

МИНИСТЕРСТВО ОБОРОНЫ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ставропольское высшее военное авиационное инженерное училище
(военный институт) имени маршала авиации В.А. Судца

А.В. Баженов

ЦИФРОВЫЕ МЕТОДЫ РЕАЛИЗАЦИИ ПРОСТРАНСТВЕННО-
ВРЕМЕННОЙ ОБРАБОТКИ СИГНАЛОВ В АВИАЦИОННЫХ
РАДИОЭЛЕКТРОННЫХ КОМПЛЕКСАХ

Монография

Ставрополь
Издание училища
2006

Баженов, А.В. ЦИФРОВЫЕ МЕТОДЫ РЕАЛИЗАЦИИ ПРОСТРАНСТВЕННО-ВРЕМЕННОЙ ОБРАБОТКИ СИГНАЛОВ В АВИАЦИОННЫХ РАДИОЭЛЕКТРОННЫХ КОМПЛЕКСАХ [Текст]/ А.В. Баженов. – Ставрополь: СВВАИУ , 2006. – 219 с.

Цифровая обработка сигналов является одним из активно развиваемых направлений современной науки. Практические приложения этого направления охватывают различные области человеческой деятельности, в том числе и радиоэлектронное оборудование современных самолетов. В монографии изложены теоретические вопросы цифровой обработки сигналов, приведены методы и алгоритмы цифровой обработки одномерных и двумерных сигналов, а также представлены справочные данные по современным техническим устройствам реализации цифровой обработки: микроконтроллерам и программируемым сигнальным процессорам.

Для научных работников. Полезна инженерам, занимающимся разработкой и исследованием устройств цифровой обработки сигналов

Рецензент: доктор технических наук, профессор Сныткин И.И.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	6
1.1 Типы сигналов. Связь между сигналами различных типов	6
1.1.1 Классификация сигналов	6
1.1.2 Связь между аналоговыми и дискретными сигналами	17
1.1.3 Связь между дискретными и цифровыми сигналами	21
1.2 Основы теории дискретных (цифровых) линейных систем	29
1.2.1 Линейные системы с постоянными параметрами	29
1.2.2 Физическая реализуемость. Устойчивость дискретных систем	31
1.2.3 Разностные уравнения	33
1.3 Частотные характеристики цифровых устройств	36
1.4 Дискретные линейная и кольцевая свертки	44
2 ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ СИГНАЛОВ	52
2.1 Представление сигнала с помощью ортогональных функций	52
2.2 Дискретное преобразование Фурье	54
2.3 Z-преобразование	58
2.4 Общая характеристика негармонических ортогональных преобразований	62
2.5 Функции Радемахера и Хаара. Преобразование Хаара	65
2.6 Функции Уолша. Дискретное преобразование Уолша	68
2.7 Ортогональное вейвлет-преобразование	80
3 БЫСТРЫЕ АЛГОРИТМЫ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЙ	92
3.1 Постановка задачи и обоснование обобщенного алгоритма быстрого преобразования Фурье (БПФ)	92
3.2 Применение алгоритмов БПФ	101
3.3 Алгоритмы быстрого преобразования Уолша	105
3.3.1 Быстрое преобразование Уолша-Адамара, упорядоченное по Адамару	105
3.3.2 Быстрое преобразование Уолша-Адамара, упорядоченное по Уолшу	110
3.4 Алгоритм быстрого преобразования Хаара	113
4 ПРИМЕНЕНИЕ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	118

4.1	Линейные алгоритмы цифровой обработки	118
4.1.1	Назначение, классификация, обобщенная схема цифрового фильтра	119
4.1.2	Структурный синтез цифрового фильтра	130
4.2	Двумерные цифровые сигналы и их обработка	132
4.2.1	Двумерные цифровые сигналы	132
4.2.2	Улучшение качества изображения на основе линейной и нелинейной фильтрации	140
4.2.3	Сжатие изображений	147
4.3	Алгоритмы нелинейной обработки речевых сигналов	150
5	ТЕХНИЧЕСКИЕ СРЕДСТВА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	156
5.1	Общие принципы технической реализации алгоритмов цифровой обработки	156
5.2	Применение однокристальных микроконтроллеров для цифровой обработки сигналов	161
5.2.1	RISC – архитектура современных микроконтроллеров	166
5.3	Программируемые процессоры цифровой обработки сигналов	172
5.3.1	Основные операции обработки сигналов в цифровых системах и их влияние на структуру сигнального процессора	172
5.3.2	Обобщенная архитектура DSP	174
5.4	Процессоры семейства DSP56000	177
5.4.1	Обзор архитектуры и шинной организации DSP56000	177
5.4.2	Порты ввода/вывода	201
5.5	Специализированная цифровая вычислительная машина «Багет-55»	207
	ЗАКЛЮЧЕНИЕ	216
	СПИСОК ЛИТЕРАТУРЫ	217

ВВЕДЕНИЕ

Цифровая обработка сигналов (ЦОС) - одно из важнейших направлений информационных технологий. Она находит широкое применение в обработке сигналов самой различной природы. С первых шагов становления этого направления едва ли не главным объектом ее применения были оптические, радио- и аналогичные им одномерные и двумерные сигналы: изображения, голограммы, интерферограммы, томограммы и т.п. Это объясняется огромной ролью этих сигналов в информационном обеспечении современной науки и техники, в особенности военного применения.

