

Компьютерная  
арифметика  
функций

*Алгоритмы и аппаратура*

Соломон Ицкович Хмельник

Россия Израиль  
2004

# Computer Arithmetic of Finctions

*Algorithms and Hardware Design*  
(in Russian)

Solomon I. Khmelnik

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, elctronic or mechanical, without written permission of the author.

Technical Editor, Cover Designer – Inna S. Doubson

**ISBN 978-0-557-07520-1**

Published by “MiC” - Mathematics in Computer Comp.

BOX 15302, Beney-Ayish, Israel, 60860

Fax: ++972-3-9471301

Printed in United States of America, Lulu Inc, ID **84116**.

Russia Israel

2004

# Summary

The book deals with a full theory, as yet not well known, and engineering solutions for the computer arithmetic of functions. Such processors can find application in medicine, meteorology, seismology, astronomy, physics, air-raid defense, etc.

The book covers the functions of one and many arguments coding theory – the codes structure, algorithms of coding and decoding, arithmetical operations. The theory is supplemented by numerous examples. Processor intended for operations with functions considered – data representation, operating blocks, hardware realization of coding, decoding and arithmetic operations algorithms. The processor's internal performance is appraised.

The book is meant for students, engineers and a users aiming to apply the computer arithmetic of functions in his own development of custom designed processors. For this purpose the book includes all information necessary for:

- understanding of the processor intended for operations with functions functioning in all details,
- application of the technical solutions presented in this book for the reader's own development.

Algorithms and units described in this book are developed into models in VHDL and FPGA. We shall welcome any kind of cooperation proposals sent to the address:

**[solik@netvision.net.il](mailto:solik@netvision.net.il)**

## Аннотация

В книге описываются малоизвестные методы построения специализированных компьютеров для обработки функций. Такие процессоры могут найти применение для медицины, метеорологии, сейсмологии, радиоастрономии, физике, противовоздушной обороне и т.п.

В книге описывается теория кодирования функций одного и многих аргументов – структура кодов, алгоритмы кодирования, декодирования, арифметические операции. Теория дополняется многочисленными примерами. Рассматривается устройство функционального процессора – представление данных, операционные блоки, техническая реализация алгоритмов кодирования, декодирования и арифметических операций. Оценивается быстродействие этого процессора.

Книга ориентирована на пользователя, который намерен применять компьютерную арифметику функций в собственных разработках специализированных процессоров. С этой целью книга включает всю информацию, необходимую для

- понимания функционирования процессора для операций с функциями во всех деталях,
- использования приведенных в книге технических решений для собственной разработки.

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](mailto:solik@netvision.net.il)

---

# Оглавление

---

Предисловие \ 11

Глава 1. Позиционные коды функций  
\ 13

1.1. Треугольные коды \ 13

*Определение 1.1.1 о разложении функции по основанию*

*Определение 1.1.2 о треугольном коде функции*

*Определение 1.1.3 об  $R$ -ом треугольном коде функции*

*Свойство 1.1.1 ТК*

*Свойство 1.1.2 ТК*

*Свойство 1.1.3 ТК*

*Свойство 1.1.4 ТК*

*Свойство 1.1.5 ТК*

1.2. Алгебраическое сложение кодов  
вещественных чисел \ 17

1.3. Алгебраическое сложение  
треугольных кодов \ 19

1.4. Деление треугольных кодов на  
параметр \ 21

1.5. Умножение треугольных кодов \ 24

1.6. Кодирование и декодирование  
треугольных кодов \ 25

*Теорема 1.6.1 о преобразовании разложений функции*

*Теорема 1.6.2 о преобразовании разложений функции*

*Определение 1.6.1 об  $R$ -ом прямоугольном коде функции*

*Свойство 1.6.1 РК*

*Свойство 1.6.2 РК*

*Свойство 1.6.3 РК*

*Свойство 1.6.4 РК*

1.7. Дифференцирование треугольных  
кодов \ 28

1.8. Ступенчатые коды \ 29

*Определение 1.8.1 о ступенчатом разложении функции*

## Глава 2. Кодирование

тригонометрических рядов \ 31

2.1. Треугольные коды функций по  
основанию  $\sin^2(x)$  \ 31

*Теорема 2.1.1 о числах  $S$*

*Теорема 2.1.2 о числах  $L$*

*Теорема 2.1.3 о функциях  $\gamma$*

*Теорема 2.1.4 о функциях  $\lambda, \varepsilon, \eta$*

*Теорема 2.1.5 о преобразовании рядов*

*Пример 2.1.1. Преобразование*

*частичного ряда при  $i=3$*

*Пример 2.1.2. Преобразование*

*частичного ряда функции  $\gamma$*

*Пример 2.1.3а. Преобразование*

*частичного ряда при  $i=0$*

*Пример 2.1.3. Взаимные преобразования*

*частичного ряда при  $i=0$ .*

*Пример 2.1.4. Взаимные преобразования*

*частичного ряда при  $i=1$ .*

*Пример 2.1.5. Взаимные преобразования*

*частичного ряда при  $i=2$ .*

*Пример 2.1.6. Взаимные преобразования*

*частичного ряда при  $i=3$ .*

2.2. Тригонометрические треугольные  
коды-ГТК \ 43

*Теорема 2.2.1 о преобразовании тригонометрического ряда*

*Теорема 2.2.2 о*

*тригонометриче*

*ском*

*треугольном*

*коде функции*

*Свойство 2.2.1 ГТК*

*Свойство 2.2.2 ТТК*

*Свойство 2.2.3 ТТК*

*Свойство 2.2.4 ТТК*

*Определение 2.2.1 о расширении и сжатии ТТК*

*Пример 2.2.1. Расширение, сдвиг и компоновка кодов*

*Пример 2.2.2. Кодирование функции*

## 2.3. Операции с ТТК \ 47

2.3.1. Короткие операции

2.3.2. Умножение

*Пример 2.3.1. Умножение тригонометрических рядов*

*Пример 2.3.2. Умножение тригонометрических рядов*

2.3.3. Дифференцирование

*Алгоритм 2.3.1. Дифференцирование кода*

2.3.4. Интегрирование

2.3.5. Инвертирование аргумента

2.3.6. Смещение оси ординат

## 2.4. Кодирование и декодирование ТТК \ 53

*Алгоритм 2.4.1. Кодирование функции*

*Алгоритм 2.4.2. Кодирование функции*

*Алгоритм 2.4.3. Декодирование кода*

## 2.5. Погрешность кодирования ТТК \ 55

*Теорема 2.5.1 о числах  $p$*

*Теорема 2.5.2 о числах  $q$*

## 2.6. Укорочение ТТК \ 59

## 2.7. Гиперболические треугольные коды \ 63

# Глава 3. Кодирование функций

## многих аргументов \ 65

### 3.1. Пирамидальные коды \ 65

*Определение 3.1.1 о разложении функции по двум основаниям*

*Определение 3.1.2 о пирамидальном коде функции*

*Определение 3.1.3 о тригонометрическом пирамидальном коде*

### 3.2. Гиперпирамидальные коды \ 67

# Глава 4. Четверичные

## тригонометрические треугольные

## коды \ 71

## 4.1. Арифметические операции с четверичными треугольными кодами \ 71

### 4.1.1. Алгебраическое сложение

*Пример 4.1.1. Умножение на параметр*

### 4.1.2. Деление на параметр

*Пример 4.1.2. Деление на параметр*

*Пример 4.1.3. Деление на параметр*

### 4.1.3. Умножение

*Пример 4.1.4. Умножение*

## 4.2. Кодирование и декодирование четверичных треугольных кодов \ 79

### 4.2.1. Кодирование и декодирование четверичных треугольных кодов

*Пример 4.2.1. Кодирование и декодирование*

*Пример 4.2.2. Преобразование кода ТК в код РК*

*Пример 4.2.3. Кодирование функции*

*Пример 4.2.4. Кодирование функции*

*Пример 4.2.5а. Декодирование кода*

*Пример 4.2.5. Декодирование кода*

### 4.2.2. Кодирование и декодирование четверичных тригонометрических треугольных кодов

*Пример 4.2.6. Кодирование функции по алгоритму 2.4.1*

*Пример 4.2.7. Кодирование функции по алгоритму 2.4.2*

*Пример 4.2.8. Декодирование кода ТК кода по алгоритму 2.4.3.*

*Пример 4.2.9. Кодирование функции в ТТК по алгоритму 2.4.1.*

*Пример 4.2.10. Коды функций  $\varepsilon$ .*

## 4.3. Математические операции с четверичными тригонометрическими треугольными кодами \ 93

*Пример 4.3.1. Сдвиг оси ординат*

*Пример 4.3.2. Умножение*

*Пример 4.3.2. Дифференцирование.*



#### 4.4. Укорочение четверичных тригонометрических треугольных кодов \ 95

*Алгоритм 4.4.1. Округление кода*

*Пример 4.4.1. Укорочение.*

*Пример 4.4.2. Округление.*

*Пример 4.4.3. Оценка погрешности округления*

#### 4.5. Погрешность кодирования четверичных тригонометрических треугольных кодов \ 100

### Глава 5. Арифметическое устройство для операций с функциями \ 105

#### 5.1. Одноразрядные схемы \ 105

- 5.1.1. Одноразрядный сумматор.
- 5.1.2. Одноразрядный вычитатель.
- 5.1.3. Одноразрядный инвертор.
- 5.1.4. Одноразрядный учетверитель.
- 5.1.5. Одноразрядный делитель на 4.

#### 5.2. Многоразрядные схемы \ 110

- 5.2.1. Столбцовый сумматор.
- 5.2.2. Строчный сумматор.
- 5.2.3. Столбцовый делитель.
- 5.2.4. Строчный делитель.
- 5.2.5. Параллельный сумматор.
- 5.2.6. Параллельный делитель.

#### 5.3. Вариант арифметического устройства \ 115

- 5.3.1. Структура арифметического устройства.
- 5.3.2. Операции с треугольными кодами.
  - 1. Алгебраическое сложение смешанных кодов.
  - 2. Учетверение смешанного кода.
  - 3. Деление треугольного кода на 4.
  - 5. Укорочение треугольного кода.
  - 7. Преобразование прямоугольного кода в треугольный код.
  - 8. Преобразование треугольного кода в прямоугольный код.
  - 9. Умножение треугольных кодов.

5.3.3. Операции с тригонометрическими

треугольными кодами.

1. Алгебраическое сложение ТТК.

2. Умножение ТТК.

3. Дифференцирование ТТК.

4. Интегрирование ТТК.

5. Кодирование тригонометрического ряда.

6. Декодирование ТТК.

7. Укорочение ТТК.

5.4. Сравнительный анализ \ 124

5.4.1. Взаимосвязь между разрядностью ТТК, рангом ряда и разрядностью коэффициентов ряда.

5.4.2. Разрядность.

5.4.3. Объем арифметического устройства.

5.4.4. Длительность элементарных операций.

5.4.5. Взаимосвязь между элементарными операциями и операциями с функциями.

5.4.6. Выводы.

Глава 6. Устройства для операций с М-кодами \ 129

6.1. Алгебраическое сложение М-кодов \ 129

6.1.1. Многоразрядные схемы для М-кодов

6.1.2. Инвертор М-кода

6.1.3. Инверсный сумматор М-кодов

6.1.4. Сумматор М-кодов

6.1.5. Вычитатель М-кодов

6.1.6. Знакоопределитель М-кодов

6.2. Устройства для кодирования и декодирования М-кодов \ 136

6.2.1. Кодер положительного Р-кода в М-код

6.2.2. Декодер М-кода в Р-код

6.2.3. Полный декодер М-кода в Р-код

Литература \ 141

Обозначения \ 142

# Предисловие

---

В книге описывается компьютер, оперирующий с функциями. Впервые вычислительная машина для операций с функциями была предложена и разработана М.А. Карцевым в 1967 году [1]. В число операций этой вычислительной машины входили «сложение, вычитание и умножение функций, сравнение функций, аналогичные операции над функцией и числом, отыскание максимума функций, вычисление неопределенного интеграла, вычисление определенного интеграла от производной двух функций, сдвиг функции по абсциссе и т.д.» По архитектуре эта вычислительная машина являлась (пользуясь современной терминологией) векторным процессором. В ней использовался тот факт, что «многие из этих операций могут быть истолкованы как известные операции над векторами: сложение и вычитание функций - как сложение и вычитание векторов, вычисление определенного интеграла от производной двух функций - как вычисление скалярного произведения двух векторов, сдвиг функций по абсциссе - как поворот вектора относительно осей координат и т.д.».

В отличие от этого предлагаемый компьютер основан на представлении функции единым двоичным кодом. В нем указанные операции с функциями (и, кроме того, дифференцирование и интегрирование функций) выполняются как уникальные машинные операции с такими кодами на единственном арифметическом устройстве. При этом объем данных сокращается, а конструкция процессора резко упрощается: нет задачи организации параллельных вычислений на множестве скалярных арифметических устройств. Однако вместо множества простых скалярных арифметических устройств в предлагаемом компьютере появляется сложное арифметическое устройство для операций с кодами функций – функциональное арифметическое устройство.

Итак, для операций с функциями можно предложить

- компьютер Т – традиционный компьютер, содержащий единственное скалярное арифметическое устройство,
- компьютер Р – векторный компьютер, содержащий несколько скалярных арифметических устройств,

- компьютер  $F$  – компьютер, содержащий функциональное арифметическое устройство.

Ниже показано, что по сравнению с компьютером  $T$  у компьютера  $P$  быстродействие растет пропорционально объему, а у компьютера  $F$  быстродействие увеличивается в 8 раз быстрее увеличения объема.

Теория предлагаемых компьютеров была предложена и разработана автором в работах [2-6].

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.

Книга содержит 6 глав. В первой главе описываются позиционные коды функций общего вида – структура кодов и алгоритмы операций с ними. Во второй главе детально рассматриваются коды тригонометрических рядов. Этому вопросу уделяется особое внимание в связи с тем, что тригонометрические ряды Фурье накладывают наименьшие ограничения на вид функции (которые представляются этими рядами), а операции с рядами Фурье очень широко используются в прикладных задачах. В третьей главе рассматриваются позиционные коды функций многих аргументов. В четвертой главе подробно рассматриваются алгоритмы операций с четверичными тригонометрическими треугольными кодами, описанные в общем случае в предыдущих главах. Целью этой главы является конкретизация теории кодирования функций до такой степени, которая допускает постановку задачи технического проектирования соответствующих компьютеров. В пятой главе рассматривается конструкция и система команд одного варианта арифметического устройства для операций с кодами функций. В этой главе производится также сравнительный анализ, на который мы уже ссылались выше. Наконец, в шестой главе описываются компьютерная устройства для операций с кодами вещественных чисел по отрицательному основанию, поскольку такие устройства входят составной частью в устройства для операций с кодами функций.

# Глава 1. Позиционные коды функций

## 1.1. Треугольные коды

Ниже предполагается, что функция определена функциональным рядом. При этом тривиальный способ кодирования функций мог бы заключаться в задании коэффициентов этих рядов. Однако такой способ создает коды большого объема и не эффективен для умножения.

В дальнейшем для общности будем полагать, что функция  $\Phi(x)$  задана отношением  $\Phi(x) = h \frac{f_1(x)}{f_2(x)}$ , где  $f_1(x)$  и  $f_2(x)$  -

функции, представленные рядами или их кодами,  $h$  - числовой множитель – *экспонента функции*. Такое задание значительно расширяет класс представимых функций и позволяет избежать операции деления. Экспонента  $h$  снимает ограничения с величины коэффициентов ряда, которые возникают при кодировании функций. Дадим некоторые определения.

**Определение 1.1.1.** Двойная сумма вида

$$F(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} R^k y^{k-m} (1-y)^m, \quad (1.1.1)$$

где  $\alpha_{mk}$  - действительные числа,

$R$  - целое положительное число,

$y=f(x)$  - некоторая функция аргумента  $x$ ,

$m, k, n$  - целые положительные числа или нули

называется разложением функции  $F(x)$  по основанию  $y$  с параметром  $R$ ; в этом разложении

- функция  $\varphi = \alpha_{mk} R^k y^{k-m} (1-y)^m$  называется  $mk$ -разрядом;

## 1.1. Треугольные коды

- функция  $\psi = R^k y^{k-m} (1-y)^m$  называется весом  $mk$ -разряда,
- число  $\alpha_{mk}$  называется величиной  $mk$ -разряда.

Таким образом,

$$F(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} \psi_{mk} \quad (1.1.2)$$

или

$$F(x) = \sum_{k=0}^n \sum_{m=0}^k \varphi_{mk} \quad (1.1.3)$$

При известном  $x$  и, следовательно, известном  $y$  значение функции  $F(x)$  может быть вычислено непосредственно по (1.1.1). Однако для такого вычисления эта формула может быть представлена в более удобном виде:

$$F = (((((c_n R y + c_{n-1}) R y + c_{n-2}) R y + \dots) R y + \dots + c_1) R y + c_0),$$

где

$$c_k = (((((\alpha_{kk} b + \alpha_{k-1,k}) b + \alpha_{k-2,k}) b + \dots) b + \dots + \alpha_{1,k}) b + \alpha_{0,k}),$$

$$b = (1-y)/y.$$

**Определение 1.1.2.** Треугольная матрица, составленная из величин  $\alpha_{mk}$  разложения функции  $F(x)$  по основанию  $y$  с параметром  $R$  таким образом, что каждая величина  $\alpha_{mk}$  принадлежит  $k$ -столбцу и  $m$ -строке этой матрицы, называется треугольным кодом функции  $F(x)$  по основанию  $y$  с параметром  $R$ ; этот треугольный код обозначается символами  $TK(F(x))$ , где обозначение функции может быть опущено, если по контексту ясно, о какой функции идет речь. Таким образом,

$$TK(F(x)) = \begin{pmatrix} & & & & & & \alpha_{nn} \\ & & & & & \dots & \dots \\ & & & & & \alpha_{kk} & \dots & \alpha_{kn} \\ & & & & \dots & \dots & \dots & \dots \\ & & & \alpha_{mm} & \dots & \alpha_{mk} & \dots & \alpha_{mn} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha_{11} & \dots & \alpha_{1m} & \dots & \alpha_{1k} & \dots & \alpha_{1n} \\ \alpha_{00} & \alpha_{01} & \dots & \alpha_{0m} & \dots & \alpha_{0k} & \dots & \alpha_{0n} \end{pmatrix}$$

Вес разрядов ТК зависит от его местоположения. В этом смысле треугольные коды функций можно называть также позиционными кодами.

**Определение 1.1.3.** Треугольный код называется  $R$ -ым и обозначается символом  $TK_R$ , если величины  $\alpha_{mk}$  принимают значения из множества  $D_R = \{-r_1, -r_1 + 1, \dots, -1, 0, 1, \dots, r_2 - 1, r_2\}$ , где  $R = r_1 + r_2 + 1$ ,  $r_1 > 0$ ,  $r_2 > 0$ .

Например, треугольный код называется троичным, если  $\alpha_{mk} \in \{-1, 0, 1\}$ , и - четверичным, если  $\alpha_{mk} \in \{-2, -1, 0, 1\}$ . Будем также использовать следующие написания ТК:

$$\begin{array}{c} A \\ B \ C \\ TK = (m) \ D \ E \ F \\ (k) \end{array}$$

или

$$\begin{array}{c} A \\ * \ * \\ TK = D \ * \ F \end{array}$$

где в скобках указаны (в случае необходимости) число отброшенных нижних строк ( $m$ ) и левых столбцов ( $k$ ), а номера разрядов не обозначены. Разряды, не участвующие в данном преобразовании, обозначаются как \*.

Рассмотрим некоторые свойства ТК.

**Свойство 1.1.1.** Треугольные коды  $mk$ -разряда и веса  $mk$ -разряда имеют соответственно вид

$$\begin{array}{c} 0 \\ 0 \ 0 \\ TK(\varphi_{mk}) = (m) \ \alpha_{mk} \ 0 \ 0 \\ (k) \end{array}$$

и

$$\begin{array}{c} 0 \\ 0 \ 0 \\ TK(\psi_{mk}) = (m) \ 1 \ 0 \ 0 \\ (k) \end{array}$$

**Свойство 1.1.2.** Преобразование кода

$$\begin{array}{c} A \\ B \ C \\ TK(f(x)) = (p) \ D \ E \ F \\ (q) \end{array}$$

В КОД

$$\begin{array}{cccc} & & & \text{A} \\ & & & \text{B C} \\ \text{TK}(h(x)) = (p+m) & \text{D} & \text{E F} & \\ & & & (q+k) \end{array}$$

называемое  $mk$ -сдвигом, соответствует следующей операции с кодируемыми функциями:  $h(x) = f(x)R^k y^{k-m}(1-y)^m$ ; в частности, если  $k > 0$  и  $m > 0$ , то  $h(x) = f(x) \cdot \psi_{mk}$ , то-есть  $mk$ -сдвиг соответствует умножению ТК на вес  $mk$ -разряда. Справедливо также обратное утверждение. Аналогично, если  $k < 0$  и  $m < 0$ , то  $h(x) = f(x) / \psi_{mk}$ , то-есть  $mk$ -сдвиг соответствует делению ТК на вес  $mk$ -разряда.

**Свойство 1.1.3.** Умножение на число, сложение и вычитание ТК суть аналогичные операции с каждым разрядом или с одноименными разрядами ТК.

**Свойство 1.1.4.** Число разрядов ТК, имеющего  $(n+1)$  столбцов,  $s = (n+1)(n+2)/2$ .

**Свойство 1.1.5.** Из очевидной формулы

$$R\psi_{mk} = \psi_{m,k+1} + \psi_{m+1,k+1}$$

следует, что

$$\begin{bmatrix} 0 \\ Ra & 0 \end{bmatrix} = \begin{bmatrix} a \\ 0 & a \end{bmatrix}, \quad \begin{bmatrix} a \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ Ra & -a \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 & a \end{bmatrix} = \begin{bmatrix} -a \\ Ra & 0 \end{bmatrix}.$$



## 1.2. Алгебраическое сложение кодов вещественных чисел.

Вначале рассмотрим алгебраическое сложение кодов  $K_R(a)$  вещественных чисел  $a$  по основанию  $R$  с разрядами, принимающими положительное и отрицательное значения из множества  $D_R$ , содержащего  $R$  чисел. Такие коды составляют прямоугольный код функции, а алгоритм их алгебраического сложения служит (как будет ясно из дальнейшего) аналогией алгебраического сложения треугольных кодов.

Алгебраическое сложение пары одноименных разрядов кодов чисел описывается формулой

$$S_{mk} = Q_{mk} + \pi'_{mk}, \quad (1.2.1)$$

причем

$$\left\{ \begin{array}{l} Q_{mk} = \alpha_{mk} + \beta_{mk} \text{ при сложении,} \\ Q_{mk} = \alpha_{mk} - \beta_{mk} \text{ при вычитании,} \\ Q_{mk} = -\alpha_{mk} - \beta_{mk} \text{ при обратном сложении,} \\ Q_{mk} = -\alpha_{mk} \text{ при инвертировании,} \\ Q_{mk} = R\alpha_{mk} \text{ при умножении на параметр,} \end{array} \right. \quad (1.2.2)$$

где  $S_{mk}$  - разрядный результат,  
 $\pi'_{mk}$  - перенос из предыдущего разряда,  
 $Q_{mk}$  - частный разрядный результат,  
 $\alpha_{mk}, \beta_{mk}$  - разряды слагаемых кодов.

Представим разрядный результат в виде

$$S_{mk} = \sigma_{mk} + R\pi_{mk}, \quad (1.2.3)$$

где

$\sigma_{mk}$  - разряд результирующего кода,  
 $\pi_{mk}$  - перенос из данного разряда.

Из изложенного следует, что алгоритм алгебраического сложения  $R$ -ых треугольных кодов чисел существует, если

- переносы  $\pi_{mk}, \pi'_{mk}$  принимают значения из общего ограниченного множества  $\mathbf{P}$ ,

## 1.2. Алгебраическое сложение кодов вещественных чисел

---

- любая сумма вида (1.2.1) представима также в виде (1.2.3), где  $\sigma_{mk} \in D_R$ .

Приведенные формулы позволяют построить таблицы, описывающие процесс одноразрядного алгебраического сложения. В применении к синтезу таких таблиц вышеприведенные условия трансформируются в условия полноты таблицы:

- если  $\pi_{mk}$  принимает в таблице некоторое значение, то  $\pi'_{mk}$  также принимает это значение,
- сумма  $S_{mk}$  любой комбинации чисел  $\alpha_{mk}$ ,  $\beta_{mk}$ ,  $\pi'_{mk}$  присутствует в таблице.

В качестве примера приведена табл. 1.2.1, описывающая одноразрядное сложение при  $R=3$  и  $D_R = \{-1,0,1\}$ . Нетрудно убедиться, что эта таблица удовлетворяет условиям полноты.

Таблица 1.2.1.

$S_{mk}$	$K_R(S_{mk})$	$\sigma_{mk}$	$\pi_{mk}$
0	0	0	0
1	1	1	0
2	1	-1	1
3	1	0	1
-1	-1	-1	0
-2	-1	1	-1
-3	-1	0	-1

### 1.3. Алгебраическое сложение треугольных кодов

Алгебраическое сложение ТК (сложение, обратное сложение, инвертирование, умножение на параметр  $R$ ) связано с выполнением одноименной операции над парой одноименных разрядов слагаемых кодов или над каждым разрядом кода, умножаемого на  $(-1)$  или  $R$ . В свойстве 1.1.3 отмечено, что эти операции выполняются чрезвычайно просто, если на величины  $\alpha_{mk}$  не накладывается ограничений. Однако для возможности использования ТК в вычислительных устройствах и упрощения технической реализации операций с ними необходимо, чтобы эти величины принимали значения из ограниченного множества. Этому требованию удовлетворяют  $\text{TK}_R$ , описанные в определении 1.1.3. Ограничения, накладываемые на величины  $\alpha_{mk}$ , естественно, вызывают усложнение алгоритмов операций, что связано, как и в обычных кодах, с возникновением переносов в старшие разряды при получении в данном разряде величины  $\alpha_{mk} \notin D_R$ . Однако, если в обычных  $R$ -ых позиционных кодах выработка переносов основывается на соотношении  $K(R)=10$ , то в данном случае используется соотношение

$$\text{TK}_R(R) = 0 \quad 1 \quad (1.3.1)$$

Таким образом, в треугольных кодах перенос распространяется в два разряда старшего столбца (перенос-вилка). Следовательно, в данный разряд могут поступить переносы также из двух младших разрядов. Поэтому алгебраическое сложение пары одноименных разрядов кодов описывается формулой

$$S_{mk} = Q_{mk} + \pi'_{mk} + \pi''_{mk}, \quad (1.3.1)$$

причем  $Q_{mk}$  определяется по (1.2.2). Представим разрядный результат в виде (1.2.3). Тогда получим:

$$\text{TK}(S_{mk}) = \begin{matrix} & \pi_{mk} \\ \sigma_{mk} & \pi_{mk} \end{matrix},$$

где  $\sigma_{mk}$  - разряд результирующего кода,

$\pi_{mk}$  - перенос из данного разряда.

### 1.3. Алгебраическое сложение треугольных кодов

Из изложенного следует, что алгоритм алгебраического сложения  $R$ -ых треугольных кодов существует, если

- переносы  $\pi_{mk}$ ,  $\pi'_{mk}$ ,  $\pi''_{mk}$  принимают значения из общего ограниченного множества  $\mathbf{P}$ ,
- любая сумма вида (1.3.1) представима также в виде (1.3.2), где  $\sigma_{mk} \in D_R$ .

Нетрудно убедиться, что эти условия выполняются при  $R > 2$ . Приведенные формулы позволяют построить таблицы, описывающие процесс одноразрядного алгебраического сложения. В применении к синтезу таких таблиц вышеприведенные условия трансформируются в условия полноты таблицы:

- если  $\pi_{mk}$  принимает в таблице некоторое значение, то  $\pi'_{mk}$  и  $\pi''_{mk}$  также принимают это значение,
- сумма  $S_{mk}$  любой комбинации чисел  $\alpha_{mk}$ ,  $\beta_{mk}$ ,  $\pi'_{mk}$  и  $\pi''_{mk}$  присутствует в таблице.

В качестве примера приведена табл. 1.3.1, описывающая одноразрядное сложение при  $R=3$ . Нетрудно убедиться, что эта таблица удовлетворяет условиям полноты.

Таблица 1.3.1.

$S_{mk}$	TK( $S_{mk}$ )	$\sigma_{mk}$	$\pi_{mk}$
0	0	0	0
1	1	1	0
-1	-1	-1	0
2	1		
	-1 1	-1	1
3	1		
	0 1	0	1
4	1		
	1 1	1	1
-2	-1		
	1 -1	1	-1
-3	-1		
	0 -1	0	-1
-4	-1		
	-1 -1	-1	-1

## 1.4. Деление треугольных кодов на параметр

Эта операция основана на использовании свойства 1.1.5

$$\begin{bmatrix} (\pi_{mk}/R) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \pi_{mk} & (-\pi_{mk}/R) \end{bmatrix},$$

откуда следует, что деление каждого разряда вызывает переносы в два нижних разряда. Следовательно, разрядный результат в этой операции является суммой частного от деления данного разряда на  $R$  и переносов из двух верхних разрядов:

$$S_{mk} = \frac{\alpha_{mk}}{R} - \frac{\pi'_{mk}}{R} + \pi''_{mk} \quad (1.4.1)$$

где

$\alpha_{mk}$  -  $mk$ -разряд делимого кода,

$\pi'_{mk}$  - перенос из  $(m+1, k)$ -разряда одноименного столбца в  $(m, k)$ -разряд,

$\pi''_{mk}$  - перенос из  $(m+1, k+1)$ -разряда из старшего столбца в  $(m, k)$ -разряд.

Представим разрядный результат в виде

$$S_{mk} = \sigma_{mk} + \frac{\pi_{mk}}{R}, \quad (1.4.2)$$

где

$\sigma_{mk}$  -  $mk$ -разряд делимого кода,

$\pi_{mk}$  - перенос из  $(m, k)$ -разряда в нижнюю строку,

Из изложенного следует, что алгоритм деления на параметр  $R$ -ых треугольных кодов существует, если

- Переносы  $(\pi_{mk}, \pi'_{mk}, \pi''_{mk}) \in D_R$ ,
- любая сумма вида (1.4.1) представима также в виде (1.4.2), где  $\sigma_{mk}$  - целое число.

Нетрудно убедиться, что эти условия выполняются при любых  $R$ . Приведенные формулы позволяют построить таблицу, описывающую процесс деления и удовлетворяющую условиям полноты:

- если  $\pi'_{mk}, \pi''_{mk}$  принимают в таблице некоторое значение, то  $\pi_{mk}$  также принимает это значение,

## 1.4. Деление треугольных кодов на параметр

- любая комбинация чисел  $\alpha_{mk}, \pi'_{mk}, \pi''_{mk}$  присутствует в таблице.

В качестве примера приведена табл. 1.4.1, описывающая одноразрядное деление на  $R=3$ . Из этой таблицы следует, что  $-2 \leq \sigma_{mk} \leq 2$ . Следовательно, после деления должно выполняться распространение переносов из разрядов, не удовлетворяющих условию  $-1 \leq \sigma_{mk} \leq 1$ .

Таблица 1.4.1.

