

# Алгоритмы и процессоры цифровой обработки сигналов

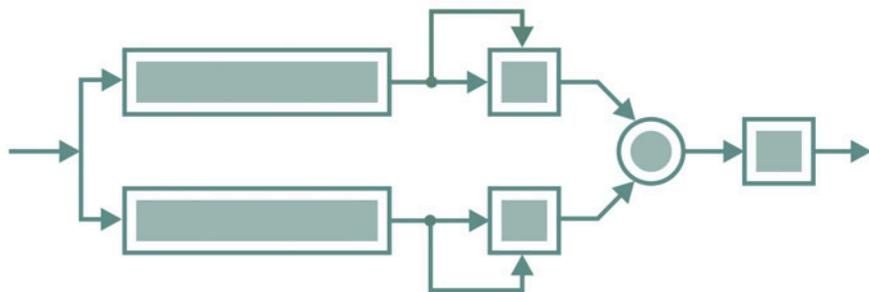


*Методы и базовые  
алгоритмы*

*Архитектура процессоров*

*Представление данных*

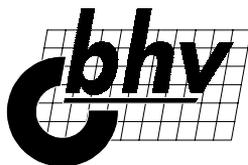
*Основы проектирования систем  
цифровой обработки сигналов*



**Алла Солонина  
Дмитрий Улахович  
Лев Яковлев**

# **Алгоритмы и процессоры цифровой обработки сигналов**

*Рекомендовано УМО по образованию в области телекоммуникаций  
в качестве учебного пособия для студентов, обучающихся  
по направлению 654400 "Телекоммуникации"*



*Санкт-Петербург*

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

УДК 681.3.06

Учебное пособие посвящено базовым алгоритмам ЦОС и архитектуре ЦПОС на примерах процессоров фирм Texas Instruments, Analog Devices и Motorola. Рассматриваются принципы построения и характеристики цифровых процессоров обработки сигналов, представление и обработка данных, команды, типы адресации операндов. Описывается работа различных устройств внутрикристальной периферии. Обсуждаются вопросы подготовки программ пользователя: этапы разработки и отладки, особенности и элементы языков ассемблера различных процессоров, состав пакетов программного обеспечения, а также использование языка С. Изложение сопровождается многочисленными примерами и иллюстрациями.

*Для студентов, преподавателей, инженеров и научных работников,  
разрабатывающих и применяющих ЦПОС*

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Наталья Таркова</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

**Солонина А. И., Улахович Д. А., Яковлев Л. А.**

Алгоритмы и процессоры цифровой обработки сигналов. — СПб.: БХВ-Петербург, 2001. — 464 с.: ил.

ISBN 5-94157-065-1

© А. И. Солонина, Д. А. Улахович, Л. А. Яковлев, 2001

© Оформление, издательство "БХВ-Петербург", 2001

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 24.08.01.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 37,41.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с диапозитивов  
в Академической типографии "Наука" РАН.  
199034, Санкт-Петербург, 9-я линия, 12.

# Содержание

<b>Введение.....</b>	<b>1</b>
<b>Глава 1. Методы и алгоритмы цифровой обработки сигналов .....</b>	<b>5</b>
1.1. Обобщенная схема цифровой обработки аналоговых сигналов .....	5
1.2. Основные направления, задачи и алгоритмы ЦОС .....	10
1.2.1. Фильтрация .....	11
1.2.2. Преобразование Гильберта.....	13
1.2.3. Дифференциатор .....	15
1.3. Цифровой спектральный анализ .....	16
1.3.1. Быстрое преобразование Фурье.....	18
1.3.2. Дискретное преобразование Хартли (ДПХ).....	20
1.3.3. Дискретное косинусное преобразование .....	21
1.4. Нелинейная обработка.....	24
1.4.1. Вычисление квадратного корня.....	24
1.4.2. Вычисление функции $\cos(\omega)$ .....	25
1.4.3. Вычисление полиномов.....	25
1.4.4. Медианные фильтры.....	26
1.4.5. Векторное квантование.....	29
1.5. Адаптивная фильтрация .....	32
1.5.1. Адаптивные фильтры .....	32
1.5.2. Линейное предсказание .....	34
1.6. Способы реализации алгоритмов ЦОС .....	39
1.6.1. Реальное время .....	40
1.6.2. Аппаратная реализация.....	42
1.6.3. Программная реализация .....	45
1.6.4. Аппаратно-программная реализация .....	47
1.7. Особенности ЦОС, влияющие на элементную базу .....	48
1.7.1. Характеристика особенностей ЦОС.....	48
1.7.2. Основные свойства ЦПОС.....	52
<b>Глава 2. Архитектура цифровых процессоров обработки сигналов.....</b>	<b>55</b>
2.1. Реализация цифровой обработки сигналов.....	55
2.2. Общие принципы построения ЦПОС и особенности их архитектуры .....	57
2.2.1. Архитектура фон Неймана и гарвардская архитектура .....	58
2.2.2. Структура процессора ЦПОС .....	60
2.2.3. Конвейерное выполнение команд.....	62

2.2.4. Аппаратная реализация программных функций.	
Параллельная работа различных функциональных узлов .....	67
2.2.5. Использование нескольких АЛУ .....	71
2.2.6. Регистровые файлы .....	73
2.2.7. Специальные методы адресации.....	74
2.2.8. Комбинированные и специализированные команды.....	76
2.2.9. Разнообразные устройства ввода/вывода и периферии .....	78
2.3. ЦПОС с фиксированной и плавающей точкой .....	79
2.4. Основные типы ЦПОС.....	80
2.4.1. Стандартные процессоры ЦПОС (Conventional DSP) .....	81
2.4.2. Улучшенные стандартные процессоры ЦПОС (Enhanced-Conventional DSP) .....	86
2.4.3. Процессоры ЦПОС с архитектурой VLIW.....	93
2.4.4. Суперскалярные процессоры .....	99
2.4.5. Гибридные процессоры .....	102
2.5. Влияние архитектуры на возможности процессора .....	105
2.6. Организация памяти ЦПОС .....	106
2.6.1. Доступ к блокам памяти. Блоки памяти.....	106
2.6.2. Внешняя память .....	111
2.6.3. Кэш .....	113
2.6.4. Защита содержимого памяти.....	114

## **Глава 3. Данные ..... 115**

3.1. Представление данных в алгоритме .....	115
3.2. Представление данных в программе .....	116
3.3. Представление данных в ЦПОС.....	117
3.4. Двоичная система счисления.....	117
3.5. Форматы данных .....	119
3.6. Формы представления данных.....	121
3.7. Представление данных с фиксированной точкой .....	122
3.7.1. Структура двойного и расширенного слова при представлении чисел с ФТ .....	123
3.7.2. Представление целых чисел .....	124
3.7.3. Представление вещественных чисел .....	126
3.7.4. Шестнадцатеричный эквивалент представления данных .....	129
3.7.5. Целочисленная и дробная арифметики .....	129
3.7.6. Коды чисел.....	131
3.7.7. Представление данных при целочисленной арифметике .....	140
3.7.8. Представление данных при дробной арифметике .....	142
3.7.9. Арифметические операции в дополнительном коде .....	143
3.7.10. Преобразование форматов в ЦПОС с фиксированной точкой.....	152
3.7.11. Диапазон, динамический диапазон и точность представления чисел с фиксированной точкой.....	156
3.7.12. Увеличение динамического диапазона и точности представления данных в ЦПОС с фиксированной точкой .....	159
3.7.13. Упакованные данные.....	160

3.8. Представление данных с плавающей точкой.....	160
3.8.1. Стандарт IEEE 754 представления данных с плавающей точкой.....	161
3.8.2. Форма представления данных с плавающей точкой.....	162
3.8.3. Форматы данных с плавающей точкой.....	163
3.8.4. Преобразование форматов в ЦПОС с плавающей точкой.....	165
3.8.5. Нормализованные числа.....	166
3.8.6. Специальные данные.....	167
3.8.7. Арифметические операции над данными с плавающей точкой.....	169
3.8.8. Диапазон, динамический диапазон и точность представления чисел с плавающей точкой.....	170
3.9. Сравнение ЦПОС с фиксированной и плавающей точками.....	171
3.10. Организация обработки данных с плавающей точкой в ЦПОС с ФТ.....	171

