Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

В. М. Чертков
Р. П. Богуш
С. В. Мальцев

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Учебно-методический комплекс для студентов
специальности 1-39 01 01 «Радиотехника»

Новополоцк
ПГУ
2013

УДК 621.37(075.8)
ББК 32.84я73
Ч50

Рекомендовано к изданию методической комиссией
радиотехнического факультета в качестве учебно-методического комплекса
(протокол № 5 от 12.04.2011)

РЕЦЕНЗЕНТЫ:

ведущий инженер группы информации и ЭВМ
РУП «Полоцкий ЦСМС» И. И. ВАРГАСОВА;
канд. техн. наук, доц., зав. каф. конструирования РЭС
УО «ПГУ» Ю. Г. ГРОЗБЕРГ

Чертков, В. М.

Ч50 Цифровая обработка сигналов : учеб.-метод. комплекс для студентов специальности 1-39 01 01 «Радиотехника» / В. М. Чертков, Р. П. Богуш, С. В. Мальцев. – Новополоцк : ПГУ, 2013. – 360 с.

ISBN 978-985-531-372-5.

Рассмотрены общие принципы и математические модели преобразования сигналов при цифровой обработке, базовые алгоритмы цифровой фильтрации сигналов, методы синтеза, программной реализации и моделирования цифровых фильтров, оценки и обеспечения их точности, методы спектрально-корреляционного анализа сигналов на основе дискретного преобразования Фурье, алгоритмы быстрого преобразования Фурье, методы переноса спектров, формирования и обработки комплексных дискретных сигналов, передискретизации сигналов в процессе их обработки, а также реализации систем цифровой обработки сигналов на основе аппаратных и аппаратно-программных средств.

Предназначен для студентов радиотехнического факультета, обучающихся по специальности «Радиотехника» и изучающих дисциплину «Цифровая обработка сигналов».

УДК 621.37(075.8)
ББК 32.84я73

ISBN 978-985-531-372-5

© Чертков В. М., Богуш Р. П., Мальцев С. В., 2013
© УО «Полоцкий государственный университет», 2013

Введение в курс «Цифровая обработка сигналов»

Цифровая обработка сигналов (ЦОС) – это область науки и техники, в которой изучаются общие для различных технических приложений принципы, методы и алгоритмы обработки сигналов и изображений средствами цифровой вычислительной техники [1].

Под сигналом понимают физическую величину, изменяющуюся во времени, содержащую некоторую информацию и пригодную для обработки.

Главная цель обработки физических сигналов заключается в необходимости получения содержащейся в них информации, которая обычно присутствует в амплитуде сигнала, в частоте или в спектральном составе, в фазе или в относительных временных зависимостях нескольких сигналов [1].

Цифровая информация, представляющая индивидуальные звуковые каналы, мультиплексируется во времени (многоканальный доступ с временным разделением, TDMA) и передается по последовательной цифровой линии связи.

Еще одна причина обработки сигналов заключается в сжатии полосы частот сигнала (без существенной потери информации) с последующим форматированием и передачей информации на пониженных скоростях, что позволяет сузить требуемую полосу пропускания канала.

В высокоскоростных модемах широко используются алгоритмы устранения избыточности данных, так же как и в цифровых системах мобильной связи, системах записи звука и видео MPEG, в телевидении высокой четкости (HDTV).

Промышленные системы сбора данных и системы управления используют информацию, полученную от датчиков, для выработки соответствующих сигналов обратной связи, которые, в свою очередь, непосредственно управляют процессом.

В некоторых случаях в сигнале, содержащем информацию, присутствует шум, и основной целью является восстановление сигнала. Такие методы, как фильтрация, автокорреляция, свертка и т.д. часто используются для выполнения этой задачи и в аналоговой, и в цифровой областях.

Сигналы могут быть обработаны с использованием аналоговых методов (ASP), цифровых методов (DSP) или комбинации аналоговых и цифровых методов (MSP).

Преимущества ЦОС [2]:

- высокая гибкость, т.е. системы ЦОС можно перепрограммировать на выполнение различных функций без изменения оборудования;
- гарантированная точность, которая определяется числом используемых битов;

- совершенная воспроизводимость, поскольку отсутствуют отклонения, обусловленные устойчивостью отдельных составляющих;
- высокая надежность, малые размеры, низкое энергопотребление.

К основному недостатку следует отнести высокие временные затраты при обработке широкополосных сигналов.

В настоящее время ЦОС широко используется для обработки речи, например, распознавание и синтез речи, озвучивание текста; обработка изображений, например, улучшение и синтез изображений, машинное зрение и распознавание образов, обработка динамических изображений; в инструментальных средствах, например, спектральный анализ, контроль положения и скорости, сжатие информации и др.

Цель преподавания дисциплины

Теоретическая и практическая подготовка студентов радиотехнической специальности по основным направлениям цифровой обработки сигналов: цифровой фильтрации, спектрально-корреляционному анализу и оценке параметров, многомерной и адаптивной обработке, алгоритмическому обеспечению.

Задачи изучения дисциплины:

- изучение особенностей дискретизации и квантования сигналов;
- изучение быстрых алгоритмов для цифровой фильтрации, корреляционной обработки и спектрального анализа;
- изучение методов синтеза цифровых фильтров и оценки их параметров.

В результате изучения курса «Прикладная теория кодирования» студент **должен знать:**

– принципы и особенности дискретизации сигналов в радиоэлектронных системах;

– преобразования сигналов при цифровой обработке и связанные с ними искажения и погрешности;

– алгоритмы цифровой фильтрации, спектрального анализа и оценки параметров сигнала;

– методы синтеза систем цифровой обработки сигналов и оценки эффективности их работы;

– основные применения ЦОС;

уметь:

– обоснованно оценить необходимые параметры дискретизации и квантования;

– выбирать наиболее эффективный алгоритм обработки;

– выполнять синтез цифрового фильтра;

– вычислять основные преобразование и базовые модели системы ЦОС;

– моделировать алгоритмы обработки на ЭВМ в средах общего и специализированного математического программного обеспечения (MathCAD, MatLAB и др.);

Подготовка специалиста должна обеспечивать формирование у студентов следующих групп компетенций:

1) *академические компетенции*:

- владение базовыми научно-теоретическими знаниями;
- владение исследовательскими навыками (знания об основных методах проектирования и оптимизаций цифровых устройств);
- умение работать самостоятельно;
- владение междисциплинарным подходом при решении инженерных проблем;

2) *социально-личностные компетенции:*

- быть способным к социальному взаимодействию;
- уметь работать в коллективе;

3) *профессиональные компетенции*:

- применять прогрессивные энергоэффективные и ресурсосберегающие технологии;
- внедрять современные технологии;
- осуществлять выбор перспективных элементов и ресурсосберегающих схем;
- определять актуальные направления научных исследований в области электронных цепей дискретного действия.

Формирование и развитие социально-профессиональной компетентности, позволяющей сочетать академические, профессиональные, социально-личностные компетенции для решения задач в сфере профессиональной и социальной деятельности.

Методы (технологии) обучения

Основными методами (технологиями) обучения, адекватно отвечающими целям изучения данной дисциплины, являются:

– проблемное обучение (проблемное изложение, частично – поисковый и исследовательский методы);

– технология обучения как учебное исследование;

– проектные технологии;

– метод кейсов (анализ ситуации);

– коммуникативные технологии (дискуссия, мозговой штурм, учебные дебаты, круглый стол и т.д.);

– использование учебно-методического комплекса по дисциплине.

Для управления учебным процессом и организации контрольно-оценочной деятельности рекомендуется использовать рейтинговые, кредитно-модульные системы оценки учебной и исследовательской деятельности студентов.

Для контроля качества образования по дисциплине рекомендуется использовать следующие **средства диагностики:**

– устный, письменный опрос;

– коллоквиумы;

– защита лабораторных работ;

– экзамен.

Обучение дисциплине «Прикладная теория кодирования» предполагает реализацию следующих видов самостоятельной работы студентов:

- подготовка к аудиторному выполнению практических работ (предварительное знакомство с методическими указаниями, программным обеспечением, вариантом индивидуального задания по работе);

- подготовка к защите практических работ (оформление отчета по индивидуальному варианту задания по работе, подготовка к презентации результатов работы);

- изучение основной и дополнительной литературы в процессе подготовки к анализу и решению проблемных задач;

- подготовка к промежуточной и итоговой диагностике компетенции. Промежуточная (аттестационная) диагностика компетенции студентов осуществляется на основании письменного теста, проводимого на момент аттестации. Итоговая (зачетная, а для студентов заочного отделения – экзаменационная) диагностика компетенции студентов проводится в виде письменной работы.

Курсовая работа и ее характеристика

Курсовая работа выполняется студентами дневной формы обучения в 6-м семестре.

Цель курсовой работы – систематизация и закрепление теоретических знаний студентов по основным разделам курса, углубленное изучение методов цифровой обработки сигналов, получение практических навыков по моделированию и разработке алгоритмов цифровой обработки.

Примерное количество часов на выполнение – 40.

Темы курсовых работ посвящены разработке специальных алгоритмов цифровой обработки и их аппаратной реализации для корреляционной обработки, цифровой фильтрации и спектрального анализа.

Организация выполнения и защита работы

Задание на работу выдается на первой лекции по курсу. В задании указывается дата защиты курсовой работы. В качестве дня защиты может быть установлена любая дата не позднее календарного начала сессии. Консультации по работе проводятся согласно расписанию в течение всего срока выполнения до даты завершения работы. Работа представляется к защите в виде распечатки. Полный перечень материалов по каждому виду указан в задании. Защита курсовой работы проводится индивидуально. В процессе защиты путем постановки трех контрольных вопросов-заданий проверяется самостоятельность выполнения работы.

Общее количество часов, отводимых на изучение дисциплины, составляет 232 часов, из них 102 аудиторных часа. Примерное распределение по видам занятий: 52 часа лекций, 16 часов практических занятий и 34 часа лабораторных занятий.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

№ п/п	Название темы	Содержание	Объем в часах
1	2	3	4
1	Введение	Содержание дисциплины и ее взаимосвязь с другими дисциплинами	**2**
2	Раздел 1	Дискретные и цифровые сигналы и системы	**6**
3		Дискретизация и квантование сигналов	2
4		Представление сигналов функциональными рядами. Математические модели дискретной и цифровой систем обработки сигналов	2
5		Спектры дискретных сигналов. Технические показатели эффективности ЦОС. Краткий обзор алгоритмов цифровой обработки стационарных сигналов. Эффективность алгоритмов и оценка их вычислительной сложности	2
6	Раздел 2	Преобразование сигналов в цифровых системах	**16**
7		Z-преобразование и преобразование Фурье	2
8		Дискретное преобразование Фурье и его свойства	2
9		Быстрое преобразование Фурье	2
10		Дискретные преобразования в поле вещественных чисел	2
11		Дискретные преобразования в поле целых чисел	2
12		Дискретная свертка и корреляция	6

Окончание табл.

1	2	3	4
13	Раздел 3	Базовые модели систем ЦОС	**16**
14		Методы и модели ЦОС. Математическое описание цифровых фильтров. Типовые цифровые фильтры	2
15		Структуры цифровых фильтров	2
16		Частотные характеристики цифровых фильтров. Синтез цифровых фильтров	2
17		Специализированные цифровые фильтры	2
18		Непараметрические методы спектрального и корреляционного анализа	4
19		Параметрические методы спектрального анализа, основанные на моделях исследуемых процессов	2
20		Фильтры оптимальной обработки цифровых сигналов	2
21	Раздел 4	Аппаратные средства реализации алгоритмов цифровой обработки сигналов	**4**
22		Построение последовательных, последовательно-параллельных и конвейерных процессоров цифровой обработки сигналов	2
23		Современные цифровые сигнальные процессора DSP 56xxx, ADSP_21xxx, TMS320Cxx	2
24	Раздел 5	Применение цифровой обработки сигналов в радиоэлектронных системах	**6**
25		Применение цифровой обработки сигналов в задачах радиолокации и навигации	2
26		Цифровая спектрально-корреляционная обработка сложных сигналов	2
27		Адаптивная цифровая фильтрация сигналов	2
28	Заключение	Тенденции, перспективы и направления применения цифровой обработки сигналов в радиоэлектронных системах	**2**
		Всего	52

МОДУЛЬ 1. ДИСКРЕТНЫЕ И ЦИФРОВЫЕ СИГНАЛЫ И СИСТЕМЫ

1.1. Дискретизация и квантование сигналов

1.1.1. Условия выбора частоты дискретизации

Сигнал может быть определен как функция, переносящая информацию о состоянии или поведении физической системы. Сигнал может принимать форму колебаний, зависящих от времени или от пространственных координат. Математически сигналы представляются в виде функций одной или более независимых переменных. Обычно при математическом представлении сигнала независимой переменной считают время.

Независимая переменная в математическом представлении сигнала может быть как непрерывной, так и дискретной. Сигналы в непрерывном времени определяются на множестве моментов времени и, следовательно, представляются как функции от непрерывной переменной. Дискретные сигналы определяются в дискретные моменты времени и представляются последовательностями чисел [3].

Для преобразования аналогового сигнала в цифровой его последовательно подвергают трем преобразованиям: дискретизации (по времени), квантованию (по уровням) и кодированию. Устройства преобразования аналогового сигнала в цифровой и обратно называются аналого-цифровыми (АЦП) и цифроаналоговыми (ЦАП) преобразователями. Дискретизация – представление непрерывного по времени аналогового сигнала дискретным. Осуществляется на основе теоремы Котельникова – Найквиста: непрерывный по времени сигнал с ограниченным высшей частотой $f_в$ спектром (полоса 0– $f_в$) может быть представлен последовательностью импульсных отсчетов (выборок), величина которых равна или пропорциональна мгновенным значениям сигнала в соответствующие моменты времени, причем частота дискретизации (отсчетов) $f_Д$ должна удовлетворять требованию

$$f_Д \geq 2 f_в. \qquad (1.1)$$

Следовательно, на входе АЦП должен быть установлен ФНЧ с частотой среза $f_{ср}$, равной $f_в$..

Дискретные сигналы (последовательности) часто изображаются графически так, как это показано на рис. 1.1. Хотя абсцисса изображена в виде непрерывной линии, важно сознавать, что $x(n)$ определена только

для целых значений n. Неправильно полагать, что $x(n)$ равна нулю для нецелых n. Просто $x(n)$ не определена для дискретного сигнала нецелых значений n.

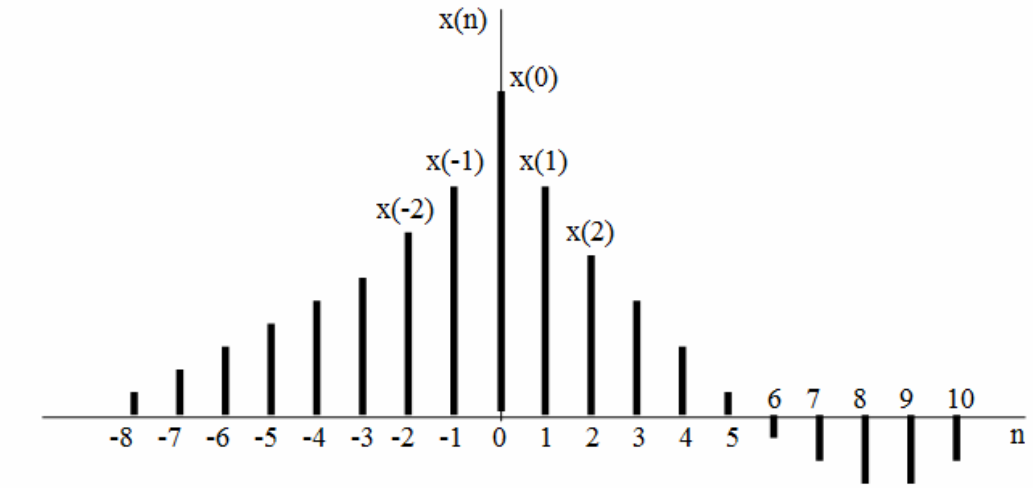


Рис. 1.1. Графическое представление дискретного сигнала

Некоторые примеры последовательностей, играющих важную роль при дискретной обработке, показаны на рис. 1.2.

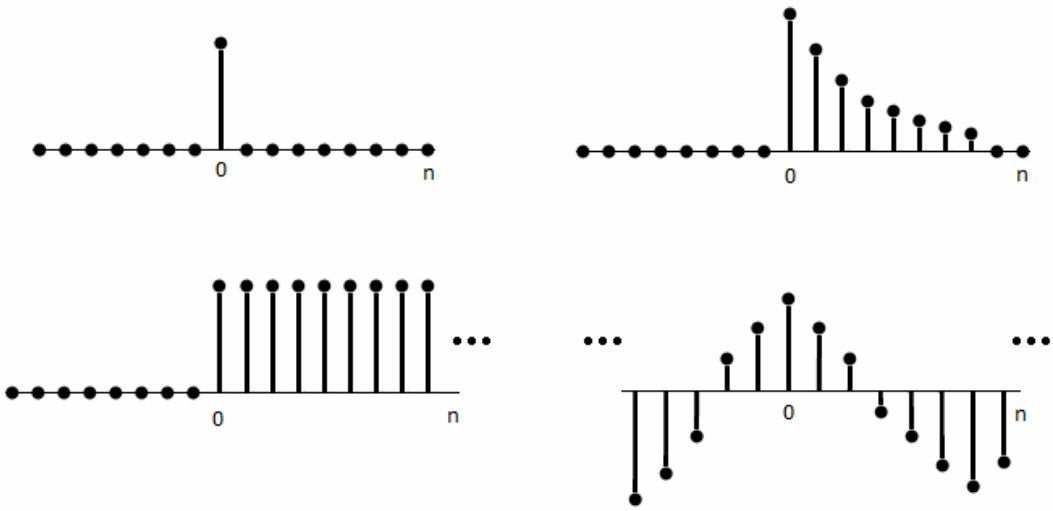


Рис. 1.2. Примеры последовательностей:
а – единичный импульс; *б* – единичная ступенчатая последовательность;
в – действительная экспоненциальная последовательность;
г – синусоидальная последовательность

Единичный импульс $\delta(n)$ определяется как последовательность со значениями

$$\begin{cases} \delta(n) = 0, n \neq 0; \\ \delta(n) = 1, n = 0. \end{cases}$$

Единичный импульс играет для дискретных сигналов и систем ту же роль, какую играет дельта-функция для аналоговых сигналов и систем. Для удобства единичный импульс часто называется просто импульсом. Важно отметить, что с единичным импульсом не связаны те математические затруднения, которые встречаются при использовании дельта-функции. Его определение просто и точно.

Единичная ступенчатая последовательность $u(n)$ имеет значение $u(n) = 1, n \geq 0$ и связана с единичным импульсом соотношением

$$u(n) = \sum_{k=-\infty}^{n} \delta(k). \qquad (1.2)$$

Соответственно, единичный импульс связан с единичной ступенчатой последовательностью соотношением

$$\delta(n) = u(n) - u(n-1). \qquad (1.3)$$

1.1.2. Квантование сигнала

При квантовании по уровню бесконечное множество возможных значений дискретного сигнала $x(n)$ в заданном максимальном диапазоне его изменения $D_{x\max} = (x_{\max} - x_{\min})$ замещается конечным числом уровней квантования m дискретного квантованного сигнала $x_{кв}(n)$. С одним из них в соответствии с определенным правилом или алгоритмом и отождествляется точное мгновенное значение дискретного сигнала $x(n)$ (рис. 1.3). Интервал между уровнями квантования называется шагом квантования по уровню $Q = D_{x\max}/m$, $e(n)$ – погрешность квантования.

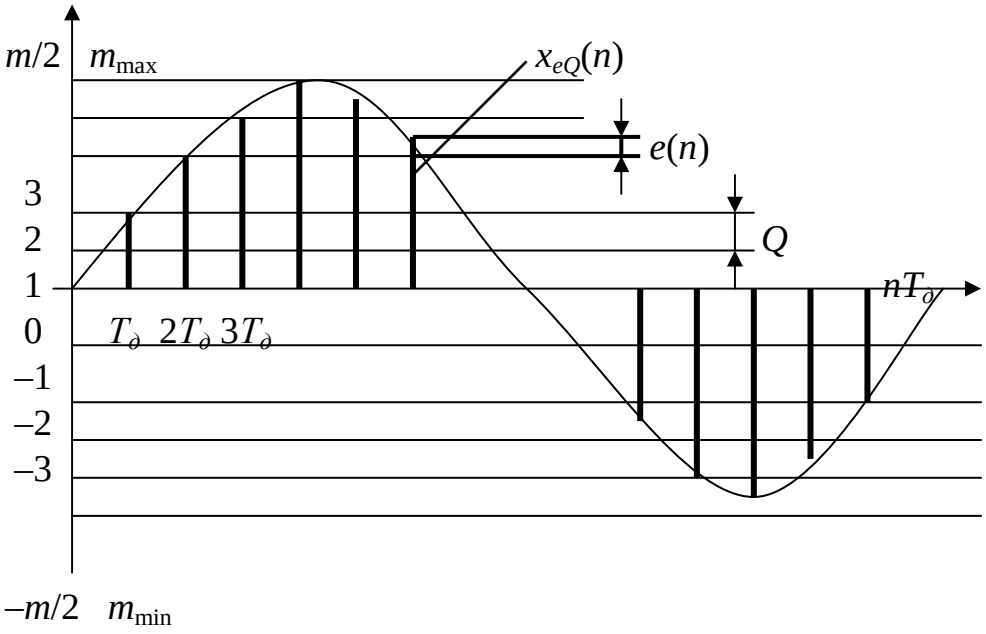


Рис. 1.3. Квантование сигнала

Квантование возможно с усечением и с округлением. Квантованный дискретный сигнал $x_{кв}(n)$ определяется при этом как

$$x_{кв}(n) = \left[x(n)/Q \right]_{ЦЧ} Q = iQ \qquad (1.4)$$

при усечении и

$$x_{кв}(n) = \left[x(n)/Q + 0{,}5 \right]_{ЦЧ} Q = iQ \qquad (1.5)$$

при округлении, где $i = [...]_{ЦЧ}$ – это целая часть отношения, заключенного в прямые скобки и соответствующая номеру уровня квантования, с которым отождествляется точное значение квантуемого дискретного сигнала с ближайшим меньшим при усечении и ближайшим большим – при округлении. Для однополярного сигнала $0 \le i \le (m-1)$, для двухполярного $(-m/2) \le i \le (m/2)-1$.

1.1.3. Погрешность квантования

Разность между квантованным и точным значениями дискретного сигнала $e(n) = x_{кв}(n) - x(n)$ определяет **погрешность, возникающую при квантовании сигнала по уровню.** В соответствии с **амплитудной характеристикой квантователя** она принимает значения $-Q/2 \le e(n) \le 0$ при усечении (рис. 1.4, *а*, *б*) и $-Q/2 \le e(n) \le Q/2$ при округлении (рис. 1.5, *а*, *б*). Предельные значения погрешности квантования $E_m = |e(n)|_{\max} = \eta Q$, где $\eta = 1$ при усечении и $\eta = 1/2$ при округлении.

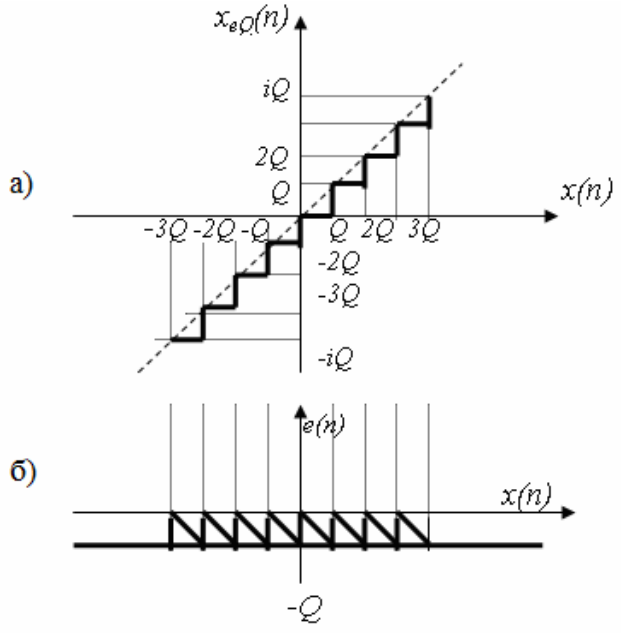


Рис. 1.4. Амплитудная характеристика квантования с усечением (*а*) и соответствующая ему погрешность квантования (*б*)

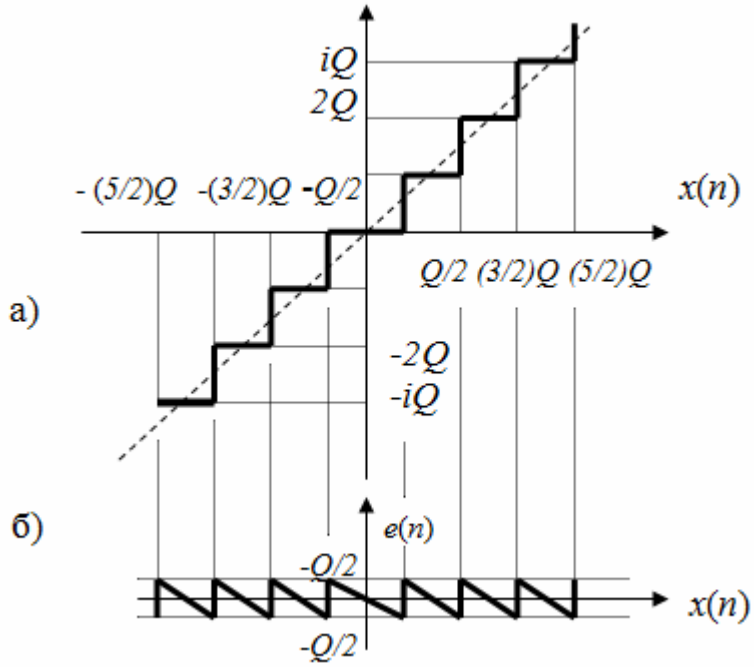


Рис. 1.5. Амплитудная характеристика квантования с округлением (*а*) и соответствующая ему погрешность квантования (*б*)

Вследствие **априорной неизвестности** значений сигнала и погрешности квантования ее рассматривают как случайный дискретный процесс или **шум квантования,** накладывающийся на квантуемый дискретный сигнал: $x_{кв}(n) = x(n) + e(n)$ и с равной вероятностью принимающий любые значения в вышеуказанных пределах. На графиках законов распределения шума квантования (рис. 1.6, *а*, *б*) обоим способам квантования соответствует одинаковая плотность вероятностей $p(e) = 1/Q$, отвечающая условию нормировки.

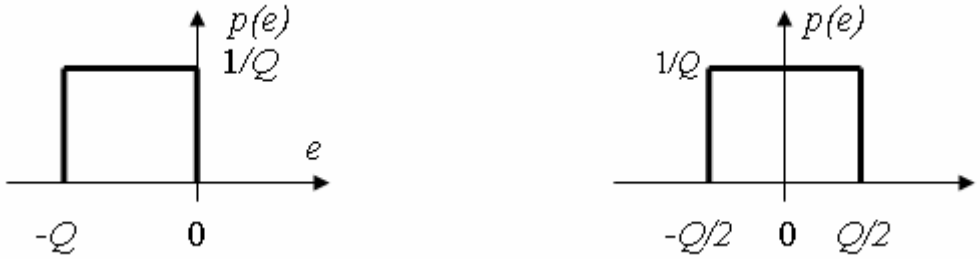


Рис. 1.6. Плотности вероятностей шума квантования при усечении (*а*) и округлении (*б*)

Шум квантования характеризуется следующими параметрами:

- математическим ожиданием

$$M_e = \int_{-Q}^{0} e \cdot p(e) de = -Q/2 \qquad (1.6)$$

при усечении и

$$M_e = \int_{-Q/2}^{Q/2} e \cdot p(e)de = 0 \qquad (1.7)$$

при округлении

- дисперсией

$$D_e = \int_{-Q/2}^{Q/2} e^2 \cdot p(e)de = Q^2/12 \qquad (1.8)$$

– для обоих способов квантования;

- среднеквадратическим значением

$$\sigma_e = \sqrt{D_e} = Q/\sqrt{12}\,. \qquad (1.9)$$

Спектрально-корреляционные свойства шума квантования существенно зависят от вида и параметров преобразуемого сигнала. В общем случае значения отсчетов шума квантования коррелированны со значениями входного сигнала и между собой. Наиболее высокая степень корреляции наблюдается при детерминированных сигналах, наименьшая – при случайных.

К последним относятся реальные речевые, музыкальные сигналы, выходные сигналы измерительных датчиков, а также любые сигналы на входе системы цифровой обработки сигналов (ЦОС) с наложенными на них шумами или случайными помехами. Наложение внешнего шума или помехи с приведенным к входу АЦП уровнем, превышающим значение единицы младшего разряда Q, существенно декоррелирует или рандомизирует шум квантования. Это позволяет рассматривать шум квантования, возникающий при преобразовании случайного сигнала, как дискретный случайный процесс типа «белый шум» с равномерной спектральной плотностью

$$P_{кв} = \sigma_{кв}^2 / f_Д\,. \qquad (1.10)$$

Повышение частоты дискретизации $f_Д$ уменьшает спектральную плотность шума квантования в основной полосе, позволяя при определенных условиях уменьшить необходимое значение разрядности АЦП, т.е. обменять разрядность сигнала на скорость его обработки. Модель шума квантования с вышеприведенными характеристиками широко используется в теории ЦОС.

1.2. Представление сигналов функциональными рядами

Пусть имеется непрерывный сигнал $x(t)$, заданный на интервале $[0,\infty)$. При переходе к оцифровке происходит следующая операция. Выбирается шаг дискретизации T, и вместо исходного сигнала получается по-

следовательность $s[n] = x(nT),\ \ n = 0,1,\ldots N-1$. Таким образом, дискретный сигнал $s(n)$, заданный на интервале $[0,\infty)$, можно записать в виде

$$s(n) = \sum_{k=0}^{\infty} c(k)\eta_k(n), \qquad (1.11)$$

где

$$c(k) = E_k^{-1} \sum_{n=0}^{N-1} s(n)\eta_k(n). \qquad (1.12)$$

Здесь $\{\eta_k(n)\}$ – совокупность базисных функций, определенных на том же интервале; $\{c(k)\}$ – коэффициенты разложения сигнала по базису $\{\eta_k(n)\}$, называемые спектром сигнала; E_k – энергия k-й базисной функции.

Из выражений (1.11) и (1.12) следует, что сигнал можно задать либо его отсчетами в дискретных точках, либо набором спектральных коэффициентов, причем спектральное представление неоднозначно и зависит от выбранной системы базисных функций. Выбор системы $\{\eta_k(n)\}$ определяется соображениями практического или математического удобства.

Наибольший интерес для задач цифровой обработки представляет случай, когда число базисных функций конечно и равно размерности сигнала, т.е. N. Далее будем рассматривать только такие системы [4].

В разложении (1.11) необходимо, чтобы базисная система удовлетворяла следующим требованиям:

1. Функции системы должны быть линейно независимыми:

$$a_0\eta_0(n) + a_1\eta_1(n) + \ldots + a_{N-1}\eta_{N-1}(n) \neq 0$$

при любых значениях коэффициентов a_i, кроме случая, когда

$$a_0 = a_1 = \ldots a_{N-1} = 0.$$

Наиболее часто в качестве базисной системы линейно независимых функций выбирают ортогональные функции, т.е. функции, для которых выполняется условие

$$\sum_{n=0}^{N-1} \eta_k(n)\eta_l^*(n) = 0,\ \ k \neq l\,.$$

В этом случае сигнал $\{s(n)\}$ можно рассматривать как вектор в декартовой системе координат, осями которой являются базисные функции.

Спектральные коэффициенты равны проекциям векторов на соответствующие оси (рис. 1.7).

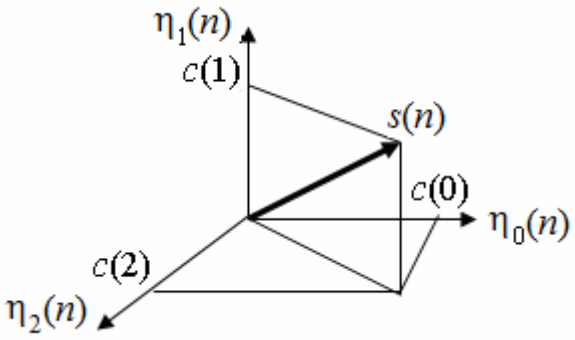


Рис. 1.7. Представление сигнала $s(n)$ в пространстве базисных функций

2. Система $\{\eta_k(n)\}$ должна быть упорядоченной. Это значит, что на множестве индексов k введено отношение порядка, показывающее, какая из функций предыдущая и какая последующая. Например, синусоидальные и косинусоидальные функции обычно располагают в порядке возрастания аргумента (частоты): $1, \cos x, \cos 2x, ..., \cos kx$. От введенного отношения порядка зависят форма спектра и удобство работы с ним. Например, при спектральном анализе по Фурье систем с ограниченной полосой пропускания часто отбрасывается составляющие спектра с высокими частотами.

3. Базисные функции на интервале ортогональности должны иметь конечную энергию, т. е.

$$\sum_{n=0}^{N-1} \eta_k^2(n) < \infty . \qquad (1.13)$$

4. Система функций $\{\eta_k(n)\}$ должна быть полной. Это означает, что к ней больше нельзя добавить ни одной функции, которая была бы ортогональна всем остальным функциям системы.

1.3. Математические модели дискретной и цифровой систем обработки сигналов

1.3.1. Основные типы сигналов и их математическое описание

Для аналитического описания сигналов и процессов их преобразований используют математические модели. Прежде всего сигналы рассматривают как функции, заданные в физических координатах. В этом смысле говорят об одномерных сигналах (например, зависящих от времени), двумерных, заданных на плоскости (например, изображениях), трехмерных (характеризующих, например, трехмерные пространственные объекты).

Обычно в теории сигналов в качестве математических моделей сигналов используются скалярные функции, но используются и более сложные модели – комплексные и векторные функции. Например, для описания электромагнитного поля как сигнала удобно использовать комплексные функции, для описания цветных изображений – трехкомпонентные векторные функции, для описания данных многоспектральной съемки – четырех- – шестикомпонентные векторные функции.

Важными общими характеристиками сигналов как математических функций являются множества значений, которые могут принимать они сами и их аргументы. С этой точки зрения целесообразно различать финитные и инфинитные, ограниченные и неограниченные, непрерывные, дискретные, квантованные и цифровые сигналы.

Финитными называются сигналы, область определения которых ограничена [5]. Например, финитным является сигнал, характеризующий кадр фотоснимка, значение электромагнитного поля в раскрыве антенны и т.п. Инфинитные сигналы имеют неограниченную область определения. Финитный сигнал можно превратить в инфинитный, если доопределить его значения за пределами заданного интервала. Однако любое доопределение должно выполняться так, чтобы при обработке сигнала не изменялись его свойства внутри области определения.

Если ограничена область значений сигнала, то сигнал называется ограниченным. Так, например, ограниченным является сигнал, характеризующий степень почернения фотонегатива или распределение яркости свечения экрана телевизионного монитора. Фактически все встречающиеся в природе сигналы являются ограниченными. Неограниченный сигнал – идеализация, принимаемая в тех случаях, когда ограничение области значений сигнала несущественно для данной задачи или просто неизвестно.

С точки зрения математического описания сигналов различают также детерминированное и вероятностное описания. При детерминированном описании сигналы рассматриваются индивидуально, независимо друг от друга, и считается, что значение сигнала может быть задано в каждой точке, где он определен. Однако иногда индивидуальное рассмотрение характеристик физических объектов невозможно, а можно измерить и учесть только некоторое число «макропараметров», характеризующих объекты в среднем. В этих случаях используется вероятностное описание, т.е. сигналы рассматриваются как выборочные функции или реализации из некоторого ансамбля сигналов и строится математическое описание не каждого отдельного сигнала, а ансамбля в целом.

Сигналы также подразделяют на дискретные, непрерывные и дискретно-непрерывные. Сигнал считают дискретным по данному параметру, если число значений, которое может принимать этот параметр, конечно (или счетно). Непрерывными называются сигналы, область определения и область значений которых непрерывны, т.е. для каждой точки области определения и области значения можно найти точку, удаленную от нее на бесконечно малое расстояние.

Сигнал, дискретный по одному параметру и непрерывный по другому, называют дискретно-непрерывным.

Математическую модель используют:

- как непрерывную функцию непрерывного аргумента (рис. 1.8);
- как непрерывную функцию дискретного аргумента, например, функцию, значения которой отсчитывают только в определенные моменты времени (рис. 1.9). Временной интервал между соседними отсчетами называется шагом дискретизации;
- как дискретную функцию непрерывного аргумента, например, функцию времени, квантованную по уровню (рис. 1.10);
- как дискретную функцию дискретного аргумента, принимающую одно из конечного множества возможных значений в дискретные моменты времени (рис. 1.11).

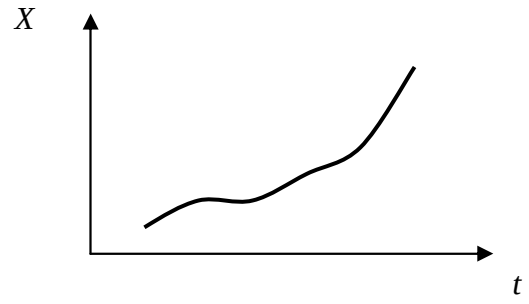


Рис. 1.8. Модель непрерывного сигнала

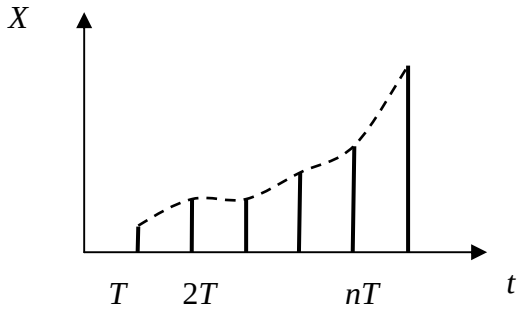


Рис. 1.9. Модель дискретного по времени сигнала

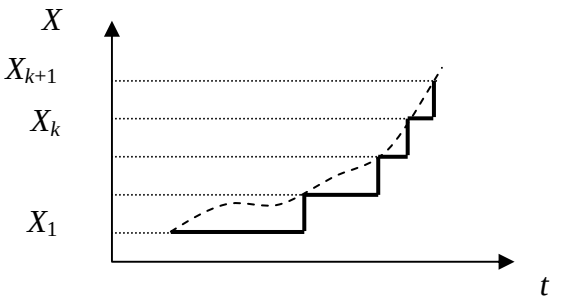


Рис. 1.10. Модель квантованного сигнала

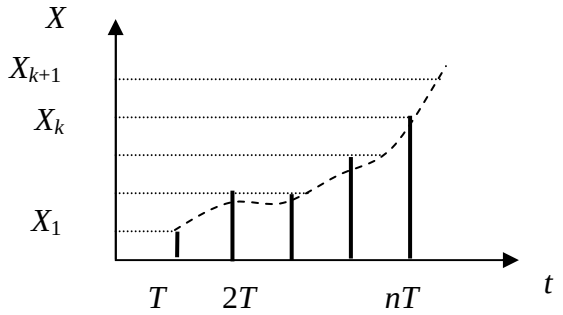


Рис. 1.11. Модель дискретного сигнала

Дискретный сигнал может быть как вещественным, так и комплексным. В последнем случае его вещественная и мнимая части описываются вещественными последовательностями

$$s(nT) = s_1(nT) + js_2(nT). \tag{1.14}$$

При описании дискретных и цифровых сигналов удобно пользоваться нормированным временем $\tilde{t}$

$$\tilde{t} = \frac{t}{T},$$

которое при $t = nT$ имеет смысл номера отсчета n

$$\tilde{t} = \frac{t}{T} = \frac{nT}{T} = n \tag{1.15}$$

и означает, что отсчет взят в момент nT. Это позволяет описывать дискретный сигнал функцией целочисленной переменной $x(n)$ и считать тождественными обозначения дискретного сигнала $s(n)$ и $s(nT)$:

$$s(nT) = s(n). \tag{1.16}$$

1.3.2. Типовые дискретные сигналы и их математическое описание. Основная полоса частот и нормирование частоты

1.3.2.1. Типовые дискретные сигналы [6]. Цифровой единичный импульс (дельта-импульс) описывается последовательностью

$$\delta_0(n) = \begin{cases} 1, n = 0; \\ 0, n \neq 0, \end{cases} \tag{1.17}$$

т.е. этот сигнал равен единице при $n = 0$ и нулю при всех остальных значениях n (рис. 1.12, *а*).

Задержанный цифровой единичный импульс описывается последовательностью

$$\delta_0(n - m) = \begin{cases} 1, n = m; \\ 0, n \neq m, \end{cases} \tag{1.18}$$

т.е. этот сигнал, в отличие от незадержанного, равен единице при $n = m$ и нулю при всех остальных значениях n (см. рис. 1.12, б).

Цифровой единичный скачок описывается последовательностью

$$u_0(n) = \begin{cases} 1, n \geq 0; \\ 0, n < 0, \end{cases} \tag{1.19}$$

т.е. этот сигнал равен единице при всех неотрицательных значениях n (рис. 1.13, *а*).

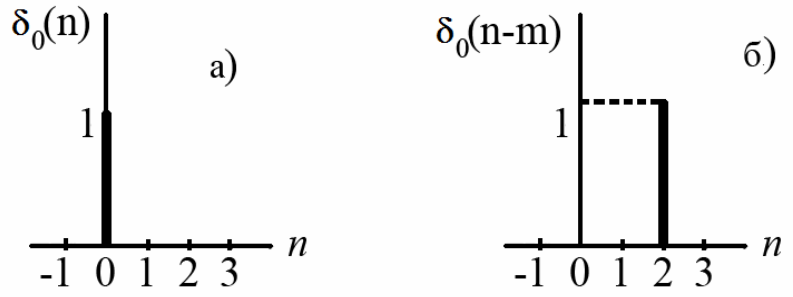


Рис. 1.12. Цифровой единичный (*а*) и задержанный цифровой единичный (*б*) импульсы

Задержанный цифровой единичный скачок (см. рис. 1.13, *б*) описывается последовательностью

$$u_0(n-m)=\begin{cases}1, n \geq m;\\ 0, n < m,\end{cases} \qquad (1.20)$$

т.е. этот сигнал, в отличие от незадержанного, равен единице при всех значениях $n \geq m$ и нулю при остальных значениях n.

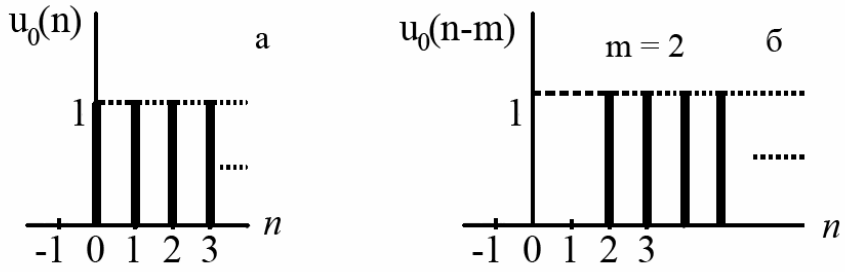


Рис. 1.13. Цифровой единичный (*а*) и задержанный цифровой единичный (*б*) скачки

Дискретная экспонента описывается последовательностью

$$s(n)=\begin{cases}a^n, n \geq 0;\\ 0, n < 0,\end{cases} \qquad (1.21)$$

где a – вещественная константа.

В зависимости от величины и знака a дискретная экспонента будет:

- $|a|<1$ и $a>0$ – убывающей знакопостоянной (рис. 1.14, *а*);
- $|a|<1$ и $a<0$ – убывающей знакопеременной (рис. 1.14, *б*);
- $a = 1$ – цифровым единичным скачком;
- $a = -1$ – знакопеременной последовательностью единиц;
- $|a|>1$ и $a>0$ – возрастающей знакопостоянной;
- $|a|>1$ и $a<0$ – возрастающей знакопеременной.

Дискретный гармонический сигнал, например, дискретная косинусоида описывается последовательностью

$$s(nT) = s(n) = A\cos(2\pi fnT) = A\cos(\omega nT), \qquad (1.22)$$

где T – период дискретизации; A – амплитуда; ω – круговая частота, связанная с частотой f коэффициентом пропорциональности $2\pi.\ \omega = 2\pi f$.

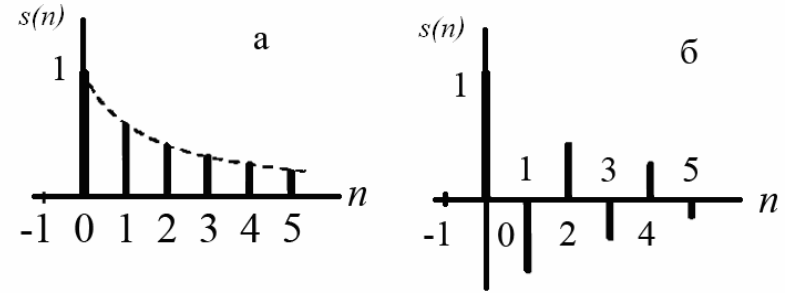


Рис. 1.14. Знакопостоянная (*а*) и знакопеременная (*б*) дискретные экспоненты

Дискретная косинусоида получается из аналоговой

$$s(t) = A\cos(2\pi ft) = A\cos(\omega t)$$

в результате замены непрерывного времени дискретным (рис. 1.15):

$$s(nT) = s(n) = A\cos(\omega t)\,|_{t=nT} = A\cos(\omega Tn). \qquad (1.23)$$

Дискретная синусоида описывается аналогично.

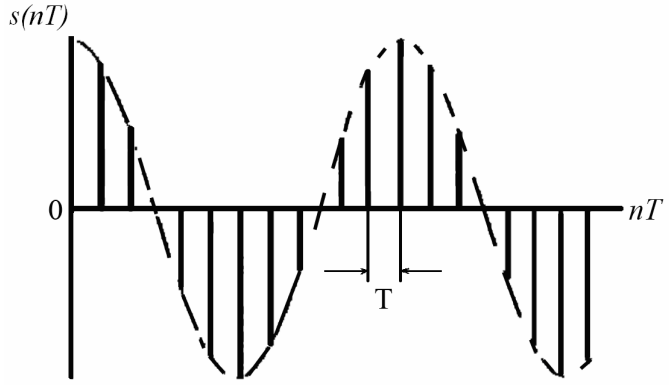


Рис. 1.15**.** Дискретная косинусоида

Дискретный комплексный гармонический сигнал описывается комплексной последовательностью

$$s(n) = Ae^{j\varpi Tn} \qquad (1.24)$$

или двумя вещественными последовательностями – косинусоидой (вещественная часть) и синусоидой (мнимая часть):

$$s(nT) = A\cos(\omega Tn) + jA\sin(\omega Tn). \qquad (1.25)$$

1.3.2.2. Основная полоса частот. Нормирование частоты. Согласно теореме Котельникова верхняя частота $f_в$ аналогового сигнала не должна превышать половины частоты дискретизации $f_Д$ этого сигнала. Следовательно, дискретные сигналы целесообразно рассматривать в области $\left[0;\frac{f_Д}{2}\right]$, которая называется основной полосой частот или основным диапазоном частот.

Это позволяет ввести понятие нормированных частот:

$\tilde{f}$ – отношение текущей частоты f к частоте дискретизации $f_Д$:

$$\tilde{f} = \frac{f}{f_Д} = fT; \tag{1.26}$$

$\tilde{\omega}$ – отношение текущей частоты ω к частоте дискретизации $f_Д$:

$$\tilde{\omega} = \frac{\omega}{f_Д} = \omega T. \tag{1.27}$$

Нормированные частоты $\tilde{f}$ и $\tilde{\omega}$ связаны отношением

$$\tilde{\omega} = \frac{\omega}{f_Д} = \frac{2\pi f}{f_Д} = 2\pi\tilde{f}. \tag{1.28}$$

В зависимости от выбранной шкалы частот основная полоса соответствует областям

$$f \to \left[0;\frac{f_Д}{2}\right];$$
$$f` \to [0;0,5];$$
$$\omega \to \left[0;\frac{\omega_Д}{2}\right] = \left[0;\frac{\pi}{T}\right];$$
$$\omega' \to [0;\pi].$$

Обычно используются абсолютная частота f и нормированная частота $\tilde{\omega}$.

Для косинусоиды в области нормированных частот выражение имеет вид:

$$s(n) = A\cos(2\pi\tilde{f}n) = A\cos(\tilde{\omega}n). \tag{1.29}$$

Введение нормированной частоты указывает на то, что в ЦОС важны не абсолютные значения частот сигнала и дискретизации, а их отношение.

Пример двух дискретных косинусоид:

$$s_1(n) = \cos(2\pi f_1 T_1 n) = \cos\left(2\pi \frac{f_1}{f_{Д1}} n\right) \text{ при } f_1 = 4 \text{ Гц}, f_{Д1} = 16 \text{ Гц};$$

$$s_2(n) = \cos(2\pi f_2 T_2 n) = \cos\left(2\pi \frac{f_2}{f_{Д2}} n\right) \text{ при } f_2 = 10 \text{ кГц}, f_{Д2} = 40 \text{ кГц}.$$

Подставив значения f_1 и $f_{Д1}$, f_2 и $f_{Д2}$, получим в области нормированных частот одинаковые дискретные сигналы

$$s_1(n) = \cos(2\pi \frac{4}{16} n) = \cos\left(\frac{\pi}{2} n\right); \quad (1.30)$$

$$s_2(n) = \cos(2\pi \frac{10000}{40000} n) = \cos\left(\frac{\pi}{2} n\right). \quad (1.31)$$

1.4. Спектры дискретных сигналов

Спектральную плотность дискретного сигнала $X(j\omega)$, для упрощения называемую в дальнейшем **спектром**, можно найти, дискретизировав по времени преобразование Фурье соответствующего ему аналогового сигнала $X_a(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$. Заменив t на $nT_Д$, интеграл на сумму и dt на $T_Д$, получим [2]:

$$X(j\omega) = T_Д \sum_{n=-\infty}^{\infty} x(nT_Д) e^{-j\omega n T_Д}. \quad (1.32)$$

С другой стороны, он может быть найден и прямым преобразованием Фурье дискретного сигнала, представленного функцией непрерывного времени [2]:

$$X_Д(j\omega) = \int_{-\infty}^{\infty} x_Д(t) e^{-j\omega t} dt = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) \delta(t - nT_Д) e^{-j\omega t} dt = \sum_{n=-\infty}^{\infty} x(nT_Д) e^{-j\omega n T_д}. \quad (1.33)$$

Выражения (1.32) и (1.33) отличаются только масштабным (и размерным) множителем $T_Д$, который при односкоростной обработке обычно опускают.

В силу периодичности комплексной экспоненты $e^{-j\omega n T_Д} = e^{-j(\omega + k\omega_Д) n T_Д}$ спектр дискретного сигнала, в отличие от аналогового, периодичен по частоте с периодом $\omega_Д$: $X(j\omega) = X[j(\omega + k\omega_Д)]$, $k = 0, \pm 1, \pm 2, \ldots$ (рис. 1.16). Периодизация спектра обусловлена дискретиза-

цией сигнала по времени. Справедливо и обратное утверждение о периодичности сигналов с дискретным по частоте (или линейчатым) спектром. Оба эти свойства отвечают фундаментальному положению о взаимосвязи дискретизации и периодизации сигналов во временной и частотной области.

Определяют спектр дискретного сигнала в основной полосе частот $(0 \pm \omega_Д / 2)$. На основе определения спектра дискретного сигнала

$$x_Д(t) = x(t)\, f_\delta(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_Д) = \sum_{n=-\infty}^{\infty} x(nT_Д)\delta(t - nT_Д), \qquad (1.34)$$

в котором дискретизирующая функция $f_\delta(t)$ представляется или заменяется аппроксимирующим ее рядом Фурье $f_\delta(t) = \sum_{k=-\infty}^{\infty} C_k e^{jk\omega_Д t}$:

$$x_Д(t) = x(t) \sum_{k=-\infty}^{\infty} C_k e^{jk\omega_Д t}. \qquad (1.35)$$

Коэффициенты ряда

$$C_k = \frac{1}{T_Д} \int_{nT_Д - T_Д/2}^{nT_Д + T_Д/2} \delta(t - nT_Д) e^{-jk\omega_Д t} dt = \frac{1}{T_Д} e^{-jk\omega_Д nT_Д} = \frac{1}{T_Д}$$

соответствуют равномерному дискретному (или линейчатому) спектру этой периодической функции. Такой спектр может быть описан также непрерывным преобразованием Фурье с помощью дискретизирующей функции:

$$F_\delta(j\omega) = \frac{1}{T_Д} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_Д).$$

Преобразование Фурье (1.34) при $C_k = 1/T_Д$ приводит к выражению

$$x_Д(j\omega) = \frac{1}{T_Д} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\omega} x(t) e^{jk\omega_Д t} e^{-j\omega t} dt = \frac{1}{T_Д} \sum_{k=-\infty}^{\infty} X_a[j(\omega - k\omega_Д)]. \qquad (1.36)$$

Из (1.36) следует, что спектр дискретного сигнала с точностью до постоянного множителя равен сумме спектров аналогового сигнала $X_a(j\omega)$, смещенных по частоте на $k\omega_Д$. Перенос спектра $X_a(j\omega)$ на частоты $k\omega_Д$ вызван умножением аналогового сигнала на множество комплексных экспонент $e^{jk\omega_Д t}$, являющихся гармониками дискретизирующей функции $f_\delta(t)$. Он физически объясняет явление размножения или периодизации спектров при дискретизации.

Выражение (1.36) может быть проиллюстрировано для различных случаев дискретизации аналоговых сигналов с ограниченным (финитным) и неограниченным спектрами, которые во временной области имеют соответственно бесконечную и конечную длительности.

На рис. 1.16 спектр дискретного сигнала в основной полосе частот $\pm\omega_Д / 2$ (при $|\omega| \le \omega_Д / 2$) точно совпадает (до постоянного множителя $T_Д$) со спектром аналогового сигнала: $T_Д X_Д(j\omega) = X_a(j\omega)$. Дискретизация не сопровождается здесь искажениями. В этом случае возможно точное восстановление аналогового сигнала по дискретным выборкам с помощью идеального ФНЧ с прямоугольной частотной характеристикой $\prod \omega_m(j\omega)$, равной $T_Д$ при $|\omega| \le \omega_Д / 2$ и равной нулю при $|\omega| > \omega_Д / 2$.

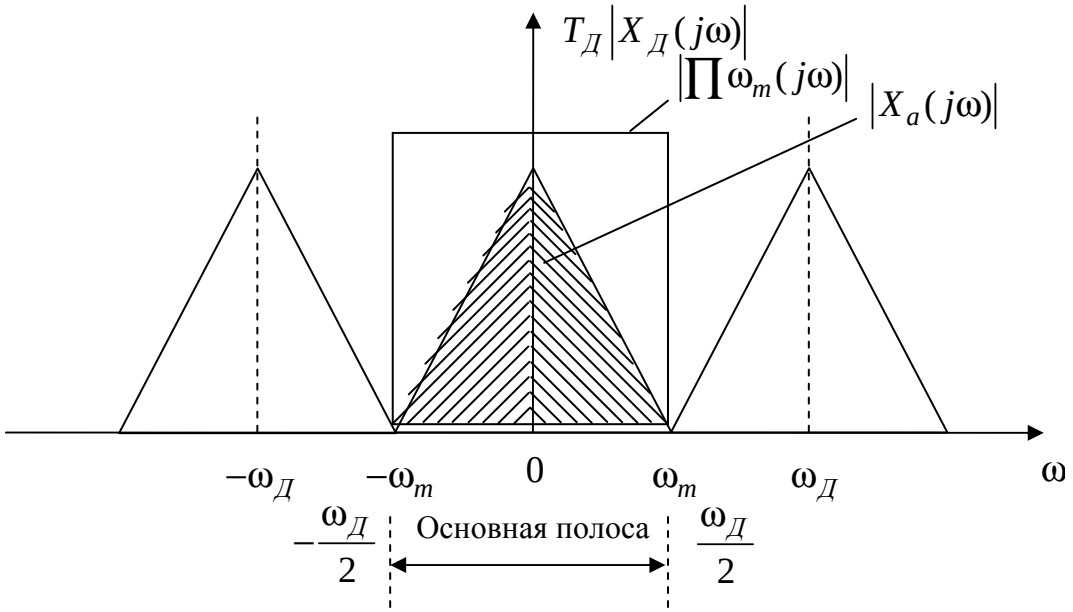


Рис. 1.16. Спектральное преобразование при дискретизации аналогового сигнала с финитным спектром в случае $\omega_Д \ge 2\omega_m$

Возможен случай, когда спектр дискретного сигнала в основной полосе частот отличается от спектра аналогового сигнала. Периодизация спектра $X_a(j\omega)$ здесь приводит к перекрытию и суммированию его с соседними смешанными по частоте спектрами (рис. 1.17). Это явление называют наложением спектров при дискретизации. Связанные с ним погрешности дискретизации также называют погрешностями или искажениями наложения.

При наложении невозможно точное восстановление аналогового сигнала по его дискретным выборкам.

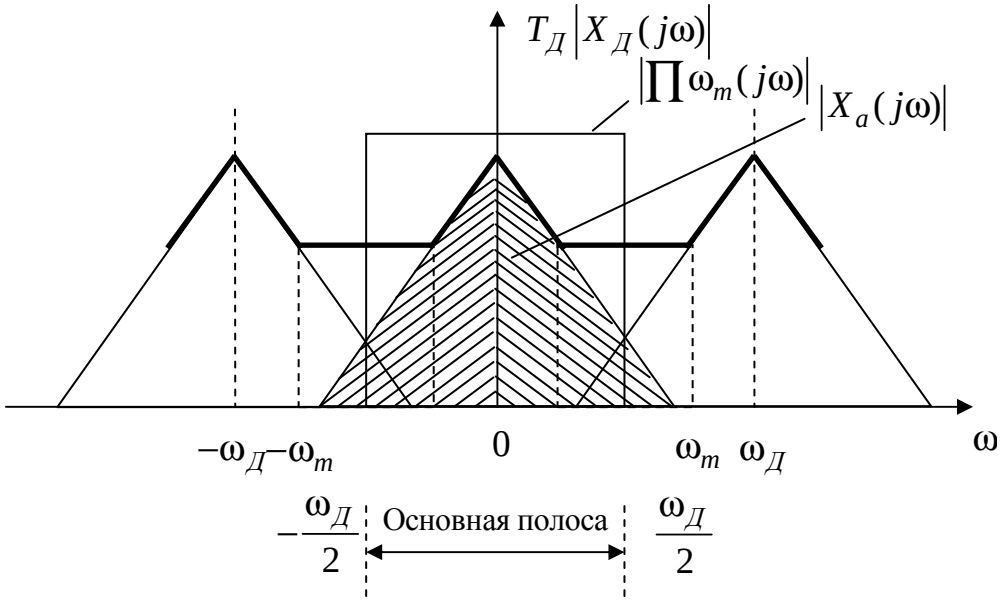


Рис. 1.17. Спектральное преобразование при дискретизации аналогового сигнала с финитным спектром в случае $\omega_Д < 2\omega_m$

1.5. Технические показатели эффективности ЦОС

Устройства цифровой обработки сигналов (ЦОС) – это устройства, реализующие тот или иной алгоритм цифровой обработки.

Основные преимущества устройства ЦОС по сравнению с устройствами аналоговой обработки и дискретными системами, реализуемыми на аналоговых элементах, следующие [7]:

1. Характеристики устройств ЦОС абсолютно стабильны и не изменяются при изменении внешних условий (температуры, влажности и т.д.) пока все элементы устройства сохраняют работоспособность.

2. Возможна реализация ряда операций и алгоритмов, принципиально не реализуемых с помощью аналоговых элементов; например, можно обрабатывать весьма низкочастотные сигналы, поскольку длительность хранения информации цифровыми элементами практически неограниченна.

3. Эти устройства весьма удобно реализовывать в виде больших и сверхбольших интегральных схем, например, в виде специальных микропроцессоров.

Основные недостатки современных устройств ЦОС:

1. Относительно низкая скорость обработки информации, которая ограничивается задержками используемых цифровых элементов.

2. Как правило, большая потребляемая мощность.

3. Относительно большая стоимость.

4. Необходимость использования на входе и выходе элементов АЦП и ЦАП.

Отмеченные выше достоинства позволяют считать устройства ЦОС весьма перспективными при значениях частот дискретизации сигналов до сотен килогерц.

Принципиально точность устройств цифровой обработки сигналов ограничена применяемыми АЦП и ЦАП. Точность вычисления в самом устройстве определяется числом двоичных разрядов, используемых для представления кодов.

1.6. Эффективность алгоритмов и оценка их вычислительной сложности

Вопросы сложности алгоритмов и устройств всегда были в центре внимания инженеров. Само понятие сложности эволюционировало с развитием техники и имеет много различных трактовок, отражающих как специфику устройства, так и уровень технического развития. Так, сложность первых радиоэлектронных устройств оценивали количеством содержащихся в них функциональных элементов. С развитием микроэлектроники отдельный элемент стал заменяться модулем – прибором, содержащим много элементов.

Понятие сложности алгоритма также допускает различные трактовки. Здесь могут учитываться или не учитываться такие факторы, как размер машинного слова, емкость памяти, различие в длительности выполнения отдельных команд и т.д.

К настоящему времени не существует достаточно универсального понятия сложности, которым можно пользоваться в любой ситуации. Вместе с тем создан ряд моделей вычислительного процесса, позволяющих раскрыть вычислительную сложность отдельных задач и сравнить различные алгоритмы. Наиболее употребительной является модель неветвящейся программы.

Неветвящаяся программа представляет собой программу без циклов, в которой цикл заменяется копированием повторяющейся команды соответствующее число раз. Число шагов такой программы, как функция от размера входа N, называется временно́й сложностью, а число переменных, участвующих в вычислениях, – емкостной сложностью.

Пример 1.1. Рассмотрим вычисление полинома

$$p(z) = a_N z^N + a_{N-1} z^{N-1} + \ldots + a_1 z + a_0.$$

Можно записать:

$$a_1 z + a_0,\ N = 1;\ (a_2 z + a_1)z + a_0,\ N = 2;\ ((a_3 z + a_2)z + a_1)z + a_0,\ N = 3.$$

Неветвящаяся программа приведена в табл. 1.1.

Таблица 1.1

N=1	N=2	N=3
$t \leftarrow a_1 z$ $p \leftarrow t + a_0$	$t \leftarrow a_2 z$ $t \leftarrow t + a_1$ $t \leftarrow tz$ $p \leftarrow t + a_0$	$t \leftarrow a_3 z$ $t \leftarrow t + a_2$ $t \leftarrow tz$ $t \leftarrow t + a_1$ $t \leftarrow tz$ $p \leftarrow t + a_0$

Временная сложность этой программы равна $2N$, а емкостная $N + 4$.

Дадим формальное определение неветвящейся программы. Пусть заданы:

1) набор входных переменных $x_0, x_1, ..., x_{N-1}$;

2) кольцо K (или поле F);

3) множество P базисных операций $P = \{+,*,/\} \cup \{x,\gamma\}$, где +, *, / – двухместные арифметические операции сложения, умножения, деления; x, γ – одноместная операция умножения на элемент кольца или поля.

Неветвящаяся программа (НП) представляет собой последовательность строк (команд), 1-я из которых имеет вид

$$z_L = f_L(z_{i_1}, z_{i_u}, x_{j_1}, ..., x_{j_v}),$$

где $i_1 < L, ..., i_u < L$, $f_L \in P$.

Для любой базисной операции из множества P фиксируется число $\lambda(f)$, называемое сложностью этой операции. Сложностью НП называется сумма всех $\lambda(f)$ по всем строкам этой программы.

Рассмотрим ряд модификаций этого понятия:

1. Пусть $\lambda(f) = 1$ для всякой $f \in P$. Тогда соответствующая сложность «считает» число всех операций НП и называется тотальной сложностью. Обозначим ее C_t .

2. Пусть $\lambda(+) = 1$, $\lambda(*) = \lambda(/) = \lambda(x\ \gamma) = 0$, т.е. учитываются только операции сложения. Соответствующая сложность называется аддитивной. Обозначим ее C_a.

3. Пусть $\lambda(*) = \lambda(/) = 1$, а $\lambda(+) = \lambda(x\ \gamma) = 0$, т.е. учитываются только нелинейные операции – умножение и деление. Сложность такого рода называется мультипликативной. Обозначим ее $C_{m.}$

Аддитивная сложность является хорошим критерием качества алгоритма при обработке бинарных или троичных сигналов, элементы которых кодируются в алфавитах (0,1), (1, –1), (1,0, –1). В этих случаях операции умножения и деления отсутствуют. Мультипликативная сложность обычно используется тогда, когда операция умножения существенно дороже операции сложения. При тотальной сложности все операции оцениваются

одинаково. Мультипликативная сложность используется при анализе вычислителя, построенного на матричных процессорах.

Заметим также, что сложность существенно зависит от выбора кольца K или поля F. Например, для вычисления выражения $x^2 + y^2$ в поле вещественных чисел необходимо выполнить два умножения, а в поле комплексных чисел – только одно, так как $x^2 + y^2 = (x + jy)\,(x - jy)$.

При оценке качества алгоритма обычно оперируют асимптотической сложностью, т.е. величиной, которая получается при неограниченном увеличении размера входа. Она в итоге определяет размер задач, которые можно решить алгоритмом. Асимптотическая сложность оценивается порядком роста функции без учета мультипликативных констант. Например, если N входных переменных обрабатываются за время cN^2, где c – некоторая постоянная, то временная сложность этого алгоритма есть $O(N^2)$ (читается: порядка N^2).

Ошибочно думать, что значение эффективных алгоритмов уменьшится с ростом быстродействия вычислительных машин. В качестве примера рассмотрим пять алгоритмов $A_1 - A_5$ различной сложности для решения одной и той же задачи (табл. 1.2).

Таблица 1.2

Алгоритм	Временная сложность
A_1	N
A_2	$N\log_2 N$
A_3	N^2
A_4	N^3
A_5	2^N

В этих алгоритмах под временной сложностью будем понимать число единиц времени, требуемого для обработки входа размером N. Пусть, например, единицей времени будет одна миллисекунда. Тогда алгоритм A_1 обработает за 1 с вход размером 1000, в то время как алгоритм A_5 – вход размером не более 9. В табл. 1.3 приведены размеры задач, которые можно решить за различное время этими алгоритмами.

Таблица 1.3

Алгоритм	Временная сложность	Максимальный размер задачи		
		1 с	1 мин	1 ч
A_1	N	1000	$6\cdot10^4$	$3{,}6\cdot10^6$
A_2	$N\log_2 N$	140	4893	$20\cdot10^5$
A_3	N^2	31	244	1897
A_4	N^3	10	39	153
A_5	2^N	9	15	21

Предположим, что быстродействие вычислительной машины увеличилось в 10 раз. В табл. 1.4 показано, как при этом возрастут размеры задач.

Таблица 1.4

Алгоритм	Временная сложность	***Максимальный размер задачи***	
		До ускорения	после ускорения
A_1	N	S_1	$10S_1$
A_2	$N\log_2 N$	S_2	$10S_2$(при больших S_2)
A_3	N^2	S_3	$3{,}16S_3$
A_4	N^3	S_4	$2{,}15S_4$
A_5	2^N	S_5	$S_5 + 3{,}3$

Видно, что увеличение быстродействия приводит к существенному увеличению размера задачи только в алгоритмах с малой временной сложностью.

Предположим, что вместо увеличения быстродействия мы будем совершенствовать алгоритм. Рассмотрим табл. 2, взяв для сравнения колонку с временем решения 1 мин. Тогда, заменив алгоритм A_4 алгоритмом A_3, можно получить шестикратное увеличение размера задачи, а при замене алгоритма A_4 на алгоритм A_2 размер задачи возрастает в 125 раз. Если выполнить сравнение за 1 ч, то различие окажется еще более существенным.

Приведенные примеры показывают, что асимптотическая сложность служит важной мерой качества алгоритма.

В практических приложениях все задачи имеют ограниченный размер. Поэтому кроме порядка роста следует учитывать и мультипликативную константу. Иногда больший порядок роста может иметь меньшую мультипликативную константу, и в этом случае такой алгоритм окажется предпочтительным при малых размерах задачи. Например, предположим, что временные сложности алгоритмов A_1, A_2, A_3, A_4, A_5, равны соответственно $1000N$, $100N\log_2 N$, $10N^2$, N^3 и 2^N. Тогда алгоритм A_5 будет наилучшим для задач размером $2 \le N \le 9$, A_3 – для задач размером $10 \le N \le 58$, A_2 – для задач размером $59 \le N \le 1024$ и A_1 – для задач размером $N > 1024$.

Вопросы и задания для самопроверки

1. Назовите и обоснуйте основные преимущества и недостатки ЦОС по сравнению с системами аналоговой обработки сигналов.
2. Что такое сигнал и что понимают под цифровой обработкой сигналов?
3. Представьте структурную схему ЦОС и поясните принцип ее работы.
4. Сформулируйте теорему Котельникова и ее назначение.
5. Дайте определение дискретному, непрерывному и дискретно-непрерывному сигналу, приведите примеры.
6. Назовите типовые дискретные сигналы и опишите их математически.
7. Для каких целей в ЦОС используется нормирование частоты?
8. В каких областях используется цифровая обработка сигналов?
9. Что такое дискретный сигнал и дискретная последовательность?
10. В чем заключаются взаимосвязь и отличие спектров дискретного и аналогового сигналов?
11. Можно ли по известному спектру дискретного сигнала найти спектр соответствующего ему аналогового сигнала?
12. В чем заключается и как проявляется наложение спектров при дискретизации сигналов?
13. Из каких условий выбирается частота дискретизации аналоговых сигналов?
14. Каким образом могут быть уменьшены искажения, связанные с дискретизацией сигнала?
15. Что такое апертурная погрешность УВХ и АЦП?
16. Как изменяется спектр сигнала при цифроаналоговом преобразовании?
17. Как зависят искажения восстановления сигнала от частоты дискретизации и максимальной частоты его спектра в основной полосе?
18. Как определяется погрешность квантования дискретного квантованного сигнала?

МОДУЛЬ 2. ПРЕОБРАЗОВАНИЯ СИГНАЛОВ В ЦИФРОВЫХ СИСТЕМАХ

2.1. Преобразование Фурье и *Z*-преобразование

2.1.1. Основы преобразования Фурье

Выражение регулярного сигнала определенной функцией времени называют временным представлением сигнала. Форма записи функции может быть различной. В частности, при некоторых ограничениях функция времени, заданная на некотором отрезке времени, может быть представлена в виде тригонометрического ряда, каждый член которого является простейшей гармонической функцией времени (косинус, синус). Эти функции называются гармониками, и каждой из них принадлежат определенные амплитуда, частота и фаза [8].

Множество амплитуд, частот и фаз называют спектром рассматриваемого сигнала. Функция времени находится в однозначном соответствии с принадлежащим ей спектром. На этом основании временное представление сигнала может быть заменено так называемым частотным представлением. Оба представления адекватны.

Представление периодического сигнала суммой гармонических составляющих осуществляется с помощью его разложения в ряд Фурье. Если функция (сигнал) $f(t)$ задана на интервале времени $t_1 \le t \le t_2$ и повторяется с периодом $T_p = 2\pi / \omega$, то тригонометрическая форма ряда Фурье для нее может быть записана следующим образом:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega T_{\text{p}} + b_n \sin n\omega T_{\text{p}}) . \qquad (2.1)$$

Амплитуды косинусоидальных и синусоидальных членов определяются выражениями

$$a_n = \frac{2}{T_p} \int_0^{T_p} f(t) \cos(n\omega t) dt ; \qquad (2.2)$$

$$b_n = \frac{2}{T_p} \int_0^{T_p} f(t) \sin(n\omega t) dt . \qquad (2.3)$$

Слагаемое

$$\frac{a_0}{2} = \frac{A_0}{2} = \frac{1}{T_p}\int_0^{T_p} f(t)dt \qquad (2.4)$$

является постоянной составляющей сигнала, которая равна среднему значению функции $f(t)$ за период.

Частоты $n\omega$ называют n-ми гармониками частоты ω. Следовательно, бесконечный ряд содержит зависящие от частоты синусоидальные и косинусоидальные члены с различными амплитудами на положительных частотах гармоник.

Амплитуда A_n и фаза ψ_n n-ой гармоники связаны с величинами a_n и b_n соотношениями

$$A_n = \sqrt{a_n^2 + b_n^2}\,; \quad \psi_n = \operatorname{arctg}(b_n / a_n)\,. \qquad (2.5)$$

Весьма удобной является комплексная форма записи ряда Фурье, к которой легко перейти, если выразить тригонометрические функции через показательные, воспользовавшись известными формулами Эйлера:

$$\cos n\omega t = \frac{1}{2}(e^{jn\omega t} + e^{-jn\omega t})\,; \qquad (2.6)$$

$$\sin n\omega t = \frac{1}{2j}(e^{jn\omega t} - e^{-jn\omega t})\,. \qquad (2.7)$$

В результате получим

$$f(t) = \frac{a_0}{2} + \frac{1}{2}\sum_{n=1}^{\infty}(A_n e^{jn\omega t} + A_n^* e^{-jn\omega t})\,, \qquad (2.8)$$

где A_n и A_n^* – комплексные амплитуды, связанные с a_n и b_n соотношениями

$$A_n e^{-j\psi_n} = a_n - jb_n\,;$$
$$A_n^* e^{j\psi_n} = a_n + jb_n\,. \qquad (2.9)$$

Таким образом, комплексные амплитуды A_n и A_n^* являются комплексно-сопряженными величинами. Действительно, каждое слагаемое ряда можно представить как вектор на комплексной плоскости (рис. 2.1), вращающийся с частотой $n\omega$ (т.е. в положительном направлении отсчета углов – против направления движения часовой стрелки). Каждое слагаемое второго ряда – вектор, вращающийся в обратном направлении.

Так как A_n и A_n^* – комплексно-сопряженные величины, то сумма векторов в любой момент времени дает вектор, направленный по вещест-

венной оси, т.е. n-ю гармоническую составляющую вещественной функции времени $f(t)$. Отрицательная частота $(-n\omega)$ только указывает направление вращения вектора.

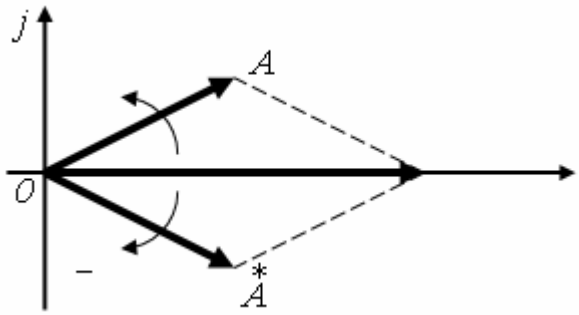


Рис. 2.1. Векторная диаграмма комплексно-сопряженных величин

Таким образом, комплексная амплитуда A_n может быть определена согласно выражению

$$A_n = \frac{1}{T_p}\int_0^T f(t)e^{-jn\omega t}dt\,. \qquad (2.10)$$

Следовательно, в экспоненциальной форме ряд Фурье имеет вид

$$f(t) = \sum_{n=1}^{\infty} A_n e^{jn\omega t}\,.$$

При такой записи ряда Фурье периодический сигнал заменяется суммой простых гармонических колебаний как с положительными частотами ($n > 0$), так и с отрицательными ($n < 0$). Конечно, отрицательные частоты не имеют здесь физического смысла, а являются формальным следствием произведенного математического преобразования.

2.1.2. Основы *Z*-преобразования

Z-преобразование представляет собой разложение функций в ряды степенных полиномов по *z*. Впервые *Z*-преобразование введено в употребление П. Лапласом в 1779 г. и повторно «открыто» В. Гуревичем в 1947 г. с изменением символики на z^{-k}. В настоящее время в технической литературе имеют место оба вида символики. На практическое использование преобразования это не влияет, т.к. смена знака только зеркально изменяет нумерацию членов полинома (относительно z^0), числовое пространство которых в общем случае от –∞ до +∞. В дальнейшем в качестве основной будем

использовать символику положительных степеней z, давая пояснения по особенностям отрицательной символики, если таковая имеется [8, 9, 10].

Произвольной непрерывной функции $s(t)$, равномерно дискретизированной и отображенной отсчетами $s_k = s(k \cdot t)$, равно как и непосредственно дискретной функции, можно поставить в однозначное соответствие степенной полином по z, последовательными коэффициентами которого являются значения s_k:

$$s_k = s(k \cdot t) \Leftrightarrow TZ[s(k \cdot t)] = \sum_{k=-\infty}^{\infty} s_k = S(z), \qquad (2.11)$$

где $z = \sigma + j\omega$ – произвольная комплексная переменная.

В показательной форме $z = r \cdot e^{-j\varphi}$,

где $r = |z| = \sqrt{\sigma^2 + \omega^2}$, $\varphi = \arg(z) = \operatorname{arctg}(\omega / \sigma)$.

Пример 2.1

$s_k = \{1, 2, 0, –1, –2, –1, 0, 0\}$.

$S(z) = 1z^0 + 2z^1 + 0z^2 – 1z^3 – 2z^4 – 1z^5 + 0z^6 + 0z^7 = 1 + 2z – z^3 – 2z^4 – z^5$.

В каузальных системах значения импульсного отклика систем существуют при $k \geq 0$ и уравнение (2.11) действует в одностороннем варианте:

$$H(z) = \sum_{k=0}^{\infty} h_k \cdot z^k . \qquad (2.12)$$

В общем случае z-преобразование – это степенной ряд с бесконечным количеством членов, поэтому он может сходиться не для всего пространства значений z. Область z, в которой z-преобразование сходится и значения $S(z)$ конечны, называют областью сходимости.

Пример 2.2. Последовательность (сигнал) конечной длины, непричинная:

$s_{-k} = \{1, 2, 3, 2, 1\}$, k = 0, 1, 2, 3, 4;

$S(z) = 1z^0 + 2z^{-1} + 3z^{-2} + 2z^{-3} + 1z^{-4} = 1 + 2/z + 3/z^2 + 2/z^3 + 1/z^4$.

Очевидно, что $S(z) = \infty$ при $z = 0$. Область сходимости – все значения z, за исключением $z = 0$.

Пример 2.3. Последовательность конечной длины, причинная (как импульсный отклик каузальной системы):

$s_k = \{1, 2, 3, 2, 1\}$, $k = 0, 1, 2, 3, 4$;

$S(z) = 1 + 2z + 3z^2 + 2z^3 + z^4$.

$S(z) = \infty$ при $z = \infty$. Область сходимости – все значения z, за исключением $z = \infty$.

Пример 2.4. Последовательность конечной длины, двусторонняя (как импульсный отклик симметричного фильтра):

$s_k = \{1, 2, 3, 2, 1\}$, $k = -2, -1, 0, 1, 2$;

$S(z) = 1z^{-2} + 2z^{-1} + 3z^0 + 2z^1 + 1z^2 = 1/z^2 + 2/z + 3 + 2z + z^2$;

$S(z) = \infty$ при $z = 0$ и $z = \infty$. Область сходимости не включает точки $z = 0$ и $z = \infty$.

Пример 2.5. Последовательность бесконечной длины, причинная (как импульсный отклик рекурсивного интегрирующего фильтра): $s_k = 0$ при $k < 0$, $s = 1$ при $k \geq 0$.

$S(z) = z^{-0} + z^1 + z^2 + z^3 + \ldots = 1 + z + z^2 + z^3 + \ldots = 1/(1 - z)$.

Ряд удовлетворяет условию сходимости только при $|z| < 1$.

Значения z, для которых $S(z) = \infty$, называются *полюсами*, а те, для которых $S(z) = 0$, называются нулями функции $S(z)$. Как видно из примеров, для последовательностей конечной длины z-преобразование сходится везде, кроме точки $z = \infty$ для имеющих правостороннюю часть ($k \geq 0$), и точки $z = 0$ для имеющих левостороннюю часть ($k < 0$), в любых их комбинациях. Для бесконечных причинных последовательностей преобразование сходится везде внутри круга единичного радиуса с центром в начале координат.

По заданному или полученному в результате анализа какой-либо системы z-полиному однозначно восстанавливается соответствующая этому полиному функция путем идентификации коэффициентов степеней при z^k с k-отсчетами функции.

Пример 2.6. Восстановление функции по z-полиному:

$S(z) = 1 + 3z^2 + 8z^3 - 4z^6 - 2z^7 = 1z^0 + 0z^1 + 3z^2 + 8z^3 + 0z^4 + 0z^5 - 0z^6 - 2z^7$;

$s_k = \{1, 0, 3, 8, 0, 0, -4, -2\}$.

Смысл величины z в z-полиноме заключается в том, что она является оператором единичной задержки по координатам функции. Умножение z-образа сигнала $s(k)$ на величину z^n означает задержку сигнала (сдвиг вправо по временной оси) на n интервалов: $z^n S(z) \Leftrightarrow s(k - n)$. Чтобы убедиться в этом, достаточно в приведенном выше примере выполнить умножение многочлена $S(z)$, например, на z^2, выполнить обратное преобразование и получить новый сигнал $s_k = \{0, 0, 1, 0, 3, 8, 0, 0, -4, -2\}$.

Z-образы с положительными степенями z соответствуют каузальным (физически реализуемым) процессам и системам, которые работают в реальном масштабе времени с текущими и «прошлыми» значениями сигналов. При обработке информации на ЭВМ каузальность сигналов не относится к числу ограничений и возможно использование отрицательных степеней z, соответствующих отсчетам сигналов «вперед». Последнее применяется, например, при синтезе симметричных операторов фильтров, что

позволяет производить обработку информации без внесения в сигнал фазовых искажений. При использовании символики z^{-1} «прошлым» значениям соответствуют значения с отрицательными степенями z, «будущим» – с положительными.

Основное достоинство z-преобразований заключается в простоте математических операций со степенными полиномами, что имеет немаловажное значение при расчетах цифровых фильтров и в спектральном анализе.

2.1.3. Связь с преобразованиями Фурье и Лапласа

Запишем дискретный сигнал s_k в виде суммы весовых импульсов Кронекера:

$$s_k = s(kt) = \sum_{n=-\infty}^{\infty} s(nt)\delta(kt - nt). \qquad (2.13)$$

Определим спектр сигнала по теореме запаздывания:

$$S(\omega) = \sum_{k=-\infty}^{\infty} s(kt)e^{-j\omega kt}. \qquad (2.14)$$

Выполним замену переменных, $z = e^{-j\omega t}$, и получим:

$$S(\omega) = \sum_{k=-\infty}^{\infty} s(kt)z^k = S(z). \qquad (2.15)$$

Отсюда следует, что дискретное преобразование Фурье является частным случаем z-преобразования при $z = e^{-j\omega t}$.

Аналогичной подстановкой $z = e^{-pt}$ может осуществляться переход к дискретному преобразованию Лапласа. В общем виде

$$S(\omega) = S(z), \quad z = e^{-j\omega t}; \; S(p) = S(z), \quad z = e^{-pt}. \qquad (2.16)$$

Обратное преобразование:

$$S(z) = S(\omega), \quad \omega = \ln z/jt; \qquad S(z) = S(p), \quad p = \ln z/t. \qquad (2.17)$$

При отрицательной символике z связь между представлениями осуществляется, соответственно, подстановками $z^{-1} = e^{j\omega t}$ и $z^{-1} = e^{pt}$.

При $z^k = e^{-j\omega tk}$ z-преобразование представляет собой особую форму представления дискретных сигналов, при которой на полином $S(z)$ можно ссылаться как на временную функцию (по значениям коэффициентов kt), так и на функцию частотного спектра сигнала (по значениям аргумента ω).

Отображение z-преобразования выполняют на комплексной z-плоскости с Re z и Im z по осям координат (рис. 2.2). В частности, спектральной оси частот ω на z-плоскости соответствует окружность радиусом

$$|z| = \left|e^{-j\omega t}\right| = \sqrt{\cos^2(\omega t) + \sin^2(\omega t)} = 1. \qquad (2.18)$$

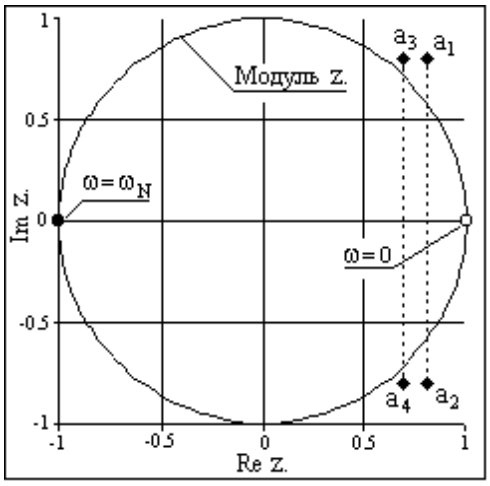


Рис. 2.2. Комплексная *z*-плоскость

Подстановка значения какой-либо частоты ω в $z = e^{-j\omega t}$ отображается точкой на окружности. Частоте ω = 0 соответствует точка Re *z* = 1 и Im *z* = 0 на правой стороне оси абсцисс.

При повышении частоты точка смещается по окружности против часовой стрелки и занимает крайнее левое положение на частоте Найквиста $\omega_N = \pi / t$ (Re *z* = –1, Im *z* = 0). Отрицательные частоты спектра отображаются аналогично по часовой стрелке на нижней полуокружности. Точки $\pm\omega_N$ совпадают, а при дальнейшем повышении или понижении частоты значения начинают повторяться в полном соответствии с периодичностью спектра дискретной функции. Проход по полной окружности соответствует одному периоду спектра, а любая гармоника спектра сигнала задается на плоскости двумя точками, симметричными относительно оси абсцисс.

Сигналы и системы непрерывного времени очень часто описываются с помощью преобразования Лапласа. Если $z = e^{-pt}$, где $p = \sigma + j\omega$, то

$$z = e^{-(\sigma + j\omega)t} = e^{-\sigma t} e^{-j\omega t}. \qquad (2.19)$$

Следовательно, $|z| = e^{-\sigma t}$, arg(z) = ω*t* = 2π*ft* = 2π*f*/*f*$_Д$, где *f*$_Д$ – частота дискретизации, при этом ось ω отображается на *z*-плоскости единичной окружностью, правая сторона *p*-плоскости отображается внутрь окружности, а левая сторона – на внешнюю сторону окружности. При использовании символики z^{-1} отображение сторон *p*-плоскости на *z*-плоскости меняется местами.

2.1.4. Пространство *Z*-полиномов

Область сходимости. Полином *S*(*z*) (2.11) называют *z*-образом или *z*-изображением функции $s(kt)$. Преобразование имеет смысл для области тех значений *z*, в которой ряд *S*(*z*) сходится, т.е. сумма ряда представляет собой аналитическую функцию переменной *z*, не имеющую полюсов и особых точек [9, 10]:

$$\sum_{k=-\infty}^{\infty} |s_k| |z|^k < \infty. \qquad (2.20)$$

В общем случае множества *z*, для которых полиномы *S*(*z*) сходятся, образуют на *z*-плоскости определенные области, показанные на рис. 2.3.

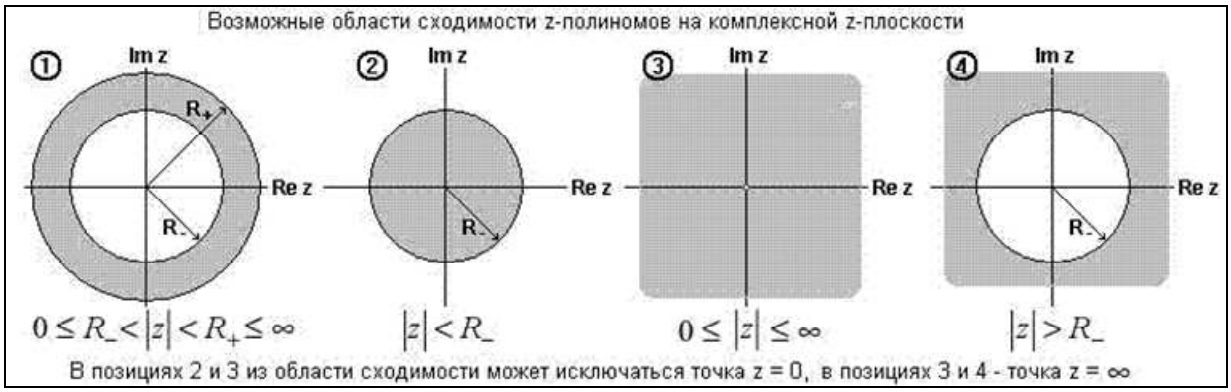


Рис. 2.3. Возможные области сходимости *z*-полиномов

Из приведенной выше связи *z*-преобразования с преобразованием Фурье следует, что если функция $s(t)$ имеет спектральное представление $S(\omega)$, то единичная окружность $|z| = \left|e^{-j\omega t}\right| = 1$ обязательно должна входить в область сходимости полинома $S(z)$. И наоборот, если область сходимости полинома $S(z)$ включает в себя единичную окружность, то дискретное преобразование Фурье функции $s(t)$ – прообраза полинома $S(z)$ должно существовать, а в противном случае – нет. Последнее следует из того, что *z*-преобразование, являясь более общим случаем преобразования дискретных функций, может существовать и для функций, для которых не существует преобразования Фурье. Примером этого может служить функция единичного скачка: $u_n = 1$, $n \geq 0$; $u_n = 0$, $n < 0$.

Для преобразования Фурье функции $u(n)$ не выполняется условие абсолютной суммируемости (энергия функции бесконечна). Но для *z*-преобразования имеем:

$$\sum_{k=-\infty}^{\infty} |u_k||z|^k = \sum_{k=0}^{\infty} |z|^k < \infty \quad \text{при } |z| < 1. \tag{2.21}$$

Рассмотрим примеры часто встречающихся на практике z-преобразований дискретных сигналов:

1. *Импульсы Кронекера*

В общем случае для импульса Кронекера в произвольной точке числовой оси

$$\begin{cases} \delta(k-n) = 1 \quad \text{при} \quad k = n; \\ \delta(k-n) = 0 \quad \text{при} \quad k \neq n. \end{cases} \tag{2.22}$$

$$X_\delta(z) = \sum_{k=-\infty}^{\infty} \delta(k-n) z^k = z^n. \tag{2.23}$$

Для импульса Кронекера в нулевой точке, соответственно,

$$X_\delta(z) = z^0 = 1.$$

Ряд $X_\delta(z)$ сходится на всей z-плоскости.

2. *Функция Хевисайда*

Единичный скачок, причинная последовательность бесконечной длины, например, импульсный отклик рекурсивного интегрирующего фильтра

$$\begin{cases} x(k) = 0 \quad \text{при} \quad k < 0; \\ x(k) = 1 \quad \text{при} \quad k \geq 0; \end{cases} \tag{2.24}$$

$$X(z) = \sum_{k=0}^{\infty} z^k = z^k. \tag{2.25}$$

Ряд сходится при $|z| < 1$, при этом его сумма равна: $X(z) = 1/(1-z)$.

Z-преобразование действительно везде внутри круга единичного радиуса с центром в начале координат.

При использовании символики z^{-1}

$$X(z) = 1/(1-z^{-1}) = z/(z-1), \quad |z| > 1. \tag{2.26}$$

На границе области аналитичности функция $X(z)$ имеет один простой полюс при $z = 1$.

3. *Экспоненциальная функция:*

$$\begin{cases} x(k) = 0 \quad \text{при} \quad k < 0; \\ x(k) = a^k \ \text{при} \quad k \geq 0; \end{cases} \tag{2.27}$$

$$X(z) = \sum_{k=-\infty}^{\infty} x(k) z^k = \sum_{k=0}^{\infty} a^k z^k = \sum_{k=0}^{\infty} (az)^k. \tag{2.28}$$

Как и в предыдущем случае, ряд сходится при $|az| < 1$, при этом

$$X(z) = 1/(1-az), \ |z| < 1/a. \tag{2.29}$$

При использовании символики z^{-1}

$$X(z) = z/(z-a), \ |z| > a. \tag{2.30}$$

4. *Комплексная экспонента*

$$x(k) = e^{j\omega k}, \ k \geq 0, \ x(k) = 0, \ k < 0; \tag{2.31}$$

$$X(z) = \sum_{k=0}^{\infty} e^{j\omega k} z^k = \sum_{k=0}^{\infty} (e^{j\omega} z)^k = 1/(1-ze^{j\omega}), \ |z| < 1. \tag{2.32}$$

Аналитическая форма *z*-образов существует для *z*-преобразований, если возможно свертывание степенного ряда в аналитическое выражение. Выше, в примерах *z*-преобразования, уже показывалось приведение к аналитической форме *z*-образов функции Хевисайда и экспоненциальной функции. Ниже в табл. 2.1 приводится *z*-трансформация ряда распространенных функций, которые могут использоваться для прямого и обратного преобразований.

Таблица 2.1

***Z*-трансформация распространенных функций**

Функция $s(k)$, $k \geq 0$	z-образ $S(z)$	Z^{-1} – образ $S(z)$
β	$\beta/(1-z)$, $\quad \lvert z\rvert < 1$	$\beta z/(z-1)$, $\quad \lvert z\rvert > 1$
$\beta\, k$	$\beta z/(1-z)^2$, $\quad \lvert z\rvert < 1$	$\beta z/(z-1)^2$, $\quad \lvert z\rvert > 1$
$\beta\, k^2$	$\beta z\,(1+z)/(1-z)^3$, $\quad \lvert z\rvert < 1$	$\beta z\,(z+1)/(z-1)^3$, $\quad \lvert z\rvert > 1$
$\beta\, \alpha^k$	$\beta/(1-z\alpha)$, $\quad \lvert z\rvert < 1/\alpha$	$\beta z/(z-\alpha)$, $\quad \lvert z\rvert > \alpha$
$\beta k\alpha^k$	$\beta\alpha z/(1-z\alpha)^2$, $\quad \lvert z\rvert < 1/\alpha$	$\beta\alpha z/(z-\alpha)^2$, $\quad \lvert z\rvert > \alpha$
$\cos \alpha k$	$(1-z\cos\alpha)/(1-2z\cos\alpha+z^2)$, $\lvert z\rvert < 1$	$z\,(z-\cos\alpha)/(z^2-2z\cos\alpha+1)$, $\lvert z\rvert > 1$
$\sin \alpha k$	$z\sin\alpha/(1-2z\cos\alpha+z^2)$, $\quad \lvert z\rvert < 1$	$z\sin\alpha/(z^2-2z\cos\alpha+1)$, $\quad \lvert z\rvert > 1$
$\beta \exp(-\alpha k)$	$\beta/(1-z\exp(-\alpha))$, $\quad \lvert z\rvert < 1/\exp(-\alpha)$	$\beta z/(z\text{-}\exp(-\alpha))$, $\quad \lvert z\rvert > \exp(-\alpha)$
$\beta k \exp(-\alpha k)$	$\beta z\exp(-\alpha)/(1-z\exp(-\alpha))^2$, $\lvert z\rvert < 1/\exp(-\alpha)$	$\beta z\exp(-\alpha)/(z\text{-}\exp(-\alpha))^2$, $\lvert z\rvert > \exp(-\alpha)$

В таблице приведены преобразования как для символики *z*, так и для символики z^{-1} (по Гуревичу), которая иногда бывает более удобной в некоторых математических операциях. Переход из одной символики в другую достаточно прост и выполняется заменой *z* в одной символике на 1/*z* в другой.

2.1.5. Обратное *Z*-преобразование

Обратное *z*-преобразование позволяет восстанавливать дискретную функцию по ее *z*-образу. Оно широко используется, например, при определении импульсных характеристик рекурсивных цифровых фильтров [1]. В символической форме

$$x(k) = TZ^{-1}[X(z)]. \qquad (2.33)$$

На практике $X(z)$ в процессе расчетов обычно выражается через отношение двух многочленов от *z*:

$$X(z) = \frac{b_0 + b_1 z + b_2 z^2 + \ldots + b_N z^N}{a_0 + a_1 z + a_2 z^2 + \ldots + a_M z^M} = x(0) + x(1)z + x(2)z^2 + \ldots \,. \qquad (2.34)$$

Самые распространенные методы обратного преобразования из этой формы $X(z)$:

- преобразование интегрированием по контуру (метод вычетов);
- метод разложения на элементарные дроби;
- метод разложения в степенной ряд.

Метод разложения в степенной ряд наиболее прост и пригоден для выполнения на компьютерах, но он не дает решения в аналитической форме. При задании большого числа точек обратного преобразования требуется также следить за возможным нарастанием числовых ошибок вследствие рекурсии его алгоритма.

Два первых метода позволяют получать результаты в аналитическом виде, но требуют вычисления полюсов функции $X(z)$, что может представлять трудности при высоком порядке функции. При высоких порядках полюсов потребуется также дифференцирование соответствующих порядков.

Преобразование интегрированием по контуру относится к числу математически строгих методов. Оно выполняется интегрированием по произвольному замкнутому контуру C, расположенному в области сходимости и окружающему все особые точки (нули и полюсы) z-образа. Интегрирование удобнее выполнять над полюсами, расположенными внутри контура, включающего центр системы координат, т.е. в символике z^{-1}. В этой символике мы и будем рассматривать примеры этого пункта. Контурный интеграл обратного преобразования

$$s_k = (1/2\pi j)\oint_c S(z) z^{k-1} d_z . \qquad (2.35)$$

Согласно теореме Коши о вычетах интеграл (2.35) равен сумме вычетов (Res) подынтегральной функции относительно всех полюсов этой функции, лежащих внутри контура интегрирования. Каждый вычет связан с определенным полюсом p_k:

$$\text{Res}[F(z), p_k] = \frac{1}{(m-1)!}\frac{d^{m-1}}{dz^{m-1}}[(z - p_k)F(z)] \quad \text{при } z = p_k , \qquad (2.36)$$

где $F(z) = z^{k-1}S(z)$, m – порядок полюса в точке p_k.

Пример 2.7

Z-образ функции: $X(z) = z^2 / (z - 0.5)(z - 1)^2$.

$x(k) = \text{Res}[F(z), p_1] + \text{Res}[F(z), p_2]$; $F(z) = z^{k-1} X(z) = z^{k+1} / (z - 0.5)(z - 1)^2$.

Функция $F(z)$ имеет простой полюс $p_1 = 0.5$ и полюс второго порядка $p_2 = 1$.

$\text{Res}[F(z), 0.5] = (z - 0.5)\, z^{k+1} / (z - 0.5)(z - 1)^2 = z^{k+1} / (z - 1)^2 \,|_{z=0.5} =$

$= 0.5\,(0.5)^k / (0.5)^2 = 2(0.5)^k$.

$\mathrm{Res}[F(z), 1] = \dfrac{d}{dz}[(z-1)^2 z^{k+1} / (z-0.5)(z-1)^2] =$

$= [(z-0.5)(k+1)z^k - z^{k+1}] / (z-0.5)^2 \,|_{z=1} = 2(k-1).$

Результат: $x(k) = 2[(k-1) + (0.5)^k]$.

Преобразование разложением на дроби. В этом методе z-образ (2.34) раскладывается на рациональные простые дроби с последующим почленным обратным преобразованием с помощью таблицы. Наиболее просто это выполняется, если функция $S(z)$ может быть разложена по степеням z в символике z^{-1}, т.е. представлена в следующем виде:

$$S(z) = s(0) + s(1)z^{-1} + s(2)z^{-2} + \dots . \qquad (2.37)$$

Соответственно, в выражении (2.34) отношение многочленов также должно быть в символике z^{-1}. Если полюсы $S(z)$ первого порядка и $N = M$, то (2.34) можно разложить на следующую сумму:

$$S(z) = B_0 + C_1 / (1 - p_1 z^{-1}) + C_2 / (1 - p_2 z^{-2}) + \dots + C_M / (1 - p_M z^{-M}) =$$

$$= B_0 + C_1 z / (z - p_1) + C_2 z / (z - p_2) + \dots + C_M z / (z - p_M) = B_0 + \sum_{k=1}^{M} C_k z / (z - p_k), \qquad (2.38)$$

где $B_0 = b_N / a_N$,.C_k – коэффициенты элементарных дробей, которые являются вычетами функции $S(z)$.

Для вычисления коэффициентов C_k умножим левую и правую стороны выражения (2.38) на $(z - p_k)/z$ и положим $z = p_k$, при этом в правой части за счет множителя $(z - p_k) = 0$ при $z = p_k$ обнуляются все члены суммы, кроме члена с C_k данного полюса, а в левой остается произведение $S(z)(z - p_k)/z$, что позволяет вычислить значения C_k:

$$C_k = S(z)(z - p_k) / z\big|_{z=p_k} . \qquad (2.39)$$

Если в (2.34) $N < M$, то значение B_0 равно нулю. Если функция $S(z)$ в точке $z = p_k$ имеет полюс m-го порядка, то коэффициент C_k заменяется суммой коэффициентов:

$$\sum_{i=1}^{m} D_i / (z - p_k)^i ; \qquad (2.40)$$

$$D_i = \frac{1}{(m-i)!} \frac{d^{m-i}}{dz^{m-i}} \left[\frac{X(z)(z - p_k)^m}{z} \right] \quad \text{при } z = p_k. \qquad (2.41)$$

Пример 2.8. Повторим пример преобразования данным способом z-образа функции $X(z) = z^2 / (z - 0.5)(z - 1)^2$, использованного в предыдущем примере.

Функция имеет простой полюс $p_1 = 0.5$ и полюс второго порядка $p_2 = 1$.

$X(z) = Cz/(z - 0.5) + D_1 z/(z - 1) + D_2 z/(z - 1)^2$;

$C = z/(z - 1)^2 = 0.5/(0.5 - 1)^2 = 2$;

$D_1 = \dfrac{d}{dz}[(z-1)^2 X(z)/z] = \dfrac{d}{dz}[z/(z-0.5)]\,|_{z=1} = -2;$

$D_2 = (z-1)^2\, X(z)/z = z/(z-0.5)\,|_{z=1} = 2;$

$X(z) = 2z/(z-0.5) + D_1 z/(z-1) + D_2 z/(z-1)^2.$

Обратное преобразование каждой простой дроби выполним по табл. 2.1.

Результат: $x(k) = 2(0.5)^k - 2 + 2k = 2[(k-1) + (0.5)^k]$. Результат аналогичен методу вычетов.

Если z-изображение имеет вид дробно-рациональной функции, то разложение на простые дроби с последующим применением таблицы соответствий обычно труда не представляет. Так, например,

$$S(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - az^{-1}} = \frac{b_0}{1 - az^{-1}} + \frac{b_1 z^{-1}}{1 - az^{-1}} + \frac{b_2 z^{-2}}{1 - az^{-1}}. \qquad (2.42)$$

По таблице соответствия 2.1

$$X(z) = 1/(1 - az^{-1}) \to x(k) = a^k. \qquad (2.43)$$

Отсюда, с учетом линейности преобразования и свойства задержки,

$$x(k) = b_0 a^k + b_1 a^{k-1} + b_2 a^{k-2}. \qquad (2.44)$$

При преобразовании функций со знаменателями более высоких порядков предварительно следует найти полюса функции. Например, для многочлена второго порядка с полюсами p_1 и p_2

$$S(z) = 1/(1 - a_1 z^{-1} + a_2 z^{-2}) = 1/\left[(1 - p_1 z^{-1})(1 - p_2 z^{-1})\right]. \qquad (2.45)$$

Представим $S(z)$ в виде суммы дробей с неизвестными коэффициентами b_1 и b_2:

$$S(z) = b_1/(1 - p_1 z^{-1}) + b_2/(1 - p_2 z^{-1}) = \frac{b_1 - b_1 p_2 z^{-1} + b_2 - b_2 p_1 z^{-1}}{\left[(1 - p_1 z^{-1})(1 - p_2 z^{-1})\right]}. \qquad (2.46)$$

При равенстве знаменателей в этих двух выражениях должны быть равны и числители:

$$b_1 + b_2 - (b_1 p_2 + b_2 p_1) z^{-1} = 1, \qquad (2.47)$$

а это обеспечивается равенством коэффициентов при одинаковых степенях z.

Отсюда получаем систему уравнений:

$$\begin{cases} b_1 + b_2 = 1; \\ b_1 p_2 + b_2 p_1 = 0. \end{cases} \qquad (2.48)$$

Решая эту систему уравнений, находим значения коэффициентов b_1 и b_2, подставляем коэффициенты в $S(z)$, выраженное в виде суммы дробей, и по таблице соответствия переводим дроби во временные функции.

Метод степенных рядов. Выражение (2.34) можно разложить непосредственно в степенной ряд путем деления в столбик, для чего числитель и знаменатель функции выражаются предварительно через нарастающий или уменьшающийся показатель степени z. Обратное z-преобразование степенного ряда очевидно.

Пример 2.9. Нарастающая степень z. $X(z) = (1 + 2z + z^2) / (1 – z + 0.4z^2)$.

$1 + 2z + z^2 \quad | \underline{1 – z + 0.4z^2}$

$\underline{1 – z + 0.4z^2} \quad 1 + 3z + 3.6z^2 + 2.4z^3 + 0.96z^4 + \ldots$ Ряд может быть бесконечным.

$3z + 0.6z^2$

$\underline{3z – 3z^2 + 1.2z^3}$

$3.6z^2 – 1.2z^3$

$\underline{3.6z^2 – 3.6z^3 + 1.44z^4}$

$2.4z^3 – 1.44z^4$

$\underline{2.4z^3 – 2.4z^4 + 0.96z^5}$

$0.96z^4 – 0.96z^5$

$0.96z^4 – 0.96z^5 + 0.384z^6$, и т.д.

Обратное преобразование выполняется путем идентификации коэффициентов степеней при z^k с k-отсчетами функции: $x(k) = \{1, 3, 3.6, 2.4, 0.96, \ldots\}$.

Пример 2.10. Уменьшающиеся номера степени z. $X(z) = (1 + 2z + z^2) : (1 – z + 0.4z^2) \rightarrow$ (деление на z^N числителя и знаменателя полинома) $\rightarrow (z^{-2} + 2z^{-1} + 1) : (z^{-2} – z^{-1} + 0.4)$.

$Z^{-2} + 2z^{-1} + 1 \quad | \underline{z^{-2} – z^{-1} + 0.4}$

$\underline{Z^{-2} – z^{-1} + 0.4} \quad 1 + 3z + 3.6z^2 + 2.4z^3 + 0.96z^4 + \ldots$ Результат тот же.

$3z^{-1} + 0.6$

$\underline{3z^{-1} – 3 + 1.2z}$

$3.6 – 1.2z$

$\underline{3.6 – 3.6z + 1.44z^2}$

$2.4z – 1.44z^2$

$\underline{2.4z – 2.4z^2 + 0.96z^3}$

$0.96z^2 – 0.96z^3$

$0.96z^2 – 0.96z^3 + 0.384z^4$, и т.д.

Метод деления полинома (2.34) можно выполнять рекурсивно:

$x(0) = b_0 / a_0$,

$x(1) = (b_1 – x(0)\, a_1) / a_0$,

$x(2) = (b_2 – x(1)\, a_1 – x(0)\, a_2) / a_0$,

…

$$x(n) = \left(b_n – \sum_{i=1}^{n} \left(x(n-i) a_i \right) \right) / a_0, \quad n = 1, 2, 3, \ldots$$

2.2. Дискретное преобразование Фурье и его свойства

2.2.1. Дискретные экспоненциальные функции

Дискретный сигнал $s(n)$, заданный на интервале $[0,\infty)$, можно записать в виде [4]:

$$s(n)=\sum_{k=0}^{\infty}c(k)\eta_k(n), \tag{2.49}$$

где

$$c(k)=E_k^{-1}\sum_{n=0}^{N-1}s(n)\eta_k(n). \tag{2.50}$$

Здесь $\{\eta_k(n)\}$ – совокупность базисных функций, определенных на том же интервале; $\{c(k)\}$ – коэффициенты разложения сигнала по базису $\{\eta_k(n)\}$, называемые спектром сигнала; E_k – энергия k-й базисной функции.

Из этих выражений следует, что сигнал можно задать либо его отсчетами в дискретных точках, либо набором спектральных коэффициентов, причем спектральное представление неоднозначно и зависит от выбранной системы базисных функций. Выбор системы $\{\eta_k(n)\}$ определяется соображениями практического или математического удобства.

Математические преобразования применяются к сигналу для того, чтобы получить о нем какую-то дополнительную информацию, недоступную в исходном виде. Среди многих известных преобразований сигналов наиболее популярным является преобразование Фурье.

Дискретное преобразование Фурье оперирует дискретной по времени выборкой периодического сигнала во временной области $\{s(n)\}=$ $=\{s(0),s(1),...,s(N-1)\}$. Для того чтобы быть представленным в виде суммы синусоид, сигнал должен быть периодическим. Но в качестве набора входных данных для ДПФ доступно только конечное число отсчетов (N).

В дискретном преобразовании Фурье используется система дискретных экспоненциальных функций (ДЭФ), которая определяется следующим выражением:

$$def(k,n)=\exp(-j\frac{2\pi}{N}kn)=\cos\left(\frac{2\pi}{N}kn\right)-j\sin\left(\frac{2\pi}{N}kn\right), \tag{2.51}$$

где $j=\sqrt{-1}$; k и n принимают целочисленные значения $(0,1,2,...,N-1)$. Переменную k отождествляют с номером функции, а переменную n – с номером отсчета.

Введем обозначение $W = \exp(-j\frac{2\pi}{N})$. Тогда $\mathrm{def}(k,n) = W^{kn} = W_N^{k\cdot n}$.

Всю систему ДЭФ можно записать в виде матрицы V, строки которой нумеруются переменной k, столбцы переменной n, а в пересечении k-й строки и n-го столбца записана величина W^{kn}. Такое представление называется матричным представлением ДЭФ.

$$V = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ k \\ \vdots \\ N-1 \end{array} \begin{bmatrix} & & & | & & \\ & & & | & & \\ & & & | & & \\ -- & -- & -- & W^{kn} & -- & -- \\ & & & | & & \\ & & & | & & \end{bmatrix}$$

$$0 \quad 1 \quad . \quad n \qquad .. \qquad N-1$$

Например, для $N = 8$ матрица V имеет следующий вид:

$$V = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} & W^{14} \\ W^0 & W^3 & W^6 & W^9 & W^{12} & W^{15} & W^{18} & W^{21} \\ W^0 & W^4 & W^8 & W^{12} & W^{16} & W^{20} & W^{24} & W^{28} \\ W^0 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} & W^{30} & W^{35} \\ W^0 & W^6 & W^{12} & W^{18} & W^{24} & W^{30} & W^{36} & W^{42} \\ W^0 & W^7 & W^{14} & W^{21} & W^{28} & W^{35} & W^{42} & W^{49} \end{bmatrix}$$

Основные свойства ДЭФ [4]:

1. *Ортогональность:*

$$\sum_{n=0}^{N-1} W^{kn} W^{-\ln} = \begin{cases} N, \text{ если } (k-l) \equiv 0 \bmod N; \\ 0, \text{ если } (k-l) \neq 0 \bmod N. \end{cases}$$

Свойство ортогональности показывает, что скалярное произведение любых двух строк матрицы V, одна из которых взята с комплексно сопря-

женными элементами, равно нулю, если строки различны, и равно N, если они совпадают. Матричная запись этого свойства имеет следующий вид:

$$V(V^T)^* = NI,$$

где знак * означает взятие комплексного сопряжения для всех элементов матрицы: I – единичная матрица.

2. *Периодичность.* Если $kn = Nl + r$, то $W^{kn} = W^{N \cdot l} W^r = W^r$, что позволяет записать элементы матрицы V с минимальными степенями (фазами). Например, для $N = 8$

$$V = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ W^0 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \\ W^0 & W^5 & W^2 & W^7 & W^4 & W^1 & W^6 & W^3 \\ W^0 & W^6 & W^4 & W^2 & W^0 & W^6 & W^4 & W^2 \\ W^0 & W^7 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix}.$$

3. *Симметричность:*

$$V = V^T.$$

Свойство симметричности позволяет легко найти обратную матрицу для матрицы V. Матричное выражение для ортогональности запишется:

$$VV^* = NI.$$

Умножив обе части этого равенства справа на V^{-1}, получим

$$V^{-1} = N^{-1}V^*.$$

4. *Мультипликативность:*

по номеру функции

$$\mathrm{def}(k_1, n)\mathrm{def}(k_2, n) = \mathrm{def}(k_1 + k_2, n);$$

по номеру отсчета

$$\mathrm{def}(k, n_1)\mathrm{def}(k, n_2) = \mathrm{def}(k, n_1 + n_2).$$

Свойство означает, что при умножении любых двух строк (столбцов) матрицы V получается соответственно строка (столбец) той же матрицы, номер строки (столбца) равен сумме номеров сомножителей.

Подобно обычным комплексным экспоненциальным функциям, ДЭФ можно изобразить на плоскости в виде вращающегося вектора единичной длины, проекции которого на оси абсцисс и ординат дают действительную и мнимую части функции (рис. 2.4). Разница будет заключаться в том, что если у обычных функций этот вектор вращается непрерывно, то в случае ДЭФ он вращается скачкообразно, проходя при изменении n на единицу угол $2\pi k/N$ рад. В общей сложности на интервале N вектор проходит угол $2\pi k$ рад, т.е. совершает ровно k оборотов. Движение комплексно-сопряженного вектора $\overline{def(k,n)} = def(N-k,n)$ при изменении n на единицу можно представить как поворот на один оборот и возврат обратно на угол $2\pi k/N$ рад, так, что создается иллюзия вращения в противоположную сторону (стробоскопический эффект). Такому представлению соответствует запись системы ДЭФ в виде матрицы с полными фазами, а матрица с минимальными фазами не дает представления об истинной скорости вращения вектора. Поэтому величину W^{kn} называют *поворачивающим множителем*.

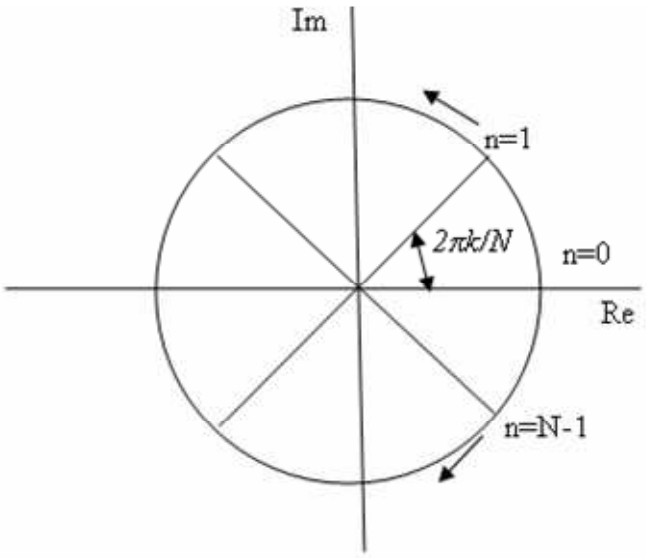


Рис. 2.4. Представление ДЭФ в виде вектора

По формуле Эйлера можно выразить значения вещественной и мнимой частей поворачивающих множителей через косинус и синус угла поворота θ соответственно:

$$e^{-j\theta} = \cos(\theta) - j\sin(\theta). \qquad (2.52)$$

Более того, при значениях $k \geq \frac{N}{2}$ значения поворачивающих множителей можно найти из соображения симметрии:

$$e^{-j\frac{2\pi(\frac{N}{2}+k)}{N}} = e^{-j\pi}e^{-j\frac{2\pi}{N}k} = -e^{-j\frac{2\pi}{N}k} \quad \text{при } 0 \leq k \leq \frac{N}{2}.$$

Вычисление поворачивающегося множителя для восьмиточечной последовательности показано на рис. 2.5, а значения поворачивающегося множителя приведены в табл. 2.2.

