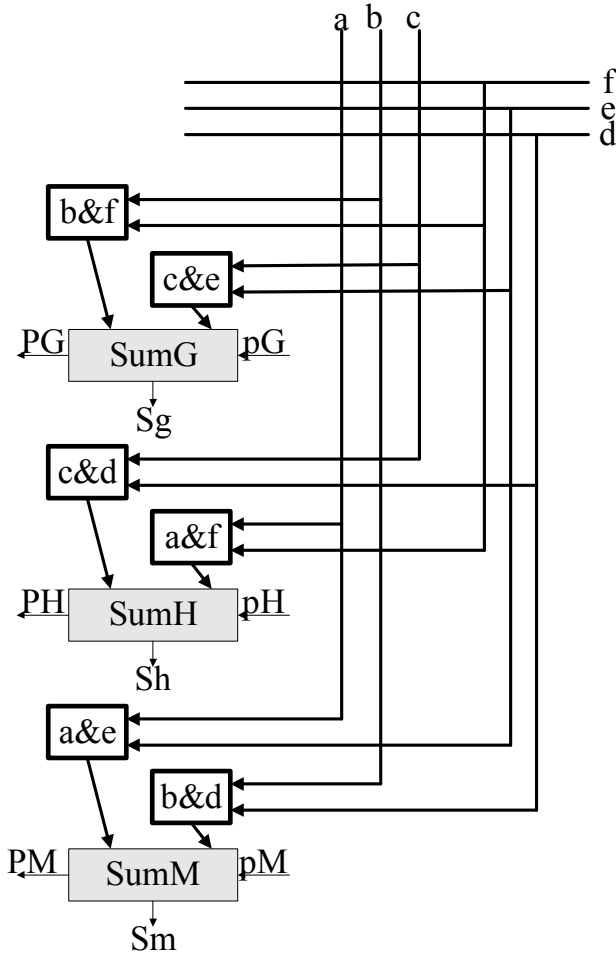


Рис. 4. Сумматор в блоке векторного умножения

Глава 6. Деление

1. Метод деления \ 105
2. Деление комплексных чисел по основанию ρ_2 \ 108
3. Деление трехмерных векторов \ 112

1. Метод деления

Деление позиционных кодов действительных чисел по отрицательному основанию и деление позиционных кодов комплексных чисел во многом аналогично обычному делению и состоит из циклов ‘сдвиг-вычитание-сравнение’. Отличие состоит только в правилах выполнения сравнения. Для действительных положительных чисел сравнение производится по предыдущему и последующему знаку остатка. Для кодов действительных чисел по отрицательному основанию знак остатка может быть определен по четности/нечетности номера старшего значащего разряда. Соответственно сравнение знаков остатков можно выполнить сравнением четности старших значащих разрядов. В нашем случае комплексных чисел этот метод не годится, поскольку не приходится говорить о знаке комплексного числа. Сравнение по знаку можно было бы заменить сравнением по модулю. Для действительных чисел это одно и то же, ибо сохранение знака остатка свидетельствует об уменьшении его модуля.

Вообще, предположим, что код комплексного числа имеет некоторую характеристику, называемую далее (условно) **размером кода**. Будем обозначать эту характеристику кода комплексного числа Z символом $NK(Z)$. В качестве размера кода может использоваться номер старшего значащего разряда или модуль кодируемого числа. Критерием правильности очередного вычитания служит сохранение или уменьшение размера кода. Ниже рассматривается алгоритм деления, основанный на сравнении длин двоичных кодов и пригодный для деления как действительных, так и комплексных чисел.

Итак, необходимо найти частное $Z=V / W$ от деления чисел - делимого и делителя, представленных своими двоичными позиционными кодами $\langle V \rangle$ и $\langle W \rangle$ соответственно.

Рассмотрим алгоритм деления, обозначив

P - предыдущий остаток

S - следующий остаток

v, w, p, s - размеры кодов V, W, P, S соответственно

Алгоритм 1.

1. Выполняется сдвиг делителя W на h разрядов влево при $h > 0$ или вправо при $h < 0$, где $h = v - w$.
2. Предыдущему остатку P присваивается значение делимого V , а частному Z - значение 0.
3. Сдвинутый делитель W вычитается из предыдущего остатка P . Разность является следующим остатком S .
4. Сравниваются размеры p и s с целью определения нового значения частного, сдвинутого делителя и величины остатка для следующего цикла деления. Эти данные определяются в соответствии с таблицей 1.
5. Выполняется переход к пункту 3 с новыми значениями сдвинутого делителя и предыдущего остатка.

