Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Ю.И. ГАГАРИН, К.Ю. ГАГАРИН

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Учебное пособие

РГГМУ

Санкт-Петербург
2012

ББК 32.811.3я7
УДК 621.3; 681.3

Гагарин Ю.И., Гагарин К.Ю. Цифровая обработка сигналов. Учебное пособие – СПб.: изд. РГГМУ, 2012, 104 с.
Y.I. Gagarin, K.Y. Gagarin. Digital signal processing. School-book – St.Peterburg: RSHU Publishers, 2012,- 104 pp.
ISBN 978-5-86813-331-2

Рецензент Фомин В.В. – д.т.н., профессор кафедры "Информационные системы и программное обеспечение" Российского государственного педагогического университета.

В учебном пособии рассмотрены различные способы формализованного представления цифровых сигналов в конечных и бесконечных числовых полях и в их расширениях. Сформированы основные направления по решению задач синтеза (кодирования) и анализа (декодирования) цифровых последовательностей. Большое внимание уделено изучению методов и компьютерных технологий статистических оценок на основе цифровой линейной фильтрации, корреляционного и спектрального анализа. Значительное место занимают методы и алгоритмы быстрой цифровой обработки одномерных и многомерных сигналов со способами организации типовых и специализированных архитектур вычислительных устройств.

Пособие предназначено для студентов старших курсов РГГМУ, обучающихся по направлению подготовки (специальности) 090302 «Информационная безопасность телекоммуникационных систем» и по направлению 180800 "Корабельное вооружение" по профилю "Морские информационные системы и оборудование", квалификация – бакалавр.

Considered different methods of formalized digital signals representation over finite and infinite fields and over their extensions. Formulated main ways of solving a problems of synthesis (coding) and analysis (decoding) of digital sequences. The major attention is focused on studying the methods and computer technologies for statistical evaluations based on digital linear filtering, correlation and spectral analysis. The considerable part covers methods and algorithms of fast digital processing of one-dimension and multi-dimension signals together with organization of signal processors and general purposes processors architectures.

This teaching aid is meant for students of senior courses studying the specialty 090302 "Information security of telecommunication systems" and specialty 180800 "Shipboard armament" at "Naval information systems and equipment" profile, for bachelors qualification.

ISBN 978-5-86813-331-2

ПРЕДИСЛОВИЕ

Современный рынок вычислительной техники и средств связи характеризуется высокой степенью миниатюризации, многофункциональностью и широкой доступностью для пользователей в их приобретении. При этом свойства многофункциональности чаще всего ориентированы на цифровую обработку сигналов (в том числе в реальном времени), которая требует использования высокопроизводительных вычислительных средств, создаваемых на основе быстрых вычислительных алгоритмов, реализуемых на векторно-конвейерных архитектурах сигнальных процессоров.

Аналогичные требования могут предъявляться к обработке гидрометеорологических сигналов и данных, полученных, например, средствами радиолокационного зондирования верхних слоёв атмосферы и ионосферы Земли.

В частности, одной из задач цифровой обработки гидрометеосигналов является сокращение избыточности - сжатие запоминаемых или передаваемых по каналам связи данных с возможностью их восстановления с высоким качеством в условиях обеспечения требуемых помехозащиты и защиты от несанкционированного доступа.

Учебное пособие включает в себя шесть разделов и Приложение.

В первом разделе в краткой форме приведены сведения об алгебраических структурах, которые получили наиболее широкое применение в цифровой обработке сигналов: группы, кольца, поля, векторные пространства и матричная алгебра.

Второй раздел посвящён математическим моделям цифровых линейных свёрток и дискретного преобразования Фурье применительно к полю комплексных чисел и полям Галуа.

В третьем разделе рассматриваются алгоритмы быстрого преобразования Фурье (БПФ) в поле комплексных чисел по основанию два , а также по основанию четыре и по смешанным основаниям.

Четвёртый раздел посвящён быстрым теоретико-числовым преобразованиям Мерсенна по основанию два в простых полях, полученным через переходные функции из их расширений.

В пятом разделе приведены матрично-векторные формы быстрых ортогональных преобразований в поле вещественных чисел. В частности, рассмотрены дискретное косинусное и вейвлет-преобразования, к которым привлечено наибольшее внимание специалистов в области сжатия цифровых сигналов.

Шестой раздел включает в себя корреляционные и спектральные методы классификации и сжатия цифровых речевых сигналов.

В Приложении представлены в формализованном виде арифметико-логические основы построения вычислительных устройств и общие принципы организации типовых однопроцессорных ЭВМ с примерами использования для цифровой обработки сигналов.Материалы пособия в большинстве своём содержат авторские научно-практические разработки в быстрой цифровой обработке сигналов.

Раздел 1. ВВЕДЕНИЕ В АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ

Под алгебраической структурой принято понимать, либо конечное, либо бесконечное множество *M*, отношения между элементами которого определены в виде алгебраических операций с аксиоматически заданными свойствами. В дальнейшем нас будут интересовать преимущественно числовые и построенные на их основе векторные и полиномиальные множества. Классификация структур обычно следует в порядке возрастания количества допустимых алгебраических операций. Структуры с допустимой одной операцией получили широкое применение в форме различных групп.

1.1. Группы

В общем, теоретико-множественном виде всякую группу можно представить

$$G=\langle M;\circ\rangle, \tag{1.1}$$

где $\circ$ - оператор допустимой одной операции над элементами множества *M*: + сложения в аддитивных группах, либо × умножения в мультипликативных группах. По своим свойствам группы делятся на коммутативные (абелевы) и некоммутативные, циклические и нециклические. В коммутативной группе соблюдается равенство

$$x \circ y = y \circ x;\ \ x, y \in M\ .$$

Свойство цикличности группы определяется наличием образующего элемента α_N, через который могут быть выражены все другие элементы группы

$$G_N = \{\dot{e}, \dot{\alpha}_N^1, \dot{\alpha}_N^2, .., \dot{\alpha}_N^{N-1}\},$$

где $\dot{e}$, $\dot{\alpha}_N^1$ - соответственно единичный и образующий элементы группы по заданной операции $\circ$, *N*- порядок группы. Здесь степенное обозначение $\dot{\alpha}_N^i$ соответствует *i*-кратному применению операции $\circ k$ *элементу* $\boldsymbol{\alpha}_N$. Во всякой группе имеются единичный $\dot{e}$ и обратный $\overline{x}$ элементы обуславливающие соответственно равенства

$$\dot{e} \circ x = \dot{e} \circ x = x\ ;$$

$$x \circ \overline{x} = \overline{x} \circ x = \dot{e}$$

для любого элемента $x \in M$.

Циклическими группами могут быть конечные множества натуральных чисел с групповой операцией сложения, либо умножения по модулю q. Например, для $q = 7$ имеем аддитивную группу $Z^{(a)}{}_7$ порядка $N = 7$ и мультипликативную группу $Z^{(m)}{}_6$ 6-го порядка

$$Z_7^{(a)} = \{0,1,2,3,4,5,6\} \quad , \quad Z_6^{(m)} = \{1,3,2,-1,-3,-2\} .$$

1.2. Кольца

В порядке увеличения до двух количества допустимых арифметических операций за группами следуют кольца, образованные также на бесконечных, либо на конечных исходных множествах, элементами которых могут быть числа-скаляры, либо полиномы.

По аналогии с выражением (1.1) кольцо можно представить как абстрактную структуру $K = \langle M ; + , \times \rangle$ с допустимыми операциями сложения и умножения. По операциям сложения всякое кольцо является группой, для которой выполняются условия замкнутости со свойствами коммутативности, ассоциативности, дистрибутивности и наличия единичного элемента. По умножению различают коммутативные кольца с единицей, которые называют просто кольцами. Обратные элементы по умножению могут быть правым или левым и существовать только в кольце с единицей. В кольце для любого элемента допустимо возведение в целочисленную степень. Большинство известных законов алгебры являются справедливыми для произвольного кольца. Это позволяет использовать при вычислениях в кольцах традиционные приёмы и действия над целыми числами и полиномами. Примерами колец могут служить Z - бесконечное кольцо целых чисел и Z_q - кольцо целых чисел по модулю q на конечном множестве.

1.3. Поля

Поле как абстрактную структуру можно рассматривать в виде кольца, в котором отсутствуют ограничения по наличию обратных элементов по умножению, т.е. обратный элемент по умножению имеется для каждого элемента исходного числового, либо полиномиального множеств.

Простыми бесконечными полями являются поле вещественных чисел R и поле рациональных чисел Q, поле комплексных чисел C – это 2-рас-

ширенное поле вещественных чисел, поле *Q(i)* – 2-расширенное поле рациональных чисел (поле комплексных рациональных чисел). Множество чисел-скаляров является простым полем, если оно замкнуто по заданным бинарным операциям (сложения и умножения). При этом выполняются условия ассоциативности и коммутативности по каждой операции и существуют единичные элементы по сложению (нуль) $a + 0 = a$ и умножению (единица) $a \times 1 = a$.

Кроме того, для каждого элемента a поля F существуют единственный обратный элемент ($-a$) по сложению $a - a = 0$ и обратный элемент по умножению $a^{-1} \times a = 1$. Элемент 0 является необратимым элементом по умножению.

Числовые конечные поля получили названия полей Галуа, обозначаемых *GF(q),* где *q*-целое число. При значениях *q*, равным простым числам *GF(q)* являются простыми, нерасширенными, не имеющими подполей. Простое поле *GF*(2) имеет минимальное исходное множество элементов {0,1} и может быть задано в виде таблиц сложения и умножения.

+	0	1	×	0	1
0	0	1	0	0	0
1	1	0	1	0	1

Примечательной особенностью полей Галуа является то, что в них присутствуют практически неограниченные возможности в выборе циклических мультипликативных групп, на основе которых строятся матрицы теоретико-числовых пребразований (преобразований Фурье в полях Галуа).

1.4. Векторные пространства

Понятие векторного пространства является фундаментальным для теории матриц и, в свою очередь, базируется на понятии поля, множества скаляров, на которые можно умножать векторы.

Под вектором для заданного поля $\Im$ принято понимать последовательность (V_0, V_1, ..., V_{n-1}) элементов поля-скаляров V_i, называемую вектором длины n.

Векторным пространством V над полем $\Im$ называется множество таких векторов, для которых определены операции векторного сложения и умножения вектора на скаляр.

Операция векторного сложения записывается в виде

$$V = V' + V'' = (V'_0, V'_1, \ldots, V'_{n-1}) + (V''_0, V''_1, \ldots, V''_{n-1}) =$$
$$= (V'_0 + V''_0), \ldots, (V'_{n-1} + V''_{n-1}).$$

Операция левого умножения вектора на скаляр осуществляется по правилу

$$C\,(V_0,\ V_1, \ldots,\ V_{n-1}) = (CV_0,\ CV_1, \ldots,\ CV_{n-1}).$$

Для указанных операций в векторном пространстве должны соблюдаться условия:

1) коммутативность, ассоциативность и наличие нулевого (нейтрального) элемента по сложению;

2) если $x,\ y \in V$ *и* $a,\ b \in \Im$, то

$a(x + y) = ax + ay$; $\qquad (a + b)\,x = ax + bx$;

$a(bx) = (ab)\,x$; $\qquad e\,x = 1x = x,\ \ e \in \Im$.

Множество векторов в векторном пространстве называется линейно зависимым, если соблюдается равенство

$$a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_{m-1} \cdot x_{m-1} = 0,$$

хотя бы при одном $a_j \neq 0$, где $a_i \in \Im$, $x_i \in V$. Для ортогональных преобразований применяют ортонормированные векторные пространства, связанные со скалярными произведениями и векторными нормами.

$$\|\cdot\| : V \to R,$$

В общем случае под векторной нормой пространства V понимается функция если для всех $x,\ y \in V$ выполняются условия [2]:

1) $\|x\| \geq 0$ – (неотрицательность);

2) $\|x\| = 0$ – только при $x = 0$ (положительность);

3) $\| cx \| = |c| \cdot \| x \|$ – где $c \in F$ (абсолютная однородность);

4) $\| x+y \| \leq \| x \| + \| y \|$ – (неравенство треугольника).

Наиболее широкое применение получили евклидовы нормы, в том числе в векторной трёхмерной пространственной алгебре. Поэтому многие термины («длина», «угол») часто используют для n-мерных векторных пространств.

В поле вещественных чисел скалярное произведение двух векторов определяется выражением

$$(x,\ y) = x_0 \cdot y_0 + x_1 \cdot y_1 + \ldots + x_{n-1} \cdot y_{n-1}.$$

Если $(x,\ y) = 0$, то говорят, что векторы x, y взаимно ортогональны. Для случая $(x,\ x)$ из выражения (1.1) можно получить векторную норму:

$$\|x\| = (x,x)^{1/2} = \sqrt{x_0^2 + x_1^2 + \ldots x_{n-1}^2}.$$

Для поля комплексных чисел скалярное произведение находят с учётом сопряженных векторов

$$(x, y) = \sum_{k=0}^{n-1} x_k \dot{y}_k$$

Вектор называется единичным или нормированным, если $(x, x) = 1$. Система векторов $e_0, e_1, \ldots, e_{rn-1}$ называется ортонормированной, если

$$(e_i, e_j) = \delta_{ij} = \begin{vmatrix} 1 \\ 0 \end{vmatrix}$$

при $i = j$, при $i \neq j$,

где $i, j = 0, 1, \ldots, m\text{-}1$.

В дальнейшем будут рассматриваться ортонормированные векторные пространства строк (или столбцов) матриц.

1.5. Матричная алгебра

Матрицей $A_{n,m}$ принято называть прямоугольную таблицу $n \times m$ элементов, состоящую из n строк и m столбцов

$$A_{n,m} = \|a_{ij}\| = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m-1} \end{pmatrix}.$$

При $n = m$ матрицу называют квадратной и обозначают A_n.

Две матрицы складывают поэлементно:

$$A + B = \begin{pmatrix} a_{0,0} + b_{0,0} & a_{0,1} + b_{0,1} & \cdots & a_{0,m-1} + b_{0,m-1} \\ a_{1,0} + b_{1,0} & a_{1,1} + b_{1,1} & \cdots & a_{1,m-1} + b_{1,m-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1} + b_{n-1,1} & a_{n-1,1} + b_{n-1,1} & \cdots & a_{n-1,m-1} + b_{n-1,m-1} \end{pmatrix}.$$

Любую матрицу A_n можно умножить на скаляр β по правилу:

$$\beta A = \begin{pmatrix} \beta a_{0,0} & \beta a_{0,1} & \cdots & \beta a_{0,m-1} \\ \beta a_{1,0} & \beta a_{1,1} & \cdots & \beta a_{1,m-1} \\ \cdots & \cdots & \cdots & \cdots \\ \beta a_{n-1,0} & \beta a_{n-1,1} & \cdots & \beta a_{n-1,m-1} \end{pmatrix}.$$

Всякую матрицу $A_{1,n}$ можно умножить на матрицу $B_{n,m}$, получив матрицу

$$C_{l,m} = \left\| C_{ij} \right\|, \text{ где}$$

$$C_{i\,j} = \sum_{k=0}^{n-1} a_{i\,k} b_{k\,j}, \quad i = \overline{0, l-1}, \quad j = \overline{0, m-1}.$$

Множество элементов $\{a_{ii}\}$ квадратной матрицы A_n, $i = \overline{0, n-1}$, получило название главной диагонали. Операция умножения матриц в общем случае некоммутативна.

Частным случаем умножения матриц является умножение вектора на матрицу. При умножении n-компонентного вектора строки X_n слева на матрицу $A_{n,m}$ в результате m-компонентный вектор-строку $X_n A_{n,m} = Y_m$.

При умножении матрицы $A_{n,m}$ на m-компонентный вектор-столбец справа имеем в результате n-компонентный вектор-столбец.

Транспонированной к матрице $A_{n,m}$ является матрица $A^T{}_{m,n}$, у которой $a^T{}_{i,j} = a_{i,j}$, т.е. строками транспонированной матрицы являются столбцы матрицы $A_{n,m}$.

Обратной к квадратной матрице A_n, если такая существует, является квадратная матрица $A^{-1}{}_n$, которая удовлетворяет равенству

$$\frac{1}{n} A_n A_n^{-1} = I_n, \qquad \text{где } I_n \text{ – единичная матрица}$$

$$I_n = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

Матрица, имеющая обратную матрицу, называется невырожденной.

Операции транспонирования и нахождения обратных матриц от произведений матриц подчиняются закону обращения порядка

$$(ABC)^T = C^T B^T A^T, \qquad (ABC)^{-1} = C^{-1} B^{-1} A^{-1}.$$

Квадратная матрица A_n называется ортогональной, если её строки (либо столбцы) образуют ортонормированную систему. Для любой ортогональной матрицы справедливо соотношение $A_n A^T{}_n = A^T{}_n A_n = I_n$, т.е. $A^T{}_n = A^{-1}{}_n$.

Ортогональные матрицы над полем C часто называют унитарными. Заметим, что в поле C для унитарной матрицы A_n обратная матрица равна транспонированной комплексно-сопряженной матрице $A^{-1}{}_n = A^T{}_n$.

Произведение двух ортогональных матриц и матрица, обратная для ортогональной, ортогональны.

Из примечательных ортогональных матриц следует отметить матрицы ортогональных преобразований (например, Фурье над полем C, Хартли над полем R и др.) и матрицы перестановки – матрицы, строки (столбцы) которых содержат один ненулевой, единичный, элемент, при этом отсутствуют одинаковые строки (столбцы). В качестве примера можно привести матрицу инверсной перестановки, которая равна

$$\bar{I}_n = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

По аналогии разбиения множества на подмножества любую матрицу можно представить в блочном, подматричном, виде. Причём каждый элемент исходной матрицы A_n входит только в одну из подматриц.

Например, можно записать матрицу A_n в виде четырех блок - матриц размерности $n/2 \times n/2$.

$$A_n = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{c|c} A_{n/2} & A_{n/2} \\ \hline A_{n/2} & -A_{n/2} \end{array}\right).$$

Умножение и сложение матриц с согласованным блочным разбиением напоминает обычное умножение и сложение матриц.

Из других примечательных матриц следует отметить циркулянтные матрицы. В общем виде матрицу-циркулянт (или просто циркулянт) можно записать

$$A_n = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_1 & a_2 & a_3 & \cdots & a_{n-1}a_0 \end{pmatrix},$$

т.е. циркулянтную матрицу задают посредством циклического сдвига на одну позицию элементов одного вектора $\boldsymbol{a}_n$.

Иногда различают лево- и правоциркулянтные матрицы в зависимости от того, осуществляется левый или правый циклический сдвиг.

Матрицы-циркулянты применяют для представления циклических сверток и корреляций, которые в последовательностной форме соответственно задают в виде

$$y_n = \sum_{k=0}^{N-1} x_k h_{n-k},$$

$$\rho_n = \frac{1}{n}\sum_{k=0}^{N-1} x_k h_{n+k},$$

$$k, n = \overline{0, N-1},$$

$\{x_k\}$ $\{h_k\}$ – исходные дискретные функции. Например, векторно-матричная форма задания сверток имеет вид

$$V_N = S_N X_N,$$

где $X_N = x_0, x_1, x_2, \ldots, x_{N-1}$, S_N – матрица-циркулянт с образующим вектором $\overline{S} = h_0, h_{N-1}, h_{N-2}, \ldots, h_1$.

С циркулянтами связаны диагональные матрицы. В общем виде диагональная матрица может быть записана

$$D_N = \begin{pmatrix} d_0 & 0 & 0 & \cdots & 0 \\ 0 & d_1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & d_{N-1} \end{pmatrix} = \mathrm{diag}\{ d_i \}_{i=0}^{N-1}$$

Две матрицы A и B называются ортогонально подобными, если имеется ортогональная (или унитарная) матрица T, удовлетворяющая следующему равенству

$$B = T^{-1} A\, T.$$

Отсюда следует, что матрица A тогда и только тогда ортогональна (унитарна), когда она ортогонально подобна диагональной матрице B.

В задачах синтеза алгоритмов быстрых ортогональных преобразований часто используются кронекеровское (прямое) произведение матриц

$$C = A \otimes^{\bullet} B = B \otimes A,$$

где $C = |c_{ij}|_1^{mn}$, т.е. матрица C составлена из всевозможных произведений элементов матриц A и B в соответствующем порядке. Здесь $^{\bullet}\otimes, \otimes$ знаки соответственно левого и правого кронекеровского произведения. Для кронекеровских произведений имеется взаимосвязь с обычными матричными умножениями: если $A = A_1 A_2$ и $B = B_1 B_2$, то $C = (A_1 \otimes B_1)(A_2 \otimes B_2)$.

Раздел 2. ЦИФРОВЫЕ СВЁРТКИ И ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

2.1. Способы формального представления цифровых сигналов

Для цифровых одномерных сигналов, представленных в исходной временной области, используется общая последовательностная модель, которую можно записать в виде

$$\{s(n)\}\ ,\ n = \overline{0, N-1},\ s(n) \in \Im, \tag{2.1}$$

где $s(n)$ – значения цифрового сигнала в момент времени nT_0, где T_0- шаг дискретизации по времени, т.е. здесь цифровой сигнал представлен в виде последовательности значений дискретной функции вещественной переменной. Значения самой функции принадлежат заданному простому, либо расширенному, бесконечному, либо конечному полю $\Im$. К простым бесконечным полям относятся поле Q-рациональных чисел и поле R-вещественных чисел. Конечные поля (поля Галуа $GF(p)$) имеют характеристику – простое число p, по модулю которого образовано простое поле.

Для расширенных полей, например, поля С комплексных чисел или m-расширенных полей Галуа $GF(p^m)$, элементами последовательности $\{s_n\}$ выступают полиномы, образованные по модулю неприводимого в основном поле полинома (для поля комплексных чисел таким полиномом является $f(z)=z^2+1$ –неприводимый в поле вещественных чисел).

От последовательностной (поэлементной) формы (2.1) задания цифрового сигнала существует переход к формальному степенному ряду [2]

$$S(z) = \sum_{n=0}^{N-1} s_n\ z^n, \tag{2.2}$$

либо к векторной форме $\overline{s}_n = s_0 s_1 \ldots s_{N-1}$.

Для кодирования и распознавания сигналов часто используются цифровые последовательности - дискретные функции, наделённые определёнными свойствами. Например, для линейной фильтрации к таким функциям в поле R можно отнести дискретную $\Delta(n)$- функцию и функцию единичного скачка $1(n)$, заданные соответственно в виде

$\Delta(n) = 1$ при $n = 0$ и $\Delta(n) = 0$ в остальных случаях,

$1(n) = 1$ при $n >= 0$ и $\Delta(n) = 0$ в остальных случаях.

В поле C можно привести дискретную комплексную экспоненциальную функцию

$$s(i\omega nT_0) = \exp(i2\pi f_s nT_0) = \cos(2\pi f_s n\, T_0) + i\sin(2\pi f_s n\, T_0)$$

Примером сигналов с заданными свойствами могут служить псевдослучайные последовательности (ПСП) , которые формируются, либо через рекуррентные соотношения, либо через формальные степенные ряды (полиномы) [2]. При этом широкое применение ПСП получили как в конечных, так и в бесконечных полях. Так при имитационном математическом моделировании, синтезируются ПСП с помощью ДСЧ-датчиков случайных чисел с заданным законом распределения, чаще всего с равномерным законом.

При исследовании реального цифрового сигнала $\{s(n)\}$ он может рассматриваться, как случайный временной ряд (случайный процесс) с определёнными статистическими свойствами . Например, для речевого цифрового сигнала можно использовать модель стационарного в широком смысле случайного процесса $\{\ \bar{s}(n)\}$, т. е. когда для каждого значения n статистическое среднее процесса есть величина постоянная. Тогда для N-выборки в общем случае комплексного сигнала $\{s(n)\}$ можно применить упрощённое известное выражение[3] для смещённой оценки апериодической автокорреляционной функции (АКФ)

$$\hat{R}(k) = \frac{1}{N}\sum_{n=0}^{N-k-1} s_n \cdot s^*_{n+k}\,. \qquad (2.3)$$

Для распознавания цифровых сигналов нашли широкое применение периодические АКФ, которые в общем виде могут быть представлены

$$\hat{R}(k) = \frac{1}{N}\sum_{n=0}^{N-1} s_n \cdot s^*_{<n+k>_N} \qquad (2.4)$$

где сумма $<n+k>_N$ вычисляется по модулю N.

Выражения (2.3) и (2.4) относятся к средствам корреляционного анализа-распознавания сигнала во временной области. Для периодических АКФ от формы (4) можно перейти соответственно к векторно-матричной форме

$$R_N = Q_N \cdot S_N, \qquad (2.5)$$

где Q_N- правоциркулянтная матрица с образующим вектором-строкой вида $\bar{Q}_N = s_0, s_1, \ldots, s_{N-1}$, $S_N = \bar{Q}'_N$- вектор-столбец отсчётов сигнала.

Форма (2.5) будет использована в дальнейшем для создания быстрых вычислительных алгоритмов.

Последовательностную модель можно применить также для двумерных цифровых сигналов, среди которых наиболее распространёнными являются цифровые видеосигналы. При этом форма (2.1) приобретает вид

$$\{ s(m,n) \}, \quad m = \overline{0, M-1}, \; n = \overline{0, N-1}, \tag{2.6}$$

где $s(m,n)$ – функция двух вещественных переменных -пространственных координат m и n соответственно с шагом приращения Δm и Δn. В теории и практике цифровой обработки сигналов получили широкое применение преобразования одномерных сигналов в двумерные и наоборот. Чаще всего такие приёмы используются при построении различных алгоритмов быстрых ортогональных преобразований. Примером может служить также дискретная комплексная экспонента, которая легко преобразуется в функцию двух переменных

$$s(m\Delta f,\, nT_0) = \exp(i2\pi m\Delta f_S nT_0) = \cos(2\pi m\Delta f_S nT_0) + i\cdot\sin(2\pi m\Delta f_S nT_0),$$

где Δf - шаг приращения по частоте (Гц) периодического сигнала.

От формы (2.6) можно перейти к пространственной векторно-матричной форме, либо к пространственно-временной, которые используются, например, при обработке подвижных изображений.

В зависимости от поля, в котором задана последовательность, используется соответствующая арифметика данного поля при компьютерной реализации алгоритмов её обработки. В компьютерных средствах и, в частности, в мультимедиа средствах, широкое применение нашли технологии обработки вещественных двоичных с равномерным квантованием аудио и видео сигналов. Цифровые аудио и видео сигналы в ЭВМ чаще всего хранятся в виде стандартных файлов, где отсчёты сигналов могут быть представлены как целыми бесзнаковыми, так и дробными со знаком вещественными ограниченной разрядности двоичными числами. Для обработки таких данных в ЭВМ, как известно, используется две формы представления чисел и соответственно две арифметики: двоично-разрядная арифметика с фиксированной точкой (запятой) и арифметика с плавающей точкой (запятой).

В арифметике с плавающей точкой вещественное число $x = (\pm a_x, \pm u_x)$ представлено двумя величинами - порядком $\pm a_x$ и значащей частью числа (мантиссой) $\pm u_x$, где порядок и мантисса представлены в форме с фиксированной точкой. С целью упрощения вычислений в арифметике с пла-

вающей точкой принято использовать нормализованные числа с положительными значениями порядка и с дробными значениями мантиссы. Главным преимуществом арифметики с плавающей точкой является высокая точность вычислений, обусловленная широким динамическим диапазоном представления данных. Однако специфика выполнения операций сложения в такой арифметике требует дополнительных вычислительных и временных затрат по сравнению с арифметикой с фиксированной точкой. Имеется большое количество изобретений на программно-аппаратные средства арифметики с плавающей точкой, обладающих высоким быстродействием. Однако в сравнении с вычислительными технологиями на основе арифметики с фиксированной точкой по быстродействию они значительно уступают.