## **Глава 4. Операции над данными..... 175**

4.1. Операции над данными в ЦПОС с фиксированной точкой.....	175
4.1.1. Умножители и устройства МАС.....	175
4.1.2. Арифметико-логические устройства.....	176
4.1.3. Аккумуляторы.....	176
4.1.4. Сдвигатели.....	177
4.1.5. Ограничители.....	184
4.1.6. Переполнение в аккумуляторе.....	184
4.1.7. Округление результатов.....	187
4.2. Операции над данными в ЦПОС с плавающей точкой.....	189
4.2.1. Умножители.....	189
4.2.2. Арифметико-логические устройства.....	190
4.2.3. Сдвигатели.....	190
4.2.4. Округление данных с плавающей точкой.....	191
4.2.5. Особые случаи при обработке данных с плавающей точкой.....	192

## **Глава 5. Адресация..... 195**

5.1. Прямая адресация.....	197
5.1.1. Прямое указание адресов.....	197
5.1.2. Прямое указание источников и приемников.....	203
5.1.3. Прямая адресация переходов.....	206
5.2. Косвенная адресация.....	207
5.2.1. Модификация адреса.....	211
5.2.2. Типы арифметики.....	221
5.2.3. Циклическая адресация.....	222
5.2.4. Бит-реверсивная адресация.....	229
5.2.5. Косвенная адресация переходов.....	233
5.3. Непосредственная адресация.....	233

## **Глава 6. Система команд..... 235**

6.1. Форматы команд.....	236
6.2. Структура слова команды.....	237

6.2.1. Структура слова команды в процессорах со стандартной архитектурой .....	238
6.2.2. Структура слова команды в процессорах с одновременным выполнением группы команд .....	242
6.3. Синтаксис команд.....	246
6.3.1. Синтаксис команд в процессорах со стандартной архитектурой.....	246
6.3.2. Синтаксис команд в процессорах с одновременным выполнением группы команд .....	252
6.3.3. Операции над упакованными данными .....	253
6.4. Формирование системы команд.....	254
6.5. Группы команд.....	255
6.5.1. Команды пересылок.....	255
6.5.2. Команды арифметических операций .....	257
6.5.3. Команды логических операций .....	261
6.5.4. Комбинированные команды .....	262
6.5.5. Команды бит-манипуляций.....	264
6.5.6. Команды управления .....	264
6.5.7. Особенности команд с плавающей точкой .....	270
<b>Глава 7. Прерывания .....</b>	<b>271</b>
7.1. Источники прерывания.....	272
7.2. Средства управления прерываниями .....	273
7.3. Типы прерываний .....	278
7.4. Инициализация процессора для работы в состоянии прерывания.....	279
7.5. Обслуживание прерываний.....	280
7.6. Состояние ожидания.....	282
<b>Глава 8. Периферийные устройства .....</b>	<b>283</b>
8.1. Основные понятия и определения.....	283
8.1.1. Функциональный интерфейс .....	284
8.2. Командеры .....	290
8.3. Генератор задержек доступа к памяти .....	294
8.4. Таймеры.....	296
8.4.1. Таймеры на декремент.....	296
8.4.2. Таймеры на инкремент .....	300
8.4.3. Работа таймера.....	301
8.5. Синхронизация портов.....	303
8.5.1. Генераторы тактовых частот.....	304
8.5.2. Синхронизация синхронных последовательных портов .....	309
8.5.3. Синхронизация асинхронных портов .....	313
8.6. Синхронные последовательные порты .....	317
8.6.1. Базовый синхронный порт.....	317
8.6.2. Буферизированный последовательный порт .....	325
8.7. Контроллер прямого доступа к памяти .....	332
8.7.1. Синхронизация каналов ПДП .....	334
8.7.2. Генерация адреса ПДП .....	335

8.7.3. Система приоритетов ПДП .....	336
8.8. Порт с временным разделением каналов .....	337
8.9. Многоканальный буферизированный последовательный порт .....	341
8.9.1. Устройство управления и синхронизации МкБПП .....	344
8.9.2. Устройство многоканального выбора .....	345
8.10. Хост-порт .....	346
<b>Глава 9. Подготовка программ пользователя. Языки программирования .....</b>	<b>349</b>
9.1. Этапы разработки программы .....	349
9.2. Языки ассемблера .....	352
9.2.1. Особенности языка .....	352
9.2.2. Структура программы .....	353
9.2.3. Мнемонические и алгебраические ассемблеры .....	358
9.2.4. Основные конструкции ассемблера .....	361
9.2.5. Средства макроассемблера .....	364
9.2.6. Средства организации стандартных структур .....	367
9.2.7. Совместимость ассемблеров различных процессоров .....	368
9.3. Получение исполняемой программы. Состав пакетов программного обеспечения процессоров DSP .....	370
9.3.1. Абсолютные и перемещаемые программные модули .....	372
9.3.2. Компоновка .....	373
9.3.3. Отладка и тестирование программы .....	375
9.3.4. Библиотеки функций и информационная поддержка .....	376
9.3.5. Использование интегрированных оболочек для подготовки и моделирования программ ЦОС .....	377
9.3.6. Matlab .....	378
9.4. Рекомендуемый путь построения программы .....	379
9.5. Размещение программ в памяти .....	382
9.5.1. Секции программы и блоки памяти .....	382
9.5.2. Начальная загрузка программы .....	383
9.5.3. Оверлейные программы .....	384
9.6. Языки С/С++ и архитектуры, дружественные к языку С .....	385
<b>Глава 10. Средства разработки и отладки систем цифровой обработки сигналов .....</b>	<b>391</b>
10.1. Аппаратные средства отладки .....	392
10.1.1. Внутрисхемные эмуляторы-приставки .....	393
10.1.2. Внутрикристалльные средства отладки .....	394
10.1.3. Проверочные модули .....	405
10.2. Программные средства отладки .....	407
10.2.1. Симуляторы системы команд .....	407
10.2.2. Отладчики .....	409
10.2.3. Интегрированные отладочные средства .....	411
<b>Глава 11. Разновидности и характеристики ЦПОС .....</b>	<b>413</b>
11.1. Квалификационные параметры и характеристики ЦПОС .....	413

11.2. Сравнение производительности процессоров.....	417
11.3. Разновидности ЦПОС с точки зрения назначения.....	421
<b>Приложение 1. Сравнительная таблица параметров процессоров .....</b>	<b>427</b>
<b>Приложение 2. Описания процессоров на русском языке.....</b>	<b>432</b>
<b>Приложение 3. Источники информации .....</b>	<b>433</b>
Периодические издания .....	433
Журналы на русском языке .....	433
Журналы на английском языке .....	433
Фирмы — производители ЦПОС.....	434
Поставщики продукции и консультанты .....	434
Разработчики систем.....	435
Web-сайты с полезной информацией .....	435
<b>Приложение 4. Маркировка процессоров TMS320 фирмы TI .....</b>	<b>437</b>
<b>Приложение 5. Принятые сокращения .....</b>	<b>438</b>
<b>Приложение 6. Список литературы.....</b>	<b>442</b>
<b>Предметный указатель .....</b>	<b>445</b>

# Введение

Цифровые процессоры обработки сигналов (ЦПОС) или их равнозначное название — цифровые сигнальные процессоры (ЦСП или просто сигнальные процессоры), англоязычное сокращение — DSP (Digital Signal Processor), предназначены для реализации алгоритмов цифровой обработки сигналов (ЦОС) и систем управления в реальном времени. ЦПОС являются одной из разновидностей микропроцессоров, четко выделившейся в отдельное направление развития электронной индустрии.

Первый ЦПОС TMS320C10 с производительностью 5 млн операций в секунду фирмы Texas Instruments появился в начале 80-х годов XX века. В последующие годы аналогичная продукция была разработана другими фирмами: Analog Devices, Motorola, Lucent Technologies и т. д. Современный рынок представлен большим разнообразием ЦПОС с производительностью, достигающей нескольких миллиардов операций в секунду. Фантастические, непрерывно возрастающие темпы совершенствования ЦПОС в свою очередь активизируют разработку новых методов и алгоритмов ЦОС все возрастающей вычислительной сложности. В настоящее время динамически развивающаяся взаимосвязь "алгоритм ЦОС ↔ ЦПОС" в значительной мере определяет научно-технический прогресс в области телекоммуникаций, включая глобальную сеть Internet, и систем мультимедиа.