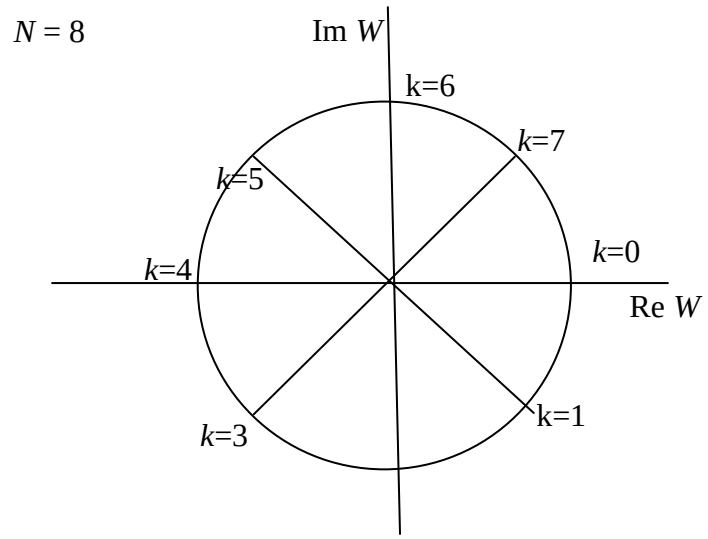


Рис. 2.5. Представление поворачивающегося множителя для восьмиточечной последовательности

Таблица 2.2

Значения поворачивающегося множителя для восьмиточечной последовательности

Значения *k*	Поворачивающий множитель
0	1
1	$\cos\left(\frac{\pi}{4}\right) - j\sin\left(\frac{\pi}{4}\right)$
2	-*j*
3	$-\cos\left(\frac{\pi}{4}\right) - j\sin\left(\frac{\pi}{4}\right)$
4	-1
5	$-\cos\left(\frac{\pi}{4}\right) + j\sin\left(\frac{\pi}{4}\right)$
6	j
7	$\cos\left(\frac{\pi}{4}\right) + j\sin\left(\frac{\pi}{4}\right)$

2.2.2. Дискретное преобразование Фурье и его свойства

Пара дискретного преобразования Фурье последовательности $\{s(n)\} = \{s(0), s(1), ..., s(N-1)\}$ определяется следующими равенствами [4]:

$$f(k) = \sum_{n=0}^{N-1} s(n) W^{kn}, \quad k = 0,1,2,...,N-1; \tag{2.53}$$

$$s(n) = N^{-1} \sum_{k=0}^{N-1} f(k) W^{-kn}, \quad n = 0,1,2,...,N-1, \tag{2.54}$$

где $\{f(k)\}$ – дискретный спектр.

Выражение (2.53) называется прямым преобразованием, а выражение (2.54) – обратным. Равенства (2.53) и (2.54) представляют собой экспоненциальную форму записи ДПФ. Как уже было сказано, ДПФ устанавливает связь между временными и частотным представлениями сигнала при разложении в базисе гармонических функций.

В матричной форме ДПФ имеет вид:

$$F = VS\ ;$$

$$S = N^{-1}V^{*}F.$$

Матрицы прямого и обратного преобразований V и V^* соответственно состоят из комплексно-сопряженных элементов W^{kn} и W^{-kn}. Для $k = n = 0$ строки обеих матриц совпадают и состоят только из единиц. Поэтому $s(0) = N^{-1}f(0)$. Для $k \neq 0$ рассмотрим следующее равенство:

$$W^{(N-k)n} = W^{Nn}W^{-kn} = W^{-kn}.$$

Из него следует, что строки матрицы V^* совпадают со строками матрицы V, но записаны в обратном порядке. Поэтому для вычисления обратного ДПФ вместо V^* можно использовать матрицу прямого преобразования V, но результат умножения на эту матрицу следует прочесть в обратном порядке и уменьшить в N раз.

Основные области применения ДПФ:

- цифровой спектральный анализ:
  - анализаторы спектра;
  - обработка речи;
  - обработка изображений;
  - распознавание образов;
  - проектирование фильтров;
  - вычисление импульсной характеристики по частотной;
  - вычисление частотной характеристики по импульсной;
- быстрое преобразование Фурье (БПФ) – простой алгоритм для эффективного вычисления ДПФ.

Пример 2.11. ДПФ для $\{s(n)\} = \{1,0,0,1\}$.

Матрица ДЭФ для выполнения ДПФ входного сигнала с количеством отсчетов $N = 4$ имеет вид:

$$V = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^3 & W^2 & W^1 \end{bmatrix}.$$

Поворачивающийся множитель W^{kn} для $N = 4$ на комплексной плоскости представлен на рис. 2.6.

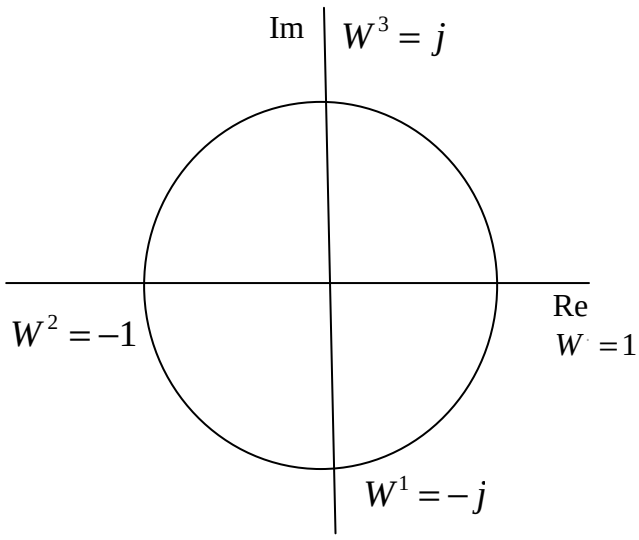


Рис. 2.6. Представление поворачивающегося множителя для четырехточечной последовательности

Следовательно,

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T = \begin{bmatrix} 2 \\ 1+j \\ 0 \\ 1-j \end{bmatrix}^T .$$

Таким образом, временному ряду $\{1,0,0,1\}$ соответствует его ДПФ-образ вида $\{2,1+j,0,1-j\}$.

ДПФ принято представлять в виде графиков зависимости $|f(k)|$ от номера гармоники k или от частоты ω. Чтобы определить ω, необходимо знать значение интервала дискретизации. Если предположить, что представленная выше последовательность была дискретизована с частотой 8 кГц, то $T = 1/(8 \cdot 10^3) = 125$ мкс. Тогда $\omega = 2\pi / (NT) = 2\pi / (4 \cdot 125 \cdot 10^{-6}) = 12{,}57 \cdot 10^3$ рад/с. Следовательно, $2\omega = 25{,}14 \cdot 10^3$ рад/с, $3\omega = 37{,}71 \cdot 10^3$ рад/с.

Свойства ДПФ [4]:

1. *Периодичность*. В силу периодичности ДЭФ функции $f(k)$ и $s(n)$ также периодичны, т.е.

$$f(Nl \pm k) = f(\pm k); \quad s(Nl \pm n) = s(\pm n). \qquad (2.55)$$

2. *Связь с коэффициентами ряда Фурье*. Если частота дискретизации выбрана в соответствии с теоремой Котельникова, то при дискретизации периодической аналоговой функций $s(t)$ ДПФ позволяет по выборкам $s(n)$ найти спектр $f(k)$, который на интервале $0 \le k \le N-1$ равен спектру исходной функции $s(t)$. При этом первые $\left(\frac{N}{2}-1\right)$ точек функции $f(k)$ соответствуют спектральным линиям на положительных частотах, а последние $\left(\frac{N}{2}-1\right)$ точек $f(k)$ – спектральным линиям на отрицательных частотах. В обратном преобразовании первые $\left(\frac{N}{2}-1\right)$ линий функции $s(n)$ со-

ответствуют области положительных времен, а последние $\left(\frac{N}{2}-1\right)$ линий – области отрицательных времен.

3. *Линейность*. Пусть даны последовательности $x(n)$ и $y(n)$, для которых ДПФ равны соответственно $f_x(k)$ и $f_y(k)$. Рассмотрим взвешенную сумму этих последовательностей $z(n)=ax(n)+by(n)$. Спектр последовательности $z(n)$ равен аналогичной взвешенной сумме спектров последовательностей $x(n)$ и $y(n)$, т.е.

$$f_z(k)=af_x(k)+bf_y(k). \tag{2.56}$$

4. *Инвариантность относительно сдвига по времени и частоте.* Пусть последовательность $z(n)$ образована сдвигом по времени последовательности $s(n)$, т.е. $z(n)=s(n\pm h)$. Тогда

$$f_z(k)=W^{\mp kh}f_x(k). \tag{2.57}$$

Данное свойство показывает, что при сдвиге по времени амплитудный спектр (величина амплитуд отдельных гармоник) не меняется. Изменениям подвергаются только фазы гармонических составляющих (фазовый спектр).

Аналогичное свойство справедливо и для обратного преобразования, а именно: если $l(k)=f(k\pm h)$, то

$$S_F(n)=W^{\pm nh}s_l(n). \tag{2.58}$$

5. *Теорема о свертке*

Пусть $\{f_s(k)\}$ и $\{f_h(k)\}$ – спектры последовательностей $\{s(n)\}$ и $\{h(n)\}$ соответственно. Теорема о свертке утверждает, что спектр свертки равен произведению спектров сворачиваемых последовательностей, т.е.

$$f_y(k)=f_s(k)\,f_h(k). \tag{2.59}$$

Матричное представление свертки имеет вид:

$$\begin{bmatrix} y(0)\\ y(1)\\ .\\ .\\ .\\ y(N-1)\end{bmatrix}=\begin{bmatrix} h(0) & h(1) & .... & ... & ... & h(N-1)\\ h(1) & h(2) & ... & ... & ... & h(0)\\ h(2) & ... & ... & ... & ... & ...\\ ... & ... & ... & ... & ... & ...\\ h(N-1) & h(0) & ... & ... & ... & h(N-2)\end{bmatrix}\begin{bmatrix} s(0)\\ s(N-1)\\ s(N-2)\\ .\\ .\\ s(1)\end{bmatrix}.$$

Теорема позволяет вычислить свертку при помощи ДПФ по формуле

$$\{y(n)\}=ДПФ^{-1}(ДПФ\{s(n)\}\,ДПФ\{h(n)\}. \tag{2.60}$$

6. *Теорема о корреляции.*

Спектр корреляционной функции последовательностей $\{s(n)\}$ и $\{h(n)\}$ равен произведению их спектров, причем один из спектров берется в комплексном сопряжении, т.е.

$$f_r(k) = f_h^*(k)\, f_s(k) = f_h(k)\, f_s^*(k). \qquad (2.61)$$

Матричное представление корреляции:

$$\begin{bmatrix} r(0) \\ r(1) \\ . \\ . \\ . \\ r(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(1) & .... & h(N-1) \\ h(N-1) & h(0) & .... & h(N-2) \\ .... & .... & .... & .... \\ h(1) & h(2) & .... & h(0) \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ . \\ . \\ . \\ s(N-1) \end{bmatrix}.$$

7. *ДПФ вещественных последовательностей*

Определяется согласно выражению

$$f(k) = \sum_{n=0}^{N-1} s(n)\cos(\frac{2\pi}{N}kn) - j\sum_{n=0}^{N-1} s(n)\sin(\frac{2\pi}{N}kn). \qquad (2.62)$$

Отсюда особенности:

а) спектральные коэффициенты комплексно сопряжены относительно счета $N/2$, т.е.

$$f\left(\frac{N}{2}+l\right) = f^*\left(\frac{N}{2}-l\right),\ l = 0,1,...,N/2,$$

а коэффициент $f(N/2)$ для четных N всегда действителен;

б) если последовательность $\{s(n)\}$ четная, т.е. $\{s(n)\} = \{s(-n)\}$, то ее ДПФ $\{f(k)\}$ – вещественная последовательность. Аналогично, если $\{s(n)\}$ нечетная, т.е. $\{s(n)\} = -\{s(-n)\}$, то ее ДПФ $f(k)$ есть чисто мнимая последовательность.

8. *Равенство Парсеваля*:

$$\sum_{n=0}^{N-1} s^2(n) = N^{-1}\sum_{k=0}^{N-1} |f(k)|^2. \qquad (2.63)$$

Данное соотношение показывает, что энергия сигнала равна суммарной энергии спектральных компонент, т.е. энергия сигнала больше никуда не расходуется.

2.3. Алгоритмы быстрого преобразования Фурье

Вычисление преобразований Фурье требует очень большого числа N^2 умножений. Существует способ выполнить эти преобразования значительно быстрее – примерно за $N \log_2 N$ операций умножения. Этот способ называется быстрым преобразованием Фурье (БПФ, FFT – fast Fourier transform). Он основан на том, что среди множителей есть много повторяющихся значений (в силу, например, своей периодичности). Алгоритм БПФ группирует слагаемые с одинаковыми множителями, значительно сокращая число умножений. В результате быстродействие БПФ может в зависимости от N в сотни раз превосходить быстродействие стандартного алгоритма. При этом следует подчеркнуть, что алгоритм БПФ является точным. Он даже точнее стандартного, т.к., сокращая число операций, он приводит к меньшим ошибкам округления.

Однако у большинства алгоритмов БПФ есть особенность: они способны работать лишь тогда, когда длина анализируемого сигнала N является степенью двойки. Обычно это не представляет большой проблемы, так как анализируемый сигнал всегда можно дополнить нулями до необходимого размера. Число N называется размером или длиной БПФ (FFT size).

Наибольшее распространение получил алгоритм БПФ с основанием 2, известный как алгоритм БПФ Кули-Тьюки (по имени разработчиков) и впервые опубликованный в 1965 г. в США. Существуют два эквивалентных по эффективности алгоритма БПФ с основанием 2 – с прореживанием по времени и с прореживанием по частоте.

2.3.1. Алгоритм БПФ с прореживанием по времени

Основная идея алгоритма БПФ с прореживанием по времени заключается в поэтапном вычислении N-точечного ДПФ ($N = 2^v$) на v этапах, на каждом из которых текущее ДПФ определяется как комбинация ДПФ вдвое меньшей размерности.

Алгоритм БПФ с прореживанием по времени в общем виде можно представить [6]:

- задание начальных условий;
- отсчеты N-точечной последовательности расставляются по определенному правилу;
- на первом этапе определяется 2-точечное ДПФ каждой пары отсчетов последовательности;
- на вторам этапе определяются 4-точечные ДПФ как комбинация 2-точечных ДПФ;
- на i-м этапе определяются 2^i-точечные ДПФ как комбинация 2^{i-1} -точечных ДПФ;

- на (v-1)-м этапе определяются N/2-точечные ДПФ как комбинация N/4-точечных ДПФ;
- на v-м (последнем) этапе определяется искомое N-точечное ДПФ как комбинация N/2-точечных ДПФ, отсчеты ДПФ следуют в естественном порядке, $k = 0,1,...,(N-1)$.

Таким образом, для реализации алгоритма БПФ необходимо знать правило начальной расстановки отсчетов N-точечной последовательности и общую формулу расчета ДПФ для произвольного i-го этапа.

Получить их можно следующим образом.

Рассмотрим случай, когда алгоритм БПФ состоит из одного, v-го этапа. Начальные условия одноэтапного алгоритма БПФ формируются в результате однократного разбиения исходной N-точечной последовательности на две N/2-точечные с выделением отдельно четных и нечетных отсчетов:

$$\{s(2n)\} = \{s(0), s(2), ..., s(N-2)\};$$
$$\{s(2n+1)\} = \{s(1), s(3), ..., s(N-1)\}.$$

Таким образом, начальная расстановка отсчетов производится по правилу:

N/2 *четных* отсчетов; *N/2 нечетных* отсчетов:

$$\{s(0), s(2), ..., s(N-2); s(1), s(3), ..., s(N-1)\}.$$

Это позволяет разбить сумму в ДПФ на две:

$$f(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n) W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} s(2n+1) W_N^{(2n+1)k} =$$

$$= \sum_{n=0}^{\frac{N}{2}-1} s(2n) W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} s(2n+1) W_N^{2nk}, k = 0,1,..., N-1. \qquad (2.64)$$

Представив поворачивающий множитель W_N^{2nk} в виде

$$W_N^{2nk} = e^{-j\frac{2\pi}{N}2nk} = e^{-j\frac{2\pi}{N/2}nk} = W_{N/2}^{nk}, \qquad (2.65)$$

ДПФ можно записать как

$$f(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n) W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} s(2n+1) W_{N/2}^{nk}, k = 0,1,..., N-1. \qquad (2.66)$$

Анализ выражения показывает, что каждая из сумм представляет собой N/2-точечное ДПФ: первая сумма – ДПФ последовательности четных отсчетов, а вторая – нечетных. Причем каждое из N/2-точечных ДПФ определяется при $k = 0, 1, ..., N-1$.

Введем обозначения:

$$f_0^{v}(k) = f(k);$$

$$f_0^{v-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n) W_{N/2}^{nk}; \qquad (2.67)$$

$$f_1^{v-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n+1) W_{N/2}^{nk},$$

где *верхний* индекс указывает номер этапа алгоритма БПФ, а нижний – номер ДПФ, четный (четное ДПФ) и нечетный (нечетное ДПФ).

С учетом введенных обозначений ДПФ можно рассчитать: *N*-точечное ДПФ определяется как комбинация двух *N*/2-точечных ДПФ – четного $f_0^{v-1}(k)$ и нечетного $f_1^{v-1}(k)$:

$$f^{v}(k) = f_0^{v-1}(k) + W_N^k f_1^{v-1}(k), \quad k = 0,1,...,N-1. \qquad (2.68)$$

N/2-точечные ДПФ $f_0^{v-1}(k)$ и $f_1^{v-1}(k)$ – периодические функции переменной k с периодом *N*/2, нет необходимости определять их при k = 0,1,..., N – 1, достаточно только определить их при k = 0,1,…, $\frac{N}{2}$ – 1, а затем повторить при $k = \frac{N}{2}, \frac{N}{2}+1, \ldots, N-1$, т.е.

$$f_0^{v-1}(k) = f_0^{v-1}(k + \frac{N}{2}), \quad k = 0,1,...,\frac{N}{2}-1;$$

$$f_1^{v-1}(k) = f_1^{v-1}(k + \frac{N}{2}), \quad k = 0,1,...,\frac{N}{2}-1; \qquad (2.69)$$

Поворачивающий множитель при $k = \frac{N}{2}, \frac{N}{2}+1, \ldots, N-1$ равен

$$W_N^{k+\frac{N}{2}} = W_N^k W_N^{N/2} = W_N^k e^{-j\frac{2\pi}{N}\frac{N}{2}} = -W_N^k, \quad k = 0,1,...,\frac{N}{2}-1, \qquad (2.70)$$

следовательно, и его достаточно определить при k = 0,1,…, $\frac{N}{2}$ – 1, а затем, изменив знак, повторить при k = $\frac{N}{2}, \frac{N}{2}+1, \ldots, N-1$.

Рассмотренные свойства позволяют представить ДПФ в виде

$$\begin{cases} f_0^{v}(k) = f_0^{v-1}(k) + W_N^k f_1^{v-1}(k); \\ f_0^{v}(k + \frac{N}{2}) = f_0^{v-1}(k) - W_N^k f_1^{v-1}(k); \\ k = 0,1,...,\frac{N}{2}-1. \end{cases} \qquad (2.71)$$

Анализ полученного выражения показывает, что расчет ДПФ упрощается за счет распараллеливания вычислений при $k = 0,1,\ldots,\frac{N}{2} - 1$ и $k = \frac{N}{2},\frac{N}{2} + 1,\ldots, N - 1$ по верхней и нижней формулам соответственно.

Если алгоритм БПФ состоит из двух, v-го и $(v - 1)$-го этапов, то начальные условия алгоритма формируются в результате последовательного двукратного разбиения N-точечной последовательности: сначала N-точечная последовательность разбивается на две $N/2$-точечные, а затем каждая из этих последовательностей вновь разбивается на две $N/4$-точечные – четных и нечетных отсчетов по порядку их следования, считая от нуля:

– $N/2$-точсчная последовательность четных отсчетов, полученная для БПФ, состоящего из одного этапа, разбивается на две $N/4$-точечные последовательности четных и нечетных отсчетов:

$$\{s(0), s(4),..., s(N-4)\};$$

$$\{s(2), s(6),..., s(N-2)\};$$

– $N/2$-точечная последовательность нечетных отсчетов, полученная для БПФ, состоящего из одного этапа, разбивается на две $N/4$-точечные последовательности четных и нечетных отсчетов:

$$\{s(1), s(5),..., s(N-3)\};$$

$$\{s(3), s(7),..., s(N-1)\}.$$

Таким образом, начальная расстановка отсчетов производится по правилу:

$$\{s(0), s(4),..., s(N-4); s(2), s(6),..., s(N-2)\};$$

$$\{s(1), s(5),..., s(N-3); s(3), s(7),..., s(N-1)\}.$$

На $(v - 1)$-м этапе определяются два $N/2$-точечных ДПФ, причем каждое из них – как комбинация двух $N/4$-точечных ДПФ:

– $N/2$-точечное ДПФ $f_0^{v-1}(k)$ – как комбинация $N/4$-точечных ДПФ – четного $f_0^{v-2}(k)$ и нечетного $f_1^{v-2}(k)$;

– $N/2$-точечное ДПФ $f_1^{v-1}(k)$ – как комбинация $N/4$-точечных ДПФ – четного $f_2^{v-2}(k)$ и нечетного $f_3^{v-2}(k)$.

$N/2$-точечные ДПФ $f_0^{v-1}(k)$ и $f_1^{v-1}(k)$ определяются но формуле (2.71), в которой индекс v уменьшается на единицу, а размерности ДПФ и поворачивающего множителя понижаются вдвое, следовательно, два $N/2$-точечных ДПФ, $f_0^{v-1}(k)$ и $f_1^{v-1}(k)$, определяются:

$$\begin{cases} f_0^{v-1}(k) = f_0^{v-2}(k) + W_{N/2}^k f_1^{v-2}(k) \\ f_0^{v-1}(k+\frac{N}{4}) = f_0^{v-2}(k) - W_{N/2}^k f_1^{v-2}(k) \\ k = 0,1,\ldots,\frac{N}{4}-1; \end{cases}$$

$$\begin{cases} f_1^{v-1}(k) = f_2^{v-2}(k) + W_{N/2}^k f_3^{v-2}(k) \\ f_1^{v-1}(k+\frac{N}{4}) = f_2^{v-2}(k) - W_{N/2}^k f_3^{v-2}(k) \\ k = 0,1,\ldots,\frac{N}{4}-1; \end{cases} \qquad (2.72)$$

На v-м этапе N-точечное ДПФ $f_0^v(k)$ определяется как комбинация $N/2$-точечных ДПФ.

Если алгоритм БПФ состоит из трех, v-го, $(v-1)$-го и $(v-2)$-го этапов, то аналогично начальные условия формируются а результате трехкратного разбиения N-точечной последовательности, на $(v-1)$-м этапе каждое из четырех $N/4$-точечных ДПФ определяется по формуле (2.72), в которой индекс v уменьшается на два, а размерности ДПФ и поворачивающего множителя понижаются в 4 раза, на $(v-1)$-м и v-м этапах два $N/2$-точечных и N-точечное ДПФ определяются согласно двухэтапному алгоритму. Подобный подход можно применить и для v-этапного алгоритма БПФ с прореживанием по времени.

Начальные условия формируются в результате v-кратного разбиения N-точечной последовательности, а сформированная последовательность называется прореженной по времени.

Процедура многократного прореживания приводит к тому, что исходные данные располагаются не в естественном, а в двоично-инверсном порядке.

Например, для $N=8$ естественный и двоично-инверсный порядок расположения отчетов запишутся следующим образом:

Естественный порядок	Двоично-инверсный порядок
000(0)	000(0)
001(1)	100(4)
010(2)	010(2)
011(3)	110(6)
100(4)	001(1)
101(5)	101(5)
110(6)	011(3)
111(7)	111(7)

Общая формула расчета ДПФ на произвольной i-м этапе, полученная на основе (2.72), имеет вид:

$$\begin{cases} f_m^i(k) = f_{2m}^{i-1}(k) + W_L^k f_{2m+1}^{i-1}(k); \\ f_m^i(k+\dfrac{L}{2}) = f_{2m}^{i-1}(k) - W_L^k f_{2m+1}^{i-1}(k); \\ i = 1,2,...,v; \\ m = 0,1,...,M-1 \\ k = 0,1,...,\dfrac{L}{2}-1, \end{cases} \tag{2.73}$$

где: i – номер этапа; m – номер ДПФ; k – номер отсчета ДПФ; M – количество L-точечных ДПФ;

$$M = \frac{N}{2^i} = \frac{2^v}{2^i} = 2^{v-i}; \tag{2.74}$$

L – размерность ДПФ;

$$L = 2^i; \tag{2.75}$$

$f_m^i(k)$ – L-точечное ДПФ;

$f_{2m}^{i-1}(k)$, $f_{2m+1}^{i-1}(k)$ – четное и нечетное $L/2$-точечное ДПФ соответственно.

Согласно (2.73) L-точечное ДПФ определяется параллельно: первая половина отсчетов L-точечного ДПФ – по верхней формуле; вторая половина отсчетов L-точечного ДПФ – по нижней формуле.

Рассмотрим особенности *первого* этапа алгоритма (i = 1). По формуле (2.75) размерность L ДПФ

$$L = 2^1 = 2,$$

а по формуле (2.74) количество 2-точечных ДПФ

$$M = 2^{v-i} = 2^{v-1} = \frac{N}{2}.$$

Общая формула для ДПФ:

$$\begin{cases} f_m^1(0) = f_{2m}^0(0) + W_2^0 f_{2m+1}^0(0); \\ f_m^1(1) = f_{2m}^0(0) - W_2^0 f_{2m+1}^0(0), \end{cases} \tag{2.76}$$

где поворачивающий множитель W_2^0 = 1 сохранен для единообразия с общей формулой (2.73). В правой части (2.76) согласно принятым обозначениям $f_{2m}^0(0)$ и $f_{2m+1}^0(0)$ — отсчеты ДПФ нулевого этапа (i = 0), однако такого этапа нет, поэтому для вычисления 2-точечных ДПФ по общей формуле (2.73) перед выполнением первого этапа задаются начальные условия: каждой паре отсчетов ДПФ, состоящей из четного и нечетного, присваивают

значения пары отсчетов прореженной последовательности четного и нечетного отсчетов соответственно:

$$\begin{cases} f_{2m}^{0}(0) = \tilde{s}_{2m}; \\ f_{2m+1}^{0}(0) = \tilde{s}_{2m+1}, \end{cases} \qquad (2.77)$$

где $\tilde{s}_{2m}$ и $\tilde{s}_{2m+1}$ – условные обозначения четного и нечетного отсчетов 2-точечной последовательности, полученные в результате v-кратного разбиения исходной N-точечной последовательности.

Анализируя выражение (2.73), можно сказать, что алгоритм БПФ с прореживанием по времени представляет собой v-этапную процедуру типа «цикл в цикле», где:

– внешний цикл организуется по переменной i: $I = 1, 2,\ldots,v$;

– первый внутренний цикл (при фиксированном i) – по переменной m: $m = 0, 1,..., M - 1$;

– второй внутренний цикл (при фиксированных i и m) – по переменной k: $k = 0,1,..., (L/2 - 1)$. В цикле вычисляется k-й отсчет L-точечного ДПФ по формуле (2.73).

На выходе алгоритма БПФ с прореживанием по времени получаем N-точечное ДПФ, отсчеты которого следуют в естественном порядке: $k = 0, 1,..., (N - 1)$.

Основной операцией алгоритма БПФ, называемой базовой, является одновременное (параллельное) вычисление двух отсчетов ДПФ – k-го и $(k + L/2)$-го по двум верхним формулам (2.73). На рис. 2.7 приведен направленный граф базовой операции, получивший название «бабочка» по ассоциации с изображением графа. Кружок обозначает арифметическую операцию сложения/вычитания, верхний выход соответствует сумме, а нижний – разности; стрелка обозначает операцию умножения на поворачивающий множитель, стоящий над ней.

Алгоритм БПФ обычно представляют в виде направленного графа – совокупности «бабочек».

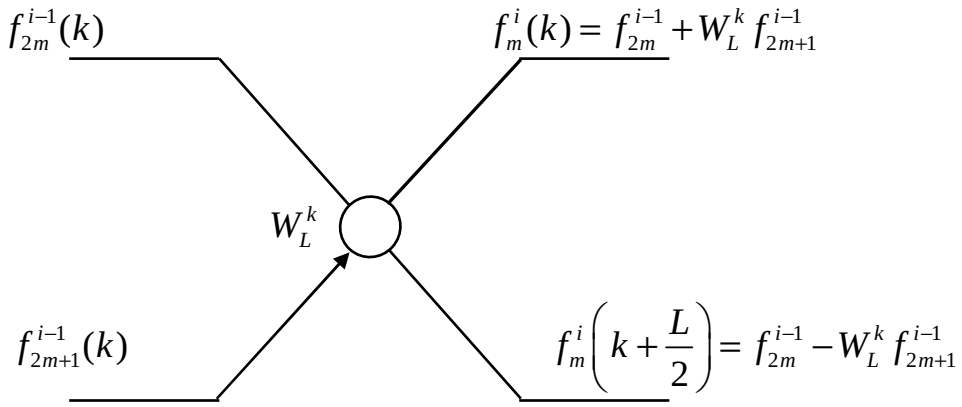


Рис. 2.7. Граф базовой операции алгоритма БПФ с прореживанием по времени

Количество «бабочек», т.е. базовых операций алгоритма, на любом i-м этапе одинаково: согласно (2.73) на i-м этапе при фиксированном значении m и k = 0, 1,..., L/2 требуется L/2 «бабочек», следовательно, при m = = 0, 1,..., (M – 1) будем иметь ML/2 «бабочек».

Алгоритм БПФ с прореживанием по времени рассмотрим на примере вычисления 8-точечного ДПФ для $N = 2^v = 2^3 = 8$, $v = 3$.

Алгоритм БПФ представляет собой трехэтапную процедуру определения ДПФ по общей формуле (2.73) при i = 1, 2, 3. Начальные условия алгоритма формируются в результате трехкратного разбиения исходной 8-точечной последовательности на четные и нечетные отсчеты и представляются в двоично-инверсном порядке:

$$\{s(0), s(4), s(2); s(6), s(1), s(5), s(3), s(7)\}.$$

На нулевом этапе каждой паре отсчетов ДПФ – четному $f_{2m}^0(0)$ и нечетному $f_{2m+1}^0(0)$ при m = 0, 1, 2, 3 присваиваются значения четного и нечетного отсчетов прореженной 8-точечной последовательности.

На первом этапе, (i = 1) определяются четыре 2-точечных ДПФ при m = 0, 1, 2, 3 с учетом начальных условий:

$$\begin{cases} f_0^1(0) = s(0) + W_2^0 s(4); \\ f_0^1(1) = s(4) - W_2^0 s(4); \end{cases} \qquad \begin{cases} f_1^1(0) = s(2) + W_2^0 s(6); \\ f_1^1(1) = s(2) - W_2^0 s(6); \end{cases}$$

$$\begin{cases} f_2^1(0) = s(1) + W_2^0 s(5); \\ f_2^1(1) = s(1) - W_2^0 s(5); \end{cases} \qquad \begin{cases} f_3^1(0) = s(3) + W_2^0 s(7); \\ f_3^1(1) = s(3) - W_2^0 s(7). \end{cases}$$

Поворачивающий множитель в данном случае $W_2^0 = 1$.

На втором этапе (i = 2) определяются два 4-точечных ДПФ при m = 0,1:

$$\begin{cases} f_0^2(k) = f_0^1(k) + W_4^k f_1^1(k); \\ f_0^2(k+2) = f_0^1(k) - W_4^k f_1^1(k); \\ k = 0,1 \end{cases}$$

$$\begin{cases} f_1^2(k) = f_2^1(k) + W_4^k f_3^1(k); \\ f_1^2(k+2) = f_2^1(k) - W_4^k f_3^1(k); \\ k = 0,1. \end{cases}$$

На третьем этапе (i = 3) определяется 8-точечное ДПФ при m = 0:

$$\begin{cases} f_0^3(k) = f_0^2(k) + W_8^k f_1^2(k); \\ f_0^2(k+4) = f_0^2(k) - W_8^k f_1^2(k); \\ k = 0,1,2,3 \end{cases}$$

Полученные отсчеты 8-точечного ДПФ следуют в естественном порядке. Граф 8-точечного БПФ с прореживанием по времени показан на рис. 2.8.

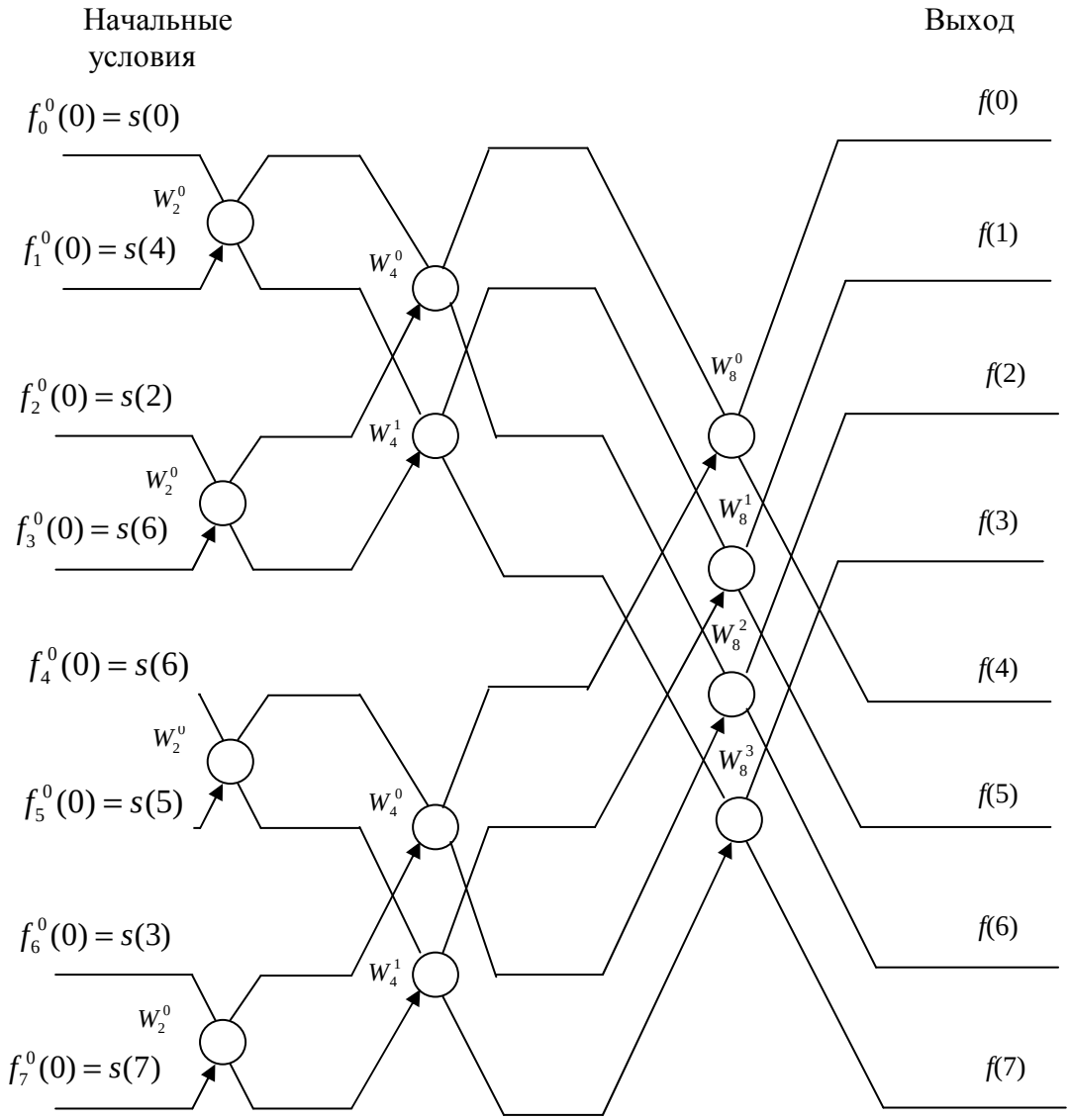


Рис. 2.8. Граф алгоритма БПФ с прореживанием по времени для 8-точечного ДПФ

2.3.2 Алгоритм БПФ с прореживанием по частоте

Основная идея алгоритма БПФ с прореживанием по частоте заключается в поэтапном вычислении N-точечного ДПФ на v этапах, на каждом из которых ДПФ определяется через ДПФ вдвое большей размерности [6].

Алгоритм БПФ с прореживанием по частоте в общем случае описывается следующим образом:

- задание начальных условий – естественный порядок следования входных отсчетов;
- на первом этапе определяются *N*/2-точечные ДПФ *N*/2-точечных последовательностей (двух половин исходной последовательности);
- на втором этапе определяются *N*/4-точечные ДПФ как комбинация *N*/2-точечных ДПФ;

……

- на *i*-м этапе определяются 2^{i-1}-точечные ДНФ как комбинация 2^i-точечных ДПФ;

……

- на *v*-м (последнем) этапе определяются 2-точечные ДПФ как комбинация 4-точечных ДПФ. Выходные отсчеты ДПФ следуют в битреверсивном порядке двоичных номеров.

Анализ алгоритмов БПФ с прореживанием по времени и частоте показывает, что общая формула расчета БПФ с прореживанием по частоте на произвольном *i*-м этапе может быть получена путем замены входа и выхода, а также обращения стрелки направленного графа – «бабочки» алгоритма БПФ с прореживанием по времени.

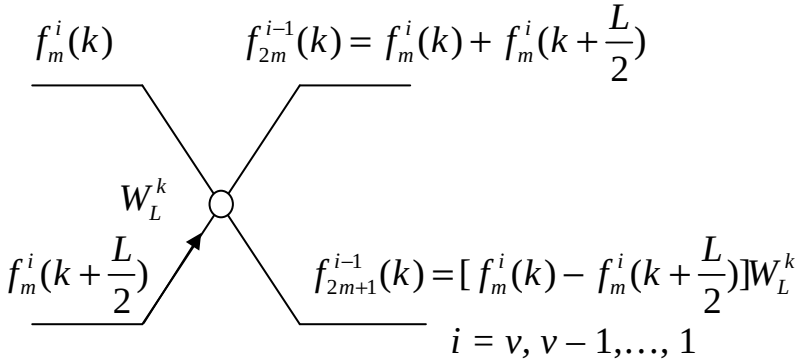


Рис. 2.9. Граф базовой операции алгоритма БПФ с прореживанием по времени

Общая формула расчета ДПФ для произвольного *i*-го этапа:

$$\begin{cases} f_{2m}^{i-1}(k) = f_m^i(k) + f_m^i(k+\frac{L}{2}); \\ f_{2m+1}^{i-1}(k) = [f_m^i(k) - f_m^i(k+\frac{L}{2})]W_L^k \\ i = v, v-1, \ldots, 1; \\ m = 0,1,\ldots,M-1; \\ k = 0,1,\ldots,\frac{L}{2}-1. \end{cases} \qquad (2.78)$$

где i – номер этапа, $i = v, v - 1,\ldots, 1$; m – номер ДПФ; k – номер отсчета ДПФ; M – количество ДПФ, определяемое по формуле

$$M = 2^{v-i};$$

L – размерное ДПФ, определяемая по формуле

$$L = 2^{i};$$

$f_m^i(k)$ – L-точечное ДПФ; $f_{2m}^{i-1}(k)$, $f_{2m+1}^{i-1}(k)$ – четное и нечетное $L/2$-точечное ДПФ соответственно.

Согласно (2.78) L-точечное ДПФ определяется параллельно:

- отсчеты *четного* $L/2$-точечного ДПФ – но *верхней* формуле;
- отсчеты *нечетного* $L/2$-точечного ДПФ – по *нижней* формуле.

Анализ выражения (2.78) описывает алгоритм БПФ с прореживанием по частоте – v-этапную процедуру типа «цикл в цикле», где:

- внешний цикл организуется по переменной i: $i = v... 1$;
- первый внутренний цикл (при фиксированном i) – по переменной m: $m = 0, 1, 2,..., M - 1$;
- второй внутренний цикл (при фиксированных i и m) – по переменной k: $k = 0, 1,\ldots, (L/2 - 1)$. В цикле вычисляется k-й отсчет $L/2$-точечного ДПФ по формуле (2.77).

На выходе алгоритма (в результате вычислений при $i = 1$ и $L = 2^1 = 2$) согласно (2.78) получаем N-точечное ДПФ, отсчеты которого следуют в бит-реверсивном порядке двоичных номеров.

На практике алгоритм БПФ с прореживанием по частоте применяют реже, чем с прореживанием по времени, т.к. последний обеспечивает естественный порядок следования отсчетов ДПФ на выходе.

Как и в алгоритме с прореживанием по времени, в данном алгоритме вычислительные затраты составляют $N\log_2 N$ операций сложения и $0{,}5N\log_2 N$ операций умножения, а вычисления можно выполнять с замещением. Величина «бабочки» по мере продвижения к концу вычислений уменьшается.

Если $N = 8$, алгоритм БПФ представляет собой трехэтапную процедуру вычисления ДПФ по общей формуле (2.78) при $i = 3, 2, 1$.

Начальные условия алгоритма задаются как

$$\begin{cases} f_0^3(k) = s(k); \\ k = 0,1,...,7. \end{cases}$$

Первый этап: $i = v = 3$.

Определяется одно 8-точечное ДПФ (2.78) при $m = 0$ с учетом начальных условий:

$$\begin{cases} f_0^{\,2}(k) = s(k) + s(k+4); \\ f_1^{\,2}(k) = [s(k) - s(k+4)]W_8^k; \\ k = 0,1,....,3. \end{cases}$$

На рис. 2.10 показан граф алгоритма БПФ с прореживанием по частоте для 8-точечного ДПФ.

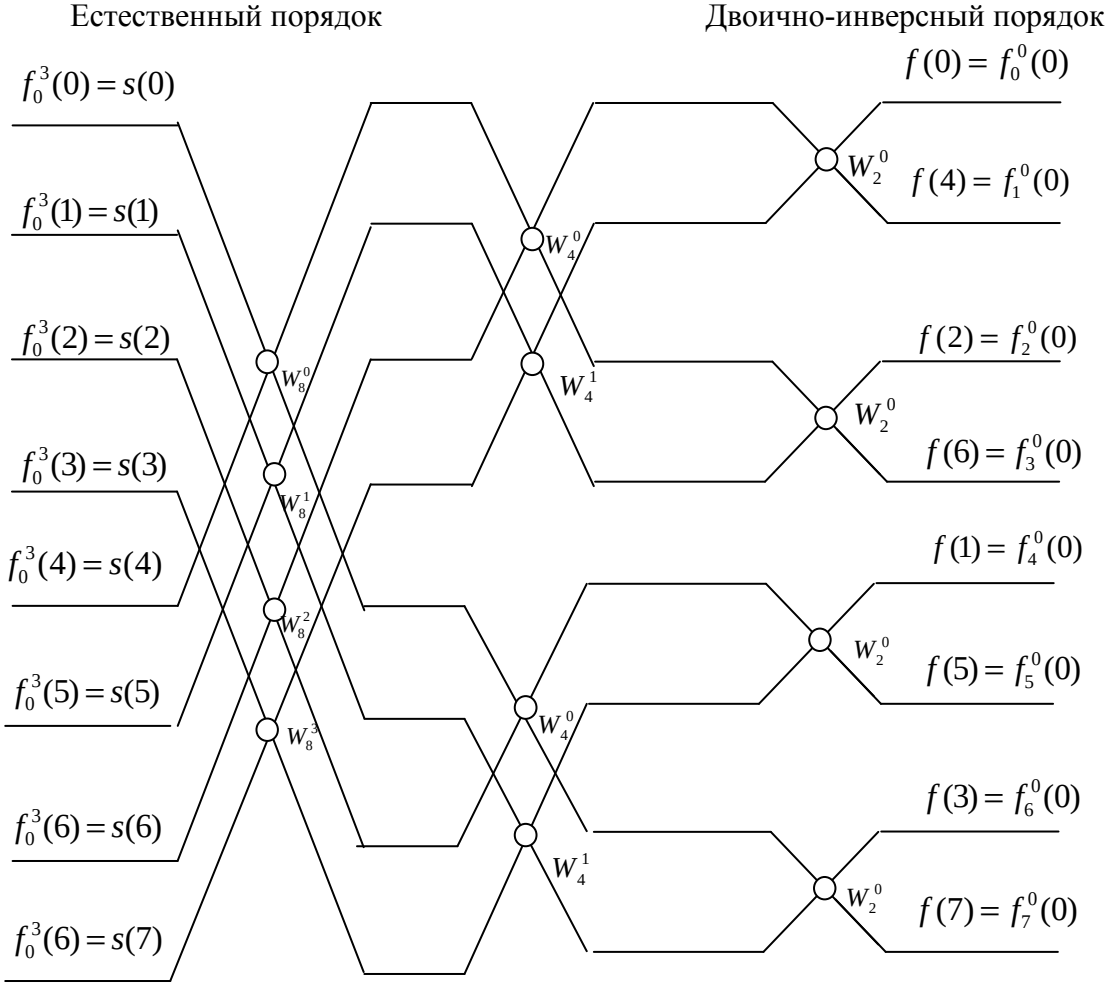


Рис. 2.10. Граф алгоритма БПФ с прореживанием по частоте для 8-точечного ДПФ

- *Второй этап:* $i = \nu - 1 = 2$.
  Определяются два 4-точечных ДПФ при $m = 0, 1$:

$$\begin{cases} f_{2m}^1(k) = f_m^2(k) + f_m^2(k+2); \\ f_{2m+1}^1(k) = [f_m^2(k) - f_m^2(k+2)]W_4^k; \\ k = 0,1. \end{cases}$$

- *Третий этап:* $i = v - 2 = 1$.

  Определяются 4 2-точечных ДПФ при $m = 0, 1, 2, 3$:

$$\begin{cases} f_{2m}^{0}(0) = f_{m}^{1}(0) + f_{m}^{1}(1); \\ f_{2m+1}^{0}(0) = [f_{m}^{1}(0) - f_{m}^{1}(1)]W_{2}^{0}. \end{cases}$$

Значения $f_{2m}^{0}(0)$ и $f_{2m+1}^{0}(0)$ при $m = 0, 1, 2, 3$ и есть искомое 8-точечное ДПФ, отсчеты которого следуют в бит-реверсивном порядке двоичных номеров:

$$\begin{cases} f(0) = f_{0}^{0}(0); \\ f(4) = f_{1}^{0}(0); \end{cases} \qquad \begin{cases} f(2) = f_{2}^{0}(0); \\ f(6) = f_{3}^{0}(0); \end{cases}$$

$$\begin{cases} f(1) = f_{4}^{0}(0); \\ f(5) = f_{5}^{0}(0); \end{cases} \qquad \begin{cases} f(3) = f_{6}^{0}(0); \\ f(7) = f_{7}^{0}(0). \end{cases}$$

2.3.3. Алгоритм действительного БПФ (АДБПФ)

Построим быстрый алгоритм для расчета ДПФ действительной N-точечной последовательности $\{s(n)\}$ [11]. С целью снижения вычислительной сложности алгоритма БПФ с прореживанием по времени N-точечной действительной последовательности воспользуемся свойством эрмитовой симметрии спектральных коэффициентов ДПФ и весовых множителей [11]:

$$\begin{aligned} f_k &= f_{N-k}^{*} \\ f_k &= f_{N/2-k}^{*} \\ w_N^k &= -\left(w_N^{N/2-k}\right)^{*}, \qquad k = 1,\ldots, N/4 - 1. \end{aligned} \tag{2.79}$$

Из соотношений (2.79) следует, что на этапе взвешивания достаточно выполнить только $N/4 - 1$ комплексных умножений $f_k = s_k W_N^k \times$ $\times\left(k = 1,.., N/4 - 1\right)$, поскольку $f_{N/2-k} = -f_k^{*}$.

Учитывая свойство (2.79), преобразуем традиционную для алгоритмов комплексного БПФ сумма-разностную операцию «бабочка» в «модифицированную бабочку», которая помимо суммирования содержит операцию разности комплексно-сопряженных чисел. На рис. 2.11 представлен граф, иллюстрирующий разложение N-точечного ДПФ действительной последовательности на два $N/2$-точечных ДПФ с помощью алгоритма действительного быстрого преобразования Фурье (АДБПФ). Исходная последовательность $\{s(n)\}$ $\left(n = 0,1,\ldots, N-1\right)$ разделена по четным и нечетным отсчетам, а последовательность коэффициентов ДПФ f_k $\left(k = 0,1,\ldots, N/2\right)$

расположена в прямом порядке. Каждый этап АДБПФ состоит из следующих операций (см. рис. 2.11):

1. Выполнение двух $N/2$-точеных ДПФ действительных последовательностей.

2. Выполнение $N/4 - 1$ комплексных взвешиваний.

3. Вычисление $N/2 + 1$ коэффициентов N-точечного ДПФ на основе $N/4 - 1$ попарных сложений и вычитаний комплексных чисел и одного сложения и вычитания пары действительных чисел при выполнении двух базовых операций.

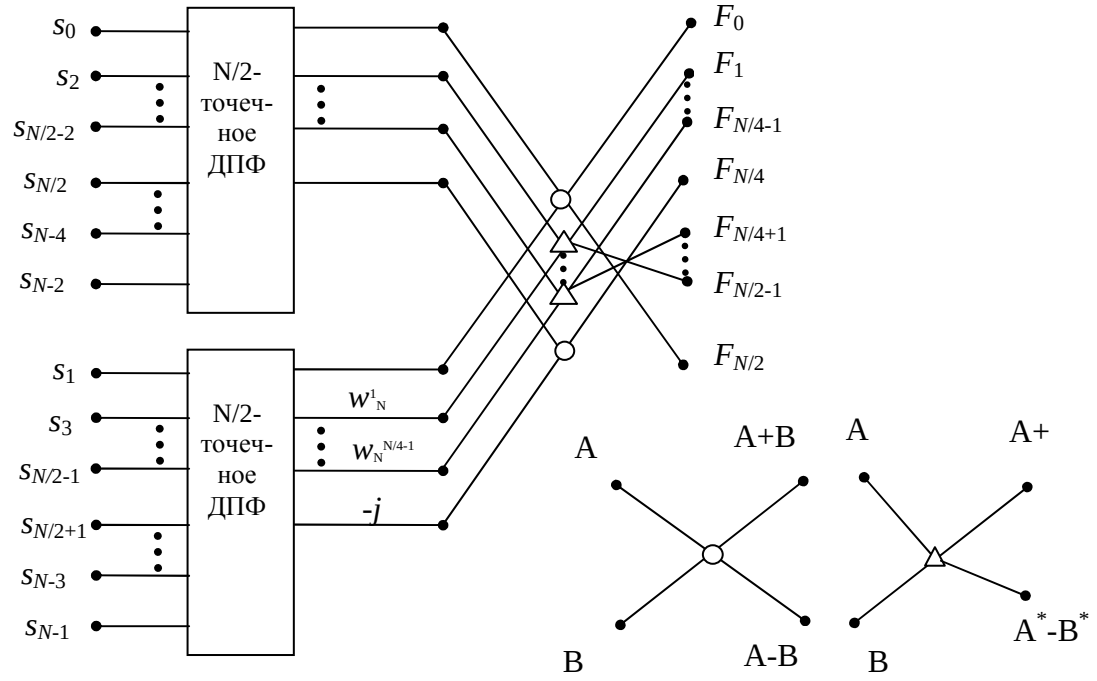


Рис. 2.11. Первый этап разложения в АДБПФ

Для АДБПФ характерно:

1. Наличие обратной двоичной перестановки входной последовательности $\{s(n)\}$ $\left(n = 0,1,...,N-1;\ N = 2^{\nu}\right)$.

2. Прямой порядок коэффициентов Фурье f_k $\left(k = 0,1,...,N/2\right)$ на выходе.

3. Рекурсивная структура, содержащая ν этапов.

4. Два вида базовых операций – «бабочка» и «модифицированная бабочка», состоящие из операций суммирования двух комплексных чисел и разности двух комплексно-сопряженных чисел.

5. Выполнение $(N/4 - 1)$ комплексных взвешиваний на W_N^k $\left(k = 1,..,N/4-1\right)$ для каждого этапа АДБПФ (рис. 2.12).

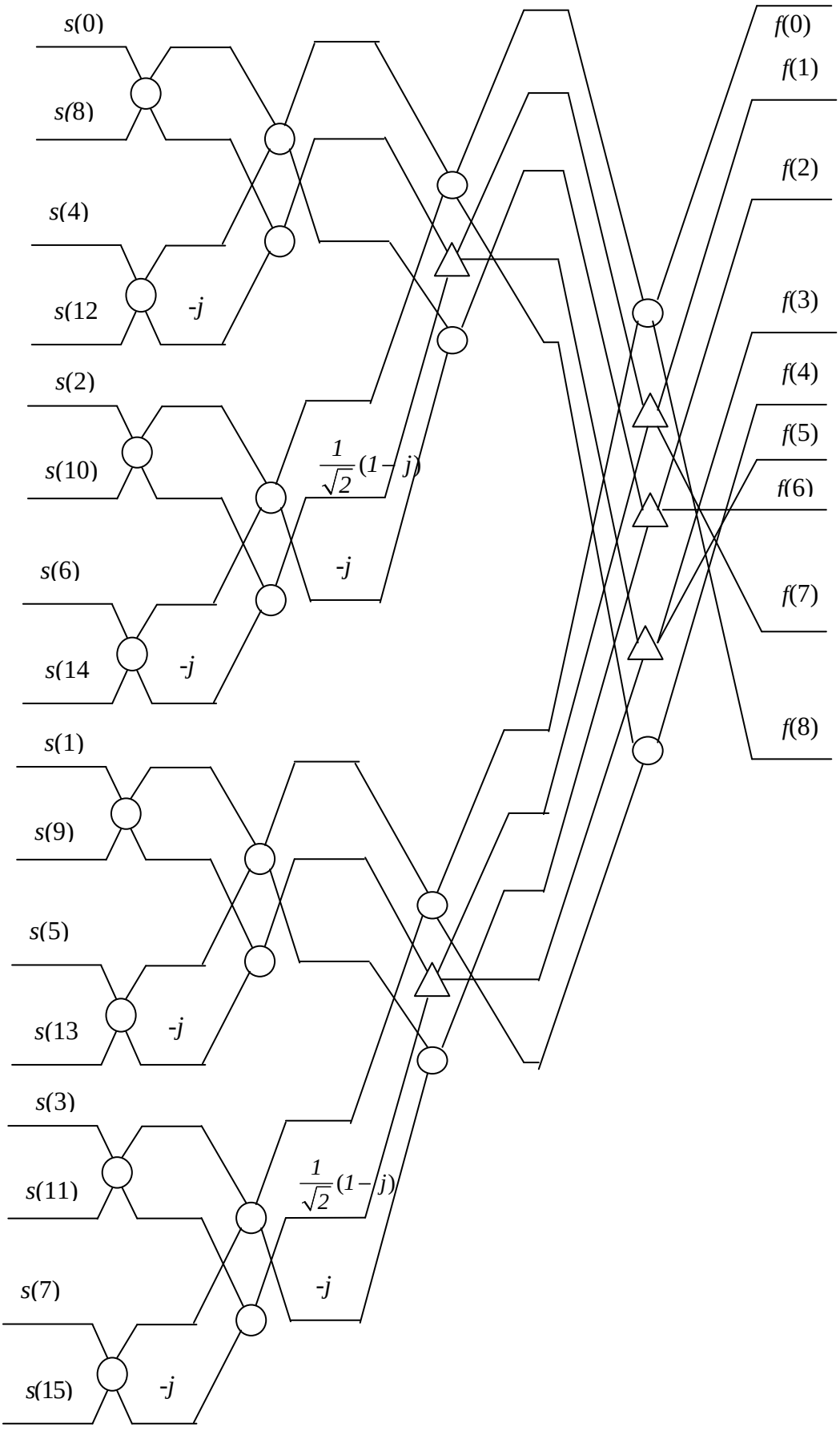


Рис. 2.12. Граф 32-точечного АДБПФ для N=16

2.4. Преобразование Уолша – Адамара

2.4.1. Функции Уолша

Функции Уолша образуют полную ортонормированную систему с количеством функций 2^l, l = 1, 2, 3 ..., и значениями ± 1 [4]. Число функций, включаемых в систему, обычно равно числу отсчетов каждой функции, так как при дискретном спектральном анализе сигналов с N отсчетами число спектральных составляющих также должно быть равно N. Поскольку нумерация (упорядочение) функций Уолша может быть произведена разными способами, то возможны различные системы функций Уолша. Наиболее широко используется представление таких систем в виде квадратных матриц, в которых каждая строка – это функция Уолша.

Адамар показал, что эти матрицы принадлежат классу матриц с максимально возможным определителем. Поэтому они получили название матриц Адамара.

Матрицей Адамара порядка N называется ортогональная матрица размером $N \times N$, элементами которой являются действительные числа «±1». Матрица Адамара порядка N обозначается H_N.

Простейшей матрицей Адамара является матрица второго порядка вида

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \qquad (2.80)$$

Если H_N – матрица Адамара порядка N, то матрица

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}. \qquad (2.81)$$

является матрицей Адамара порядка $2N$.

Например, матрицы четвертого и восьмого порядка имеют, соответственно, следующий вид:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix};$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hdashline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right].$$

В более компактном виде матрицу Адамара можно представить как

$$H_8 = H_2 \otimes H_4, \qquad 2.82)$$

где $\otimes$ – знак прямого или кронекерова умножения матриц, которое выполняется путем умножения каждого элемента матрицы H_4 на матрицу H_2.

Для функций, принадлежащих системе Уолша – Адамара, используется обозначение $had(k,n)$, где k – номер строки, а n – номер столбца матрицы Адамара. Пронумеруем строки и столбцы матрицы Адамара двоичными числами от 0 до $N - 1$. Тогда элемент матрицы, стоящий на пересечении строки с номером k и столбца с номером n, вычисляется по формуле

$$had\,(k, n) = (-1)^{(k \cdot n)}, \qquad (2.83)$$

где $(k \cdot n) = \sum_{i=0}^{l-1} k_i n_i$ – скалярное произведение векторов k и n; k – двоичный номер строки, n – двоичный номер столбца.

Например, для матрицы H_8 с учетом (2.83):

$$\begin{array}{ccccccccc} 0 & & 0 & & & & 1 & & \begin{pmatrix} n_2 \\ n_1 \\ n_0 \end{pmatrix} = n \\ 0 & .. \; .. & 1 & .. \; .. \; .. & & & .....1 & & \\ 0 & & 1 & & & & 1 & & \end{array}$$

$$k = \begin{pmatrix} k_2 & k_1 & k_0 \end{pmatrix}$$

$$\begin{array}{c} 0\;0\;0 \\ \vdots \\ 0\;1\;0 \\ \vdots \\ \vdots \\ \vdots \\ 1\;1\;1 \end{array} \left[\begin{array}{c|c} & \\ & \\ - - - \; (-1)^{kn} \; - - - - - - & \\ & \\ & \\ & \\ & \end{array}\right] = \begin{bmatrix} had(000,n) \\ had(001,n) \\ had(010,n) \\ had(011,n) \\ had(100,n) \\ had(101,n) \\ had(110,n) \\ had(111,n) \end{bmatrix}$$

Строки матриц Адамара *had*(*000*,*n*), *had*(001,*n*), …, *had*(*k*,*n*) можно рассматривать как функции дискретных переменных *n* и *k* , определенные в целочисленных точках 0, 1,2, ... *N* – 1 интервала [0, *N*). Эти функции называются дискретными функциями Уолша. Первая переменная *k* обозначает номер функции, вторая переменная *n* – дискретное время. В качестве номера функции берется номер соответствующей строки матрицы Адамара. При этом говорят, что функции упорядочены по Адамару.

Важнейшими свойствами функций Уолша являются:

1. *Ортогональность*

$$\sum_{n=0}^{N-1} had(k_1,n)had(k_2,n) = \begin{cases} N, k_1 = k_2; \\ 0, k_1 \neq k_2, \end{cases}$$

или в матричной записи

$$H_N H_N^T = NI,$$

где *I* – единичная матрица.

2. *Симметричность*

$$H_N = H_N^T.$$

Используя свойства ортогональности и симметричности, легко получить $H^{-1} = N^{-1}H$.

3. Мультипликативность по номеру функции

$$had(k_1,n)had(k_2,n) = had(k_1 \oplus k_2, n)$$

и номеру отсчета

$$had(k,n_1)had(k,n_2) = had(k,n_1 \oplus n_2).$$

Постоянная функция и функции меандрового типа с одной ненулевой позицией в двоичной записи номера называются функциями Радемахера. Для $N = 8$ они равны:

$$R_0 = had(000,n); \quad R_2 = had(010,n);$$
$$R_1 = had(100,n); \quad R_3 = had(001,n).$$

Из свойства мультипликативности следует, что любая функция Уолша может быть получена перемножением функций Радемахера. Например:

$$had(011,n) = R_2 R_3;$$
$$had(110,n) = R_1 R_2.$$

В практических приложениях используется и другая нумерация функций, что соответствует другому способу ранжирования обобщенных гармоник. Наиболее употребительными из них являются упорядочения по Пэли и по Уолшу.

В некоторых практических приложениях, например, в аналого-цифровых преобразованиях, желательно использовать коды, у которых все следующие друг за другом кодовые слова различаются только одной цифрой в некотором разряде. Коды, обладающие таким свойством, называются циклическими. Очень важным циклическим кодом является код Грея (табл. 2.3). Двоичное представление числа может быть легко преобразовано в код Грея с помощью полусумматоров.

Пусть последовательность $g_{n-1}, g_{n-2}, ..., g_0$ является кодовым словом в n-разрядном двоичном коде Грея и соответствует двоичному числу $b_{n-1}, b_{n-2}, ..., b_0$. Тогда g_i может быть определена как

$$g_i = b_i \oplus b_{i+1},\ g_{n-1} = b_{n-1},\ 0 \le i \le n-2.$$

Таблица 2.3

Трехразрядный код Грея

Десятичное число	Двоичный код			Код Грея		
	b_2	b_1	b_0	g_2	g_1	g_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

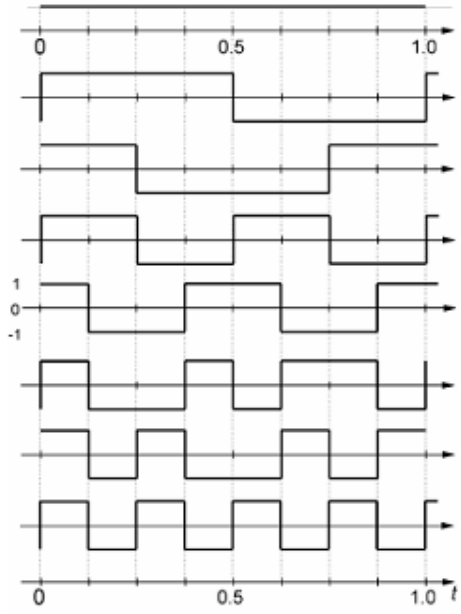


Рис. 2.13. Функции Уолша

Система Пэли образуется из системы Адамара двоичной инверсией номеров функций, т.е. путем записи разрядов двоичного представления номера k в обратном порядке. Система Уолша располагает функции в порядке возрастания числа смен знака на интервале (по «частотам» следования, рис 2.13), что соответствует расположению их номеров в двоично-инверсном коде Грея.

Для $N = 4$ в рассмотренных способах упорядочения строки расположены следующим образом:

по Адамару	по Пэли	по Уолшу
00	00	00
01	10	10
10	01	11
11	11	01

Соответствующие матрицы при упорядочении по Пэли и по Уолшу:

$$P=\{pal(k,n)\}=\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&1&-1\\1&-1&-1&1\end{bmatrix};\quad W=\{wal(k,n)\}=\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}.$$

2.4.2. Преобразование Уолша – Адамара и его свойства

Пусть $\{s(n)\}=\{s(0), s(1), \ldots, s(N-1)\}$ – совокупность равноотстоящих отсчетов сигнала. Пара дискретного преобразования Уолша – Адамара в показательной форме представляется в виде [4]:

$$b(k)=\sum_{n=0}^{N-1}s(n)(-1)^{(kn)},\quad k=0,1,\ldots,N-1; \tag{2.84}$$

$$s(n)=N^{-1}\sum_{k=0}^{N-1}b(k)(-1)^{(kn)},\quad n=0,1,\ldots,N-1. \tag{2.85}$$

Равенство (2.84) называется *прямым преобразованием* и дает спектр сигнала в базисе Уолша. Равенство (2.85) называют *обратным преобразованием*.

Используя матрицу Адамара порядка N, можно записать преобразование в матричной форме:

$$B=HS;$$
$$s=N^{-1}HB, \tag{2.86}$$

где $s = [s(0), s(1), \ldots, s(N-1)]^T$, $B = [b(0), b(1), \ldots, b(N-1)]^T$ – векторы-столбцы отсчетов сигнала и спектральных коэффициентов соответственно.

Основными свойствами преобразования являются:

1. *Линейность*. Если $\{x(n)\}$ и $\{y(n)\}$ – две последовательности со спектрами $\{b_x(k)\}$ и $\{b_y(k)\}$ соответственно, то спектр их взвешенной суммы $\{z(n)\} = \{A_x(n)\}+\{B_y(n)\}$ равен

$$\{b_z(k)\}=\{Ab_x(k)\}+\{Bb_y(k)\}. \tag{2.87}$$

2. *Инвариантность к диадному сдвигу*. Рассмотрим функцию $s(n)$ дискретной переменной n. Функция $S(n\oplus\tau)$ называется диадным сдвигом

функции $s(n)$. Сущность диадного сдвига заключается в перестановке отсчетов исходной функции. В частности, на место отсчета с номером n ставится отсчет с номером $(n \oplus \tau)$. Например, пусть $\{s(n)\} = \{0011)$ и $\tau = 2$.

Значения $(n \oplus \tau)$ для $n = 0, 1, 2, 3$ равны

$$\begin{array}{cccc} n & & n \oplus \tau \\ 00 & \oplus\ 10 & = & 10 \\ 01 & \oplus\ 10 & = & 11 \\ 10 & \oplus\ 10 & = & 00 \\ 11 & \oplus\ 10 & = & 01 \end{array}$$

Из этой записи следует, что отсчет с номером ноль ставится на второй отсчет, а на место отсчета с номером один – третий и т.д. В результате получим

$$\{s(n)\} = \{0011\};$$
$$\{s(n \oplus \tau)\} = \{1100\}.$$

Из свойств мультипликативности следует, что при фиксированном τ

$$had(k, n \oplus \tau) = had\,(k, n)\, had(k, \tau) = \pm had(k, n),$$

т.е. спектральные составляющие исходного и диадно-сдвинутого сигналов могут отличаться только знаком. Амплитудный же спектр при диадном сдвиге не меняется.

3. *Теорема о свертке и корреляции.* Понятие диадного сдвига позволяет обобщить понятия свертки и корреляционной функции. Так как суммирование и вычитание по модулю два совпадают, то диадная свертка совпадает с диадной корреляцией и определяется выражением

$$y(n) = r(n) = \sum_{l=0}^{N-1} h(l)s(n \oplus l), \qquad (2.88)$$

где $n = 0, 1, \ldots, (N - 1)$.

Теорема о свертке утверждает, что спектр свертки равен произведению спектров сворачиваемых последовательностей:

$$b_y(k) = b_h(k)b_s(k). \qquad (2.89)$$

Это позволяет для вычисления диадной свертки и корреляционной функции использовать преобразование Адамара:

$$R = H^{-1}(B_h B_s) = N^{-1}H(B_h B_s).$$

2.4.3. Быстрое преобразование Уолша – Адамара

Вычисление преобразования прямым способом требует выполнения $N(N - 1)$ операций сложения. Существуют быстрые алгоритмы, которые требуют только $N\log_2 N$ операций.

Для построения быстрого алгоритма рассмотрим матричное равенство $B = HS$.

Запишем в виде:

$$\begin{bmatrix} b(0) \\ b(1) \\ . \\ . \\ . \\ b(\frac{N}{2}-1) \\ b(N/2) \\ . \\ . \\ . \\ b(N-1) \end{bmatrix} = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ . \\ . \\ . \\ s(\frac{N}{2}-1) \\ s(N/2) \\ . \\ . \\ . \\ s(N-1) \end{bmatrix},$$

что позволит отдельно вычислить первую и вторую половины спектра.

Получим

$$\begin{bmatrix} b(0) \\ b(1) \\ . \\ . \\ . \\ b(\frac{N}{2}-1) \end{bmatrix} = H_{N/2} \begin{bmatrix} s(0)+s(N/2) \\ s(1)+s(N/2+1) \\ . \\ . \\ . \\ s(\frac{N}{2}-1)+s(N-1) \end{bmatrix} \qquad \begin{bmatrix} b(N/2) \\ b(\frac{N}{2}-1) \\ . \\ . \\ . \\ b(N-1) \end{bmatrix} = H_{N/2} \begin{bmatrix} s(0)-s(N/2) \\ s(1)-s(N/2) \\ . \\ . \\ . \\ s(\frac{N}{2}-1)-s(N-1) \end{bmatrix}$$

Из этих выражений следует, что вычисление N-точечного преобразования сводится к предварительному суммированию (вычитанию) входных данных и последующему вычислению двух $N/2$-точечных преобразований (рис. 2.14).

Процедуру снижения размерности преобразованиями можно продолжить до получения двухточечного преобразования. Для этого потребуется $\log_2 N$ шагов. На каждом шаге производится N сложений, поэтому общее количество операций сложения равно $N\log_2 N$. Рассмотренный алгоритм называется быстрым преобразованием Уолша – Адамара (БПУА).

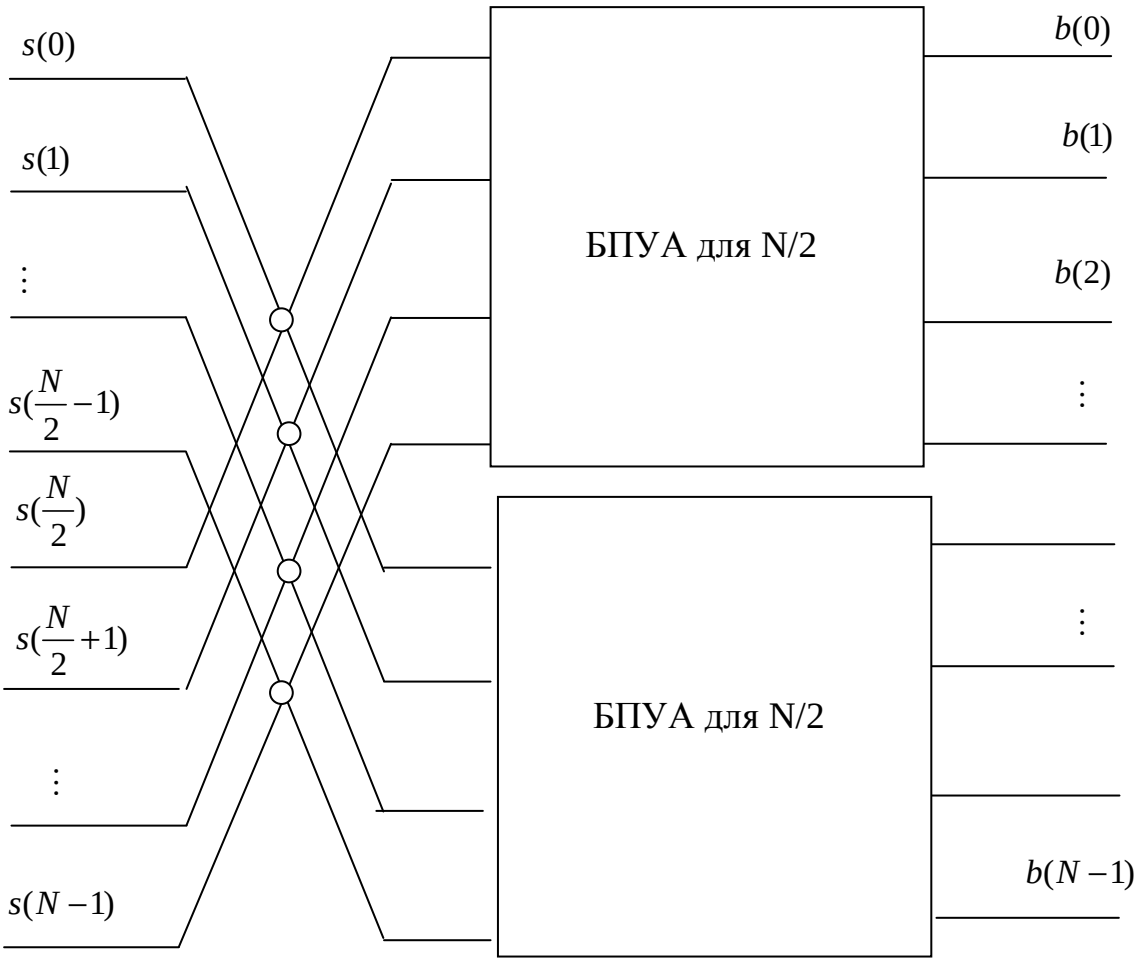


Рис. 2.14. Принцип построения алгоритма БПУА

На рис. 2.15 приведен граф вычислительного процесса для N = 8. БПУА можно трактовать как разложение (факторизацию) матрицы Адамара в произведение слабозаполненых сомножителей.

Возможность представления матриц Адамара в виде произведения слабозаполненных матриц показал Гуд. Факторизация методом Гуда для матриц, представляющих собой кронекеровское произведение нескольких одинаковых матриц, основана на следующих теоремах:

1. Если $E_m^{[n]}$ – произвольная матрица размером $m \times m$, то

$$E_m^{[n]} = \underbrace{E_m \otimes E_m \otimes \ldots \otimes E_m}_{n \text{ раз}} = B_{m^n}^{n}. \qquad (2.90)$$

Другими словами, n-я кронекеровская степень матрицы E_m размером $m \times m$ равна обычной n-й степени матрицы $B_m{}^n$ размером $m^n \times m^n$.

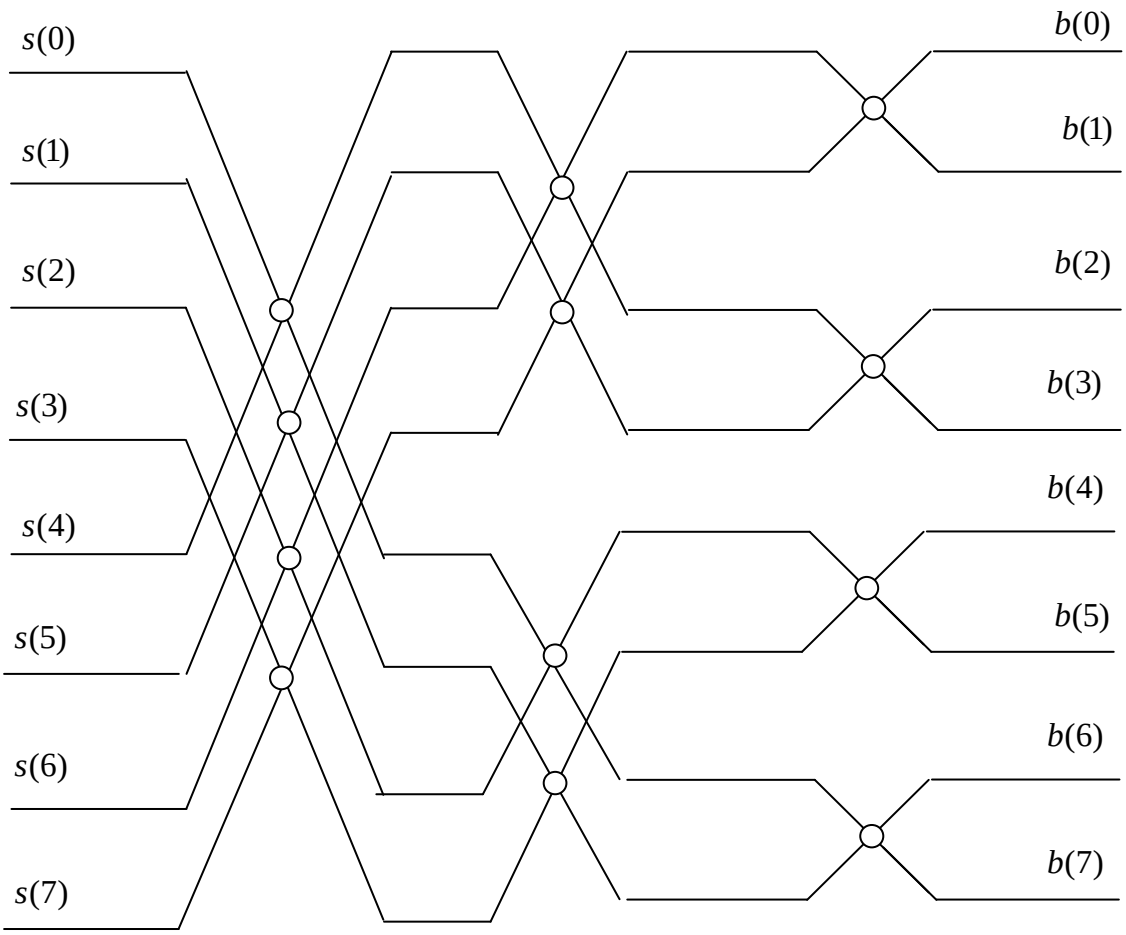


Рис. 2.15. Граф алгоритма БПУА для $N = 8$

2 Если $E_m^{[n]}$ – произвольная матрица размером $m \times m$, то

$$E_m^{[n]} = \underbrace{E_m \otimes E_m \otimes ... \otimes E_m}_{n \text{ раз}} = C_{m^n}^{(1)} \cdot C_{m^n}^{(2)} .... C_{m^n}^{(n)}, \qquad (2.91)$$

где

$$\begin{aligned} C_{m^n}^{(1)} &= E_m \otimes 1_m \otimes ... \otimes 1_m; \\ C_{m^n}^{(2)} &= 1_m \otimes E_m \otimes ... \otimes 1_m; \\ C_{m^n}^{(n)} &= 1_m \otimes 1_m \otimes ... \otimes E_m. \end{aligned} \qquad (2.92)$$

Здесь 1_m – единичная матрица размером $m \times m$, а матрицы $C_{m^n}^{(1)}, C_{m^n}^{(2)}, ..., C_{m^n}^{(n)}$ – все размером $m^n \times m^n$.

Особенностью матриц ${B_m}^n$ и матриц $C_{m^n}^{(1)}, C_{m^n}^{(2)}, ..., C_{m^n}^{(n)}$ является то, что они имеют не больше m^{n+1} отличных от нуля элементов, поэтому общее число операций типа сложение – вычитание не превышает величины $n \times m^n$.

Для матриц Адамара

$$H_{2^n} = E_2^{[n]},$$

где

$$E_2^{[n]} = E_2^{[1]} \otimes E_2^{[2]} \otimes ... \otimes E_2^{[i]} \otimes ... \otimes E_2^{[n]},$$

$$E_2^{[l]} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad l = 1, 2, \ldots, n.$$

Согласно теореме 1 матрица Адамара размером 8×8 раскладывается на произведение трех слабозаполненных сомножителей:

$$H_8 = H_2 \otimes H_2 \otimes H_2 = B^3, \tag{2.93}$$

где

$$B = \begin{bmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & 1 \\ 1 & -1 & & & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & -1 & & \\ & & & & & & 1 & -1 \end{bmatrix}.$$

Согласно теореме 2 матрица Адамара может быть факторизована следующим образом:

$$H_8 = C_8^{(1)} \cdot C_8^{(2)} \cdot C_8^{(3)}, \tag{2.94}$$

где

$$C_8^{(1)} = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & 1 & & \\ & & & & 1 & -1 & & \\ & & & & & & 1 & 1 \\ & & & & & & 1 & -1 \end{bmatrix};$$

$$C_8^{(2)} = \begin{bmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ 1 & & -1 & & & & & \\ & 1 & & -1 & & & & \\ & & & & 1 & & 1 & \\ & & & & & 1 & & 1 \\ & & & & 1 & & -1 & \\ & & & & & 1 & & -1 \end{bmatrix};$$

$$C_8^{(3)} = \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 1 \\ 1 & & & & -1 & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & -1 & \\ & & & 1 & & & & -1 \end{bmatrix}.$$

Вычисление векторно-матричного произведения при использовании рассмотренных методов факторизации матриц Адамара требует выполнения 24 операций типа сложение – вычитание. Прямой метод умножения вектора на матрицу H_8 предполагает 56 вычислений.

2.5. Преобразование Хаара

Функции Хаара $\{ har(r,k,t) \}$ были построены в 1910 г. Они образуют полную систему ортонормированных функций, определенных на интервале [0, N), $N = 2^l$, $l = 1, 2...$ и описываются следующими рекуррентными соотношениями [4]:

$$har(0,0,t) = 1, \quad t \in x[0,1);$$

$$har(r,k,t) = \begin{cases} 2^{r/2}, & (k-1)2^{-r} \le t < (k-0{,}5)2^{-r}; \\ -2^{r/2}, & (k-0{,}5)2^{-r} \le t < k2^{-r}; \\ 0, & \text{при остальных } t \in x[0,1); \end{cases} \qquad (2.95)$$

$0 \le r < \log_2 N \quad 1 \le k \le 2^r$.

Для $N = 4$ (рис. 2.16) $r \in 0,1$, $k \in 1,2$

$$har(0,0,t) = 1, \quad t \in x[0,1); \quad har(0,1,t) = \begin{cases} 1, & 0 \le t < 0,5; \\ -1, & 0,5 \le t < 1. \end{cases}$$

$$har(1,1,t) = \begin{cases} 2^{1/2}, & 0 \le t < 0,25; \\ -2^{1/2}, & 0,25 \le t < 0,5; \\ 0, & \text{для других } t; \end{cases} \quad har(1,2,t) = \begin{cases} 2^{1/2}, & 0,5 \le t < 0,75; \\ -2^{1/2}, & 0,75 \le t < 1; \\ 0, & \text{для других } t; \end{cases} \qquad (2.96)$$

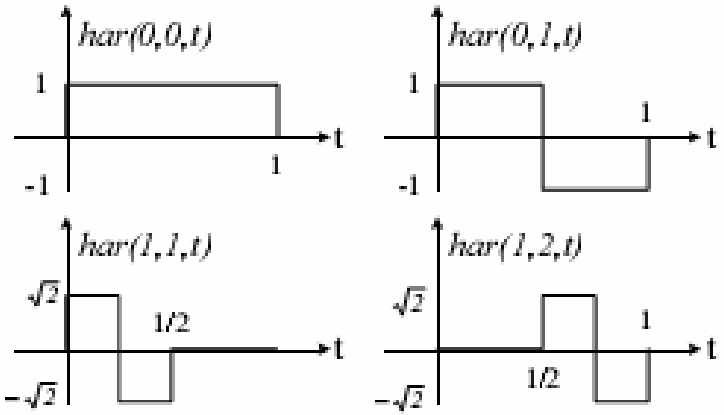


Рис. 2.16. Функции Хаара для $N = 4$

При дискретизации системы функций Хаара, представленной выше, получаются дискретные функции Хаара $har(r,k,t)$, определенные в целочисленных точках 0, 1, 2,.... $N - 1$.

Для $N = 8$ матрица функций Хаара имеет вид:

$$Hr_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}.$$

Для $N = 8$ матрица функций Хаара

$$Hr_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}.$$

Так как функции Хаара ортогональны, то $Hr_N Hr_N^T = NI$. Множество функций Хаара, в отличие от функций Уолша, не является мультипликативным, т.е. произведение функций Хаара не приводит вновь к функции Хаара. С увеличением переменной r уменьшается интервал, на котором функция Хаара отлична от нуля. Это обстоятельство позволяет использовать разложение Хаара для получения спектральных коэффициентов, учитывающих локальное поведение функции (рис. 2.17).

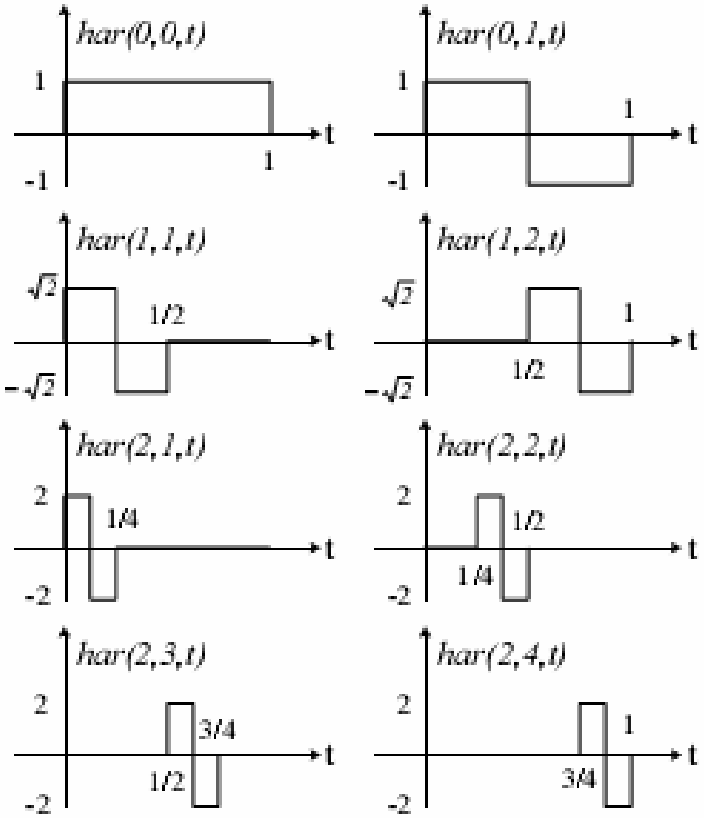


Рис. 2.17. Функции Хаара для $N = 8$

Пара дискретного преобразования Хаара сигнала S определяется следующими соотношениями:

$$Y = HrS;$$
$$S = N^{-1}Hr^{T}Y. \qquad (2.97)$$

Первое выражение называется прямым преобразованием Хаара, а второе выражение – обратным преобразованием Хаара.

Например, для сигнала вида $S = (2,4,1,4)$ спектр сигнала в базисе Хаара определяется:

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \\ 1 \\ = \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ -2\sqrt{2} \\ -3\sqrt{2} \end{bmatrix}.$$

В результате выполнения обратного преобразования Хаара получаем исходный сигнал:

$$S = \frac{1}{4}\begin{bmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 11 \\ 1 \\ -2\sqrt{2} \\ -3\sqrt{2} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 8 \\ 16 \\ 4 \\ 16 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 1 \\ 4 \end{bmatrix}.$$

Для выполнения преобразования Хаара существует ряд быстрых алгоритмов. На рисунках, представленных ниже, показаны графы быстрого преобразования Хаара – для $N = 4$ (рис. 2.18) и $N = 8$ (рис. 2.19).

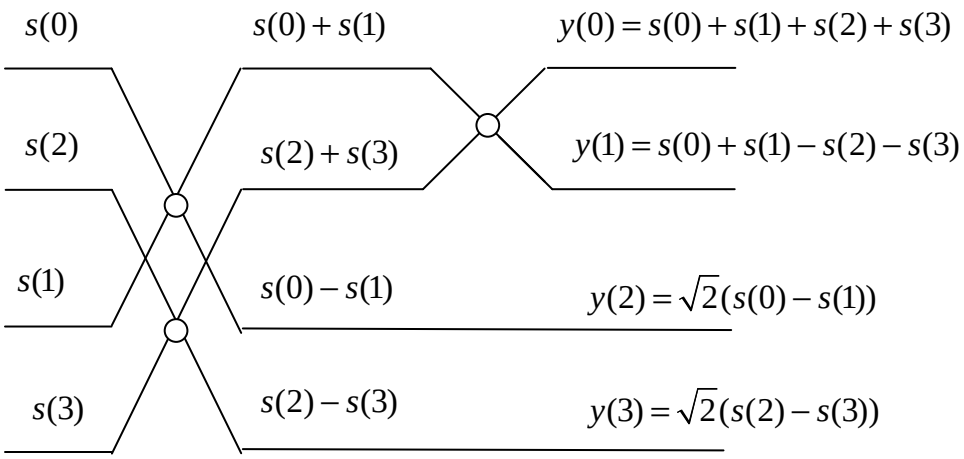


Рис. 2.18. Граф быстрого преобразования Хаара для $N = 4$

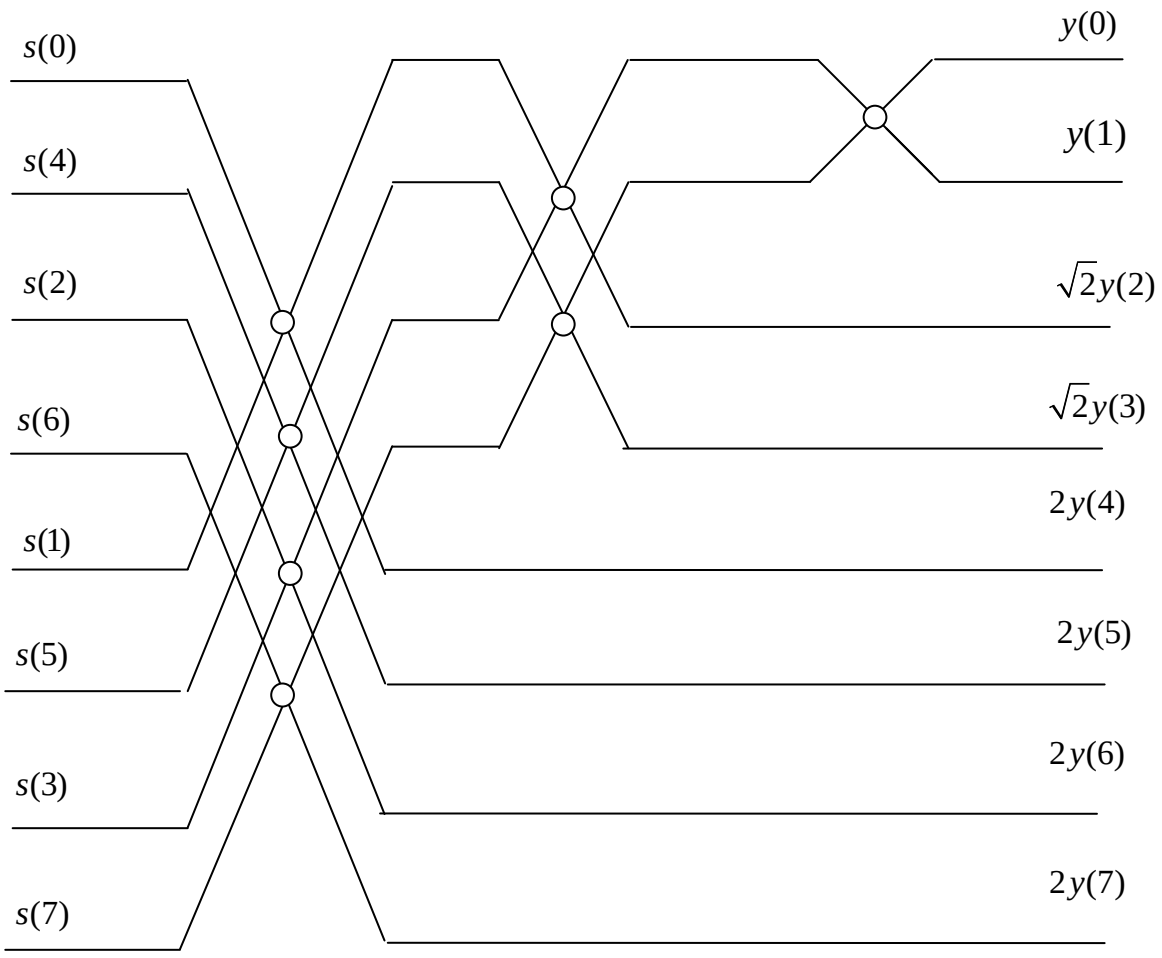


Рис. 2.19. Граф быстрого преобразования Хаара для $N = 8$

2.6. Кольцо и поле

Кольцом R называется множество элементов $(a,b,c,...)$, над которыми определены две операции – сложение и умножение (может быть, не совсем обычные) и выполняются следующие аксиомы [4]:

1) если $a,b \in R$, то $a+b \in R$ и $a \cdot b \in R$. Эта аксиома называется аксиомой замкнутости;

2) в кольце существует нейтральный элемент 0, такой, что для любого элемента а кольца $a+0=a$, и обратный элемент $(-a)$, такой, что $a+(-a)=0$;

3) для элементов кольца выполняются свойства ассоциативности

$$a(bc)=(ab)c$$

и дистрибутивности

$$a(b+c)=ab+ac\,.$$

Если в R существует нейтральный элемент по умножению –1, такой, что для любого элемента a кольца справедливо равенство $a \cdot 1 = 1 \cdot a = a$, то такое кольцо называется унитарным или кольцом с единицей.

Если в унитарном кольце для каждого элемента a существует мультипликативно обратный элемент a^{-1}, т.е. справедливо равенство $aa^{-1} = 1$, то такое кольцо называется полем.

В задачах цифровой обработки сигналов (в частности, в теоретико-числовых преобразованиях) большое значение имеет кольцо целых чисел по модулю числа М. Элементами кольца являются вычеты $0,1,2,...,M-1$, а операциями – операции сложения и умножения по модулю M.

Например, если M = 14, то элементами кольца будут числа 0, 1, 2,…, 13. Эти числа не образуют поле, так как не для всякого элемента кольца существует обратный элемент. В частности, число 2 не имеет обратного числа. Числа же 3 и 5 являются взаимно обратными, так как $3 \cdot 5 = 15 = 1 \bmod 14$.

Можно доказать, что число a имеет обратное число a^1, если оно взаимно просто с модулем. Поэтому если модуль равен простому числу, то все числа кольца имеют обратные и такое кольцо будет полем. Наличие обратного элемента важно для последующего построения теоретико-числового преобразования.

Рассмотрим некоторый элемент кольца a и его степени a^2, a^3, ... Поскольку число элементов в кольце конечно и справедлива аксиома замкнутости, то на каком-то шаге получим $a^{k+1} = a$, что дает $a^k = 1$. Наименьшее k, удовлетворяющее этому равенству, называется порядком элемента a. Для предыдущего примера легко проверить, что $3^6 = 1 \bmod 14$, т.е. число 3 имеет порядок 6.

Если a и M взаимно просты, то

$$a^{\varphi(M)} \equiv 1 \bmod M, \qquad (2.98)$$

где $\varphi(M)$ – функция Эйлера, равная числу чисел ряда $0,1,...,(M-1)$, взаимно простых с M. Следовательно, порядок числа a должен быть делителем функции Эйлера и все порядки лежат среди всевозможных делителей. Равенство (2.97) называется теоремой Эйлера. С ее помощью можно доказать следующую теорему.

Теорема 1. Если $M = p_1^{i_1} \cdot p_2^{i_2} ... p^{il}$, то порядок k числа a является делителем наибольшего общего делителя (НОД) чисел $p_1 - 1,\ p_2 - 1,...,\ p_l - 1,$ т.е.

$$k \mid \text{НОД}(p_1 - 1,\ p_2 - 1,...,\ p_l - 1). \qquad (2.99)$$

Читается: k делит НОД.

Теорему Эйлера можно использовать для нахождения обратного элемента. Умножив обе части сравнения (2.97) на a^{-1},получим:

$$a^{-1} \equiv a^{\varphi(M)-1} \bmod M , \qquad (2.100)$$

что дает

$$a^{-1} =< a^{\varphi(M)-1} > M .$$

2.7. Теоретико-числовое преобразование

2.7.1. Прямое и обратное теоретико-числовые преобразования, условия их существования

Рассмотрим кольцо целых чисел по модулю числа $M = p_1^{i_1} \cdot p_2^{i_2} \cdot \ldots \cdot p_l^{i_l}$, где p – простые числа [4]. Пусть a – элемент кольца порядка N (т.е. $a^N \equiv 1 \bmod M$). Из теоремы 1 следует, что N|НОД ($p_1 - 1$, $p_2 - 1$,..., $p_l - 1$).

Пусть $\{s(n)\} = \{s(0), s(1), \ldots, s(N-1)\}$ – последовательность отсчетов входного сигнала. Пара теоретико-числового преобразования последовательности $\{s(n)\}$ определяется в экспоненциальной форме следующими равенствами:

$$a(k) = \sum_{n=0}^{N-1} s(n)\alpha^{kn}, k = 0,1,\ldots, N-1; \qquad (2.101)$$

$$s(n) = N' \cdot \sum_{k=0}^{N-1} a(k)\alpha^{-kn}, n = 0,1,\ldots, N-1, \qquad (2.102)$$

где N' – число, обратное числу N, т.е. $NN' \equiv 1 \bmod M$; α – такое число, что N является наименьшим положительным числом, для которого справедливо $\alpha^N \equiv 1 (\bmod M)$.

Соответствующие матричные выражения имеют вид:

$$A = TS; \qquad (2.103)$$

$$S = NT^{-1}A, \qquad (2.104)$$

где $S = [s(0), s(1),\ldots, s(N-1)]^T$, $A = [a(0), a(1),\ldots, a(N-1)]^T$, а

$$T = k\begin{bmatrix} & & n & & \\ & & | & & \\ - & - & a^{kn} & - & - \\ & & | & & \\ & & | & & \end{bmatrix}; \qquad T^{-1} = k\begin{bmatrix} & & n & & \\ & & | & & \\ - & - & a^{-kn} & - & - \\ & & | & & \\ & & | & & \end{bmatrix}$$

есть матрицы прямого и обратного преобразований соответственно.

Сравнивая формулы (2.101) – (2.104) с соответствующими формулами для ДПФ, легко заметить полную аналогию в записи этих двух преобразований. Отличие заключается в том, что поворачивающие множители W заменены элементом a порядка N. Заметим также, что ТЧП длины N существует, если только в кольце имеется элемент, обратный элементу N. Поскольку в поле наличие обратных элементов гарантируется простотой модуля, то в поле ТЧП существует всегда.

Так как ТЧП по своей структуре подобно ДПФ, то для него справедливы все свойства ДПФ с заменой W на a и выполнением арифметических операций по модулю M. В частности, строки матриц T и T^{-1} ортогональны по модулю M и справедлива теорема о свертке.

Основным применением ТЧП является вычисление сверток и корреляционных функций методом двойного преобразования. Поскольку a – целое число, а не комплексное, то операция умножения существенно упрощается и выполняется точно. Шумы округления полностью отсутствуют, что может быть важно, когда входное отношение сигнал/шум мало.

Чтобы ТЧП давало еще дополнительные преимущества, необходимо заботиться о хорошем выборе параметров a и M. В частности, умножение на a и деление на M должны быть простыми операциями. Лучшим выбором для a является $a = 2^i$. В этом случае умножение на a сводится к сдвигу числа на i позиций. Деление проще всего осуществляется, если двоичное представление модуля содержит только два ненулевых символа. В этом плане заслуживают внимания числа вида $2^{2^i}+1$ – числа Ферма. Для них $N \le 2^{i+1}$ и a является степенью двойки, поэтому матрица преобразования разбивается на максимальное количество сомножителей, структура которых аналогична соответствующим матрицам ДПФ. Однако существенным недостатком чисел Ферма является большая длина машинного слова. Например, если $M = 2^{64} + 1$, то $N = 128$, т.е. приходится работать с 65-разрядными двоичными числами.

При выборе меньшего модуля не всегда удается выполнить условие $a = 2^i$, кроме того, деление на M происходит сложнее.

Как уже указывалось, при выполнении ТЧП вычисления производятся по модулю M, поэтому величина M должна быть больше ожидаемого ответа. Иначе будет получен искаженный результат.

Приведенные рассуждения показывают, что параметры N, a и M жестко связаны между собой теоретико-числовой конструкцией кольца или поля и не могут варьироваться произвольно. Это создает определенные трудности при практическом использовании ТЧП.

Для иллюстрации свойств и возможностей ТЧП рассмотрим пример.

Пример 2.12. Вычислить свертку последовательностей S = [2, –2, 1,0], $H = [1,2,0,0]^{\mathrm{T}}$.

Поскольку N = 4, то можно взять $M = 2^{2^2} + 1 = 17$. Целое число 2 имеет порядок 8, в чем можно убедиться непосредственно проверкой. Поэтому целое число 4 имеет порядок 4 и можно положить a = 4. Далее получим: $a^2 = 16$, $a^3 = 13$, $a^4 = 1$.

Матрица прямого преобразования

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 4^2 & 4^3 \\ 1 & 4^2 & 4^4 & 4^6 \\ 1 & 4^3 & 4^6 & 4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 13 \\ 1 & 16 & 1 & 16 \\ 1 & 13 & 16 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & -1 & -4 \\ 1 & -1 & 1 & -1 \\ 1 & -4 & -1 & 4 \end{bmatrix}.$$

Как и в преобразовании Фурье, строки матрицы обратного преобразования T^{-1} расположены в обратном порядке, т. е.

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -4 & -1 & 4 \\ 1 & -1 & 1 & -1 \\ 1 & 4 & -1 & -4 \end{bmatrix}.$$

Число, обратное N, равно N^1 = 13, так как $13 \cdot 4 = 52 \equiv 1 \bmod 17$. Прямые преобразования сворачиваемых последовательностей равны

$$A_s = TS = [1,10, 5,9]^{T};$$

$$A_н = TH = [3,9,16, 10]^{T},$$

а их произведение

$$A_s A_н = [3,90,80,90]^{\mathrm{T}} = [3,5,12,5]^{\mathrm{T}}.$$

Обратное преобразование этой последовательности дает свертку

$$Y = 13T^{-1}(A_S A_H) = (2,2,14, 2)^{T} = (2,2,-3,2)^{T}.$$

2.7.2. ТЧП Мерсенна

Наибольшее распространение на практике получили теоретико-числовые преобразования с числами Ферма (ТЧПФ) и Мерсенна (ТЧПМ) [7].

Теоретико-числовым преобразованием Мерсенна называется пара следующих преобразований:

$$X(k) \equiv \left(\sum_{n=0}^{p-1} x(nT) \pmod q\right), k = 0,..., p-1; \qquad (2.105)$$

$$x(nT) \equiv \left(p^{-1} \sum_{k=0}^{p-1} X(k) 2^{-nk}\right) \pmod q, n = 0,..., p-1, \qquad (2.106)$$

где p – простое положительное число; $q = 2^p - 1$ – простое число (число Мерсенна).

В качестве p могут быть выбраны числа 2, 3, 5, 7, 13, 17, 19, 31, 61. С точки зрения обеспечиваемого динамического диапазона наиболее широко используются числа 31 и 61. Преобразование Мерсенна не обладает структурой БПФ и для своей реализации требует $(p-1)^2$ операций сдвига и $p(p-1)$ – сложения.

Пример. С помощью ТЧПМ найдем свертку последовательностей: $x = [2, -2, 1, 0, 0]^T$; $h = [1, 2, 0, 0]^T$. Выберем $p = 5$, тогда $q = 2^5 - 1 - 31$. Матрица ТЧПМ имеет вид:

$$T_M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \\ 1 & 2^2 & 2^4 & 2^6 & 2^8 \\ 1 & 2^3 & 2^6 & 2^9 & 2^{12} \\ 1 & 2^4 & 2^8 & 2^{12} & 2^{16} \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 4 & 16 & 2 & 8 \\ 1 & 8 & 2 & 16 & 4 \\ 1 & 16 & 8 & 4 & 2 \end{bmatrix} (\mathrm{mod}\, 31).$$

Так как $5^{-1} \equiv 25 (\mathrm{mod}\ 31)$, то матрица ОТЧПМ

$$T_M{}^{-1} \equiv 5^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \\ 1 & 2^{-2} & 2^{-4} & 2^{-6} & 2^{-8} \\ 1 & 2^{-3} & 2^{-6} & 2^{-9} & 2^{-12} \\ 1 & 2^{-4} & 2^{-8} & 2^{-12} & 2^{-16} \end{bmatrix} \equiv 25 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 16 & 8 & 4 & 2 \\ 1 & 8 & 2 & 16 & 4 \\ 1 & 4 & 16 & 2 & 8 \\ 1 & 2 & 4 & 8 & 16 \end{bmatrix} (\mathrm{mod}\, 31).$$

Теперь вычисляем:

1) ТЧПМ последовательности x: $X \equiv T_M x \equiv [1, 2, 10, 19, 9]^T (\mathrm{mod}31)$;
2) ТЧПМ последовательности h: $H = T_M h \equiv [3, 5, 9, 17, 2]^T (\mathrm{mod}31)$;
3) Произведение коэффициентов полученных ТЧПМ:
$$Y \equiv HX^T \equiv [3,10,28,13,18]^T (\mathrm{mod}\ 31);$$
4) ОТЧПМ последовательности Y: $y \equiv T_M{}^{-1} Y \equiv [2, 2, 28, 2, 0]^T (\mathrm{mod}31)$.

Так как результат должен лежать в диапазоне [–15,15], то искомая свертка будет равна $y = [2, 2, -3, 2, 0]^T$.

2.7.3. ТЧП Ферма

Теоретико-числовое преобразование Ферма (ТЧПФ) является наиболее перспективным, так как позволяет использовать эффективные алгоритмы типа БПФ. В качестве модуля М выбирается одно из чисел Ферма:

$$M = F_t = 2^{2^t} + 1 = 2^b + 1, u = 2^t. \qquad (2.107)$$

Здесь F_t называется t-м числом Ферма.

Первые семь чисел равны:
$F_0 = 3$; $F_1 = 5$; $F_2 = 17$; $F_3 = 257$; $F_4 = 65537$; $F_5 = 641 \times 6700417$;
$F_6 = 274177 \times 67280421310721$

Пример 2.13. С помощью ТЧПФ вычислить свертку последовательностей: $x = [2,-2,1,0]^{-T}$; $h = [1,2,0,0]^{T}$. В качестве модуля выберем число $M = F_2 = 17$. При $N = 4$ $\alpha = 4$. Матрица ТЧПФ принимает вид:

$$T_{\Phi} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4^1 & 4^2 & 4^3 \\ 1 & 4^2 & 4^4 & 4^6 \\ 1 & 4^3 & 4^6 & 4^9 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & -1 & -4 \\ 1 & -1 & 1 & -1 \\ 1 & -4 & -1 & 4 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 13 \\ 1 & 16 & 1 & 16 \\ 1 & 13 & 16 & 4 \end{bmatrix} (\mathrm{mod}17)$$

Так как $4^{-1} \equiv -4(\mathrm{mod}17)$, то матрица обратного ТЧПФ:

$$T_{\phi}^{-1} = 4^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4^{-1} & 4^{-2} & 4^{-3} \\ 1 & 4^{-2} & 4^{-4} & 4^{-6} \\ 1 & 4^{-3} & 4^{-6} & 4^{-9} \end{bmatrix} = -4 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -4 & -1 & 4 \\ 1 & -1 & 1 & -1 \\ 1 & 4 & -1 & -4 \end{bmatrix} = 13 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 13 & 16 & 4 \\ 1 & 16 & 1 & 16 \\ 1 & 4 & 16 & 13 \end{bmatrix} (\mathrm{mod}\,17).$$

Вычисление принимает вид:

1) ТЧПФ последовательности x:

$$X \equiv T_{\Phi} x(\mathrm{mod}17) \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 13 \\ 1 & 16 & 1 & 16 \\ 1 & 13 & 16 & 4 \end{bmatrix} \times \begin{bmatrix} 2 \\ 15 \\ 1 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 18 \\ 78 \\ 243 \\ 213 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 10 \\ 5 \\ 9 \end{bmatrix} (\mathrm{mod}17).$$

2) ТЧПФ последовательности h:

$$H \equiv T_{\Phi} h(\mathrm{mod}17) \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 13 \\ 1 & 16 & 1 & 16 \\ 1 & 13 & 16 & 4 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 3 \\ 9 \\ 33 \\ 27 \end{bmatrix} \equiv \begin{bmatrix} 3 \\ 9 \\ 16 \\ 10 \end{bmatrix} (\mathrm{mod}17).$$

3) $Y \equiv XH^{T} \equiv [3, 90, 80, 90]^{\mathrm{T}} \equiv [3, 5, 12, 5]^{\mathrm{T}}(\mathrm{mod}17)$.

4) ОТЧПФ от Y дает искомую свертку: $y = T_{\Phi}^{-1} Y = [2,2,14,2]^{T}(\mathrm{mod}17)$. Так как результаты должны лежать в диапазоне $[-8,8]$, то окончательно $y = [2,2,-3,2]^{\mathrm{T}}$.

2.8. Ключевые операции ЦОС – свертка и корреляция

2.8.1. Общие сведения о корреляции и свертке

Часто бывает необходимо определить степень независимости одного процесса от другого или установить сходство одного набора данных с другим [1]. Другими словами, искомой является корреляция процессов или

данных, которую можно определить математически и измерить. Вообще, процесс корреляции занимает значительное место в обработке сигналов. Этот математический аппарат нашел применение в обработке изображений в сфере компьютерного зрения или дистанционного зондирования со спутников, где сравниваются данные с различных изображений, в радарных или гидроакустических установках для дальнометрии и местоопределения (пеленгации), в которых сравниваются переданные и отраженные сигналы. Он используется в детектировании и идентификации сигналов в шуме, в организации технического контроля для наблюдения за влиянием входа на выход, в идентификации двоичных кодовых слов в системе с импульсно-кодовой модуляцией, в обычных схемах оценки по методу наименьших квадратов и во многих других областях, например, в климатологии. Корреляция также является неотъемлемой частью процесса свертки, который, по сути, – та же корреляция двух последовательностей данных, при вычислении которой одна из последовательностей обращена во времени. Это означает, что для вычисления корреляции и свертки могут использоваться одни и те же алгоритмы, только в случае свертки одна из последовательностей обращается. Отметим также, что спектр записанного сигнала состоит из свертки спектра сигнала со спектром вырезающей функции [1].

Определение импульсной характеристики неизвестной системы называется *идентификацией системы*. Определение неизвестного входа по импульсной характеристике системы по выходу называется *обращением свертки* (нахождением оригинала функции). Если импульсная характеристика неизвестна, определение искомого входного сигнала называется *слепым обращением свертки* (*blind deconvolution*). Все эти важные темы рассмотрены ниже

2.8.2. Описание корреляции

Рассмотрим, как можно сравнить две последовательности данных, состоящие из значений, одновременно выбираемых из двух соответствующих сигналов. Если два сигнала похоже меняются при переходе от точки к точке, то меру их *корреляции* можно вычислить, взяв сумму произведений соответствующих пар точек. Это предположение становится более аргументированным, если рассмотреть две независимые и случайные последовательности данных. В этом случае сумма произведений стремится к нулю по мере увеличения пар точек. Это объясняется тем, что все числа, положительные и отрицательные, равновероятны, так что пары произведений компенсируются при сложении. В то же время, если сумма конечна, это указывает на наличие корреляции. Отрицательная сумма указывает на от-

рицательную корреляцию, т.е. увеличение одной переменной связано с уменьшением другой. Таким образом, взаимную корреляцию $r_{12}\ (n)$ двух последовательностей данных $x_1(n)$ и $x_2\ (n)$, содержащих по N элементов, можно записать как

$$r_{12} = \sum_{n=0}^{N-1} x_1(n)x_2(n)\,. \qquad (2.108)$$

Впрочем, такое определение взаимной корреляции дает результат, который зависит от числа взятых точек. Чтобы это исправить, результат нормируется на число точек (делится на N). Данную операцию можно также рассматривать как усреднение суммы произведений. Итак, получаем следующее улучшенное определение:

$$r_{12} = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n)\,. \qquad (2.109)$$

Пример 2.14

Ниже иллюстрируется расчет r_{12}. Число точек данных в последовательности равно n, используются последовательности x_1 и x_2.

n	1	2	3	4	5	6	7	8	9
x_1	4	2	−1	3	−2	−6	−5	4	5
x_2	−4	1	3	7	4	−2	−8	−2	1

$$r_{12} = \frac{1}{9}(4\times(-4) + 2\times1 + (-1)\times3 + 3\times7 + (-2)\times4 + (-6)\times(-2) +$$
$$+ (-5)\times(-8) + 4\times(-2) + 5\times1) = 5.$$

Впрочем, чтобы данное определение можно было использовать, его также нужно модифицировать. В некоторых случаях корреляция, определенная указанным выше способом, может быть нулевой, хотя две последовательности коррелируют на 100 %. Это может произойти, например, когда два сигнала идут не в фазе (как часто и бывает). Данная ситуация иллюстрируются сигналами на рис. 2.20. На рисунке показано, что каждая пара произведений в функции корреляции равна нулю, следовательно, вся корреляция равна нулю, поскольку нулю всегда равно одно из значений x_1 или x_2. Впрочем, очевидно, что сигналы сильно коррелируют, хотя и идут не в фазе. Разность фаз может, например, объясняться тем, что x_1 – некий эталонный сигнал, а x_2 – запаздывающий выход схемы. Чтобы преодолеть подобный сдвиг фаз, необходимо сдвинуть (или задержать) один из сигналов относительно другого. Обычно, чтобы выровнять сигналы перед определением корреляции, x_2 смещается влево. Как показано на рис. 2.21, это эквивалентно замене $x_2(n)$ на $x_2(n + j)$, где j представляет величину за-

держки – число точек выборки, на которое x_2 смещается влево. Альтернативной и эквивалентной процедурой является смещение x_1 вправо. В результате получаем такую формулу для взаимной корреляции:

$$r_{12}(j) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n+j) = r_{12}(-j) = \frac{1}{N}\sum_{n=0}^{N-1} x_2(n)x_1(n-j). \quad (2.110)$$

На практике, когда два сигнала коррелируют, их фазовая связь скорее всего неизвестна, так что корреляцию нужно находить для нескольких различных задержек, чтобы установить наибольшее значение корреляции, которое затем считается истинным.

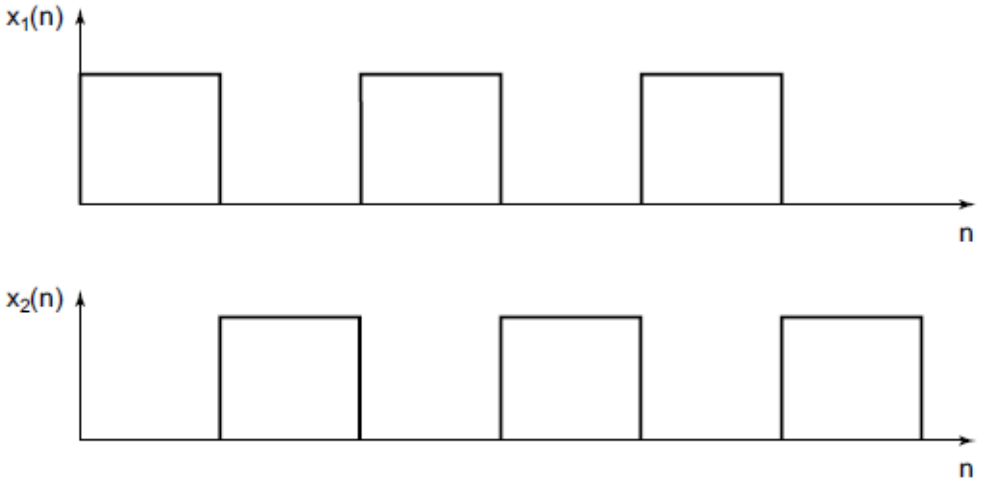


Рис. 2.20. Сигналы со 100 %-й корреляцией, идущие не в фазе, – при нулевой задержке корреляция равна нулю.