Таблица 1

Условие	Результат сравнения		
	Частное	h	Предыдущий остаток для следующего цикла
$s < p$	увеличивается на единицу h -го разряда	увеличивается на единицу	$P=S$
$s = p$	увеличивается на единицу h -го разряда	остается без изменений	$P=S$
$s > p$	остается без изменений	увеличивается на единицу	$P=P$

Деление производится до получения заданного числа разрядов в частном, которое образуется суммированием единиц h -го разряда, полученных в пункте 4. Важно отметить, что случай ' $s=p$ ' может повторяться, то-есть делитель может вычитаться более одного раза, не сдвигаясь.

Пример 1. Деление кодов комплексных чисел в системе $\langle \rho = j\sqrt{2}, \{0, 1\} \rangle$ - см. теорему 1.8. Пусть $\langle V \rangle = 1001$ и $\langle W \rangle = 1010$. Воспользуемся алгоритмом 1.

	1001	
$h=0:$	-1010	
	1011	$s=p, Z=1$
$h=0:$	-1010	
	00010====→	$s < p, Z=10100$
$h=-1:$	-01010	
	101000	$s > p, Z$ не изменяется
	0000100←==	
$h=-2:$	-0001010	
	0000110==→	$s=p, Z=10100.01$
$h=-2:$	-0001010	
	0101100	$s > p, Z$ не изменяется
	00001100←=	
$h=-3:$	-00001010	
	00001110	$s=p, Z=10100.011$
$h=-3:$	-00001010	
	00000100	$s < p, Z=10110.110$
$h=-3:$	-00001010	
	00000110	$s=p, Z=10110.111$

и т. д. При прекращении деления на данном шаге получаем: $(1001-0.11):1010 = 10110.11$. Правильность деления легко проверить: $1001 - 0.11 = 11110.11$; $10110.111 * 1010 = 11110.11$.

Номер старшего значащего разряда в общем случае не может быть использован как размер кода, поскольку убывание такого размера не всегда соответствует убыванию модуля комплексного числа. Все дальнейшее изложение ведется в предположении, что размер кода – это модуль числа.

2. Деление комплексных чисел по основанию ρ_2

Рассмотрим подробнее деление кодов комплексных чисел по основанию – см. (1.2). Вначале рассмотрим операции декомпозиции и композиции.

Декомпозиция для деления состоит в разложении данного *нормализованного* кода комплексного числа Z по формуле $(-2)^{-4} = \sum_h [Za_h \rho_2(h)]$. Композиция состоит в вычислении по

формуле $\frac{(-2)^{-4}}{Z} = \sum_h [a_h \rho_2(h)]$. При аппаратной реализации

этих операций используются сумматор, сдвигатель, компаратор (блок сравнения по модулю) и коллектор. Последний представляет собой два регистра для накопления разрядов частного, которые мы будем обозначать как Acc1 и Acc2.

Схема алгоритма декомпозиции DecompDevis представлена на рис. 1, где следующие обозначения:

Z – данное комплексное число,

R – текущее значение разряда коллектора,

Accum – функция записи в данный разряд регистров частного; эта функция устанавливает текущие разряды регистров Acc1 и Acc2 по следующему правилу:

R	Acc1	Acc2
0	0	0
1	1	0
2	0	1

N – текущий номер разряда, $N_{\min} \leq N \leq N_{\max}$

Even – функция проверки четности действительного целого положительного числа N (которая эквивалентна проверке значения младшего разряда кода числа),