В условиях, когда жесткая конкуренция диктует высочайшие темпы производства ЦПОС и реализации на их базе новых алгоритмов ЦОС, появилась возможность, благодаря сети Internet, одновременного, столь же быстрого распространения соответствующей технической информации. В последнее время руководства для пользователей ЦПОС все реже выпускаются в виде книг, все чаще их размещают на сайтах фирм и дублируют на компакт-дисках. Поэтому основным способом изучения ЦПОС становится *индивидуальная и самостоятельная* работа с информацией, предоставляемой фирмами в Сети.

Главная трудность для начинающих самостоятельно изучать ЦПОС — колоссальный объем информации в Internet. Можно потратить массу времени и усилий, а в результате получить лишь набор фрагментарных знаний, не дающий общего представления об основах построения ЦПОС, причинах, по которым они выделены в отдельный класс микропроцессоров, признаках, отличающих архитектуры различных ЦПОС, и т. д. В этих условиях очень важна начальная (базовая) подготовка.

В настоящее время отечественных книг по ЦПОС очень немного и предназначены они для изучения конкретных процессоров (или базовых семейств).

Учебные пособия по изучению основ ЦПОС отсутствуют. Существует американский вариант подобной книги [4], приводимый в списке литературы.

Авторы полагают, что базовая подготовка должна предусматривать:

- знакомство с типовыми методами и алгоритмами ЦОС, оценку их вычислительной сложности, анализ возможности разработки на базе ЦПОС соответствующих устройств ЦОС, работающих в реальном времени;
- изучение единых или сходных принципов формирования архитектуры ЦПОС, основных вариантов и характерных особенностей их различных реализаций.

Авторы считают, что такая подготовка, *поддерживаемая данным пособием*, будет способствовать развитию общей эрудиции в области ЦОС и ЦПОС, позволит создать цельное представление о принципах организации ЦПОС и познакомит с многообразием их новейших технических решений.

Пособие может стать основой для последующего более обстоятельного изучения конкретных ЦПОС различных фирм. Оно научит понимать, какие характеристики, в каком приложении являются определяющими, ориентироваться в оценке сильных и слабых сторон различных ЦПОС и принимать правильные компромиссные решения при выборе сигнальных процессоров для требуемых приложений.

Данное пособие может использоваться при изучении дисциплин, включенных в государственные стандарты специальностей направления "Телекоммуникации" (654400):

- вычислительная техника и информационные технологии (раздел "Сигнальные процессоры и их применение в системах цифровой обработки сигналов");
- микропроцессоры и цифровая обработка сигналов;
- цифровая обработка сигналов и сигнальные процессоры в системах подвижной радиосвязи.

Пособие может быть также полезно для инженеров и разработчиков цифровых систем связи на основе ЦПОС.

Изложение основ ЦПОС сопровождается многочисленными примерами общего или конкретного характера в зависимости от иллюстрируемого положения. В приводимых примерах в основном использовались процессоры фирм Texas Instruments, Analog Devices и Motorola, получившие на российском рынке наибольшее распространение. Ссылки на источники информации по продукции данных фирм, а также о российских процессорах, содержатся в *приложении 3*. При написании пособия использовались многочисленные руководства пользователя по конкретным процессорам. Однако все эти руководства не включены в список литературы из-за их большого количества и постоянной модификации.

Читатель заметит определенные повторы материала в разных главах. Авторы сохранили их намеренно в случаях, когда желательнее не отвлекать внимание читателя на поиск информации в других главах, либо, когда уместно подчеркнуть, что один и тот же вопрос может рассматриваться в разном контексте.

Учебное пособие содержит 11 глав; главы 1, 8, 10 написаны Д. А. Улаховичем, главы 2, 9, 11, приложения — Л. А. Яковлевым, главы 3, 4, 5, 6, 7 — А. И. Солониной.

- В *главе 1* рассматриваются основные направления ЦОС, приводятся примеры базовых методов и алгоритмов ЦОС, анализируются методы их реализации и делается вывод о требованиях, которым должны отвечать цифровые процессоры обработки сигналов для обеспечения эффективной реализации алгоритмов ЦОС.
- *Глава 2* посвящена принципам построения и архитектуре ЦПОС; в ней также рассматриваются основные типы существующих архитектур и их сравнение, организация памяти в ЦПОС.
- *Глава 3* посвящена подробному знакомству с представлением данных в ЦПОС с фиксированной и плавающей точками и арифметическим операциям с данными.
- В *главе 4* рассматриваются специальные операции с данными: округление, арифметика насыщения при переполнении, обработка особых случаев для данных с плавающей точкой.
- В *главе 5* рассматриваются основные типы адресации операндов.
- В *главе 6* изучаются принципы формирования системы команд и основные группы команд.
- В *главе 7* рассматривается работа процессора в состоянии прерывания.
- *Глава 8* посвящена изучению принципов работы разнообразных устройств внутрикристальной периферии: компандеров, таймеров, контроллеров прямого доступа к памяти, портов, синхронизации и др.; приводятся примеры процессоров, в которых применяются эти устройства.
- *Глава 9* посвящена вопросам подготовки программ пользователя: этапы разработки, используемые для программирования ЦПОС языки, общие особенности и элементы языков ассемблера различных процессоров, состав пакетов программного обеспечения процессоров, а также использование языка С.
- В *главе 10* рассматриваются программные и аппаратные средства разработки устройств ЦОС: симуляторы, внутрисхемные и внутрикристальные эмуляторы, отладочные модули, приводятся конкретные примеры.
- *Глава 11* посвящена рассмотрению основных характеристик ЦПОС, сравнению производительности различных процессоров (по зарубежным ма-

териалам), а также существующим разновидностям ЦПОС с точки зрения назначения.

- В приложениях приводятся сравнительная таблица параметров некоторых упоминаемых в пособии процессоров, список книг на русском языке, посвященных описаниям различных ЦПОС, а также список источников информации: периодические издания, сайты дистрибьюторов и фирм-разработчиков.

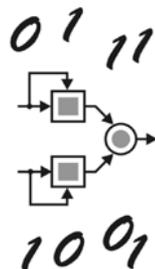
Предполагается, что читатели данного пособия уже знакомы с основами микропроцессорных систем, цифровой и вычислительной техники.

При написании пособия использован опыт чтения лекций и проведения лабораторно-практических занятий по соответствующим курсам на кафедре цифровой обработки сигналов Государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича.

Авторы выражают благодарность заведующему кафедрой ЦОС и руководителю Центра ЦОС профессору Артуру Абрамовичу Ланне за помощь в подготовке книги.

Все предложения и замечания по книге просим присылать по адресу: 191186, Санкт-Петербург, наб. р. Мойки, 61, Государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, кафедра цифровой обработки сигналов или электронному адресу **info@dsp-sut.spb.ru**.

# Глава 1



## Методы и алгоритмы цифровой обработки сигналов

Прежде чем изучать принципы, лежащие в основе конструкции цифровых процессоров обработки сигналов (ЦПОС), целесообразно рассмотреть базовые методы и алгоритмы цифровой обработки сигналов (ЦОС), которые оказали и продолжают оказывать непосредственное влияние как на всю элементную базу ЦОС, так, в частности, и на архитектуру ЦПОС. Знание методов и алгоритмов ЦОС, их особенностей, сложности, возможностей применения в разнообразных задачах позволяет оценить вычислительные проблемы, свойственные данному алгоритму в конкретном приложении, и выбрать оптимальную элементную базу для его реализации. Кроме того, это знание помогает понять, почему развитие цифровой вычислительной техники пришло к цифровым процессорам обработки сигналов.

Общие же проблемы ЦОС, характерные для любых алгоритмов, нетрудно обнаружить на обобщенной схеме цифровой обработки аналоговых сигналов, которую и рассмотрим.

### 1.1. Обобщенная схема цифровой обработки аналоговых сигналов

Цифровая обработка принципиально связана с представлением любого сигнала в виде последовательности чисел. Это означает, что исходный аналоговый сигнал должен быть преобразован в исходную последовательность чисел (рис. 1.1), которая вычислителем по заданному алгоритму преобразуется в новую последовательность, однозначно соответствующую исходной. Из полученной новой последовательности формируется результирующий аналоговый сигнал.