Использование цифровой техники для обработки изображений началось в конце 50-х - начале 60-х годов прошлого столетия с применением универсальных ЦВМ для моделирования методов кодирования изображений и передачи изображений по каналам связи. Но с развитием вычислительной техники (конец 60-х - начало 70-х годов) цифровые вычислительные машины стали использовать и для решения других задач обработки сигналов. А к середине 70-х годов, благодаря развитию микроэлектронной базы вычислительной техники, появились первые специализированные цифровые системы обработки сигналов, в том числе и бортового размещения.

В отличие от аналоговых устройств обработки информации цифровые процессоры обладают высокой способностью к адаптации и гибкостью. Цифровая вычислительная машина способна перестраивать структуру обработки сигнала без перестройки своей физической структуры, поэтому она является идеальным средством адаптивной обработки сигналов и быстрой перестройки ее на решение разных задач. Другим достоинством ЦОС является простота и естественность получения и переработки количественной информации, содержащейся в сигналах. Цифровой сигнал содержит переносимую аналоговым сигналом информацию в чистом виде, вне ее физической оболочки. Благодаря своему универсальному характеру цифровой сигнал служит идеальным средством объединения различных информационных систем.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

1.1 Типы сигналов. Связь между сигналами различных типов

1.1.1 Классификация сигналов

Радиотехнические системы (РТС), входящие в состав бортового радиоэлектронного оборудования, относятся к классу информационно-управляющих систем. Отличительной особенностью РТС является передача информации, формирование информации или разрушение информации с помощью электромагнитных волн радиодиапазона. Радиоволны, несущие ту или иную информацию, называются радиосигналами.

В общем случае сигналом называется процесс изменения во времени физического состояния какого-либо объекта, служащий для отображения, регистрации и передачи сообщений.

Сигналы, как некоторые физические процессы, можно наблюдать с помощью различных приборов и устройств – осциллографов, вольтметров, приемников. Такой эмпирический подход имеет существенный недостаток. Явления, изучаемые экспериментатором, всегда выступают как частные, единичные проявления, лишенные той степени обобщенности, которая позволила бы судить об их фундаментальных свойствах, предсказывать результаты в изменившихся условиях.

Для того чтобы сделать сигналы объектами теоретического изучения и расчетов, следует указать способ их математического описания, или, говоря языком современной науки, создать математическую модель исследуемого сигнала.

Математическая модель сигнала представляет собой функциональную зависимость, в которой аргументом чаще всего является время. Способы задания модели могут быть разнообразными – математическая формула, вычислительный алгоритм и даже словесное описание. Знание математических моделей сигналов дает возможность сравнивать их между собой, устанавливать их тождество и различие и, в конечном итоге, проводить их классификацию.

Радиосигнал как электромагнитная волна в некоторой диэлектрической материальной среде может быть задан законом изменения вектора напряженности электрического поля в зависимости от време-

ни и пространственной координаты /1,2 /:

$$\vec{E}(t, z) = E_0 \cdot e^{-j(\omega t + \varphi_0)} \cdot e^{-j\frac{2\pi}{\lambda}z}, \quad (1.1)$$

где E_0 - амплитудное значение напряженности электрического поля;

ω - циклическая (круговая) частота;

φ_0 - начальная фаза;

λ - длина волны.

С учетом обозначения комплексной амплитуды $\dot{E}_0 = E_0 \cdot e^{-j\varphi_0}$ выражение (1.1) можно переписать в виде

$$\vec{E}(t, z) = \dot{E}_0 \cdot e^{-j(\omega t + \frac{2\pi}{\lambda}z)}. \quad (1.2)$$

Наблюдаемый электромагнитный процесс может быть представлен функциями напряжений и токов, формируемых в антенной системе под воздействием $\vec{E}(t, z)$. Для общности обозначения радиосигнала перепишем выражение (1.2) в следующем виде:

$$\vec{S}(t, z) = \dot{S}_0 \cdot e^{-j(\omega t + \frac{2\pi}{\lambda}z)}, \quad (1.3)$$

где обозначение переменных имеет тот же смысл, что и в выражении (1.2).

Информационную нагрузку могут нести практически все параметры, входящие в выражение (1.3). Характер воздействия на параметры несущего колебания определяется назначением РТС /3,4/.

В РТС передачи информации (различные системы радиосвязи, телеметрии, передачи команд и т.д.) амплитуда радиосигнала, частота или начальная фаза являются функциями времени и изменяются по закону передаваемого сообщения. Такие радиосигналы называются, соответственно, амплитудно-модулированными, частотно-модулированными и фазомодулированными.

В РТС формирования (извлечения) информации (радиолокационные системы, радионавигационные системы, системы радиотехнической разведки) в принимаемом радиосигнале так же присутствует функциональная зависимость параметров сигнала, отражающая информацию о характеристиках лоцируемого объекта. В общем случае, воздействию могут подвергаться и амплитуда, и частота, и начальная

фаза, а также и другие параметры зондирующего сигнала. Задачей РТС в этом случае является извлечение информации об объекте по закону изменения параметров радиосигнала.

РТС разрушения информации (противодействия) предназначены для создания условий, в которых работа радиосистем противника становится невозможной. Радиосигналы, используемые в таких системах, называются активными помехами. Закон изменения параметров помехового сигнала выбирается таким образом, чтобы затруднить противнику извлечение информации из смеси помехи и полезного сигнала.

В зависимости от длительности отрезка времени, в течение которого существует радиосигнал, различают непрерывные и импульсные сигналы. Непрерывным называется радиосигнал, описываемый выражением (1.3), если для любого интервала $[t_1, t_2]$ в точке пространства, удаленной от источника на расстояние z , значение $\vec{S}(t, z)$ определено.