R_{prev} – предыдущее значение результата,

R_{next} – следующее значение результата,

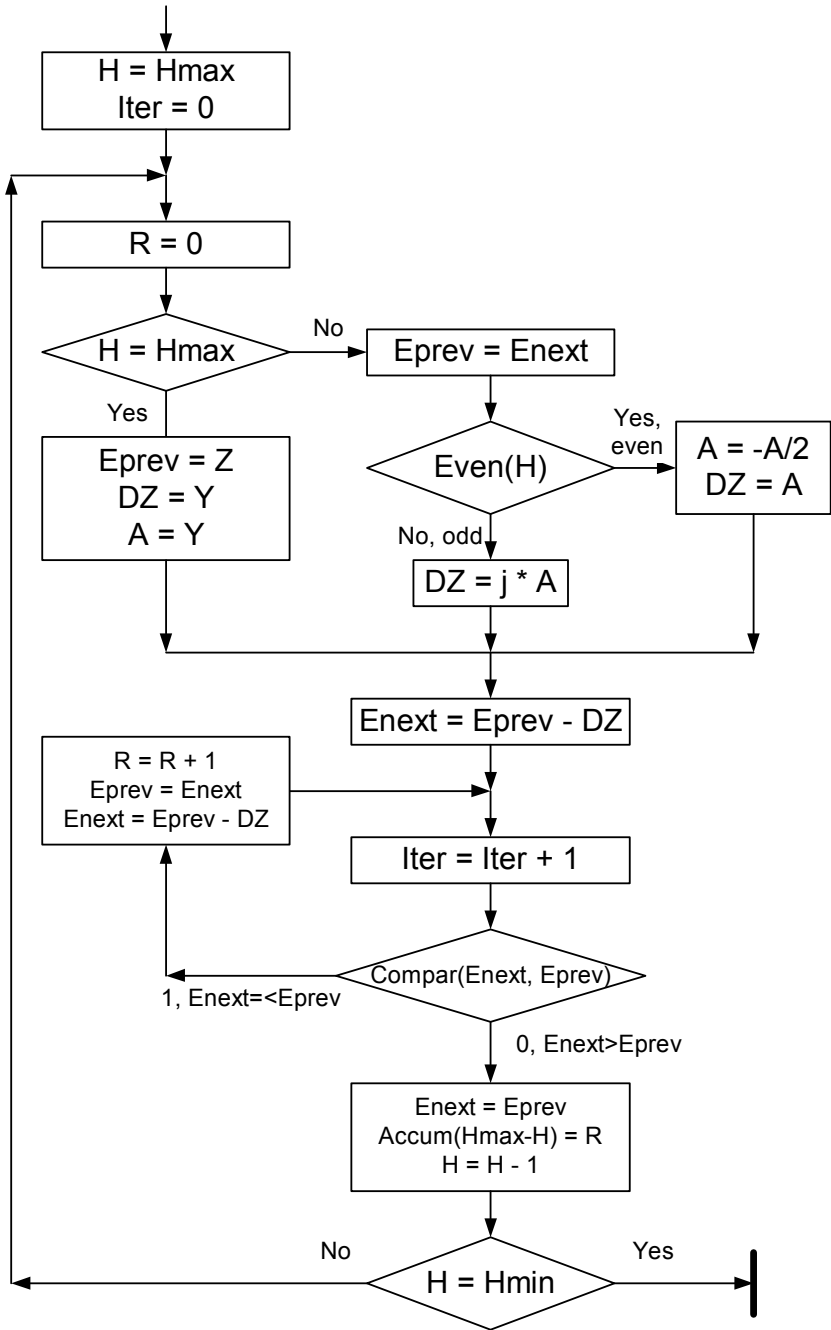


Рис. 1.

Compar— функция сравнения комплексных чисел по модулю, которая выполняется компаратором_и возвращает значение «1», если $E_{next} < E_{prev}$ и — «0», если $E_{next} > E_{prev}$,

DZ— приращение,

Iter— счетчик итераций,

A — промежуточный результат (комплексное число).

Важно отметить, что при декомпозиции число R не превышает 2, т.е. $a_h = \{0, 1, 2\}$. Поэтому композиция CompDivis эквивалентна сложению кодов, полученных в двух регистрах Acc1 и Acc2, как кодов по основанию ρ_2 , т.е. выполняется по формуле $W = \langle Acc1 \rangle - 2 \cdot \langle Acc2 \rangle$.

Декомпозиция и композиция могут быть совмещены. На рис. 3.4 представлена схема совмещенного алгоритма, где

W_{prev} — предыдущее значение результата,

W_{next} — следующее значение результата.

Теперь рассмотрим *обращение* числа Z , которое состоит в следующем:

1. Декомпозиция DecompDivis.

2. Композиция CompDivis.