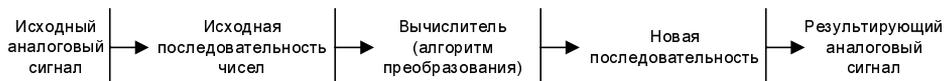


Рис. 1.1. Этапы цифровой обработки

Перечисленные преобразования должны происходить по определенным правилам, смысл которых отображен на рис. 1.2, где показаны основные элементы обобщенной схемы цифровой обработки аналоговых сигналов:

- аналоговый антиэлайсинговый фильтр низких частот (АФНЧ);
- аналого-цифровой преобразователь (АЦП);
- устройство цифровой обработки сигналов (вычислитель);
- цифро-аналоговый преобразователь (ЦАП);
- аналоговый сглаживающий фильтр низких частот (СФНЧ).

На этом же рисунке приведен пример цифровой обработки аналогового сигнала и временные и спектральные диаграммы на входе и выходе основных элементов. Устройство, объединяющее АФНЧ и АЦП, называется *кодером*. Как следует из рис. 1.2, а, кодер формирует последовательность чисел, соответствующую обрабатываемому аналоговому сигналу. Устройство, объединяющее ЦАП и СФНЧ, называется *декодером*. Декодер по принятому цифровому сигналу формирует аналоговый сигнал, т. е. производит преобразования, обратные происходившим в кодере.

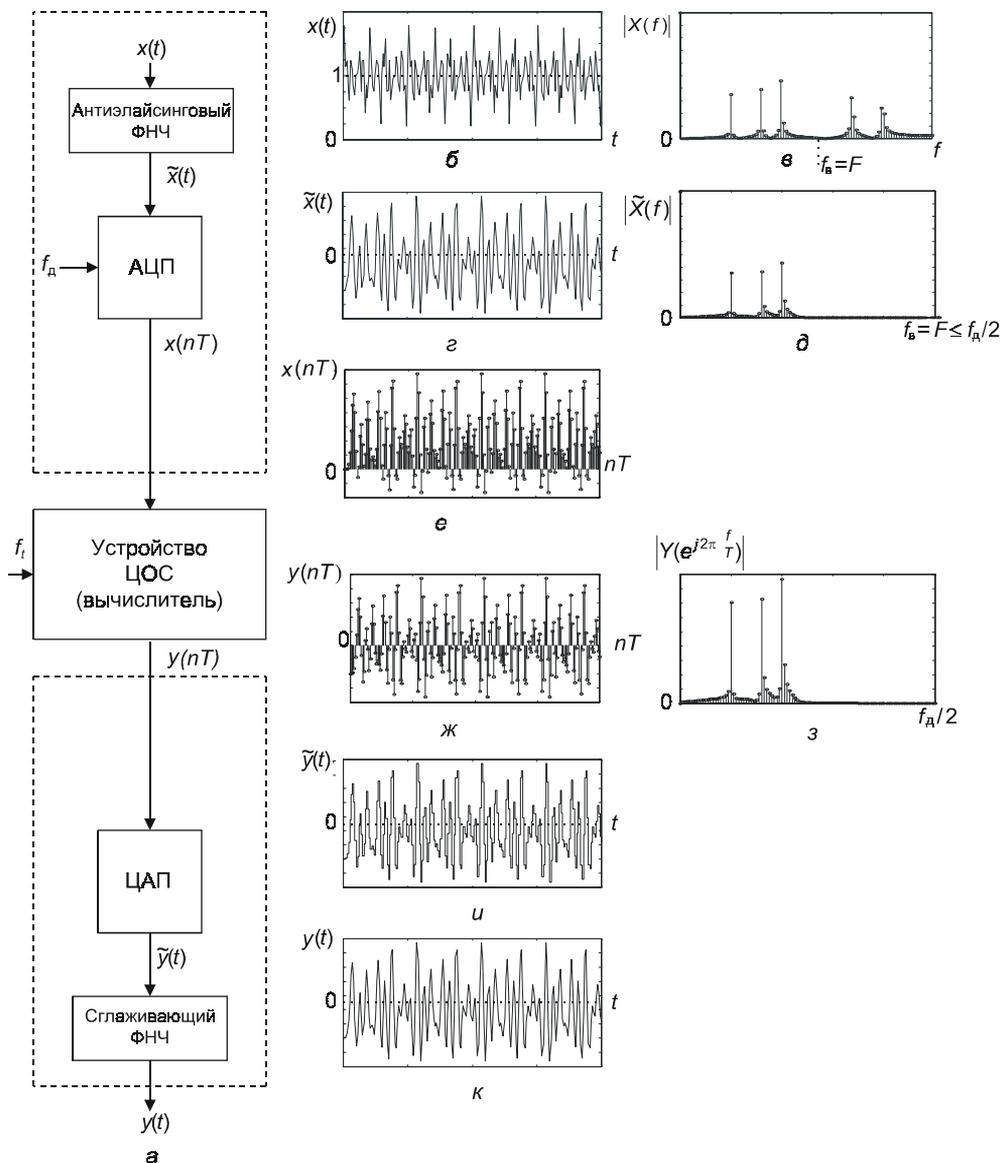
На вход системы поступает ограниченный по длительности сигнал  $x(t)$ , имеющий в своем составе постоянную составляющую  $x_0 = 1$ , явно выраженные высокочастотные составляющие, которые несколько затемяют сигнал небольшими всплесками (рис. 1.2, б). Спектр амплитуд, или просто спектр, такого сигнала представлен на рис. 1.2, в. В силу конечной длительности сигнала его спектр бесконечен!

Бесконечность спектра является препятствием для преобразования сигнала  $x(t)$  в цифровую форму, что связано с природой аналого-цифрового преобразования, осуществляемого в два этапа: дискретизации по времени и квантования по уровню.

*Дискретизация по времени* (или *дискретизация*) представляет собой процедуру взятия мгновенных значений сигнала  $x(t)$  через равные промежутки времени  $T$ . Мгновенные значения  $x(nT)$  называются *выборками*, или *отсчетами*, время  $T$  — *периодом дискретизации*,  $n$  указывает порядковый номер отсчета. Ясно, что чем чаще брать отсчеты, т. е. чем меньше период дискретизации  $T$ , тем точнее последовательность отсчетов  $x(nT)$  будет отображать исходный сигнал  $x(t)$ . Период дискретизации  $T$  определяет *частоту дискретизации*.

$$f_d = \frac{1}{T}; \quad T = \frac{1}{f_d}, \quad (1.1)$$

откуда чем меньше  $T$ , тем выше  $f_d$ . С другой стороны, чем выше частота дискретизации, тем труднее вычислителю выполнять большое количество операций над отсчетами в темпе их поступления на переработку и тем сложнее должно быть его устройство.



**Рис. 1.2.** Обобщенная схема цифровой обработки аналоговых сигналов

Таким образом, точность представления сигнала требует увеличивать  $f_d$ , а стремление сделать вычислитель как можно более простым приводит к желанию понизить  $f_d$ . Однако существует ограничение на минимальное значение  $f_d$ : для полного восстановления непрерывного сигнала по его отсче-

там  $x(nT)$  необходимо и достаточно, чтобы частота дискретизации  $f_d$  была, как минимум, в два раза больше наивысшей частоты  $F$  в спектре передаваемого сигнала  $x(t)$ , т. е.

$$f_d \geq 2F, \quad T \leq 1/2F. \quad (1.2)$$

Отсюда следует, что при бесконечном спектре, когда  $F \rightarrow \infty$ , дискретизация невозможна. Тем не менее, в спектре любого конечного сигнала есть такие высшие составляющие, которые, начиная с некоторой верхней частоты  $f_v$ , имеют незначительные амплитуды, и потому ими можно пренебречь без заметного искажения самого сигнала. Значение  $f_v$  определяется конкретным типом сигнала и решаемой задачей. Например, для стандартного телефонного сигнала  $f_v = 3,4$  кГц минимальная стандартная частота его дискретизации  $f_d = 8$  кГц. Ограничение спектра до частоты  $F = f_v$  осуществляется аналоговым ФНЧ, получившим название *антиэлайсингового*, поскольку он предотвращает искажения спектра типа *наложения* (*aliasing*), которые возникают в спектре цифрового сигнала при недостаточной частоте дискретизации. Во временной области эффект наложения означает необратимую потерю возможности точного восстановления аналогового сигнала по его отсчетам.