Основными задачами при обработке цифровых сигналов является их кодирование–синтез новых последовательностей с заданными свойствами и их распознавание–декодирование в том числе на фоне активных и пассивных помех. Приэтом кодирование и распознавание сигналов может осуществляться как в исходной временной (или пространственной) области, так и в области преобразований. Для решения таких задач используются различные методы и алгоритмы, которые определяют выбор тех или иных форм представления исходных сигналов. Среди прочих наибольшее применение получили методы и алгоритмы линейной цифровой фильтрации и быстрых ортогональных преобразований, на основе которых строятся технологии сжатия, корреляционного и спектрального анализа сигналов [4].

2.2. Математические модели цифровой линейной фильтрации

При распознавании цифровых сигналов нашли широкое применение модели цифровых линейных систем. Одномерная цифровая линейная инвариантная к сдвигу система (ЛИС) - система с постоянными параметрами, во временной области может быть задана соотношением вход-выход в виде разностного уравнения

$$\sum_{k=0}^{K-1} a_k\, y(n-k) = \sum_{k=0}^{L-1} b_k\, x(n-k)\,, \qquad (2.7)$$

где $x(n)$ и $y(n)$ - дискретные функции соответственно входа и выхода системы.

Без потери общности принимая $a_0 = 1$, из выражения (2.7) получим

рекуррентное соотношение для определения отклика системы на входное воздействие при заданном исходном состоянии системы

$$y(n) = -\sum_{k=0}^{K-1} a_k\, y(n-k) + \sum_{k=0}^{L-1} b_k\, x(n-k). \qquad (2.8)$$

Из-за присутствия в системе (2.8) рекурсивной составляющей её импульсный отклик по длительности является бесконечным. Примечательно, что каждая из суммируемых частей в (2.7) является цифровой свёрткой, которая может быть записана в общем виде

$$y(n) = \sum_{k=0}^{\infty} x(k)\ h(n-k);$$
$$y(n) = \sum_{k=0}^{N-1} h(k)\, x(n-k) \qquad (2.9)$$

где $h(n)$- импульсный отклик системы,являющийся по длине в первом случае бесконечным, а во втором- конечным. Вторая форма приведена с учётом коммутативности операции свёртки.

Из разностных уравнений (2.8) и (2.9) следует реальный способ практической реализации цифровых линейных фильтров, например, на основе сдвиговых регистров с сумматорами и умножителями в заданном числовом поле можно представить [5] следующие схемные реализации (Рис.2.1-Рис.2.4):

1) схемы рекурсивных фильтров:

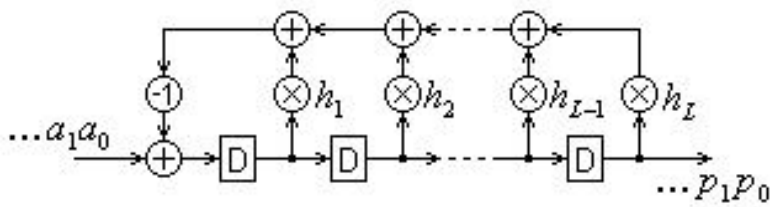


Рис.2.1. Схема рекурсивного фильтра с сумматорами вне цепи задержки, заданного разностным уравнением

$$p_{j+L} = \sum_{i=1}^{k-1} h_i\, p_{j+L-i} + a_j$$

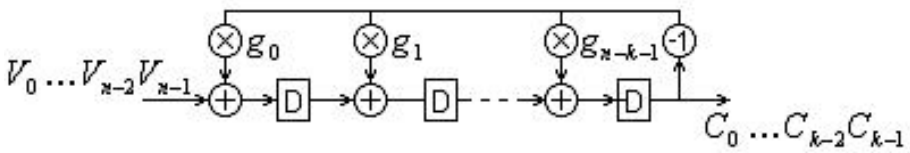


Рис.2.2. Схема рекурсивного фильтра с сумматорами в цепи задержки и заданного в полиномиальной форме C(x)=V(x)/g(x)

2) схемы КИХ - фильтров, заданных в форме цифровой свёртки вида -

$$b_j = \sum_{i=0}^{L} g_i a_{j-i},$$

либо произведением многочленов $b(x)= g(x)\ a(x)$.

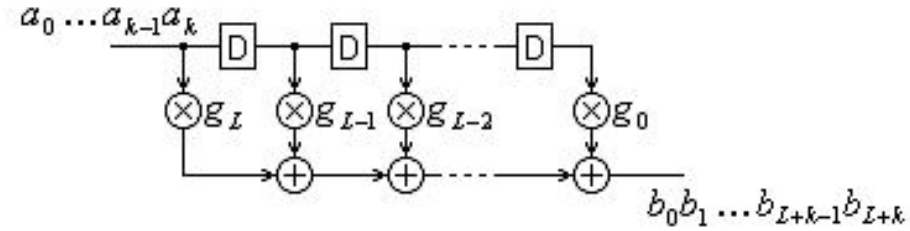


Рис.2.3. Схема КИХ-фильтра с сумматорами вне цепи задержки

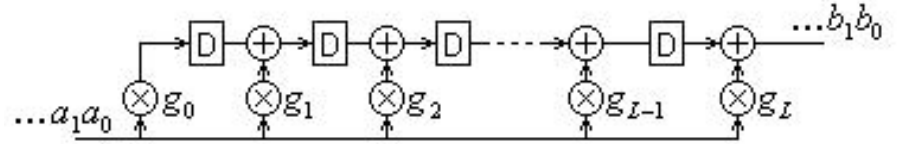


Рис.2.4. Схема КИХ-фильтра с сумматорами в цепи задержки

При этом импульсная характеристика определена через последовательности коэффициентов $\{a_k\}$, $\{b_k\}$ фильтра. Для циклической свёртки во втором выражении длина последовательностей является одинаковой $n,k = \overline{0,N-1}$, значение функции $x(n\text{-}k)$ вычисляется в виде $x<n\text{-}k>_N$, т.е. значение аргумента в угловых скобках вычисляется по модулю N.

Операция свёртки (2.9) может быть применена для вычисления комплексных взаимно-корреляционных последовательностей

$$R_{xy}(n) = x(-n) * \dot{y}\ (n),$$

где $(*)$ - оператор свёртки. Поэтому существует общность в формах представления математических моделей линейных свёрток и взаимных и ав-

токорреляций. Например, цифровые циклические свёртки по аналогии с выражением (2.5) для периодических корреляциий могут быть заданы также в векторно-матричном виде

$$Y_N = S_N \cdot \bar{x}_N ,$$

где S_N - левоциркулянтная матрица с образующим вектором - строкой $\bar{h}_N = h_0, h_1,....,h_{N-1}$, соответствующим последовательности $\{h(n)\}$, $\bar{x}_N$ - вектор - столбец, соответствующий инвертированной последовательности $\{x(n)\}$.

Существует также возможность представления линейных свёрток в полиномиальном виде после замены каждой из исходных последовательностей формально-степенным рядом, т.е. можно записать полиномиальную форму

$$Y(z)=X(z)\ H(z) , \qquad (2.10)$$

которая непосредственно следует из разностных уравнений (2.9).

Для циклической свёртки форма (2.10) приобретает вид

$$Y(z)=X(z)\ H(z) \bmod(z^N - 1). \qquad (2.11)$$

Наряду с математическими моделями во временной области для анализа сигналов и самих линейных систем используются методы и модели в области преобразований, из которых наиболее широкое применение получили z- и Фурье-преобразования.

Причём z-преобразование используется только в поле комплексных чисел и основой его построения служит применение в качестве входной экспоненциальной последовательности $\{x(n)\}=\{e^{sn}\}$, являющейся для ЛИС собственной функцией. В этом случае в области z-преобразования цифровой сигнал, например, $\{x(n)\}$ приобретает полиномиальную форму

$$X(z)=\sum_{n=0}^{N-1} x_n\ z^{-n} ,$$

Подобно преобразованию Лапласа для непрерывных ЛИС, с помощью дробно-полиномиальной передаточной функции $W(z)=Y(z)/X(z)$ в области z-преобразования можно осуществлять анализ устойчивости дискретных ЛИС. Заметим, что для элемента задержки в один такт передаточная функция равна

$$W(z) = z^{-1} .$$

Используя принятое обозначение $H(z)$, как импульсного отклика в области z-преобразования, получим выражение, совпадающее с выражением (2.10), которое чаще всего представляют в виде одного из свойств z-преобразования. Подстановкой в $H(z)$ значения $z = e^{i\omega}$ получим частотную характеристику $H(i\omega)$ системы.

В общем случае от z-преобразования сигнала $x(n)$ осуществляется переход

$$\textit{в прямое}\quad X(i\omega) = \sum_{n=\infty}^{\infty} x(n)\, e^{-i\omega} \quad \textit{и обратное}\quad x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(i\omega)\, e^{i\omega}\, d\omega$$

-Фурье-преобразования соответственно при $z = e^{i\omega}$ и при $z = e^{-i\omega}$, а полиномиальная форма (2.11) приобретает вид

$$Y(i\omega) = X(i\omega)\, H(i\omega)\,.$$

Для сигнала $s(n)$ обратный переход от полиномиальной формы в области z-преобразования во временную область можно выполнить простой заменой полиномиальной на последовательностную форму. Заметим, что именно форма (2.10) используется для синтеза быстрых алгоритмов вычисления цифровых свёрток, представленных через формальные степенные ряды [2].

Для вычисления двумерных линейных цифровых свёрток во временной области можно воспользоваться соотношением

$$y(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} h(k_1, k_2)\;\; x(n_1 - k_1, n_2 - k_2)\;\;,$$

где параметрами $n_1 = \overline{0, N_1 - 1},\quad n_2 = \overline{0, N_2 - 1}$ определена опорная область фильтра.

Полагая, что входной сигнал представлен собственными функциями $x(n_1,n_2) = {z_1}^{n1}\, {z_2}^{n2}$, получим передаточную функцию в области z-преобразования

$$H(z_1, z_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} h(k_1, k_2)\; {z_1}^{-k_2}\; {z_2}^{-k_2}\;\;, \qquad (2.12)$$

являющейся z-преобразованием импульсного отклика $h = (k_1, k_2)$ системы. В общем случае выражение (2.12) может быть использовано для получе-

ния z-преобразования любой заданной на конечном интервале дискретной функции. Определяя таким способом z-преобразования для функций $x(n_1,n_2)$ и $y(n_1,n_2)$ можно записать линейную двумерную свёртку в виде произведения полиномов

$$Y(z_1, z_2) = X(z_1, z_2)\, H(z_1, z_2)\,.$$

При значениях переменных $z_1 = e^{i\omega_1}$, $z_2 = e^{i\omega_2}$ передаточная функция $H(i\omega_1, i\omega_2)$ становится частотным откликом двумерной ЛИС.

Для многомерных цифровых свёрток можно использовать более компактную в записи векторную форму представления:
– во временной области

$$y(\bar{n}) = \sum_{\bar{k}} x(\bar{k})\, h(\bar{n} - \bar{k})\,,$$

где $\bar{n}$ *и* $\bar{k}$ - целочисленные векторы-аргументы дискретных функций,
– и в области *z*-преобразования $Y(\bar{z}) = X(\bar{z})\, H(\bar{z})$, где $\bar{z}$- вектор полиномиальных переменных.

2.3. Дискретное преобразование Фурье

Математические модели одномерного дискретного преобразования Фурье (ДПФ) чаще всего связывают с полем комплексных чисел, где дискретные базисные функции могут быть представлены двумерными комплексными экспонентами

$$f(k,n) = \exp(-i2\pi\, k\, n / N) = \cos(2\pi\, k\, n / N) - i\sin(2\pi\, k\, n / N)\,, \quad (2.13)$$

где $k, n = \overline{0, N-1}$, *N*- длина базиса. Широкое использование базисных функций (2.13) обусловлено значительным сходством их гармонической формы с сигналами, полученными при описании многих физических явлений и процессов. Заметим, что форма (2.13) записи дискретных базисных функций является формой представления двумерного цифрового сигнала в поле комплексных чисел. Приэтом двумерный цифровой сигнал $f(k,n)$ можно рассматривать, как множество одномерных сигналов $f_k(n)$.

Форма (2.13) является одной из разновидностей математических моделей, позволяющая осуществлять смысловую взаимосвязь базисных функций с физическими сигналами. Однако для построения быстрых вычисли-

тельных алгоритмов нашло применение общее для всех полей (конечных и бесконечных, простых и расширенных) матрично-полиномиальное определение ДПФ через корни α_N^j многочлена $f(z) = z^N - 1$, образующих циклическую мультипликативную группу $G_N = \{1, \alpha_N^1, \alpha_N^2, ..., \alpha_N^{N-1}\}$ в виде матриц прямого F_N и обратного F_{N-1} преобразований:

$$F_N = \left\| \alpha_N^{kn} \right\|, \quad F_N^{-1} = \left\| \alpha_N^{-kn} \right\|$$

В развёрнутом виде матрица ДПФ формируется с учётом вычисления показателей степеней корней по модулю N

$$F_N = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha_N & \alpha_N^2 & \alpha_N^3 & \cdots & \alpha_N^{N-1} \\ 1 & \alpha_N^2 & \alpha_N^4 & \alpha_N^6 & \cdots & \alpha_N^{N-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \alpha_N^{N-1} & \alpha_N^{N-2} & \alpha_N^{N-3} & \cdots & \alpha_N \end{pmatrix}.$$

Из бесконечных полей возможность задания матриц ДПФ степенями корней многочлена $f(z) = z^N - 1$ существует только в поле комплексных чисел, где $\alpha_N = \exp(-i2\pi/N)$, а сам многочлен получил название многочлена деления круга (деления на N равных частей окружности единичного радиуса с центром на комплексной плоскости) .

В полях Галуа существуют неограниченные возможности в построении матриц ДПФ, которые непосредственно следуют из того, что все ненулевые элементы конечного поля образуют циклическую мультипликативную группу. В качестве примера приведём матрицы ДПФ в поле $GF(2^3-1)$, образованном по модулю 7 (простого числа Мерсенна)

$$F_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & -1 & -3 & -2 \\ 1 & 2 & -3 & 1 & 2 & -3 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -3 & 2 & 1 & -3 & 2 \\ 1 & -2 & -3 & -1 & 2 & 3 \end{pmatrix}.$$

Матрицей обратного преобразования будет $F_6^{-1} = (-1) \cdot F_6$,

$$\widehat{F}_6 = J_6 \cdot F_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & -3 & -1 & 2 & 3 \\ 1 & -3 & 2 & 1 & -3 & 2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 2 & -3 & 1 & 2 & -3 \\ 1 & 3 & 2 & -1 & -3 & -2 \end{pmatrix}$$

где $\bar{J}_6$-матрица инверсной перестановки, а множитель (-1) выполняет роль нормирующего коэффициента в поле *GF*(7).

В заданном базисе коэффициенты ДПФ можно определить тремя видами математических моделей :

1) в виде дискретной функции

$$X(k) = \sum_{n=0}^{N-1} x_n \alpha_N^{kn} \text{ , где } k = \overline{0, N-1} \text{ ;}$$

2) в векторно-матричном виде

$X_N = F_N \bar{x}_N$, где X_N *и* $\bar{x}_N$ - векторы-столбцы соответственно коэффициентов ДПФ и отсчётов сигнала $\{x(n)\}$.

3) в полиномиальном виде

$X(k) \equiv X(z) \bmod (z - \alpha_N^k)$, где $X(z) = \sum_{n=0}^{N-1} x_n z^n$ -полином, соответствующий последовательности отсчётов сигнала .

Из свойств ДПФ, являющихся общими для всех полей, следует отметить наиболее часто используемое для практики свойство циклической свёртки, базирующееся на теореме о сдвиге во временной и в частотной областях

$$\Im(x(n \pm n_0)) = e^{\pm i 2\pi k n_0 / N} X(k); \quad \Im(e^{\pm i 2\pi k_0 n / N} x(n)) = X(k \mp k_0),$$

где $\Im$ -оператор ДПФ.

В применении к векторно-матричной форме свойство циклической свёртки можно записать

$$Y_N = S_N \bar{x}_N = F_N^{-1} D_N F_N \bar{x}_N \text{ ,}$$

где $D_N = diag(\bar{d}_N), \quad \bar{d}_N = F_N \bar{h}_N$.

Свойство циклической свёртки можно записать также в последовательностной форме

$$\Im(x(n) \cdot y(n)) = X(k) * Y(k); \Im(x(n) * y(n)) = X(k) \cdot Y(k). \quad (2.14)$$

В поле комплексных чисел ДПФ обладает целым рядом свойств, связанных: во первых- со спецификой структуры расширенных полей, а во вторых- со спектральным и корреляционным анализом дискретизованных во времени и квантованных по уровню вещественно-значных сигналов. Заметим, что при распознавании таких сигналов анализу подлежат их энергетические характеристики, основанные на известном по теореме Парсеваля свойстве

$$\frac{1}{N}\sum_{n=0}^{N-1}|x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2.$$

Свойство (2.14) применительно к корреляционным функциям в поле *C* принимает вид

$$\Im(x(n) * y(-n)) = X(k) \cdot Y^*(k), \quad \Im(x(n) * x^*(-n) = |X(k)|^2.$$

Для вещественных данных справедливо свойство симметрии коэффициентов ДПФ $X(k) = X^*(N - k)$, которое по отношению к ДПФ в расширенных полях Галуа является частным случаем.

При использовании ДПФ для спектрального анализа важнейшим является свойство спектрального разрешения. Общепринятым приближённым определением спектрального разрешения в герцах считается величина, обратная длине выборки анализируемого сигнала в секундах.

При этом шаг дискретизации по частоте определяется в виде отношения $\Delta f_d = f_d / N$, где $f_d = 1/\Delta t$. Другим, также общепринятым в литературе, определением спектрального разрешения считается способность различать спектральные отклики двух синусоид, близких по частоте. Однако сам процесс дискретизации, как во временной,так и в частотной областях, сопровождается рядом факторов, снижающих различимость сигналов с помощью ДПФ.

Одним из таких факторов является эффект «размывания» спектральных оценок, сопровождаемый помехами в виде боковых лепестков спектральных составляющих сигнала. Формально причиной данного эффекта

является ограничение длины выборки анализируемого сигнала с помощью прямоугольной весовой единичной функции-окна

$$w(n) = \begin{vmatrix} 1, & 0 \le n \le N-1 \\ 0 & otherwise \end{vmatrix},$$

так, что из бесконечной последовательности $\{\hat{x}(n)\}$ формируется сигнал

$$x(n) = \hat{x}(n)\ w(n)\ .$$

Тогда спектральные оценки X(k) принимают форму свёртки

$$X(k) = \hat{X}(k) * D(k), \qquad (2.15)$$

где $D(k) = \exp(-i\pi k)\dfrac{\sin(\pi\, k\, N)}{\sin(\pi\, k)}$ - дискретная *sinc*-функция.

Пример спектрограммы для прямоугольной единичной весовой функции длины N=8 приведён на рис.2.5, откуда видны боковые спекральные составляющие (дополнительные к основной $W(0)$). На рис2.6 представлена спектрограмма тестового синусного сигнала $x(n) = A\sin 2\pi T^{-1} n\Delta t$, где видны боковые лепестки, проявившиеся через свёртку (2.15) в спектральной области. Наличие помех в виде боковых лепестков в спектрограммах ДПФ существенно снижает спектральное разрешение в смысле различения сигналов с близко расположенными частотами. На практике для уменьшения влияния эффекта размывания частот используются во временной области весовые функции, сглаживающие сигнал на концах выборки. В настоящее время имеется достаточно много таких окон, которые качественно оцениваются по ширине полосы главного лепестка на уровне 3 дБ и по максимальному (пиковому) уровню боковых лепестков и скорости их снижения. При выборе окон важную роль играет их вычислительная сложность, представленная чаще всего количеством и типом арифметических операций. С этой точки зрения окна могут быть использованы, как во временной, так и в спектральной областях.

В практических приложениях наиболее часто используются окна, заданные во временной области в виде синусных, либо косинусных дискретных функций. Например, окно Ханна может быть задано двумя равноценными в приложениях функциями

$$w_1(n) = \sin^2(\pi\, n / N) = (1 - \cos(2\pi\, n / N)\, 2^{-1},$$

либо $w_2(n) = \cos^2(\pi n / N) = (1 + \cos(2\pi n / N)2^{-1}$.

В спектральной области данное окно записывается выражением

$$W(k)=0.5[D(k) \pm 0.5\ (D(k\text{-}1)+D(k+1))]\ ,$$

где $D(k)$ – k-е значение окна Дирихле.

Близким к окну Ханна является окно Хэмминга, имеющего форму приподнятого косинуса и записываемого во временной и спектральной областях соответственно

$$w(\mathrm{n})= 0.54+0.46cos(2\pi\ n/N),$$
$$W(\mathrm{k})=0.54D(\mathrm{k})+ 0.23(D(\mathrm{k\text{-}1})+D(\mathrm{k+1})).$$

В работе [6] было предложено использовать для спектральных окон компенсационные последовательности, которые для окон Хана и Хэмминга могут быть представлены соответственно в виде

$$B_1(k) = \{2^{-1}[|D(k)| - 2^{-1}(|D(k-1)| + |D(k+1)|)]\}^1,$$

$$B_2(k) = 0.54\,|D(k)| + 0.23[|D(k-1)| + |D(k+1)|]\}^1.$$

Поскольку эти последовательности образованы через модульное представление k-х значений окна Дирихле, то удобно назвать вновь полученные окна модульно-комбинированными спектральными весовыми функциями. В итоге можно использовать компенсированное окно

$$Y(k) = B(k) + |D(k)|.$$

Данные окна обладают значительными преимуществами перед другими спектральными окнами, совмещая низкий уровень боковых лепестков с минимальной шириной главного лепестка.

В качестве примера ниже на рис.2.5 приведен фрагмент текста программы, написанной в редакторе пакета Mathcad, с помощью которой формируется тестовый гармонический цифровой сигнал $y = 2_v$, представленный суммой двух гармоник, на порядок отличающихся по амплитуде с длиной выборки $N = 64$ и с частотой дискретизации 700 Гц. На рисунке представлены также график самого сигнала и его спектрограмма с окном Дирихле. Для сравнения на рис. 2.6а,б представлены спектрограммы соответственно с окном Ханна и с компенсированным окном Ханна, из которых можно видеть значительное преимущество по спектральному разрешению компенсационного окна.

$n1 := 0..63 \qquad m1 := 0..63$

$f3_{m1,n1} := e^{-i\cdot 2\cdot\pi\cdot m1\cdot\frac{n1}{64}}$

matrix(m1, n1, f3)

$\nu := 0..63$

$$y2_\nu := 10\cdot\sin\left(2\cdot\pi\cdot 100\cdot\frac{\nu}{700}\right) + \cos\left(2\cdot\pi\cdot 130\cdot\frac{\nu}{700}\right)$$

$Y4 := f3\cdot y2$

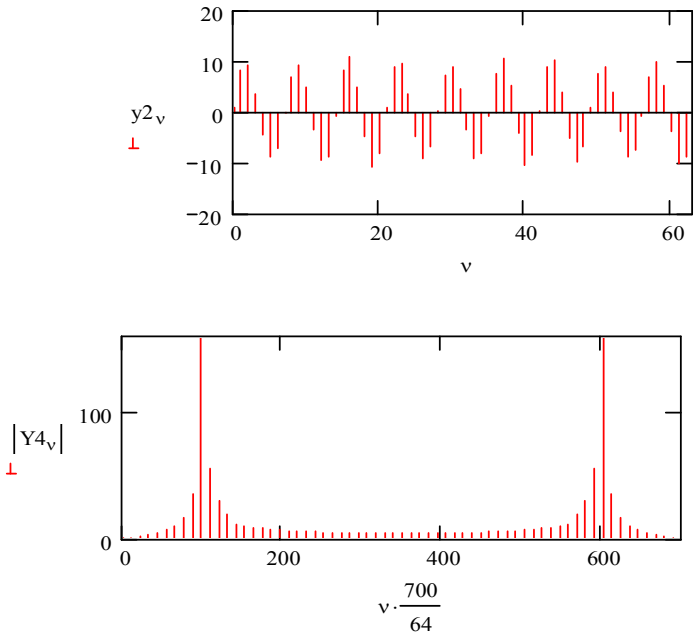
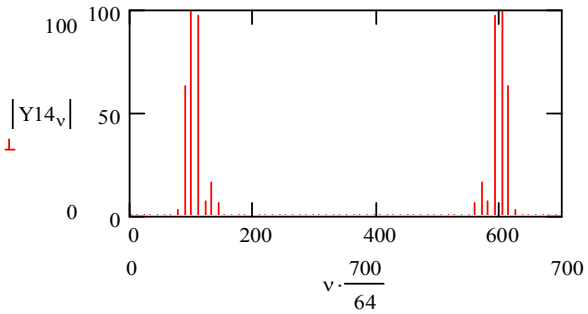


Рис.2.5. Исходный тестовый сигнал и его спектрограмма с окном Дирихле

а)



б)

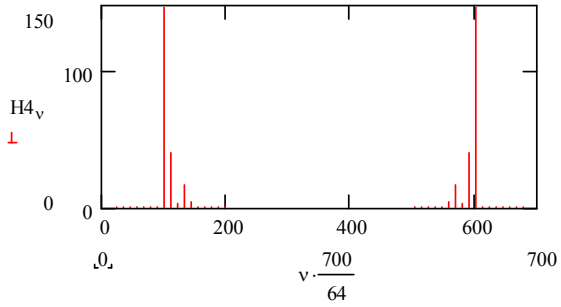


Рис. 2.6 а) - Спектрограмма с окном Ханна,
б) - Спектрограмма с компенсированным окном Ханна

Таблица 2.1

Сравнительные характеристики спектральных окон

Тип окна	Макс. уровень бок.лепестков $V_{макс}$, *дБ*	Ширина гл. лепестка по $V_{макс}$, *дБ*	Когерентное усиление	Эквивал. шумовая полоса, бин
Компенс. окно Ханна	N=8 -49 N=16 -66 N=128 -117	1,6 1,8 2	1	1
Дирихле	-13	1,64	1	1
Ханна	-32	4	0,55	1,5
Хемминга	-43	4	0,54	1,36
Блекмана-Хэрриса	-90	8	0,36	2

Раздел 3. БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ (БПФ) В ПОЛЕ КОМПЛЕКСНЫХ ЧИСЕЛ

3.1. Последовательностные формы представления БПФ

Для построения различных алгоритмов БПФ первоначально чаще использовалась последовательностная форма представления ДПФ. В частности, вывод наиболее часто используемых алгоритмов БПФ Кули-Тьюки [7] базировался на выражениях

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} W^{2mk} + W^k \sum_{m=0}^{N/2-1} x_{2m+1} W^{2mk},$$
$$X_{k+N/2} = \sum_{m=0}^{N/2-1} x_{2m} W^{2mk} - W^k \sum_{m=0}^{N/2-1} x_{2m+1} W^{2mk}, \qquad (3.1)$$
$$k = \overline{0, N/2-1}.$$

Эти выражения служат основой для построения алгоритмов БПФ с основанием два. Суть данного метода синтеза алгоритмов БПФ сводится к тому, что вычисление N-точечного ДПФ осуществляется рекурсивно через $N/2$-точечные ДПФ. При этом исходная (во временной области) последовательность отсчётов сигнала делится на две подпоследовательности с чётными и нечётными индексами. По этой причине данный алгоритм получил название алгоритма БПФ с прореживанием по времени. К образованным подпоследовательностям снова применяют выражения (3.1). Деление заканчивается на подпоследовательностях длины два. Поэтому условием построения БПФ по основанию два является факторизация длины N по степеням числа два, т.е. $N = 2^n$.