$S_{mk}$	TK( $S_{mk}$ )	$\sigma_{mk}$	$\pi_{mk}$
0	0	0	0
1	1	1	0
-1	-1	-1	0
1/3	0		
	1	-1/3	0
2/3	0		
	-1	1/3	1
4/3	1		
	1	-1/3	1
5/3	2		
	-1	1/3	2
-1/3	0		
	-1	1/3	0
-2/3	-1		
	1	-1/3	-1
-4/3	-1		
	-1	1/3	-1
-5/3	-2		
	1	-1/3	-2

Переносы, возникающие в разрядах 0-строки треугольного кода, не могут распространяться ниже. В связи с этим линейный код, разрядами которого являются переносы  $\pi_{0k}$ , рассматривается как  $R$ -ый линейный код функции - остатка  $o(x)$ , вес  $0k$ -разряда которого равен функции  $\psi_{0k}$ :

$$\text{ЛК}_R(o(x)) = \pi_{00} \dots \pi_{0k} \dots \pi_{0k}$$

Коды функций частного и остатка могут быть представлены в виде единого так называемого смешанного кода. Подробнее линейные и смешанные коды будут рассмотрены ниже.

$$\begin{array}{ccccccc}
 & & & & & & \beta_{0n} \\
 & & & & & & \dots & \dots \\
 & & & & & & \beta_{0n} & \dots & \beta_{0n} \\
 & & & & & & \beta_{0n} & \beta_{0n} & \dots & \beta_{0n} \\
 \text{SK}_R(F(x)/R) = & \beta_{0n} & \beta_{0n} & \beta_{0n} & \dots & \beta_{0n} \\
 & \pi_{00} & \pi_{00} & \pi_{00} & \dots & \pi_{00}
 \end{array}$$

Иллюстрация деления на параметр  $\bar{\beta}$  будет рассмотрена далее.

## 1.5. Умножение треугольных кодов

Произведение  $P(x)=M(x)F(x)$ , где  $F(x)$  имеет разложение (1.1.2), может быть определено следующим образом:

$$P(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} \psi_{mk} M(x), \quad (1.5.1)$$

Из свойства 1.1.2 следует, что умножение кода некоторой функции на вес  $\psi_{mk}$   $mk$ -разряда соответствует  $mk$ -сдвигу кода функции. Умножение кода функции на целое (положительное или отрицательное) число  $\alpha_{mk}$  состоит из нескольких операций алгебраического сложения. Таким образом, если функции представлены своими  $R$ -ыми треугольными кодами, то вычисление по формуле (1.5.1) состоит лишь из операций  $mk$ -сдвига и алгебраического сложения кодов. При этом обход  $mk$ -разрядов множителя  $F(x)$  может производиться в любой последовательности.

Итак, для  $R$ -ых треугольных кодов функций соблюдается общий для всех позиционных кодов принцип выполнения умножения. Заметим в заключение, что, согласно (1.3.1), умножение на параметр  $R$  описывается формулой

$$R \cdot F(x) = \psi_{01} F(x) + \psi_{11} F(x),$$

то есть состоит из двух сдвигов и сложения.



# 1.6. Кодирование и декодирование треугольных кодов

Вначале сформулируем без доказательств две теоремы.

**Теорема 1.6.1.** Функция, имеющая разложение вида (1.1.1), представима также разложением вида

$$F(x) = \sum_{v=0}^n A_v \psi_{0v}, \quad (1.6.1)$$

где

$$A_v = \sum_{k=v}^n \sum_{m=k-v}^k R^{k-v} (-1)^{m+k+v} \alpha_{mk} C_m^{k-v} \quad (1.6.2)$$

**Теорема 1.6.2.** Функция, представленная в виде (1.6.1), где

$$A_v = \sum_{i=0}^j R^i \beta_{iv}, \quad \beta_{iv} \in D_R, \quad (1.6.3)$$

представима также разложением вида

$$F(x) = \sum_{i=0}^j \sum_{v=0}^n R^i \psi_{ov} \beta_{iv}, \quad (1.6.4)$$

или

$$F(x) = \sum_{i=0}^j R^i f_i(x), \quad (1.6.5)$$

где

$$f_i(x) = \sum_{v=0}^n \beta_{iv} \psi_{0v}. \quad (1.6.6)$$

**Определение 1.6.1.** Прямоугольная матрица, составленная из величин  $\beta_{iv} \in D_R$  разложения (1.6.4) функции  $F(x)$  таким образом, что каждая величина  $\beta_{iv}$  принадлежит  $v$ -столбцу и  $i$ -строке этой матрицы, называется  $R$ -ым прямоугольным кодом функции  $F(x)$  по основанию  $u$  и обозначается как  $\text{PK}_R(F(x))$

	$\text{K}_R(A_v):$					
	$\beta_{j0}$	$\beta_{j1}$	...	$\beta_{jv}$	...	$\beta_{jv}$
	...	...	...	...	...	...
$\text{LK}_R(f_i(x)):$	$\beta_{i1}$	$\beta_{i1}$	...	$\beta_{iv}$	...	$\beta_{in}$
	...	...	...	...	...	...
$\text{PK}_R(F(x))=$	$\beta_{00}$	$\beta_{01}$	...	$\beta_{0v}$	...	$\beta_{0n}$

Отметим некоторые свойства кодов  $\text{PK}_R(F(x))$ .

**Свойство 1.6.1.** В  $v$ -столбце кода  $\text{PK}_R(F(x))$  расположен код  $\text{K}_R(A_v) = \beta_{0v} \dots \beta_{iv} \dots \beta_{jv}$  числа  $A_v$  по основанию  $R$ .

**Свойство 1.6.2.** В  $i$ -строке кода  $\text{PK}_R(F(x))$  расположен код  $\text{LK}_R(f_i(x)) = \beta_{i0} \dots \beta_{iv} \dots \beta_{in}$  функции  $f_i(x)$ , который будем называть  $R$ -ым линейным кодом функции  $f_i(x)$ . Отметим, что выражения  $\text{K}_R(A_v)$  и  $\text{LK}_R(f_i(x))$ , несмотря на внешнее сходство, имеют принципиальное различие, так как вес  $i$ -го разряда первого кода равен числу  $R^i$ , а вес  $v$ -го разряда второго кода равен функции  $\psi_{0v}$ .

**Свойство 1.6.3.** Вес  $iv$ -разряда  $\beta_{iv}$  кода  $\text{PK}_R(F(x))$  есть функция  $R^i \psi_{0v}$ .

**Свойство 1.6.4.** Сдвиг кода  $\text{PK}_R(F(x))$  на одну строку вверх/вниз эквивалентен умножению/делению функции  $F(x)$  на  $R$ .

Обозначим: 
$$F_i(x) = \sum_{e=i}^j R^{e-i} f_e(x).$$

Тогда имеем: 
$$F_0(x) = F(x), \quad F_j(x) = f_j(x),$$

$$F_{i-1}(x) = RF_i(x) + f_{i-1}(x), \quad (1.6.7)$$

$$\frac{1}{R} F_{i-1}(x) = F_i(x) + \frac{1}{R} f_{i-1}(x). \quad (1.6.8)$$

Заметим, что структура смешанных кодов допускает выполнение с ними операций алгебраического сложения. Действительно, при выполнении этих операций код  $\text{SK}_R$  может рассматриваться как совокупность независимых позиционных кодов действительных чисел (разряды кода  $\text{LK}_R$ ) и кода  $\text{TK}_R$ . Взаимодействие между этими кодами возникает только при возникновении переносов из строки  $\text{LK}_R$  в 0-строку. Однако переносы между составляющими кодами имеют тот же характер, что и переносы внутри этих кодов. Поэтому такое взаимодействие происходит по обычным правилам распространения переносов в соответствующих кодах.

На использовании этих свойств основаны алгоритмы преобразования прямоугольных кодов  $\text{PK}_R(F(x))$  в треугольные коды  $\text{TK}_R(F(x))$  и обратно. Действительно, каждая строка  $\text{LK}_R(f_i(x))$  кода  $\text{PK}_R(F(x))$  может рассматриваться как код

$\text{TK}_R(f_i(x))$ , содержащий нулевые разряды во всех строках, за исключением нулевой, которая совпадает с  $\text{ЛК}_R(f_i(x))$ . При этом рекуррентная формула (1.6.7) может рассматриваться как формула, содержащая операции умножения на параметр и сложения кодов  $\text{TK}_R$ . Ее  $(j+1)$ -кратное применение позволяет вычислить искомый код  $\text{TK}_R(F(x))$  при известном коде  $\text{PK}_R(F(x))$ . С другой стороны, формула (1.6.8) описывает операцию деления на параметр кода  $\text{TK}_R(F_{i-1}(x))$ , в результате чего образуется код частного  $\text{TK}_R(F_i(x))$  и код остатка  $\text{TK}_R(f_{i-1}(x))$ . Последовательное  $(j+1)$ -кратное деление исходного кода  $\text{TK}_R(F(x))$  на параметр дает в результате код  $\text{PK}_R(F(x))$ .

Таким образом, при известных позиционных кодах чисел  $A_v$  разложения (1.1.2) функции  $F(x)$  может быть (в силу свойства 1.6.1 кодов  $\text{PK}_R$ ) построен код  $\text{PK}_R(F(x))$ , который преобразуется в код  $\text{TK}_R(F(x))$ . Обратное преобразование позволяет вычислить позиционные коды чисел  $A_v$ . Наконец, связь чисел  $A_v$  и коэффициентов степенного ряда функции  $F(x)$  по переменной  $y$  очевидна. Следовательно, любой полином вида

$$F(x) = \sum_{v=0}^n A_v y^v, \quad (1.6.9)$$

коэффициенты  $A_v$  которого суть целые и кратные  $R^v$  действительные числа, представим кодом  $\text{TK}_R(F(x))$ .

# 1.7. Дифференцирование треугольных кодов

Производная функции (1.1.1) равна  $\frac{d}{dx}F(x) = \frac{\partial}{\partial y}F(x)\frac{dy}{dx}$ .

Поэтому дифференцирование треугольных кодов функции  $F(x)$  заключается в определении треугольного кода частной производной  $\frac{\partial}{\partial y}F(x)$  и умножении его на известный треугольный

код производной  $\frac{dy}{dx}$ . Рассмотрим определение треугольного кода

частной производной  $\frac{\partial}{\partial y}F(x)$ . Оно основано на очевидном

соотношении

$$\frac{\partial}{\partial x}\psi_{mk} = R(k-m)\psi_{m,k-1} - Rm\psi_{m-1,k-1}. \quad (1.7.1)$$

В соответствии со свойством 1.1.5 треугольных кодов имеем:

$$R\psi_{m-1,k-1} = \psi_{m-1,k} + \psi_{m,k}. \quad (1.7.2)$$

Из (1.7.1) и (1.7.2) находим:

$$\frac{\partial}{\partial x}\psi_{mk} = -m\psi_{m-1,k} + (k-2m)\psi_{m,k} + (k-m)\psi_{m+1,k}. \quad (1.7.3)$$

Из (1.7.1) получаем:

$$\frac{\partial}{\partial x} \begin{Bmatrix} 0 \\ 0 & \alpha_{mk} \\ 0 & 0 & 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ R(k-m)\alpha_{mk} & 0 \\ 0 & -Rm\alpha_{mk} & 0 \end{Bmatrix}. \quad (1.7.4)$$

Из (1.7.3) получаем:

$$\frac{\partial}{\partial x} \begin{Bmatrix} 0 \\ 0 & \alpha_{mk} \\ 0 & 0 & 0 \end{Bmatrix} = \begin{Bmatrix} (k-m)\alpha_{mk} \\ 0 & (k-2m)\alpha_{mk} \\ 0 & 0 & -m\alpha_{mk} \end{Bmatrix}. \quad (1.7.5)$$

Аппаратный способ дифференцирования заключается в схемной организации переносов из  $mk$ -разряда в  $(m+1, k)$ -разряд и в  $(m-1, k)$ -разряд. Величина этих переносов определяется согласно (1.7.5), а их суммирование в данном разряде производится по тем же правилам, что и в коротких операциях.

## 1.8. Ступенчатые коды

Рассмотрим еще один способ позиционного кодирования функций, являющийся обобщением вышеизложенного способа “треугольного кодирования”. Предлагаемый способ позволяет использовать в качестве исходных более сложные разложения кодируемых функций по некоторому основанию  $u$ , в которые  $u$  входит в дробных и отрицательных степенях. Рассмотрим выражение следующего вида:

$$F(x) = \sum_{k=0}^n \sum_{m=0}^{kr} \alpha_{mk} R^k u^{k-m} u^m \quad (1.8.1)$$

где  $\alpha_{mk}$  - действительные числа,  
 $R$  - целое положительное число,  
 $y(x)$ ,  $u(x)$  - функции аргумента  $x$ ,  
 $m$ ,  $k$ ,  $r$ ,  $n$  - целые положительные числа или нули.

**Определение 1.8.1.** Двойная сумма вида (1.8.1), в которой функции  $y$  и  $u$  связаны зависимостью

$$y \sum_{i=0}^r u^i = 1, \quad (1.8.2)$$

называется ступенчатым разложением порядка  $r$  функции  $F(x)$  по основанию  $u$  с параметром  $R$ .

Нетрудно заметить аналогию между разложениями (1.1.1) и (1.8.1). Более того, при  $r=1$  разложение (1.8.1) превращается в разложение (1.1.1), ибо в этом случае  $u=(1-y)/y$  и  $y^r u^m = y^{k-m}(1-y)^m$ . При  $r>1$  зависимость между  $y$  и  $u$  оказывается более сложной. В частности, при  $r=2$

$$u = \frac{-1}{2} \pm \sqrt{\frac{1}{y} - \frac{3}{4}}.$$

По аналогии с определением 1.1.1 для ступенчатого разложения (1.8.1) также вводятся понятия  $mk$ -разряда, веса  $mk$  - разряда, величины  $mk$  -разряда и, наконец, ступенчатого кода функции  $F(x)$  – **STK**( $F(x)$ ). Рассмотрим в качестве иллюстрации выражение (1.8.1) при  $r=2$  в развернутом виде:

$$\begin{array}{cccc}
 & & \alpha_{42} & R^2 y^2 u^4 \\
 & & \alpha_{32} & R^2 y^2 u^3 \\
 \alpha_{21} & R y u^2 & \alpha_{22} & R^2 y^2 u^2 \\
 \alpha_{11} & R y u & \alpha_{12} & R^2 y^2 u \\
 \text{STK} = \alpha_{00} & \alpha_{01} & R y & \alpha_{02} & R^2 y^2
 \end{array}$$

Продолжая эту аналогию, отметим, что среди ступенчатых кодов можно выделить  $R$ -ые ступенчатые коды функций  $\text{STK}_R(F(x))$ , у которых разряды  $\alpha_{mk}$  принимают значения из множества  $D_R$ . Для кодов  $\text{STK}_R$  существуют правила выполнения арифметических операций, аналогичные тем, которые описаны для кодов  $\text{TK}_R$ . Основное отличие связано с тем, что выработка переноса при операциях с кодами  $\text{STK}_R$  основывается на соотношении

$$\text{STK}_R(R) = \begin{array}{c} 1 \\ 1 \\ \dots \\ 1 \\ 0 \end{array} \left. \vphantom{\begin{array}{c} 1 \\ 1 \\ \dots \\ 1 \\ 0 \end{array}} \right\} (r+1)$$

то есть перенос распространяется в  $(r+1)$  разряд старшего столбца. Поэтому в данный разряд могут поступить переносы одновременно из  $(r+1)$  разрядов. Таким образом, для ступенчатых кодов должно выполняться условие  $R > 2r$ .

## Глава 2. Кодирование тригонометрических рядов

### 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

Рассмотренные выше свойства треугольных кодов не зависят от вида основания. В частности, в качестве основания может быть взят сам аргумент:  $y=x$ . Однако для приложений необходимо выбирать такие основания, которые позволяют кодировать функции наиболее общего вида. Поэтому ниже рассматриваются коды по основанию  $y = \text{Sin}^2(x)$ , которыми могут быть представлены тригонометрические ряды и, в частности, ряды Фурье. Кроме того, коды тригонометрических рядов обладают (как будет видно) рядом достоинств по сравнению с общим случаем кодирования функций.

Далее обозначено:  $i = \{0, 1, 2, 3\}$  - верхний индекс соответствующей величины,  $C_b^a$  - число сочетаний из  $b$  по  $a$ .

Предварительно сформулируем ряд теорем, опуская их доказательства для ограничения объема книги. Заметим только, что эти доказательства сводятся к достаточно элементарным, но громоздким преобразованиям.

**Теорема 2.1.1.** Числа  $S^i(v, n)$  определены по формулам табл. 2.1.1 и при  $v > 0$  связаны следующими рекуррентными зависимостями:

$$S^i(v, v+1) = 2 + S^i(v-1, v);$$

$$S^i(v, n) = 2S^i(v, n-1) - S^i(v, n-2) + S^i(v-1, n-1), \quad n > v+1.$$

## 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

Таблица 2.1.1.

$i$	$S^i(v, \omega)$	$S^i(0, \omega), \omega > 0$
0	$\frac{2\omega}{\omega+v} C_{\omega+v}^{2v} = C_{\omega+v}^{2v} + C_{\omega+v-1}^{2v}$	2
1	$\frac{2\omega+1}{\omega+v+1} C_{\omega+v+1}^{2v+1} = C_{\omega+v+1}^{2v+1} + C_{\omega+v}^{2v+1}$	$2\omega+1$
2	$C_{\omega+v}^{2v}$	1
3	$C_{\omega+v+1}^{2v+1}$	$\omega+1$

В соответствии с этим числа  $S^i(v, \omega)$  могут быть вычислены по следующей схеме:

	n-2	n-1	n
v-1		$+S^i(v-1, n-1)$	
v	$-S^i(v, n-2)$	$+2S^i(v, n-1)$	$S^i(v, n) =$

В таблицах 2.1.2. $i$  приведены значения чисел  $S$  при  $i=(0,1,2,3)$  соответственно и при  $v < 5$  и  $n < 5$ .

Таблица 2.1.2.0. Числа  $S^0(v, n)$ .

$v \backslash n$	0	1	2	3	4
0	1	2	2	2	2
1		1	4	9	16
2			1	6	20
3				1	8
4					1

Таблица 2.1.2.1. Числа  $S^1(v, n)$ .

$v \backslash n$	0	1	2	3	4
0	1	3	5	7	2
1		1	5	14	16
2			1	7	20
3				1	8
4					1



Таблица 2.1.2.2. Числа  $S^2(v, n)$ .

$v \setminus n$	0	1	2	3	4
0	1	1	1	1	1
1		1	3	6	10
2			1	5	15
3				1	7
4					1

Таблица 2.1.2.3. Числа  $S^3(v, n)$ .

$v \setminus n$	0	1	2	3	4
0	1	2	3	4	5
1		1	4	10	20
2			1	6	21
3				1	8
4					1

**Теорема 2.1.2.** Числа  $L^i(v, n)$  определены по формулам табл. 2.1.3 и связаны следующими рекуррентными зависимостями:

$$L^i(v, v) = 1, \quad v \geq 0;$$

$$L^i(v, 0) = L^i(v-1, 0)l^i(v), \quad v \geq 1;$$

$$L^i(v, v-1) = 2L^i(v-1, v-2), \quad v \geq 2;$$

$$L^i(v, n) = L^i(v-1, n-1) + 2L^i(v-1, n) + L^i(v-1, n+1),$$

$$v-2 \geq n > 0, \quad v \geq 3.$$

Таблица 2.1.3.

$i$	$L^i(v, \omega)$	$l^i(v)$
0	$C_{2v}^{v-\omega}$	$2(2v-1)/v$
1	$C_{2v+1}^{v-\omega}$	$2(2v+1)/(v+1)$
2	$C_{2v}^{v-\omega} - C_{2v}^{v-\omega-1}$	$2(2v-1)/(v+1)$
3	$C_{2v}^{v-\omega} - C_{2v}^{v-\omega-2}$	$2(2v+1)/(v+2)$

В соответствии с этим числа  $L^i(v, n)$  могут быть вычислены по следующей схеме:

## 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

	n-1	n	n+1
v-1	$+L^i(v-1, n-1)$	$+2L^i(v-1, n)$	$+L^i(v-1, n+1)$
v		$L^i(v, n) =$	

В таблицах 2.1.4. $\dot{i}$  приведены значения чисел  $L$  при  $\dot{i}=(0,1,2,3)$  соответственно и при  $v < 5$  и  $n < 5$ .

Таблица 2.1.4.0. Числа  $L^0(v, n)$ .

v \ n	0	1	2	3	4
0	1				
1	2	1			
2	6	4	1		
3	20	15	6	1	
4	70	56	28	8	1

Таблица 2.1.4.1. Числа  $L^1(v, n)$ .

v \ n	0	1	2	3	4
0	1				
1	3	1			
2	10	5	1		
3	35	21	7	1	
4	126	84	36	9	1

Таблица 2.1.4.2. Числа  $L^2(v, n)$ .

v \ n	0	1	2	3	4
0	1				
1	1	1			
2	2	3	1		
3	5	9	5	1	
4	14	28	20	7	1

Таблица 2.1.4.3. Числа  $L^3(v, n)$ .

v \ n	0	1	2	3	4
0	1				
1	2	1			
2	5	4	1		
3	14	14	6	1	
4	42	48	27	8	1

**Теорема 2.1.3.** Функции  $\gamma_n^i$ , определенные по формулам

табл. 2.1.5, имеют разложение по основанию  $y = \text{Sin}^2(x)$  в виде

$$\gamma_n^i = \sum_{v=0}^n (-1)^v \left(\frac{R}{4}\right)^{n-v} S^i(v, n) \cdot \psi_{ov} \quad (2.1.1)$$

и при  $n > 1$  связаны рекуррентной зависимостью

$$\gamma_n^i = 2\text{Cos}(2x)\gamma_{n-1}^i - \left(\gamma_{n-2}^i b^i\right) / a_{n-2}^i \quad (2.1.2)$$

Напомним, что  $\psi_{ov} = y^v = \text{Sin}^{2v}(x)$ .

Таблица 2.1.5.

$i$	$\left(\frac{4}{R}\right)^\omega \gamma_\omega^i$	$\gamma_0^i$	$\left(\frac{4}{R}\right)\gamma_1^i$	$a_\omega^i$	$b^i$
0	$2\text{Cos}(2\omega x)$	1	$2-4\text{Sin}^2 x$	1 ( $\omega=0$ ) 2 ( $\omega>0$ )	2
1	$\frac{\text{Sin}(2\omega+1)x}{\text{Sin}(x)}$	1	$3-4\text{Sin}^2 x$	1	1
2	$\frac{\text{Cos}((2\omega+1)x)}{\text{Cos}(x)}$	1	$1-4\text{Sin}^2 x$	1	1
3	$\frac{\text{Sin}(2(\omega+1)x)}{\text{Sin}(2x)}$	1	$2-4\text{Sin}^2 x$	1	1

Рассмотрим теперь коды  $\text{TK}(\gamma_\omega^i)$  по основанию  $\text{Sin}^2 x$ . В соответствии с табл. 2.1.5 и формулой (2.1.2), а также учитывая, что,

$2\text{Cos}(2x) = \gamma_1^i$  находим:

$$\begin{aligned} \gamma_2^0 &= \gamma_1^0 \gamma_1^0 - 2\gamma_0^0, & \gamma_2^i &= \gamma_1^0 \gamma_1^i - \gamma_0^i, \\ \gamma_n^0 &= \gamma_1^0 \gamma_{n-1}^0 - 4\gamma_{n-2}^0, & \gamma_n^i &= \gamma_1^0 \gamma_{n-1}^i - 2\gamma_{n-2}^i. \end{aligned}$$

Табл. 2.1.6 кодов  $\text{TK}(\gamma_\omega^i)$  для начальных значений  $\omega$  приведена ниже.

## 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

Таблица 2.1.6.

$i$	$\text{TK}(\gamma_0^i)$	$\text{TK}\left(\left(\frac{4}{R}\right)\gamma_1^i\right)$		$\text{TK}\left(\left(\frac{4}{R}\right)\gamma_2^i\right)$		
0	1	2	-4/R	2	-16/R	$16/R^2$
1	1	3	-4/R	8	-24/R	$16/R^2$
2	1	1	-4/R	0	-8/R	$16/R^2$
3	1	2	-4/R	3	-16/R	$16/R^2$

**Теорема 2.1.4.** Функции  $\lambda_\omega^i$ ,  $\varepsilon_\omega^i$ ,  $\eta^i$  определены по формулам табл. 2.1.7 и связаны следующими соотношениями:

$$\eta^i \gamma_\omega^i = \left(\frac{R}{4}\right)^\omega a_\omega^i c^i \varepsilon_\omega^i \quad (2.1.3)$$

$$\lambda_\omega^i = \left(\frac{R}{4}\right)^\omega \eta^i \gamma_\omega^i \quad (2.1.4)$$

$$\lambda_\omega^i = a_\omega^i c^i \varepsilon_\omega^i \quad (2.1.5)$$

Таблица 2.1.7.

$i$	$\varepsilon_\omega^i$	$\lambda_\omega^i$	$c^i$	$\eta^i$
0	$\text{Cos}2\omega x$	$2\text{Cos}2\omega x$	1	1
1	$\text{Sin}(2\omega + 1)x$	$\sqrt{R}\text{Sin}(2\omega + 1)x$	$\sqrt{R}$	$\sqrt{R}\text{Sin}x$
2	$\text{Cos}(2\omega + 1)x$	$\sqrt{R}\text{Cos}(2\omega + 1)x$	$\sqrt{R}$	$\sqrt{R}\text{Cos}x$
3	$\text{Sin}(2(\omega + 1))x$	$R/2 \text{Sin}(2(\omega + 1))x$	$R/2$	$\frac{R}{2}\text{Sin}2x$

Рассмотрим теперь разложения функций по основанию  $y = \text{Sin}^2(x)$  с весом  $m$ -разряда, соответственно равным

$$\psi_{mk} = R^k [\text{Sin}^2(x)]^{k-m} [\text{Cos}^2(x)]^m. \quad (2.1.6)$$

В частности,

$$\psi_{ok} = R^k \text{Sin}^{2k}(x). \quad (2.1.7)$$

Далее мы будем часто оперировать функциональными рядами вида

$$F^i(x) = \sum_{\omega=0}^n D_\omega^i \varepsilon_\omega^i, \quad (2.1.8)$$

$$F^i(x) = \eta^i \sum_{v=0}^n A_v \psi_{0v}, \quad (2.1.9)$$

$$F^i(x) = \sum_{\omega=0}^n E_{\omega}^i \gamma_{\omega}^i, \quad (2.1.10)$$

$$F^i(x) = \sum_{\omega=0}^n H_{\omega}^i \lambda_{\omega}^i, \quad (2.1.11)$$

где  $D_{\omega}^i$ ,  $A_v$ ,  $E_{\omega}^i$ ,  $H_{\omega}^i$  - коэффициенты рядов, действительные числа. Далее утверждается, что тригонометрический ряд общего вида может быть представлен четырьмя составляющими – частичными рядами функций  $F^i(x)$  по функциям  $\mathcal{E}_{\omega}^i$ , а каждый из этих частичных рядов может быть преобразован в ряд по функциям  $\lambda_n^i$ ,  $\gamma_n^i$ ,  $\psi_{0v}$  и, далее, каждый из этих рядов представим ТТК по основанию  $y = \text{Sin}^2(x)$  с весом разряда в виде (2.1.6)

**Теорема 2.1.5.** Функция  $F^i(x)$ , представленная одним из функциональных рядов вида (2.1.8-11), разложима также в любой из этих рядов и их коэффициенты связаны следующими соотношениями:

$$H_{\omega}^i = \sum_{v=\omega}^n (R/4)^v A_v L^i(v, \omega), \quad (2.1.12)$$

$$E_{\omega}^i = (4/R)^{\omega} H_{\omega}^i, \quad (2.1.13)$$

$$E_{\omega}^i = (4/R)^{\omega} \sum_{v=\omega}^n (R/4)^v A_v L^i(v, \omega), \quad (2.1.14)$$

$$D_{\omega}^i = a_{\omega}^i c^i (-1)^{\omega} H_{\omega}^i, \quad (2.1.15)$$

$$A_v = (-1)^v (4/R)^v \sum_{v=\omega}^n S^i(v, \omega) H_{\omega}^i (-1)^{\omega}, \quad (2.1.16)$$

$$D_{\omega}^i = a_{\omega}^i c^i (-1)^{\omega} \sum_{v=\omega}^n (R/4)^v A_v L^i(v, \omega), \quad (2.1.17)$$

$$A_v = \frac{(-1)^v}{c^i} (4/R)^v \sum_{v=\omega}^n S^i(v, \omega) \frac{D_{\omega}^i}{a_{\omega}^i} \quad (2.1.18)$$

$$E_{\omega}^i = \frac{(-1)^{\omega}}{a_{\omega}^i c^i} \left( \frac{4}{R} \right)^{\omega} D_{\omega}^i \quad (2.1.19)$$

Из этих формул следует, что

$$\left\| \begin{array}{l} \text{функция } F^i(x) \text{ имеет равное количество} \\ \text{членов во всех разложениях вида (2.1.8-11)} \end{array} \right\| \quad (2.1.20)$$

## 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

Из (2.1.17) в частности следует, что

$$\eta^i \psi_{on} = \sum_{\omega=0}^n D_{\omega}^i \varepsilon_{\omega}^i, \quad (2.1.21)$$

где

$$D_{\omega}^i = a_{\omega}^i c^i (-1)^{\omega} (R/4)^n L^i(n, \omega). \quad (2.1.22)$$

В частности, последний член ряда (2.1.21)

$$g_n^i = (-1)^n (R/4)^n a_n^i c^i \varepsilon_n^i. \quad (2.1.23)$$

### **Пример 2.1.1. Преобразование частичного ряда при $i=3$ .**

Рассмотрим функцию

$$\Phi_1(x) = 4\text{Sin}x\text{Cos}x \cdot \left\{ \begin{array}{l} -3408 + 13538\psi_{01} - 16776\psi_{02} + 11056\psi_{03} \\ -4310\psi_{04} + 1032\psi_{05} - 140\psi_{06} + 8\psi_{07} \end{array} \right\}$$

Найдем для этой же функции коэффициенты ряда (2.1.8) по формуле (2.1.17) – в верхней строке таблицы указан коэффициент  $A_{\nu}$ , во втором столбце таблицы указан коэффициент  $a_{\omega}^i c^i (-1)^{\omega}$ , а в таблице приведены коэффициенты  $L^i(\nu, \omega)$ :

$\omega$		-3408	13538	-16776	11056	-4310	1032	-140	8	$D_{\omega}^3$
0	2*	1	2	5	14	42	132	429	1430	2312
1	-2*		1	4	14	48	165	572	2002	-1108
2	2*			1	16	27	110	429	1638	-492
3	-2*				1	8	44	208	910	-288
4	2*					1	10	65	350	-580
5	-2*						1	12	90	-144
6	2*							1	14	-56
7	-2*								1	-16

Таким образом,

$$\Phi_1(x) = \left\{ \begin{array}{l} 2312\text{Sin}2x - 1108\text{Sin}4x - 492\text{Sin}6x - \\ 288\text{Sin}8x - 580\text{Sin}10x - 144\text{Sin}12 - \\ 56\text{Sin}14x - 16\text{Sin}16x \end{array} \right\}$$

Далее будем применять следующее обозначение:

$[F \parallel h_k] = \{h_0, h_1, \dots, h_k, \dots\}$  – множество коэффициентов  $h_k$  функционального ряда для функции  $F$ . Таким образом, для

некоторая функция  $F$  может быть представлена несколькими множествами коэффициентов:

$$[F \parallel D_{\omega}^i], [F \parallel A_{\nu}], [F \parallel E_{\omega}^i], [F \parallel H_{\omega}^i].$$

**Пример 2.1.2. Преобразование частичного ряда функции  $\gamma$ .**

Построим функциональный ряд (2.1.9) для функции  $F(x) = \gamma_3^0(x)$  при  $R=4$ . В соответствии с (2.1.1) находим:

$$\gamma_3^0 = \begin{bmatrix} S^0(0,3)\psi_{00} - S^0(1,3)\psi_{01} - S^0(2,3)\psi_{02} - S^0(3,3)\psi_{03} \\ 2\psi_{00} - 9\psi_{01} + 6\psi_{02} - \psi_{03} \end{bmatrix}$$

В соответствии с (2.1.9) при  $\eta^0 = 1$  имеем:

$$\left[ \gamma_3^0 \parallel A_{\nu} \right] = \{+2, -9, +6, -1\}.$$

Применяя формулу (2.1.12) для функции  $H_{\omega}^0$ , найдем (в каждом слагаемом первый сомножитель - число  $A_{\nu}$ , а второй сомножитель - число  $L^i(\nu, \omega)$ ):

$\nu$	0	1	2	3	
$H_0^0$	+2*1	-9*2	+6*6	-1*20	=0
$H_1^0$	0	-9*1	+6*4	-1*15	=0
$H_2^0$			+6*1	-1*6	=0
$H_3^0$				-1*1	=-1

Таким образом,  $\left[ \gamma_3^0 \parallel H_{\omega}^0 \right] = \{0,0,0,-1\}$ .