Антиэлайсинговый фильтр формирует аналоговый сигнал со значительно подавленными верхними частотными составляющими (рис. 1.2, *з, д*) в полосе задерживания, начиная с частоты  $F = f_v$ . Это дает основание считать сигнал практически ограниченным по частоте и неподверженным эффекту наложения при частоте дискретизации не менее  $2F$ .

*Квантование отсчетов по уровням* (или *квантование*) производится с целью формирования последовательности чисел: весь диапазон изменения величины отсчетов разбивается на некоторое количество дискретных уровней (рис. 1.2, *е*), и каждому отсчету по определенному правилу присваивается значение одного из двух ближайших уровней квантования, между которыми оказывается данный отсчет. В результате получается последовательность чисел  $x(nT) = x(n)$ , представляемых в двоичном коде. Количество уровней определяется разрядностью  $b$  АЦП; так, если  $b = 3$ , всего можно иметь  $k = 2^b = 2^3 = 8$  уровней квантования, а минимальное и максимальное значения отсчетов равны соответственно  $0 \Leftrightarrow 000$  и  $7 \Leftrightarrow 111$ . Ясно, что квантованный отсчет отличается от выборки  $x(nT)$ . Это отличие выражается ошибкой квантования

$$\varepsilon_{кв} = x_{ц}(nT) - x(nT), \quad (1.3)$$

которая тем больше, чем меньше  $b$ . Максимальная ошибка квантования при использовании округления в качестве приближения равна половине шага квантования  $Q$

$$\max|\varepsilon_{кв}| = Q/2, \quad \text{где } Q = q_1 = 2^{-b}. \quad (1.4)$$

Отсюда следует, что чем больше разрядность  $b$  АЦП, тем точнее представляется отсчет, но тем сложнее и дороже оказывается АЦП, который необходим для решения поставленной задачи. Современные АЦП имеют разрядность от 8 до 20.

Последовательность  $x(nT) = x(n)$  поступает на вычислитель, который по заданному алгоритму каждому отсчету  $x(n)$  ставит в однозначное соответствие выходной отсчет  $y(nT) = y(n)$

$$x(n) \Rightarrow y(n). \quad (1.5)$$

Результатом переработки исходного сигнала является новая цифровая последовательность — цифровой сигнал (рис. 1.2, ж), уже не имеющий постоянной составляющей и существенно отличающийся от  $x(n)$ . Амплитудный спектр (рис. 1.2, з) оказывается более обостренным на частотах, близких к частоте  $f_d/4$ . Количество операций (умножений, сложений, пересылок и т. д.) для получения одного отсчета  $y(n)$  может исчисляться тысячами, поэтому вычислитель должен работать на более высокой тактовой частоте  $f_\tau$ , чтобы успеть произвести все необходимые действия до поступления очередного отсчета  $x(n)$ , т. е. какой бы сложности ни был алгоритм, время переработки  $t_{\text{пер}}$  не должно превышать периода дискретизации  $T$

$$t_{\text{пер}} \leq T. \quad (1.6)$$

Но это может быть обеспечено лишь в случае, когда тактовая частота  $f_\tau$  вычислителя существенно превышает частоту дискретизации  $f_d$ . Именно при этих условиях возможна работа вычислителя в реальном времени, т. е. в темпе поступления входных отсчетов. Например, при обработке стандартного телефонного сигнала с частотой дискретизации 8 кГц для обеспечения работы вычислителя в реальном времени тактовая частота должна быть равной, по крайней мере, 6 МГц, как в процессоре первого поколения TMS320C10.

Полученные выходные отсчеты подаются на цифро-аналоговый преобразователь (ЦАП), формирующий ступенчатый сигнал  $\tilde{y}(t)$  — рис. 1.2, и, который затем с помощью сглаживающего фильтра НЧ преобразуется в аналоговый непрерывный сигнал  $y(t)$  — рис. 1.2, к.

Из всего сказанного вытекает ряд ограничений, влияющих на характер и выбор элементной базы для реализации вычислителя:

- разрядность регистров должна быть большой и превышать разрядность ЦАП во избежание дополнительных ошибок при округлении результатов вычислений;

- тактовая частота, на которой работает вычислитель, должна в сотни раз превосходить частоту дискретизации, если предъявляется требование реального времени;
- малое энергопотребление;
- компактность.

## 1.2. Основные направления, задачи и алгоритмы ЦОС

Среди многочисленных задач, решаемых на базе ЦОС, можно выделить группу наиболее полно характеризующих как традиционные, так и нетрадиционные области применения ЦОС. Каждая задача — в зависимости от конкретного приложения — может решаться с использованием различных методов и алгоритмов; например, задача выделения сигнала из помех может решаться методами линейной, адаптивной и нелинейной фильтрации. В свою очередь, цифровая линейная фильтрация может быть осуществлена алгоритмами КИХ- или БИХ-фильтрации.

В настоящее время выделяют следующие основные направления ЦОС (табл. 1.1):

- линейная фильтрация;
- спектральный анализ;
- частотно-временной анализ;
- адаптивная фильтрация;
- нелинейная обработка;
- многоскоростная обработка.

**Таблица 1.1.** Основные задачи ЦОС

№ п/п	Направление	Примеры задач
1	Линейная фильтрация	Селекция сигнала в частотной области; синтез фильтров, согласованных с сигналами; частотное разделение каналов; цифровые преобразователи Гильберта и дифференциаторы; корректоры характеристик каналов
2	Спектральный анализ	Обработка речевых, звуковых, сейсмических, гидроакустических сигналов; распознавание образов
3	Частотно-временной анализ	Компрессия изображений, гидро- и радиолокация, разнообразные задачи обнаружения

Таблица 1.1 (окончание)

№ п/п	Направление	Примеры задач
4	Адаптивная фильтрация	Обработка речи, изображений, распознавание образов, подавление шумов, адаптивные антенные решетки
5	Нелинейная обработка	Вычисление корреляций, медианная фильтрация; синтез амплитудных, фазовых, частотных детекторов, обработка речи, векторное кодирование
6	Многоскоростная обработка	Интерполяция (увеличение) и децимация (уменьшение) частоты дискретизации в многоскоростных системах телекоммуникации, аудиосистемах

### 1.2.1. Фильтрация

Фильтрация может осуществляться с помощью цифровых фильтров (ЦФ), описываемых во временной области линейными разностными уравнениями вида:

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i); \quad (1.7)$$

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{k=1}^{M-1} a_k y(n-k), \quad (1.8)$$

где  $x(n)$  — отсчеты воздействия;  $y(n)$  — отсчеты реакции;  $\{b_i, a_k\}$  — вещественные коэффициенты, полностью определяющие свойства ЦФ;  $M$  и  $N$  — константы, задающие сложность ЦФ;  $x(n-i)$  и  $y(n-k)$  — отсчеты воздействия и реакции, задержанные на  $i$  и  $k$  периодов дискретизации  $T$  соответственно.

Фильтр, описываемый выражением (1.7), называют *нерекурсивным*, или *КИХ-фильтром* (фильтр с конечной импульсной характеристикой); фильтр, описываемый выражением (1.8), называют *рекурсивным*, или *БИХ-фильтром* (фильтр с бесконечной импульсной характеристикой).