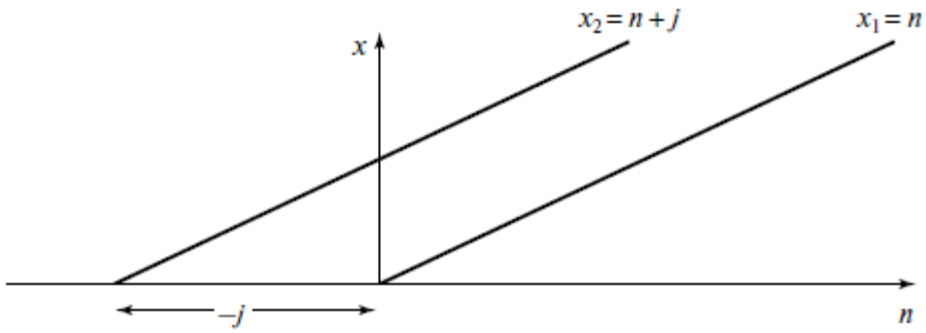


Рис. 2.21. Сигнал $x_2 = x_1 + j$, смещенный на j промежутков времени влево от сигнала x_1

Пример 2.15

Рассмотрим взаимную корреляцию указанных выше двух последовательностей $x_1(n)$ и $x_2(n)$ с задержкой $j = 3$, т.е. $r_{12}(3)$. Итак, используются следующие последовательности:

n	1	2	3	4	5	6	7	8	9
x_1	4	2	−1	3	−2	−6	−5	4	5
x_2	7	4	−2	−8	−2	−1,			

так что

$$r_{12} = \frac{1}{9}(4\times 7 + 2\times 4 + (-1)\times(-2) + 3\times(-8) + (-2)\times(-2) +$$
$$+(-6)\times(-1)) = 2,667.$$

Разумеется, также можно рассмотреть корреляцию в непрерывной временной области, некоторые аналоговые схемы корреляции организованы именно так. В непрерывной области $n \to t$, $j \to \tau$ и

$$r_{12}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x_1(t)x_2(t+\tau)dt. \qquad (2.111)$$

В то же время, если $x_1(t)$ и $x_2(t)$ – периодические с периодом T, формула (2.111) упрощается до

$$r_{12}(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x_1(t)x_2(t+\tau)dt. \qquad (2.112)$$

Если данные – это сигналы с конечной энергией, например, непериодические импульсообразные сигналы, то усреднение по времени T при $T \to \infty$ не выполняется, поскольку в этом случае было бы $1/T \to 0$ и $r_{12}(\tau)$ было бы исчезающе малой величиной. В таком случае используется следующая формула:

$$r_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t)x_2(t+\tau). \qquad (2.113)$$

На практике обрабатываться будут записи конечной длины, так что применяется формула (2.110) или

$$r_{12}(\tau) = \frac{1}{T}\int_0^T x_1(t)x_2(t+\tau)dt. \qquad (2.114)$$

Есть и другая сложность, связанная с нахождением взаимной корреляции последовательностей данных конечной длины. Это иллюстрирует приведенный выше пример, в котором было найдено значение $r_{12}(3) = 2,667$. При смещении x_2 влево сигналы уже не перекрываются и

данные в конце последовательностей не формируют парные произведения – возникает так называемый *краевой эффект*. В рассмотренном примере число пар при задержке 3 уменьшилось с девяти до шести. В результате наблюдалось линейное уменьшение $r_{12}(j)$ при увеличении j и получены спорные значения $r_{12}(j)$. Одно из возможных решений возникшей проблемы заключается в том, чтобы длину одной последовательности сделать в два раза больше длины, необходимой для нахождения корреляции. Для этого можно записать больше данных или, если одна из последовательностей периодична, повторить последовательность (особое внимание следует обратить на согласование краев). Другое возможное решение – скорректировать все рассчитанные значения. На рис. 2.22 показано уменьшение r_{12} с ростом j в результате краевого эффекта, т.е. реальное изменение $r_{12}(j)$ не показано. При $j = 0$ $r_{12}(j) = r_{12}(0)$, что можно вычислить. При $j = N$, $r_{12}(N) = 0$, поскольку сигналы уже не перекрываются. В промежуточных случаях при некоторых значениях задержки j истинное значение $r_{12}(j)$ равно $r_{12}(j)_{true}$, тогда как действительное значение, искаженное краевым эффектом, равно $r_{12}(j)$. Далее на основе рисунка получаем

$$\frac{r_{12}(j)_{true} - r_{12}(j)}{j} = \frac{r_{12}(0)}{N}, \tag{2.115}$$

откуда

$$r_{12}(j)_{true} = r_{12}(j) + \frac{j}{N} r_{12}(0). \tag{2.116}$$

Таким образом, вычисленные значения взаимной корреляции легко скорректировать для учета краевых эффектов, прибавив к $r_{12}(j)$ величину $jr_{12}(0)/N$.

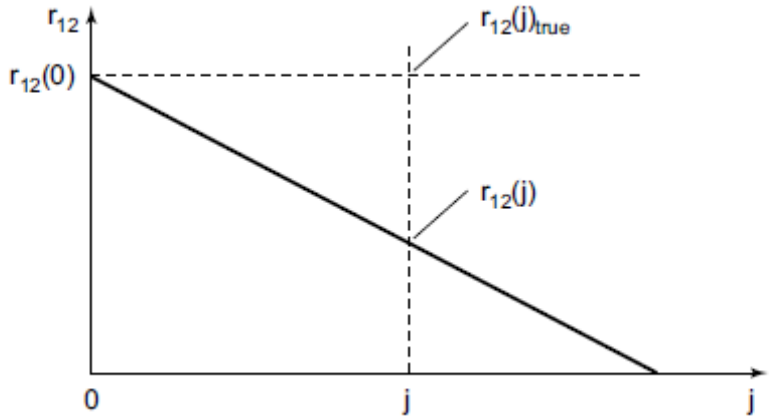


Рис. 2.22. Влияние краевого эффекта на взаимную корреляцию $r_{12}(j)$

Значения функции взаимной корреляции вычисляются согласно приведенным выше формулам в зависимости от абсолютных значений данных. Часто бывает необходимо измерить взаимную корреляцию в фиксированном масштабе между –1 и +1. Чтобы определить значения из указанного диапазона, полученные величины нормируют на величину, зависящую от энергетического содержания данных. Например, рассмотрим две пары сигналов $x_1(n)$, $x_2(n)$ и $x_3(n)$, $x_4(n)$. Значения элементов данных приведены ниже.

n	0	1	2	3	4	5	6	7	8
$x_1(n)$	0	3	5	5	5	2	0,5	0,25	0
$x_2(n)$	1	1	1	1	1	1	0	0	0
$x_3(n)$	0	9	15	15	15	6	1,5	0,75	0
$x_4(n)$	2	2	2	2	2	2	0	0	0

Как показано на рис. 2.23, сигналы $x_1(n)$ и $x_3(n)$ подобны и отличаются только амплитудой. То же справедливо для пары $x_2(n)$ и $x_4(n)$. Таким образом, корреляция между $x_1(n)$ и $x_2(n)$ равна корреляции между $x_3(n)$ и $x_4(n)$. В то же время параметры корреляции $r_{12}(2)$ и $r_{34}(2)$ равны соответственно 1,44 и 8,67. Они отличаются, поскольку зависят от абсолютных значений элементов данных. Чтобы поправить эту ситуацию, нормируем взаимную корреляцию $r_{12}(j)$ на коэффициент

$$\left[\frac{1}{N}\sum_{n=0}^{N-1}x_1^2(n)\times\frac{1}{N}\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}=\frac{1}{N}\left[\sum_{n=0}^{N-1}x_1^2(n)\times\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}, \quad (2.117)$$

и подобным образом нормируем $r_{34}(j)$. В результате нормированное выражение для $r_{12}(j)$ приобретает такую форму:

$$\rho_{12}(j)=\frac{r_{12}(j)}{\frac{1}{N}\left[\sum_{n=0}^{N-1}x_1^2(n)\times\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}}. \quad (2.118)$$

Величина $\rho_{12}(j)$ известна как *коэффициент взаимной корреляции*. Значение этого коэффициента всегда лежит между –1 и +1, причем означает 100 %-ю корреляцию в прямом смысле, «–1» – 100 %-ю корреляцию в противоположном смысле, например, как для сигналов в противофазе. Значение «0» указывает на нулевую корреляцию. Это означает, что сигналы совершенно независимы, например, если один из сигналов абсолютно случаен.

Малые значения $\rho_{12}(j)$ указывают на незначительную корреляцию. Нормировочный коэффициент для $r_{12}(j)$ из приведенного выше примера равен

$$\frac{1}{N}\left[\sum_{n=0}^{N-1}x_1^2(n)\times\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}=\frac{1}{9}(88{,}31\times6)^{1/2}=2{,}56,$$

а для $r_{34}(j)$

$$\frac{1}{N}\left[\sum_{n=0}^{N-1} x_1^2(n)\times\sum_{n=0}^{N-1} x_2^2(n)\right]^{1/2} = \frac{1}{9}(794{,}8\times 24)^{1/2} = 15{,}35.$$

Следовательно,

$$\rho_{12}(2) = \frac{r_{12}(2)}{2{,}56} = \frac{1{,}44}{2{,}56} = 0{,}56$$

и

$$\rho_{34}(2) = \frac{r_{34}(2)}{15{,}35} = \frac{8{,}67}{15{,}35} = 0{,}56\,.$$

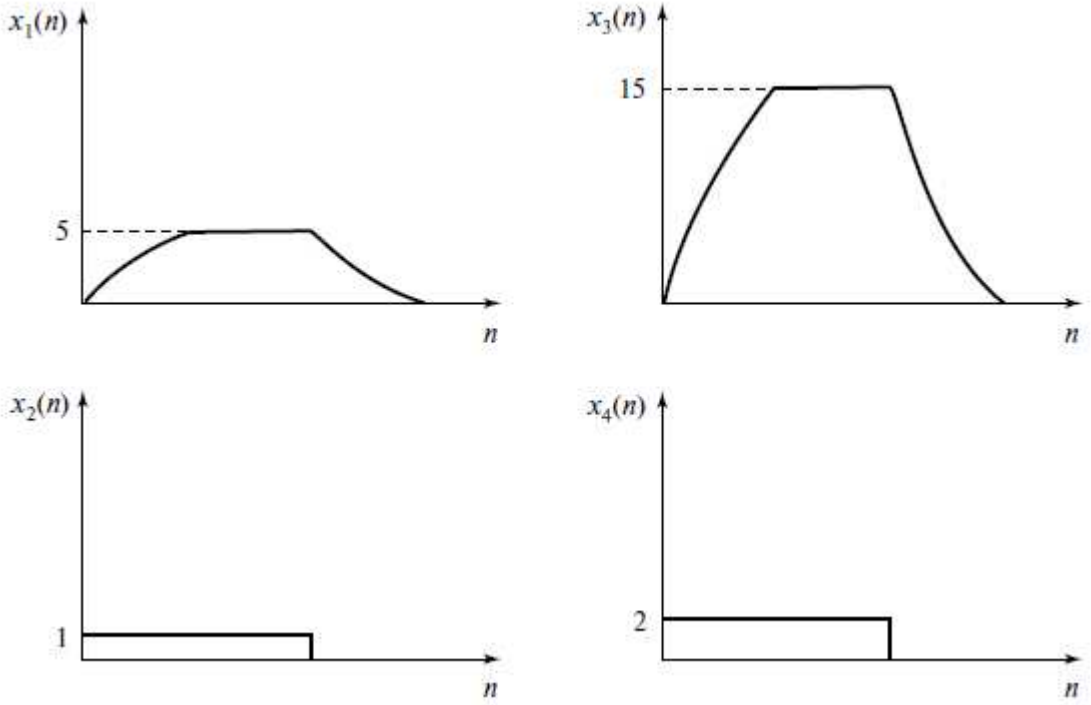


Рис. 2.23. Пары сигналов $\{x(n)x(n)\}$ и $\{x(n)x(n)\}$ различных амплитуд, но с равными взаимными корреляциями

Теперь $\rho_{12}(2) = \rho_{34}(2)$, откуда видно, что данный процесс нормировки действительно позволяет независимо сравнивать взаимные корреляции абсолютных значений данных.

2.8.3. Взаимная корреляция и автокорреляция

Рассмотрим частный случай $x_1(n) = x_2(n)$, т.е. найдем корреляцию сигнала с самим собой. Данный процесс называется *автокорреляцией*. Автокорреляционная функция сигнала определяется как

$$r_{11}(j) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_1(n+j). \qquad (2.119)$$

Автокорреляционная функция имеет одно весьма полезное свойство:

$$r_{11}(0) = \frac{1}{N}\sum_{n=0}^{N-1} x_1^2(n) = S, \qquad (2.120)$$

где S – нормированная энергия сигнала.

В результате получаем метод расчета энергии сигнала. Если сигнал абсолютно случаен, например, сигнал, соответствующий белому гауссову шуму в электрической системе, его автокорреляция будет максимальной при нулевой задержке и уменьшается до случайных флуктуаций малой амплитуды возле нуля для задержек, превышающих единицу (рис. 2.24). Кроме того, справедливо следующее соотношение:

$$r_{11}(0) \geq r_{11}(j). \qquad (2.121)$$

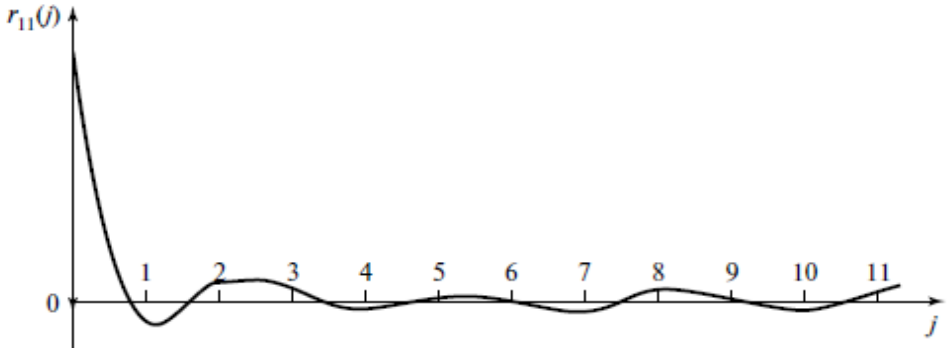


Рис. 2.24. Автокорреляционная функция случайного сигнала

Определение взаимной корреляции двух периодических последовательностей неравной длины требует аккуратности. Это объясняется тем, что результат корреляции будет повторяться с периодом более короткой последовательности. Этот результат не отражает полной периодичности более длинной последовательности, следовательно, неверен. Продемонстрируем это, найдя взаимную корреляцию $r_{ab}(j)$ последовательностей $a = \{4, 3,1, 6\}$ и $b = \{5, 2, 3\}$. Последовательность b записывается под a и поэтапно смещается на одну позицию влево, в последнем столбце записываются соответствующие значения взаимной корреляции.

Последовательность				Задержка	$r_{ab}(j)$
4	3	1	6		
3	5	2	3	0	47
5	2	3	5	1	59
2	3	5	2	2	34
3	5	2	3	3	47 $r_{ab}(j)$ повторяется
5	2	3	5	4	59

и т.д.

Результат показывает, что $r_{ab}(j)$ циклично с периодом в три задержки, т.е. период $r_{ab}(j)$ равен периоду более короткой последовательности b. Описанная процедура называется *циклической корреляцией*. Чтобы получить правильное значение, в котором каждое значение a умножается на каждое значение b, все элементы b нужно последовательно сместить под каждым значением a:

```
                4   3   1   6
                        5   2   3
                    5   2   3
                5   2   3
            5   2   3
        5   2   3
    5   2   3
5   2   3
```

Видно, что для того, чтобы последовательность b стала повторяться, требуется 6 задержек. Длины последовательностей равны 4 и 3, а число требуемых задержек: 4 + 3 – 1 = 6. Таким образом, получаем общее правило нахождения линейной взаимной корреляции двух периодических последовательностей длины N_1 и N_2: дополнить нулями каждую последовательность, чтобы их длины были равны $N_1 + N_2 – 1$ (т.е. добавить $N_2 – 1$ нулей к последовательности длиной N_1 и $N_1 – 1$ нулей к последовательности длиной N_2). Вышесказанное иллюстрируется для указанных выше последовательностей a и b.

Последовательность	Задержка	$r_{ab}(j)$	
4 3 1 6 0 0			
5 2 3 0 0 0	0	29	
2 3 0 0 0 5	1	17	
3 0 0 0 5 2	2	12	
0 0 0 5 2 3	3	30	
0 0 5 2 3 0	4	17	
0 5 2 3 0 0	5	35	
5 2 3 0 0 0	6	29	r_{ab} (j) повторяется

Следовательно, искомая взаимная корреляция a и b равна

$$r_{ab}(\text{j}) = \{29,17,12, 30,17, 35\}.$$

2.8.4. Быстрая корреляция

Расчет корреляции можно ускорить, используя теорему о корреляции, которая обычно формулируется следующим образом:

$$r_{12}(j) = F_D^{-1}[X_1^*(k)X_2(k)], \tag{2.122}$$

хотя корректной является такая формулировка:

$$r_{12}(j) = \frac{1}{N} F_D^{-1}[X_1^*(k)X_2(k)], \tag{2.123}$$

где F_D^{-1} обозначает обратное дискретное преобразование Фурье. Данный подход требует выполнения двух дискретных преобразований Фурье (ДПФ) и одного обратного ДПФ, что легче всего сделать, используя алгоритм БПФ. Если число членов в последовательностях достаточно велико, данный метод БПФ дает результат быстрее, чем непосредственный расчет взаимной корреляции.

Доказательство теоремы о корреляции

Пусть $x_1(r)$, $x_2(r)$ и $x_3(n)$ – периодические последовательности длины N, и пусть их ДПФ-образы равны соответственно $X_1(k)$, $X_2(k)$ и $X_3(k)$. Более того, пусть

$$X_3(k) = X_1^*(k)X_2(k). \tag{2.124}$$

В этом случае

$$X_1^*(k) = \sum_{i=0}^{N-1} x_1(l) e^{(2\pi i/N)lk} \tag{2.125}$$

и

$$X_2^*(k) = \sum_{i=0}^{N-1} x_1(l) e^{(2\pi i/N)(-rk)}. \tag{2.126}$$

Подставляя уравнения (2.124) и (2.125) в (2.123), получаем

$$X_3(k) = \sum_{i=0}^{N-1} x_1(l) e^{(2\pi i/N)lk} \sum_{i=0}^{N-1} x_2(l) e^{(2\pi i/N)(-rk)} = \tag{2.127}$$

$$= \sum_{l=0}^{N-1}\sum_{r=0}^{N-1} x_1(l)x_2(r) e^{(2\pi i/N)(lk-rk)}, \tag{2.128}$$

далее

$$x_2(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_3(k) e^{(2\pi i/N)nk}. \tag{2.129}$$

Таким образом, подставляя уравнение (2.127) в (2.128), получаем

$$\begin{aligned} x_3(n) &= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\sum_{r=0}^{N-1} x_1(l)x_2(r) e^{(2\pi i/N)(lk-rt+nk)} = \\ &= \frac{1}{N}\sum_{l=0}^{N-1} x_1(l)\sum_{r=0}^{N-1} x_2(r)\left[\sum_{k=0}^{N-1} e^{(2\pi i/N)(l-r+n)k}\right]. \end{aligned} \tag{2.130}$$

При $r = n + l$ выражение в квадратных скобках равно N. При $r \neq n + l$ его можно рассматривать как геометрическую прогрессию вида $\sum ax^n$, сумма которой по N членам равна $\frac{a(1-x^N)}{1-x}$. В нашем случае сумма равна

$$\frac{1[1-e^{(2\pi i/N)(l-r+n)N}]}{1-e^{(2\pi i/N)(l-r+n)}}. \qquad (2.131)$$

Показатель экспоненты в числителе всегда кратен 2π, так что ее значение равно 1. Таким образом, сумма равна нулю при $r \neq n + l$. Следовательно, уравнение (2.130) можно переписать как

$$x_3(n) = \frac{1}{N}\sum_{l=0}^{N-1} x_1(l) \sum_{r=0}^{N-1} x_2(r) N\delta(l - r + n), \qquad (2.132)$$

где $\delta(l - r + n) = 1$ при $r = n + l$ и $\delta(l - r + n) = 0$ при $r \neq n + l$.

Положив $r = n + l$ и упростив, получаем:

$$x_3(n) = \sum_{l=0}^{N-1} x_1(l) x_2(l + n), \qquad (2.133)$$

или

$$\frac{1}{N} x_3(n) = \frac{1}{N}\sum_{l=0}^{N-1} x_1(l) x_2(l + n). \qquad (2.134)$$

Правая часть этого выражения эквивалентна взаимной корреляции $x_1(n)$ и $x_2(n)$, и, как легко видеть, она равна $(1/N)x_3(n)$. Из уравнения (2.129) получаем

$$x_3(n) = F_D^{-1}[X_3(n)]. \qquad (2.135)$$

Далее, объединяя уравнения (2.134), (2.135) и (2.124), получаем

$$\frac{1}{N} F_D^{-1}[X_3(n)] = r_{12}(n) = \frac{1}{N} F_D^{-1}[X_1^*(n) X_2(n)]. \qquad (2.136)$$

Наконец, заменяя n на j, получаем

$$r_{12}(j) = \frac{1}{N} F_N^{-1}[X_1^*(n) X_2(n)]. \qquad (2.137)$$

Пример 2.16

Используя теорему о корреляции, найдем взаимную корреляцию двух последовательностей $x_1(n)$ и $x_2(n)$.

$$x_1(n) = \{1;0;0;1\};$$
$$x_2(n) = \{0,5;1;1;0,5\}.$$

Используем теорему о корреляции – уравнение (2.137). Для этого необходимо вычислить ДПФ $x_1(n)$. Из примера 2.11 $X_1(n)$ равно

$$X_1(n) = 2;1+i;0;1-i,$$

так что

$$X_1^*(n) = 2;1-i;0;1+i.$$

Для получения $X_2(n)$ проще всего использовать алгоритм БПФ. Таким образом, при $x_0 = 0\,,5$, $x_2 = 1$, $x_1 = 1$ и $x_3 = 0{,}5$ получаем

$$X_{21}(0) = x_0 + x_2 = 1,5,$$
$$X_{21}(1) = x_0 - x_2 = -0,5,$$
$$X_{22}(0) = x_1 + x_3 = 1,5,$$
$$X_{22}(1) = x_1 - x_3 = 0,5,$$
$$X_{11}(0) = X_{21}(0) + X_{22}(0) = 3,$$
$$X_{11}(1) = X_{21}(1) + (-i)X_{22}(1) = -0,5 - 0,5i,\ X_{11}(2) = X_{21}(0) - X_{22}(0) = 0,$$
$$X_{11}(1) = X_{21}(1) - (-i)X_{22}(1) = -0,5 + 0,5i.$$

Собирая значения БПФ, получаем

$$X_1^* = 2;1-i;0;1+i,$$
$$X_2(n) = 3;-0,5-0,5i;0;-0,5+0,5i.$$

Так что

$$X_1^*(0)X_2(0) = 2\times 3 = 6,$$
$$X_1^*(1)X_2(1) = (1-i)\times(-0,5-0,5i) = -1,$$
$$X_1^*(2)X_2(2) = 0\times 0 = 0,$$
$$X_1^*(3)X_2(3) = 0,5\times(1+i)\times(-1+i) = -1.$$

Следовательно,

$$[X_1^*(n)X_2(n)] = 6;-1;0;-1.$$

Теперь необходимо к этому результату применить обратное ДПФ. Обратное ДПФ получается путем замены знаков экспонент (в весовых коэффициентах W_N) приведенного выше алгоритма БПФ и деления результата на N. Следовательно, получаем такой результат:

$$X_{21}(0) = x_0 + x_2 = 6,$$
$$X_{21}(1) = x_0 - x_2 = 6,$$
$$X_{22}(0) = x_1 + x_3 = -2,$$

$$X_{22}(1) = x_1 - x_3 = 0,$$
$$X_{11}(0) = X_{21}(0) + X_{22}(0) = 4,$$
$$X_{11}(1) = X_{21}(1) + iX_{22}(1) = 6,$$
$$X_{11}(2) = X_{21}(0) - X_{22}(0) = 8,$$
$$X_{11}(1) = X_{21}(1) - iX_{22}(1) = 6.$$

Компоненты $F_D^{-1}[X_1^*(n)X_2(n)]$ получаются делением значений $X_{11}(0)$, $X_{11}(1)$, $X_{11}(2)$ и $X_{11}(3)$ на $N = 4$. Таким образом,

$$F_D^{-1}[X_1^*(n)X_2(n)] = 1;1,5;2;1,5.$$

Далее из уравнения (2.136) определяем

$$r_{12}(i) = \frac{1}{4}F_D^{-1}[X_1^*(n)X_2(n)] = \{0,25;0,375;0,5;0,375\}.$$

Эта корреляция будет круговой, поскольку все данные периодичны с периодом N. Если непосредственно посчитать взаимную корреляцию $r_{12}(i)$, то получатся такие значения:

$$r_{12}(0) = (1\times0,5 + 0 + 0 + 1\times0,5)/4 = 0,25,$$
$$r_{12}(1) = (1\times1 + 0 + 0 + 1\times0,5)/4 = 0,375,$$
$$r_{12}(2) = (1\times1 + 0 + 0 + 1\times1)/4 = 0,5,$$
$$r_{12}(3) = (1\times0,5 + 0 + 0 + 1\times1)/4 = 0,25.$$

Следующее значение $r_{12}(4)$ снова равно 0,25, и последовательность периодически повторяется. Данный результат согласуется с данными, полученными выше с помощью теоремы о корреляции.

Теорему о корреляции можно использовать для получения линейной корреляции путем добавления к двум последовательностям дополняющих нулей. Следовательно, если последовательность $x_1(n)$ имеет длину N_1, а последовательность $x_2(n) - N_2$, то $x_1(n)$ дополняется $(N_2 - 1)$ нулями, а $x_2(n) - (N_1 - 1)$ нулями. Далее на основе двух расширенных последовательностей рассчитывается взаимная корреляция. Этот метод вычисления взаимной корреляции с помощью теоремы о корреляции и БПФ называется *быстрой корреляцией*.

2.8.5. Описание свертки. Линейная свертка

Свертка, помимо прочего, описывает, как выход системы определяется взаимодействием входа с самой системой. Обычно выход системы является запаздывающей и подавленной или усиленной версией входа. В этой связи особенно полезно рассмотреть выход системы, порожденный

импульсным входом. Это объясняется тем, что любой вход можно представить как последовательность импульсов разной мощности. Итак, выход системы, инициированный импульсным входом, не будет соответствующим импульсом, а будет меняться со временем, в некоторый момент достигая максимального значения, как показано на рис. 2.25. Из этого рисунка видно, что в момент взятия выборки m выход, порожденный единичным импульсом, поданным в момент 0, равен $h(m)$. Данная величина называется *импульсной характеристикой* системы или ее *импульсным откликом*.

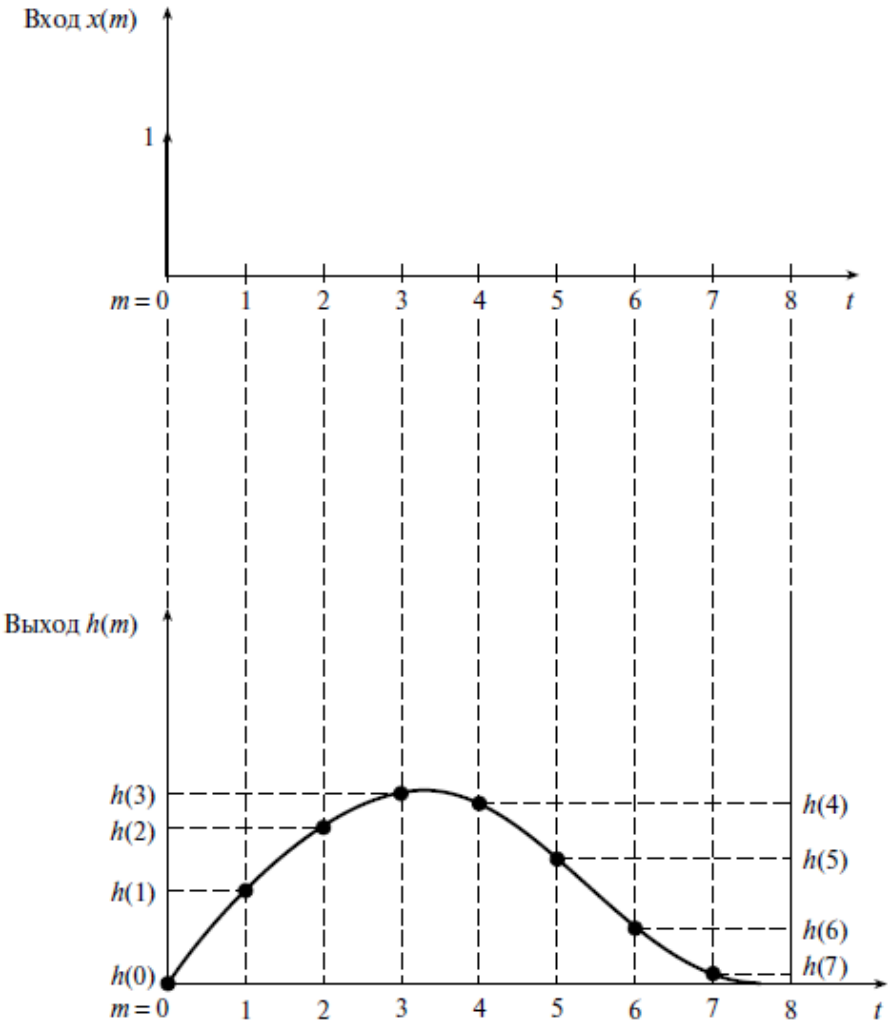


Рис. 2.25. Входной импульс и соответствующая импульсная характеристика системы

Рассмотрим подачу на вход системы последовательности импульсов $x(m)$ в моменты времени m. В контексте рис. 2.26 – это выход в момент времени $y(0)$, причем

$$y(0) = h(0)x(0).$$

В дискретный момент времени $m=1$ выход равен $h(0)x(1)$ (влияние текущего входа $x(1)$) плюс $h(1)x(0)$ (запаздывающее влияние входа, поданного в момент $m=0$). Следовательно,

$$y(1)=h(1)x(0)+h(0)x(1).$$

Таким образом, последующие выходы запишутся так:

$$\begin{aligned} y(2)&=h(2)x(0)+h(1)x(1)+h(0)x(2);\\ y(3)&=h(3)x(0)+h(2)x(1)+h(1)x(2)+h(0)x(3);\\ &\vdots\\ y(n)&=h(n)x(0)+h(n-1)x(1)+\ldots+h(0)x(n). \end{aligned} \qquad (2.138)$$

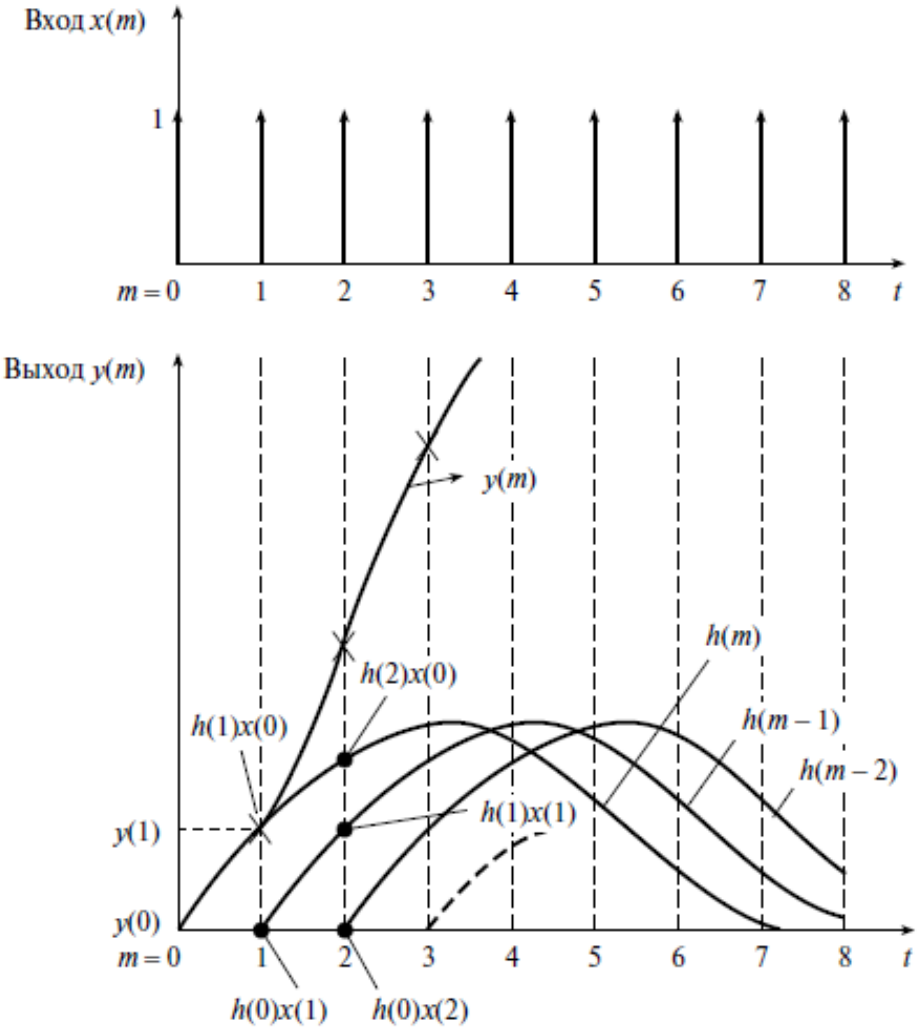


Рис. 2.26. Поданная на вход импульсная характеристика системы, полученная из отдельных откликов

Если система линейна, выход можно записать как линейную сумму влияния предыдущих входов. Выход линейной системы первого порядка описывается уравнением (2.138).

Изучая приведенные выражения, находим, что выход получается умножением входной последовательности на соответствующие точки обращенной во времени функции импульсной характеристики. Альтернативный вариант – записать уравнение (2.138) в таком виде:

$$y(n) = h(0)x(n) + h(1)x(n-1) + \ldots + h(n)x(0) \qquad (2.139)$$

и рассматривать выход как произведение соответствующих пар точек в функции импульсной характеристики и обращенной во времени входной последовательности. Следовательно, сверточная сумма эквивалента взаимной корреляции одной последовательности и обращенной во времени другой.

Уравнения (2.138) и (2.139) можно записать компактно:

$$y(n) = \sum_{m=0}^{n} h(n-m)x(m) \qquad (2.140)$$

и

$$y(n) = \sum_{m=0}^{n} h(m)x(n-m). \qquad (2.141)$$

Эти функции называются *сверточными суммами* входов с импульсной характеристикой, а выход *находится как свертка входа с импульсной характеристикой системы*.

Уравнения (2.140) и (2.141) можно расширить на сигналы бесконечной длительности, записав их следующим образом:

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = x(n) \otimes h(n) \qquad (2.142)$$

и

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) = h(n) \otimes x(n). \qquad (2.143)$$

В приведенных уравнениях символ «$\otimes$» обозначает операцию свертки.

Если вход состоит из непрерывной последовательности импульсов, приведенную выше сумму можно заменить интегралом, например, уравнение (2.142) при этом приводится к виду

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda, \qquad (2.144)$$

что называется *интегралом свертки*.

До этого момента термином *свертка* назывался результат свертки импульсной характеристики системы со входом системы. В то же время данную концепцию можно расширить на свертку любых двух наборов данных и рассматривать термин в более широком смысле.

В качестве примера свернем две периодические во времени последовательности {4,3,2,1} ($h(m)$) и {1,2,3,4} ($x(m)$). На рис. 2.27, *а* показана периодическая последовательность {4,3,2,1} ($h(m)$), а на рис. 2.27, *б* – обращенная во времени последовательность ($h(-m)$), {1,2,3,4}. (Напомним, что сверточная сумма требует, чтобы одна из последовательностей умножалась на обращенную во времени вторую последовательность, т.е. свертка соответствует взаимной корреляции одной последовательности с обращенной во времени второй.) На рисунке также показано окно, ширина которого равна одному периоду, по которому выполняется свертка. Очевидно, что полученный результат будет периодическим, как и при нахождении взаимной корреляции, так что необходимо найти только свертку по выделенному интервалу. На рис. 2.27, *е* для справки показана вторая последовательность {1,2,3,4} ($x(m)$).

При $n = 0$ уравнение (2.139) переходит в следующее:

$$y(0) = \sum_{m=0}^{n} h(-m)x(m).$$

Согласно этой формуле взаимная корреляция данных в окне на рис. 2.27, *б* и *е* равна

$$y(0) = 4 \cdot 1 + 1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 = 24.$$

При $n = 1$ уравнение (2.139) переходит в следующее:

$$y(1) = \sum_{m=0}^{n} h(1-m)x(m),$$

и согласно этой формуле взаимная корреляция данных в окне на рис. 2.27, *в* и *е* равна

$$y(1) = 3 \cdot 1 + 4 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 = 22.$$

Следовательно,

$$y(2) = 2 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 + 1 \cdot 4 = 24.$$

и

$$y(3) = 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 + 4 \cdot 4 = 30.$$

Далее выходная последовательность (см. рис. 2.27, *ж*) циклично повторяется.

Если сигналы поддаются строгому математическому описанию, свертку можно выполнить аналитически. Чтобы лучше понять процесс свертки, рассмотрим подобный пример и проиллюстрируем необходимые этапы графически.

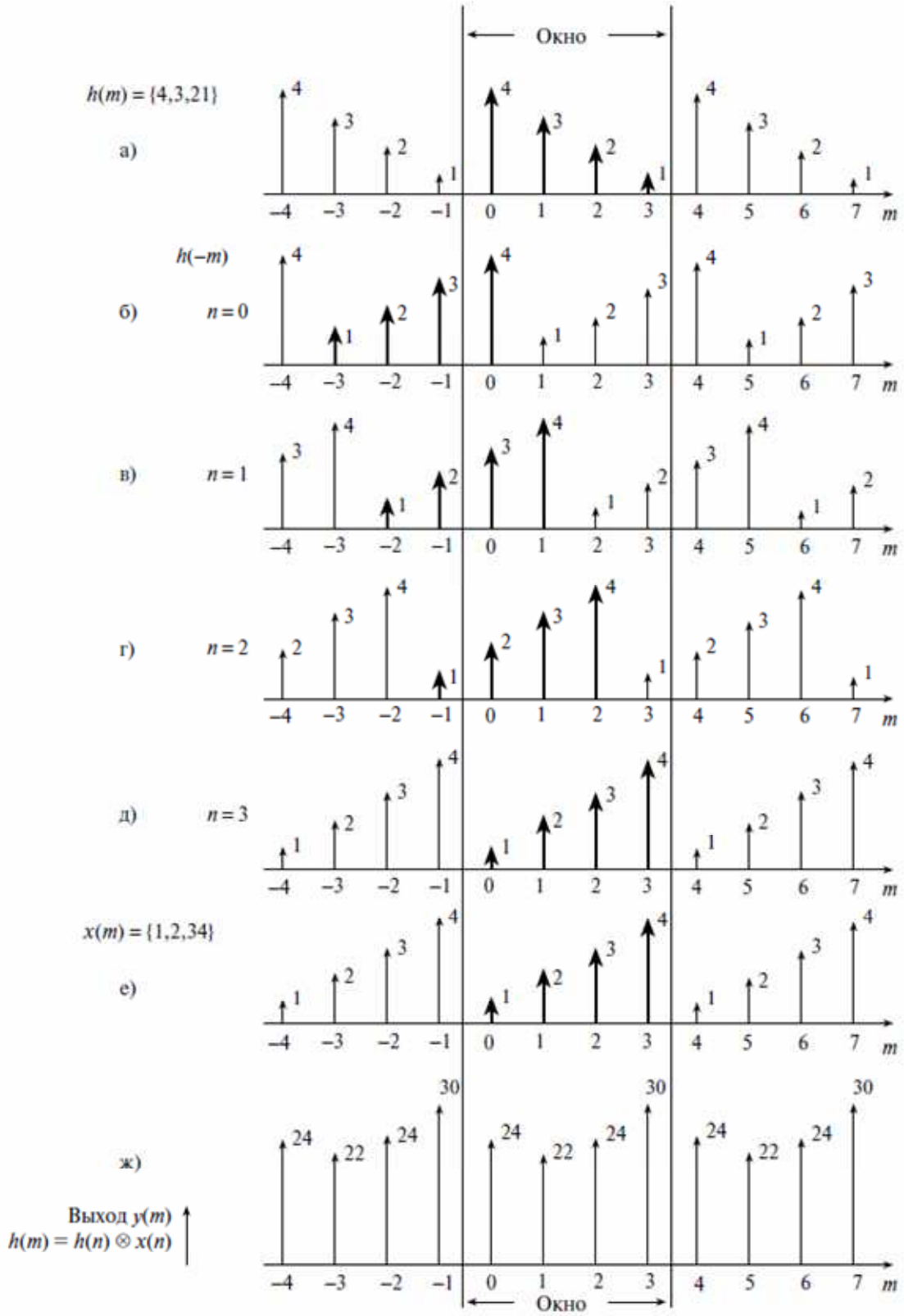


Рис. 2.27. Свертка $y(m)$ последовательностей $h(n)$ и $x(n)$: *а* – периодическая последовательность $h(m)$; *б* – обращенная во времени последовательность $h(-m)$; *в – д* – копии $h(-m)$, последовательно смещенные в вправо; *е* – последовательность $x(m)$; *ж* – выходная последовательность $y(m) = h(n) \otimes x(n)$

Пример 2.17

Найдем аналитически свертку сигналов $x(t)$ и $h(t)$, изображенных на рис. 2.28, *а*. Пусть интеграл свертки записывается как

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau . \qquad (2.145)$$

Уравнение (2.145) соответствует уравнению (2.144), в котором переменную λ заменили на τ, чтобы указать, что теперь рассматривается задача с задержками. Интеграл свертки зависит от переменной τ, так что рис. 2.28, *а* нужно заменить на рис. 2.28, *б*.

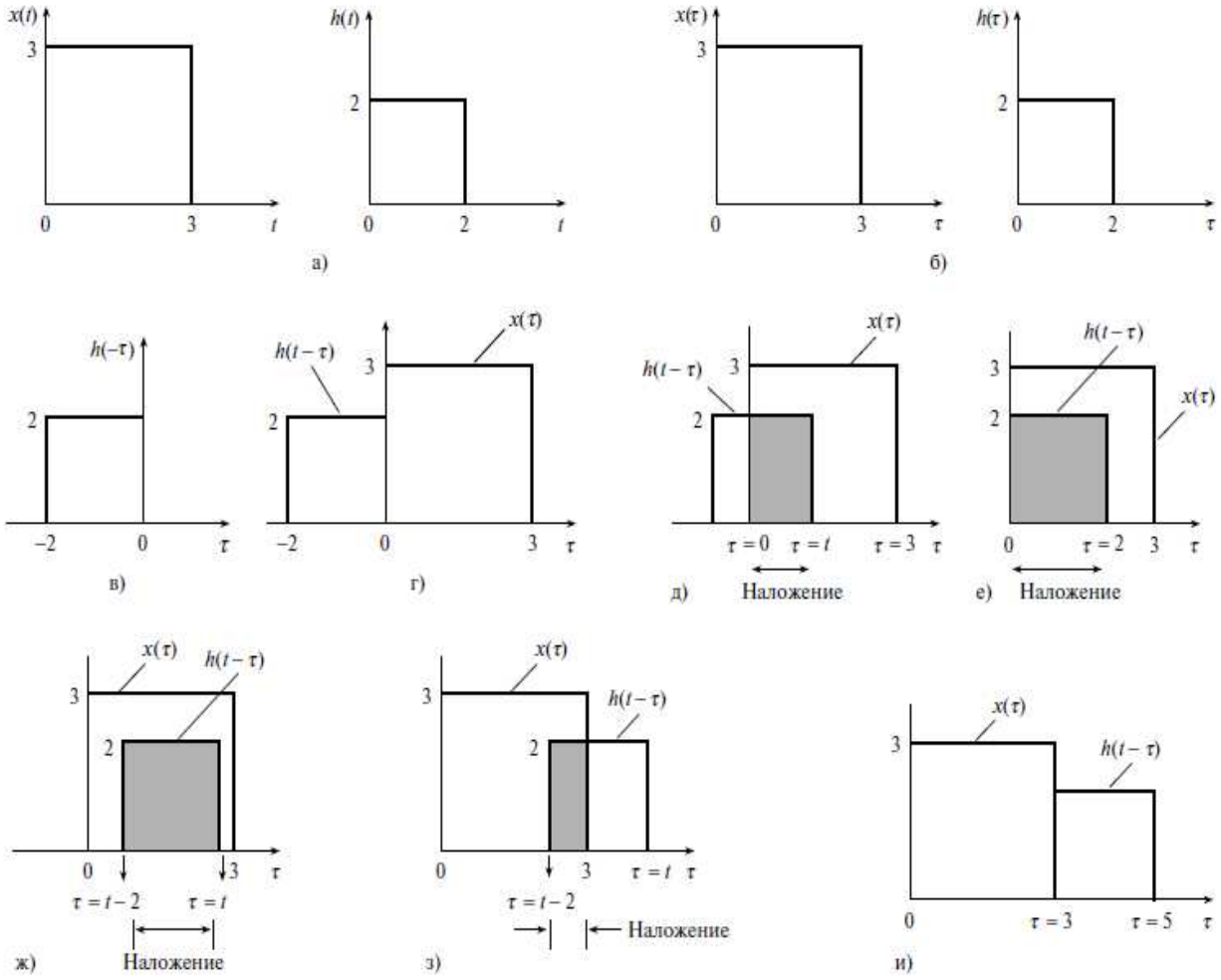


Рис. 2.28. Аналитическая свертка: *а* –сигналы $x(t)$ и $h(t)$; *б* – $x(\tau)$ и $h(\tau)$; *в* – $h(-\tau)$; *г* – $h(t-\tau)$ и $x(\tau)$; $t<0$, сигнал $h(t-\tau)$ не накладывается на $x(\tau)$; *д* – $h(t-\tau)$ и $x(\tau)$; $0<t\le 2$, первое частичное наложение $h(t-\tau)$ и $x(\tau)$; *е* – $h(t-\tau)$ и $x(\tau)$; $t=2$, конец первого частичного наложения; *ж* – $h(t-\tau)$ и $x(\tau)$; $2<t\le 3$, полное наложение $h(t-\tau)$ и $x(\tau)$; *з* – $h(t-\tau)$ и $x(\tau)$; $3<t\le 5$, второе частичное наложение $h(t-\tau)$ и $x(\tau)$; *и* – $h(t-\tau)$ и $x(\tau)$; $t>5$, наложение $h(t-\tau)$ и $x(\tau)$ нет

Далее необходимо обратить во времени сигнал $h(\tau)$, как показано на рис. 2.28, *в*. Далее $h(-\tau)$ смещается относительно $x(\tau)$ в положительном направлении τ. Получающийся в результате сигнал $h(t-\tau)$ накладывается на $x(\tau)$ за пять отдельных этапов, проиллюстрированных на рис. 2.28, *г-з*. Для каждого из этих этапов существует соответствующий интеграл свертки. Следовательно, $x(t) \otimes h(t)$ существует в пяти отдельных непрерывных областях.

Этап 1. $t < 0$, и $h(t-\tau)$ не накладывается на $x(\tau)$ (см. рис. 2.28, *г*). Поскольку функции не перекрываются, $x(\tau)h(t-\tau) = 0$ для всех t и не дает вклада в интеграл свертки.

Этап 2. $0 < t \le 2$, и существует частичное наложение $h(t-\tau)$ и $x(\tau)$ (см. рис. 2.28, *д*). В этой области

$$y(t) = \int_{\tau=0}^{\tau=t} x(\tau)h(t-\tau)d\tau = \int_{\tau=0}^{\tau=t} (3)\cdot(2)d\tau;$$
$$y(t) = 6[\tau]_0^t = 6t, \quad 0 < t \le 2. \tag{2.146}$$

Как показано на рис. 2.28, *е*, данный этап завершается в момент $t = 2$.

Этап 3. $2 < t \le 3$, и сигналы $h(t-\tau)$ и $x(\tau)$ полностью перекрываются (см. рис. 2.28, *ж*). В этой области

$$y(t) = \int_{\tau=t-2}^{\tau} (3)\cdot(2)d\tau = 6[\tau]_{t-2}^t;$$
$$y(t) = 6(t-t+2) = 12, \quad 2 < t \le 3. \tag{2.147}$$

Этап 4. $3 < t \le 5$. Еще одна область частичного наложения, показанная на рис. 2.28, *з*:

$$y(t) = \int_{\tau=t-2}^{\tau=3} (3)\cdot(2)d\tau = 6[\tau]_{t-2}^3 = 6(5-t)3 - 6t. \tag{2.148}$$

Этап 5. $t > 5$. Как показано на рис. 2.28, *и*, это вторая область без наложения, которая также не дает вклада в интеграл свертки.

Таким образом, вклад в интеграл свертки дают этапы 2 – 4, причем эти вклады выражаются следующим образом:

$$0 < t \le 2 \quad y(t) = 6t;$$
$$2 < t \le 3 \quad y(t) = 12;$$
$$3 < t \le 5 \quad y(t) = 30 - 6t.$$

Из этих выражений можно получить зависимость $y(t)$, изображенную на рис. 2.29.

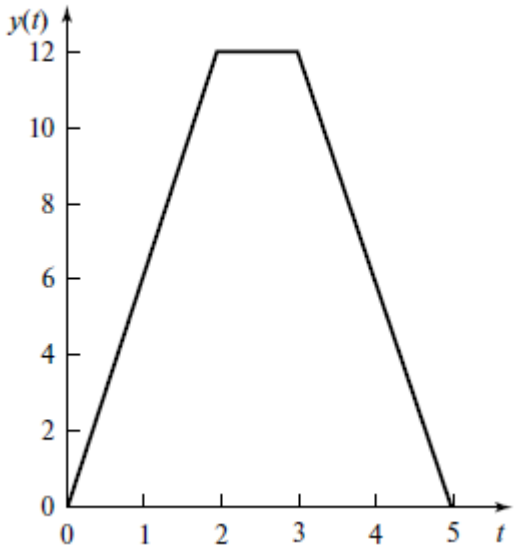


Рис. 2.29. Зависимость свертки $y(t) = x(t) \cdot h(t)$ от t

Приведем повторно уравнения (2.142) и (2.144) и продолжим наше обсуждение:

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = x(n) \otimes h(n)$$

и

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda.$$

Изучая эти уравнения, следует помнить, что свертка выполняется во времени – возникает так называемая *свертка во временной области*. Известно, что в частотной области выход системы на частоте f равен $Y(f)$:

$$Y(f) = H(f)X(f), \qquad (2.149)$$

где $H(f)$ – частотная характеристика системы на частоте f; $X(f)$ – Фурье-образ входа $x(t)$.

Кроме того, можно показать, что $H(f)$ – Фурье-образ $h(t)$. Применяя обратное преобразование Фурье к обеим частям уравнения (2.149), получаем

$$F^{-1}[H(f)] = y(t) = F^{-1}[H(f)X(f)]. \qquad (2.150)$$

Объединяя уравнения (2.144) и (2.150), получаем, что

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda = x(t) \otimes h(t) = F^{-1}[H(f)X(f)]. \qquad (2.151)$$

Таким образом, видно, что свертка двух сигналов во временной области эквивалентна применению обратного преобразования Фурье к про-

изведению Фурье-образов двух сигналов. Данный полезный факт часто формулируют в сокращенной форме: *свертка во временной области эквивалентна умножению в частотной*.

Существует соотношение, дуальное приведенному, т.е. свертка в частотной области эквивалентна умножению во временной. Таким образом, можно показать [14], что

$$Y(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega - u)H(u)du = X(f) \otimes H(f) = \\ = F[y(t)] = F[x(t)h(t)]. \tag{2.152}$$

Следовательно, Фурье-образ произведения двух временных последовательностей соответствует свертке Фурье-образов двух последовательностей. В данной процедуре оцифрованная последовательность данных точка за точкой умножается на другую последовательность, которая состоит из дискретных значений весовой функции. Этот процесс называется *взвешиванием* и выполняется для уменьшения ошибок при вычислении энергетического спектра данных. После этого к взвешенным данным применяется дискретное преобразование Фурье, по результату которого вычисляется энергетический спектр. Задачей является получение энергетического спектра последовательности данных, но из сказанного выше следует, что в действительности получается спектр последовательности данных, свернутый со спектром весовой последовательности.

2.8.6. Свойства свертки

1. *Закон коммутативности:*

$$x_1(t) \otimes x_2(t) = x_2(t) \otimes x_1(t). \tag{2.153}$$

Отметим, что это выражение идентично следующему:

$$\int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)d\tau = \int_{-\infty}^{\infty} x_2(\tau)x_1(t-\tau)d\tau.$$

2. *Закон дистрибутивности:*

$$x_1(t) \otimes [x_2(t) + x_3(t)] = x_1(t) \otimes x_2(t) + x_1(t) \otimes x_3(t). \tag{2.154}$$

3. *Закон ассоциативности:*

$$x_1(t) \otimes [x_2(t) \otimes x_3(t)] = [x_1(t) \otimes x_2(t)] \otimes x_3(t). \tag{2.155}$$

Эти свойства можно доказать, либо расписав соответствующие интегралы, либо рассмотрев свертку как взаимную корреляцию одной последовательности с обращенной во времени другой.

2.8.7. Циклическая свертка

Циклическая свертка определяется для периодических последовательностей длины N выражением

$$y(n)=\sum_{l=0}^{N-1}h(l)s(n-l),\qquad n=0,1,...,N-1. \tag{2.156}$$

В силу периодичности последовательностей номера отсчетов берутся по модулю N, поэтому $s(-n)=s(N-n)$, $h(-n)=h(N-n)$. Матричная запись циклической свертки имеет вид:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ ... \\ y(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(2) & ... & h(N-1) \\ h(1) & h(3) & ... & h(0) \\ h(2) & h(4) & ... & h(1) \\ ... & ... & ... & ... \\ h(N-1) & h(0) & ... & h(N-2) \end{bmatrix} \cdot \begin{bmatrix} s(0) \\ s(N-1) \\ s(N-2) \\ ... \\ s(1) \end{bmatrix}.$$

При вычислениях по модулю N из свойств сравнений следует, что $N\equiv 0$, поэтому в полиномиальном представлении $z^N=1$ и циклическая свертка может рассматриваться как произведение двух полиномов по модулю полинома $z^N=1$:

$$y(z)=h(z)s(z)\bmod(z^N-1). \tag{2.157}$$

Обозначим $h(z)s(z)=w(z)=w(0)+w(1)z+...+w(2N-2)z^{2N-2}$.

Произведение по модулю z^N-1 просто означает, что

$$\begin{gathered} y(0)=w(0)+w(N); \\ y(1)=w(1)+w(N+1); \\ ..... \\ y(N-2)=w(N-2)+w(2N-2); \\ y(N-1)=w(N-1). \end{gathered} \tag{2.158}$$

Таким образом, если положить равными нулю значения $w(N)$, $w(N+1)$, $w(2N-2)$, то линейную свертку можно вычислить через циклическую[15].

Пример 2.18

Для трехточечной циклической свертки с учетом равенства $z^3 = 1$ получим:

$$y(z) = \left(y'(0) + y'(1)z + y'(2)z^2 + y'(3)z^3 + y'(4)z^4 \right) \bmod (z^3 - 1) =$$

$$= \left(y'(0) + y'(3) \right) + \left(y'(1) + y'(4) \right) z + y'(2)z^2.$$

Используя соотношение (2.155), запишем коэффициенты циклической свертки:

$$y(0) = h(0)s(0) + h(2)s(1) + h(1)s(2);$$
$$y(1) = h(1)s(0) + h(0)s(1) + h(2)s(2);$$
$$y(2) = h(2)s(0) + h(1)s(1) + h(0)s(2)$$

Эти равенства можно получить и с помощью матричного соотношения.

2.8.8. Диадная свертка

Диадная свертка определяется выражением

$$y(n) = \sum_{l=0}^{N-1} h(l)s(n \oplus l), n = 0,1,..., N-1. \qquad (2.159)$$

Здесь номера отсчетов представляются в двоичной системе счисления, арифметические операции над числами выполняются посимвольно по модулю два, а длина сворачиваемых последовательностей равна $N = 2^v$, v – целое число. При этом сдвиг функции фактически заключается в перестановке отсчетов. Таким образом, коэффициентами свертки являются суммы произведений $h(l)s(n \oplus l)$, полученные при различных перестановках функций $s(l)$.

2.8.9. Быстрая линейная свертка

В п. 2.9.4 было показано, что вычисление свертки можно ускорить с помощью теоремы о корреляции.

Существует также подобная *теорема о свертке*. Итак, используя дискретную терминологию и временную область, можно записать:

$$x_1(l) \otimes x_2(r) = F_D^{-1}[X_1(k)X_2(k)]. \qquad (2.160)$$

Уравнение (2.160) является формулировкой теоремы о свертке, где F_D^{-1} обозначает обратное дискретное преобразование Фурье, $X_1(l)$ – ДПФ-образ $x_1(l)$, $X_2(k)$ – ДПФ-образ $x_2(r)$. Как и в п. 2.9.4, $x_1(l)$ и $x_2(r)$ – периодические последовательности длины *N*.

2.8.10. Связь между сверткой и корреляцией

При свертке значение n-го выхода определяется сверточной суммой, представленной в формуле (2.140):

$$y(n) = \sum_{m=0}^{n} h(m)x(n-m) = h(0)x(n) + h(1)x(n-1) + \ldots + h(n)x(0). \quad (2.161)$$

Значение функции взаимной корреляции для сигналов $h(n)$ и $x(n)$ для i-й задержки представлено в формуле (2.110), которая в несколько модифицированном виде выглядит так:

$$\begin{aligned} r_{hx}(j) &= \frac{1}{N}\sum_{n=0}^{N-1} h(n)x(j+n) = \\ &= \frac{1}{N}[h(0)x(j) + h(1)x(j+1) + \ldots + h(N-1)x(j+N-1)]. \end{aligned} \quad (2.162)$$

Сравнить $y(n)$ и $r_{hx}(j)$ проще всего, если j = 0, т.е. при нулевой задержке. В этом случае уравнение (2.162) приводится к такому виду:

$$\begin{aligned} r_{hx}(0) &= \frac{1}{N}\sum_{n=0}^{N-1} h(n)x(n) = \\ &= \frac{1}{N}[h(0)x(0) + h(1)x(1) + \ldots + h(N-1)x(N-1)]. \end{aligned} \quad (2.163)$$

Сравнивая уравнения (2.161) и (2.163), находим, что они имеют сходную форму, за исключением того, что последовательность $x(n)$ в функции взаимной корреляции идет в обратном порядке по сравнению с тем, как она идет в свертке. Следовательно, свертка эквивалентна функции взаимной корреляции двух сигналов, в которой одна из исходных последовательностей обращена во времени, а нормировочный коэффициент $1/N$ равен единице. Это означает, что свертки и корреляции можно вычислять с помощью одной компьютерной программы, просто обращая одну из последовательностей.

Вопросы и задания для самопроверки

1. Запишите выражение для тригонометрической формы ряда Фурье.
2. На основе каких выражений можно определить амплитуду и фазу n-й гармоники?
3. Запишите выражение для системы ДЭФ.
4. Представьте матрицу ДЭФ для $N = 5$ с минимальными фазами.
5. Сформулируйте и запишите свойства ДЭФ.
6. Графически представьте поворачивающий множитель для $N = 4$ ($N = 6$, $N = 8$).
7. Запишите пару ДПФ в показательной (матричной) форме.
8. Докажите, что Фурье-базис является ортогональным.
9. Запишите свойства ДПФ.
10. Сформулируйте алгоритм вычисления свертки (корреляции) с использованием ДПФ.
11. Постройте матрицу ДЭФ для выполнения обратного преобразования Фурье для сигнала длиной $N = 6$.
12. Назначение и особенности алгоритмов БПФ.
13. Какую операцию алгоритма БПФ называют базовой?
14. Что понимается под числом этапов прореживания БПФ?
15. Как определяется вычислительная сложность алгоритма БПФ с прореживанием по времени (частоте) по основанию 2?
16. Каков порядок следования входных и выводных отсчетов в алгоритме БПФ с прореживанием по времени (частоте)?
17. Дайте определение импульсной характеристике линейной дискретной системы. Как определить импульсную характеристику системы?
18. Дайте определение переходной характеристике линейной дискретной системы.
19. Что такое свертка и для каких целей она используется в ЦОС? Какие виды свертки вы можете назвать и в чем их отличие?
20. Вычислить линейную свертку двух сигналов: $s(n) = (10,10,8,6,8,10,10)$ и $h(n) = (-2,5,-2)$.
21. Что такое корреляция и как она может вычисляться?
22. Докажите, что функции Уолша образуют ортонормированную систему.
23. Дайте определение матрице Адамара и запишите способы построения матриц Адамара.
24. Сформулируйте и запишите основные свойства функций Уолша.
25. По какому правилу образуется диадный сдвиг функции?
26. Запишите пару дискретного преобразования Хаара и сформулируйте его особенности по сравнению с ДПФ.
27. В чем заключаются сущность и особенности вейвлет-преобразования сигналов?
28. Для каких целей используется Z-преобразование и каким образом оно определяется?
29. Приведите пример прямого и обратного Z-преобразования сигнала.

МОДУЛЬ 3. БАЗОВЫЕ МОДЕЛИ СИСТЕМ ЦОС

3.1. Методы и модели ЦОС

Основными направлениями использования методов цифровой обработки являются цифровая фильтрация и спектральный анализ. К цифровым фильтрам относятся КИХ-фильтры и БИХ-фильтры. Спектральный анализ можно проводить путем вычисления спектров с помощью дискретного преобразования Фурье (ДПФ) или путем вычисления спектров с применением статистических методов, например, при анализе случайных сигналов (в частности, шума квантования, возникающего в цифровых системах). На практике при спектральном анализе, как правило, используются быстрое преобразование Фурье (БПФ) и основанная на нем методика вычисления быстрой свертки.

3.1.1. Разностные уравнения и метод пространства состояний

Системы, у которых входная и выходная последовательности $x(n)$ и $y(n)$ связаны линейным разностным уравнением с постоянными коэффициентами, образуют подмножество класса линейных систем с постоянными параметрами. Описание ЛПП-систем разностными уравнениями очень важно, так как оно часто позволяет найти эффективные способы построения таких систем. Более того, по разностному уравнению можно определить многие характеристики рассматриваемой системы, включая собственные частоты, соответствующие нулевому коэффициенту передачи, и т.д. [16].

В самом общем случае линейное разностное уравнение M-го порядка с постоянными коэффициентами относится к физически реализуемой системе, имеет вид:

$$y(n) = \sum_{i=0}^{M} b_i x(n-i) - \sum_{i=0}^{M} a_i y(n-i), n \geq 0, \qquad (3.1)$$

где коэффициенты $\{a_i\}$ и $\{b_i\}$ описывают конкретную систему, причем $a_M \neq 0$. Каким именно образом порядок системы M характеризует математические свойства разностного уравнения, будет показано ниже. Уравнение (3.1) записано в виде, удобном для решения методом прямой подстановки. Имея набор начальных условий (например, $x(i)$, $y(i)$ для $i = -1, -2, -M$) и входную последовательность $x(n)$, по формуле (3.1) можно непосредственно вычислить выходную последовательность $y(n)$ для $n \geq 0$.

Например, разностное уравнение

$$y(n) = x(n) - 3y(n-1) \qquad (3.2)$$

с начальным условием $y(-1) = 0$ и $x(n) = n^2 + n$ можно решить подстановкой, что дает:

$$\begin{aligned}
&y(0) = x(0) - 3y(-1) = 0;\\
&y(1) = x(1) - 3y(0) = 2;\\
&y(2) = x(2) - 3y(1) = 0;\\
&y(3) = x(3) - 3y(2) = 12;\\
&y(4) = x(4) - 3y(3) = -16;\\
&y(5) = x(5) - 3y(4) = 78;\\
&y(6) = x(6) - 3y(5) = 100;\\
&y(7) = x(7) - 3y(6) = -192.\\
&\quad\vdots \qquad\quad \vdots \qquad\quad \vdots
\end{aligned}$$

Хотя решение разностных уравнений прямой подстановкой и целесообразно в некоторых случаях, значительно полезнее получить решение уравнения в явном виде. Основная идея сводится к получению двух решений разностного уравнения – однородного и частного. Однородное решение получается путем подстановки нулей вместо всех членов, содержащих элементы входной последовательности $x(n)$, и определения отклика при нулевой входной последовательности. Именно этот класс решений описывает основные свойства заданной системы. Частное решение получают, подбирая вид последовательности $y(n)$ на выходе при заданной входной последовательности $x(n)$. Для определения произвольных постоянных однородного решения используются начальные условия. В качестве примера решим этим методом уравнение (3.2). Однородное уравнение имеет вид

$$y(n) + 3y(n-1) = 0. \qquad (3.3)$$

Известно, что характеристическими решениями однородных уравнений, соответствующих линейным разностным уравнениям с постоянными коэффициентами, являются решения вида $A\alpha^n$. Поэтому, подставив в уравнение (3.3) $A\alpha^n$ вместо $y(n)$, получим:

$$\begin{aligned}
&A\alpha^n + 3A\alpha^{n-1} = 0;\\
&A\alpha^{n-1}(\alpha + 3) = 0;\\
&\alpha = -3;\\
&y_0(n) = A(-3)^n.
\end{aligned} \qquad (3.4)$$

Частное решение, соответствующее входной последовательности $x(n) = n^2 + n$, попробуем найти в виде

$$y_ч(n) = Bn^2 + Cn + D. \tag{3.5}$$

Из уравнения (3.2) получаем:

$$Bn^2 + Cn + D + 3B(n-1)^2 + 3C(n-1) + 3D = n^2 + n. \tag{3.6}$$

Поскольку коэффициенты при равных степенях n должны совпадать, B, C, D должны быть равны

$$B = \frac{1}{4},\ C = \frac{5}{8},\ D = \frac{9}{32}. \tag{3.7}$$

Таким образом, общее решение имеет вид:

$$y(n) = \frac{n^2}{4} + \frac{5n}{8} + \frac{9}{12} + A(-3)^n, \tag{3.8}$$

коэффициент A определяется из начального условия $y(n-1) = 0$, откуда $A = -\frac{9}{12}$, тогда

$$y(n) = \frac{n^2}{4} + \frac{5n}{8} + \frac{9}{12}[1-(-3)^n]. \tag{3.9}$$

Выборочная проверка решения (3.9) при $n \neq 0$ показывает полное его совпадение с приведенным решением. Очевидное преимущество решения (3.9) состоит в том, что оно позволяет весьма просто определить $y(n)$ для любого конкретного $n = n_0$.

Важное значение разностных уравнений состоит в том, что они непосредственно определяют способ построения цифровой системы. Так, разностное уравнение первого порядка самого общего вида

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

можно реализовать с помощью схемы, изображенной на рис. 3.1.

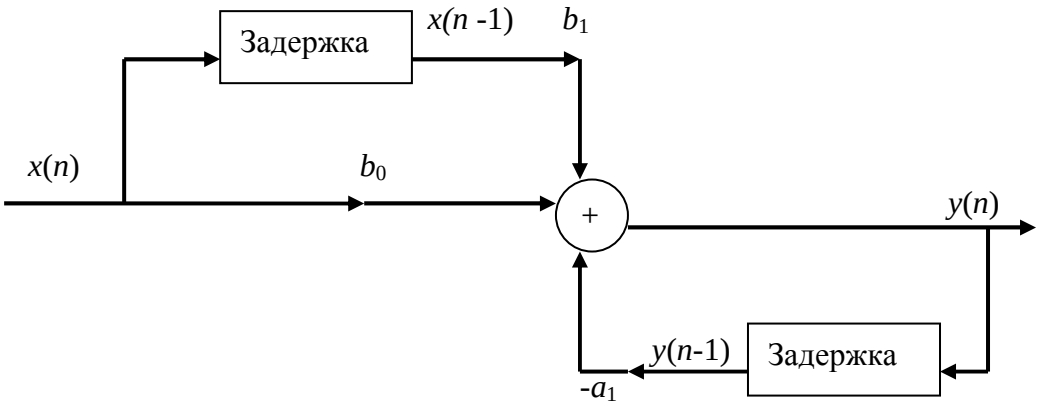


Рис. 3.1. Схема реализации простого разностного уравнения первого порядка

Блок «задержка» осуществляет задержку на один отсчет. Рассмотренная форма построения системы, в которой для входной и выходной последовательностей используются раздельные элементы задержки, называется прямой формой 1.

3.1.2. Оператор сдвига

Пусть a и b – это целые положительные числа, данные на интервале $N=m^2$, где m и n – также целые положительные числа, причем $2 \le m \le N$. Числа a и b можно представить в m-ичной системе счисления в виде

$$a = (a_1 a_2 ... am)m = \sum_{i=1}^{n} a_i m^{n-i};$$

$$b = (b_1 b_2 ... bm)m = \sum_{i=1}^{n} b_i m^{n-i}.$$

Поразрядной суммой чисел a и b по модулю m, которую обозначим $a \underset{m}{\oplus} b$, называется число

$$c = a \underset{m}{\oplus} b = (c_1 c_2 ... c_n)_m = \sum_{i=1}^{n} c_i m^{n-1}, \qquad (3.10)$$

где

$$c_i = ((a_i + b_i))_m.$$

Введем m-ичное противоположное число b^*, удовлетворяющее условию

$$b \underset{m}{\oplus} b^* = 0. \qquad (3.11)$$

Очевидно, что это условие выполняется, если для каждого разряда чисел b и b^* справедливо одно из следующих равносильных соотношений:

$$((b_i + b_i^*))_m = 0, b_i^* = ((m - b_i))_m;$$

$$b_i^* = \begin{cases} m - b_i, b_i \neq 0; \\ 0, b_i = 0. \end{cases} \qquad (3.12)$$

Тогда поразрядной разностью чисел a и b по модулю m будет положительное число:

$$d = a \underset{m}{\oplus} b_i^*. \qquad (3.13)$$

Другими словами, поразрядное вычитание по модулю m – это то же самое, что и поразрядное сложение с m-ично противоположным числом по тому же модулю [6].

При $m = 2$ каждая точка числовой оси m-ично противоположна самой себе, а при $m=N$ m-ично противоположные точки равноудалены от концов интервала ($x^* = N - x$).

Поразрядная сумма по модулю m образуется путем поразрядного суммирования без переноса единицы в старший разряд. Понятно, что при таком способе образования сумма c или разность d всегда будут принадлежать тому же интервалу N, что и слагаемые a и b.

В дальнейшем для краткости такие операции, связанные с таким способом сложения или вычитания, будем называть m-сдвигом.

Для выяснения сущности m-сдвига рассмотрим конкретный пример. Пусть $m = 5$, $n = 2$, $N = m^n = 25$, $a = 0,1,2,\ldots,24$ и $b = 13$. Имея в виду, что число 13 в пятеричной системе записывается как $(b_1\ b_2)_5 = (23)_5$, получим для $d = a \underset{m}{\oplus} b$ результат, изображенный на рис. 3.2. Из него видно, что m-сдвиг массива чисел $\{a\}$ на b единиц сводится к перестановке чисел.

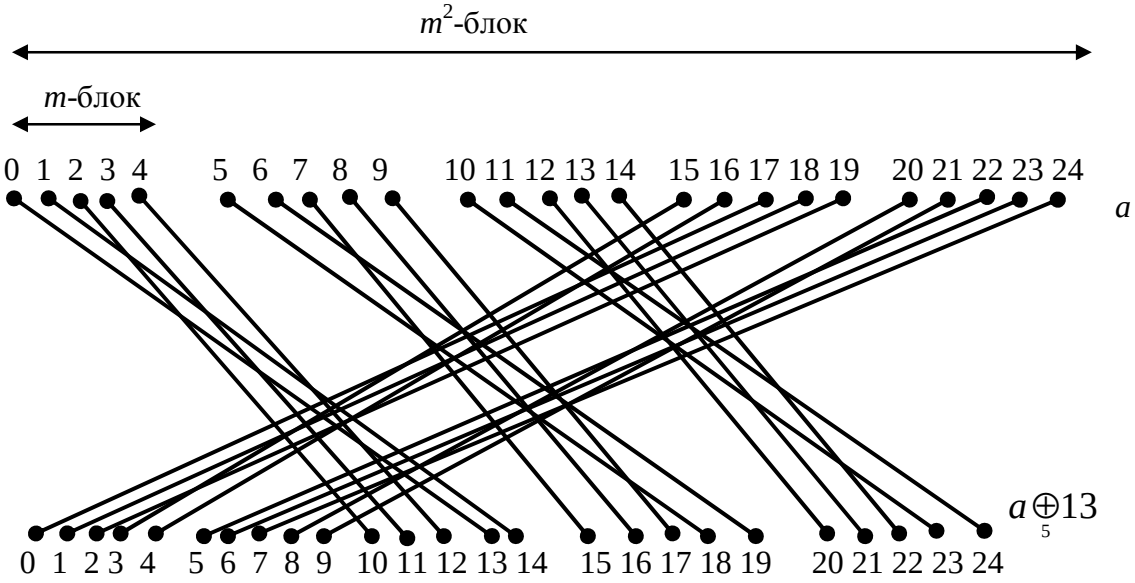


Рис. 3.2. *M*-сдвиг последовательности чисел

Для того чтобы пояснить механизм такой перестановки, разобьем интервал N определения массива $\{a\}$ сначала на самые мелкие подынтервалы по m чисел в каждом, затем разобьем тот же интервал N на более крупные подынтервалы по m^2 чисел в каждом, на еще более крупные подынтервалы по m^3 чисел в каждом и т.д. Для простоты назовем все эти по-

дынтервалы соответственно m-блоками, m^2-блоками и т.д. Очевидно, что их число на интервале N соответственно будет составлять m^{m-1}, m^{m-2}, ..., m, 1.

Тогда перестановка чисел при m-сдвиге $d = a \underset{m}{\oplus} b$ сводится к следующему:

1) внутри каждого m-блока производится круговая перестановка чисел путем сдвига на b_n единиц;

2) внутри каждого m^2-блока производится круговая перестановка m-блоков путем сдвига на $b_n - 1$ m-блоков;

3) внутри каждого m^3-блока производится круговая перестановка m^2-блоков путем сдвига на b_{n-2} m^2-блоков и т.д.

Этот процесс продолжается n раз, пока не будут использованы все разрядные коэффициенты m-ичного представления числа b. В общем случае будет произведено n круговых перестановок в блоках, содержащих m, m^2,.., $m^n = N$ чисел так, что m-сдвиг – это как бы многопетлевая круговая перестановка на интервале из N чисел.

Заметим, что m-сдвиг массива из N чисел вырождается при $m = N$ в однократную круговую перестановку. В другом частном случае, когда $m = 2$ и $N=2^n$, для m-сдвига справедливо выражение:

$$c = a \underset{2}{\oplus} b = a + b - 2\sum_{i=1}^{n} a_i b_i 2^{n-i} . \qquad (3.14)$$

Введенное понятие m-сдвига массива чисел требует изменения некоторых привычных представлений. Запаздывание дискретной функции $f(x)$ на τ единиц необходимо теперь рассматривать как ее m-сдвиг на τ точек, т.е. как образование функции $f(x \underset{m}{\oplus} \tau)$ путем указанной выше перестановки отсчетов исходной функции.

Симметричными относительно точки x будут отсчеты $f(x \underset{m}{\Theta} \tau)$ и $f(x \underset{m}{\oplus} \tau)$, а соседними с отсчетом $f(x)$ – отсчеты $f(x \underset{m}{\Theta} 1)$ и $f(x \underset{m}{\oplus} 1)$ и т.д.

Меняются также привычные представления о четности дискретных функций. Эти представления теперь необходимо связывать с понятием m-ично противоположных чисел. Определим четную и нечетную дискретные функции в смысле m-сдвига как функции, обладающие, соответственно, свойствами:

$$f_ч(x^*) = f_ч(x),\ f_н(x^*) = -f_ч(x). \qquad (3.15)$$

Такие функции для краткости будем называть m-четной и m-нечетной. Следовательно, если $m = 2$, то любая дискретная функция является m-четной, а в случае $m = N$ m-четная и m-нечетная функции на интервале N имеют обычный смысл.

В заключение приведем сводку основных свойств операции m-сдвига (их доказательство элементарно, поэтому оно не приводится):

1) $x \underset{m}{\oplus} y = y \underset{m}{\oplus} x$ (коммутативность);

2) $(x \underset{m}{\oplus} y) \underset{m}{\oplus} z = x \underset{m}{\oplus} (y \underset{m}{\oplus} z)$ (ассоциативность);

3) $x \underset{m}{\Theta} y = x \underset{m}{\oplus} y^* = y^* \underset{m}{\Theta} x^*$.

В некоторых случаях в одних и тех же выражениях наряду с m-сдвигом встречаются обычные сложение и умножение. Следует помнить, что в общем случае

$$a(x \underset{m}{\oplus} y) \neq ax \underset{m}{\oplus} ay, (x \underset{m}{\oplus} y) + a \neq x \underset{m}{\oplus} (y + a).$$

Например, если $3(5 \underset{4}{\oplus} 7) = 24$, то $15 \underset{4}{\oplus} 21 = 16$, и если $(5 \underset{4}{\oplus} 7) + 3 = 11$, то $5 \underset{4}{\oplus} (7 + 3) = 15$.

3.2. Математическое описание и типы ЦФ

3.2.1. Понятие цифрового фильтра и его математическое описание

Цифровая фильтрация является одним из наиболее мощных инструментальных средств ЦОС. Кроме очевидных преимуществ устранения ошибок в фильтре, связанных с флуктуациями параметров пассивных компонентов во времени и по температуре, дрейфом ОУ (в активных фильтрах) и т.д., цифровые фильтры способны удовлетворять таким техническим требованиям по своим параметрам, которых было бы чрезвычайно трудно или даже невозможно достичь в аналоговом исполнении. Кроме того, характеристики цифрового фильтра могут быть легко изменены программно. Поэтому они широко используются в телекоммуникациях, в приложениях адаптивной фильтрации, таких как подавление эха в модемах, подавление шума и распознавание речи.

В отличие от аналоговых фильтров в цифровых используются элементы, которые выполняют функции, отличающиеся от функций резисторов, индуктивностей и емкостей. К таким элементам относятся умножители, сумматоры и элементы задержки, характеризующиеся зависимостью между

входными и выходными параметрами, а не соотношением между током и напряжением. Элементы цифрового фильтра представлены на рис. 3.3.

Функционирование ЦФ описывается разностным уравнением [6]

$$y(nT) = \sum_{i=0}^{N} a_i x(nT - iT) - \sum_{i=0}^{M} b_i y(nT - iT), \qquad (3.16)$$

где $y(nT)$ – отсчеты выходного сигнала; $x(nT)$ – отсчеты входного сигнала; n – порядковый номер отсчета; a_i, b_i – коэффициенты цифрового фильтра; i – порядковый номер коэффициентов фильтра; N, M – максимальное значение порядковых номеров коэффициентов a_i и b_i соответственно; T – интервал дискретизации.

Однако поскольку разностные уравнения не позволяют судить о передаточной характеристике ЦФ, для ее определения применяют Z-преобразование разностного уравнения. Цифровая фильтрация сигналов в основном применяется в четырех видах систем:

- системы локального типа – предназначены для предварительной обработки сигналов с целью улучшения отношения сигнал/помеха, сжатия данных, выделения определенного диапазона частот и т.п.;
- системы обработки речевых сигналов;
- системы управления динамическими испытаниями – служат для генерирования на управляющих входах стендов процессов, которые по своему воздействию на объект идентичны эксплуатационным;
- системы обработки сигналов с элементами приема/передачи.

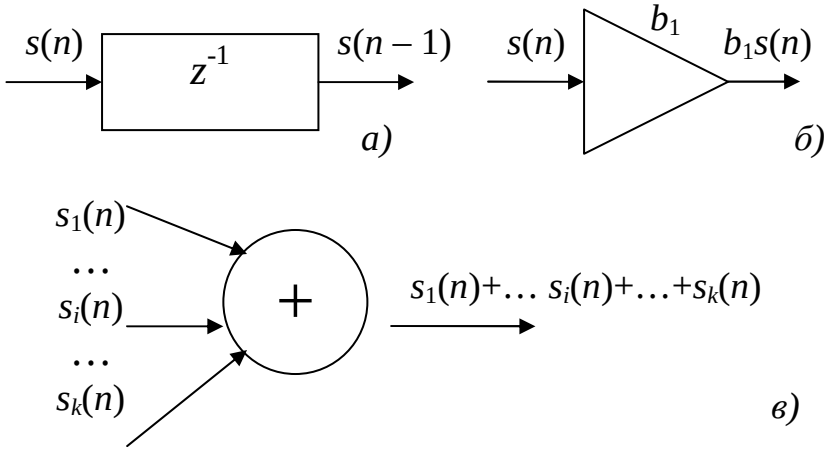


Рис. 3.3. Элементы цифрового фильтра:
а – элемент задержки; *б* – умножитель; *в* – сумматор

Цифровой фильтр, соотношение вход/выход которого описывается в виде разностного уравнения (3.16), отвечает условиям физической реализуемости: при нулевых начальных условиях реакция не может возникнуть раньше воздействия. Значения реакции $y(n)$ в каждый момент времени n зависят от текущего и предшествующих значений воздействия, но не зависят от его последующих значений.

Однако поскольку разностные уравнения не позволяют судить о передаточной характеристике ЦФ, для ее определения применяют Z-преобразование разностного уравнения.

При этом входной и выходной сигнал являются цифровыми, так что в устройстве циркулируют только двоичные коды. Поскольку операция умножения отсчетов цифрового сигнала на число иногда выполняется неточно, за счет округления или усечения произведений, в общем случае цифровое устройство не точно реализует алгоритм (3.16) и выходной сигнал отличается от точного решения (3.16). Однако в ЦФ погрешность выходного сигнала не зависит от условий, при которых работает фильтр, – температуры, влажности и т.п. Кроме того, эта погрешность контролируема – ее можно уменьшить, увеличивая число разрядов, используемых для представления отсчетов цифровых сигналов. Именно этим и определяются основные преимущества ЦФ – высокая точность обработки сигналов и стабильность характеристик по сравнению с аналоговыми и дискретными фильтрами [7].

На практике чаще всего берут $N = M = N_0$. Тогда уравнение (3.16) имеет вид:

$$y(nT) = \sum_{i=0}^{N_0} a_i x(nT - iT) - \sum_{i=0}^{N_0} b_i y(nT - iT). \qquad (3.17)$$

В том случае, когда известны коэффициенты a_i, и b_i, отсчеты входного сигнала $x(nT)$ при $n \geq 0$ и начальные условия, используя уравнения (3.16), (3.17), можно определить $y(nT)$ для любого $n \geq 0$. За начальные условия принимаются значения x и y при отрицательных значениях времени, обычно x и y принимаются равными нулю, т.е.

$$x(-T), x(-2T), ..., x(-N_0T) = 0;$$
$$y(-T), y(-2T), ..., y(-N_0T) = 0.$$

Если все коэффициенты b_i в уравнениях (3.16), (3.17) равны нулю, фильтр называется нерекурсивным, в противном случае – рекурсивным.

В зависимости от значений коэффициентов a_i и b_i различают цифровые фильтры следующих типов: полосовой (ЦПФ), нижних частот (ЦФНЧ), верхних частот (ЦФВЧ) и режекторный (ЦРФ).

Разностное уравнение (3.17) не позволяет судить о передаточной характеристике фильтра. Для получения передаточной характеристики фильтра $H(jw)$, где $j=\sqrt{-1}$; w – круговая частота, необходимо сначала по его коэффициентам найти данную характеристику на Z-плоскости, а затем заменить в ней оператор Z^{-l} на e^{-jwT}. Очевидно, что такая замена невозможна без Z-преобразования. Поэтому остановимся на нем и некоторых его свойствах, позволяющих получить из разностного уравнения (3.17) передаточную характеристику фильтра на Z-плоскости.

Для последовательности $x(nT)$, заданной при всех n, Z-преобразование имеет вид:

$$x(Z)=\sum_{n=-\infty}^{\infty} x(nT)Z^{-n}, \tag{3.18}$$

где Z – комплексная переменная.

Передаточной функцией $H(z)$ называют отношение Z-образов выходного $Y(z)$ и входного $X(z)$ сигналов фильтра при нулевых начальных условиях: $H(z)=Y(z)/X(z)$. Для рекурсивного и нерекурсивного фильтров

$$H_p(x)=\frac{\sum_{i=0}^{N-1} b_i z^{-1}}{1+\sum_{j=0}^{M-1} a_j z^{-1}}; \tag{3.19}$$

$$H_н(x)=\sum_{i=0}^{N-1} b_i z^{-1}. \tag{3.20}$$

Коэффициенты фильтров являются коэффициентами соответствующих передаточных функций.

Цифровой фильтр, работающий в реальном масштабе времени, оперирует с дискретными по времени данными в противоположность непрерывному сигналу, обрабатываемому аналоговым фильтром. При этом очередной отсчет, соответствующий отклику фильтра, формируется по окончании каждого периода дискретизации. Вследствие дискретной природы обрабатываемого сигнала на отсчеты данных зачастую ссылаются по их номерам, например, отсчет 1, отсчет 2, отсчет 3 и т.д.

На рис. 3.4 представлен низкочастотный сигнал, содержащий высокочастотный шум, который должен быть отфильтрован.

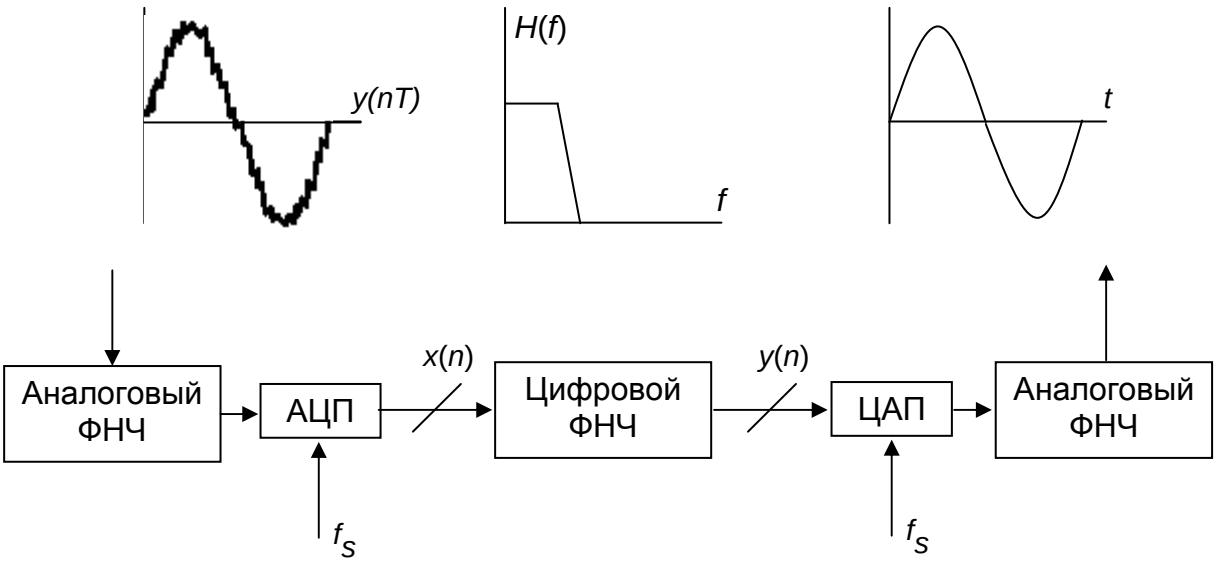


Рис. 3.4. Цифровая фильтрация

Вначале сигнал должен быть оцифрован с помощью АЦП для получения выборки $x(n)$. Далее эта выборка поступает на цифровой фильтр, который в данном случае является НЧ-фильтром. Отсчеты выходных данных $y(n)$ используются для восстановления аналогового сигнала с использованием ЦАП с низким уровнем ложного сигнала.

3.2.2. Рекурсивные и нерекурсивные ЦФ

Цифровой фильтр называется рекурсивным, если хотя бы один из коэффициентов $a(k), k = 1,2,...M-1$ разностного уравнения (3.16) не равен нулю [6].

Порядок рекурсивного ЦФ (РЦФ) определяется как

$$\max\{(M-1),(N-1)\}.$$

Согласно (3.16) реакция $y(n)$ РЦФ в каждый момент времени n определяется: текущим отсчетом воздействия $s(n)$; предысторией воздействия $s(n-k), k = 1,2,..., N-1$; предысторией реакции $y(n-k), k = 1,2,..., M-1$.

Примеры разностных уравнений РЦФ:

– первого порядка

$$y(n) = b_0 s(n) + b_1 s(n-1) - a_1 y(n-1),$$

– второго порядка

$$y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2) - a_1 y(n-1) - a_2 y(n-2).$$

Цифровой фильтр называется нерекурсивным (НЦФ), если все коэффициенты $a(k)$ разностного уравнения (3.16) равны нулю. Для НЦФ разностное уравнение принимает вид

$$y(n) = \sum_{k=0}^{N-1} b(k)s(n-k). \qquad (3.21)$$

Порядок НЦФ определяется как (N – 1).

Согласно (3.21) реакция $y(n)$ НЦФ в каждый момент времени n определяется: текущим отсчетом воздействия $s(n)$; предысторией воздействия $s(n-k), k = 1, 2, ..., N-1$.

Пример разностного уравнения НЦФ второго порядка:

$$y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2). \qquad (3.22)$$

Для РЦФ и НЦФ определим импульсную характеристику по разностному уравнению и сравним результаты на примерах.

Для вычисления импульсной характеристики НЦФ второго порядка подадим на вход цифровой единичный импульс, соотношение вход/выход которого описывается разностным уравнением (3.22).

Решим уравнение (3.22) методом подстановки при нулевых начальных условиях:

$$y(0) = b_0 s(0) + b_1 s(-1) + b_2 s(-2) = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 0 = b_0;$$
$$y(1) = b_0 s(1) + b_1 s(0) + b_2 s(-1) = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 = b_1;$$
$$y(2) = b_0 s(2) + b_1 s(1) + b_2 s(0) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 1 = b_2;$$
$$y(3) = b_0 s(3) + b_1 s(2) + b_2 s(1) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 = 0.$$

Очевидно, что $y(n) = 0$ при $n > 2$.

Таким образом, обобщая полученные результаты на НЦФ произвольного порядка, можно сделать выводы, что импульсная характеристика НЦФ имеет конечную длительность; значения отсчетов импульсной характеристики равны коэффициентам разностного уравнения.

Поэтому НЦФ называют фильтрами с конечной импульсной характеристикой (КИХ-фильтрами).

Теперь вычислим импульсную характеристику РЦФ первого порядка, соотношение вход/выход которой описывается разностным уравнением

$$y(n) = b_0 s(n) + b_1 s(n-1) - a_1 y(n-1);$$

при $b_1 = 0$ получим:

$$y(n) = b_0 x(n) - a_1 y(n-1).$$

Для данного уравнения методом прямой подстановки при нулевых начальных условиях имеем:

$$y(0) = b_0 s(0) - a_1 s(-1) = b_0;$$
$$y(1) = b_0 s(1) - a_1 y(0) = -a_1 b_0;$$
$$y(2) = b_0 s(2) - a_1 y(1) = -a_1(-a_1 b_0) = a_1^{\,2} b_0;$$
$$y(3) = b_0 s(3) - a_1 y(2) = -a_1(a_1^{\,2} b_0) = -a_1^{\,3} b_0;$$
$$\ldots$$

Анализируя выражения, можно записать общую формулу вычисления импульсной характеристики для данного случая:

$$y(n) = (-1)^n a_1^{\,n} b_0, n = 4,5\ldots$$

Таким образом, очевидно, что импульсная характеристика РЦФ имеет бесконечную длительность. Поэтому РЦФ называют системами с бесконечной импульсной характеристикой (БИХ-системами). Однако РЦФ могут иметь и конечную импульсную характеристику. Вид импульсной характеристики БИХ- и КИХ-фильтра представлены на рис. 3.5.

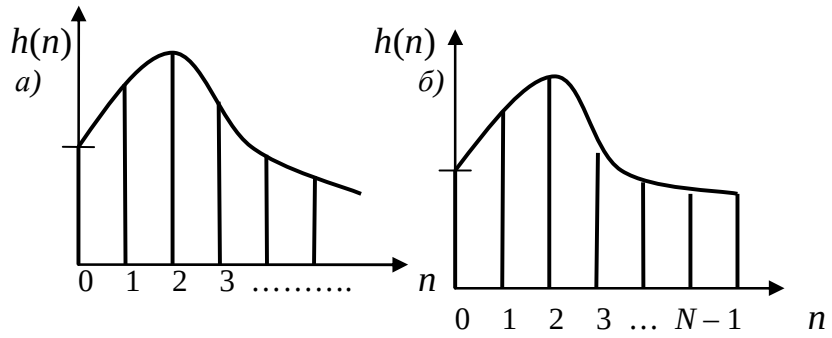


Рис. 3.5. Виды импульсных характеристик: *а* – БИХ; *б* – КИХ

По импульсной характеристике можно судить об устойчивости РЦФ. Устойчивому РЦФ отвечает затухающая со временем импульсная характеристика, что математически выражается как

$$\sum_{n=0}^{\infty} |h(n)| < \infty. \qquad (3.23)$$

3.2.3. Передаточная функция

Передаточной функцией $H(z)$ ЦФ называется отношение Z-преобразования выходной последовательности к Z-преобразованию входной последовательности при нулевых начальных условиях [6]:

$$H(z)=\frac{Z\{y(n)\}}{Z\{x(n)\}}=\frac{Y(z)}{X(z)}. \qquad (3.24)$$

Таким образом, передаточная функция ЦФ может быть получена путем применения Z-преобразования к разностным уравнениям. Для РЦФ передаточная функция имеет вид:

$$H(z)=\frac{\sum_{k=0}^{N-1}b_k z^{-k}}{1+\sum_{k=1}^{M-1}a_k z^{-k}}. \qquad (3.25)$$

Для НРЦ передаточная функция имеет вид:

$$H(z)=\sum_{k=0}^{N-1}b_k z^{-k}. \qquad (3.26)$$

Анализ представленных выражений и разностных уравнений показывает, что:

– коэффициенты разностного уравнения являются коэффициентами передаточной функции;

– коэффициенты разностного уравнения $b(k)$ при $s(n-k)$ равны коэффициентам числителя передаточной функции при z^{-k}, при $k=0,1,...,N-1$;

– коэффициенты разностного уравнения $-a(k)$ при $y(n-k)$ равны коэффициентам знаменателя передаточной функции (с обратным знаком) при z^{-k}, при $k=1,...,M-1$.

Пример 3.1

Известна передаточная функция РЦФ

$$H(z)=\frac{0,5z^{-1}-0,3z^{-2}}{1+0,7z^{-1}-0,55z^{-2}}.$$

Разностное уравнение для данного РЦФ имеет вид:

$$y(n)=0,5s(n)-0,3s(n-1)-0,7y(n-1)+0,55y(n-2).$$

3.3. Структуры ЦФ

3.3.1. Основные структуры РЦФ

Цифровой фильтр может быть реализован аппаратно или программно. В первом случае – в виде специализированного цифрового устройства на интегральных логических элементах, во втором – в виде программы на компьютере или цифровом процессоре обработки сигналов. В последнее время программная реализация является преобладающей.

Для РЦФ определяют три основные структуры [6]: прямую; каскадную; параллельную.

3.3.2. Прямая структура

Прямая структура определяется передаточной функцией $H(z)$:

$$H(z)=\frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1+\sum_{k=1}^{M-1} a_k z^{-k}} \tag{3.27}$$

и отображает разностное уравнение

$$y(n)=\sum_{k=0}^{N-1} b(k)s(n-k)-\sum_{k=1}^{M-1} a(k)y(n-k). \tag{3.28}$$

На рис. 3.6 приведена прямая структура звена 2-го порядка, описываемого передаточной функцией

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}$$

и разностным уравнением

$$y(n)=b_0x(n)+b_1s(n-1)+b_2s(n-2)-a_1y(n-1)-a_2y(n-2).$$

В общем случае прямая структура РЦФ содержит $[(N-1)+(M-1)]$ элементов задержки, из них: $(N-1)$ – для предшествующих отсчетов воздействия и $(M-1)$ – для предшествующих отсчетов реакции.

Для уменьшения числа элементов задержки разработана каноническая структура. Структуру называют канонической, если число элементов задержки в ней минимально и равно порядку передаточной функции – $\max\{(M-1),(N-1)\}$.

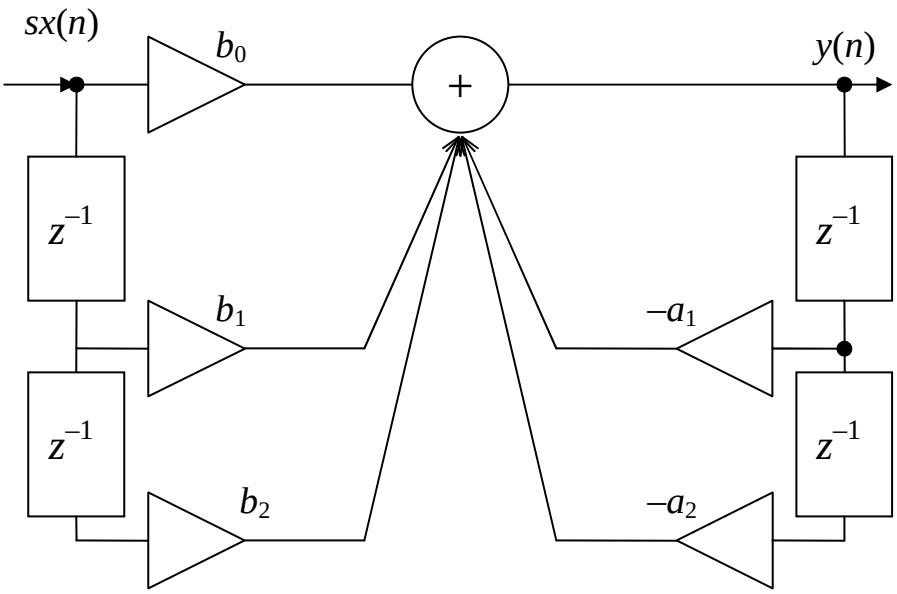


Рис. 3.6. Прямая структура РЦФ 2-го порядка

Прямая каноническая структура 1 определяется эквивалентным представлением передаточной функции $H(z)$ в виде произведения двух передаточных функций:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{1}{1+\sum_{k=1}^{M-1}a_k z^{-k}}\sum_{k=0}^{N-1}b_k z^{-k}=\frac{V(z)}{X(z)}\cdot\frac{Y(z)}{V(z)}=H_1(z)H_2(z), \qquad (3.29)$$

одна из которых описывает рекурсивную часть ЛДС:

$$H_1(z)=\frac{V(z)}{X(z)}=\frac{1}{1+\sum_{k=1}^{M-1}a_k z^{-k}},$$

а вторая – нерекурсивную:

$$H_2(z)=\frac{Y(z)}{V(z)}=\sum_{k=0}^{N-1}b_k z^{-k}.$$

Передаточным функциям $H_1(z)$, $H_2(z)$, согласно их определению, соответствуют разностные уравнения

$$v(n)=s(n)-\sum_{k=1}^{M-1}a_k v(n-k),\ y(n)=\sum_{k=0}^{N-1}b_k v(n-k),$$

отображаемые прямыми структурами.

На рис. 3.7 показана прямая структура звена 2-го порядка.

Аналогичные канонические структуры 1типа имеют РЦФ меньшего и большего порядков.

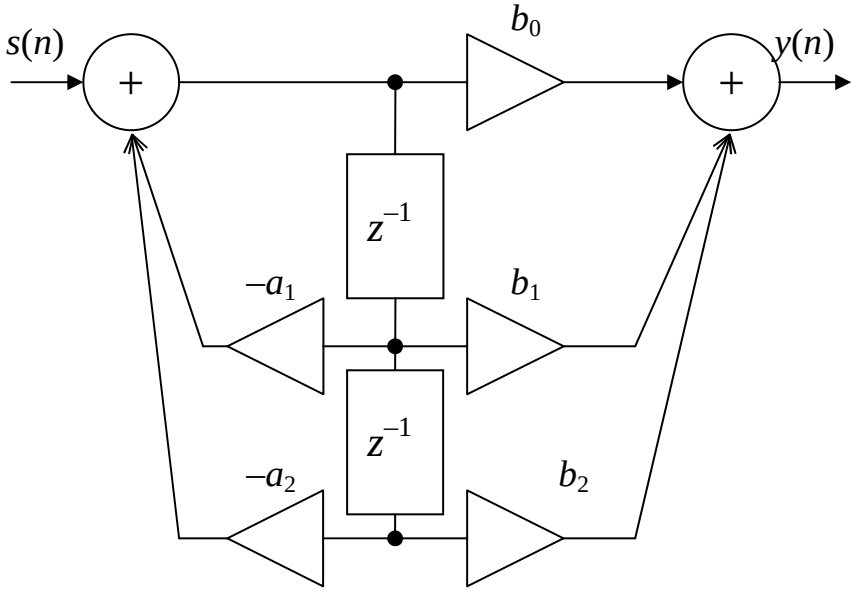


Рис. 3.7. Прямая каноническая структура 1 РЦФ 2-го порядка

Каноническая структура 2 (рис. 3.8) определяется другим эквивалентным представлением передаточной функции $H(z)$ и системой разностных уравнений:

$$H(z) = b_0 + \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} \left[(b_1 - b_0 a_1) z^{-1} + (b_2 - b_0 a_2) z^{-2} \right]; \qquad (3.30)$$

$$\begin{cases} y(n) = b_0 s(n) + (b_1 - b_0 a_1) v(n-1) + (b_2 - b_0 a_2) v(n-2); \\ v(n) = s(n) - a_1 v(n-1) - a_2 v(n-2). \end{cases} \qquad (3.31)$$

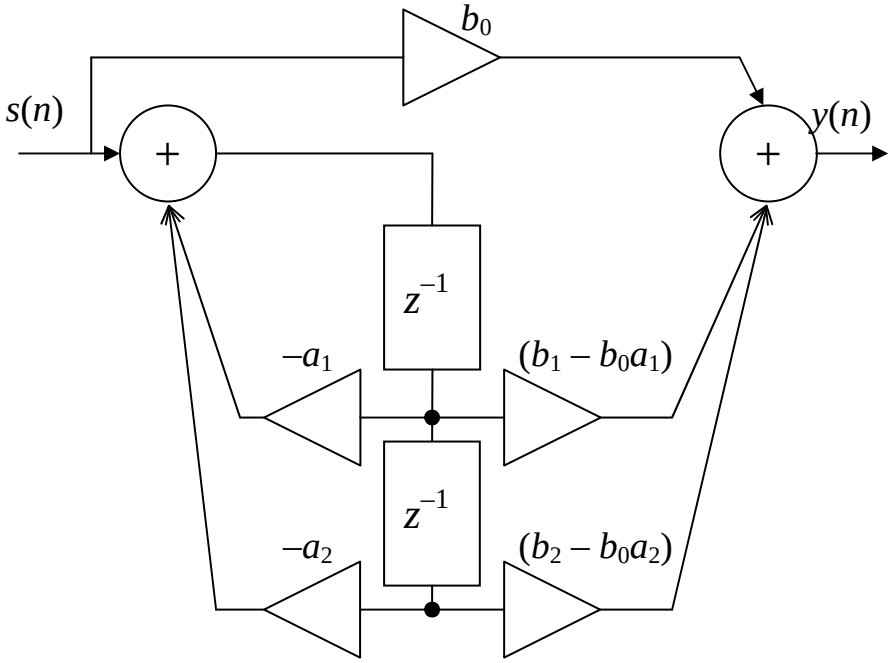


Рис. 3.8. Каноническая структура 2 РЦФ 2-го порядка

На рис. 3.9 приведена каноническая структура 3 РЦФ 2-го порядка. В этом случае передаточная функция и разностное уравнение имеют вид:

$$\begin{cases} H(z) = b_0 + z^{-1}H_1(z); \\ H_1(z) = [b_1 - a_1 H(z)] + z^{-1}H_2(z); \\ H_2(z) = b_2 - a_2 H(z); \end{cases} \tag{3.32}$$

$$\begin{cases} y(n) = b_0 s(n) + v_1(n-1); \\ v_1(n) = b_1 s(n) - a_1 y(n) + v_2(n-1); \\ v_2(n) = b_2 s(n) - a_2 y(n). \end{cases} \tag{3.33}$$

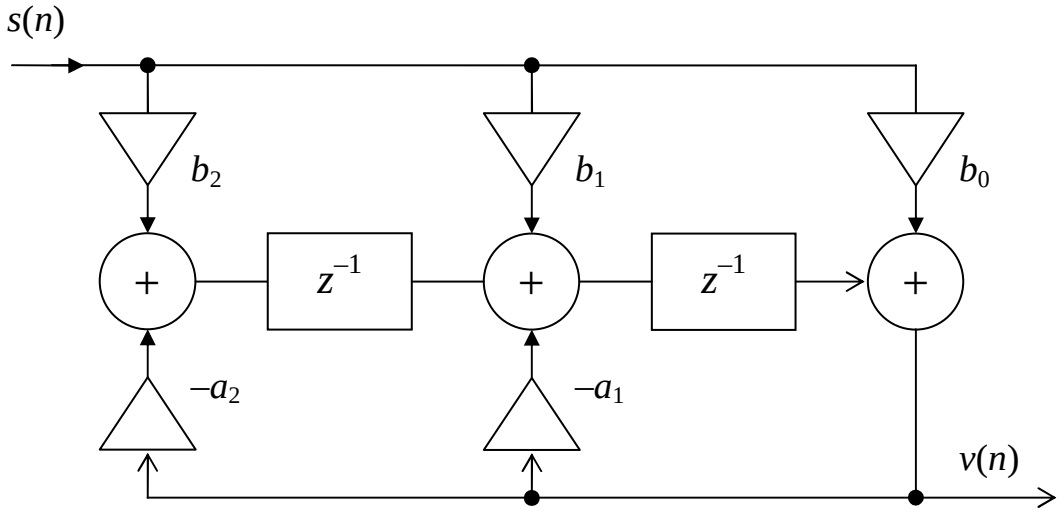


Рис. 3.9. Каноническая структура 3 РЦФ 2-го порядка

3.3.3. Каскадная структура

Каскадная структура определяется передаточной функцией $H(z)$, представленной в виде произведения множителей второго порядка:

$$H(z) = \prod_{i=1}^{K} H_k(z) = \prod_{i=1}^{K} \left(\frac{b_{0i} + b_{1i} z^{-1} + b_{2i} z^{-2}}{1 + a_{1i} z^{-1} + a_{2i} z^{-2}} \right), \tag{3.34}$$

где b_{0i}, b_{1i}, b_{2i}, a_{1i}, a_{2i} – вещественные коэффициенты, а K – количество РЦФ 2-го порядка.

При прямой структуре всех звеньев данному виду передаточной функции соответствует система разностных уравнений

$$\begin{cases} v_1(n) = b_{01}s(n) + b_{11}s(n-1) + b_{21}s(n-2) - a_{11}v_1(n-1) - a_{21}v_1(n-2); \\ v_2(n) = b_{02}v_1(n) + b_{12}v_1(n-1) + b_{22}v_1(n-2) - a_{12}v_2(n-1) - a_{22}v_2(n-2); \\ \ldots \\ y(n) = b_{0.K-1}v_{K-1}(n) + b_{1.K-1}v_{K-1}(n-1) + b_{2.K-1}v_{K-1}(n-2) - a_{1.K-1}y(n-1) - \\ -a_{2.K-1}y(n-2), \end{cases} \quad (3.35)$$

из которой следует, что реакция i-го звена, i = 1, 2,…, (K – 1), служит воздействием для (i + 1)-го звена, поэтому данная система отображается каскадным соединением РЦФ 2-го порядка – каскадной (последовательной) структурой (рис. 3.10).

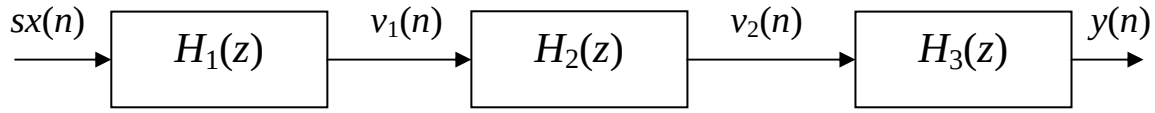


Рис. 3.10. Каскадная структура из трех РЦФ 2-го порядка

3.3.4. Параллельная структура

Параллельная структура определяется передаточной функцией $H(z)$, представленной в виде суммы дробей второго порядка (в частном случае):

$$H(z) = \sum_{i=1}^{K} H_i(z) = \sum_{i=1}^{K} \left(\frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-1}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \right), \quad (3.36)$$

где b_{0k}, b_{1k} , b_{1k} , a_{1k}, a_{2k} – вещественные коэффициенты, а K – количество звеньев 2-го порядка.

Разностное уравнение

$$y(n) = \sum_{i=1}^{K} v_i(n), \quad (3.37)$$

где $v_i(n) = b_{0i}x(n) + b_{1i}s(n-1) + b_{2i}s(n-2) - a_{1i}v_i(n-1) - a_{2k}v_i(n-2)$.

Из разностного уравнения для параллельной структуры следует, что воздействие для всех РЦФ одинаковое, а реакция равна сумме реакций отдельных РЦФ, поэтому данное РУ отображается параллельным соединением рекурсивных звеньев 2-го порядка – параллельной структурой (рис. 3.11).

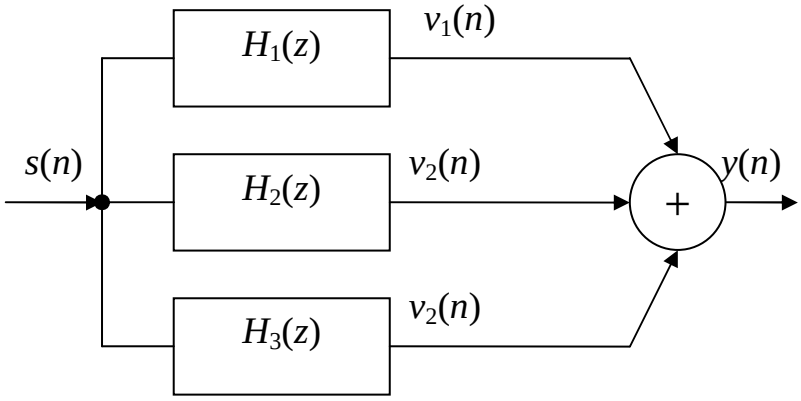


Рис. 3.11. Параллельная структура из трех РЦФ 2-го порядка

3.3.5. Структуры НЦФ

Передаточная функция НЦФ может рассматриваться как частный случай $H(z)$ общего вида при $a_k = 0$, $k = 1,2,..., M - 1$. В этом случае знаменатель $H(z)$ оказывается равным 1.

НЦФ соответствуют два основных типа структур – прямая и каскадная.

Прямая структура (рис. 3.12) определяется передаточной функцией $H(z)$, представленной в виде рациональной функции

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k},$$

и отображает разностное уравнение $y(n) = \sum_{k=0}^{N-1} b_k s(n-k)$.

Каскадная структура определяется передаточной функцией, представленной в виде произведения множителей второго порядка

$$H(z) = \prod_{i=1}^{K} H_i(z) = \prod_{i=1}^{K} (b_{0i} + b_{1i} z^{-1} + b_{2i} z^{-2}), \qquad (3.38)$$

где b_{0i}, b_{1i}, b_{2i} – вещественные коэффициенты, а K – количество НЦФ 2-го порядка.

Система разностных уравнений НЦФ звеньев 2-го порядка описывается:

$$\begin{cases} v_1(n) = b_{01} s(n) + b_{11} s(n-1) + b_{21} s(n-2); \\ v_2(n) = b_{02} v_1(n) + b_{12} v_1(n-1) + b_{21} v_1(n-2); \\ \ldots \\ y(n) = b_{0.K-1} v_{K-1}(n) + b_{1.K-1} v_{K-1}(n-1) + b_{2.K-1} v_{K-1}(n-2). \end{cases} \qquad (3.39)$$

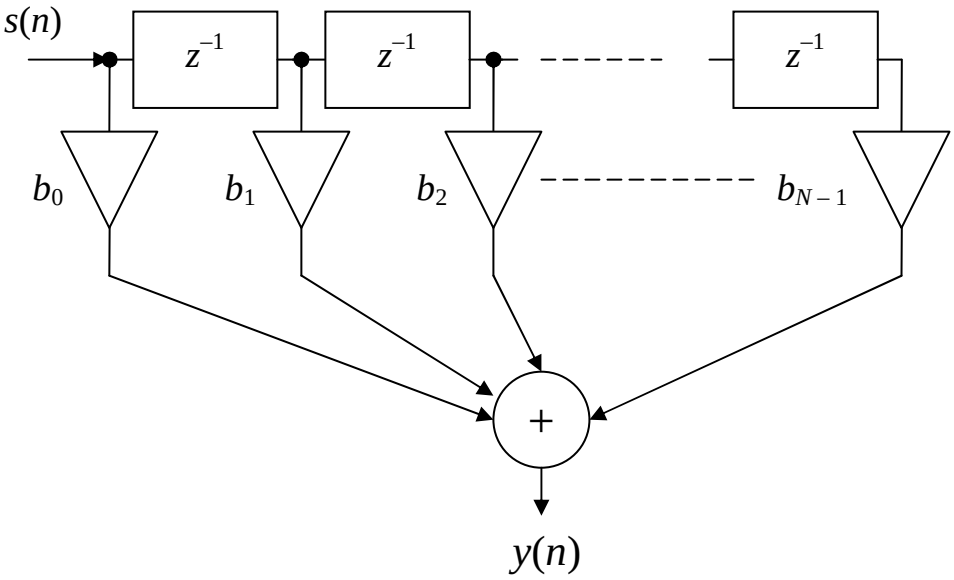


Рис. 3.12. Прямая структура НЦФ

Поскольку одна и та же передаточная функция может быть представлена в разных видах, возникает вопрос о выборе структуры ЦФ, для решения которого прежде всего используются следующие положения:

- для ЦФ вид передаточной функции определяется методом синтеза, тем самым структура по существу оказывается автоматически выбранной: для рекурсивных фильтров обычно она каскадная или параллельная, а для нерекурсивных – прямая приведенная;
- при выборе структуры отдельных ЦФ и последовательности их расположения в каскадном соединении необходимо иметь в виду, что от этого зависит погрешность вычислений – собственные шумы системы.