Импульсный радиосигнал описывается выражением вида:

$$\vec{S}(t, z) = \begin{cases} \dot{S}_0 \cdot e^{-j(\omega t + \frac{2\pi}{\lambda}z)}, & \text{при } \frac{-\tau}{2} \leq t \leq \frac{\tau}{2}, \\ 0, & \text{при } t < \frac{-\tau}{2} \text{ и } t > \frac{\tau}{2} \end{cases}, \quad (1.4)$$

где τ - длительность импульса.

Огибающая высокочастотной составляющей радиоимпульса носит название видеоимпульса.

Импульсный сигнал, повторяющийся через равные промежутки времени T_n , называется импульсной последовательностью, а интервал T_n , соответственно, называется периодом повторения.

Законы изменения параметров непрерывных и импульсных сигналов являются физическим носителем или, иначе, аналогом переносимой информации, поэтому для их обозначения используется еще одно общее название – аналоговые сигналы.

В общем случае, аналоговый сигнал описывается непрерывной (или кусочно-непрерывной для импульсной последовательности) функцией $x_a(t)$, причем и аргумент и сама функция могут принимать любые значения из некоторых интервалов /5/:

$$t' \leq t \leq t'', \quad x_a' \leq x_a \leq x_a''.$$

На рисунке 1.1, а изображен график аналогового сигнала:

$$x_a(t) = U_m \sin \omega t; \quad \omega = 2\pi f; \quad U_m = 1B; \quad f = \frac{1}{2\pi} \Gamma\text{ц}.$$

К аналоговым сигналам относятся, например, речевые сигналы в телефонии и радиосвязи, телевизионные сигналы и т.п.

В тех случаях, когда значения аналогового сигнала измеряются лишь в строго определенные моменты времени, говорят о дискретизации по времени аналогового сигнала.

Аналитически дискретный сигнал описывается решетчатой функцией (последовательностью, временным рядом) $x(nT)$, которая может принимать любые значения в некотором интервале $x' \leq x \leq x''$, только лишь в моменты времени $t = 0T, 1T, 2T, \dots, nT$, где $T = \frac{1}{f_\Delta}$ - интервал (период) дискретизации; f_Δ - частота дискретизации.

На рисунке 1.1, б изображен график дискретного сигнала:

$$x(nT) = U_m \cdot \sin n\omega T; \quad \omega = 2\pi f; \quad U_m = 1B; \quad f = \frac{1}{2\pi} \Gamma\text{ц}; \quad T = \frac{\pi}{4} c.$$

Используются и другие способы обозначения решетчатой функции $x(n), x_n$, когда период дискретизации тем или иным способом нормирован и остается постоянным, или $\{x(nT)\}$, когда необходимо подчеркнуть, что речь идет о решетчатой функции в целом, а не об отдельном значении (отсчете) этой функции при $t = nT$.

Дискретный и аналоговый сигналы могут быть как вещественными, так и комплексными в соответствии с принимаемыми отсчетами значениями. К дискретным относятся сигналы в системах связи с амплитудно-импульсной модуляцией или любые аналоговые сигналы после дискретизации по времени перед последующим преобразованием в цифровую форму.

Цифровой сигнал описывается квантованной решетчатой функцией (квантованной последовательностью, квантованным временным рядом) $x_\kappa(nT)$, т.е. решетчатой функцией, принимающей лишь ряд дискретные значения – уровни квантования h_1, h_2, \dots, h_k , в то время как независимая переменная $n = 0, 1, \dots, N$.

На рисунке 1.1, в изображен график цифрового сигнала:

$$\begin{aligned}
x_y(nT) &= O_k(p)U_m \sin(n\omega T); \\
\xi &= 2\pi f; \quad U_m = 1B; \\
f &= \frac{1}{2\pi} \Gamma y; \quad T = \frac{\pi}{4} c; \\
h_l &= -1 + 0,25(l - 1),
\end{aligned}$$

где $l = 1, 2, \dots, K = 9$ и нелинейная функция $O_k(p)$ определяется следующим образом:

$$O_k(p) = \begin{cases} h_1 & \text{при } p \leq (h_2 + h_1)/2; \\ h_l & \text{при } (h_l + h_{l-1})/2 < p \leq (h_{l+1} + h_l)/2; \\ h_k & \text{при } (h_k + h_{k-1})/2 < p. \end{cases}$$

Каждый из уровней квантования представляется кодом, состоящим из двоичных цифр, так что передача или обработка отсчета цифрового кодированного сигнала сводится к операциям над безразмерным двоичным кодом. Число K уровней квантования и число S разрядов соответствующих кодов связаны следующим отношением:

$$S \geq \text{int}(\log_2 K),$$

где значение функции $\text{int}(B)$ - обозначает наименьшее целое число, не меньшее числа B .

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 используются скалярные функции. Но иногда применяются и более сложные модели – комплексные и 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 (конечные) и инфинитные (бесконечные), ограниченные и неограниченные сигналы.