Передаточные функции КИХ- и БИХ-фильтров определяются из (1.7) и (1.8) с помощью Z-преобразования и имеют вид соответственно

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i}; \quad (1.9)$$

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}, \quad (1.10)$$

откуда после подстановки  $z = e^{j\omega T} = e^{j\hat{\omega}}$  получают комплексные частотные характеристики

$$H(e^{j\hat{\omega}}) = \sum_{i=0}^{N-1} b_i e^{-j\hat{\omega}i}; \quad (1.11)$$

$$H(e^{j\hat{\omega}}) = \frac{\sum_{i=0}^{N-1} b_i e^{-j\hat{\omega}i}}{1 + \sum_{k=1}^{M-1} a_k e^{-j\hat{\omega}k}}. \quad (1.12)$$

Из (1.11) и (1.12) нетрудно получить выражения для амплитудно-частотной характеристики (АЧХ)  $A(\hat{\omega})$  и фазо-частотной характеристики (ФЧХ)  $\varphi(\hat{\omega})$ :

$$A(\hat{\omega}) = |H(e^{j\hat{\omega}})|, \quad (1.13)$$

$$\varphi(\hat{\omega}) = \arg|H(e^{j\hat{\omega}})|, \quad (1.14)$$

или в явном виде

$$A(\hat{\omega}) = \frac{\sqrt{\left[ \sum_{i=0}^{N-1} b_i \cos(\hat{\omega}i) \right]^2 + \left[ \sum_{i=0}^{N-1} b_i \sin(\hat{\omega}i) \right]^2}}{\sqrt{\left[ \sum_{k=0}^{M-1} a_k \cos(\hat{\omega}k) \right]^2 + \left[ \sum_{k=0}^{M-1} a_k \sin(\hat{\omega}k) \right]^2}}; \quad (1.15, a)$$

$$\varphi(\hat{\omega}) = \arctg \frac{-\sum_{i=0}^{N-1} b_i \sin(\hat{\omega}i)}{\sum_{i=0}^{N-1} b_i \cos(\hat{\omega}i)} - \arctg \frac{-\sum_{k=0}^{M-1} a_k \sin(\hat{\omega}k)}{\sum_{k=0}^{M-1} a_k \cos(\hat{\omega}k)}. \quad (1.15, б)$$

Связь между воздействием и реакцией фильтра устанавливается не только разностными уравнениями, но и с помощью *свертки*

$$y(n) = \sum_{i=0}^K h(i)x(n-i) = \sum_{i=0}^K x(i)h(n-i), \quad (1.16)$$

где  $h(k)$  — импульсная характеристика фильтра. Для КИХ-фильтра  $K = N - 1$ , отсчеты импульсной характеристики равны коэффициентам фильтра  $h(i) = b_i$ ; для БИХ-фильтра  $K = \infty$ , а отсчеты импульсной характеристики определяются через коэффициенты  $a_k$  и  $b_i$  сложным образом.

Из (1.7), (1.8) и (1.16) видно, что для вычисления результатов фильтрации необходимо многократно выполнять следующие операции:

- сложение;
- вычитание;
- умножение;
- сдвиг, реализующий задержку сигнала на один период дискретизации.

Перечисленные операции называются *основными* (или *базовыми*) *операциями*.

## 1.2.2. Преобразование Гильберта

Среди задач обработки сигналов важное место принадлежит модуляции и демодуляции узкополосных сигналов и задачам сдвига частоты (например, в модемах в режиме подстройки), что характерно для систем радиосвязи. Примером такой задачи является демодуляция однополосного сигнала, который получается выделением одной из боковых полос амплитудно-модулированного сигнала. Результатом демодуляции является низкочастотный сигнал, представляющий собой *огибающую* узкополосного сигнала. Демодулируемый сигнал  $x(n)$  можно представить в комплексном виде:

$$\dot{x}(n) = x(n) + j\tilde{x}(n); \quad (1.17)$$

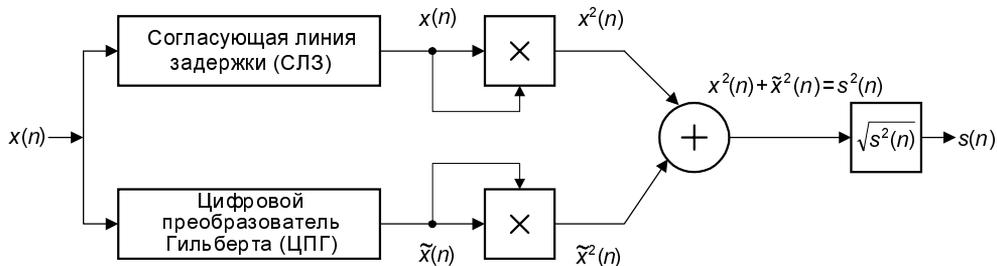
$$x(n) = s(n) \cos \varphi(\omega n); \quad (1.18)$$

$$\tilde{x}(n) = s(n) \sin \varphi(\omega n), \quad (1.19)$$

где  $x(n)$  — вещественный сигнал;  $\tilde{x}(n)$  — мнимый сигнал;  $s(n)$  — огибающая сигнала  $x(n)$ , которая, как следует из (1.17), определяется по формуле

$$s(n) = \sqrt{x^2(n) + \tilde{x}^2(n)}. \quad (1.20)$$

Из (1.18) и (1.19) видно, что  $x(n)$  и  $\tilde{x}(n)$  находятся в квадратуре относительно друг друга, т. е. их фазы отличаются на  $\pi/2$ . Следовательно, необходимо иметь фазовращатель на  $\pi/2$ . Такие сигналы называются сопряженными по Гильберту, а устройство, формирующее пару сопряженных сигналов, называется *цифровым преобразователем Гильберта* (ЦПГ) (рис. 1.3), который позволяет организовать вычисление огибающей  $s(n)$  сигнала  $x(n)$ . Согласующая линия задержки (СЛЗ) обеспечивает временное согласование сигналов  $x(n)$  и  $\tilde{x}(n)$ .

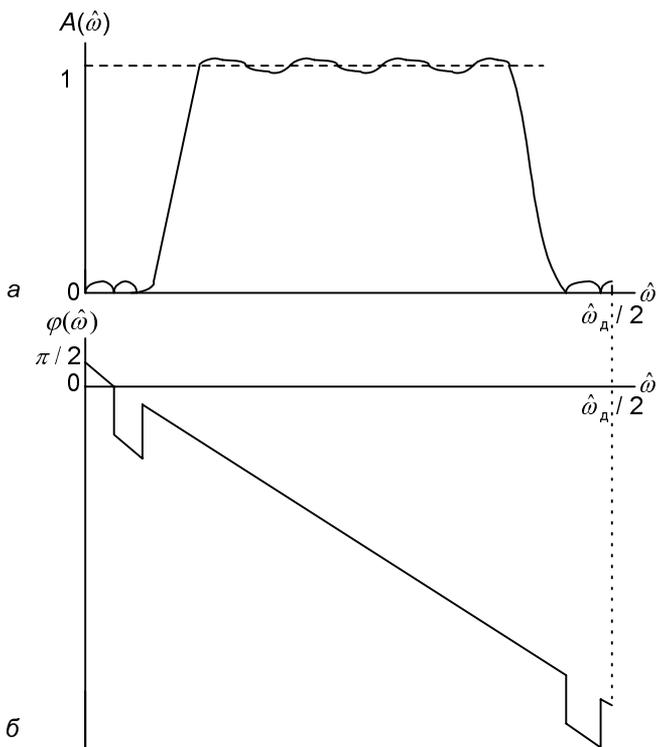


**Рис. 1.3.** Структурная схема выделения огибающей  $s(n)$  сигнала  $x(n)$

Собственно ЦПГ представляет собой полосовой КИХ-фильтр с линейной ФЧХ, имеющий антисимметричную импульсную характеристику нечетной длины. ФЧХ такого фильтра имеет постоянную составляющую  $\pi/2$  (рис. 1.4, а, б).

Антисимметричность означает

$$h(n) = -h(N - 1 - n). \quad (1.21)$$



**Рис. 1.4.** Частотные характеристики ЦПГ: а — АЧХ, б — ФЧХ

Структурная схема ЦПГ изображена на рис. 1.5, где каждый второй коэффициент (или отсчет импульсной характеристики) равен нулю. Коэффициенты ЦПГ рассчитываются по формулам:

$$b(n) = h(n) = \begin{cases} \frac{2 \sin 2(\pi n / 2)}{\pi}, & n \neq 0; \\ 0, & n = 0. \end{cases} \quad (1.22)$$

В качестве СЛЗ используются первые  $(N - 1)/2$  регистров всей линии задержки фильтра, содержащей  $N$  регистров.