3.4. Частотные характеристики и элементы синтеза ЦФ

3.4.1. Общие сведения

Частотной характеристикой (ЧХ) $H(e^{j2\pi\hat{f}})$ дискретного фильтра называется отношение преобразования Фурье выходной последовательности к преобразованию Фурье входной последовательности при нулевых начальных условиях:

$$H(e^{j2\pi\hat{f}}) = \frac{F\{y_n\}}{F\{x_n\}} = \frac{Y\{e^{j2\pi\hat{f}}\}}{X\{e^{j2\pi\hat{f}}\}}. \qquad (3.40)$$

Частотная характеристика представляет собой коэффициент передачи фильтра для каждого значения частоты

$$\hat{f} = \frac{f}{f_Д} = fT. \tag{3.41}$$

Из сравнения выражений для частотной характеристики и передаточной функции видно, что ЧХ равна передаточной функции фильтра, вычисленной на единичной окружности z-плоскости (при $z = e^{j2\pi\hat{f}} = e^{j\omega}$):

$$H(e^{j2\pi\hat{f}}) = H(z)\Big|_{z = e^{j2\pi\hat{f}}}. \tag{3.42}$$

Можно установить связь между значениями нормированной частоты $\hat{f}$ и соответствующим углом ω в z-плоскости. Поскольку

$$e^{j2\pi\hat{f}} = \cos 2\pi\hat{f} + j\sin 2\pi\hat{f} = e^{j\omega} = \cos\omega + j\sin\omega,$$

то

$$\hat{f} = \omega / 2\pi.$$

На рис. 3.13 показана связь точек единичной окружности и частот $\hat{f}$:

– точке $z = 1 = \cos 0 + j\sin 0$ соответствует $\hat{f} = 0$;

– точкам $z = \pm j = \cos(\pm\pi/2) + j\sin(\pm\pi/2)$ соответствуют $\hat{f} = \pm 0.25$.

Подставляя значения нормированной частоты, получаем:

$$e^{j2\pi\hat{f}} = \cos 2\pi\left(\pm\frac{1}{4}\right) + j\sin 2\pi\left(\pm\frac{1}{4}\right) = \pm j,$$

точке $z = -1 = \cos(\pm\pi) + j\sin(\pm\pi)$ соответствуют $\hat{f} = \pm 0.5$.

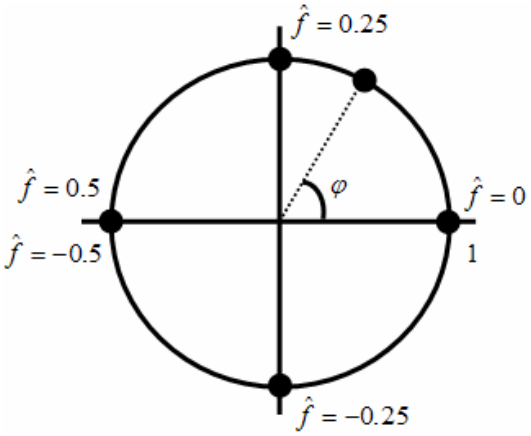


Рис. 3.13. Связь точек единичной окружности и частот $\hat{f}$

3.4.2. Характеристики КИХ-фильтров и БИХ-фильтров

Используя выражения для передаточных функций для нерекурсивных и рекурсивных фильтров, можно записать общие формулы для частотных характеристик дискретных фильтров.

КИХ-фильтр:

$$H(e^{j2\pi\hat{f}}) = \sum_{k=0}^{N-1} b_k e^{-j2\pi\hat{f}k}. \qquad (3.43)$$

БИХ-фильтр:

$$H(e^{j2\pi\hat{f}}) = \frac{\sum_{k=0}^{N-1} b_k e^{-j2\pi\hat{f}k}}{\left(1+\sum_{k=1}^{M-1} a_k e^{-j2\pi\hat{f}k}\right)}. \qquad (3.44)$$

Как уже известно, частотная характеристика $H(e^{j2\pi\hat{f}})$ – комплексная функция. Ее можно представить в виде

$$H(e^{j2\pi\hat{f}}) = H_{\mathrm{Re}}(e^{j2\pi\hat{f}}) + jH_{\mathrm{Im}}(e^{j2\pi\hat{f}}) = A(\hat{f})\cdot e^{j\omega(\hat{f})}. \qquad (3.45)$$

Амплитудно-частотной характеристикой (АЧХ) называется функция

$$A(\hat{f}) = \left|H(e^{j2\pi\hat{f}})\right| = \sqrt{H^2_{\mathrm{Re}}(e^{j2\pi\hat{f}}) + H^2_{\mathrm{Im}}(e^{j2\pi\hat{f}})}. \qquad (3.46)$$

Фазо-частотной характеристикой (ФЧХ) называется функция

$$\varphi(\hat{f}) = \arg H(e^{j2\pi\hat{f}}) = \tan^{-1}(H_{\mathrm{Im}}(e^{j2\pi\hat{f}}) \big/ H_{\mathrm{Re}}(e^{j2\pi\hat{f}})). \qquad (3.47)$$

АЧХ устанавливает связь между амплитудами, а ФЧХ – между фазами выходного $y(n)$ и входного $s(n)$ сигналов фильтра. Если

$$s(n) = A_{вх}\cos(2\pi\hat{f}n + \varphi_{вх}), \qquad (3.48)$$

то в установившемся режиме

$$y(n) = A_{вых}\cdot\cos(2\pi\hat{f}n + \varphi_{вых}), \qquad (3.49)$$

где

$$A_{вх} = A_{вых}A(\hat{f});$$
$$\varphi_{вых} = \varphi_{вх} + \varphi(\hat{f}).$$

Общие формулы АЧХ КИХ-фильтра и БИХ-фильтра имеют вид:

$$A_{КИХ}(\hat{f}) = \sqrt{\left(\sum_{k=0}^{N-1} b_k \cos 2\pi \hat{f} k\right)^2 + \left(\sum_{k=0}^{N-1} b_k \sin 2\pi \hat{f} k\right)^2}; \qquad (3.50)$$

$$A_{БИХ}(\hat{f}) = \sqrt{\frac{\left(\sum_{k=0}^{N-1} b_k \cos 2\pi \hat{f} k\right)^2 + \left(\sum_{k=0}^{N-1} b_k \sin 2\pi \hat{f} k\right)^2}{\left(1 + \sum_{k=1}^{M-1} a_k \cos 2\pi \hat{f} k\right)^2 + \left(\sum_{k=0}^{N-1} a_k \sin 2\pi \hat{f} k\right)^2}}. \qquad (3.51)$$

Частотную характеристику можно рассматривать как спектр импульсной характеристики:

$$H(e^{j2\pi\hat{f}}) = \sum_{k=0}^{\infty} h_k e^{-j2\pi\hat{f}k}. \qquad (3.52)$$

Следовательно, свойства ЧХ фильтра во многом совпадают со свойствами спектра дискретного сигнала. Ниже приведены основные свойства частотных характеристик дискретных фильтров:

- частотная характеристика является периодической функцией по частоте, с периодом, равным частоте дискретизации;
- АЧХ и ФЧХ также являются периодическими функциями с тем же периодом;
- для фильтров с вещественными коэффициентами АЧХ – четная функция, а ФЧХ – нечетная функция;
- для полного описания частотной характеристики фильтра с вещественными коэффициентами достаточно задать ее в основной полосе частот $\hat{f} \in [0, 0.5]$.

Дискретный фильтр называют нормированным, если максимальное значение АЧХ равно 1. Нормировка фильтра достигается путем умножения коэффициентов числителя передаточной функции на нормирующий множитель $q_i = 1/A_{\max}$, где $A_{\max}$ – максимальное значение АЧХ на интервале $\hat{f} \in [0, 0.5]$.

3.5. Синтез цифровых фильтров

Каждый из классов ЦФ (БИХ и КИХ) имеет свои, принципиально отличные, методы синтеза, которые изучаются отдельно для каждого класса. При этом выделяются фильтры, в которых реализованы принципы оптимального синтеза, и фильтры, при синтезе которых принципы оптимальности не принимаются во внимание.

3.5.1. Требования к цифровым фильтрам

Требования к фильтрам могут формулироваться как во временной (рис. 3.14, *а*), так и в частотной областях (см. рис. 3.14, *б*), что определяется назначением фильтра и областью его описания.

Во временной области (см. рис. 3.14, *а*) требования могут задаваться к импульсной $h(n)$ или переходной $g(n)$ характеристике при широких допусках к частотным свойствам фильтра. Так, согласованные с сигналами фильтры целиком определяются импульсной характеристикой, в то время как фильтры, входящие в состав высокоскоростных систем, весьма критичны к длительности переходных процессов, поэтому такие фильтры удобнее описывать переходными характеристиками.

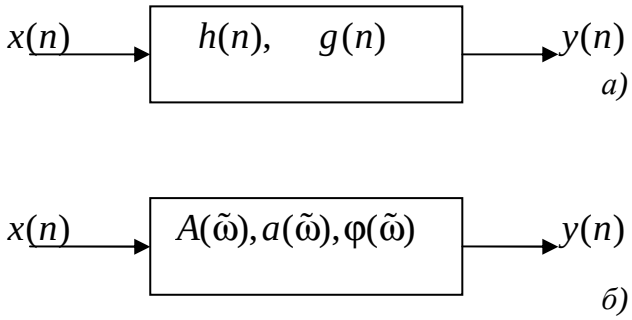


Рис. 3.14. Описание фильтра во временной *(а)* и в частотной *(б)* областях

В частотной области (см. рис. 3.14, *б*) обычно синтезируются избирательные фильтры. При этом требования могут предъявляться:

- только к АЧХ или к характеристике ослабления (затухания) без каких-либо ограничений на ФЧХ;
- только к ФЧХ, когда важно сохранение фазовых, а потому и временных соотношений между гармоническими составляющими принимаемого сигнала. Это фазовые корректоры, у которых АЧХ не зависит от частоты, т. е. $A(\tilde{\omega}) = \text{const}$ на всем интервале частот $0 < \tilde{\omega} < \pi$;
- одновременно и к АЧХ, и к ФЧХ. Например, как будет показано в дальнейшем, возможен синтез КИХ-фильтров с заданной АЧХ и линейной ФЧХ, которая в общем виде представляется формулой

$$\varphi(\tilde{\omega}) = -C\omega T + \varphi_0, \qquad (3.53)$$

где T – период дискретизации; C – константа; $\omega = 2\pi f$; φ_0 – начальная фаза.

При этом групповое время задержки (синонимом этого термина является «групповое время прохождения» – ГВП) (ГВЗ) оказывается постоянным,

$$\tau_{ГВЗ} = \frac{d\varphi(\omega)}{d\omega} = CT = \text{const.} \qquad (3.54)$$

При задании требований в частотной области и в процессе синтеза сами частоты могут задаваться как действительными (f [Гц], ω [рад/с]), так и нормированными $\tilde{\omega}$ [рад], что определяется удобством и традициями. Поскольку связи между перечисленными представлениями частот известны, будет не трудно переходить от одной частоты к другой.

Рассмотрим требования к частотно-избирательным ЦФ. На рис. 3.15 – 3.18, *а* представлены идеальные АЧХ фильтров – НЧ, ВЧ, полосового и режекторного соответственно, которые, вследствие их физической невозможности, необходимо аппроксимировать при заранее определяемых допусках, зависящих от ряда практических ограничений. Эти ограничения связаны, прежде всего, с назначением синтезируемого фильтра. Все ограничения и допуски составляют требования, предъявляемые к фильтру.

Требования к фильтрам включают в себя:

- задание частоты дискретизации $f_Д$ [Гц] и типа избирательности (НЧ, ВЧ и т.д.);
- задание требований к АЧХ $A(f)$ или к характеристике ослабления (затухания) $a(f)$, представляющей собой логарифмическую АЧХ. При этом необходимо помнить, что:
  - задание требований начинается с установки всех граничных частот фильтра *только в основной полосе частот* $0 \le f \le f_Д/2$;
  - требования в переходных полосах избирательных фильтров не задаются;
  - требования формулируются в виде допустимых отклонений от нормированной АЧХ $\tilde{A}(f)$ [или от $a(f)$] в полосах пропускания и задерживания и отображаются на диаграмме допусков;
  - АЧХ $A(f)$ является безразмерной величиной и выражается в абсолютных единицах;

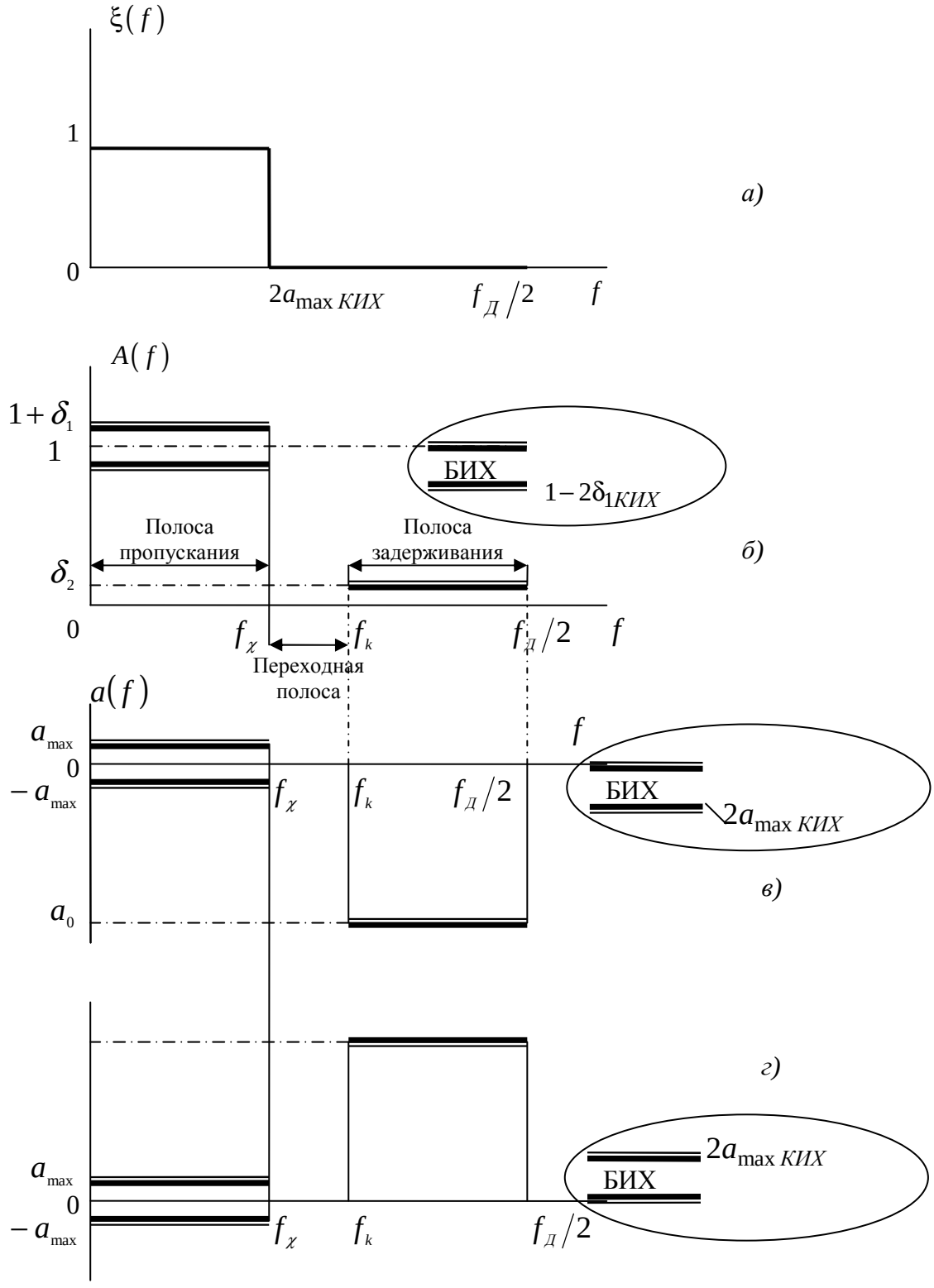


Рис. 3.15. Диаграмма требований к ФНЧ: идеальная АЧХ (*а*), требования к АЧХ (*б*), требования к характеристике ослабления (*в*), требования к характеристике затухания (*г*)

– характеристики ослабления и затухания обозначаются одинаково $a(f)$, размерность $a(f)$ выражается в децибелах, связь между $A(f)$ и характеристикой ослабления $a(f)$ определяется формулой

$$a(f) = 20\lg \tilde{A}(f), \tag{3.55}$$

а при отображении требований в виде характеристики затухания

$$a(f) = -20\lg \tilde{A}(f); \tag{3.56}$$

- выбор метода аппроксимации АЧХ $\tilde{A}(f)$ (или $a(f)$).

3.5.2. Типы избирательных фильтров и задание требований к ним

Здесь изучаются требования, предъявляемые к АЧХ и характеристикам ослабления (затухания) избирательных фильтров. Все требования отображаются на диаграммах. При постановке задач оптимального синтеза идеальную АЧХ обычно обозначают $\xi(f)$. Наиболее подробно излагаются требования к ФНЧ, которые нетрудно перенести и на другие фильтры.

1. ***Фильтр нижних частот*** (ФНЧ) имеет три частотных полосы (см. рис. 3.15, *б*): полосу пропускания (ПП), полосу задерживания (ПЗ) или ослабления (см. рис. 3.15, *в*) и затухания (см. рис. 3.15, *г*), и переходную полосу.

Полоса пропускания (ПП) ограничивается *частотой среза* f_{χ}; ширина полосы пропускания $\Delta f_{ПП} = f_{\chi}$; σ_1 – максимально допустимое отклонение от 1 (см. рис. 3.15, *б*); $a_{\max}$ – максимально допустимое ослабление (см. рис. 3.15, *в*) в полосе пропускания,

$$a_{\max} = 20\lg(1-\sigma_1). \tag{3.57}$$

Соответственно, максимально допустимое *затухание в полосе пропускания* (см. рис. 3.15, *г*) имеет вид

$$a_{\max} = -20\lg(1-\sigma_1). \tag{3.58}$$

Важным является то обстоятельство, что в соответствии с методом синтеза КИХ-фильтров отклонение АЧХ от 1 (соответственно, $a(f)$ от 0) задается симметрично $1-\sigma_1 \le 1+\sigma_1$.

Для БИХ-фильтров отклонение задается только в одну сторону так, чтобы АЧХ не превышала единицы ($1-\sigma_1 \le A(f) \le 1$) и характеристика ослабления, соответственно, не превышала нуля ($a(f) \le 0$), что отображено на рис. 3.15, *б, в* вынесенными диаграммами, что означает, что для БИХ-фильтра, синтезируемого с теми же допусками, что и КИХ-фильтр, необходимо задавать в полосе пропускания допустимое отклонение АЧХ $\sigma_{1\text{БИХ}} = 2\sigma_{1\text{КИХ}}$, допустимое отклонение характеристики ослабления $a_{\max\text{ БИХ}} = 2a_{\max\text{ КИХ}}$ [дБ], допустимое отклонение характеристики затухания $a_{\max\text{ БИХ}} = -2a_{\max\text{ КИХ}}$ [дБ].

Причина, по которой нормированная АЧХ БИХ-фильтров в полосе пропускания не превышает единицы, объясняется методикой синтеза БИХ-фильтров по аналоговым фильтрам-прототипам. *Полоса задерживания* (ослабления) (ПЗ) лежит в пределах от граничной частоты f_k до половины частоты дискретизации $f_д/2$; ее ширина $\Delta f_{ПЗ} = f_д/2 - f_k$; σ_2 – максимально допустимое отклонение АЧХ от 0; a_0 – максимально допустимое ослабление в полосе задерживания,

$$a_0 = 20\lg\sigma_2 < 0, \tag{3.59}$$

а в терминах характеристики затухания a_0 – минимально допустимое затухание в полосе задерживания,

$$a_0 = a_{\min} = -20\lg\sigma_2 < 0. \tag{3.60}$$

Переходная полоса располагается между полосами пропускания и задерживания, ее ширина $\Delta f_{ПЗ} = f_k - f_\chi$. Поскольку в этой полосе требования не задаются, *удовлетворительным окажется любое решение, если оно соответствует требованиям в полосах пропускания и задерживания.*

2. ***Фильтр верхних частот*** (ФВЧ) имеет, как и ФНЧ, три частотных полосы, которые расположены в обратном, относительно ФНЧ, порядке (рис. 3.16):

полоса задерживания в пределах от 0 до f_k, $\Delta f_{ПЗ} = f_k$;

переходная полоса в пределах от f_k до f_χ, $\Delta f = f_\chi - f_k$;

полоса пропускания в пределах от f_χ до $f_д/2$, $\Delta f_{ПП} = f_д/2 - f_\chi$.

Остальные параметры определяются так же, как и для ФНЧ.

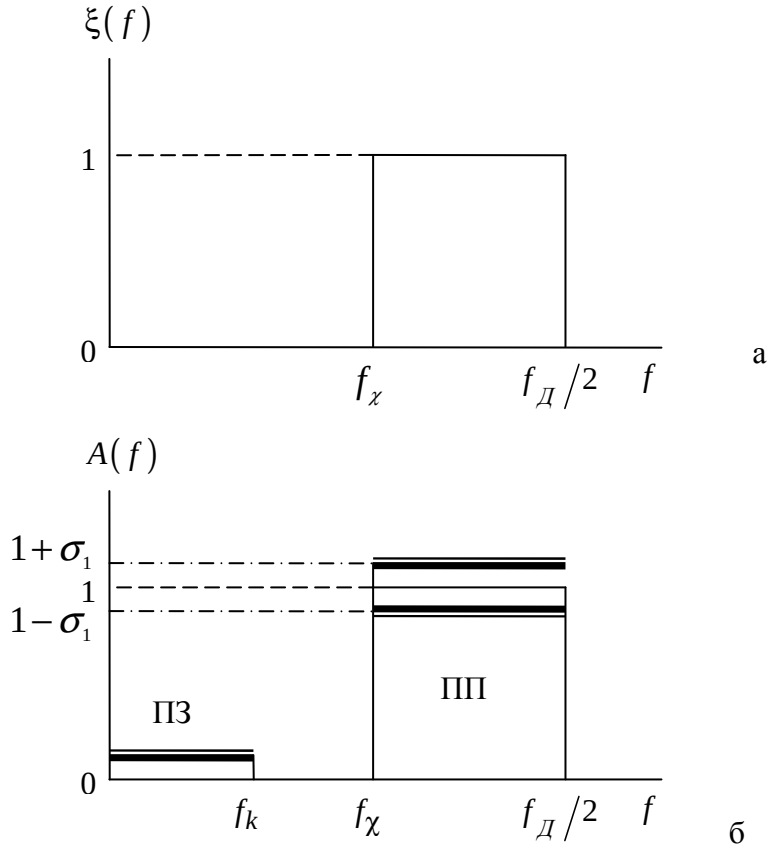


Рис. 3.16. Диаграммы требований к ФВЧ: идеальная АЧХ (*а*), требования к АЧХ (*б*)

3. ***Полосовой фильтр*** (ПФ) характеризуется пятью частотными полосами (рис. 3.16), из которых центральная – полоса пропускания, две полосы задерживания и две переходных полосы. Отрицательный индекс частоты означает, что частота расположена слева от середины полосы пропускания. На рис. 3.17, *б* обозначены:

f_{-k} – граничная частота первой полосы задерживания ПЗ1, ширина которой $\Delta f_{ПЗ1} = f_{-k}$; $f_{-\chi}$ – левая частота среза полосы пропускания: f_χ – правая частота среза полосы пропускания, ширина полосы пропускания $\Delta f_{ПП} = f_\chi - f_{-\chi}$; f_k – граничная частота второй полосы задерживания ПЗ2, ширина которой $\Delta f_{ПЗ2} = f_Д/2 - f_k$.

Переходные полосы 1 и 2 имеют ширину $\Delta f_1 = f_{-\chi} - f_{-k}$ и $\Delta f_2 = f_k - f_\chi$ соответственно.

Примечание. При задании требований к ПФ следует иметь в виду, что в общем случае $\Delta f_{ПЗ1} \neq \Delta f_{ПЗ2}, \Delta f_1 \neq \Delta f_2$ – допустимые отклонения от нуля в полосах задерживания не обязаны быть одинаковыми.

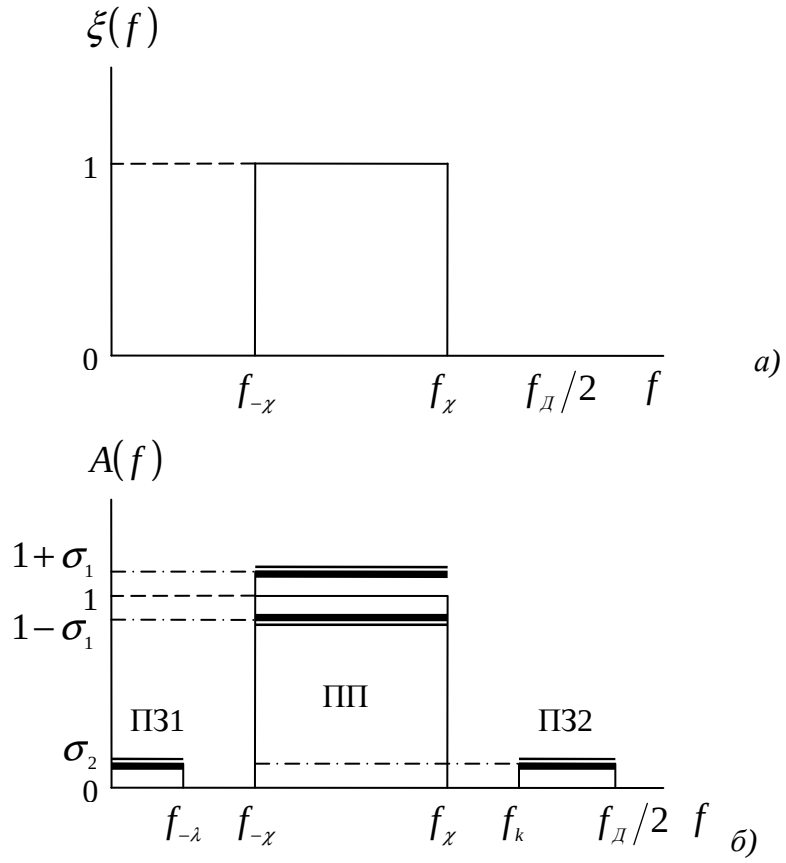


Рис. 3.17. Диаграммы требований к ПФ: идеальная АЧХ (*а*), требования к АЧХ (*б*)

4. ***Режекторный фильтр*** (РФ), иногда называемый полосно-заграждающим, подобно ПФ характеризуется пятью полосами (рис. 3.18), из которых две полосы пропускания, одна полоса задерживания и две переходных полосы. Отрицательный индекс частоты означает, что частота расположена слева от середины полосы задерживания. На рис. 3.18, *б* обозначены: $f_{-\chi}$ – частота среза первой полосы пропускания, ширина которой $\Delta f_{ПП1} = f_{-\chi}$; f_{-k} и f_k – нижняя и верхняя граничные частоты полосы задерживания, $\Delta f_{ПЗ} = f_k - f_{-k}$; f_χ – частота среза второй полосы пропускания, ширина которой $\Delta f_{ПП2} = f_Д/2 - f_\chi$.

Переходные полосы 1 и 2 имеют ширину $\Delta f_1 = f_{-k} - f_{-\chi}$ и $\Delta f_2 = f_\chi - f_k$ соответственно.

Примечание. При задании требований к РФ следует иметь в виду, что в общем случае $\Delta f_{ПП1} \neq \Delta f_{ПП2}, \Delta f_1 \neq \Delta f_2$ – допустимые отклонения от единицы в полосах пропускания не обязаны быть одинаковыми.

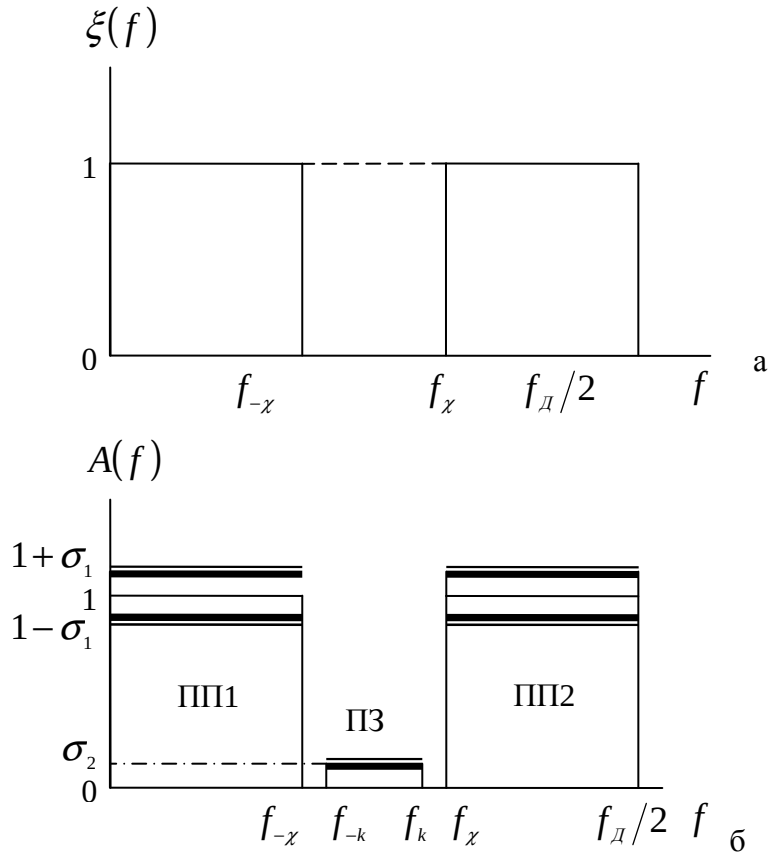


Рис. 3.18. Диаграммы требований к РФ: идеальная АЧХ (*а*), требования к АЧХ (*б*)

3.5.3. Характеристика задачи оптимального синтеза

Цель оптимального синтеза в самом общем смысле состоит в том, чтобы при заданных условиях и ограничениях получить фильтр, который имел бы наилучшие параметры: минимальный порядок, минимум максимального отклонения реальной частотной или временной характеристики от желаемой, малый собственный шум, желаемые особенности структуры и т.д.

Типовой, весьма распространенной задачей оптимального синтеза является получение цифрового фильтра минимального порядка при условии достижения наилучшего воспроизведения (наилучшей аппроксимации) требуемой АЧХ при заданных ограничениях.

Преследуемая цель формально может быть выражена как функция одного или нескольких аргументов (частоты f и вектора коэффициентов $\vec{b} = \{b_0, b_1 \ldots b_k\}$), которую требуется минимизировать (или максимизировать). Минимизируемая функция называется целевой – ею определяется качество достижения поставленной цели; степень же приближения оцени-

вается одним числом. К примеру, степень полученного воспроизведения АЧХ оценивается допустимым отклонением σ_1 в полосе пропускания фильтра.

Однако достижение определенной цели может иметь разный смысл, определяемый мерой близости, или критерием оптимальности.

Любое решение, удовлетворяющее заданным ограничениям, называется допустимым. *Оптимальное решение является лучшим допустимым решением в смысле выбранного критерия*, т.е. меры близости.

Критерием оптимальности называют показатель, который характеризует общую ценность решений таким образом, что решение признается тем лучшим, чем меньше (или больше) значение показателя при заданных условиях и ограничениях задачи. Это означает, что *любая оценка решения в виде некоторого числа является оценкой по определенному критерию.* Критерий оптимальности включается в постановку задачи.

Задача приближения реальной АЧХ $A(f)$ к желаемой $\xi(f)$ решается на совокупности F ограниченных областей частот $F_i \in (0 + f_д/2)$, которые называются *интервалами аппроксимации.* В эту область входят *только полосы пропускания и задерживания фильтра.* Так, фильтры нижних и верхних частот имеют по два интервала аппроксимации, а полосовые и режекторные фильтры имеют по три интервала аппроксимации (см. рис. 3.15).

Чаще всего при конструировании передаточной функции $H(z)$ в качестве критерия оптимальности используется мера близости ρ реальной АЧХ $A(f)$ или амплитудной функции $B(f)$ к желаемой $\xi(f)$. Выбранная мера близости ρ определяет характер задачи оптимизации, смысл которой состоит в достижении минимума (или максимума) целевой функции.

3.6. Специализированные цифровые фильтры

3.6.1. Сглаживающие цифровые фильтры

К специализированным сглаживающим фильтрам, которые здесь рассматриваются, относятся рекурсивный фильтр экспоненциального весового усреднения, нерекурсивные фильтры на основе различных весовых функций – однородный, триангулярный и другие, нерекурсивные фильтры на основе параболической аппроксимации, медианный фильтр. Они используются для подавления аддитивных ВЧ-составляющих, высших гармоник и шумов и выделения медленно изменяющейся полезной составляющей, например, на выходе цифрового или аналогового детектора или демодулятора.

3.6.2. Рекурсивный фильтр весового экспоненциального усреднения

Рекурсивное звено первого порядка (рис. 3.19) с вещественными коэффициентами b_0, a_1, связанными соотношением

$$b_0 = 1/\mu, a_1 = -(\mu - 1)/\mu = -1 + b_0, \qquad (3.61)$$

в соответствии с его разностным уравнением

$$y(n) = b_0 x(n) = -a_1 y(n-1) = (1/\mu)x(n) + \left[(\mu - 1)/\mu\right] y(n-1) \qquad (3.62)$$

реализует так называемый алгоритм экспоненциального весового усреднения или сглаживания [18, 19]. Параметром его является коэффициент сглаживания $\mu > 1$.

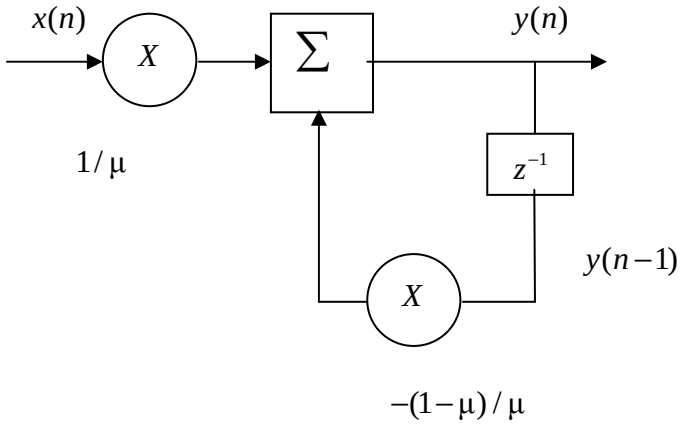


Рис. 3.19. Структурная схема сглаживающего рекурсивного звена

Если представить сигналы на входе и выходе звена в виде суммы их математического ожидания (МО) и случайной аддитивной помехи

$$x(n) = M\left[x(n)\right] + e_x(n), y(n) = M\left[y(n)\right] + e_y(n), \qquad (3.63)$$

то в случае идеального сглаживающего фильтра выходной сигнал $y(n)$ должен являться несмещенной оценкой МО входного сигнала: $M\left[y(n)\right] = M\left[x(n)\right]$, а его дисперсия $D\left[y(n)\right] = D\left[e_y(n)\right]$ должна быть минимальной в соответствии с критерием минимума среднего квадрата ошибки (СКО). Используя разностное уравнение звена, можно найти связь между МО и дисперсией его выходного и входного сигналов:

$$M[y(n)] = (1/\mu)M[x(n)] + [(\mu-1)/\mu]M[y(n-1)];$$
$$D[y(n)] = (1/\mu^2)D[x(n)] + [(\mu-1)/\mu]^2 D[y(n-1)]. \quad (3.64)$$

Как видим, в общем случае $M[y(n)] \neq M[x(n)]$,т.е. выходной сигнал является смещенной оценкой МО входного сигнала, и это смещение при $n \leq \mu$ определяется выражением

$$M[y(n)] = M[x(n)]\left[1-(\mu-1)^n/\mu^n\right] \quad (3.65)$$

и составляет до $36{,}8\%$. При $n >> \mu$, т.е. в установившемся режиме справедливы допущения:

$$M[y(n-1)] \approx M[y(n)], D[y(n-1)] \approx D[y(n)], \quad (3.66)$$

при которых $M[y(n)] \approx M[x(n)]$ − смещение оценки с ростом n стремится к нулю, а дисперсия ее $D[y(n)] = D[x(n)]/(2\mu-1)$ в $(2\mu-1)$ раз меньше дисперсии входного сигнала (в том числе и при $n = \mu$. По эффективности, т.е. дисперсии оценка экспоненциального сглаживания при $n = \mu$ почти в 2 раза лучше оценки линейного сглаживания, для которой $D[y(n)] = D[x(n)]/n$. Таким образом, чем больше μ, тем меньше случайная погрешность оценки, но больше ее систематическая погрешность при значениях n, соизмеримых с μ, и медленнее ее убывание во времени, т.е. с ростом n. Физически это означает возрастание инерционности и ухудшение способности фильтра отслеживать изменения входного сигнала (множитель $1/\mu$ перед $x(n)$).

Отклик фильтра на единичный импульс при $n = 0,1,...$ равен:

$$y(0) = 1/\mu, y(1) = (1/\mu)[(\mu-1)/\mu], y(2) = (1/\mu)[(\mu-1)/\mu]^2, \quad (3.67)$$

что соответствует импульсной характеристике фильтра

$$h(n) = \frac{1}{\mu}\left(\frac{\mu-1}{\mu}\right)n. \quad (3.68)$$

При $\mu \geq (5-10)$ множитель $[(\mu-1)/\mu]^n$ стремится к $e^{-n/\mu}$ и импульсная характеристика принимает вид экспоненциальной весовой функции $h(n) \approx \frac{1}{\mu}e^{-n/\mu}$. Это объясняет название данного фильтра и реализуемого им алгоритма экспоненциального усреднения или взвешивания.

Фильтр имеет вещественный полюс $r_p = (\mu - 1)\mu < 1$ и описывается передаточной функцией и АЧХ вида

$$H(z) = \frac{1/\mu}{1 - \frac{\mu - 1}{\mu} z^{-1}}; \left|H(j\lambda)\right| = \frac{1}{\left|\mu - (\mu - 1)e^{-j\lambda}\right|} = \frac{1}{\sqrt{1 + 2\mu(\mu - 1)(1 - \cos\lambda)}}. \quad (3.69)$$

С ростом μ уменьшается полоса пропускания фильтра. Поэтому в следящих системах значение μ задают небольшим в режиме установления или захвата и увеличивают его в режиме слежения.

Предлагается самостоятельно построить и проанализировать графики АЧХ и импульсной характеристики фильтра.

3.6.3. Нерекурсивные сглаживающие фильтры на основе весовых функций

Нерекурсивный фильтр осуществляет линейное сглаживание сигнала в соответствии с алгоритмом весового усреднения

$$y(n) = \frac{1}{U} \sum_{m=0}^{N-1} w(m) x(n - m), \quad (3.70)$$

где U – масштабирующий множитель, зависящий от вида весовой функции $w(m)$. Такой алгоритм обеспечивает несмещенную оценку входного сигнала: $M\left[y(n)\right] = M\left[x(n)\right]$. Прямоугольной весовой функции соответствует алгоритм скользящего среднего

$$y(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(n - m) \quad (3.71)$$

с минимальной (для белого шума) дисперсией оценки $D_y = D_x / N$. Такой фильтр называют однородным [20], так как все сглаживаемые отсчеты у него имеют одинаковые веса, равные единице. Его передаточная функция

$$H(z) = \frac{1}{N} \sum_{m=0}^{N-1} z^{-m} \quad (3.72)$$

может быть преобразована к виду

$$H(z) = \frac{1}{N} \frac{1 - z^{-N}}{1 - z^{-1}}, \quad (3.73)$$

которому отвечает разностное уравнение

$$y(n) = N^{-1}\left[x(n) - x(n-N) + y(n-1)\right].$$

Структура реализуемого в соответствии с ним фильтра (рис. 3.20) содержит нерекурсивную и рекурсивную части и при больших N требует намного меньшего объема вычислений.

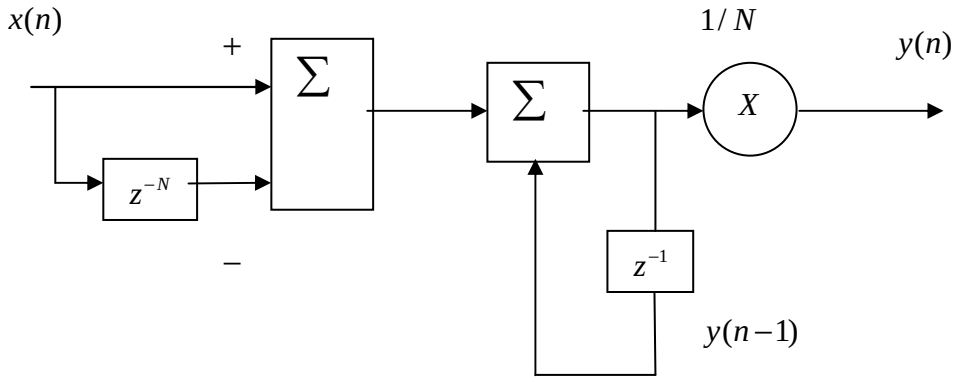


Рис. 3.20. Структурная схема сглаживающего рекурсивного звена

Однако в данном случае невозможно прореживание выходного сигнала в процессе обработки, предоставляемое алгоритмом на основе ДВС [7, 20]. Коэффициент прореживания для сглаживающих НФ на основе ДВС может достигать значения, равного N.

Другие весовые функции имеют более широкую по сравнению с прямоугольной шумовую полосу и менее эффективны относительно белого шума. Однако они лучше подавляют сосредоточенные или узкополосные помехи вне главного лепестка частотной характеристики весовой функции, определяющей частотную характеристику сглаживающего НФ. Важные для задач сглаживания параметры весовых функций приводятся в [7, 18, 21].

НФ с прямоугольной и треугольной весовыми функциями (последний фильтр называют еще триангулярным) применяют также в качестве фильтров-интерполяторов соответственно нулевого и первого порядка в восходящих дискретных системах, т.е. системах с повышением частоты дискретизации [7, 20]. В случае нулевого (или ступенчатого) интерполятора недостающие (нулевые) отсчеты входного сигнала на каждом интервале интерполяции $T'_Д$ замещаются в выходном сигнале $(L-1)$-м узловым значением входного сигнала на границе интервала интерполяции. Здесь

$L = T'_Д / T_Д$ соответствует коэффициенту повышения частоты дискретизации и длине импульсной характеристики фильтра N. При интерполяции по линейному закону длина треугольной импульсной характеристики фильтра вдвое превышает коэффициент передискретизации сигнала $L : N = 2L$. Линейную интерполяцию сигнала можно выполнить также с помощью двух каскадно включенных интерполяторов нулевого порядка.

В качестве примера на рис. 3.21 показано применение сглаживающего однородного фильтра в устройстве обнаружения и оценки параметров гармонических сигналов [20].

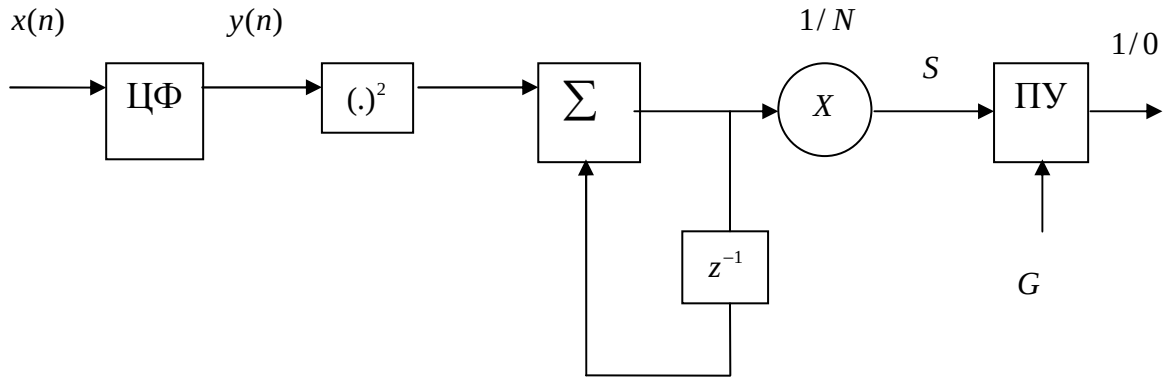


Рис. 3.21. Пример структурной схемы применения сглаживающего однородного фильтра

Отсчеты выделяемого избирательным цифровым фильтром сигнала $y(n)$ возводятся в квадрат и в течение времени $NT_Д$ накапливаются с помощью реализуемого в рекурсивной форме сглаживающего однородного фильтра. Выходной сигнал сглаживающего фильтра

$$s = (1/N)\sum_{n=0}^{N-1} y^2(nT_Д) \qquad (3.74)$$

на каждом N-м такте сравнивается в пороговом устройстве ПУ с некоторым пороговым значением G, при превышении которого принимается решение о присутствии на входе устройства полезного гармонического сигнала. При этом длина импульсной характеристики сглаживающего фильтра должна выбираться из условия $NT_Д = kT_0$, где T_0 – период обнаруживаемого или выделяемого сигнала, $k = 1, 2, ...$ – целые числа. В этом случае значение сигнала на выходе сглаживающего фильтра после установления

переходных процессов в ЦФ равно квадрату действующего значения входного синусоидального сигнала: $s = a^2 / 2$.

Полоса пропускания ЦФ соответствует частоте полезного сигнала, а полоса задерживания – частотам подавляемых помех.

Аналогичная структура устройства (без ПУ) используется при полосовом спектральном анализе случайных сигналов [18].

3.6.4. Режекция фиксированных частот с помощью нерекурсивных фильтров

Режекцию фиксированных частот сигнала λ_{ci} можно осуществить с помощью НФ, нули которого, размещенные на единичной окружности, совпадают по частоте с подавляемыми составляющими сигнала: $\lambda_{0i} = \lambda_{ci}$. Фильтру, подавляющему одну частоту λ_c, соответствуют комплексносопряженная пара нулей $z_{01,2} = e^{\pm j\lambda c} = \cos\lambda_c + j\sin\lambda_c$ и коэффициенты $b_0 = 1;\ b_1 = -2\cos\lambda_c;\ b_2 = 1$. Передаточная функция и частотная характеристика такого заграждающего фильтра определяются выражениями

$$H(z) = 1 - 2\cos\lambda_c z^{-1} + z^{-2};\ \ H(j\lambda) = 1 - 2\cos\lambda_c z^{-j\lambda} + z^{-2j\lambda}. \quad (3.75)$$

Рекомендуется самостоятельно получить выражение и исследовать АЧХ данного фильтра.

Для L фиксированных частот режекции коэффициенты НФ можно найти по заданным значениям нулей путем представления передаточной функции его в виде

$$H(z) = \prod_{i=1}^{L}(1 - z_{0i}z^{-1}) = \prod_{i=1}^{L}(1 + b_{1i}z^{-1} + z^{-2}) = \sum_{n=0}^{N-1} h(n)z^{-n}, \quad (3.76)$$

где $b_{1i} = -2\cos\lambda_c$.

Фильтр можно реализовать по каскадной или прямой структуре.

Режекцию применяют для подавления сосредоточенных помех, например, сетевой помехи в медицинских приборах.

3.6.5. Согласованные (оптимальные) нерекурсивные фильтры

Согласованные НФ (СФ) используются для обнаружения и оценки временного положения зашумленных периодических сигналов и сигналов конечной длительности $T_c = (N-1)T_Д$, обеспечивая максимальное отноше-

ние мощности сигнала к мощности шума в момент окончания сигнала, а также сжатие сигнала по длительности. Применяются они также для полосового спектрального анализа сигналов (ПСА) и оценки их параметров. Выходной сигнал СФ соответствует автокорреляционной функции (АКФ) входного сигнала.

При обнаружении сигнала с помощью СФ (рис. 3.22) возведенное в квадрат значение модуля выходного сигнала фильтра $|y(N-1)|^2$ в момент времени $(N-1)T_Д$, которое соответствует его пиковой мощности P_c, сравнивается в пороговом устройстве (ПУ) с некоторым порогом G, превышающим значение мощности выходного шума

$$P_ш = \sigma_x{}^2 \sum_{n=0}^{N-1} |h(n)|^2, \qquad (3.77)$$

где $\sigma_x{}^2$ – мощность входного шума, $h(n)$ – импульсная характеристика СФ.

Обычно $G \geq 1{,}5P_ш$ [20]. При превышении порога G принимается решение о наличии сигнала, в противном случае – об его отсутствии.

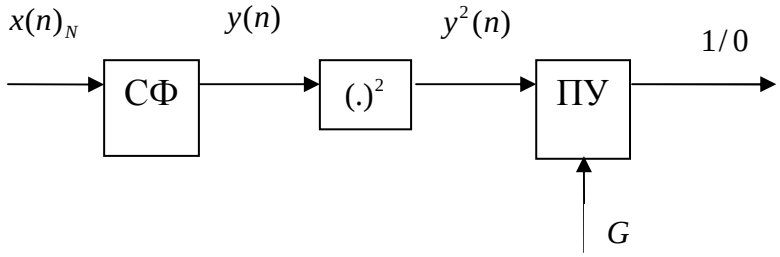


Рис. 3.22. Общая структура обнаружителя сигнала на основе СФ

Для сигнала конечной длительности $x(n)_N$ импульсная характеристика СФ определяется как $h(n)_N = x(N-1-n)$, а его частотная характеристика – комплексно сопряженная спектру входного сигнала:

$$H(j\omega) = X^*(j\omega). \qquad (3.78)$$

Спектр выходного сигнала

$$Y(j\omega) = X(j\omega)X^*(j\omega) = |X(j\omega)|^2, \qquad (3.79)$$

при этом он чисто вещественный, так как фазовый спектр его $\varphi_y(\omega) = 0$. За счет компенсации (подкрутки) фаз составляющих сигнала они суммируют-

ся синфазно, образуя пик выходного сигнала. Наибольшее значение максимума отношения сигнал/шум имеют сложные сигналы с большой базой, т.е. произведением полоса×длительность. Это ЛЧМ сигналы с квадратичной фазой, специальные кодовые видео и радио (фазоманипулированные) последовательности (М-последовательности, коды Баркера и др.).

3.6.6. Нерекурсивные фильтры, согласованные с гармоническим сигналом

С помощью СФ эффективно решаются важные для практики задачи обнаружения гармонических сигналов и оценки их параметров. Для гармонического сигнала

$$x(n) = a\sin(\lambda_0 n + \varphi) = a\cos\varphi\sin\lambda_0 n + a\sin\varphi\cos\lambda_0 n \qquad (3.80)$$

с известной начальной частотой $\lambda_0 = \omega_0 T_Д$ и неизвестной начальной фазой φ импульсной характеристикой согласованного с ним фильтра является комплексная экспонента

$$h(n)_N = e^{j\lambda_0(N-1-n)} = \cos\lambda_0(N-1-n) + j\sin\lambda_0(N-1-n). \qquad (3.81)$$

Она соответствует комплексному или квадратурному НФ, состоящему из двух согласованных вещественных фильтров $СФ_c$, $СФ_s$. Один из них с импульсной характеристикой $\cos\lambda_0(N-1-n)$ ($СФ_c$) согласован с косинусной составляющей сигнала, а другой с импульсной характеристикой $\sin\lambda_0(N-1-n)$ ($СФ_s$) – с синусной составляющей сигнала. Структура построенного на основе комплексного СФ обнаружителя приведена на рис. 3.23.

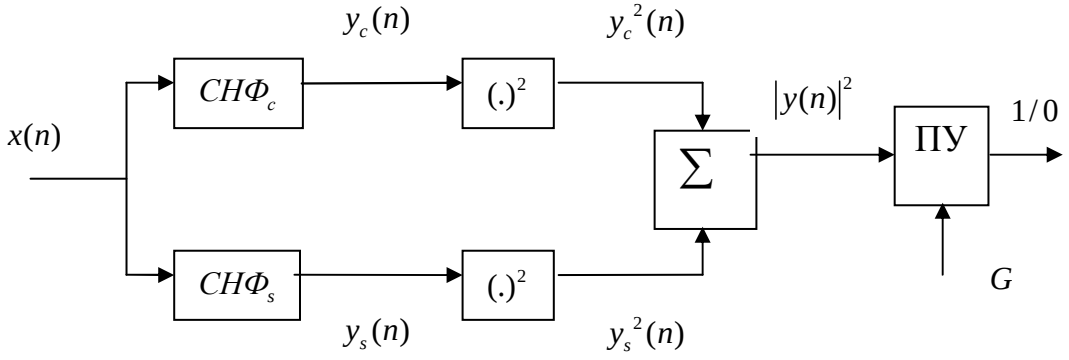


Рис. 3.23. Структурная схема обнаружителя гармонического сигнала на основе согласованного с ним комплексного нерекурсивного фильтра

Для изучения свойств такого обнаружителя и выбора длины импульсной характеристики СФ N запишем выражение для выходного сигнала комплексного фильтра при воздействии на него гармонического сигнала

$$x(n) = a\sin(\lambda n + \varphi) = (a/2j)(e^{j\lambda n} - e^{-j\lambda n}) \tag{3.82}$$

с частотой λ, в общем случае не равной частоте λ_0, с которой согласован фильтр:

$$\begin{aligned} y(n) &= y_c(n) + jy_s(n) = \sum_{l=0}^{n} x(l)h(n-1) = \\ &= \frac{a}{2j} e^{j\lambda_0(N-1-n)} \left[e^{j\varphi} \frac{1-e^{j(\lambda-\lambda_0)(n+1)}}{1-e^{j(\lambda-\lambda_0)}} - e^{-j\varphi} \frac{1-e^{-j(\lambda+\lambda_0)(n+1)}}{1-e^{-j(\lambda+\lambda_0)}} \right] \end{aligned} \tag{3.83}$$

(при выводе использована формула суммы конечного числа членов геометрической прогрессии).

Рассмотрим случай $\lambda = \lambda_0$. При $n = N-1$ выражение для выходного сигнала фильтра приводится к виду

$$y(N-1) = \frac{a}{2j} \left[e^{j\varphi} N - e^{-j\varphi} \frac{1-e^{-j2\lambda_0 N}}{1-e^{-j2\lambda_0}} \right]. \tag{3.84}$$

При произвольном значении длины импульсной характеристики N квадрат модуля выходного сигнала СФ будет зависеть от неизвестной фазы сигнала φ. Эта зависимость устраняется при выборе N из условия: $2\lambda_0 N = 2k\pi,$ где $k = 1, 2, ..., N-1$. При этом условии

$$y(N-1) = (a/2j)e^{j\varphi} N = (a/2j)N(\cos\varphi + j\sin\varphi); \tag{3.85}$$

$$\left|y(N-1)\right|^2 = {y_c}^2(N-1) + {y_s}^2(N-1) = (a^2/2)N^2 \tag{3.86}$$

и, следовательно, возможно обнаружение сигнала с помощью порогового устройства ПУ. Значения частот f_0, с которыми может быть согласован фильтр, определяются соотношением $f_0 = kf_Д / 2N$. С увеличением N возрастает мощность полезной составляющей сигнала.

Ограничение на возможные значения центральных частот НФ, согласованного с гармоническим сигналом, и длины импульсной характеристики (времени обнаружения сигнала) снимаются в случае комплексного входного гармонического сигнала: $x(n) = ae^{j(\lambda n+\varphi)}$. По аналогии получа-

ются следующие выражения для выходного сигнала комплексного СФ и квадрата его модуля при $n = N - 1$:

$$y(N-1) = ae^{j\varphi}\frac{1-e^{j(\lambda-\lambda_0)N}}{1-e^{j(\lambda-\lambda_0)}}; \tag{3.87}$$

$$\left|y(N-1)\right|^2 = a^2\left|\frac{1-e^{j(\lambda-\lambda_0)N}}{1-e^{j(\lambda-\lambda_0)}}\right|^2. \tag{3.88}$$

Квадрат модуля не зависит от начальной фазы при любом значении частоты сигнала λ, а обнаружение сигнала с частотой $\lambda = \lambda_0$ возможно при любых значениях частоты λ_0 и длины импульсной характеристики $N: \left|y(N-1)\right|^2 = a^2N^2$.

Зависимость

$$A(\lambda, \lambda_0) = \left|y(N-1)\right| / a = \left|\frac{1-e^{j(\lambda-\lambda_0)N}}{1-e^{j(\lambda-\lambda_0)}}\right| = \left|\frac{\sin(\lambda-\lambda_0)N/2}{\sin(\lambda-\lambda_0)/2}\right| \tag{3.89}$$

называется АЧХ обнаружителя, согласованного с частотой λ_0. Она соответствует частотной характеристике прямоугольной весовой функции, смещенной по частоте на λ_0. С помощью параметра N можно управлять положением нулей АЧХ, обеспечивая таким образом наряду с выделением полезного гармонического сигнала с частотой λ_0 (или близкой к ней) подавление гармонических помех за пределами полосы пропускания фильтра. Значение N определяет также время наблюдения сигнала и эффективность подавления белого шума.

Преимущества согласованной фильтрации комплексных гармонических сигналов могут быть в полной мере реализованы и в случае вещественного гармонического сигнала, если использовать преобразователь Гильберта, преобразующий вещественный сигнал в комплексный. Структура обнаружителя в этом случае принимает вид, приведенный на рис. 3.24 [20].

При этом входной и выходной сигналы преобразователя Гильберта (ПГ) соответствуют косинусной $x_c(n)$ и синусной $x_s(n)$ составляющим в случае обработки комплексного входного сигнала, рассмотренной выше. Преобразователь Гильберта также реализуется на основе НФ.

Квадратурные обнаружители позволяют оценивать амплитуду и фазу сигнала, с которым согласован обнаружитель, при условии, что амплитуды помех достаточно малы.

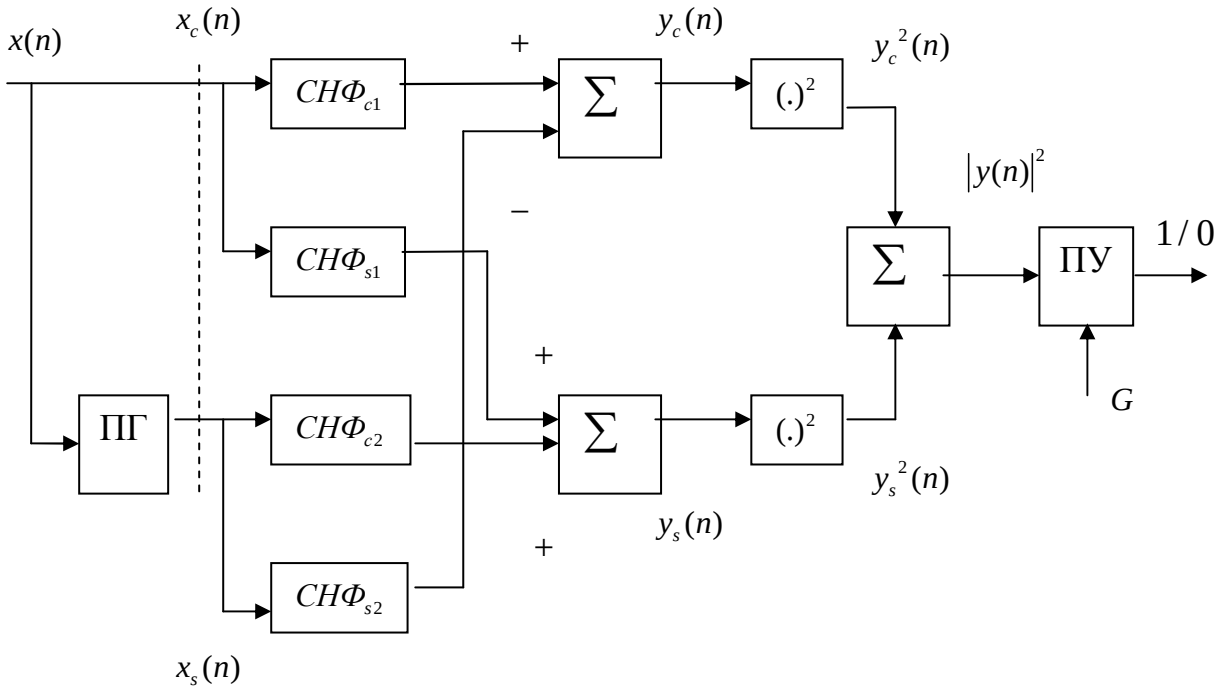


Рис. 3.24. Структурная схема обнаружителя комплексного гармонического сигнала или вещественного сигнала с преобразователем Гильберта

3.6.7. Простые алгоритмы цифрового дифференцирования сигналов

С помощью нерекурсивных фильтров реализуются простые алгоритмы приближенного цифрового дифференцирования сигналов, используемые, например, в радиолокации (селекция движущихся целей) [19], электрокардиографии (выделение R-зубцов) [19, 22] и других областях.

Двухточечный алгоритм дифференцирования описывается уравнением вида $y(n) = (1/T_Д)\left[x(n) - x(n-1)\right]$ и реализуется в соответствии со структурной схемой НФ – рис. 3.25, *а*.

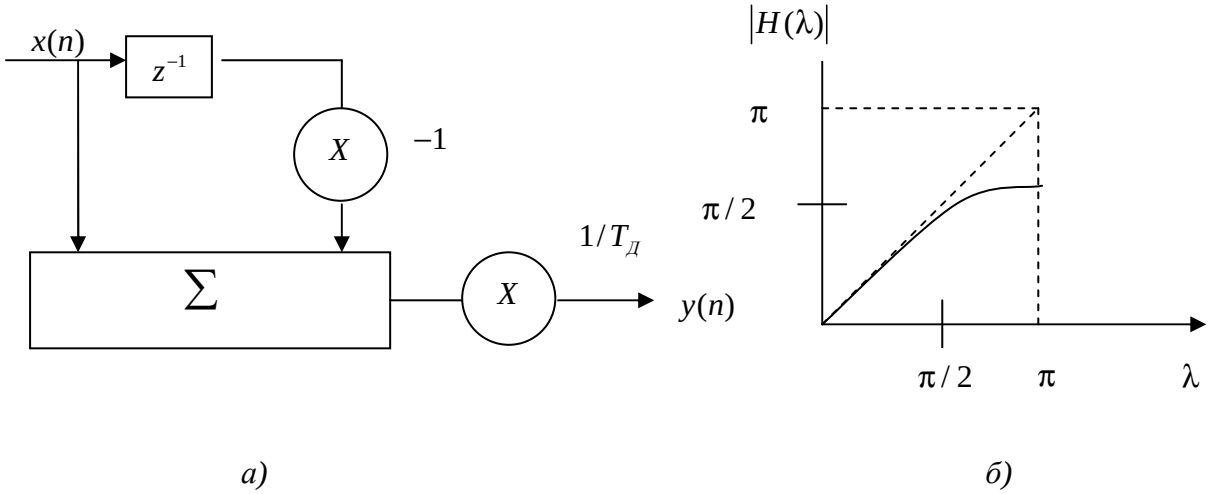
Ему отвечает передаточная функция

$$H(z) = (1/T_Д)(1 - z^{-1}) \tag{3.90}$$

с нулем в точке $z_0 = 1$, АЧХ и ФЧХ выражаются:

$$\left|H(j\lambda)\right| = (2/T_Д)\sin\lambda/2; \tag{3.91}$$

$$\varphi(\lambda) = (\pi/2) - (\lambda/2). \tag{3.92}$$

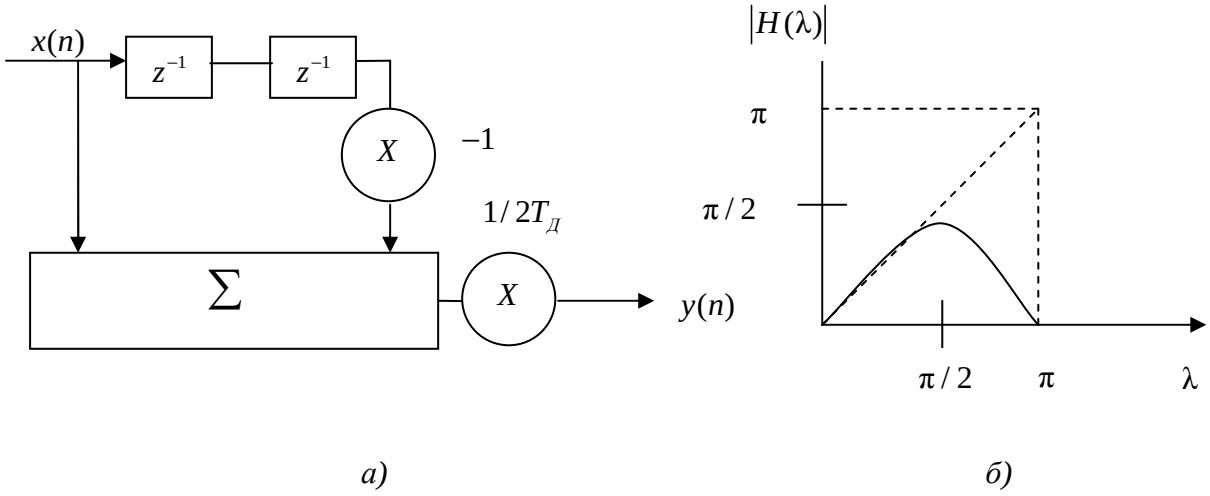


а) *б)*

Рис. 3.25. Структурная схема (*а*) и АЧХ (*б*) двухточечного дифференцирующего нерекурсивного фильтра

АЧХ фильтра практически линейна (представлена на рис. 3.25, *б*), т.е. близка к АЧХ идеального дифференциатора до значений $\lambda \approx \pi / 4$ или $f_Д / 8$. Данный алгоритм наряду с сигналом усиливает также высокочастотные шумы и помехи, которые могут присутствовать на входе фильтра.

Трехточечный алгоритм дифференцирования описывается уравнением вида $y(n) = (1/2T_Д)\left[x(n) - x(n-2)\right]$ и реализуется в соответствии со структурной схемой НФ – рис. 3.26, *а*.



а) *б)*

Рис. 3.26. Структурная схема (*а*) и АЧХ (*б*) трехточечного дифференцирующего нерекурсивного фильтра

Ему отвечает передаточная функция

$$H(z) = (1/2T_Д)(1 - z^{-2}) \qquad (3.93)$$

с нулями в точках $z_{01,2} = \pm 1$, АЧХ и ФЧХ выражаются:

$$\left|H(j\lambda)\right| = (1/T_Д)\sin\lambda; \qquad (3.94)$$

$$\varphi(\lambda) = (\pi/2) - \lambda. \qquad (3.95)$$

АЧХ фильтра практически линейна (см. рис. 3.26, *б*), т.е. близка к АЧХ идеального дифференциатора до значений $\lambda \approx \pi/6$ или $f_Д/12$.

В отличие от предыдущего, данный алгоритм подавляет возможные высокочастотные шумы и помехи на входе фильтра.

Алгоритм дифференцирования на основе полиномиальной аппроксимации по конечному набору данных позволяет получить сглаженные оценки первой и высших производных сигнала.

Например, в случае параболической 5-точечной аппроксимации дифференцирование осуществляется в соответствии с разностным уравнением

$$y_1(n) = (1/10)\left\{2\left[x(n+2) - x(n-2)\right] + \left[x(n+1) - x(n-1)\right]\right\},$$

которому отвечают передаточная функция, АЧХ и ФЧХ НФ:

$$H(z) = (1/10)\left[2 + z^{-1} - z^{-3} - 2z^{-4}\right];,$$

$$\left|H(j\lambda)\right| = (2/10)(\sin\lambda + 2\sin 2\lambda), \varphi(\lambda) = (\pi/2) - 2\lambda.$$

Предлагается самостоятельно проанализировать АЧХ данного фильтра.

С увеличением порядка фильтра (при $N = 7,9,11$) улучшаются точность аппроксимации производной и подавление высокочастотных шумов и помех.

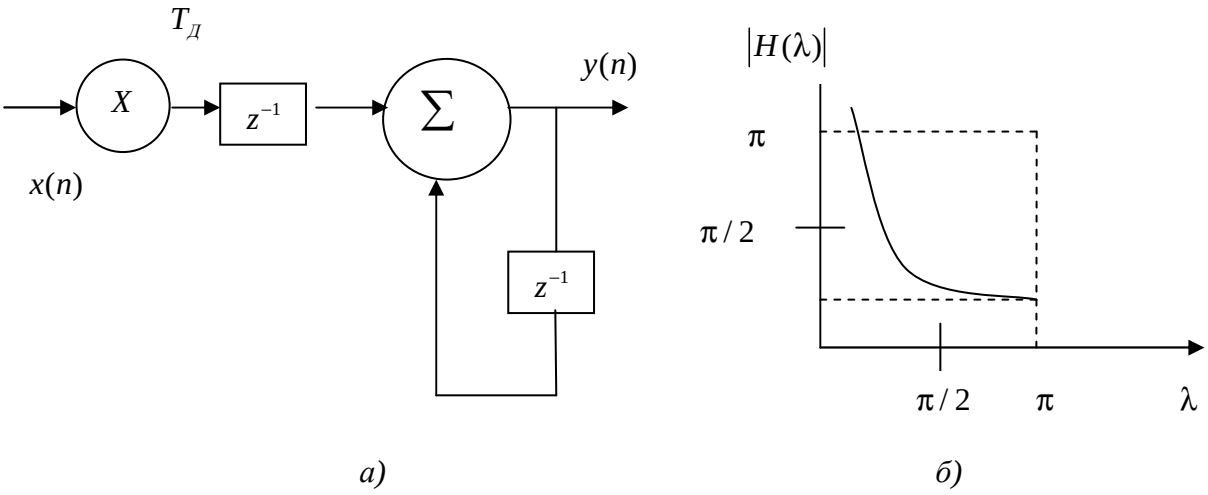
3.6.8. Простые алгоритмы цифрового интегрирования сигналов

С помощью цифровых фильтров реализуются простые алгоритмы приближенного цифрового интегрирования сигналов, такие, как алгоритм прямоугольников, трапеций и Симпсона.

Алгоритм прямоугольников определяется разностным уравнением

$$y(n) = T_Д x(n-1) + y(n-1), \qquad (3.96)$$

реализуемым рекурсивным ЦФ первого порядка (рис. 3.27, *а*).



а) *б)*

Рис. 3.27. Структурная схема (*а*) и АЧХ (*б*) рекурсивного фильтра, интегрирующего по алгоритму прямоугольников

Он описывается передаточной функцией, АЧХ и ФЧХ вида

$$H(z) = T_Д z^{-1}(1 - z^{-1}); \qquad (3.97)$$

$$|H(j\lambda)| = T_Д / 2\sin\lambda / 2; \qquad (3.98)$$

$$\varphi(\lambda) = -(\pi / 2) - (\lambda / 2). \qquad (3.99)$$

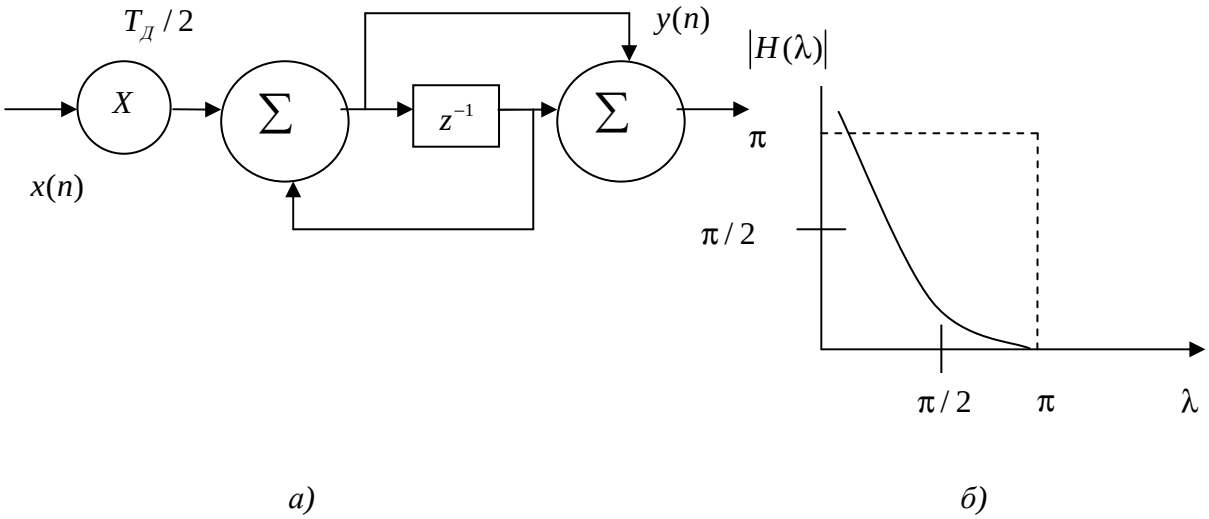
АЧХ фильтра представлена на рис. 3.27, *б*.

Фильтр имеет один нуль $z_0 = 0$ в начале координат и полюс $z_p = 1$ на единичной окружности. Фильтр потенциально неустойчив, что проявляется в нарастании со временем погрешности интегрирования.

Более точное интегрирование обеспечивает алгоритм трапеций, определямый разностным уравнением

$$y(n) = T_Д x(n-1) + y(n-1) + (T_Д / 2)\left[x(n) - n(n-1)\right], \qquad (3.100)$$

реализуемым рекурсивным ЦФ первого порядка (рис. 3.28, *а*).



а) *б)*

Рис. 3.28. Структурная схема (*а*) и АЧХ (*б*) рекурсивного фильтра, интегрирующего по правилу трапеций

Он описывается передаточной функцией, АЧХ и ФЧХ вида

$$H(z) = (T_Д / 2)(1 + z^{-1})(1 - z^{-1}); \qquad (3.101)$$

$$\left|H(j\lambda)\right| = (T_Д / 2)\mathrm{ctg}\lambda / 2; \qquad (3.102)$$

$$\varphi(\lambda) = -(\pi / 2). \qquad (3.103)$$

АЧХ фильтра представлена на рис. 3.28, *б*.

Фильтр имеет один нуль в точке $z_0 = -1$ и полюс $z_p = 1$ на единичной окружности. АЧХ фильтра достаточно близка к АЧХ идеального интегратора $(1/\lambda)$ на частотах менее $\lambda \approx \pi / 2$ или $f_Д / 4$. Вследствие потенциальной неустойчивости фильтра со временем происходит нарастание погрешности интегрирования.

Цифровое интегрирование по правилу Симпсона выполняется в соответствии с разностным уравнением

$$y(n) = y(n-2) + (T_Д / 3)\left[x(n) + 4x(n-1) + x(n-2)\right], \qquad (3.104)$$

реализуемым РФ второго порядка (рис. 3.29, *а*).

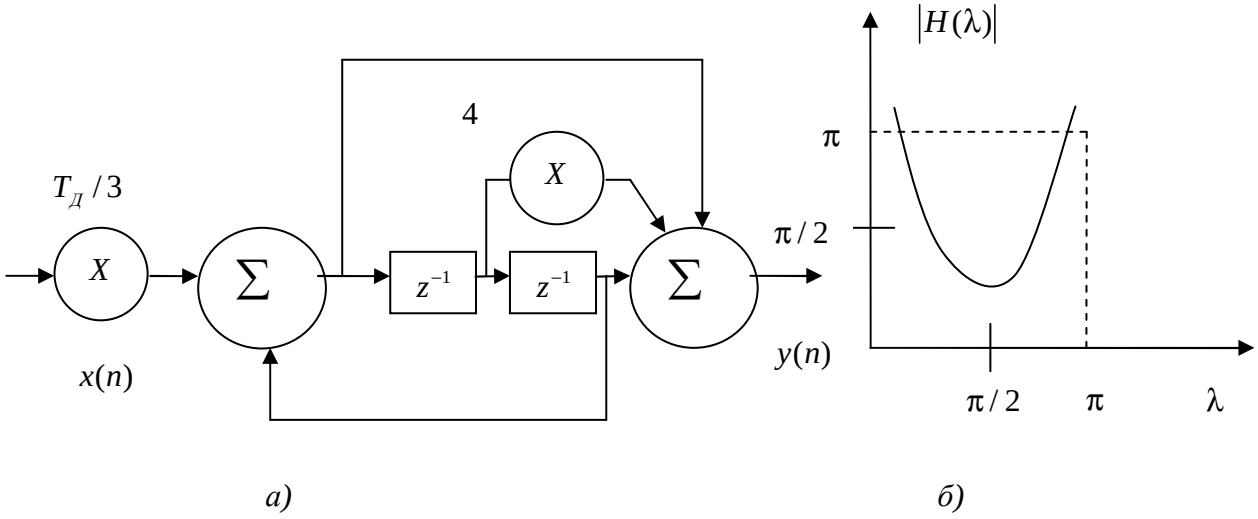
Он описывается передаточной функцией, АЧХ и ФЧХ вида

$$H(z) = (T_Д / 3)(1 + 4z^{-1} + z^{-2}) / (1 - z^{-2}); \qquad (3.105)$$

$$\left|H(j\lambda)\right| = (T_Д / 3)(2 + \cos\lambda) / \sin\lambda; \qquad (3.106)$$

$$\varphi(\lambda) = -(\pi / 2). \qquad (3.107)$$

АЧХ фильтра представлена на рис. 3.29, *б*.



а) *б)*

Рис. 3.29. Структурная схема (*а*) и АЧХ (*б*) рекурсивного фильтра, интегрирующего по правилу Симпсона

Фильтр имеет пару нулей $z_{01} = -0,27$, $z_{02} = -3,73$ и полюсов $z_{p1,2} = \pm 1$ на единичной окружности. Следствием потенциальной неустойчивости данного фильтра, как и рассмотренных выше, является ограниченное время непрерывного интегрирования и необходимость периодического приведения его в исходное состояние, т.е. сброса. АЧХ фильтра достаточно близка к АЧХ идеального интегратора $(1/\lambda)$ на частотах не выше $\lambda \approx \pi / 2$ или $f_Д / 4$. Однако такой интегратор усиливает высокочастотные шумы. Поэтому интегрирование зашумленных сигналов лучше выполнять по правилу трапеций.

3.6.9. Уменьшение частоты дискретизации (децимация) цифрового сигнала

Структура и описание однократной системы децимации [20]

Собственно операция децимации выполняется с помощью компрессора частоты дискретизации (КЧД), осуществляющего уменьшение частоты дискретизации в целое число раз *M*, рис. 3.30.

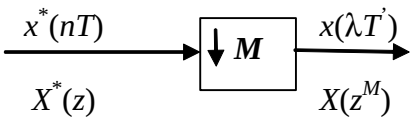


Рис. 3.30. Компрессор частоты дискретизации (КЧД)

Компрессор частоты дискретизации представляет собой ключ, который замыкается в моменты $t = nMT = \lambda T'$ ($n = 0, 1, 2\ldots$), т.е. из входного сигнала $x^*(nT)$ с интервалом дискретизации T берется только каждый M-й отсчет и формируется выходной сигнал $x(\lambda T') = x^*(\lambda MT)$ с интервалом дискретизации $T = MT$. Иными словами, выходная последовательность $x(\lambda T')$ КЧД формируется путем прореживания последовательности $x^*(nT)$ по алгоритму

$$x(\lambda T' + kT') = x^*(nT), \qquad (3.108)$$

где $\lambda = 0, 1, 2,\ldots$; $n = \lambda M + k$, а k – целое фиксированное число ($0 \le k \le M$). Операция, выполняемая КЧД, называется прореживанием, а последовательность $x(\lambda T')$ – прореженной. На рис. 3.31 показаны последовательности $x^*(nT)$ и $x(\lambda T')$ на входе и выходе КЧД при уменьшении частоты дискретизации в 4 раза ($M = 4$, $k = 2$).

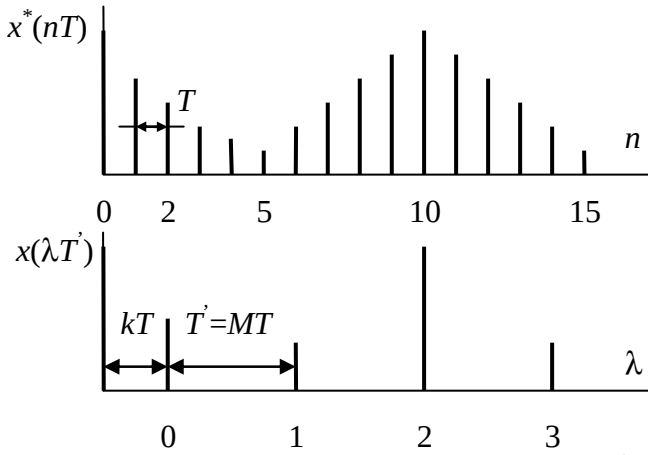


Рис. 3.31. Прореживание последовательности $x^*(nT)$

Простейшая система уменьшения частоты дискретизации (децимации) в целое число раз M представлена на рис. 3.32. Принцип работы схемы поясняется на рис. 3.33 (для $M = 4$). Предполагается, что спектр входного сигнала $x(nT)$ занимает полосу нормировочных частот [0; 0,5] (см. рис. 3.33, график 1). Задачей схемы является уменьшение частоты дискретизации сигнала в M раз с сохранением спектра, расположенного в полосе $[0, w_{\max}]$.

Собственно операция уменьшения частоты дискретизации в M раз осуществляется с помощью КЧД, формирующего сигнал $y(\lambda T') = y(\lambda MT)$ путем взятия только каждого M-го отсчета из последовательности $y^*(nT)$, т.е. $y(\lambda T') = y^*(M\lambda T)$, $\lambda = 0, 1, 2,\ldots$. Для предотвращения явления наложения

спектров операции, выполняемой КЧД, предшествует операция фильтрации децимируемого сигнала. Выходной сигнал $x(nT)$ обрабатывается фильтром, назначение которого состоит в подавлении составляющих спектра в частотных диапазонах $\left[\frac{r}{2M}, \frac{r+1}{2M}\right]$ (r = 1, 2,…, M – 1), которые при последующем уменьшении частоты дискретизации в M раз попадут в частотный диапазон [0; 1/(2M)].

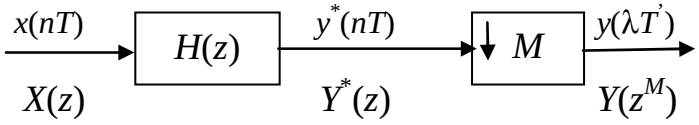


Рис. 3.32. Простейшая система децимации:
$x(nT)$ – входной цифровой сигнал; $y(\lambda T')$ – выходной децимированный сигнал;
$H(z)$ – КИХ-фильтр, M – КЧД

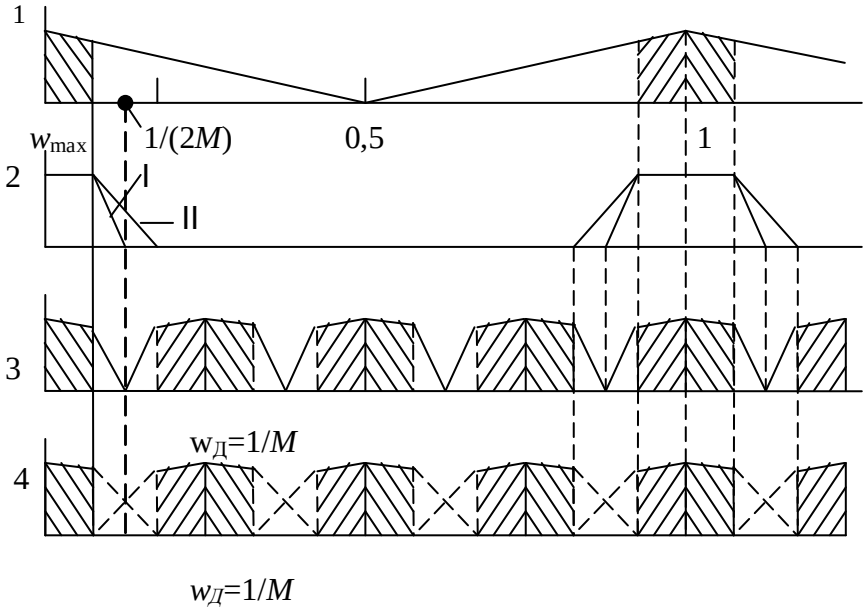


Рис. 3.33. Процесс децимации

Идеализированная АЧХ фильтра нижних частот схемы должна удовлетворять требованиям:

$$A(w) = \left|H(e^{j2\pi\varpi})\right| = \begin{cases} 1 \text{ при } w \in [0; w_{\max}]; \\ 0 \text{ при } w \in [0;0{,}5], \end{cases} \qquad (3.109)$$

где $w_{\max} < \theta \le \frac{1}{M} - w_{\max}$; $\theta = w_{г.з.}$ – граничная частота полосы задерживания фильтра.

АЧХ показана на рис.3.33 (график 2) для случаев $\theta = 1/2M$ (I, график 2, см. рис. 3.33) и $\theta = 1/2M - w_{\max}$ (II, график 2, см. рис. 3.33) ($1/10 < w_{\max} < 1/8$, $M = 4$).

Спектр сигнала $y^*(nT)$ на выходе фильтра равен:

$$Y^*(e^{j2\pi w}) = X(e^{j2\pi w})H(e^{j2\pi w}). \qquad (3.110)$$

Следовательно, спектр выходного сигнала $y(\lambda T')$ схемы децимации в основной полосе частот $w \in [0;0,5]$ (что соответствует частотному диапазону $w \in [0;1/(2M)]$)

$$Y(e^{j2\pi Mw}) = \frac{1}{M}\sum_{x=0}^{M-1} X(e^{j2\pi(w+\frac{x}{M})})H(e^{j2\pi(w+\frac{x}{M})}). \qquad (3.111)$$

Если АЧХ фильтра $\left|H(e^{j2\pi w})\right|$ удовлетворяет условию (3.109), то в полосе частот $[0,w_{\max}]$ спектр выходного сигнала равен спектру входного сигнала. В полосе $[w_{\max},1/2M]$ может отсутствовать наложение спектров (при $\theta \le 1/(2M)$), см. рис. 3.33 (график 3) либо могут иметь место наложения спектров (при $1/(2M) < \theta \le 1/M - w_{\max}$, см. рис. 3.33, график 4).

Оба случая допустимы, поскольку от схем требуется только сохранение спектра в полосе $[0,w_{\max}]$.

В реальных фильтрах, используемых для децимации, АЧХ аппроксимирует (3.109) с определенной степенью точности. В полосе пропускания АЧХ имеет неравномерность $\Delta A_П$, а в полосе задерживания – отклонение от нуля $\Delta A_З$. При уменьшении частоты дискретизации имеет место наложение спектров. Спектр выходного сигнала определяется (3.111). Первое слагаемое в правой части (3.111) при $\chi = 0$ для $w \in [0,w_{\max}]$ можно рассматривать как спектр полезного сигнала, равный спектру входного сигнала в данной полосе измененного в соответствии с АЧХ фильтра в полосе пропускания. Слагаемые для $\chi = 1, 2, \ldots, M-1$ и $|w| \in [0,w_{\max}]$ следует рассматривать как спектры помех, искажающие спектр полезного сигнала в полосе $[0,w_{\max}]$.

Выбор величин $\Delta A_П$ и $\Delta A_З$ при решении аппроксимационной задачи основывается на требованиях конкретной проектируемой системы и аналогичен выбору соответствующих величин в системах интерполяции. Требования к АЧХ (3.109) могут быть заметно смягчены, если $w_{\max} << 1/(2M)$:

$$A(w) \approx \begin{cases} 1 \text{ при } w \in [0,w_{\max}]; \\ 0 \text{ при } w \in \left[\frac{r}{M} - w_{\max}, \frac{r}{M} + w_{\max}\right], r = 1,2\ldots,\left[\frac{M}{2}\right]. \end{cases} \qquad (3.112)$$

Рассмотрим теперь особенности использования КИХ- и БИХ-фильтров при децимации. Передаточная функция $H(z)$ фильтра (как КИХ, так и БИХ) и его частотная характеристика определяются «высокой» (входной) частотой дискретизации.

Однако КИХ-фильтр работает фактически на «низкой» (выходной) частоте, поскольку нет необходимости рассчитывать $M - 1$ отсчет выходной последовательности $y(nT)$ фильтра (см. рис. 3.32), которые будут отброшены КЧД.

БИХ-фильтр, используемый в схеме децимации, работает на «высокой» (входной) частоте дискретизации, поскольку при вычислении любого отсчета последовательности $y^*(nT)$ необходимо иметь значения всех предыдущих отсчетов (и тех, которые далее будут отброшены КЧД).

Использование БИХ-фильтра может оказаться более предпочтительным при минимизации емкости оперативной памяти или объема оборудования.

Число операций умножения в единицу времени для БИХ- ($V_Б$) и КИХ-фильтров ($V_к$) равно:

$$V_Б = (N_Б + M_Б - 1)f_Д; \quad V_к = N_к(f_Д/M), \tag{3.113}$$

где $N_Б$ и $M_Б$ – количество коэффициентов в числителе и знаменателе передаточной функции БИХ-фильтра; $N_к$ – количество коэффициентов передаточной функции КИХ-фильтра; $f_Д$ – частота дискретизации входного сигнала; M – коэффициент децимации.

Применение БИХ-фильтра оказывается предпочтительным (по критерию минимума операций умножения в единицу времени) при условии

$$N_к > M(N_Б + M_Б - 1). \tag{3.114}$$

При требовании сохранения фазовых соотношений между составляющими спектра входного сигнала в полосе частот $[0, w_{\max}]$ в системе децимации целесообразно использовать КИХ-фильтр с линейной фазовой характеристикой.

Рассмотрим структуры систем децимации с целочисленным коэффициентом M при использовании в схеме КИХ-фильтра с передаточной функцией

$$H(z) = \sum_{l=0}^{N-1} b_l z^{-l}. \tag{3.115}$$

Алгоритм работы схемы описывается разностным уравнением

$$y(\lambda T') = y^*(\lambda MT) = \sum_{l=0}^{N-1} b_l x(\lambda MT - lT), \lambda = 0,1,2,... \qquad (3.116)$$

Структура 1. Уравнению (3.116) соответствует схема, показанная на рис. 3.34. Входная последовательность $x(nT)$ поступает в цепь из $N-1$ элемента задержки на интервал дискретизации T. Выходы элементов задержки подключены к умножителям на коэффициенты b_l через КЧД. Компрессоры частоты дискретизации работают как ключи, замыкающиеся в моменты $t = \lambda T' = \lambda MT$, т.е. в моменты 0, MT, $2MT$ и т.д. Полученные произведения последовательно складываются в сумматорах на два входа, образуя отсчет выходного сигнала $y(\lambda T')$. Нужно отметить, что умножители в этой схеме работаю на «низкой» (выходной) частоте дискретизации.

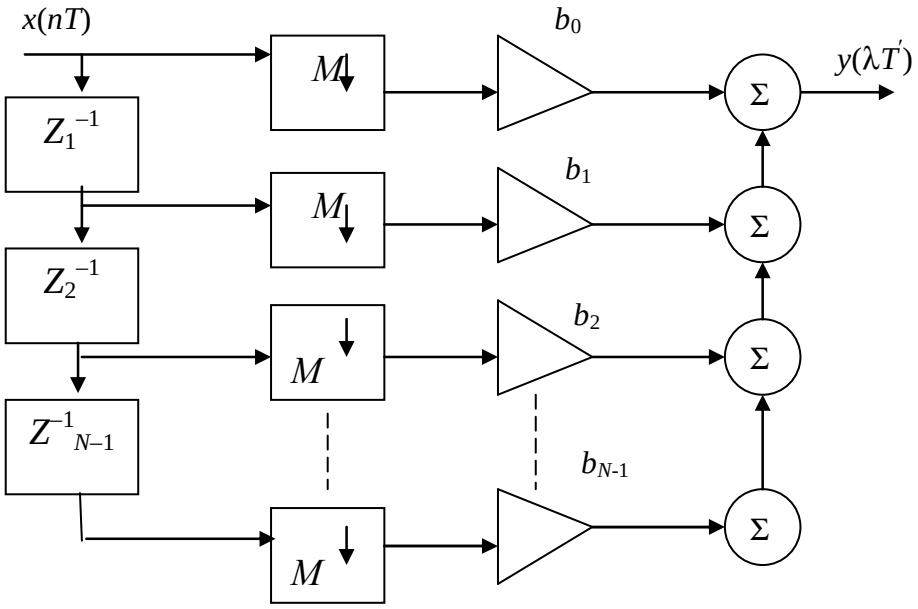


Рис. 3.34. Структура децимации

Структура 2 (полифазная). Полифазная структура основана на преобразовании уравнения (3.116) к виду

$$y(\lambda T') = \sum_{k=0}^{M-1} \sum_{r=0}^{G-1} b_{rM+k} x((\lambda - r)MT - kT), \qquad (3.117)$$

причем $N = GM$, G – целое.

Уравнение (3.117), описывающее полифазную структуру децимации сигнала с целочисленным коэффициентом M, можно интерпретировать следующим образом: выходная последовательность $y(\lambda T')$ схемы есть сумма M последовательностей $y_k(kMT)$, $k = 0, 1,...,M - 1$, каждая из которых, в свою очередь, – результат фильтрации последовательности $x_k^*(kMT) = x(\lambda MT - kT)$ дискретным фильтром с передаточной функцией $H_k^*(z^M)$ и импульсной характеристикой $b^k_r = b_{rM+k}$, причем отсчеты импульсной характеристики k-го фильтра – это отсчеты импульсной характеристики b_l, взятые через $M - 1$ отсчет: $b_r^k = b_{rM} + {}_k$, $k = 0, 1,..., M - 1$; $r = 0, 1,2,...$

Полифазная структура при децимации сигнала содержит (как и при интерполяции) M параллельных ветвей обработки, в каждой из которых находится фильтр, работающий на «низкой» (выходной) частоте дискретизации.

3.6.10. Увеличение частоты дискретизации (интерполяция) цифрового сигнала

До сих пор мы рассматривали алгоритмы и устройства цифровой обработки сигналов при определенной (фиксированной) частоте дискретизации $f_Д$ и соответствующем интервале дискретизации $T = 1/f_Д$. Вместе с тем в современных системах связи и радиотехники часто используются устройства и каналы связи с различными частотами дискретизации. Так, в современном цифровом оборудовании радиодомов и телецентров приняты следующие стандарты на частоту дискретизации: для обработки сигнала – 48 кГц; для передачи по каналу связи – 32 кГц; для лазерного проигрывателя – 44,1 кГц. Для обеспечения совместной работы различных источников сигнала, системы обработки и каналов связи необходимо осуществить сопряжения частот дискретизации, т.е. преобразование сигнала с частотой дискретизации $f_Д = 1/T$ в частоту $f'_Д = 1/T'$. Аналогичная задача возникает в технике многоканальной связи (преобразование стандартной частоты дискретизации 60-канальной группы с частотным разделением каналов, равной 576 кГц, в частоту 12 кГц для передачи по каналу связи) [20].

Процесс преобразования цифрового сигнала от более низкой частоты дискретизации к более высокой традиционно называют интерполяцией цифрового сигнала. Процесс преобразования сигнала от более высокой частоты дискретизации к более низкой называют децимацией цифрового сигнала.

Появившиеся как средство сопряжения систем с различными частотами дискретизации алгоритмы преобразования частоты дискретизации (ПЧД) стали в дальнейшем использоваться и при построении эффективных в смысле вычислительных затрат многоскоростных систем обработки цифровых сигналов, в которых различные этапы обработки выполняются на различных частотах дискретизации.

В математике задачей интерполирования решетчатой функции является задача построения решетчатой функции $y(nT)$ с интервалом дискретизации T по конечному множеству $\{x(0), x(T'), x(2T'),\ldots, x(kT')\}$ известных значений решетчатой функции $x(vT')= x(vLT)$, $T'= LT$, такой, что в заданных точках $y(nT) = y(LvT) = x(vT')$, $v = 0, 1,...,k$, а в остальных точках ($nT = 0, LT,2LT,..,kLT$) функция $y(nT)$ приблизительно равна функции $x(t)$, из которой образована исходная решетчатая функция $x(vT')$.

Иными словами, по заданным отсчетам исходной решетчатой функции $x(vT')$ с относительно большим интервалом дискретизации T' строится решетчатая функция $y(nT)$ с интервалом дискретизации T, в L раз меньшим исходного ($T – T'/L$), причем в исходных точках $\{0, T', 2T',...\}$ или $(0, LT, 2LT,...\}$ отсчеты обеих функций совпадают ($y(nT) – y(LvT) = x(vLT)$, $v = 0, 1,...,k$. Таким образом, если исходная решетчатая функция $x(vT')$ содержала $k + 1$ отсчет, то полученная решетчатая функция $y(nT)$ содержит $kL + 1$ отсчет (между каждой парой отсчетов $x(lt)$ и $x((l+1)T)$ добавляется $L – 1$ отсчет).

Интерполяция – процесс цифровой обработки сигналов, приводящий к формированию сигнала $y(nT)$ с повышенной частотой дискретизации из сигнала $x(vT') = x(vLT)$ с более низкой частотой дискретизации при определенных ограничениях на временные и спектральные изменения исходного сигнала.

Можно выделить три разновидности процесса интерполяции при ЦОС:

1) увеличение частоты дискретизации осуществляется в соответствии с математическим понятием интерполяции;

2) при увеличении частоты дискретизации исходные отсчеты дискретного сигнала $x(vT')$ оказываются утерянными, однако отсчеты выходного сигнала $y(nT)$ могут рассматриваться как отсчеты исходного аналогового сигнала $x(t)$, из которого путем дискретизации с интервалом T' образован исходный дискретный сигнал $x(vT')$. В этом случае форма огибающей сигналов $x(vT')$ и $y(nT)$ и их спектр не изменяются;

3) увеличение частоты дискретизации приводит к изменению формы интерполируемого сигнала, однако модуль спектра не меняется.

Чтобы пояснить эти разновидности процесса интерполяции при ЦОС, рассмотрим схему, показанную на рис. 3.35. Входной аналоговый сигнал $x(t)$ поступает на идеальный дискретизатор Д, работающий с интервалом дискретизации $T' = LT$ (L – целое число). Выходной дискретный сигнал дискретизатора $x(vT')$, $v = 0,1,2,\ldots n$, поступает на идеальный интерполятор ИИ, увеличивающий частоту дискретизации в целое число раз L.

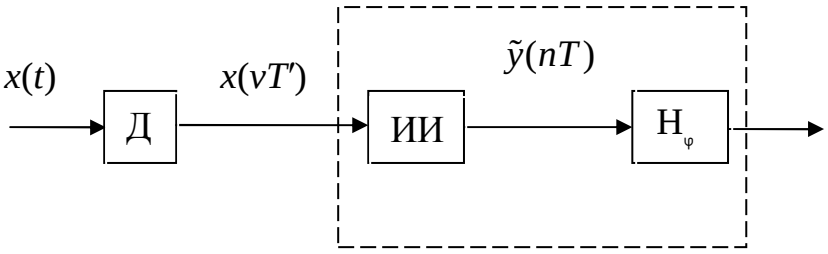


Рис. 3.35. Схема интерполяции

Выходной сигнал схемы $y(nT)$, $n = 0,1,2,\ldots n$, получается в результате преобразования выходного сигнала ИИ $\tilde{y}(nT)$ линейной дискретной системой $H\varphi$ с частотной характеристикой

$$H_{\varphi}(e^{j2\pi\omega}) = e^{-jR(\omega)2\pi\omega}. \qquad (3.118)$$

Иными словами, АЧХ системы $\left|H_{\varphi}(\cdot) = 1\right|$ (т.е. форма спектра не изменяется).

Процесс интерполяции осуществляется путем цифровой фильтрации с целью подавления «лишних» составляющих спектра. Однако цифровой фильтр работает на определенной частоте дискретизации. Следовательно, перед цифровой фильтрацией сигнала необходимо осуществить предварительное увеличение частоты дискретизации исходного сигнала в нужное число раз L. Эту операцию осуществляет экспандер частоты дискретизации (ЭЧД).

Условное изображение ЭЧД, осуществляющего предварительное увеличение частоты дискретизации в целое число раз L, показано на рис. 3.36. Входным сигналом ЭЧД является дискретный сигнал с интервалом дискретизация T' (частотой дискретизации $f'_{Д}$), описываемый решетчатой функцией $x(vT')$, $v = 0, 1, 2. \ldots$ Сигнал $x(vT')$ преобразуется по алгоритму

$$x^{*}(nT) = \begin{cases} x(\frac{n}{L}T') \text{при } n=0,L,2L,\ldots; \\ 0 \text{ при других } n. \end{cases} \qquad (3.119)$$

Таким образом формируется выходной дискретный сигнал с интервалом дискретизации $T = T'/L$, в L раз меньшим исходного (частотой дискретизации $f_д = 1/T$, в L раз большей исходной), описываемый решетчатой функцией $x^*(nT) = x^*(nT'/L)$, $n = 0, 1, 2,\ldots$

Из (3.119) видно, что последовательность $x^*(nT)$ получается из последовательности $x(vT')$ путем ввода $L - 1$ нулевого отсчета между каждой парой отсчетов исходной последовательности. На рис. 3.37 показаны последовательности $x(vT')$ и $x^*(nT)$ на входе и выходе ЭЧД при увеличении частоты дискретизации в $L = 3$.

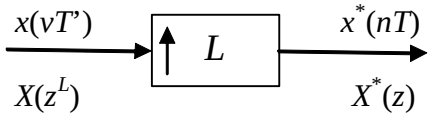


Рис. 3.36. Увеличение частоты дискретизации

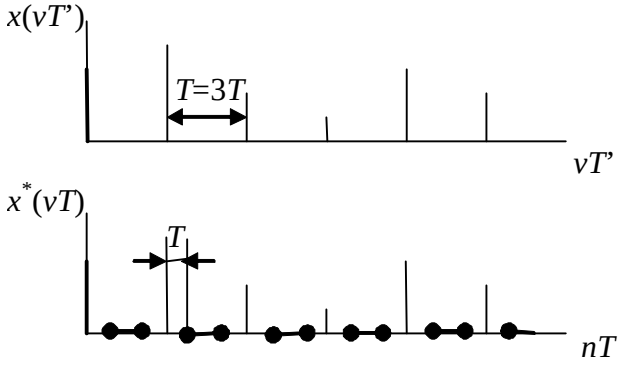


Рис. 3.37. Процесс интерполяции

Перейдем теперь к вопросу о спектральных изменениях сигнала в ЭЧД.

Запишем вначале выражения для z-преобразований входной и выходной последовательностей:

$$X(z') = \sum_{v=0}^{\infty} x(vT')(z')^{-v}, \tag{3.120}$$

где $z' = \exp(j\omega T')$,

$$X^*(z) = \sum_{n=0}^{\infty} x^*(nT)z^{-n}, \tag{3.121}$$

где $z = \exp(j\omega T)$.

Поскольку $T' = LT$, очевидно, что $z' = z^L$. Преобразуем теперь (3.121), учтя, что $x^*(nT) = 0$ при $n \neq 0, L, 2L,\ldots$:

$$X^*(z) = \sum_{k=0}^{\infty} x^*(kLT)z^{-kL}. \tag{3.122}$$

Если учесть, что $LT = T'$, а $z^L = z'$ и, следовательно, $z^{-kL} = (z')^{-k}$, то, сравнив (3.122) с (3.120), получим:

$$X^*(z) = X(z'). \qquad (3.123)$$

Это означает, что z-преобразование входного и выходного сигналов ЭЧД тождественны.

Соотношение для спектров входного и выходного сигналов ЭЧД можно получить из (3.123), если подставить $z' = \exp(j\omega T') = e^{j\omega LT}$ и $z = e^{j\omega T}$:

$$X^*(e^{j\omega T}) = X(e^{j\omega LT}), \qquad (3.124)$$

или

$$X^*(e^{j2\pi w}) = X(e^{j2\pi Lw}), \qquad (3.125)$$

где $w = \omega/\omega_Д = \omega T/(2\pi)$ – нормированная частота.

Таким образом, выходной сигнал ЭЧД $x^*(nT)$, формируемый из входного сигнала $x(vT')$ по алгоритму (3.119), имеет тот же спектр, что и входной сигнал. На рис. 3.38 условно показаны модули спектров входного и выходного сигналов ЭЧД при $L = 3$.

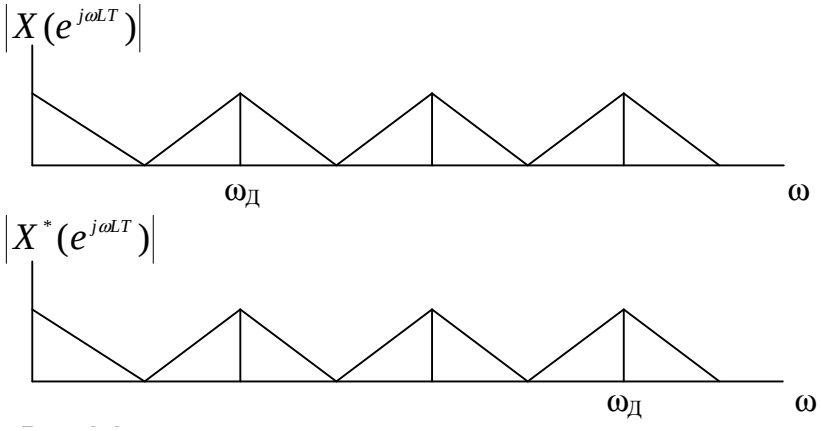


Рис. 3.38. Модули спектров входного
и выходного сигналов ЭЧД при $L = 3$

Простейшая схема повышения частоты дискретизации (интерполяции) дискретного сигнала в целое число раз (с целочисленным коэффициентом L) показана на рис. 3.39. Входным сигналом схемы является сигнал $x(vT')$, $v = 0, 1,2,...$, с интервалом дискретизации T'. Предполагается, что сигнал $x(vT')$ с финитным спектром $X(j\omega)$, занимающим полосу частот $\omega \in [0, \omega_{max}]$. Частота дискретизации $f'_Д \geq \omega_{max}/\pi$. На рис. 3.40 (графики 1 и 2) показаны модули спектров сигналов $x(t)$ и $x(vT')$ на нормированной оси частот ($w = \omega/\omega_Д = \omega/(L\omega'_Д)$) для случая $L = 3$ ($\omega_Д = 3\omega'_Д = 6\omega_{max}$).

Входной сигнал $x(vT')$ поступает на экспандер частоты дискретизации, осуществляющий предварительное увеличение частоты дискретизации в L раз по алгоритму (3.119). Спектры выходного и входного сигналов ЭЧД равны и периодичны с частотой $w'_Д$ = l/L. Связь спектров сигналов $x^*(nT)$ и $x(vT')$ со спектром $X(j\omega)$ исходного аналогового сигнала $x(t)$ определяется соотношением

$$X^*(e^{j2\pi w}) = X(e^{j2\pi Lw}) = \frac{1}{T'}\sum_{k=-\infty}^{\infty} X(jw + jk\frac{1}{L}), \qquad (3.126)$$

т.е. периодический спектр дискретных сигналов $x^*(nT)$ и $x(vT')$ представляет собой сумму спектральных компонент (спектров) исходного аналогового сигнала, расположенных симметрично относительно центральных частот k/L. Модуль спектров $X^*(e^{j2\pi w})$ и $X(e^{j2\pi Lw})$ показан на рис. 3.40 (график 2). Следует обратить внимание на коэффициент $1/T'$ перед знаком суммы в правой части (3.119).

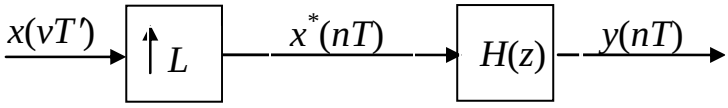


Рис. 3.39. Простейшая схема повышения частоты дискретизации

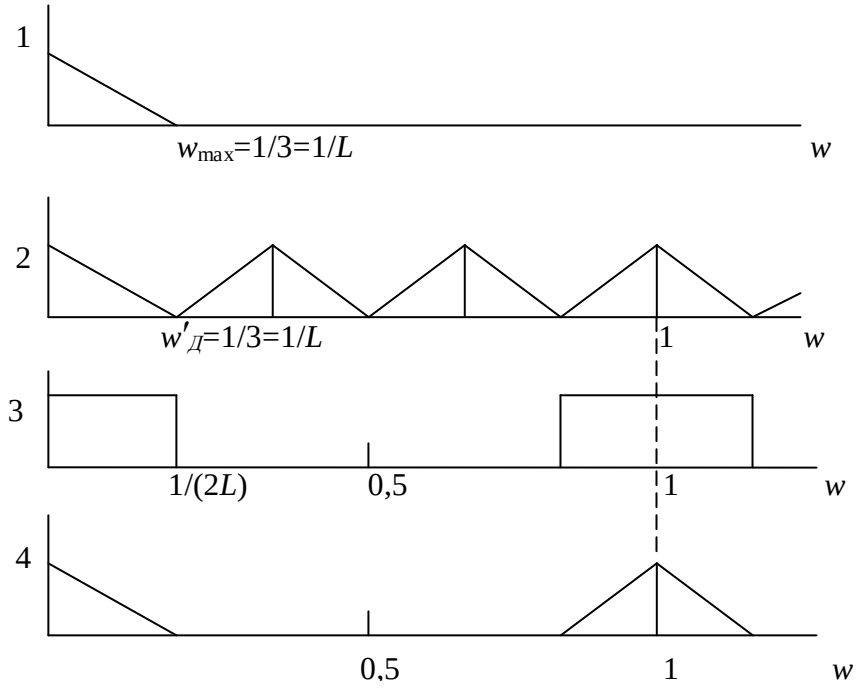


Рис. 3.40. Спектры выходного и входного сигналов ЭЧД

Выходной сигнал ЭЧД обрабатывается «идеальным» фильтром нижних частот с передаточной функцией $H(z)$ и импульсной характеристикой $h(nT)$, задачей которого является подавление «лишних» частотных составляющих спектра $X^*(e^{j2\pi w})$, занимающих область частот $w\in$ [1/(2L); 0,5]. Амплитудно-частотная характеристика фильтра показана на рис. 3.40 (график 3). Полоса пропускания «идеального» фильтра занимает область частот [0; 1/(2L)], полоса задерживания занимает область частот [1/(2L); 0,5]. Коэффициент усиления фильтра в полосе пропускания равен коэффициенту интерполяции, т.е. АЧХ «идеального» ФНЧ должна удовлетворять требованиям:

$$\left|H(e^{j2\pi w})\right| = \begin{cases} L & \text{при} \quad w\in[0,1/(2L)]; \\ 0 & \text{при} \quad w\in[1/(2L);0,5]. \end{cases} \tag{3.127}$$

Поясним формулу (3.127). Требуется получить сигнал $y(nT)$, который можно было бы рассматривать как результат дискретизации аналогового сигнала $x(t)$ с высокой частотой ω_∂. Спектр же такого сигнала $\tilde{y}(nT)$, получаемого путем непосредственной дискретизации сигнала $x(t)$, связан со спектром сигнала $x(t)$ соотношением

$$\tilde{Y}(e^{j2\pi w}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X(jw + jk). \tag{3.128}$$

Если сравнить коэффициенты перед знаком суммы в (3.128) и (3.126), очевидно, что для получения требуемой амплитуды спектра необходимо иметь фильтр, удовлетворяющий (3.127).

Итак, фильтр должен иметь коэффициент усиления L в полосе пропускания, определяемой шириной спектра исходного сигнала, и подавлять «лишние» частотные составляющие спектра, лежащие в диапазоне частот от l/(2L) (граничной частоты основного спектра исходного сигнала) до 0,5.

Запишем теперь выражение для z-преобразования выходного сигнала:

$$Y(z) = X^*(z)H(z) = X(z^L)H(z). \tag{3.129}$$

Выражение для спектра выходного сигнала можно получить из (3.129), подставляя $z = e^{j2\pi w}$:

$$\begin{aligned} Y(e^{j2\pi w}) = X^*(e^{j2\pi w})H(e^{j2\pi w}) = X(e^{j2\pi Lw})H(e^{j2\pi w}) = \\ = X(e^{j2\pi Lw})\left|H(e^{j2\pi w})\right|e^{j\varphi(w)}. \end{aligned} \tag{3.130}$$

Следовательно, в рассматриваемом идеализированном случае (при АЧХ фильтра, удовлетворяющей условиям (3.127) спектр сигнала на выходе схемы в основной полосе частот [0; 0,5] с учетом (3.126) будет иметь вид

$$Y(e^{j2\pi w}) = \frac{1}{T} X(jw) e^{j\varphi(w)}, \qquad (3.131)$$

т.е. спектр выходного сигнала представляет собой спектр исходного аналогового сигнала, измененный в соответствии с фазовой характеристикой фильтра $\varphi(w)$.

Поскольку спектр сигнала $\tilde{y}(nT)$, получаемого путем непосредственной дискретизации $x(t)$, определяется в основной полосе частот [0;0,5] соотношением, получаемым из (3.128) при $k = 0$,

$$\tilde{Y}(e^{j2\pi w}) = \frac{1}{T} X(jw), \qquad (3.132)$$

очевидно, что

$$\tilde{Y}(e^{j2\pi w}) = Y(e^{j2\pi w}) e^{j\varphi(w)}. \qquad (3.133)$$

Из выражений (3.132) и (3.131) можно сделать следующие выводы:

а) спектр сигнала $y(nT)$ на выходе схемы можно рассматривать как спектр сигнала $\tilde{y}(nT')$, полученного путем непосредственной дискретизации сигнала с частотой ω_∂, но измененного в соответствии с фазовой характеристикой ФНЧ $\varphi(w)$;

б) схему повышения частоты дискретизации (интерполяции), содержащую ЭЧД и «идеальный» фильтр с АЧХ, удовлетворяющую условию (3.127), можно рассматривать как совокупность идеального интерполятора ИИ и линейной системы с частотной характеристикой $H_\varphi(e^{j2\pi w}) = e^{j\varphi(w)}$;

в) форма и спектр выходного сигнала схемы интерполяции существенно зависят от типа используемого фильтра и его ФЧХ.

Остановимся более подробно на последнем выводе. В качестве ФНЧ могут использоваться как КИХ-, так и БИХ-фильтры. Рассмотрим вначале особенности использования КИХ-фильтров, тем более что конечная импульсная характеристика последних позволяет достаточно просто строить эффективные структуры систем преобразования частоты дискретизации сигналов. КИХ-фильтры с передаточной функцией

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n} \qquad (3.134)$$

и линейностью ФЧХ $\varphi(w) = -R2\pi w$ могут иметь нечетное (КИХ-фильтры вида 1) и четное (КИХ-фильтры вида 2) числа отсчетов N импульсной характеристики $h(nT) = b_n$.

Для КИХ-фильтров вида 1 (N – нечетное) $R = (N - 1)/2$ – целое число. Поскольку величина R определяет задержку во временной области, вносимую фильтром при обработке сигнала, для фильтров вида 1 задержка равна целому числу $(N - 1)/2$ интервалов дискретизации T. Таким образом, при интерполяции сигнала схема (рис. 3.39), содержащая КИХ-филътр вида 1, сохраняет как вид модуля спектра, так и форму исходного сигнала.

Для КИХ-фильтров вида 2(N – четное) $R = (N - 1)/2$ – нецелое число. Процесс интерполяции в этом случае соответствует варианту 2 интерполяционного процесса, схема (рис. 3.39), содержащая КИХ-фильтр вида 2, сохраняет как вид модуля спектра, так и форму исходного сигнала, однако отсчеты последнего не сохраняются.