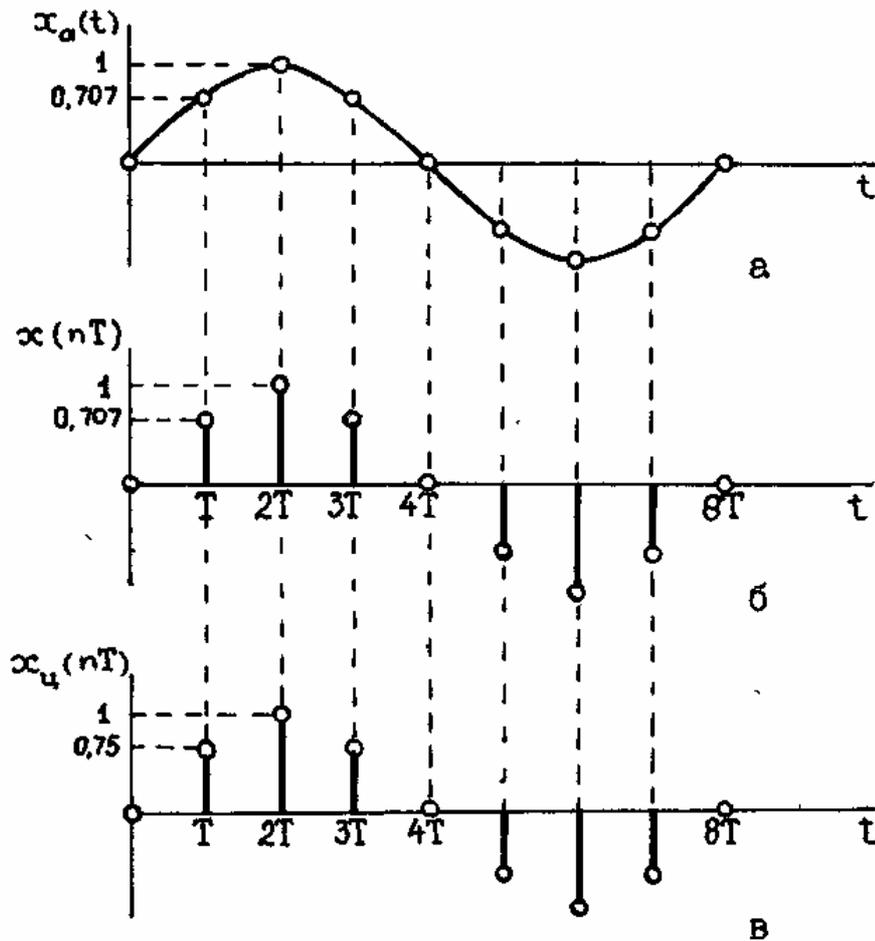


Рисунок 1.1 - Сигналы различных типов:
 а – аналоговый; б – дискретный; в – цифровой

Финитными называются сигналы, область определения которых ограничена. 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 негатива или распределения яркости свечения экрана телевизионного монитора.

Фактически все встречающиеся в природе сигналы являются ог-

раниченными. Неограниченный сигнал – идеализация, принимаемая в тех случаях, когда ограничение области значений сигнала несущественно для заданной задачи или просто неизвестно.

Дискретная (цифровая) последовательность может быть получена несколькими способами /6/. Проще всего можно взять набор чисел и расположить их в виде последовательности. Например, числа $0, 1, 2, \dots, (N - 1)$ образуют «пилообразную» последовательность $h(n) = n, 0 \leq n \leq N - 1$. Другой способ состоит в использовании некоторого рекуррентного соотношения. Например, равенство $h(n) = h(n - 1)/2$ с начальным условием $h(0) = 1$ дает последовательность $h(n) = (1/2)^n, 0 \leq n \leq \infty$. Третий способ — взять равноотстоящие отсчеты непрерывного колебания и из их величин образовать последовательность, т. е. положить $h(nt) = h(t) |_{t=nt}, -\infty \leq n \leq \infty, T$ – интервал дискретизации. Первые два метода получения последовательностей не связаны с временем, тогда как третий существенно от него зависит.

Часто полезным и информативным является графическое изображение последовательностей. Ниже приведены (и графически изображены на рисунке 1.2) некоторые важные последовательности, часто используемые при цифровой обработке сигналов /6/. На рисунке 1.2, а показан цифровой единичный импульс (или единичный отсчет) $u_0(n)$, который определяется следующим образом:

$$u_0(n) = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0. \end{cases}$$

В дискретных системах этот импульс играет такую же роль, как аналоговый единичный импульс (или дельта-функция Дирака) $\delta(t)$ в аналоговых системах /2/. Важное различие между ними состоит в том, что первый является физически реализуемым сигналом, тогда как второй рассматривается только как обобщенная функция (или распределение).

На рисунке 1.2, б изображен единичный импульс, задержанный на n_0 отсчетов, который определяется как

$$u_0(n - n_0) = \begin{cases} 1, & n = n_0, \\ 0, & n \neq n_0. \end{cases}$$

А на рисунке 1.2, в представлен единичный скачок $u_{-1}(n)$, задаваемый следующим образом:

$$u_{-1}(n) = \begin{cases} 1, & n \geq 0, \\ 0, & n < 0. \end{cases}$$

Нетрудно показать, что единичный скачок связан с единичным импульсом соотношением

$$u_{-1}(n) = \sum_{l=-\infty}^n u_0(n-l).$$

На рисунках 1.2, з и д изображены убывающая (при $a < 1$) показательная функция $g(n)$ и косинусоида $h(n)$, определяемые соответственно как

$$g(n) = \begin{cases} a^n, & n \geq 0, \\ 0, & n < 0 \end{cases}$$

и

$$h(n) = \cos\left(\frac{2\pi n}{n_0}\right) \text{ для всех } n.$$

Особенно важной последовательностью является комплексная экспонента $e^{j\omega n} = \cos(\omega n) + j \sin(\omega n)$. Поскольку эта последовательность является комплексной, для ее изображения необходимы отдельные графики вещественной и мнимой частей. Позднее мы увидим, что многие из вышеупомянутых последовательностей играют важную роль в теории цифровой обработки сигналов.