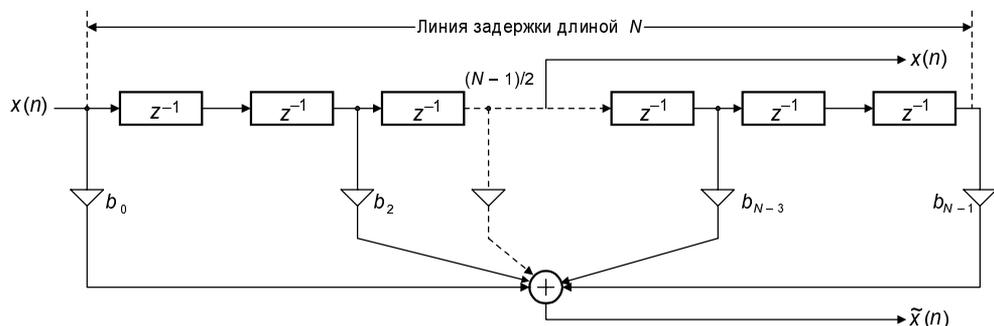


Рис. 1.5. Структурная схема цифрового преобразователя Гильберта

### 1.2.3. Дифференциатор

Дифференциаторы — это КИХ-фильтры с линейной ФЧХ, имеющие антисимметричную импульсную характеристику четной длины, линейно возрастающую АЧХ в рабочем диапазоне частот  $0 \leq \omega \leq \omega_p$  и постоянную составляющую ФЧХ, равную  $\pi/2$  (рис. 1.6).

Свое название такие КИХ-фильтры получили благодаря свойству дифференцирования сигнала. Покажем это.

На рис. 1.6 видно, что отношение приращения АЧХ  $\Delta A(\omega)$  к приращению частоты  $\Delta \omega$  постоянно

$$K = \frac{\Delta A(\omega)}{\Delta \omega} = \text{const}. \quad (1.23)$$

При устремлении  $\Delta \omega \rightarrow 0$  получаем

$$A'(\omega) = \frac{dA(\omega)}{d\omega}, \quad (1.24)$$

откуда

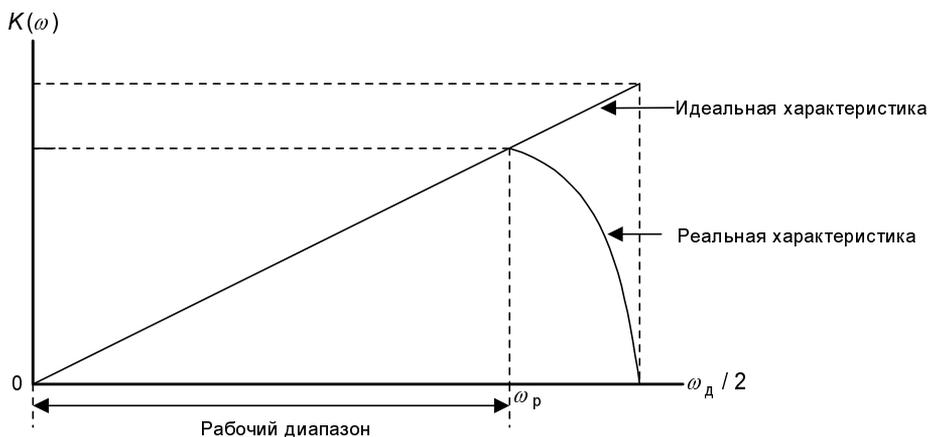
$$dA(\omega) = A'(\omega)d\omega. \quad (1.25)$$

Аналогично можно записать для сигнала  $s(\omega)$ :

$$ds(\omega) = s'(\omega)d\omega. \quad (1.26)$$

Последнее равенство означает, что сигнал на выходе рассматриваемого фильтра представляет собой производную входного сигнала, задержанного на некоторое время, определяемое фазочастотной характеристикой, т. е. такое устройство действительно обладает свойством дифференцирования.

В системах управления часто требуется линейное изменение коэффициента управления  $K(\omega)$  (например, усиление или ослабление нагрузки) в зависимости от частоты  $\omega$  воздействия. Для этой цели используют дифференциаторы, линейно возрастающую АЧХ которых можно истолковать как частотнозависимый коэффициент управления  $K(\omega)$  в этом же диапазоне частот: подавая на вход дифференциатора гармонические колебания, получаем линейно зависящие от частоты коэффициенты управления, рассчитываемые согласно (1.15, а).



**Рис. 1.6.** АЧХ дифференциатора  $[K(\omega)]$

Заметим, что вычисление огибающей сигнала и частотнозависимого коэффициента управления в дополнение невозможно без многократных вычислений тригонометрических функций и извлечения квадратного корня из суммы двух величин.

### 1.3. Цифровой спектральный анализ

*Цифровой спектральный анализ* — это совокупность разнообразных методов обработки цифровых сигналов, которые позволяют оценить частотный состав (спектр) исследуемого сигнала. Задача спектрального анализа может

носить как самостоятельный характер (например, в сейсмологии для определения типа сейсмического события, в геофизике для поиска месторождений полезных ископаемых и разработки новых методов поиска и т. п.), так и вспомогательный (в системах компрессии речи и изображений, компенсации помех и фильтрации).

В ЦОС важнейшими сигналами являются периодические последовательности с периодом отсчетов  $N$  и последовательности конечной длины в  $N$  отсчетов.

Для периодических последовательностей вводится *дискретное преобразование Фурье (ДПФ)*:

$$\text{прямое ДПФ} \quad X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad 0 \leq k \leq N-1; \quad (1.27)$$

$$\text{обратное ДПФ} \\ \text{(ОДПФ)} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad 0 \leq n \leq N-1, \quad (1.28)$$

где  $X(k)$  —  $k$ -я комплексная амплитуда (составляющая) спектра (ДПФ);  $x(n)$  — отсчеты дискретного сигнала (периодического с периодом  $N$  или конечной длины  $N$ );  $W_N^{nk}$  — поворачивающий множитель (или ядро преобразования), равный

$$W_N^{nk} = e^{-j\frac{2\pi}{N}nk}. \quad (1.29)$$

Составляющие спектра  $X(k)$  имеют период  $N$  и располагаются по частотной оси с интервалом

$$\Delta\omega = 2\pi / NT, \quad T = 1/f_d. \quad (1.30)$$

Этот интервал называется *частотой преобразования*. Составляющая с номером  $k$  располагается на частоте

$$\omega_k = \Delta\omega k, \quad X(k) = X\left(e^{jk\Delta\omega}\right). \quad (1.31)$$

Говорят, что ДПФ является  $N$ -точечным, если оно содержит  $N$  составляющих спектра. Модуль ДПФ  $|X(k)|$  называется *спектром амплитуд* (часто просто *спектром*, если нет каких-либо оснований для уточнения), аргумент ДПФ называется спектром фаз

$$\varphi(k) = \arg X(k). \quad (1.32)$$

Это же преобразование можно применить и к последовательности конечной длины, рассматривая ее как один период повторяющейся последовательности.

### Замечание

Следует иметь в виду, что смысл ДПФ для дискретных периодических сигналов и сигналов конечной длины различен. Спектр периодического дискретного сигнала с периодом  $N$  является также периодическим и дискретным, имеющим на одном периоде ровно  $N$  комплексных составляющих, поэтому ДПФ *точно выражает его спектр*. С дискретным сигналом конечной длины  $N$  дело обстоит несколько сложнее: его спектр, являясь периодическим с периодом  $\omega_d$ , представляет собой непрерывную функцию частоты, однако по  $N$  равноотстоящим отсчетам спектра  $X(k)$ , т. е. по ДПФ, гарантируется возможность *точного восстановления* как непрерывного спектра  $X(e^{j\omega})$ , так и последовательности  $x(n)$ .

## 1.3.1. Быстрое преобразование Фурье

Прямое вычисление ДПФ по формулам (1.27) и (1.28) для больших  $N$  (например, при обработке речевых сигналов длина одного фрагмента  $N$  достигает значения  $2^{10} = 1024$ ) крайне неэффективно и может стать препятствием для обеспечения реального времени. Действительно, для вычисления  $N$ -точечного преобразования требуется произвести  $(N-1)^2$  комплексных умножений и  $N(N-1)$  комплексных сложений, т. е. объем вычислений имеет порядок  $N^2$  операций сложения и умножения комплексных чисел.

Для уменьшения вычислительных затрат разработаны *алгоритмы быстрого вычисления ДПФ*, называемые *быстрым преобразованием Фурье (БПФ)*. Эти алгоритмы основаны на периодичности ядра преобразования  $W_N^{nk}$ . Идея БПФ состоит в том, чтобы разделить  $N$ -точечную последовательность на две, из ДПФ которых можно получить ДПФ исходной последовательности, и продолжать такое деление каждой новой последовательности до тех пор, пока не останутся последовательности, состоящие только из двух элементов. Конечно, такое деление возможно лишь при  $N = 2^m$ .