Минимально-фазовые КИХ-фильтры обладают нелинейной ФЧХ $\varphi(w) = R(w)2\pi w$.

БИХ-фильтры обладают, как правило, нелинейной ФЧХ. В случае использования БИХ-фильтра в схеме на рис. 3.39 также сохраняется вид модуля спектра, но не сохраняется форма исходного сигнала.

3.7. Непараметрические методы спектрального и корреляционного анализа

3.7.1. Параметры анализаторов спектра

Спектральный анализ заключается в разложении сигнала на его частотные или спектральные составляющие и оценке или измерении их спектральных характеристик − амплитуды, фазы, мощности, спектральной плотности мощности и др. [20].

К задачам, решаемым методами спектрального анализа, относятся: обнаружение, разрешение и оценивание параметров сигналов, сжатие данных, выделение информативных признаков, идентификация объектов (определение частотных, импульсных и других характеристик), распознавание образов (речи, изображений) и др. Для случайных сигналов с помощью спектрального анализа решается общая задача выявления скрытых периодичностей и статистических (корреляционных) связей. Спектральный анализ детерминированных периодических (регулярных) сигналов и сигналов конечной длительности называют также гармоническим анализом.

Основными методами спектрального анализа являются фильтровые (методы полосового анализа), бесфильтровые (основанные на ДПФ), параметрические (на основе параметрических моделей случайных процессов), текущего, скользящего и скачущего спектрального анализа, последовательного и параллельного, одноканального и многоканального, в реальном времени и нереальном (по записям сигнала).

К параметрам анализаторов спектра относятся число каналов анализа; время наблюдения или анализа (ширина окна) $T_a = NT_Д$ и соответствующее ему число отсчетов или длина N обрабатываемой реализации сигнала; полоса анализа Δf_a, не превышающая для дискретных сигналов основной полосы спектра $\pm f_Д / 2$; разрешение по частоте, обратно пропорциональное времени анализа, $\Delta f_p \sim 1/T_a$, и соответствующее разности частот двух соседних разрешаемых (разделяемых) частотных составляющих сигнала.

Спектральный анализ широко применяется в радиотехнике и смежных с нею областях.

Особенности классического спектрального анализа на основе ДПФ связаны с оценкой спектра сигнала по его реализациям конечной длины, т.е. на конечном интервале наблюдения. При этом полагается, что за пределами этого интервала сигнал равен нулю или является периодическим продолжением считанной реализации. Широкое применение анализаторов спектра на основе ДПФ обусловлено наличием высокоэффективных вычислительных алгоритмов БПФ.

3.7.2. Базовая структура анализатора спектра на основе ДПФ и БПФ

В основе анализаторов спектра, использующих ДПФ, лежит базовая структура, приведенная на рис 3.41. Она реализует базовые операции анализатора спектра – взвешивание и вычисление ДПФ. Ее выходом является вектор ДПФ входной в общем случае не ограниченной по длине последовательности $x(n)$, усеченной весовой функцией $w(n)$ конечной длины N:

$$\widetilde{X}(j\omega_k) = \sum_{n=0}^{N-1} x(n)w(n)e^{-j\omega_k nT_Д} = \sum_{n=0}^{N-1} \tilde{x}(n)e^{-j(2\pi/N)kn} = \text{ДПФ}_N[\tilde{x}(n)], \quad (3.135)$$

$$k = 0,1,\ldots,N-1.$$

Здесь $\tilde{x}(n) = x(n)w(n)$, $n = 0,1,...,N-1$ – преобразуемая входная последовательность ДПФ; $\omega_k = k\omega_Д / N$ или $f_k = kf_Д / N$ – частоты анализа, называемые также бинами ДПФ – 1 бин равен шагу дискретизации сигнала в частотной области $f_Д / N$. Анализатор имеет N разнесенных по частоте на 1 бин ($f_Д / N$) каналов анализа с центральными частотами $\omega_k(f_k)$, при этом значения $k = 0,1,...,N-1$ соответствуют номеру канала, номеру бина или номеру частотной выборки ДПФ $\widetilde{X}(j\omega_k) = \widetilde{X}(jk)$. Весовая функция представляет окно, через которое наблюдается входной сигнал, длиной ее определяется время анализа $T_a = NT_Д$ или время наблюдения сигнала.

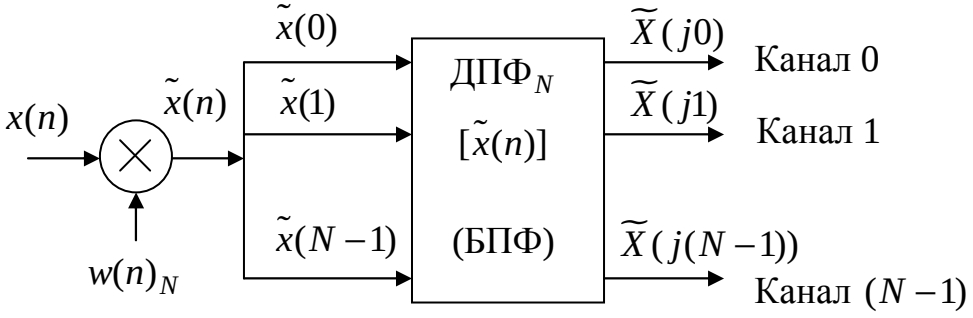


Рис. 3.41. Структурная схема анализатора спектра на основе ДПФ, представленная его базовыми элементами

Умножению или взвешиванию во временной области соответствует свертка в частотной, поэтому вычисляемое ДПФ фактически является дискретизированной сверткой истинного спектра анализируемого сигнала $X(j\omega)$ с частотной характеристикой (спектром) весовой функции $W(j\omega)$: $\widetilde{X}(j\omega_k) = X(j\omega) * W(j\omega)|\omega = \omega_k$, где * – символ свертки, т.е. содержит систематическую (методическую) погрешность анализа. Она является следствием ограничения сигнала по длительности, искажающего результаты спектрального анализа. Применение специальных весовых функций или окон позволяет сгладить или ослабить вызываемое временным усечением влияние или эффект разрывов сигнала на краях. Дальнейшая обработка выходных данных ДПФ осуществляется с учетом измеряемых или оцениваемых с помощью ДПФ спектральных характеристик, зависящих от вида анализируемых сигналов.

Дальнейшая обработка выходных данных ДПФ осуществляется с учетом измеряемых или оцениваемых с помощью ДПФ спектральных характеристик, зависящих от вида анализируемых сигналов.

Для периодических сигналов $x_p(n)_N$ с периодом $NT_Д$ оценивают амплитуды $A_m(\omega_i)$ и фазы $\varphi(\omega_i)$ гармоник с частотой $if_Д / N$ или их средние за период мощности $[A_m(\omega_i)]^2 / 2$.

Для детерминированных сигналов конечной длительности $x(n)_N$ (непериодических) оценивают:

– спектральную плотность $X(j\omega)$ размерностью [В/Гц], определяемую ее модулем $|X(j\omega)|$ и аргументом $\varphi(\omega)$, т.е. амплитудным и фазовым спектрами или реальной $X_{\mathrm{Re}}(j\omega)$ и мнимой $X_{\mathrm{Im}}(j\omega)$ частями и вычисляемую на частотах анализа $\omega = \omega_k$ или бинах ДПФ;

– энергетический спектр или спектральную плотность энергии $S_x(\omega)$ $\left(|X(j\omega)|^2\right)$ размерностью [$В^2$с/Гц], показывающую распределение энергии сигнала по частоте и также вычисляемую на дискретных частотах ω_k.

Для случайных сигналов $x(n)$ оценивают спектральную плотность мощности $P_x(\omega)$ размерностью [$В^2$/Гц], являющуюся статистическим расширением понятия спектральной плотности энергии на случайные сигналы, т.е. сигналы с бесконечной энергией, и также вычисляемую на дискретных частотах ω_k. Для стохастически связанных случайных сигналов $x(n)$, $y(n)$ с помощью ДПФ измеряют их взаимную спектральную плотность мощности $P_{xy}(j\omega)$.

При реализации конкретных алгоритмов спектрального анализа различных сигналов большое значение имеет правильное масштабирование результатов анализа и учет их размерности. Так, по вычисленному ДПФ $X(j\omega_k) = \sum_{n=0}^{N-1} x(n) e^{-j\omega_k n T_Д}$ вещественного периодического сигнала $x_p(n)_N$ с периодом $NT_Д$ и частотами гармоник $if_Д / N$, совпадающими с бинами ДПФ, амплитуды гармоник $A_m(\omega_i)$ определяются как $(2/N)|X(j\omega_i)|$, фазы $\varphi(\omega_i)$ как $\mathrm{arctg}[X_{\mathrm{Im}}(j\omega_i) / X_{\mathrm{Re}}(j\omega_i)]$, и средние мощности – как $2|(1/N)X(j\omega_i)|^2$. По ДПФ $X(j\omega_k)$ детерминированного сигнала конечной длительности $NT_Д$ аналогичным образом находятся амплитуды, фазы и мощности k-й частотной выборки спектра сигнала, а спектральная плотность сигнала на частотах ω_k определяется как $T_Д X(j\omega_k)$. Другие спек-

тральные характеристики такого сигнала связаны с его ДПФ соотношениями: $S_x(k) = \left|T_Д X(j\omega_k)\right|^2$ – спектральная плотность энергии на частоте ω_k; $P_x(k) = T_Д / N \left|X(j\omega_k)\right|^2$ – спектральная плотность мощности на частоте ω_k; $S_x = \frac{1}{NT_Д}\sum_{k=0}^{N-1} S_x(k)$ – полная энергия сигнала; $P_x = \frac{1}{NT_Д}\sum_{k=0}^{N-1} P_x(k)$ – средняя мощность сигнала. Можно отметить, что корректность масштабов при вычислении энергии и мощности сигнала обеспечивает пара преобразований с размерным множителем $T_Д$ перед ДПФ и $1/T_Д$ – перед ОДПФ, называемая дискретно-временным рядом Фурье (ДВРФ).

3.7.3. Частотная характеристика анализатора спектра на основе ДПФ

Существует тесная связь между ДПФ и цифровой фильтрацией. Для ее установления запишем отклик нерекурсивного фильтра (НФ) с конечной импульсной характеристикой $h(n)$ длиной N на анализируемый с помощью ДПФ сигнал $x(n)$ в момент времени $nT_Д = (N-1)T_Д$ (при $n = N-1$):

$$y(n) = \sum_{m=0}^{N-1} h(m)x(n-m) = \sum_{m=0}^{N-1} x(m)h(n-m);$$

$$y(N-1) = \sum_{m=0}^{N-1} x(m)h(N-1-m). \qquad (3.136)$$

Из сопоставления (3.135) и (3.136) следует, что выход канала ДПФ $X(j\omega_k)$ совпадает с выходом НФ с импульсной характеристикой, отвечающей условию:

$$h(N-1-m) = w(m)e^{-j\omega_k mT_Д}$$

или

$$h(m) = w(N-1-m)e^{-j\omega_k (N-1-m)T_Д}.$$

Это означает, что k-й канал ДПФ эквивалентен полосовому НФ с импульсной характеристикой $h(m)$, согласованной со взвешенным комплексным гармоническим сигналом $w_k(m) = w(m)e^{j\omega_k mT_Д}$ в соответствии с

условием согласования: $h(n) = w*_k\,(N-1-m)$. Такой фильтр имеет частотную характеристику

$$H(j\omega) = e^{-j\omega(N-1)T_Д} W^*[j(\omega-\omega_k)], \qquad (3.137)$$

являющуюся комплексно-сопряженной частотной характеристики весовой функции $W^*(j\omega)$, смещенной вправо к частоте $\omega_k > 0$ (центральной частоте канала $k > 0$) или влево к частоте $\omega_k < 0$ (центральной частоте канала $k < 0$).

Частотную характеристику НФ можно определить также по его отклику $y(N-1)$ на комплексный гармонический сигнал $x(n) = e^{j\omega nT_Д}$, отнесенному к значению этого сигнала при $n = N-1$:

$$H(j\omega) = y(N-1) / e^{j\omega(N-1)T_Д} = e^{-j\omega(N-1)T_Д} W^*[j(\omega-\omega_k], \qquad (3.138)$$

что совпадает с выражением для частотной характеристики (3.137).

Это позволяет аналогичным образом определить и частотную характеристику канала ДПФ, обозначаемую $H(j\omega_k, \omega)$, в виде отношения его отклика на комплексный гармонический сигнал $x(n) = e^{j\omega nT_Д}$ к значению сигнала при $n = N-1$:

$$H(j\omega_k, \omega) = ДПФ_N\left\{e^{j\omega nT_Д} w(n)\right\} / e^{j\omega(N-1)T_Д} = e^{-j\omega(N-1)T_Д} W^*[j(\omega-\omega_k)]. \qquad (3.139)$$

Максимум частотной характеристики канала имеет место на частоте $\omega = \omega_k$, при этом на частоты $\omega > 0$, совпадающие с бинами ДПФ, откликаются каналы анализатора спектра с номерами $k > 0$ в полосе частот $(0 - \omega_Д / 2)$, а на частоты $\omega < 0$ (иначе на сигнал $x(n) = e^{-j\omega nT_Д}$), совпадающие с бинами ДПФ, откликаются каналы с номерами $k < 0$ в полосе частот $(-\omega_Д / 2 - 0)$.

Таким образом, анализатор спектра на основе ДПФ обеспечивает разрешение комплексных гармонических сигналов не только по частоте, но и по знаку частоты. При вычислении ДПФ в диапазоне частот $(0 - f_Д)$, т.е. для $k = 0,1,...(N-1)$, отрицательным значениям $k\_ = -1,-2,...-N/2$ соответствуют значения $k = k\_ + N$.

В качестве примера найдем частотную характеристику анализатора спектра с прямоугольной весовой функцией $w_R(n)$ длиной N. Ее преобразование Фурье $W_R(j\omega) = e^{-j\omega(N-1)/2}[\sin(\omega NT_Д / 2] / [\sin(\omega T_Д / 2)]$.

Каналы $k = 0,1,...(N-1)/2$ анализатора спектра, откликающиеся на сигнал $x(n) = e^{j\omega nT_Д}$, для данной весовой функции описываются частотной характеристикой

$$H_R(j\omega_k, \omega) = e^{-j\omega(N-1)/2} \frac{\sin[(\omega - \omega_k)NT_Д / 2]}{\sin[(\omega - \omega_k)T_Д / 2]}. \qquad (3.140)$$

Выражение для частотных характеристик каналов $k\_ = -1,-2,...-(N-1)/2$ или $k = (N-1)/2,...(N-1)$, откликающихся на сигнал $x(n) = e^{-j\omega nT_Д}$, имеет вид:

$$H_R(j\omega_k, \omega) = e^{j\omega(N-1)/2} \frac{\sin[(\omega + \omega_k)NT_Д / 2]}{\sin[(\omega + \omega_k)T_Д / 2]}. \qquad (3.141)$$

Важной для практики является задача обнаружения и оценки параметров вещественных гармонических сигналов $x(n) = (\cos\omega nT_Д)$. Такой сигнал можно представить в виде суммы двух комплексных гармонических сигналов с частотой $\omega > 0$ и $\omega < 0$:

$$x(n) = \frac{1}{2}[e^{j\omega nT_Д} + e^{-j\omega nT_Д}] = \frac{1}{2}[x_1(n) + x_2(n)]. \qquad (3.142)$$

В соответствии с принципом суперпозиции ДПФ сигнала $x(n)$ также можно найти как сумму откликов на его комплексно-сопряженные составляющие, определяемых, как показано выше, частотной характеристикой анализатора спектра.

3.7.4. Особенности гармонического анализа сигналов

Гармонический спектральный анализ, как и спектральный анализ других сигналов, требует учета особенностей решаемой задачи и творческого подхода к выбору типа весовой функции, времени анализа, использованию других приемов спектрального анализа, в том числе дополнения обрабатываемой последовательности нулевыми отсчетами.

Анализ и оценка параметров полигармонических сигналов с когерентной выборкой. В данном случае обрабатывается сигнал $x(n) = \sum_{i=1}^{L} A_{mi} \sin(\omega_i nT_Д + \varphi_i)$ состоящий из одной или L синусоидальных составляющих с частотами f_i, отвечающими условию $f_i = kF$, т.е. являю-

щихся гармониками некоторой известной или неизвестной частоты F, связанной с частотой дискретизации сигнала $f_Д$ соотношением $F = pf_Д / N$ (так называемый режим когерентной выборки), где N и p – целые взаимно простые числа. С помощью спектрального анализа нужно оценить частоты присутствующих в сигнале составляющих, их амплитуды и фазы.

Такая задача спектрального анализа возникает при тестировании (идентификации) объектов гармоническими или сложными полигармоническими сигналами, частоты которых формируются от общего задающего генератора, который является одновременно и источником частоты дискретизации как тестового сигнала, так и считываемого отклика объекта. Режим когерентной выборки реализуется также путем фазовой синхронизации или автоподстройки частоты дискретизации $f_Д$ по сигналу основной (опорной) частоты F входного сигнала.

При исследовании характеристик моделируемых объектов, в том числе систем ЦОС, по откликам их на гармонические и полигармонические сигналы также используется спектральный анализ сигналов на основе ДПФ. При этом значения частот тестовых сигналов и время анализа, т.е. число точек ДПФ при заданной частоте дискретизации выбирают с учетом указанных выше условий однозначного разрешения всех частотных составляющих, присутствующих в тестовом сигнале: $N = kf_Д / f_i$.

В качестве весовой функции анализаторов спектра с когерентной выборкой используется прямоугольная весовая функция. Значения неизвестных частот составляющих определяются номером канала (бина ДПФ), откликающегося на данную частоту, т.е. $f_i = kf_Д / N$, амплитуда их $A_{mi} = 2|X(jk)| / N$, фаза $\varphi_i = \text{arctg}[X(jk]$.

При наложении на сигнал белого шума и/или шума квантования АЦП со спектральной плотностью $(\sigma^2_ш / f_Д)$, обусловленная ими случайная оценка определяется мощностью шума на выходе канала ДПФ $\sigma^2_{шк} = [\sigma^2_ш / f_Д]\Delta F_ш / N = (\sigma^2_ш / N)\Delta F_ш$. Наилучшей в данном случае также является прямоугольная весовая функция, имеющая минимальную эквивалентную шумовую полосу $\Delta F_ш = \Delta F_{ш\ \min} = 1$. Относительная погрешность оценки среднеквадратичного значения сигнала при этом составляет $\delta_{Ai} = A_{mi} / \sqrt{2}\sigma_{шк}$, а фазы – $\varphi_i = A_{mi} / \sqrt{2}\sigma_{шк}$. Для повышения точности оценок можно произвести предварительное усреднение K реализаций сигнала или K оценок ДПФ или увеличить в K раз время анализа (число точек

ДПФ N). При этом погрешность оценок амплитуды и фазы уменьшается в $\sqrt{K}$ раз.

Анализ и оценка параметров полигармонических сигналов с некогерентной выборкой. Некогерентная выборка означает отсутствие жесткой связи между частотными составляющими сигнала и частотой его дискретизации.

Для одночастотного сигнала в этом случае целесообразно использовать прямоугольную весовую функцию, а время анализа выбрать исходя из требуемого частотного разрешения $\Delta f_p = \Delta F_{гл(6)} f_Д / N$, откуда $N \geq \Delta F_{гл(6)} f_Д / \Delta f_p$. Частота сигнала здесь отождествляется с центральной частотой канала ДПФ, имеющего максимальное выходное значение, что соответствует погрешности оценки частоты, не превышающей $f_Д / 2N$. Погрешность оценки амплитуды сигнала при этом является большой и при $f = f_к \pm f_Д / 2N$ достигает максимального значения A_n =–3,92 дБ. Меньшую погрешность можно получить, используя другие весовые функции с меньшим значением коэффициента модуляции A_n или подбором шага дискретизации по частоте $f_Д / N$ путем изменения числа точек ДПФ N, так чтобы в случае прямоугольной весовой функции получить на выходе ДПФ одну частотную выборку с максимальной амплитудой. При этом можно говорить о режиме квазикогерентной выборки.

При большом числе частотных составляющих время анализа или число точек ДПФ N также выбирается из требуемого частотного разрешения, соответствующего, например, априорной информации о минимальном разносе по частоте спектральных составляющих сигнала, а в качестве весовой функции применяются весовые функции с малым уровнем боковых лепестков, так как прохождение внеканальных составляющих через боковые лепестки частотной характеристики анализатора спектра существенно влияет на погрешность спектральных оценок сигнала.

Аналогичным образом осуществляется анализ и в том случае, когда известно, что одна или несколько присутствующих совместно с сигналом составляющих представляют собой мощную узкополосную помеху. При этом уровни боковых лепестков частотной характеристики, через которые помехи просачиваются в каналы, соответствующие полезному сигналу, должны обеспечивать их необходимое ослабление, что достигается выбором удовлетворяющей этому весовой функции.

При наличии широкополосных шумов точность оценок можно повысить с помощью усреднения ДПФ по K реализациям сигнала или увели-

чением в K раз числа точек N с последующим усреднением K соседних частотных выборок ДПФ и их прореживанием в K раз.

При некогерентном гармоническом анализе, как и при анализе случайных сигналов, используется дополнение преобразуемой последовательности определенным числом нулевых отсчетов N_0, когда $\tilde{x}(n) = x(n)w(n)$ при $0 \le n \le N_1 - 1$ и $\tilde{x}(n) = 0$ при $N_1 \le n \le N - 1$. При этом ДПФ N_1-точечной последовательности $\tilde{x}(n)$ вычисляется по числу точек $N = N_1 + N_0$, где N_1 соответствует длине весовой функции:

$$\widetilde{X}(j\omega_k) = \sum_{n=0}^{N_1-1} w(n)x(n)e^{-j2\pi kn/N}, \; k = 0,1,..N-1. \qquad (3.143)$$

Это приводит к уменьшению шага дискретизации $\Delta f = f_Д / N$, к увеличению числа каналов анализа и интерполяции сигнала в частотной области. Интерполированные частотные выборки улучшают разрешение анализатора спектра по частоте. Однако такое улучшение является условным, так как ширина главного и боковых лепестков частотной характеристики канала анализатора спектра при этом остаются прежними и определяются длиной весовой функции N_1. Например, при $N_0 = N_1$ число каналов анализа удваивается, а частотный интервал между ними уменьшается вдвое. Если частота сигнала при этом точно совпадет с частотой k-го канала, в том числе и интерполированного, то даже в случае прямоугольной весовой функции на данный сигнал будут откликаться 3 канала анализатора спектра. Предлагается самостоятельно построить графики частотных характеристик анализатора спектра с добавлением нулевых отсчетов для прямоугольной весовой функции и весовой функции Хэмминга при $N_0 = N_1$.

Нулевые отсчеты используют также для дополнения обрабатываемой последовательности до числа точек $N = 2^i$ с целью применения высокоэффективных алгоритмов спектрального анализа на основе БПФ.

Дополнение последовательности нулевыми отсчетами может также помочь в разрешении вопроса, являются ли наблюдаемые частотные выборки ДПФ результатом размытия спектральной линии одночастотного сигнала или соответствуют выборкам частотных составляющих, реально присутствующих в обрабатываемом сигнале. Предлагается самостоятельно подумать над данной практической задачей спектрального анализа.

3.7.5. Спектральные представления случайных процессов

Дискретный случайный процесс можно рассматривать как некоторую совокупность или ансамбль действительных или комплексных дискретных временных последовательностей $\{x_i(n)\}, i = 0,1,2,3,...$, каждую из которых можно было бы наблюдать как результат проведения некоторого эксперимента. При фиксации временного индекса n в сечениях случайного процесса образуются случайные величины, при задании распределений плотностей вероятностей которых можно определить соответствующие моменты распределений.

Среднее значение (момент 1-го порядка) определяется выражением

$$< x(n) >= \lim_{M\to\infty} \frac{1}{M} \sum_{i=0}^{M-1} x_i(n) = \int_{-\infty}^{\infty} x p_1(x,n) dx . \qquad (3.144)$$

Автокорреляция случайного процесса в два различных момента времени n_1 и n_2 определяется выражением

$$r_{xx}(n_1,n_2) =< x(n_1)x^*(n_2) >= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2^* p_2(x_1,n_1,x_2^*,n_2) dx_1 dx_2^* , \qquad (3.145)$$

где * – знак комплексного сопряжения.

Автокорреляция центрированного случайного процесса, т.е. с удаленным средним значением, называется автоковариацией:

$$\begin{aligned} c_{xx}(n_1,n_2) &=< (x(n_1) - < x(n_1) >)(x^*(n_2) - < x^*(n_2) >) >= \\ &= r_{xx}(n_1,n_2) - < x(n_1) >< x^*(n_2) > . \end{aligned} \qquad (3.146)$$

Случайный процесс называется стационарным в широком смысле, если его среднее значение не зависит от времени, а автокорреляционная последовательность (АКП) зависит только от разности индексов времени $m = n_1 - n_2$.

Спектральная плотность мощности (СПМ) стационарного случайного процесса определяется как дискретно-временное преобразование Фурье автокорреляционной последовательности

$$P_{xx}(f) = T \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi f mT} . \qquad (3.147)$$

СПМ дискретного случайного процесса периодически (с периодом $1/T$ Гц) размножает по оси частот СПМ исходного непрерывного во време-

ни процесса в предположении, что она локализована в интервале частот $-1/2T \le f \le 1/2T$.

Обратное ДВПФ имеет вид

$$r_{xx}(m) = \int_{-1/2T}^{1/2T} P_{xx}(f) e^{j2\pi fmT} df \,, \qquad (3.148)$$

поэтому при $m = 0$ имеем:

$$r_{xx}(0) = < |x(n)|^2 > = \int_{-1/2T}^{1/2T} P_{xx}(f) df \,. \qquad (3.149)$$

Последнее соотношение оправдывает трактовку $P_{xx}(f)$ как распределение средней мощности процесса по частоте. Дискретное преобразование Фурье называют соотношениями Винера – Хинчина для случая дискретного времени. Поскольку $r_{xx}(-m) = r_{xx}^*(m)$, то СПМ должна быть действительной и неотрицательной функцией. Если АКП – действительная функция, то $r_{xx}(-m) = r_{xx}(m)$ и СПМ можно представить в форме

$$P_{xx}(f) = 2T \sum_{m=0}^{\infty} r_{xx}(m) \cos(2\pi fmT) \,, \qquad (3.150)$$

а это означает четность СПМ как функции частоты, т.е. $P_{xx}(f) = P_{xx}(-f)$.

Если стационарный случайный процесс обладает свойством эргодичности, то процедуру усреднения по ансамблю при вычислении среднего значения и автокорреляции можно заменить усреднением по времени с использованием одной единственной реализации процесса.

Допущение об эргодичности позволяет ввести еще одно определение СПМ:

$$P_{xx}(f) = \lim_{M \to \infty} < \frac{1}{(2M+1)T} \left| T \sum_{n=-M}^{M} x(n) e^{-j2\pi f nT} \right|^2 > . \qquad (3.151)$$

Эта эквивалентная форма СПМ получается путем статистического усреднения квадрата модуля ДВПФ взвешенной совокупности данных, поделенного на длину записи данных, при увеличении числа отсчетов данных до бесконечности. Метод определения СПМ на основе оценки автокорреляционной последовательности называют косвенным (коррелограммным), поскольку случайный процесс $x(n)$ непосредственно не используется для оценки СПМ. Определение СПМ по формуле (3.151) называют прямым методом (периодограммным), т.к. процесс $x(n)$ непосредственно используется в расчете СПМ.

3.7.6. Коррелограммный метод оценки СПМ

Рассмотрим подробнее практическую реализацию коррелограммного метода оценивания СПМ. Автокорреляционная последовательность $r_{xx}(m)$ эргодического процесса как предел среднего по времени определяется соотношением

$$r_{xx}(m) = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x(n+m)\,x^*(n). \qquad (3.152)$$

На практике эта последовательность, как правило, не известна и поэтому должна оцениваться по имеющейся конечной последовательности данных.

Предположим, что мы располагаем N отсчетами данных $x(n), n = \overline{0,...,N-1}$, тогда одним из вариантов соотношения для оценки *АКП* может быть, например, следующее:

$$\hat{r}_{xx}(m) = \frac{1}{N-m} \sum_{n=0}^{N-m-1} x(n+m)x^*(n), \qquad (3.153)$$

применимое при $0 \le m \le N-1$.

При отрицательных значениях индекса m значения *АКП* получаются из свойства сопряженной симметрии $\hat{r}_{xx}(-m) = \hat{r}^*(m)$. Таким образом, $(2N-1)$ значений *АКП* могут быть оценены по N отсчетам данных. Дискретная последовательность $\hat{r}_{xx}(m)$ является несмещенной оценкой истинной *АКП*, поскольку

$$< \hat{r}_{xx}(m) >= \frac{1}{N-m} \sum_{n=0}^{N-m-1} < x(n+m)x^*(n) > = r_{xx}(m)\,. \qquad (3.154)$$

При увеличении N значение дисперсии оценки $\hat{r}_{xx}(m)$ стремится к нулю, поэтому это статистически состоятельная оценка *АКП*.

Другой вариант соотношения для оценки *АКП* имеет вид

$$\check{r}_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} x(n+m)x^*(n)\,,\ 0 \le m \le N-1. \qquad (3.155)$$

Эта оценка отличается от $\hat{r}_{xx}(m)$ только нормирующим множителем $\check{r}_{xx}(m) = \frac{N-|m|}{N}\hat{r}_{xx}(m)$, поэтому при конечном N она является смещенной, поскольку

$$< \check{r}_{xx}(m) >= (1 - \frac{|m|}{N}) r_{xx}(m). \qquad (3.156)$$

Для типичных приложений средний квадрат ошибки будет, как правило, больше для оценки $\hat{r}_{xx}(m)$, чем для оценки $\check{r}_{xx}(m)$. По этой и ряду других причин предпочтение часто отдается смещенной оценке АКП.

Суммирование, предусматриваемое в выражениях (3.153) и (3.156), можно записать как линейную свертку

$$\sum_{n=0}^{N-m-1} x(n+m)x^*(n) = x(n) \cdot x^*(-n) \ . \qquad (3.157)$$

Следовательно, для вычисления оценок дискретной автокорреляции можно применить эффективные в вычислительном отношении процедуры, такие как «быстрая свертка», в которых используются алгоритмы БПФ.

Коррелограммный метод оценивания СПМ есть подстановка в соотношение (3.152) конечной последовательности значений оценки автокорреляции (коррелограммы) на место бесконечной последовательности неизвестных истинных значений АКП. Так, при подстановке несмещенной оценки АКП $\hat{r}_{xx}(m)$ с максимальными значениями временного сдвига $\pm L$ получается оценка СПМ в виде

$$\hat{P}_{xx}(f) = T \sum_{m=-L}^{L} \hat{r}_{xx}(m) e^{-j2\pi fmT} \ , \ -\frac{1}{2T} \le f \le \frac{1}{2T} . \qquad (3.158)$$

Максимальный индекс временного сдвига L обычно ограничивают величиной $L \approx N/10$. Причина выбора такого максимального значения – стремление устранить большие значения дисперсии, связанные с оценками автокорреляции при больших временных сдвигах.

Среднее значение оценки $\hat{P}_{xx}(f)$ определяется соотношением

$$\begin{aligned} < \hat{P}_{xx}(f) > &= T \sum_{m=-L}^{L} < \hat{r}_{xx}(m) > e^{-j2\pi f mT} = \\ &= T \sum_{m=-L}^{L} r_{xx}(m) e^{-j2\pi f mT} = T \sum_{m=-\infty}^{\infty} w(m) r_{xx}(m) e^{-j2\pi f mT} \ , \end{aligned} \qquad (3.159)$$

где прямоугольная функция (окно) единичной амплитуды

$$w(m)=\begin{cases} 1, & -L\le m\le L, \\ 0, & m<-L; m>L. \end{cases} \tag{3.160}$$

Соотношение (3.159) есть ДВПФ произведения последовательностей АКП $r_{xx}(m)$ и $w(m)$, которое выражается через свертку ДВПФ сомножителей:

$$<\hat{P}_{xx}(f)>=P_{xx}(f)*D_{2L+1}(f), \tag{3.161}$$

где ДВПФ прямоугольного окна

$$D_{2L+1}(f)=\frac{\sin(2L+1)\pi fT}{\sin\pi fT}. \tag{3.162}$$

Таким образом, среднее значение оценки $\hat{P}_{xx}(f)$ отличается от истинного спектра $P_{xx}(f)$, поскольку это свертка истинного спектра и спектра прямоугольного окна.

Для уменьшения эффекта просачивания (растекания) спектра из-за влияния прямоугольного окна и уменьшения смещения оценки в соотношение (3.158) следует ввести весовую функцию (корреляционное окно) $w(m)$, $-L\le m\le L$, симметричное относительно начала отсчета. Тогда оценка СПМ, получаемая на основе коррелограммного метода, принимает форму

$$\hat{P}_{xx}(f)=T\sum_{m=-L}^{L} w(m)\hat{r}_{xx}(m)e^{-j2\pi f mT}. \tag{3.163}$$

Среднее значение этой оценки будет сверткой истинного спектра и спектра окна $W(f)$:

$$<\hat{P}(f)>=T\sum_{m=-L}^{L} w(m)r_{xx}(m)\,e^{-j2\pi f mT}=P_{xx}(f)*W(f). \tag{3.164}$$

Правильный выбор окна позволит уменьшить растекание спектра и его смещение.

Для вычисления оценки СПМ, определяемой на сетке из K частот $f_{i=i/KT}$, где $0\le i\le K-1$, можно использовать алгоритм БПФ. Обычно значение $K=2^M\gg L$, а это требует доопределения $\hat{r}^*_{xx}(m)=0$ при $L<m<K-L-1$ и $\hat{r}_{xx}(K-m)=\hat{r}^*_{xx}(m)$ при $0<m\le L$.

3.7.7. Периодограммный метод оценки СПМ

Как уже отмечалось выше, для эргодического случайного процесса эквивалентным определением СПМ является следующая дискретно-временная форма:

$$P_{xx}(f)=\lim_{N\to\infty}\left(\frac{1}{(2N+1)T}\left|T\sum_{n=-N}^{N}x(n)e^{-j2\pi fnT}\right|^2\right). \qquad (3.165)$$

При наличии конечного множества данных $x(n)$, $0\le n\le N-1$ и единственной реализации это соотношение преобразуется в СПМ выборки или периодограмму:

$$\tilde{\tilde{P}}_{xx}(f)=\frac{1}{NT}\left|T\sum_{n=0}^{N-1}x(n)\,e^{-j2\pi fnT}\right|^2=\frac{T}{N}\left|\sum_{n=0}^{N-1}x(n)e^{-j2\pi fnT}\right|^2. \qquad (3.166)$$

Несложно показать, что среднее значение оценки $\tilde{\tilde{P}}_{xx}(f)$ такое же, как и у коррелограммной оценки, дисперсия оценки не стремится к нулю при $N\to\infty$, а среднеквадратичное значение оценки при любом N сравнимо со средним значением. Следовательно, выборочный спектр не является состоятельной (статистически устойчивой) оценкой СПМ.

Предложен ряд модификаций соотношения (3.166), суть которых сводится к сглаживанию оценки путем введения псевдоусреднения по некоторому ансамблю. Наибольшую популярность в настоящее время получил метод Уэлча, идея которого состоит в образовании псевдоансамбля периодограмм за счет разбиения наблюдаемых данных на перекрывающиеся сегменты и обработки каждого сегмента с помощью окна данных.

Если запись комплексных данных $x(n)$, $0\le n\le N-1$ разбита на P сегментов по D отсчетов в каждом со сдвигом S отсчетов между соседними сегментами, то максимальное число сегментов P есть целая часть величины $(N-D)/S+1$. После взвешивания окном $w(n)$ сегмент с номером $p(0\le p\le P-1)$ будет содержать отсчеты

$$x^{(p)}(n)=w(n)x(n+pS),0\le n\le D-1.$$

Выборочный спектр P-го сегмента определяется выражением

$$\tilde{\tilde{P}}_{xx}^{(p)}(f)=\frac{1}{UDT}X^{(p)}(f)\left[X^{(p)}(f)\right]^*=\frac{1}{UDT}\left|X^{(p)}(f)\right|^2, \qquad (3.167)$$

где $X^{(p)}(f)=T\sum_{n=0}^{D-1}x^{(p)}(n)\,e^{-j2\pi fnT}$ – ДВПФ сегмента, $U=T\sum_{n=0}^{D-1}w^2(n)$ – энергия окна.

Среднее значение сегментных периодограмм определяет оценку *СПМ*:

$$\tilde{\tilde{P}}_{xx}(f)=\frac{1}{P}\sum_{p=0}^{P-1}\tilde{P}_{xx}^{(p)}(f). \qquad (3.168)$$

Уэлч, в частности, предложил использовать окно Ханна и 50 %-е перекрытие сегментов, которое обеспечивало очень эффективные реализации его метода на основе алгоритма БПФ. Кроме того, при 50 %-м перекрытии сегментов все данные используются дважды, за исключением $D/2$ отсчетов на каждом конце исходной N-точечной последовательности данных, а это выравнивает обработку большинства отсчетов данных, поскольку те отсчеты, которые имели малые веса на одном сегменте, получают большие веса на следующем сегменте. Анализ поведения периодограммы Уэлча для гауссовских процессов показал, что минимальная дисперсия для окна Ханна достигается при 65 %-м перекрытии, при этом величина дисперсии увеличивается приблизительно на 8 % при использовании 50 %-го перекрытия сегментов. Дисперсия периодограммы Уэлча примерно обратно пропорциональна числу сегментов, т.е.

$$\mathrm{var}\{\tilde{\tilde{P}}_{xx}(f)\}=\frac{P_{xx}^2(f)}{P} \qquad (3.169)$$

в предположении независимости сегментов (хотя перекрытие сегментов приводит, конечно, к некоторой их взаимозависимости). Благодаря перекрытию по заданной записи данных можно сформировать большее число сегментов и уменьшить величину дисперсии.

3.8. Современные параметрические методы оценки

Для непараметрических методов, описанных в предыдущих разделах данной главы и использующих периодограммы и БПФ, характерны упомянутые выше ограничения: низкое спектральное разрешение при коротких сегментах данных, необходимость взвешивания для предотвращения просачивания спектральных составляющих. Этих сложностей можно избежать, используя параметрические методы. Ценой улучшения является необходимость всестороннего исследования подходящей модели каждого процесса, определения необходимого порядка выбранной модели для адекватного представления данных и вычисления параметров модели. В число

получаемых преимуществ входят повышенное спектральное разрешение, возможность применения к кратким сегментам данных, отсутствие просачивания спектральных составляющих, гребешкового искажения, размывания спектра и смещения весовой функции. Поскольку параметрические методы очень важны, ниже представлен самый распространенный из них – метод авторегрессионного моделирования. Тем не менее, кроме преимуществ, параметрические методы имеют и недостатки, которые можно устранить с помощью альтернативных современных подходов, например, последовательного или адаптивного и метода максимального правдоподобия.

Параметрический подход требует параметрического моделирования данных, процесса, хорошо изученного в сфере анализа с помощью временных рядов, плюс интерполяции данных, которые рассматриваются как выход линейной системы, возбужденной белым шумом. Данная система представляется полиномиальной передаточной функцией, выраженной через параметры модели. Далее по этой передаточной функции вычисляется спектр данных [1].

3.8.1. Авторегрессионная оценка спектра

В данном методе оцифрованный сигнал моделируется как авторегрессионный (АР) временной ряд плюс слагаемое белого шума. После этого спектр находится из параметров АР-модели и дисперсии слагаемого шума. Параметры модели – это решение системы линейных уравнений, полученных минимизацией среднеквадратической ошибки (мощности белого шума) по всем данным. Несколько способов решения данной системы уравнений описано ниже. Важным моментом является выбор числа членов в АР-модели – *порядка* модели. При слишком маленьком порядке оценка плотности мощности будет чрезмерно сглаженной, так что некоторые максимумы могут скрадываться. Если порядок слишком велик, могут вводиться ложные максимумы. Следовательно, для каждого набора данных нужно определить подходящий порядок модели, и ниже эта тема обсуждается подробно. Описанный метод применим к сигналам, спектр плотности мощности которых содержит острые максимумы. Если сигналы не удовлетворяют этому критерию, можно использовать другие модели, такие как модель скользящего среднего или авторегрессионного движения. Поскольку авторегрессионный подход дает решаемые уравнения, его стоит использовать, когда только возможно.

3.8.2. Авторегрессионная модель и авторегрессионный фильтр

В АР-модели временного ряда текущее значение ряда $x(n)$ представляется линейной функцией предыдущих значений плюс ошибка $e(n)$:

$$x(n) = -a(1)x(n-1) + a(2)x(n-2) - \ldots - a(k)x(n-k) - \\ - a(p)x(n-p) + e(n). \tag{3.170}$$

Это уравнение содержит p предыдущих членов и представляет модель порядка p. Его можно записать в более компактной форме:

$$x(n) = -\sum_{k=1}^{p} a(k)x(n-k) + e(n) = -\sum_{k=1}^{p} a(k)z^{-k}x(n) + e(n), \tag{3.171}$$

где z^{-k} – оператор запаздывания, которым обозначена задержка на k интервалов выборки. Далее перепишем уравнение (3.171):

$$x(n) + \sum_{k=1}^{p} a(k)z^{-k}x(n) = \left(1 + \sum_{k=1}^{p} a(k)z^{-k}\right)x(n) = e(n). \tag{3.172}$$

Отсюда

$$x(n) = \frac{e(n)}{1 + \sum_{k=1}^{p} a(k)z^{-k}}. \tag{3.173}$$

Выражая отношение $x(n)/e(n)$, получаем

$$\frac{x(n)}{e(n)} = \frac{1}{1 + \sum_{k=1}^{p} a(k)z^{-k}} = H(z). \tag{3.174}$$

Здесь $H(z)$ интерпретируется как z-преобразование цифрового БИХ-фильтра с одними полюсами, который имеет коэффициенты $a(k)$. Подобный фильтр (рис. 3.42) называется авторегрессионным. Из уравнения (3.174) $x(n)$ можно рассматривать как выходы данного фильтра, порожденные случайными входами $e(n)$.

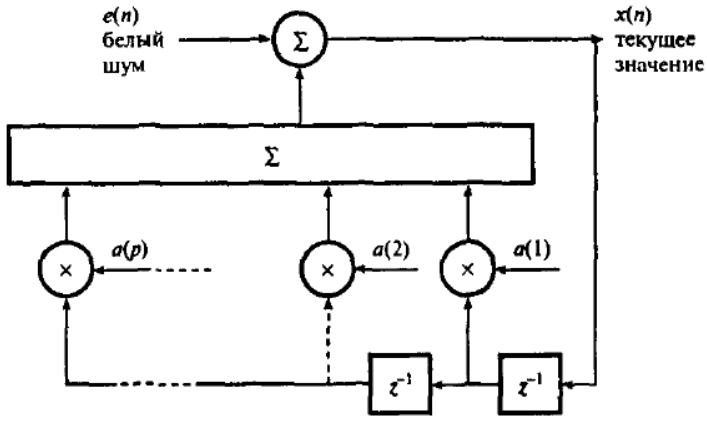


Рис. 3.42. АР-фильтр

Величина $e(n)$ представляет ошибку между значением, предсказанным моделью, и истинным значением выборки $x(n)$. Как правило, предполагается, что $e(n)$ имеет свойства белого шума, т.е. гауссово распределение плотности вероятностей и равномерный спектр плотности мощности. Следовательно, можно считать, что $x(n)$ генерируется АР-фильтром, на входе которого – источник белого шума. Частотная характеристика фильтра $H(f)$ получается подстановкой ω в уравнение (3.172), где $z = e^{j\omega T}$ – угловая частота, а T – период дискретизации. Следовательно,

$$H(f) = \frac{1}{1 + \sum_{k=1}^{p} a(k) e^{-jk\omega T}}. \qquad (3.175)$$

3.8.3. Спектральная плотность мощности авторегрессионного ряда

Требуется найти спектральную плотность мощности $P_x(f)$ АР-ряда $x(n)$. Данная величина связана со спектральной плотностью мощности сигнала белого шума $P_x(f)$, равной ее дисперсии $\sigma_e^2(n)$:

$$P_x(f) = |H(f)|^2 P_e(f) = |H(f)|^2 \sigma_e^2(n) = \frac{\sigma_e^2(n)}{\left|1 + \sum_{k=1}^{p} a(k) e^{-jk\omega T}\right|^2}. \qquad (3.176)$$

Дисперсия белого шума равна ее среднеквадратическому значению, которое, в свою очередь, равно квадрату среднего значения $e(n)$, а эту величину мы обозначили через E. Итак, значение $\sigma_e^2(n)$ (или E) можно выразить через параметры модели, а поскольку параметры модели определить можно, то спектральная плотность мощности также находится.

3.8.4. Порядок авторегрессионной модели

Порядок авторегрессионной модели, которая наилучшим образом аппроксимирует данные, нужно выбирать аккуратно для каждого набора данных, поскольку порядок зависит от статистических свойств данных. Данный момент можно проиллюстрировать на данных ЭЭГ, где различные сегменты данных требуют моделей различных порядков [1]. Модели низкого порядка предпочтительнее, поскольку нужно подбирать меньше параметров. Впрочем, если порядок слишком низок, оценка спектра будет слишком сглаженной. С другой стороны, слишком высокий порядок приведет к появлению ложных максимумов и спектральной неустойчивости.

Для оценки порядка Экейком (Akaike) было предложено два параметра. Первый – это ошибка окончательного предсказания (*Final Prediction Error* – FPE), которая определяется следующим образом [1]:

$$FPE(p) = \frac{N+p}{N-p} E(p). \qquad (3.177)$$

Второй параметр – это информационный критерий Экейка (*Akaike Information Criterion*), который определяется так [1]:

$$AIC(p) = N \ln E(p) + 2p. \qquad (3.178)$$

Параметр $AIC(p)$ применяется для небольших наборов данных, тогда как $FPE(p)$ рекомендуется использовать для более крупных наборов. На практике обычно пытаются так выбрать p, чтобы одновременно минимизировать $FPE(p)$ и $AIC(p)$.

3.9. Методы и алгоритмы проекционной обработки сигналов

Как косвенные, так и прямые методы спектрального анализа имеют несколько недостатков. Хотя в настоящее время они сравнительно просто могут быть реализованы на нескольких серийно выпускаемых специализированных системах, им свойственны сильное размытие за счет высокого уровня боковых лепестков и недостаточное разрешение. Детально исследованы два метода: метод максимальной энтропии (или так называемая процедура авторегрессии) и метод максимального правдоподобия; кроме них начинает привлекать внимание и третий метод, разработанный Писаренко[21].

3.9.1. Метод максимальной энтропии

В литературе оценку спектральной плотности мощности авторегрессионного процесса часто называют спектральным анализом на основе *метода максимальной энтропии* (ММЭ). Этот подход был введен Бергом [23]. Если конечная автокорреляционная последовательность $r_{xx}[0]$, ..., $r_{xx}[p]$ полагается известной, то может быть поставлен вопрос о том, как следует определять оставшиеся неизвестные члены этой последовательности $r_{xx}[p+1]$, $r_{xx}[p+2]$, ..., для того чтобы полная автокорреляционная последовательность обязательно была бы положительно-полуопределенной. Существует бесконечное число возможных экстраполяций, которые будут

давать автокорреляционную последовательность, отвечающую этому требованию. Берг показал, что эту экстраполяцию следует выполнять таким образом, чтобы максимизировать энтропию временного ряда, характеризуемого этой экстраполированной АКП. Получаемый при этом временной ряд будет наиболее случайным (в энтропийном смысле) из всех рядов, которым соответствует заданная АКП с временными сдвигами от 0 до p. Спектральная оценка, получаемая по этой экстраполированной АКП, будет в этом случае оценкой для процесса с максимальной энтропией.

Для гауссовского случайного процесса удельная энтропия пропорциональна величине

$$\int_{-1/2T}^{1/2T} \ln P_{MMЭ}(f)df\ , \tag{3.179}$$

где $P_{MMЭ}(f)$ – СПМ этого процесса. Значение $P_{MMЭ}(f)$ определяется посредством максимизации величины (3.179) при наложенных ограничениях, что она удовлетворяет соотношению Винера – Хинчина для p + 1 известных значений автокорреляции:

$$\int_{-1/2T}^{1/2T} P_{MMЭ}(f)\exp(j2\pi fnT)df = r_{xx}[n], \tag{3.180}$$

где $0 \le n \le p$. Это решение, которое находится с помощью метода множителей Лагранжа имеет вид

$$P_{MMЭ}(f) = \frac{T\rho_\omega}{\left|1+\sum_{k=1}^{p} a_p[k]\exp(-j2\pi fkT)\right|^2}, \tag{3.181}$$

где ρ_ω, $a_p[1]$, …, $a_p[p]$ определяются из решения уравнений Юла – Уолкера.

Таким образом, авторегрессионная СПМ и СПМ, получаемая методом максимальной энтропии, идентичны в случае гауссовского случайного процесса и известной автокорреляционной последовательности с равноотстоящими значениями. Однако неправильно говорить, что уравнение (3.181) является основой спектрального анализа по методу максимальной энтропии в тех случаях, когда исследуемый процесс не является гауссовским или когда значения параметров оцениваются не по известной АКП, а по отсчетам данных. Метод максимальной энтропии применим и в случае неэквидистантной автокорреляционной последовательности. Однако авторегрессионная СПМ и СПМ, получаемая методом максимальной энтропии, не будут идентичны.

3.9.2. Метод максимального правдоподобия

Метод максимального правдоподобия (ММП) первоначально был предназначен для обработки сейсмических сигналов, а также сигналов антенных решеток в гидролокации. Потом он применялся для анализа различных временных сигналов [24].

При использовании данного метода спектрального анализа искомая оценка спектра представляет собой мощность шума на выходе фильтра с конечной импульсной характеристикой, который синтезирован таким образом, что его характеристика имеет единичное (т. е. неискаженное) значение на интересующей нас частоте и минимальное значение на всех остальных частотах, а спектр получается просто как оценка мощности шума в зависимости от частоты. В этом случае находится оценка корреляционной функции входных данных, формируется корреляционная матрица Теплица, которая затем обращается, и вычисляется ее квадратичная форма как функция частоты. Корреляционная функция может быть найдена любым из всех упоминавшихся выше методов, и в частности, перемножением и сложением задержанных сигналов, перемножением и сложением задержанных жестко ограниченных сигналов, а также методами БПФ; кроме того, может быть использована корреляционная функция теплицевой формы в обращенном виде. Однако большую часть вычислений составляет вычисление квадратичной формы.

Из сравнения двух описанных алгоритмов, которые иллюстрируются на рис. 3.43, следует несколько существенных выводов [25]. ММЭ обеспечивает лучшее частотное разрешение, однако в спектре появляются ложные пики, что обусловлено большим уровнем боковых лепестков. Кроме того, получаемая оценка спектра зашумлена значительно сильнее, хотя, согласно опубликованным данным, стабильность измерений в окрестности тональных пиков вполне достаточная. То, что оценка спектра состоятельная, можно доказать ее асимптотической несмещенностью при увеличении порядка фильтра и объема данных. Однако это свойство сильно зависит от стабильности частоты дискретизации, что ограничивает возможность распространения этого метода на случай обработки сигналов антенных решеток. При использовании метода максимального правдоподобия получается намного более стабильная оценка спектра со значительно меньшим уровнем ложных боковых лепестков. Однако этот метод обеспечивает худшее разрешение, чем ММЭ, причем оценка получается смещенной, хотя смещение можно скомпенсировать подбором коэффициента, зависящего от длины характеристики фильтра.

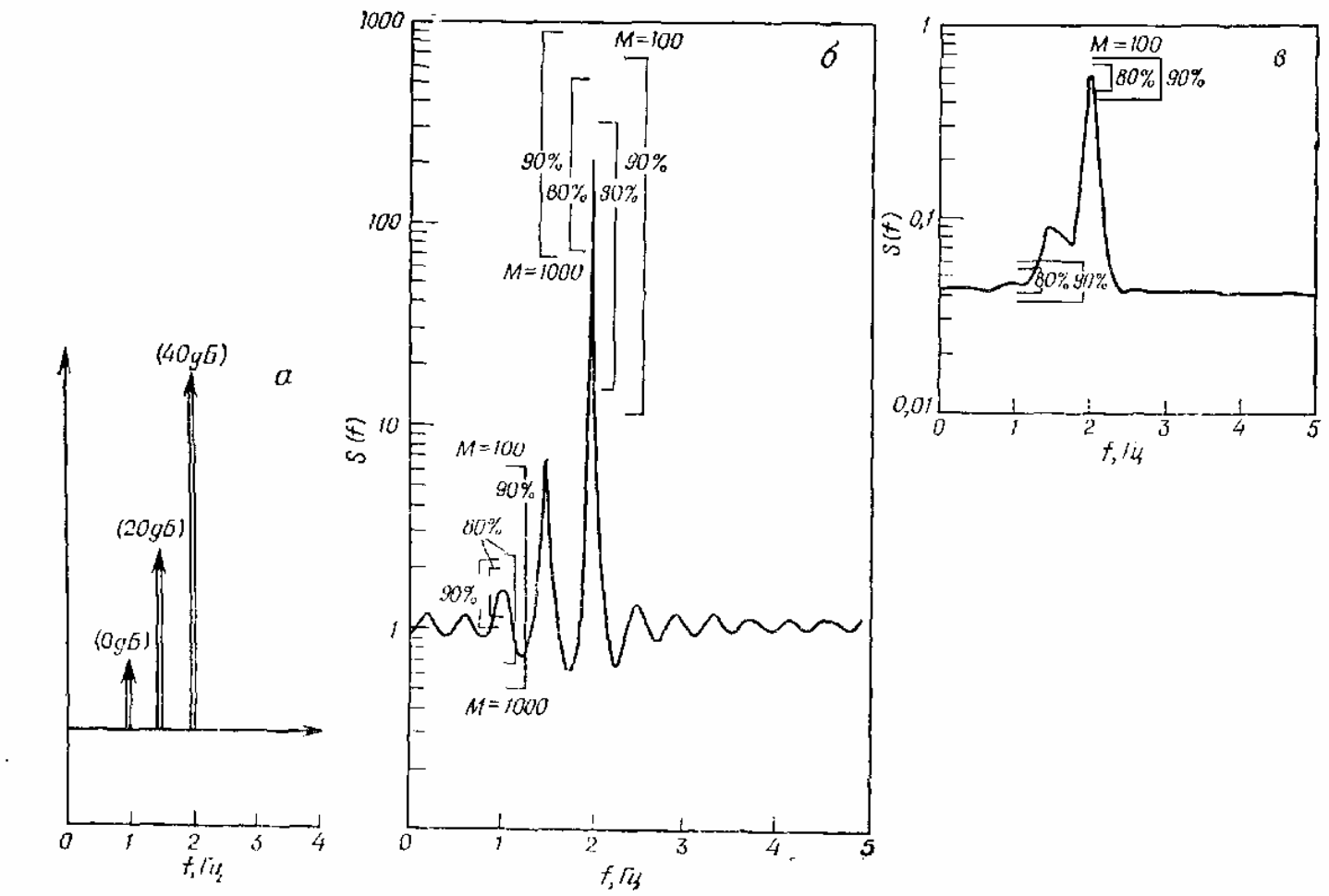


Рис. 3.43. Сравнение метода максимальной энтропии и метода максимального правдоподобия спектрального анализа (задержка коррелирования равна 2,4 с): *а* – истинный спектр, состоящий из трех тональных линий на фоне белого шума; *б* – оценка методом максимальной энтропии; *в* – оценка методом максимального правдоподобия

3.9.3. Метод гармонического разложения Писаренко

Метод гармонического разложения Писаренко (ГРП) относится к числу самых первых процедур спектрального оценивания, основанных на анализе собственных значений. В этом методе предполагается, что анализируемый процесс состоит из смеси M комплексных синусоид и аддитивного комплексного белого шума, а для оценки частот синусоид, их мощности и дисперсии белого шума используется известная автокорреляционная последовательность от $r_{xx}[0]$ до $r_{xx}[M]$. Если анализируемый процесс состоит из смеси M действительных синусоид и аддитивного действительного белого шума, то АКП должна быть известна для корреляционных сдвигов от 0 до $2M$. В комплексном случае автокорреляционная матрица R_{M+1}, сформированная из АКП, будет иметь в подпространстве шума только один собственный вектор v_{M+1} и одно связанное с ним собственное значение p_{ω}, соответствующее дисперсии белого шума [21]. Это собственное значение является также минимальным собственным значением матрицы R_{M+1} что может быть записано в следующем виде:

$$R_{M+1}v_{M+1} = p_{\omega}v_{M+1}. \qquad (3.182)$$

Если матрица R_{M+1} задана, то с помощью подпрограммы MINEIGVAL, приведенной в [21, гл. 3], можно определить это минимальное собственное значение (дисперсию шума) и соответствующий ему собственный вектор. Алгоритм, реализуемый данной подпрограммой, учитывает комплексно-сопряженную симметрию собственного вектора. После определения этого вектора частоты синусоид находятся посредством факторизации полинома фильтра, соответствующего полученному собственному вектору. При этом гарантируется, что корни указанного полинома будут лежать на единичной окружности.

После выполнения процедуры факторизации полинома и определения частот синусоид определяют мощности этих синусоид. Процедура вычислений основана на использовании матричного уравнения, приведенного ниже, которое можно решить относительно вектора неизвестных мощностей синусоид P_1, ..., P_M с помощью любой стандартной программы, предназначенной для решения комплексных линейных уравнений. На практике значения АКП должны оцениваться по отсчетам данных

$$\begin{pmatrix} \exp(j2\pi f_1 T) & \exp(j2\pi f_2 T) & \cdots & \exp(j2\pi f_M T) \\ \exp(j2\pi f_1 2T) & \exp(j2\pi f_2 2T) & \cdots & \exp(j2\pi f_M 2T) \\ \vdots & \vdots & & \vdots \\ \exp(j2\pi f_1 MT) & \exp(j2\pi f_2 MT) & \cdots & \exp(j2\pi f_M MT) \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_M \end{pmatrix} = \begin{pmatrix} r_{xx}[1] \\ r_{xx}[2] \\ \vdots \\ r_{xx}[M] \end{pmatrix}, \quad (3.183)$$

К числу трудных относится и задача определения значения порядка (т.е. числа синусоид) M, которое должно использоваться в методе ГРП.

В тех случаях, когда значение порядка превышает число синусоид, собственные значения, соответствующие дисперсии шума, должны, как это следует из теории, повторяться, поэтому проверка наличия повторяющихся собственных значений у автокорреляционных матриц очень больших порядков могла бы, по-видимому, стать одним из методов выбора порядка. Однако использование на практике оцененных автокорреляционных последовательностей обычно не приводит к появлению повторяющихся собственных значений. Следовательно, характеристики оценок, получаемых с помощью метода ГРП, в случае коротких последовательностей данных будут, как правило, ухудшаться.

Адаптивный вариант метода Писаренко для слежения за узкополосными сигналами в широкополосном шуме был предложен Томпсоном, а характеристики этого метода были тщательно исследованы Ларимором. В его основе лежит последовательно-временной алгоритм реализации метода ГРП, который, в отличие от описанного выше алгоритма блочной реализации этого метода, осуществляет обновление оценки собственного вектора (связанного с минимальным собственным значением) по мере поступления очередного отсчета данных. Этот алгоритм представляет собой один из вариантов алгоритма наименьших средних квадратов, реализуемого с помощью процедуры условного (т.е. с наложенными ограничениями) градиентного поиска:

$$\begin{aligned} \mathrm{v}'[k+1] &= \mathrm{v}[k] - 2\mu e_k[k+1]x[k]; \\ \mathrm{v}[k+1] &= \frac{\mathrm{v}'[k+1]}{(\mathrm{v}'[k+1])^H (\mathrm{v}'[k+1])}; \end{aligned} \quad (3.184)$$

$$e_k[k+1] = x[k+1] + \sum_{n=1}^{M} \upsilon[n] x[k+1-n].$$

Когда алгоритм сходится, вектор υ оказывается вектором единичной нормы в подпространстве порядка $p = 2M$, связанным с минимальным собственным значением. Опыт работы с этим рекурсивным по времени алгоритмом показал, что его сходимость может быть немонотонной и что для выделения спектральных линий необходимо применять длинные окна данных.

3.10. Оптимальные линейные фильтры

3.10.1. Модели случайных процессов и шумов

Результаты практических измерений, подлежащие обработке, содержат определенный полезный сигнал на фоне различного рода помех (шумов), при этом спектр помех в общем случае недетерминирован и в той или иной мере представлен по всему интервалу главного частотного диапазона. Другими словами, спектр полезного сигнала наложен на спектр шумов. В этих условиях ставится задача реализации так называемых оптимальных фильтров, которые, в зависимости от своего конкретного назначения, позволяют достаточно надежно производить обнаружение сигнала, наилучшим образом выделять сигнал на фоне помех или в максимальной степени подавить помехи без существенного искажения сигнала.

Случайные процессы и шумы описываются функциями автокорреляции и спектрами мощности. Модели случайных процессов (сигналов) с заданными статистическими характеристиками обычно получают фильтрацией белого шума.

3.10.1.1. Белый шум. Белый шум $q(t)$ можно представлять как случайную по времени (аргументу) последовательность дельта-импульсов $\delta(t_i)$ со случайными амплитудными значениями a_i

$$q(t) = \sum_i a_i \delta(t - t_i), \qquad (3.185)$$

которая удовлетворяет условиям статистической однородности: постоянное среднее число импульсов в единицу времени и статистическая независимость появления каждого импульса от предыдущих. Такой поток им-

пульсов, который называют пуассоновским, является некоррелированным и имеет равномерный спектр мощности:

$$R_q(\tau) = c^2\delta(t);$$

$$W_q(\omega) = c^2,$$

где $c^2 = N\sigma_a^2$, N – число импульсов на интервале T реализации случайного процесса, σ_a^2 – дисперсия амплитуд импульсов.

3.10.1.2. Фильтрация белого шума. Если на входе фильтра с импульсным откликом $h(t)$ действует белый шум $q(t)$, то сигнал на выходе фильтра

$$g(t) = h(t)q(t) = h(t)\sum_i a_i\delta(t-t_i) = \sum_i a_i h(t-t_i), \qquad (3.186)$$

т.е. выходной сигнал будет представлять собой последовательность сигналов импульсной реакции фильтра $h(t)$ с амплитудой a_i, при этом автокорреляционная функция и спектр мощности выходного потока также становятся подобными ФАК и спектру мощности импульсной реакции фильтра и в первом приближении определяются выражениями

$$R_q(\tau) \approx N\delta_a^2(t)R_h(\tau) = c^2R_h(\tau); \qquad (3.187)$$

$$W_g(\omega) \approx N\delta_a^2(t)\left|H(\omega)\right|^2 = c^2\left|H(\omega)\right|^2. \qquad (3.188)$$

Этот результат известен как теорема Кэмпбелла.

3.10.2. Критерии построения оптимальных фильтров

В практике обработки геофизических данных используются три основных критерия построения оптимальных фильтров: минимум среднего квадратического отклонения профильтрованного сигнала от его действительного или заданного значения, максимум отношения сигнал/шум и максимум энергетического отношения сигнал/шум на выходе фильтра. При анализе и синтезе фильтров используется аддитивная модель входного сигнала: $x(k) = s(k) + q(k)$, где $s(k)$ – полезная составляющая сигнала; $q(k)$ – составляющая помех. Синтез оптимальных фильтров производится с максимальным использованием известной априорной информации как о сигналах, которые необходимо выделить, так и о шумах и помехах. Как

правило, используется информация о природе полезного сигнала и шума, об их спектральном составе, о корреляционных и взаимно корреляционных характеристиках. Наличие определенных особенностей (различий) в характеристиках сигнала и шума позволяет реализовать фильтр вообще и оптимальный фильтр в частности. Если такие особенности отсутствуют, постановка задачи становится некорректной.

В геофизической практике априорные данные о полезных сигналах, как правило, являются достаточно определенными, особенно для активных методов геофизики (сейсмические методы, электроразведка на переменном токе, индукционные методы ядерной геофизики и пр.). Определение характеристик действующих помех представляет собой более сложную проблему, но даже при полной неопределенности можно допустить, что помеха является нормальным стационарным процессом с нулевым средним значением.

3.10.2.1. Среднее квадратическое отклонение. При наличии помех абсолютно точное выделение полезного сигнала методами линейной фильтрации, как правило, невозможно. Результат фильтрации

$$y(k) = h(n) \cdot x(k-n) \tag{3.189}$$

отличается от $s(k)$ на величины $\varepsilon(k) = y(k) - s(k)$, которые являются абсолютными значениями погрешности воспроизведения полезного сигнала по координатам k. Качество фильтра оценивается средним значением квадрата величины $\varepsilon(k)$:

$$\overline{\varepsilon^2} = \overline{[y(k) - s(k)]^2}\,. \tag{3.190}$$

Во многих задачах обработки геофизических данных не требуется восстановления исходной формы сигнала $s(k)$, т.к. в процессе его дальнейшей обработки осуществляется преобразование сигнала $s(k)$ в сигнал $z(k)$, форма которого может быть более удобной для извлечения (измерения) каких-либо информационных параметров сигнала (например, амплитудного значения, ширины сигнала на половине максимального значения и т.п.). В этом случае оптимальный фильтр может проектироваться непосредственно на получение выходного сигнала $z(k)$. Качество таких фильтров, получивших название формирующих, оценивается средним значением квадрата величины $\varepsilon(k)$ получения сигнала заданной формы:

$$\overline{\varepsilon^2} = \overline{[y(k) - s(k)]^2}\,. \tag{3.191}$$

Выражение (3.190) дает возможность определить значения $h(k)$ фильтра по критерию минимума среднего квадратического отклонения выходного сигнала от его действительной или заданной формы. Еще раз отметим, что данный критерий исходит из вероятностно-статистической модели экспериментальных данных и хорошо себя показал при обработке геофизических данных, но его возможности могут быть ограничены при количественной интерпретации геофизических аномалий.

3.10.2.2. Амплитудное отношение сигнал/шум. При постановке задачи обнаружения (установления факта наличия) в экспериментальных данных сигнала известной формы для проектирования фильтра используется, как правило, критерий максимума пикового отношения сигнал/шум на выходе фильтра:

$$\rho_a = y_{экс} / \sigma, \tag{3.192}$$

где $y_{экс}$ – экстремальное (максимальное или минимальное) значение амплитуды сигнала; σ – средний квадратический уровень амплитудных значений помех (σ^2 – дисперсия помех).

Если в полезном сигнале отсутствует четко выраженный экстремум, а сам сигнал достаточно протяженный по аргументу (что обычно имеет место в геофизической практике), то в качестве критерия используется отношение средних квадратов амплитуд сигнала и шума:

$$\overline{\rho^2} = \overline{y^2} / \sigma, \tag{3.193}$$

где $\overline{y^2}$ – средний квадрат амплитуды сигнала в пределах его формы.

3.10.2.3. Энергетическое отношение сигнал/шум. При узкоконкретной задаче обнаружения сигнала степень искажения самого сигнала может не ограничиваться. Если кроме обнаружения сигнала, как основной цели обработки данных, ставится и задача оценки его формы, то в этом случае для проектирования фильтра обычно используется критерий максимума энергетического отношения сигнал/шум:

$$\rho = E_{sy} / E_{qh}, \tag{3.194}$$

где E_{sy} и E_{qh} – энергия, соответственно, сигнала и шума на выходе фильтра.

3.10.3. Фильтр Колмогорова – Винера

3.10.3.1. Условие оптимальности фильтра. Подставим (3.189) в выражение (3.191) и получим среднее квадратическое отклонение ε^2 формы выходного сигнала $y(k) = h(n)x(k-n)$ от оптимальной формы $z(k)$ по всем точкам массива данных (в частном случае $z(k) = s(k)$):

$$\varepsilon^2 = \overline{\left[\sum_n h(n)x(k-n) - z(k)\right]^2}. \qquad (3.195)$$

Минимум функции (3.195) определяет значения коэффициентов $h(n)$ оптимального фильтра. Для нахождения их значений продифференцируем выражение (3.195) по коэффициентам фильтра и приравняем полученные уравнения нулю. В итоге получаем:

$$d(\varepsilon^2) / dh(n) = -\overline{z_k x_{k-m}} + \sum_n h_n \overline{x_{k-m}x_{k-n}} = 0,$$

где $\overline{x_{k-m}x_{k-n}} = R(m-n)$ – корреляционная функция входного сигнала, $\overline{z_k x_{k-m}} = B(m)$ – взаимная корреляционная функция сигналов $z(k)$ и $x(k)$.

Отсюда

$$\sum_n h_n R(m-n) = B(m), \quad n = m = 0,1,2, ..., M. \qquad (3.196)$$

В краткой форме записи:

$$h(n)R(m-n) = B(m). \qquad (3.197)$$

Другими словами, ***свертка функции отклика фильтра с функцией автокорреляции входного сигнала должна быть равна функции взаимной корреляции входного и выходного сигналов.***

3.10.3.2. Система линейных уравнений фильтра. Выражение (3.196) может быть записано в виде системы линейных уравнений – однострочных равенств левой и правой частей для фиксированных значений m. С учетом четности функции автокорреляции

$$m = 0: \; h_0R(0) + h_1R(1) + h_2R(2) + h_3R(3) + ... + h_MR(M) \quad = B(0);$$
$$m = 1: \; h_0R(1) + h_1R(0) + h_2R(1) + h_3R(2) + ... + h_MR(M-1) \; = B(1);$$
$$m = 2: \; h_0R(2) + h_1R(1) + h_2R(0) + h_3R(1) + ... + h_MR(M-2) \; = B(2);$$
$$.....$$
$$m = M: \; h_0R(M) + h_1R(M-1) + h_2R(M-2) + h_3R(3) + ... + h_MR(0) \; = B(M).$$

Решение данной системы уравнений относительно значений h_i дает искомый оператор фильтра.

Отметим, что $R(i) = R_s(i) + R_q(i)$, где R_s – функция автокорреляции сигнала; R_q – функция автокорреляции шума, а $B(i) = B_{zs}(i) + B_{zq}(i)$, где B_{zs} – функция взаимной корреляции сигналов $z(k)$ и $s(k)$; B_{zq} – функция взаимной корреляции сигнала $z(k)$ и помех $q(k)$.

Подставляя эти выражения в (3.197), получаем:

$$h(n)[R_s(m-n) + R_q(m-n)] = B_{zs}(m) + B_{zq}(m). \qquad (3.198)$$

3.10.3.3. Частотная характеристика фильтра. Частотная характеристика фильтра находится преобразованием Фурье левой и правой частей уравнения (3.198):

$$\begin{aligned} &H(\omega)[W_s(\omega) + W_q(\omega)] = W_{zs}(\omega) + W_{zq}(\omega); \\ &H(\omega) = [W_{zs}(\omega) + W_{zq}(\omega)] / [W_s(\omega) + W_q(\omega)], \end{aligned} \qquad (3.199)$$

где $W_s(\omega) \Leftrightarrow R_s(m)$ и $W_q(\omega) \Leftrightarrow R_q(m)$ – энергетические спектры сигнала и помех; $W_{zs}(\omega) \Leftrightarrow B_{zs}(m)$ – взаимный энергетический спектр входного и выходного сигналов; $W_{zq}(\omega) \Leftrightarrow B_{zq}(m)$ – взаимный энергетический спектр выходного сигнала и помех.

В геофизической практике обычно имеет место статистическая независимость полезного сигнала, а, следовательно, и сигнала $z(k)$ от шумов, при этом $B_{zq} = 0$ и фильтр называют оптимальным по сглаживанию шумов при заданной форме выходного сигнала:

$$H(\omega) = W_{zs}(\omega) / [W_s(\omega) + W_q(\omega)]. \qquad (3.200)$$

Если при этом заданная форма сигнала $z(k)$ совпадает с формой полезного сигнала $s(k)$, то $B(m) = B_{zs} = R_s$ и фильтр называют фильтром воспроизведения полезного сигнала:

$$H(\omega) = W_s(\omega) / [W_s(\omega) + W_q(\omega)]. \qquad (3.201)$$

3.10.3.4. Эффективность фильтра. Из выражений (3.199) – (3.201) наглядно видно, что с позиции минимального искажения полезного сигнала при максимальном подавлении шумов фильтр Колмогорова – Винера эффективен в тем большей степени, чем больше отношение сигнал/шум на входе фильтра. В пределе, при $W_q(\omega) << W_s(\omega)$ имеем $H(\omega) \Rightarrow 1$ и фильтр

воспроизводит входной сигнал без искажений. Отметим также, что помеха, коррелированная с полезным сигналом, как это следует из (3.199), используется фильтром для повышения точности воспроизведения сигнала. С другой стороны, при $W_q(\omega) >> W_s(\omega)$ имеем $H(\omega) \Rightarrow 0$ и сигнал будет сильно искажен, но никакой другой фильтр лучшего результата обеспечить не сможет.

Если сигнал, подлежащий воспроизведению, не относится к числу случайных и представляет собой какую-либо детерминированную функцию, то $W_s(\omega) = |S(\omega)|^2$.

3.10.4. Оптимальные фильтры сжатия сигналов

3.10.4.1. Частотная характеристика. Частотная характеристика идеального фильтра, осуществляющего сжатие сигнала $s(k)$ к дельта-функции, определяется выражением

$$H(\omega) = 1 / S(\omega) = S^*(\omega) / |S(\omega)|^2, \qquad (3.202)$$

где $S^*(\omega)$ – комплексно сопряженный спектр полезного сигнала. На выходе такого фильтра имеем:

$$Y(\omega) = H(\omega) \cdot X(\omega) \to 1 \quad \text{при} \quad X(\omega) \to S(\omega).$$

Реализация фильтра возможна только при условии $S(\omega) > 0$ на всех частотах в главном частотном диапазоне. В противном случае, при $S(\omega_i) \to 0 \quad H(\omega_i) \to \infty$ и фильтр становится неустойчивым. Для исключения возможности такого явления в фильтр (3.202) вводится стабилизатор a:

$$H(\omega) = S^*(\omega) / \left[|S(\omega)|^2 + a \right], \qquad (3.203)$$

где $|S(\omega)|^2 + a > 0$ во всем частотном диапазоне.

3.10.4.2. Условие оптимальности. Фильтр сжатия сигнала может быть получен с использованием уравнения (3.197). Положим, что $z(k) = \delta(k)$ при статистической независимости сигнала и шума.

Отсюда

$$\begin{aligned} &B_{sz}(m) = \delta(m) \cdot s(k+m) = s(-m); \\ &h(n) \cdot (R_s(m-n) + R_q(m-n)) = s(-m); \\ &H(\omega) = S^*(\omega) / \left(|S(\omega)|^2 + W_q(\omega) \right). \end{aligned} \qquad (3.204)$$

Сравнение выражений (3.203) и (3.204) показывает, что оптимальной величиной стабилизатора a является значение спектральной плотности помех. При некоррелированной помехе с дисперсией σ^2 система уравнений для определения значений коэффициентов $h(n)$:

$$\begin{aligned}
&h_0(R(0)+\sigma^2)+h_1R(1)+h_2R(2)+h_3R(3)+...+h_MR(M) &&= s(0);\\
&h_0R(1) &&+h_1R(0)+h_2R(1)+h_3R(2)+...+h_MR(M-1) = 0;\\
&h_0R(2) &&+h_1R(1)+h_2R(0)+h_3R(1)+...+h_MR(M-2) = 0;\\
&\ldots\ldots\ldots

ции необходимо только зафиксировать наличие сигнала в массиве данных, если он там присутствует (а может и не присутствовать), при этом сохранения формы сигнала не требуется. Сама форма сигнала полагается известной либо по теоретическим данным (путем решения прямых задач геофизики или при активном воздействии на геологическую среду сигналами известной формы с учетом соответствующей реакции среды), либо по результатам предшествующих измерений на моделях или на аналогичных средах. Для уверенного обнаружения сигнала фильтр должен обеспечить максимально возможную амплитуду выходного сигнала над уровнем помех и, соответственно, выполняется на основе критерия максимума пикового отношения сигнал/помеха.

3.10.6.1. Частотная характеристика. Для расчета фильтра требуется задать известную форму полезного сигнала $s(k) \Leftrightarrow S(\omega)$ и функцию автокорреляции или спектр мощности помех $R_q(m) \Leftrightarrow W_q(\omega)$. Полный входной сигнал принимается по аддитивной модели: $x(t) = s(t) + q(t)$. На выходе проектируемого фильтра $h(n) \Leftrightarrow H(\omega)$ для составляющих выходного сигнала имеем:

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) S(\omega) e^{j\omega t} d\omega; \qquad (3.206)$$

$$\sigma^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left|H(\omega)\right|^2 W_q(\omega) d\omega, \qquad (3.207)$$

где σ – средняя квадратическая амплитуда выходной помехи. Значения (3.206), (3.207) используются для задания критерия максимума отношения сигнал/шум (3.193) для произвольной точки t_i:

$$\rho = [y(t_i)]^2 / \delta^2. \qquad (3.208)$$

Исследование функции (3.208) на максимум показывает, что он достигается при частотной характеристике фильтра

$$H(\omega) = e^{-j\omega t_i} S^*(\omega) / W_q(\omega). \qquad (3.209)$$

Без потери общности можно принять $t_i = 0$:

$$H(\omega) = S^*(\omega) / W_q(\omega) = \left|S(\omega)\right| e^{j\varphi_s(\omega)} / W_q(\omega). \qquad (3.210)$$

При переходе во временную (координатную) область

$$H(\omega) W_q(\omega) = S^*(\omega) \Leftrightarrow h(n) R_q(n-m) = s(-m). \qquad (3.211)$$

Система линейных уравнений для расчета фильтра:

$$\begin{aligned}
&h_0R_q(0)+h_1R_q(1)+h_2R_q(2)+h_3R_q(3)+\ldots+h_MR_q(M) &&=S(-M);\\
&h_0R_q(1)+h_1R_q(0)+h_2R_q(1)+h_3R_q(2)+\ldots+h_MR_q(M-1) &&=S(-M+1);\\
&h_0R_q(2)+h_1R_q(1)+h_2R_q(0)+h_3R_q(1)+\ldots+h_MR_q(M-2) &&=S(-M+2);\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&h_0R_q(M)+h_1R_q(M-1)+h_2R_q(M-2)+h_3R_q(3)+\ldots+h_MR_q(0) &&=S(0).
\end{aligned}$$

3.10.6.2. Эффективность фильтра. Из выражения (3.210) можно видеть, что фильтр имеет максимальный коэффициент передачи на частотах доминирования сигнала и минимальный коэффициент передачи на частотах доминирования помех. Кроме того, фазовая характеристика фильтра $\varphi(\omega)=-\varphi_s(\omega)$ обеспечивает синфазность всех частотных составляющих выходного сигнала и, соответственно, максимальную его амплитуду в заданный момент времени $t_i = 0$:

$$y(0)\Leftrightarrow S(\omega)H(\omega)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{S(\omega)S^*(\omega)}{W_q(\omega)}=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{|S(\omega)|^2}{W_q(\omega)}. \qquad (3.212)$$

Отметим также, что коэффициент передачи фильтра тем больше и эффективность его работы тем выше, чем больше различия в форме частотных спектров сигнала и шумов. Для постоянной формы спектров сигнала и шума любой другой фильтр уступает данному фильтру как по пиковому, так и по энергетическому отношению сигнал/шум на выходе фильтра.

3.10.6.3. Согласованный фильтр. При помехах типа белого шума $W_q(\omega)=\sigma^2$ и $H(\omega)=S^*(\omega)/\sigma^2$. Множитель $1/\sigma^2$ не влияет на отношение сигнал/помеха и может быть опущен. Частотная характеристика фильтра определяется только спектром сигнала, при этом

$$h(n)=s(-n). \qquad (3.213)$$

Фильтр получил название согласованного (по частотной характеристике со спектром сигнала). Он малоэффективен при коротком импульсном или длинном гармоническом сигнале.

3.10.6.4. Обратный фильтр. Допустим, что помеха имеет такой же частотный состав, что и полезный сигнал, т.е.

$$W_q = \sigma^2 \left| S(\omega) \right|^2 . \qquad (3.214)$$

Выделение полезного сигнала в таких условиях весьма сомнительно. Тем не менее, определим оптимальный фильтр:

$$H(\omega) = S^*(\omega) / \left[\sigma^2 \left| S(\omega) \right|^2 \right] = 1 / \left[\sigma^2 S(\omega) \right]. \qquad (3.215)$$

Выражение (3.215) с точностью до постоянного множителя соответствует фильтру сжатия сигнала. Но если согласованный фильтр и фильтр сжатия рассматривать в качестве предельных случаев при полной неопределенности характеристики помех, то в качестве модели помех можно принять их суперпозицию:

$$W_q = a^2 \left| S(\omega) \right|^2 + b^2 . \qquad (3.216)$$

Подставляя это выражение в (3.210), с точностью до множителя получаем:

$$H(\omega) = S^*(\omega) / \left[\left| S(\omega) \right|^2 + \gamma^2 \right], \qquad (3.217)$$

где $\gamma = b / a$ – отношение дисперсий шума и сигнала. Фильтр стремится к согласованному при больших γ, и к обратному (фильтру сжатия) при малых.

Вопросы и задания для самопроверки

1. Представьте разностное уравнение, которое описывает ЦФ.
2. По какому правилу определяется порядок РЦФ?
3. Каким образом можно определить импульсную характеристику ЦФ?
4. Что понимают под передаточной функцией ЦФ?
5. Известно разностное уравнение ЦФ $y(n) = 0,5s(n) - 0,3s(n-1) - -0,7y(n-1) + 0,55y(n-2)$, запишите передаточную функцию.
6. Приведите классификацию структур ЦФ.
7. В чем преимущество канонической структуры ЦФ по сравнению с прямой?
8. Представьте обобщенную схему канонической структуры ЦФ.
9. На основе каких правил следует выбирать тип структуры ЦФ при его реализации?
10. В каких областях используется спектральный анализ сигналов?
11. Назовите основные методы спектрального анализа.
12. Назовите основные параметры анализаторов спектра.
13. Представьте значения основных параметров известных вам весовых функций.
14. В чем отличие между скачущим и скользящим ДПФ?
15. В соответствии с каким алгоритмом осуществляется обработка сигнала рекурсивным цифровым фильтром?
16. Как определяется импульсная характеристика цифрового фильтра, какие цифровые фильтры называют фильтрами БИХ и КИХ-типа?
17. Какой смысл имеют коэффициенты нерекурсивных цифровых фильтров?
18. В чем особенности частотных характеристик цифровых фильтров и чем они обусловлены?
19. Как определяется передаточная функция рекурсивного фильтра по его разностному уравнению?
20. Каковы структура и математическое описание прямой и канонической форм реализации рекурсивных звеньев второго порядка?
21. Какой объем вычислительных операций выполняется в рекурсивном и нерекурсивном фильтрах при обработке одного отсчета сигнала?
22. Как аналитически может быть найден отклик рекурсивного и нерекурсивного фильтров на заданное входное воздействие?
23. Что характеризуют СПМ и ВСПМ и на основе каких выражений их можно определить?
24. В чем сущность коррелограммного метода вычисления спектральной плотности мощности дискретных случайных сигналов?

МОДУЛЬ 4. АППАРАТНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

4.1. Универсальные и специализированные процессоры ЦОС

Здесь объясняются ключевые фундаментальные вопросы, связанные с универсальными и специализированными процессорами для цифровой обработки сигналов (ЦОС), влиянием алгоритмов ЦОС на аппаратную и программную архитектуру данных процессоров, а также с реализацией ключевых алгоритмов ЦОС на блоке специализированной аппаратуры или на универсальных процессорах ЦОС для работы в реальном времени.

Обработка в реальном времени обычно означает «как можно скорее», но в заданных временных рамках. Обработку в реальном времени можно разбить на две обширные категории (хотя данные категории также можно разбить на более мелкие классы): потоковая обработка, например, цифровая фильтрация, где данные обрабатываются по выборке за такт, и блоковая обработка, например, нахождение БПФ и корреляции, где за такт обрабатывается блок данных фиксированной длины. Отметим, что для реализации алгоритмов ЦОС в реальном времени требуется и программное, и аппаратное обеспечение. В качестве аппаратной поддержки можно использовать матрицы процессоров, стандартные микропроцессоры, чипы ЦОС или микропрограммируемые специализированные устройства. Программной поддержкой служат коды или микрокоды на языке Ассемблер низкого уровня, «родном» для аппаратуры ЦОС, и/или коды на эффективном языке высокого уровня, таком, как С или С++. В настоящее время обычно используют именно языки высокого уровня, особенно с новейшими процессорами ЦОС, имеющими сложную и запутанную структуру. Процессоры ЦОС, появившиеся в начале 1980-х, стали в настоящее время значительно сложнее, повысили свои возможности и расширили число сфер применения. Все это также привело к значительному увеличению числа доступных процессоров ЦОС.

Процессоры ЦОС можно для удобства разделить на две обширные категории – универсальные и специализированные. В число распространенных процессоров ЦОС входят такие устройства с фиксированной запя-

той, как TMS320C54x от Texas Instruments и DSP563x от Motorola, и такие устройства с плавающей запятой, как TMS320C4x (Texas Instruments) и ADSP21xxx SHARC (Analog Devices). Существуют два типа специализированных устройств:

1. Аппаратное обеспечение, разработанное для эффективного выполнения специальных алгоритмов ЦОС, таких, как цифровая фильтрация, быстрое преобразование Фурье. Устройства данного типа иногда называют *алгоритмическими процессорами ЦОС.*

2. Аппаратное обеспечение, разработанное для специального приложения, например, в сфере контроля, телекоммуникаций или цифрового аудио. Устройства данного типа иногда называют *процессорами ЦОС специального назначения (специализированными).*

В большинстве случаев специализированные процессоры выполняют такие алгоритмы ЦОС, как кодирование – декодирование PCM. Кроме того, они должны выполнять другие операции, отражающие специфику приложения. В качестве примеров специализированных процессоров ЦОС можно привести процессор Cirrus CS8420 для конвертеров частоты дискретизаций в цифровой аудиоаппаратуре, подавитель речевого эха в многоканальной телефонии Mitel MT9300, процессор БПФ PDSP16515A и программируемый КИХ-фильтр VPDSP16256.

Все универсальные и специализированные процессоры можно построить с помощью отдельных чипов или блоков умножителей, АЛУ, ячеек памяти и т.д.

Рассмотрим архитектурные особенности процессоров ЦОС, которые позволили применить цифровую обработку в реальном времени во многих областях.

4.2. Компьютерные архитектуры обработки сигналов

Большинство доступных сейчас универсальных процессоров основаны на архитектуре фон Неймана, при которой операции выполняются последовательно. Упрощенная архитектура стандартного процессора фон Неймана показана на рис. 4.1. При обработке инструкции в таком процессоре блоки процессора, не задействованные в каждой фазе инструкции, находятся в холостом состоянии до передачи им управления. Повышение скорости процессора достигается за счет ускорения работы отдельных блоков, хотя существует определенный предел, ограничивающий их возможную скорость работы.

Если устройство должно работать в реальном времени, архитектуру процессора ЦОС нужно оптимизировать под выполнение функций ЦОС. Например, на рис. 4.2 показана общая аппаратная архитектура, подходящая для цифровой обработки сигналов в реальном времени. Она характеризуется следующими особенностями:

1. Многошинная структура с раздельной памятью для данных и инструкций программы. Обычно память для хранения данных содержит входные данные, промежуточные значения и выходные выборки, а также фиксированные коэффициенты, например, для цифровой фильтрации или БПФ. Команды программы хранятся в специально отведенных ячейках памяти.

2. Порт ввода-вывода позволяет обмениваться данными с внешними устройствами (АЦП, ЦАП) или передавать цифровые данные другим процессорам. Прямой доступ к памяти (*Direct Memory Access* – DMA) (если он есть) позволяет быстро обмениваться блоками данных с памятью (ОЗУ) для хранения данных, причем обычно это происходит под внешним управлением.

3. Арифметические устройства для логических и арифметических операций, в число которых выходят АЛУ, аппаратные умножители и схемы сдвига (или умножители-накопители).

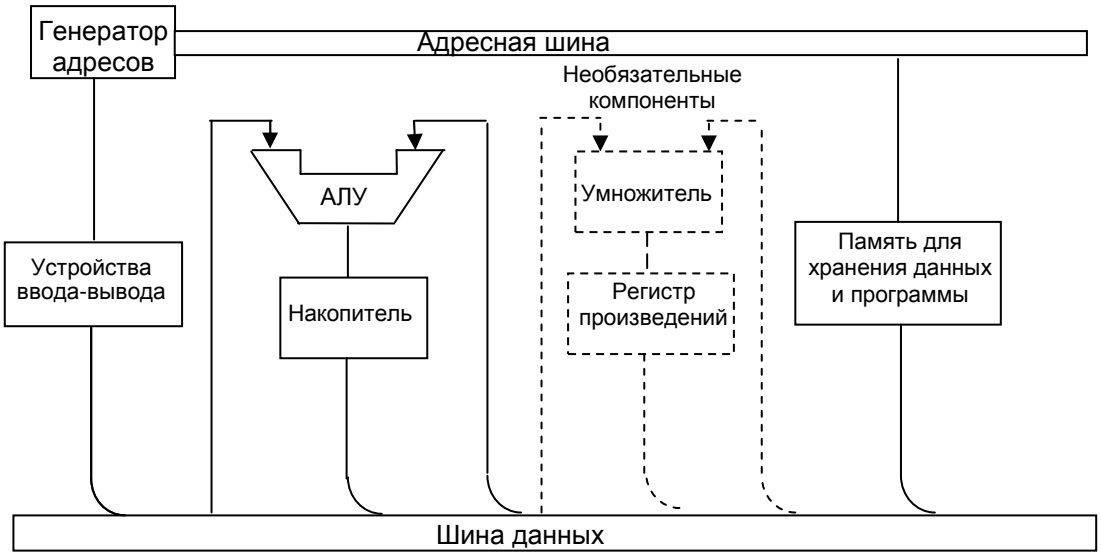


Рис. 4.1. Упрощенная архитектура стандартных микропроцессоров

Почему необходима именно такая архитектура? Большинство алгоритмов ЦОС (фильтрация, корреляция, преобразование Фурье) включают повторяющиеся арифметические операции, такие, как умножение, сложение, обращение к памяти и интенсивная передача данных через центральный процессор.

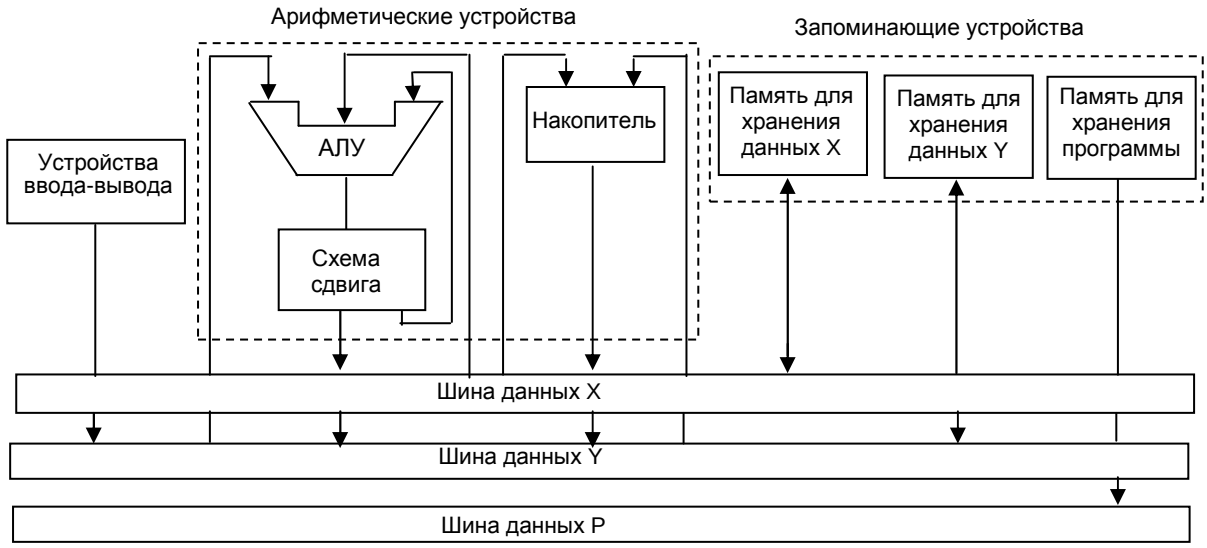


Рис. 4.2. Стандартная универсальная аппаратная архитектура обработки сигналов

Архитектура стандартных микропроцессоров не предназначена для такого типа деятельности. При разработке аппаратуры ЦОС важно оптимизировать под операции ЦОС и аппаратную архитектуру, и систему команд. В процессорах ЦОС для этого широко используется концепция параллелизма. В частности, применяются следующие средства:

- гарвардская архитектура;
- конвейерная обработка;
- быстрые специализированные аппаратные умножители-накопители;
- специальные команды, предназначенные для ЦОС;
- копирование;
- встроенная память/кэш;
- расширенный параллелизм – векторная архитектура (SIMD), архитектура с командными словами сверхбольшой длины (VLIW) и статическая суперскалярная обработка. Для успешной разработки структуры ЦОС важно понимать и правильно использовать все названные ключевые архитектурные особенности.

4.2.1. Гарвардская архитектура

Принципиальная особенность гарвардской (двухшинной) архитектуры заключается в том, что память для хранения данных и память для хранения программы располагаются в разных местах, допуская полное совмещение во времени операций вызова команды из памяти и ее выполнения. Стандартные микропроцессоры, такие как Intel 6502, характеризуются

одношинной структурой, через которую передаются и данные, и команды, как показано на рис. 4.1.

Предположим, что при использовании стандартного микропроцессора требуется считать значение opl, которое находится по адресу памяти ADR1, в аккумулятор, а затем сохранить его по двум другим адресам, ADR2 и ADR3. В этом случае могут использоваться такие команды:

LDA ADR1 загрузить операнд opl в аккумулятор из ячейки
ADR1 STA ADR2 сохранить opl по адресу ADR2 STA
ADR3 сохранить opl по адресу ADR3

Обычно каждая из приведенных команд включает три различных этапа:

1) выборка команды из памяти;
2) декодирование команды;
3) выполнение команды.

В нашем случае выборка команды включает вызов следующей команды из памяти, а выполнение команды предполагает чтение данных из памяти либо запись данных в память. В стандартном процессоре без гарвардской архитектуры команды программы (т.е. программный код) и данные (операнды) содержатся в одной области памяти (рис. 4.3).

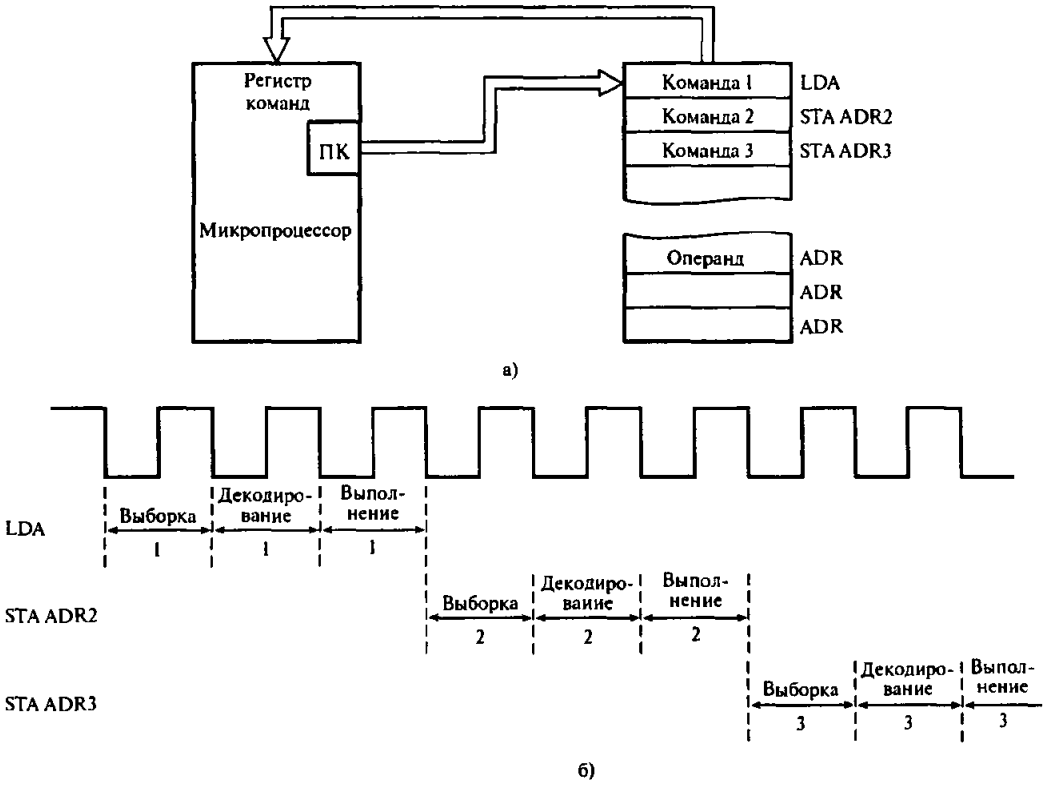


Рис. 4.3. Иллюстрация выборки, декодирования и выполнения команды в негарвардской архитектуре с единой областью памяти:
а – вызов команды из памяти; *б* – временная диаграмма процесса

Следовательно, вызов следующей команды при выполнении текущей невозможен, поскольку обе операции требуют обращения к памяти.

В гарвардской архитектуре (рис. 4.4), где команды программы и данные хранятся в различных областях памяти, вызов следующей команды может совпадать с выполнением текущей команды (рис. 4.5). Обычно память программы содержит программный код, тогда как память для хранения данных включает переменные, например, выборки входных данных.

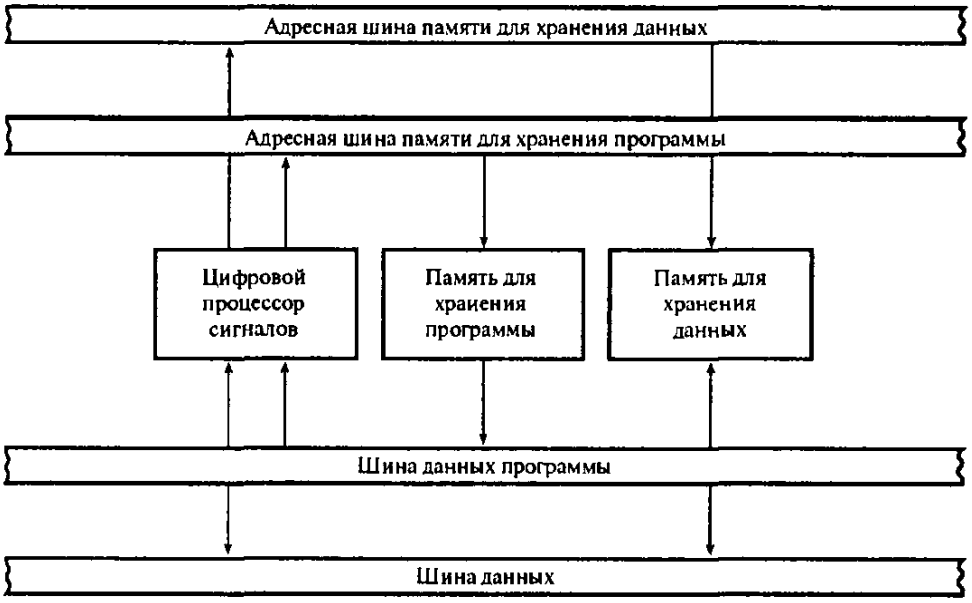


Рис. 4.4. Стандартная гарвардская архитектура с раздельными областями памяти для хранения данных и программы

Вызов команд данных и команд программы может выполняться одновременно, поскольку используются две независимые памяти.

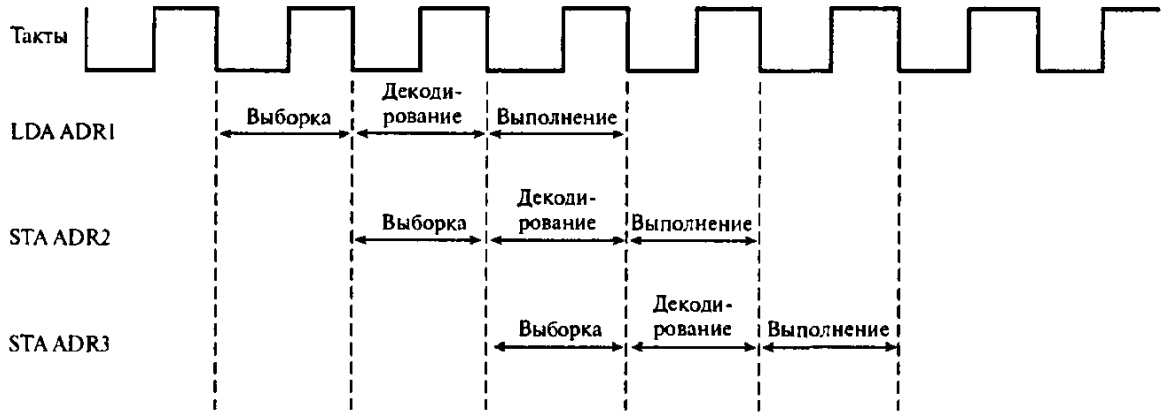


Рис. 4.5. Иллюстрация совмещения команд, возможного при использовании гарвардской архитектуры

Строгая гарвардская архитектура используется лишь в нескольких процессорах ЦОС (например, Motorola DSP56000), в большинстве устройств применяется модифицированная гарвардская архитектура (например, семейство процессоров TMS320). В модифицированной архитектуре также выделяются раздельные области памяти для хранения программы и данных, но в отличие от строгой гарвардской архитектуры здесь разрешена связь между двумя областями памяти.

4.2.2. Конвейерная обработка

Конвейерная обработка – это метод, который позволяет совмещать несколько операций в процессе выполнения. При конвейерной обработке задача разбивается на несколько различных подзадач, которые затем совмещаются при выполнении. Данный метод широко используется в цифровой обработке сигналов для повышения скорости. Конвейер в цифровой обработке подобен обычной производственной линии на фабрике, например, на предприятии по сборке телевизоров или автомобилей. Как и на производственной линии, задача разбивается на небольшие независимые подзадачи, именуемые *каскадами конвейера*. Последовательное соединение каскадов формирует конвейер.

Как было показано в предыдущем разделе, команду можно разбить на три этапа. Каждый этап команды рассматривается как каскад конвейера. Таким образом можно организовать наложение команд, при котором новая команда будет начинать выполняться в первый момент каждого такта (рис. 4.6, *а*).

На рис. 4.6, *б* приведена временная диаграмма трехкаскадного конвейера, на которой выделены этапы команд. Обычно каждый этап в конвейере занимает один машинный цикл. Следовательно, в течение данного цикла могут быть одновременно активны до трех различных команд, хотя все они будут на разных стадиях завершения. Основной момент при использовании конвейерной обработки команд заключается в том, что три части команды (т.е. выборка из памяти, декодирование и выполнение) независимы, так что выполнение нескольких команд может совмещаться. На рис. 4.6, *б* показано, что в ходе i-го цикла процессор может одновременного извлекать из памяти i-ю команду, декодировать $(i - 1)$-ю команду и в то же время выполнять $(i - 2)$-ю команду.

Трехкаскадный конвейер, рассмотренный выше, основан на технике, которая применяется в процессорах TMS320 (*Texas Instruments*). В том же процессоре TMS320 конвейерная обработка используется и по-

другому, когда для получения конвейера применяется несколько регистров. Счетчик упреждающей выборки содержит адрес следующей команды, которая будет вызываться из памяти, регистр команд содержит команду, которая будет выполняться, и регистр команды в очереди хранит команды, которые будут выполняться при выполнении текущей команды. Адрес следующей выполняемой команды фиксирует программный счетчик.

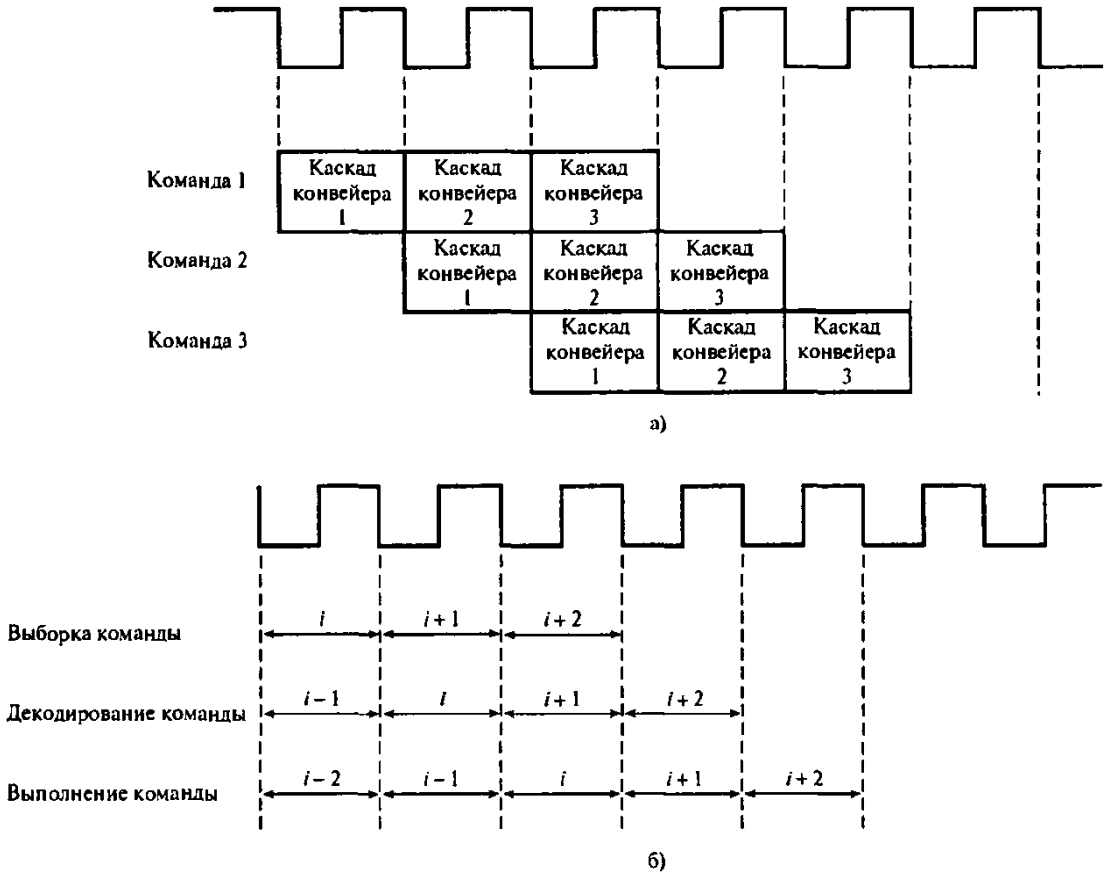


Рис. 4.6. Иллюстрация концепции конвейерной обработки

Благодаря использованию внутреннего параллелизма потока команд конвейерная обработка позволяет существенно снизить в среднем время выполнения одной команды. Пропускная способность машины с конвейерной обработкой определяется числом команд, пропущенных через конвейер за единицу времени. Как и в производственной линии, все каскады конвейера должны быть синхронизированы. Время перехода команды на следующий этап (см. рис. 4.6, *а*) равно одному циклу и зависит от наиболее медленного каскада конвейера. В идеальном конвейере среднее время выполнения одной команды равно [6]

$$\frac{\text{Время на команду (без конвейерной обработки)}}{\text{Число каскадов конвейера}}. \qquad (4.1)$$

В идеальном случае увеличение скорости равно числу каскадов конвейера (частей, на которые разбита команда). На практике увеличение скорости будет меньше из-за служебных издержек на организацию конвейера, задержек в регистрах конвейера и т.п.

Пример 4.1

В машине без конвейерной обработки вызов, декодирование и выполнение команды занимает 35, 25 и 40 не соответственно. Определите увеличение пропускной способности, если этапы команд обрабатываются конвейером. Предполагайте, что служебные издержки составляют 5 не на каждом этапе и что другие задержки несущественны.

Решение

В машине без конвейерной обработки среднее время выполнения команды просто равно сумме времен, необходимых на реализацию отдельных ее этапов: 35 + 25 + 40 = 100 нс. В то же время, если предположить, что процессор имеет фиксированный машинный цикл и что этапы команды синхронизированы по системным часам, завершение каждой команды потребует трех машинных циклов: 40 · 3 = 120 нс. Эта величина соответствует пропускной способности $8{,}3 \cdot 10^6$ команд $с^{-1}$.

В машине с конвейерной обработкой тактовая частота определяется скоростью самого медленного каскада плюс служебные издержки. В нашем случае машинный цикл равен 40 + 5 = 45 не, что вводит ограничение на среднее время выполнения команды. Пропускная способность (при заполненном конвейере) равна $22{,}2 \cdot 10^6$команд $с^{-1}$. Следовательно,

$$\text{Ускорение вычислений} = \frac{\text{Среднее время выполнения команды (без конвейера)}}{\text{Среднее время выполнения команды (с конвейером)}} = \qquad (4.2)$$

$$= \frac{120}{45} = 2{,}67 \text{ раза} \text{ (предполагается, что без конвейера команда выполняется за три цикла).}$$

В машине с конвейерной обработкой каждая команда по-прежнему требует трех тактов, но в каждом такте процессор занимается выполнением до трех различных команд. Конвейерная обработка увеличивает пропускную способность системы, но не время выполнения каждой команды. Как правило, из-за служебных издержек на организацию конвейера время выполнения каждой команды даже немного увеличивается.

Конвейерная обработка существенно влияет на память системы. Число обращений к памяти в машине с конвейерной обработкой увеличивается, осо-

бенно с увеличением числа этапов. При цифровой обработке сигналов конвейерная обработка сигналов облегчается при использовании гарвардской архитектуры, где данные и команды размещаются в различных областях памяти.

При последовательном соединении такого медленного устройства, как память для хранения данных, и арифметического элемента последний часто простаивает довольно долго. В таких случаях можно применять конвейерную обработку, что позволит полнее использовать возможности арифметического устройства. Ниже данная концепция иллюстрируется на примере.

Пример 4.2

Большинство алгоритмов ЦОС характеризуются операциями умножения-накопления, например:

$$a_0 x(n) + a_1(n-1) + a_2 x(n-2) + \ldots + a_{N-1} x(n-(N-1)).$$

На рис. 4.7 представлена конфигурация системы без конвейерной обработки, в которой приведенное выше уравнение реализовано с помощью арифметического элемента.

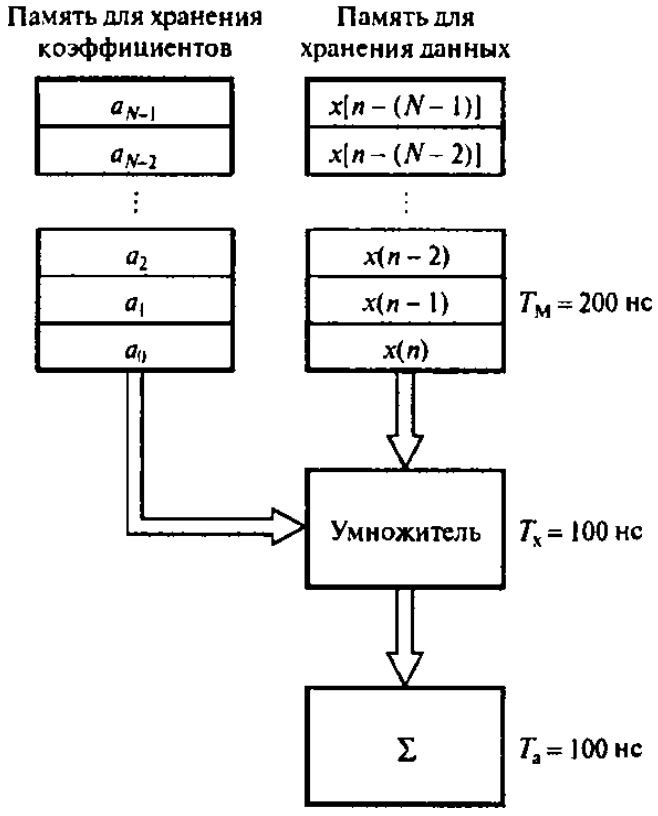


Рис. 4.7. Конфигурация умножителя-накопителя без конвейерной обработки. Произведения подаются в накопитель каждые 400 нс

Транспортные задержки для памяти, умножителя и накопителя равны соответственно 200, 100 и 100 нс.

Задание:

1. Определите, чему равна пропускная способность системы.
2. Измените конфигурацию системы для конвейерной обработки с двукратным увеличением скорости. Проиллюстрируйте операции в новой конфигурации с помощью временной диаграммы.

Решение

1. Коэффициенты $a_к$ и массивы данных хранятся в памяти, как показано на рис. 4.7, *а*. В режиме без конвейерной обработки коэффициенты и данные вызываются последовательно и подаются на умножитель. Произведения суммируются в накопителе (накапливающем сумматоре). Последовательные операции умножения-накопления будут завершаться каждые 400 не (200 + 100 + 100), так что пропускная способность равна $2{,}5 \cdot 10^6$ операций в с.

2. Фигурирующие в вычислениях арифметические операции можно разбить на три различных этапа – считывание из памяти, умножение и накопление. Для увеличения скорости данные шаги разных операций можно совместить. Для двукратного увеличения скорости следует ввести конвейерные регистры между памятью и умножителем и между умножителем и накопителем, как показано на рис. 4.8. Временная диаграмма полученного блока с конвейерной обработкой представлена на рис. 4.9. Как видно из диаграммы, операция умножения-накопления завершается каждые 200 не.

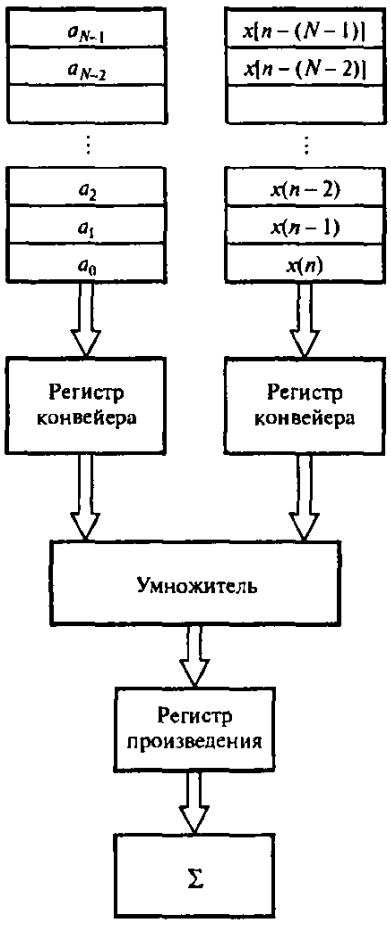


Рис. 4.8. Конфигурация умножителя-накопителя с конвейерной обработкой

Поскольку по условию задачи издержки на организацию конвейера можно не учитывать, ограничивающим фактором является стандартная задержка передачи через самый низкоскоростной элемент, в данном случае – память.