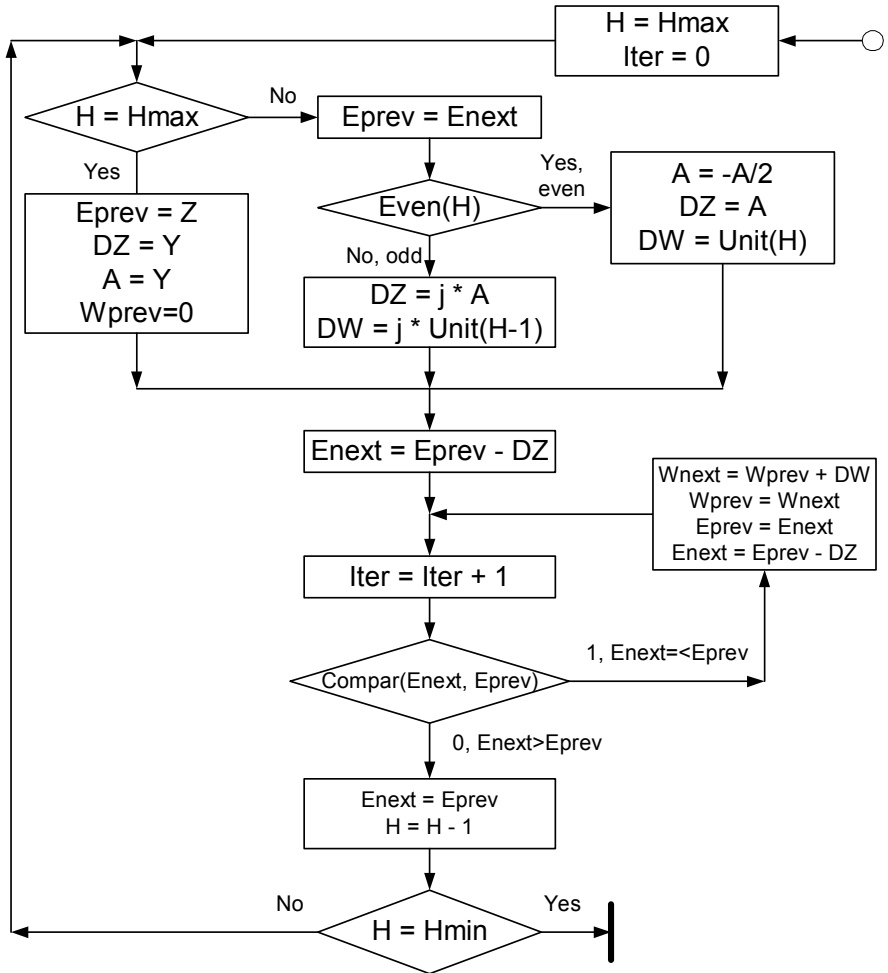
При этом определяется число $\frac{1}{16Z}$.

Наконец рассмотрим *деление* чисел $V = \frac{Y}{Z}$. Пусть делитель $Z = (-2)^k \cdot M$, где M — мантисса, k — экспонента. Деление целесообразно выполнять по следующему алгоритму:

1. Определение числа $W = \frac{1}{16 \cdot M}$, т.е. обращение

числа M .

2. Умножение $V = (-2)^{-k+4} Y \cdot W$.



Puc. 3.4

3. Деление трехмерных векторов

Рассмотрим **деление векторов**, как операцию, обратную умножению в алгебре, рассмотренной в главе 5. Итак,

$$\text{if } U = U_1 * U_2, \text{ then } U_1 = U / U_2.$$

Рассмотрим частный случай – определение обратного вектора, когда $U=1$. Пусть

$$U_1 = x \cdot i + y \cdot j + z \cdot k,$$

$$U_2 = a \cdot i + b \cdot j + c \cdot k.$$

Тогда

$$ax - yc - zb = 1,$$

$$bx + ya - zc = 0,$$

$$cx + yb + za = 0.$$

Эта система уравнений неразрешима, если ранг R расширенной матрицы коэффициентов не равен рангу P матрицы коэффициентов. Решение существует и единственно, если $R=P=3$. Матрица коэффициентов имеет вид

$$A = \begin{vmatrix} a & -c & -b \\ b & a & -c \\ c & b & a \end{vmatrix},$$

а расширенная матрица коэффициентов имеет вид

$$B = \begin{vmatrix} a & -c & -b & 1 \\ b & a & -c & 0 \\ c & b & a & 0 \end{vmatrix}.$$

Применяя разложение определителя $\det(B)$ матрицы B по четвертому столбцу, находим, что он и определитель $\det(A)$ матрицы A связаны соотношением: $\det(B) = \det(A)$. Следовательно, $R=P$.