При известных  $H_{\omega}^0$  вновь по формуле (2.1.16) найдем числа  $A_{\nu}$  для функции  $\gamma_3^0$  (в каждом слагаемом первый сомножитель - число  $(H_{\omega}^0(-1)^{\omega})$ , а второй сомножитель - число  $S^i(\nu, \omega)$ ):

Учитывая формулу (2.1.15), находим  $\left[ \gamma_3^0 \parallel D_{\omega}^0 \right] = \{0,0,0,-2\}$

$\omega$	0	1	2	3	
$A_0$	0*1	0*2	0*2	1*2	=2
$-A_1$		0*1	0*4	1*9	=9
$A_2$			0*1	1*6	=6
$-A_3$				1*1	=1

Полученный результат совпадает с приведенным в начале примера.

В следующих примерах приняты следующие обозначения:

$$d = \sum A_\nu L^i(\nu, \omega), \quad D_\omega^i = (-1)^\omega a_\omega^i c^i d,$$

$$z = \sum D_\omega^i \frac{S^i(\nu, \omega)}{a_\omega^i}, \quad A_\nu = z \frac{(-1)^\nu}{c^i}.$$

**Пример 2.1.3а. Преобразование частичного ряда при  $i=0$ .**

Рассмотрим преобразование функции вида (2.1.9)  $f(x) = (-11 + 3\psi_{01} - 2\psi_{02})$  по (2.1.17) при  $R=4, i=0$ . При этом  $c^i = 1, a_{\omega=0}^i = 1, a_{\omega>0}^i = 2$ . В следующей таблице множителями являются числа  $L^i(\nu, \omega)$ .

0	1	2	$d$	$D_\omega^i$
$A_0$	$2 A_1$	$6 A_2$	-17	-17
	$A_1$	$4 A_2$	-5	-10
		$A_2$	-2	-4

Таким образом, данная функция имеет разложение вида (2.1.8):

$$f(x) = -17 - 10\text{Cos}(2x) - 4\text{Cos}(4x).$$

**Пример 2.1.3. Взаимные преобразования частичного ряда при  $i=0$ .**

Рассмотрим преобразования функций (2.1.17) и (2.1.18) при  $R=4, i=0$ . При этом  $c^i = 1, a_{\omega=0}^i = 1, a_{\omega>0}^i = 2$ . В следующей таблице множителями являются числа  $L^i(\nu, \omega)$ .



0	1	2	3	4	$d$	$D_{\omega}^i$
$A_0$	$2 A_1$	$6 A_2$	$20 A_3$	$70 A_4$	32	32
	$A_1$	$4 A_2$	$15 A_3$	$56 A_4$	21	-42
		$A_2$	$6 A_3$	$28 A_4$	-4	-8
			$A_3$	$8 A_4$	2	-4
				$A_4$	-1	-2

Таким образом, данная функция имеет разложение вида (2.1.8):

$$F(x) = 32 - 42\cos 2x - 8\cos 4x - 4\cos 6x - 2\cos 8x.$$

В следующей таблице множителями являются числа

$$S^i(v, \omega) / a_{\omega}^i.$$

0	1	2	3	4	$z$	$A_v$
$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	-24	-24
	$\frac{D_1}{2}$	$2 D_2$	$\frac{9 D_3}{2}$	$8 D_4$	-71	71
		$\frac{D_2}{2}$	$6 D_3$	$10 D_4$	-36	-36
			$\frac{D_3}{2}$	$4 D_4$	-10	10
				$\frac{D_4}{2}$	-1	-1

Таким образом, данная функция имеет разложение вида (2.1.9):

$$F(x) = -24 + 71\psi_{01} - 36\psi_{02} + 10\psi_{03} - \psi_{04}.$$

**Пример 2.1.4. Взаимные преобразования частичного ряда при  $i=1$ .**

Рассмотрим преобразования функций (2.1.17) и (2.1.18) при  $R=4$ ,

$i=1$ . При этом  $c^i = 2$ ,  $a_{\omega}^i = 1$ . В следующей таблице

множителями являются числа  $L^i(v, \omega)$ .

## 2.1. Треугольные коды по основанию $\text{Sin}^2(x)$

0	1	2	$d$	$D_{\omega}^i$
$A_0$	$3 A_1$	$10 A_2$	128	256
	$A_1$	$5 A_2$	-20	10
		$A_2$	4	2

Таким образом, данная функция имеет разложение вида (2.1.8):

$$F'(x) = 256\text{Sin}x + 10\text{Sin}3x + 2\text{Sin}5x.$$

В следующей таблице множителями являются числа

$$S^i(v, \omega) / a_{\omega}^i = S^i(v, \omega).$$

0	1	2	$z$	$A_v$
$D_0$	$3 D_1$	$5 D_2$	296	148
	$D_1$	$5 D_2$	20	-10
		$D_2$	2	1

Таким образом, данная функция имеет разложение вида (2.1.9):

$$F'(x) = 2\text{Sin}x(148 - 10\psi_{01} + \psi_{02}).$$

### **Пример 2.1.5. Взаимные преобразования частичного ряда при $i=2$ .**

Рассмотрим преобразования функций (2.1.17) и (2.1.18) при  $R=4$ ,

$i=2$ . При этом  $c^i = 2$ ,  $a_{\omega}^i = 1$ . В следующей таблице

множителями являются числа  $L^i(v, \omega)$ .

0	1	2	$d$	$D_{\omega}^i$
$A_0$	$A_1$	$A_2$	128	256
	$A_1$	$3 A_2$	5	-10
		$A_2$	1	2

Таким образом, данная функция имеет разложение вида (2.1.8):

$$F''(x) = 256\text{Cos}x - 10\text{Cos}3x + 2\text{Cos}5x.$$

В следующей таблице множителями являются числа  $S^i(v, \omega) / a_\omega^i = S^i(v, \omega)$ .

0	1	2	$z$	$A_v$
$D_0$	$D_1$	$D_2$	248	124
	$D_1$	$3 D_2$	-4	2
		$D_2$	2	1

Таким образом, данная функция имеет разложение вида (2.1.9):

$$F''(x) = 2\text{Cos}x(124 + 2\psi_{01} + \psi_{02}).$$

**Пример 2.1.6. Взаимные преобразования частичного ряда при  $i=3$ .**

Рассмотрим преобразования функций (2.1.17) и (2.1.18) при  $R=4$ ,  $i=3$ . При этом  $c^i = 2$ ,  $a_\omega^i = 1$ . В следующей таблице множителями являются числа  $L^i(v, \omega)$ .

0	1	2	3	$d$	$D_\omega^i$
$A_0$	$2 A_1$	$5 A_2$	$14 A_3$	42	84
	$A_1$	$4 A_2$	$14 A_3$	-16	32
		$A_2$	$6 A_3$	12	24
			$A_3$	-8	16

Таким образом, данная функция имеет разложение вида (2.1.8):

$$F'''(x) = 84 + 32\text{Sin}2x + 24\text{Sin}4x + 16\text{Sin}8x.$$

В следующей таблице множителями являются числа  $S^i(v, \omega) / a_\omega^i = S^i(v, \omega)$ .

0	1	2	3	$z$	$A_v$
$D_0$	$2 D_1$	$3 D_2$	$4 D_3$	284	142
	$D_1$	$4 D_2$	$10 D_3$	288	-144
		$D_2$	$6 D_3$	120	60
			$D_3$	16	-8

Таким образом, данная функция имеет разложение вида (2.1.9):

$$F'''(x) = 4\text{Sin}x\text{Cos}x(142 - 144\psi_{01} + 60\psi_{02} - 8\psi_{03}).$$

## 2.2. Тригонометрические треугольные коды

Вначале сформулируем две теоремы.

**Теорема 2.2.1.** Функция  $\Phi(x)$ , заданная тригонометрическим рядом общего вида

$$\Phi(x) = \sum_{u=0}^U (a_u \text{Cos}(ux) + b_u \text{Sin}(ux)),$$

представима также разложениями вида

$$\Phi(x) = \sum_{i=0}^3 F^i(x), \quad \Phi(x) = \sum_{i=0}^3 \eta^i f^i(x),$$

где функции  $F^i(x) = \eta^i f^i(x)$  имеют разложения вида (2.1.8), где

$$D_{\omega}^i = a_u \quad \text{при} \quad u = 2\omega \quad i = 0,$$

$$D_{\omega}^i = b_u \quad \text{при} \quad u = 2\omega + 1, \quad i = 1,$$

$$D_{\omega}^i = a_u \quad \text{при} \quad u = 2\omega + 1, \quad i = 2,$$

$$D_{\omega}^i = b_u \quad \text{при} \quad u = 2\omega + 2, \quad i = 3,$$

**Теорема 2.2.2.** Функция  $\Phi(x)$ , определенная в соответствии с теоремой 2.2.1, имеет разложение вида

$$\Phi(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} \psi'_{mk} \quad (2.2.1)$$

где  $\psi'_{mk} = \sqrt{\psi_{mk}} = \psi_{m/2, k/2}$  или

$$\psi'_{mk} = R^{k/2} \text{Sin}^{k-m} x \text{Cos}^m x. \quad (2.2.2)$$

Сравнивая формулы (1.1.2) и (2.2.1), замечаем их полную идентичность, откуда следует, что на основе последней формулы также может быть построен треугольный код, который, в отличие от рассмотренного ранее, будем называть тригонометрическим треугольным кодом функции  $\Phi(x)$  и обозначать как  $\text{ТТК}(\Phi(x))$ . В этом случае также вводится понятие  $\text{ТТК}_R(\Phi(x))$ . Для иллюстрации запишем разложение (2.2.1) в развернутом виде:

				$\alpha_{nn} R^{n/2} \text{Cos}^n x$
			...	...
			$\alpha_{kk} R^{k/2} \text{Cos}^k x$	$\alpha_{kn} R^{n/2} \text{Sin}^{n-k} x \text{Cos}^k x$
		...	...	...
	$\alpha_{11} \sqrt{R} \text{Cos} x$	...	$\alpha_{1k} R^{k/2} \text{Sin}^{k-1} x \text{Cos} x$	$\alpha_{1n} R^{n/2} \text{Sin}^{n-1} x \text{Cos} x$
$\alpha_{00}$	$\alpha_{01} \sqrt{R} \text{Sin} x$	...	$\alpha_{ok} R^{k/2} \text{Sin}^k x$	$\alpha_{on} R^{n/2} \text{Sin}^n x$

**Свойство 2.2.1.** Тригонометрические треугольные коды обладают свойствами 1, 2, 3, 4 обычных треугольных кодов.

**Свойство 2.2.2.** Имеет место соотношение

$$R\psi'_{mk} = \psi'_{m,k+2} + \psi'_{m+2,k+2}.$$

**Свойство 2.2.3.** ТТК( $\Phi(x)$ ) является композицией четырех кодов ТТК( $F^i(x)$ ), где функции  $F^i(x)$  удовлетворяют условиям теоремы 2.2.1; на схеме цифрами 0, 1, 2, 3 обозначены разряды кодов при  $i = 0, 1, 2, 3$  соответственно:

$$\begin{array}{ccccccc}
 & & & & & & 2 \\
 & & & & & & 0 & 3 \\
 & & & & & & 2 & 1 & 2 \\
 & & & & & & 0 & 3 & 0 & 3 \\
 & & & & & & 2 & 1 & 2 & 1 & 2 \\
 & & & & & & 0 & 3 & 0 & 3 & 0 & 3
 \end{array}$$

**Свойство 2.2.4.** Каждая из функций  $\eta^i$  имеет ТТК $_R(\eta^i(x))$  - см. теорему 2.1.4 и табл. 2.2.1. Умножение кода ТТК $_R$  некоторой функции на код ТТК $_R(\eta^i(x))$  равносильно сдвигу кода этой функции, точнее, 00-, 01-, 11-, 12- сдвигу при  $i = 0, 1, 2, 3$  соответственно.

Таблица 2.2.1.

$i$	0	1	2	3
$\eta^i$	1	$\sqrt{R} \text{Sin} x$	$\sqrt{R} \text{Cos} x$	$\frac{R}{2} \text{Sin} 2x$
ТТК $_R(\eta^i)$	0 0 0 1 0 0	0 0 0 0 1 0	0 1 0 0 0 0	0 0 1 0 0 0

Таким образом,

$$\psi'_{ab} = \left\{ \begin{array}{l} \eta^0 \psi_{mk}, \text{ if } a = 2m, b = 2k; \\ \eta^1 \psi_{mk}, \text{ if } a = 2m, b = 2k + 1; \\ \eta^2 \psi_{mk}, \text{ if } a = 2m + 1, b = 2k + 1; \\ \eta^3 \psi_{mk}, \text{ if } a = 2m + 1, b = 2k + 2; \end{array} \right. \quad (2.2.3)$$

**Определение 2.2.1.** *Расширением* кода **ТК** в код **ТТК** называется перестановка *mk*-разрядов исходного кода **ТК** на место  $(2m, 2k)$ -разрядов результирующего кода **ТТК**. Обратное преобразование называется *сжатием*.

Например, если  $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ , то  $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ .  
 $\text{ТК}_R = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$        $\text{ТТК}_R = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$

Каждая из функций  $f^i(x)$  может быть представлена кодом  $\text{ТТК}(f^i(x))$  по основанию  $y = \text{Sin}^2(x)$ , что следует из предыдущих теорем. Расширением каждый из этих кодов может быть преобразован в тригонометрический код  $\text{ТТК}(f^i(x))$ , вес *mk*-разряда которого имеет вид (2.2.2). Из сказанного, а также из теоремы 2.2.1 и свойств 2.2.3 и 2.2.4 следует, что код  $\text{ТТК}(\Phi(x))$  может быть получен расширением, сдвигом и совмещением кодов  $\text{ТТК}(f^i(x))$ . Такое преобразование будем называть *компоновкой*  $\langle \text{ТТК}(f^i(x)) \rangle \dots \langle \text{ТТК}(\Phi(x)) \rangle$ , а обратное ему - *декомпоновкой*  $\langle \text{ТТК}(\Phi(x)) \rangle \dots \langle \text{ТТК}(f^i(x)) \rangle$ .

**Пример 2.2.1. Расширение, сдвиг и компоновка кодов.**

Пусть  $R = 4$ ,  $\alpha \in \{-2, -1, 0, 1\}$ ,

$$\begin{matrix} & & & & & & & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 & 0 \\ & & & & & & & 0 & 1 & 0 & -1 \\ & & & & & & & 1 & -1 & 0 & 0 & 0 \\ & & & & & & & 1 & -1 & -1 & 0 & 0 & 0 \\ \text{ТТК}(\Phi(x)) = -2 & & & & & & & 1 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

Выделяя из этого кода составляющие коды и сжимая их, то есть производя декомпоновку, получаем:

$$\begin{aligned} \text{TK}_4(f^0(x)) &= \begin{pmatrix} 1 & -1 \\ -2 & 1 \end{pmatrix}, & \text{TK}_4(f^1(x)) &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \\ \text{TK}_4(f^2(x)) &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}, & \text{TK}_4(f^3(x)) &= \begin{pmatrix} -1 & -1 \\ 0 & 0 \end{pmatrix} \end{aligned}$$

Расширение, сдвиг и совмещение, то есть компоновка, дает в результате снова исходный код.

**Пример 2.2.2. Кодирование функции.** Построим код функции  $\Phi(x) = F(x) + F'(x) + F''(x) + F'''(x)$ , где слагаемые функции определены в примерах 2.1.3-6:

$$F(x) = 32 - 42\text{Cos}2x - 8\text{Cos}4x - 4\text{Cos}6x - 2\text{Cos}8x,$$

$$F'(x) = 256\text{Sin}x + 10\text{Sin}3x + 2\text{Sin}5x,$$

$$F''(x) = 256\text{Cos}x - 10\text{Cos}3x + 2\text{Cos}5x,$$

$$F'''(x) = 84 + 32\text{Sin}2x + 24\text{Sin}4x + 16\text{Sin}8x,$$

или

$$F(x) = -24 + 71\psi_{01} - 36\psi_{02} + 10\psi_{03} - \psi_{04},$$

$$F'(x) = 2\text{Sin}x(148 - 10\psi_{01} + \psi_{02}),$$

$$F''(x) = 2\text{Cos}x(124 + 2\psi_{01} + \psi_{02}),$$

$$F'''(x) = 4\text{Sin}x\text{Cos}x(142 - 144\psi_{01} + 60\psi_{02} - 8\psi_{03}).$$

Следовательно,

$\text{ТТК}F(x) =$		0	0	0	0	0	0	0	0
	-24	0	71	0	-36	0	10	0	-1
$\text{ТТК}F'(x) =$		0	0	0	0	0	0	0	0
	0	148	0	-10	0	1	0	0	0
$\text{ТТК}F''(x) =$		124	0	2	0	1	0	0	0
	0	0	0	0	0	0	0	0	0
$\text{ТТК}F'''(x) =$		0	142	0	-144	0	60	0	-8
	0	0	0	0	0	0	0	0	0
$\text{ТТК}\Phi(x) =$		124	142	2	-144	1	60	0	-8
	-24	148	71	-10	-36	1	10	0	-1

## 2.3. Операции с тригонометрическими треугольными кодами

**2.3.1. Короткие операции.** Этим термином мы будем называть операции алгебраического сложения, умножения и деления на параметр. Следствием свойства 2.2.3 является то, что при выполнении коротких операций тригонометрические треугольные коды можно рассматривать как состоящие из четырех независимых частей - треугольных кодов и выполнять операции над этими частями по правилам, описанным выше. Независимость четырех частей тригонометрического треугольного кода следует из того, что в треугольном коде выработка переноса основывается на соотношении (1.3.1), а в тригонометрическом треугольном коде - на соотношении

$$\text{ТТК}_R(R) = \begin{pmatrix} 1 & & & \\ & 0 & 0, & \\ & & 0 & 0 & 1 \end{pmatrix}$$

являющимся следствием свойства 2. Отсюда, например, получаем:

$$\left\{ \begin{matrix} & & & 0 \\ & & 0 & 0 \\ & R & R & 0 \\ R & R & 0 & 0 \end{matrix} \right\} = \left\{ \begin{matrix} & & & 0 \\ & & 1 & 1 \\ & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{matrix} \right\}$$

На этом примере видно, что переносы, возникающие в четырех младших разрядах, не взаимодействуют между собой.

**2.3.2. Умножение** тригонометрических треугольных кодов полностью аналогично умножению треугольных кодов. Эта аналогия объясняется тем, что для построения и тех и других используются одностипные разложения, а умножение кода на вес разряда в обоих случаях эквивалентно сдвигу.

Для сравнения рассмотрим обычное умножение тригонометрических рядов.

**Пример 2.3.1. Умножение тригонометрических рядов.**

Найдем произведение тригонометрических рядов



$\Phi_1(x) = F(x) \cdot F'''(x)$ , где функции  $F(x)$ ,  $F'''(x)$  определены в примере 2.2.2. При этом

$$\Phi_1(x) = \left\{ \begin{array}{l} (32 - 42\cos 2x - 8\cos 4x - 4\cos 6x - 2\cos 8x) \cdot \\ (84 + 32\sin 2x + 24\sin 4x + 16\sin 8x) \end{array} \right\}$$

Применяя формулу  $\sin \alpha \cos \beta = \frac{1}{2} \{ \sin(\alpha - \beta) + \sin(\alpha + \beta) \}$ , получаем:

$$\Phi_1(x) = \left\{ \begin{array}{l} 2312\sin 2x - 1108\sin 4x - 492\sin 6x - \\ 288\sin 8x - 580\sin 10x - 144\sin 12 - \\ 56\sin 14x - 16\sin 16x \end{array} \right\}$$

**Пример 2.3.2. Умножение тригонометрических рядов.**

Найдем произведение рядов вида (2.1.9)  $\Phi_1(x) = F(x) \cdot F'''(x)$ , где функции  $F(x)$ ,  $F'''(x)$  определены в примере 2.2.2. При этом

$$\Phi_1(x) = \left\{ \begin{array}{l} (-24 + 71\psi_{01} - 36\psi_{02} + 10\psi_{03} - \psi_{04}) \cdot \\ (4\sin x \cos x \cdot (142 - 144\psi_{01} + 60\psi_{02} - 8\psi_{03})) \end{array} \right\}$$

Применяя формулу  $(a\psi_{0i}) \cdot (b\psi_{0j}) = ab\psi_{0,i+j}$ , получаем:

$$\Phi_1(x) = 4\sin x \cos x \cdot \left\{ \begin{array}{l} -3408 + 13538\psi_{01} - 16776\psi_{02} + 11056\psi_{03} \\ -4310\psi_{04} + 1032\psi_{05} - 140\psi_{06} + 8\psi_{07} \end{array} \right\}$$

В примере 2.1.1 показано, что эта же функция может быть представлена рядом вида (2.1.8):

$$\Phi_1(x) = \left\{ \begin{array}{l} 2312\sin 2x - 1108\sin 4x - 492\sin 6x - \\ 288\sin 8x - 580\sin 10x - 144\sin 12 - \\ 56\sin 14x - 16\sin 16x \end{array} \right\}$$

Таким образом, результат этого примера совпадает с полученным в примере 2.3.1.

**2.3.3. Дифференцирование** тригонометрических

треугольных кодов основано на очевидном соотношении

$$\frac{d}{dx} \psi'_{mk} = (k - m)\psi'_{m+1,k} - m\psi'_{m-1,k}. \quad (2.3.1)$$

В соответствии со свойством 1.1.5 треугольных кодов имеем:

$$R\psi'_{m-1,k-2} = \psi'_{m+1,k} + \psi'_{m-1,k}. \quad (2.3.2)$$

Из (2.3.1) и (2.3.2) находим:

$$\frac{d}{dx}\psi'_{mk} = k\psi'_{m+1,k} - Rm\psi'_{m-1,k-2}. \quad (2.3.3)$$

Из (2.3.1) получаем:

$$\frac{d}{dx} \begin{Bmatrix} & & 0 \\ & 0 & \alpha_{mk} \\ 0 & 0 & 0 \end{Bmatrix} = \begin{Bmatrix} & & (k-m)\alpha_{mk} \\ & 0 & 0 \\ 0 & 0 & -m\alpha_{mk} \end{Bmatrix}. \quad (2.3.4)$$

Из (2.3.3) получаем:

$$\frac{d}{dx} \begin{Bmatrix} & & 0 \\ & 0 & \alpha_{mk} \\ 0 & 0 & 0 \end{Bmatrix} = \begin{Bmatrix} & & k\alpha_{mk} \\ & 0 & 0 \\ -Rm\alpha_{mk} & 0 & 0 \end{Bmatrix}. \quad (2.3.5)$$

Возможны, по крайней мере, три способа дифференцирования.

Первый из них заключается в том, что по последним формулам определяются коды  $\text{ТТК}_R\left(\frac{d}{dx}\psi'_{mk}\right)$  и дифференцирование выполняется по формуле

$$\frac{d}{dx}\Phi(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} \frac{d}{dx}\psi'_{mk}.$$

Недостаток этого способа заключается в необходимости использовать таблицы кодов функций  $\left(\frac{d}{dx}\psi'_{mk}\right)$ , требующие определенного объема памяти и времени для обращения. Заметим, что коды можно хранить в виде треугольных кодов  $\text{ТК}_R\left(\frac{d}{dx}\psi'_{mk}\right)$ , т.к. они содержат значащие разряды только в одной составляющей тригонометрического кода.

Второй способ дифференцирования - аппаратный - заключается в схемной организации переносов из  $mk$ -разряда в  $(m+1, k)$ -разряд и в  $(m-1, k)$ -разряд. Величина этих переносов определяется согласно (2.3.4), а их суммирование в данном разряде производится по тем же правилам, что и в коротких операциях.

Наконец, третий способ состоит в вычислении по формуле

$$\frac{d}{dx}\Phi(x) = \left[ \begin{array}{l} \psi'_{10} \sum_{k=0}^n k \sum_{m=0}^k \alpha_{mk} \psi'_{mk} - \\ \frac{R}{\psi'_{12}} \sum_{m=0}^n m \sum_{k=m}^m \alpha_{mk} \psi'_{mk} \end{array} \right],$$

которая является следствием соотношения (2.3.5). Для вычисления по этой формуле используется следующий

**Алгоритм 2.3.1. Дифференцирование кода  $\text{ТТК}_R(\Phi(x))$ .**

1. Все столбцы исходного кода последовательно умножаются на номер столбца, а полученные произведения суммируются, т.е. вычисляется величина

$$\sum_{k=0}^n k \sum_{m=0}^k \alpha_{mk} \psi'_{mk}$$

2. Производится 10-сдвиг результата п.1, т.е. вычисляется величина

$$\psi'_{10} \sum_{k=0}^n k \sum_{m=0}^k \alpha_{mk} \psi'_{mk}$$

3. Все строки исходного кода последовательно умножаются на номер строки, а полученные произведения суммируются, т.е. вычисляется величина

$$\sum_{m=0}^n m \sum_{k=m}^n \alpha_{mk} \psi'_{mk}$$

4. Производится (-1,-2)-сдвиг результата п.3. В результате этого сдвигается 0-строка и диагональ сдвигаемого кода. Это соответствует упрощению формулы (2.3.1) для теряемых разрядов:

$$\frac{d}{dx} \psi'_{0k} = k \psi'_{1k} \psi, \quad \frac{d}{dx} \psi'_{kk} = -k \psi'_{k-1,k}.$$

5. Производится умножение на параметр результата по п. 4, т.е. вычисляется величина

$$\frac{R}{\psi'_{12}} \sum_{m=0}^n m \sum_{k=m}^n \alpha_{mk} \psi'_{mk}.$$

6. Из результата п. 2 вычитается результат п. 5, т.е. определяется искомая функция.

**2.3.4. Интегрирование** тригонометрических треугольных кодов. Введем обозначение

$$J_{mk} = \int \psi'_{mk}(x) dx. \tag{2.3.6}$$

При этом

$$\int \Phi(x) = \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} J_{mk}.$$

### 2.3. Операции с тригонометрическими треугольными кодами

Интегрирование можно выполнять непосредственно по этой формуле. Для этого предварительно необходимо вычислить коды  $\text{ТТК}_R(J_{mk}(x))$  для всех  $m$  и  $k$ . Как и при дифференцировании, коды можно хранить в виде треугольных кодов  $\text{ТК}_R(J_{mk}(x))$ , т.к. они содержат значащие разряды только в одной составляющей тригонометрического кода.

При интегрировании функции, имеющей постоянную составляющую, возникает составляющая, пропорциональная аргументу  $x$ . Кодирование этой составляющей должно производиться с учетом области изменения аргумента  $x$ .

Формулы для вычисления  $J_{mk}$ , которые должны быть использованы для составления таблиц кодов  $\text{ТТК}_R(J_{mk}(x))$ , могут быть определены непосредственным интегрированием функций веса  $\psi'_{mk}$ . Заменяя в (2.3.1)  $(m)$  на  $(m-1)$  и интегрируя находим:

$$\psi'_{m-1,k} = (k - m + 1)J_{m,k} - mJ_{m-2,k}$$

или

$$J_{m,k} = \frac{\psi'_{m-1,k}}{(k - m + 1)} + \frac{mJ_{m-2,k}}{(k - m + 1)}. \quad (2.3.7)$$

Заменяя в (2.3.3)  $(m)$  на  $(m-1)$  и интегрируя находим:

$$\psi'_{m-1,k} = kJ_{mk} - RmJ_{m-2,k-2}$$

или

$$J_{m,k} = \frac{\psi'_{m-1,k}}{k} + \frac{RmJ_{m-2,k-2}}{k}. \quad (2.3.8)$$

Из (2.3.7) получаем:

$$\left\{ \begin{array}{ccc} & & 0 \\ & 0 & J_{mk}\alpha_{mk} \\ 0 & 0 & 0 \end{array} \right\} = \left\{ \begin{array}{ccc} & \frac{\psi'_{m-1,k}\alpha_{mk}}{k - m + 1} & \\ & 0 & \\ 0 & 0 & \frac{-J_{m+1,k}m\alpha_{mk}}{k - m + 1} \end{array} \right\}.$$

Из (2.3.8) получаем:

$$\frac{d}{dx} \left\{ \begin{array}{ccc} & & 0 \\ & 0 & J_{mk}\alpha_{mk} \\ 0 & 0 & 0 \end{array} \right\} = \left\{ \begin{array}{ccc} & & \frac{\psi'_{m-1,k}\alpha_{mk}}{k} \\ & 0 & 0 \\ -\frac{J_{m+1,k-2}Rm\alpha_{mk}}{k} & & 0 \end{array} \right\}.$$

По последним формулам могут быть подготовлены необходимые для интегрирования коды  $\text{ТТК}_R(J_{mk}(x))$  для всех  $m$  и  $k$ .

**2.3.5. Инвертирование аргумента.** Так мы будем называть операцию вычисления функции  $\Phi(-x)$  по известной функции  $\Phi(x)$ . Очевидно,

$$\psi'_{mk}(-x) = (-1)^{k-m} \psi'_{mk}(x).$$

Следовательно, инвертирование аргумента заключается в инвертировании тех разрядов кода, у которых  $(k-m)$  нечетно. Возникающие при этом переносы учитываются также, как и в коротких операциях.

**2.3.6. Смещение оси ординат.** Под этим термином понимается вычисление функции  $\Phi(a \pm x)$  по известной функции  $\Phi(x)$ . Ясно, что в частном случае эта операция превращается в инвертирование аргумента. Но и при  $a = \pi$  она во многом аналогична предыдущей операции. Действительно,

$$\psi'_{mk}(\pi + x) = (-1)^k \psi'_{mk}(x).$$

$$\psi'_{mk}(\pi - x) = (-1)^m \psi'_{mk}(x).$$

Следовательно, и в этом случае смещение оси ординат заключается в инвертировании некоторых разрядов исходного кода. При других значениях  $a$  требуется более сложное преобразование исходного кода. Приведем формулы для некоторых случаев:

$$\psi'_{mk}\left(\frac{\pi}{2} \pm x\right) = \mp \psi'_{k-m,k}(x).$$

$$\psi'_{mk}\left(-\frac{\pi}{2} + x\right) = (-1)^{k-m} \psi'_{k-m,k}(x).$$

$$\psi'_{mk}\left(-\frac{\pi}{2} - x\right) = (-1)^k \psi'_{k-m,k}(x).$$

Из этих формул следует, что при  $a = \pm\pi/2$  требуется перестановка разрядов исходного кода с последующим инвертированием некоторых разрядов.