По аналогии с выражениями (3.1) можно записать соотношения

$$X_{2k} = \sum_{m=0}^{N/2-1} (x_m + x_{m+N/2}) W^{2mk}, \quad k = \overline{0, N/2-1},$$
$$X_{2k+1} = \sum (x_m - x_{m+N/2}) W^m W^{2mk}, \quad k = \overline{0, N/2-1}, \qquad (3.2)$$

которые служат основой для построения алгоритмов БПФ с основанием два с прореживанием частотных коэффициентов ДПФ. Суть построения алго-

ритмов БПФ с прореживанием по частоте остается той же – рекурсивное сведение вычислений N-точечного ДПФ к двум $N/2$-точечным ДПФ [7].

На основании выражений (3.1) и (3.2) можно построить графовые модели алгоритмов БПФ, которые соответствуют конкретным значениям N и являются наиболее доступными для построения программных моделей БПФ.

Для того, чтобы убедить читателя в возможности использования выражений (3.1) и (3.2) для построения графов, достаточно предложить самостоятельно попытаться построить граф БПФ с большей размерностью, например, для $N = 16$.

По аналогии с (3.1) и (3.2) могут быть записаны рекурсивные выражения для БПФ по основанию четыре, где N-точечное БПФ сводится к вычислению четырех $N/4$-точечных БПФ. Например, для БПФ с прореживанием по времени можно записать четыре выражения [7]

$$X_k = \sum_{l=0}^{3} W_N^{lk} \sum_{m=0}^{N/4-1} x_{4m+l} W_N^{4mk},$$

$$X_{k+N/4} = \sum_{l=0}^{3} (-i)^l W_N^{lk} \sum_{m=0}^{N/4-1} x_{4m+l} W_N^{4mk},$$

(3.3)

$$X_{k+N/2} = \sum_{l=0}^{3} (-1)^l W_N^{lk} \sum_{m=0}^{N/4-1} x_{4m+l} W_N^{4mk},$$

$$X_{k+3N/4} = \sum_{l=0}^{3} i^l W_N^{lk} \sum_{m=0}^{N/4-1} x_{4m+l} W_N^{4mk},$$

$$k = \overline{0, N/4-1}.$$

В работе [7] не приведён пример построения графа по выражениям (3.3). Надо заметить, что это не простая задача, так как рекурсивные выражения (3.3) составлены для одного коэффициента относительно четырех отсчётов сигнала. Еще сложнее дело обстоит с более старшими основаниями, не говоря уже об их сравнительном анализе. Поэтому обратимся к методам синтеза БПФ на основе их векторно-матричных форм представления. В частности, будут рассмотрены методы характеризационно-инвариантного синтеза быстрых ортогональных преобразований (БОП). Суть метода состоит в использовании такого набора свойств или

признаков базисных функций, через которые можно было бы обобщить их на различные ортогональные базисы и быстрые алгоритмы [10].

3.2. Матрично-рекурсивные формы БПФ по основанию два

Представим выражение для ДПФ над полем комплексных чисел C в матрично-векторной форме:

$$X_N = F_N \cdot x_N ,$$

$$F_N = \left\| \exp(-i\,2\pi\, km / N) \right\|, \quad k, m = \overline{0, N-1};$$

где X_N – вектор коэффициентов над полем C, x_N – вектор, соответствующий последовательности отчетов входного сигнала

Матрицу F_N запишем в блочном виде с использованием J_N-матрицы чёт-нечёт перестановки строк

$$F_N = J'_N \left(\begin{array}{c|c} F_{N/2} & F_{N/2} \\ \hline R_{N/2} & -R_{N/2} \end{array} \right). \tag{3.4}$$

В свою очередь матрицу-блок $R_{N/2}$ можно выразить через матрицы $F_{N/2}$ и $F^{-1}{}_{N/2}$ в двух формах:

$$R_{N/2} = R_{N/2} F^{-1}{}_{N/2} F_{N/2} = S_{N/2} F_{N/2}, \tag{3.5}$$

$$R_{N/2} = F_{N/2} F^{-1}{}_{N/2} R_{N/2} = F_{N/2} D_{N/2}, \tag{3.6}$$

где $S_{N/2} = R_{N/2} F_{N/2}^{-1}$, $\quad D_{N/2} = F_{N/2}^{-1} R_{N/2}$.

С учётом (3.5) и (3.6) из (3.4) получим следующие две блочно-рекурсивные формы для матрицы F_N

$$F_N = J'_N \left(\begin{array}{c|c} F_{N/2} & F_{N/2} \\ \hline F_{N/2} D_{N/2} & -F_{N/2} D_{N/2} \end{array} \right) \tag{3.7}$$

$$F_N = J'_N \left(\begin{array}{c|c} F_{N/2} & F_{N/2} \\ \hline S_{N/2} F_{N/2} & -S_{N/2} F_{N/2} \end{array} \right). \tag{3.8}$$

В результате для обобщённого числа шагов рекурсии можно получить

две факторизованные формы представления алгоритмов БПФ по основанию два с прореживанием по частоте:

$$F_N = \widetilde{J}_N \operatorname{diag}\begin{Bmatrix} 1 & 1 \\ 1 & -1 \end{Bmatrix} \operatorname{diag}\left\{\begin{pmatrix} I_2 & I_2 \\ D_2 & -D_2 \end{pmatrix}\right\} \text{x} \ldots \text{x} \begin{pmatrix} I_{N/2/} & I_{N/2} \\ D_{N/2} & -D_{N/2} \end{pmatrix}, \qquad (3.9)$$

$$F_N = J'_N \begin{pmatrix} I_{N/2} & I_{N/2} \\ S_{N/2} & -S_{N/2} \end{pmatrix} \text{x} \ldots \text{x} \operatorname{diag}\left\{ J'_4 \begin{pmatrix} I_2 & I_2 \\ S_2 & -S_2 \end{pmatrix}\right\} \operatorname{diag}\left\{\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right\}, \qquad (3.10)$$

где $\widetilde{J}_N$ - матрица двоично-инверсной перестановки строк получается в результате рекурсивного применения J_N – матриц).

Факторизованные формы (3.9) и (3.10) являются основными. С помощью их, например, посредством транспонирования можно строить различные модификации алгоритмов БПФ по основанию два. Естественно, что любая такая модифицированная форма алгоритма БПФ по отношению к исходной будет давать одинаковую вычислительную сложность.

Принципиальное отличие алгоритмов БПФ, построенных по формам (3.9) и (3.10), состоит в том, что первый из них построен на рекурсивном использовании слабо заполненных диагональных матриц D_N, а второй – на использовании циркулянтных матриц S_N.

Диагональная матрица D_N определяется выражением

$$D_N = \operatorname{diag}\left\{\exp(-i\pi j/N)\right\}_{j=0}^{N-1}.$$

Таким образом, с помощью блочно-рекурсивной формы (3.9) получили алгоритмы БПФ по основанию два, известные как алгоритмы Кули-Тьюки [3]. В соответствии с матричной факторизацией (3.9) строится векторный ориентированный граф (рис.3.1). Заметим, что представление БПФ в форме сигнального ориентированного графа является наиболее удобным для построения вычислительных алгоритмов БПФ, пригодных для их интерпретации символическими языками программирования.

3.3. Алгоритмы БПФ по μ-основанию

Рассмотрим сначала вывод блочно-рекурсивных и факторизованных форм для случая $\mu = 4$, воспользовавшись при этом матрично-рекурсивной формой:

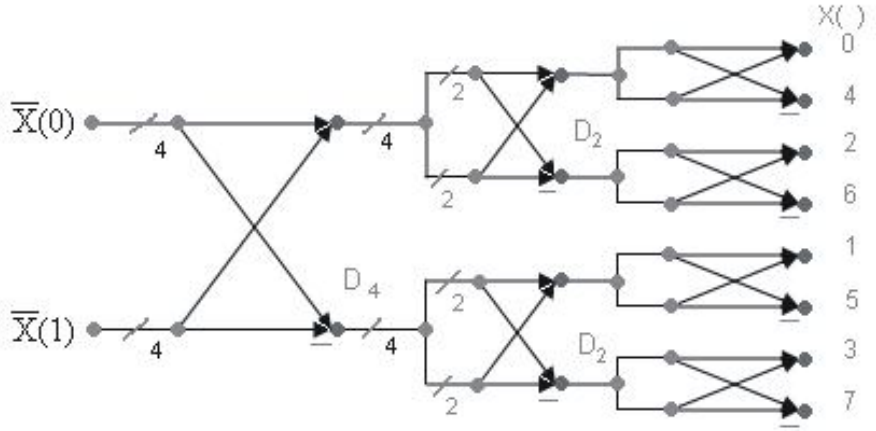


Рис.3.1. Векторный граф восьмиточечного БПФ по основанию два с прореживанием по частоте

$$\widetilde{F}_N = \begin{pmatrix} \widetilde{F}_{N/2} & \widetilde{F}_{N/2} \\ \widetilde{F}_{N/2} D_{N/2} & -\widetilde{F}_{N/2} D_{N/2} \end{pmatrix}.$$

Применяя к матрицам эту же рекурсию, получим блочно-рекурсивную форму:

$$\widetilde{F}_N^{(4)} = \begin{pmatrix} \widetilde{F}_{N/4} & \widetilde{F}_{N/4} & \widetilde{F}_{N/4} & \widetilde{F}_{N/4} \\ \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} & \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} \\ \begin{pmatrix} \widetilde{F}_{N/4} & \widetilde{F}_{N/4} \\ \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} \end{pmatrix} D_{N/2} & -\begin{pmatrix} \widetilde{F}_{N/4} & \widetilde{F}_{N/4} \\ \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} \end{pmatrix} D_{N/2} \end{pmatrix} =$$

$$= \begin{pmatrix} \widetilde{F}_{N/4} & \widetilde{F}_{N/4} & \widetilde{F}_{N/4} & \widetilde{F}_{N/4} \\ \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} & \widetilde{F}_{N/4} D_{N/4} & -\widetilde{F}_{N/4} D_{N/4} \\ \widetilde{F}_{N/4} D_{N/4}^{(1)} & \widetilde{F}_{N/4} D_{N/4}^{(1)} i & -\widetilde{F}_{N/4} D_{N/4}^{(1)} & -\widetilde{F}_{N/4} D_{N/4}^{(1)} i \\ \widetilde{F}_{N/4} D_{N/4}^{(2)} & -\widetilde{F}_{N/4} D_{N/4}^{(2)} i & -\widetilde{F}_{N/4} D_{N/4}^{(2)} & \widetilde{F}_{N/4} D_{N/4}^{(2)} i \end{pmatrix}, \qquad (3.11)$$

где $$D_{N/2} = \operatorname{diag}\{\exp(-i\,2\pi\,k/N)\}_{k=0}^{N/2-1},$$

$$D_{N/4}^{(1)} = \operatorname{diag}\{\exp(-i\,2\pi\,k/N)\}_{k=0}^{N/4-1},$$

$$D_{N/4}^{(2)} = D_{N/4} D_{N/2},$$

$$D_{N/4} = \mathrm{diag}\left\{\exp(-i\,4\pi\,k/N)\right\}_{k=0}^{N/4-1}.$$

Из (3.11) можно получить факторизованную форму

$$\widetilde{F}_N = (I_4 \otimes F_{N/4})(I_{N/4} \oplus D_{N/4}^{(1)} \oplus D_{N/4}^{(2)} \oplus D_{N/4}^{(3)})(F_4 \otimes I_{N/4}),$$

или с учетом матриц перестановки $J_N^{(4)}$ строк по mod 4

$$F_N = (J_N^{(4)})'(I_4 \otimes \widetilde{F}_{N/4})(I_{N/4} \oplus D_{N/4}^{(2)} \oplus D_{N/4}^{(3)})(\widetilde{F}_4 \otimes I_{N/4}), \qquad (3.12)$$

где матрицы $D_{N/4}^{(j)}$ определяются из общего выражения

$$D_{N/4}^{(j)} = \mathrm{diag}\left\{\exp(-i\,2\pi k\; j/N)\right\}_{k=0}^{N/4-1}.$$

Из (3.12) видно, что перестановки строк по основанию μ возникают из двоично - инверсных перестановок после применения к матрицам $\widetilde{F}_4 \otimes I_{N/4}$ и $I_4 \otimes \widetilde{F}_{N/4}$ БПФ с основанием два.

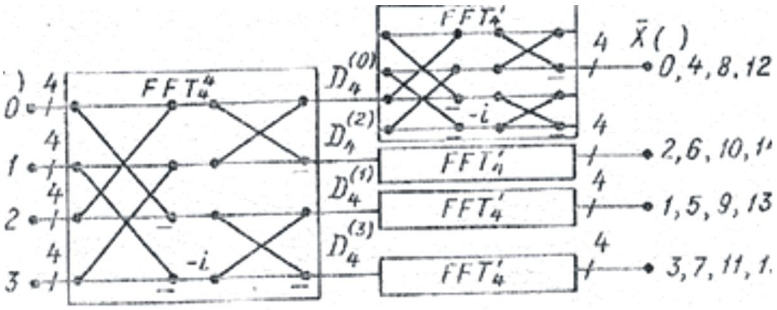


Рис.3.2. Векторный граф 16- точечного БПФ по основанию четыре

Методом индукции несложно выполнить переход для одного шага рекурсии от основания μ = 4 для общего случая μ-основания:

$$F_N = (J_N^{(\mu)})'(I_\mu \otimes F_{N/\mu})\left(\bigoplus_{j=0}^{\mu-1} D_{N/\mu}^{(j)}\right)(F_\mu \otimes I_{N/\mu}), \qquad (3.13)$$

где $D_{N/\mu}^{(j)} = \mathrm{diag}\left\{\exp(-i\,2\pi k\; j/N)\right\}_{k=0}^{N/\mu-1}, \quad j = \overline{0, N/\mu - 1}.$

Обращает на себя внимание тесная взаимосвязь между матрично-рекурсивными формами типа (3.11.) и смешанными факторизованными формами, содержащими операции не только обычного, но и кронекеровского умножения матриц. Но матрично-рекурсивные формы являются наиболее предпочтительными для содержательного, смыслового, анализа алгоритмов БПФ и позволяют получить факторизованные формы как простую характеризацию блочно-рекурсивных форм. Обобщая (3.13), на общее число шагов рекурсии можно получить факторизованную форму матрицы F_N

$$F_N = \widetilde{J}_N^{(\mu)} \prod_{k=0}^{n-1} (I_{N/\mu^{k+1}} \otimes \left[(\bigoplus_{t=0}^{\mu-1} D_{\mu^k}^{(t)}) \cdot (F_\mu \otimes I_{\mu^k}) \right]) =$$

$$= \widetilde{J}_N^{(\mu)} \prod_{k=0}^{n-1} \left[(I_{N/\mu^{k+1}} \otimes (\bigoplus_{t=0}^{\mu-1} D_{\mu^k}^{(t)})) \cdot (I_{N/\mu^{k+1}} \otimes F_\mu \otimes I_{\mu^k}) \right] =$$

$$J_N^{(\mu)} \prod_{k=0}^{n-1} (D_N^{t_k} \cdot F_\mu^{(\phi_k)}), \qquad (3.14)$$

где $\widetilde{J}_N^{(\mu)}$ - матрица μ-ично-инверсной перестановки строк,

$$F_\mu^{(\phi_k)} \overset{\Delta}{=} I_{N/\mu^{k+1}} \otimes F_\mu \otimes I_{\mu^k},$$

$$D_N^{(t_k)} \overset{\Delta}{=} I_{N/\mu^{k+1}} \otimes (\bigoplus_{t=0}^{\mu-1} D_{\mu^k}^{(t)}),$$

$$D_{\mu^k}^{(t)} \overset{\Delta}{=} \bigoplus_{j=0}^{\mu^k-1} \exp(-i\,2\pi t\, j / \mu^{k+1}), \quad t = \overline{0, \mu-1}.$$

На основании уравнения (3.14) можно сделать оценку оптимальности выбора величины μ с точки зрения вычислительной сложности БПФ. Например, если предположить, что число нетривиальных множителей в одной *k*-й матрице мало зависит от величины μ, то дополнительные нетривиальные множители будут определяться алгоритмом БПФ, используемым для матрицы F_μ. Поскольку число матриц определяется $log_\mu N$, величину μ желательно выбирать по возможности большую. Для старших оснований ($\mu > 2$) только для $\mu = 4$ алгоритм БПФ не имеет нетривиальных множителей. Поэтому ясно, что значение $\mu = 4$ и будет оптимальным. Заметим, что для матриц F_μ может быть использован БПФ с любым основанием $\mu_0 \mid \mu$.

В некоторых работах как отдельное подмножество алгоритмов БПФ были выделены так называемые БПФ с расщеплённым основанием (*split radix*). Обращаясь к (3.14), несложно показать, например, что в случае, когда N не является степенью числа μ, и $N = a\mu^n$, где a - целое число либо степень целого числа, можно использовать БПФ со смещённым основанием. Если $a \mid \mu$, то факторизация $N = a\mu^n$ позволяет использовать алгоритм БПФ с расщеплённым основанием. Например, один из вариантов записи факторизации F_N имеет вид

$$F_N = \tilde{J}'_N (I_{N/a} \otimes F_a) \cdot \prod_{k=0}^{\log_\mu (N/a-1)} \left(I_{N/\mu^{k+1} \cdot a} \otimes \left[\left(\bigoplus_{t=0}^{\mu-1} D_{\mu^k \cdot a}^{(t)} \right) \left(F_\mu \otimes I_{\mu^k \cdot a} \right) \right] \right). \quad (3.15)$$

Сигнальный граф, соответствующий факторизации (3.15) для $N = 32 = 4^2 \cdot 2$, когда $\mu = 4$, $a = 2$, приведён на рис. 3.3. Отметим, что при описании алгоритмов БПФ с расщепленным основанием чаще всего приводятся примеры для $N = 32$.

При использовании матрично-рекурсивных и следующих из них факторизованных форм очевидно то, что алгоритмы БПФ с расщепленным основанием являются лишь частным случаем алгоритма (3.15), когда $N = 2^m \neq \mu^n$, $\mu > 2$.

Заметим, что данный быстрый алгоритм имеет число умножений определяемое из выражения $M = N/2 \log_2 N - N/4$, $N \geq 8$.

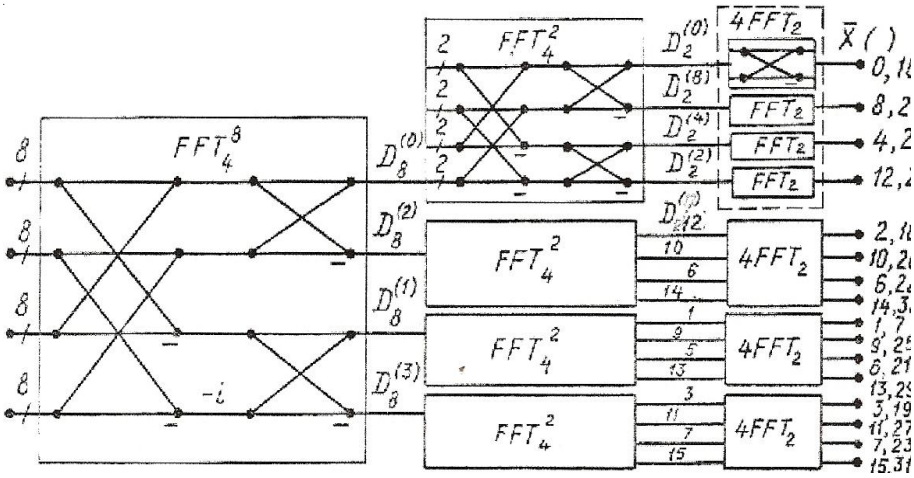


Рис.3.3. Векторный граф БПФ с расщеплённым основанием, N=32

Раздел 4. БЫСТРЫЕ ТЕОРЕТИКО-ЧИСЛОВЫЕ ПРЕОБРАЗОВАНИЯ (ТЧП) МЕРСЕННА ПО ОСНОВАНИЮ ДВА В ПРОСТЫХ ПОЛЯХ

Применение числовых преобразований Мерсенна и Ферма [9] чаще всего связывают с быстрыми алгоритмами вычисления линейных свёрток в цифровой фильтрации, где арифметику с плавающей точкой в поле вещественных чисел можно заменить арифметикой в конечном поле, увеличив при этом скорость вычислений. Высокая точность вычислений в задачах цифровой фильтрации обусловлена значительным влиянием точности представления коэффициентов фильтров на качество их фактических частотных характеристик. Данная проблема является актуальной и для ортонормированных дискретных вейвлет-пребразований, где для представления отсчётов базисных вейвлетов (например, вейвлетов Добеши) требуется высокая точность.

4.1. Способы задания комплексного ТЧП Мерсенна

Учитывая то, что в большинстве практических приложений отсчётами обрабатываемых цифровых сигналов являются вещественные числа, рассмотрим кратко возможности построения быстрых *Ф*-преобразований в полях $GF(2^p-1)$, образованных по модулю простых чисел Мерсенна [6].

Пусть задана матрица прямого (комплексного) преобразования Мерсенна в поле Галуа $GF((2^p-1)^2)$, $p = 2, 3, 5, 7, 13, 17, 19, 31, 61$

$$F_N = \left\| \alpha_N^{k \cdot m} \right\|,$$

где $\alpha_N^{k \cdot m} = a_N^{(km)} + b_N^{(km)} x = a_N^{(km)} + \hat{i}\, b_N^{(km)}$ – первообразный элемент порядка N в поле $GF((2^p-1)^2)$: $k, m = 0, 1, \ldots, N-1$, $a_N^{(km)}$, $b_N^{(km)} \in GF(2^p-1)$.

В дальнейшем рассмотрим матрицы F_N, построенные с помощью первообразных элементов $\alpha_N^1 = a_N^{(1)} + \hat{i}\, b_N^{(1)}$ с периодом $N = 2^{p+1}$, определяемые выражением

$$a_{2^{p+1}} \equiv 2^{2^{p-2}} \bmod(2^p-1), \quad b_{2^{p+1}} \equiv \pm(-3)^{2^{p-2}} \bmod(2^p-1).$$

Степени первообразного элемента α_N^{2m} образуют мультипликативную подгруппу G_N порядка $N = 2^{p+1}/2^{p+1-m}$.

Наиболее часто используемому диапазону представления входных сигналов (от восьми до двенадцати двоичных разрядов) при вычислении свёрток через числовые преобразования соответствует величина p=31. В этом случае имеем для $N_{\max} = 2^{32}, \alpha_{2^{32}} = -10000 - \hat{i}\,1DCD932F$ (здесь и далее значения первообразных корней представлены в шестнадцатиричной системе). Для различных значений m первообразные элементы мультипликативных групп G_N порядка $N = 2^{32-m}$ приведены в таблице 4.1.

Заметим, что здесь всякая подгруппа G_8 представлена элементами типа $\alpha_8^k = \pm c_k 2^{15 m_k} \pm i\hat{}\, d_k 2^{15 m_k}, c_k, d_k, m_k \in \{0,1\}$. Например, можно представить подгруппу

$$\alpha_8 = -2^{15} - i\hat{}\, 2^{15}, \alpha_8^2 = i\hat{},$$
$$\alpha_8^3 = 2^{15} - i\hat{}\, 2^{15}, \alpha_8^4 = -1,$$
$$\alpha_8^5 = 2^{15} + i\hat{}\, 2^{15}, \alpha_8^6 = -i\hat{},$$
$$\alpha_8^7 = -2^{15} + i\hat{}\, 2^{15}, \alpha_8^0 = 1$$

Для обратного комплексного преобразования Мерсенна имеем

$$F_N^{-1} = N^{-1} \left\| (\alpha_N^{-1})^{km} \right\|,$$

где $\alpha_N^{-1} = a_N^{(-1)} + i\hat{}\, b_N^{(-1)}$ - обратный элемент по умножению для элемента α_N в поле $GF((2^p-1)^2)$.

Таблица 4.1

Первообразные элементы мультипликативных групп G_N порядка $N = 2^{32-m}$

N	$a_N^{(1)}$	$b_N^{(1)}$	N	$a_N^{(1)}$	$b_N^{(1)}$
2^3	8000	8000	2^{11}	3*C*0821*E*8	05*C*47*D*82
2^4	5*CC*9971*D*	3*A*542275	2^{12}	6*EE*8448*D*	1*EA*048*B*8
2^5	49*FB*524*B*	46515668	2^{13}	1*E*03*D*28*B*	292*FD*64*C*
2^6	61*DE*0753	18*F*3*DC*5	2^{14}	6757221*C*	52007839
2^7	*C*48366*D*	7356603*A*	2^{15}	5*CC*505*EE*	415*E*694*B*
2^8	*E*12*ACC*7	5*E*064*BDC*	2^{16}	77*E*1*D*46*C*	40*E*16*FA*5
2^9	19*ADD*024	44*B*40*CE*6	2^{17}	1*B*2*AC*490	5*EE*6*D*8*BA*
2^{10}	7*E*4*A*851*A*	6*A*07*F*0*E*7	2^{32}	-10000	-1*DCD*932*F*

4.2. Быстрое ТЧП Мерсенна по основанию два в простом поле

Используя переходную функцию к матрице F_N, можно построить матрицу ортогонального преобразования в простом поле $GF(2^p-1)$, которую представим следующим образом

$$\Phi_N = \left\| a_N^{(km)} + b_N^{(km)} \right\|.$$

Матрица Φ_N^{-1} обратного преобразования Мерсенна над полем $GF(2^p-1)$ должна удовлетворять тождеству $N^{-1}\Phi_N\Phi_N^{-1} = I_N$, где I_N - единичная матрица. В общем случае обратный элемент $N^{-1} \in GF(2^p-1)$ определяется из сравнения $N \cdot N^{-1} \equiv 1 \bmod(2^p-1)$. В нашем случае $N=2^n$, и тогда получаем $N^{-1}= 2^{p-m}$. Здесь $\Phi_N^{-1}= \Phi_N^{'}= \Phi_N$, т. е. матрицы прямого и обратного преобразований совпадают и являются симметрическими.

Имеется очевидное сходство матриц Φ-преобразований над полем $GF(2^p- 1)$ и ДПХ в поле вещественных чисел. Для примера приведём матрицы Ф-преобразования и ДПХ длины $N=8$ (матрицы Φ_8 и H_8).

Заметим при этом, что матрица Φ_8 в качестве нетривиальных множителей имеет лишь 16-ю степень числа 2.

Взаимосвязь матриц Φ_N с матрицами-циркулянтами, через которые вычисляются свёртки (корреляции), по аналогии с ДПХ можно рассмотреть, выразив матрицы F_N в виде

$$F_N = P_N \Phi_N, \quad F_N^{-1} = \Phi_N P_N^{-1};$$

$$P_N = 1 \oplus \left(\oplus_{j=1}^{N-1} \left(\frac{1+i}{2} \right)_j + \left(\oplus_{l=1}^{N-1} \left(\frac{1-i}{2} \right)_l \right) \bar{I}_{N-1} \right), \qquad (4.1)$$

где $\oplus$ - знак прямой суммы.