Ранг матрицы коэффициентов $P=3$, если ее определитель не равен нулю, т.е.

$$(a^3 - b^3 + c^3 + 3abc) \neq 0. \quad (1)$$

Таким образом, $R=P$ и существует единственное решение при выполнении условия (1). Другими словами, при выполнении условия (1) возможно вычисление обратного вектора.

Преобразуем (1) к виду

$$a(bc + a^2) - b(ac + b^2) + c(ab + c^2) \neq 0. \quad (2)$$

Деление может быть заменено умножением на обратный вектор. Следовательно, деление векторов возможно, если координаты делителя удовлетворяют условию (1) или (2).

История вопроса

Кратко остановимся на истории вопроса. Компьютерная арифметика сложных математических объектов берет свое начало в статье Шеннона о позиционном кодировании действительных чисел по отрицательному основанию [1]. Эта идея, по-видимому, впервые была реализована в Польше [2] и побудила (по-видимому) нескольких авторов к разработке методов кодирования комплексных чисел. Практически одновременно Кнут [3] предложил систему кодирования по основанию $j\sqrt{2}$. Хмельник [4] предложил несколько систем, в т.ч. по основаниям $j\sqrt{2}$ и $(-1+j)$. Основание $(-1+j)$ позднее рассмотрел Penney [5]. Хмельник в диссертации [6] рассмотрел комплекс вопросов конструирования арифметического устройства для операций с комплексными числами. Эти результаты развивались затем в работах [7, 8, 9, 11, 12, 13, 14, 33, 34, 35, 44].

В нескольких работах [16, 17, 18] рассмотрены методы построения умножителей комплексных чисел. При этом основное внимание уделяется способам реализации этих устройств на чипе. Для этого предлагаются избыточные системы кодирования, которые, по мнению авторов, позволяют построить более регулярные схемы. Однако при этом не рассматриваются другие операции с предлагаемыми кодами (например, деление).

Для кодов действительных чисел известен метод ‘цифра за цифрой’ [19, 20] для аппаратного вычисления элементарных функций. Он может быть обобщен на позиционные коды комплексных чисел, что впервые было сделано Хмельником в [6, 11]. При этом часто достаточно иметь аппаратную реализацию только потенцирования и логарифмирования, поскольку через эти функции в комплексной области можно выразить все элементарные функции. Кроме того, этот метод применим для построения алгоритмов аппаратного решения трансцендентных уравнений и систем таких уравнений. При использовании кодов комплексных (а не действительных) чисел класс таких уравнений расширяется, а алгоритмы их решения существенно упрощаются. В [6, 11] описан один из таких алгоритмов.

Дальнейшее развитие идеи позиционного кодирования шло по пути построения позиционных кодов векторов [21, 22], матриц [36, 37], функций [23, 24, 25, 33, 47], геометрических фигур [22, 26, 27, 28, 32, 38, 39, 46]. Следует отметить, что коды геометрических фигур

могут рассматриваться как коды числовых массивов и для них могут быть построены эффективные поисковые алгоритмы [29, 30, 31]. Многие из этих результатов обобщены в книге [32].

Предпочтение, отдаваемое именно позиционным кодам, объясняется, главным образом, тем, что с ними очень просто выполняются арифметические операции. Так, вне зависимости от объекта кодирования, сложение позиционных кодов связано с распространением переносов от младших разрядов к старшим, а умножение состоит из сдвигов (то-есть перенумераций разрядов) и сложений. Упомянутый выше метод ‘цифра за цифрой’ вообще применим только в сочетании с позиционной системой кодирования.