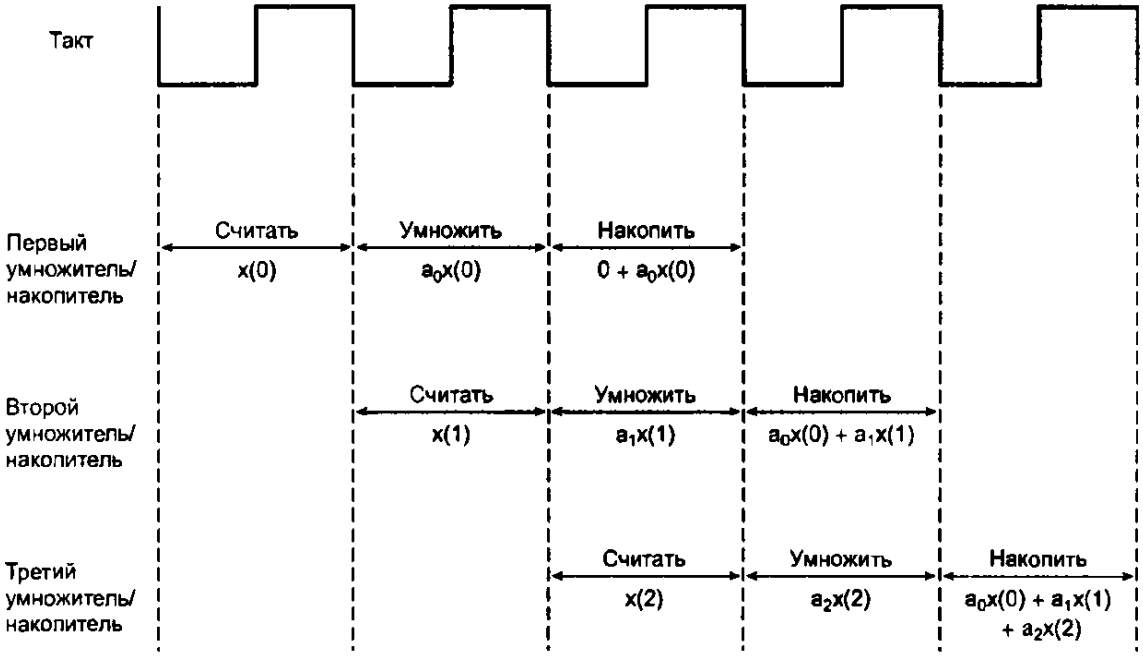


Рис. 4.9. Временная диаграмма блока умножителя-накопителя с конвейерной обработкой

Алгоритмы ЦОС часто имеют итерационную, но сложную структуру, что делает их идеально подходящими для многоуровневой конвейерной обработки. Например, БПФ требует постоянного вычисления по схеме «бабочка». Хотя каждое такое вычисление требует иных данных и коэффициентов, основная арифметическая операция в схеме «бабочка» остается без изменений. Следовательно, такие арифметические устройства, как процессоры БПФ, можно модернизировать, сыграв на этой особенности вычислений. Конвейерная обработка гарантирует постоянный поток команд на центральный процессор и в общем случае приводит к значительному увеличению пропускной способности системы. В то же время, иногда конвейерная обработка может порождать проблемы. Например, в некоторых процессорах ЦОС конвейерная обработка может привести к выполнению нежелательной команды, особенно при наличии команд ветвления, и разработчику следует помнить, что такая возможность существует.

Регистры конвейера используются как временное хранилище для коэффициентов и пар выборок данных. Регистр произведения также служит временным хранилищем произведения.

4.2.3. Аппаратный умножитель-накопитель

Основными численными операциями в цифровой обработке сигналов являются умножение и сложение. Умножение в программной форме печально знаменито своей трудоемкостью, а если используется арифметика с плавающей запятой, сложение оказывается даже еще более трудоемкой операцией. Чтобы максимально ускорить цифровую обработку сигналов в реальном времени, рекомендуется использовать специализированные аппаратные умножители-накопители с арифметикой с плавающей или фиксированной запятой. Такое аппаратное обеспечение теперь стандартно используется во всех цифровых процессорах сигналов. В процессоре с фиксированной запятой аппаратный умножитель за один такт (обычно 25 нс) принимает два 16-битовых дробных числа, представленных в форме дополнения до двух, и вычисляет их 32-битовое произведение. Среднее время выполнения команды умножения-накопления можно значительно уменьшить, если использовать специальные команды повторения.

При заполненном конвейере операция в блоке завершается каждый такт (200 нс).

Типичная конфигурация аппаратного умножителя-накопителя ЦОС изображена на рис. 4.10. В такой конфигурации умножитель имеет пару входных регистров, которые содержат входы умножителя, и 32-битовый регистр произведения, который содержит результат умножения. Выход регистра P (*product* – произведение) соединяется с накопителем двойной точности, в котором накапливаются произведения.

Принцип весьма сходен с тем, что используется в аппаратных умножителях-накопителях с плавающей запятой, за исключением того, что входы и произведения – это нормированные числа, представленные в форме с плавающей запятой. Умножители-накопители с плавающей запятой позволяют быстро вычислять результаты алгоритмов ЦОС с минимальным ошибками. Такие алгоритмы ЦОС, как КИХ- и БИХ-фильтрация, подвержены эффектам конечной разрядности (квантование коэффициентов и арифметические ошибки). Операции с плавающей запятой позволяют использовать большой динамический диапазон и снижают арифметические ошибки, хотя во многих приложениях достаточно динамического диапазона, который предлагает представление с фиксированной запятой.

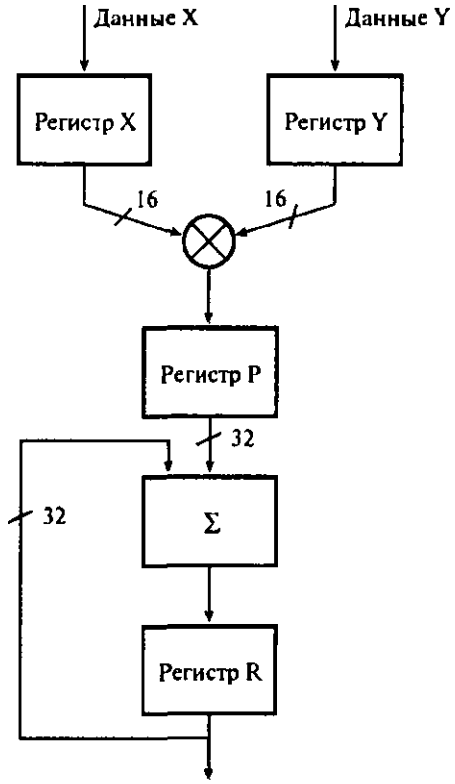


Рис. 4.10. Типичная для ЦОС конфигурация умножителя-накопителя

4.2.4. Специальные команды

Процессоры ЦОС предлагают некоторые специальные команды, оптимизированные для цифровой обработки сигналов. Эти команды обладают двойным преимуществом. Во-первых, они позволяют создавать более компактный код, который занимает меньше места в памяти (получается практически такой же компактный код, как и программа, написанная на языке высокого уровня, например, С). Во-вторых, они увеличивают скорость выполнения алгоритмов ЦОС. Специальные команды, внедренные в микросхемы ЦОС, включают: 1) команды, поддерживающие базовые операции ЦОС; 2) команды, которые снижают служебные издержки для организации циклов команд; 3) команды, ориентированные на конкретное приложение.

Многие ключевые алгоритмы ЦОС, такие как алгоритмы цифровой фильтрации и поиска корреляции, требуют смещения или задержки данных, чтобы освободить место для новых выборок данных. Процессоры ЦОС предлагают специальные команды, которые позволяют копировать выборку, находящуюся в ячейке памяти со следующим порядковым адре-

сом, так же, как будто выборка извлечена из памяти или обрабатывается в данный момент, причем все это за один такт. Например, во втором поколении семейства TMS320 процессоров ЦОС пара команд LTD и MPY позволяет одновременно загружать данные во временный регистр для умножителя, смещать данные (для реализации единичной задержки, которая обозначается z^{-1}) и накапливать произведения. Специальные команды, которые поддерживают операции умножения – накопления со сдвигом данных, в современных процессорах ЦОС стали стандартными.

Специальные команды также позволяют ускорить часто повторяющиеся операции ЦОС. Например, во втором поколении процессоров семейства TMS320 есть команда, которая позволяет повторить следующую за ней команду заданное число раз. Поскольку команда повтора требует только одной операции извлечения из памяти, фрагмент кода, который обычно требует цикла в несколько тактов, эффективно превращается в однотактовую команду. Команды повтора особенно полезны в цифровой обработке сигналов для расчета внутренних циклов, например, в КИХ- и адаптивной фильтрации, и записи данных – в порядке, определяемом битреверсивными битами при вычислении БПФ.

Напомним, что КИХ-фильтры характеризуются следующим уравнением:

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

где N – длина фильтра.

В процессоре TMS320C50, например, уравнение КИХ-фильтрации можно эффективно реализовать с использованием пары команд:

```
RPT
NM1
MACD      HNM1,XNM1
```

Первая команда RPT NM1 загружает длину фильтра минус 1 (т.е. $N - 1$) в счетчик повторения команды и приводит к N-кратному повторению следующей за ней команды умножения – накопления с перемещением данных (MACD). Команда MACD выполняет за один такт несколько команд:

1) умножает выборку данных $x(n - к)$, которая находится в памяти для хранения данных, на коэффициент $h(k)$, который находится в памяти для хранения программы;

2) добавляет предыдущее произведение к накапливающему сумматору;

3) реализует единичную задержку, которая обозначается как z^{-1}, путем смещения выборки данных $x(n - \kappa)$ в ячейку со следующим по возрастанию адресом для обновления линии задержки с отводами.

В семействе процессоров ЦОС Motorola DSP56000, как и в семействе TMS320, для эффективной реализации КИХ-фильтра можно использовать команду MAC вместе с командой повторения (REP):

```
RPT   #N-1
MAC   X0,Y0,A      X:(R0)+,X0          У:(R4)+,Y0
```

Здесь команда повторения используется вместе с командой MAC для выполнения длительных умножений и суммирования произведений. Снова отметим возможность осуществить умножение с помощью одной команды, что позволяет иметь несколько тактов обработки данных. Содержимое регистров XO и Y0 перемножается, а результат добавляется к накопителю. В это же время следующая выборка данных и соответствующий коэффициент извлекаются для умножения из ячеек X и Y.

В наиболее современных процессорах ЦОС была развита концепция повторения команды и введена команда, которая позволяет повторять заданное число раз не только одну команду, но и блок кода. В семействе TMS320 (например, TMS320C50, TMS320C54 и TMS320C30) используется такой формат вычисления повторяющего блока команд (цикл с нулевыми служебными издержками):

```
RPTB loop
:
:
loop (последняя команда)
```

Команды повторения, предлагаемые некоторыми процессорами ЦОС, имеют функции, характерные для языков высокого уровня. В семействах Motorola DSP56000 и DSP56300 есть циклы DO с нулевыми служебными издержками, которые также могут быть вложенными. Приведенный ниже пример иллюстрирует вложенный цикл DO, в котором внешний цикл выполняется *N* раз, а внутренний – *NM* раз.

```
DO #N, LOOP1
:
DO #M, LOOP2
:
:
LOOP2 (здесь располагается последняя команда)
:
LOOP1 (здесь располагается последняя команда внешнего цикла)
```

Вложенные циклы полезны при эффективной реализации таких функций ЦОС, как алгоритмы БПФ и двумерная обработка сигналов.

Процессоры ЦОС, производимые Analog Devices (например, ADSP-2115 и SHARC), также позволяют организовать вложенные циклы. Процессор ADSP-2115 поддерживает до четырех уровней вложения циклов. При вложении используется следующий формат:

```
CNTR=N
DO LOOP UNTIL CE
:
:
LOOP: (последняя команда цикла)
```

Цикл повторяется до обнуления счетчика. Цикл может содержать большой блок команд. Формат вложения циклов, по сути, идентичен формату, который используется в процессорах семейства DSP56000.

Адаптивная фильтрация – это еще одна ключевая характерная функция современной обработки сигналов, и для ее поддержки также были реализованы специальные команды. В адаптивной фильтрации важным шагом является обновление коэффициентов и вычисление нового набора коэффициентов на основе старого. Например, в адаптивных фильтрах, основанных на схеме наименьших квадратов, задача обновления коэффициентов характеризуется следующим уравнением:

$$h_{k+1}(i) = h_k(i) + 2\mu e_k x(k-i), \tag{4.3}$$

где

$$e_k = y_k - \hat{y}_k, \tag{4.4}$$

а $\hat{y}_k$, выход адаптивного фильтра, выражается следующим образом:

$$\hat{y}_k = \sum_{i=0}^{N-1} h_k(i) x(k-i). \tag{4.5}$$

Обычно часть задачи обновления коэффициентов, которая относится к КИХ-фильтрации (уравнение (4.5), реализуется с использованием специальных команд умножения – накопления, как описывалось ранее. Задачу обновления коэффициентов (уравнение (4.3) можно выполнить с помощью команды повторения блока с нулевыми служебными издержками на организацию цикла. В процессоре TMS320C54 рассчитывать выход адаптивного фильтра и обновлять коэффициенты фильтра можно с помощью команды адаптивной фильтрации на базе схемы наименьших квадратов LMS, команды умножения ST//MPY и команды повтора блока RPTBD. Указан-

ные команды позволяют снизить время выполнения и размер кода адаптивных фильтров.

В быстром преобразовании Фурье (БПФ), другой ключевой операции ЦОС, всегда требуется скремблировать последовательность входных данных до БПФ и восстанавливать обычный порядок после БПФ, чтобы гарантировать, что точки данных идут в правильной последовательности. Все высококачественные универсальные процессоры ЦОС предлагают специальные команды для адресации с инвертированием битов, позволяющие выполнять требуемое скремблирование/восстановление одновременно с извлечением выборок данных из памяти или перемещением выборок.

Например, адресацию с инвертированием битов в процессорах TMS320 можно использовать для обращения битов при занесении в память *N*-точечной комплексной последовательности данных:

```
RPT N2
BLDD #XN,   *BR0+
```

В данном случае используются комплексные входные данные, так что каждая выборка состоит из действительной и мнимой частей. Следовательно, каждая выборка требует двух ячеек памяти.

Современные процессоры ЦОС также предлагают ориентированные на приложения команды для таких областей, как кодирование речи (например, при поиске в кодовой книге), цифровое аудио (например, в системах объемного звучания) и телекоммуникации (например, в декодировании Витерби).

4.2.5. Средства дублирования

В цифровой обработке сигналов дублирование подразумевает использование нескольких стандартных блоков, например, более одного АЛУ, умножителя или ячейки памяти. Часто блоки располагаются так, чтобы их можно было использовать параллельно. В ЦОС нормой является наличие одного центрального процессора с дублированными одним или несколькими арифметическими устройствами.

В то же время в области ЦОС уже используется концепция полнофункциональной параллельной обработки, когда, например, над одной задачей работают несколько независимых процессоров или несколько процессоров под общим управлением. На настоящий момент на рынке представлено несколько процессоров ЦОС с параллельной обработкой, например, TMS320C40 и ADSP-21060 SHARC.

4.2.6. Встроенная память/кэш

В большинстве случаев чипы ЦОС работают настолько быстро, что медленная недорогая память не может поддерживать такой темп. Обычно при этом процессор замедляется – вводятся состояния ожидания. В некоторых процессорах состояния ожидания имеют программное управление, хотя иногда для замедления процессора требуются внешние аппаратные блоки. Разумеется, наличие состояний ожидания означает, что процессор не может работать с полной скоростью.

Чтобы решить описанную проблему, во многие чипы ЦОС встраивают скоростные внутренние ОЗУ и/или ПЗУ данных. В таких процессорах низкоскоростная внешняя память может использоваться для хранения программного кода. При инициализации код может передаваться в быструю внутреннюю память для выполнения операций на полной скорости. Для реализации в реальном времени и создания окончательных прототипов полезны скоростные встроенные стираемые программируемые ПЗУ (СППЗУ). Некоторые чипы имеют также встроенный программный кэш, который может использоваться часто повторяемыми блоками программы. Если код выполняется в кэше, можно не вызывать лишний раз команды из памяти, ускоряя выполнение программы.

В настоящее время встроенная память используется повсеместно.

4.2.7. Расширенный параллелизм – SIMD, VLIW и статическая суперскалярная обработка

В настоящее время в архитектуре процессоров ЦОС наблюдается тенденция к повышению производительности за счет увеличения числа команд, выполняемых в каждом такте, и числа операций, выполняемых при вызове одной команды [26 – 29].

В новейших архитектурах процессоров ЦОС для повышения вычислительной эффективности интенсивно используются методы параллельной обработки. Наиболее часто применяются (иногда совместно) следующие три техники: архитектура SIMD (*single instruction, multiple data* – с одним потоком команд и многими потоками данных, также ОКМД – одна команда, много данных), архитектура VLIW (*very-large-instruction-word* – архитектура с командными словами сверхбольшой длины) и суперскалярная обработка [30].

Обработка с архитектурой SIMD используется для увеличения числа операций, выполняемых при вызове одной команды. Обычно в процессо-

рах ЦОС с архитектурой SIMD процессор имеет несколько трактов передачи данных и несколько операционных блоков. Следовательно, одна команда может передаваться нескольким операционным блокам для обработки блоков данных одновременно, увеличивая число операций, выполняемых за один такт (см., например, рис. 4.11). В контексте рис. 4.11 процессор ЦОС имеет два операционных блока, каждый со своим АЛУ, умножителем-накопителем и схемой сдвига. Процессор может выполнять две различных арифметических операции (например, сложение и умножение – накопление) одновременно в течение одной команды. В число процессоров ЦОС с архитектурой SIMD входят DSP16000 (Lu-cent), TMS320C62x (*Texas Instruments*) и TigerSHARC (*Analog Devices*) ADSP-TS001 (*Analog Devices*)[2].

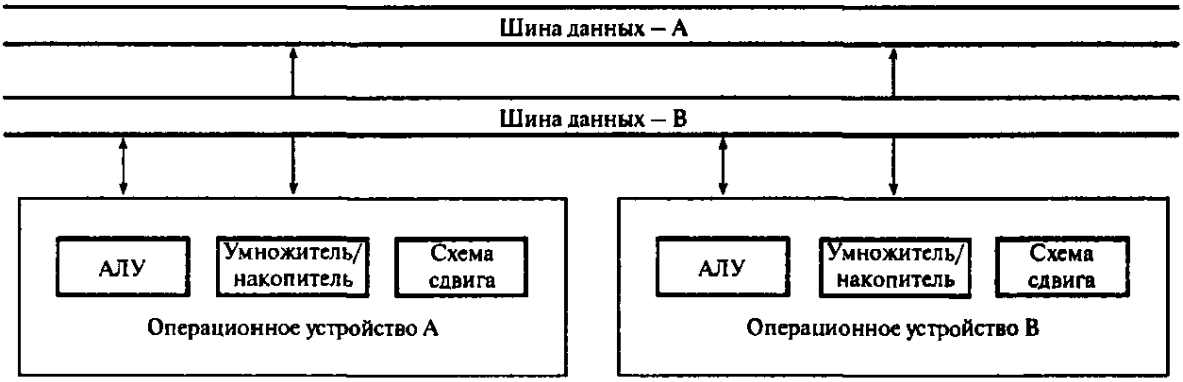


Рис. 4.11. Двойные арифметические устройства и двойные тракты передачи данных для обработки по схеме SIMD

Привлекательной особенностью архитектур SIMD, особенно архитектур, которые поддерживают данные нескольких размеров, является возможность эффективного увеличения числа доступных операционных блоков, а, следовательно, числа операций, выполняемых за такт, путем разделения блоков. Например, в процессоре TigerSHARC два доступных умножения можно разделить для одновременного выполнения четырех операций умножения – накопления 16 × 16 бит вместо одной операции умножения – накопления 32 × 32 бит (рис. 4.12).

Очевидно, что в приложениях, где данные обрабатываются параллельно, архитектура SIMD может значительно повысить эффективность работы процессора. Впрочем, в приложениях с последовательными данными возможность увеличения вычислительной эффективности за счет обработки по схеме SIMD незначительна. По этой причине в следующем поколении процессоров ЦОС, нацеленных на многоканальные приложения

(такие, как системы мобильной связи третьего поколения), используются возможности обработки с одним потоком команд и многими потоками данных (SIMD).

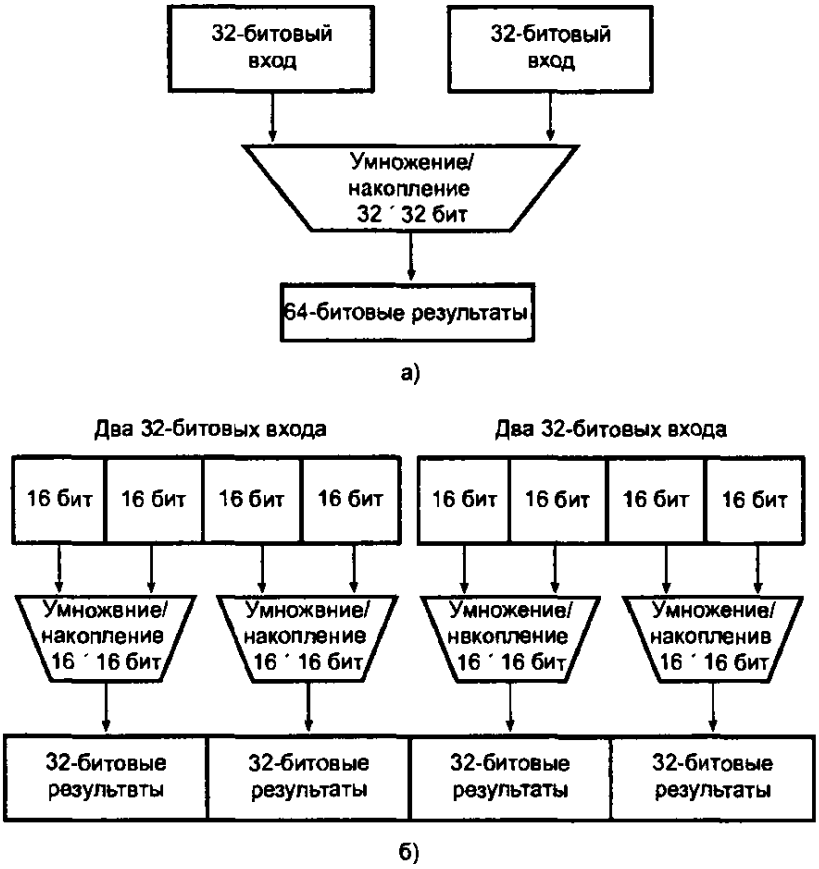


Рис. 4.12. Иллюстрация использования обработки по схеме SIMD и поддержки данных нескольких размеров для увеличения в процессоре TigerSHARC числа умножителей-накопителей с одного до четырех

Обработка *с командными словами сверхбольшой длины* (*Very Long Instruction Word* – VLIW) позволяет существенно увеличить число команд, обрабатываемых за такт [29]. По сути, *командное слово сверхбольшой длины* – это конкатенация нескольких *коротких* команд, и для выполнения нескольких коротких команд за один такт требуется несколько операционных блоков, функционирующих параллельно.

Принципы архитектуры VLIW и схема потока данных улучшенных процессоров ЦОС с фиксированной запятой TMS320C62x иллюстрируются на рис. 4.13. Центральный процессор содержит два тракта передачи данных и восемь независимых операционных блоков, организованных в два набора – (L1,S1, M1 и D1) и (L2, S2, M2 и D2). В данном случае каждая *короткая* команда имеет длину 32 бит. Восемь таких команд вместе

образуют пакет *командного слова сверхбольшой длины*, который может обрабатываться параллельно несколькими устройствами.

Обработка по схеме VLIW начинается с извлечения центральным процессором из встроенной программной памяти пакета команд (восемь 32-битовых команд). Восемь команд в *вызванном* пакете формируют *исполняемый* пакет, если они могут выполняться параллельно, а затем (по обстановке) передается восьми операционным блокам. Следующий 256-битовый пакет команд вызывается из программной памяти во время декодирования и выполнения исполняемого пакета. Если восемь команд в вызванном пакете нельзя выполнять параллельно (например, если все восемь команд – команды умножения/накопления, в течение одного такта можно выполнять только две из них, так как существует всего два умножителя), тогда формируется несколько исполняемых пакетов, которые передаются операционным блокам по одному за раз. Вызываемый пакет всегда имеет длину 256 бит (восемь команд), но размер исполняемого пакета может составлять от одной до восьми команд.

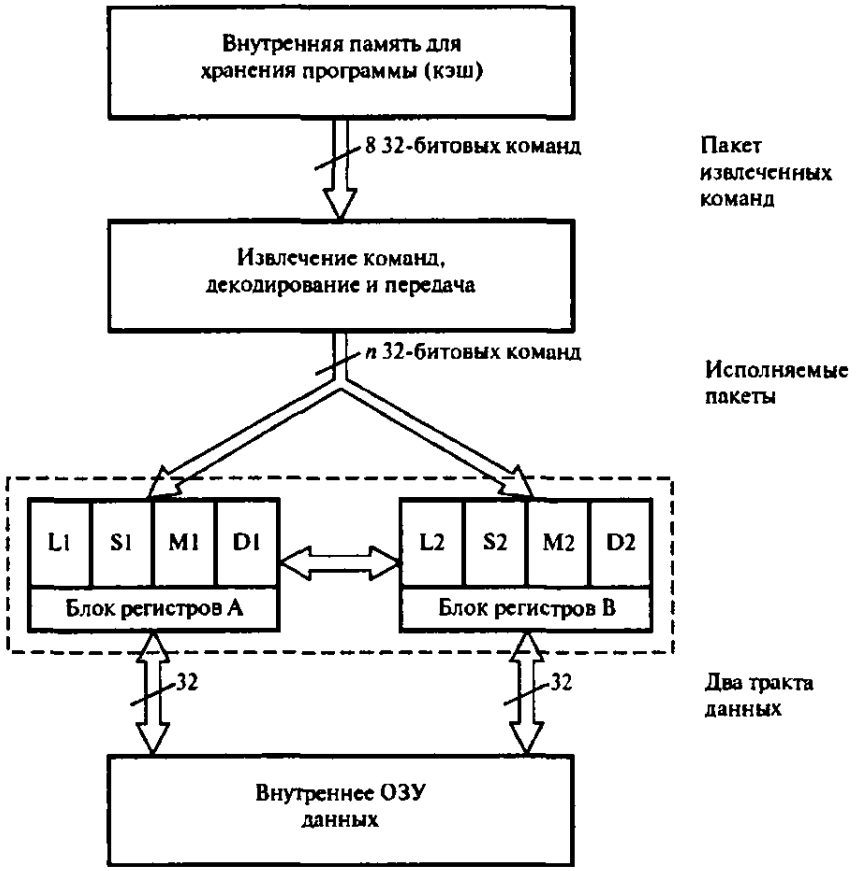


Рис. 4.13. Принципы архитектуры с командными словами сверхбольшой длины (*Very Long Instruction Word* – VLIW) и схема потока данных в процессорах TMS320C62X: LI, L2 – логические элементы; SI, S2 – схемы сдвига/логические элементы; Ml, M2 – умножители; Dl, D2 – адресные элементы

Архитектура VLIW предназначена для поддержки параллелизма на уровне команд. Если объединить эту архитектуру с высокой тактовой частотой (обычно 200 МГц), обычно получаются весьма эффективные процессоры ЦОС. Следует, правда, помнить, что в семействе TMS320C62x, например, параллельная обработка команд планируется во время компиляции. Впрочем, если команды нельзя выполнять параллельно, вычислительная эффективность подобных процессоров не используется.

Другой способ увеличить скорость выполнения команд процессора ЦОС (число команд, обработанных за такт) состоит в *суперскалярной* обработке, в которой используется параллелизм на уровне команд. Традиционно термином *суперскалярный* называют архитектуру компьютера, позволяющую выполнять несколько команд за один такт [30].

Подобные архитектуры широко используются в таких универсальных процессорах, как PowerPC и Pentium.

Суперскалярные процессоры ЦОС содержат несколько операционных блоков, и несколько команд могут передаваться этим блокам для параллельной обработки. Кроме того, для дальнейшего увеличения производительности интенсивно эксплуатируется метод параллельной обработки.

Наиболее известным суперскалярным процессором ЦОС является TigerSHARC (*Analog Devices*) (рис. 4.14). Процессор TigerSHARC описан в спецификации как статический суперскалярный процессор ЦОС, поскольку параллелизм команд определяется до их выполнения. Фактически процессор TigerSHARC объединяет концепции SIMD, VLIW и суперскалярной обработки. Этот улучшенный процессор ЦОС имеет несколько тактов обработки данных и два набора независимых операционных блоков, в каждом из которых имеется умножитель, АЛУ, 64-битовая схема сдвига и блок регистров (см. рис. 4.14). TigerSHARC – это процессор с плавающей запятой, но он также поддерживает арифметику с различными типами данных (8-, 16- и 32-битовые числа).

Длина команды в процессоре TigerSHARC не фиксирована. В каждом такте из внутренней памяти извлекается до четырех 32-битовых команд, которые передаются двум параллельным наборам операционных блоков. Команды могут передаваться и параллельно двум блокам (архитектура SIMD – Single Instruction, Multiple Data – с одним потоком команд и многими потоками данных) или каждому операционному блоку независимо. Каждый операционный блок (АЛУ, умножитель или схема сдвига) получает свой вход от блока регистров и возвращает результаты ему же. Блоки регистров соединены с тремя трактами передачи данных, так что

они могут одновременно в одном такте считывать два входа и записывать выход в память. Такая архитектура загрузки – хранения подходит для стандартных операций ЦОС, у которых часто бывает два входных значения и одно выходное.

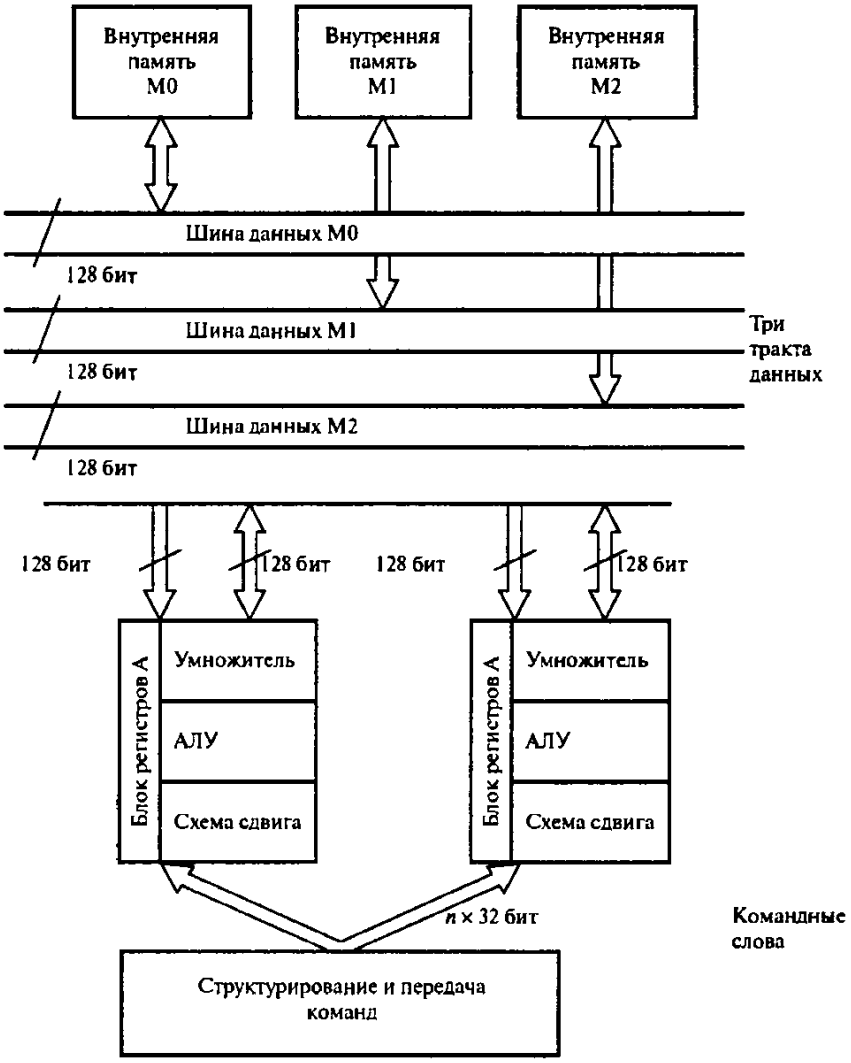


Рис. 4.14. Принципы суперскалярнон архитектуры и схема потока данных в процессоре ЦОС TigerSHARC

Как обсуждалось ранее, поскольку процессор может работать с несколькими размерами данных (8-, 16-, 32- и 64-битовые числа), возможна организация параллельной обработки на следующих уровнях. Следовательно, в каждом такте процессор TigerSHARC может выполнять до восьми операций сложения – вычитания и восемь операций умножения – накопления с 16-битовыми входами вместо двух операций умножения – накопления с 32-битовыми входами. Возможность обработки смешанных типов данных и разбиения больших командных слов на отдельные коман-

ды для операционных блоков позволяет процессору интенсивно использовать параллелизм на уровне команд.

Стоит также отметить, что в процессорах ЦОС, использующих такие улучшенные архитектуры, как VLIW и суперскалярная обработка, для эффективного использования параллельных операционных блоков необходима некоторая форма статического планирования команд перед выполнением программы. Кроме того, следует учитывать возможную зависимость по данным (например, результаты требуются до того, как они готовы) и зависимость по управлению (например, при использовании команд ветвления), что может представлять проблемы при параллельной обработке.

4.3. Универсальные процессоры ЦОС

Универсальные процессоры ЦОС – это, по сути, высокоскоростные микропроцессоры с гарвардской архитектурой и наборами команд, оптимизированных под операции ЦОС. В этих процессорах везде, где только можно, выполнение трудоемких операций (таких, как сдвиг/масштабирование, умножение и т.д.) облегчается за счет интенсивного использования гарвардской архитектуры, конвейерной обработки и специализированного аппаратного обеспечения.

Универсальные процессоры ЦОС значительно развились за последнее десятилетие в результате непрекращающихся поисков лучших способов выполнения операций ЦОС с точки зрения вычислительной эффективности, легкости использования, стоимости, расхода энергии, размера и требований приложения [26]. Ненасытный спрос на улучшенную вычислительную эффективность привел к существенному снижению времени выполнения команд, увеличению тактовой частоты и, что более важно, – к усложнению аппаратных и программных архитектур. В настоящее время обычным является наличие специализированных встроенных арифметических аппаратных устройств (например, для поддержки быстрых операций умножения – накопления), большой встроенной памяти со множественным доступом и специальных команд для эффективного выполнения операций ЦОС во внутреннем ядре. Авторы также отмечают тенденцию к увеличению размера слова (например, для поддержания качества сигнала) и более интенсивному использованию параллелизма (для увеличения числа команд, выполняемых в одном такте, и числа операций, выполняемых при вызове одной команды). Таким образом, по мнению авторов, в новых универсальных процессорах ЦОС увеличивается число трактов обработки данных и используются арифметики, поддерживающие параллельные

операции. Вводятся процессоры ЦОС, основанные на архитектурах SIMD (*Single Instruction, Multiple Data* – архитектура с одним потоком команд и многими потоками данных), VLIW (*Very Long Instruction Word* – архитектура с командными словами сверхбольшой длины) и суперскалярной архитектуре для поддержки эффективной параллельной обработки. В некоторых процессорах ЦОС, чтобы дополнительно повысить производительность, используют специализированные встроенные сопроцессоры для ускорения таких специфических алгоритмов ЦОС, как КИХ-фильтрация и декодирование Витерби. Вообще, на развитие процессоров ЦОС значительное влияние оказало взрывоподобное развитие технологий в сфере связи и цифровых аудиосистемах, а также рост числа приложений, в которых используются встроенные процессоры ЦОС.

В следующих двух пунктах кратко описаны архитектурные особенности нескольких поколений процессоров ЦОС с фиксированной и плавающей запятой.

4.3.1. Процессоры ЦОС с фиксированной запятой

Стандартная архитектура семейства процессоров ЦОС с фиксированной запятой *первого поколения* TMS320Clx, впервые использованная в 1982 году Texas Instruments, изображена на рис. 4.15. Ключевые особенности TMS320Clx – выделенные арифметические устройства, которые включают умножитель и накопитель. Процессоры семейства имеют модифицированную гарвардскую архитектуру с двумя отдельными областями памяти для данных и программы. Кроме того, процессоры имеют встроенную память и несколько специальных команд для выполнения основных алгоритмов ЦОС.

Доступные на сегодня процессоры ЦОС с фиксированной запятой отличаются друг от друга деталями архитектуры и встроенными ресурсами. Резюме по ключевым особенностям четырех поколений процессоров ЦОС с фиксированной запятой, созданных основными производителями полупроводниковых технологий, представлено в табл. 4.1. Условная классификация процессоров ЦОС на четыре поколения основана на исторических причинах, архитектурных особенностях и вычислительной эффективности.

Процессоры ЦОС с фиксированной запятой *второго поколения* имеют значительно улучшенные архитектуры по сравнению с процессорами первого поколения. В большинстве случаев в число этих особенно-

стей входили большая встроенная память и много специальных команд для поддержки эффективного выполнения алгоритмов ЦОС. В результате вычислительная производительность процессоров ЦОС второго поколения в 4 – 6 раз превышает производительность процессоров первого поколения.

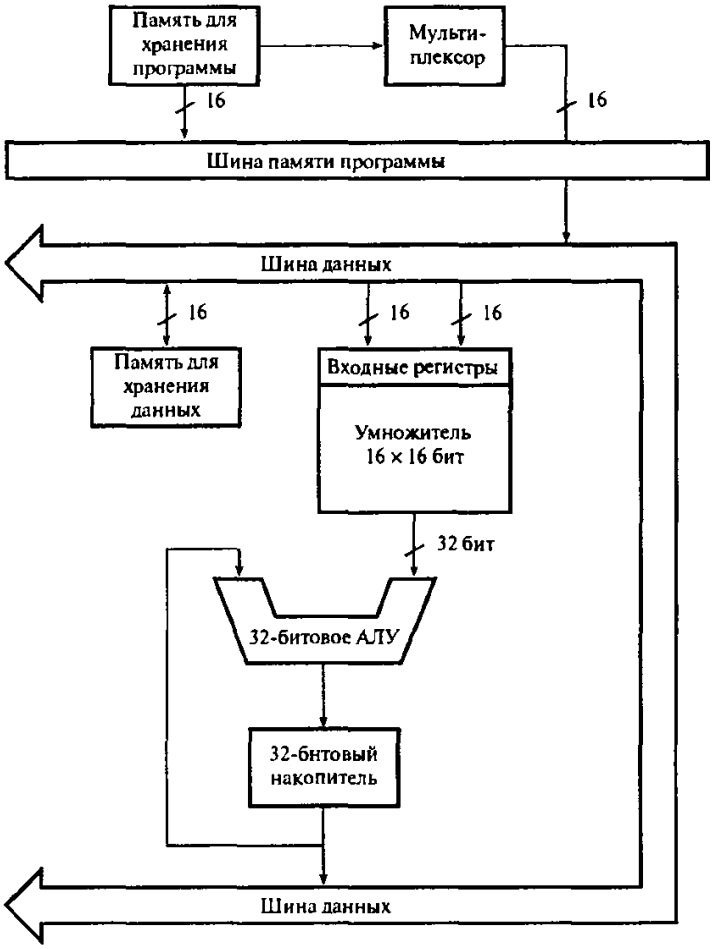


Рис. 4.15. Упрощенная архитектура процессора ЦОС с фиксированной запятой первого поколения (Texas Instruments TMS320C10)

Типичными представителями процессоров ЦОС второго поколения являются семейства TMS320C5x (Texas Instruments), DSP5600x (Motorola), ADSP21xx (Analog Devices) и DSP16xx (Lucent Technologies). Процессоры ЦОС первого и второго поколений производства Texas Instruments имеют много общего архитектурно, но процессоры второго поколения имеют больше особенностей и более высокую скорость (см. табл. 4.1). Внутренняя архитектура процессоров семейства TMS320C5x показана на рис. 4.16 (в упрощенном виде, чтобы акцентировать внимание на двойной области памяти, характерной для гарвардской архитектуры).

Таблица 4.1

Особенности универсальных процессоров ЦОС с фиксированной запятой производства Texas Instruments, Motorola и Analog Devices

Поколение	ЦОС с фиксированной запятой	Ширина тракта данных (биты)	Число трактов данных	Длина слова накопителя (биты)	Длина слова (биты)	Длина команды (биты)	Размер встроенного ОЗУ (слова)	Размер кэша команд (число команд)	Число умножителей	Коэффициент производительности
1	TMS320C10 (Texas Instruments)	16	1	16	32	16	144	–	1	–
2	TMS320C50 (Texas Instruments)	16	2	16	32	16	10 К	–	1	10 @ 50 МГц
	DSP56002 (Motorola)	24	2	24	56	24	1 К	–	1	13 @ 50 МГц
	DSP-2100 (Analog Devices)	16	2	16	40	24	32 К	16	1	13 @ 52 МГц
	1600 (Lucent Technologies)	16	2	16	36	16	–	15	1	22 @ 120 МГц
3	TMS320C54 (Texas Instruments)	16	3	16	40	16	32 К	–	1	25 @ 100 МГц
	DSP56300 (Motorola)	24	3	24	56	24	–	3 К	1	25 @ 100 МГц
	16000 (Lucent Technologies)	32	2	32	40	32	127 К	31	2	36 @ 100 МГц
4	TMS320C6200	–	2	–	40	256	17 К	64 К	2	86 @ 133,6 МГц

* Коэффициент производительности рассчитывается на основе скорости расчета контрольных ядер/вычисления контрольных алгоритмов ЦОС.

В число специальных команд для выполнения операций ЦОС входит команда умножения и накопления со смещением данных, которую, например, можно объединить с командой повтора и реализовать КИХ-фильтр со значительной экономией времени. Возможность адресации с инвертированием битов полезна при БПФ. Кроме того, в отличие от процессоров первого поколения, процессоры семейства С5х имеют большую встроенную память.

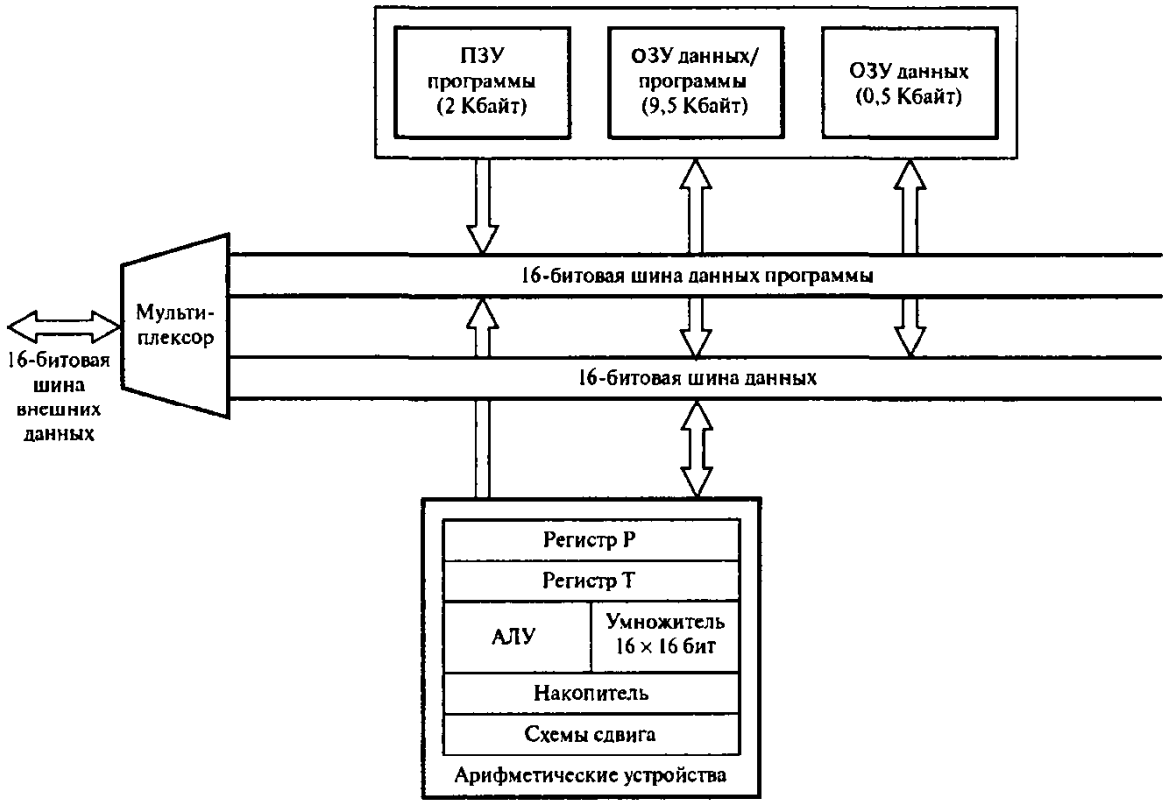


Рис. 4.16. Упрощенная архитектура процессора ЦОС с фиксированной запятой второго поколения (Texas Instruments TMS320C50)

Процессор DSP5600x (Motorola) является высокоточным цифровым процессором сигналов с фиксированной запятой. Его архитектура изображена на рис. 4.17. Он имеет две независимые области памяти для хранения данных (области X и У) и одну область для хранения программ. Наличие двух отдельных областей для хранения данных позволяет естественно разбить данные для операций ЦОС и облегчает выполнение алгоритмов ЦОС. Например, в графических приложениях данные могут записываться как данные, относящиеся к координате X, и данные, относящиеся к координате У, при КИХ-фильтрации – как коэффициенты и данные, в БПФ – как действительная и мнимая части числа. В ходе выполнения программы за

один такт можно одновременно извлечь пару выборок из внутренней памяти или занести пару выборок во внутреннюю память. Внешне две области данных уплотняются в единую шину данных, что несколько снижает преимущества двойной внутренней памяти. Кроме того, процессоры семейства имеют следующие арифметические устройства: два 56-битовых накопителя и однотактовый аппаратный умножитель-накопитель с фиксированной запятой.

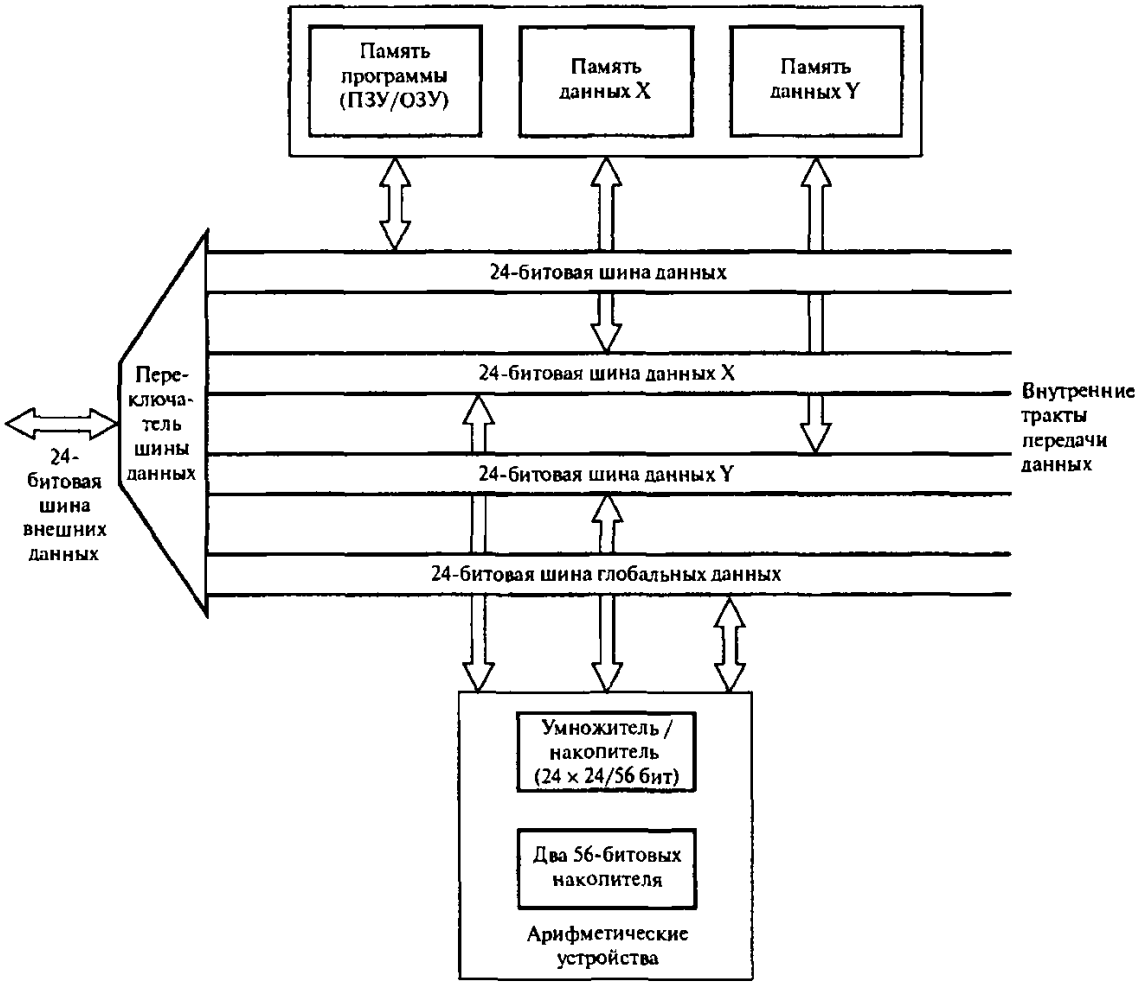


Рис. 4.17. Упрощенная архитектура процессора ЦОС с фиксированной запятой второго поколения (Motorola DSP56002)

Умножитель-накопитель принимает 24-битовые входы и выдает 56-битовое произведение. Размер 24 бит обеспечивает достаточную точность для представления большинства переменных ЦОС, тогда как 56-битовый накопитель (включая 8 защитных бит) предотвращает арифметическое переполнение. Указанных длин слов достаточно для большинства приложений, включая цифровое аудио с его весьма строгими требованиями. Процессоры TMS5600x предлагают также специальные команды, по-

зволяющие организовывать циклы с нулевыми служебными издержками, кроме того, существует возможность адресации с инвертированием битов, чтобы скремблировать входные данные перед быстрым преобразованием Фурье или восстанавливать правильный порядок данных после БПФ.

Другим семейством процессоров ЦОС с фиксированной запятой второго поколения является ADSP21xx (Analog Devices), имеющий две раздельные области памяти – в одной хранятся только данные, а в другой – данные и программный код. Упрощенная блок-схема внутренней архитектуры процессоров ADSP21xx изображена на рис. 4.18.

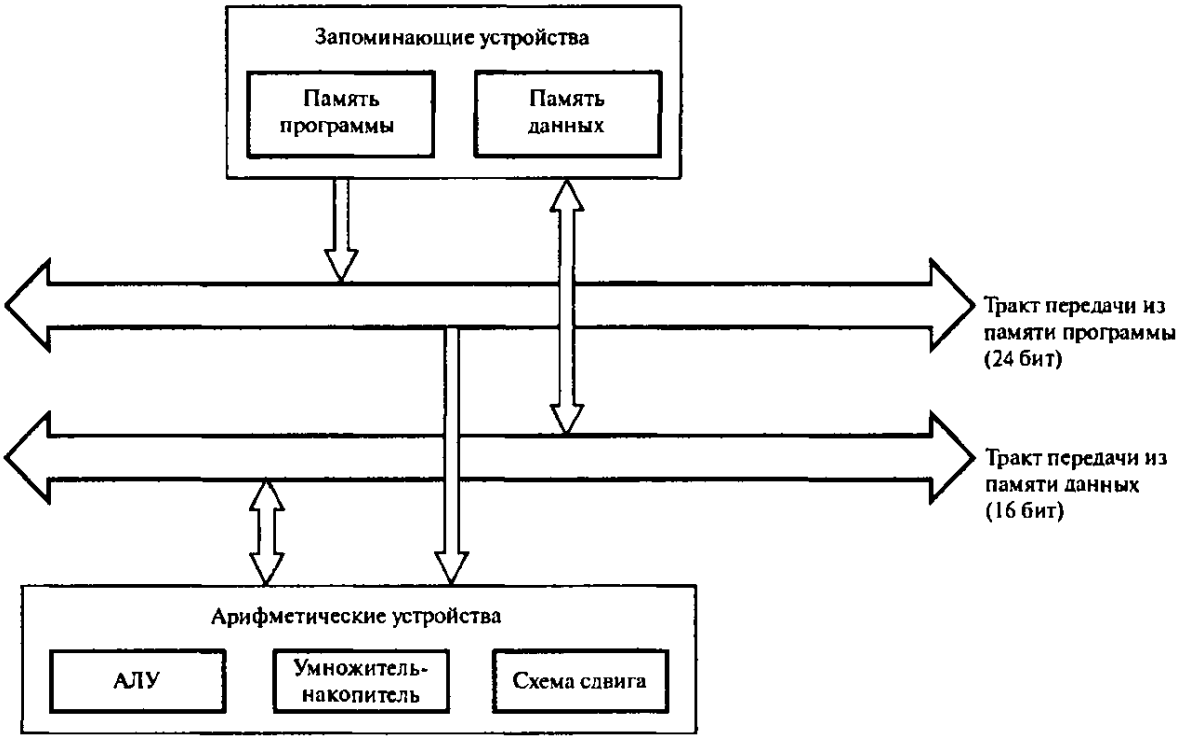


Рис. 4.18. Упрощенная архитектура процессора ЦОС с фиксированной запятой второго поколения (Analog Devices ADSP2100)

Основными компонентами являются АЛУ, умножитель-накопитель и схемы сдвига. Умножитель-накопитель за один такт принимает вход 16 × 16 бит и выдает 32-битовое произведение. Накопитель процессора ADSP21xx имеет 8 защитных бит, которые могут использоваться для расчетов с повышенной точностью. При создании процессора ADSP21xx разработчики отошли от стандартной гарвардской архитектуры, поскольку процессор позволяет хранить в программной памяти данные и команды программы. Чтобы указать, когда из программной памяти извлекаются данные, а не команды программы, используется сигнальная шина. Хранение данных в программной памяти препятствует равномерному потоку

данных через центральный процессор, поскольку извлечение данных и команд не может происходить одновременно. Чтобы избежать в этой связи значительного снижения производительности, в семействе ADSP21xx используется встроенный кэш программной памяти, который содержит 16 последних выполненных команд. Это устраняет потребность в извлечении из программной памяти повторяющихся команд (данная особенность весьма существенна при выполнении циклов программы). Процессоры ADSP21xx предлагают также специальные команды для организации циклов с нулевыми служебными издержками и поддерживают возможность адресации с инвертированием разрядов битов (требуется при БПФ). Процессоры семейства имеют большую встроенную память (для более интенсивной передачи данных предоставляется до 64 Кбайт внутреннего ОЗУ). Стоит также отметить, что данные процессоры поддерживают прямой доступ к памяти, и внешние устройства могут обмениваться данными и командами с ОЗУ процессора ЦОС без участия самого процессора.

Семейство процессоров ЦОС с фиксированной запятой DSP16xx (Lucent Technologies) (рис. 4.19) нацелено на рынок телекоммуникаций и модемов. С точки зрения вычислительной производительности это один из самых мощных процессоров второго поколения. Процессор имеет гарвардскую архитектуру и подобно другим процессорам второго поколения имеет два тракта передачи данных, тракты X и У. В набор его арифметических устройств входят специализированный умножитель 16×16 бит, 36-битовое АЛУ/схема сдвига (включает четыре защитных бит) и двойные накопители. Предлагаются специальные команды, например, для организации циклов команд и выполнения блоков команд с нулевыми служебными издержками.

Процессоры ЦОС с фиксированной запятой *третьего поколения* являются, по сути, улучшением процессоров предыдущего поколения. В общем случае повышение производительности достигается за счет более интенсивного и/или эффективного использования доступных встроенных ресурсов. По сравнению с процессорами ЦОС второго поколения процессоры третьего поколения отличаются большим числом трактов передачи данных (обычно три по сравнению со старыми двумя), более широкими трактами передачи данных, большими встроенной памятью и кэшем команд и в некоторых случаях – двойными умножителями-накопителями. В результате производительность процессоров ЦОС третьего поколения обычно в 2 – 3 раза превышает производительность процессоров второго поколения того же семейства (согласно [26]). Упрощенные архитектуры процессоров ЦОС третьего поколения TMS320C54x, DSP563x и DSP 16000 изображены на рис. 4.20, 4.21 и 4.22 соответственно.

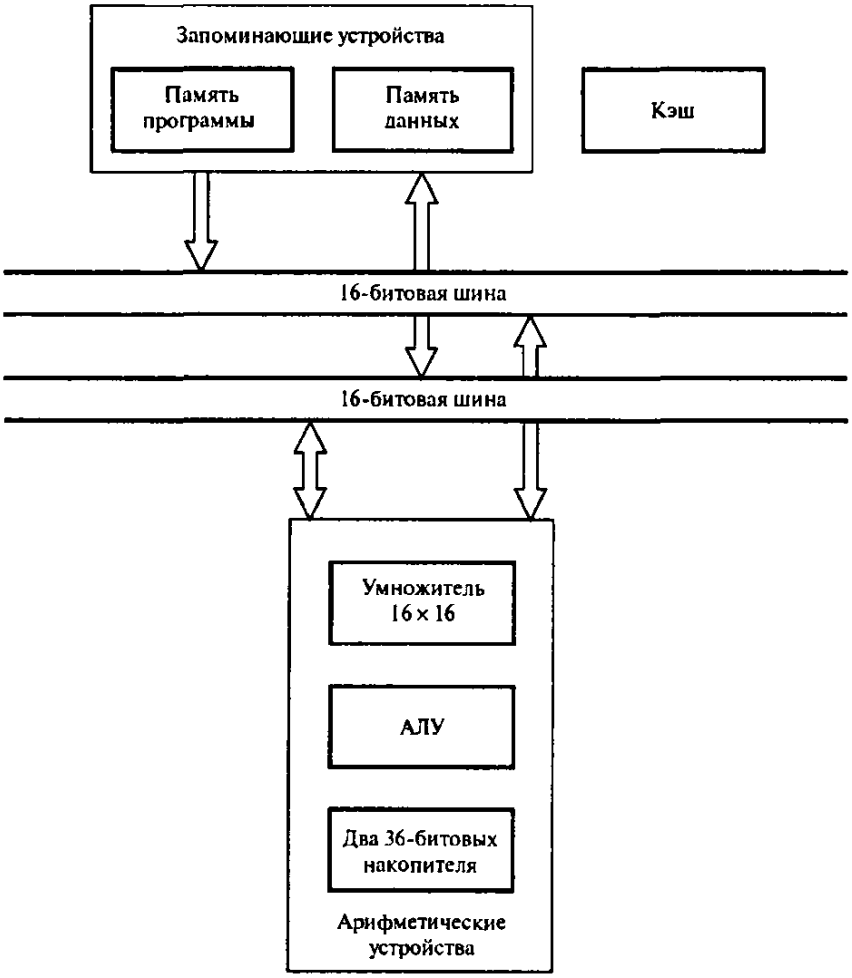


Рис. 4.19. Упрощенная архитектура процессора ЦОС с фиксированной запятой DSP16xx (Lucent Technologies)

Большинство процессоров ЦОС с фиксированной запятой третьего поколения нацелены на применение в цифровой связи и цифровом аудио (вообще, развитие этих приложений оказало огромное влияние на развитие процессоров ЦОС). Таким образом, авторы обнаружили в некоторых процессорах особенности, облегчающие использование процессоров в указанных сферах. Процессор TMS320C54x, например, включает специальные команды для адаптивной фильтрации (которая часто используется для эхоподавления и адаптивного выравнивания в сфере телекоммуникаций) и для поддержки декодирования Витерби. В процессорах третьего поколения производители полупроводниковых технологий также серьезно отнеслись к вопросу потребляемой мощности (вследствие его важности в таких портативных устройствах, как мобильные телефоны). Большинство процессоров ЦОС третьего поколения имеют низкий расход энергии и средство управления режимом электропитания.

Процессоры ЦОС с фиксированной запятой *четвертого поколения* с их новыми архитектурами нацелены в первую очередь на большие и/или развивающиеся многоканальные приложения, такие как построение медицинских изображений, цифровые абонентские линии, модемы удаленного доступа, беспроводные базовые станции и мобильные системы третьего поколения. Новой архитектурой с фиксированной запятой, которая привлекла значительную долю внимания в обществе ЦОС, стала *архитектура с командными словами сверхбольшой длины* (VLIW) (подробности представлены в п. 4.2). В архитектуре новых процессоров интенсивно используется параллелизм, при этом сохранены некоторые преимущества предыдущих процессоров ЦОС.

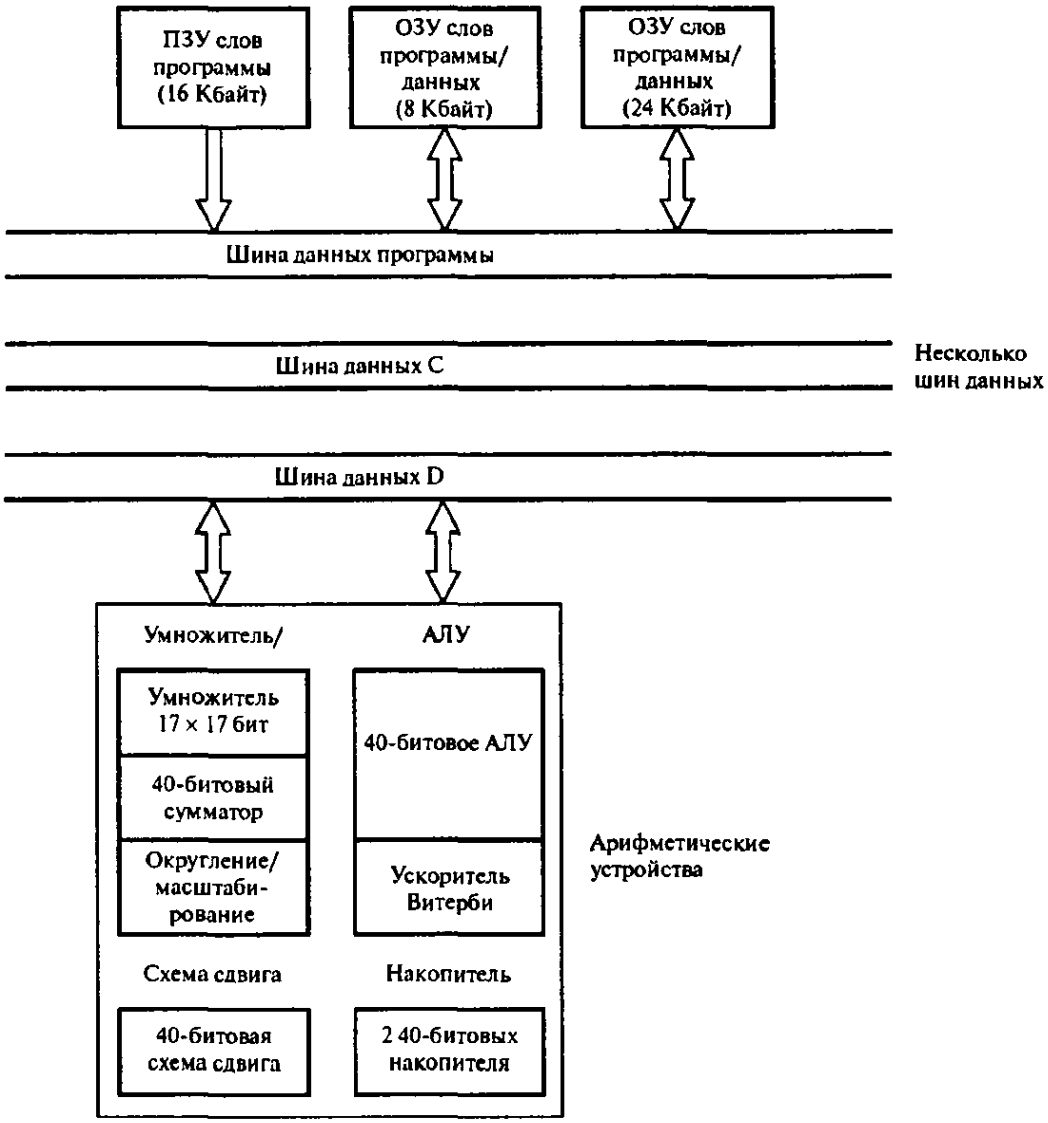


Рис. 4.20. Упрощенная архитектура процессора ЦОС с фиксированной запятой третьего поколения (Texas Instruments TMS320C54x)

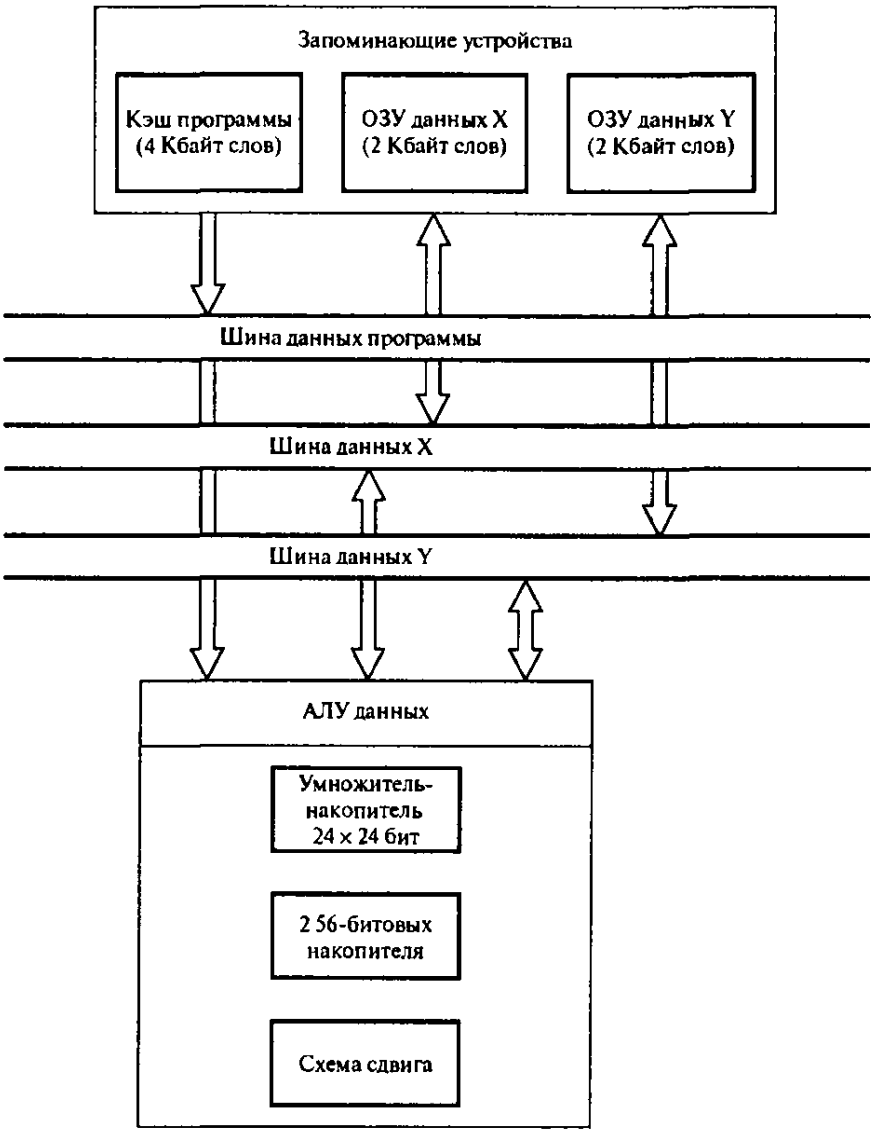


Рис. 4.21. Упрощенная архитектура процессора ЦОС с фиксированной запятой третьего поколения (Motorola DSP56300)

По сравнению с предыдущими поколениями четвертое поколение процессоров с фиксированной запятой отличается более длинными командными словами, более широкими трактами передачи данных, большим числом регистров, большим кэшем команд и наличием нескольких арифметических устройств, что позволяет выполнять значительно больше команд и операций за такт.

Семейство процессоров ЦОС с фиксированной запятой TMS320C62x (Texas Instruments) основано на архитектуре VLIW (рис. 4.23). Процессор ядра имеет два независимых арифметических тракта, в каждом из которых четыре операционных блока – логическое устройство (Li), схема сдви-

га/логическое устройство (Si), умножитель (Mi) и устройство адреса данных (Di). Обычно процессор ядра извлекает из памяти восемь 32-битовых команд за раз, так что длина получающегося слова равна 256 бит (отсюда и название – *с командными словами сверхбольшой длины).*

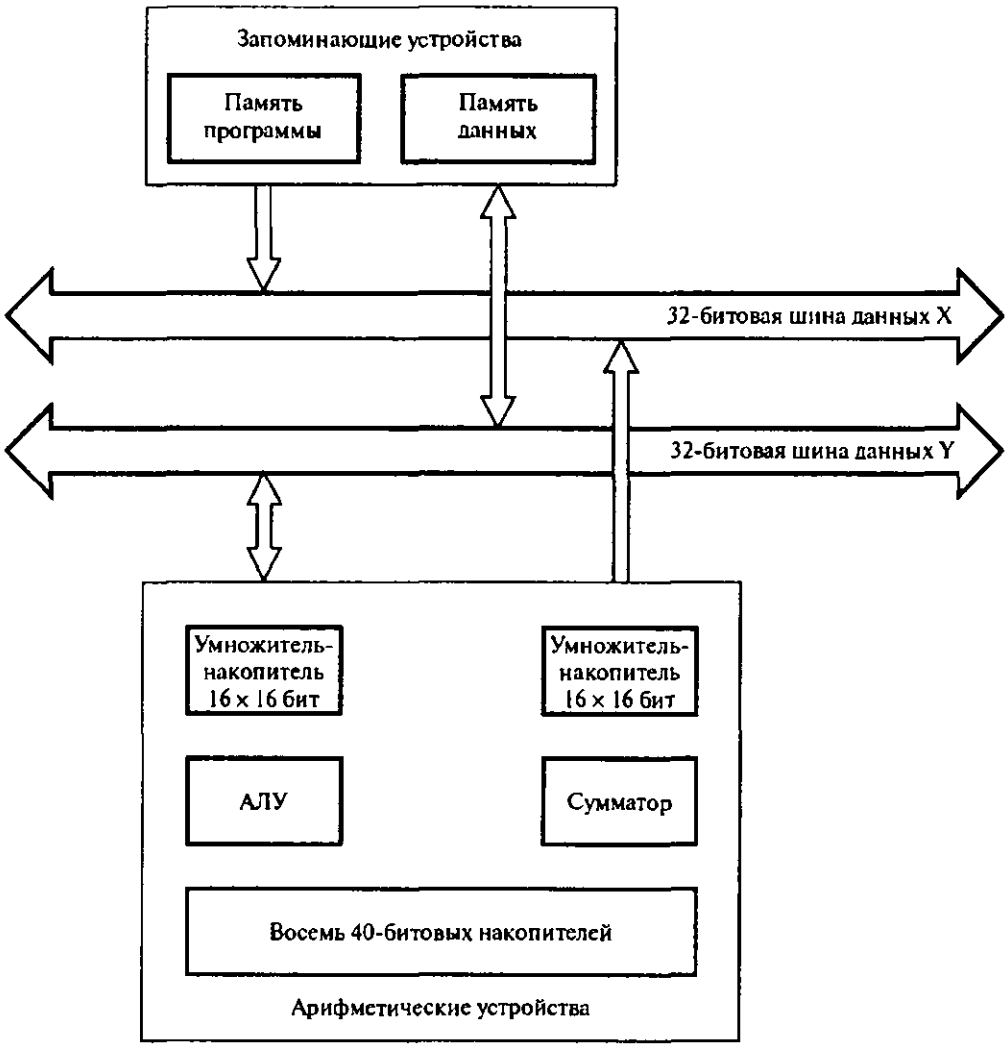


Рис. 4.22. Упрощенная архитектура процессора ЦОС с фиксированной запятой третьего поколения (Lucent Technologies DSP16000)

При наличии в сумме восьми операционных блоков, по четыре в каждом тракте, процессор TMS320C62x может выполнять до восьми команд параллельно в одном такте. Процессор имеет значительную программную память и большой кэш данных (обычно 4 Кбайт программного кэша/кэша данных уровня 1 и 64 Кбайт программного кэша/кэша данных уровня 2). Каждый тракт данных имеет собственный блок регистров (шестнадцать 32-битовых регистров), но может также обращаться к регистрам другого тракта данных. Преимущества архитектуры VLIW включают простоту и

значительную вычислительную эффективность. Недостатки – повышенное использование программной памяти (организация кодов для согласования внутреннего параллелизма процессора может привести к неэффективному использованию памяти). Более того, оптимальной производительности процессоров можно достичь, только когда все операционные блоки заняты, что не всегда возможно вследствие зависимости по данным, задержек команд и ограничений при использовании операционных блоков. Впрочем, существуют сложные программные средства для сжатия кода, планирования команд, распределения ресурсов и использования богатого потенциала процессора вообще.

На рис. 4.23 приведена упрощенная архитектура процессора ЦОС с фиксированной запятой четвертого поколения. Обратите внимание на два независимых арифметических тракта передачи данных, каждый из которых имеет четыре операционных блока – LI, SI, Ml и D1; L2, S2, M2 и D2.

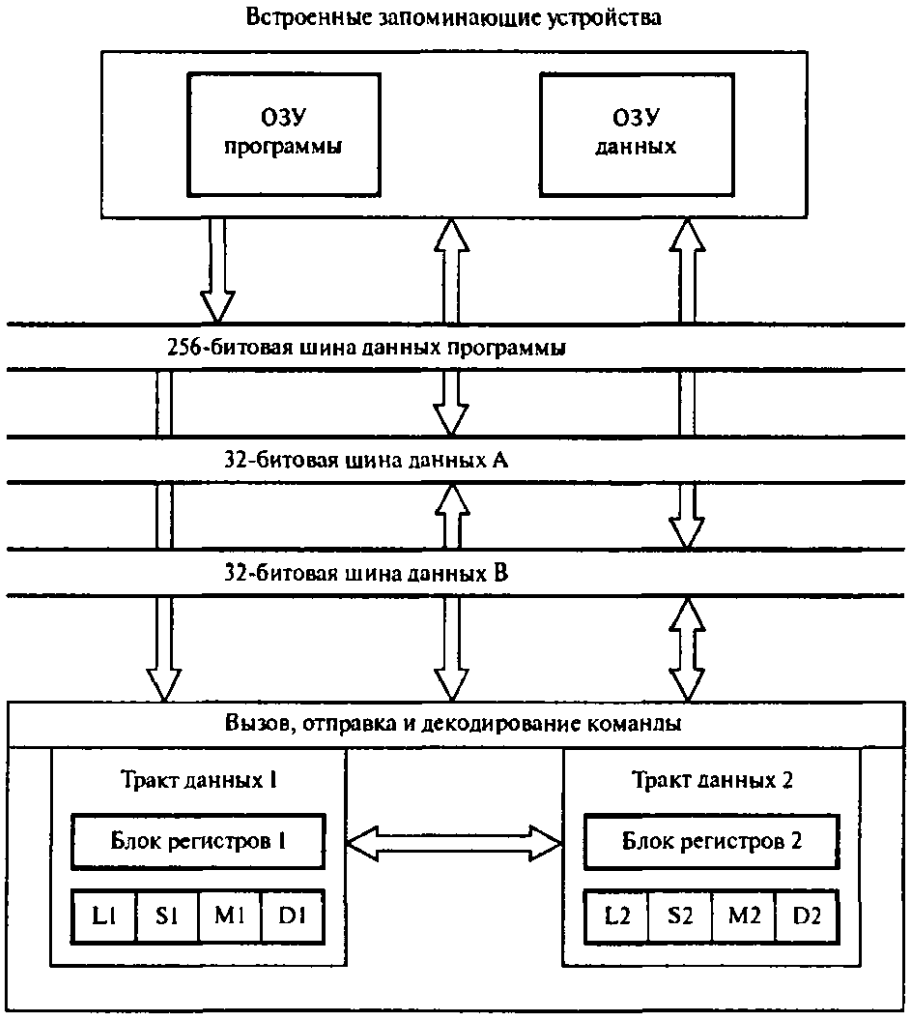


Рис. 4.23. Упрощенная архитектура процессора ЦОС с фиксированной запятой четвертого поколения (Texas Instruments TMS320C62x)

4.3.2. Процессоры ЦОС с плавающей запятой

Возможность разработки процессоров для выполнения операций ЦОС с высокой скоростью, высокой точностью и использованием арифметики с плавающей запятой стала долгожданным открытием. Такие операции минимизируют влияние конечной разрядности, которое проявляется как переполнение, ошибки округления и ошибки квантования коэффициентов. Кроме того, такой подход облегчает разработку алгоритмов, поскольку разработчик может создать алгоритм на большом компьютере на языке высокого уровня, а затем перенести его на устройство ЦОС проще, чем он бы это сделал при арифметике с фиксированной запятой.

Ключевые особенности процессоров ЦОС с плавающей запятой аналогичны свойствам процессоров с фиксированной запятой – существуют специальные команды для операций ЦОС и множественные тракты передачи данных для выполнения нескольких операций. При этом, как и процессоры ЦОС с фиксированной запятой, доступные процессоры ЦОС с плавающей запятой существенно отличаются архитектурно. Некоторые ключевые особенности трех поколений процессоров ЦОС с плавающей запятой производства Texas Instruments и Analog Devices сведены в табл. 4.2.

Пожалуй, самым известным семейством универсальных процессоров ЦОС с плавающей запятой первого поколения является TMS320C3x, которое представлено 32-битовыми одночиповыми цифровыми процессорами сигналов и поддерживает как арифметические операции с целыми числами, так и действия с плавающей запятой. Процессоры имеют большую память и множество встроенных периферийных средств, упрощающих разработку систем. В число этих средств входят программный кэш (позволяет ускорить выполнение часто вызываемых кодов) и встроенная память двойного доступа. Размеры областей памяти удовлетворяют запросам приложений с большими требованиями к памяти, как, например, графические приложения и задачи из сферы обработки изображений. В процессоре TMS320C30 умножение с плавающей запятой требует 32-битовых операндов и выдает 40-битовое нормированное произведение, представленное в форме с плавающей запятой. Целочисленное умножение требует 24-битовых входов и дает 32-битовый результат. Всего поддерживается три формата с плавающей запятой. Первый – короткий 16-битовый формат с 4-битовой экспонентой, одним битом знака и 11 битами мантиссы. Данный формат используется в тех операциях с плавающей запятой, где требуется немедленный результат.

Таблица 4.2

Особенности универсальных процессоров ЦОС с фиксированной запятой производства Texas Instruments, Motorola и Analog Devices

Поко-ление	ЦОС с фиксированной запятой	Ширина тракта данных (биты)	Число трактов данных	Длина слова накопителя (биты)	Длина слова (биты)	Длина команды (биты)	Размер встроенного ОЗУ (слова)	Размер кэша команд (число команд)	Число умножи-телей	Коэффициент производи-тельности*
1	TMS320C10 (Texas Instruments)	16	1	16	32	16	144	–	1	–
2	TMS320C50 (Texas Instruments)	16	2	16	32	16	10 К	–	1	10 @ 50 МГц
	DSP56002 (Motorola)	24	2	24	56	24	1 К	–	1	13 @ 50 МГц
	DSP-2100 (Analog Devices)	16	2	16	40	24	32 К	16	1	13 @ 52 МГц
	1600 (Lucent Technologies)	16	2	16	36	16	–	15	1	22 @ 120 МГц
3	TMS320C54 (Texas instruments)	16	3	16	40	16	32 К	–	1	25 @ 100 МГц
	DSP56300 (Motorola)	24	3	24	56	24	–	3 К	1	25 @ 100 МГц
	16000 (Lucent Technologies)	32	2	32	40	32	127 К	31	2	36 @ 100 МГц
4	TMS320C6200	–	2	–	40	256	17 К	64 К	2	86 @ 133,6 МГц

* Коэффициент производительности рассчитывается на основе скорости расчета контрольных ядер/вычисления контрольных алгоритмов ЦОС.

Второй формат – формат обычной точности с 8- битовой экспонентой, одним битом знака и 23-битовой мантиссой (всего 32 бита). Третий – это 40-битовый формат повышенной точности, который имеет 8-битовую экспоненту, один бит знака и 31-битовую мантиссу. Отметим, что описанное представление в форме с плавающей запятой отличается от стандарта IEEE, но процессор имеет средства конвертации между двумя форматами. Процессор TMS320C3x объединяет особенности гарвардской архитектуры (отдельные шины для команд программы, данных и ввода – вывода) и фоннеймановской архитектуры (унифицированное адресное пространство).

Во втором поколении универсальных процессоров ЦОС с плавающей запятой акцент был сделан на многопроцессорной обработке и поддержке нескольких процессоров. Ключевыми вопросами здесь являются передача данных между процессорами, передача данных с прямым доступом к памяти и совместное использование глобальной памяти.

Наиболее известными семействами процессоров ЦОС с плавающей запятой второго поколения являются TMS320C4x (Texas Instruments) и ADSP-2106x SHARC (Analog Devices). Семейство C4x имеет некоторые архитектурные особенности семейства C3x, но разработано для многопроцессорной обработки. Семейство C40x имеет хорошие возможности ввода-вывода – шесть последовательных портов для передачи данных между процессорами и шесть 32-битовых каналов прямого доступа к памяти для быстрой передачи данных. Данная архитектура позволяет выполнять несколько операций параллельно в одном такте. Семейство C4x поддерживает арифметику с плавающей запятой и фиксированной запятой. Родной формат данных с плавающей запятой в процессоре C40 отличается от стандарта IEEE 754/854, хотя преобразовать формат довольно легко.

Процессоры ЦОС ADSP-2106x SHARC (Analog Devices) также являются 32-битовыми устройствами с плавающей запятой. Они имеют большую внутреннюю память и впечатляющие возможности ввода – вывода – 10 каналов прямого доступа к памяти, позволяющих обращаться к внутренней памяти без вмешательства процессора, и шесть портов связи для высокоскоростной передачи данных между процессорами. Архитектура допускает совместное использование глобальной памяти, так что до шести процессоров SHARC могут обращаться к внутренним ОЗУ друг друга на скорости передачи данных, достигающей максимальной. Семейство ADSP-2106x поддерживает арифметику с фиксированной запятой и плавающей запятой. Формат обычной точности согласуется со стандартом IEEE 754/854 (24-битовая мантисса и 8-битовая экспонента). Архитектура также позволяет выполнять за такт несколько операций. В процессорах

ЦОС *третьего поколения* еще больше была развита концепция параллелизма, увеличено число команд и число операций за такт, чтобы процессоры удовлетворяли требованиям многоканальных и вычислительно интенсивных приложений. Это стало возможным за счет применения новых архитектур, в частности, VLIW (*Very Long Instruction Word* – архитектура с командными словами сверхбольшой длины) и суперскалярной архитектуры. Двумя лидирующими семействами процессоров ЦОС с плавающей запятой третьего поколения являются TMS320C67x (Texas Instruments) и ADSP-TS001 (Analog Devices). Семейство TMS320C67x имеет ту же архитектуру VLIW, что и улучшенные процессоры ЦОС с фиксированной запятой четвертого поколения TMS320C62x (детальная информация представлена в п. 4.3.1 и на рис. 4.23).

Семейство TigerSHARC поддерживает смешанные арифметические типы (арифметику с плавающей и фиксированной запятой) и типы данных (8-, 16- и 32-битовые числа). Такая гибкость позволяет повышать эффективность за счет выбора типа арифметики и формата данных, наиболее подходящих для данного приложения. Как и семейство TMS320C67x, процессоры TigerSHARC нацелены на масштабные многоканальные приложения, такие как мобильные системы третьего поколения (беспроводные системы третьего поколения), цифровые абонентские линии (xDSL) и модемы удаленного множественного доступа для служб Internet. Процессор TigerSHARC с его статической суперскалярной архитектурой (см. рис. 4.14 и п. 4.3.1) объединяет хорошие особенности архитектуры VLIW, обычной архитектуры ЦОС и компьютеров RISC (*reduced instruction set computer* – компьютер с сокращенным набором команд). Процессор имеет два операционных блока, в каждом из которых есть умножитель, АЛУ и 64-битовая схема сдвига. Процессор может выполнять до восьми операций умножения – накопления за такт с 16-битовыми входами и 40-битовым накоплением, две 40-битовые операции умножения – накопления с 16-битовыми комплексными данными или две 80-битовые операции умножения – накопления с 32-битовыми данными. При 8-битовых данных TigerSHARC может выполнять до 16 операций в такте. TigerSHARC имеет значительную пропускную способность памяти, а сама память организована в три 128-битовых блока. При работе с данными могут использоваться разные форматы – обычные 32-битовые слова, длинные 64-битовые слова и учетверенные 128-битовые слова. В одном такте может обрабатываться до четырех 32-битовых команд. Чтобы избежать использования больших служебных издержек (недостаток структуры VLIW), длинные командные слова могут разбиваться на отдельные короткие команды, которые затем передаются независимо каждому блоку.

4.4. Выбор цифрового процессора сигналов

В последние годы вопрос выбора процессора ЦОС для конкретного приложения становится важным, поскольку число доступных процессоров продолжает расти [26]. В число специфических факторов, которые следует рассмотреть при выборе процессора ЦОС для данного приложения, входят архитектурные особенности, скорость выполнения, тип арифметики и длина слова.

1. *Архитектурные особенности*. Большинство доступных сейчас процессоров ЦОС имеют хорошие архитектуры, ресурсов которых, впрочем, может не хватать для определенного приложения. Ключевыми характеристиками процессоров являются размер встроенной памяти, наличие специальных команд и возможности ввода – вывода. Наличие встроенной памяти – необходимое требование в большинстве приложений ЦОС реального времени, поскольку это означает быстрый доступ к данным и быстрое выполнение программы. Для приложений со строгими требованиями к памяти (цифровое аудио, система Dolby, факс-модем, кодирование – декодирование MPEG) размер внутренней памяти ОЗУ может стать решающим фактором при выборе процессора. Если внутренней памяти недостаточно, ее можно дополнить высокоскоростной внешней памятью, хотя это может привести к увеличению стоимости системы. Для приложений, требующих быстрых и эффективных вычислений или обмена потоками данных с внешним миром, весьма важны такие средства ввода – вывода, как интерфейсы к АЦП и ЦАП, возможность прямого доступа к памяти и поддержка многопроцессорной обработки. В зависимости от приложения важен богатый набор специальных команд поддержки операций ЦОС, например, возможность организации циклов с нулевыми служебными издержками, специализированные команды ЦОС и круговая адресация.

2. *Скорость выполнения.* Поскольку большинство задач ЦОС требуют срочного решения, важной мерой производительности является скорость процессоров ЦОС. Традиционно двумя основными единицами измерения этой величины являются тактовая частота процессора в мегагерцах и число выполняемых команд в миллионах команд за секунду (*Million Instructions Per Second* – MIPS) или, если используются процессоры ЦОС с плавающей запятой, в миллионах операций с плавающей запятой в секунду (*million floating-point operations per second* – MFLOPS). Впрочем, подобные меры могут в некоторых случаях не подходить из-за значительных отличий в принципах работы различных процессоров ЦОС, большинство из которых может выполнять несколько операций в одной машинной ко-

манде. Например, в семействе процессоров C62x возможно выполнение восьми команд за такт. Для разных процессоров также отличается число операций, выполняемых в каждом такте. Следовательно, сравнивать скорости работы процессоров на основе названных выше мер бессмысленно. Альтернативная мера основана на скорости выполнения контрольных алгоритмов [26], например, центральных алгоритмов ЦОС, таких как БПФ, КИХ- и БИХ-фильтрация. В табл. 4.1 и 4.2 представлены коэффициенты производительности, рассчитанные на основе подобных контрольных алгоритмов; изучая эти данные, можно получить представление об относительной производительности различных популярных процессоров ЦОС.

3. *Тип арифметики*. Двумя наиболее распространенными типами арифметики, используемыми в современных процессорах ЦОС, являются арифметики с фиксированной и плавающей запятой. Арифметика с плавающей запятой – это естественный выбор в приложениях с широкими и переменными требованиями к динамическому диапазону (динамический диапазон можно определить как разность между наибольшим и наименьшим уровнем сигнала, который можно представить, или как разность между наибольшим сигналом и минимальным уровнем шума, измеренными в децибелах). Процессоры с фиксированной запятой предпочтительны с точки зрения низкой стоимости, они подходят для массового производства (например, сотовые телефоны и компьютерные дисководы). При использовании арифметики с фиксированной запятой возникают вопросы, связанные с ограничениями динамического диапазона, которые обязательно должен рассмотреть разработчик. Вообще процессоры с плавающей запятой дороже процессоров с фиксированной запятой, хотя в последние годы разница в цене существенно уменьшилась. Отметим также, что большинство существующих процессоров ЦОС с плавающей запятой также поддерживают арифметику с фиксированной запятой.

4. *Длина слова*. Еще одним важным параметром в ЦОС является длина слова данных процессора, поскольку она может существенно влиять на качество сигнала. Этот параметр определяет, насколько точно можно представить параметры и результаты операций ЦОС. Вообще, чем длиннее слово данных, тем меньше ошибки при цифровой обработке сигнала. В аудиообработке с фиксированной запятой, например, для поддержания CD-качества длина слова процессора должна быть не меньше 24 бит, что позволит поддержать наименьший уровень сигнала достаточно выше минимального уровня шума, генерируемого обработкой сигнала. В процессорах ЦОС с фиксированной запятой используются разнообразные длины слов процессоров, в зависимости от приложения (см. табл. 4.1).

В процессорах ЦОС с фиксированной запятой, нацеленных на рынок телекоммуникаций, обычно используются слова 16-битовой длины (например, TMS320C54x), тогда как в процессорах, нацеленных на аудиоприложения высокого качества, используются 24 бит (например, DSP56300). В последние годы авторы отмечают тенденцию к использованию большего числа битов для АЦП и ЦАП (например, 24-битовый аудиокодек Cirrus, CS4228), поскольку стоимость подобных устройств постоянно снижается. Таким образом, авторы предвидят повышение спроса на процессоры для аудиообработки с большими длинами слов. В накопителях процессоров с фиксированной запятой также могут требоваться защитные биты (обычно 1-8 бит) для предотвращения арифметического переполнения в процессе операций умножения и накопления повышенной точности. Дополнительные биты эффективно расширяют динамический диапазон, доступный для процессоров ЦОС. В большинстве процессоров ЦОС арифметика обычной точности подразумевает использование 32-битовых данных (24-битовая мантисса и 8-битовая экспонента). Данный размер также совместим с форматом IEEE представления в форме с плавающей запятой (IEEE 754). Большинство процессоров ЦОС с плавающей запятой также позволяют выполнять операции с фиксированной запятой и часто поддерживают арифметику с фиксированной запятой с переменным размером данных.

На практике при выборе процессора в расчет могут приниматься и такие факторы, как опыт работы (знакомство) с конкретным семейством процессоров ЦОС, легкость использования, срок присутствия на рынке и стоимость.

4.5. Реализация алгоритмов ЦОС

4.5.1. Цифровая КИХ-фильтрация

Нерекурсивные *N*-точечные КНХ-фильтры со структурой, представленной на рис. 4.24, *а*, характеризуются следующим разностным уравнением:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k). \qquad (4.6)$$

В программе 4.1 представлен фрагмент реализации на языке С общего КИХ-фильтра. При фильтрации в реальном времени предполагается, что данные и коэффициенты хранятся в памяти, как показано на рис. 4.24, *б*. Чтобы понять, как работает КИХ-фильтр, рассмотрим простой случай $N = 3$, который описывается следующим разностным уравнением:

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2), \qquad (4.7)$$

где $x(n)$ представляет последнюю входную выборку, $x(n-1)$ – предпоследнюю входную выборку, а $x(n-2)$ – предпредпоследнюю входную выборку.

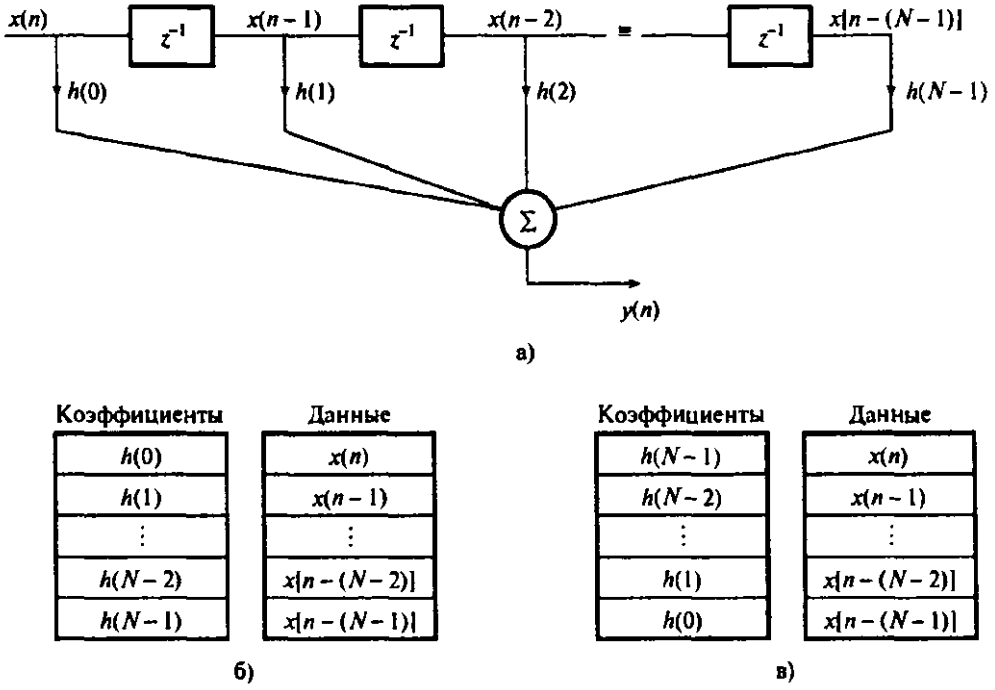


Рис. 4.24. Реализация КИХ-фильтра: *а* – структура фильтра;
б– карты памяти коэффициентов и данных; *в* – альтернативная карта памяти

Программа 4.1. Псевдокод С для выполнения КИХ-фильтрации

```
nml=N-l; yn=0;
for(k=0; k,nml;++k){ /* данные смещаются, чтобы освободить место
                                    для новой выборки */
    x[nml-k]=x[nml-k-l]; x[0]=xп;
}
for(k=0; k,N;++k){
    yn=yn+h[k]*x[к]; /* данные фильтруются и вычисляется
                                    выходная выборка*/
}
return(yn);                          /* выходная выборка фильтра */
```

Предположим, что данные на трехкоэффициентный КИХ-фильтр поступают с АЦП. Первое, что нужно сделать – выделить два набора соседствующих ячеек памяти (в ОЗУ), один для хранения входных данных ($x(n)$, $x(n-1)$, $x(n-2)$), а другой – для хранения коэффициентов фильтра, ($h(0)$, $h(1)$, $h(2)$), как показано ниже:

ОЗУ данных Память коэффициентов

0 $h(0)$

0 $h(1)$

0 $h(2)$

Во время инициализации значения ячеек ОЗУ, в которых планируется хранить выборки данных, обнуляются, поскольку фильтр всегда начинает работу, не имея данных. Затем выполняются следующие операции:

1. *Первый момент выборки.* Считывается выборка данных с АЦП, ОЗУ данных смещается на одну позицию (чтобы освободить место для новых данных), сохраняется новая входная выборка, вычисляется выходная выборка согласно формуле (4.5), а затем выходная выборка посылается ЦАП:

ОЗУ данных Память коэффициентов

$\rightarrow x(1)$ $h(0)$

$y(1) = h(0)x(1) + h(1)x(0) + h(2)x(-1)$

0 $h(1)$

0 $h(2)$

2. *Второй момент выборки*. Повторяются описанные выше действия, и новая выходная выборка посылается ЦАП:

ОЗУ данных Память коэффициентов

$\rightarrow x(2)$ $h(0)$

$y(2) = h(0)x(2) + h(1)x(1) + h(2)x(0)$

0 $h(1)$

0 $h(2)$

3. *Третий момент выборки.* Повторяются описанные выше действия, и новая выходная выборка посылается ЦАП:

ОЗУ данных Память коэффициентов

$\rightarrow x(3)$ $h(0)$

$y(3) = h(0)x(3) + h(1)x(2) + h(2)x(1)$

0 $h(1)$

0 $h(2)$

4. *Четвертый момент выборки.* Повторяются описанные выше действия, и новая выходная выборка посылается ЦАП:

ОЗУ данных Память коэффициентов

$\rightarrow x(4)$ $h(0)$

$y(4) = h(0)x(4) + h(1)x(3) + h(2)x(2)$

0 $h(1)$

0 $h(2)$

Обратите внимание на то, что старейшая входная выборка удалена из памяти.

5. *Момент выборки n*. Повторяются описанные выше действия, и новая выходная выборка посылается ЦАП:

ОЗУ данных	Память коэффициентов
$\rightarrow x(n)$	$h(0)$
$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2)$	
0	$h(1)$
0	$h(2)$

Реализация трехточечного КИХ-фильтра на процессоре ЦОС с фиксированной точкой первого поколения (TMS320C10) представлена в программе 4.2. В данном случае расчет произведений начинается с верхнего адреса памяти данных и коэффициентов, что позволяет использовать команду TMS320C10 для передвижения данных. Для реализаций КИХ-фильтров на базе TMS320C10 основными являются пара команд LTD и MPY. Например, приведенная ниже пара команд выполняет сдвиг, подразумеваемый в уравнении (4.4) или обозначенный на рис. 4.24, *a* через z^{-1}, добавляет предыдущее произведение в накопитель и рассчитывает следующее произведение $h(k)x(n - \kappa)$.

Программа 4.2. Линейный код трехточечного КИХ-фильтра

```
NXTPT IN XN, ADC
ZAC
LT      XNM2
MPY     H2      ; h(2)x(n – 2)
LTD     XNM1    ; 0 + h(2)x(n – 2); x(n – 2) = x(n – l)
MPY     H1      ; h(l)x(n – l); x(n – l) = x(n – 2)
LTD     XN      ; h(2)x(n – 2) + h(l)x(n – l); x(n – l) = x(n)
MPY     H0      ; h(0)x(n)
APAC            ; h(2)x(n – 2) + h(l)x(n – l) + h(0)x(n)
SACH    YN, 1   ; сохранить выходную выборку
OUT     YN, DAC ; передать выходную выборку ЦАП
B       NXTPT

LTD     XNM1
MPY     H1
```

В частности, команда LTD XNM1 загружает в регистр T (временный) выборку данных $x(n - 1)$ (хранилась в ОЗУ данных по адресу XNM1),

добавляет предыдущее произведение $h(2)x(n-2)$, которое все еще находится в регистре P (произведение), к накопителю и смещает $x(n-1)$ в ячейку со следующим адресом, т.е. $x(n-2) = x(n-1)$. Вторая команда MPY множит содержимое регистра T на $h(1)$ и оставляет результат в регистре произведения. Схема сдвигов гарантирует, что, когда потребуется вычислять следующую выборку, входные выборки данных будут находиться в правильных ячейках.

Линейный код КИХ-фильтра, подобный программе 4.2, – это быстрая реализация, которая, впрочем, не является универсальной и для больших N-точечных фильтров не позволяет получить компактную программу. В частности, для реализации универсального КИХ-фильтра требуется организовать внутренний цикл, реализующий уравнение КИХ-фильтра и вычисляющий выход фильтра, как задано в уравнении (4.4).

Функциональная схема ДГ-точечной КИХ-фильтрации, на которой показан внутренний цикл, представлена на рис. 4.25. В процессоре ЦОС первого поколения внутренний цикл КИХ-фильтра можно реализовать с помощью следующих команд:

```
LOOP LTD  *,ARO      ;сдвиг/обновление линии задержки и
                     ;накопление произведений
     MPY  *-,ARl     ;умножение следующего коэффициента и элемента
                     данных
     BANZ LOOP
```

В рассматриваемом случае вспомогательные регистры AR0 и AR1 применяются для указания на элемент данных и коэффициент, которые нужно перемножить. Вспомогательный регистр AR1 содержит длину фильтра и используется как счетчик цикла. Для управления циклом используется команда ветвления BANZ и проверка содержимого регистра AR1. Отметим, что реализация КИХ-фильтра на процессоре ЦОС первого поколения неэффективна из-за служебных издержек, связанных с управлением циклом.

Процессоры ЦОС с фиксированной запятой второго поколения, такие как TMS320C50 и DSP56000 (Motorola), позволяют создавать циклы с нулевыми служебными издержками и содержат специальные команды умножения и накопления, которые помогают сократить время выполнения внутреннего цикла КИХ-фильтрации. В процессоре TMS320C50, например, внутренний цикл N-точечного КИХ-фильтра, показанный на рис. 4.25, можно эффективно выполнить с помощью следующих команд:

```
RPT     NM1
MACD    HNM1, XNM1
```

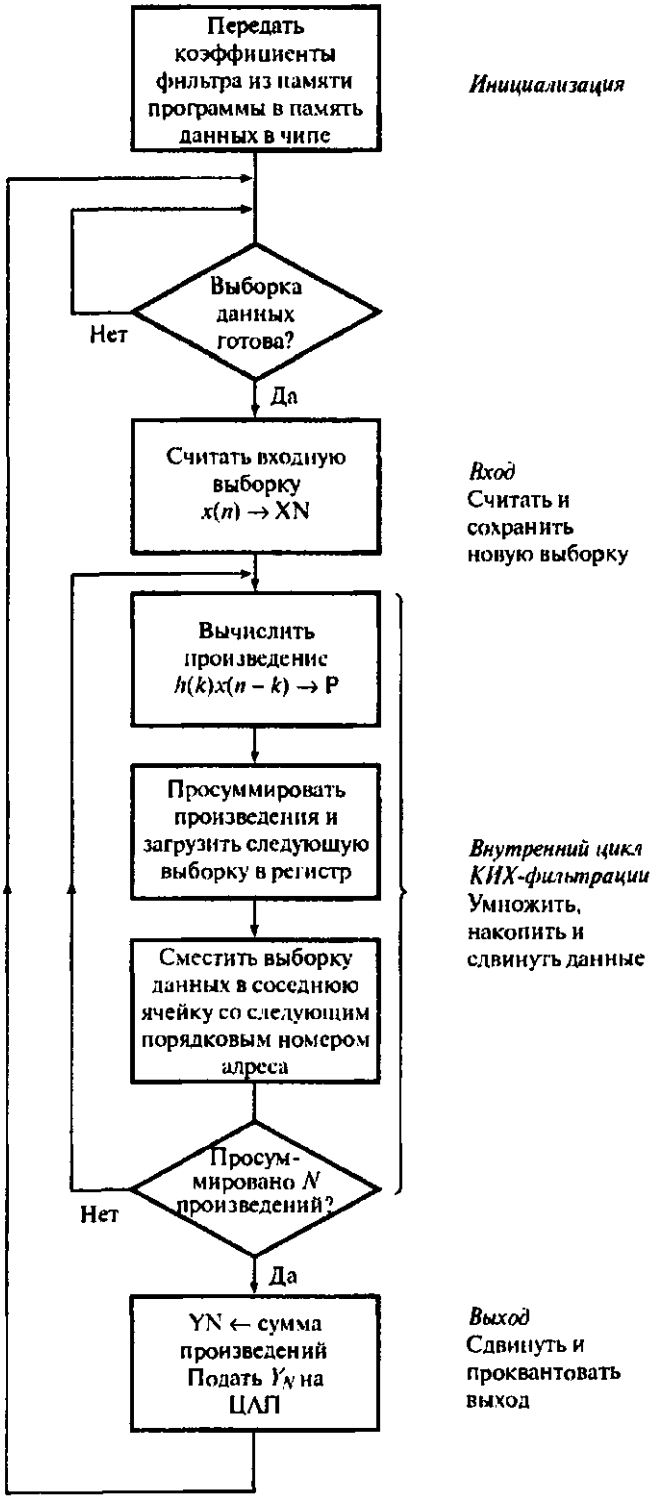


Рис. 4.25. Функциональная схема КИХ-фильтра

Внутренний цикл КИХ-фильтра вычисляет сверточную сумму.

Команда RPT NM1 загружает длину фильтра минус 1 (т.е. (N – 1)) в регистр повтора и инициирует (N – 1)-кратное выполнение с нулевыми служебными издержками команды умножения и накопления со сдвигом

данных MACD. Команда MACD объединяет пару команд LTD MPY в одну, что позволяет быстрее их выполнить. Вообще, пара команд RPT и MACD – это хороший пример экономных специальных команд, доступных в процессорах ЦОС.

Альтернативным подходом к реализации N-точечных КИХ-фильтров на процессорах ЦОС второго и последующих поколений является использование круговых буферов. Очевидно, что в КИХ-фильтрации содержимое памяти коэффициентов статично, а вот память данных меняется с поступлением каждой новой входной выборки. С точки зрения эффективности выгоднее подавать последующие выборки данных в раздвижное окно, удаляя старейшие выборки из памяти. Для обработки изменений в блоке входных данных, используемых при КИХ-фильтрации, можно применять круговой буфер, при этом данные не придется сдвигать, как в линейных буферах.

Концептуально круговой буфер идентичен линейному, если рассматривать два конца линейного буфера как соседние ячейки памяти, т.е. самая новая и самая старая выборки данных $x(n)$ и $x[n – (N – 1)]$ находятся в соседних ячейках памяти (рис. 4.26, *а*). В круговом буфере на рис. 4.26 указатель данных (обозначен стрелочкой) ссылается на ячейку памяти с новейшей входной выборкой $x(n)$, а предыдущие входные выборки данных $x(n – 1)$, $x(n – 2)$, … $x(n – 7)$ хранятся в последующих ячейках памяти по часовой стрелке. В каждый момент получения выборки, как и ранее, выполняется внутренний цикл КИХ-фильтра – каждая выборка данных множится на соответствующий коэффициент фильтра $x(n)$, и произведения накапливаются. Единственное отличие состоит в том, что выборки данных не смещаются. После вычисления внутреннего цикла указатель находится возле ячейки $x(n – 7)$ со старейшей выборкой данных, на место которой затем записывается следующая входная выборка $x(n)$. Работа кругового буфера для трех последовательных выборок данных иллюстрируется на рис. 4.26, *а – в*.

На практике круговая адресация реализуется за счет использования для автоматического циклического возврата, когда указатели адреса выходят за границы буфера модульной арифметики. Обычно в таких случаях требуется задать начальный адрес кругового буфера и размер буфера (или размер модуля). Ниже приводится реализация внутреннего цикла N-точечного КИХ-фильтра с использованием круговой адресации на базе процессора DSP56000.

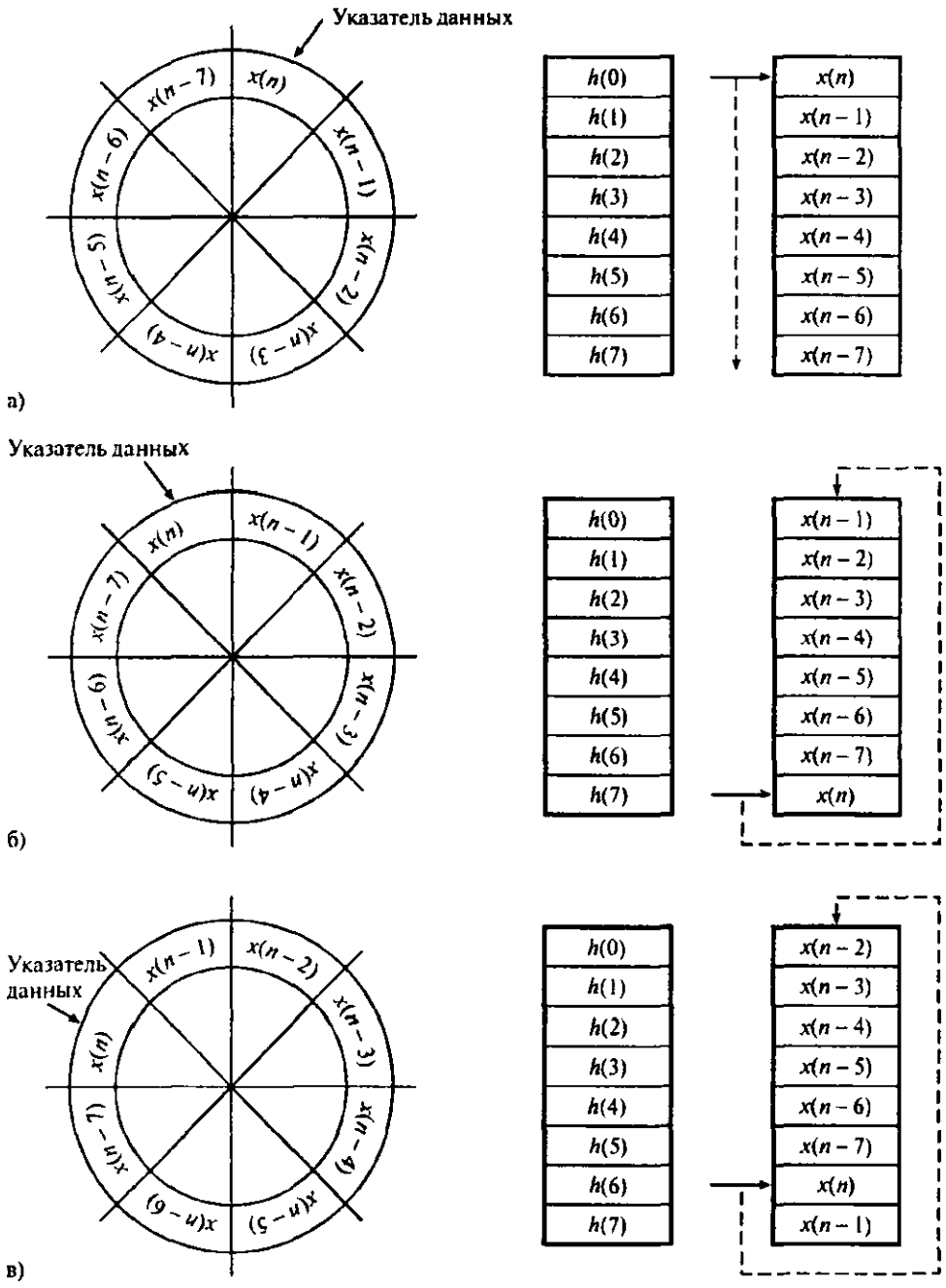


Рис. 4.26. Иллюстрация принципов реализации КИХ-фильтра на основе кругового буфера

```
MOVE    #XDATA, RO
MOVE    #COEFF, R4
MOVE    #N-1, MO            ;размер буфера/модуля
MOVEP   X: INPUT, X: (RO)   ;считывание и запись входной выборки
CLR     A                   ;очистка накопителя
REP     #N-1 /выполнение внутреннего цикла КИХ-фильтра
MAC     XO, Y0, A X:(R0)+, XO Y:(R4)+, Y0
MACR    XO, XO, A (R0) –
```

В представленном случае круговые буферы используются для хранения данных и коэффициентов. Круговой буфер данных выполняет неявный временной сдвиг, как описано выше. В то же время для удобства автоматического циклического возврата указателя коэффициентов используется круговой буфер коэффициентов. Первые четыре команды в приведенной выше программе устанавливают указатели адреса R0 и R4. Пара команд REP и MAC выполняют внутренний цикл КИХ-фильтра. Команда повтора REP повторяет следующую за ней команду N – 1 раз. В следующей строке для выполнения набора операций умножения используется архитектура и параллелизм DSP56000 – здесь умножаются элементы данных и коэффициенты, расположенные в ячейках X0 и Y0, произведение добавляется к накопителю, из ячеек X и Y извлекается следующая пара «элемент данных – коэффициент» и обновляются указатели.

Кроме области КИХ-фильтрации, круговая адресация применяется при эффективной реализации нескольких других функций ЦОС, требующих временных сдвигов или простых очередей, например, корреляции, фильтров с обработкой при нескольких скоростях (фильтров децимации и интерполяции) и генерации периодических сигналов. Использование круговой адресации делает ненужным смещение данных или постоянную проверку (обновление) указателей адреса. Все процессоры ЦОС старших поколений имеют улучшенные возможности круговой адресации.

Пример 4.3

Цифровой фильтр, удовлетворяющий приведенным ниже спецификациям, нужно реализовать на базе процессора ЦОС с фиксированной запятой второго поколения TMS320C50.

Частота режекции	1,875 кГц.
Подавление на частоте режекции	60 дБ.
Граничные частоты полосы пропускания	1,575 и 2,175 кГц.
Неравномерность в полосе пропускания	0,01 дБ.
Частота дискретизации	7,5 кГц.

Решение

1. Приведенным спецификациям удовлетворяет 61-точечный оптимальный КИХ-фильтр. Разработка этого фильтра подробно рассмотрена в [1, п. 7.6.5]. Здесь мы остановимся только на его реализации. Коэффициенты фильтра квантуются до 16 бит (формат Q15) путем умножения каждого коэффициента на 2^{15} с последующим округлением ре-

зультата до ближайшего целого. Квантованные и неквантованные коэффициенты фильтра перечислены в табл. 4.3.

Таблица 4.3

Коэффициенты фильтра из примера 4.3

$Я(1) = 0{,}12743640E-02 = H(61)$	42
$Я(2) = 0{,}26730640E-05 = H(60)$	0
$Я(3) = -0{,}23681110E-02 = H(59)$	–78
$Я(4) = -0{,}17416350E-05 = H(58)$	0
$Я(5) = 0{,}43428480E-02 = H(57)$	142
$Я(6) = 0{,}53579250E-05 = H(56)$	0
$Я(7) = -0{,}71570240E-02 = H(55)$	–235
$Я(8) = -0{,}49028620E-05 = H(54)$	0
$Я(9) = 0{,}10897540E-01 = H(53)$	357
$Я(10) = 0{,}89629280E-05 = H(52)$	0
$Я(11) = -0{,}15605960E-01 = H(51)$	–511
$Я(12) = -0{,}85508990E-05 = H(50)$	0
$Я(13) = 0{,}21226410E-01 = H(49)$	695
$Я(14) = 0{,}12250150E-04 = H(48)$	0
$Я(15) = 0{,}27630130E-01 = H(47)$	–905
$Я(16) = -0{,}11091200E-04 = H(46)$	0
$Я(17) = 0{,}34579770E-01 = H(45)$	1133
$Я(18) = 0{,}13800660E-04 = H(44)$	0
$Я(19) = -0{,}41774130E-01 = H(43)$	–1369
$Я(20) = -0{,}11560390E-04 = H(42)$	0
$Я(21) = 0{,}48832790E-01 = H(41)$	1600
$Я(22) = 0{,}12787590E-04 = H(40)$	0
$Я(23) = -0{,}55359840E-01 = H(39)$	–1814
$Я(24) = -0{,}90065860E-05 = H(38)$	0
$Я(25) = 0{,}60944450E-01 = H(37)$	1997
$Я(26) = 0{,}88997300E-05 = H(36)$	0
$Я(27) = 0{,}65232190E-01 = H(35)$	–2137
$Я(28) = 0{,}38167120E-05 = H(34)$	0
$Я(29) = 0{,}67925720E-01 = H(33)$	2226
$Я(30) = 0{,}27041150E-05 = H(32)$	0
$Я(31) = 0{,}93115220E+00 = H(31)$	30512

Как показано на функциональной схеме на рис. 4.25, полный КИХ-фильтр имеет по крайней мере четыре необходимых элемента.