Пользуясь известным выражением для циркулянта S_N над полем $GF((2^P - 1)^2)$

$$S_N = F_N^{-1} D_N^{(F)} F_N, \text{ где } D_N^{(F)} = diag\{d_j^{(F)}\}_{j=0}^{N-1};$$

$$\bar{d}^{(F)} = F_N \bar{s}_n$$

где $\bar{s}_N$ - транспонированный образующий вектор-строка циркулянта S_N, с помощью (4.1) запишем выражение для циркулянта S_N через матрицы P_N и Φ_N:

$$\Phi_8 = \begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2^{16} & 1 & 0 & -1 & 2^{16} & -1 & 0 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & -2^{16}1 & -1 & 0 & 1 & 2^{16} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 2^{16} & 1 & 0 & -1 & -2^{16} & -1 & 0 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 0 & -1 & 2^{16} & -1 & 0 & 1 & -2^{16} \end{Vmatrix}$$

$$H_8 = \begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\sqrt{2} & 1 & 0 & -1 & \sqrt{2} & -1 & 0 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & -\sqrt{2} & -1 & 0 & 1 & \sqrt{2} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \sqrt{2} & 1 & 0 & -1 & -\sqrt{2} & -1 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & -1 & \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} \end{Vmatrix}$$

$$S_N = \Phi_N D_N^{(\Phi)} \Phi_N; \quad D_N^{(\Phi)} = P_N^{-1} D_N^{(F)} P_N =$$
$$= 1 \oplus \frac{1}{2}\left(\oplus_{k=1}^{N-1}\left(d_k^{(\Phi)} + d^{(\Phi)}{}_{N-k-1}\right) + \left(\oplus_{k=1}^{N-1}\left(d_{N-k-1}^{(\Phi)} - d_k^{(\Phi)}\right)\right)\bar{I}_{N-1}\right).$$

Матрично-рекурсивные и следующие из них матрично-факторизованные формы Φ-преобразования Мерсенна над полем $GF(2^p–1)$ можно получить на основе характеризацинно-инвариантных методов синтеза быстрых ортогональных преобразований [4]. Тогда для быстрого Φ-преобразования по основанию 2 имеем матрично-факторизованную форму

$$\Phi_N = diag\left\{\begin{Vmatrix} 1 & 1 \\ 1 & -1 \end{Vmatrix}\right\} diag\left\{\begin{Vmatrix} I_2 & I_2 \\ D_2^{(\Phi)} & -D_2^{(\Phi)} \end{Vmatrix}\right\} \times \ldots \times \begin{Vmatrix} I_{N/2} & I_{N/2} \\ D_{N/2}^{(\Phi)} & -D_{N/2}^{(\Phi)} \end{Vmatrix} \cdot \tilde{J}_N . \quad (4.2)$$

Граф, соответствующий быстрому Ф-преобразованию в поле $GF(2^{31}—1)$, построенный по матрично-факторизованной форме (4.2) для длины $N = 16$, приведен на рис.4.1, где весовые коэффициенты определены

степенями первообразных элементов в поле $GF((2^{31}—1)^2)$:

$$a_{16}^{(1)} = 5CC9971D,\ b_{16}^{(1)} = -3A542275, a_{16}^{(2)} = a_8^{(1)} = 8000;$$

$$b_8^{(1)} = b_{16}^{(2)} = -8000, a_{16}^{(3)} = 3A542275, b_{16}^{(3)} = -5CC9971D.$$

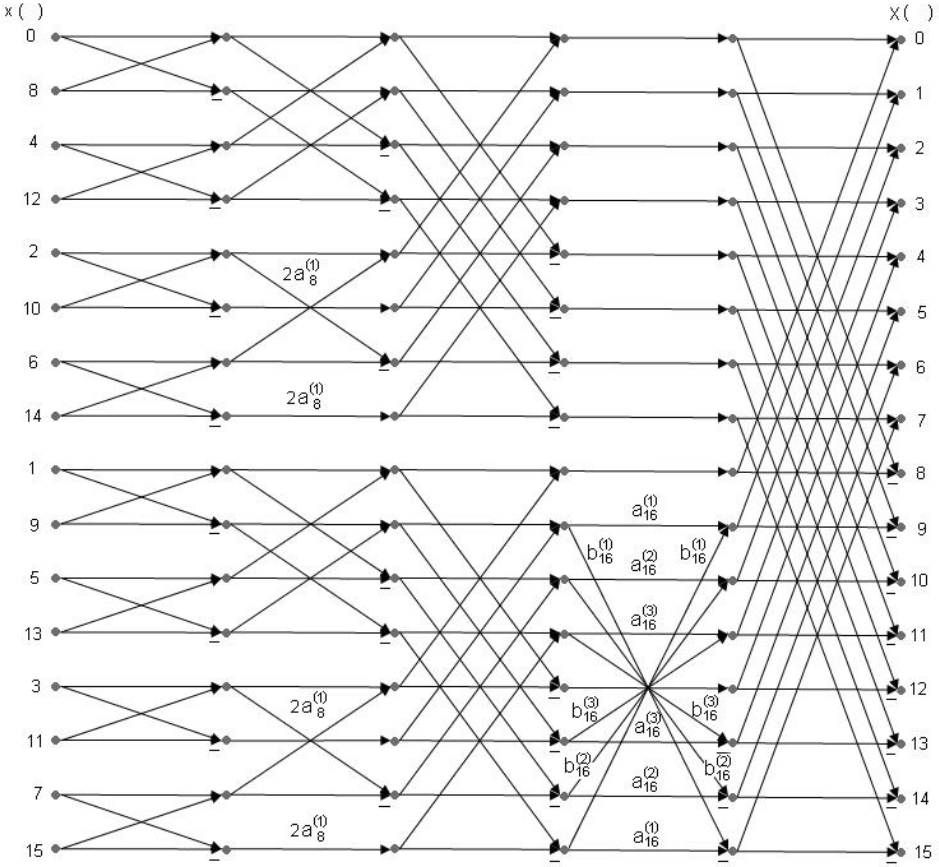


Рис.4.1. Граф, соответствующий быстрому *Ф*-преобразованию в поле $GF(2^{31}\text{-}1)$, N=16

Раздел 5. БЫСТРЫЕ ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ В ПОЛЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ

5.1. Быстрые алгоритмы косинусного преобразования (БКП) с однодиагональными матрицами весовых коэффициентов

В общем виде матрицу дискретного косинусного преобразования можно получить от матрицы комплексного ДПФ в смещённом базисе посредством переходной функции в виде

$$T_N = \mathrm{Re}(F_{2N}^{(0,1/2)}) = \mathrm{Re}[\exp(-i2\pi\, k\,(m+1/2)/2N]\,,\ \ k,m = \overline{0,N-1}\,.$$

В среде Mathcad целесообразнее использовать ортонормированные базисы, т.е. матрицу преобразования домножают на диагональную матрицу нормирующих коэффициентов, как показано в приведённом примере

$m := 0..7$

$f_{m,k} := e^{-i\cdot 2\cdot\pi\cdot(m)\cdot\frac{k+.5}{16}}$

$T8 := \mathrm{Re}(f)$

$d8 := \begin{pmatrix} .354 \\ .5 \\ .5 \\ .5 \\ .5 \\ .5 \\ .5 \\ .5 \end{pmatrix}$

$D8 := \mathrm{diag}(d8)$

$T8n := D8 \cdot T8$

$k := 0..7$

matrix(m,k,f)

$\sqrt{\frac{1}{8}}$ $\sqrt{\frac{2}{8}}$

норм. к.

Алгоритмы БКП с однодиагональными матрицами весовых множителей обладают важным преимуществом при их реализации на конвейерных архитектурах вычислительных устройств. Воспользуемся для БКП матрично-факторизованной формой, которую можно было бы использо-вать также для получения псевдогнездовых алгоритмов двумерного ДКП [10].

Представим матрицу ДКП в виде:

$$T_N = \left(1/\sqrt{2} \oplus I_{N-1}\right)\cdot Q_N\,, \qquad (5.1)$$

где $Q_N = \left\| \cos\frac{m\ (2k+1)\,\pi}{2N} \right\|, m,\kappa = \overline{0,N-1}, N = 2^e.$

Матрицу обратного ДКП можно получить, транспонируя форму (5.1)

$$T_N^{-1} = T_N' = Q_N'(1/\sqrt{2} \oplus I_{N-1}). \tag{5.2}$$

Выразим Q_N в виде произведения двух матриц

$$Q_N' = \hat{Q}_N \cdot \hat{P}_N, \tag{5.3}$$

где $\hat{P}_N$ и $\hat{Q}_N$ в свою очередь также факторизованы. Для матрицы P_N имеем

$$\hat{P}_N = diag\{P_4\} \cdot diag\{P_8\} \times \ldots \times diag\{P_{N/2}\} \cdot P_N = \hat{P}_N^{(0)} \cdot \hat{P}_N^{(1)} \times \ldots \times \hat{P}_N^{(l-2)},$$
$$l = \log_2 N \tag{5.4}$$

P_N – слабозаполненные матрицы с элементами 0 и 1, определяемые выражением

$$P_N = \begin{vmatrix} \oplus_{\nu=0}^{N/2-1} (10)_\nu \\ \hline 0\left(\left(\oplus_{\nu=0}^{N/2-2} (10)_\nu\right) \oplus 1\right) + \left(0 \oplus \left(\oplus_{\nu=0}^{N/2-2} (10)_\nu\right) 0\right) \end{vmatrix},$$

$$P_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}, P_2 = I_2.$$

Для матрицы $\hat{Q}_N$ имеем факторизацию очень сходную с факторизациями матриц ДПФ:

$$\hat{Q}_N = \begin{pmatrix} I_{N/2} & D_{N/2} \\ \bar{I}_{N/2} & -\bar{D}_{N/2} \end{pmatrix} \cdot diag\left\{\begin{pmatrix} I_{N/4} & D_{N/4} \\ \bar{I}_{N/4} & -\bar{D}_{N/4} \end{pmatrix}\right\} \times \ldots$$
$$\times diag\left\{\begin{pmatrix} 1 & D_1 \\ 1 & -\bar{D}_1 \end{pmatrix}\right\} = \hat{Q}_N^{(0)} \cdot \hat{Q}_N^{(1)} \times \ldots \times \hat{Q}^{(l-1)},$$

где $D_n = diag\left\{\left(2\cos\frac{(2j+1)\cdot\pi}{4n}\right)^{-1}\right\}_{j=0}^{n-1}$,

$\overline{D}_n = \bar{I}\cdot D_n$ – матрица с инверсно переставленными строками.

Подставляя в выражение (5.2) факторизированные формы (5.3), (5.4) и (5.5), получим для матрицы T_N^{-1} следующее выражение, соответствующее быстрому алгоритму ОДКП:

$$
\begin{aligned}
&T_N^{-1} = \begin{pmatrix} I_{N/2} & D_{N/2} \\ \bar{I}_{N/2} & -\overline{D}_{N/2} \end{pmatrix} \cdot diag\left\{\begin{pmatrix} I_{N/4} & D_{N/4} \\ \bar{I}_{N/4} & -\overline{D}_{N/4} \end{pmatrix}\right\} \times \ldots \\
&\times diag\left\{\begin{pmatrix} 1 & D_1 \\ 1 & -\overline{D}_1 \end{pmatrix}\right\} \times diag\{P_4\} \times \ldots \times diag\{P_{N/4}\} \times \\
&\times P_N\left(1/\sqrt{2} \oplus I_{N-1}\right) = \widehat{Q}_N^{(l-1)} \times \ldots \times \widehat{Q}_N^{(1)} \cdot \widehat{Q}_N^{(0)} \cdot \widehat{P}_N^{(0)} \times \ldots \times \widehat{P}_N^{(l-2)} \times \\
&\times \left(1/\sqrt{2} \oplus I_{N-1}\right) = V_N^{(l-1)} \cdot D_N^{(l-1)} \times \ldots \times V_N^{(1)} \cdot D_N^{(1)} \cdot V_N^{(0)} \cdot D_N^{(0)} \cdot \widehat{P}_N^{(0)} \times \ldots \\
&\times \widehat{P}_N^{(l-2)}\left(1/\sqrt{2} \oplus I_{N-1}\right),
\end{aligned}
\tag{5.6}
$$

где $\hat{Q}_N^{(j)} \triangleq diag\left\{\begin{pmatrix} I_{2^j} & D_{2^j} \\ \bar{I}_{2^j} & -\overline{D}_{2^j} \end{pmatrix}\right\}$, $\hat{V}_N^{(j)} \triangleq diag\left\{\begin{pmatrix} I_{2^j} & I_{2^j} \\ \bar{I}_{2^j} & -\bar{I}_{2^j} \end{pmatrix}\right\}$,

$D_N^{(j)} \triangleq diag\left\{I_{2^j} \oplus D_{2^j}\right\}$.

Тогда факторизованная форма матрицы T_N может быть получена транспонирование формы (5.6)

$$
\begin{aligned}
&T_N = (1/2 \oplus I_{N-1}) \times P_N^{'} \cdot diag\{P_{N/2}^{'}\} \times \ldots \times \{P_4^{'}\} \cdot diag\left\{\begin{pmatrix} 1 & D_1 \\ 1 & -\overline{D}_1 \end{pmatrix}\right\} \times \ldots \\
&\times diag\left\{\begin{pmatrix} I_{N/4} & \bar{I}_{N/4} \\ D_{N/4}^{'} & -\overline{D}_{N/4}^{'} \end{pmatrix} \cdot \begin{pmatrix} I_{N/2} & \bar{I}_{N/2} \\ D_{N/2}^{'} & -\overline{D}_{N/2}^{'} \end{pmatrix}\right\} = (1/2 \oplus I_{N-1}) \cdot \left(\widehat{P}_N^{(l-2)}\right)^{'} \times \ldots \\
&\times \left(\widehat{P}_N^{(0)}\right)^{'} \times \left(D_N^{(0)}\right)^{'} \cdot \left(V_N^{(0)}\right)^{'} \times \ldots \times \left(D_N^{(l-2)}\right)^{'} \cdot \left(V_N^{(l-2)}\right)^{'}.
\end{aligned}
\tag{5.7}
$$

На рис. 5.1. представлен граф БКП для N =8.

Для построения быстрого двумерного БКП воспользуемся кронекеровсеой факторизацией, в виде которой может быть задана матрица двумерного преобразования, выраженная через матрицы одномерного преобразования

$$T_{NN} = T_N \otimes T_{N.}.$$

Подставляя в нее выражение (5.7) для БКП, получим:

$$T_{NN} = (1/\sqrt{2} \oplus I_{N-1})^{[2]} \cdot (P_N^{'})^{[2]} \times ... \times (diag\{P_4^{'}\})^{[2]} \times$$
$$\times \left(diag\left\{ \begin{pmatrix} 1 & 1 \\ D_1 & -\overline{D}_1 \end{pmatrix} \right\} \right)^{[2]} \times ... \times \left(diag\left\{ \begin{pmatrix} I_{N/4} & \bar{I}_{N/4} \\ D_{N/4} & -\overline{D}_{N/4} \end{pmatrix} \right\} \right)^{[2]} \cdot$$
$$\cdot \begin{pmatrix} I_{N/2} & \bar{I}_{N/2} \\ D_{N/2}^{'} & -\overline{D}_{N/2}^{'} \end{pmatrix} = (1/\sqrt{2} \oplus I_{N-1})^{[2]} \cdot \left(\left(\widehat{P}_N^{(l-2)} \right)' \right)^{[2]} \times ... \qquad (5.8)$$
$$\times \left(\left(\widehat{P}_N^{(0)} \right)' \right)^{[2]} \times \left(\left(D_N^{(0)} \right)' \right)^{[2]} \cdot \left(\left(V_N^{(0)} \right)' \right)^{[2]} \times ... \times \left(\left(D_N^{(l-1)} \right)' \right)^{[2]} \cdot \left(\left(V_N^{(l-1)} \right)' \right)^{[2]}.$$

$$\left(\left(V_N^{(j)} \right)' \right)^{[2]} = \left(diag\left\{ \begin{pmatrix} I_{2^j} & \bar{I}_{2^j} \\ I_{2^j} & -\bar{I}_{2^j} \end{pmatrix} \right\} \right)^{[2]},$$

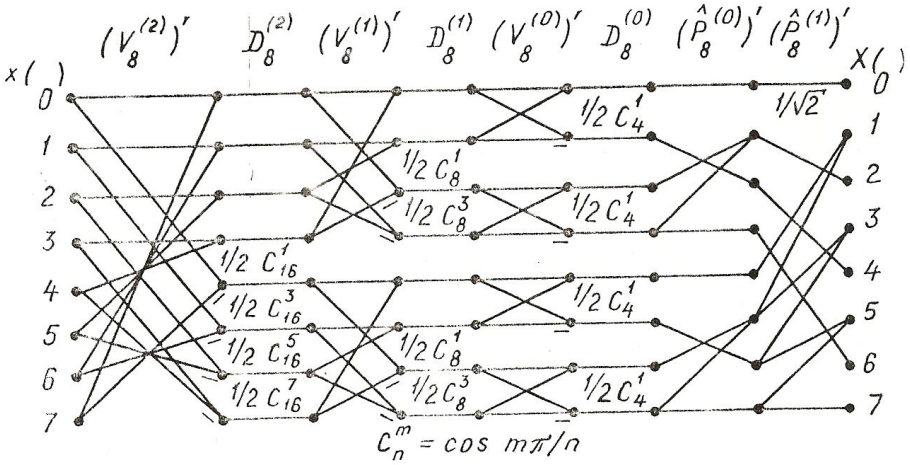
где $$\left(\left(D_N^{(j)} \right)' \right)^{[2]} = \left(diag\left\{ \left(I_{2^j} \oplus D_{2^j} \right) \right\} \right)^{[2]}, j = 0, l-1.$$

Алгоритму обратного двумерного БКП будет соответствовать форма

$$T_{NN}^{-1} = T_N^{-1} \otimes T_N^{-1} = \left(V_N^{(l-1)} \right)^{[2]} \cdot \left(D_N^{(l-1)} \right)^{[2]} \times ...$$
$$\times \left(V_N^{(0)} \right)^{[2]} \cdot \left(D_N^{(0)} \right)^{[2]} \times \left(P_N^{(0)} \right)^{[2]} \times ... \times \left(P_N^{(0)} \right)^{[2]} \cdot \left(1/\sqrt{2} \oplus I_{N-1} \right)^{[2]}, \qquad (5.9)$$

$$\left(V_N^{(j)} \right)^{[2]} = \left(diag\left\{ \begin{pmatrix} I_{2^j} & I_{2^j} \\ I_{2^j} & I_{2^j} \end{pmatrix} \right\} \right)^{[2]},$$

где $$\left(D_N^{(j)} \right)^{[2]} = \left(diag\left\{ \left(I_{2^j} \oplus D_{2^j} \right) \right\} \right)^{[2]}, j = \overline{0, l-1}.$$



5.1. Граф БКП с однодиагональной матрицей весовых множителей, N=8

В таблице 5.1 приведены оценки вычислительной сложности быстрых алгоритмов двумерных псевдогнездовых и построчно-столбцовых БКП для случая использования в качестве одномерных БКП алгоритмов (5.6) и (5.7). Граф двумерного псевдогнездового 8 × 8 БКП представлен на рис. 5.2.

Несложно заметить, что формы (5.8) и (5.9) обобщаются на случай L – мерного БКП посредством введения в произведения кроне-керовских L- степеней.

Таблица 5.1

Оценки количества арифметических операций быстрых алгоритмов

Длина *N ЧN* преобразования (БКП)	Псевдогнездовой		Построчно-столбцовый	
	M (вещ.ум.)	*A* (вещ.сл.)	*M*	*A*
8 Ч8	142	464	208	464
16 Ч16	734	2592	1056	2592
32 Ч32	3646	13376	5120	13376
64 Ч64	17534	65664	25402	65664

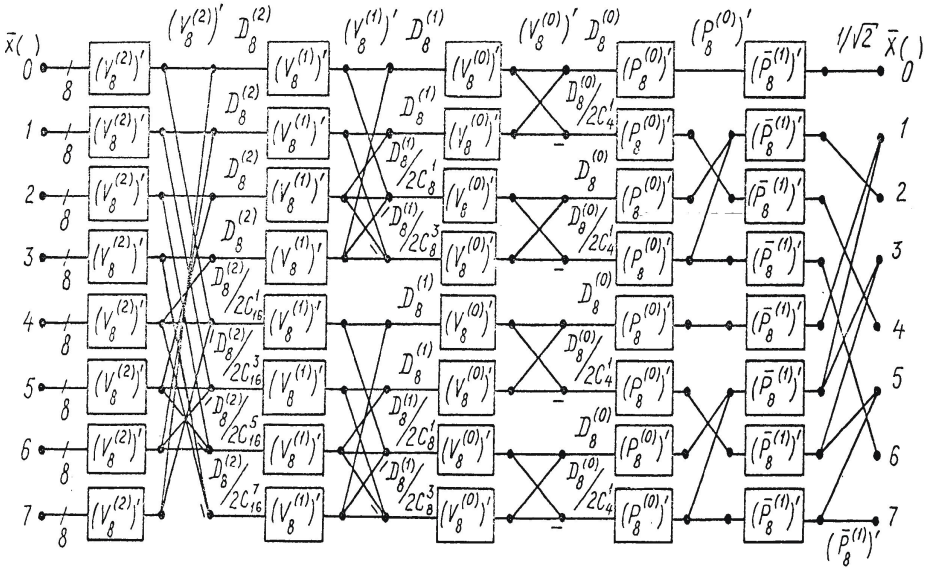


Рис. 5.2. Граф двумерного псевдогнездового быстрого ДКП, N2 = 8×8

В отличие от алгоритмов БКП с двудиагональными матрицами, алгоритмы (5.8) и (5.9) согласно таблице 5.1 позволяют сократить по сравнению с построчно-столбцовыми число нетривиальных умножений примерно на 25% при одинаковой сложности управления данными.

5.2. Способы задания и примеры применения ортонормированных базисов дискретных вейвлет-преобразований

5.2.1. *Общие блочно-матричные формы задания дискретных вейвлет-преобразований в ортонормированных базисах*

Матрицу прямого дискретного вейвлет преобразования (ВП) будем задавать общепринятой двух блоковой формой

$$\Psi_N = \begin{pmatrix} \Psi^{(1)} \\ \Psi^{(2)} \end{pmatrix}, \qquad (5.10)$$

где $\Psi^{(1)}$ – прямоугольная блок-матрица, соответствующая низкочастотной части ВП коэффициентов ,

$\Psi^{(2)}$ – прямоугольная блок-матрица, соответствующая высокочастотной части ВП коэффициентов .

Будем полагать, что строки матрицы образуют ортонормированное векторное пространство, т.е. функция скалярного произведения векторов удовлетворяет условию

$$\left(\bar{h}_i, \bar{h}_j\right) = \begin{vmatrix} 1, & \textit{если } i = j, \\ 0, & \textit{если } i \neq j. \end{vmatrix}$$

Матрица обратного ВП над полем вещественных чисел принимает вид

$$\Psi_N^{-1} = \Psi_N' = \left(\left(\Psi^{(1)}\right)' \mid \left(\Psi^{(2)}\right)' \right).$$

Исходя из матричной формы (5.10), коэффициенты одномерного ВП определяются в блочно-векторном виде

$$X_N = \begin{pmatrix} X_{N/2}^{(1)} \\ \hline X_{N/2}^{(2)} \end{pmatrix} = \Psi_N \cdot x_N,$$

где x_N– вектор, соответствующий последовательности отсчётов цифрового сигнала. Коэффициенты двумерного ВП можно определить двумя способами, используемыми, как ранее было показано в общей теории синтеза быстрых ортогональных преобразований. Тогда построчно-столбцовый способ для двумерного ВП записывается матричным выражением

$$[X_N] = \Psi_N \cdot [x_N] \cdot \Psi_N', \qquad (5.11)$$

где $[X_N]$ и $[x_N]$ – матрицы соответственно коэффициентов двумерного ВП и отсчётов сигнала.

Второй способ вычисления коэффициентов двумерного, а в общем случае многомерного преобразования, сводится к построению матрицы, представленной в виде кронекеровского произведения матриц одномерного преобразования, например, используемых в выражении (5.11). Тогда для двумерного ВП можно записать

$$\widehat{\Psi}_{N^2} = \Psi_N \otimes \Psi_N, \qquad (5.12)$$

и коэффициенты ВП определяются

$$X_{N^2} = \left(\Psi_N \otimes \Psi_N\right) \cdot \bar{x}_{N^2},$$

где X_{N^2} - блоковый вектор-столбец, образованный транспонированными строками матрицы коэффициентов двумерного преобразования, используемой в выражении (5.11),

$\overline{x}_{N^2}$ - блоковый вектор-столбец, образованный транспонированными строками матрицы двумерного сигнала $[x_N]$.

5.2.2. *Вейвлет-преобразования Хаара*

В соответствии с блоковой формой (5.10) матрицу ВП Хаара можно записать в виде

$$\Psi_N = \begin{pmatrix} \Psi^{(1)}_{N/2} \\ \Psi^{(2)}_{N/2} \end{pmatrix} = \begin{pmatrix} diag\left(h_0^{(1)}, h_1^{(1)}\right) \\ diag\left(h_0^{(2)}, -h_1^{(2)}\right) \end{pmatrix}, \qquad (5.13)$$

где оператор *diag*(*x*) соответствует оператору прямой суммы матриц *x*.

Матричную форму (5.13) можно обобщить на *L*- длину базового вейвлета (длину анализирующей части ВП) и на *K*- количество матриц-блоков.

Однако для практических приложений ВП с блочно-диагональной формой матриц из-за отсутствия при сдвигах перекрытий в цифровом сигнале целесообразно использовать короткие ортонормированные базисные анализирующие функции.

В случае использования многоблоковых матричных форм в представлении ВП могут быть обобщены с подполосным кодированием.

Для сокращения количества арифметических операций при вычислении коэффициентов в матрицу ВП вводится нормирующий коффициент так, что матрица принимает вид

$$\Psi_N = K_N \begin{pmatrix} \widehat{\Psi}^{(1)}_{N/2} \\ \widehat{\Psi}^{(2)}_{N/2} \end{pmatrix}.$$

5.2.3. *Блочно-циклическая матричная форма задания ВП*

Типичными представителями ВП, которые можно задать в блочно-векторной циклической матричной форме, являются добешиподобные ВП. Общий вид таких ортонормированных матриц для длины базисного вейвлета *l*=6 и с величиной его смещения *t* = 2 можно записать

$$\Psi_8 = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \\ h_4 & h_5 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & h_4 & h_5 & 0 & 0 & h_0 & h_1 \\ h_5 & -h_4 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & h_5 & -h_4 & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & 0 & 0 & h_5 & -h_4 & h_3 & -h_2 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & h_5 & -h_4 \end{pmatrix} \quad (5.14)$$

Из матричной формы (5.14) видны следующие ограничения:

- длина ВП всегда является чётным числом;

- образующий вектор нижней блок-матрицы представляет собой инвертированную последовательность компонентов базового вейвлета с через один изменённым знаком.

Для длин $l = 4$ и $l = 6$ можно построить практически неограниченное множество вейвлетов , удовлетворяющих условию ортонормированности матрицы Ψ_N. Однако вейвлеты Добеши обладают так называемыми свойствами «исключения моментов» [12], которые сводятся, например для $l = 4$, к соблюдению условия равенства нулю скалярного произведения векторов

$$(\bar{h}, \bar{x}) = \sum_{n=0}^{3} h_{3-n} \cdot x_n = 0$$

1) при $x_0 = x_1 = x_2 = x^3$, (5.15)
2) при $x_1 = 2\,x_0$, $x_2 = 3\,x_0$, $x_3 = 4\,x_0$,

где $\bar{h}$ - образующий вейвлет-вектор, $\bar{x}$ - вектор отсчётов анализируемого фрагмента цифрового сигнала. В матрице Ψ_N данные условия должны быть отражены в образующем векторе нижней блок-матрицы так, что максимальные значения верхней половины коэффициентов ВП будут соответствовать одинаковым, либо плавно-линейно изменяемым значениям отсчётов анализируемого фрагмента сигнала.

Условие (5.15) значительно уменьшает возможности синтезирования новых аналогичных ВП, в том числе наиболее предпочтительных для программно-аппаратной реализации.