Важно отметить, что в программировании для предлагаемых компьютеров используется существующий математический аппарат, не учитывающий, естественно, специфических возможностей этих компьютеров. Можно надеяться, что при распространении таких компьютеров будут найдены не только другие методы решения задач, но и другие неожиданные области применения, как это непрерывно происходит с существующими компьютерами. Например, существует теория функций пространственного комплексного переменного [40]. Алгебра четырехмерных векторов [21, 32], предложенная для их кодирования, совпадает с алгеброй пространственных комплексных чисел, используемой в [40]. В связи с этим появляется возможность разработки компьютерной арифметики пространственных комплексных чисел с аппаратным вычислением функций этого переменного, как дальнейшим обобщением метода ‘цифра за цифрой’ (подобно тому, как это было сделано для комплексных чисел [6, 11]). В этом есть практический смысл, поскольку теория функций пространственного комплексного переменного используется в весьма сложных задачах теоретической физики [40].

В предлагаемой книге можно обнаружить много аналогий с традиционной компьютерной арифметикой. Можно указать ряд книг, где подробно рассматривается эта арифметика [41, 42, 43]. Арифметика кодов по основанию (-2) широко используется при кодировании комплексных чисел и векторов. Однако в данной книге она не рассматривается, поскольку подробно описана в [45].

Литература

1. Shannon C. E., **A Symmetrical Notation of Number**, Amer. Math. Month., 57, 1950, 90.
2. Шевчик Ю., **Универсальная цифровая машина УМЦ-10**, сб. "Цифровая вычислительная техника и программирование", выпуск 2, "Советское радио", 1967.
3. Knuth D. E., **An Imaginary Number System**, Communication of the ACM-3, 1960, № 4.
4. Хмельник С. И., **Специализированная ЦВМ для операций с комплексными числами**. Вопросы радиоэлектроники, серия XII, выпуск 2, 1964 (указано, что поступила в редакцию в марте 1962 г.).
5. Penney W., **A 'Binary' System for Complex Numbers**, Journal of ACM 12, No. 2, 1965, pp. 247-248.
6. Хмельник С. И., **Методы синтеза ЦВМ, оперирующих с комплексными числами, и их использование для решения некоторых задач ПВО**, Москва, ГКРЭ при СМ СССР, ГСНИИ-5, №7907, 1964.
7. Хмельник С. И., **Сумматор кодов комплексных чисел**. Вопросы радиоэлектроники, серия XII, выпуск 3, 1965.
8. Хмельник С. И., **Позиционное кодирование комплексных чисел**. Вопросы радиоэлектроники, серия XII, выпуск 9, 1966.
9. Хмельник С. И., **Системы счисления с комплексными основаниями**. В книге [12].
10. Поспелов Д. А., **Арифметические основы вычислительных машин дискретного действия**, изд. "Высшая школа", 1970.
11. Хмельник С. И., **Решение навигационных задач на ЦВМ, оперирующих с кодами комплексных чисел**. Вопросы специальной радиоэлектроники, серия "Телемеханика и системы управления", выпуск 6, 1971.
12. Хмельник С. И., **Арифметическое устройство для цифровой вычислительной машины**. Авт.св. 266362, 1970, БИ-11.
13. Хмельник С. И., **Арифметическое устройство для операций с комплексными числами** Авт. св. 377769, 1973, БИ-18.

14. Хмельник С. И., Свистунов Ю. А, **Узлы процессоров цифровых устройств**, в книге [15].
15. **Цифровые устройства на микросхемах**. Изд. "Энергия", 1975.
16. T. Aoki, H. Amada, and T. Higuchi: **Real/Complex Reconfigurable Arithmetic Using Redundant Complex Number Systems**. In Proc. 13th Symposium on Computer Arithmetic, 1997.
17. Y. Chang and K. Parhi. **High Performance Digit Serial Complex Number Multiplier-Accumulator**, Proc. Int. Conf. on Computer Design, 1998.
18. A. M. Nielsen and P. Kornerup, **Redundant Radix Representation of Ring**, IEEE Transactions on Computers, Vol. 48 (11), November 1999
19. В. Д. Байков, В. Б. Смолов, **Специализированные процессоры: итерационные алгоритмы и структуры**, "Радио и связь", М., 1985.
20. Miller J. M., **Elementary Functions. Algorithms and Implementation**. Birkhauser, 1997, Boston.
21. Хмельник С. И., **Кодирование векторов**. Кибернетика, АН УССР, 1969, №5.
22. Хмельник С. И., **Алгебра многомерных векторов и кодирование пространственных фигур**. Автоматика и вычислительная техника, АН АССР, 1971, №1.
23. Хмельник С. И., **Кодирование функций**. Кибернетика, АН УССР, 1966, №4.
24. Хмельник С. И., **Несколько типов позиционных кодов функций**. Кибернетика, АН УССР, 1970, №5.
25. Хмельник С. И., **Алгоритмы кодирования и декодирования функциональных рядов**. Сб. "Цифровая вычислительная техника и программирование", выпуск 8, 1974.
26. Хмельник С. И., **Кодирование плоских фигур**. Автоматика и вычислительная техника, АН АССР, 1970, №6.
27. Хмельник С. И., **Цифровое устройство для геометрических преобразований изображения**. Авт. св. 333573, 1972, БИ-11
28. Хмельник С. И., **Устройство для геометрических преобразований изображения**. Авт. св. 1030816, БИ-27, 1983.
29. Хмельник С. И., **Быстродействующие поисковые процедуры**. Третий международный симпозиум по теории информации, часть II, Таллин, 1973.