## 2.4. Кодирование и декодирование тригонометрических треугольных КОДОВ

Излагаемые здесь алгоритмы кодирования и декодирования содержат два этапа:

- предварительный, на котором вычисляются некоторые константы и строятся таблицы величин, необходимых при кодировании/декодировании функций/кодов,
- оперативный, на котором производится собственно кодирование или декодирование с использованием полученных на предварительном этапе величин.

### Алгоритм 2.4.1. Кодирование функции $\Phi(x)$ .

#### Предварительный этап:

Вычисление кодов  $\text{TK}_R(\gamma_\omega^i)$  в соответствии с теоремой 2.1.3.

Заметим, что при  $R \neq 4$  функции  $\gamma_\omega^i$  могут быть представлены кодами  $\text{TK}_R(\gamma_\omega^i)$  лишь приближенно. Поэтому целесообразно применять именно четверичные тригонометрические треугольные коды - см. раздел 4.1.

#### Оперативный этап

1. Выделение из тригонометрического ряда функции  $\Phi(x)$  рядов

$$\eta^i f^i(x) = \sum_{\omega=0}^n D_\omega^i \varepsilon_\omega^i \quad (2.4.1)$$

- см. теорему 2.2.1. Числа  $D_\omega^i$  должны быть целыми.

2. Вычисление чисел  $E_\omega^i$  в зависимости от чисел  $D_\omega^i$  в соответствии с теоремой 2.1.5 – см. (2.1.19).

3. Кодирование чисел  $E_\omega^i$ , то есть образование кодов  $\text{TK}_R(E_\omega^i)$ .

4. Вычисление кодов  $\text{TK}_R(\eta^i f^i(x))$  по известным кодам  $\text{TK}_R(E_\omega^i)$

и  $\text{TK}_R(\gamma_\omega^i)$  по формуле

$$\eta^i f^i(x) = \sum_{\omega=0}^n E_\omega^i \gamma_\omega^i. \quad (2.4.2)$$

Эта формула следует из теоремы 2.1.5: если функция имеет разложение (2.4.1), то она имеет также разложение (2.4.2), где коэффициенты  $E_{\omega}^i$  и  $D_{\omega}^i$  связаны соотношением (2.1.14).

5. Компоновка кода  $\text{ТТК}_R(\Phi(x))$  из кодов  $\text{ТК}_R(f^i(x))$  - см. определение 2.2.1.

**Алгоритм 2.4.2. Кодирование функции  $\Phi(x)$ .**

Предварительный этап: определение чисел  $S^i(v, n)$  согласно теореме 2.1.1.

Оперативный этап

1. Выделение из тригонометрического ряда функции  $\Phi(x)$  рядов (2.4.1) аналогично п.1 алгоритма 2.4.1.
2. Определение чисел  $A_v^i$  в зависимости от чисел  $D_{\omega}^i$  в соответствии с теоремой 2.1.5 – см. (2.1.18).
3. Кодирование функций  $f^i(x)$  при известных числах  $A_v^i$ , то есть образование кодов  $\text{ТК}_R(f^i(x))$  - см. раздел 1.6.
4. Компоновка кода  $\text{ТТК}_R(\Phi(x))$  из кодов  $\text{ТК}_R(f^i(x))$  - см. определение 2.2.1.

**Алгоритм 2.4.3. Декодирование кода  $\text{ТТК}_R(\Phi(x))$ .**

Предварительный этап: определение чисел  $L^i(v, n)$  согласно теореме 2.1.2.

Оперативный этап:

1. декомпоновка кода  $\text{ТТК}_R(\Phi(x))$ , то есть образование кодов  $\text{ТК}_R(f^i(x))$ ;
2. декодирование кодов  $\text{ТК}_R(f^i(x))$ , то есть определение чисел  $A_v^i$  - см. раздел 1.6;
3. вычисление коэффициентов  $D_{\omega}^i$  тригонометрического ряда функций  $f^i(x)$  в зависимости от значений чисел  $A_v^i$  в соответствии с теоремой 2.1.5 – см. (2.1.17);
4. формирование тригонометрического ряда функции  $\Phi(x)$ , в соответствии с теоремой 2.2.1.

## 2.5. Погрешность кодирования тригонометрических треугольных КОДОВ

При оценке погрешности кодирования тригонометрических рядов следует, очевидно, говорить о погрешности кодирования коэффициентов  $D_{\omega}^i$  этих рядов - абсолютной и относительной.

Как указано в описании алгоритмов 2.4.1 и 2.4.2, числа  $D_{\omega}^i$  должны быть целыми. Следовательно, абсолютная погрешность кодирования  $\Delta = 1$ . Относительную погрешность кодирования  $\delta$  естественно относить к величине максимального по модулю коэффициента:

$$\delta = \Delta / \left( \max_{\omega, i} |D_{\omega}^i| \right). \quad (2.5.1)$$

Относительная погрешность кодирования отдельных гармоник тригонометрического ряда увеличивается с уменьшением амплитуды этих гармоник. Структура кода тригонометрического ряда такова, что допустимые при заданной разрядности кода амплитуды гармоник уменьшаются с ростом их частоты. К счастью, это же характерно для большинства реальных процессов. Поэтому увеличение относительной погрешности кодирования высших гармоник вполне допустимо. Оценим величину  $\delta$  в зависимости от разрядности более строго, для чего рассмотрим (без доказательства) две теоремы.

**Теорема 2.5.1.** Числа

$$p(m, k, \omega) = \sum_{v=(\omega, k-m)}^k (-1)^{v+k+m+\omega} 4^{k-v} C_m^{k-v} C_{2v}^{v-\omega} \quad (2.5.2)$$

связаны следующими рекуррентными соотношениями:

$$p(0, 0, 0) = 1;$$

$$p(m, m, \omega) = C_{2m}^{m-\omega}, \quad \text{if } 0 \leq \omega \leq m;$$

$$p(m, k, 0) = 2p(m, k-1, 0) - p(m, k-1, 1), \quad \text{if } k \geq m;$$



$$p(m, k, \omega) = \begin{bmatrix} -p(m, k-1, \omega-1) \\ +2p(m, k-1, \omega) - \\ p(m, k-1, \omega+1) \end{bmatrix}, \quad \text{if } k \geq m \text{ and } 1 \leq \omega < k;$$

$$p(m, k, k) = (-1)^{m+k}, \quad \text{if } k \geq m;$$

$$p(m, k, \omega) = 0, \quad \text{if } \omega > k \text{ or } k < m.$$

В табл. 2.5.1.  $\omega$  приведены значения чисел  $p(m, k, \omega)$  при  $\omega = (0, 1, 2, 3, 4)$  соответственно и при  $k < 6$  и  $m < 6$ . В этих же таблицах указаны числа

$\Pi_1(k, \omega)$  - сумма всех положительных чисел  $p(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

$\Pi_2(k, \omega)$  - сумма всех отрицательных чисел  $p(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

Таблица 2.5.1.0. Числа  $p(m, k, 0)$ .

$k \setminus m$	0	1	2	3	4	5	$\Pi_1(k, 0)$	$\Pi_2(k, 0)$
0	1						1	0
1	2	-1					4	0
2	6	-4	1				14	0
3	20	-15	6	-1			48	0
4	70	-56	28	-8	1		166	0
5	252	-210	120	-45	10	-11	584	0

Таблица 2.5.1.1. Числа  $p(m, k, 1)$ .

$k \setminus m$	0	1	2	3	4	5	$\Pi_1(k, 0)$	$\Pi_2(k, 0)$
1	2	1					1	-1
2	2	0	-1				4	-4
3	4	-1	-2	1			16	-16
4	10	-4	-4	4	-1		60	-60
5	28	-14	-8	13	-6	1	226	-226

Таблица 2.5.1.2. Числа  $p(m, k, 2)$ .

$k \setminus m$	0	1	2	3	4	5	$\Pi_1(k, 0)$	$\Pi_2(k, 0)$
2	6	4	1				2	-1
3	4	1	-2	-1			12	-4
4	6	0	-4	0	1		56	-12
5	12	-2	-8	3	2	-1	240	-32

Таблица 2.5.1.3. Числа  $p(m, k, 3)$ .

$k \backslash m$	0	1	2	3	4	5	$\Pi_1(k, 0)$	$\Pi_2(k, 0)$
3	20	15	6	1			2	-2
4	10	4	-4	-4	-1		12	-12
5	12	2	-8	-3	2	1	61	-61

Таблица 2.5.1.4. Числа  $p(m, k, 4)$ .

$k \backslash m$	0	1	2	3	4	5	$\Pi_1(k, 0)$	$\Pi_2(k, 0)$
4	70	56	28	8	1		3	-2
5	28	14	-8	-13	-6	-1	24	-12

В табл. 2.5.2 приведены значения чисел  $p(m, k, \omega)$  при

$$k < 10, \omega < 10, m = \begin{cases} k/2, & \text{if } k - \text{even}, \\ (k-1)/2, & \text{if } k - \text{odd}. \end{cases}$$

Таблица 2.5.2. Числа  $p(m, k, \omega)$ .

$m$	$k \backslash \omega$	0	1	2	3	4	5	6	7	8	9
0	0	1									
0	1	-2	-1								
1	2	-2	0	1							
1	3	4	1	-2	-1						
2	4	6	0	-4	0	1					
2	5	-12	-2	8	3	-2	-1				
3	6	-20	0	15	0	-6	0	1			
3	7	40	5	-30	-9	12	5	-2	-1		
4	8	70	0	-56	0	28	0	-8	0	-1	
4	9	-140	-14	112	28	-56	-20	16	7	-2	-1

Определим еще следующие числа

$$p^0(m, k, \omega) = p(m, k, \omega); \quad (2.5.3)$$

$$p^1(m, k, \omega) = p(m, k, \omega) - p(m, k, \omega + 1); \quad (2.5.4)$$

$$p^2(m, k, \omega) = p(m, k, \omega) + p(m, k, \omega + 1); \quad (2.5.5)$$

$$p^3(m, k, \omega) = p(m, k, \omega) - p(m, k, \omega + 2); \quad (2.5.6)$$

**Теорема 2.5.2.** Функция  $\eta^i \psi_{mk}$  имеет разложение вида

$$\eta^i \psi_{mk} = \sum_{\omega=0}^k q^i(m, k, \omega) \cdot \varepsilon^i, \quad (2.5.7)$$

где

$$q^i(m, k, \omega) = a_{\omega}^i c^i \left( \frac{R}{4} \right)^k p^i(m, k, \omega). \quad (2.5.8)$$

Из последней теоремы следует, что функция  $F^i(x)$ , имеющая разложение вида

$$F^i(x) = \eta^i \sum_{k=0}^n \sum_{m=0}^k \alpha_{mk} \psi_{mk}, \quad (2.5.9)$$

представима также рядом (2.1.8), где

$$D_{\omega}^i = \sum_{k=\omega}^n \sum_{m=0}^k q^i(m, k, \omega) \cdot \alpha_{mk}^i. \quad (2.5.10)$$

Используя эти теоремы, можно по знакам чисел  $p^i(m, k, \omega)$  определить вид треугольного кода (расположение разрядов с минимальным и максимальным значениями), максимизирующий значение коэффициента  $D_{\omega}^i$  при данном  $\omega$ , и вычислить это значение. Обозначим его через  $M_{\omega}^i$ . В разделе 4.5 рассмотрены величины  $M_{\omega}^0$  для кодов ТТК<sub>4</sub> $(F^0(x))$  при  $\alpha_{mk}^i \in \{-2, -1, 0, 1\}$ . Там приведены зависимости  $\lg(M_{\omega}^0)$  от  $\omega$  при различных значениях  $n$  и зависимости  $\lg(M_{\omega}^0)$  от  $n$  при различных значениях  $\omega$ .

Итак, для оценки относительной погрешности кодирования при данном  $n$  достаточно найти максимум величины  $M_{\omega}^i$ : величина, обратная этому максимуму, и является относительной погрешностью  $\delta$  кодирования тригонометрического ряда.

## 2.6. Укорочение тригонометрических треугольных КОДОВ

При кодировании функций и при выполнении операций с кодами функций может возникать переполнение разрядной сетки. Для ликвидации такого переполнения вводится операция *укорочения* треугольных кодов, рассматриваемая ниже.

Назовем *значащим столбцом* треугольного кода тот столбец, в котором имеется хотя бы один разряд, отличный от нуля. Назовем, далее, номер  $N$  старшего значащего столбца *длиной* треугольного кода. Некоторая функция  $\Phi_0(x)$  в общем случае может быть представлена в виде произведения

$$h_0 \cdot [\text{ТТК}_R(\Phi_0(x))], \quad (2.6.1)$$

где

$h_0$  - числовой множитель, экспонента,

$\text{ТТК}_R(\Phi_0(x))$  - код длины  $N_0$ .

Укорочение  $\text{ТТК}_R(\Phi_0(x))$  заключается в преобразовании выражения (2.6.1) к виду

$$h_1 \cdot [\text{ТТК}_R(\Phi_1(x))], \quad (2.6.2)$$

где

$h_1$  - новая экспонента,

$\text{ТТК}_R(\Phi_1(x))$  - код длины  $N_1 < N_0$ ,

$$\Phi_1(x) \approx \Phi_0(x) \frac{h_0}{h_1}. \quad (2.6.3)$$

Последнее соотношение показывает, что укорочение в общем случае приносит некоторую погрешность.

Алгоритмически укорочение заключается в последовательном  $u$ -кратном делении исходного кода длины  $N_0$  на параметр  $R$  до получения кода заданной длины  $N_1$  с одновременной коррекцией числового множителя:  $h_1 = h_0 R^u$ . В результате деления  $R$ -го треугольного кода на параметр образуется  $R$ -ый смешанный код - см. раздел 1.4. В связи с этим возникает задача *округления смешанного кода*, которая более точно формулируется следующим образом:

- известен  $SK_R(f(x))$ ;
- необходимо вычислить  $TK_R(f_0(x))$  такой функции  $f_0(x)$ , что функция  $(f(x) - f_0(x))$  имеет тригонометрический ряд вида (2.1.8) с коэффициентами, достаточно малыми по абсолютной величине.

Прежде, чем описывать алгоритм округления, рассмотрим функцию

$$\delta_n^i = (-1)^{n+1} \left( \gamma_n^i - (-1)^n \psi_{0n} \right) \quad (2.6.4)$$

или

$$\psi_{0n} = \delta_n^i + (-1)^n \gamma_n^i \quad (2.6.5)$$

Учитывая (2.1.1) из (2.6.4) находим:

$$\delta^i(n) = (-1)^{n+1} \sum_{v=0}^n \left[ (-1)^v \left( \frac{R}{4} \right)^{n-v} S^i(v, n) \cdot \psi_{0v} \right] - (-1)^n \psi_{0n}$$

или

$$\delta^i(n) = (-1)^{n+1} \sum_{v=0}^{n-1} \left[ (-1)^v \left( \frac{R}{4} \right)^{n-v} S^i(v, n) \cdot \psi_{0v} \right]. \quad (2.6.6)$$

Учитывая (2.1.3) из (2.6.5) находим:

$$\eta^i \psi_{0n} = \eta^i \delta_n^i + (-1)^n \left( R/4 \right)^n a_n^i c^i \varepsilon_n^i. \quad (2.6.7)$$

Учитывая, далее, (2.1.23), отсюда получаем:

$$\eta^i \psi_{0n} = \eta^i \delta_n^i + g_n^i; \quad (2.6.8)$$

Таблица 2.6.1.

Ряд	Функция	Количество членов ряда	Обоснование
(2.1.9) по $\psi_{0v}(x)$	$\eta^i \psi_{0n}$	$(n+1)$	единственный $n$ -член ряда
	$\eta^i \delta_n^i$	$n$	(2.6.6)
	$\eta^i \gamma_n^i$	$(n+1)$	(2.1.1)
(2.1.8) по $\varepsilon_\omega^i(x)$	$\eta^i \psi_{0n}$	$(n+1)$	(2.1.21)
	$\eta^i \delta_n^i$	$n$	(2.6.6) и (2.1.20)
	$\eta^i \gamma_n^i$	$(n+1)$	(2.1.1) и (2.1.20)

Сравним количество членов в функциональных рядах функций  $\eta^i \psi_{0n}$ ,  $\eta^i \delta_n^i$ ,  $\eta^i \gamma_n^i$  - см. табл. 2.6.1. Из этих замечаний следует, что ряды (2.1.8-11) функции  $\eta^i \delta_n^i$  короче рядов (2.1.8-11) функции  $\eta^i \psi_{0n}$  на один член. Этот факт может быть положен в основу алгоритма округления. Действительно, функция  $\eta^i \psi_{0n}$  может быть заменена функцией  $\eta^i \delta_n^i$  отбрасыванием старшей гармоники  $g_n^i$  тригонометрического ряда функции  $\eta^i \psi_{0n}$ . Эту операцию мы и будем называть *округлением тригонометрического ряда*. Очевидно, абсолютная погрешность такого округления

$$\Delta_n^i = \left(\frac{R}{4}\right)^n a_n^i c^i.$$

В частности, при  $n > 0$  и  $R = 4$  имеем:  $\Delta_n^i = 2$ . Итак, при округлении функции  $\eta^i \psi_{0n}$  образуется округленная функция  $\eta^i \delta_n^i$ , ряд (2.1.9) которой короче на один член одноименного ряда округляемой функции. Из этого, вообще говоря, не следует, что код  $\text{TK}_R(\eta^i \delta_n^i)$  округленной функции короче кода  $\text{TK}_R(\eta^i \psi_{0n})$  округляемой функции. Однако *в среднем* при округлении код  $R$ -**ТК** укорачивается и на этом основан алгоритм укорочения. Таким образом, округление эквивалентно отбрасыванию старших гармоник тригонометрического ряда, имеющих относительно малую амплитуду.

Итак, некоторый разряд  $\psi_{0n}$  кода  $\text{TK}_R(f^i(x))$  может быть заменен кодом  $\text{TK}_R(\delta_n^i)$  с погрешностью округления  $\Delta_n^i$ . Аналогично, разряд  $\psi_{-1,n}$  смешанного кода функции  $f^i(x)$  может быть заменен кодом функции  $\delta_n^i/R$  с погрешностью округления  $\Delta_n^i/R$ . Смешанные коды  $B_n^i = \text{SK}_R\left(\delta_n^i/R\right)$  функций  $\delta_n^i/R$  могут быть получены по формуле (2.6.4) при известных  $\eta^i \gamma_n^i$ .

## 2.7. Гиперболические треугольные КОДЫ

Подробное рассмотрение именно тригонометрических треугольных кодов связано с тем, что тригонометрические ряды занимают центральное место в теории и практике функциональных рядов. Если же отвлечься от этого обстоятельства, то можно указать еще одно основание кодирования функций  $Ch^2(x)$ , обладающее наименьшими возможностями по сравнению с основанием  $Sin^2(x)$  с точки зрения формальной теории треугольных кодов. Это утверждение основано на известной аналогии между соотношениями обычной и гиперболической тригонометрии: для получения соотношений гиперболической тригонометрии достаточно в формулах обычной тригонометрии произвести замену  $Sin(x)$  на  $jSh(x)$  и  $Cos(x)$  на  $Ch(x)$ .

Треугольные коды по основанию  $Ch^2(x)$  назовем гиперболическими треугольными кодами. Очевидно, свойства гиперболических треугольных кодов аналогичны свойствам тригонометрических треугольных кодов.





# Глава 3. Кодирование функций многих аргументов

## 3.1. Пирамидальные коды

Выше рассматривались коды функций одного аргумента. Дальнейшее обобщение приводит нас к понятию кодов функций нескольких переменных. При этом естественно применить следующую схему рассуждений: код числа имеет линейную структуру; код функции одного аргумента является плоским (треугольным, ступенчатым); очевидно, код функции двух аргументов должен быть трехмерным; код функции четырех аргументов - четырехмерным и так далее.

Вначале рассмотрим объемное обобщение треугольного кода - пирамидальный код функции двух аргументов. Пусть функция  $F(x, v)$  двух аргументов  $x$  и  $v$  имеет разложение следующего вида:

$$F(x, v) = \sum_{k=0}^n \sum_{m_2=0}^k \sum_{m_1=0}^k \alpha_{m_1, m_2, k} R^k \psi_{m_1, m_2, k}, \quad (3.1.1)$$

$$\psi_{m_1, m_2, k} = y^{k-m_1} (1-y)^{m_1} z^{k-m_2} (1-z)^{m_2}, \quad (3.1.2)$$

где  $\alpha_{m_1, m_2, k}$  - действительные числа;

$R$  - целое положительное число, параметр;

$y=y(x)$  - некоторая функция аргумента  $x$ ;

$z=z(v)$  - некоторая функция аргумента  $v$ ;

$m_1, m_2, k, n$  - целые положительные числа или нули.

Имеет место следующее соотношение:

$$R \psi_{m_1, m_2, k} = \begin{bmatrix} \psi_{m_1, m_2, k+1} + \psi_{m_1+1, m_2, k+1} \\ \psi_{m_1, m_2+1, k+1} + \psi_{m_1+1, m_2+1, k+1} \end{bmatrix} \quad (3.1.3)$$

**Определение 3.1.1.** Выражение (3.1.1) называется *разложением функции  $F(x, v)$*  по основаниям  $y$  и  $z$  с параметром  $R$ .

### 3.1. Пирамидальные коды

По аналогии с определением 1.1.1 для разложения (3.1.1) также вводятся понятия  $(m_1, m_2, k)$ -разряда, веса  $(m_1, m_2, k)$ -разряда, величины  $(m_1, m_2, k)$ -разряда.

**Определение 3.1.2.** Объемная матрица, имеющая вид четырехугольной пирамиды и составленная из величин  $\alpha_{m_1, m_2, k}$  разложения (3.1.1) таким образом, что

- все величины  $\alpha_{00k}$  располагаются на одной горизонтали,
- все величины  $\alpha_{m_1, m_2, k}$  при  $k = \text{const}$  располагаются в одной вертикальной плоскости и смещены относительно величины  $\alpha_{00k}$  на  $m_1$  шагов влево и  $m_2$  шагов вверх,

называется пирамидальным кодом функции  $F(x, v)$  по основаниям  $u$  и  $z$  с параметром  $R$ . Этот код обозначается как  $PRK(F(x, v))$  - см рис. 3.1.1.

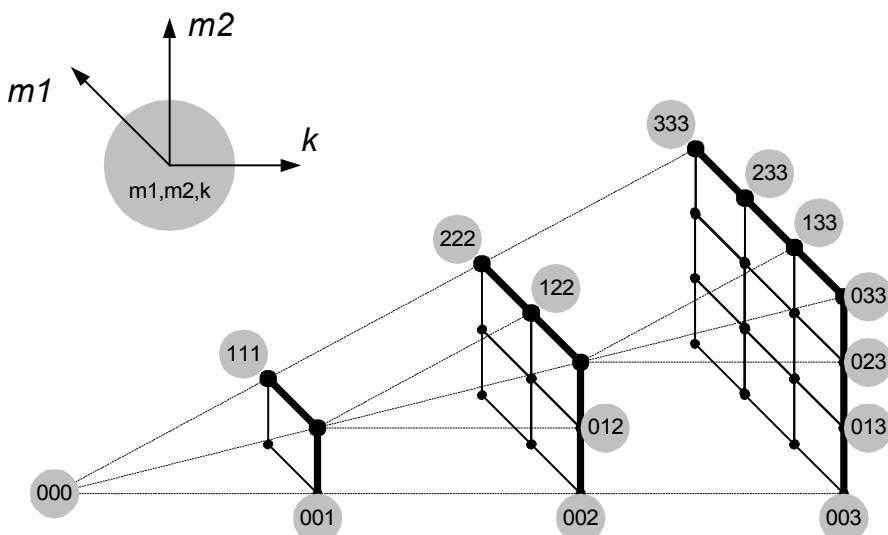
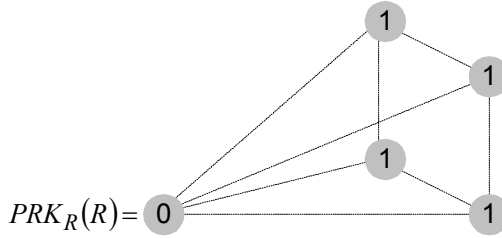


Рис. 3.1.1. Пирамидальный код

Несомненно, допускается любой поворот пирамидального кода относительно того положения, которое зафиксировано в этом определении. По аналогии с предыдущим выделим среди этих кодов  $R$ -ые пирамидальные коды  $PRK_R(F(x, v))$ , у которых разряды  $\alpha_{m_1, m_2, k} \in D_R$  - см. определение 1.1.3. Из (3.1.3) находим:



то есть перенос в рассматриваемом случае распространяется в четыре разряда и поэтому  $R \geq 7$ . Для  $PRK_R(F(x, v))$  также соблюдается общий для всех позиционных кодов принцип выполнения арифметических операций.

Очевидно, структура пирамидальных кодов и алгоритмы операций с ними не накладывают ограничений на вид функций  $u$  и  $z$ . Но по соображениям, изложенным выше для кодов функций одного аргумента, и в этом случае желательно выбрать основания вида  $\sin^2(x)$  и  $\sin^2(v)$ , поскольку они допускают кодирование тригонометрических рядов функций двух аргументов.

Тригонометрический ряд функции двух аргументов имеет следующий вид:

$$\Phi(x, v) = \sum_{\omega_1=0}^n \sum_{\omega_2=0}^n \begin{bmatrix} a_{\omega_1\omega_2} \cos(\omega_1x)\cos(\omega_2v) + \\ b_{\omega_1\omega_2} \sin(\omega_1x)\cos(\omega_2v) + \\ c_{\omega_1\omega_2} \cos(\omega_1x)\sin(\omega_2v) + \\ d_{\omega_1\omega_2} \sin(\omega_1x)\sin(\omega_2v) \end{bmatrix} \quad (3.1.4)$$

Рассуждая аналогично предыдущему, можно показать, что такой ряд представим в виде следующей суммы:

$$\Phi(x, v) = \sum_{k=0}^n \sum_{m_2=0}^k \sum_{m_1=0}^k \alpha_{m_1, m_2, k} \psi'_{m_1, m_2, k}, \quad (3.1.5)$$

где

$$\psi'_{m_1, m_2, k} = R^{k/2} \sin^{k-m_1}(x) \cos^{m_1}(x) \sin^{k-m_2}(v) \cos^{m_2}(v). \quad (3.1.6)$$

**Определение 3.1.3.** Объемная матрица, составленная из величин  $\alpha_{m_1, m_2, k}$  разложения (3.1.5) по правилам определения 3.1.2, называется *тригонометрическим пирамидальным кодом* функции  $\Phi(x, v)$ . Этот код обозначается как  $TPRK(\Phi(x, v))$ . Код, в котором

### 3.1. Пирамидальные коды

---

$\alpha_{m_1, m_2, k} \in D_R$ , называется  $R$ -ым и обозначается как  $TPRK_R(\Phi(x, v))$ .

Из последней формулы следует, что

$$\psi'_{m_1, m_2, k} = \left[ \begin{array}{l} \psi'_{m_1, m_2, k+2} + \psi'_{m_1+2, m_2, k+2} + \\ \psi'_{m_1, m_2+2, k+2} + \psi'_{m_1+2, m_2+2, k+2} \end{array} \right] \quad (3.1.7)$$

В свою очередь, из этого соотношения следует, что **TPRK** состоит из десяти частей, причем

- каждый разряд кода принадлежит только одной из этих частей,
- индексы  $m_1, m_2, k$  разрядов, принадлежащих одной части, отличаются только на четное число и, таким образом, разряды одной части образуют пирамидальный код некоторой функции по основаниям  $\text{Sin}^2(x)$  и  $\text{Sin}^2(v)$ ,
- угловые разряды пирамидальных кодов десяти частей имеют индексы 000, 001, 011, 101, 111, 012, 102, 112, 122, 212.

Таким образом, при выполнении коротких операций с  $R$ -**TPRK** их можно рассматривать состоящими из десяти независимых кодов и выполнять операции независимо с каждой из этих частей. В остальном свойства **TPRK** и правила выполнения различных операций с ними аналогичны тому, что имеет место для кодов **ТТК** и **PRK**.

## 3.2. Гиперпирамидальные коды

При кодировании функций нескольких переменных рассматривается функция, зависящая от  $a$  аргументов  $x_i$  и имеющая разложение следующего вида:

$$F(x_1, \dots, x_i, \dots, x_a) = \sum_{k=0}^n \sum_{m_1=0}^k \dots \sum_{m_a=0}^k \left( \alpha_{m_1, \dots, m_a, k} \prod_{i=1}^a y_i^{k-m_i} (1-y_i)^{m_i} \right) \quad (3.2.1)$$

где

$\alpha_{m_1, \dots, m_a, k}$  - действительные числа;

$R$  - целое положительное число, параметр;

$y_i = f_i(x_i)$  - некоторая функция аргумента  $x_i$ ;

$m_i, k, n$  - целые положительные числа или нули.

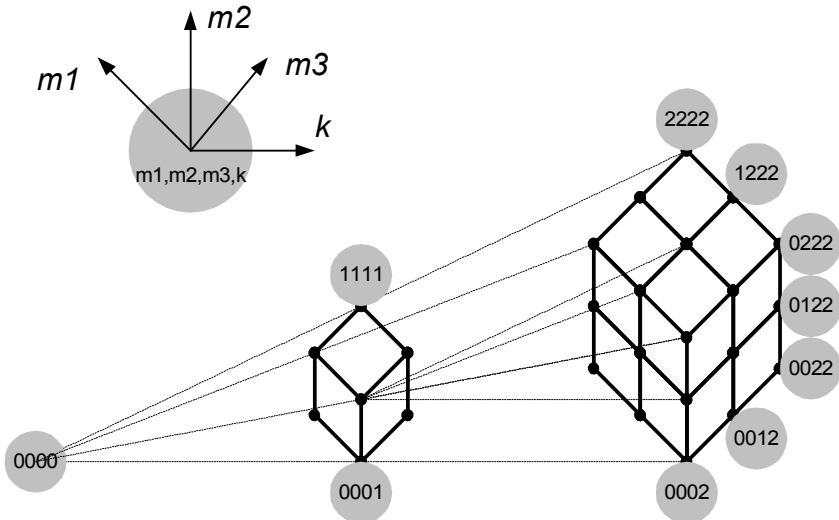


Рис. 3.2.1. Гиперпирамидальный код.