2. *Система инициализации*, которая может включать настройку таблицы коэффициентов.

3. *Вводная часть программы*, которая может включать считывание входной выборки $x(n)$, например, с АЦП через последовательный порт.

4. *Вычисление внутреннего цикла.* Реализация уравнения КИХ-фильтрации для получения $y(n)$.

Выходная часть программы, включающая сдвиг (округление) результата, полученного при расчете внутреннего цикла, и передачу его, например, ЦАП через последовательный порт. Поскольку этапы 1, 2 и 4 зависят от системы, сконцентрируемся на вычислении внутреннего цикла. Внутренний цикл КИХ-фильтра можно реализовать с помощью следующих команд TMS320C50:

```
SACL    XN          ;записать новейшую выборку x(n) в память данных
LAR     AR1, #XNM1  ;указать на положение старейшей выборки данных
                     x[n – (N – 1)]
ZAP                 ;очистить накопитель и регистр произведения
MAR     *, AR1      ;сделать AR1 текущим вспомогательным регистром
RPT     #60         ;выполнить внутренний цикл КИХ-фильтрации
MACD    #COEFF, *-  ;умножить и накопить со сдвигом данных
APAC                ;прибавить последнее произведение
```

В данном случае памяти данных и коэффициентов организованы так, как показано на рис. 4.24, *в*. Вспомогательный регистр AR1 используется для организации непрямой адресации при вычислении внутреннего цикла (команда MACD) и изначально указывает на старейшую выборку данных, XNM1, в памяти для хранения данных.

Во внутреннем цикле команда MACD делает следующее:

- добавляет предыдущее произведение к накопителю – изначально произведение равно нулю;
- умножает коэффициент $h(k)$ на данные, на которые ссылается указатель AR1 – изначально $h(k) = h(N – 1)$, и устанавливает во вспомогательном регистре ссылку на $x[n – (N – 1)]$;
- копирует данные, на которые указывает AR1, в ячейку памяти со следующим адресом – изначально $x[n – (N – 1)]$ копируется в $x(n – N)$; т.е. старейшая выборка данных отбрасывается. Последняя команда MACD копирует $x(n)$ в $x(n – 1)$, чтобы освободить место для следующей входной выборки;
- уменьшает AR1 на 1 (т.е. указатель теперь ссылается на следующую выборку в памяти для хранения данных) – изначально AR1 указывает

на $x[n-(N-1)]$, а затем последовательно указывает на $x[n-(N-2)]$, $x[n-(N-3)]$,. . $x(n)$ по мере кругового прохождения цикла;

- увеличивает адрес COEFF на 1; последовательно получаются следующие адреса: $h(N-1)$, $h(N-2)$,…, $h(0)$.

4.5.2. Цифровая БИХ-фильтрация

4.5.2.1. Стандартные компоновочные блоки БИХ-фильтров. В качестве стандартных компоновочных блоков при создании цифровых БИХ-фильтров используются звенья БИХ-фильтров второго порядка. Двумя наиболее используемыми структурами второго порядка являются каноническое звено (рис. 4.27) и реализация в прямой форме (рис. 4.28). Каноническое звено второго порядка характеризуется следующим уравнением:

$$\omega(n) = \mathrm{SF}_1 x(n) - a_1\omega(n-1) - a_2\omega(n-2), \qquad (4.8)$$

где $x(n)$ представляет входные данные, а $\omega(n)$ – внутренний узел; $y(n)$ – выходная выборка фильтра, SF_1 – масштабный множитель, равный $1/s_1$.

Разностное уравнение звена БИХ-фильтра, реализованного в прямой форме (см. рис. 4.28, *б*), записывается как

$$y(n) = b_0x(n) + \mathrm{bix}(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2), \qquad (4.9)$$

где $x\{n-к)$ – последовательность входных, а $y(n-к)$ – выходных данных. Память для хранения данных и коэффициентов для реализации в прямой форме изображена на рис. 4.28, *б*.

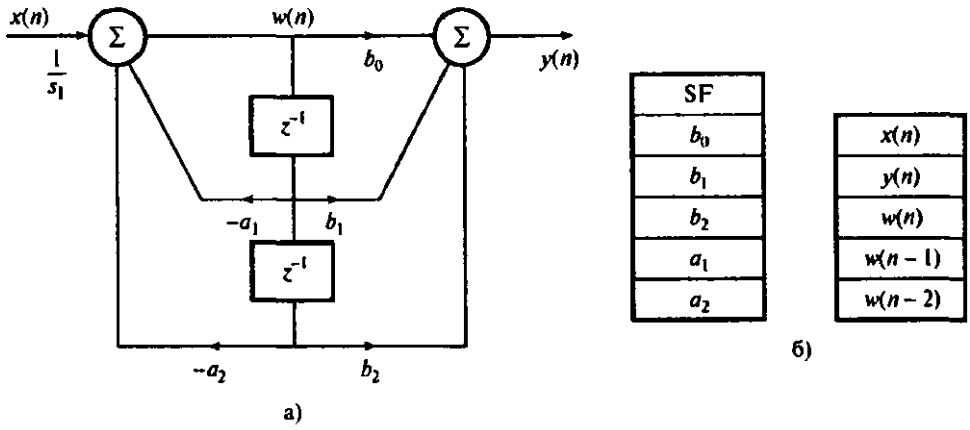


Рис. 4.27. Каноническое звено второго порядка: *а* – диаграмма реализации; *б* – память для хранения данных и коэффициентов

Фильтр в прямой форме проще программировать, и такая реализация может быть несколько быстрее, чем реализация в канонической форме, поскольку в ней используется более простая система индексации – сравните, например, уравнения (4.8) и (4.9).

БИХ-фильтры больших порядков реализуются как каскадная либо параллельная комбинация звеньев фильтров второго порядка.

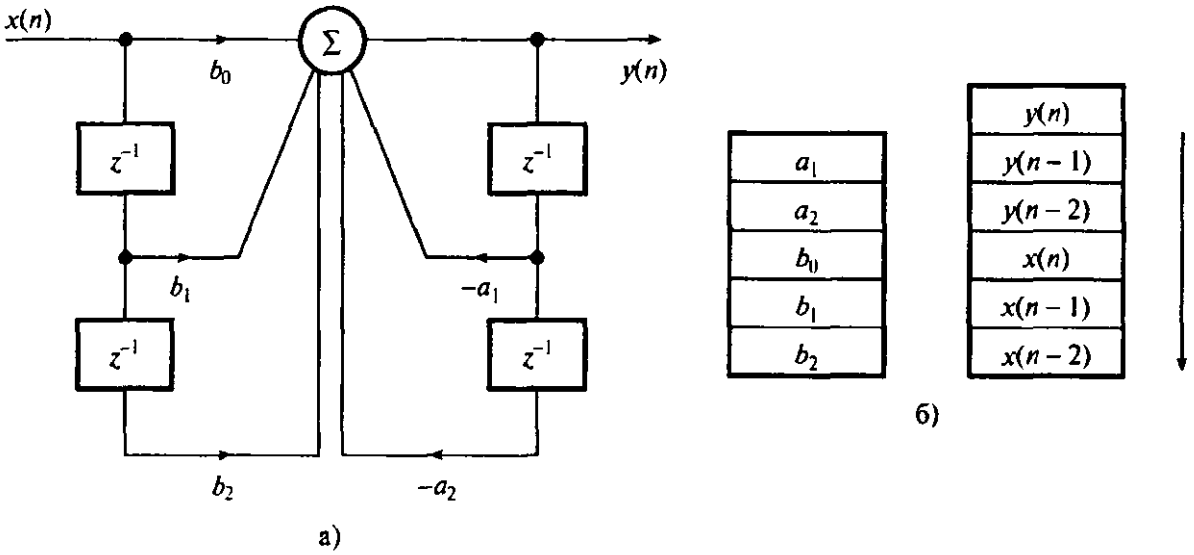


Рис. 4.28. Реализация звена второго порядка в прямой форме: *а* – диаграмма реализации; *б* – память для хранения данных и коэффициентов

Каскадная реализация

Передаточная функция $H(z)$ БИХ-фильтра N-го порядка, полученного как каскад звеньев второго порядка, записывается следующим образом:

$$H(z)=\prod_{k=1}^{N/2}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{1-a_{1k}z^{-1}-a_{2k}z^{-2}}. \qquad (4.10)$$

Каскадная реализация БИХ-фильтра четвертого порядка (N = 4) с использованием канонических звеньев второго порядка показана на рис. 4.29, *а*. Память для хранения переменных фильтра (данных и коэффициентов) показана на рис. 4.29, *б*. набор разностных уравнений БИХ-фильтра четвертого порядка, полученного с помощью канонических звеньев, записывается следующим образом:

$$\omega_1(n)=\mathrm{SF}_1x(n)-a_{11}\omega_1(n-1)-a_{21}\omega_1(n-2); \qquad (4.11)$$

$$y_1(n)=b_{01}\omega_1(n-1)-b_{21}\omega_1(n-2); \qquad (4.12)$$

$$\omega_2(n)=y_1(n)-a_{12}\omega_1(n-1)-a_{22}\omega_2(n-2); \qquad (4.13)$$

$$y_2(n)=b_{02}\omega_2(n)-b_{12}\omega_2(n-1)-b_{22}\omega_2(n-2). \qquad (4.14)$$

Псевдокод С для реализации БИХ-фильтра в форме каскада канонических звеньев второго порядка представлен в программе 4.3.

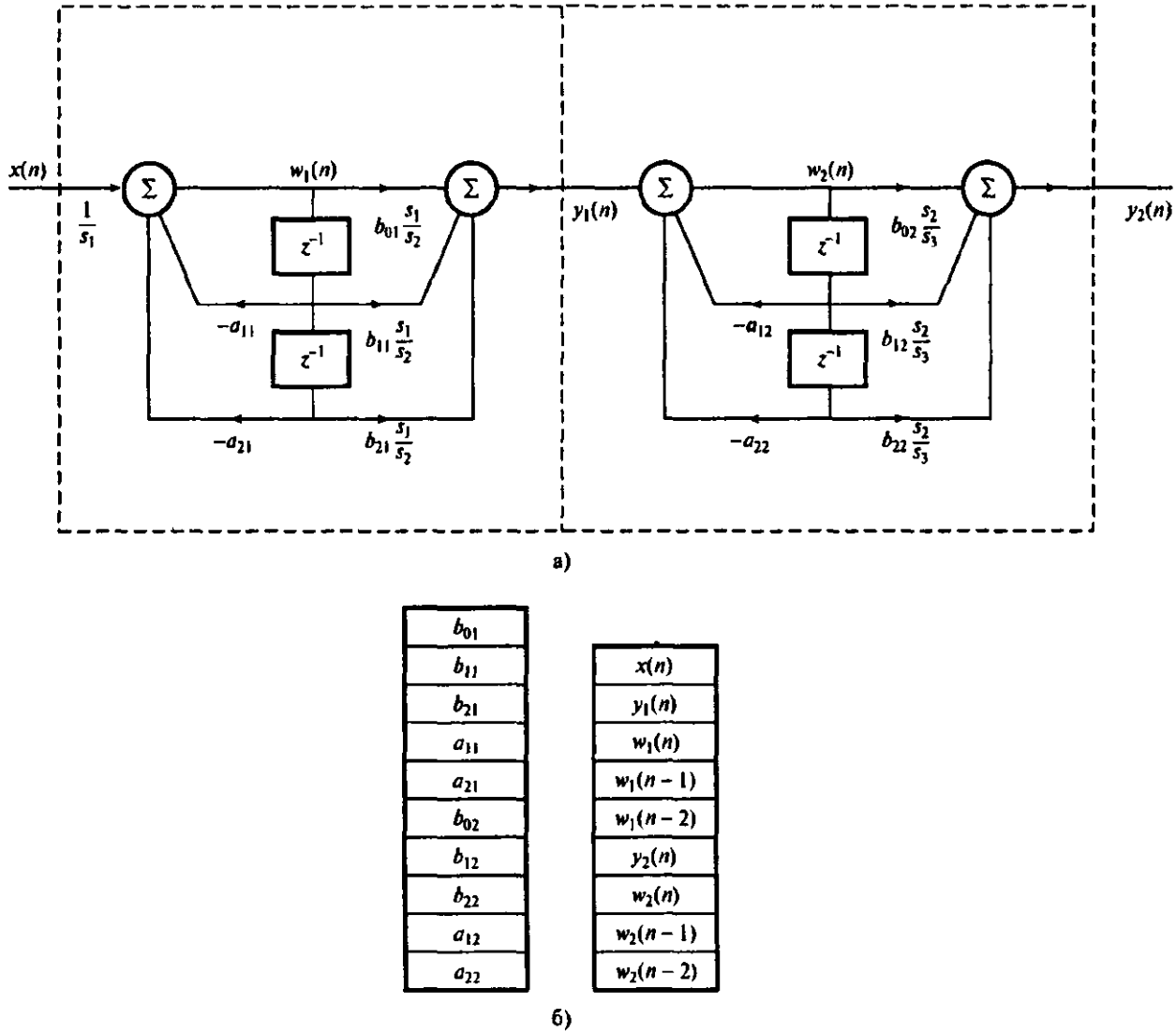


Рис. 4.29. Каскадная реализация БИХ-фильтра: *а* – диаграмма реализации; *б* – память для хранения данных и коэффициентов

В программах 4.4 и 4.5 соответственно представлены фрагменты реализаций БИХ-фильтра *N*-го порядка, который состоит из каскадного соединения *M* биквадратных звеньев (где *M* = *N*/2), на базе процессоров TMS320C50 и DSP56000.

Программа 4.3. Псевдокод С для реализации каскадного БИХ-фильтра

```
for(n=0; n,(Nsamples-1); ++ n){ /* Nsamples – число выборок данных */
    xn = x[n];
    for(k=l; k,N; ++k) {
        wk=sk[k]*xn-al[k]*wl[k]-a2[k]*w2[k];
        yk=(b0[k]*wk+bl[k]*wl[k]+b2[k]*w2[k]); /* выход первого звена */
```

```
        w2[k] = wl[k];              /* сдвиг и запись данных элемента задержки */
        wl[k] = wk;
        xn = yk;/* выход k-го звена подается на вход следующего звена */
    }
    y[n] = yk;                      /* n-я выходная выборка */
    }
```

Программа 4.4. Реализация M каскадно соединенных канонических звеньев фильтров второго порядка на базе процессора TMS320C50

```
 SPL        #M-l, BRCR      ; число биквадратных звеньев

 RPTB       M_IIR           ;расчет M биквадратных звеньев, соединенных
                            каскадом
 LT         *-, AR2         ; загрузка wk(n-2)
 MPYA       + , AR1         ; расчет wk(n-2)*ak(2)
 MPY        +               ; wk(n-l)* ak{l)
 LTA        *+, AR1         ; расчет и запись в память wk(n)=x(n)+
                            ; wk(n-l)*ak(l)+wk{n-2)*ak(2)
 SACH       *0+, 1
 MPY                        ; расчет wk(n-2) * bk(2)
 LACL       #0              ;
 LTD        AR2             ; сдвиг данных wk(n-2) = wk(n-l)
 MPY        *+, AR1         ; yk=yk+wk<n-2) * bk(2), wk(n-l)*bk(l)
 LTD        *-, AR2         ; сдвиг данных wk(n-l) = wk(n)
 MPY        *+, ARl         ; расчет wk(n-2) * bk(2) +
                            ; wk(n-l) * bk(l), wk(n) * bk(0)
M_IIR :
 LTA        *, AR4          ; прибавить последнее произведение
 SACH       *, 1            ; квантование и запись выходной выборки
```

Программа 4.5. Реализация M каскадно соединенных канонических звеньев фильтров второго порядка на базе процессора DSP56000

```
 DO         #M, M_IIR       ; расчет M биквадратных звеньев
 MAC        -X0, Y0, A      X: (R0)-, XI Y: (R4)+, Y0 ;
 MACR       -XI, Y0, A      XI, X: (RO)+ Y: (R4)+, Y0 ; сдвиг данных
                            ; w(n-2)= w(n-l)
 MAC        X0, Y0, A       A, X: (RO)+ Y: (R4)+, Y0
 MAC        XI, Y0, A       X: (RO) +, X0 Y: (R4)+, Y0
M_IIR
 RND
 MOVEP      A, Y: OUTPUT
```

Пример 4.4

1. Разработайте и реализуйте БИХ-фильтр нижних частот, используя процессор ЦОС с фиксированной запятой TMS320C50. Фильтр должен удовлетворять следующим спецификациям:

частота дискретизации	15 кГц,
полоса пропускания	0 – 3 кГц,
ширина полосы перехода	450 Гц,
неравномерность в полосе пропускания	0,5 дБ,
затухание в полосе подавления	45 дБ.

2. Повторите п. 1, используя процессор ЦОС с фиксированной запятой TMS320C54.

3. Повторите п. 1, используя процессор ЦОС с фиксированной запятой DSP56000.

4. Повторите п. 1, используя процессор ЦОС с фиксированной запятой DSP56300.

Решение

Подробно разработка фильтра рассмотрена в [1, п. 8.8]. Показано, что спецификации удовлетворяет эллиптический фильтр четвертого порядка со следующей передаточной функцией:

$$H(z)=\frac{1+0,675718z^{-1}+z^{-2}}{1-0,495935z^{-1}+0,761864z^{-2}}\cdot\frac{1+1,649656z^{-1}+z^{-2}}{1-0,829328z^{-1}+0,307046z^{-2}}.$$

Разностные уравнения каскадной реализации с использованием канонических звеньев совпадают с приведенными в формуле (4.9).

Коэффициенты, масштабированные, чтобы избежать переполнения, и квантованные до 16 бит, перечислены в табл. 4.4. Коды реализации фильтра четвертого порядка с использованием фильтров в канонической форме на базе процессоров TMS320C50 и TMS320C54 не приводятся из-за недостатка места, но их можно найти в [1]. Для реализаций на базе DSP56000 и DSP56300 коэффициенты квантуются до 24 бит (длина коэффициентов).

Таблица 4.4

Коэффициенты фильтра до и после квантования до 16 бит

	Коэффициент	Масштабированный коэффициент	Квантованный коэффициент
b_{02}	**1**	**0,9999695**	**32767**
b_{12}	**0,675718**	**0,675718**	**22142**
b_{22}	**1**	**0,9999695**	**32767**

Окончание табл. 4.4

	Коэффициент	Масштабированный коэффициент	Квантованный коэффициент
a_{12}	**-0,495935**	**-0,495935**	**–16251**
a_{22}	**0,761864**	**0,761864**	**24965**
b_{01}	**1**	**0,1311136**	**4296**
b_{11}	**1,649656**	**0,2162924**	**7087**
b_{21}	**1**	**0,1311136**	**4296**
a_{11}	**-0,829328**	**-0,829328**	**–27175**
a_{21}	**0,307046**	**0,307046**	**10061**
s_1 = 2,479 158 (L_1); SF = 0,403 362 7; s_2 = 18,908 47 (L_1).			

Параллельная реализация

Передаточная функция БИХ-фильтра N-го порядка в параллельной реализации записывается следующим образом:

$$H(z)=\prod_{k=1}^{N/2}\frac{b_{0k}+b_{1k}z^{-1}}{1+a_{1k}z^{-1}+a_{2k}z^{-2}}+C\,. \qquad (4.15)$$

Диаграмма реализации с использованием канонических звеньев второго порядка с N – 4 представлена на рис. 4.30.

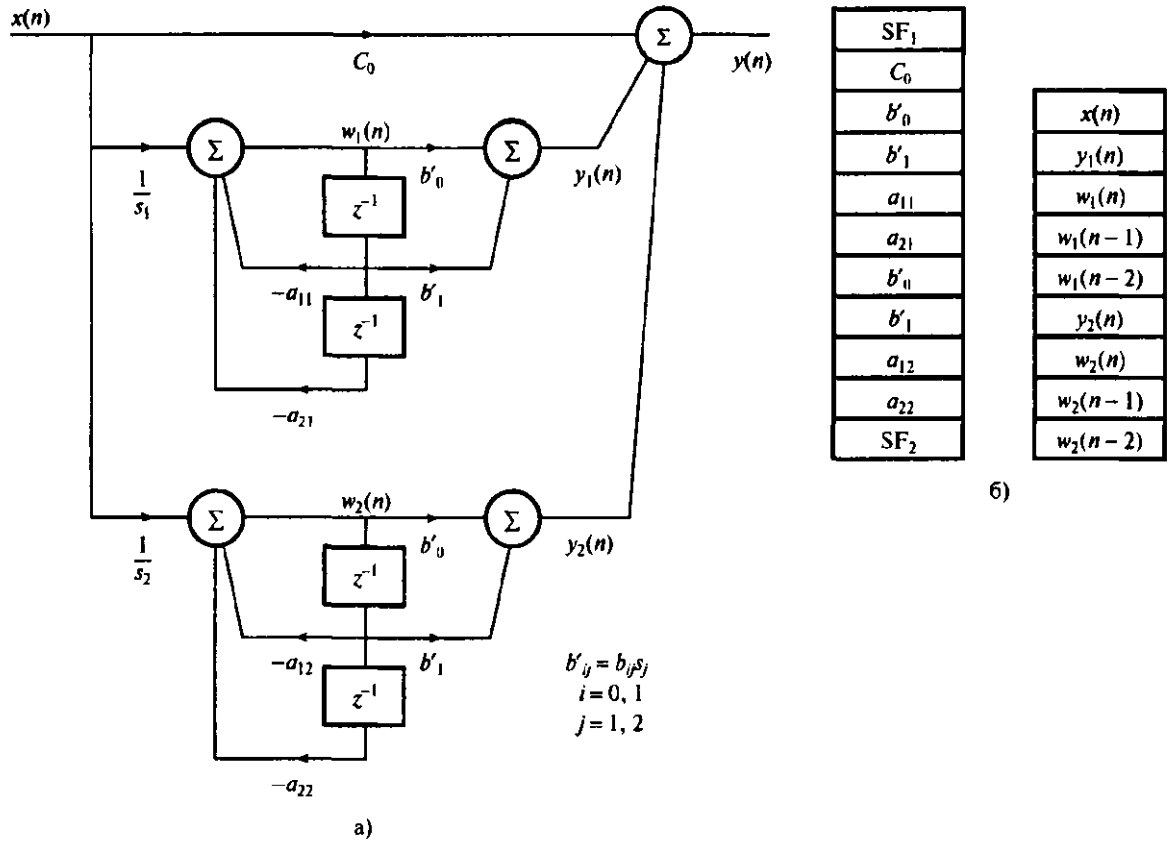


Рис. 4.30. Реализация БИХ-фильтра четвертого порядка: *а* – диаграмма реализации; *б* – схема памяти для хранения данных и коэффициентов

Разностные уравнения канонического звена записываются следующим образом:

$$\omega_1(n) = \mathrm{SF}_1 x(n) - a_{11}\omega_1(n-1) - a_{21}\omega_1(n-2);$$

$$y_1(n) = b_{01}\omega_1(n) + b_{11}\omega_1(n-1);$$

$$\omega_2(n) = \mathrm{SF}_2 x(n) - a_{12}\omega_2(n-1) - a_{22}\omega_1(n-2);$$

$$y_2(n) = b_{02}\omega_2(n) + b_{12}\omega_2(n-1);$$

$$y(n) = c_0 x(n) + y_1(n) + y_2(n).$$

Простой код С для БИХ-фильтра, реализованного как параллельная комбинация канонических звеньев второго порядка, представлен в программе 4.6.

Программа 4.6. Псевдокод С параллельной реализации фильтра четвертого порядка

```
for(n=0; n,(Nsamples-1); ++n) {          /* Nsamples – число выборок данных */
    y[n] = c*x[n];                       /* выход через постоянный тракт */
    for(k=l; k,N; ++k) {
        wk = sk[k]*x[n]-al[k]*wl[k]-a2[k]*w2[k];
        yk = (b0[k]*wk+bl[k]*wl[k])/sk[k];  /* выход первого звена */
        w2[k]=wl[k];                     /* сдвиг и запись данных элемента
        wl[k]=wk;                        задержки*/
        y[n]=yk+y[n];
    }
}
```

Пример 4.5

Представьте передаточную функцию, приведенную в примере 4.4, в параллельной форме, используя в качестве элементарных составляющих канонические звенья второго порядка. Реализуйте фильтр на базе аппаратного обеспечения, указанного в предыдущем примере.

Решение

Используя рассмотренную в главе 4 программу разложения на элементарные дроби, по коэффициентам каскадной реализации получим коэффициенты параллельной реализации. В результате передаточная функция преобразуется к следующему виду:

$$H(z)=\frac{-0{,}1329225-0{,}180523z^{-1}}{1-0{,}0284899z^{-1}+0{,}307046z^{-2}}+$$

$$+\frac{-0{,}058534+0{,}508420z^{-1}}{1-0{,}048489z^{-1}+0{,}0079511z^{-2}}+0{,}24992379;$$

$$s_1=5{,}5244844,\quad s_2=2{,}4794.$$

Значения коэффициентов до и после квантования до 16 бит приведены в табл. 4.5. Коды реализации фильтра на базе процессоров TMS320C50, TMS320C54, DSP56000 и DSP56300 можно найти в [1]. Для реализации на базе процессоров DSP56000 и DSP56300 коэффициенты квантовались до 24 бит.

Расширение методов реализации, описанных выше, на каскадную и параллельную структуры БИХ-фильтров высоких порядков выполняется относительно просто. Впрочем, если реализовать расчет звена второго порядка как подпрограмму, можно получить более компактный код.

Таблица 4.5

Реализация БИХ-фильтра четвертого порядка из примера 4.5: коэффициенты фильтра до и после квантования до 16 бит

	Неквантованные коэффициенты	Квантованные коэффициенты
SF_1	0,18100	5931
C_0	0,24992379	8190
a_{01}	–0,1329225	–24063
a_{11}	–0,1805232	–32670
b_{11}	0,028994	16251
b_{21}	–0,0445416	–24965
a_{02}	–0,058534	–4756
a_{12}	0,5084205	20653
b_{12}	0,0484899	27178
b_{22}	–0,017951	–10061
SF_2	0,40332	13216

Алгоритмы ЦОС подразумевают трудоемкие арифметические операции, в частности, умножение и сложение с интенсивным потоком данных через центральный процессор. Эффективное выполнение подобных алгоритмов в реальном времени требует аппаратной архитектуры и набора команд, радикально отличающихся от архитектуры и команд стандартных микропроцессоров. В цифровых процессорах сигналов для этого используются гарвардская архитектура, конвейерная обработка и специализиро-

ванное аппаратное обеспечение, например, быстрые аппаратные умножители-накопители и схемы сдвига, скоростная внешняя память, также в процессоры внедряется множество команд, ориентированных на ЦОС.

Чтобы удовлетворять требованиям таких многоканальных вычислительно-интенсивных приложений, как современные модемы удаленного доступа к серверу, системы мобильной связи третьего поколения, обработка мультимедийной информации, вводятся новые архитектуры. В частности, это архитектура *VLIW* (*very long instruction word* – с командными словами сверхбольшой длины) и *статическая суперскалярная архитектура*, которые используются в последнем поколении процессоров ЦОС. В обеих архитектурах применяются множественные тракты обработки данных и арифметические устройства, также для повышения эффективности используется параллелизм на уровне команд.

Существует два типа цифровых процессоров сигналов – универсальные процессоры (которые сродни стандартным микропроцессорам, за исключением архитектур и наборов команд, приспособленных для выполнения операций ЦОС) и специализированные процессоры. Последние используются для выполнения специфических алгоритмов ЦОС, например, цифровой КИХ-фильтрации (алгоритмические процессоры ЦОС), или для эффективного выполнения некоторых операций конкретных приложений (прикладные процессоры, или процессоры специального назначения). По сравнению с универсальными процессорами специализированные более быстрые, но менее гибкие.

Здесь подробно рассмотрены основные идеи, реализованные в аппаратуре ЦОС, и влияние алгоритмов ЦОС на архитектуру процессоров ЦОС. Для иллюстрации существенных моментов разобрана реализация нескольких ключевых алгоритмов ЦОС с использованием универсальных цифровых и специализированных процессоров сигналов.

4.6. Микропроцессорный комплекс TMS320VC5510

4.6.1. Структура процессора

В курсовом проектировании используется цифровой сигнальный процессор (ЦСП) с фиксированной точкой TMS320VC5510/5510A, базирующийся на ядре TMS320C55x. Это связано с тем, что данная архитектура обладает высокой производительностью при низком энергопотреблении благодаря повышению параллелизма и уделению особого внимания вопросам снижения потребляемой мощности. ЦПУ обладает внутренней шинной структурой, состоящей из программной шины, трех шин чтения данных, двух шин записи данных и вспомогательных шин для периферии и кон-

троллера прямого доступа к памяти (DMA). Это позволяет выполнять до трех операций чтения данных и двух операций записи данных за один цикл, при этом контроллер DMA может осуществить до двух операций перемещения данных без задействования ЦПУ.

Ядро ЦСП C55x обладает двумя модулями умножения – накопления (MAC), каждый из которых способен выполнять операции типа «умножение 17-бит x 17-бит» за один цикл. Центральное 40-битное арифметическо-логическое устройство (АЛУ) сопровождается вспомогательным 16-битным АЛУ. Алгоритм совместного использования двух АЛУ определяется набором инструкций, обеспечивая оптимальную параллельную работу и снижение энергопотребления. Распределение ресурсов возложено на адресное устройство (АУ) и устройство данных (УД) ядра ЦСП C55x.

Семейством цифровых сигнальных процессоров C55x поддерживаются инструкции с переменным числом байт, что позволяет увеличить плотность кода. Модуль инструкций (МИ) осуществляет 32-битную выборку инструкций из внутренней либо внешней памяти и определяет очередь инструкций для программного модуля (ПМ). В свою очередь, ПМ декодирует инструкции, определяет задачи для АУ и УД и управляет защищенным конвейером. Для предотвращения переполнения конвейера при выполнении условных переходов используется предсказание переходов. Процессоры 5510/5510A также имеют встроенный кэш инструкций размером 24 Кбайт для снижения числа операций доступа к внешней памяти, повышения производительности и снижения энергопотребления.

Набор периферийных устройств процессоров 5510/5510A включает в себя интерфейс внешней памяти (EMIF), обеспечивающий непосредственное подключение различных типов асинхронной памяти, таких, как EPROM и SRAM, а также высокоскоростных запоминающих устройств высокой плотности, таких, как синхронные DRAM и SRAM с пакетной выборкой. Три полнодуплексных многоканальных буферизованных последовательных порта (McBSP) обеспечивают непосредственное подключение большого ряда устройств со стандартными последовательными интерфейсами и многоканальный обмен (до 128 каналов с индивидуальным запретом). Порт управляющего контроллера (EHPI) представляет собой 16-битный параллельный интерфейс, обеспечивающий внешнему управляющему процессору доступ к встроенной памяти процессоров 5510/5510A, и может быть сконфигурирован как в мультиплексный, так и в немультиплексный режим, что позволяет использовать его совместно с самыми различными управляющими процессорами. Контроллер прямого доступа к памяти (DMA) обеспечивает перемещение данных по шести независимым

каналам без вмешательства ЦПУ, его суммарная пропускная способность составляет до двух 16-битных слов за цикл. Кроме этого, набор периферийных устройств включает в себя два таймера общего назначения, восемь выводов общего назначения (GPIO) и генератор с ФАПЧ и цифровым управлением (DPLL).

Процессоры 5510/5510A поддерживаются программным обеспечением eXpressDSP™, которое состоит из интегрированной среды разработки (IDE) Code Composer Studio™, РТОС DSP/BIOS™ и документации TMS320™ DSP Algorithm Standard; кроме этого, существует большое количество разработок сторонних производителей. IDE Code Composer Studio состоит из компилятора языка С, линкера Visual Linker, симулятора, поддержки обмена данными в реальном режиме времени Real-Time Data Exchange (RTDX™), драйверов эмулятора XDS510™ и библиотек Chip Support Libraries (CSL). DSP/BIOS представляет собой расширяемое модульное программное обеспечение, бесплатно доступное пользователям цифровых сигнальных процессоров фирмы Texas Instruments и состоящее в основном из планировщика задач и средств поддержки систем реального времени с весьма экономичным расходованием памяти и других ресурсов процессора. TMS320 DSP Algorithm Standard – это набор спецификаций, выполнение которых обеспечивает совместимость кода, созданного различными разработчиками, что значительно облегчает его дальнейшую интеграцию в проекты. Сеть сторонних разработчиков фирмы Texas Instruments объединяет более 400 участников, предлагающих пользователям массу готовых комплексных и компетентных решений.

Ядро процессоров TMS320C55x базируется на открытой архитектуре с добавлением специфических модулей, позволяющих значительно ускорить обработку некоторых алгоритмов. Данные модули обеспечивают процессорам 5510/5510A оптимальное соотношение производительности, присущей процессорам с фиксированной точкой, низкого энергопотребления и цены, что является уникальным сочетанием на рынке видеопроцессоров. Дополнительные модули обеспечивают процессорам 5510/5510A высокую производительность видеокодека, в то же время оставляя незадействованными более половины ресурсов ядра, что позволяет реализовывать параллельно такие функции, как преобразование цветовых схем, организация пользовательского интерфейса, функции безопасности, стек TCP/IP, распознавание голоса, преобразование текста в голос и т.д. В результате цифровой сигнальный процессор 5510/5510A способен в одиночку решить практически все задачи, стоящие перед системами фильтрации.

4.6.2. Отличительные особенности микропроцессорного комплекса

1. Высокопроизводительное ядро для операций с фиксированной точкой с низким энергопотреблением TMS320C55™ (рис. 4.31):
   а) время цикла 6.25 / 5 нс;
   б) тактовая частота 160 / 200 МГц;
   в) выполнение одной или двух операций за такт;
   г) два умножителя (производительность до 400 миллионов умножений с накоплением в секунду (MMACS);
   д) два арифметическо-логических устройства (АЛУ);
   е) внутренняя программная шина;
   ж) три внутренних шины чтения данных/операндов;
   з) две внутренних шины записи данных/операндов.
2. Кэш инструкций (24 КБайта).
3. Встроенное ОЗУ 160К х 16 бит, состоящее из:
   а) 8 блоков по 4К х 16 бит двухпортовой ОЗУ (DARAM) (всего 64 КБайт);
   б) 32 блоков по 4К х 16 бит однопортовой ОЗУ (SARAM) (всего 256 КБайт).
4. Встроенное ПЗУ 16К х 16 бит (32 КБайт).
5. Общий объем адресуемой памяти 8М х 16 бит.
6. 32-битный интерфейс внешней памяти (EMIF) с возможностью подключения памяти типа:
   а) асинхронное статическое ОЗУ (SRAM);
   б) асинхронное EEPROM;
   в) синхронное динамическое ОЗУ (SDRAM);
   г) синхронное статическое ОЗУ с пакетной выборкой (SBSRAM).
7. Программный контроль энергопотребления шести функциональных блоков из внутренних устройств.
8. Встроенные периферийные устройства:
   а) два 20-битных таймера;
   б) шестиканальный контроллер прямого доступа к памяти (DMA);
   в) три многоканальных буферизованных последовательных порта (McBSP);
   г) 16-битный параллельный порт управляющего контроллера (EHPI).

9. Программируемый тактовый генератор с ФАПЧ:
   а) восемь портов ввода – вывода общего назначения (GPIO) и выход общего назначения (XF);
   б) встроенный эмулятор;
   в) поддержка периферийного сканирования по стандарту JTAG[1].
10. Варианты корпусов:
   а) 240-выводный корпус MicroStar BGA™ (суффикс GCW);
   б) 240-выводный корпус MicroStar BGA™ (суффикс ZGW) без содержания свинца (Pb-free).
11. Питание ядра 1,6 В.
12. Питание портов ввода – вывода 3,3 В.

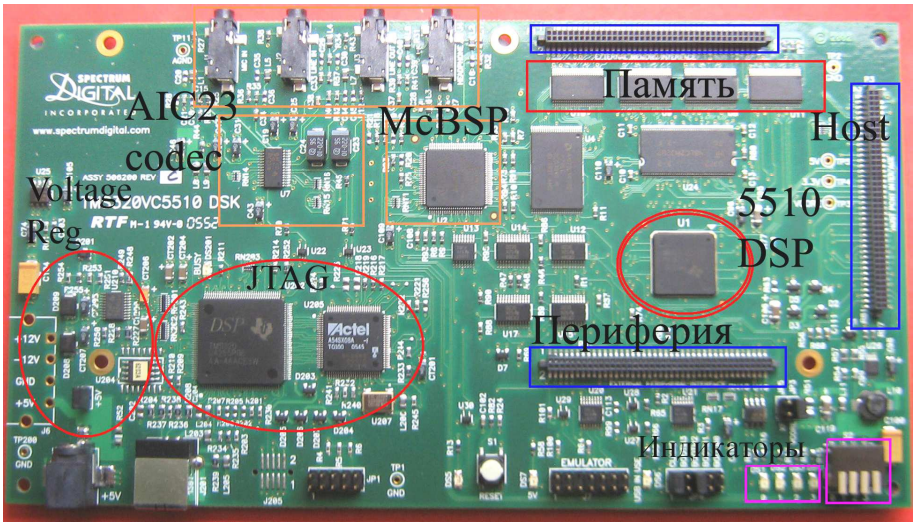


Рис. 4.31. Внешний вид TMS320VC5510

Память процессора состоит из адресов регистров и адресов данных программы, содержащей константы и тело программы. Обращение к памяти процессора может осуществляться по адресам. Аппаратно доступ к блокам, описанным в табл. 4.6, может осуществляться параллельно, т.е. два доступа за один цикл (два чтения, две записи или чтение и запись). DRAM – *Dual-Access RAM*(Память с двойным доступом).

Таблица 4.6

Разбиение на блоки областей памяти с двойным доступом

Шестнадцатеричный адрес	Память
000000h − 001 FFFh	DARAM 0
002000h − 003FFFh	DARAM 1
004000h − 005FFFh	DARAM 2
006000h − 007FFFh	DARAM 3

[1] Стандарт IEEE 1149.1-1990[1]

Внутри кристалла также имеются области памяти с одиночным доступом (SDRAM) (рис. 4.32).

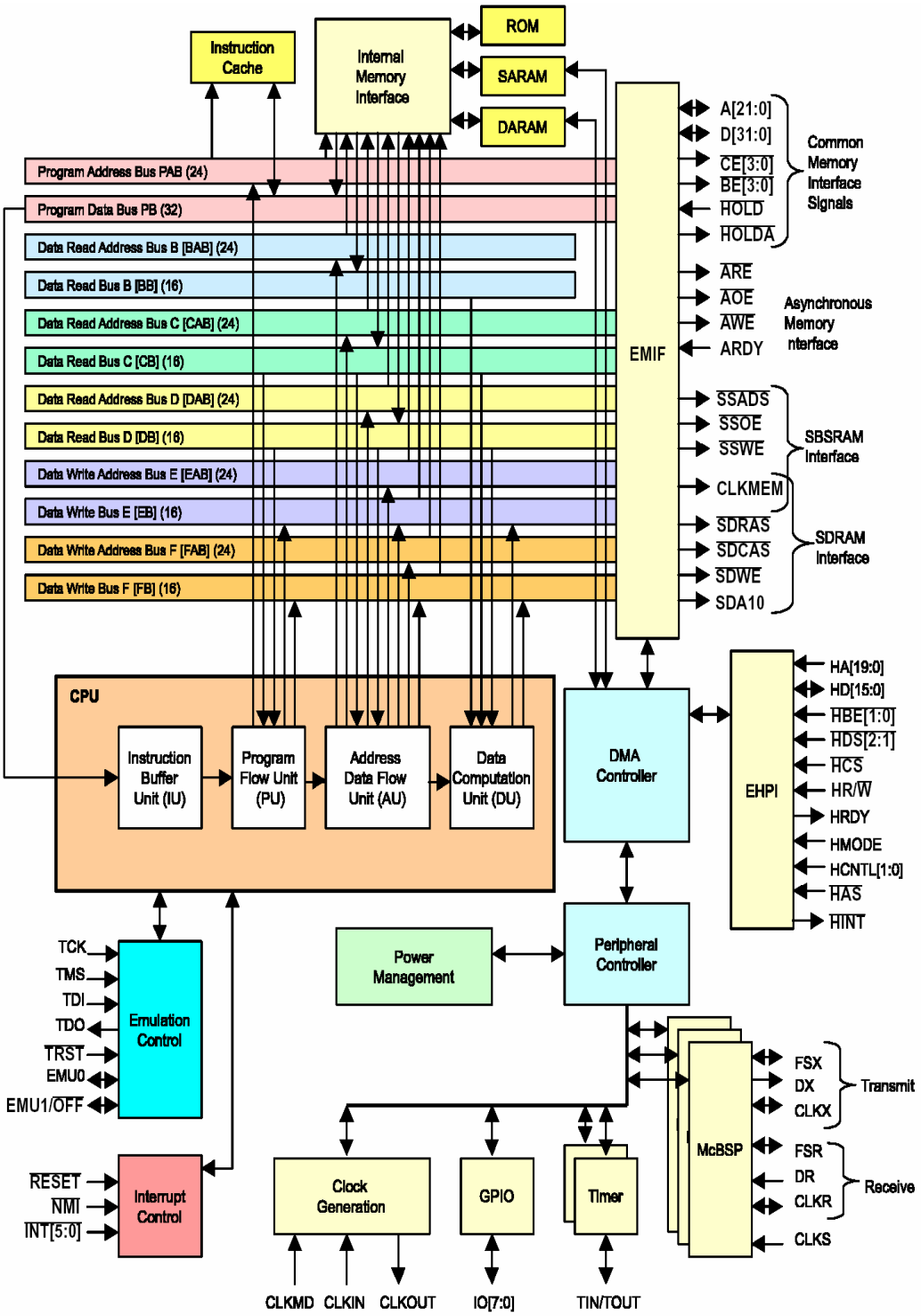


Рис. 4.32. Блок-схема внутрикристальных компонентов с функциональными выводами

В табл. 4.7 обозначено деление этих областей по адресам.

Таблица 4.7

Области памяти с одиночным доступом

Шестнадцатеричный адрес	Область памяти	Шестнадцатеричный адрес	Область памяти
010000h−011 FFFh	SARAM 0	030000h−031 FFFh	SARAM 16
012000h−013FFFh	SARAM 1	032000h − 033FFFh	SARAM 17
014000h−015FFFh	SARAM 2	034000h − 035FFFh	SARAM 18
016000h−017FFFh	SARAM 3	036000h − 037FFFh	SARAM 19
018000h−019FFFh	SARAM 4	038000h − 039FFFh	SARAM 20
01A000h−01BFFFh	SARAM 5	03A000h − 03BFFFh	SARAM 21
01C000h−01DFFFh	SARAM 6	03C000h − 03DFFFh	SARAM 22
01E000h−01FFFFh	SARAM 7	03E000h − 03FFFFh	SARAM 23
020000h − 021 FFFh	SARAM 8	040000h − 041 FFFh	SARAM 24
022000h − 023FFFh	SARAM 9	042000h − 043FFFh	SARAM 25
024000h − 025FFFh	SARAM 10	044000h − 045FFFh	SARAM 26
026000h − 027FFFh	SARAM 11	046000h − 047FFFh	SARAM 27
028000h − 029FFFh	SARAM 12	048000h − 049FFFh	SARAM 28
02A000h − 02BFFFh	SARAM 13	04A000h − 04BFFFh	SARAM 29
02C000h − 02DFFFh	SARAM 14	04C000h − 04DFFFh	SARAM 30
02E000h − 02FFFFh	SARAM 15	04E000h − 04FFFFh	SARAM 31

Регистры управления процессора

SYSR – Регистр системы (SYSR), обеспечивает управление над определенными для устройства функциями. SYSR расположен в 07FDh (адреса порта).

DMA_CCR – Регистр контроля прямого доступа к памяти. Регистр прямого доступа к памяти позволяет синхронизировать каналы передачи данных по отобранным событиям. Поддерживается 14 отдельных синхронизирующих событий, и каждый канал может быть привязан, чтобы отделить синхронизирующие события, независимые от других каналов. Синхронизирующие события могут быть отобраны путем программирования области SYNC в определенном для канала регистре контроля – канала прямого доступа к памяти (DMA_CCR).

EHPI – обеспечивает 16-битный параллельный интерфейс удаленного управления со следующими особенностями:

1. 20- битовую шину адреса хоста.
2. 16-битовый шину данных хоста.
3. Мультиплексированные и демультиплексированные способы передачи по шинам.

4. Доступ хоста к памяти SARAM, DARAM и внешней памяти.

5. 20-битовый регистр адреса (в мультиплексном способе) со способностью автоприращения к более быстрым передачам.

6. HRDY используют для подтверждения связи с ведущим.

IODATA, IODIR − регистры управления портами ввода вывода интерфейса GPIO. Этот интерфейс обеспечивает восемь точек ввода/вывода общего назначения IO0 – IO7. Каждая точка может независимо формироваться как вход или вывод, используя регистр руководства ввода/вывода (IODIR). Регистр данных ввода/вывода (IODATA) используется, чтобы контролировать логическое состояние точек, формируемых как входы, и управлять логическим состоянием точек, формируемых как выходы. IODIR и IODATA доступны для центрального процессора и диспетчеру прямого доступа к памяти в адресах в месте ввода/вывода.

Системный регистр (***SYSR***) обеспечивают управление над определенными для устройства функциями. SYSR расположен по адресу 07FDh порта (табл. 4.8).

Таблица 4.8

Системный регистр (побитное представление)

Резерв	HPE	BH	HBH	BOOTM3	Резерв	Резерв	Резерв	CLKDIV
15 – 10	9	8	7	6	5	4	3	2 – 0
R-0000	R/W-1	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-000

Легенда: R – чтение; W – запись; *n* – значение после рестарта.

Расшифровка битов системного регистра представлена в прил. А. Регистры процессора **VC5510** представлены в прил. Б.

Вопросы и задания для самопроверки

1. Цифровые сигнальные процессоры: определение, особенности и области применения, классификация ЦСП.
2. ЦСП фирмы Texas Instruments семейства TMS320: классификация, условные обозначения. Особенности процессоров TMS320C55x.
3. Назовите отличительные черты процессоров ЦОС.
4. Сформулируйте основные особенности реализации КИХ-фильтров на ЦСП.
5. Каковы источники погрешностей ЦОС, связанные с конечной разрядностью чисел?
6. Какие способы представления и форматы чисел используются в ЦОС?
7. Перечислите архитектуры ЭВМ.
8. Приведите упрощенную архитектуру стандартных микропроцессоров.
9. Приведите стандартную универсальную аппаратную архитектуру обработки сигналов.
10. Принципиальная особенность гарвардской архитектуры.
11. Сформулируйте концепцию конвейерной обработки данных.
12. Приведите структурную схему умножителя-накопителя.
13. Приведите специализированные команды процессоров ЦОС, которые оптимизированы для цифровой обработки сигналов.
14. Раскройте принципы архитектуры VLIW.
15. Сформулируйте основные особенности SIMD-архитектуры.
16. Какие процессоры подразумеваются под универсальными?
17. Объясните принцип представления чисел с фиксированной запятой.
18. Объясните принцип представления чисел с плавающей запятой.
19. В чем отличие упрощенной архитектуры процессора ЦОС с фиксированной запятой третьего поколения от упрощенной архитектуры процессора ЦОС с фиксированной запятой второго поколения?
20. На основании каких параметров выбирают цифровой процессор для обработки сигналов?
21. Опишите особенности реализации цифровой КИХ-фильтрации.
22. Опишите особенности реализации цифровой БИХ-фильтрации.

МОДУЛЬ 5. ПРИМЕНЕНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ

5.1. Области применения цифровой обработки сигналов

Цифровая обработка сигналов (ЦОС) является одной из самой мощных технологий, которая охватит науку и технику в двадцать первом веке. Революционные изменения уже коснулись *широкого* круга областей – коммуникация, медицинская визуализация изображений, радиолокация и гидролокация, высококачественное воспроизведение музыки, разведка нефтяных месторождений и многие другие. В каждой их этих областей произошло *глубокое* проникновение технологии цифровой обработки сигналов, включающее разработку собственных алгоритмов, математических выводов и специальных методов. Эта комбинация «ширины» и «глубины» проникновения цифровой обработки сигналов делает невозможным индивидуальное овладение всей технологией, которая уже разработана к настоящему времени. Обучение ЦОС включает в себя две задачи: усвоение основных принципов и изучение специализированных методов, которые необходимы для применения ЦОС в конкретных областях науки и техники. Эта глава начинает наше приключение в мир Цифровой Обработки Сигналов, описывая эффект, который произвела технология ЦОС в различных областях науки и техники. Революция началась!

5.1.1. Истоки DSP

Цифровая обработка сигналов выделяется из других областей знаний вычислительной техники уникальным типом данных, который она использует – *сигналы*. В большинстве случаев эти сигналы представляют собой информацию от датчиков из окружающего мира – сейсмические колебания, визуальные изображения, звуковые волны и т.п. ЦОС представляет собой математику, алгоритмы и методы, используемые для обработки этих сигналов, после того как они были переведены в дискретную форму. Обработка может содержать в себе несколько задач: улучшение визуального отображения, распознавание и синтез речи, сжатие данных для их хранения и передачи и др. Предположим, что вы подсоединили аналого-цифровой преобразователь к компьютеру и используете его для сбора некоторого количества данных. ЦОС ответит на вопрос, *что делать дальше.*

История ЦОС начинается в 60-х годах, когда появились первые цифровые компьютеры, но в то время они были дорогими, и поэтому область применения ЦОС ограничивалась несколькими приложениями. Первыми из них стали: *гидро- и радиолокация,* в которых был интерес национальной обороны, *разведка нефтяных месторождений*, где можно было заработать большие деньги, *освоение космоса,* где существовала нехватка данных, и медицинская визуализация изображений, где речь шла о жизни людей.

Революция, произведенная персональными компьютерами в 80-х и 90-х годах, расширила применения ЦОС новыми приложениями. Кроме военных и правительственных нужд, технология цифровой обработки сигналов широко распространилась на коммерческий рынок. ЦОС получила распространение в таких общественных приложениях, как мобильные телефоны, проигрыватели компакт-дисков и электронная речевая почта. На рис. 5.1 показаны некоторые из этих областей применения.

Техническая революция привела к тому, что в 80-х годах ЦОС была включена в курс обучения электротехники для получения ученой степени. Десятью годами позже ЦОС стала обязательной частью обучения студентов. Сегодня технология цифровой обработки сигналов – базовые знания, которые необходимы ученым и инженерам.

ЦОС

Космос	- улучшение космических фотографий - сжатие данных - анализ сигналов от удаленных космических зондов, полученных от интеллектуальных сенсоров
Медицина	- диагностика медицинских изображений - анализ электрокардиограмм - хранение и чтение медицинских изображений
Коммерция	- сжатие изображений и звука для мультимедийных презентаций - специальные эффекты в кино - созыв видео конференций
Телефония	- сжатие данных и речевых сигналов - подавление эха - мультиплексирование сигналов - фильтрация
Армия	- радиолокация - ультразвуковая локация - наведение на цели - защищенная связь
Промышленность	- предсказание месторождений нефти и минералов - процессы проверки и управления - неразрушающий контроль - CAD и инструменты проектирования
Наука	- запись и анализ землетрясений - сбор данных - спектральный анализ - моделирование

Рис. 5.1. Области применения ЦОС

По аналогии ЦОС может быть сравнен с предыдущей технической революцией – *электроникой*. Пока эра электротехники продолжается, почти каждый ученый и инженер имеет основные навыки проектирования принципиальных схем. Без этого они бы потерялись в технологическом мире. ЦОС ждет то же будущее.

Предположим, что вы столкнулись с задачей, которая требует знаний ЦОС, и обратились к учебникам и другим публикациям, чтобы найти решение. В обычном случае вы найдете страницы уравнений, неизвестных математических символов и незнакомой терминологии. Большинство литературы по ЦОС непонятно даже для тех, кто обладает опытом в этой области. Это не означает, что материал, который там представлен, неправильный, просто он предназначен для очень специализированной аудитории. Современные исследователи нуждаются в этой детализированной математике, чтобы понять теоретические предпосылки работы.

Основным предположением этой главы является то, что методы ЦОС могут быть изучены и использованы без традиционных препятствий в виде детализированной математики и теории.

Далее в этой главе иллюстрируются области, в которых ЦОС произвела революционные изменения. Читая про каждое приложение, заметьте, что ЦОС является междисциплинарной наукой, поскольку включает в себя множество смежных областей. Как показано на рис. 5.2, границы между ЦОС и другими техническими дисциплинами определены не строго, а скорее размыты или перекрываются. Поэтому, если вы хотите специализироваться по ЦОС, это потребует дополнительного обучения другим дисциплинам.

Рис. 5.2. Границы между ЦОС и другими техническими дисциплинами определены не строго, а скорее размыты и перекрываются

5.1.2. Телекоммуникации

Телекоммуникация осуществляет передачу информации из одного местоположения в другое. Типы информации включают в себя телефонные разговоры, телевизионные сигналы, компьютерные файлы и др. Чтобы передать информацию, требуется канал связи, соединяющий два пункта. Это может быть витая пара, радиосигнал, светопровод и т.д. Телекоммуникационные компании взимают *плату* за передачу информации заказчика, поскольку они должны оплатить проведение канала и поддерживать его работоспособность. Финансовая стратегия компании проста: чем больше информации она может пропустить через один канал, тем больше денег будет заработано. ЦОС совершила революцию телекоммуникационной индустрии во многих областях: генерирование и детектирование тоновых сигналов, сдвиг полосы частот, фильтрация сетевых радиопомех и т.д. Мы обсудим три специфических примера из телефонных сетей: мультиплексирование, сжатие и подавление отраженных сигналов.

5.1.3. Мультиплексирование

В мире приблизительно *один биллион* телефонов. При нажатии нескольких кнопок коммутируемая сеть позволяет за пару секунд установить соединение между двумя людьми. Необъятность этой задачи просто поражает! До 60-х годов соединение требовало прохождения аналоговых речевых сигналов через механические ключи и усилители. Одно соединение требовало одну пару проводов. ЦОС преобразует звуковые сигналы в поток последовательных цифровых данных. Поскольку биты могут упаковываться и далее разделяться, несколько телефонных разговоров могут передаваться по одному каналу связи. Например, телефонный стандарт, известный как *система T-carrier*, может одновременно передавать 24 речевых сигнала. Каждый речевой сигнал дискретизируется 8000 раз в секунду с использованием 8-битного аналого-цифрового преобразования с компандированием (логарифмическое сжатие). Результат по каждому речевому сигналу будет представлять собой поток со скоростью 64000 бит/с, а все 24 канала будут заключены в потоке 1,544 Мб/с. Этот сигнал может передаваться по обыкновенной телефонной линии на расстояние до 1800 м. Экономическое преимущество цифровой передачи громадно. Провода и аналоговые ключи дорогостоящие, цифровые логические элементы дешевы.

5.1.4. Сжатие

Когда речевой сигнал оцифровывается с дискретизацией 8000 выборок в секунду, большая часть полученной информации *избыточна*. То есть, информация, содержащаяся в одном отсчете, часто повторяется в последующих отсчетах. Для того чтобы уменьшить поток данных, разработаны множество алгоритмов *сжатия* данных. Для восстановления оригинальной формы сигнала применяются соответствующие алгоритмы *распаковки данных*. Эти алгоритмы различаются по степени достигаемого сжатия и качества восстановленного звука. Например, уменьшение потока от 64 Кб/с до 32 Кб/с не оказывает особенного влияния на качество звука. При сжатии потока данных до 8 Кб/с качество звукового сигнала заметно ухудшается, но является приемлемым для передачи по телефонным линиям. Наибольшее достигаемое сжатие потока данных составляет около 2 Кб/с. При этом звуковой сигнал получается значительно искаженным, но может использоваться в некоторых приложениях, например, в военной или подводной коммуникации.

5.1.5. Подавление отраженных сигналов

Отраженные сигналы (эхо) представляют собой серьезную проблему при передаче сигналов на длинные расстояния. При телефонном разговоре сигнал, представляющий голос, доходит до приемного устройства, после чего его часть возвращается в виде отраженного сигнала. Если расстояние между передатчиком и приемником несколько сотен километров, время возврата отраженного сигнала составляет несколько миллисекунд. Ухо человека не замечает отраженный сигнал при таких небольших задержках, и качество воспринимаемого звука будет нормальным. Если расстояние значительно больше, отраженные сигналы становятся заметными и раздражающими. Время возврата эха при межконтинентальных соединениях может составлять несколько сотен миллисекунд. ЦОС борется с этой проблемой, измеряя возвращающийся сигнал и генерируя соответствующий *антисигнал* для подавления эха. Такой метод позволяет пользователям телефонов одновременно и говорить, и слушать без задержки звукового сигнала. Метод может применяться и для подавления шума окружающей среды генерируемым *антишумом*.

5.1.6. Обработка звуковых сигналов

Два основных чувства восприятия человека – это зрение и слух. Поэтому большая часть ЦОС посвящена обработке изображений и звуковых сигналов. Люди слушают как *музыку*, так и *речь*. ЦОС совершила революционные изменения в этих областях.

5.1.7. Музыка

Путь от микрофона музыканта до акустической системы любителя музыки очень долог. Цифровое представление данных в основном используется для предотвращения ухудшения, связанного с аналоговыми запоминающими устройствами и их работой. Это легко понять, если сравнить качество музыки, записанной на аудиокассете и компакт-диске. Обычно музыкальное произведение записывается в звуковой студии на нескольких каналах или дорожках. В некоторых случаях каждый инструмент и голос исполнителя записываются отдельно. Это предоставляет звукорежиссеру большую гибкость в работе над финальным продуктом. Комплексный процесс объединения отдельных треков называется *сведением фонограмм*. ЦОС обеспечивает множество важных функций, необходимых при сведении фонограмм, включая фильтрацию, сложение и вычитание сигналов, редактирование сигнала и т.п.

Одно из самых интересных приложений ЦОС в подготовке музыки – это *искусственная реверберация*. Если отдельные каналы просто складываются вместе, полученный звук будет слабым и приглушенным, как если бы музыканты играли на открытом воздухе. ЦОС позволяет добавить искусственное эхо и реверберацию, чтобы имитировать различные условия, окружающие слушателя. Например, эхо с задержкой несколько сотен миллисекунд дает эффект исполнения музыкального произведения в кафедральном соборе. Добавление эха с задержкой 10 – 20 секунд вызывает ощущения прослушивания музыки в меньшем по размерам помещении.

5.1.8. Генерация речи

Генерация и распознавание речи используется для установления связи между человеком и машиной. Человек чаще использует свои рот и уши, чем руки и глаза. Это особенно удобно, если руки и глаза заняты чем-то другим, например, вождением автомобиля или проведением хирургической операции. Для генерации речи используются два подхода – *цифровая*

запись и *имитация речевого тракта*. При цифровой записи речевой сигнал дискретизируется и хранится, обычно в сжатой форме. При воспроизведении записанные данные восстанавливаются и преобразуются в аналоговый сигнал. Целый час записанного речевого сигнала требует всего лишь три мегабайта памяти и может быть размещен даже на машинах с небольшими аппаратными ресурсами. Такой подход является на сегодняшний день самым распространенным.

Имитаторы речевого тракта значительно сложнее, так как они пытаются имитировать физический механизм, которым люди создают речь. Речевой тракт человека – это акустический резонатор с частотами, определяемыми размерами и формой полости рта. Звук в речевом тракте создается из двух основных компонент, названных *вокализованными* и *фрикативными* звуками. При вокализованных звуках вибрация голосовых связок производит близкие к периодическим вибрации воздуха в ротовой полости. Фрикативные звуки порождаются при прохождении воздуха через сжатые губы или зубы. Действие имитаторов речевого тракта основано на генерации цифровых сигналов, которая имеет похожие два типа возбуждения. Характеристики акустического резонатора имитируются прохождением сигнала через цифровой фильтр с подобными резонансами. Этот подход использовался в одном из успешных ранних применений ЦОС – Speak & Spell, электронном обучающем помощнике для детей.

5.1.9. Распознавание речи

Автоматизированное распознавание речи гораздо сложнее, чем генерация. Распознавание речи – это классический пример того, что человек делает хорошо, а цифровой компьютер плохо. Цифровой компьютер может хранить и вызывать большое количество данных, выполнять математические расчеты с огромной скоростью и повторять одну и ту же задачу, не уставая и без потери качества. К сожалению, современные компьютеры работают плохо, когда сталкиваются с необработанными данными от датчиков. Обучить компьютер ежемесячно посылать счет на оплату легко. Научить тот же компьютер распознавать ваш голос – сложное дело. ЦОС пытается решить проблему распознавания речи в два этапа: *выделение признаков* с последующим *сопоставлением признаков*. Каждое слово речевого сигнала выделяется и анализируется, чтобы определить тип возбуждения и резонансные частоты. *Эти* параметры сравниваются с записанными ранее примерами слов, чтобы найти среди них наиболее похожее. Часто такие системы ограничены одной или несколькими сотнями слов; спо-

собны воспринимать речь с отчетливыми паузами между словами; должны быть натренированы для индивидуального голоса. Эти ограничения некритичны для некоторых коммерческих приложений, однако в этой области предстоит еще большая работа.

5.1.10. Эхолокация

Самый распространенный метод получения информации об удаленном объекте состоит в получении от него отраженных волн. Например, радар посылает импульсы радиоволн и принимает отраженный сигнал от воздушных объектов. В сонаре для определения подводных объектов через воду посылаются звуковые волны. Геофизики зондируют землю, производя взрывы и анализируя отраженный сигнал от залегающих горных пород. В то время, как эти приложения имеют некоторые общие принципы, каждое из них обладает собственными специфическими проблемами и потребностями. ЦОС произвела революционные преобразования во всех трех областях.

5.1.11. Радиолокация

Слово *radar* (радар) является аббревиатурой от *Radio Detection And Ranging* (радиообнаружение и определение расстояния). В простейшей радиолокационной системе радиопередатчик излучает импульсы радиочастотной энергии длительностью несколько микросекунд. Этот импульс подается на передающую антенну, откуда радиоволна распространяется со скоростью света. Воздушное пространство на пути этой волны будет отражать некоторую часть энергии обратно в направлении приемной антенны, размещенной около передающей. Дистанция до объекта вычисляется из прошедшего времени между переданным импульсом и принятым отражением. Направление на объект находят просто, так как известно, куда была направлена антенна, когда был принят отраженный сигнал.

Диапазон действия радиолокационной системы определяется двумя параметрами: энергией в начальном импульсе и уровнем шума радиоприемника. К сожалению, увеличение энергии в импульсе обычно сопровождается его удлинением. В свою очередь, увеличение продолжительности импульса понижает точность измерения времени запаздывания. Это отражается в противоречии двух важных параметров – возможностью определения объектов на дальних расстояниях и возможностью точно определить расстояние до объекта.

ЦОС произвела революцию в трех областях радиолокации. Во-первых, ЦОС может *сжать* импульс после приема, что приводит к лучшему определению расстояния до объекта без уменьшения рабочего диапазона. Во-вторых, ЦОС может фильтровать принятый сигнал, чтобы снизить шум. Это увеличивает диапазон без снижения точности определения расстояния. В-третьих, ЦОС способна быстро выбирать и генерировать импульсы различной формы и длительности. Кроме прочего, это позволяет оптимизировать импульс для специфичных задач обнаружения. И самое удивительное – многие эти функции выполняются со скоростями выборок, сравнимыми с используемой радиочастотой – выше нескольких сотен мегагерц! После прихода ЦОС в радиолокацию потребовалось как проектирование высокоскоростного оборудования, так и разработка эффективных алгоритмов.

5.1.12. Гидролокация

Слово *sonar* (сонар) – это аббревиатура от *Sound Navigation And Ranging* (звуковая навигация и определение расстояния). Сонары (звуковые локаторы) делятся на две категории – *активную* и *пассивную*. В активном звуковом локаторе звуковые импульсы в диапазоне от 2 КГц до 40 КГц передаются в воду и полученные отраженные сигналы обнаруживаются и анализируются. Использование активного сонара включает в себя обнаружение и фиксацию местоположения подводных объектов, навигацию, связь, картографирование морского дна. Типичный максимальный диапазон действия составляет от 10 до 100 километров.

Пассивным звуковым локатором просто *прослушивают* подводные звуки, которые включают в себя природную турбулентность, морскую жизнь и механические звуки от подводных лодок и надводных кораблей. Поскольку пассивный сонар не излучает энергии, он является идеальным для секретных операций, поскольку можно обнаружить кого-либо без того, чтобы обнаружили вас. Основное применение пассивной звуковой локации – военные разведывательные системы, которые обнаруживают курсы подводных лодок. Пассивный сонар обычно использует более низкие частоты, чем активный, так как они распространяются в воде с меньшими потерями. Диапазон обнаружения может составлять тысячи километров.

В гидролокации ЦОС произвела революцию в тех же областях, что и в радиолокации: генерация импульса, сжатие импульса и фильтрация обнаруженных сигналов. На первый взгляд, сонар проще, чем радар, потому что он связан с более низкими частотами. С другой стороны, сонар более

сложен, чем радар, потому что окружающая среда намного менее однородна и стабильна. Системы гидролокации обычно включают в себя множество передающих и приемных элементов. При корректном управлении и смешивании сигналов в этом множестве элементов сонары могут фокусировать передаваемый импульс в желаемом направлении и определять направление, откуда принят отраженный сигнал. Чтобы управлять этим множеством каналов, системы гидролокации требуют такую же вычислительную мощность, как и радары.

5.1.13. Сейсморазведка методом отраженных волн

В 20-х годах геофизики открыли, что структура земной коры может быть исследована с помощью звука. Геологоразведчики научились устраивать взрывы и записывать отражения от граничных слоев до десяти километров ниже поверхности земли. Эти отраженные сейсмограммы интерпретировались для представления внутренних структур. Метод отраженных волн быстро стал основным методом обнаружения залежей нефти и минералов и остается им до сегодняшнего дня.

В идеальном случае звуковой импульс, посылаемый в землю, производит единичное отражение, проходя через каждый граничный слой. К сожалению, ситуация не так проста. Каждое отражение, вернувшееся к поверхности, должно пройти через все другие граничные слои, расположенные выше. Это может быть причиной увеличения отражения между слоями, вызывающее *отражения от отражений*, которые будут обнаружены на поверхности. Такие вторичные отражения могут усложнить обнаруженный сигнал и, соответственно, его интерпретацию. С 60-х годов ЦОС широко используется для разделения начального и вторичного отражения в сейсмограммах. Как же обходились геофизики без ЦОС? Ответ прост – они искали в *простых* местах, где многочисленные отражения были минимизированы. ЦОС позволяет находить нефть в *труднодоступных* местах, например, под океаном.

5.1.14. Обработка изображений

Изображения являются сигналами с особенными характеристиками. Во-первых, они измеряются параметрами *пространства* (расстояния), в то время как большинство сигналов измеряются параметрами во *времени*. Во-вторых, они содержат много информации, например, для записи одной

секунды телевизионного видео может потребоваться более 10 Мб. Это в тысячу раз больше, чем для речевого сигнала той же продолжительности. В-третьих, окончательная оценка качества чаще зависит от человеческого вкуса, чем от объективных критериев. Эти особенные характеристики сделали обработку изображений отдельной подгруппой в ЦОС.

5.1.15. Медицина

В 1895 году Вильгельм Конрад Рентген обнаружил, что *X-rays* (рентгенолучи) могут проходить через материю. С появлением возможности наблюдать изнутри человеческое тело в медицине произошла революция. Медицинские системы на рентгенолучах распространились в мире за несколько лет. Несмотря на этот очевидный успех, рентгеновские изображения были ограничены четырьмя проблемами, до тех пор, пока в 70-х годах не появилась ЦОС и соответствующая техника. Во-первых, перекрывающиеся структуры в теле могли закрываться одна другой. Например, часть сердца могла быть не увидена за ребрами. Во-вторых, не всегда можно различить похожие ткани. Например, можно отличить кости от мягких тканей, но не опухоль от легких. В-третьих, рентгеновские лучи показывают *анатомию*, структуру тела, но не *физиологию*, процессы в организме. Рентгеновское изображение живого человека точно такое же, как изображение мертвого! В-четвертых, рентгеновское воздействие может быть причиной возникновения рака и требует умеренного использования.

Проблема перекрывающих структур была решена в 1971 году с появлением первого компьютерного томографического сканера (прежде называемого компьютерной осевой томографией, или КОТ сканером). Компьютерная томография (КТ) – это классический пример ЦОС. Анализируются рентгеновские лучи, пропущенные через часть тела пациента с нескольких направлений. Вместо простого формирования изображения по принятым рентгеновским лучам сигналы преобразуются в цифровые данные и записываются в компьютер. Эта информация используется, чтобы *вычислить* изображения, которые представляют собой срезы исследуемой части. Эти изображения показывают гораздо больше деталей и позволяют проводить существенно лучшее диагностирование и лечение. Революция, которую произвела компьютерная томография, была так же велика, как внедрение самих рентгеновских изображений. В течение только нескольких лет каждый крупный госпиталь в мире получил доступ к сканеру КТ.

В 1979 году двое из основоположников компьютерной томографии, Годфри Хаунсфилд (Godfrey N. Hounsfield) и Алан Кормак (Allan M. Cormack), были удостоены Нобелевской премии в области медицины. Благодаря ЦОС!

Оставшиеся три проблемы рентгеновских лучей были решены с использованием другой, нежели рентгеновские лучи, проникающей энергии, такой, как радио- и звуковые волны. ЦОС играет здесь ключевую роль. Например, магниторезонансное изображение (МРИ) использует магнитные поля в сочетании с радиоволнами для исследования внутренностей человеческого тела. Соответствующая установка силы и частоты полей вызывает резонанс атомных ядер между состояниями энергии квантов в локализованных областях тела. В результате резонанса возникает вторичное радиоизлучение, которое обнаруживается антенной, размещаемой вблизи тела. Мощность и другие характеристики этого сигнала дают информацию о локализованной области в состоянии резонанса. Регулирование магнитного поля позволяет сканировать область резонанса через тело, фиксируя внутренние структуры. Эта информация обычно представляется в виде изображения, как и в компьютерной томографии. Кроме способности точного определения разных типов мягких тканей, МРИ может сообщать информацию о физиологии, например, кровотоке через артерии. Метод МРИ полностью реализован на алгоритмах ЦОС.

5.1.16. Космос

Иногда необходимо улучшить качество плохого изображения. Эта задача часто встречается в случаях с изображениями, взятыми со спутников и космических исследовательских кораблей. Никто не согласится быть репортером, посланным на Марс только для того, чтобы нажать кнопку на камере! ЦОС может улучшить качество изображений, полученных в экстремальных неблагоприятных условиях, несколькими способами: регулирование яркости и контрастности, выделение контуров, подавление шума, регулировка фокуса, коррекция нерезкости, вызванной движением, и т.д. Изображения, которые имеют пространственные искажения, например, когда плоское изображение снимается со сферических планет, могут быть преобразованы в корректное представление. Несколько изображений могут быть соединены в единую базу данных, позволяющую представлять информацию особенным способом. Например, последовательность видеокадров, представляющих полет над поверхностью удаленной планеты.

5.1.17. Коммерческие видеопродукты

Большое количество информации, содержащееся в изображениях, является проблемой для систем, продаваемых массовым потребителям. Продаваемые системы должны быть *дешевы*, при том, что они включают в себя большое количество памяти и устройства с высокой скоростью передачи. Единственным решением этой проблемы является сжатие изображения. Как и речевые сигналы, изображение содержит большое количество избыточной информации и может быть обработано специальными алгоритмами, которые понижают количество разрядов, требуемых для его представления. Телевидение и кинофильмы особенно предрасположены к сжатию, поскольку большая часть изображения остается неизменной от кадра к кадру. Коммерческие видеопродукты, которые используют эту технологию, включают в себя видеотелефоны, компьютерные программы для демонстрации кинофильмов, цифровое телевидение.

5.2. Применение ЦОС при записи и воспроизведении звука

5.2.1. Цифровое микширование

Цифровое микширование звука – это отличный пример успешного применения ЦОС для улучшения качества записи и воспроизведения звука, а также повышения функциональности системы [1].

Микширование звука применяется в профессиональных и полупрофессиональных аудиоприложениях, например, для студийной звукозаписи, радиовещания, при усилении звука и в системах публичных выступлений. Микшер позволяет регулировать, смешивать и выводить на экран характеристики многоканальных аудиосигналов от различных источников для подстройки их под требования определенных приложений.

Цифровая система микширования состоит из средств аудиовыравнивания, аудиомикширования и обработки после микширования (рис. 5.3). Цифровой аудиоэквалайзер – это набор цифровых фильтров с регулируемыми характеристиками, т.е. с возможностью выполнять действия над различными частями полос входного спектра аудиосистемы для достижения желаемого звука (например, усиливать или обрезать определенные тона) подобно настройке высоких и низких частот. Затем выровненные звуковые сигналы смешиваются с помощью матрицы смешивания (позволяет смешивать любой произвольный звуковой вход с любым выходом). После смешивания обработка сигнала может продолжаться путем реверберации и выравнивания.

Система микширования обладает такими интерактивными средствами контроля для управления параметрами микшера, как значения операторов микшера (регуляторов уровня сигнала) и параметры контроля эквалайзера (частота, добротность и коэффициент усиления фильтра) в реальном времени. Одна из сложных проблем цифрового микширования звука – это достижение пользовательского контроля над параметрами микшера при относительно высокой скорости обработки данных без заметных искажений. Каждый раз, когда пользователь передвигает рычаги управления, параметры микшера должны изменяться, чтобы соответствовать новым требованиям. Такая регулировка может привести к заметным искажениям, что недопустимо в профессиональных системах смешивания звука. Тщательное следование алгоритму смешивания является главным условием достижения профессионального уровня звучания.

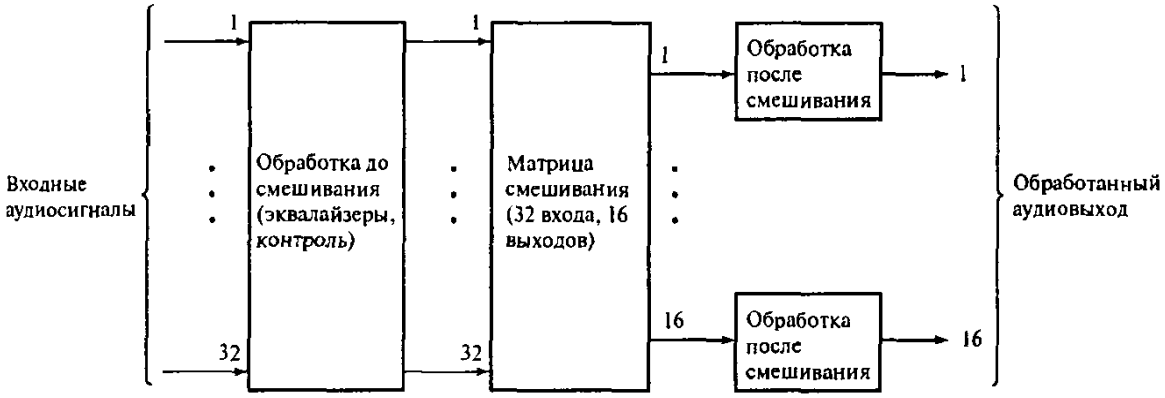


Рис. 5.3. Упрощенная блок-схема цифровой стереосистемы микширования на 32 входа и 16 выходов

На рис. 5.4 показан обычный микшер. Его характерные особенности:

- 8 моновходов (8 микрофонов или 8 линий);
- 2 пары стереоканалов (левый и правый) сигнала;
- главная кодовая шина для централизованного контроля, например, выбора области памяти, усиления по мощности и обработки сигнала.

Рис. 5.4. Средства управления на обычном 8-канальном микшере

В цифровом микшере, изображенном на рис. 5.4, используется современный процессор ЦОС, позволяющий реализовывать новейшие алгоритмы, необходимые при смешивании цифровых сигналов (например, выравнивание, стробирование шума, динамический контроль) и выполнении операций после смешивания.

5.2.2. Синтез и распознавание речи

5.2.2.1. Синтез речи. В прошлом синтезированная речь воспринималась как звучание механического голоса. Однако прогресс в области полупроводниковых технологий и ЦОС сделали экономически возможным достижение такого качества речи, что ее нельзя отличить от человеческой.

Примером успешного коммерческого продукта с речевым выходом является система Speak and Spell, с которой, вероятно, знакомы многие читатели. Это электронное пособие для обучения детей, в котором используется метод линейного кодирования с предсказанием (*linear predictive coding* – LPC), где настоящая человеческая речь, которую нужно воспроизвести, моделируется как отклик переменного во времени цифрового фильтра на периодический или случайный сигнал возбуждения (рис. 5.5).

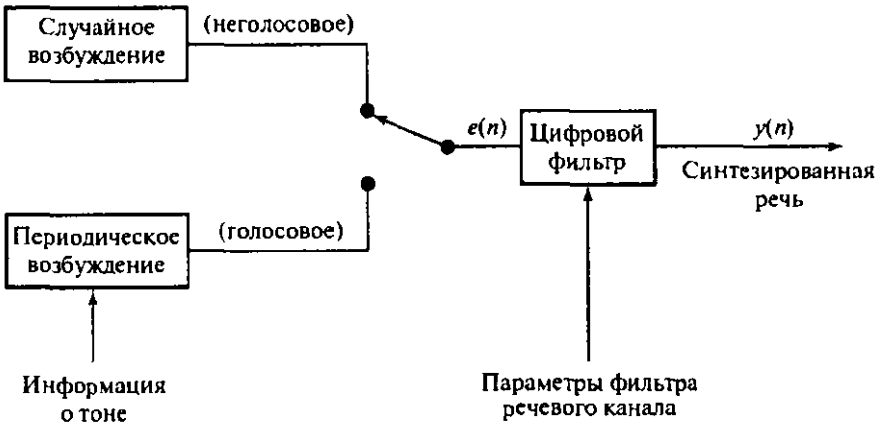


Рис. 5.5. Линейное кодирование речи с предсказанием

Периодическое возбуждение используется для генерации голосовых звуков (например, гласных) и представляет поток воздуха через вибрирующие голосовые связки. Случайное возбуждение используется для неголосовых звуков (таких, как С, Ш) и представляет шум, создаваемый при проталкивании воздуха через преграды в речевом канале. Человеческая речь содержит очень много лишней информации. Кодер оставляет только важную информацию, необходимую для сохранения таких характеристик

речи, как интонация, акцент и диалект, позволяя держать целые минуты звука высокого качества в памяти среднего размера.

В пособии Speak and Spell используется микросхема синтезатора речи TMS5100, который объединяет все элементы модели LPC (цифровой фильтр и источники возбуждения), а также декодер и 8-битовый цифроаналоговый преобразователь (ЦАП). Микросхема синтезатора действует в тесной связи с 4-битовым микропроцессором и двумя 128-килобитовыми ПЗУ, которые вместе вмещают словарь из приблизительно 300 слов и фраз (рис. 5.6). Информация о речи хранится в ПЗУ в форме кадров (содержащих 25 мс речи), каждый кадр характеризуется набором из 10 или 12 параметров LPC. Параметры кадра сообщаются синтезатору каждые 25 мс и используются для обновления коэффициентов цифрового фильтра и выбора источника возбуждения и его уровня энергии. Выход цифрового фильтра преобразуется в аналоговый и подается на громкоговоритель для создания требуемого звука с определенным тоном, амплитудой и гармоническим содержанием. Для сглаживания переходов в речевом спектре каждые 3 мс синтезатор обновляет параметры LPC, проводя интерполяцию между параметрами предыдущего и следующего кадра.

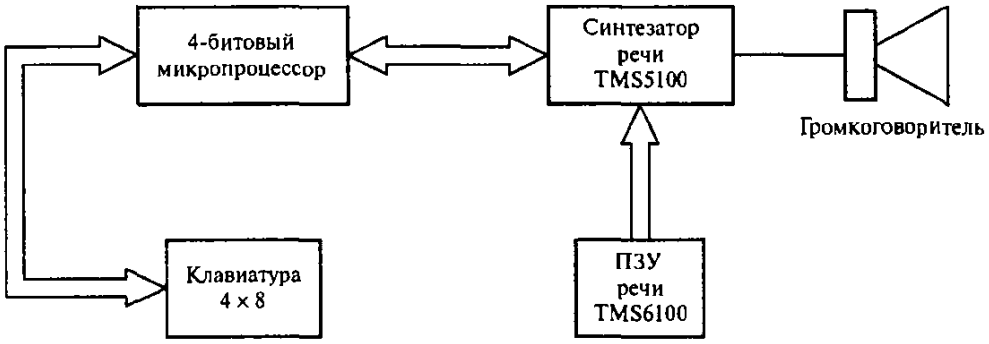


Рис. 5.6. Структура пособия Speak and Spell

В первом режиме операции ребенка просят сказать слово по буквам. Ребенок вводит слово по одной букве с помощью клавиатуры. Если написание верно, то при нажатии клавиши <Enter> программа отвечает «Правильно» или «Верно». Если слово написано неверно, программа говорит «Неправильно, попробуй еще раз». Если же и следующая попытка неверна, она указывает: «Это неправильно» и добавляет: «Правильно это слово пишется так: ...».

5.2.2.2. Распознавание речи

Под распознаванием голоса подразумевается, что информация вводится в компьютер с помощью человеческого голоса, а компьютер слуша-

ет и распознает человеческую речь. Распознавание голоса все еще активно изучается, так как поставленные задачи намного сложнее, чем те, которые возникают при синтезе речи. Поэтому успешные коммерческие системы распознавания речи немногочисленны и встречаются редко. Самыми удачными из них оказались настроенные на диктора системы распознавания изолированных слов. Такие системы работают в одном из двух режимов. В режиме обучения пользователь обучает систему распознавать его или ее голос, произнося каждое слово, подлежащее распознанию, в микрофон. Система оцифровывает и создает эталон каждого слова, сохраняя его в памяти. В режиме распознавания каждое произнесенное слово снова оцифровывается и его эталон сравнивается с эталонами из памяти. Если есть соответствие, значит, слово распознано и система сообщает об этом пользователю или выполняет какое-то действие. Работа таких систем зависит от того, делает ли говорящий достаточно длинные паузы перед каждым словом, существует ли некий шумовой фон и от того, насколько четко и ясно произносятся слова. Двумя самыми важными операциями ЦОС при распознавании являются извлечение параметров, когда из сказанного слова получаются отдельные образцы и создаются эталоны, и подбор по образцу, когда эталоны сравниваются с записями, которые хранятся в памяти (рис. 5.7).

Рис. 5.7. Блок-схема системы распознавания речи

Для большинства людей разговор – это самая естественная форма общения, ведь говорить намного быстрее, чем писать или печатать. Поэтому сейчас в офисах есть системы, которые позволяют управлять программами не нажатием клавиш, а произнесением команд. Разрабатываются системы, которые позволят создавать и отправлять с помощью голоса обычные офисные документы, такие как письма и служебные записки. Системы распознавания слов встречаются в товарах широкого потребления, таких как системы набора телефонного номера с речевым управлением, и в управляемых голосом предметах домашнего обихода для инвали-

дов с ограниченной способностью передвижения. Это делает таких людей более независимыми, позволяет самостоятельно выполнять некоторые простые действия, например, включать или выключать свет, радио или телевизор.

Конечно, существует еще множество возможных применений распознавания речи. Впрочем, дальнейшее развитие в этой области зависит, главным образом, от технологий искусственного интеллекта, поскольку машины должны не только распознавать, но и понимать человеческую речь.

5.2.3. Цифровая аудиосистема воспроизведения компакт-дисков

Многим читателям знакомы неприятные звуки, доносящиеся из проигрывателя, если пластинка повреждена либо на ней есть царапины, грязь или отпечатки пальцев. Компакт-диск – это современная аудиосистема, у которой уже нет недостатков долгоиграющих пластинок. Основные характеристики пластинок и компакт-дисков сравниваются в табл. 5.1.

На компакт-дисках информация записывается в цифровой форме в виде спиральной дорожки, которая состоит из последовательности впадин (рис. 5.8). Каждый бит, записанный на компакт-диске, занимает площадь всего лишь 1 мкм^2, т.е. 10^6 бит на квадратный миллиметр, что обеспечивает очень высокую плотность хранения информации.

Таблица 5.1

Сравнение характеристик долгоиграющей пластинки и компакт-диска

Характеристика	Долгоиграющая пластинка	Компакт-диск
Частотная характеристика	От 30 Гц до 20 кГц (± 3 дБ)	От 20 Гц до 20 кГц (от 0,5 до –1 дБ)
Динамический диапазон	70 дБ (при 1 кГц)	>90 дБ
Соотношение сигнал-шум	60 дБ	>90 дБ
Гармоническое искажение	1 – 2 %	0,004 %
Разделение стереоканалов	25 – 30 дБ	>90 дБ
Коэффициент детонации	0,03 %	Не регистрируется
Действие пыли, царапин и отпечатков пальцев	Вызывает шум	Приводит к исправимым или маскируемым ошибкам
Долговечность носителей	Высокочастотная характеристика при проигрывании ухудшается	Почти вечные
Время жизни иглы	500 – 600 ч	Почти бесконечно
Длительность звучания	40 – 45 мин (обе стороны)	50 – 75 мин (или больше)

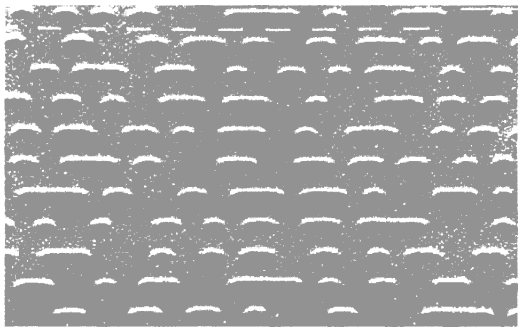


Рис. 5.8. Выжженные лазером впадины на компакт-диске. Ширина каждой впадины – 0,5 мкм, длина – 0,8 – 3,5 мкм, а глубина – 0,11 мкм. Расстояние между дорожками составляет 1,6 мкм.

Упрощенная блок-схема обработки звукового сигнала на компакт-диске во время записи изображена на рис. 5.9. Аналоговый звуковой сигнал в каждом стереоканале дискретизуется с частотой 44,1 кГц и оцифровывается. Каждая выборка представляется 16-битовым кодом, множество всех кодов образует динамический диапазон 90 дБ. Таким образом, в каждый момент дискретизации получается 32 бит, по 16 бит с левого и правого звуковых каналов. Цифровые выборки кодируются с помощью двухуровневого кода Рида – Соломона, что дает возможность обнаруживать и исправлять или скрывать ошибки при воспроизведении звукового сигнала. Для контроля и отображения информации для слушателей добавляются дополнительные биты. Затем полученные потоки битов модулируются для преобразования в форму, более удобную для хранения на диске. С помощью EFM-модуляции (преобразования 8-разрядного кода в 14-разрядный) каждый байт потока данных преобразуется в 14-разрядный код. Полученный поток битов в канале после дальнейшей обработки используется для управления лазерным пучком, который записывает цифровую информацию на светочувствительный слой вращающегося стеклянного диска. С помощью процесса проявки создается узор из впадин на мастер-диске, с которого впоследствии делаются пользовательские компакт-диски.

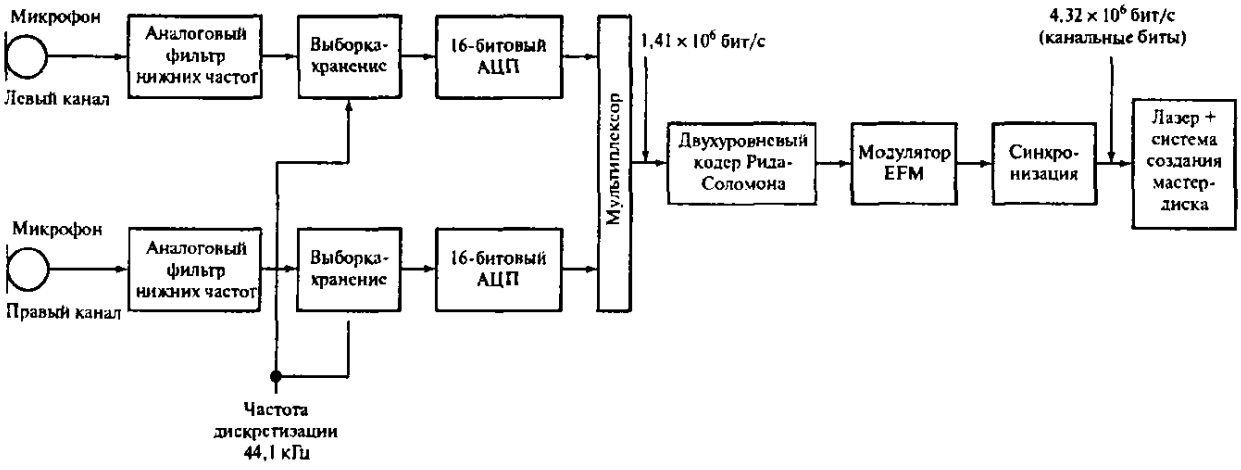


Рис. 5.9. Упрощенная блок-схема обработки аудиосигнала и записи его на компакт-диск

Во время воспроизведения записи на проигрывателе компакт-дисков дорожки на диске оптически сканируются с постоянной скоростью 1,2 м/с, в то время как диск вращается со скоростью от 8 об./с до приблизительно 3,5 об./с, и таким образом считывается записанная информация (рис. 5.10). Прежде всего, цифровой сигнал с диска демодулируется, детектируется, и по возможности в нем исправляются ошибки. Ошибки могут возникнуть вследствие производственных дефектов, повреждений, наличия отпечатков пальцев или пыли на поверхности диска. Если ошибки неисправимы, они либо маскируются посредством замены неправильного элемента выборки новым, полученным путем интерполяции между соседними элементами, либо, если ошибка обнаружена более чем в одном элементе, обнуляются (приглушаются).

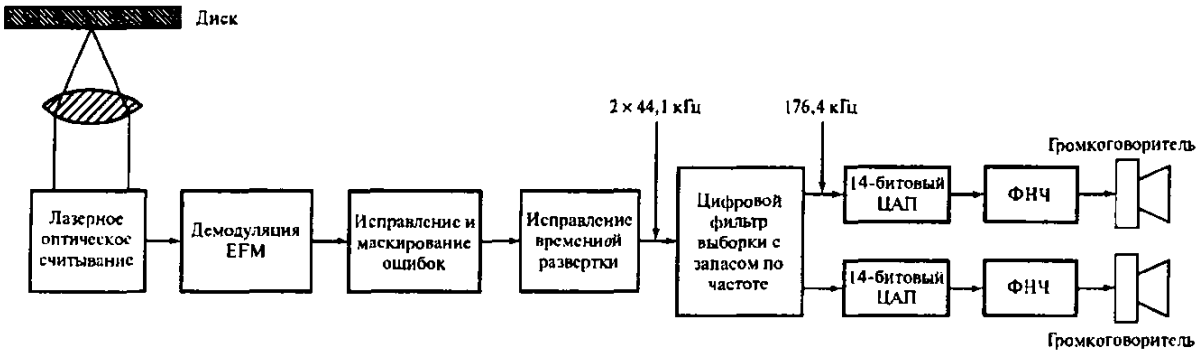


Рис. 5.10. Воспроизведение звукового сигнала с компакт-диска

В результате полученные данные представляют собой ряды из 16-битовых слов, каждое из которых является элементом выборки звукового сигнала. Эти элементы могут непосредственно направляться в 16-битовый цифроаналоговый преобразователь, а затем в аналоговый фильтр нижних частот. Однако для этого понадобились бы аналоговые фильтры с весьма строгими спецификациями. В частности, уровень частот, лежащих за 20 кГц, должен быть хотя бы на 50 дБ меньше максимального, а фильтр должен иметь линейную фазовую характеристику во избежание ухудшения звукового сигнала. Для этого цифровые сигналы предварительно пропускаются через цифровой фильтр, работающий с частотой, в четыре раза большей, чем частота дискретизации 44,1 кГц. Эффект повышения частоты дискретизации помогает сглаживать выход цифроаналогового преобразователя, упрощает требования к аналоговым фильтрам. Он также позволяет достичь 16-битового отношения сигнал – шум с помощью 14-битового цифроаналогового преобразователя. Использование цифрового

фильтра также способствует получению линейной фазовой характеристики, снижает вероятность взаимной модуляции и дает фильтр с характеристикой, которая изменяется с тактовой частотой, делая его нечувствительным к скорости вращения диска. На рис. 5.11 изображена печатная плата декодирующей схемы первого поколения проигрывателей компакт-дисков фирмы Philips. Хорошо различимы основные микросхемы.

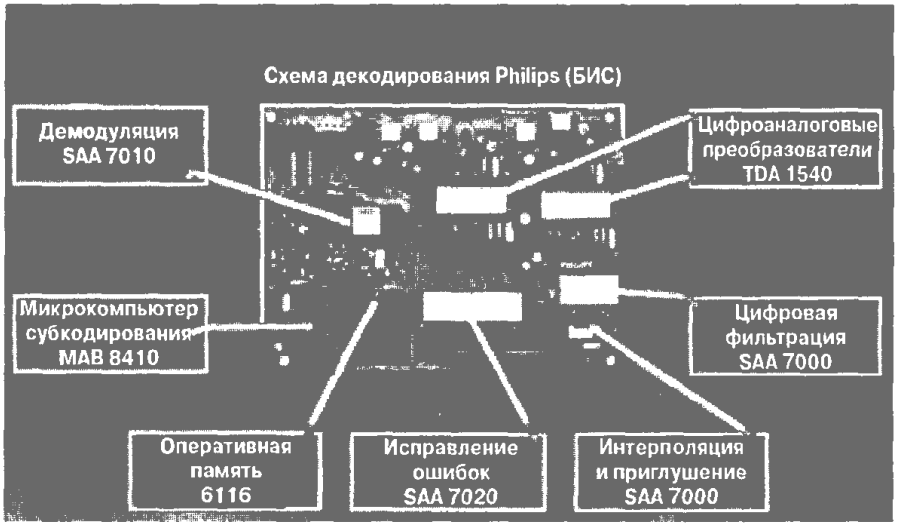


Рис. 5.11. Печатная плата декодирующей схемы проигрывателя компакт-дисков фирмы Philips)

Кроме работы с компакт-дисками, цифровая обработка сигналов также играет большую роль и в других областях цифровой аудиотехники. Она широко используется как в товарах массового потребления, так и для профессиональной работы, например, на студиях звукозаписи, при передаче телепрограмм телерадиокомпаниями, при создании фильмов и в музыкальной индустрии. Ниже перечислены специальные области применения ЦОС в цифровой звукозаписи и звуковоспроизведении:

- использование передовых технологий ЦОС для кодирования, выявления, исправления или маскировки ошибок при воспроизведении, для того чтобы ограничения носителя информации (магнитного или оптического) не влияли на уровень качества, которого можно достичь при записи и воспроизведении. Таким образом, одна и та же запись будет звучать одинаково на носителях различного качества, с одинаковой частотой появления ошибок;
- расширение области слышимости и обогащение звука. Например, простые схемы цифровых фильтров применяются для создания эха, естественной реверберации и эффектов хорового звучания;

- синтез звуков, напоминающих звучание музыкальных инструментов, и даже таких, которые не может издать ни один инструмент;
- на телевидении, в мультипликации и кинофильмах при создании и использовании различных звуковых эффектов, таких как выстрелы, шаги, аплодисменты, звуки машин, удары кулаком, чтобы усилить чувство реальности или сделать сцену более правдоподобной;
- улучшение качества архивных записей или стенограмм.

5.3. Применение ЦОС в телекоммуникации

5.3.1. Цифровые сотовые мобильные телефоны

Сегодня мобильная связь – это одна из самых быстрорастущих отраслей в мире, мобильный телефон стал незаменимым средством постоянной связи в информационном обществе. По оценкам специалистов, в течение нескольких лет количество людей, пользующихся мобильными телефонами, превысит число тех, кто пользуется стационарными телефонными линиями. В некоторых странах, например, в Финляндии, это уже произошло. ЦОС – это одна из ключевых технологий, делающих возможной революцию в области мобильной связи. ЦОС интенсивно используется для обработки сигналов и информации на базовых радиостанциях и в самих мобильных телефонах (например, для кодирования речи, многоканального выравнивания, измерения силы сигнала, голосовых сообщений, защиты от ошибок, модуляции и демодуляции). Широко доступны ЦОС-микросхемы, предназначенные для беспроводной связи, благодаря чему в области мобильной связи появилась возможность выпускать доступные высококачественные товары широкого потребления.

В современных системах мобильной связи используется принцип цифровой сотовой радиосвязи, хотя в первом поколении мобильных телефонов, с которых началось развитие сотовой радиосвязи, использовались аналоговые методы обработки и передачи голосовых сигналов. К самым удачным системам аналоговой мобильной связи можно отнести системы Advanced Mobile Phone System (AMPS), которая использовалась в Северной Америке, Nordic Mobile Telephone (NMT), разработанную совместно Данией, Финляндией, Норвегией и Швецией, и Total Access Communication Systems (TACS), использовавшуюся в Великобритании. Первые системы были несовместимы друг с другом, поскольку разрабатывались, в основном, для отдельно взятой страны, и не могли в полной мере удовлетворить растущую потребность в мобильной связи. Современные цифровые сотовые сети предлагают более широкие возможности, расширена об-

ласть покрытия, возросли качество, безопасность и надежность. Проиллюстрируем основные принципы мобильной связи на примере системы GSM (*Global System for Mobile communication*).

Система GSM была первой полностью цифровой системой радиотелефонной связи и сейчас считается фактически мировым стандартом цифровой мобильной связи. GSM была разработана в 1992 году, а к концу 1998-го у нее было более 130 миллионов пользователей в более чем ста странах мира. По прогнозам к 2005 году количество абонентов мобильной связи буде превышать 1 миллиард. Значительный процент из них будут составлять пользователи GSM. Некоторые характеристики системы GSM 900, которой пользуются в Европе, приведены в табл. 5.2.

Таблица 5.2

Основные характеристики системы GSM 900

Параметры сотовой системы мобильной связи	Показатели системы GSM 900
Частота: мобильный передатчик/приемник базовой станции мобильный приемник/передатчик базовой станции	 890 – 915 МГц 935 – 960 МГц
Дуплексное разделение по частоте	45 МГц
Частотный интервал канала	200 кГц
Количество каналов	124
Скорость передачи речи	13 Кбит/с (половинная скорость 5,6 Кбит/с)
Общая скорость передачи информации	270 Кбит/с
Способ передачи сигнала	Узкополосный CDMA
Характерный размер ячейки	300 м – 35 км

5.3.1.1. Архитектура сотовой телефонной сети

Мобильная сотовая телефонная сеть – это двусторонняя телефонная система, которая позволяет отправлять и принимать информацию (речь, данные и сообщения) с помощью мобильного телефона через радиоканалы. В сотовых радиосистемах область покрытия делится на небольшие единицы, называемые *ячейками* (или *сотами*), каждая из которых обслуживается своей базовой радиостанцией. Доступная полоса частот делится на радиочастоты, или *каналы*, с каждой ячейкой связан набор радиочастот. Радиочастоты в пределах области покрытия используются многократно, чтобы эффективно использовать доступную полосу частот, как показано на рис. 5.12 [31]. Для минимизации внутриканальной интерференции между базовыми станциями, работающими на одних радиочастотах, ячейки с одинаковым набором частот располагаются как можно дальше друг от друга.

На практике размер ячейки зависит от интенсивности передвижения. В городах и густонаселенных территориях ячейки небольшие, они удовлетворяют требованиям интенсивного передвижения (часто используются микросоты с размерами, не превышающими 300 м). В сельской местности с низким уровнем передвижения могут использоваться ячейки размером до 35 км. Для таких больших ячеек необходима большая мощность передачи сигнала, при этом могут возникать дыры в области покрытия.

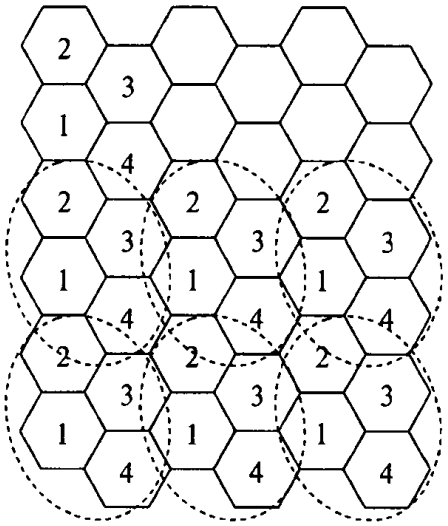


Рис. 5.12. Сотовый принцип, иллюстрирующий схему повторяющихся ячеек, возникающую из-за многократного использования частоты

Связь между мобильным телефоном и сетью осуществляется через базовую радиостанцию. Сотовая сеть объединяет базовые радиостанции в единую радиотелефонную систему по всей области покрытия. Каждая базовая станция имеет определенное количество голосовых каналов и связана с мобильным коммутационным центром (*mobile switching center* – MSC); рис. 5.13 и [31]. Некоторые коммутационные центры непосредственно связаны с коммутируемой телефонной системой (PSTN) и таким образом связывают сотовую сеть с PSTN. Базовая радиостанция состоит из контроллеров базовых станций (для управления радиоканалами), каждому из которых соответствует набор базовых станций-передатчиков. MSC содержит информацию о расположении мобильных телефонов и обеспечивает их мобильность.

Двумя ключевыми особенностями мобильной сотовой радиосети, которые позволяют пользователям звонить и принимать звонки на большой территории, является ее способность определять местонахождение мобильного телефона и передавать линии радиосвязи с телефоном от одной базовой станции к другой. В сети сохраняется информация о местоположении каждого мобильного телефона, зарегистрированного в ней, так что входящие и исходящие звонки могут направляться к мобильному телефону или от него, где бы он ни находился в пределах области покрытия. Это крайне важно, поскольку местонахождение пользователя мобильного телефона не фиксированно. При каждом включении мобильного телефона он регистрируется в сети, и это позволяет обновлять информацию о его местонахождении. У каждого мобильного телефона есть, по меньшей ме-

ре, два регистра местоположения в сети – домашний (HLR) и регистр местоположения посетителей (VLR). HLR размещается в собственной сети пользователя и содержит информацию о его профиле, в частности, перечень услуг, которыми может пользоваться владелец данного телефона. В VLR записывается информация о мобильном телефоне, когда он используется за пределами собственной сети. При регистрации мобильного телефона в сети через ближайшую базовую станцию в сеть передается характерная только для этого телефона информация. Затем она используется для получения необходимых сведений о HLR данного телефона, чтобы он был опознан и получил доступ в сеть.

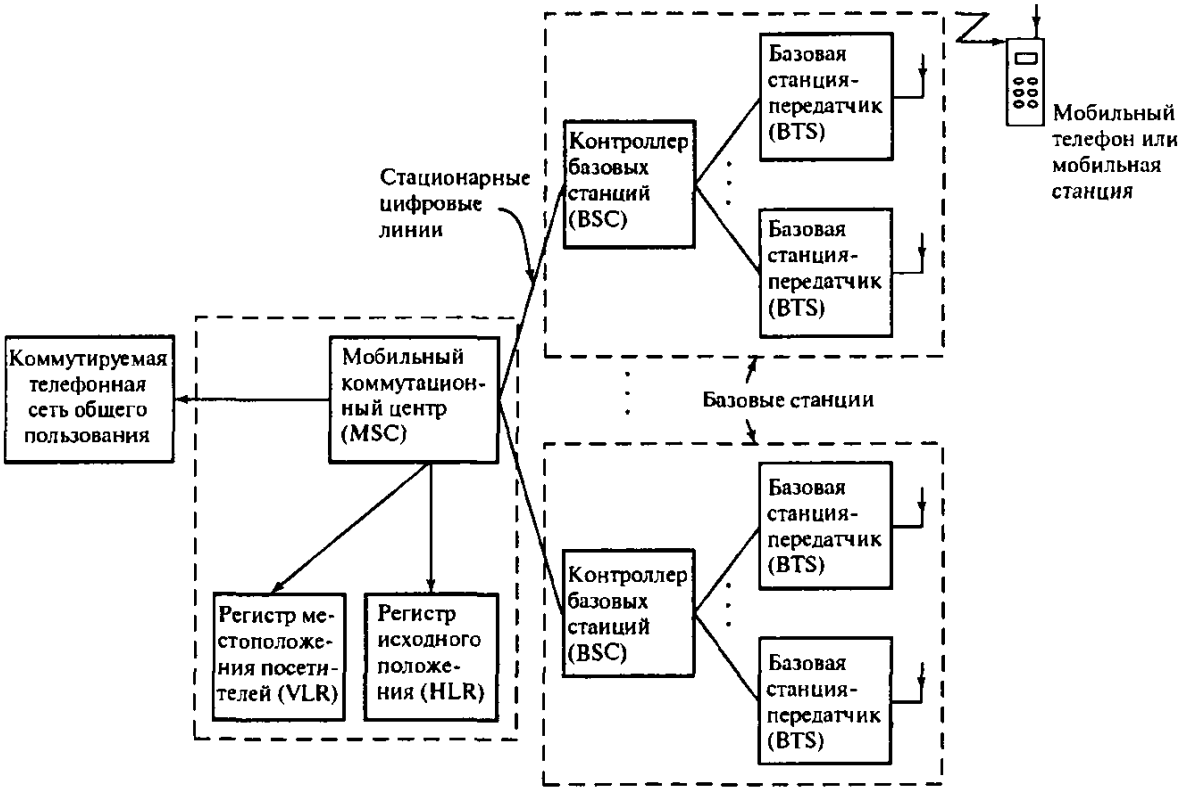


Рис. 5.13. Упрощенная блок-схема мобильной сотовой телефонной системы

Как указывалось выше, радиосвязь между мобильным телефоном и сетью осуществляется через базовую станцию. Все базовые станции регулярно передают закодированную контрольную информацию. Мобильный телефон, не задействованный в разговоре, ловит контрольный канал ближайшей базовой станции. Перемещаясь по области покрытия, мобильный телефон при необходимости ловит новую базовую станцию, и информация о его местонахождении обновляется. Осуществляется постоянный контроль за силой сигнала от мобильного телефона, по которому разговаривают. Если сила сигнала оказывается ниже определенного порогового зна-

чения, то сеть автоматически переключает радиосвязь на другую базовую станцию, которая обеспечит лучший прием, если такая есть. Возможность перебрасывать радиосвязь от одной базовой станции к другой при плохом сигнале – это как раз то, что позволяет пользователям мобильных телефонов свободно перемещаться в пределах области покрытия, не прекращая делать и принимать звонки. Этот процесс передачи занимает несколько секунд, хотя перерыв в разговоре на 200 – 300 мс уже может быть заметен. Как знает большинство пользователей мобильных телефонов, бывает, что в некоторых отдаленных областях поблизости нет такой базовой станции, которая может обеспечить лучшую связь. Если сила сигнала слишком мала, то голосовой канал обрезается и пользователь не может делать или принимать звонки. Информация о силе сигнала отображается на экране мобильного телефона в том или ином виде (например, в виде нескольких полосок).

5.3.1.2. Элементы обработки сигнала

В современных сотовых системах радиотелефонной связи, таких как GSM, используются цифровые методы, и поэтому ЦОС – это естественный выбор для обработки и передачи информации. В мобильной связи ЦОС используется для кодирования речи, выравнивания сигналов после многолучевого распространения, измерения силы и качества сигнала, передачи голосовых сообщений, кодирования с исправлением ошибок, модуляции и демодуляции [31].

В системе GSM кодек речи основан на системе возбуждения регулярными импульсами с линейным кодированием с предсказанием (*Regular Pulse Excitation, Linear Predictive Coding* – RPE-LPC). В отличие от системы PSTN, в которой речь кодируется со скоростью 64 или 32 Кбит/с (адаптивная дифференциальная импульсно-кодовая модуляция), в мобильной радиотелефонной связи кодирование речи происходит с относительно невысокой скоростью 13 Кбит/с с целью эффективного использования спектра радиочастот. Алгоритм кодирования речи для системы GSM реализован в самых распространенных процессорах ЦОС (таких, как Motorola ЦОС56000, Texas Instrument TMS320C50). Скорость обработки данных кодека – 13 Кбит/с, причем данный кодек заменяет 13-разрядные линейные цифроаналоговый и аналого-цифровой преобразователи.

Если мобильный телефон оказывается в неподходящей среде, часто возникают проблемы, связанные с многолучевым распространением. На частотах, которые используются для сотовой радиотелефонной связи, передаваемые сигналы часто могут отражаться от высотных зданий и т.п. Отраженный сигнал задерживается и поступает в принимающее устройст-

во позже, чем неотраженный, поскольку преодолевает большее расстояние. Это приводит к флуктуациям амплитуды и фазы суммарного сигнала в приемнике, зависящим от природы многолучевого распространения и перемещения мобильного телефона. Эффект многолучевого распространения можно снизить, воспользовавшись цифровым выравниванием в приемнике. Для этого через регулярные промежутки передается известная последовательность сигналов длиной 26 бит. По окончании приема эквалайзер, используя эту опорную последовательность, определяет коэффициенты цифрового фильтра, оценивает характеристики полученного радиосигнала и таким образом устраняет эффект многолучевого распространения из полученных данных. Если известна передаточная функция, приемник может определить наиболее вероятную последовательность передаваемых битов и демодулировать сигнал. Алгоритм выравнивания GSM также реализован во многих процессорах ЦОС.

Кроме кодирования речи и выравнивания эффектов многолучевого распространения, методы ЦОС применяются также для цифровой модуляции, в частности, для обработки при нескольких скоростях. Также ЦОС дает возможность измерять силу полученного сигнала, чтобы переключать связь, и регулировать через базовую станцию уровень выходящей мощности мобильного телефона. В сети GSM мобильный телефон следит за сигналами от окружающих базовых станций. В системах сотовой радиосвязи всегда существует проблема внутриканальной интерференции между ячейками с одинаковым набором частот, которая зависит от передаваемой мощности. В системе GSM передаваемую мощность можно контролировать автоматически, снижая таким образом уровень внутриканальной интерференции и увеличивая время работы телефона без подзарядки. Решение об изменении передаваемой мощности принимается на основе уровня и качества принимаемого сигнала. Для того чтобы можно было оценить величину этих параметров, при анализе полученного сигнала применяются методы ЦОС.

При мобильной радиосвязи возникает много ошибок, например, обусловленных случайной интерференцией или затуханием. В системе GSM для уменьшения влияния этих ошибок используется метод сверточного кодирования.

5.3.2. Приставка для приема цифрового телевизионного сигнала

Цифровое телевидение возникло не так давно и много обещает потребителям – интерактивность, больший выбор, лучшее качество изображения и звука. Интерактивность дает абонентам возможность играть в игры, иметь доступ в сеть Internet, делать покупки, мгновенно выполнять по-

вторное воспроизведение и т.д. Телевидение снова заявляет о себе как о неотъемлемой части информационного общества.

В цифровом телевидении цифровая информация (видео, аудио, текст) может передаваться на домашние телевизоры через спутник (с помощью спутника и существующих спутниковых антенн), по кабелю (кабельное телевидение) и по наземным линиям связи (с помощью существующих телевизионных антенн и передатчиков). Почти все телевизоры в наших домах могут принимать только аналоговый сигнал, поэтому для приема цифрового телевидения им нужен цифровой декодер (цифровая приставка) (рис. 5.14). Цифровая приставка преобразует цифровой сигнал в подходящий для приема аналоговым телевизором вид. Новейшие телевизоры оснащены встроенным декодером.

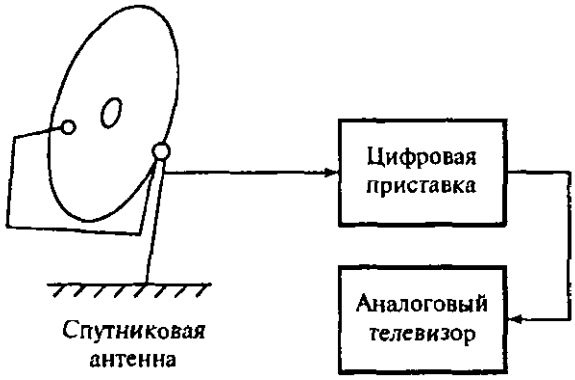


Рис. 5.14. Использование цифровой приставки для приема сигнала цифрового телевидения

В цифровом телевидении ЦОС играет ключевую роль в обработке сигнала, кодировании и модуляции/демодуляции видео- и аудиосигналов от точки захвата сигнала до момента его появления на экране телевизора. Без ЦОС замечательное качество изображения и звука, которое мы сейчас воспринимаем как должное, было бы невозможным. Например, ЦОС лежит в основе алгоритмов кодирования MPEG, которые используются для сжатия видео- и аудиоинформации перед их передачей (для эффективного использования ширины полосы). В цифровой приставке для декодирования информации используется декодер MPEG. Главный элемент алгоритма MPEG – это дискретное косинус-преобразование.

5.3.3. Адаптивное телефонное эхоподавление

Эффект эха в системах связи возникает преимущественно тогда, когда на пути сигнала встречается неподходящее сопротивление. На рис. 5.15 показана упрощенная схема телефонной цепи для больших расстоя-

ний. Гибридная схема телефонного аппарата преобразовывает двухпроводную линию, идущую от абонента, в четырехпроводную, и предоставляет отдельный тракт для передачи сигнала в каждом направлении. Это делается в основном из соображений экономии, например, чтобы сделать возможной множественную или одновременную передачу нескольких звонков.

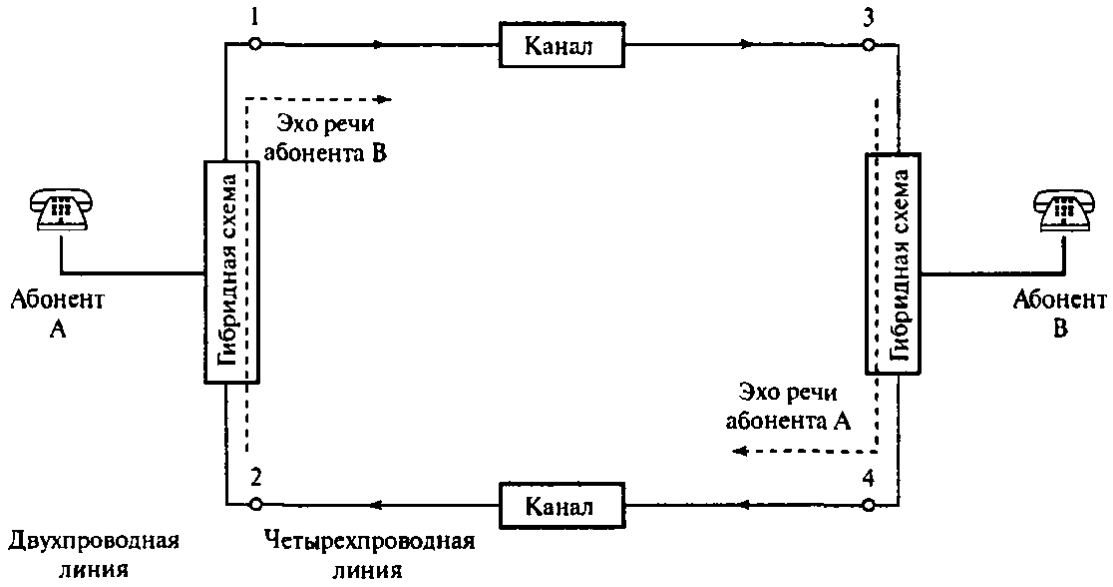


Рис. 5.15. Упрощенная схема телефонной цепи для больших расстояний

В идеале речевой сигнал от абонента А распространяется по верхнему тракту передачи к гибридной схеме справа, а оттуда к абоненту В, тогда как сигнал от пользователя В распространяется по нижней линии передачи к пользователю А. Гибридная схема на каждом конце обеспечивает подачу речевого сигнала от удаленного пользователя только на ее двухпроводный выход (на внешний выход сигнал не поступает). Однако из-за несоответствий в сопротивлении гибридная сеть пропускает некоторые входные сигналы на выход, которые слышны говорящему как эхо. Если телефонный звонок делается с большого расстояния (например, через геостационарный спутник), то эхо может задерживаться на 540 мс и вызывать помехи, раздражающие пользователей. Эти помехи будут возрастать с увеличением расстояния. Для решения этой проблемы в сети устанавливаются пары эхоподавителей, как показано на рис. 5.16.