30. Хмельник С. И., **Многокритериальная задача о назначениях**. Изв.АН СССР, Техническая кибернетика, №4, 1977.
31. Хмельник С. И., **Поисковые процедуры с геометрическими кодами**, ж. "Кибернетика", АН УССР, 1990, №6.
32. С. Хмельник. **Компьютерная арифметика векторов, фигур и функций**, изд. «Mathematics in Computers», Москва – Тель-Авив, 1995.
33. S. Khmelnik. **A method and system for implementing a coprocessor**. Canadian Intellectual Property Office. Application No 2,293,953. Priority 05.01.00.
34. S. Khmelnik. **A Method and System for Processing Complex Numbers**. International patent application under PCT. PCT/CA01/00007, WO 01 50332 A2. Priority 05.01.00.
35. S. Khmelnik. **A Method and System for Processing Complex Numbers**. USA, United States Patent Application No. 10/189,195. Priority 05.06.02.
36. S. Khmelnik. **Method and System for Processing Matrices of Complex Numbers**. Canadian Intellectual Property Office. Application No 2,339,919. Priority 07.03.01.
37. S. Khmelnik. **Method and System for Processing Matrices of Complex Numbers and Complex Fast Fourier Transformations**. International patent application under PCT. PCT/CA02/00295, WO 02 071254 A2. Priority 07.03.01.
38. S. Khmelnik. **Method and System for Processing Geometrical Figures**. Canadian Intellectual Property Office. Application No 2,349,924. Priority 07.07.01.
39. S. Khmelnik. **Method and System for Processing Geometrical Figures**. International patent application under PCT. PCT/CA02/00835, WO 02 099625 A2. Priority 07.06.01.
40. В. И. Елисеев. **Введение в методы теории функций пространственного комплексного переменного**, Центр научно-технического творчества молодежи «Алгоритм». - М.; НИАТ. - 1990. Шифр Д7-90/83308 в каталоге Государственной публичной научно-технической библиотеки, <http://www.maths.ru/>
41. L. Wanhammar. **DSP Integrated Circuits**, Academic Press, 1999.
42. B. Parhami. **Computer Arithmetic. Algorithms and Hardware Design**, Oxford University Press, 2000.
43. M. J. Flynn, S. F. Oberman. **Advanced Computer Arithmetic Design**, A wiley interscience Publication John Wiley & Sons, Inc., 2001.

-
44. Khmelnik S.. **A Method and System for Processing Complex Numbers.** European Patent Office, EP 1248993. Priority 12.07.01.
 45. Хмельник С. И., Дубсон И. С., Хмельник С. М., Видуецкий А. Е. **Арифметическое устройство кодов по отрицательному основанию**, изд. «Mathematics in Computers», Израиль, 2004, <http://www.lulu.com/content/93352>.
 46. Хмельник С. И. **Компьютерная арифметика геометрических фигур. Алгоритмы и аппаратура**, изд. «Mathematics in Computers», Израиль, 2004, <http://www.lulu.com/content/71555>.
 47. Хмельник С. И. **Компьютерная арифметика функций. Алгоритмы и аппаратура**, изд. «Mathematics in Computers», Израиль, 2004, <http://www.lulu.com/content/84116>.