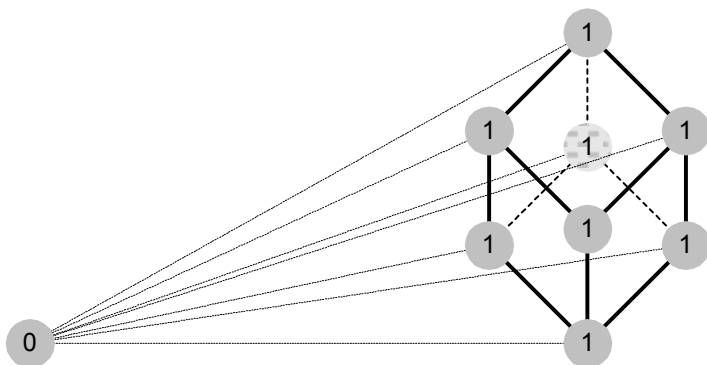
Естественно называть формулу (3.2.1) разложением функции  $F(x_1, \dots, x_i, \dots, x_a)$  по основаниям  $y_i, i = \overline{1, a}$  с параметром  $R$ ; объемную матрицу, составленную из величин  $\alpha_{m_1, \dots, m_a, k}$  разложения (3.2.1) - гиперпирамидальным кодом этой функции и так далее. Очевидно, разложение (3.2.1) при  $a=1$  превращается в разложение (1.1.1), а гиперпирамидальный код - в треугольный. При

### 3.2. Гиперпирамидальные коды

$a=2$  разложение (3.2.1) превращается в разложение (3.1.1), а гиперпирамидальный код - в пирамидальный. При дальнейшем увеличении структура кода усложняется: вместо  $k$ -столбца в треугольном коде и  $k$ -квадрата в пирамидальном коде появляется  $a$ -мерный  $k$ -куб. В качестве иллюстрации на рис. 3.2.1 изображен гиперпирамидальный код при  $a = 3$ .

Очевидно, что все сказанное относительно пирамидального кода обобщается и на гиперпирамидальные коды. Дополнительно отметим только два факта.

В гиперпирамидальном коде перенос распространяется в  $a$ -мерный куб, содержащий  $2^a$  разрядов. Следовательно параметр  $R \geq (2^{a-1} - 1)$ . Ниже представлен код параметра  $R$ :



Гиперпирамидальные коды тригонометрических рядов - тригонометрические гиперпирамидальные коды функций нескольких переменных строятся по аналогии с предыдущим и состоят из нескольких независимых частей - гиперпирамидальных кодов по основаниям  $\text{Sin}^2(x_i)$ ,  $i = \overline{1, a}$ . Число таких частей равно  $(1 + 3^a)$ .

# Глава 4. Четверичные тригонометрические треугольные коды

---

В данной главе подробно рассматриваются алгоритмы операций с четверичными тригонометрическими треугольными кодами, описанные в общем случае в предыдущих главах. Целью этой главы является конкретизация теории кодирования функций до такой степени, которая допускает постановку задачи технического проектирования соответствующих компьютеров. Предпочтение, отдаваемое четверичным кодам, вызвано следующими обстоятельствами:

1. не существует двоичных треугольных кодов,
2. схемы для операций с четверичными кодами легко реализуются на двоичных элементах,
3. большинство формул, используемых в алгоритмах кодирования и декодирования тригонометрических треугольных кодов, приобретают для четверичных кодов более простой вид.

В связи с тем, что далее будут рассматриваться исключительно четверичные коды, символ  $R$  в обозначениях кода будет опускаться. Условимся далее, что множество значений величин разрядов  $D_R = \{-2, -1, 0, 1\}$ .

## 4.1. Арифметические операции с четверичными треугольными кодами

В общем виде арифметические операции с треугольными кодами описаны выше. Здесь будут указаны лишь особенности четверичных кодов.

### 4.1.1. Алгебраическое сложение.

Вначале рассмотрим алгебраическое сложение четверичных кодов чисел  $K_4(A)$ , которые составляют прямоугольный код. Вообще

#### 4.1. Арифметические операции с четверичными треугольными кодами

говоря, такие коды при  $R=4$  и  $D_R=\{-2,-1,0,1\}$  эквивалентны двоичным кодам вещественных чисел по основанию  $(-2)$ . Тем не менее мы рассмотрим (для дальнейших аналогий) табл. 4.1.1, которая описывает алгебраическое сложение кодов  $K_4(A)$  при  $R=4$  и  $D_R=\{-2,-1,0,1\}$ . Эта таблица построена с использованием формул (1.2.1), (1.2.3) и удовлетворяет условиям полноты.

Таблица 4.1.1.

S	$\sigma$	$\pi$					
5	1	1					
4	0	1					
3	-1	1					
2	-2	1					
1	1	0					
0	0	0					
-1	-1	0					
-2	-2	0					
-3	1	-1					
-4	0	-1					
-5	-1	-1					
-6	-2	-1					
-7	1	-2					
-8	0	-2					
-9	-1	-2					
-10	-2	-2					

В табл. 4.1.2 перечислены варианты алгебраического сложения треугольных кодов при  $R=4$  и  $D_R=\{-2,-1,0,1\}$ . Эта таблица построена с использованием формул (1.3.1), (1.3.2) и удовлетворяет условиям полноты.



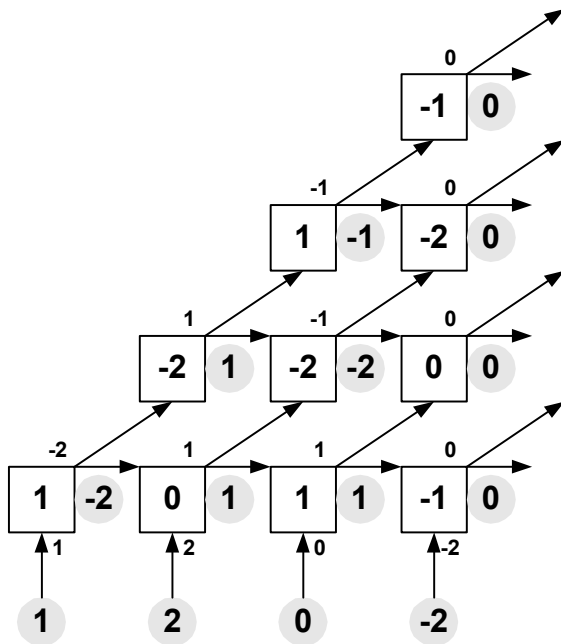
Таблица 4.1.2.

S	$\sigma$	$\pi$
8	0	2
7	-1	2
6	-2	2
5	1	1
4	0	1
3	-1	1
2	-2	1
1	1	0
0	0	0
-1	-1	0
-2	-2	0
-3	1	-1
-4	0	-1
-5	-1	-1
-6	-2	-1
-7	1	-2
-8	0	-2
-9	-1	-2
-10	-2	-2
-11	1	-3
-12	0	-3
-13	-1	-3
-14	-2	-3

**Пример 4.1.1. Умножение на параметр** кода функции

$$SK_4(f(x)) = \begin{matrix} & & & & 0 \\ & & & & -1 & 0 \\ & & & 1 & 1 & 0 \\ & & -2 & 0 & 2 & 0 \\ & 1 & -2 & -2 & 0 & \end{matrix}$$

По табл. 4.1.2 строим схему распространения переносов, где в кругах записаны разряды  $\alpha_{mk}$  исходного кода, в квадратах - разряды  $\sigma_{mk}$  произведения, а рядом со стрелками - переносы  $\pi_{mk}$  из разряда, откуда выходят стрелки.



Следовательно, код произведения

$$TK_4(4f(x)) = \begin{matrix} & & & -1 \\ & & & 1 & -2 \\ & & -2 & -2 & 0 \\ & 1 & 0 & 1 & -1 \end{matrix}$$

#### 4.1.2. Деление на параметр.

В данном случае  $R=4$ ,  $D_R = \{-2, -1, 0, 1\}$  и  $(\alpha_{mk}, \pi'_{mk}, \pi''_{mk}) \in D_R$ . При этом по формуле (1.4.1) находим:  $\frac{-11}{4} \leq S_{mk} \leq \frac{7}{4}$ . Каждое из этих значений представляется в виде удовлетворяет условию (1.4.2), где  $\sigma_{mk}$  - целое число. В табл. 4.1.3 перечислены все возможные значения и соответствующие им значения.

Таблица 4.1.3.

$4S_{mk}$	$\pi_{mk}$	$\sigma_{mk}$
-11	1	-3
-10	-2	-2
-9	-1	-2
-8	0	-2
-7	1	-2
-6	-2	-1
-5	-1	-1
-4	0	-1
-3	1	-1
-2	-2	0
-1	-1	0
0	0	0
1	1	0
2	-2	1
3	-1	1
4	0	1
5	1	1
6	-2	2
7	-1	2

Важно отметить, что в этой таблице  $-3 \leq \sigma_{mk} \leq 2$ . Это означает, что разряды частного должны иметь 2 бита (а не 2, как в других случаях) и после деления должно выполняться распространение переносов из разрядов, не удовлетворяющих условию  $\sigma_{mk} \in D_R$ .

**Пример 4.1.2. Деление на параметр.** Рассмотрим код

$$\begin{array}{r}
 1 \\
 0 \ 0 \\
 0 \ 0 \ 0 \\
 \text{TK}_4 = 0 \ 0 \ 0 \ 0
 \end{array}$$

После деления первой строки получаем:

$$\begin{array}{cccc} & & & 1/4 \\ & & & 0 \ 0 \\ & & 0 \ 0 & 0 \\ \text{TK}_4 = & 0 \ 0 & 0 \ 0 & 0 \end{array}$$

После передачи переноса из третьей строки получаем:

$$\begin{array}{cccc} & & & 0 \\ & & & 1 \ (-1/4) \\ & & 0 \ 0 & 0 \\ \text{TK}_4 = & 0 \ 0 & 0 \ 0 & 0 \end{array}$$

После передачи переноса из второй строки получаем:

$$\begin{array}{cccc} & & & 0 \\ & & & 1 \ 0 \\ & & 0 \ -1 & (1/4) \\ \text{TK}_4 = & 0 \ 0 & 0 \ 0 & 0 \end{array}$$

После передачи переноса из первой строки получаем:

$$\begin{array}{cccc} & & & 0 \\ & & & 1 \ 0 \\ & & 0 \ -1 & 0 \\ \text{TK}_4 = & 0 \ 0 & 1 \ (-1/4) & \end{array}$$

После передачи переноса из нулевой строки получаем частное

$$\begin{array}{cccc} & & & 0 \\ & & & 1 \ 0 \\ & & 0 \ -1 & 0 \\ \text{TK}_4 = & 0 \ 0 & 1 \ 0 & \end{array}$$

и остаток

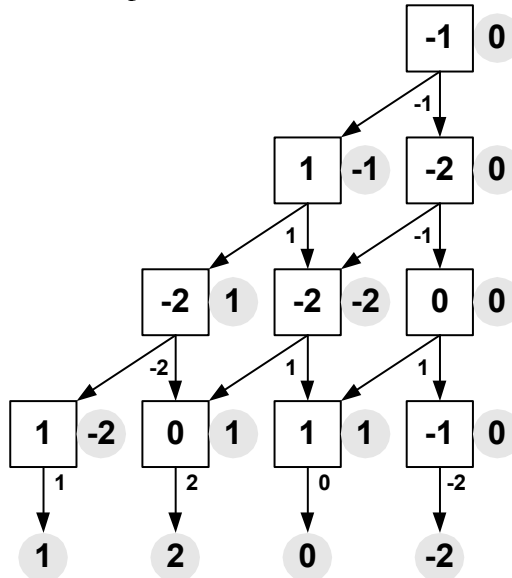
$$\text{ЛК}_4 = 0 \ 0 \ 0 \ (1/4)$$

**Пример 4.1.3. Деление на параметр** кода функции

$$\begin{array}{cccc} & & & -1 \\ & & & 1 \ -2 \\ & & -2 \ -2 & 0 \\ \text{TK}_4(F(x)) = & 1 \ 0 & 1 \ -1 & \end{array}$$

По табл. 4.1.3 строим схему распространения переносов, где в квадратах записаны разряды  $\alpha_{mk}$  делимого, в кругах – разряды

частного  $\sigma_{mk}$ , между стрелками – переносы  $\pi_{mk}$  из разряда, расположенного над стрелками.



Следовательно, смешанный код частного

$$\begin{array}{r}
 0 \\
 -1 \ 0 \\
 1 \ 1 \ 0 \\
 SK_4(F(x)/R) = -2 \ 0 \ 2 \ 0 \\
 1 \ -2 \ -2 \ 0
 \end{array}$$

Сравнивая примеры 4.1.2 и 4.1.3 замечаем, что в них описаны обратные действия – коды на рисунках совпадают, рисунки различаются направлением стрелок-переносов и

$$F(x) = Rf(x), \quad SK_4(F(x)/R) = SK_4(f(x)), \quad TK_4(R \cdot f(x)) = TK_4(F(x))$$

### 4.1.3. Умножение.

Умножение четверичных треугольных кодов не имеет особенностей по сравнению с общим случаем.

**Пример 4.1.4. Умножение.** Найдем код  $TK_4(z(x))$  произведения  $\xi = uv$  функций, коды  $TK_4(u(x))$  и  $TK_4(v(x))$  которых приведены в примерах 4.2.3 и 4.2.4 соответственно.

4.1. Арифметические операции с четверичными треугольными кодами

									0
								0	0
							0	0	0
						0	0	0	1
				0	0	0	1	-1	1
			0	-2	0	0	-1	-1	1
		-2	-1	-1	0	-2	-2	1	
TK <sub>4</sub> (z(x))		-2	1	1	-1	0	-1	0	0
	-2	1	1	-1	0	0	-1	0	0

В примере 4.2.5 этот код декодирован и тем самым найдено произведение  $\mathfrak{Z}^{-III}$  функций, определенных в примерах 4.2.3 и 4.2.4.

## 4.2. Кодирование и декодирование четверичных треугольных кодов

В общем виде кодирование и декодирование четверичных треугольных кодов описано в разделе 2.4. Здесь будут указаны лишь особенности четверичных кодов.

### 4.2.1. Кодирование и декодирование четверичных треугольных кодов

Рассмотрим примеры.

**Пример 4.2.1. Кодирование и декодирование** при  $R = 4$ ,  $D_R = \{-2, -1, 0, 1\}$ . Вначале рассмотрим декодирование треугольного кода  $\text{TK}(F(x))$  из примера 4.1.3. При делении кода  $\text{TK}(F_0(x)) = \text{TK}(F(x))$  на 4 образуется код частного  $\text{TK}(F_1(x))$  и код остатка  $\text{AK}(f_0(x))$ . При делении кода  $\text{TK}(F_1(x))$  образуется код частного  $\text{TK}(F_2(x))$  и код остатка  $\text{AK}(f_1(x))$ . При делении кода  $\text{TK}(F_2(x))$  образуется код частного  $\text{TK}(F_3(x))$  и код остатка  $\text{AK}(f_2(x))$ . Последнее деление кода  $\text{TK}(F_3(x))$  дает нулевое частное и остаток  $\text{TK}(f_3(x)) = \text{TK}(F_3(x))$ . В обратном направлении этот же пример иллюстрирует процесс кодирования прямоугольного кода последовательным умножением его на параметр.

	0				0				
		0	0		-1 0				
			0	-1 0	1 -1 1				
$\text{TK}(F_0(x)) =$	1	0	1	-1	$\text{TK}(F_1(x)) =$	-2	-2	-2	1
					$\text{AK}(f_0(x)) =$	1	-2	0	-2
			0			0			
			-1	0		0	0		
$\text{TK}(F_2(x)) =$	1	-1	1		$\text{TK}(F_3(x)) =$	-1	0	0	
$\text{AK}(f_1(x)) =$	-2	1	-2		$\text{AK}(f_2(x)) =$	1	0	1	

#### 4.2. Кодирование и декодирование четверичных треугольных кодов

Следовательно,

$$\begin{aligned}
 & \begin{array}{cccc} -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -2 & 1 & -2 & 0 \\ 1 & -2 & 0 & -2 \end{array} = \begin{array}{l} \Lambda K(f_3(x)) \\ \Lambda K(f_2(x)) \\ \Lambda K(f_1(x)) \\ \Lambda K(f_0(x)) \end{array} \\
 PK(F(x)) = & \\
 A_{\psi} = & \begin{array}{cccc} -55 & 2 & 8 & -2 \end{array}
 \end{aligned}$$

Итак,  $F(x) = -55 + 2\psi_{01} + 8\psi_{02} - 2\psi_{03}$ .

#### Пример 4.2.2. Преобразование кода ТК(F(x)) в код РК(F(x))

последовательным делением на параметр.

								0
								0 1
							-1	-2 1
						1	0	-1 1
					-1	1	0	0 1
				-1	-1	1	-2	1 1
			-1	-1	-1	1	-1	-1 1
		0	-1	-1	1	1	0	1 0
TK =	0	-2	0	-1	0	1	0	0 0

								0
								0 1
							-1	-2 0
						1	1	1 1
					-1	0	-1	-1 1
				-1	0	1	-1	-2 1
			-1	0	-1	1	1	1 0
		0	0	-1	-2	-2	1	0 0
СК1 =	0	-2	0	0	-2	0	0	0 0

								0
								0 0
							0	1 0
						-1	-1	0 1
					1	-2	0	-2 1
				-1	-1	-2	1	-1 1
			-1	1	-2	-1	-1	-1 1
		-1	1	-2	-1	-1	0	-2 1
	0	1	-2	0	-1	-2	1	-2
СК2 =	0	-2	0	0	-2	0	0	0 0



							0
						0	0
				0	1	0	
			-1	-1	-1	1	
		1	-1	1	-1	0	
	-1	-1	-1	0	0	1	
	-1	-1	0	0	-1	-1	0
	-1	-2	0	-1	-1	1	-1
	0	1	-2	0	-1	-2	1
CK3=	0	-2	0	0	-2	0	0

							0
						0	0
				0	0	0	
			0	1	0	1	
		-1	-1	-1	-2	1	
	1	0	-2	-2	0	1	
	-1	-2	0	-2	-1	-2	1
	-1	0	-2	1	1	0	-2
	-1	-2	0	-1	-1	1	-1
	0	1	-2	0	-1	-2	1
CK4=	0	-2	0	0	-2	0	0

							0
						0	0
				0	0	0	
			0	1	0	1	
		-1	-1	-2	-2	1	
	1	0	-1	0	-2	1	
	-1	1	-1	-1	-1	0	0
	-1	0	-2	1	1	0	-2
	-1	-2	0	-1	-1	1	-1
	0	1	-2	0	-1	-2	1
CK5=	0	-2	0	0	-2	0	0

							0
						0	0
			0	0	0		
		0	1	-1	1		
	-1	-1	1	-2	0		
	1	1	0	-1	0	1	0
	-1	1	-1	-1	-1	0	0
	-1	0	-2	1	1	0	-2
	-1	-2	0	-1	-1	1	-1
	0	1	-2	0	-1	-2	1

## 4.2. Кодирование и декодирование четверичных треугольных кодов

СК6=	0	-2	0	0	-2	0	0	0
	0	1	-1	1	0	0	0	0
	-1	-1	0	-1	-1	0	0	0
	1	1	0	-1	0	1	0	0
	-1	1	-1	-1	-1	0	0	0
	-1	0	-2	1	1	0	-2	0
	-1	-2	0	-1	-1	1	-1	1
	0	1	-2	0	-1	-2	1	-2
PK=	0	-2	0	0	-2	0	0	0

A0	-3408
A1	13538
A2	-16776
A3	11056
A4	-4310
A5	1032
A6	-140
A7	8

### Пример 4.2.3. Кодирование функции

$$u(x) = -4 + 7\psi_{01} + 12\psi_{02} - 6\psi_{03} + \psi_{04} :$$

					0
					0
			0	1	1
TK <sub>4</sub> (u(x))		-1	-2	1	1
	0	-2	-2	-2	1

### Пример 4.2.4. Кодирование функции

$$v(x) = 10 + 48\psi_{01} - 36\psi_{02} + 8\psi_{03} :$$

					0
					0
			1	0	1
TK <sub>4</sub> (v(x))		-1	-2	-2	1
	-2	-1	1	-2	1

### Пример 4.2.5а. Декодирование кода

TK <sub>4</sub> =			-1
		1	-1
	1	0	-2

Выполним последовательное деление кода TK<sub>4</sub> на параметр:

			0
		-1	0
	1	-1	0
$\Delta K1=$	1	-1	-2
			0
		0	0
	-1	0	0
$\Delta K2=$	1	1	0
			0
		0	0
	0	0	0
$\Delta K3=$	-1	0	0

Формируем прямоугольный код:

	-1	0	0	$\Delta K3$
$RK_4$	1	1	0	$\Delta K2$
	1	-1	-2	$\Delta K1$

Декодирование прямоугольного кода  $RK_4$  дает функцию  $(-11 + 3\psi_{01} - 2\psi_{02})$

**Пример 4.2.5. Декодирование кода  $TK_4(z(x))$**  из примера 4.1.4. Выполним последовательное деление кода  $TK_4(z(x))$  на параметр:

				0	0	0	1
			0	0	1	-1	0
		0	-2	1	-2	0	1
	-2	0	-2	1	0	-2	0
	-2	0	0	-2	1	-1	-2
$\Delta K1=$	0	-2	0	0	-2	0	-1

				0	0	0	1	0
			0	0	1	-2	-1	0
		0	-2	1	1	1	1	0
	-2	1	0	0	-1	1	-2	1
$\Delta K2=$	-2	-2	-2	-2	1	0	1	-2

			0	0	1	1	0	0
		0	0	-2	-1	-1	-1	0
	0	-2	-2	1	-1	0	0	0
$\Delta K3=$	-2	1	-2	-1	-1	-2	-1	1

#### 4.2. Кодирование и декодирование четверичных треугольных кодов

		0	0	1	1	0	1	0
	0	0	-2	-1	-2	-2	1	0
ЛК4=	0	-2	-2	-1	1	-2	-1	0

	0	0	1	1	-1	1	0	0
ЛК5=	0	0	-2	-2	1	-2	0	0

Формирование прямоугольного кода:

	0	0	1	1	-1	1	-1	0	ЛК6
	0	0	-2	-2	1	-2	0	0	ЛК5
	0	-2	-2	-1	1	-2	-1	0	ЛК4
	-2	1	-2	-1	-1	-2	-1	1	ЛК3
РК <sub>4</sub> (z(x))	-2	-2	-2	-2	1	0	1	-2	ЛК2
	0	-2	0	0	-2	0	0	-1	ЛК1

Декодирование прямоугольного кода РК<sub>4</sub>(z(x)):

$$z = \begin{bmatrix} -40 - 122\psi_{01} + 344\psi_{02} + 424\psi_{03} \\ -718\psi_{04} + 352\psi_{05} - 76\psi_{06} + 7\psi_{07} \end{bmatrix}$$

#### 4.2.2. Кодирование и декодирование четверичных тригонометрических треугольных кодов

Рассмотрим вначале функции  $\gamma_{\omega}^i$  – см. теорему 2.1.3. Можно убедиться в следующих свойствах функций  $\gamma_{\omega}^i$  при  $R=4$  и  $n>1$ :

$$\psi_{01} \cdot \gamma'_{\omega} = \gamma_{\omega} + \gamma_{\omega-1},$$

$$\gamma_{\omega} = \gamma'_{\omega} + \gamma'_{\omega-1},$$

$$\psi_{01} \cdot \gamma'''_{\omega} = \gamma''_{\omega} + \gamma''_{\omega-1},$$

$$\gamma''_{\omega} = \gamma'''_{\omega} + \gamma'''_{\omega-1}.$$

Из этих формул при  $n>2$  следует:

$$\gamma_{\omega}^i = (\psi_{01} - 2)\gamma_{\omega-1}^i + \gamma_{\omega-2}^i.$$

Четверичные коды этих функций упрощаются по сравнению с общим случаем. Коды ТК<sub>4</sub>( $\gamma_{\omega}^i$ ) для младших значений  $n$  приведены в табл. 4.2.1. На использовании заранее рассчитанных таблиц кодов ТК<sub>4</sub>( $\gamma_{\omega}^i$ ) основан алгоритм 2.4.1 кодирования функций.

Таблица 4.2.1.

n	$TK_4(\gamma_\omega)$					$TK_4(\gamma'_\omega)$				
1		1					1			
	-2	0				-1	0			
2			0					0		
		1	-1				1	-1		
	-2	1	0			1	0	0		
3				0					0	
			0	1				0	1	
		0	-2	0			-2	-1	0	
	-2	1	0	0		1	0	0	0	
4					0					0
				0	0				0	0
			0	-1	1			1	-1	1
		1	0	-1	0		-2	-2	0	0
	-2	1	0	0	0	1	0	0	0	0
5					0					0
					0	0				0
				0	1	0			0	1
			0	-2	-1	1		-1	-1	-2
		0	-2	-1	0	0	1	0	-1	0
	-2	1	0	0	0	0	1	0	0	0
n	$TK_4(\gamma''_\omega)$					$TK_4(\gamma'''_\omega)$				
1		0					1			
	1	-1				-2	0			
2			0					0		
		0	-1				1	-1		
	1	1	0			-1	1	0		
3				0					0	
			0	1				0	1	
		0	-2	0			-1	-2	0	
	-1	-2	1	0		0	1	0	0	
4					0					0
				0	0				0	0
			0	0	1			0	-1	1
		0	-2	-1	0		1	-1	-1	0
	1	-2	1	0	0	1	1	0	0	0
5					0					0
					0	0				0
				0	0	0			0	1
			0	1	-2	1		0	-2	-1
		0	0	1	-1	0	-1	1	-2	0
	-1	-1	1	0	0	0	-2	-2	1	0

## 4.2. Кодирование и декодирование четверичных треугольных кодов

Рассмотрим примеры кодирования и декодирования четверичных кодов. Некоторые числа, которые будут использоваться в этих примерах, сведены в табл. 4.2.2-4.

Таблица 4.2.2. К примерам кодирования и декодирования.

$i$	0	1	2	3	3	3
$\omega$	0	1	1	0	1	2
$TK_4(\gamma_\omega^i)$	0 1 0	1 -1 0	0 1 -1	0 1 0	1 -2 0	1 -1 0
$D_\omega^i$	2	2	-2	-4	-2	-2
$E_\omega^i$	2	-1	1	-2	1	-1
$TK_4(E_\omega^i)$	1 -2 1	1	-1	-2	1	-1

Таблица 4.2.3. К примерам кодирования и декодирования.

$i$	0	1	1	2	2	3	3
$v$	0	0	1	0	1	0	1
$A_v^i$	2	-3	1	2	-2	3	-2

Таблица 4.2.4. К примерам кодирования и декодирования.

$i$	0	1	3	3
$PK_4(f^i(x))$	1 0 -2 0	-1 0 1 1	0 0 1 -1	1 0 -1 -2
$TK_4(f^i(x))$	1 -2 1	-1 1 0	0 -1 -1	-1 1 -1

**Пример 4.2.6. Кодирование функции по алгоритму 2.4.1** при  $R = 4$  и  $\alpha \in \{-2, -1, 0, 1\}$ .

Рассмотрим функцию

$$\Phi(x) = 2 - 4\sin 2x - 2\sin 3x + 2\cos 3x + 2\sin 4x - 2\sin 6x$$

1. выделяем из ряда  $\Phi(x)$  функции  $\eta^i f^i(x)$ , равные при  $i = 0, 1, 2, 3$  соответственно  $(2)$ ,  $(2\sin 3x)$ ,  $(-2\cos 3x)$ ,  $(-4\sin 2x - 2\sin 4x - 2\sin 6x)$ ; отсюда находим числа  $D_\omega^i$ , приведенные в табл. 4.2.2;

2. находим числа  $E_{\omega}^i$  при известных числах  $D_{\omega}^i$  по формуле (2.1.19) - см. табл. 4.2.2;
3. кодируем числа  $E_{\omega}^i$  и получаем коды  $\text{TK}_4(E_{\omega}^i)$ ;
4. пользуясь формулой (2.4.2), известными кодами  $\text{TK}_4(\nu_{\omega}^i)$  и вычисленными кодами  $\text{TK}_4(E_{\omega}^i)$ , вычисляем коды  $\text{TK}_R(\eta^i f^i(x))$ , равные при  $i = 0, 1, 2, 3$  соответственно:
 
$$\left\{ \begin{array}{c} 1 \\ -2 \end{array} \right\}, \left\{ \begin{array}{cc} -1 & \\ & 0 \end{array} \right\}, \left\{ \begin{array}{cc} 0 & \\ & -1 \end{array} \right\}, \left\{ \begin{array}{cc} 0 & \\ & -2 \end{array} \right\} + \left\{ \begin{array}{cc} 1 & \\ & -2 \end{array} \right\} + \left\{ \begin{array}{cc} -1 & \\ & 0 \end{array} \right\} = \left\{ \begin{array}{cc} -1 & \\ & 1 \end{array} \right\}$$
 - см. также табл. 4.2.4;
5. компонуем код

$$\text{TKK}_4(\Phi(x)) = \begin{pmatrix} 0 \\ 1 & -1 \\ 1 & -1 & 0 \\ -2 & 1 & -1 & -1 \\ -2 & 1 & 1 & 0 & 0 \end{pmatrix}$$

**Пример 4.2.7. Кодирование функции по алгоритму 2.4.2**  
при условиях примера 4.2.6

1. находим числа  $D_{\omega}^i$  - см. п. 1 примера 4.2.6
2. при известных числах  $D_{\omega}^i$  и  $S^i(\nu, \omega)$  находим числа  $A_{\nu}^i$  по формуле (2.1.18) - см. табл. 4.2.3;
3. кодируем функции  $f^i(x)$ , заданные числами  $A_{\nu}^i$ , то есть формируем коды  $\text{PK}_4(f^i(x))$ , а затем - коды  $\text{TK}_4(f^i(x))$  - см. табл. 4.2.4; эти коды совпадают с теми, которые получены в п. 4 примера 4.2.6;
4. компонуем код  $\text{TK}_4(\Phi(x))$  - см. п. 5 примера 4.2.6.

**Пример 4.2.8. Декодирование кода по алгоритму 2.4.3.**

Рассмотрим код  $\text{TK}_4(\Phi(x))$ , полученный в примере 4.2.6.

1. декомпозируем код  $\text{TK}_4(\Phi(x))$ , что приводит к образованию кодов  $\text{TK}_4(f^i(x))$  - см. табл. 4.2.4;

2. декодируем полученные коды для получения чисел  $A_v^i$  - см. табл. 4.2.3;
3. при известных числах  $A_v^i$  и  $L^i(v, n)$  вычисляем числа  $D_\omega^i$  по формуле (2.1.17) - см. табл. 4.2.2;
4. формируем тригонометрический ряд функции  $\Phi(x)$  при известных числах  $D_\omega^i$  - см. условие примера 4.2.6.