Произвольные последовательности легко выразить через основную последовательность (единичный импульс), используя задержку и масштабирование. Рассмотрим числовую последовательность $a(0), a(1), a(2), \dots$, где $a(n)$ - величина n -го элемента. Такая последовательность описывается равенством

$$\{a(n)\} = \sum_{m=-\infty}^{\infty} a(m)u_0(n-m). \quad (1.5)$$

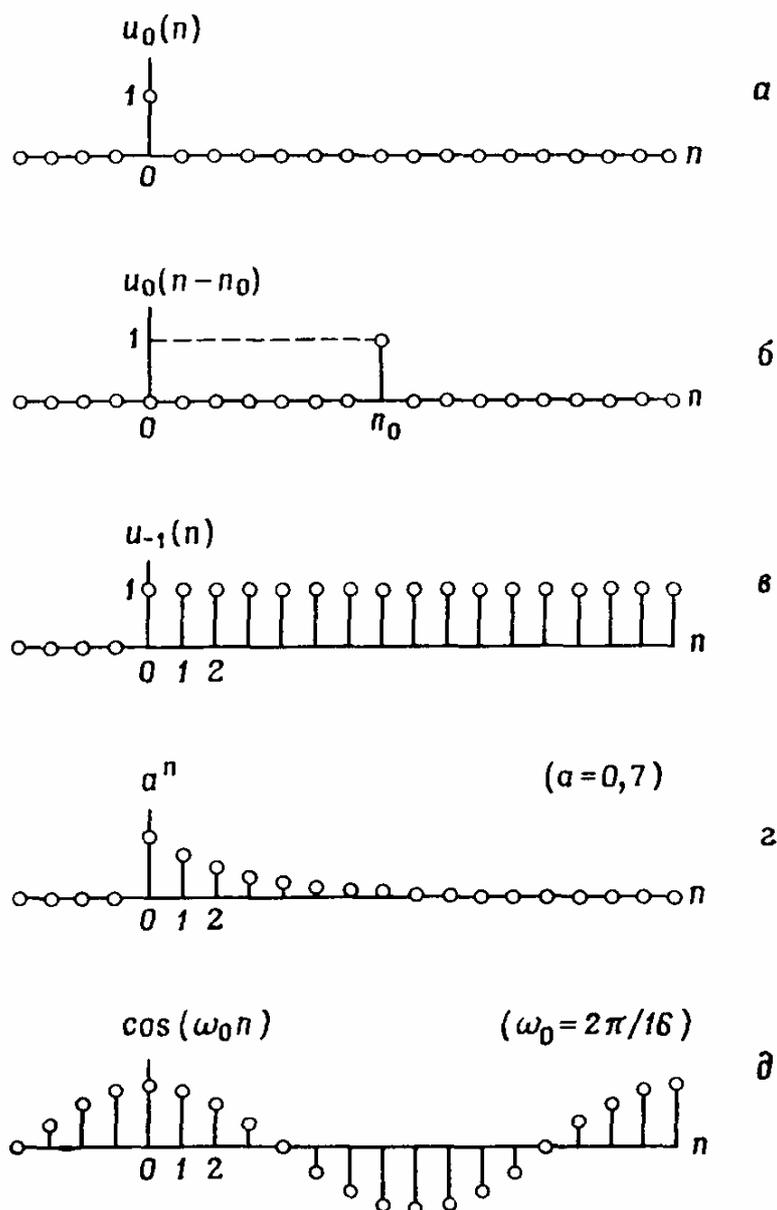


Рисунок 1.2 - Некоторые важнейшие последовательности, используемые при цифровой обработке сигналов

С точки зрения прогнозируемости мгновенного значения различают детерминированные и вероятностные (случайные) сигналы.

Отличительная черта случайного сигнала состоит в том, что его мгновенные значения не могут быть заранее предсказаны и вычислены. При детерминированном описании сигналы рассматриваются индивидуально, независимо друг от друга, и считается, что значение сигнала может быть задано в каждой точке, где он определен. А при вероятностном описании сигналы рассматриваются уже как выборочные функции, или реализации из некоторого ансамбля сигналов, и

строится математическое описание не каждого отдельного сигнала, а ансамбля в целом.

Давая математическое описание сигналов, удобно рассматривать их как точки или векторы в некотором функциональном пространстве (пространстве сигналов), а преобразование сигналов – как отображения в этом пространстве. При этом свойства сигналов трактуются как свойства пространства. Слово «пространство» используется, чтобы придать понятию множества сигналов геометрический смысл и, тем самым, наглядность.

Для того чтобы математически описать различия между сигналами, вводится понятие метрики пространства, т.е. способа, в соответствии с которым каждой паре точек пространства, например a_1 и a_2 , может быть поставлено в соответствие некоторое вещественное неотрицательное число $d(a_1, a_2)$, имеющее смысл расстояния между ними. Обычно этот способ удовлетворяет следующим правилам:

$$d(a_1, a_2) = 0, \text{ если } a_1 = a_2;$$

$$d(a_1, a_2) = d(a_2, a_1);$$

$$d(a_1, a_3) \leq d(a_1, a_2) + d(a_2, a_3).$$

Смысл первых двух условий очевиден. Смысл введения третьего условия, которое называется «правилом треугольника», в том, что оно является формальным выражением следующего требования в метрике: если две точки близки к третьей, то они должны быть близки между собой.