Заметим, что в отличие от ортонормированных двумерных ВП Хаара, где элементами матрицы Ψ_N являются 1 и -1 с нормирующим коэффициентом равным 1/2, в матрицах ВП Добеши элементами являются иррациональные числа, для выполнения операций над которыми в многоитерационных технологиях сжатия потребуется арифметика с плавающей точкой. Кроме того, количество самих операций многократно больше ВП Хаара.

5.2.4. *Аппроксимированные вейвлеты Добеши*

С целью сокращения вычислительных затрат можно использовать вейвлеты, аппроксимированные к вейвлетам Добеши [13]. Например, для длин $l = 4$ и $l = 6$ можно использовать вейвлеты, элементами которых являются преимущественно степени числа два

$AD4 = 1\ ,\ 2,\ 0.5,\ -0.25;\quad AD6 = 0.5\ ,\ 0.31,\ -0.062,\ -0.125,\ 0.98\ ,\ 0.02$

вместо соответствующих вейвлетов Добеши

$D4 = 0.48296\ ,\ 0.836516\ ,\ 0.224143\ ,\ -0.129409;$

$D6 = 0.33267\ ,\ 0.80689\ ,\ 0.459877\ ,\ -0.135011\ ,\ -0.085441,\ 0.035226.$

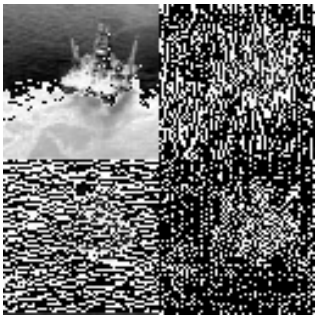
Ниже приведены примеры программных средств формирования в Mathcad ортонормированных матриц Q и Qa, построенных на основе соответственно вейвлетов $D4$ и $AD4$ с экспериментальными данными по их применению для сжатия изображений (аэрофотоснимок разлива нефти в Мексиканском заливе).

$$q2 := \begin{pmatrix} .483 \\ .836 \\ .224 \\ -.129 \end{pmatrix} \qquad q3 := \begin{pmatrix} -.129 \\ -.224 \\ .836 \\ -.483 \end{pmatrix} \qquad q := \begin{pmatrix} 1 \\ 1.875 \\ .5 \\ -.25 \end{pmatrix} \qquad q1 := \begin{pmatrix} -.25 \\ -.5 \\ 1.875 \\ -1 \end{pmatrix}$$

```
Qa :=  for k ∈ 0..3
         for i ∈ 0..63
           a[i, mod(k+2·i, 128)] ← q[k] / √4.82
         for i ∈ 64..127
           a[i, mod(k+2·i, 128)] ← q1[k] / √4.82
       a
```

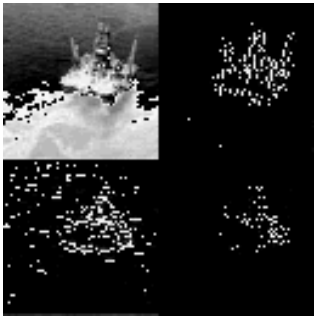
```
Q :=  ( for k ∈ 0..3
          for i ∈ 0..63
            a[i, mod(k+2·i, 128)] ← q2[k]
          for i ∈ 64..127
            a[i, mod(k+2·i, 128)] ← q3[k]
        a )
```

T1 Рис.5.3. Исходное изображение



Y Рис.5.4. Изображение в области ВП A*D*4

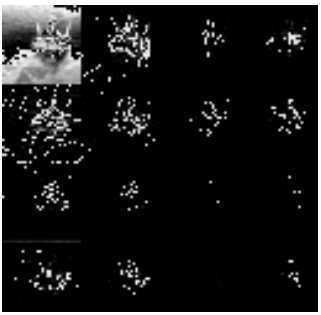
(T20 + 128)



Y3

Рис. 5.5. Восстановленное (слева) и сжатое (справа) изображения после первой итерации

(T40 + 128)



Y41

Рис. 5.6. Восстановленное (слева) и сжатое (справа) изображения после второй итерации

Раздел 6. КОРРЕЛЯЦИОННАЯ И СПЕКТРАЛЬНАЯ ОБРАБОТКА ЦИФРОВЫХ РЕЧЕВЫХ СИГНАЛОВ

6.1. Скалярно-разностное кодирование

Разностное кодирование относится к методам сжатия цифровых сигналов через кодирование их формы. Наиболее широкое практические применение получили скалярно-разностные алгоритмы кодирования [14], которые строятся на основе скалярно-разностных уравнений

$$d(n) = x(n) - \tilde{x}(n), \tag{6.1}$$

где $x(n)$ – n-й отсчет кодирования сигнала, $\tilde{x}(n)$ – предсказанное значение для $x(n)$, $d(n)$ – погрешность предсказания.

На основе разностного уравнения (6.1) разработаны алгоритмы дельта-модуляции (двухуровневое квантование) и дифференциальной импульсно-кодовой модуляции (многоуровневое квантование).

Применительно к уравнению (6.1) получили распространение алгоритмы сжатия с линейным предсказанием, где предсказанное $\tilde{x}(n)$ значение для каждого кодируемого отсчёта сигнала $x(n)$ формируется не по одному, а по последовательности предшествующих ему отсчётов. Математическую модель такого предсказателя можно записать в виде линейной свёртки

$$\tilde{x}(n) = \sum_{k=1}^{p} a(k)\hat{x}(n-k),$$

где $\hat{x}(n)$ – квантованное значение n-го отсчета, $a(k)$ – коэффициенты предсказания (коэффициенты линейного фильтра).

Наиболее простым способом разностного кодирования является дельта – модуляция, сокращенно *DM* – (*Delta Modulation*). Метод был разработан еще в 40 – х годах 20 века для использования в телефонии. В системах такого типа частота дискретизации выбирается во много раз больше, чем частота Найквиста. В результате соседние отсчёты оказываются в большей степени коррелированными.

Большая корреляция между отсчётами означает, что при уменьшении периода T можно более точно предсказать текущий отсчёт по предшествующим и, следовательно, уменьшить дисперсию погрешности предсказания. Поэтом более «грубый» квантователь может дать хорошие результаты. Схема простейшей системы *DM* приведена на рис. 6.1.

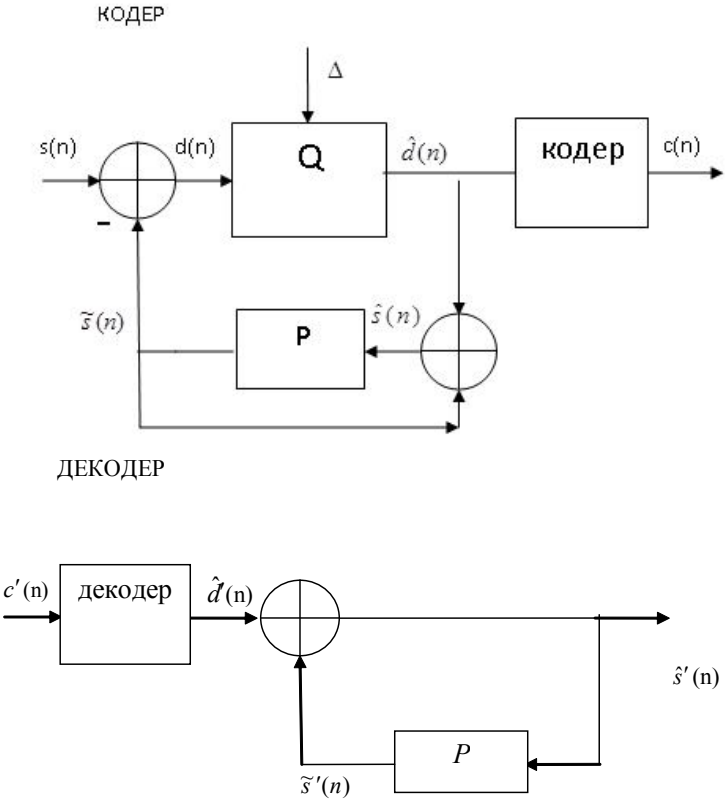


Рис. 6.1 Структурная схема *DM* кодера и декодера

Суть такого кодирования заключается в следующем – на вход одноразрядного квантователя на n – ом шаге преобразования подаётся сигнал ошибки предсказания

$$d(n) = s(n) - \tilde{s}(n),$$

где $s(n)$ – измеренное значение сигнала на шаге дискретизации n, а величина $\tilde{s}(n)$ определяется разностным уравнением

$$\tilde{s}(n) = \sum_{j=1}^{k} a_j \hat{s}(n-j) = a_1\,\hat{s}(n-1) + a_2\,\hat{s}(n-2) + \ldots.. + a_k \hat{s}(n-k)$$

где k - порядок предсказателя,

n = 0, 1, 2, 3… - номер текущего шага дискретизации,

$\tilde{s}(n-j)$ - численное значение итогового импульса сигнала на предшествующем (n - j) шаге преобразования j = 1, 2,..., k,

a_k – коэффициент предсказания на предыдущем шаге.

При $k = 1$, $a_1 = 1$, $d(n) = s(n) - \tilde{s}(n-1)$.

Одноразрядный квантователь имеет только два уровня квантования «0» и «1», при этом шаг квантования Δ между этими уровнями является постоянным на каждом шаге дискретизации (поэтому такую ДМ называют линейной ДМ). Квантователь, квантуя поданный на вход сигнал $d(n)$ выводит «1», т.е. $\hat{d}(n) = \Delta$, если значение сигнала $s(n)$ на n – ом шаге оказалось равным нулю; и «0», т.е. $\hat{d}(n) = -\Delta$, если $s(n) = 1$.

Скалярно-разностные уравнения нашли применение в алгоритмах с кратко- и долгосрочным (двухитерационным) предсказанием, получившие известность в виде *Code Excited Linear Prediction* (*CELP*)-алгоритмов [15],. К схеме такого разностного кодирования (рис.6.2) помимо уравнений (6.1) добавляются разностные уравнения второго порядка

$$e(n) = d(n) - \tilde{d}(n)$$

$$\tilde{d}(n) = \sum_{k=1}^{P_1} b(n)\,\hat{d}(n-k)$$

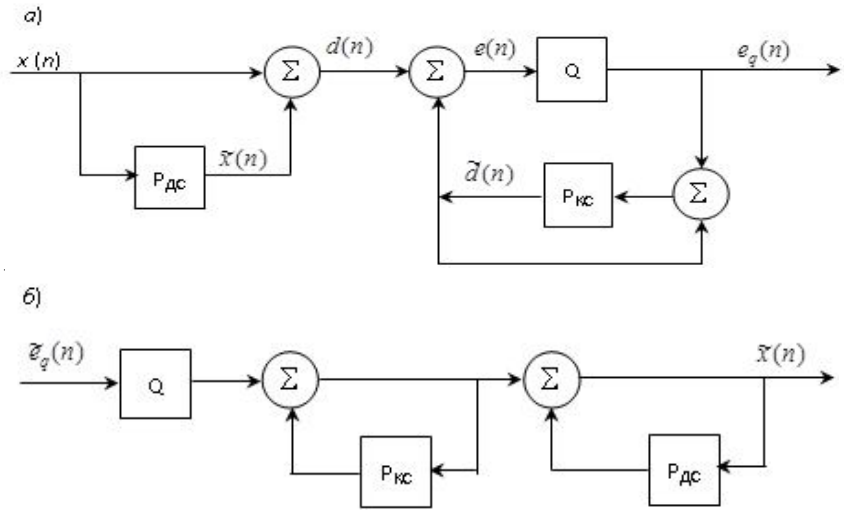


Рис.6.2. Структурные схемы разностного кодирования: *а*) – кодера; *б*) – декодера с кратко- и долгосрочным предсказанием

В работе [15] приведены сравнительные качественные характеристики различных методов компрессии аудиосигналов, из которых следует, что методы кодирования с кратко и долгосрочным скалярным кодированием и с векторным квантованием обладают предпочтениями в практических применениях.

Векторное квантование [15] относится также к кодированию формы сигнала. В отличие от скалярно-разностных алгоритмов квантуются и кодируются не отдельные отсчёты, а векторы (последовательность) отсчётов. Кодами в этом случае выступают номера (признаки) наиболее вероятной принадлежности вектора тому или иному классу кодовой книги, которая имеется на передающей и на приёмной стороне и может изменяться в процессе кодирования.

6.2. Векторно-разностное кодирование

В работе [16] было предложено использовать для кодирования формы речевых сигналов векторно-разностные (ВР) уравнения

$$\overline{d}(m) = X(m) - \widetilde{X}(m), \tag{6.3}$$

где $\overline{d}(m)$, $X(m)$ и $\overline{X}(m)$ – векторы соответственно погрешностей предсказания, кодируемых отсчётов сигнала $\{x(n)\}$ и предсказанных значений $\{\widetilde{x}(n)\}$. Схемы кодера и декодера ВР-кодирования (рис. 6.3) отличаются от скалярного разностных схем наличием $\Phi_{\text{В}}$ - блока формирования векторов в кодере и $\Phi_{\text{П}}$ - блока преобразования векторной формы данных в последовательностную. Кроме того, операции сложения-вычитания и квантования выполняются над векторами, т. е. над каждым элементом вектора отдельно и независимо друг от друга. Такой способ кодирования относится к кодированию формы сигналов, образованных одноименными $x_i^{(k)}$ компонентами последовательности векторов $\{X(k)\}$.

Для случая, когда длина вектора определяется величиной периода основного тона речевого сигнала вектора на рис.6.3 приведены графики сигналов $\{x(k)\}$ и $\{x(n)\}$. Приведенные примеры форм сигналов x_i и x_n указывают на высокую степень корреляции между соседними отсчётами как внутри векторов, так и между векторами.

Поэтому, как практический, так и теоретический интерес представляют алгоритмы ВР-кодирования, позволяющие учитывать оба вида корреляции отсчётов сигнала.

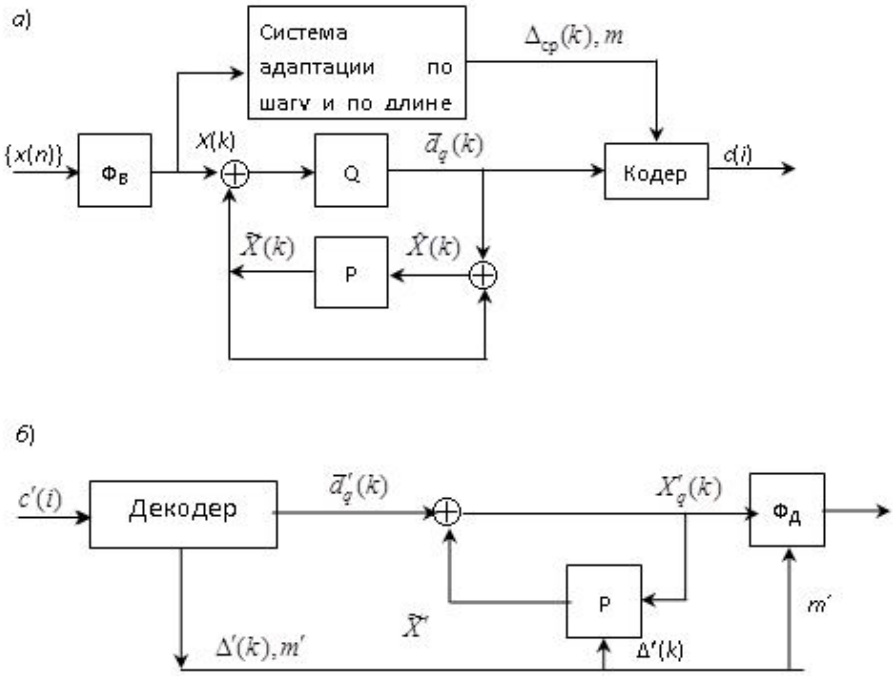


Рис.6.3. Схема ВР-кодирования с адаптацией по шагу квантования и по длине вектора: *а*) кодер, *б*) декодер

В работе [16] были предложены математические модели предсказателей в схемах ВР-кодирования в виде многоканальных линейных систем.

Рассмотрим более подробно методы ВР-кодирования с учётом особенностей используемых математических моделей.

При независимой межканальной обработке отсчётов сигнала квантование и кодирование осуществляется для каждого *i*-го сигнала, по которому передаётся сигнал $x_i(k)$. При этом, согласно векторно-разностному уравнению (6.4) предсказанные значения $\{\tilde{x}_i(k)\}$ формируются независимо друг от друга.

Для определения вектора $X_i(n)$ предсказанных значений кодируемого вектора $X(k)$ можно записать векторно-матричное разностное уравнение:

$$\tilde{X}_m(k) = \sum_{l=1}^{p} D_m(l)\,\hat{X}(k-l)$$

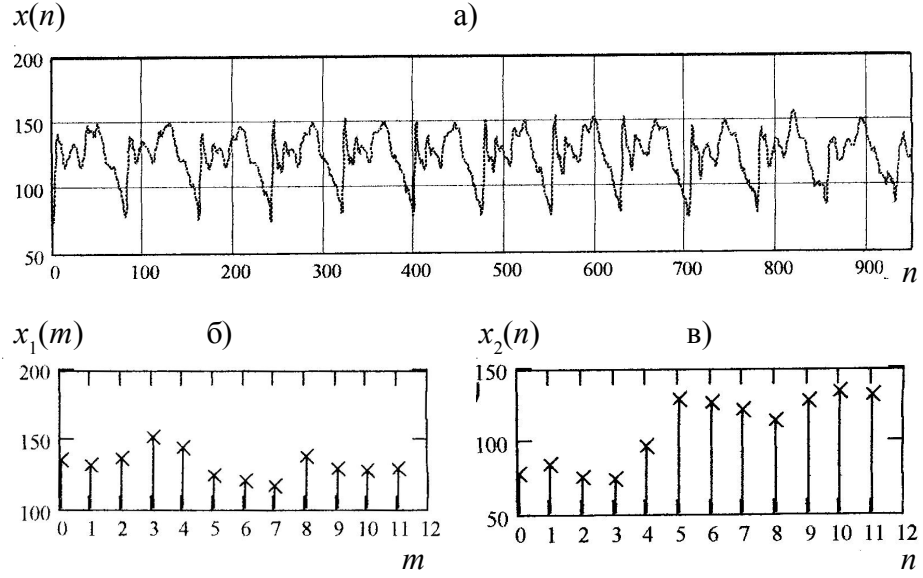


Рис.6. 4. Графики сигналов: *а)* $x(n), n = \overline{0, 900}$, *б)* $x_1(n), n = \overline{0, 13}$, *в)* $x_2(k), k = \overline{1, 12}$

где $D_m(l) = diag\{a_i(l)\}_{i=0}^{m-1}$ – диагональная матрица коэффициентов предсказания; $\widehat{X}(n-l)$ – вектор квантованных значений кодируемого сигнала. Для $p = 1$ выражение может быть записано в виде $\widetilde{X}_m(k) = D_m(1)\widehat{X}(k-1)$, где $D_m(1) = I_m$, т. е. может быть равно единичной матрице.

Для *i*-го сигнала $\{x_i(k)\}$ являются справедливыми известные соотношения [14], определяющие качество кодирования. Например, отношение сигнал/шум (с/ш) можно записать через отношение дисперсии погрешности $\sigma_\alpha^2(i)$ предсказания к дисперсии погрешности квантования $\sigma_e^2(i)$

$$(\text{с/ш})_i = \sigma_{\alpha_i}^2 / \sigma_{e_i}^2 .$$

Для коэффициентов усиления в *i*-ом канале имеем

$$G_{ri} = \sigma_{x_i}^2 / \sigma_{d_i}^2 .$$

Таким образом, ВР-кодирование с независимой межканальной обработкой может быть осуществлено с использованием в каждом из каналов скалярно-разностных алгоритмов, в том числе алгоритмов с линейным предсказанием.

При этом для определения оптимальных коэффициентов предсказания представленных матрицей $D_m^{(v)}(l) = diag\left\{\alpha_v^{(i)}(l)\right\}_{i=1}^{m}$ можно, например, вос-

пользоваться известным соотношением, связывающим коэффициенты предсказания на *n*-м интервале времени с кратковременной автокорреляционной функцией

$$R_v^2(j) = \sum_{l=1}^{p} \alpha_v^{(i)}(l) R_l(j-l), \qquad j = \overline{1, p},$$

которое может быть представлено в векторно-матричной форме

$$\overline{R}_p = [R_p]\overline{\alpha}_p,$$

где $\overline{R}_p$ – вектор значений $R_l^{(i)}(j)$, $[R_p]$ – автокорреляционная матрица, $\overline{\alpha}_p$ – вектор коэффициентов предсказания $\alpha_v^{(i)}(l)$.

Главной отличительной особенностью ВР-кодирования является обновление на каждом периоде основного тона среднего значения приращения величины отсчётов в каждом канале, либо среднего приращения величины отсчетов по всему периоду в целом. Это позволяет значительно расширить возможности ВР-кодирования в части повышения коэффициентов сжатия и помехоустойчивости с сохранением высокого качества восстановленной речи. Эксперимента-льные исследования ВР-кодирования с дельта-модуляцией показали их значительные преимущества по сравнению с дельта-модуляцией скалярно-разностных алгоритмов. При двумерном предсказании речевого сигнала $x(i, n)$ уравнения разностного кодирования могут быть записаны в следующем виде

$$d(i, n) = x(i, n) - \tilde{x}(i, n).$$

В этом случае в качестве математической модели предсказателя можно использовать двумерное разностное уравнение

$$\tilde{x}(i, n) = \sum_{l=1}^{p_1-1} \sum_{k=1}^{p_2-1} a(l, k)\hat{x}(i-l, n-k).$$

Таким образом, математические методы разностного кодирования с двумерным линейным предсказанием являются эквивалентными математическим моделям многоканального векторно-разностного кодирования.

6.3. Быстрые алгоритмы вычисления оценок АКФ

Методы скалярно и векторно-разностного кодирования с линейным предсказанием базируются на решении системы линейных уравнений относительно коэффициентов предсказания, заданной через автокорреляционную матрицу. В свою очередь автокорреляционная матрица строится на основе апериодической автокорреляционной функции, заданной в последовательностной форме

$$\hat{r}(k) = \frac{1}{N_1} \sum_{n=0}^{N_1-k-1} x(n)\, x(n+k), \tag{6.4}$$

где $k = \overline{0, N_1 - 1}$.

Выражение (6.4) характеризует смещённую оценку последовательности, которая может рассматриваться как стационарный в широком смысле случайный процесс. Согласно [3] использование смещенной оценки АКФ гарантирует положительную полуопределенность автокорреляционной матрицы, необходимую для разрешимости системы линейных уравнений. Заметим, что пользуясь выражением (6.4), можно получить лишь первую половину значений АКФ. Вторая половина значений может быть получена за счёт симметрии значений АКФ.

Быстрые алгоритмы вычисления оценки АКФ можно получить либо через векторно-матричные, либо через полиноминальные формы представления периодических АКФ, через которые в том числе можно выразить апериодические АКФ, заданные выражением (6.4).

Оценку периодической АКФ можно также задать в последовательностной форме

$$\hat{r}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) x\langle n+k \rangle_N,$$

где аргумент $n + k$ дискретной функции $x(n + k)$ вычисляется по mod N.

Легко показать, что после дополнения в выражении (6.4) последовательности $\{x(n)\}_{n=0}^{N_1-1}$ N_1 нулями с помощью периодической АКФ можно вычислить оценки апериодической АКФ длины $2N_1 = N$. В векторно-матричной форме оценку периодической АКФ можно записать в виде

$$\hat{r}_N = \frac{1}{N} \hat{S}_N x_N, \tag{6.5}$$

где $\widehat{r}_N$ – вектор значений оценки АКФ, $\widehat{S}_N$ – матрица-правый циркулянт с образующим вектором-строкой $\overline{S}_N = \mathbf{x}'_N = (x_0, \ldots, x_{N-1})$ соответствующим последовательности $\{x(n)\}$. Для выражения (6.5) известен быстрый алгоритм вычисления АКФ через быстрое преобразование Фурье (БПФ), который можно записать

$$\widehat{r}_N = \frac{1}{N}(F_N^{(\Phi)})^{-1} D_N^* F_N^{(\Phi)} \mathbf{x}_N, \tag{6.6}$$

где $F_N^{(\Phi)}$ – факторизованная форма матрицы преобразования Фурье в поле комплексных чисел, соответствующая тому или иному алгоритму БПФ, D_N^* – диагональная матрица, элементами которой являются компоненты вектора комплексных коэффициентов БПФ от вектора $\mathbf{X}_N^*$, т. е.

$$D_N^* = diag\{d_0^*, d_1^*, \ldots, d_{N-1}^*,\},$$

$$d'_N = d_0, d_2, \ldots, d_{N-1}, \quad d_N = F_N \mathbf{x}_N$$

Если ввести $\overline{P} = \left|\overline{X}_N\right|^2 = D_N^* F^{(\Phi)} \mathbf{x}_N$ – вектор квадрата модуля коэффициентов БПФ от вектора $\mathbf{x}_N$, то из выражения (6.6) можно получить известное соотношение для корреляционного метода оценки спектральной плотности мощности сигнала

$$\widehat{P}_x(K) = \sum_{n=0}^{N-1} \widehat{r}(m) \exp(-i2\pi km / N).$$

Поэтому предлагаемые далее быстрые алгоритмы вычислений оценок АКФ в равной степени могут быть пригодны также для цифрового спектрального анализа.

6.4. Алгоритмы вычисления АКФ на основе быстрых гиперкомплексных преобразований Фурье

В работе [16] были предложены алгоритмы быстрых гиперкомплексных преобразований Фурье (ГПФ). Рассмотрим возможности применения данных алгоритмов для быстрого вычисления АКФ.

Матрица гиперкомплексных преобразований Фурье (ГПФ) задается в виде:

$$F_N^{(\Gamma)} = [j_1^{km}],$$

где $k, m = \overline{0, N-1}$, j_1 – мнимая единица, образующая мультипликативную циклическую группу из всех мнимых единиц. Данные ГПФ образованы в расширениях поля рациональных чисел, являющихся также полями.

Матрично-факторизованную форму представления алгоритмов БПФ по основанию два с прореживанием по частоте можно записать

$$F_N^{(\Gamma\Phi)} = \tilde{J}_N diag\left\{\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right\} diag\left\{\begin{pmatrix} I_2 & I_2 \\ D_2 & -D_2 \end{pmatrix}\right\} \times \ldots \times \begin{pmatrix} I_{N/2} & I_{N/2} \\ D_{N/2} & -D_{N/2} \end{pmatrix}$$

где $D_{N/2} = diag\{1, j_1, j_2, \ldots, j_{N/2-1}\}$, $D_2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$,

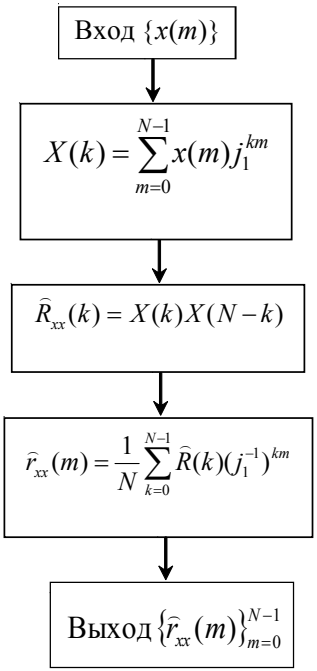


Рис.6.5 Блок-схема алгоритма вычисления апериодической АКФ через ГПФ

$i = \sqrt{-1}$ – мнимая единица в поле комплексных чисел, j_k – к-я мнимая единица в гиперкомплексной алгебре с (N/2-1) мнимыми единицами.