**Пример 4.2.9. Кодирование функции в ГТК по алгоритму 2.4.1.**

Рассмотрим функцию из примера 2.2.2 с использованием известных кодов  $TK_4(\gamma_\omega^i)$ , приведенных в табл. 4.2.1. Применяя формулу (2.1.19) для преобразования ряда (2.1.8) в ряд (2.1.10), из примера 2.2.2 находим:

$$F(x) = 32 + 21\gamma_1^0 - 4\gamma_2^0 + 2\gamma_3^0 - \gamma_4^0,$$

$$F'(x) = 2\text{Sin}x(128 - 5\gamma_1^1 + \gamma_2^1),$$

$$F'(x) = 2\text{Cos}x(128 + 5\gamma_1^2 + \gamma_2^2)$$

$$F'''(x) = 2\text{Sin}x\text{Cos}x(42 - 16\gamma_1^3 + 12\gamma_2^3 - 8\gamma_3^3)$$

	32				
$21\gamma_1^0$	0				
	-42	21			
$-4\gamma_2^0$			0		
		-4	4		
$2\gamma_3^0$	8	-4	0		
				0	
			0	2	
$-\gamma_4^0$		0	-4	0	
	-4	2	0	0	
					0
				0	0
			0	1	-1
$F^0(x)$		-1	0	1	0
	2	-1	0	0	0
					0
				0	0
			0	3	-1
$F^0(x)$		-5	0	1	0
	-4	18	0	0	0



	128		
$-5\gamma_1^1$		5	
	-5	0	
$\gamma_2^1$			0
		1	-1
	1	0	0
$F^1(x)$			0
		6	-1
$2\text{Sin}x$	124	0	0

	128		
$5\gamma_1^2$		0	
	-5	5	
$\gamma_2^2$			0
		0	-1
	1	1	0
$F^2(x)$			0
		0	-1
$2\text{Cos}x$	124	6	0

	42			
$-16\gamma_1^3$		0		
	32	-16		
$12\gamma_2^3$			0	
		12	-12	
	-12	12	0	
$-8\gamma_3^3$				0
			0	-8
		8	16	0
	0	-8	0	0
$F^3(x)$				0
			0	-8
$2\text{Sin}2x$		20	4	0
	62	-12	0	0

Кодируя полученные ТК  $(F^i(x)/\eta^i)$  в  $\text{TK}_4^i = \text{TK}_4(F^i(x)/\eta^i)$ , находим:

4.2. Кодирование и декодирование четверичных треугольных кодов

$$TK_4^0 = \begin{matrix} & & & & & 0 \\ & & & & 0 & 1 \\ & & & -1 & 0 & 1 \\ & & -2 & -1 & -1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{matrix}$$

$$TK_4^1 = \begin{matrix} & & & & & & & & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & 0 & 0 & 0 \\ & & & & & 1 & 1 & 1 & 1 \\ & & & -2 & -1 & -2 & -2 & 1 \\ & & 1 & -2 & 0 & -2 & 1 & 0 \\ 0 & 1 & 0 & -2 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -2 & 1 & 0 & 0 & 0 \end{matrix}$$

$$TK_4^2 = \begin{matrix} & & & & & & & & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & 0 & 0 & 0 \\ & & & & & 1 & 1 & 1 & 1 \\ & & & -2 & -1 & -2 & -2 & 1 \\ & & 0 & -2 & 0 & -2 & 1 & 0 \\ 0 & -1 & 0 & -2 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & -2 & 1 & 0 & 0 & 0 \end{matrix}$$

$$TK_4^3 = \begin{matrix} & & & & & 1 \\ & & & & -2 & 1 \\ & & & 1 & -2 & 0 \\ 0 & 0 & -2 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \end{matrix}$$

Расширяя коды  $TK_4^i$ , получаем тригонометрические четверичные коды:

$$TTK_4(F(x)) =$$

$$\begin{matrix} & & & & & & & & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & 0 & 0 & 1 \\ & & & & & 0 & 0 & 0 & 0 \\ & & & -1 & 0 & 0 & 0 & 1 \\ & & 0 & 0 & 0 & 0 & 0 & 0 \\ & -2 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{matrix}$$

$$\text{ТТК}_4(F''(x)) =$$

								1	0	1	0	1	0	1
								0	0	0	0	0	0	0
							-2	0	-1	0	-2	0	-2	0
						0	0	0	0	0	0	0	0	0
					0	0	-2	0	0	0	-2	0	1	0
				0	0	0	0	0	0	0	0	0	0	0
			-1	0	0	0	-2	0	-1	0	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	-2	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$\text{ТТК}_4(F'(x)) =$$

								0	1	0	1	0	1	0
								0	0	0	0	0	0	0
						0	-2	0	-1	0	-2	0	-2	0
				0	0	0	0	0	0	0	0	0	0	0
			0	1	0	-2	0	0	0	-2	0	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	-2	0	-1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	-2	0	1	0	0	0	0	0

$$\text{ТТК}_4(F'''(x)) =$$

														0
														0
														0
									0	-2	0	1		
							0	0	0	0	0			
						0	1	0	-2	0	1			
					0	0	0	0	0	0	0			
				0	0	0	-2	0	0	0	0	1		
			0	0	0	0	0	0	0	0	0			
		0	-2	0	0	0	1	0	0	0	0			
0	0	0	0	0	0	0	0	0	0	0	0			

**Пример 4.2.10. Коды функций  $\varepsilon_{\omega}^i$ .** Еще раз отметим, что функции  $\varepsilon_{\omega}^i$  имеют простые коды  $\text{ТТК}_4(\varepsilon_{\omega}^i)$ . В частности, из

табл. 2.2.1 следует, что

$$\text{ТТК}_4(2\text{Sin}(x)) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{matrix} & & 0 & & & & 0 \\ & & 1 & 0 & & & 0 & 1 \end{matrix} \text{ . Далее}$$

$$\underline{\text{ТТК}}_4(2\text{Cos}(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \underline{\text{ТТК}}_4(2\text{Sin}(2x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

коды  $\text{ТТК}_4(\varepsilon_\omega^i)$  могут быть получены по одному из вышеописанных алгоритмов кодирования. Другой способ – использование простых формул тригонометрии. Например, из

$$2\text{Cos}(2x) = 2 - 4\text{Sin}^2(x), \quad \text{ТТК}_4(4\text{Sin}^2(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ следует,}$$

что  $\underline{\text{ТТК}}_4(2\text{Cos}(2x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$

Далее, из  $2\text{Sin}(3x) = 6\text{Sin}(x) - 8\text{Sin}^3(x),$

$$\text{ТТК}_4(6\text{Sin}(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad \text{ТТК}_4(8\text{Sin}^3(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

следует, что  $\underline{\text{ТТК}}_4(2\text{Sin}(3x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}.$  Наконец, из

$$2\text{Cos}(3x) = -6\text{Cos}(x) + 8\text{Cos}^3(x),$$

$$\text{ТТК}_4(6\text{Cos}(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{ТТК}_4(8\text{Cos}^3(x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

следует, что  $\underline{\text{ТТК}}_4(2\text{Cos}(3x)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$

### 4.3. Математические операции с четверичными тригонометрическими треугольными кодами

В общем виде эти операции описаны в разделе 2.3. Здесь будут указаны лишь особенности четверичных кодов. Рассмотрим примеры.

#### **Пример 4.3.1. Сдвиг оси ординат.**

Проверим преобразованием ТТК справедливость равенства

$$F''(x) = F'(x + \pi/2)$$

где функции  $F''(x)$ ,  $F'(x)$  определены в примере 2.2.2. Из примера 4.2.9 находим ТТК<sub>4</sub>( $F'(x)$ ). Применяя к нему формулу

$$\psi'_{mk}(x) = (-1)^m \psi_{k-m,k}(x)$$

получаем ТТК<sub>4</sub>( $F''(x)$ ), приведенный в примере 3.3.

#### **Пример 4.3.2. Умножение.**

Найдем произведение  $\Phi_1(x) = F(x) \cdot F'''(x)$ , где коды функций  $F(x)$ ,  $F'''(x)$  определены в примере 4.2.9 – см. коды ТК<sub>4</sub><sup>0</sup> и ТК<sub>4</sub><sup>3</sup> соответственно. При этом

									0
								0	1
							-1	-2	1
					1	0	-1	1	
				-1	1	0	0	1	
			-1	-1	1	-2	1	1	
		-1	-1	-1	1	-1	-1	1	
ТК <sub>4</sub> ( $\Phi_1(x)$ ) =	0	0	-1	-1	1	1	0	1	0
	0	-2	0	-1	0	1	0	0	0

#### **Пример 4.3.3. Дифференцирование.**

Проверим дифференцированием ТТК справедливость равенства

$$F'''(x) = \frac{dF(x)}{dx}$$

где функции  $F'''(x)$ ,  $F(x)$  определены в примере 2.2.2. Из примера 4.2.9 находим  $\text{ТТК}_4(F(x))$ . Производя вычисление по формуле (2.3.1), получаем

									0
								0	2
							0	0	0
							0	0	-2
						0	0	0	0
				0	2	0	-4	0	2
			0	0	0	0	0	0	0
ТТК( $F'''(x)$ ) =		0	6	0	2	0	8	0	-2
	0	0	0	0	0	0	0	0	0

Преобразуя этот код в четверичный, находим  $\text{ТТК}_4(F'''(x))$ , приведенный в примере 4.2.9.

## 4.4. Укорочение четверичных тригонометрических треугольных КОДОВ

В общем виде укорочение четверичных тригонометрических треугольных кодов описано в разделе 2.6. Четверичные тригонометрические треугольные коды имеют особенность, состоящую в том, что функции вычисляются по упрощенным формулам и коды этих функций являются целыми. Действительно, из (2.6.6) следует:

$$\delta^i(n) = \sum_{v=0}^{n-1} \left[ (-1)^{n+v+1} S^i(v, n) \cdot \psi_{0v} \right] \quad (4.4.1)$$

Но числа  $S^i(v, n)$  являются целыми – см. теорему 2.1.1. Следовательно, коды  $\text{TK}_4(\delta^i(n))$  также являются целыми – см. теорему 1.6.1. Таким образом, функции  $\delta^i(n)$  имеют целые коды (а не смешанные, как это предполагалось для общего случая). Кроме того, коды  $\text{TK}_4(\delta^i(n))$  имеют ровно  $(n+1)$  значащих столбцов.

Смешанные коды  $B_n^i = \text{SK}_4\left(\frac{\delta_n^i}{4}\right)$  функций  $\frac{\delta_n^i}{4}$  могут быть получены (как и в общем случае) по формуле (2.6.4) при известных  $\eta^i \gamma_n^i$ . Но в данном случае эти смешанные коды имеют только одну строку в прямоугольной части кода (из-за того, что коды  $\text{TK}(\delta^i(n))$  являются целыми). Все это существенно упрощающими выполнение приближенных операций с  $\text{ТТК}_4$ . Для иллюстрации в табл. 4.4.1 приведены коды функций,  $\gamma_n^0$ ,  $\delta_n^0$ ,  $\frac{\delta_n^0}{4}$  при  $n < 5$ . Для вычисления формула (2.6.4) была преобразована к виду

$$\delta_n^i = (-1)^{n+1} \gamma_n^i + \psi_{0n}.$$

Таблица 4.4.1.

$n$	$TK_4(\gamma_n^0)$	$TK_4(\delta_n^0)$	$B_n^0 = SK_4\left(\frac{\delta_n^0}{4}\right)$
0	1	0	0
1	1 -2 0	1 -2 1	1 ----- -2
2	1 -1 -2 1 0	0 1 -2 0 1	0 0 1 ----- -2 0
3	1 -2 1 -2 0 0 0	1 -2 1 -2 0 0 1	0 0 1 1 -2 -2 1 ----- -2 -1 -2 0
4	0 -1 1 1 0 -1 0 -2 1 0 0 0	0 1 -1 0 0 1 0 -2 0 0 0 1	0 0 0 1 -1 0 0 0 1 ----- -2 0 0 0

Как указывалось, укорочение сосчит в делении на параметр и округлении полученного смешанного кода, имеющего одну строку в прямоугольной части. Рассмотрим алгоритм округления такого смешанного кода, использующий указанные свойства кодов

$B_n^i = SK_4\left(\frac{\delta_n^i}{4}\right)$  и таблицы этих кодов. Этот алгоритм основан на

замене разряда  $\pi_{0v}$  этого смешанного кода  $SK_4^{old}$  на код  $\pi_{0v}B_v^i$ .

Новое значение округляемого смешанного кода вычисляется как сумма смешанных кодов:

$$SK_4^{new} = SK_4^{old} + \pi_{0v}(B_v^i - \psi_{-1,v}). \quad (4.4.2)$$

При этом могут встретиться случаи сложения с кодом  $B_v^i$  и вычитания кодов  $B_v^i$  или  $2B_v^i$ , так как  $\pi_{0v} \in \{-2, -1, 0, 1\}$ .



**Алгоритм 4.4.1. Округление кода  $SK_4(f^i(x))$ .**

1. Присвоение значения  $SK_4^{old} = SK_4(f^i(x))$ .
2. Последовательный поиск старшего значащего разряда  $\pi_{0v}$  в младшей (-1)-строке округляемого кода  $SK_4^{old}$ . Если значащий разряд при  $v > 0$  есть, то выполняется переход к следующему пункту. В противном случае выполняется переход к пункту 5.
3. Стирание разряда  $\pi_{0v}$  и алгебраическое сложение результата с кодом  $\pi_{0v}B_v^i$ , то есть реализация формулы (4.4.2).
4. Присвоение значения  $SK_4^{old} = SK_4^{new}$  и переход к пункту 2.
5. Прекращение счета, поскольку  $SK_4^{old}$  не имеет (-1)-строки, то есть является треугольным кодом  $SK_4(f_0^i(x))$ , где  $f_0^i(x)$  - округленная функция  $f^i(x)$ . При этом погрешность округления  $\Delta = \Delta_n^i \left| \pi_{0v} \right|_{\max} / R$ . Учитывая что  $\Delta_n^i = 2$  и  $\left| \pi_{0v} \right|_{\max} = 2$ , находим  $\Delta_{\max} = 1$

**Пример 4.4.1. Укорочение.**

Пусть  $R=4$ ,  $\alpha \in \{-2, -1, 0, 1\}$  и известен код, который необходимо укоротить на один столбец:

					0
				-1	-1
			1	1	-1
$TK_4(F^0(x)) =$		-2	0	-2	0
	1	0	-1	-2	0

После деления кода данной функции на параметр 4 получаем смешанный код функции  $f^0(x) = F^0(x)/4$ :

					0
				-1	0
$SK_4(f^0(x)) =$		1	-2	0	
		-2	0	0	0
	1	-2	0	-2	

После округления этого кода, выполненного в примере 4.4.2, получаем

$$TK_4(f_0^0(x)) = \begin{array}{|c|c|c|} \hline & & -1 \\ \hline & 1 & -1 \\ \hline 1 & 0 & -2 \\ \hline \end{array}$$

**Пример 4.4.2. Округление.**

Округлим код  $SK_4(f_0^0(x))$ , полученный в примере 4.4.1, по алгоритму 4.4.1.

1. Присваиваем  $SK_4^{old} = SK_4(f_0^0(x))$ .
2. Находим  $\pi_{03} = -2$ .
3. Выбираем из табл. 4.4.1 код  $B_3^0$  и вычисляем код

$$\pi_{03} B_3^0 = -2 B_3^0:$$

$$B_3^0 = \begin{array}{|c|c|c|c|} \hline & 0 & 0 & 1 \\ \hline 1 & -2 & -2 & 1 \\ \hline -2 & -1 & -2 & 0 \\ \hline \end{array}$$

$$-2 B_3^0 = \begin{array}{|c|c|c|} \hline & 0 & 1 \\ \hline -1 & 1 & -2 \\ \hline 0 & -2 & 0 \\ \hline \end{array}$$

4. Складываем полученный код с кодом  $SK_4^{old}$ , в котором стерт разряд  $\pi_{03}$ , т.е. вычисляем код  $SK_4^{new}$ :

$$\begin{array}{|c|c|c|} \hline & & -1 \\ \hline & 1 & -2 \\ \hline -2 & 0 & 0 \\ \hline 1 & -2 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline & & 0 \\ \hline & 0 & 1 \\ \hline -1 & 1 & -2 \\ \hline 0 & -2 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & & -1 \\ \hline & 0 & -1 \\ \hline 1 & 0 & -2 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$$

Это эквивалентно тому, что к функции  $f^0(x)$ , представленной кодом  $SK_4^{old}$ , прибавлена функция  $(\delta_3^n - \psi_{0n})/4 = (2\gamma_3^0)/4$ .

5. Присваиваем  $SK_4^{old} = SK_4^{new}$ .
6. Обнаруживая, что  $\pi_{0v} = 0$  при  $v > 0$ , прекращаем счет. Это эквивалентно тому, что к функции, представленной кодом  $SK_4^{new}$ , прибавлена (-1). В результате получаем

$$TK_4(f_0^0(x)) = \begin{array}{|c|c|c|} \hline & & -1 \\ \hline & 1 & -1 \\ \hline 1 & 0 & -2 \\ \hline \end{array}$$

Таким образом,  $4[f^0(x) - f_0^0(x)] = [2\gamma_3^0 - 1]$ .

**Пример 4.4.3. Оценка погрешности округления.**

Для оценки погрешности округления в примере 4.4.2 заметим, что по существу вычислений  $4[f^0(x) - f_0^0(x)] = [2\gamma_3^0 - 1]$ . Из примера 2.1.2 следует, что  $[2\gamma_3^0 - 1] = [-1 - 4\text{Cos}(6x)]$ . Таким образом, абсолютная погрешность вычислений является функцией

$$[f^0(x) - f_0^0(x)] = -0.25 - \text{Cos}(6x),$$

то есть абсолютная погрешность коэффициентов тригонометрического ряда  $\Delta = 1$ . Для определения относительной погрешности округления кода в примере 4.4.1 декодируем результат округления  $\text{TK}_4(f_0^0(x))$ , как показано в примере 4.2.5а, и получаем функцию

$$f_0^0(x) = (-11 + 3\psi_{01} - 2\psi_{02})$$

Далее, преобразуем эту функцию, как показано в примере 2.1.3а, и получаем

$$f_0^0(x) = -17 - 10\text{Cos}(2x) - 4\text{Cos}(4x).$$

Далее находим:

$$f^0(x) = -17.25 - 10\text{Cos}(2x) - 4\text{Cos}(4x) - \text{Cos}(6x).$$

Таким образом, относительная погрешность округления

$$\Delta / \left| D_\omega^0 \right|_{\max} = \Delta / \left| D_2^0 \right|_{\max} = 1/17 \approx 6\%.$$

## 4.5. Погрешность кодирования четверичных тригонометрических треугольных кодов

Выше (в разделе 2.5) было показано, что для оценки максимальной относительной погрешности приближенных вычислений необходимо построить функцию  $M_\omega$  максимально возможного модуля коэффициента  $D_\omega$ . В качестве иллюстрации рассмотрим построение кривой для кодов ТТК( $F(x)$ ) – см. свойство 2.2.3. Напомним, что числа  $p(m, k, \omega)$  и  $q(m, k, \omega)$  определены в теоремах 2.5.1 и 2.5.2 соответственно. Обозначим:

$\Pi_1(k, \omega)$  - сумма всех положительных чисел  $p(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

$\Pi_2(k, \omega)$  - сумма всех отрицательных чисел  $p(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

$Q_1(k, \omega)$  - сумма всех положительных чисел  $q(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

$Q_2(k, \omega)$  - сумма всех отрицательных чисел  $q(m, k, \omega)$  при постоянных  $k$  и  $\omega$ ;

Числа  $\Pi_1(k, \omega)$  и  $\Pi_2(k, \omega)$  приведены в табл. 2.5.1 раздела 2.5. Из формулы (2.5.8) при  $R=4$  находим:

$$Q(k, \omega) = \begin{cases} \Pi(k, \omega) & \text{if } \omega = 0, \\ 2\Pi(k, \omega) & \text{if } \omega > 0. \end{cases} \quad (4.5.1)$$

Далее при  $\alpha_{mk} \in \{-2, -1, 0, 1\}$  имеем:

$$\left[ \sum_{m=0}^k (\alpha_{mk} \cdot q(m, k, \omega)) \right]_{\max} = Q_1(k, \omega) + 2Q_2(k, \omega),$$

$$\left[ \sum_{m=0}^k (\alpha_{mk} \cdot q(m, k, \omega)) \right]_{\min} = 2Q_1(k, \omega) - Q_2(k, \omega).$$

Отсюда, учитывая (2.5.10), получаем:

$$|D_{\omega} > 0|_{\max} = \sum_{k=\omega}^n [Q_1(k, \omega) + 2Q_2(k, \omega)], \quad (4.5.2)$$

$$|D_{\omega} < 0|_{\max} = \sum_{k=\omega}^n [2Q_1(k, \omega) - Q_2(k, \omega)]. \quad (4.5.3)$$

Обозначим:

$$\mu_1(\omega, n) = \sum_{k=\omega}^n [\Pi_1(k, \omega)], \quad (4.5.4)$$

$$\mu_2(\omega, n) = \sum_{k=\omega}^n [\Pi_2(k, \omega)]. \quad (4.5.5)$$

Числа  $\mu_1(\omega, n)$ ,  $\mu_2(\omega, n)$  при  $\omega \leq 5$ ,  $n \leq 5$  приведены в табл. 4.5.1 и 4.5.2.

Таблица 4.5.1. Числа  $\mu_1(\omega, n)$

$n \setminus \omega$	0	1	2	3	4	5
0	1					
1	5	1				
2	19	5	2			
3	67	21	14	2		
4	233	81	70	14	3	
5	817	307	310	75	27	3

Таблица 4.5.2. Числа  $\mu_2(\omega, n)$

$n \setminus \omega$	0	1	2	3	4	5
0	0					
1	0	-1				
2	0	-5	-1			
3	0	-21	-5	-2		
4	0	-81	-17	-14	-2	
5	0	-307	-49	-75	-14	-3

Имея в виду формулу (4.5.3), а также то, что  $p(m, k, \omega=0) > 0$  и, следовательно,  $\Pi_2(k, \omega=0) = 0$ , находим:

$$|D_{\omega} > 0|_{\max} = \begin{cases} \mu_1(\omega, n) & \text{if } \omega = 0, \\ (2\mu_1(\omega, n) - 4\mu_2(\omega, n)) & \text{if } \omega > 0, \end{cases} \quad (4.5.6)$$

$$|D_{\omega} < 0|_{\max} = \begin{cases} 2\mu_1(\omega, n) & \text{if } \omega = 0, \\ (4\mu_1(\omega, n) - 2\mu_2(\omega, n)) & \text{if } \omega > 0, \end{cases} \quad (4.5.7)$$

Числа (4.5.6) и (4.5.7) при  $\omega \leq 5$ ,  $n \leq 5$  приведены в табл. 4.5.3 и 4.5.4. В этих же таблицах приведены аналогичные величины, подсчитанные для  $n$ -разрядного четверичного кода числа по формулам:

$$|D > 0|_{\max} = \sum_{k=0}^n 4^k = \frac{1}{3}(4^{n+1} - 1),$$

$$|D < 0|_{\max} = 2 \sum_{k=0}^n 4^k = \frac{2}{3}(4^{n+1} - 1).$$

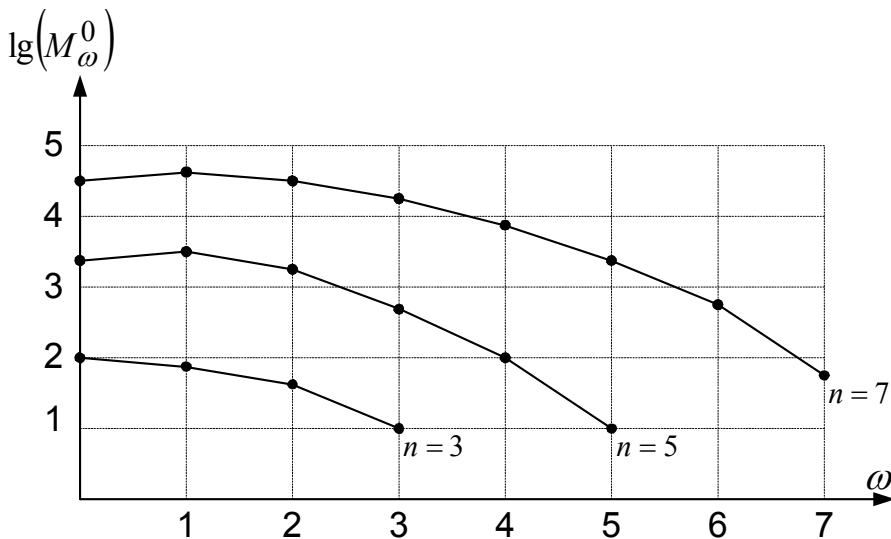
Таблица 4.5.3. Числа  $|D_{\omega} > 0|_{\max}$ .

$n \setminus \omega$	0	1	2	3	4	5	$ D > 0 _{\max}$
0	1						1
1	5	6					5
2	19	30	8				21
3	67	126	48	12			85
4	233	1842	208	84	14		341
5	817	6930	816	450	110	18	1365

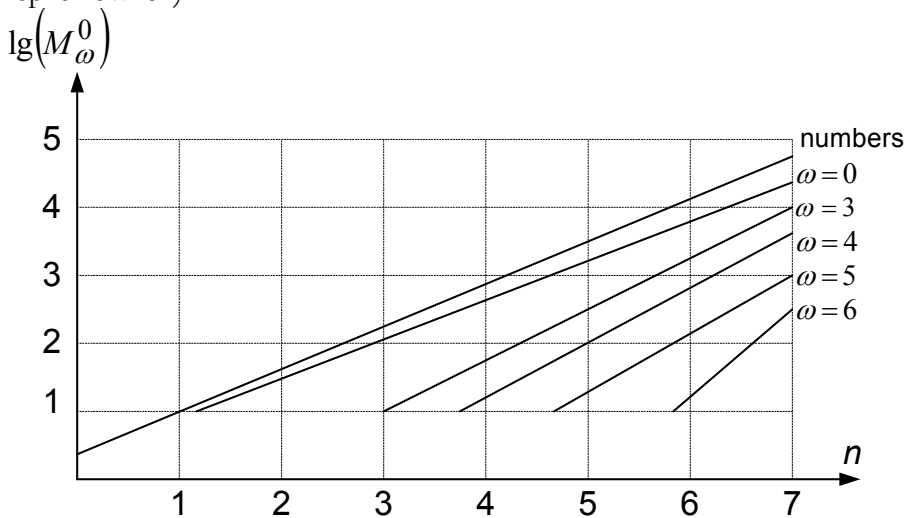
Таблица 4.5.4. Числа  $|D_{\omega} < 0|_{\max}$ .

$n \setminus \omega$	0	1	2	3	4	5	$ D < 0 _{\max}$
0	2						2
1	10	6					10
2	38	30	10				42
3	134	126	66	12			170
4	466	486	314	84	16		682
5	1634	1842	1338	450	136	18	2730

Из табл. 4.5.3 и 4.5.4 видно, что  $M_{\omega} = |D_{\omega} < 0|_{\max}$ . Зависимости  $\lg(M_{\omega}^0)$  от  $\omega$  при различных значениях  $n$  приведены на рис. 4.5.1.

Рис. 4.5.1. Зависимость  $M_\omega^0$  от  $\omega$ .

На рис. 4.5.2 изображены зависимости  $\lg(M_\omega^0)$  от  $n$  при различных значениях  $\omega$ . Там же для сравнения приведена зависимость максимального модуля действительного числа в системе кодирования  $\langle \rho = 4, A_4 = \{-2, -1, 0, 1\} \rangle$  (в этом случае индекс  $\omega$  теряет смысл).

Рис. 4.5.2. Зависимость  $M_\omega^0$  от  $n$ .





# Глава 5. Арифметическое устройство для операций с функциями

В данной главе подробно рассматривается арифметическое устройство для операций четверичными тригонометрическими треугольными кодами функций. Операционные устройства этого АУ должны оперировать с четверичными разрядами, принимающими значения из множества  $D_R = \{-2, -1, 0, 1\}$ . Для представления таких разрядов и описания операций с ними мы воспользуемся методом кодирования вещественных чисел по отрицательному основанию «-2». Будем называть **М-кодами** коды вещественных чисел по отрицательному основанию «-2» (в отличие от прямых двоичных кодов положительного числа, которые будем называть **Р-кодами**). Удобство такого кодирования заключается в том, что с М-кодами просто реализуются операции с положительными и отрицательными числами – не требуется схем преобразования чисел в обратный или дополнительный коды, не требуется схем управления для анализа знаков.

М-код некоторого числа  $a$  будем обозначать как  $K(a)$ . Например,  $K(0)=00$ ,  $K(1)=01$ ,  $K(-1)=11$ ,  $K(-2)=10$ . Схемы алгебраического суммирования М-кодов строятся на основе т.н. обратного сумматора, реализующего формулу  $c = (-a-b)$ . Объем обратного сумматора М-кодов равен объему обычного сумматора Р-кодов.

## 5.1. Одноразрядные схемы

### 5.1.1. Одноразрядный сумматор.

Из формул (1.3.1), (1.3.2) и табл. 4.1.2 следует, что числа  $\alpha_{mk}$ ,  $\beta_{mk}$ ,  $\pi'_{mk}$ ,  $\pi''_{mk}$ ,  $\pi_{mk}$ ,  $\sigma_{mk}$  имеют двухразрядные М-коды, а число  $S_{mk}$  имеет четырехразрядный М-код. Кроме того, число  $S_{mk}$  может быть представлено в виде:

$$S_{mk} = \sigma_{mk} + (-2)^2 \pi_{mk} = (\alpha_{mk} + \pi'_{mk}) + (\beta_{mk} + \pi''_{mk}).$$

Схема одноразрядного сумматора четверичных кодов, в которой учтены эти особенности, приведена на рис. 5.1.1, где

- $\sum_n$  -  $n$ -разрядный сумматор М-кодов,
- $\xrightarrow{n}$  -  $n$ -канал передачи разрядов.

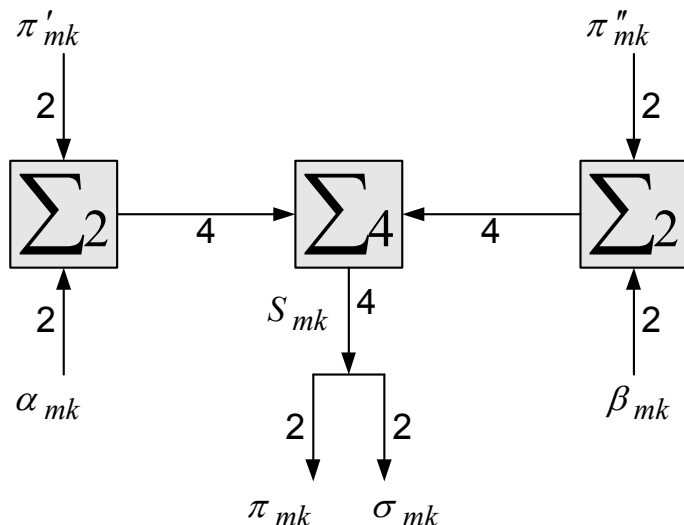


Рис. 5.1.1. Одноразрядный сумматор четверичных кодов

Заметим, что объем этого сумматора равен объему 8-ми одноразрядных сумматоров чисел. Поскольку этот сумматор складывает двухразрядные коды, можно сказать, что относительный (отнесенный к разрядности) объем сумматора ТК в 4 раза больше относительного объема сумматора чисел.

Задержка в этом сумматоре в 6 раз больше задержки в одноразрядном сумматоре чисел.

### 5.1.2. Одноразрядный вычитатель.

По аналогии с одноразрядным сумматором строится одноразрядный вычитатель четверичных кодов. В этом случае число  $S_{mk}$  может быть представлено в виде:

$$S_{mk} = \sigma_{mk} + (-2)^2 \pi_{mk} = (\alpha_{mk} + \pi'_{mk}) + (-\beta_{mk} + \pi''_{mk}).$$

Схема одноразрядного вычитателя четверичных кодов приведена на рис. 5.1.2 и отличается от схемы сумматора только тем, что в ней один из сумматоров заменен на вычитатель, обозначенный так:

- $\overline{\sum}_n$  -  $n$ -разрядный вычитатель М-кодов.