Поскольку в теории сигналов понятие расстояния используется для трактовки отличия одного сигнала от другого или ошибки представления одного сигнала другим, для характеристики пространства сигналов должна выбираться такая метрика, которая наиболее полно может описать это отличие одним числом.

Вероятностное описание сигнала является более общим, чем детерминированное, поэтому определим метрику пространства для случайных сигналов.

Пусть, например, отличия одного сигнала от другого возникают в результате действия на сигналы аддитивного некоррелированного гауссовского шума. Рассмотрим случай различения двух дискретных сигналов $\{a_{1k}\}$ и $\{a_{2k}\}$, таких, что

$$a_{2k} = a_{1k} + n_k, \quad k = 0, 1, \dots, N-1, \quad (1.6)$$

где n_k - случайные величины с нормальной плотностью вероятностей

$$p(n) = (1/\sqrt{2\pi\sigma}) \exp(-n^2 / 2\sigma^2)$$

и дисперсией σ^2 .

Очевидно, все различия между сигналами a_1 и a_2 заключены в сигнале $n\{nk\}$, а он может быть полностью статистически описан многомерной плотностью вероятностей

$$p(n_0, n_1, \dots, n_{N-1}) = (2\pi\sigma^2)^{-N/2} \exp\left[-\frac{1}{2}\sigma^2 \sum_{k=0}^{N-1} (a_{2k} - a_{1k})^2\right],$$

которая в свою очередь полностью определяется величиной

$$d(a_1, a_2) = \sqrt{\sum_{k=0}^{N-1} |a_{2k} - a_{1k}|^2},$$

называемой эвклидовым расстоянием между a_2 и a_1 . Так порождается эвклидова метрика [7].

Эвклидова метрика очень популярна в теории сигналов по двум причинам. Во-первых, она удобна в расчетах и имеет определенный физический смысл: это мера энергии разности двух сигналов, измерение которой легко воплотить в физическом приборе. Во-вторых, эта метрика в точности адекватна задачам, где отличия между сигналами порождаются суммарным действием большого числа помех или ошибок измерения.

Эвклидову метрику часто называют также среднеквадратической, ибо она дает квадрат разности сигналов, усредненный по области их определения. В этом смысле ее обобщением является взвешенная среднеквадратическая метрика, определяемая для дискретного случая как

$$d(a_1, a_2) = \sqrt{\sum_{k=0}^{N-1} \omega_k (a_{2k} - a_{1k})^2},$$

где $\{\omega_k\}$ - набор весовых констант. Такая метрика потребовалась бы, например, если бы в (1.6) мы предположили, что $\{n_k\}$ имеют разные значения дисперсии σ_k^2 .

1.1.2 Связь между аналоговыми и дискретными сигналами

Взаимосвязанный переход от аналогового сигнала к дискретному и наоборот заключается в выполнении операций дискретизации и восстановления.

Операция дискретизации состоит в том, что по заданному аналоговому сигналу $x_a(t)$ строится дискретный сигнал $x(nT)$, причем $x(nT) = x_a(nt)$.

Операция восстановления состоит в том, что по заданному дискретному сигналу $x(nT)$ строится аналоговый сигнал $x_a(t)$, $x(nT) \rightarrow x_a(t)$.

Операции дискретизации и восстановления взаимно обратны в том случае, когда дискретизируемый аналоговый сигнал удовлетворяет условиям теоремы Котельникова: если аналоговый сигнал $x_a(t)$ имеет ограниченный (финитный) спектр, принимающий отличные от нуля значения лишь при $\omega_{a \min} \leq \omega_a \leq \omega_{a \max}$ (где $\omega_a = 2\pi f_a$ - круговая частота аналогового сигнала), и дискретизация этого сигнала выполняется с частотой f_δ такой, что

$$2\omega_{a \max} / q \leq 2\pi f_\delta \leq 2\omega_{a \min} / (q - 1),$$

где $q = 1, 2, \dots, \text{int} \left[\frac{\omega_{a \max}}{\omega_{a \max} - \omega_{a \min}} \right]$, то этот аналоговый сигнал может быть точно восстановлен по отсчетам соответствующего дискретного сигнала.

Другой вариант записи теоремы Котельникова имеет вид:

$$f_\delta \geq 2f_{\max}, \quad (1.7)$$

где f_{\max} - максимальная частота в спектре аналогового сигнала.

Как уже отмечалось, последовательность $x(nT)$ часто получают

путем дискретизации с периодом T непрерывного сигнала $x(t)$.

Пара преобразований Фурье для непрерывного колебания $x(t)$ имеет вид /5/

$$X_H(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt, \quad (1.8)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_H(j\Omega)e^{j\Omega t} d\Omega. \quad (1.9)$$

Аналогичные соотношения для дискретизированного сигнала имеют вид

$$X(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}, \quad (1.10)$$

$$x(nT) = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} X(e^{j\omega T})e^{j\omega nT} d\omega. \quad (1.11)$$

Поскольку $x(nT) = x(t)|_{t=nT}$, то можно связать $X_H(j\Omega)$ и $X(e^{j\omega T})$, вычислив интеграл (1.9) для $t = nT$, причем интеграл с бесконечными пределами следует заменить бесконечной суммой интегралов на интервалах длиной $2\pi/T$. Таким образом,

$$x(nT) = \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \int_{(2m-1)\pi/T}^{(2m+1)\pi/T} X_H(j\Omega)e^{j\Omega nT} d\Omega. \quad (1.12)$$