Блок-схема алгоритма вычисления N-точечной периодической АКФ через ГПФ представлена на рис.6.5, где через $\hat{R}_{xx}(k)$ обозначена оценка АКФ в области ГПФ. Применяя к ней обратное ГПФ, получаем значения АКФ во временной области $r_{xx}(m)$.

Использование алгоритмов ГБПФ лучше всего продемонстрировать на примере. На рис.6.6 приведены графы прямого и обратного ГБПФ, соответствующий вычислению апериодической АКФ через 8 точечную периодическую АКФ. Значения коэффициентов ГПФ представлены в гипер-комплексной 4-х ортовой алгебре квантернионов $\{1, j, i, k\}$, $i=\sqrt{-1}$, $j=\sqrt{-i}$, $k=\sqrt{-i}$.

Приведенный пример позволяет выявить два основных преимущества технологии вычисления АКФ на основе ГБПФ:

- все арифметические операции выпол-

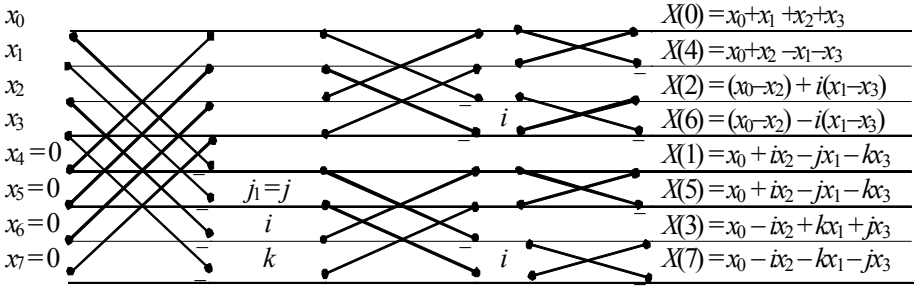


Рис.6.6. Граф прямого ГБПФ для исходного сигнала

няются в поле рациональных чисел, если полагать, что цифровые отсчёты $x(m)$ рациональные числа;

- количество арифметических операций значительно меньше, чем, например, при использовании БПФ, либо быстрого преобразования Хартли: количество умножений 10 против 12, количество сложений 17 против 39.

Значения АКФ в области ГПФ с учётом симметрии коэффициентов можно представить в виде

$$R(1) = R(7) = a + jb - kb,\ a = x_0^2 + x_1^2 + x_2^2 + x_3^2,$$
$$b = x_0x_3 + x_1x_2 + x_0x_1 + x_2x_3 = x_0(x_1 - x_3) + x_2(x_1 - x_3);$$
$$R(3) = R(5) = a - jb + kb,\ R(2) = R(6) = (x_0 - x_2)^2 + (x_1 - x_3)^2;$$
$$R(4) = (x_0 + x_2 - x_1 - x_3)^2.$$

Заметим, что алгоритм вычисления апериодических АКФ через быстрое преобразование Хартли (БПХ) может быть получен посредством подстановки в выражение (6.6) матрицы

$$F_N = P_N H_N,$$

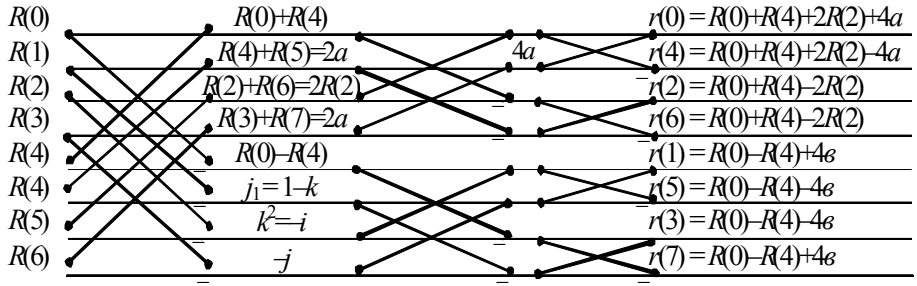


Рис. 6.7. Граф обратного ГБПФ

где $P_N = \frac{1}{N} F_N \cdot H_N$ – переходная матрица от матрицы H_N к матрице F_N.

В результате получим

$$\hat{R}_N = \frac{1}{N} H_N^{(\Phi)} [D_N^{(h)}] H_N^{(\Phi)} X_N = \frac{1}{N} \bar{J}_N H_N^{(\Phi)} D_N^{(h)} H_N^{(\Phi)} \mathbf{x}_N$$

где $[D_N^{(h)}] = P_N^{-1} D_N^* P_N$,

$$D_N^{(h)} = d_o^{(h)} \oplus 1/2 \left[\left(\bigoplus_{k=1}^{N-1} \left(d_{N-k}^{(h)} + d_k^{(h)} \right) \right) + \left(\bigoplus_{k=1}^{N-1} \left(d_{N-k}^{(h)} - d_k^{(h)} \right) \right) \bar{I}_{N-1} \right],$$

$\bar{d}_N^{(h)} = H_N \mathbf{x}_N$, $d_k^{(h)}$ – κ-ый компонент вектора $\bar{\mathbf{d}}_N^{(h)}$,

$\bar{\bar{J}}_k = 1 \oplus \bar{I}_{N-1}$, $\bar{\bar{I}}_{N-1}$ – матрица инверсной перестановки.

Виды факторизованных форм $H_N^{(\Phi)}$ матриц H_N приведены в работе [8].

Покажем каким образом можно использовать ФСР для аппроксимации нормированных апериодических АКФ, через которые строятся корреляционные матрицы.

Для упрощения выкладок в дальнейшем будем использовать прямоугольные весовые функции с амплитудой равной единице, так что выражение (17) приобретает вид

$$\hat{\gamma}(k) = \frac{1}{N} \sum_{m=0}^{N-1} \left| x(m) - x(m+k) \right|, \qquad (6.7)$$

где $m, k = \overline{0, N-1}$, $m + k = \langle m + k \rangle_N$.

Функцию (6.7) по аналогии с АКФ можно рассматривать как периодическую, если параметр (m + k) вычисляется по mod N.

Периодическая функция $\hat{\gamma}(k)$ также как и периодическая АКФ обладает симметрией, т. е. $\hat{\gamma}(k) = \hat{\gamma}(N-k)$, $k = \overline{1, N-1}$.

Введем дискретную функцию, образованную через дополнения $\tilde{y}(k) = \gamma_{\max} - \hat{\gamma}(k)$. Полученная функция $\tilde{y}(k)$ будет иметь максимальное значение при $k = 0$ (при нулевом сдвиге окна, также, как АКФ).

Если допустить, что с помощью N-точечной периодической ФСР вычисляется $N/2$ точечная апериодическая ФСР, то исходная последовательность отсчётов цифрового сигнала $\{x_i\}_{i=0}^{N/2-1}$ должна быть дополнена $N/2$ нулями, (также, как АКФ). Тогда минимальное значение периодической

ФСР, равно $\tilde{\gamma}_{\min} = \gamma(N/2) = 0$. Пронормировав значения $\tilde{y}(k)$, получим соотношения

$$\tilde{\gamma}_{\min} = \hat{\gamma}(N/2) = \hat{R}(N/2) = 0,\ \tilde{\gamma}_{\max} = \hat{\gamma}(0) = \hat{R}_{\max} = \hat{R}(0).$$

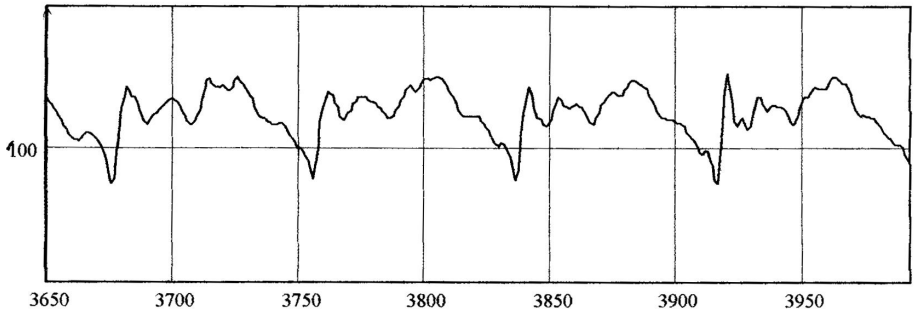
Остаётся проанализировать поведение функций $\hat{\gamma}(k)$ и $\hat{R}(k)$ в промежуточных точках между значениями $\tilde{\gamma}_{\max}$ и $\tilde{\gamma}_{\min}$.

Введем функцию Д(k), отражающую погрешность отклонения

$$\Delta(k) = \frac{\hat{R}(k)}{\hat{R}_{\max}} - \frac{\tilde{\gamma}(k)}{\hat{\gamma}_{\max}},$$

где $$\hat{R}_{\max} = R(0) = \frac{1}{N}\sum_{i=0}^{N-1} x_i^2,\ \tilde{\gamma}_{\max} = \tilde{\gamma}(0) = \frac{1}{N}\sum_{i=0}^{N-1}\left|x_i\right|.$$

а) $x(n)$



б)

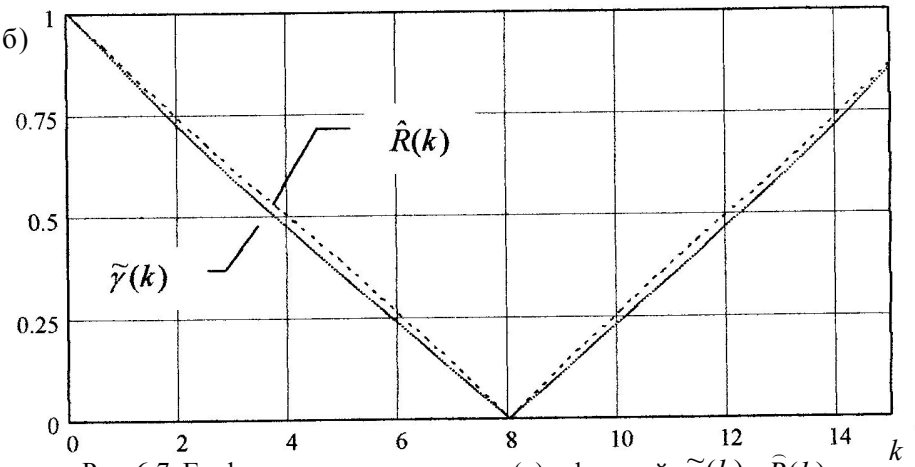


Рис. 6.7. Графики исходного сигнала $x(n)$ и функций $\tilde{\gamma}(k)$, $\hat{R}(k)$

Из графиков можно заключить, что характер изменения значений $\bar{R}(k)$ и $\tilde{\gamma}(k)$ является одинаковым, хотя в отдельных случаях наблюдается отклонение значений $\bar{R}(k)$ и $\tilde{\gamma}(k)$ до 30%.

Для практики интерес представляет возможность использования дополненных нормированных ФСР для вычисления $N/2$ оптимальных коэффициентов предсказания из системы линейных уравнений, заданных через тёплицеву матрицу, образующей строкой которой может быть использованы $N/2$ значений ФСР вместо $N/2$ значений АКФ.

6.5. Примеры использования корреляционных функций для оценки свойств кодовых последовательностей, связанных с речевыми сигналами

С помощью пакета Mathcad организован доступ к тестовому wav-файлу цифрового сигнала

```
file := "Test.wav"
r := GETWAVINFO(file)
x := READWAV(file)
N := length (x)
t := 0.. N
N := 4000
```

$$r = \begin{pmatrix} 1 \\ 2.205 \times 10^4 \\ 8 \\ 2.205 \times 10^4 \end{pmatrix}$$

и выведен на отображение (рис.6.8) подлежащий обработке фрагмент сигнала.

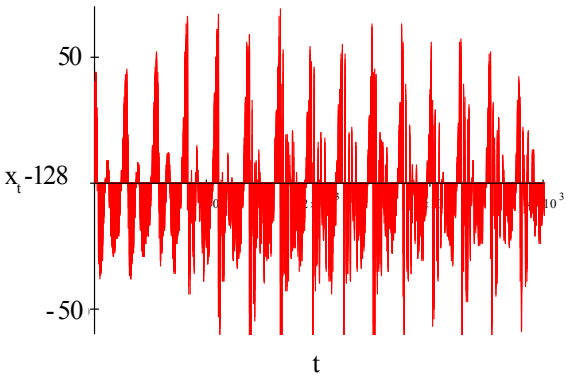
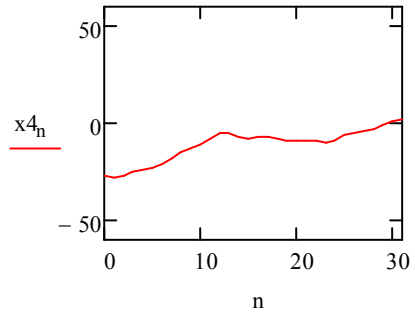


Рис. 6.8. Выборка исходного цифрового сигнала, N=4000

Формы сигналов и АКФ для фрагментов сигнала малых длин N=32 и N=128 представлены соответственно на рис.6.9 и рис. 6.10.

$$x4 := \left| \begin{array}{l} \text{for } i \in 0..31 \\ a_i \leftarrow x_{i+1300} - 128 \\ a \end{array} \right.$$

$$R_i := \sum_{n=0}^{31} \frac{\left(x4_n \cdot x4_{\text{mod}(n+i,32)}\right)}{32}$$



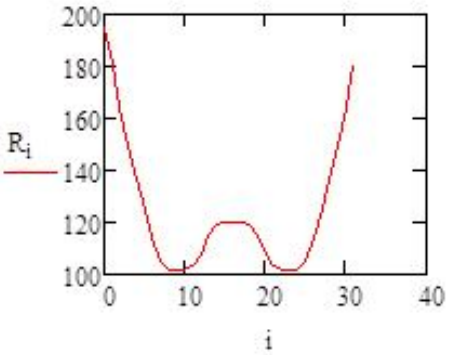
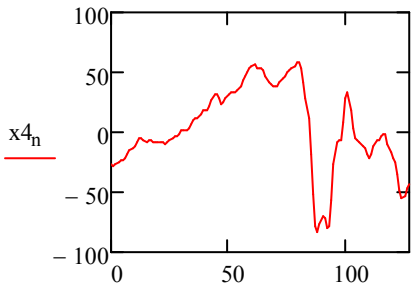


Рис. 6. 9. Исходный сигнал, N=32, АКФ сигнала, N=32



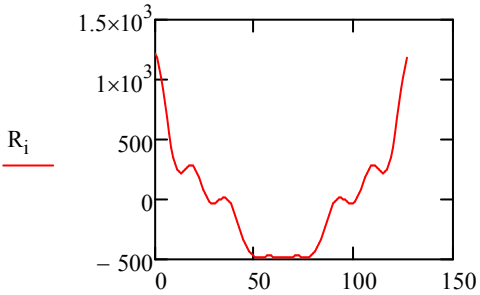
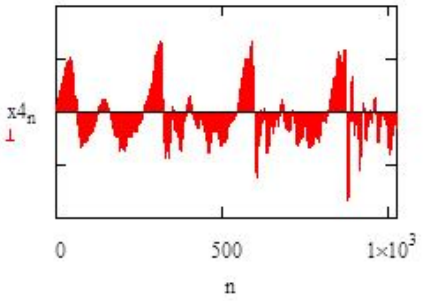


Рис. 6.10. Исходный сигнал, N=128 , АКФ сигнала, N=128

Оценка качества криптозащиты по АКФ представлена сранительного визульного сравнения АКФ исходного речевого сигнала и зашифрованного этого же сигнала двоичной псевдослучайной последовательностью длиной, равной длине выборки в двоичном представлении (рис. 6.11, 6.12).

В заключение читателю предлагается организовать попытку взлома использованного в данном примере шифра.



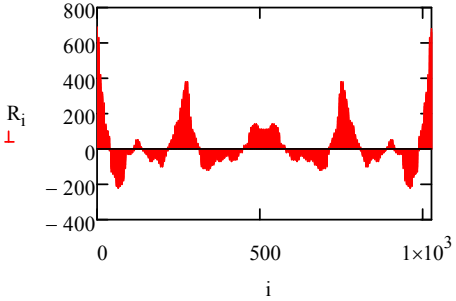


Рис. 6.11. Исходный речевой сигнал АКф исходного речевого сигнала

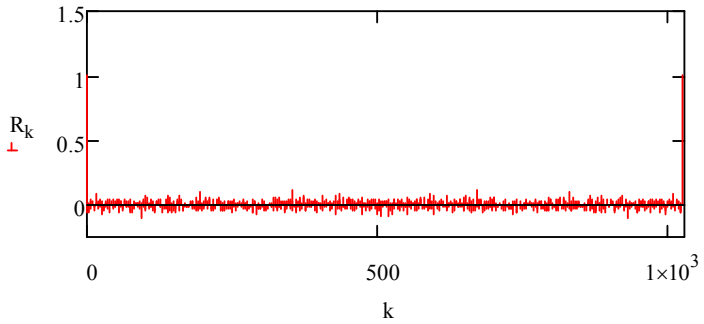


Рис. 6.12. АКФ речевого сигнала, защищённого блочным криптошифром, N=1024

ЛИТЕРАТУРА

1. Хорн Р., Джонсон Ч. Матричный анализ. М.: Мир, 1989.
2. Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир, 1988, т.1, т.2.
3. Марпл С. Л. Цифровой спектральный анализ и его приложения. М., Мир, 1990.
4. Ахмед Н., Рао К.Р. Ортогональные преобразования для цифровой обработки сигналов. М.: Радио и связь, 1980.
5. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1985.
6. Гагарин Ю.И., Шифрин В.В. Модульно-комбинированные спектральные весовые функции для ДПФ // Радиотехника и электроника, Вып.2, 1992, с.247.
7. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свёрток. М.: Радио и связь, 1984.
8. Гагарин Ю. И. Математические модели и алгоритмы быстрых ортогональных преобразований. СПбГТУ, 1999.
9. Макклеллан Дж., Рейдер Ч. Применение теории чисел в цифровой обработке сигналов. М.: Радио и связь, 1983.
10. Гагарин Ю.И. Характеризационно-инвариантный синтез быстрых ортогональных и теоретико-числовых преобразований: Дис. д-ра техн. наук. СПб, 1997.
11. Гагарин Ю.И., Гагарин К.Ю. Теоретико-числовые преобразования Мерсенна для быстрого вычисления вещественных свёрток, длина которых факторизована степенями числа два // Известия ВУЗов СССР , сер. Радиоэлектроника. Киев, 1991, т.12, с. 2182.
12. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии. М.: Триумф, 2003.
13. Соколов В.И. Вейвлет-пребразования в аппроксимированных ортогональных базисах// Учёные записки РГГМУ, Вып.18, 2011,с.94.
14. Рабинер Л. В., Шафер Р. В. Цифровая обработка речевых сигналов. М., Радио и связь, 1981.
15. Schroeder M.R., Atal B. Code Excited Linear Prediction (CELP): High Quality Speech at Very how Rates. Proc. ICASSP-85, p.937.
16. Гагарин К. Ю. Математические модели и быстрые алгоритмы векторно-разностного кодирования цифровых речевых сигналов// Информационно-управляющие системы, №1(14), 2005, с.2.

ПРИЛОЖЕНИЕ

ОРГАНИЗАЦИЯ И ПРИМЕНЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

1. Формальные системы представления данных и описания функций элементов ЭВМ

1.1. Арифметический тип данных: системы счисления и правила выполнения операций над двоично-кодированными вещественными числами в машинной арифметике

Под системой счисления понимают правила обозначения и размещения цифр при записи вещественных чисел. В ЭВМ используются только позиционные системы счисления, где значение каждой цифры определяется её местоположением в записи числа. Каждая позиционная система счисления характеризуется основанием *b,* и в общем случае вещественное число может быть записано в форме с фиксированной запятой (точкой), разделяющей целую и дробную части

$$X = x_{n-1} \ldots x_3\ x_2\ x_1\ x_0, x_{-1}\ x_{-2}\ x_{-3} \ldots x_{-k}\,. \qquad (\text{П1})$$

Отдельные цифры в составе числа получили название разрядов, определяемых множеством значений $x_i \in \{0,1,\ldots,\ b-1\}$. Наиболее распространенными в ЭВМ являются системы со следующими основаниями: b=10 - десятичная; b=2 - двоичная; b=*8* - восьмеричная; b=16 - шестнадцатеричная. Для систем, в которых b<10, обозначение цифр совпадает с обозначением части цифр десятичной системы, например, для восьмеричной системы $x_i^{(8)} \in \{0,1,\ldots 7\}$, а для двоичной. Разряды шестнадцатеричных чисел принимают следующие значения:

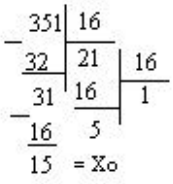
$$x_i^{(16)} \in \{0,1,\ldots,15\} = \{0,1,2,\ldots,9, A, B, C, D, E, F\}.$$

Существует общая форма определения десятичного эквивалента числа, заданного в любой другой системе счисления. Применительно к примеру (П1) можно записать

$$X = \sum_{j=-k}^{n-1} x_j b^j\,.$$

Выполнение арифметических операций над числами связано с системами счисления, в которых они представлены. Число *X,* представленное в различных системах, может быть выражено десятично-эквивалентной формой $X_2 = X_{16} = X_8 = X_{10}$. Например, $28_{10} = 1C_{16} = 11100_2$.

При использовании ЭВМ в ряде случаев требуется осуществлять преобразование данных из одной системы счисления в другую программными средствами. Преобразование десятичных чисел в любую другую систему счисления осуществляется поочерёдным делением на основание данной системы исходного десятичного числа и всех последующих полученных от деления целых частных . При этом первый остаток от деления есть младший разряд, а последний остаток есть старший разряд искомой формы числа. Пример перевода десятичного числа в шестнадцатеричную систему схематично можно представить в виде



Окончательно имеем $351_{10} = 15F_{16}$.

Существует простое правило перевода шестнадцатеричных чисел в двоичные и наоборот. При переводе требуется записать каждый разряд переводимого шестнацатеричного числа четырехразрядным двоичным числом - эквивалентом. Например, полученное ранее число приобретает форму

$$351_{10} = 15F_{16} = \overbrace{0001}^{1}\ \overbrace{0101}^{5}\ \overbrace{1111_2}^{F} = 101011111_2.$$

Обратный перевод выполняется сопоставлением каждой четвёрке двоичных разрядов, начиная с младших, их шестнадцатеричного эквивалента, как показано в примере

$$\overbrace{0010}^{2}\ \overbrace{1000}^{8}\ \overbrace{0110_2}^{6} = 286_{16}.$$

Перевод десятичной дроби в b-ичную систему можно свести к последовательности следующих операций. Исходное число умножается на осно-

вание b с фиксацией разряда целой части, т.е. разряда переполнения в представлении дроби, который является старшим значащим разрядом искомой b-ичной дроби. Данная процедура применяется для каждой десятичной дроби, полученной от умножения предидущей на основание b. Ниже приведён пример перевода десятичной дроби в двоичную

Разряд двоичного числа	0.672 2
1	0.344 2
0	0.688 2
1	0.376 2
0	
1	.....

В результате получили $0.672_{10} = .10101..._2$.

При выполнении арифметических операций с двоично-кодированными вещественными числами со знаком в представлении их с фиксированной точкой принято использовать прямой, обратный и дополнительный коды. Запись числа X со знаком в отличие от бесзнаковой (П1) отличается появлением старшего знакового разряда z, принимающего значения 0 для положительного и b-1 для отрицательного числа. Так что форма (П1) для прямого кода приобретает обобщённый вид

$$X= z\ , x_{m-1}\ x_{m-2}\ \dots\ x_1\ x_0\ .\ x_{-1} \dots x_{-n}$$

где для отрицательных двоичных чисел знаковый разряд z =1, а для отрицательных шестнадцатеричных $z=F$.

Прямой код числа соответствует общепринятым обозначениям, однако его недостатком является неоднозначность представления нуля ±0. Кроме того прямой код не позволяет осуществлять заём старшего разряда при выполнении операции вычитания чисел. В связи с этим в ЭВМ принято использовать формы представления для вычитаемых чисел через их дополнения. По сути такой приём сводит вычисления в бесконечном поле вещественных чисел через модулярную арифметику, используемую, например, в конечных полях, названных полями Галуа (*GF*) в честь французского математика Эвариста Галуа (*Evariste Galois*, 1811-1832).

Формально процесс использования дополнений для замены операции

вычитания через сложение можно записать следующим выражением

$$x - y = x + (C - y) - C, \qquad (П2)$$

где $x, y - p$ - разрядные b-ично- кодированные целые числа, а величина C принимает значение $b^p - 1$.

Для двоично-кодированных чисел при b=2 и $C = 2^p - 1$ обратный код числа (-y) определяется, как его дополнение до числа $2^p - 1$, *т.е. можно записать* $\bar{y} = 2^p - y - 1$. Но в соответствии с выражением (2) из полученной суммы $x + \bar{y}$ *необходимо вычесть* $2^p - 1$. В итоге получим выражение $x - y = x + \hat{y} - 2^p$, в котором вычитаемое число принято запоминать отрицательным знаком (старшим значением знакового p-го разряда) дополнительного кода, который участвует в последующем определении фактического значения разности. Например, из-за возможного переноса из значащей части полученной суммы, может изменяться знаковый разряд. При отрицательном знаке для получения фактической разности необходимо взять дополнительный код от полученной суммы.

Ниже приведены примеры выполнения арифметических операций с числами, представленными в форме с фиксированной запятой. Заметим, что эта форма преимущественно используется либо для целых чисел, либо для правильных дробей.

Пример 1. Вычитание двоичных чисел с использованием дополнительного кода.

Пусть заданы два целых четырёх разрядных числа со знаком в прямом коде x = 01011 и y = 00111. Требуется вычислить разность $s = x - y$.

Находим дополнительный код числа $-y$: $\hat{y} = \bar{y} + 1 = 11001$.

Затем получаем значение разности:

$$s = x + \hat{y} = 01011 + 11001 = 00100 = 4.$$

Теперь пусть требуется найти разность $s_1 = y - x$. Дополнительным кодом для числа (-x) будет $\hat{x}$ = 10101. Складывая y и $\hat{x}$, получим $\hat{s}_1 = y + \hat{x}$ = 00111 + 10101=11100. Отрицательный знак суммы указывает на то, что фактическое значение разности равно дополнительному коду полученной суммы s_1 =10100 =- 4.

При реализации двоичной арифметики с фиксированной точкой необходимо предусмотреть меры борьбы с переполнением разрядной сетки программируемого арифметико-логического блока процессора, т.е. превышения количества двоичных разрядов необходимого для результата вы-

полненной арифметической операции по сравнению с разрядностью арифметико-логического блока процессора.

Наиболее распространённой мерой борьбы с переполнением является размещение результата в разрядную сетку процессора посредством понижения его точности. Различают при этом погрешности масштабирования для сложений и погрешности округлений для умножений. В первом случае при p-разрядной сетке сумматора слагаемые x и y сдвигаются на один разряд вправо с потерей значения их младшего разряда. Полученная таким образом p-разрядная сумма, домноженнная на масштабирующий коэффициент, равный двум, будет обладать величиной погрешности, определяемой статистикой младших загрубляемых разрядов операндов.

В арифметике с плавающей точкой общая форма записи числа x имеет вид $x = (m_x, \pm f_x)$, что соответствует значению числа $x = b^{m_x} \cdot (\pm f_x)$, где f_x- дробная часть числа, m_x- положительный порядок числа. При использовании формы с плавающей запятой дробную часть обычно приводят к значению с ненулевым старшим разрядом, т.е. нормализуют форму числа. Нормализация дробной части сопровождается коррекцией показателя m.