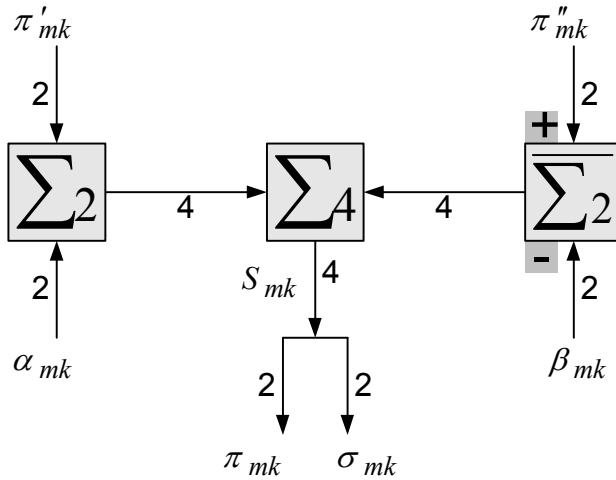


Рис. 5.1.2. Одноразрядный вычитатель четверичных кодов

### 5.1.3. Одноразрядный инвертор.

По аналогии с вычитателем строится одноразрядный инвертор четверичных кодов. В этом случае числа  $\alpha_{mk}$ ,  $\sigma_{mk}$  имеют двухразрядные М-коды, числа  $\pi'_{mk}$ ,  $\pi''_{mk}$ ,  $\pi_{mk}$  имеют двухразрядные М-коды, число  $S_{mk}$  имеет трехразрядный М-код и может быть представлено в виде:

$$S_{mk} = \sigma_{mk} + (-2)^2 \pi_{mk} = (\pi'_{mk} + \pi''_{mk}) - (\alpha_{mk}).$$

Схема одноразрядного инвертора четверичных кодов приведена – см. рис. 5.1.3.

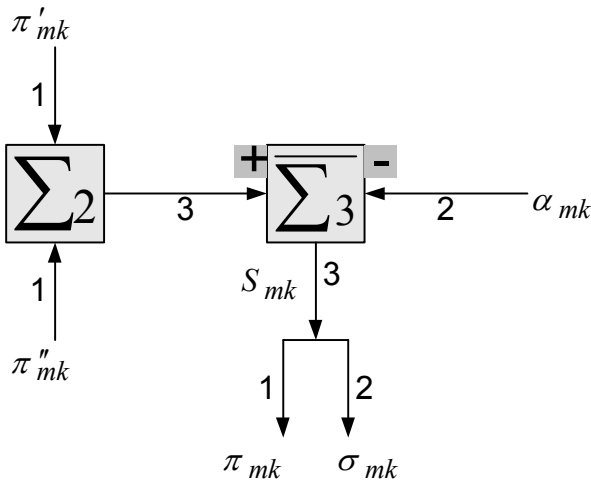


Рис. 5.1.3. Одноразрядный инвертор четверичных кодов

### 5.1.4. Одноразрядный учетверитель.

По аналогии с сумматором строится одноразрядный инвертор четверичных кодов. В этом случае числа  $\alpha_{mk}$ ,  $\sigma_{mk}$  имеют двухразрядные М-коды, числа  $\pi'_{mk}$ ,  $\pi''_{mk}$ ,  $\pi_{mk}$  имеют четырехразрядные М-коды, число  $S_{mk}$  имеет 6-тиразрядный М-код и может быть представлено в виде:

$$S_{mk} = \sigma_{mk} + (-2)^2 \pi_{mk} = (\pi'_{mk} + \pi''_{mk}) + (-2)^2 (\alpha_{mk}).$$

Схема одноразрядного учетверителя четверичных кодов приведена – см. рис. 5.1.4.

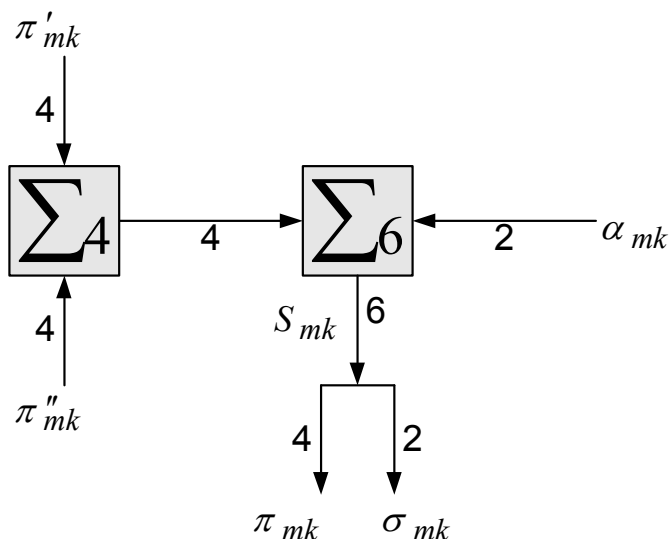


Рис. 5.1.4. Одноразрядный учетверитель четверичных кодов

### 5.1.5. Одноразрядный делитель на 4.

Из формул (1.4.1), (1.4.2) и табл. 4.1.3 следует, что числа  $\alpha_{mk}$ ,  $\pi'_{mk}$ ,  $\pi''_{mk}$ ,  $\pi_{mk}$  имеют двухразрядные М-коды, число  $\sigma_{mk}$  имеет четырехразрядный М-код, число  $4S_{mk}$  имеет 6-тиразрядный М-код и может быть представлено в виде:

$$4S_{mk} = \left( (-2)^2 \sigma_{mk} \right) + (\pi_{mk}) = (\alpha_{mk} - \pi'_{mk}) + \left( (-2)^2 \pi''_{mk} \right),$$

Схема одноразрядного сумматора четверичных кодов, в которой учтены эти особенности, приведена на рис. 5.1.5.

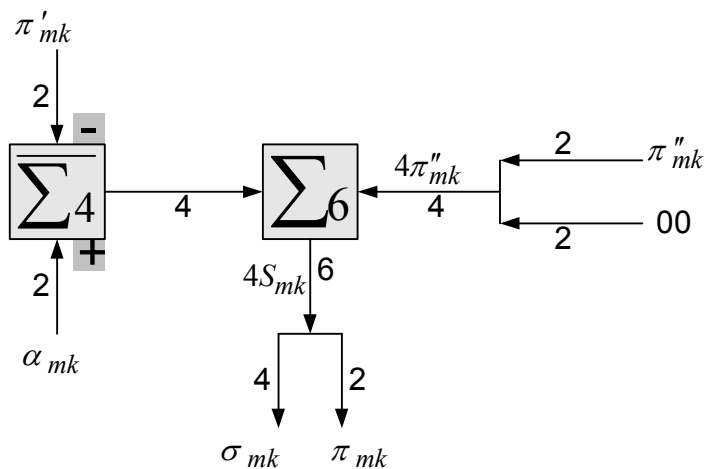


Рис. 5.1.5. Одноразрядный делитель на 4 четверичных кодах

## 5.2. Многоразрядные схемы

Многоразрядные схемы для операций с треугольными кодами строятся на основе описанных выше одnorазрядных схем. Можно предложить несколько типов многоразрядных схем, различающихся количеством параллельно работающих одnorазрядных схем и порядком чередования групп разрядов, обрабатываемых параллельно. Кроме того, тип схемы зависит от направления распространения переносов и по этому критерию схемы разбиваются на две группы:

- алгебраические сумматоры, в которых перенос распространяется вверх и влево,
- делители на параметр, в которых перенос распространяется вниз и вправо.

В дальнейшем из первой группы мы рассмотрим только сумматор, поскольку остальные схемы этой группы полностью ему аналогичны.

### 5.2.1. Столбцовый сумматор.

Этот сумматор изображен на рис. 5.2.1. Он складывает одноименные столбцы  $\alpha$  и  $\beta$  слагаемых кодов. При этом одноименные разряды столбцов складываются параллельно. При сложении образуется столбец результата  $\sigma$  и столбец выходного переноса  $\bar{\pi}$  в следующий столбец. На входе сумматора присутствует столбец входного переноса  $\underline{\pi}$  из предыдущего столбца. Очевидно, разряд входного переноса используется как входной перенос  $\pi'$  для одного из одnorазрядных сумматоров и как входной перенос  $\pi''$  для другого из одnorазрядных сумматоров.

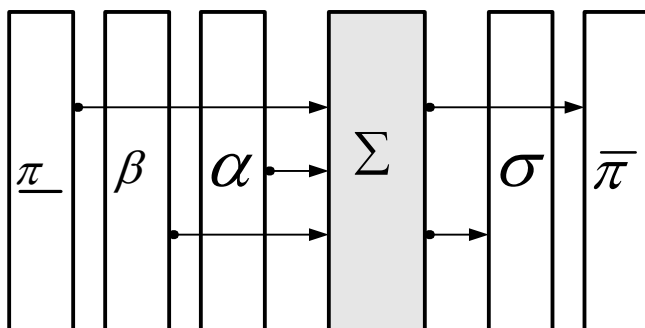


Рис. 5.2.1. Столбцовый сумматор.

Сложение кодов в целом состоит в последовательном сложении столбцов слева направо. Обозначим:

- $\tau$       длительность сложения на одноразрядной схеме,
- $t'$       длительность передачи из регистра в регистр,
- $t''$       длительность записи в регистр,
- $N$       количество столбцов и строк в треугольном коде.

Тогда полная длительность сложения на столбцовом сумматоре

$$t_1 = N(\tau + t' + t'')$$

### 5.2.2. Строчный сумматор.

Этот сумматор изображен на рис. 5.2.2. Он складывает одноименные строки  $\alpha$  и  $\beta$  слагаемых кодов. При этом выполняется распространение переносов  $\pi'$  через одноразрядные схемы. При сложении образуется строка результата  $\sigma$  и строка выходного переноса  $\bar{\pi}$  в следующую строку. На входе сумматора присутствует строка входного переноса  $\underline{\pi}$  из предыдущей строки. Очевидно, разряд входного переноса используется как входной перенос  $\pi''$  для одного из одноразрядных сумматоров.

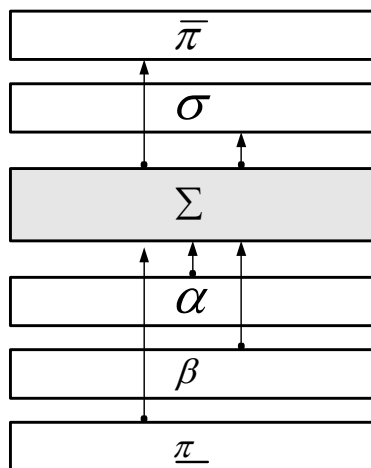


Рис. 5.2.2. Строчный сумматор.

Сложение кодов в целом состоит в последовательном сложении строк снизу вверх. Полная длительность сложения на строчном сумматоре

$$t_2 = N(N\tau + t' + t'')$$

### 5.2.3. Столбцовый делитель.

Этот делитель изображен на рис. 5.2.3. Он делит столбец  $\alpha$  исходного кодов. При этом выполняется распространение переносов  $\pi'$  через одноразрядные схемы. При делении образуется столбец результата  $\sigma$  и столбец выходного переноса  $\bar{\pi}$  в следующий столбец. На входе делителя присутствует столбец входного переноса  $\underline{\pi}$  из предыдущего столбца. Очевидно, разряд входного переноса используется как входной перенос  $\pi''$  для одной из одноразрядных схем.

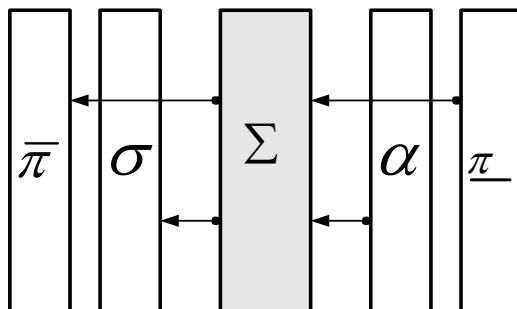


Рис. 5.2.3. Столбцовый делитель.

Деление кода в целом состоит в последовательном делении столбцов справа налево. Полная длительность деления на столбцовом делителе

$$t_3 = N(N\tau + t' + t'')$$

### 5.2.4. Строчный делитель.

Этот делитель изображен на рис. 5.2.4. Он делит строку  $\alpha$  исходного кодов. При делении образуется строка результата  $\sigma$  и строка выходного переноса  $\bar{\pi}$  в следующую строку. На входе сумматора присутствует строка входного переноса  $\underline{\pi}$  из предыдущей строки. Очевидно, разряд входного переноса используется как входной перенос  $\pi'$  для одной из одноразрядных схем и как входной перенос  $\pi''$  для другой из одноразрядных схем.



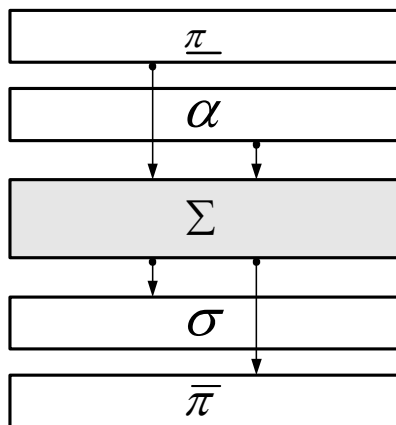


Рис. 5.2.4. Строчный делитель.

Деление кода в целом состоит в последовательном делении строк сверху вниз. Полная длительность деления на столбцовом делителе

$$t_4 = N(\tau + t' + t'')$$

### 5.2.5. Параллельный сумматор.

Этот сумматор изображен на рис. 5.2.5, где показаны связи одноразрядных сумматоров цепями переносов. Здесь указаны также цепи переносов из прямоугольной составляющей в смешанном коде. Полная длительность сложения на этом сумматоре

$$t_5 = N\tau + t''$$

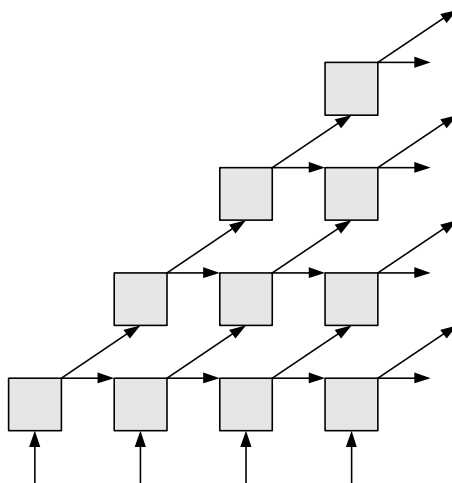


Рис. 5.2.5. Параллельный сумматор.

**5.2.6. Параллельный делитель.**

Этот делитель изображен на рис. 5.2.6, где показаны связи одноразрядных схем цепями переносов. Здесь указаны также цепи переносов в прямоугольную составляющую смешанного кода. Полная длительность деления на этом делителе

$$t_6 = N\tau + t''$$

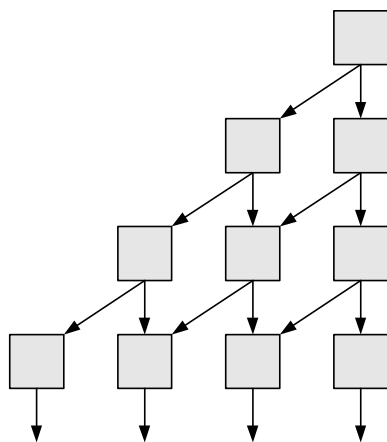


Рис. 5.2.6. Параллельный делитель.

## 5.3. Вариант арифметического устройства

### 5.3.1. Структура арифметического устройства.

Прежде всего, рассмотрим место арифметического устройства для операций с функциями в центральном процессоре – см. рис.

5.3.1, где

CPU - центральный процессор,

TAU - традиционное арифметическое устройство,

KDM - кодер-декодер M-кодов,

FAU - арифметическое устройство для операций с функциями

Как будет ясно из дальнейшего, все эти устройства должны участвовать в операциях кодирования и декодирования функций.

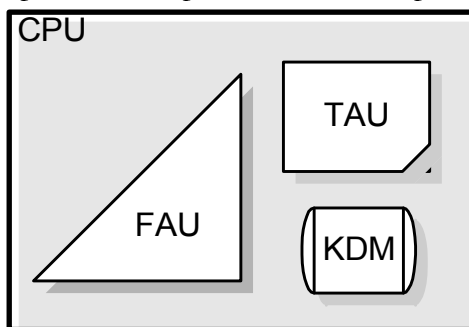


Рис. 5.3.1. Центральный процессор

Ниже рассматривается один из возможных вариантов построения FAU - арифметического устройства, оперирующего с четверичными тригонометрическими кодами функций. Блок-схема этого устройства приведена на рис. 5.3.2, где

RegТТК – регистр кода ТТК,

RegТК – регистр кода ТК,

RegЛК – регистр кода ЛК,

RegSK – регистр кода SK, который объединяет регистры RegТК и RegЛК,

RegPK – регистр кода PK,

ROM – таблицы кодов,

ADDSK – сумматор кодов SK,

DIVIS - делитель на 4 кода ТК,

QUADR – умножитель на 4 кода ТК,

ANL – анализатор разрядов множителя,

SHIFT – сдвигатель.

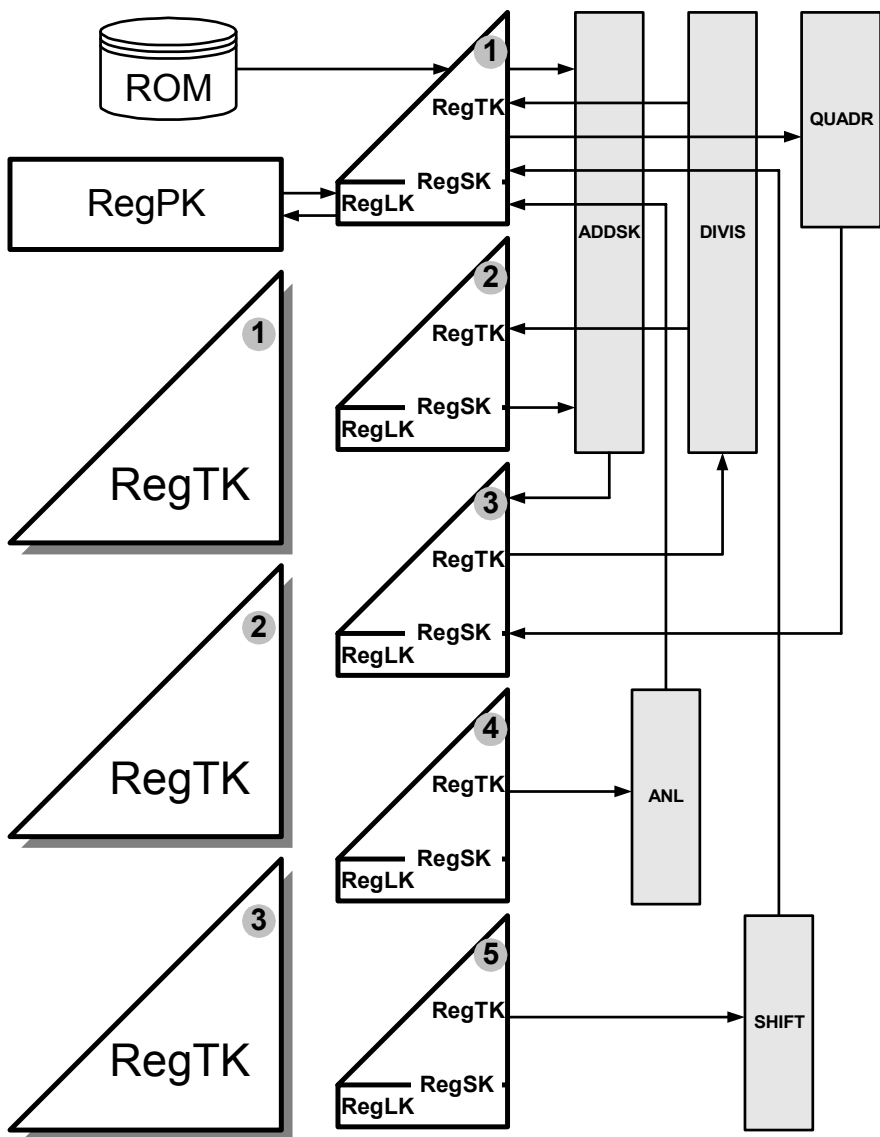


Рис. 5.3.2. FAU

Рассмотрим последовательность выполнения различных операций на этом FAU.

### 5.3.2. Операции с треугольными кодами.

#### 1. Алгебраическое сложение смешанных кодов.

Оно выполняется в устройстве ADDSK, причем исходные коды находятся в регистрах RegSK1 и RegSK2, а результат формируется в регистре RegSK3.

#### 2. Учетверение смешанного кода.

Оно выполняется в устройстве QUADR, причем исходный код находится в регистре RegSK1, а результат формируется в регистре RegTK3.

#### 3. Деление треугольного кода на 4.

Оно выполняется в устройстве DIVIS, причем исходный код находится в регистре RegTK3, а результат – смешанный код формируется в регистрах RegSK1 и RegSK2 (напомним, что результат деления формируется в двух регистрах). После деления выполняется сложение кодов в регистрах RegSK1 и RegSK2 на устройстве ADDSK. Результат формируется в регистре RegSK3, т.е. там же, где находился исходный код.

#### 4. Округление смешанного кода.

Оно выполняется согласно алгоритму 4.4.1. В устройстве RAM-ТК хранятся коды  $aB_v^i$ ,  $a \in \{1, -1, -2\}$ . Алгоритм реализуется следующим образом:

1. Округляемый код  $SK_4 = SK_4(f^i(x))$  передается в регистр RegSK3.
2. Округляемый код  $SK_4$ , находящийся в регистре RegSK3, передается в регистр RegSK2.
3. В младшей (-1)-строке регистра RegSK2 отыскивается старший значащий разряд  $\pi_{ov}$  округляемого кода. Если такой разряд отсутствует, то округление заканчивается с результатом в регистре RegTK3.

4. В регистр RegSK1 передается код  $\pi_{ov}B_v^i$ , а в регистре RegSK2 стирается разряд  $\pi_{ov}$ , т.е. в этом регистре образуется код  $SK_4 \leftarrow SK_4 - \pi_{ov}W_{-1,v}$
5. На устройстве ADDSK выполняется сложение:  $RegSK3 = RegSK1 + RegSK2$  или  $SK_4 \leftarrow SK_4 + \pi_{ov}B_v^i$ . Два последних пункта реализуют формулу (4.4.2).
6. Выполняется переход к п. 2.

#### 5. Укорочение треугольного кода.

Оно выполняется следующим образом:

1. Укорачиваемый код передается в регистр RegTK3.
2. Выполняется деление на 4 кода, находящегося в регистре RegTK3. Смешанный код результата оказывается в регистре RegSK3.
3. Если в младшей (-1)-строке регистра RegSK3 все разряды равны нулю, то укорочение заканчивается с результатом в регистре RegTK3. В противном случае выполняется округление кода, находящегося в регистре RegSK3. Результат образуется в регистре RegTK3.
4. Выполняется переход к п. 2.

#### 7. Преобразование прямоугольного кода в треугольный код.

Оно выполняется следующим образом:

1. Прямоугольный код  $PK(F(x))$  передается в регистр RegPK, а регистр RegSK1 обнуляется.
2. В регистр RegLK1 (являющийся частью регистра RegSK1) записывается очередная  $i$ -строка прямоугольного кода – линейный код  $LK(f_i(x))$ . Если перебор строк закончился, то преобразование заканчивается с результатом в регистре RegTK1.
3. Выполняется учетверение кода, находящегося в регистре RegSK1. Результат формируется в регистре RegTK3.
4. Выполняется перепись из регистра RegTK3 в регистр RegTK1.
5. Выполняется переход к п. 2.

Заметим, что в частности прямоугольный код может иметь только один столбец, т.е. представлять *число*. В этом случае данная операция соответствует кодированию числа в треугольный код.

### **8. Преобразование треугольного кода в прямоугольный код.**

Оно выполняется следующим образом:

1. Преобразуемый треугольный код  $TK(F(x))$  передается в регистр  $RegTK3$ .
2. Выполняется деление кода, находящегося в регистре  $RegTK3$ . Результат формируется в регистре  $RegSK1$ .
3. В очередную  $i$ -строку прямоугольного кода, находящегося в регистре  $RegPK$ , записывается линейный код  $LK(f_i(x))$ , образовавшийся в регистре  $RegLK1$  (являющимся частью регистра  $RegSK1$ ) Если перебор все строки записались, то преобразование заканчивается с результатом в регистре  $RegPK$ .
4. Выполняется перепись из регистра  $RegTK1$  в регистр  $RegTK3$ .
5. Выполняется переход к п. 2.

### **9. Умножение треугольных кодов.**

Оно состоит (как указывалось) из чередующихся операций «сдвиг-сложение». При этом:

1. Множитель находится в регистре  $RegTK4$ . Очередной разряд  $a$  множителя анализируется в устройстве ANL. При этом определяется вид алгебраического сложения частичного произведения и множимого: множимое может складываться ( $a=1$ ), вычитаться ( $a=-1$ ) или удваиваться и вычитаться ( $a=-2$ ).
2. Множимое находится в регистре  $RegTK5$  и сдвигается на устройстве SHIFТ в регистр  $RegTK1$ .
3. Частичное произведение перед очередным сложением находится в регистре  $RegTK2$ .
4. Новое частичное произведение формируется в регистре  $RegTK3$ , а затем передается в регистр  $RegTK2$ .

### 5.3.3. Операции с тригонометрическими треугольными кодами.

В данном АУ принят последовательный способ операций с ТТК – отдельные составляющие кодов ТТК, находящиеся в регистрах  $\text{RegТТК1}$  и  $\text{RegТТК2}$ , последовательно передаются в регистры  $\text{RegТТК1}$  и  $\text{RegТТК2}$ . Результат, образующийся в регистре  $\text{RegТТК3}$ , переписывается в регистр  $\text{RegТТК3}$ . Таким образом, АУ оперирует с ТК, разрядность которых в 4 раза меньше разрядности ТТК. Таким образом, во всех операциях с ТТК используется по определению 2.2.1

- компоновка кода  $\text{ТТК}_R(\Phi(x))$  из кодов  $\text{ТК}_R(f^i(x))$ ,
- декомпоновка кода  $\text{ТТК}_R(\Phi(x))$  в коды  $\text{ТК}_R(f^i(x))$ .

#### 1. Алгебраическое сложение ТТК.

Алгебраическое сложение ТТК распадается на 4 алгебраических сложения ТК.

#### 2. Умножение ТТК.

Каждая составляющая результирующего ТТК образуется в результате четырех умножений. Поэтому умножение ТТК содержит 16 умножений ТК и 12 сложений ТК.

#### 3. Дифференцирование ТТК.

В данном варианте АУ реализуется первый способ дифференцирования – сложение заранее вычисленных кодов

$$\text{ТК}^{mk} = \text{ТК} \left( \frac{d}{dx} \psi'_{mk} \right),$$

которые хранятся в устройстве RAM-ТК.

При этом каждая составляющая ТТК дифференцируется независимо. Рассмотрим дифференцирование треугольного кода очередной составляющей ТТК:

1. Дифференцируемый код находится в регистре  $\text{RegТТК4}$ . Очередной разряд  $a$  этого кода анализируется в устройстве ANL. При этом определяется вид алгебраического сложения частичного результата и табличного кода  $\text{ТК}^{mk}$ : этот код может складываться ( $a=1$ ), вычитаться ( $a=-1$ ) или удваиваться и вычитаться ( $a=-2$ ).



2. Очередной табличный код  $TK^{mk}$  переписывается из устройства в регистр RegTK1.
3. Частичный результат перед очередным сложением находится в регистре RegTK2.
4. Новое частичный результат формируется в регистре RegTK3, а затем передается в регистр RegTK2.

#### 4. Интегрирование ТТК.

В данном варианте АУ реализуется первый способ интегрирования с использованием сложения заранее вычисленных кодов  $TK_R(J_{mk}(x))$ , которые хранятся в устройстве RAM-ТК. При этом интегрирование выполняется аналогично дифференцированию.

#### 5. Кодирование тригонометрического ряда.

В данном варианте АУ реализуется алгоритм 2.4.1 кодирования ТТК. При этом используются заранее вычисленные коды  $TK\gamma_\omega^i$ , которые хранятся в устройстве RAM-ТК. При этом каждая  $i$ -составляющая тригонометрического ряда кодируется независимо. Итак, компоненты тригонометрического ряда представлены в виде (2.1.10) и известны числа  $E_\omega^i$ . Кодирование  $i$ -составляющей заключается в следующем:

1. Вычисление чисел  $E_\omega^i$  при известных коэффициентах  $D_\omega^i$  тригонометрического ряда. Это вычисление выполняется по формуле (2.1.19) на ТАУ.
2. Представление чисел  $E_\omega^i$  в виде М-кодов или, что одно и то же, в виде четверичных кодов. Это вычисление выполняется на KDM.
3. Кодирование четверичных кодов чисел  $E_\omega^i$ , то есть образование кодов  $TK E_\omega^i$ . Это кодирование описано выше.
4. Вычисление кода  $i$ -составляющей  $TK(\eta^i f^i(x))$  по известным кодам  $TK E_\omega^i$  и  $TK\gamma_\omega^i$  по формуле (2.4.2), т.е. суммирование произведений  $(TK E_\omega^i) \cdot (TK\gamma_\omega^i)$ .

### 6. Декодирование ТТК.

Декодирование выполняется по В. В данном варианте АУ реализуется алгоритму 2.4.3. Декодирование каждой  $i$ -составляющей ТТК выполняется независимо и заключается в следующем.

1. Треугольный код  $i$ -составляющей преобразуется в прямоугольный код, представляющий собой множество четверичных чисел  $A_v^i$ . Эта операция описана выше.
2. Четверичные числа  $A_v^i$  преобразуются в обычные двоичные коды. Это вычисление выполняется на **KDM**.
3. По (2.1.17) вычисляются коэффициенты  $D_\omega^i$  тригонометрического ряда функций  $f^i(x)$  в зависимости от значений чисел  $A_v^i$ . При этом используются числа  $L^i(v, n)$ . Это вычисление выполняется на **TAU**.

### 7. Укорочение ТТК.

После каждой операции может возникнуть переполнение разрядной сетки – увеличение количества столбцов свыше установленного числа  $N$ . Укорочение ТТК заключается в следующем:

1. Определение числа  $M$  столбцов ТТК.
2. Укорочение каждой составляющей ТК на  $(N-M)$  столбцов.
3. Умножение экспоненты ТТК на  $R^{(N-M)}$ . Это вычисление выполняется на **TAU**.

## 5.4. Сравнительный анализ

В этом разделе будет выполнен сравнительный анализ различных компьютеров, предназначенных для операций с тригонометрическими рядами вида (2.1.8). Такой ряд имеет коэффициенты  $D_{\omega}^i$  и представляет некоторую функцию  $\Phi(x)$ . Мы рассмотрим три компьютера:

- компьютер Т – традиционный компьютер, оперирующий (в данном случае) коэффициентами  $D_{\omega}^i$  тригонометрических рядов,
- компьютер Р – компьютер, содержащий несколько арифметических устройств, оперирующих с коэффициентами  $D_{\omega}^i$  параллельно,
- компьютер F – компьютер, содержащий FAU и оперирующий с кодами  $\text{ТТК}_4(\Phi(x))$ .

Для удобства дальнейшего изложения условимся называть ряд (2.1.8) тригонометрическим рядом  $\Omega$ -ранга. Очевидно, для решения любой практической задачи с рядами, требующей, по крайней мере, умножения рядов, необходимо ограничиться максимально допустимым рангом ряда. При отсутствии этого ограничения многократное умножение неизбежно выведет объем информации о ряде за пределы границ, обусловленных конструкцией компьютера или программы. Таким образом, в любом случае должна быть предусмотрена операция укорочения ряда, заключающаяся в отбрасывании старших его членов с одновременной коррекцией младших членов ряда. В связи с этим в качестве пробной для сравнения компьютеров рассмотрим задачу, содержащую лишь операции сложения, умножения и укорочения тригонометрических рядов.