Изменив в выражении (1.12) порядок действий и заменив Ω на ω , получим:

$$x(nT) = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \left[\frac{1}{T} \sum_{m=-\infty}^{\infty} X_H(\omega + \frac{2\pi}{T}m) \right] \cdot e^{j\omega nT} d\omega. \quad (1.13)$$

Приравнивая подынтегральные выражения в (1.13) и (1.11), получаем искомое соотношение

$$X(e^{j\omega T}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X_H(\omega + \frac{2\pi}{T}m). \quad (1.14)$$

Из формулы (1.14) видно, что периодическая спектральная функция последовательности состоит из суммы бесконечного числа спектральных компонент непрерывного сигнала. Если спектр непрерывного сигнала ограничен по полосе диапазоном частот $|\Omega| \leq \pi/T$, т.е. $X_H(j\Omega) = 0$ при $|\Omega| > \pi/T$, то из соотношения (1.14) следует, что в диапазоне частот $|\omega| \leq \pi/T$

$$X(e^{j\omega T}) = \frac{1}{T} X_H(\omega).$$

В этом случае спектр последовательности непосредственным образом связан со спектром непрерывного колебания (рисунок 1.3, а, б).

Если же $X_H(j\Omega)$ не ограничен диапазоном $|\Omega| \leq \pi/T$, то соотношение между спектрами дискретизированного и непрерывного сигналов оказывается более сложным.

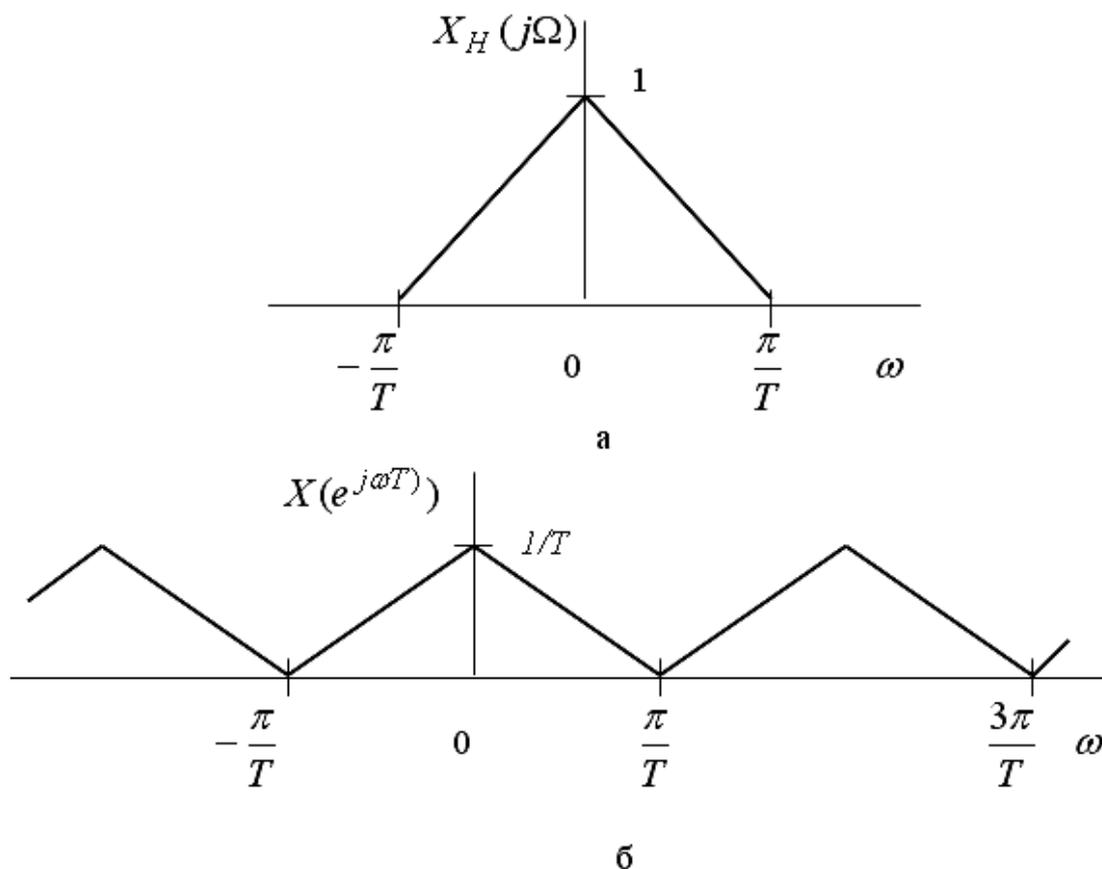


Рисунок 1.3 - Связь между спектрами непрерывного и дискретного сигналов при правильном выборе частоты дискретизации

Типичный пример показан на рисунке 1.4, а-в. Спектр непрерывного сигнала (рисунок 1.4,а) ограничен полосой $|\Omega| \leq 3\pi/2T$. Из формулы (1.14) следует, что члены с $m = 0 \pm 1$ формируют $X(e^{j\omega T})$ в диапазоне частот $|\omega| \leq \pi/T$ (рисунок 1.4, б).