### Алгоритм БПФ с прореживанием по времени

Исходная  $N$ -точечная последовательность  $x(n)$  делится на две  $N/2$ -точечных последовательности, одна из которых содержит отсчеты с нечетными номерами, а другая — с четными номерами:

□ четная последовательность  $x_1(n) = x(2n)$ ;

□ нечетная последовательность  $x_2(n) = x(2n+1)$

при  $n = 0, 1, \dots, (N/2)-1$ . Тогда  $N$ -точечное ДПФ исходной последовательности  $x(n)$  преобразуется в два  $N/2$ -точечных ДПФ:

$$X(k) = \begin{cases} X_1(k) + W_N^k X_2(k); \\ X_1(k) - W_N^k X_2(k), \end{cases} \quad (1.33)$$

где верхняя строка дает первые  $N/2$  составляющих ДПФ  $X(k)$ , а нижняя строка — вторые  $N/2$  составляющих,  $0 \leq k \leq N/2-1$ . Далее аналогичным образом  $N/2$ -точечные ДПФ заменяются двумя  $N/4$ -точечными каждое, и т. д. Такая сортировка осуществляется до тех пор, пока не образуются  $N/2$  последовательностей по два элемента в каждой. В результате  $N$ -точечное ДПФ сводится к  $m = \log_2 N$  этапам, на каждом из которых требуется вычислить  $N$  коэффициентов. Выражения (1.33) показывают, что на каждом этапе требуется  $N$  комплексных сложений и  $N/2$  комплексных умножений. Это легко видеть из направленного графа (рис. 1.7), напоминающего крылья бабочки, чем и объясняется название "бабочка" самой операции (1.33) и графа, на котором стрелкой обозначено умножение на  $W_N^{nk}$ . Использование базовой операции "бабочка" снижает количество требуемых для вычисления  $N$ -точечного ДПФ комплексных сложений с  $N^2$  до  $\frac{3}{2}N \log_2 N$ , что является существенной экономией вычислительных, а потому и временных ресурсов.

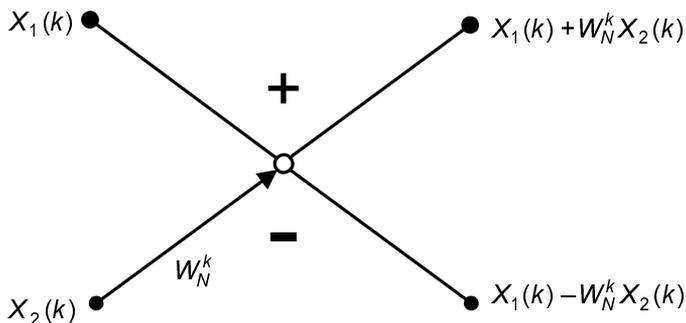


Рис. 1.7. Операция "бабочка" с прореживанием по времени

В результате сортировки отсчетов  $x(n)$  по нечетным и четным номерам входные данные записываются в необычном порядке, который называется *двоичной инверсией*, или *бит-реверсией*: например, если трехразрядное двоичное представление номера  $n$  отсчета  $x(n)$  имеет вид  $n = \{n_2 n_1 n_0\}$ , то отсчет  $x\{n_2 n_1 n_0\}$ , должен располагаться на месте  $x\{n_0 n_1 n_2\}$ . Это означает, что для правильного выполнения БПФ необходимо в исходном двоичном номере заменить порядок расположения разрядов на обратный (инверсный). Например, входной отсчет  $x(6) = x(110)$  должен разместиться на  $011 \Rightarrow 3$  третьем месте, а третий отсчет  $x(3)$  окажется на шестом месте. При двоичной инверсии входной последовательности составляющие  $X(k)$  ДПФ будут расположены в естественном порядке.

## Алгоритм БПФ с прореживанием по частоте

В этом случае входная последовательность  $x(n)$  делится пополам на  $N/2$  первых и  $N/2$  последних отсчетов и так до тех пор, пока не сформируются  $N/2$  двухэлементных последовательностей. Базовая операция "бабочка" будет описываться выражением (1.34), а ее направленный граф примет вид, показанный на рис. 1.8.

$$X(k) = \begin{cases} X_1(k) + X_2(k); \\ (X_1(k) - X_2(k))W_N^k. \end{cases} \quad (1.34)$$

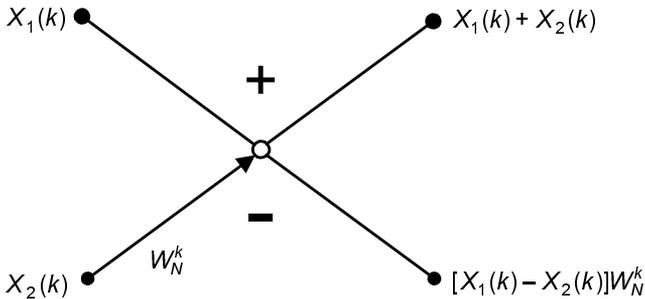


Рис. 1.8. Операция "бабочка" с прореживанием по частоте

Вычисление согласно данному алгоритму приводит к тому, что составляющие  $X(k)$  ДПФ располагаются в порядке, соответствующем бит-реверсии, поэтому их необходимо пересортировать согласно естественному порядку. Например,  $X(3) = X(011)$  следует разместить на шестом месте, т. к. инверсный номер оказывается равным  $110 \Rightarrow 6$ .

### Замечание

Алгоритмы БПФ являются рекурсивными: невозможно рассчитать  $N/2$ -точечное ДПФ, не рассчитав предварительно  $N/4$ -точечное ДПФ.

## 1.3.2. Дискретное преобразование Хартли (ДПХ)

Дискретное преобразование Фурье отображает последовательность вещественных данных в комплексную область, где хорошо разработаны методы анализа, существенно облегчающие изучение и трактовку колебательных процессов. Однако обработку вещественных данных желательно выполнять в вещественной области. Эту задачу решает дискретное преобразование Хартли (ДПХ), которое, как и ДПФ, может применяться в задачах спектрального анализа и цифровой фильтрации.

Прямое и обратное дискретное преобразование Хартли вещественной последовательности  $x(n)$  длины  $N$  определяются соотношениями:

$$H(k) = \sum_{n=0}^{N-1} x(n) \text{cas} \left( \frac{2\pi nk}{N} \right); \quad (1.35)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) \text{cas} \left( \frac{2\pi nk}{N} \right), \quad (1.36)$$

где  $\text{cas}\theta = \cos\theta + \sin\theta$ . Между ДПФ и ДПХ существует простая прямая связь:

$$\text{Re}\{X(k)\} = \frac{H(k) + H(-k)}{2}; \quad (1.37)$$

$$\text{Im}\{X(k)\} = \frac{H(k) - H(-k)}{2}. \quad (1.38)$$

С другой стороны, по известным составляющим  $X(k)$  ДПФ можно получить ДПХ

$$H(k) = \text{Re}\{X(k)\} - \text{Im}\{X(k)\}. \quad (1.39)$$

Симметричность формул прямого и обратного ДПХ, отсутствие комплексного представления данных и ряд других свойств ДПХ обеспечивают по сравнению с БПФ более высокую вычислительную эффективность при обработке вещественных данных, что особенно важно при двумерном анализе.

### 1.3.3. Дискретное косинусное преобразование

Одной из весьма сложных задач ЦОС является построение видеокodeков для сжатия изображения с целью уменьшения скорости передачи, а потому и уменьшения спектра, занимаемого видеосигналом. Рассмотрим существо проблемы на примере телевизионного сигнала.

Применяемая система со строчно-переменной фазой PAL (Phase Alternate Line) для образования одного кадра черно-белого изображения использует 625 строк, а на каждой строке 625 элементов (пикселей). Следовательно, для получения одного кадра изображения всего необходимо 380 625 пикселей. Для воспроизведения движения приемлемого качества и исключения эффекта дрожания экрана требуется формировать около 50 кадров в секунду. Отсюда нетрудно рассчитать скорость передачи

$$380 \ 625 \times 50 = 19 \ 031 \ 250 \text{ пикселей/с,}$$

что соответствует полосе пропускания около 20 МГц. Практически используется более узкая полоса около 6 МГц, что достигается, во-первых, сокращением числа передаваемых строк до 575 и, во-вторых, распределением строк по двум кадрам: нечетные строки выводятся в первом кадре, четные —