Операция умножения чисел в форме с плавающей запятой сводится к умножению нормализованных дробных частей и сложению показателей. Сложение чисел в форме с плавающей запятой включает выравнивание показателей слагаемых по большему показателю, затем сложение дробных частей (возможно уже ненормализованных) с последующей нормализацией полученной суммы.

Пример 2. Пусть заданы двоичные числа:

$$x_1 = 2^{010} \cdot 0.11000_2 = 3.0 \text{ и } x_2 = 0.10100_2 = 0.625.$$

После выравнивания показателей, заданных здесь так же, как и дробных частей, в форме с фиксированной запятой, получим

$$x_1 + x_2 = 2^{010} \cdot 0.11101_2 = 3.625.$$

Нормализация суммы здесь не потребовалась. Основным преимуществом формы с плавающей точкой является широкий диапазон представления чисел, что практически исключает проблему переполнения разрядной сетки и значительно повышает точность вычислений. В реальных ЭВМ отношение количества байтов в представлении порядка к количеству байтов дробной части обычно равно одной трети, а в целом их суммарная длина равна длине разрядной сетки процессора.

1.2. Логический тип данных. Основные сведения из булевой алгебры и её применение к синтезу логических устройств

Помимо рассмотренного формального аппарата описания двоичноразрядной арифметики в поле вещественных чисел, для анализа и синтеза элементов и устройств ЭВМ нашла широкое применение булева алгебра, названная так в честь великого французского математика Джоржа Буля (*George Boole*) .

Всякую алгебру можно представить в теоретико-множественном виде, где указывается множество исходных элементов и перечисляются допустимые для них операции-отношения. Тогда для булевой алгебры можно записать

$$K = < B, \vee , \wedge , \neg >, \qquad (П3)$$

где $B=\{0,1\}$- исходное числовое множество с допустимыми операциями $\vee , \wedge , \neg$ соответственно логического сложения, логического умножения и логического отрицания (дополнения).

Булева алгебра построена на следующих аксиомах:

1. Для любых $x, y \in B$ удовлетворяются отношения $x \vee y = z_1 \in B$ и $x \wedge y = z_2 \in B$.

2. В множестве B есть такие элементы 0 и 1, что для всякого $x \in B$ $x \vee 0 = x$, $x \wedge 1 = x$.

3. Для всех $x, y, z \in B$ справедливы следующие свойства дистрибутивности

$$(x \vee y) \wedge z = x\, z \vee y\, z,$$
$$(x\ \ y) \wedge z = x\, z \vee y\, z,$$

4. Для всякого $x \in B$ имеется его дополнение (отрицание) $\bar{x} \in B$ - такой, что $x \vee \bar{x} = 1$, $x \wedge \bar{x} = 0$.

На основании аксиом выводятся свойства булевой алгебры, как, например, коммутативность, ассоциативность и идемпотентность по логическим операциям сложения и умножения ; далее сложение переменной с константой единица и умножение переменной на константу нуль: $x \vee 1 = 1$, $x \wedge 0 = 0$, свойства поглощения $x \vee xy = x$, склеивания по переменной $x\, y \vee \bar{x}\, y = y$, $(x \vee y) \wedge (x \vee \bar{y}) = x$, и правило де Моргана $\overline{x\, y} = \bar{x} \vee \bar{y}$.

Полезно отметить здесь очевидное смысловое сходство логических операций сложения, умножения и отрицания с соответствующими операциями алгебры множеств $\cup$ – объединение, $\cap$ – пересечение, $\neg$ – дополнения до универсальности.

Для формального описания, анализа и синтеза элементов и устройств ЭВМ широкое применение нашли булевы логические функции. Неформальным способом булеву функцию можно представить как выражение из булевых переменных с символами логических операций. Например, выражение $x \vee y = f(x, y)$ соответствует функции логического сложения двух переменных x, y. Кроме способа задания булевых функций в форме логического выражения существует также табличный способ, в виде таблицы истинности. Например, для функции $f(x, y) = x \vee y$ таблица истинности имеет следующий вид

Таблица П1

Таблица истинности

x	y	$f(x, y)$
0	0	0
0	1	1
1	0	1
1	1	1

Из таблицы истинности булеву функцию можно записать в двух эквивалентных формах либо в совершенной дизъюнктивной (СДНФ) через дизъюнкцию элементарных логических произведений всех переменных, взятых с учётом отрицаний, на которых функция принимает значения единицы, либо в совершенной конъюнктивной форме (СКНФ) через конъюнкцию элементарных дизъюнкций всех переменных. В практических применениях и в публикациях преимущественно используется СДНФ. В нашем примере СДНФ $f(x,y)$ записывается в виде $f(x,y) = xy \vee \bar{x}y \vee x\bar{y}$. Пользуясь свойством склеивания логических произведений по переменным x и y, получаем выражение $f(x,y) = x \vee y$ функции логического ИЛИ, являющейся математической моделью соответствующего элементарного логического вентиля.

В схемотехнике для каждого типа логического вентиля используются стандартные обозначения в форме прямогольника с указанием символа реализуемой функции [1]. Набор логических функций, через которые можно представить любой сложности логическое выражение, получил название функционально полного базиса. Например, каждый из наборов (И-НЕ), (ИЛИ-НЕ), (И, ИЛИ, НЕ) является функционально полным.

В практике схемотехнического проектирования дискретных утройств формальный логический синтез нашёл широкое применение, особенно для

схем, представленных однотипными функциональными модулями. В качестве примера можно привести схему (рис.П1) сумматора для четырёхразрядных двоично-кодированных вещественных чисел [П1] , состоящую из четырёх последовательно соединённых одинаковых модулей ПС-схем полных одноразрядных сумматоров

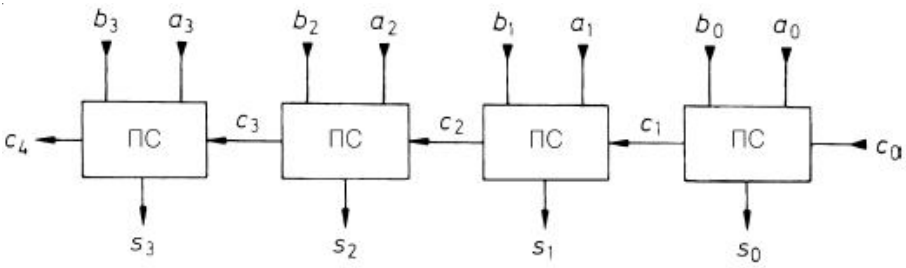


Рис. П1. Логическая схема 4-х разрядного сумматора

Очевидно, что использование формального аппарата логического синтеза в целом для схемы 4-х разрядного сумматора будет сопровождаться рутинными вычислениями, связанными с построением и преобразованием пяти булевых функций от девяти переменных. Значительно проще применить формальный аппарат синтеза на уровне даже не одноразрядного сумматора, а на уровне полусумматора. Ниже приведён пример построения таблицы истинности логических функций полусумматора, на основе которой построена схема полного одноразрядного сумматора (рис.П2).

Таблица П2

Таблица истинности функций полусумматора

ai	***bi***	***pi***	***gi***
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Заметим, что схема 4-х разрядного сумматора (Рис.П1) может быть успешно использована в качестве вычитателя, если вычитаемое представлено в дополнительном коде. Знаковый разряд результата при этом формируется в виде инвертированного выхода переноса старшего разряда числа.

При рассмотрении структур аппаратных средств реализации операций сложения и умножения двоично-кодированных вещественных чисел уде-

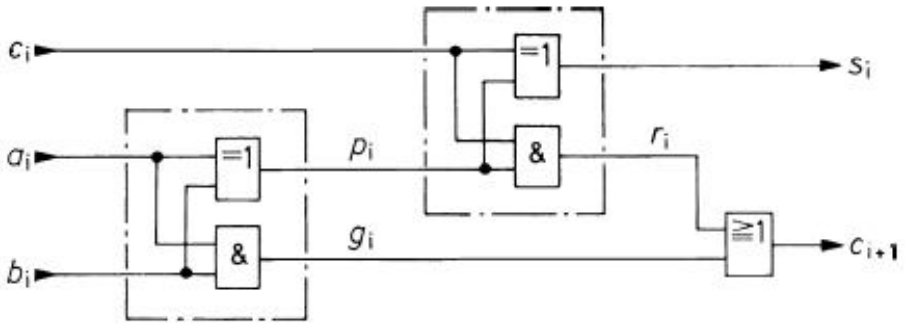


Рис. П2. Логическая ПС - схема на основе полусумматоров

ляется большое внимание повышению их быстродействия. Например, в арифметике с фиксированной точкой используются средства параллельных вычислений при формировании переносов в сумматорах, а для умножений нашли применение параллельные (однотактные) умножители. Следует отметить, что проблема ускорения переносов в двоичных сумматорах рассмотрена в фундаментальной работе [П2], когда электронные вычислительные устройства находились на самой ранней (ламповой) стадии организации. В работе уделено внимание также сравнительным оценкам качественных характеристик арифметических устройств с фиксированной и с плавающей точкой.

В настоящее время существуют два принципиально отличных направления проектирования схем логических устройств: на жесткой логике и на программируемой логике. В первом случае в схеме устройства используются стандартные логические элементы, связи между которыми жестко зафиксированы под реализуемые функции. Во втором случае используется программируемая логика, например, в виде постоянных запоминающих устройств (ПЗУ) или программируемых логических матриц (ПЛМ). Схемы ПЗУ или ПЛМ могут быть однократного либо многократного использования. Схемы многократного использования получили название репрограммируемых, а одноразового – маскируемых .

В наиболее известной практике схемотехнического проектирования ПЗУ используются для хранения двоичных кодов программ и данных в определённых форматах-словах, каждое из которых рамещено по своему адресу. Использование схем-модулей ПЗУ для логического синтеза основано на представлении адресного дешифратора набором логических схем и с количеством двоичных входов в каждой, равным числу адресных

входов ПЗУ. Так, что на выходе адресного дешифратора ПЗУ образуются всевозможные логические произведения из всех переменных, которые могут быть использованы для составления СДНФ произвольных булевых функций от заданного количества переменных. Объединение этих произведений (термов) осуществляется на логических схемах ИЛИ под определённые функции методом разрушения либо создания электрических соединений. Наличие или отсутствие соединения входов схем ИЛИ с выходами схем И определяется двоичной информацией, заносимой с помощью специального устройства-программатора в ПЗУ. Типовые ПЗУ и их характеристики приведены в работах [П1, П3].

Структурная организация программируемых логических матриц (ПЛМ) отличается от ПЗУ тем, что электрические соединения программируются не только на входах схем ИЛИ, но и на выходах схем И. Таким образом, ПЛМ позволяют реализовать булевы функции, представленные не только в СДНФ (для ПЗУ это единственная форма представления логических функций), но и в других преобразованных дизъюнктивных формах. Сведения о типовых ПЛМ можно получить также из источника [П1,П3].

1.3. Математические модели конечных дискретных автоматов

Все ранее рассмотренные логические схемы являются схемами комбинационного типа, т.е. схемами без элементов памяти. В реальных вычислительных устройствах комбинационные схемы используются совместно с элементами памяти, простейшими из которых являются двоичные триггеры. Для описания структур и схем с памятью используется формальный аппарат конечных дискретных автоматов, абстрактную общую модель которых можно представить в теоретико-множественном виде $A=\{X, Z, Y, S, s\}$, где X и Z - множества состояний соответственно входов и выходов автомата, Y- множество состояний выходов комбинационной схемы, соединённых со входами элементов памяти (ЭП) S-множество внутренних состояний автомата, s-начальное состояние автомата. Абстрактной модели автомата может быть сопоставлена структурная модель в виде его структурной схемы (рис. П3), пригодной для логического синтеза на уровне булевых функций, составленных для каждого выхода автомата с учётом изменения состояний элементов памяти по временным тактам. Структурная схема автомата сопровождается таблицей переходов, либо диаграммой состояний, выполненной, например, в

виде ориентированного графа, вершинам которого соответствуют внутренние состояния автомата, а рёбрам-векторы входных и выходных состояний. Применение формального аппарата конечных дискретных автоматов можно продемонстрировать на схеме двоичного *RS*-триггера (рис. П4).

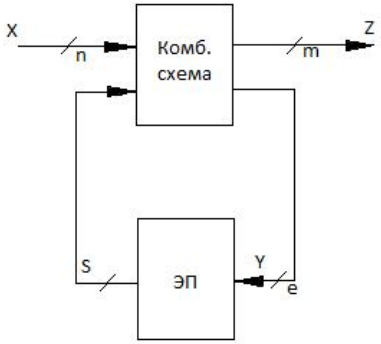


Рис. П3. Общая структурная схема автомата

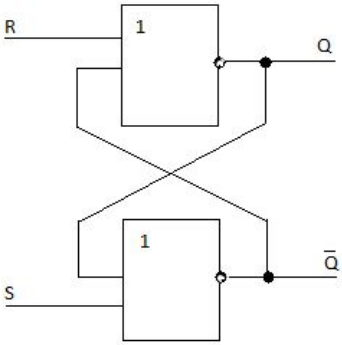


Рис. П4. Логическая схема *RS*-триггера

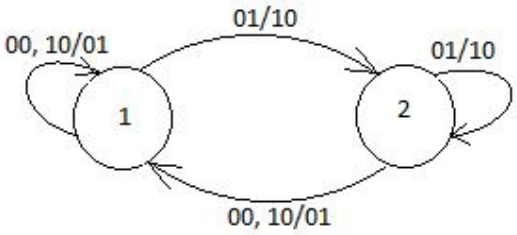


Рис. П5. Диаграмма состояний *RS*-триггера

Таблица П3

Таблица переходов

R	***S***	***Q***	$\bar{Q}$
0	0	0	1
1	0	0	1
0	1	1	0
1	1	-	-

Из приведённого примера можно заметить, что модели дискретных конечных автоматов наиболее удобно использовать при малой размерности задач синтеза логических схем. В сложившихся реальных условиях

проектирование схем логических устройств осуществляется с использованием компьютерных средств моделирования, включённых в состав систем автоматизированного проектирования (САПР).

2. Общие принципы построения ЭВМ и МП-систем

2.1. Основы организации однопроцессорных ЭВМ

Наибольшее распространение получили ЭВМ, построенные по принципам, рассмотренным в указанной выше работе [П2], и получившим название неймановских. Основы построения неймановского типа ЭВМ можно свести к нескольким положениям (постулатам), сформулированым следующим образом:

- решаемая задача должна быть представлена в виде последовательности шагов или действий, причём в каждый заданный момент времени выполняется лишь один шаг (действие), что обусловлено наличием одного арифметико-логического блока в центральном процессоре;
- ЭВМ должна работать под управлением двоичных слов, символы которых сгруппированы по смысловому назначению, отражающему вид и последовательность требуемых для выполнения операций;
- последовательность управляющих двоичных слов (программа) должна храниться в памяти открытого доступа, т. е. каждому слову должен соответствовать свой адрес, по которому осуществляется его считывание, либо запись.

На основании сформулированных трёх положений можно кратко охарактеризовать состояние организации и применения современных ЭВМ. Наиболее специфичным для неймановских (однопроцессорных) ЭВМ является первое положение, по которому решаемая задача должна быть формализована в итерационно-циклической форме (чаще всего в виде ориетированного графа) с возможностью последующей её интерпретации выбранными средствами программирования. Такая форма представления задачи, т.е. в виде вычислительного алгоритма, является наиболее важной и трудной в технологиях программирования. Профессиональный уровень разработки вычислительных алгоритмов существенно влияет на возможности выявления допущенных неточностей, логических и математических несоответствий поставленной задаче.

В процессе алгоритмизации можно выделить общие составляющие, независимые от типов ЭВМ и языков программирования. Например, вся-

кий вычислительный алгоритм содержит стадии описания исходных данных и итерационного управления вычислительным процессом с выводом результатов. Обычно алгоритмизация задачи включает несколько этапов, позволяющих представить задачу с различной степенью детализации и приближения к выбранным средствам программирования. В ЭВМ, организованных по другим принципам (неймановским), например, с конвейерными или векторными процессорами проблема алгоритмизации становится значительно сложнее.

Второй и третий принципы организации ЭВМ связаны с управлением вычислительным процессом. Здесь различают управление выбором самих управляющих слов, получивших название двоичных команд, и управление данными (операндами). Заметим, что в современных средствах программирования ЭВМ используются символические - операторные языки, с имеющимися автоматизированными системами редактирования и отладки.

Сформулированные постулаты нашли широкое воплощение в современных ЭВМ и микропроцессорных системах. Далее при описании типовых структурных схем вычислительных машин и устройств для обозначений элементов и узлов будет использована русскоязычная символика.

2.2. Общая схема аппаратных средств ЭВМ

Главными функциональными узлами общей схемы аппаратных средств, ориентированной на широкого пользователя однопроцессорной ЭВМ, (рис. П6) являются центральный процессор (Пр), оперативное запоминающее устройство (ОЗУ) для хранения выполняемых программ с данными, постоянное запоминающее устройство (ПЗУ) для хранения программ, инициирующих начало работы компьютера после его подключения к электросети, ИМ-интерфейсные модули (иначе-порты, или контроллеры), предназначенные для сопряжения ЭВМ с внешними устройствами (ВУ) и системный интерфейс, включающий три многопроводные шины: адреса, данных и управления. Назначением системного интерфейса ЭВМ является организация взаимодействия между её функциональными блоками под управлением Пр.

В блоке Пр следует выделить АЛУ-арифметико-логическое устройство, РОН-регистры общего назначения (для кратковременного хранения данных, адресов ОЗУ и другой информации), ПС - программный счётчик (формирователь адресов ячеек ОЗУ, ПЗУ и ИМ), Ф-флаговый регистр

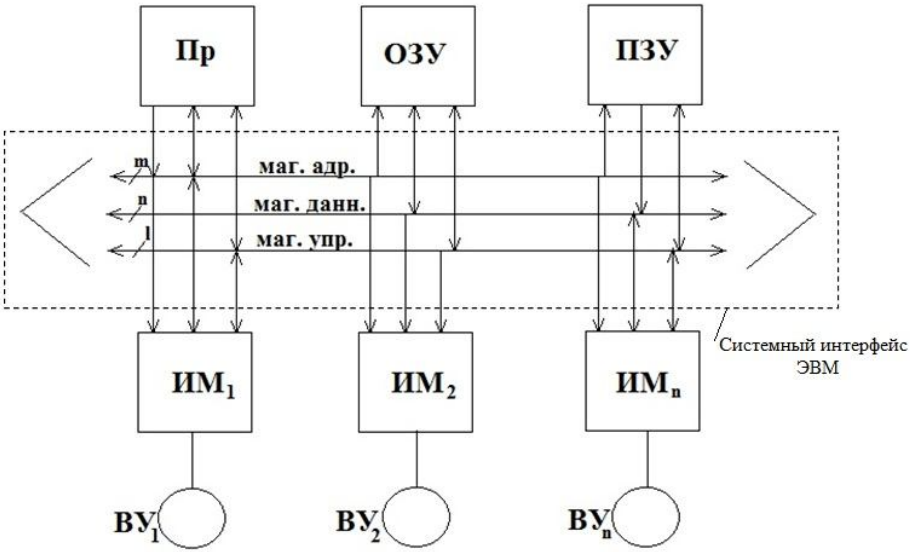


Рис. П6. Общая структурная схема аппаратных средств однопроцессорной ЭВМ

(регистр признаков результата выполненой предыдущей команды), декодер команд, устройство управления, синхронизации и согласования сигналов первичного, микропроцессорного, интерфейса с сигналами шин системного интерфейса ЭВМ. Применительно к типовым, например, IBM – совместимым ЭВМ, системный интерфейс может быть представлен несколькими уровнями обслуживания внешних устройств и блоков памяти, в зависимости от быстродействия [П4].

Из рис. П6 следует, что на аппаратном уровне доступ с ВУ в ЭВМ может осуществляться только через интерфейсные модули, которые по своей сложности представлены широким ассортиментом: от буферного регистра до микропроцессорного устройства управления (контроллера). Примером одного из таких ВУ служит внешняя память ЭВМ в виде накопителя на жёстком магнитном диске (НМД), где хранятся различными способами организованные программы и данные. Здесь уместно отметить о пользовательском интерфейсе ЭВМ, возможности которого могут быть расширены в зависимости от используемых ВУ, например, мультимедиа средств, ориентированных на интеллектуализацию интерфейса. В общем случае любой интерфейс (на аппаратном, программном или на пользовательском уровнях) можно представить в формализованном, теоретико-множественном виде:

$$Mc = < D, C >,$$

где D - множество параметров данных, С - множество управляющих параметров, Mc - сопрягающий элемент (*mating component*).

2.3. Система команд ЭВМ

Структура аппаратных средств ЭВМ чаще всего компонуется из фирменных микропроцессорных наборов(МПН), главными составляющими которых являются процессорный элемент, системный контроллер и различного типа интерфейсные модули [П2]. Приэтом процессорный элемент может иметь первичную (микропрцессорную) систему команд, либо систему микрокоманд в микропрограммируемых наборах. В настоящее время на рынок поставляются МПН многими фирмами-изготовителями, каждая из которых имеет свою специфику в организации системы команд (или микрокоманд). В то же время существует определённая традиционно сложившаяся общность в принципах организации системы команд микропроцессоров, обусловленных общими технологиями программирования на символических языках программирования низкого уровня (ассемблерах). Для начинающего пользователя достаточно иметь несколько примеров, чтобы приобрести элементы самостоятельности в практических применениях той или иной системы команд. Далее представлены общие сведения по организации и применению системы команд восьмиразрядного микропроцессора *Intel* 8080.

С полным описанием системы команд можно ознакомиться по фирменному руководству для пользователей. С сокращённым вариантом описания системы команд указанного микропроцессора можно ознакомиться по работе [П4].

Рассмотрим основные общие принципы построения системы команд, необходимые для пользователя-программиста.

Система характеризуется следующими форматами команд: однобайтный, двухбайтный и трёхбайтный. Форматы данных могут быть однобайтными и двухбайтными. Способами адресации данных являются: прямая - данные размещены в ОЗУ и их адреса данных формируются программным счётчиком, непосредственная адресация - данные размещены в формате текущей команды, прямая регистровая-данные находятся, либо в одном из восьми регистров *B,C,D,E,H,L,Ф,A* , либо в одной из пар регистров *B-C, D-E, H-L,* - косвенная регистровая - адреса формируются для ОЗУ парой регистров. Регистр *A* выполняет функции рабочего

регистра - в нём размещается результат выполненной текущей команды и его содержимое всегда участвует в выполняемой очередной арифметической, либо логической операциях. Кроме того, *A*-регистр используется в качестве буферного в операциях ввода-вывода с внешними устройствами. В микропроцессоре имеется флаговый Ф - регистр, где хранятся признаки условных переходов. Все указанные регистры имеют двоичные трёхразрядные, а для пар регистров - двухразрядные адреса, проставляемые в форматах команд. Нумерация регистров идёт от 000 до 111 с приращением на единицу при расположении их в следующем порядке *B , C, D, E, H, L, Ф, A*. Нумерация пар регистров идёт от 00 до 11 сответственно порядку их следования в указанной записи.

В системе команд по типу выполняемых операций можно выделить следующие подгруппы: арифметические, логические, передачи данных, условных переходов, ввода-вывода, преобразования указателя стека. Оставшиеся команды составляют подгруппу различного назначения. Смысловой, функциональный, оттенок каждого типа команды снабжён соответствующей символикой (оператором) с соответствующими параметрами, на основе которой создан свой язык программирования - ассемблер.

В качестве примера рассмотрим вариант программной реализации операции умножения двух целых восьмиразрядных двоичных чисел α, β без знака с построением блок-схемы вычислительного алгоритма (рис. П7) и его интерпретацией принятой символикой системы команд (табл. П4) микропроцессора *Intel* 8080 [П3]. Ввод множимого и множителя осуществляется с внешних устройств $ВУ_1$ и $ВУ_2$ с указанием их адресов.

Полученный результат в виде Σ-суммы частичных произведений выводится на $ВУ_3$ и $ВУ_4$. Перед интерпретацией вычислительного алгоритма символикой двоичных команд процессора для размещения данных выбраны соответствующие регистры: в *C*-регистре размещается множимое, *D*-регистр используется в качестве счётчика в организации цикла вычислений, а пара *H-L*-для размещения суммы частичных произведений (для вычислений использована арифметика удвоенной точности, где пара *H-L* исполняет функции аккумулятора).

В примере использован обычный способ умножения двоичных чисел столбиком, но с учётом требуемых сдвигов накапливаемой Σ-суммы частичных произведений. В блоках условных переходов 3 и 7 использованы сответственно флаг *C*-переполнение АЛУ и флаг *Z*-признак нуля в состоянии Сч, зафиксированного в регистре-аккумуляторе.

Текст программы с комментариями приведён в табл. П3.

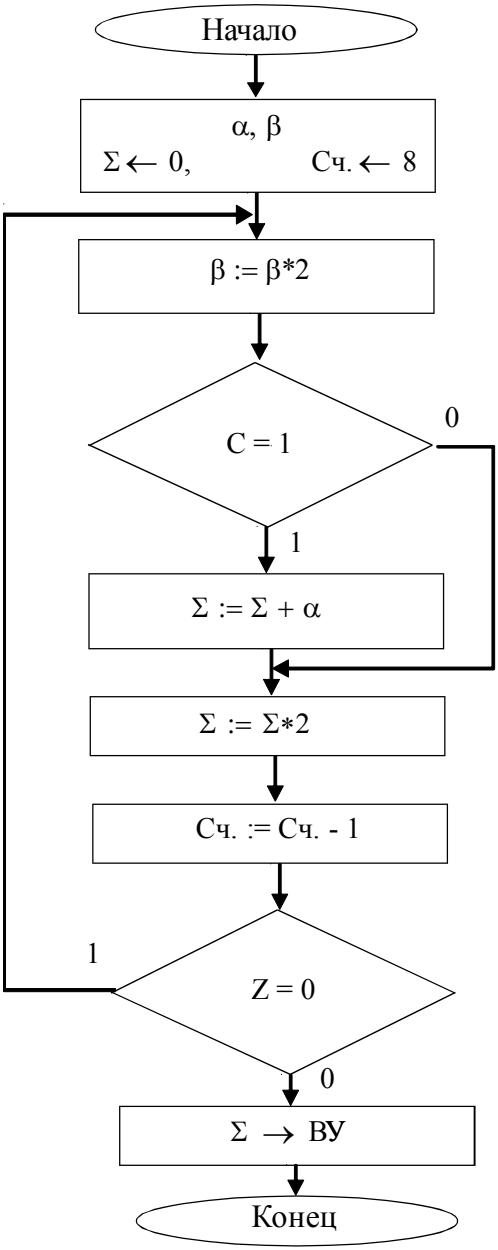


Рис. П7. Блок-схема вычислительного алгоритма умножения целых двоичных чисел без знака

Таблица П3

Текст программы с комментариями

Метка	Адрес ОЗУ	Мнемоника команды	Двоичный код команды	Комментарий
	*F*000 *F*001	*MVI B*,00	00000110 00000000	Обнуление содержимого *B*-рег.
	*F*002 *F*003	*IN 01*	11011011 00000001	Ввод множителя β в регистр *A* с ВУ2 с адресом 01
	*F*004	*MOV C,A*	01010111	Пересылка α из *A*-регистра в *C*-регистр
	*F*005 *F*006	*IN 02*	11011011 00000010	Ввод множителя β в регистр *A* с ВУ2 с адресом 02
	*F*007 *F*008 *F*009	*LXI H*,0000	00100001 00000000 00000000	Обнуление содержимого пары регистров *H, L*
	*F*00*A* *F*00B	*MVI D*,07	00001110 00000111	Занесение числа 8 в регистр *D*
*q*2	*F*00C	*ADD A*	10000111	Сдвиг множителя β влево
	*F*00D *F*00E *F*00F	*JNZ q*1	11010010 11110000 00010001	Переход на метку *q*1=*F*011, если флаг *C*=0
	*F*010	*DAD C*	00001001	$\sum = \sum + \alpha$
*q*1	*F*011	*DADH*	00101001	$\sum = 2 \cdot \sum$
	*F*012	*DCR D*	00000101	$Сч := Сч - 1$
	*F*013 *F*014 *F*015	*JNZ q*2	11000010 00000000 00001010	Переход по условию *Z*=0 на метку *q*2=*F*00*C*
	*F*016	*MOV A,H*	01111100	Пересылка старшего байта произведения в регистр *A*
	*F*017 *F*018	*OUT* ВУ3	11010011 00000011	Вывод старшего байта на ВУ3
	*F*019	*MOV A,L*	01111101	Пересылка младшего байта произведения в регистр *A*
	*F*01A *F*01B	*OUT* ВУ4	11010011 00000100	Вывод младшего байта на ВУ4

3. Алгоритмизация и программирование задач обработки данных в модулярной арифметике полей Галуа с быстрыми ортогональными преобразованиями

Рассмотренные ранее основные способы организации аппаратных средств типовых ЭВМ преимущественно ориентированы на использование машинной арифметики двоично-кодированных вещественных чисел. В то же время имеется множество вычислительных задач, решаемых с использованием модулярных арифметик конечных числовых полей (простых полей Галуа (*GF(p)*) и их расширений $GF(p^m)$. Областями применения, где решаются такие задачи, являются, например, компьютерные технологии реализации алгебраических методов помехоустойчивого кодирования и криптографии в телекоммуникационных системах. Одним из наиболее значимых отличий от ЭВМ данными в телекоммуникационных системах являются цифровые (кодированные) последовательности большой длины. Формами их одномерного представления могут быть: теоретико-множественная $\{s_n\}$, векторная и полиномиальная. Две последние формы непосредственно могут быть получены через элементы $\{sn\}$, которые принадлежат чаще всего полю $GF(2)$, в котором операции сложения и вычитания совпадают и отсутствуют нетривиальные умножения, т.е. используются операции по модулю два.