### 5.4.1. Взаимосвязь между разрядностью ТТК, рангом ряда и разрядностью коэффициентов ряда.

Если  $n$  – максимальный номер старшего столбца треугольного кода четырех составляющих ТТК некоторой функции, то ранг тригонометрического ряда этой функции

$$\Omega = 4n. \quad (7.1)$$

Из требования об ограничении ранга рядов следует, что амплитуды  $|D_\omega^i|$  гармоник должны, как правило, убывать с увеличением их частоты  $\omega$ . Именно такой характер имеет зависимость максимально возможного модуля  $M_\omega = |D_\omega^i|$  от частоты  $\omega$ , построенная в разделе 4.5 – см. рис. 4.5.1. Поэтому в дальнейшем будем полагать, что частота  $\omega'$ , при которой для кодируемого ряда  $|D_{\omega'}^i| = \max_\omega |D_\omega^i|$ , и частота  $\omega''$ , при которой  $M_{\omega''} = \max_\omega M_\omega$ , совпадают, т.е.

$$\max_\omega |D_\omega^i| = \max_\omega M_\omega.$$

Отсюда следует, что разрядность  $b_D$  максимального коэффициента тригонометрического ряда и разрядность  $b_M$  максимального из чисел  $M_\omega$  совпадают.

Найдем величину  $b_M$ . Диапазон изменения коэффициента  $D_\omega$  в разложении функции, представленной треугольным кодом, имеющим  $(n+1)$  столбцов, равен

$$\xi(\omega, n) = |D_\omega > 0|_{\max} + |D_\omega < 0|_{\max} + 1.$$

Модули чисел  $D_\omega$  (присутствующие в этой формуле) и их свойства описаны в разделе 4.5. На основании этого можно рассчитать числа  $\xi(\omega, n)$ . В табл. 5.4.1 приведены эти числа  $\xi(\omega, n)$  при  $\omega \leq 5$ ,  $n \leq 5$ .

Таблица 5.4.1. Числа  $\xi(\omega, n)$ .

$n \setminus \omega$	0	1	2	3	4	5
0	4					
1	16	13				
2	58	61	19			
3	202	253	115	25		
4	700	973	523	169	31	
5	2450	3685	2155	901	247	37

Далее, в табл. 5.4.2 приведены для каждого  $n$  числа  $\xi_{\max}(n) = \max_\omega (\xi(\omega, n))$  и разрядность  $b_M$  двоичного кода этих чисел.

Таблица 5.4.2. Числа  $\xi_{\max}(n)$ .

$n$	0	1	2	3	4	5
$\xi_{\max}(n)$	4	16	61	253	973	3685
$b_M$	2	4	6	8	10	12

Из этой таблицы следует, что

$$b_M = 2(n+1) \quad (7.2)$$

#### 5.4.2. Разрядность.

Пусть допустимая относительная погрешность такова, что максимальный коэффициент ряда  $\Omega$ -ранга должен представляться  $b_D$ -разрядным двоичным кодом. Количество коэффициентов  $D_\omega^i$  ряда равно  $\Omega$ . Из (7.1) и (7.2) имеем:

- разрядность компьютера Т равна  $b_T = b_D$ ;
- разрядность массива кодов, представляющих функцию  $\Phi(x)$  в компьютере Р, равна  $b_P = 4nb_D$ ;
- разрядность компьютера F, т.е. разрядность ТТК (или разрядность 4-х ТК), равна  $b_F = 2(n+1)(n+2)$ .

Итак,

$$\begin{aligned} b_T &= b_D = 2(n+1), \\ b_P &= 4nb_D = 8n(n+1), \\ b_F &= 2(n+1)(n+2). \end{aligned}$$

#### 5.4.3. Объем арифметического устройства.

Этот объем определяется, в основном, разрядностью регистров и типом сумматора (при данном наборе команд). Как показано в разделе 5.1.1, объем одноразрядного сумматора ТК превышает объем обычного одноразрядного сумматора в 4 раза. Но FAU содержит сумматор ТК, разрядность которых в 4 раза меньше разрядности ТТК. Таким образом, относительный (отнесенный к разрядности) объем сумматора FAU совпадает с относительным объемом сумматора в компьютере Т. Следовательно, объемы арифметических устройств сравниваемых компьютеров относятся как разрядности этих компьютеров - см. табл. 5.4.3.

Таблица 5.4.3. Относительный объем

Компьютер	Относительный объем
T	(2)
P	(4n)
F	(n + 2)

#### 5.4.4. Длительность элементарных операций.

Длительность алгебраического сложения пропорциональна длине цепочки распространения переносов. В сумматоре компьютера T эта длительность равна  $t_T^S = \tau \cdot b_D = 2\tau(n + 1)$ , где  $\tau$  - время задержки в одном разряде. В сумматоре компьютера P эта длительность такая же и, поскольку все сумматоры работают параллельно, длительность алгебраического сложения в компьютере P равна  $t_P^S = t_T^S$ .

Длина цепочки распространения переносов в ТК равна сумме длин «катетов ТК», увеличенной вдвое, т.к. переполнение может вдвое увеличить количество столбцов суммарного кода. Поэтому длительность алгебраического сложения в компьютере F равна  $t_F^S = 4\tau(n + 1)$ .

Длительность умножения пропорциональна количеству разрядов множителя. Поэтому в компьютере T длительность умножения равна  $t_T^m = t_T^S b_D = 4\tau(n + 1)^2$ . В компьютере P длительность умножения всех коэффициентов на один из них такая же, т.е. равна  $t_T^m = t_P^S$ . Длительность умножения в компьютере F равна  $t_F^m = t_F^S b_F = 8\tau(n + 1)^2(n + 2)$ .

Длительность деления в компьютере T будем считать равным длительности умножения, т.е.  $t_T^d = t_T^m$ . Длительность деления в компьютере P всех коэффициентов на один из них такая же, т.е. равна, т.е.  $t_P^d = t_P^m$ .

Длительность деления на 4 в компьютере F равна  $t_F^d = 2t_F^S = 8\tau(n + 1)$ , что следует из алгоритма этой операции.

Длительность округления в компьютере F равна  $t_F^o = 2nt_F^S = 8\tau \cdot n(n + 1)$ , что следует из алгоритма этой операции.

Длительность укорочения в компьютере F равна  $t_F^u = 2n(t_F^d + t_F^o) = 16\tau \cdot n(n+1)^2$ , что также следует из алгоритма этой операции.

#### 5.4.5. Взаимосвязь между элементарными операциями и операциями с функциями.

В компьютере F операции с функциями соответствуют элементарным операциям. В компьютере T операции с функциями выполняются по некоторым очевидным программам. В компьютере P используются групповые операции, а операции с функциями также выполняются по некоторым очевидным программам, включающим и групповые операции. Кроме того, эти программы содержат некоторое количество операций управления и доступа к данным, которыми мы в дальнейших расчетах пренебрегаем (хотя это и ухудшает расчетные характеристики компьютера F).

В табл. 5.4.4 и 5.4.5 приведено количество и длительность операций в сравниваемых компьютерах.

Таблица 5.4.4. Длительность умножения

Комп	Операции	Длительность	
T	$16n^2$ умножений $16n^2$ сложений	$16n^2(t_T^s + t_T^m)$	$32\tau \cdot n^2(n+1)(2n+3)$
P	$4n$ умножений $4n$ сложений	$4n(t_P^s + t_P^m)$	$8\tau \cdot n(n+1)(2n+3)$
F	1 умножение 1 укорочение	$(t_F^m + t_F^u)$	$8\tau \cdot (n+1)^2(3n+2)$

Таблица 5.4.5. Длительность сложения

Комп.	Операции	Длительность	
T	$4n$ сложений	$4nt_T^s$	$8\tau \cdot n(n+1)$
P	1 сложение	$t_P^s$	$2\tau \cdot (n+1)$
F	1 сложение	$(t_F^s)$	$4\tau \cdot (n+1)$

#### 5.4.6. Выводы.

Результаты сравнения компьютеров сведены в табл. 5.4.6, где приведены их сравнительные характеристики. Они показывают, что

компьютеры Р и F сравнимы по быстродействию, но компьютер Р в 4 раза больше компьютера F по объему. Следовательно, дальнейшему сравнению подлежат только компьютеры F и T.

Учитывая (7.1), замечаем, что  $B_F/B_T \approx n/2 = \frac{\Omega}{8}$  и  $U_T/U_F \approx 4n = \Omega$ .

Таблица 5.4.6. Сравнение компьютеров.

Комп	Относительный объем (В)	Длительность умножения	Относительная длительность умножения (U)
T	(2)	$32\tau \cdot n^2(n+1)(2n+3)$	$\approx (4n)$
P	(4n)	$8\tau \cdot n(n+1)(2n+3)$	$\approx 1$
F	(n+2)	$8\tau \cdot (n+1)^2(3n+2)$	$\approx 1$

Таким образом,

у компьютера F по сравнению с компьютером T быстродействие увеличивается в  $\Omega$  раз за счет увеличения объема в  $\frac{\Omega}{8}$  раз.

Следовательно, быстродействие компьютера F увеличивается в 8 раз быстрее увеличения объема.



# Глава 6. Устройства для операций с М-кодами

В главе 5 было показано, что арифметические операции с четверичными кодами эквивалентны операциям с М-кодами, а в состав процессора, оперирующего с функциями, должен быть включен КДМ - кодер-декодер М-кодов. Поэтому в данной главе рассматриваются устройства алгебраического сложения и кодирования\декодирования М-кодов.

## 6.1. Алгебраическое сложение М-КОДОВ

### 6.1.1. Многоразрядные схемы для М-кодов

Рассматриваемые далее устройства для алгебраического сложения, кодирования и декодирования представляют собой линейные многоразрядные схемы. Они состоят из последовательно соединенных одnorазрядных схем – см. рис. 6.1.1, где

$N$  - разрядность кодов,

$k = \{0, 1, 2, \dots, n-2, n-1\}$  - номера разрядов и одnorазрядных схем,

$cop$  - код операции, общий для всех одnorазрядных схем,

$V1, V2$  – разряды входного переноса, изображающие число  $V$ ,

$W1, W2$  - разряды выходного переноса, изображающие число  $W$ ,

$A, B$  – операнды,

$C$  - результат.

В частных случаях линейных схем код операции и\или второй операнд могут отсутствовать. Входной перенос  $V$ , как правило, равен нулю. Выходной перенос  $W$ , отличный от нуля, свидетельствует о переполнении.

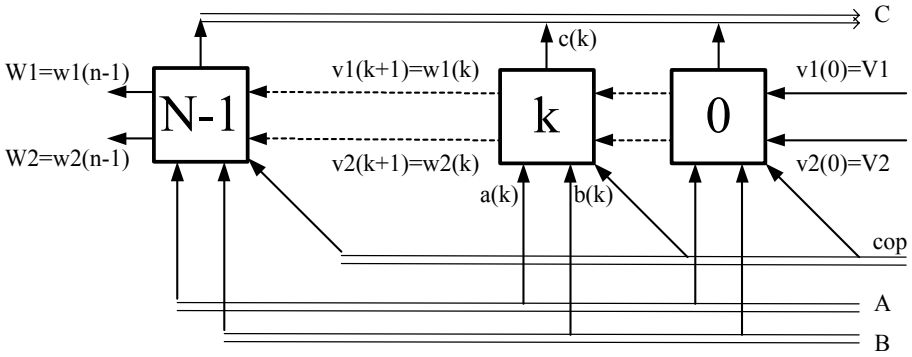


Рис. 6.1.1. Многоразрядная схема алгебраического сложения

Ниже при рассмотрении конкретных схем алгебраического сложения описываются (как правило) только одnorазрядные схемы.

### 6.1.2. Инвертор М-кода

На рис. 6.1.2 представлена одnorазрядная схема инвертирования *Inv*. Ее функционирование описывается таблицей истинности табл. 6.1.2. Эта таблица вычисляет величину  $c-2*w=-a+v$ .

Таблица 6.1.2. Одnorазрядная схема инвертирования

<b>a</b>	<b>v</b>	<b>w</b>	<b>c</b>
0	0	0	0
1	0	1	1
0	1	0	1
1	1	0	0

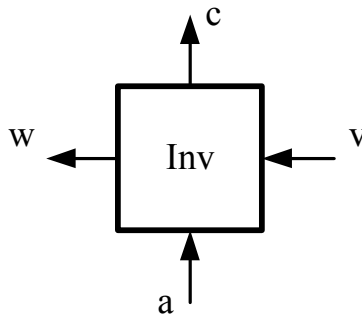


Рис. 6.1.2. Одnorазрядная схема инвертирования.

## 6.1.3. Инверсный сумматор М-кодов

На рис. 6.1.3 представлена одноканальная схема инверсного сумматора InvAdd. Ее функционирование описывается таблицей истинности табл. 6.1.3. Эта таблица вычисляет сумму  $c-2*w=(-a-b+v)$ .

Таблица 6.1.3. Одноканальная схема инверсного суммирования

<b>a</b>	<b>b</b>	<b>v</b>	<b>w</b>	<b>c</b>
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	0	1	0
0	0	1	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	1

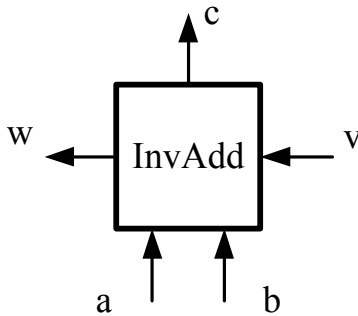


Рис. 6.1.3. Одноканальная схема инверсного сумматора.

## 6.1.4. Сумматор М-кодов

На рис. 6.1.4 представлена одноразрядная схема сумматора **Add**. Ее функционирование описывается таблицей истинности табл. 6.1.4. Эта таблица вычисляет сумму  $c-2*w = (a + b + v)$

Таблица 6.1.4. Одноразрядная схема суммирования

a	b	v1	v2	w1	w2	c
0	0	0	0	0	0	0
0	1	0	0	0	0	1
1	0	0	0	0	0	1
1	1	0	0	1	1	0
0	0	1	1	0	1	1
0	1	1	1	0	0	0
1	0	1	1	0	0	0
1	1	1	1	0	0	1
0	0	0	1	0	0	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	0	1	1	1	1

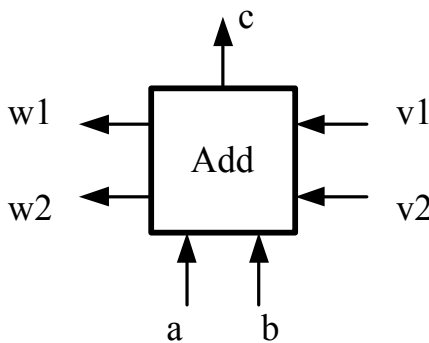


Рис. 6.1.4. Одноразрядная схема сумматора.

## 6.1.5. Вычитатель М-кодов

На рис. 6.1.5 представлена одноразрядная схема вычитателя **Sub**. Ее функционирование описывается таблицей истинности табл. 6.1.5. Эта таблица вычисляет сумму  $c-2*w = (a - b + v)$ .

Таблица 6.1.5. Одноразрядная схема вычитания

a	b	v1	v2	w1	w2	c
0	0	0	0	0	0	0
0	1	0	0	0	1	1
1	0	0	0	0	0	1
1	1	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	1	0	1	0
1	0	1	1	0	0	0
1	1	1	1	0	1	1
0	0	0	1	0	0	1
0	1	0	1	0	0	0
1	0	0	1	1	1	0
1	1	0	1	0	0	1

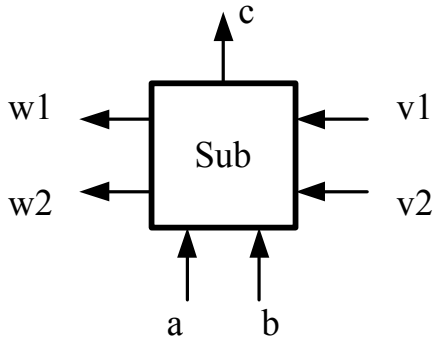


Рис. 6.1.5. Одноразрядная схема вычитателя.

## 6.1.6. Знакоопределитель М-кодов

Знакоопределитель определяет знак числа, представленного М-кодом. В нем используются одноразрядные знакоопределители, представленные на рис. 6.1.6.1 и имеющие две модификации:

*Seven* – одноразрядная схема знакоопределителя для разряда с четным номером,

*Sodd* - одноразрядная схема знакоопределителя для разряда с нечетным номером,

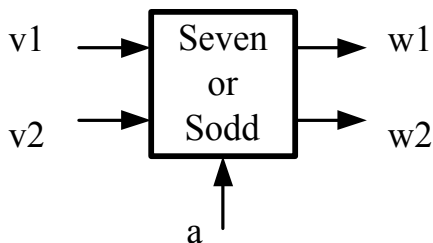


Рис. 6.1.6.1. Одноразрядная схема знакоопределителя.

Коды переносов в этих схемах интерпретируется следующим образом:

00 – код имеет нулевое значение,

01 – код имеет положительное значение,

10 – код имеет отрицательное значение.

Функционирование одноразрядных знакоопределителей *Seven* и *Sodd* описывается табл. 6.1.6.1 и 6.1.6.2 соответственно.

Таблица 6.1.6.1. Одноразрядная схема знакоопределителя для четного разряда

a	v2	v1	w2	w1
0	0	0	0	0
0	0	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	1	0	1

Табл. 6.1.6.1 реализует правило:

'w2, w1' = 'v2 v1', если a = '0',

'w2, w1' = '01', если a = '1'.

Табл. 6.1.6.2 реализует правило:

'w2, w1' = 'v2 v1', если a = '0',

'w2, w1' = '11', если a = '1'.

Таблица 6.1.6.2. Одноразрядная схема знакоопределителя для нечетного разряда

a	v2	v1	w2	w1
0	0	0	0	0
0	0	1	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	1	1	1

Знакоопределитель **nSign** в целом представлен на рис. 6.1.6.2, где показана схема соединения одноразрядных блоков *Seven* и *Sodd* между собой и с регистром М-кода. При этом приняты следующие обозначения:

$N$ -разрядность знакоопределителя,

$A$  – входной код,

$W1, W2$  - выходные переносы.

Код выходных переносов ( $W2, W1$ ) интерпретируется следующим образом:

00 – код имеет нулевое значение,

01 – код имеет положительное значение,

10 – код имеет отрицательное значение.

Таким образом, если  $W2=1$ , то  $A < 0$ , а если  $W2=0$ , то  $A \geq 0$ .

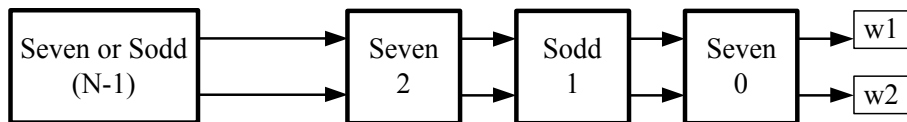


Рис. 6.1.6.2. Знакоопределитель

## 6.2. Устройства для кодирования и декодирования М-кодов

### 6.2.1. Кодер положительного Р-кода в М-код - CoderPM.

Этот кодер преобразует Р-код положительного числа в М-код этого числа. Его схема представлена на рис. 6.2.1.1, где

$N$ -разрядность кодера,

*Meven* – одноразрядная схема кодирования для разряда с четным номером,

*Modd* - одноразрядная схема кодирования для разряда с нечетным номером.

$A$  – входной Р-код.

$C$  – выходной М-код.

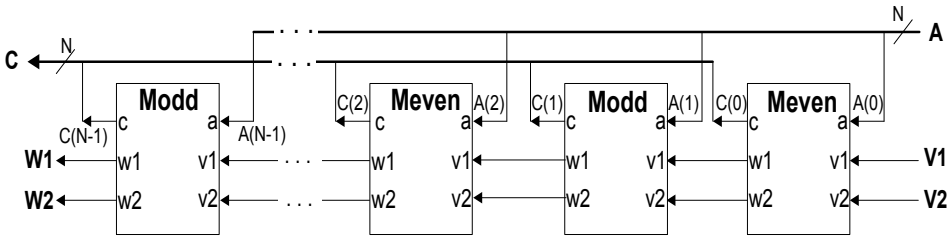


Рис. 6.2.1.1. Кодер положительного Р-кода в М-код

По-существу преобразование заключается в том, что из кода, составленного из четных разрядов Р-кода вычитается код, составленный из нечетных разрядов Р-кода, а вычитание выполняется по правилам вычитания М-кодов. Одноразрядные схемы *Meven* и *Modd* представлены на рис. 6.2.1.2. Их функционирование описывается таблицей истинности табл. 6.2.1.1 и 6.2.1.2 соответственно.



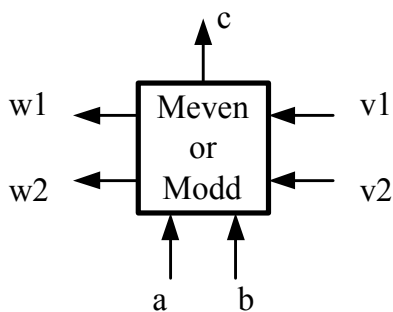


Рис. 6.2.1.2. Одноразрядная схема кодера.

Таблица 6.2.1.1. Одноразрядная схема кодера для четного разряда

<b>a</b>	<b>v1</b>	<b>v2</b>	<b>w1</b>	<b>w2</b>	<b>c</b>
0	0	0	0	0	0
1	0	0	0	0	1
0	0	1	0	0	1
1	0	1	1	1	0
0	1	1	0	1	1
1	1	1	0	0	0

Таблица 6.2.1.2. Одноразрядная схема кодера для нечетного разряда

<b>a</b>	<b>v1</b>	<b>v2</b>	<b>w1</b>	<b>w2</b>	<b>c</b>
0	0	0	0	0	0
1	0	0	0	1	1
0	0	1	0	0	1
1	0	1	0	0	0
0	1	1	0	1	1
1	1	1	0	1	0

### 6.2.2. Декодер M-кода в P-код – DecoderMP.

Этот декодер преобразует M-код некоторого числа в P-код этого числа. Его схема представлена на рис. 6.2.2.1, где

$N$  – разрядность декодера,

$Deven$  – одноразрядная схема декодирования для разряда с четным номером,

$Dodd$  – одноразрядная схема декодирования для разряда с нечетным номером,

$A$  – входной M-код,

$C$  – выходной P-код,

$COP$  – код операции,

$W$  – выходной перенос.

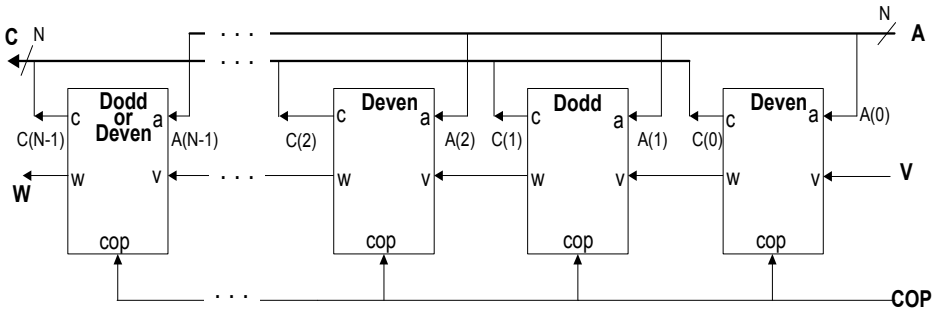


Рис. 6.2.2.1. Декодер M-кода в P-код

По-существу преобразование заключается в том, что

- если M-код представляет *положительное* число, то из кода, составленного из *четных* разрядов M-кода вычитается код, составленный из *нечетных* разрядов M-кода (в этом случае  $cop=0$ ),
- если M-код представляет *отрицательное* число, то из кода, составленного из *нечетных* разрядов M-кода вычитается код, составленный из *четных* разрядов M-кода (в этом случае  $cop=1$ ), а вычитание выполняется по правилам вычитания P-кодов.

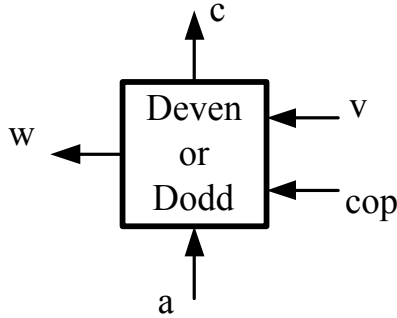


Рис. 6.2.2.2. Одноразрядная схема декодера.

Одноразрядные схемы *Deven* и *Dodd* представлены на рис. 6.2.2.2. Их функционирование описывается таблицей истинности табл. 6.2.2.1 и 6.2.2.2 соответственно.

Табл. 6.2.2.1 вычисляет

$$(-2w+c) = \begin{cases} (a - v), & \text{если } \text{cop} = 0 \\ (-a - v), & \text{если } \text{cop} = 1 \end{cases}$$

Табл. 6.2.2.2 вычисляет

$$(-2w+c) = \begin{cases} (-a - v), & \text{если } \text{cop} = 0 \\ (a - v), & \text{если } \text{cop} = 1 \end{cases}$$

Таблица 6.2.2.1. Одноразрядная схема декодера для четного разряда

	cop	a	v	w	c
<b>Even - Odd</b>	0	0	0	0	0
	0	1	0	0	1
	0	0	1	1	1
	0	1	1	0	0
<b>Odd - Even</b>	1	0	0	0	0
	1	1	0	1	1
	1	0	1	1	1
	1	1	1	1	0

Таблица 6.2.2.2. Одноразрядная схема декодера для нечетного разряда

	cop	a	v	w	c
Even - Odd	0	0	0	0	0
	0	1	0	1	1
	0	0	1	1	1
	0	1	1	1	0
Odd - Even	1	0	0	0	0
	1	1	0	0	1
	1	0	1	1	1
	1	1	1	0	0

### 6.2.3. Полный декодер М-кода в Р-код – mDecoderMP

Декодер *DecoderMP* управляется кодом операции *Cop*. В связи с этим полный декодер должен (кроме декодера *DecoderMP*) содержать еще и знакоопределитель *nSign*. При этом схема блока принимает вид, представленный на рис. 6.2.7.

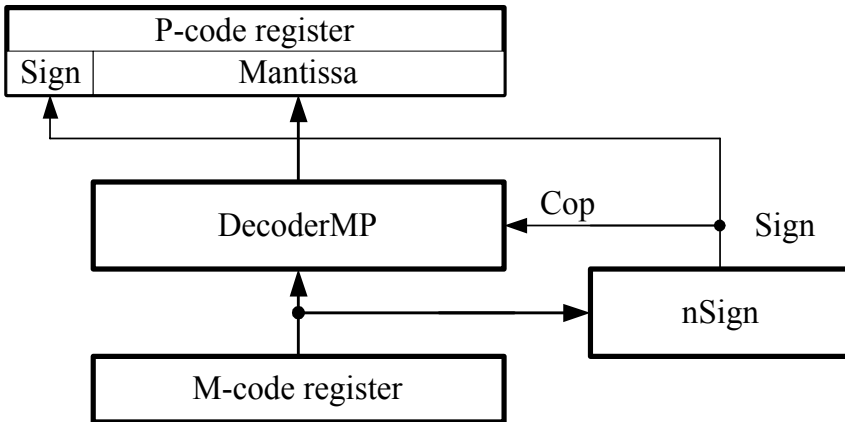


Рис. 6.2.7. Полный декодер.

---

# Литература

---

1. Малиновский Б.Н., История вычислительной техники в лицах. Киев, Фирма "КИТ", ПТОО "АСК", 1995 г.
2. Хмельник С.И., Кодирование функций. Кибернетика, АН УССР, 1966, №4.
3. Хмельник С.И., Несколько типов позиционных кодов функций. Кибернетика, АН УССР, 1970, №5.
4. Хмельник С.И., Алгоритмы кодирования и декодирования функциональных рядов. Сб. "Цифровая вычислительная техника и программирование", выпуск 8, 1974.
5. Хмельник С.И., Арифметическое устройство для цифровой вычислительной машины. Авт.св. 266362, 1970, БИ-11.
6. Хмельник С., Компьютерная арифметика векторов, фигур и функций, изд. «Mathematics in Computers», Москва – Тель-Авив, 1995.

# Обозначения

**Add** - сумматор М-кодов,

$A_v$  - коэффициент функционального ряда по функциям  $\psi_{ok}(x)$  – см. теорему 2.1.5,

$a_\omega^i$  - числа, определенные в теореме 2.1.3,

$c^i$  - числа, определенные в теореме 2.1.4,

**CoderPM** - кодер положительного Р-кода в М-код,

$D_\omega^i$  - коэффициент функционального ряда по функциям  $\mathcal{E}_\omega^i(x)$  – см. теорему 2.1.5,

**DecoderMP** - декодер Р-кода в М-код,

**Deven** – одноразрядная схема декодирования для разряда с четным номером,

**Dodd** - одноразрядная схема декодирования для разряда с нечетным номером,

$E_\omega^i$  - коэффициент функционального ряда по функциям  $\gamma_n^i(x)$  – см. теорему 2.1.5,

$[F \parallel h_k]$  - множество коэффициентов  $h_k$  функционального ряда для функции  $F$ ,

**FAU** - арифметическое устройство для операций с функциями,

$H_\omega^i$  - коэффициент функционального ряда по функциям  $\lambda_\omega^i(x)$  – см. теорему 2.1.5,

**Inv** - инвертор М-кода,

**InvAdd** - инверсный сумматор М-кодов,

$K_R(A)$  - код числа  $A$  по основанию  $R$ ,

$L^i(v, n)$  - числа, определенные в теореме 2.1.2,

**Meven** – одноразрядная схема кодирования для разряда с четным номером,

**Modd** - одноразрядная схема кодирования для разряда с нечетным номером,

**mDecoderMP** - полный декодер P-кода в M-код,

**nSign** - знакоопределитель M-кода,

$S^i(v, \omega)$  - числа, определенные в теореме 2.1.1,

**M-код** - код вещественного числа по отрицательному основанию «-2»,

**P-код** - прямой код положительного числа по основанию «2»,

$PK_R(F(x))$  - прямоугольный код функции  $F(x)$ ,

$PRK(F(x, v))$  - пирамидальный код функции  $F(x, v)$ ,

**Seven** – одноразрядная схема знакоопределителя для разряда с четным номером,

$SK_R(F(x))$  - смешанный код функции  $F(x)$ ,

**Sodd** - одноразрядная схема знакоопределителя для разряда с нечетным номером,

$STK_R(F(x))$  - ступенчатый код функции  $F(x, v)$ ,

**Sub** – вычитатель M-кодов,

$TK_R(F(x))$  - треугольный код функции  $F(x)$ ,

$TPRK_R(\Phi(x, v))$  - тригонометрический пирамидальный код функции  $\Phi(x, v)$ ,

$TTK_R(F(x))$  - тригонометрический треугольный код функции  $F(x)$ ,

$LK_R(f(x))$  - линейный код функции  $f(x)$ .

$\mathcal{E}_\omega^i(x)$  - функции, определенные в теореме 2.1.4,

$\mathcal{Y}_n^i(x)$  - функции, определенные в теореме 2.1.3,

$\eta^i(x)$  - функции, определенные в теореме 2.1.4,

$\lambda_\omega^i(x)$  - функции, определенные в теореме 2.1.4,

$\psi_{mk}$  - вес  $mk$ -разряда в треугольного кода по основанию  $y = \sin^2(x)$ ,

$\psi'_{mk}$  - вес  $mk$ -разряда в тригонометрического треугольного кода.