На каждом конце системы связи (см. рис. 5.16) входной сигнал x_k подается на гибридную схему и на адаптивный фильтр (АФ). Компенсация

осуществляется с помощью оценки эха и вычитания его из возвращенного сигнала. Оценка эха осуществляется по формуле

$$y_k = \sum_{i=0}^{N-1} \omega_{k+1}(i) x_{k-i}, \tag{5.1}$$

где x_k – это входной сигнал от дальнего абонента, а $\omega_k(i)$, $i = 0,1,\ldots,$ $N - 1$ – оценка импульсной характеристики тракта передачи эха в дискретный момент времени k.

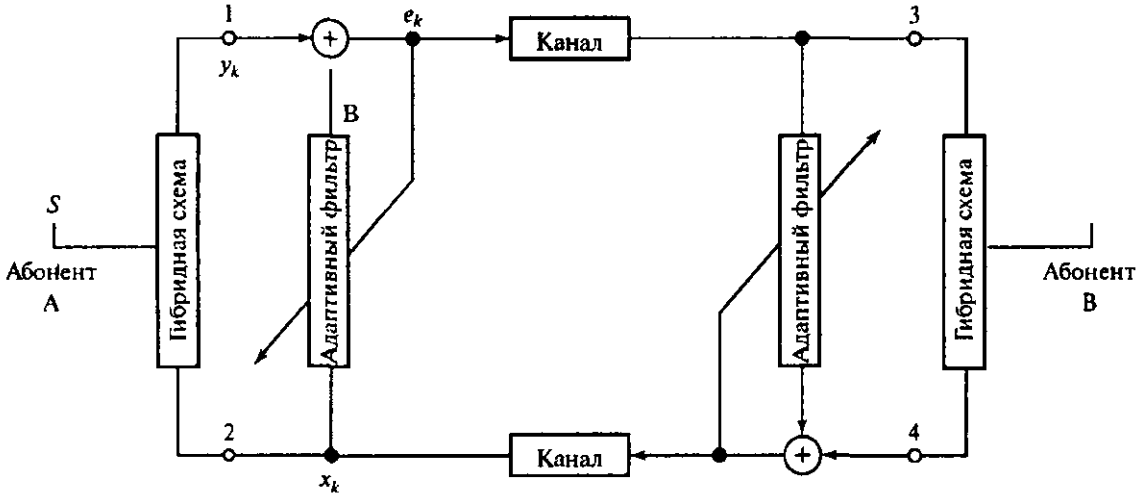


Рис. 5.16. Эхоподавление в междугородной телефонной связи

5.4. Применение ЦОС в биомедицине

Биомедицина – это важная и очень продуктивная область как для применения традиционной цифровой обработки сигнала, так и для разработки новых мощных алгоритмов ЦОС. Часто медицинские данные не отличаются хорошими характеристиками, и это бросает вызов профессионалам в области ЦОС, которые должны предложить новый способ работы с данными. В большинстве случаев медицинские данные попадают в диапазон аудиочастот. Поэтому методы ЦОС, появившиеся благодаря биомедицинским проблемам, нашли применение и в других областях, таких как системы звукозаписи, звуковоспроизведения, телекоммуникации, и наоборот.

К ряду многочисленных приложений ЦОС в биомедицине относятся усиление сигнала и/или извлечение из него характеристик, представляющих медицинский интерес. Необходимость в усилении сигнала возникает

из-за проблемы артефактов (ложных изображений) или загрязнения сигналов, что очень часто встречается в биомедицине. Артефакты могут возникать как из-за внешних источников (например, электросети или другого медицинского оборудования), так и из-за внутренних (движения головы и тела, мышечная и сердечная активность, движения глаз). Артефакты снижают медицинскую информативность сигналов и усложняют как ручной, так и автоматический их анализ, а в некоторых случаях даже делают его невозможным из-за сходства артефактов с сигналом, представляющим медицинский интерес.

Задачи усиления сигнала зачастую сопровождаются двумя неразрывно связанными проблемами: низкий уровень сигнала по сравнению с шумом и перекрывание спектров сигнала и шума. Следовательно, нужно приложить много усилий, чтобы минимизировать искажение сигнала, представляющего медицинский интерес. В следующих двух пунктах будут описаны два новых биомедицинских приложения ЦОС, которые подразумевают усиление сигнала и/или извлечение необходимых характеристик. Основные интересующие нас здесь биомедицинские сигналы имеют физиологическую природу, в частности, это сигналы электрокардиограмм (электрической активности сердца) и электроэнцефалограмм (электрической активности мозга).

5.4.1. Изучение ЭКГ плода

Электрокардиограмма (ЭКГ) плода показывает электрическую активность сердца ребенка, которая измеряется на поверхности тела. Частота сердцебиения плода (*Fetal Heart Rate* – FHR) определяется по расстоянию R-R на кардиограмме (рис. 5.17). Для оценки состояния плода во время родов обычно применяют визуальный анализ непрерывного изображения FHR вместе с сокращениями матки (маточной активности), которое называют *кардиотокограммой* (КТГ). Сложности, возникающие при интерпретации КТГ во время родов, могут привести к ненужному медицинскому вмешательству (например, кесаревому сечению или наложению щипцов), повреждению плода или неоказанию необходимой помощи [32].

Правильное применение комбинированного анализа ЭКГ плода и КТГ может значительно снизить процент ненужных медицинских вмешательств без каких-либо неблагоприятных последствий для младенцев. В частности, для этого был создан коммерческий аппарат для ЭКГ плода, *ST Analyser* (STAN, Neoventa AB, Швеция).

Упрощенная схема снятия ЭКГ плода изображена на рис. 5.18. ЭКГ снимается с черепного электрода, чтобы получить хорошее отношение сигнал – шум, полоса ограничивается диапазоном 0,05 – 100 Гц, дискретизация проводится с точностью до 12 бит с частотой 500 раз в секунду. Данные ЭКГ плода обрабатываются для снижения шума и извлечения необходимых характеристик, а затем анализируются, чтобы можно было измерить и вывести на экран изменения формы сигнала, связанные с КТГ.

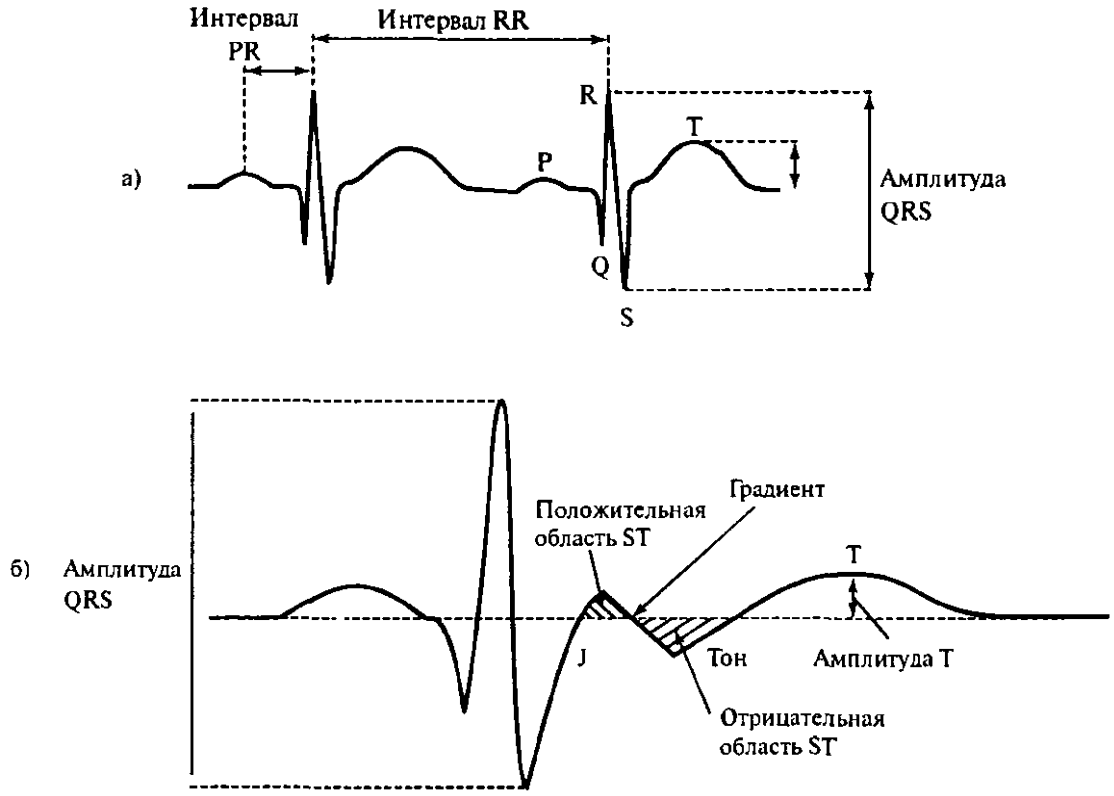


Рис. 5.17. ЭКГ плода, на которой показаны основные элементы, представляющие медицинский интерес

Важная характеристика ЭКГ плода – это форма сегмента *ST*. К числу существенных изменений формы сигнала, связанных со сжатием или растяжением, относятся постоянное увеличение амплитуды *T* сигнала, отрицательные значения волны *T* и подавление элементов *ST*. Изменения формы сигнала можно измерить как отношение амплитуды *T* к амплитуде *QRS*, которое называют *отношением T/QRS* (см. рис. 5.17). К числу остальных характеристик можно отнести область *S* изменения отрезков *R-R*, продолжительность *P*-волны и ширину комплекса *QRS*.

На рис. 5.18 показано снятие ЭКГ плода во время родов.

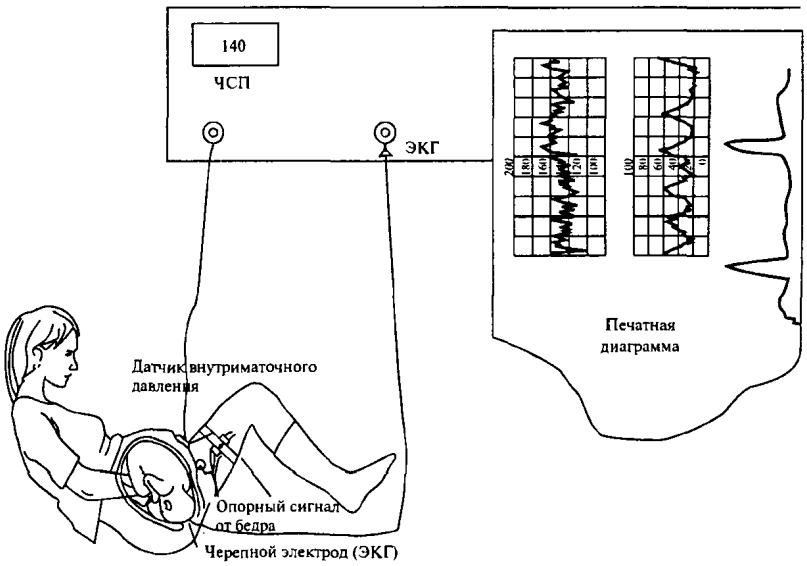


Рис. 5.18. Снятие ЭКГ плода во время родов

Черепная ЭКГ плода восприимчива к низкочастотному шуму и другим артефактам, дающим ложные изменения формы сигнала, например, шум от электросети, смещение базовой линии, мышечная активность или случайные шумы. Артефакты препятствуют извлечению необходимых характеристик и могут привести к неточностям анализа характеристик ЭКГ и формы сигнала.

Для снижения шума и извлечения основных характеристик из ЭКГ применяется множество методов обработки сигнала (включая подгонку кривой и цифровую фильтрацию при нескольких скоростях). Общая схема обработки ЭКГ плода показана на рис. 5.19. Первая главная задача – точно определить *R*-волну. Затем из необработанной ЭКГ удаляют смещение базовой линии, шум, вызванный мышечной активностью, и частоты питающей линии, чтобы получить форму сигнала, подходящую для надежного анализа. Пример оценки и устранения смещения базовой линии показан на рис. 5.20.

Очевидно, что исследование ЭКГ плода невозможно без ЦОС. Однако, хотя проблемы обработки сигнала, которые ранее препятствовали анализу ЭКГ плода, решены, исследование ЭКГ плода все еще ограничено в некоторых вопросах. В настоящее время анализ и интерпретация изменений в ЭКГ вместе с КТГ выполняются с помощью визуального наблюдения. Врачей нужно еще научить правильно снимать ЭКГ плода, но даже после соответствующего обучения существует вероятность того, что без постоянного наблюдения специалиста могут быть пропущены важные

особенности ЭКГ. Сейчас разрабатывается «сиделка» – разумная система контроля для непрерывного анализа и интерпретации изменений в ЭКГ и КТГ, которая будет помогать врачам [32].

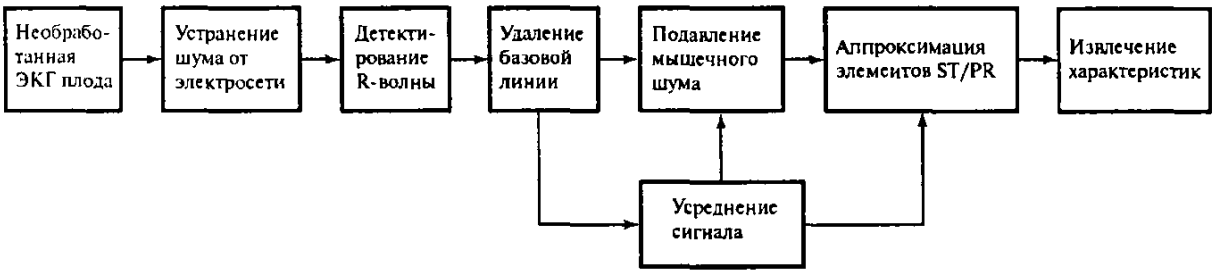


Рис. 5.19. Схема обработки сигнала ЭКГ плода

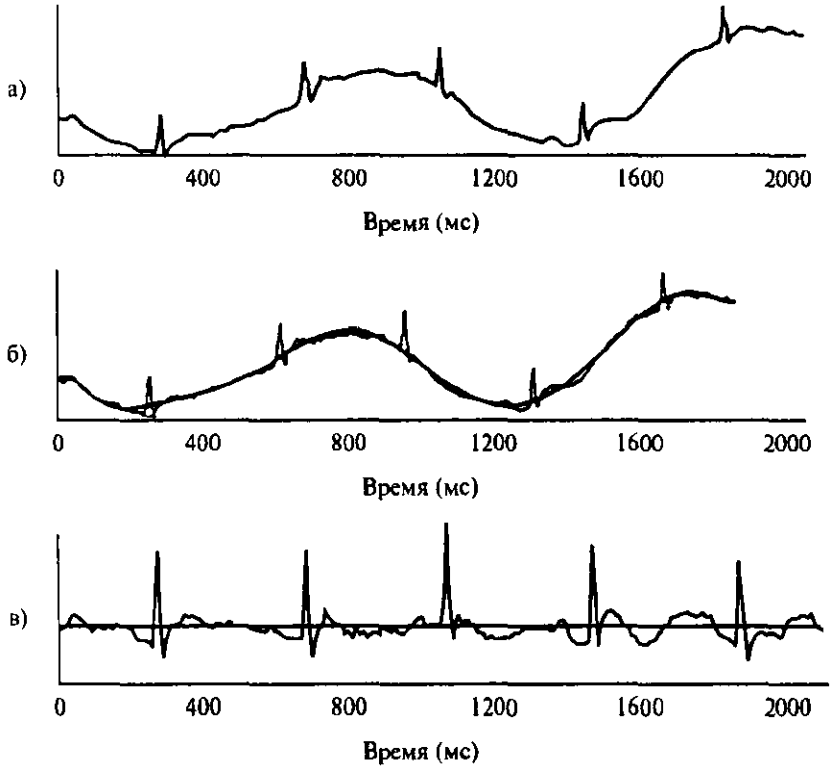


Рис. 5.20. Аппроксимация (панель *а*) и устранение (панель *б*) смещения базовой линии из необработанной ЭКГ плода

5.4.2. Анестезия с регулированием по замкнутому циклу на основе ЦОС

Огромное количество примеров применения ЦОС в медицине можно встретить в отделениях интенсивной терапии всех крупных больниц. Постоянно создаются новые современные методы работы с анестезией. Во время операции пациенты обычно находятся под наркозом, например, по-

сле введения анестезирующих препаратов внутривенно, так что они не чувствуют боли, и хирург может оперировать в спокойной обстановке. Задача анестезиолога – ввести правильную дозу препарата, чтобы как можно быстрее вызвать настолько глубокое состояние анестезии, насколько это нужно, и поддерживать это состояние без изменений на протяжении необходимого времени. Если ввести слишком большую дозу, это может привести к осложнениям или другим побочным эффектам, а при недостаточной дозе пациент может очнуться во время операции, что также может иметь долговременные физиологические последствия. В большинстве случаев глубину бессознательного состояния пациента оценивает опытный анестезиолог, наблюдая за клиническими признаками. При этом анестезиолог соответствующим образом изменяет дозу препарата, чтобы контролировать состояние анестезии. Автоматический ввод препаратов, основанный на методике регулирования по замкнутому циклу, дает большие преимущества загруженным работой анестезиологам и обеспечивает лучший уход за пациентом при меньших затратах. Эта методика снижает вероятность превышения дозы и дает анестезиологу возможность заметить и должным образом отреагировать на нарушения, которые могли бы остаться незамеченными или показаться слишком незначительными, чтобы послужить причиной ручного ввода препарата.

Однако для ввода наркотических препаратов с регулированием по замкнутому циклу необходимы надежные средства контроля глубины анестетического состояния, которые могли бы определять необходимые для поддержания этого состояния изменения в дозе. В современных анестетических системах с регулированием по замкнутому циклу для измерения глубины анестетического состояния используются биологические сигналы, также они применяются для организации обратной связи, по ним определяют поправки, которые нужно сделать при вводе препаратов. В частности, для обработки электроэнцефалограмм (ЭЭГ) применяются различные методы обработки сигнала, помогающие извлечь такие характеристики, как слуховой отклик (*Auditory Evoke Response* – AER) и биспектральный коэффициент, и по ним оценить глубину анестетического состояния. ЭЭГ – это электрическая активность мозга, измеренная с помощью электродов, прикрепленных к голове, а слуховой отклик – это электрическая реакция мозга на внешний звуковой раздражитель. Сигналы AER незаменимы при определении момента перехода от бессознательного состояния в сознательное, но их сложно получить, поскольку они скрыты в сигналах ЭЭГ, которые в несколько раз превышают AER. Для их выявления часто пользуются усреднением сигналов отклика на эффективный слуховой раз-

дражитель. Итак, сигналы AER необходимо обработать, извлечь их из фона ЭЭГ, а затем определить по ним характеристики, представляющие медицинский интерес (например, пики, времена задержки и форму). Биспектральный коэффициент определяется путем спектрального анализа ЭЭГ высшего порядка. Он представляет собой количественную меру комплексных изменений и взаимосвязей между частотными компонентами ЭЭГ при различных уровнях сознания. Известно, что он линейно зависит от концентрации в крови пропофола, самого распространенного анестетического препарата.

Упрощенная блок-схема анестетической системы с регулированием по замкнутому циклу, основанной на ЭЭГ, изображена на рис. 5.21. Главный элемент системы – это анализатор ЭЭГ, подсоединенный к пациенту через бифронтальные электроды для получения необработанных сигналов ЭЭГ. Для снижения шума, извлечения характеристик, анализа изменений этих характеристик и вычисления соответствующего показателя ЭЭГ в анализаторе используются различные методы обработки сигнала. К ним относятся импульсные преобразования, усреднение сигнала, биспектральный анализ и нейронные сети.

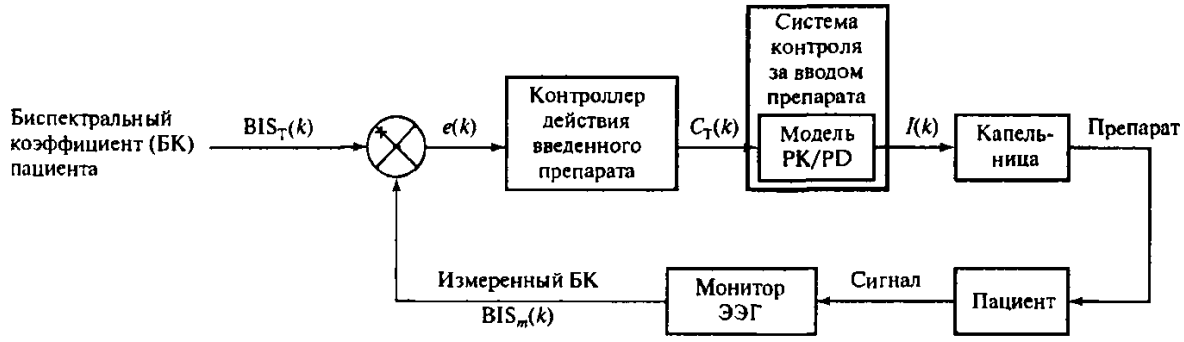


Рис. 5.21. Основанная на ЦОС анестетическая система с регулированием по замкнутому циклу

Рассчитанный показатель ЭЭГ, такой как биспектральный коэффициент, говорит о мере воздействия введенного препарата на пациента. Кроме того, рассчитанный показатель ЭЭГ играет роль сигнала обратной связи, который сравнивается с показателем ЭЭГ пациента, чтобы определить изменения, которые нужно внести в концентрацию крови пациента $C_m(k)$. Для вычисления биспектрального коэффициента и других показателей (например, A-1000, *Aspect Medical System*) создан коммерческий анализатор ЭЭГ. С помощью фармакинетической/фармадинамической (PK/PD) модели определяют скорость введения препарата пациенту $1(k)$.

Эта скорость зависит от концентрации крови пациента и других его характеристик, таких как возраст, вес и пол. Внутривенные препараты обычно вводят пациенту с помощью соответствующей анестетической капельницы под контролем специальной программы (например, STANPUMP®, S. L. Shafer, Stanford University). Пример изменения биспектрального коэффициента во время анестезии приведен на рис. 5.22.

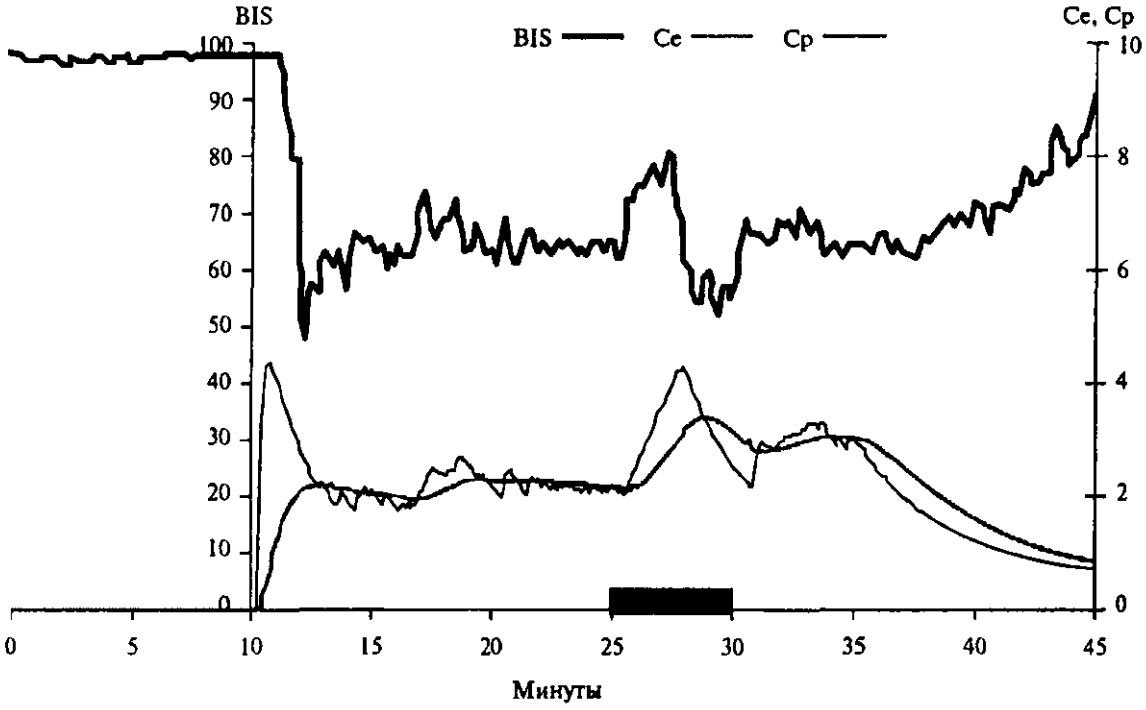


Рис. 5.22. Пример изменения биспектрального коэффициента пациента во время анестезии с регулированием по замкнутому циклу

5.5. Адаптивная цифровая фильтрация сигналов

5.5.1. Определение и назначение адаптивной обработки сигналов

Адаптацией называют процесс изменения параметров и структуры системы, а возможно, и управляющих воздействий на основе текущей информации с целью достижения определенного, обычно оптимального, состояния системы при начальной неопределенности и изменяющихся условиях работы [7].

Адаптивный автомат представляет собой систему, структура которой изменяется или приспосабливается таким образом, чтобы его поведение или функционирование улучшалось (в соответствии с некоторым подходящим критерием) в результате взаимодействия с окружающей его средой.

Простым примером автомата для автоматической адаптивной системы является автоматическая регулировка усиления (АРУ), применяемая в радио- и телевизионных приемниках. Функция этой системы – уменьшение чувствительности приемника при увеличении среднего уровня входного сигнала. Таким образом, приемник может адаптироваться к широкому диапазону уровней входных сигналов и формировать значительно более узкий диапазон уровней выходных сигналов.

5.5.2. Применение адаптивного моделирования при синтезе цифровых КИХ-фильтров

Методы адаптивного моделирования можно использовать при синтезе цифровых фильтров. На рис. 5.23 приведена структурная схема, иллюстрирующая основной принцип синтеза цифровых фильтров с конечной импульсной характеристикой. В результате процесса адаптации адаптивный фильтр имеет импульсную характеристику, наилучшим образом удовлетворяющую заданным требованиям. Этим требованиям отвечает некоторое эталонное устройство, представленное в схеме в виде эталонного фильтра. Эталонный фильтр может не существовать, поскольку в общем случае фильтр, полностью удовлетворяющий заданным требованиям, может быть физически нереализуемым. В этом случае эталонный фильтр является только некоторым описанием, вводимым для приведения задачи синтеза фильтра к задаче моделирования неизвестной системы. Далее будет показано, что для синтеза реальных фильтров не обязательно, чтобы эталонный фильтр существовал физически [17].

Предположим, что требования к фильтру заданы в виде частотной характеристики, т.е. в виде заданных значений коэффициента передачи и фазы для дискретных частот $f_1, f_2, ..., f_N$ измеренных в герцах. Вообще говоря, задается также число весовых коэффициентов цифрового фильтра, что определяет размерность адаптивного фильтра L. В процессе адаптации находятся оптимальные весовые коэффициенты (по критерию минимума среднеквадратической ошибки), при которых наилучшим образом выполняются заданные требования.

В схеме на рис. 5.23 адаптивный фильтр моделирует эталонный фильтр, построенный исходя из заданных требований, которые в большинстве случаев нельзя выполнить полностью. Однако можно предположить, что эталонный фильтр, в полной мере обладающий заданными амплитудой и фазовой частотными характеристиками, существует.

В схеме на рис. 5.23, *а* на входы идеализированного и адаптивного фильтров подается входной синусоидальный сигнал вида

$$x(t) = \sin 2\pi f_1 t\,. \tag{5.2}$$

Здесь f_1 – одна из заданных требованиями частот. Полагаем, что эталонный фильтр является линейным и его выходной сигнал

$$d(t) = a_1 \sin(2\pi f_1 t + \theta_1)\,. \tag{5.3}$$

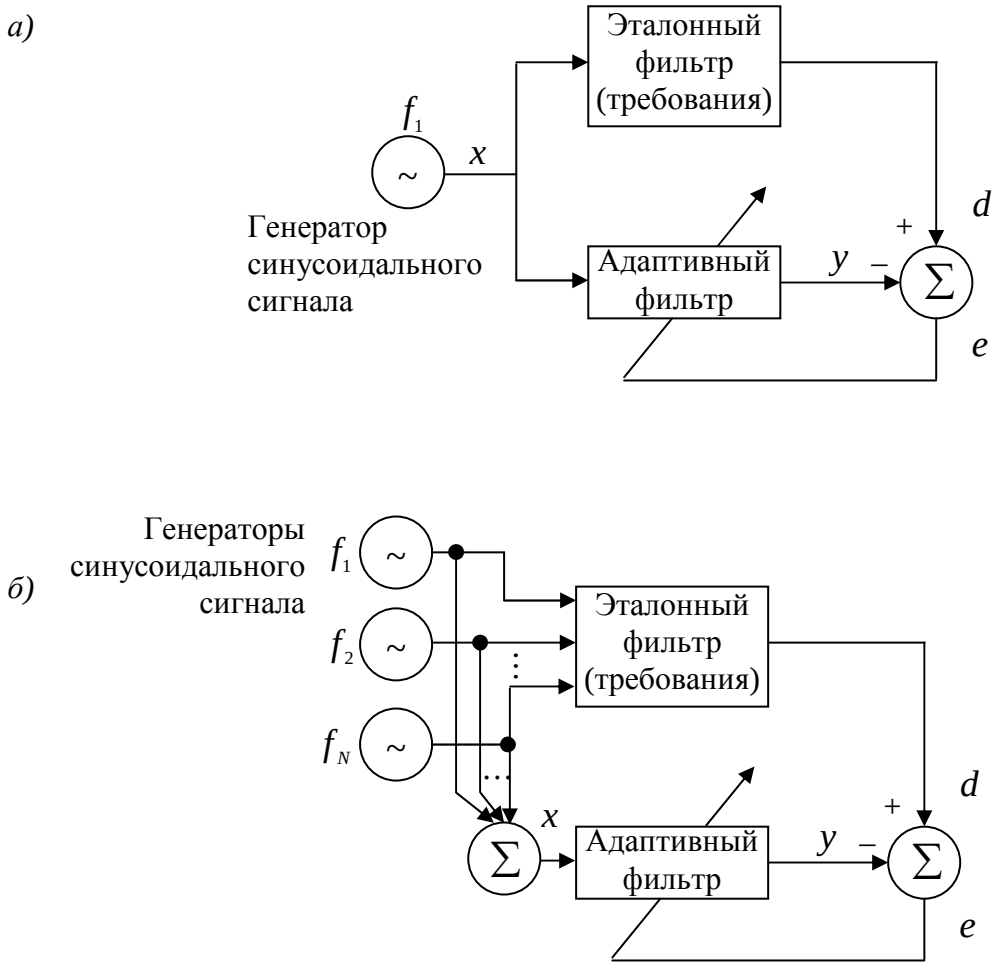


Рис. 5.23. Схема адаптивного синтеза заданного фильтра с одной частотой f_1 (*а*) и с N различными частотами (*б*)

Для адаптивного фильтра этот сигнал представляет собой полезный отклик. Отметим, что существование эталонного фильтра необязательно. Для адаптивного процесса необходим только его выходной сигнал, который можно легко построить по заданным требованиям. Коэффициент a_1 и угол θ_1 – заданные коэффициент передачи и сдвиг фазы на частоте f_1.

Для точного или, по крайней мере, приближенного выполнения требований одновременно для многих частот входной сигнал, равный сумме синусоидальных сигналов каждой из N заданных частот, подается на входы как эталонного, так и адаптивного фильтров, как показано на рис. 5.23, *б*. Этот сигнал имеет вид

$$x(t) = \sum_{l=1}^{N} \sin 2\pi f_l t \,. \tag{5.4}$$

Выходной сигнал эталонного фильтра, или полезный отклик адаптивного фильтра,

$$d(t) = \sum_{l=1}^{N} a_l \sin(2\pi f_l t + \theta_l) \,. \tag{5.5}$$

Иногда, если нельзя полностью выполнить требования для всех частот, то задают некоторые частоты, для которых выполнение требований более критично, чем для других. Это легко учесть, если ввести входные синусоидальные сигналы с различными амплитудами, при этом составляющие сигнала $d(t)$ имеют соответствующий масштабный множитель. Чем больше амплитуда входного синусоидального сигнала, тем более критично выполнение требований на этой частоте. При таком подходе l-й входной синусоидальный сигнал умножается на масштабный множитель c_l и равен

$$c_l \sin 2\pi f_l t \,, \tag{5.6}$$

где c_l – положительная константа (функция стоимости) для всех l. Входной сигнал, равный сумме синусоидальных сигналов, имеет вид

$$x(t) = \sum_{l=1}^{N} c_l \sin 2\pi f_l t \,, \tag{5.7}$$

а полезный отклик, или выходной сигнал эталонного фильтра,

$$d(t) = \sum_{l=1}^{N} a_l c_l \sin(2\pi f_l t + \theta_l) \,. \tag{5.8}$$

Здесь снова a_l и θ_l – заданные требованиями амплитуда и сдвиг фазы на частоте f_l.

Адаптивный фильтр находит оптимальное решение, обеспечивающее наилучшее приближение к заданным требованиям.

$$W^* = R^{-1}P = \begin{bmatrix} \varphi_{xx}(0) & . & . & . & \varphi_{xx}(L) \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \varphi_{xx}(L) & . & . & . & \varphi_{xx}(0) \end{bmatrix}^{-1} \begin{bmatrix} \varphi_{dx}(0) \\ . \\ . \\ . \\ \varphi_{dx}(L) \end{bmatrix}. \qquad (5.9)$$

Здесь $L+1$ – число весовых коэффициентов адаптивного линейного сумматора. Поскольку d и x известны, можно вычислить корреляционные функции в (5.9):

$$\varphi_{xx}(n) = E[x(t-nT)x(t)] = E[\sum_{l=1}^{N} c_l \sin 2\pi f_l (t-nT) \sum_{m=1}^{N} c_m \sin 2\pi f_m t]. \quad (5.10)$$

Поскольку математическое ожидание произведения двух синусоидальных функций времени с разными частотами равно нулю, выражение (5.10) принимает вид

$$\varphi_{xx}(n) = E[\sum_{l=1}^{N} c_l^2 \sin 2\pi f_l (t-nT) \sin 2\pi f_l t]. \qquad (5.11)$$

Используя тригонометрическое тождество

$$\sin 2\pi f_l (t-nT) = \sin 2\pi f_l t \cos 2\pi f_l nT - \cos 2\pi f_l t \sin 2\pi f_l nT, \qquad (5.12)$$

приводим выражение (5.11) к виду

$$\varphi_{xx}(n) = \sum_{l=1}^{N} \frac{1}{2} c_l^2 \cos 2\pi f_l nT. \qquad (5.13)$$

Равенство (5.13) получено исходя из того, что синусоидальная и косинусоидальная функции некоррелированы, а средний квадрат синусоидального сигнала равен половине квадрата его амплитуды. Таким образом, из (5.13) находятся все элементы матрицы R в (5.9). Аналогичным образом могут быть найдены элементы вектора взаимокорреляционных функций P.

Находим явное выражение для оптимального вектора весовых коэффициентов адаптивного фильтра:

$$W^* = \begin{bmatrix} \sum_{l=1}^{N} c_l^2 & \sum_{l=1}^{N} c_l^2 \cos 2\pi f_l T & . & . & . & \sum_{l=1}^{N} c_l^2 \cos 2L\pi f_l T \\ \sum_{l=1}^{N} c_l^2 \cos 2\pi f_l T & \sum_{l=1}^{N} c_l^2 & & & & \\ . & & . & & & \\ . & & & . & & \\ . & & & & . & \\ \sum_{l=1}^{N} c_l^2 \cos 2L\pi f_l T & & & & & \sum_{l=1}^{N} c_l^2 \end{bmatrix} \times$$

$$\times \begin{bmatrix} \sum_{l=1}^{N} a_l c_l^2 \cos(\theta_l) \\ \sum_{l=1}^{N} a_l c_l^2 \cos(2\pi f_l T + \theta_l) \\ . \\ . \\ . \\ \sum_{l=1}^{N} a_l c_l^2 \cos(2L\pi f_l T + \theta_l) \end{bmatrix}. \qquad (5.14)$$

При одинаковых требованиях для различных частот, т.е. при входных синусоидальных сигналах с одним значением амплитуды общее выражение (5.14) несколько упрощается.

Само по себе выражение (5.14) во многом не определяет процесс синтеза фильтра. В некотором смысле более предпочтительным является использование адаптивного процесса в схеме на рис. 5.23. Однако часто при применении в процессе синтеза ЭВМ оптимальное решение является переменным, так как матрицу R и вектор P в (5.14) можно вычислить непосредственно по заданным требованиям. Если весовые коэффициенты адаптивного фильтра соответствуют оптимальному решению W^*, то СКО

$$\xi_{\min} = \sum_{l=1}^{N} \frac{1}{2} c_l^2 \left| S_l - H_l^* \right|^2, \qquad (5.15)$$

где S_l – комплексная передаточная функция заданного эталонного фильтра на частоте f_l, а H_l^* – комплексная передаточная функция оптимального адаптивного линейного сумматора с вектором весовых коэффициентов W^* на частоте f_l.

Передаточная функция адаптивного трансверсального фильтра

$$H^*(z) = \sum_{n=0}^{L} w_n^* z^{-n}, \tag{5.16}$$

где для получения частотной характеристики вместо z подставляется $e^{j\omega} = e^{j\Omega T}$.

Нормированная угловая частота, соответствующая f_l,

$$\omega_l = 2\pi f_l T, \tag{5.17}$$

откуда H_l^* в (5.15) принимает вид

$$H_l^* = \sum_{n=0}^{L} w_n^* e^{-j2\pi n f_l T}. \tag{5.18}$$

При выводе соотношения (5.15) использован тот факт, что сигнал ошибки равен сумме синусоидальных сигналов, а если их частоты различны, то мощность суммы равна сумме мощностей.

Введение положительных множителей c определяется методом проб и ошибок. До сих пор не создано общего аналитического метода для их определения. При синтезе фильтра можно положить все c равными, затем проанализировать полученную частотную характеристику фильтра и далее увеличить значения множителей c на тех частотах, для которых выполнение заданных требований наиболее критично, и т.д. Как показывает опыт, это не представляет трудности. Хотя желательно иметь аналитический метод определения множителей c, метод проб и ошибок оказывается достаточно эффективным.

Не используя (5.14), можно проводить коррекцию адаптивного фильтра методом наименьших квадратов, получая тем самым приближенные решения. Этот способ является простым и заключается в формировании сумм синусоидальных сигналов (5.7) и (5.8) для создания соответствующих обучающих входных сигналов адаптивного фильтра, как показано на рис. 5.23. Адаптивный метод обычно используется при необходимости синтеза фильтра с большим числом весовых коэффициентов. Применение соотношения (5.14) требует решения большого числа совместных линейных уравнений. Например, если число весовых коэффициентов 256, то необходимо решить 256 линейных уравнений с 256 неизвестными. В зависимости от типа ЭВМ для синтеза фильтра более удобным может оказаться использование метода наименьшего квадратов, а не аналитического подхода.

5.5.3. Адаптивный фильтр как линейное предсказывающее устройство

Метод линейного предсказания используется для решения задач анализа и синтеза речи и для спектрального анализа. Суть метода состоит в следующем. Обрабатываемая последовательность разбивается на части, каждая из которых представляет собой конечную последовательность из N отсчетов $x(nT)$, $x((n-1)T)$, ..., $x((n-N+1)T)$. Этой конечной последовательности ставится в соответствие Q величин $a_{Q1}, a_{Q2}, ..., a_{QQ}$. Эти величины определяются так, чтобы погрешность $E_Q(a_{Q1}, a_{Q2}, ..., a_{QQ})$ была минимальной. Существуют три варианта критериев метода линейного предсказания [7].

Первый критерий соответствует линейному предсказанию «вперед». При этом оценка $\hat{x}(nT)$ очередного отсчета $x(nT)$ рассчитывается как линейная комбинация предыдущих отсчетов последовательности:

$$\hat{x}(nT) = -\sum_{j=1}^{Q} a_{Qj}^{B} x((n-j)T), \tag{5.19}$$

а текущая ошибка имеет вид

$$e_{Q_B}(nT) = x(nT) - \hat{x}(nT) = x(nT) + \sum_{j=1}^{Q} a_{Qj}^{B} x((n-j)T). \tag{5.20}$$

Коэффициенты $a_{Q1}^{B}, a_{Q2}^{B}, ..., a_{QQ}^{B}$ определяются из условия минимума общей ошибки предсказания «вперед»

$$E_{Q_B} = \sum_{n} e_{Q_B}^2(nT) = \sum_{n} (x(nT) + \sum_{j=1}^{Q} a_{Qj}^{B} x((n-j)T))^2. \tag{5.21}$$

Второй критерий соответствует линейному предсказанию «назад». При этом оценка $\hat{x}((n-Q)T)$ отсчета $x((n-Q)T)$ рассчитывается как линейная комбинация последующих отсчетов последовательности:

$$\hat{x}((n-Q)T) = -\sum_{j=1}^{Q} a_{Q_j}^{H} x((n-Q+j)T), \tag{5.22}$$

а текущая ошибка имеет вид

$$e_{Q_H}(nT) = x((n-Q)T) + \sum_{j=1}^{Q} a_{Q_j}^{H} x((n-Q+j)T). \tag{5.23}$$

Величины $a_{Q1}^{н}, a_{Q2}^{н}, ..., a_{QQ}^{н}$ определяются из условия минимума общей ошибки предсказания «назад»

$$E_{Q_H} = \sum_n e_{Q_H}^2(nT) = \sum_n (x((n-Q)T) + \sum_{j=1}^{Q} a_{Qj}^{н} x((n-j)T))^2 . \qquad (5.24)$$

Третий критерий соответствует линейному предсказанию «вперед» и «назад». При этом неизвестные параметры определяются из условия минимума общей ошибки предсказания

$$E_{Q_{B.H}} = E_{Q_B} + E_{Q_H} , \qquad (5.25)$$

где E_{Q_B} и $E_{Q_н}$ рассчитываются, соответственно, по формулам (5.22) и (5.24). Любой из описанных критериев может использоваться, например, для реализации адаптивных фильтров в системах анализа и синтеза речи. Рассмотрим принцип построения такой системы на основе линейного предсказания «вперед». Для анализа речи отсчеты речевого сигнала $x(nT)$ подаются на нерекурсивный адаптивный фильтр (рис. 5.24, *а*), так называемый обратный или отбеливающий фильтр, описываемый разностным уравнением (5.21). Коэффициенты $a_{Q1}^{B}, a_{Q2}^{B}, ..., a_{QQ}^{B}$ изменяются только после обработки очередных N отсчетов $x(nT)$.

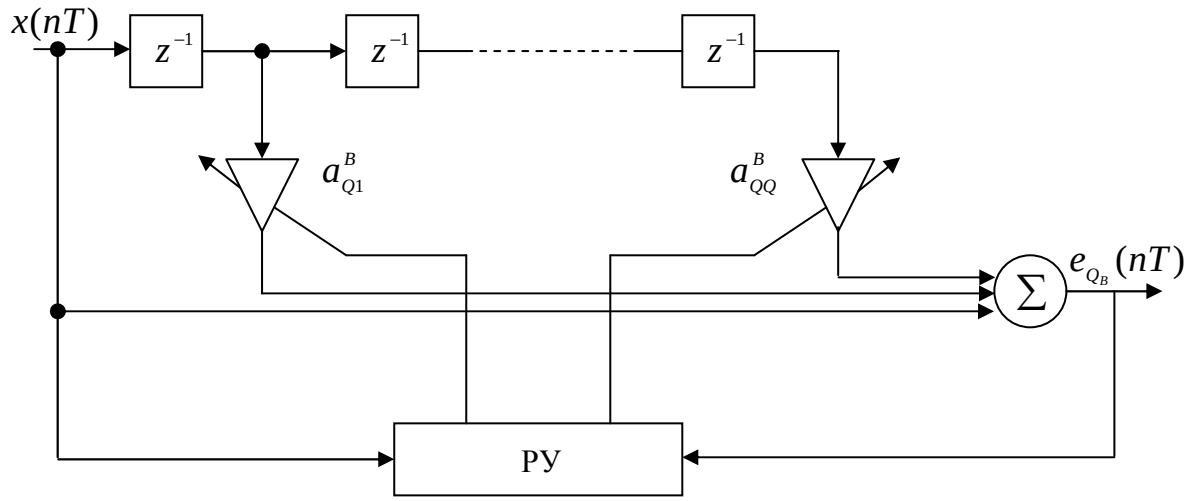
Поэтому можно принять, что

$$E_{Q_B}(z) = H(z)X(z), \qquad (5.26)$$

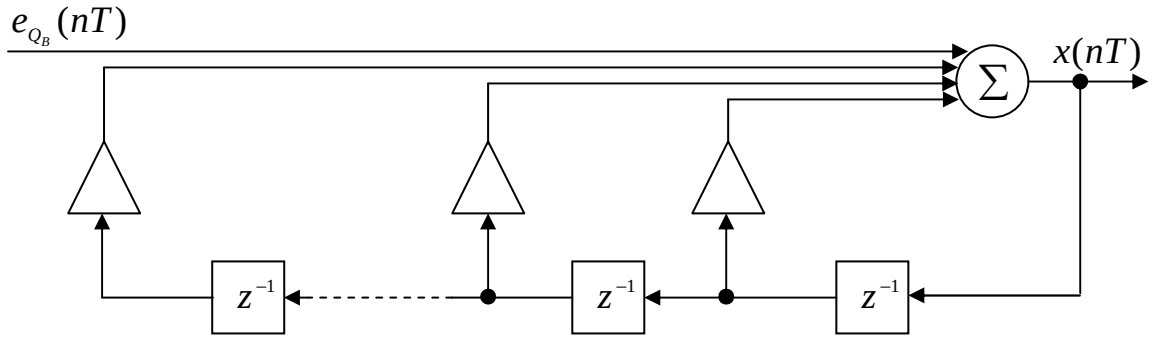
где $E_{Q_B}(z)$ – z-образ текущей ошибки $e_{Q_B}(nT)$; $H(z) = 1 + a_{Q1}^{B} z^{-1} + + ... + a_{QQ}^{B} z^{-Q}$ $H(z) = 1 + a_{Q1}^{B} z^{-1} + ... + a_{QQ}^{B} z^{-Q}$ – передаточная функция НАФ до очередного обновления коэффициентов; $X(z)$ – z-образ обрабатываемой части речевого сигнала.

Результатом анализа N отсчетов речевого сигнала являются значения коэффициентов $a_{Q1}^{B}, a_{Q2}^{B}, ..., a_{QQ}^{B}$ текущая ошибка $e_{Q_B}(nT)$ и значение общей ошибки E_{Q_B}, определяемое (5.22). Эти данные могут быть использованы для синтеза (восстановления) речевого сигнала. Для точного восстановления достаточно, согласно (5.26), подать сигнал $e_{Q_B}(nT)$ на фильтр с передаточной функцией

$$H_1(z) = \frac{1}{H(z)} = \frac{1}{1 + a_{Q1}^{B} z^{-1} + ... + a_{QQ}^{B} z^{-Q}} . \qquad (5.27)$$



а)



б)

Рис. 5.24. Нерекурсивный адаптивный фильтр

Схема восстанавливающего рекурсивного фильтра, соответствующая (5.27), изображена на рис. 5.24, *б*. Практически при передаче данных об анализируемом речевом сигнале по каналу связи с целью его восстановления на приемном конце передают лишь значения коэффициентов и некоторую информацию о текущей ошибке $e_{Q_B}(nT)$. Эта информация позволяет генерировать функцию возбуждения восстанавливающего фильтра на приемном конце. Очевидно, что восстанавливающий фильтр должен быть устойчивым, т.е. полюсы передаточной функции (5.27) должны находиться внутри единичной окружности на комплексной z-плоскости.

Существуют различные алгоритмы определения параметров фильтров по методу линейного предсказания, которые отличаются друг от друга

как выбранным критерием (5.22), (5.24), (5.25), так и областью значений переменной n, по которой выполняется суммирование в (5.22), (5.24), (5.25).

5.5.4 Перестраиваемые цифровые фильтры

К перестраиваемым относятся ЦФ с изменяемой (аппаратно или программно) частотной характеристикой. Перестройка ЦФ возможна изменением частоты дискретизации $f_Д$ и изменением значений его коэффициентов. Первый способ имеет технические ограничения и обеспечивает только масштабную перестройку фильтра. Второй способ является более универсальным. Им реализуются такие виды перестройки, как масштабная, полосовая, полная, со сменой типа ЧХ. Способ основывается на установлении однозначной связи коэффициентов фильтра с параметрами его ЧХ – частотами среза, полосой пропускания, центральной частотой, добротностью и др. Перестройка изменением коэффициентов фильтра обеспечивает независимое управление параметрами ЧХ, осуществляемое либо дискретно путем смены коэффициентов из сохраняемого в памяти их набора, либо непрерывно, путем непосредственного вычисления новых значений коэффициентов арифметическим устройством фильтра или системным котроллером [2, п. 6.2].

В качестве перестраиваемых ЦФ различного типа (ФНЧ, ФВЧ, ППФ, ПЗФ) широко используют рекурсивные звенья 2-го и 4-го порядка и многозвенные фильтры на их основе. Они применяются в многоканальных устройствах приема и передачи сигналов, в многоканальных полосовых анализаторах спектра, в следящих системах и системах управления. Перестраиваемый рекурсивный фильтр заданного типа может быть синтезирован с помощью обобщенного билинейного преобразования передаточной функции нормализованного аналогового ФНЧ-прототипа. Получаемая с помощью замены оператора $p = f_{об}(z)$ передаточная функция ЦФ $H(z) = H(p)\Big|_{p=f_{об}(z)}$ зависит как от коэффициентов передаточной функции АФПНЧ (или его нулей и полюсов), так и от коэффициентов преобразующих функций α и β, которые непосредственно связаны с перестраиваемыми параметрами ЧХ фильтра.

Так, для перестраиваемых цифровых ФНЧ и ФВЧ коэффициент α определяется их частотой среза $\lambda_c = \omega_c T_Д$: $\alpha = \mathrm{ctg}\lambda_c / 2$ и $\alpha = \mathrm{tg}\lambda_c / 2$; для ППФ и ПЗФ коэффициенты α и β являются функциями центральной час-

тоты λ_0 и полосы пропускания $\Delta\lambda = \lambda_{c2} - \lambda_{c1}$: $\alpha = \mathrm{ctg}\Delta\lambda / 2$ (ППФ), $\alpha = \mathrm{tg}\Delta\lambda / 2$ (ПЗФ), $\beta = \cos[(\lambda_{c2} + \lambda_{c1}) / 2] / \cos\Delta\lambda / 2 = \cos\lambda_0$ (для ППФ и ПЗФ). Путем дополнительного преобразования передаточной функции перестраиваемого ЦФ $H(z)$ она приводится к виду, необходимому для взаимно независимой регулировки параметров ЧХ.

Например, передаточная функция звена перестраиваемого полосового фильтра 2-го порядка может быть найдена путем преобразования передаточной функции нормализованного аналогового ФНЧ первого порядка $H(p) = 1/(p+1)$:

$$H(z) = H(p)\Big|_{p=f_{j,(z)}} = [(1 - z^{-2}) / \alpha] / [\alpha^{-1} + 1 - 2\beta z^{-1} + (1 - \alpha^{-1}) z^{-2}] =$$

$$= k(1 - A)(1 - z^{-2}) / [1 - (1 + A) B z^{-1} + A z^{-2}]. \qquad (5.28)$$

Коэффициенты ее $B = \beta = \cos\lambda_0$ и $A = (\alpha - 1) / (\alpha + 1) = (\mathrm{tg}\Delta\lambda / 2 - 1) / (\mathrm{tg}\Delta\lambda / 2 + 1)$ раздельно определяют центральную частоту λ_0 и полосу пропускания $\Delta\lambda$ ЦФ, обеспечивая независимое управление обоими параметрами ЧХ. Одновременное изменение параметров λ_0, $\Delta\lambda$ позволяет также управлять добротностью фильтра $Q = \lambda_0 / \Delta\lambda$. Коэффициент $k = 0{,}5$ является масштабирующим.

На основе перестраиваемых полосовых звеньев строятся банки равнополосных ($\Delta\lambda = \mathrm{const}$) и равнодобротных ($Q = \mathrm{const}$) ЦФ для полосового спектрального анализа сигналов.

К перестраиваемым фильтрам специального назначения относятся также амплитудные (АК) и фазовые (ФК) корректоры. Они имеют управляемую ЧХ всепропускающего фильтра и используются для компенсации искажений АЧХ и ФЧХ линейных трактов, например, каналов связи, цепей управления и др. Синтез цифровых перестраиваемых АК и ФК основывается на преобразовании передаточной функции аналоговых АК и ФК с помощью преобразующей функции $p = f_{об}(z)$, используемой для синтеза ППФ. При этом граничные частоты полосы пропускания λ_{c2}, λ_{c1} и центральная частота $\lambda_0 = (\lambda_{c2}\lambda_{c1})^{1/2}$ цифрового ФК соответствую частотам, на которых ФЧХ принимает значения, соответственно, $(-\pi / 4)M$, $(-3\pi / 4)M$ и $(-\pi / 2)M$, где M – порядок цифрового корректора. Для АК центральной

частоте $\lambda_0 = (\lambda_{c2}\lambda_{c1})^{1/2}$ соответствует минимум АЧХ, а граничным частотам λ_{c2}, λ_{c1} – уровень, определяемый коэффициентом глубины коррекции $h = 0 \pm 1$.

Корректирующие аналоговые звенья первого порядка (фазовые и амплитудные) описываются передаточными функциями вида

$$H_{ф}(p) = k(p - c)/(p + c);\ H_{a}(p) = k(p - ch)/(p + c).$$

Их коэффициенты k (масштабирующие) и c (нули и полюсы) рассчитываются хорошо разработанными методами синтеза аналоговых корректоров. Путем замены оператора $p = f_{об}(z)$ и дополнительного преобразования передаточная функция аналогового прототипа приводится к одной из возможных передаточных функций цифрового ФК и АК второго порядка:

$$H_{фк}(z^{-1}) = k_{ф}\frac{2A - 1 - 2ABz^{-1} + z^{-2}}{1 - 2ABz^{-1} + (2A - 1)z^{-2}};$$

$$H_{ак}(z^{-1}) = k_{a}\frac{A(1-h) + h - 2ABz^{-1} + [A(1+h) - h]z^{-2}}{1 - 2ABz^{-1} + (2A - 1)z^{-2}}.$$

Коэффициент $B = \cos\lambda_0$ передаточных функций ФК и АК управляет центральной частотой λ_0 полосы коррекции ФЧХ и АЧХ, коэффициент $A = \alpha/(\alpha + c)$ регулирует полосу коррекции $\Delta\lambda$, а коэффициенты k_a и $k_ф$ являются масштабирующими.

Для систем спектрального анализа и многоканального приема и передачи сигналов представляют интерес универсальные цифровые звенья, обеспечивающие наряду с перестройкой параметров также и смену вида фильтра или корректора. Их реализация основывается на соотношениях между передаточными функциями ЦФ различного вида, таких как:

$$H_{\hat{o}\hat{a}\div}(z^{-1}) = 1 - H_{\hat{o}\acute{i}\div}(z^{-1});\ H_{\ddot{i}\c{c}\hat{o}}(z^{-1}) = 1 - H_{\ddot{i}\ddot{i}\hat{o}}(z^{-1});$$

$$H_{\hat{o}\hat{e}}(z^{-1}) = H_{\ddot{i}\c{c}\hat{o}}(z^{-1}) - H_{\ddot{i}\ddot{i}\hat{o}}(z^{-1});\ H_{\grave{a}\hat{e}}(z^{-1}) = H_{\ddot{i}\c{c}\hat{o}}(z^{-1}) + hH_{\ddot{i}\ddot{i}\hat{o}}(z^{-1}).$$

Так, три последних соотношения позволяют создать универсальное звено на базе перестраиваемого ППФ с раздельными выходами ППФ, ПЗФ, АК и ФК и общим входом (рис. 5.25, *а*), или звено с раздельными входами ППФ, ПЗФ, ФК и АК и общим выходом (см. рис. 5.25, *б*).

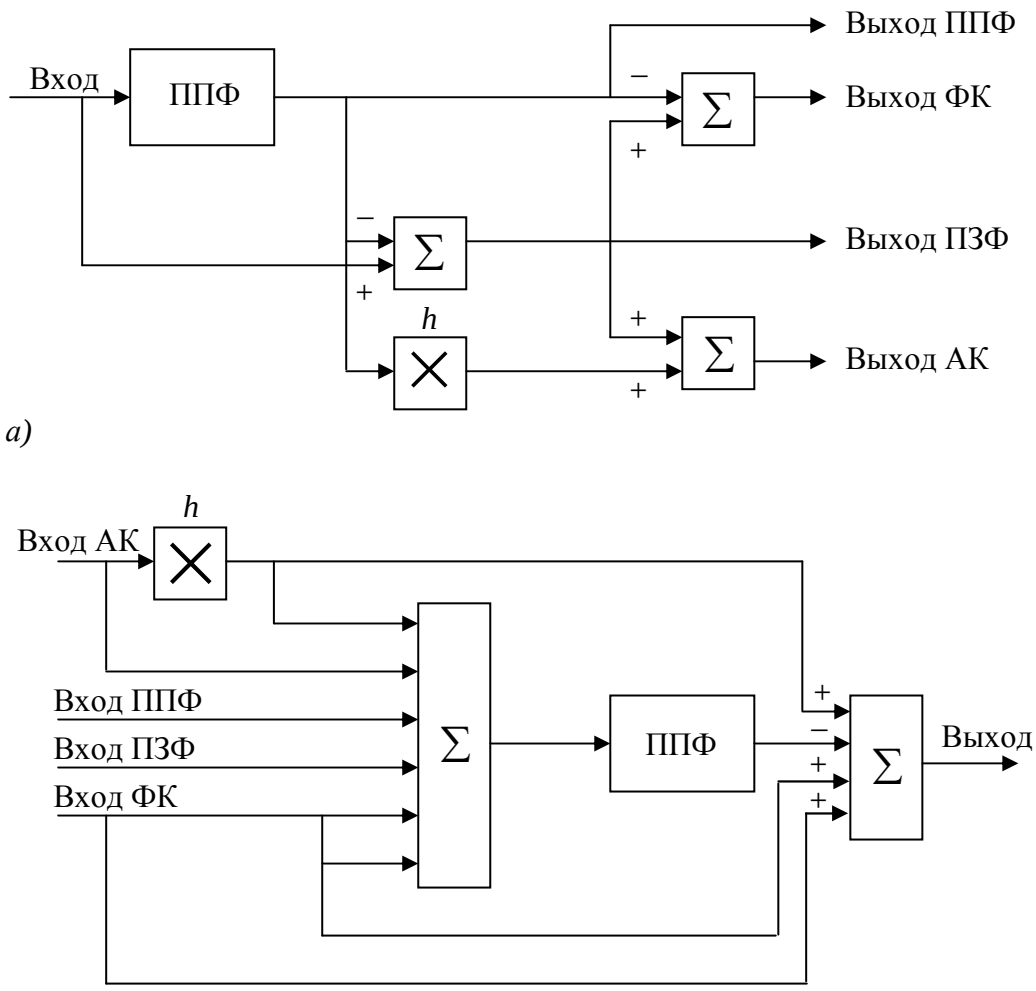


Рис. 5.25. Структурные схемы универсального звена на базе ППФ с многомерным выходом (*а*) и многомерным входом (*б*)

5.6. Применение низкоскоростных кодов в глобальных навигационных и мобильных системах телекоммуникаций

Исследования по теории и практическому использованию низкоскоростных кодов довольно долго не выходили за пределы научно-исследовательских лабораторий. Их первое широкое применение было связано прежде всего с космическими исследованиями. В ряде проектов в системах космической связи были использованы сигналы, основу которых составляли низкоскоростные коды (ЛРД-последовательности и коды РМ-1 и проекте «Маринер»). Затем было реализовано большое количество проектов военного назначения. Последние десять лет активно разрабатываются коммерческие системы различного назначения, многие хорошие каче-

ственные показатели которых невозможно получить без применения низкоскоростных кодов. Широко известными примерами использования низкоскоростных кодов являются глобальные навигационные спутниковые системы и системы мобильной связи с кодовым разделением каналов. Рассмотрим подробнее особенности применения КП в этих системах.

Глобальные навигационные спутниковые системы двойного применения GPS (Global Positioning System) и ГЛОНАСС (глобальная навигационная спутниковая система), использующие низкоскоростные коды, предназначены для высокоточного определения местоположения, направления и скорости движения объектов в любой точке земного шара. В GPS используются широкополосные сигналы, получаемые модуляцией кодовыми последовательностями Голда высокочастотного несущего колебания. Система имеет в своем составе 24 активных спутника, равномерно распределенных вокруг Земли на высоких орбитах. Для гражданского применения предназначен сигнал С/А. Сигналы всех спутников размещены в общей полосе частот и отличаются друг от друга модулирующими кодовыми последовательностями, т.е. в GPS используется кодовое разделение каналов передачи дальномерных сигналов и навигационной информации различными спутниками. Дальномерный код сигнала С/А формируется на основе кодов Голда длиной 1023 символа. Для этого выбраны проверочные полиномы $h_1(x) = x^{10}+x^3+1$ и $h_2(x) = x^{10}+x^9+x^8+x^6+x^3+x^2+1$. М-последовательность, формируемая по полиному $h_1(x)$, для всех спутников имеет одинаковую начальную фазу. М-последовательность, формируемая по полиному $h_2(x)$, для каждого спутника имеет свою начальную фазу, чем обеспечивается получение 24 различных последовательностей Голда. Схема генератора кодов Голда для GPS приведена на рис. 5.26.

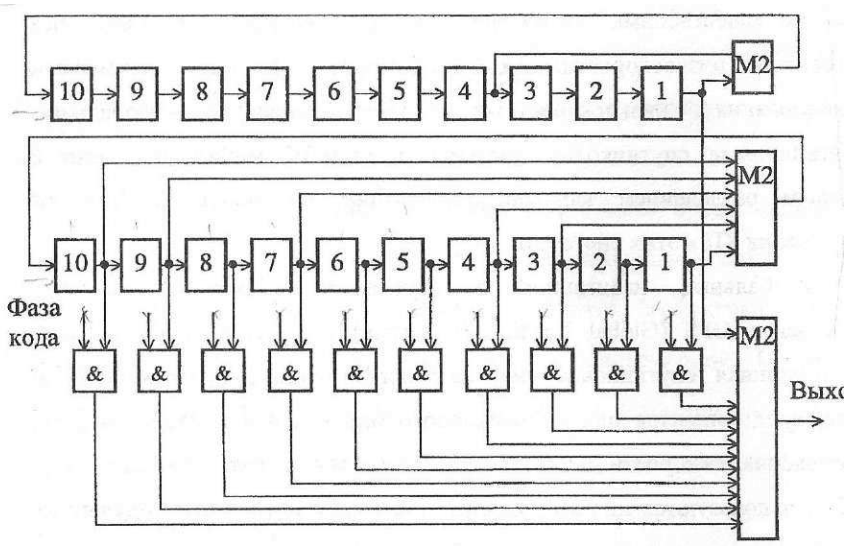


Рис. 5.26. Формирователь кода (последовательностей) Голда

Десятиразрядное двоичное число, подаваемое на входы логических схем И (см. рис. 5.26), позволяет менять форму генерируемого кода Голда. Дальномерный код дополнительно используется для передачи закодированного кодом Хэмминга навигационного сообщения, называемого альманахом. Длительность последовательности Голда равна 10^{-3} с, а ее стабильность поддерживается на высоком уровне для обеспечения высокоточных навигационных измерений.

На рис. 5.27 приведена структурная схема формирователя сигнала. Он содержит генератор кода Голда, выбранного для данного конкретного спутника, кодер кода Хэмминга для кодирования навигационного сообщения и генератор несущего колебания. Для совмещения закодированного навигационного сообщения сигналы с выходов генератора кода Голда и кодера суммируются по модулю два.

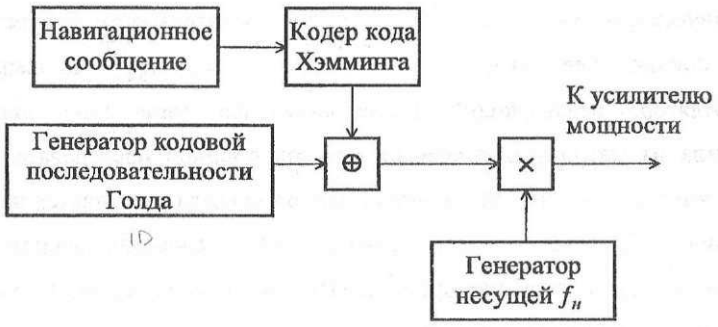


Рис. 5.27. Структурная схема формирователя сигнала навигационной системы GPS

Длительность одного кодового символа равна 20 мс, поэтому его передача происходит в течение длительности 20 периодов дальномерного кода. Затем полученная последовательность модулирует несущее колебание. Значение несущей частоты одинаково для всех спутников системы.

Навигационный приемник обнаруживает сигналы всех спутников, находящихся в данный момент времени в зоне видимости; синхронизируется с ними по временному положению и по несущей частоте. В режиме слежения за параметрами дальномерных сигналов происходит определение относительных задержек сигналов отдельных спутников, измерение доплеровских сдвигов несущих частот, прием и декодирование навигационного сообщения. По этим данным компьютер рассчитывает местоположение, направление и скорость перемещения приемника (или объекта, на котором расположен навигационный приемник).

При первоначальном включении навигационный приемник должен обнаружить сигналы отдельных спутников, находящихся в зоне приема, и

декодировать последовательности кода Голда. В данном случае под декодированием понимается определение номера или величины циклического сдвига каждой принятой последовательности. В простейших приемниках для каждого спутника используется одноканальный коррелятор, реализующий последовательное умножение принятого сигнала на матрицу циклических сдвигов опорной последовательности. При этом каждый новый принятый период сигнала умножается на новый циклический сдвиг опорного сигнала. В более сложных навигационных приемниках для ускорения процесса вычисления корреляционного вектора используются алгоритмы БПФ.

Навигационная система ГЛОНАСС использует аналогичные принципы для навигационных измерений. Основное отличие от GPS состоит в способе разделения сигналов отдельных спутников. Для этого используются различные несущие частоты, модулированные одинаковой кодовой последовательностью. В качестве ее используется одно кодовое слово кода максимальной длины с $n = 511$, задаваемое проверочным полиномом $h(x) = x^9 + x^5 + 1$, с одинаковой начальной установкой генераторов всех спутников, равной (111111100). Длительность последовательности, как и в GPS, равна 10^{-3}с, а навигационное сообщение закодировано кодом Хэмминга. В настоящее время разрабатывается несколько перспективных спутниковых навигационных систем (например, Европейская система «Галилео»), в которых также предполагается использование низкоскоростных кодов.

Структурная схема формирователя сигнала навигационной системы ГЛОНАСС приведена на рис. 5.28. В отличие от формирователя сигнала системы GPS он содержит генератор М-последовательности, одинаковый для всех спутников.

Рис. 5.28. Структурная схема формирователя сигнала навигационной системы ГЛОНАСС

Генератор несущего колебания формирует гармоническое колебание, частота которого зависит от номера спутника. Разнос частот соседних сигналов равен 562,5 кГц. Закодированное навигационное сообщение

суммируется по модулю два с М-последовательностью, а затем осуществляется модуляция несущего колебания.

Мобильные системы связи с CDMA (*Code Division Multiple Access*) широкого применения появились в результате модернизации систем связи с кодовым разделением военного назначения. В них сочетается использование всех способов кодирования информации. Это кодирование источника информации с целью устранения избыточности (сжатие речи), высокоскоростное сверточное кодирование для борьбы с одиночными ошибками, перемежение кодов для борьбы с пакетами ошибок, низкоскоростное кодирование кодом РМ-1 для передачи информации в канале с большим уровнем помех, использование М-последовательностей для расширения спектра и для разделения каналов, умножение потока данных на длинную М-последовательность для защиты передаваемой информации от несанкционированного доступа.

В системе связи с CDMA различают прямой канал связи – канал передачи информации от базовой станции к мобильным станциям и обратный канал связи – канал передачи информации от мобильной станции к базовой станции. Прямой и обратные каналы разнесены по несущей частоте. Прямой канал состоит из пилотного канала, канала синхронизации и 62 каналов передачи прямого трафика. При этом семь из них могут быть использованы в качестве каналов персонального вызова. Для разделения каналов служат ортогональные функции Уолша длиной 64 (коды РМ-1). Пилотный канал и канал синхронизации предназначены для синхронизации по несущей и тактовым частотам приемников мобильных станций. Структурная схема передающего тракта базовой станции представлена на рис. 5.29, где имеется пилотный канал, канал синхронизации и разделенные каналы вызова и передачи прямого трафика.

Во всех каналах, кроме пилотного, применен сверточный код со скоростью 1/2. Для выделения пилотного канала используется нулевая функция Уолша W_0, а для выделения канала синхронизации – W_{32}, представляющая собой меандр. Остальные функции Уолша служат для разделения каналов прямого трафика. Поскольку ансамбль функций Уолша ортогонален, то этим достигается полное разделение всех каналов в приемниках мобильных станций.

Суммирование по модулю два данных в каналах прямого трафика с М-последовательностью длиной $2^{42} - 1$ обеспечивает их защиту. Суммирование с М-последовательностью длиной $2^{15} - 1$ необходимо для уменьшения влияния сигналов соседних сот системы связи. Суммарные сигналы в квадратурных каналах подаются на модулятор передатчика базовой станции.

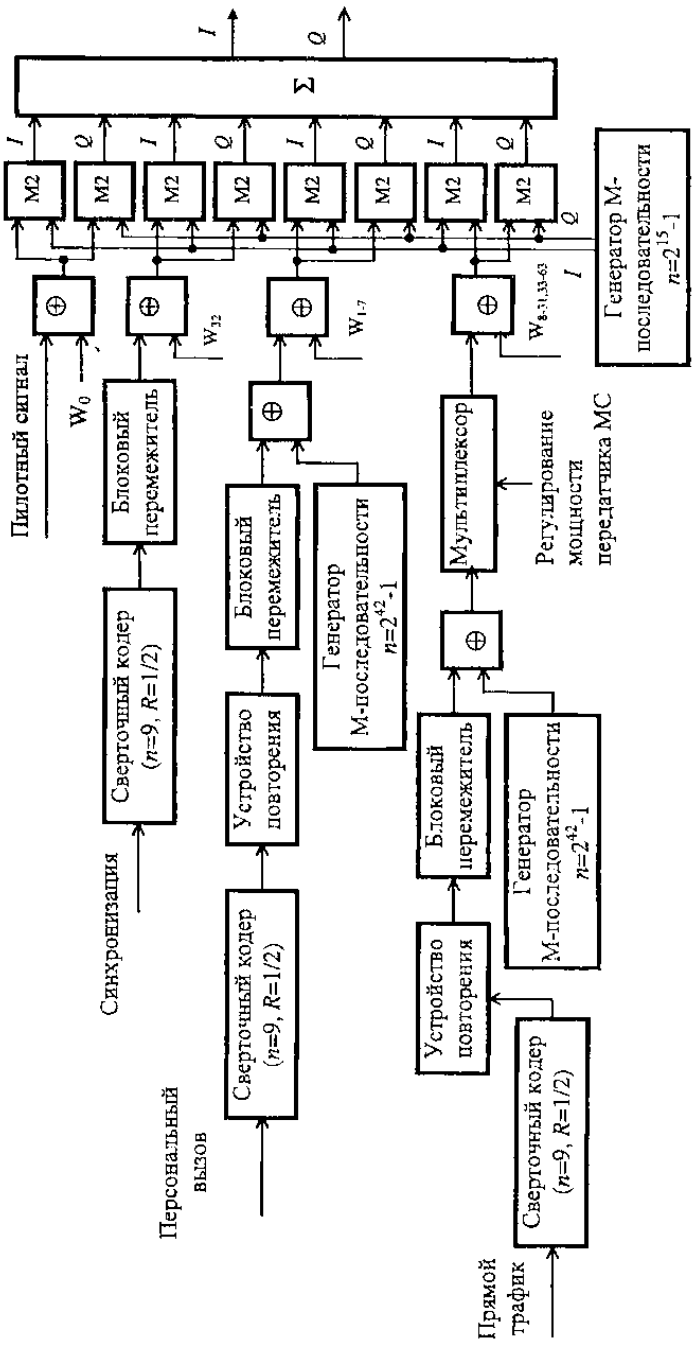


Рис. 5.29. Структурная схема передающего тракта базовой станции

На рис. 5.30 приведена структурная схема передающего тракта мобильной станции, практически повторяющая канал прямого трафика.

Основное отличие состоит в том, что ансамбль ортогональных функций Уолша использован не для разделения каналов. Низкоскоростной код РМ-1 использован здесь по прямому назначению – для помехоустойчивого кодирования. При этом поток данных делится на блоки по 6 символов; каждому двоичному числу, полученному таким образом, соответствует свое кодовое слово. Для всех мобильных станций одной соты фазы М-последовательности

длиной $2^{15} - 1$ одинаковы, а фазы М-последовательности длиной $2^n - 1$ являются строго индивидуальными, что обеспечивает их разделение в приемнике базовой станции.

В приемнике базовой станции используется декодирование в квадратурных каналах, так как неизвестна фаза несущей частоты принимаемого сигнала. Упрощенная структурная схема обработки сигнала одной мобильной станции приведена на рис. 5.31. После демодуляции и фильтрации в квадратурных каналах сигналы умножаются на синфазные М-последовательности длиной $2^{42} - 1$ и $2^{15} - 1$. Результаты поступают на многоканальные корреляторы, в качестве которых используются процессоры быстрого преобразования Уолша – Адамара. Одноименные компоненты с выходов процессоров квадратурных каналов обрабатываются в соответствии со схемой. В системе связи с CDMA используют разнесенный прием в четырех временных каналах для борьбы с многолучевостью. После объединения сигналов с выходов всех четырех корреляторов при помощи решающего устройства находится максимальная компонента. Ее порядковый номер соответствует блоку из шести двоичных символов. Далее декодированные символы следуют на деперемежитель и сверточный декодер. Такая обработка сигналов в декодере позволяет свести к минимуму вероятность ошибки и обеспечить качественную мобильную связь в сложных условиях многолучевого распространения и большого числа абонентов.

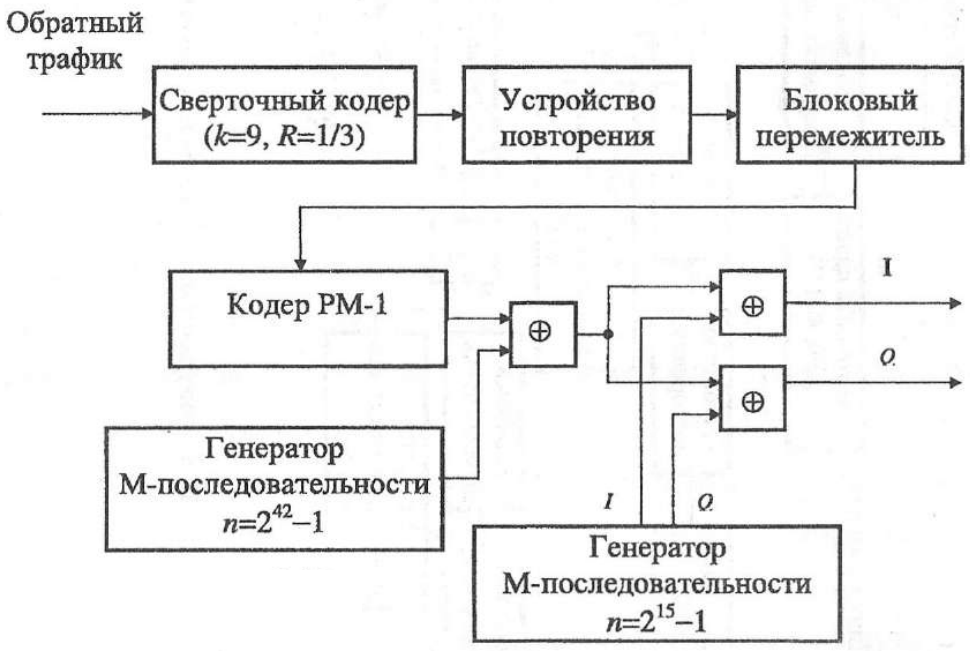


Рис. 5.30. Структурная схема передающего тракта мобильной станции

За последние 10 лет найдено несколько новых типов низкоскоростных кодов, обладающих идеальными корреляционными свойствами и имеющих высокую линейную сложность. В настоящее время не известны достаточно простые способы их формирования и обработки. Поэтому актуальными являются исследования комбинационной структуры новых кодов, методов кодирования и, что особенно важно, высокоэффективных быстрых методов обработки.

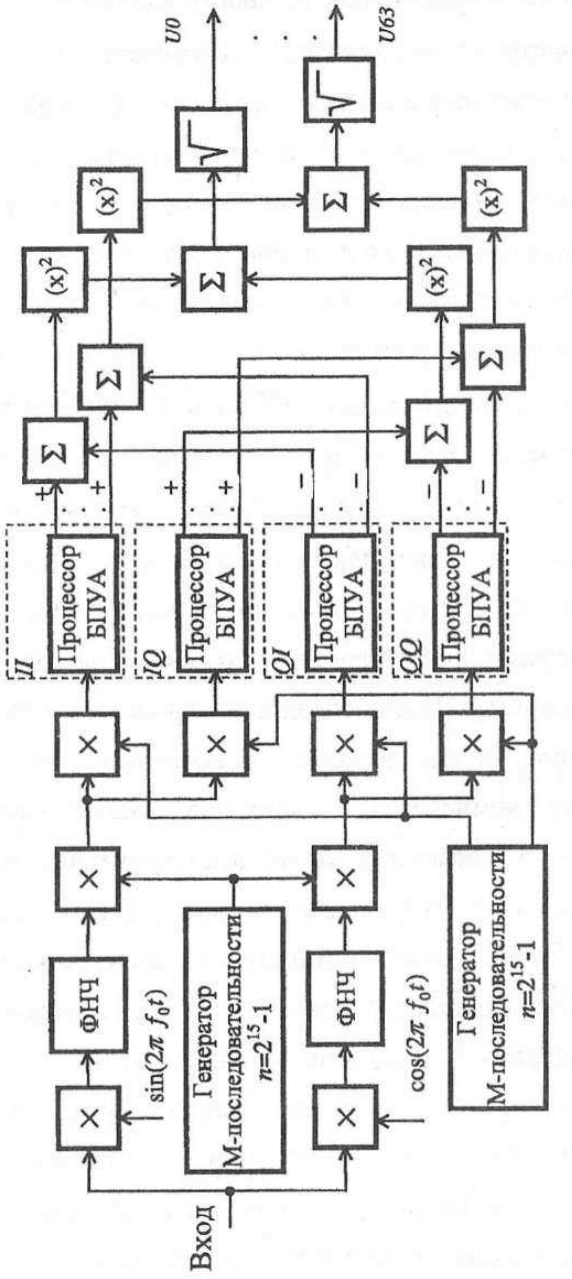


Рис. 5.31. Структура схемы канала корреляционного декодирования сигнала в приемнике базовой станции

Вопросы и задания для самопроверки

1. Перечислите основные области применения ЦОС.
2. Изобразите схематически связь ЦОС с другими техническими дисциплинами.
3. Раскройте понятие «мультиплексирование» в области телекоммуникаций ЦОС.
4. Раскройте понятие «сжатие данных» в ЦОС.
5. Какие подходы используют для генерации речи?
6. Назовите основные проблемы ЦОС при распознавании речи.
7. В каких областях радиолокации применяют ЦОС?
8. Перечислите основные виды обработки изображений.
9. Перечислите основные виды обработки видеоданных.
10. Назовите основные направления применения ЦОС при записи и воспроизведении звука.
11. Приведите блок-схему распознавания речи.
12. Изобразите структурную схему воспроизведения звука с компакт-диска.
13. Поясните архитектуру сотовой телефонной сети.
14. Раскройте принцип адаптивного телефонного эхоподавления.
15. Представьте схему обработки сигнала ЭКГ плода.
16. Раскройте сущность адаптивной цифровой фильтрации сигналов.
17. Объясните принцип работы перестраиваемого цифрового фильтра.
18. Сравните характеристики долгоиграющей пластинки и компакт-диска.
19. Перечислите основные характеристики системы сотовой связи GSM 900.
20. Изобразите упрощенную блок-схему мобильной сотовой телефонной системы.
21. Для решения каких задач используют метод линейного предсказания?
22. Какие коды используются в навигационных спутниковых системах?
23. Для решения каких задач используют дальномерный код в навигационных спутниковых системах?

ПРИЛОЖЕНИЕ А

Таблица А.1

Системный регистр (расшифровка битов)

№ бита	Название бита	Значение по умолчанию	Функции
15 – 10	Резерв	000000	Эти биты сохранены от записи
9	HPE	1	**EHPI** возможность pullup/pulldown**.** Позволяет повысить напряжение EHPI на контрольных точках HCS, HAS, HR/W, HMODE, HCNTL0, и HA1/HCNTL1. Позволяет понизить напряжение EHPI на контрольных точках HBE0 и HBE1. HPE = 0 Повышение / понижение напряжения отключается HPE = 1
8	BH	0	**EMIF** позволяет установить задержку**.** Устанавливает внутрикристальное прерывание D[31:0]. BH = 0 EMIF порта отключает. BH = 1 EMIF порта включает.
7	HBH	0	**EHPI позволяет установить возможность задержки порта данных.** Устанавливает задержку HD[15:0]. HBH = 0 – EHPI порта отключается. HBH = 1 – EHPI порта включается.
6	BOOTM3	0	**Статус BOOTM3.** Этот бит представляется только для чтения и указывает на состояние BOOTM3 контакта.
5	Резерв	0	Этот бит защищен от записи
4	Резерв	0	Этот бит защищен от записи и должен содержать 0
3	Резерв	0	Этот бит защищен от записи
2 − 0	CLKDIV	000	**CLKOUT разделяющий фактор.** Позволяет представить CLKOUT как делитель основной частоты процессора. Эта область не затрагивает программирование PLL CLKDIV = 000 CLKOUT представляет деление частоты процессора на 1 CLKDIV = 001 CLKOUT представляет деление частоты процессора на 2 CLKDIV = 010 CLKOUT представляет деление частоты процессора на 4 CLKDIV = 011 CLKOUT представляет деление частоты процессора на 6 CLKDIV = 100 CLKOUT представляет деление частоты процессора на 8 CLKDIV = 101 CLKOUT представляет деление частоты процессора на 10 CLKDIV = 110 CLKOUT представляет деление частоты процессора на 12 CLKDIV = 111 CLKOUT представляет деление частоты процессора на 14

ПРИЛОЖЕНИЕ Б

Таблица Б.1

Список регистров процессора

Регистры **VC5510**	Шестнадцатеричный адрес **(HEX)**	Описание
IER0	00	Регистр маски прерывания 0
IFR0	01	Регистр флага прерывания 0
ST0_55	02	Регистр статуса 0 для C55x
ST1_55	03	Регистр статуса 1 для C55x
ST3_55	04	Регистр статуса 3 для C55x
-	05	Зарезервировано
ST0	06	Регистр статуса ST0 (для совместимости с 54x)
ST1	07	Регистр статуса ST1 (для совместимости с 54x)
AC0L	08	Аккумулятор 0 (Эквивалент аккумулятору A в C54x)
AC0H	09	
AC0G	0A	
AC1L	0B	Аккумулятор 1 (Эквивалент аккумулятору A в C54x)
AC1H	0C	
AC1G	0D	
T3	0E	Временный регистр
TRN0	0F	Регистр перехода
AR0	10	Вспомогательный Регистр 0
AR1	11	Вспомогательный Регистр 1
AR2	12	Вспомогательный Регистр 2
AR3	13	Вспомогательный Регистр 3
AR4	14	Вспомогательный Регистр 4
AR5	15	Вспомогательный Регистр 5
AR6	16	Вспомогательный Регистр 6
AR7	17	Вспомогательный Регистр 7
SP	18	Регистр указателя стека
BK03	19	Циркулярный буферный регистр размера
BRC0	1A	Граница повторения блока
RSA0L	1B	Адрес начала блока повторения
REA0L	1C	Адрес конца блока повторения
PMST	1D	Регистр статуса состояния процессора
XPC	1E	Регистр расширения границы программы
-	1F	Зарезервировано
T0	20	Временный регистр данных 0
T1	21	Временный регистр данных 1
T2	22	Временный регистр данных 2
T3	23	Временный регистр данных 3
AC2L	24	Аккумулятор 2
AC2H	25	
AC2G	26	
CDP	27	Коэффициент указателя данных
AC3L	28	Аккумулятор 3
AC3H	29	
AC3G	2A	

Окончание табл. Б.1

Регистры **VC5510**	Шестнадцатеричный адрес **(HEX)**	Описание
DPH	2B	Расширенный указатель страницы данных
MDP05	2C	Зарезервировано
MDP67	2D	Зарезервировано
DP	2E	Начальный адрес страницы данных
PDP	2F	Начальный адрес периферийной страницы данных
BK47	30	Регистр размера циркулярного буфера для AR[4−7]
BKC	31	Регистр размера циркулярного буфера для CDP
BSA01	32	Регистр начального адреса циркулярного буфера для AR[0−1]
BSA23	33	Регистр начального адреса циркулярного буфера для AR[2−3]
BSA45	34	Регистр начального адреса циркулярного буфера для AR[4−5]
BSA67	35	Регистр начального адреса циркулярного буфера для AR[6−7]
BSAC	36	Регистр начального адреса коэффициента циркулярного буфера
BIOS	37	Местоположение хранения указателя страницы данных для 128-word Data Table
TRN1	38	Регистр перехода 1
BRS1	3A	Сохранение блока повторения 1
CSR	3B	Вычисление единственного повторения
RSA0H	3C	Адрес начала повторения 0
RSA0L	3D	
REA0H	3E	Адрес конца повторения 0
REA0L	3F	
RSA1H	40	Адрес начала повторения 1
RSA1L	41	
REA1H	42	Адрес конца повторения 1
REA1L	43	
RPTC	44	Граница повторения
IER1	45	Регистр маски прерывания 1
IFR1	46	Регистр флага прерывания 1
DBIER0	47	Отладка IER0
DBIER1	48	Отладка IER1
IVPD	49	Указатель вектора прерывания DSP
IVPH	4A	Указатель вектора прерывания HOST
ST2_55	4B	Регистр статуса 2 для C55x
SSP	4C	Указатель системного стека
SP	4D	Указатель пользовательского стека
SPH	4E	Расширенный указатель страницы данных для SP и для SSP

ПРИЛОЖЕНИЕ В

Таблица В.1

Базовые функции цифровой обработки сигналов

Функции	Описание
БПФ	
void cfft (DATA *x, ushort nx, type)	Прямое комплексное преобразование Фурье
void cfft32 (LDATA *x, ushort nx, type);	Прямое комплексное 32-х битное преобразование Фурье
void cifft (DATA *x, ushort nx, type)	Обратное комплексное преобразование Фурье
void cifft32 (LDATA *x, ushort nx, type)	Обратное комплексное 32-х битное преобразование Фурье
void cbrev (DATA *x, DATA *r, ushort n)	Комплексная бит-риверсия
void cbrev32 (LDATA *a, LDATA *r, ushort)	32-х битная бит-риверсия
void rfft (DATA *x, ushort nx, type)	Вещественное прямое преобразование Фурье
void rifft (DATA *x, ushort nx, type)	Вещественное обратное преобразование Фурье
Фильтрация и свертка	
ushort fir (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)	Фильтрация прямая форма
ushort fir2 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)	Фильтрация прямая форма (оптитимизированный алгоритм)
ushort firs (DATA *x, DATA *h, DATA *r,)	Симметричный фильтр
ushort cfir (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)	Комплексный фильтр
ushort convol (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)	Свертка
ushort convol1 (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)	Свертка (с применением двойного сдвига с накоплением)
ushort convol2 (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)	Свертка (с применением двойного сдвига с накоплением)
ushort iircas4 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)	Фильтр IIR, прямая форма II с использованием 4-х коэффициентов Бигвуда
ushort iircas5 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)	Фильтр IIR, прямая форма II с использованием 5-и коэффициентов Бигвуда

Продолжение табл. В.1

Функции	Описание
ushort iircas51 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)	Каскадный фильтр IIR, прямая форма I с использованием 4-х коэффициентов Биг-вуда
ushort iirlat (DATA *x, DATA *h, DATA *r, DATA *pbuffer, int nx, int nh)	Инверсия решетки фильтра IIR
ushort firlat (DATA *x, DATA *h, DATA *r, DATA *pbuffer, int nx, int nh)	Прямая решетка фильтра FIR
ushort firdec (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nh, ushort nx, ushort D)	Фильтр-дециматор
ushort firinterp (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nh, ushort nx, ushort I)	Фильтр-интерполятор
ushort hilb16 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)	Фильтр-трансформатор Хиберта
ushort iir32 (DATA *x, LDATA *h, DATA *r, LDATA *dbuffer, ushort nbiq, ushort nr)	Фильтр с двойной точностью
Адаптивная фильтрация	
ushort dlms (DATA *x, DATA *h, DATA *r, DATA *des, DATA *dbuffer, DATA step, ushort nh, ushort nx)	LMS фильтр (с задержками)
ushort oflag = dlmsfast (DATA *x, DATA *h, DATA *r, DATA*des, DATA *dbuffer, DATA step, ushort nh, ushort nx)	Адаптивный фильтр с задержками
Корреляция	
ushort acorr (DATA *x, DATA *r, ushort nx, ushort nr, type)	Автокорелляция
ushort corr (DATA *x, DATA *y, DATA *r, ushort nx, ushort ny, type)	Корелляция
Тригонометрия	
ushort sine (DATA *x, DATA *r, ushort nx)	Синус вектора
ushort atan2_16 (DATA *i, DATA *q, DATA *r, ushort nx)	Инверсия тангенса четвертого квадранта вектора
ushort atan16 (DATA *x, DATA *r, ushort nx)	Арктангенс
ushort add (DATA *x, DATA *y, DATA *r, ushort nx, ushort scale)	Оптимизированный вектор суммы
ushort expn (DATA *x, DATA *r, ushort nx)	Вектор экспоненты

Окончание табл. В.1

Функции	Описание
short bexp (DATA *x, ushort nx) ushort logn (DATA *x, LDATA *r, ushort nx)	Экспонента всех величин вектора
ushort log_2 (DATA *x, LDATA *r, ushort nx)	Логарифм по основанию 2
ushort log_10 (DATA *x, LDATA *r, ushort nx)	Логарифм по основанию 10
short maxidx (DATA *x, ushort ng, ushort ng_size)	Индекс максимальной величины вектора
short maxidx34 (DATA *x, ushort nx)	Индекс максимального элемента вектора ≤ 34
short maxval (DATA *x, ushort nx)	Максимальный элемент вектора
void maxvec (DATA *x, ushort nx, DATA *r_val, DATA *r_idx)	Индекс и величина максимального элемента в векторе
short minidx (DATA *x, ushort nx)	Индекс минимальной величины в векторе
short minval (DATA *x, ushort nx)	Минимальная величина вектора
void minvec (DATA *x, ushort nx, DATA *r_val, DATA *r_idx)	Индекс и величина минимального элемента в векторе
ushort mul32 (LDATA *x, LDATA *y, LDATA *r, ushort nx)	32-хбитный вектор умножения
short neg (DATA *x, DATA *r, ushort nx)	16-тибитный вектор отрицания
short neg32 (LDATA *x, LDATA *r, ushort nx)	32-хбитный вектор отрицания
short power (DATA *x, LDATA *r, ushort nx)	Сумма квадратов (мощность)
void recip16 (DATA *x, DATA *r, DATA *rexp, ushort nx)	Вектор аналог
Продолжение таблицы В.1	
void ldiv16 (LDATA *x, DATA *y, DATA *r, DATA *rexp, ushort nx)	32-хбитный с 16-тибитным разделением
ushort sqrt_16 (DATA *x, DATA *r, short nx)	Квадратный корень
short sub (DATA *x, DATA *y, DATA *r, ushort nx, ushort scale)	Вектор вычитания
Работа с матрицами	
ushort mmul (DATA *x1, short row1, short col1, DATA *x2, short row2, short col2, DATA *r)	Матричное умножение
ushort mtrans (DATA *x, short row, short col, DATA *r)	Транспонирование

ПРИЛОЖЕНИЕ Г

Алгоритм фильтрации

```
float kaf[16]={
0, 0.0005,
0.0034, 0.0149,
0.0446, 0.0982,
0.1637, 0.2104,
0.2104, 0.1637,
0.0982, 0.0446,
0.0149, 0.0034,
0.0005, 0,};
int a;
int x;
int y[1];
int inp[16]; /*Обьявление буфера входных данных, он равен размеру окна фильтра */
int i;

float rez[16]; /*Объявление буфера выходных данных, он равен размеру окна фильтра */
float out[1];  /*буфер выходных отсчетов*/
int s;

main(){
  while (1) { /*бесконечный цикл*/
  float *rez_buf = &rez[0]; /*присвоение адреса одной переменной другой переменной */
  a=16;            /*устанавливаем размер окна*/
  x=0;
   while (a--) { /*обнуляем память по адресам выходного буфера*/
   *rez_buf++ = 0;}
   while (x!=16)/*Цикл формирования выходного буфера данных*/
   {a=15;
    while (a!=-1) /*цикл сдвига данных в буфере на 1 адрес вверх*/
     { inp[a]=inp[a-1];
     a--;}
     inp[0]=y[0]; /*чтение данных из внешнего файла*/
     s=15;
     while (s!=-1) /*перемножение буфера входных данных на коэффициенты и формиро-
                     вание одного элемента буфера выходных данных*/
      { rez[x]=rez[x]+inp[s]*kaf[s];
      s--;}
      out[0]=rez[x]; /*запись результата по адресу выхода*/
      x++;
   }
  }
}
```

ПРИЛОЖЕНИЕ Д

Листинг файла dsk5510_aic.h

```
 *  ======== dsk5510_aic23.h ========
 *  Codec interface for AIC23 on the DSK5510 board
#ifndef DSK5510_AIC23_
#define DSK5510_AIC23_
#ifdef __cplusplus
extern "C" {
#endif
#include <csl.h>
#include <csl_mcbsp.h>
/* AIC23 McBSP defines */
#define DSK5510_AIC23_DATAHANDLE    C55XX_DMA_MCBSP_hMcbsp
#define DSK5510_AIC23_CONTROLHANDLE C55XX_CONTROLHANDLE_hMcbsp
/* McBSP handles */
extern MCBSP_Handle DSK5510_AIC23_DATAHANDLE,
DSK5510_AIC23_CONTROLHANDLE;
/* Codec module definitions */
#define DSK5510_AIC23_NUMREGS        10
#define DSK5510_AIC23_LEFTINVOL      0
#define DSK5510_AIC23_RIGHTINVOL     1
#define DSK5510_AIC23_LEFTHPVOL      2
#define DSK5510_AIC23_RIGHTHPVOL     3
#define DSK5510_AIC23_ANAPATH        4
#define DSK5510_AIC23_DIGPATH        5
#define DSK5510_AIC23_POWERDOWN      6
#define DSK5510_AIC23_DIGIF          7
#define DSK5510_AIC23_SAMPLERATE     8
#define DSK5510_AIC23_DIGACT         9
#define DSK5510_AIC23_RESET          15
/* Frequency Definitions */
#define DSK5510_AIC23_FREQ_8KHZ      1
#define DSK5510_AIC23_FREQ_16KHZ     2
#define DSK5510_AIC23_FREQ_24KHZ     3
#define DSK5510_AIC23_FREQ_32KHZ     4
#define DSK5510_AIC23_FREQ_44KHZ     5
#define DSK5510_AIC23_FREQ_48KHZ     6
#define DSK5510_AIC23_FREQ_96KHZ     7
/* Codec Handle */
typedef int DSK5510_AIC23_CodecHandle;
/* Parameter Structure for the DSK5510 AIC23 Codec */
typedef struct DSK5510_AIC23_Config {
   int regs[DSK5510_AIC23_NUMREGS];
} DSK5510_AIC23_Config;
#define DSK5510_AIC23_DEFAULTCONFIG { \
   0x0017, /* Set-Up Reg 0      Left line input channel volume control */
            /* LRS     0        simultaneous left/right volume: disabled */
           /* LIM     0        left line input mute: disabled */
```

```
        /* XX      00        reserved */\
        /* LIV     10111     left line input volume: 0 dB */\
0x0017, /* Set-Up Reg 1      Right line input channel volume control */ \
        /* RLS     0         simultaneous right/left volume: disabled */\
        /* RIM     0         right line input mute: disabled */       \
        /* XX      00        reserved */                              \
        /* RIV     10111     right line input volume: 0 dB */
0x008d, /* Set-Up Reg 2      Left channel headphone volume control */  \
        /* LRS     0         simultaneous left/right volume: enabled */ \
        /* LZC     1         left channel zero-cross detect: enabled */ \
        /* LHV     1011000   left headphone volume: 0 dB */          \
0x008d, /* Set-Up Reg 3      Right channel headphone volume control */ \
        /* RLS     0     simultaneous right/left volume: enabled */ \
        /* RZC     1     right channel zero-cross detect: enabled */\
        /* RHV     1011000   right headphone volume: 0 dB */         \
0x0011, /* Set-Up Reg 4      Analog audio path control */             \
        /* X       0         reserved */                            \
        /* STA     00        sidetone attenuation: -6 dB */          \
        /* STE     0         sidetone: disabled */                  \
        /* DAC     1         DAC: selected */                       \
        /* BYP     0         bypass: off */                         \
        /* INSEL   0         input select for ADC: line */          \
        /* MICM    0         microphone mute: disabled */           \
        /* MICB    1         microphone boost: enabled */           \
0x0000, /* Set-Up Reg 5      Digital audio path control */           \
        /* XXXXX   00000     reserved */                            \
        /* DACM    0         DAC soft mute: disabled */             \
        /* DEEMP   00        deemphasis control: disabled */        \
        /* ADCHP   0         ADC high-pass filter: disabled */      \
0x0000, /* Set-Up Reg 6      Power down control */                  \
        /* X       0         reserved */                            \
        /* OFF     0         device power: on (i.e. not off) */     \
        /* CLK     0         clock: on */                           \
        /* OSC     0         oscillator: on */                      \
        /* OUT     0         outputs: on */                         \
        /* DAC     0         DAC: on */                             \
        /* ADC     0         ADC: on */                             \
        /* MIC     0         microphone: on */                      \
        /* LINE    0         line input: on */                      \
0x0043, /* Set-Up Reg 7      Digital audio interface format */      \
        /* XX      00        reserved */                            \
        /* MS      1         master/slave mode: master */           \
        /* LRSWAP  0         DAC left/right swap: disabled */       \
        /* LRP     0         DAC lrp: MSB on 1st BCLK */            \
        /* IWL     00        input bit length: 16 bit */            \
        /* FOR     11        data format: DSP format */             \
0x0081, /* Set-Up Reg 8      Sample rate control */                 \
        /* X       0         reserved */                            \
        /* CLKOUT  1         clock output divider: 2 (MCLK/2) */    \
```

```
            /* CLKIN   0         clock input divider: 2 (MCLK/2) */        \
            /* SR,BOSR 00000     sampling rate: ADC  48 kHz DAC  48 kHz */ \
             /* USB/N   1         clock mode select (USB/normal): USB */    \
   0x0001  /* Set-Up Reg 9      Digital interface activation */          \
            /* XX..X   00000000   reserved */                          \
            /* ACT     1         active */                             \
}
/* Set codec register regnum to value regval */
void DSK5510_AIC23_rset(DSK5510_AIC23_CodecHandle hCodec, Uint16 regnum,
Uint16 regval);
/* Return value of codec register regnum */
Uint16 DSK5510_AIC23_rget(DSK5510_AIC23_CodecHandle hCodec, Uint16 regnum);
/* Open the codec with id and return handle */
DSK5510_AIC23_CodecHandle DSK5510_AIC23_openCodec(int id,
DSK5510_AIC23_Config *Config);
/* Close the codec */
void DSK5510_AIC23_closeCodec(DSK5510_AIC23_CodecHandle hCodec);
/* Configure the codec register values */
void DSK5510_AIC23_config(DSK5510_AIC23_CodecHandle hCodec,
DSK5510_AIC23_Config *Config);
/* Write a 16-bit value to the codec */
CSLBool DSK5510_AIC23_write16(DSK5510_AIC23_CodecHandle hCodec, Int16 val);
/* Write a 32-bit value to the codec */
CSLBool DSK5510_AIC23_write32(DSK5510_AIC23_CodecHandle hCodec, Int32 val);
/* Read a 16-bit value from the codec */
CSLBool DSK5510_AIC23_read16(DSK5510_AIC23_CodecHandle hCodec, Int16 *val);
/* Read a 32-bit value from the codec */
CSLBool DSK5510_AIC23_read32(DSK5510_AIC23_CodecHandle hCodec, Int32 *val);
/* Set the codec output gain */
void DSK5510_AIC23_outGain(DSK5510_AIC23_CodecHandle hCodec, Uint16 outGain);
/* Set the codec loopback mode */
void DSK5510_AIC23_loopback(DSK5510_AIC23_CodecHandle hCodec, CSLBool
mode);
/* Enable/disable codec mute mode */
void DSK5510_AIC23_mute(DSK5510_AIC23_CodecHandle hCodec, CSLBool mode);
/* Enable/disable codec powerdown modes for DAC, ADC */
void DSK5510_AIC23_powerDown(DSK5510_AIC23_CodecHandle hCodec, Uint16 sect);
/* Set the codec sample rate frequency */
void DSK5510_AIC23_setFreq(DSK5510_AIC23_CodecHandle hCodec, Uint32 freq);
#ifdef __cplusplus
}
#endif
#endif
 *  ======== tone.c ========
 */
#include "tonecfg.h"
#include "Fil_h.h"
#include "math.h"
#include "tms320.h"
```

```
#include "dsplib.h"
#include "dsk5510.h"
#include "dsk5510_aic23.h"
#define TRUE 1
DSK5510_AIC23_Config config = {
   0x0017, // 0 DSK5510_AIC23_LEFTINVOL  Left line input channel volume
   0x0017, // 1 DSK5510_AIC23_RIGHTINVOL Right line input channel volume
   0x008d, // 2 DSK5510_AIC23_LEFTHPVOL  Left channel headphone volume
   0x008d, // 3 DSK5510_AIC23_RIGHTHPVOL Right channel headphone volume
   0x0011, // 4 DSK5510_AIC23_ANAPATH    Analog audio path control
   0x0000, // 5 DSK5510_AIC23_DIGPATH    Digital audio path control
   0x0000, // 6 DSK5510_AIC23_POWERDOWN  Power down control
   0x0043, // 7 DSK5510_AIC23_DIGIF      Digital audio interface format
   0x0081, // 8 DSK5510_AIC23_SAMPLERATE Sample rate control
   0x0001  // 9 DSK5510_AIC23_DIGACT     Digital interface activation
};
int i,a;
float out;
Int16 x;
Int16 inp[16];
void main()
{
   DSK5510_AIC23_CodecHandle hCodec;
Int16 *adr=&x;
    /* Initialize the board support library, must be called first */
   DSK5510_init();
   /* Start the codec */
   hCodec = DSK5510_AIC23_openCodec(0, &config);
 DSK5510_AIC23_setFreq(hCodec, 5);
  while (TRUE)
   {
    while(!DSK5510_AIC23_read16(hCodec, adr));
      a=15;
          while (a!=-1) /*цикл сдвига данных в буфере на 1 адрес вверх*/
           {  inp[a]=inp[a-1];  a--;  }
     inp[0]=*adr;
     i=14;
    out=0;
/* перемножение буфера входных данных   на коэффициенты и формирование одного
элемента   буфера выходных данных*/
    while (i!=0)
   { out=out+inp[i]*kaf[i];  i--;  }
    x=out;
    x=*adr;*/
         while (!DSK5510_AIC23_write16(hCodec, x));
         while (!DSK5510_AIC23_write16(hCodec, x));
   }
   /* Close the codec */
   DSK5510_AIC23_closeCodec(hCodec);
}
```

СОДЕРЖАНИЕ

Введение в курс «Цифровая обработка сигналов» ... 3
Модуль 1. ДИСКРЕТНЫЕ И ЦИФРОВЫЕ СИГНАЛЫ И СИСТЕМЫ ... 9
1.1. Дискретизация и квантование сигналов ... 9
1.2. Представление сигналов функциональными рядами ... 14
1.3.. Математические модели дискретной и цифровой систем обработки сигналов ... 16
1.4. Спектры дискретных сигналов ... 23
1.5. Технические показатели эффективности ЦОС ... 26
1.6. Эффективность алгоритмов и оценка их вычислительной сложности ... 27
Вопросы и задания для самопроверки ... 31
Модуль 2 ПРЕОБРАЗОВАНИЯ СИГНАЛОВ В ЦИФРОВЫХ СИСТЕМАХ ... 32
2.1. Преобразование Фурье и *Z*-преобразование ... 32
2.2. Дискретное преобразование Фурье и его свойства ... 46
2.3. Алгоритмы быстрого преобразования Фурье ... 55
2.4. Преобразование Уолша – Адамара ... 70
2.5. Преобразование Хаара ... 80
2.6. Кольцо и поле ... 84
2.7. Теоретико-числовое преобразование ... 86
2.8. Ключевые операции ЦОС – свертка и корреляция ... 90
Вопросы и задания для самопроверки ... 116
Модуль 3. БАЗОВЫЕ МОДЕЛИ СИСТЕМ ЦОС ... 117
3.1. Методы и модели ЦОС ... 117
3.2. Математическое описание и типы ЦФ ... 123
3.3. Структуры ЦФ ... 131
3.4. Частотные характеристики и элементы синтеза ЦФ ... 137
3.5. Синтез цифровых фильтров ... 140
3.6. Специализированные цифровые фильтры ... 149
3.7. Непараметрические методы спектрального и корреляционного анализа ... 179
3.8. Современные параметрические методы оценки ... 195
3.9. Методы и алгоритмы проекционной обработки сигналов ... 199
3.10. Оптимальные линейные фильтры ... 205
Вопросы и задания для самопроверки ... 216
Модуль 4. АППАРАТНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ ... 217
4.1. Универсальные и специализированные процессоры ЦОС ... 217
4.2. Компьютерные архитектуры обработки сигналов ... 218
4.3. Универсальные процессоры ЦОС ... 241
4.4. Выбор цифрового процессора сигналов ... 258
4.5. Реализация алгоритмов ЦОС ... 260
4.6. Микропроцессорный комплекс TMS320VC5510 ... 279
Вопросы и задания для самопроверки ... 287
Модуль 5. ПРИМЕНЕНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В РАДИОЭЛЕКТРОННЫХ СИСТЕМАХ ... 288
5.1. Области применения цифровой обработки сигналов ... 288
5.2. Применение ЦОС при записи и воспроизведении звука ... 300
5.3. Применение ЦОС в телекоммуникации ... 309
5.4. Применение ЦОС в биомедицине ... 317
5.5. Адаптивная цифровая фильтрация сигналов ... 324
5.6. Применение низкоскоростных кодов в глобальных навигационных и мобильных системах телекоммуникаций ... 337
Вопросы и задания для самопроверки ... 345
Приложение А. Системный регистр (расшифровка битов) ... 346
Приложение Б. Список регистров процессора ... 347
Приложение В. Базовые функции цифровой обработки сигналов ... 349
Приложение Г. Алгоритм фильтрации ... 352
Приложение Д. Листинг файла dsk5510_aic.h ... 353

ЛИТЕРАТУРА

1. Айфичер, Э. Цифровая обработка сигналов: практический подход: пер. с англ. – 2-е изд. / Э. Айфичер, Б. Джервис. – М.: Изд. дом «Вильямс», 2004. – 992 с.

2. Глинченко, А.С. Цифровая обработка сигналов: учеб. пособие. В 2 ч. / А.С. Глинченко. – Красноярск: ИПЦ КГТУ, 2001. – 184 с.

3. Оппенгейм, А.В. Цифровая обработка сигналов: пер. с англ. / А.В. Оппенгейм, Р.В. Шафер; под ред. С.Я. Шаца. – М.: Связь, 1979. – 416с.

4. Лосев, В.В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: учеб. пособие для вузов / В.В. Лосев. – Минск: Выш. шк., 1990. – 132 с.

5. Ярославский, Л.П. Цифровая обработка сигналов в оптике и голографии: введение в цифровую оптику / Л.П. Ярославский. – М.: Радио и связь, 1987. – 296 с.

6. Солонина, А.И. Основы цифровой обработки сигналов: курс лекций / А.И. Солонина, Д.А. Улахович, Е.Б. Соловьева. – 2-е изд., испр. и перераб. – СПб.: БХВ-Петербург, 2005. – 768 с.

7. Гольденберг, Л. М. Цифровая обработка сигналов: справочник / Л.М. Гольденберг, Б.Д.Матюшкин, М.Н. Поляк. – М.: Радио и связь, 1985.

8. Бендат, Дж. Прикладной анализ случайных данных / Дж. Бендат, А. Пирсол. – М.: Мир, 1989. – 540 с.

9. Канасевич, Э.Р. Анализ временных последовательностей в геофизике / Э.Р. Канасевич. – М.: Недра, 1985. – 300 с.

10. Рапопорт, М.Б. Вычислительная техника в полевой геофизике: учебник для вузов / М.Б. Рапопорт. – М.: Недра, 1993.

11. Крот, А.М. Быстрые алгоритмы и программы цифровой спектральной обработки сигналов и изображений / А.М. Крот, Е.Б. Минервина. – Минск, 1995. – 407 с.

12. Новиков, Л.В. Основы вейвлет-анализа сигналов: учеб. пособие / Л.В. Новиков. – СПб.: МОДУС+, 1999.

13. McGillem, C.D. Continuous and Discrete Signal and System Analysis / C.D. McGillem, G.R. Cooper. – New York: Holt, Rinehart and Winston, 1974.

14. Саломатин, С.Б. Цифровая обработка сигналов в радиоэлектронных системах: учеб. пособие по курсу «Цифровая обраб. сигналов» для студентов специальности 1-39 01 02 «Радиоэлектрон. системы» днев. форма обучения / С.Б. Саломатин. – Минск, 2002. – 88 с.

15. Рабинен, Л. Теория и применение цифровой обработки сигналов: / Л. Рабинен, Б. Гоулд; пер. с англ. Л. Якименко. – М.: Мир, 1978, 848 с.

16. Голд, Б. Цифровая обработка сигналов: пер. с англ. / Б. Голд, Ч. Рэйдер; под ред. А.М. Трахмана. – М.: Сов. радио, 1973. – 368 с.

17. Уидроу, Б. Адаптивная обработка сигналов: пер. с англ. / Б. Уидроу, С. Стирнз. – М.: Радио и связь, 1989. – 440 с.

18. Макс, Ж. Методы и техника обработки сигналов при физических измерениях. В 2 т. Т. 2 / Ж. Макс. – М.: Мир, 1983.

19. Микрокомпьютерные медицинские системы: проектирование и применения / под ред. У. Томпкинса, Дж. Уэбстера. – М.: Мир, 1983.

20. Гольденберг, Л.М. Цифровая обработка сигналов: учеб. пособие / Л.М. Гольденберг, Б.Д. Матюшкин, М.Н. Поляк. – М.: Высш. шк., 1990.

21. Марпл, С.Л. Цифровой спектральный анализ и его приложения / С.Л. Марпл. – М.: Мир, 1990.

22. Кардиомониторы. Аппаратура непрерывного контроля ЭКГ: учеб. пособие / под ред. А.Л. Барановского. – М.: Радио и связь, 1993.

23. Кей, С.М. Современные методы спектрального анализа / С.М. Кей, С.Л. Марпл-мл. // Обзор, ТИИЭР. – 1981. – № 11.

24. Ulrych, T.J. Maximum Entropy Spectral Analysis and Autorepressive Decomposition / T.J. Ulrych, T.N. Bishop: Reviews of Geophysics and Space Physics, 13, 1 – 19 (1975).

25. Кейпон, Д. Распределение вероятности оценок пространственно-временного спектра / Д. Кейпон, Н. Гудмен // ТИИЭР – 1970. – №11. – С. 81.

26. Berkeley Design Technology (1999) Buyer's Guide ti DSP Processors. Fremont CA: Berkeley Design Technology Inc.

27. Blalock G. (1997) General-purpose µPs for DSP applications: consider the trade-offs. EDN, 23 October. – pp. 165 – 172.

28. Hacker S. (1999) Static superscalar design: a new architecture for the TigerSHARC DSP processor. Analog Devices Whitepaper.

29. Texas Instrumens (1999) TMS320C6000 Technical Brief, Literature No. SPRU197D, Texas Insatruments, Austrin TX.

30. Hayes J.P. (1998) Computer Architecture and Organization, 3rd edn. Boston MA: McGraw-Hill.

31. Macario R.C.V. (ed.) (1996) Modern Personal Radio Systems (Chapters 3, 8, 11 and 12). The Institution of Electrical Engineers, London.

32. Keith R.D.F., Beckley S. Garibaldi J.M., Westgate J., Ifeachor E.C. and Greene K.R. (1995) A multicenter comparison study of 17 experts and an intelligent computer system for managing labour using the cardiotocogram. Brit. J. Obstet. Gynaecol., 102, 688-700.

Учебное издание

ЧЕРТКОВ Валерий Михайлович
БОГУШ Рихард Петрович
МАЛЬЦЕВ Сергей Васильевич

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Учебно-методический комплекс для студентов
специальности 1-39 01 01 «Радиотехника»

Редактор *Т. В. Булах*

Дизайн обложки *В. А. Виноградовой*

Подписано в печать 21.02.2013. Формат 60×84 1/16. Бумага офсетная. Ризография
Усл. печ. л. 20,88. Уч.-изд. л. 19,4. Тираж 30 экз. Заказ 239.

Издатель и полиграфическое исполнение –
учреждение образования «Полоцкий государственный университет».

ЛИ № 02330/0548568 от 26.06.2009 ЛП № 02330/0494256 от 27.05.2009

Ул. Блохина, 29, 211440, г. Новополоцк.