Рассмотрим теперь каким образом можно обобобщить алгоритмы умножения целых двоично-кодированных вещественных чисел с полиномиальным умножением в поле $GF(2)$, которые нашли широкое применение в алгоритмах кодирования и декодирования линейных циклических кодов.

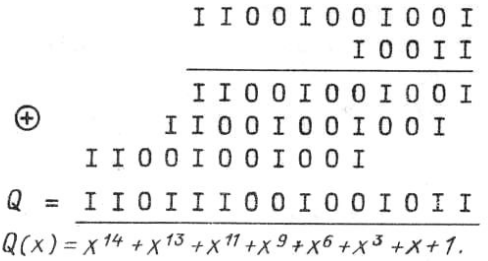
Одним из способов построения линейного двоичного несистематического циклического (n,k)- кода, где n-длина кода, к-количество информационных символов, является задание его в виде произведения многочленов над полем коэффициентов $GF(2)$

$$Q(x) = C(x) \cdot g(x),$$

где $C(x)= \sum_{0}^{k-1} c_i \cdot x^i$ - многочлен, сответствующий *к*-элементной двоично-кодируемой (информационной) последовательности, $g(x)$- неприводимый многочлен в поле $GF(2)$ степени n-k, равной величине избыточности кода. Многочлен $G(x)$ принято называть информационным, а $g(x)$- порождающим. Примечательной особенностью несистематических циклических кодов является то, что они наряду с защитой от помех могут быть исполь-

зованы для защиты от несанкционированного доступа. Например, для контроля целостности двоичных кодов программ и данных при борьбе с вирусами. Кроме того, циклические коды могут быть самосинхронизируемыми, что делает их широко используемыми в системах связи. В качестве примера рассмотрим алгоритм кодирования для циклического (15,11)-кода с порождающим многочленом $g(x)=x^4+x+1$ (здесь и далее будем записывать многочлены со старших степеней в порядке их уменьшения) и с информационным многочленом $C(x)= x^{10}+x^9+x^6+x^3+1$.

Представляя коэффициенты многочленов форме двоично-кодированных вещественных чисел, алгоритм кодирования можно записать [П5] в виде обычного алгоритма умножения целых чисел с тем отличием, что цифровые двоичные последовательности значений коэффициентов многочленов формально представлены в виде двоичных целых вещественных чисел без знака, но суммирование частичных произведений осуществляется в поле GF(2) (т.е. по *mod 2*- без переноса) и тем, что один из сомножителей является константой.



Примечательно, что алгоритм умножения многочленов над полем коэффициентов GF(2) можно использовать для циклических кодов, которые строятся на основе псевдослучайных последовательностей (ПСП) максимальной длины $n = 2^k$-1. Такие коды являются дуальными по отношению к ранее рассмотреным кодам $n=2^{n-k}$-1. Для этого в алгоритме умножения необходимо поменять старшие степени многочленов - информационного $C(x)$ с порождающим $g(x)$. В нашем примере порождающим выберем многочлен

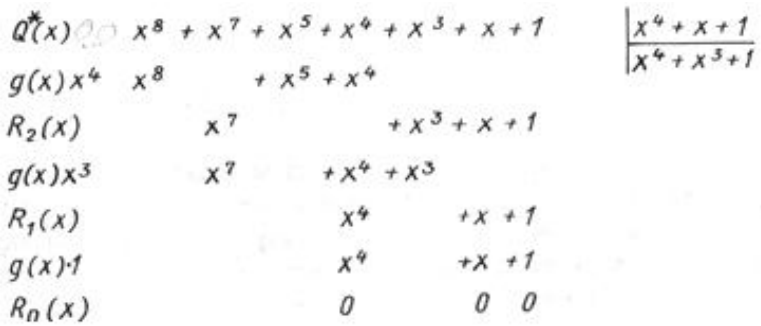
$$g(x)=x^{15}-1/x^4+x+1 = x^{11}+x^8+x^7+x^5+x^3+x+1,$$

инвертированную последовательность значений коэффициентов которого можно задать в виде двоичной псевдопоследовательности:

$$П_{15}=1101010110\ 010000.$$

Данную ПСП можно взять в качестве образующего вектора-строки циклической матрицы, в которой будут представлены все 2^4-1=15 разрешённых кодовых слов данного самосинхронизируемого циклического (15,4)-кода.

Рассмотрим теперь алгоритм деления многочленов над полем коэффициентов *GF*(2) применительно к декодированию циклических линейных кодов [П5]. Деление полиномов над полем коэффициентов *GF*(2) продемонстрируем для того же (15,11)-кода. Пусть принятому, возможно искажённому, кодовому слову соответствует в традиционной полиномиальной форме алгоритм деления можно записать в виде:



Принятый многочлен *Q(x)* делится нацело на многочлен *g(x),* что соответствует случаю, когда принятое кодовое слово относится к множеству разрешённых кодовых слов рассматриваемого циклического кода. Алгоритм деления многочлена *Q(x) на g(x)* можно представить в формальном числовом виде

```
0 0 I I 0 I I I 0 I I | I 0 0 I I
    I 0 0 I I 0 0 0 0 | I I 0 0 I
  ---------------------
    0 I 0 0 0 I 0 I I
      I 0 0 I I 0 0 0
  -----------------
    0 0 0 0 I 0 0 I I
          I 0 I I
          0 0 0
            0
```

Приведённый вариант блок-схемы вычислительного алгоритма [П5] деления полиномов с коэффициентами в поле *GF*(2) мало отличается от блок-схем алгоритмов умножения целых двоично-кодированных чисел.

4. Аппаратные средства реализации функций умножения полиномов в поле *GF*(2)

Приведённые выше примеры демонстрируют возможности использования одинаковых или очень схожих вычислительных алгоритмов для программной реализации в системе команд задач синтеза и анализа цифровых последовательностей, элементы которых могут принадлежать, как полю R- вещественных чисел, так и полю *GF*(2).

Рассмотрим теперь возможность дальнейшего обобщения подходов к реализации сформулированных задач на уровне аппаратных средств. Для этого нам необходимо прежде всего вернуться к формальному логическому синтезу схем дискретных устройств, представленных на функционально-модульном уровне с использованием моделей конечных дискретных автоматов (рис.П3), потому что структурными составляющими устройств будут использованы сдвиговые регистры, необходимые для хранения чисел - элементов обрабатываемых последовательностей и арифметических устройств-сумматоров и умножителей. Схемы таких устройств, названных цифровыми линейными фильтрами, нашли широкое применение в виде специализированных вычислительных устройств. Учитывая то, что длины обрабатываемых последовательностей могут измеряться сотнями, тысячами, иногда миллионами элементов, то для таких случаев универсальный аппарат формального логического синтеза явно не пригоден.

Пользуясь материалами раздела 2, приведём две схемы:

- схему умножения многочлена на многочлен

$$A(x) = \sum_{i=0}^{k} a_i \cdot x^i \text{ на } g(x) = \sum_{i=0}^{L} g_i \cdot x^i \text{ (рис.П8),}$$

- схему деления многочлена на многочлен

$$V(x) = \sum_{i=0}^{n-1} v_i \, x^i \text{ на } g(x) \text{ (рис.П9).}$$

Многочлены здесь представлены в общем виде и предполагается, что параметры k, n, L удовлетворяют условию $k, n >> L$.

Для арифметики поля *GF*(2) в качестве элементов задержки используются двоичные *D*-триггеры, а вместо $\otimes$ умножителя присутствует (или отсутствует) электрическое соединение между триггерами и $\oplus$ сумматорами по модулю два. Таким образом, можно заключить ,что приведённая схема может обладать значительно более высоким быстродействием по

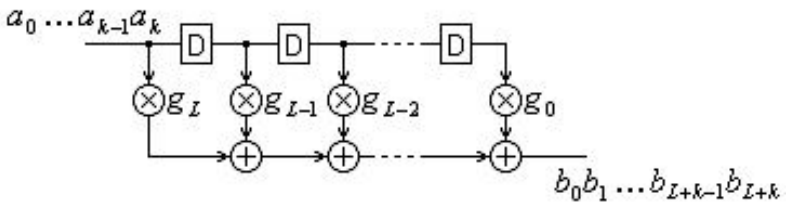


Рис. П8. Схема умножения многочленов в поле коэффициентов GF(2)

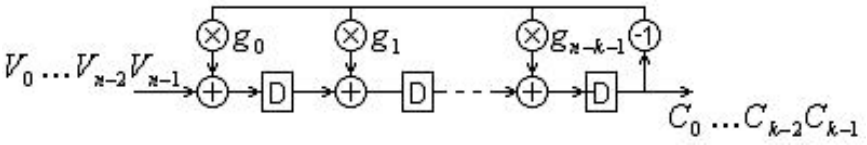


Рис. П9. Схема деления многочленов в поле коэффициентов GF(2)

сравнению с программной реализацией. В случае реализации алгоритмов в арифметике поля вещественных чисел приведённый вариант схем значительно усложняется по своей структуре и практического применения не находит.

Рассмотренным вариантам схем цифровых фильтров обычно сопоставляют их математические модели в виде разностных уравнений или уравнений цифровых линейных свёрток, либо рекуррентных выражений, которые являются наиболее пригодными для программной реализации на языках высокого уровня.

5. Рекурсивные вычисления быстрого преобразования Адамара

Пусть задана известная блочно-матричная факторизованная форма быстрого преобразования Адамара

$$A_N^{(\phi)} = diag\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \times diag\begin{pmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{pmatrix} \times \ldots \times diag\begin{pmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{pmatrix}, \qquad (П4)$$

которая может быть получена из блочно - рекурсивной формы матрицы Адамара

$$A_N = \begin{pmatrix} A_{N/2} & A_{N/2} \\ A_{N/2} & -A_{N/2} \end{pmatrix},$$

которая в свою очередь может быть представленав кронекеровским произведением $A_N = A_2 \otimes A_{N/2}$, где $N = 2^n$. В соответствии с факторизацией (П4) строится ориентированный граф быстрого преобразования (рис. П10). На основе графа, значительно проще определить общую закономерность управления в вычислительном алгоритме, ориертированном на интерпретацию операторами выбранного языка программирования.

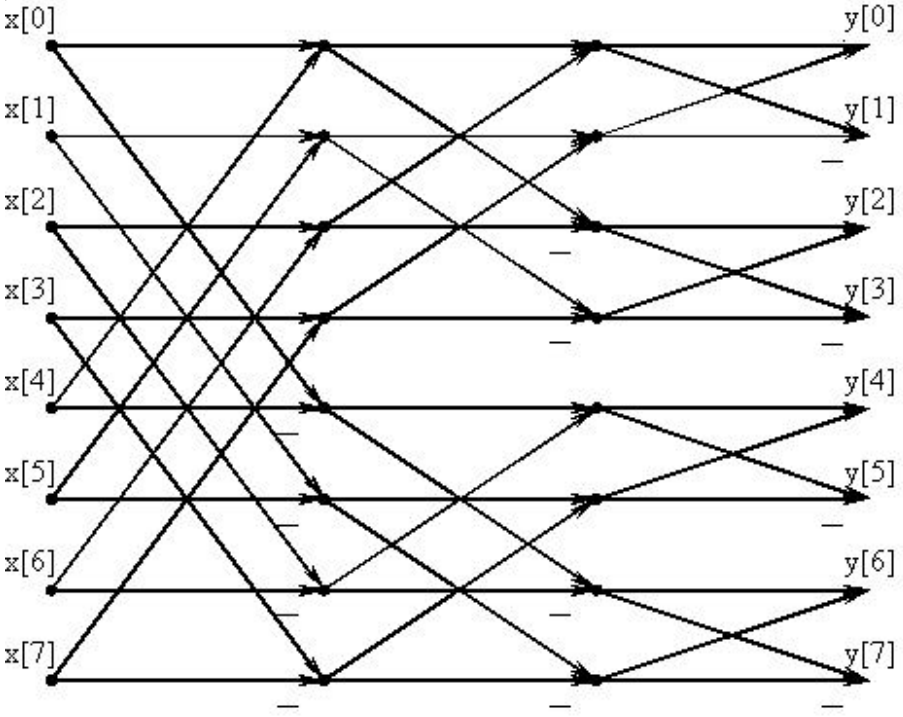


Рис. П10. Сигнальный граф БПА для N=8

В приведённой блок-схеме (рис. П11) вычислительного алгоритма рекурсивные вычисления объединены в процедуру, обладающую возможностью сопряжения выхода со входом.

5. Архитектура сигнального процессора быстрых Фурье, Харли, Уолша и косинусного преобразований

Рассмотрим одноконвейерную микропрограммируемую архитектуру сигнального процессора, ориентированную на реализацию одномерных

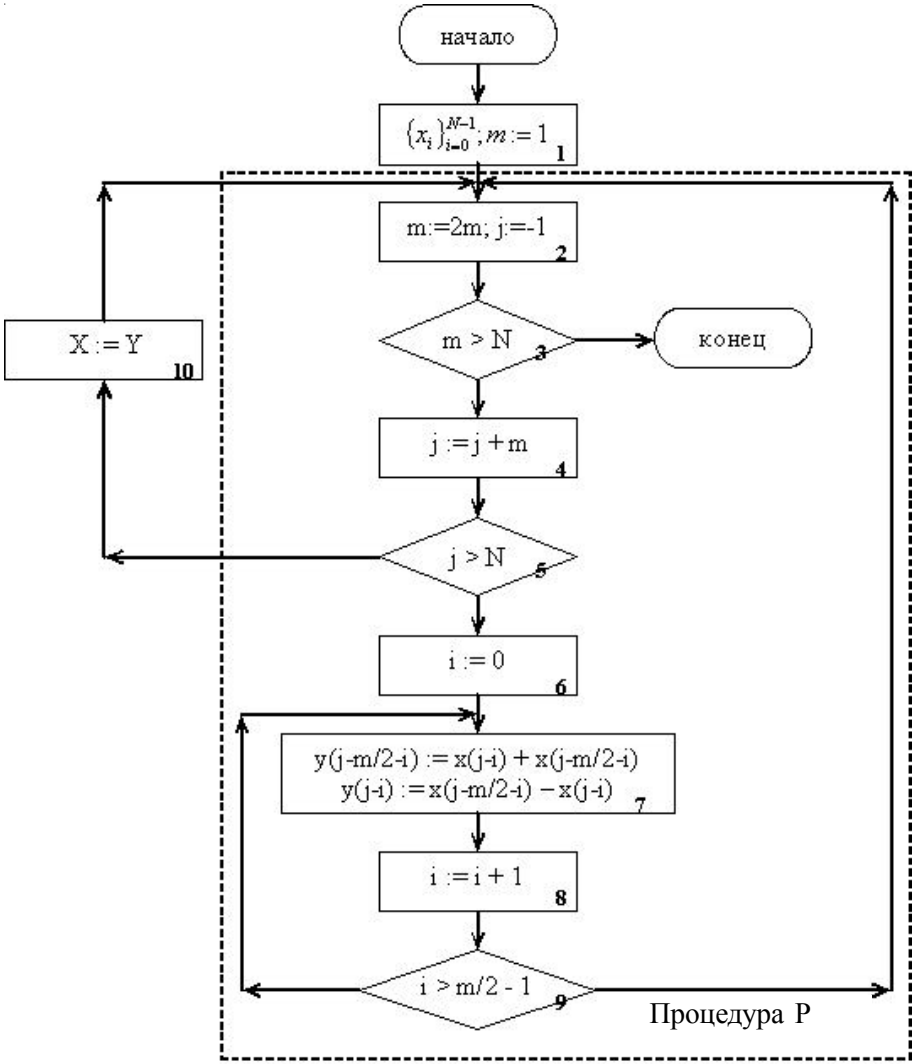


Рис. П11. Блок-схема вычислительного алгоритма БПА для программирования на ЯВУ

многомерных гнездовых, псевдогнездовых и простых множителей алгоритмов БПФ и БПХ, а также алгоритмов БКП, спецификой которых является, небольшое число разнотипных нетривиальных множителей (например, для длины N не более 120 количество разнотипных весовых множителей может составлять не более шести) и весовые множители пред-

ставлены однодиагональными матрицами. Причем для БПФ в поле *C* элементами таких матриц являются либо вещественные, либо мнимые числа, т.е. не требуются общие комплексные умножения, содержащие четыре вещественных умножения.

Основными функциональными блоками устройства (Рис.П12) являются конвейерное арифметическое устройство (АУ), включающее регистры (Р1 ÷ Р5), сумматор-вычитатель (С-В) и умножитель (УМН) параллельного типа, два блока оперативной памяти данных ($ОЗУ_1$ и $ОЗУ_2$), образующее с АУ общий конвейер, и блок микропрограммного управления с коммутаторами.

Особенностью структуры является то, что микрокоманды являются одноформатными с прямой адресацией данных и без возможности условных и безусловных переходов внутри микропрограммы.

В микропрограммной памяти (ПЗУмк) могут храниться одна или несколько микропрограмм, соответствующие определенным быстрым алгоритмам и (или) различной длине отдельного преобразования. Программирование типа и длины преобразования осуществляется двоичным кодом КП, подаваемым на адресные входы ПЗУмк.

В процессоре предусмотрены аппаратные средства для обмена данными с внешним устройством, например, с процессором основной типовой или специализированной ЭВМ. Причём на время обмена данными сигнальный процессор является пассивным устройством, т.е. выставляет нейтральное состояние на адресные или управляющие входы блоков $ОЗУ_1$ и $ОЗУ_2$.

Устройство работает следующим образом: начальной установке триггера Т соответствует состояние, определяемое состоянием его выхода, запрещающее счёт адресного счетчика АСЧ, который установлен в нулевое состояние. Код длины и типа преобразования устанавливается на заданную длину и тип преобразования, для каждого из которых в блоке постоянной памяти ПЗУмк хранятся микропрограммы; по приходу сигнала РСч “Разрешение счёта” триггер Т изменяет свое начальное состояние и счетчик формирует адрес первой микрокоманды из микропрограммы, соответствующей заданному быстрому алгоритму. Код первой микрокоманды записывается на первом такте в регистр микрокоманд РгМК.

На представленную схему процессора авторами получено изобретение [П6].

Формат микрокоманды выбран таким, что в нём присутствует поле адреса первого блока памяти $ОЗУ_1$ (разряды a_4), поле адреса второго $ОЗУ_2$

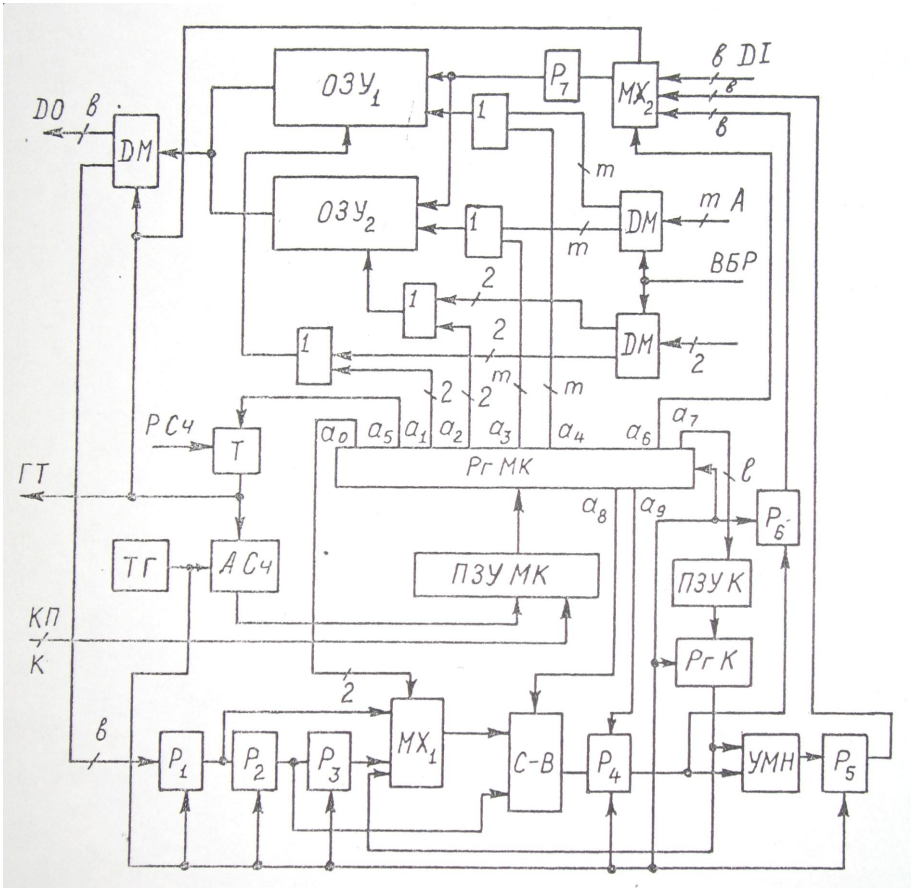


Рис. П12. Конвейерная архитектура сигнального процессора БПФ,БПХ ,БКП и БПА

блока памяти (разряды a_3), поле сигналов управления первого и второго блоков ОЗУ – разряды a_1 и a_2. Поле a_7 соответствует адресу второго блока постоянной памяти ПЗУк, где хранятся весовые коэффициенты.

Разряды микрокоманды a_0 через один из входов коммутатора $МХ_1$ управляют коммутацией операндов, размещенных в регистрах Р1, Р2, Р3, на входы сумматора-вычитателя С-В. Полем a_8 осуществляется управление сумматором-вычитателем С-В и коммутатором $МХ_1$. Разрядом a_9 осуществляется управление сдвигами в случае вычисления коэффициентов обратного преобразования.

В качестве примеров быстрых ортогональных преобразований, на которые с большим предпочтением ориентирована данная архитектура

сигнального процессора, следует указать однодиагональные алгоритмы БКП (Рис.5.1), гнездовые, псевдо-гнездовые и простых множителей алгоритмы БПФ (Рис. П13) и преобразования Хартли [8], а также быстрые алгоритмы преобразований Уолша и Хаара в совмещённых базисах.

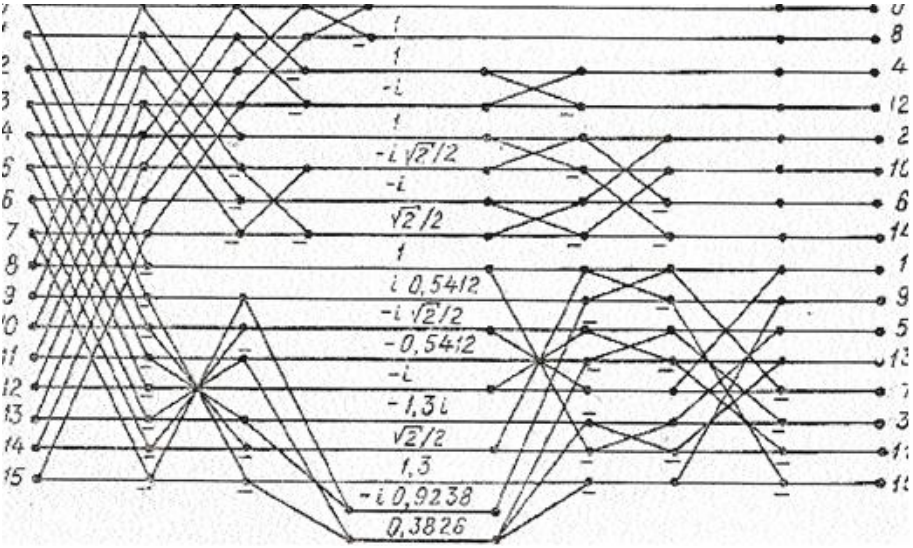


Рис. П13. Граф гнездового алгоритма БПФ, N=16

Литература

П1. Титце У., Шенк К. Полупроводниковая схемотехника. М., Мир, 2009.(справочник)
www.kodges.ru/2007/04/11/titce_u_shenk_k_poluprovodnikovaja_skhem.html.
П2. Костров Б.В., Ручкин В.Н. Микропроцессорные системы и микроконтроллеры. М., Техбук, 2007,(учебное пособие).
П3. А. Беркс, Г. Голдстейн, Дж. Нейман. Предварительное рассмотрение логических конструкций электронного вычислительного устройства, М., Мир , Кибернетический сборник № 9, 1964, с.7-67.
Burks A. W.,Goldstein H. H., von Neumann J., Preliminary discussion of the logical design of an electronic computing instrument, von Neumann J., Collected works, v. V, Pergamon Press, Oxford — Lnd. — N. Y. —Paris, 1963, pp. 34—79. Сокращенный вариант статьи опубликован в журнале Datamation, 8, № 9-10 (1962).

П4. Юэн Ч., Бичем К., Робинсон Дж. Микропроцессорные системы и их применение при обработке сигналов. М., Радио и связь, 1986.
П5. Гагарин Ю.И., Скорубский В.И., Смирнов Ю.М. Микропроцессор К584. Организация, функционирование и применение. М., Изд. “ЭКОС” Министерства промышленности средств связи СССР, 1982.
П6. SU1606977 (A1), Gagarin Yurij, Gagarin Konstantin «Device for making fast orthogonal transforms»

СОДЕРЖАНИЕ

Contents

Учебное издание

Юрий Иванович Гагарин,
Константин Юрьевич Гагарин

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Учебное пособие

Редактор *И.Г. Максимова*
Компьютерная вёрстка *К.П. Ерёмин*

ЛР № 020309 от 30.12.96

Подписано в печать 20.12.12. Формат 60x90 1/16. Гарнитура “Таймс”.
Печать цифровая. Усл. печ. л. 6,5. Тираж 200 экз. Заказ № 142.
РГГМУ, 195196, Санкт-Петербург, Малоохтинский пр. 98.
Отпечатано в ЦОП РГГМУ