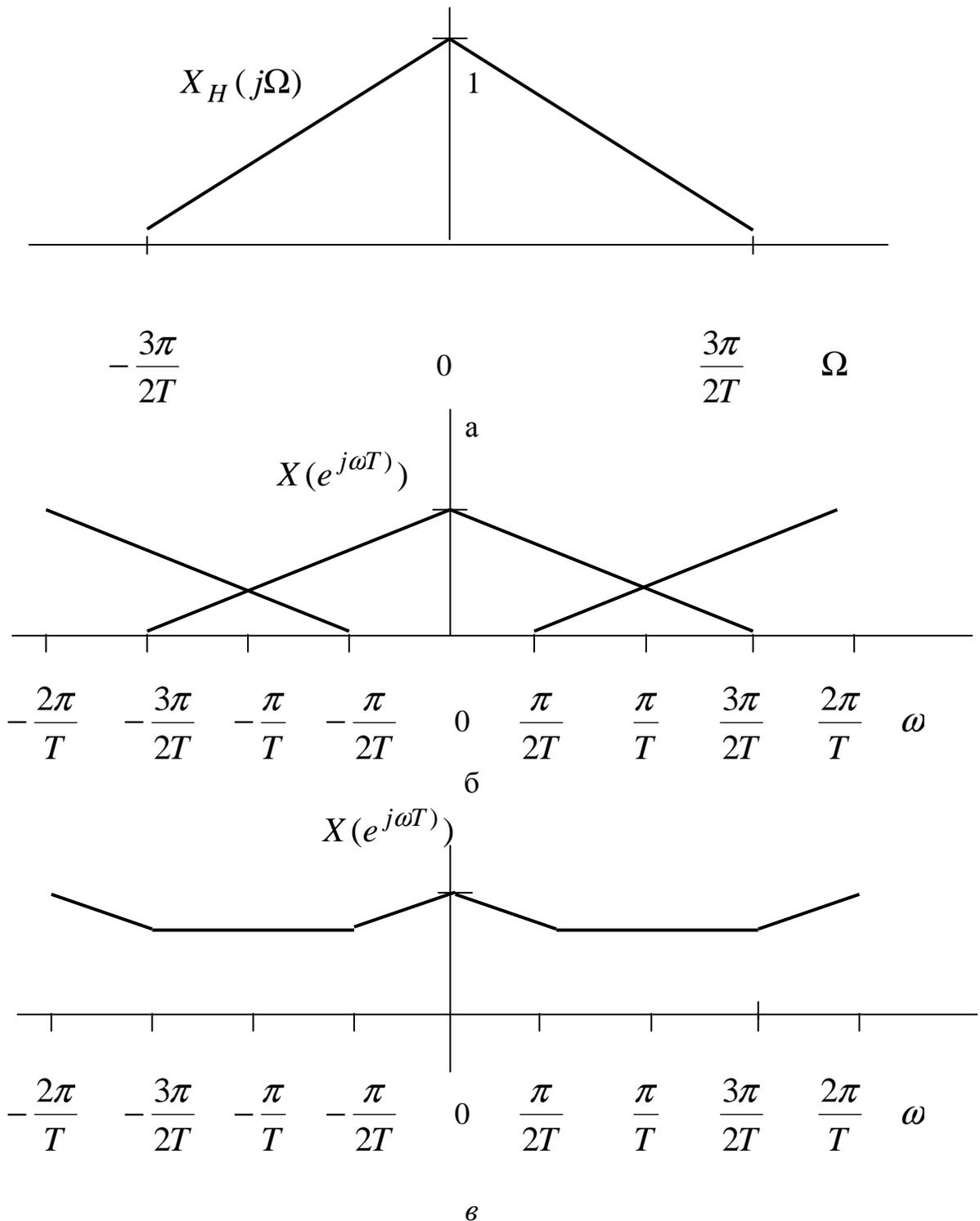


Рисунок 1.4 - Эффекты наложения в спектре дискретизированного сигнала при недостаточной частоте дискретизации

Поэтому в отличие от предыдущего примера спектр последовательности (рисунок 1.4, в) связан со спектром исходного сигнала значительно более сложным образом. Причина заключается в том, что частота дискретизации $f_d = 1/T$ была выбрана недостаточно большой и высокочастотные составляющие спектра $X_H(j\Omega)$ попали в область более низких частот в спектре $X(e^{j\omega T})$. Такое смещение спектральных составляющих из одного диапазона частот в другой называют наложением спектров, а последовательность, соответствующую спектру на рисунке 1.4, в, – представлением сигнала $x(t)$ с наложением.

Таким образом, согласно теореме Котельникова аналоговый сигнал с ограниченным спектром может быть точно (без потери информации) преобразован в дискретный сигнал и затем точно восстановлен по отсчетам этого дискретного сигнала при выполнении условия (1.7). Практически любой аналоговый сигнал имеет ограниченный спектр и поэтому может быть заменен при правильно выбранной частоте дискретизации соответствующим дискретным сигналом.

1.1.3 Связь между дискретными и цифровыми сигналами

Операция квантования и кодирования (аналого-цифрового преобразования) состоит в том, что по заданному дискретному сигналу $x(nT)$ строится цифровой кодированный сигнал $x_u(nT)$, $x(nT) \rightarrow x_u(nT)$ так, что $x_u(nT) \approx x(nT)$, $n = 0, 1, \dots$.

Операция цифроаналогового преобразования состоит в том, что по заданному цифровому кодированному сигналу $x_u(nT)$ строят дискретный сигнал $x(nT)$, $x_u(nT) \rightarrow x(nT)$ так, что $x(nT) = x_u(nT)$.

Операции квантования, кодирования и цифроаналогового преобразования не являются точно взаимно обратными, поскольку квантование в общем случае выполняется с неустранимой погрешностью.

Ошибка при переходе от дискретного сигнала к цифровому, т.е. при выполнении операции квантования, уменьшается при увеличении количества уровней квантования (уменьшении шага квантования) и увеличении числа разрядов для представления отсчетов.

Работу аналого-цифрового преобразователя (АЦП) можно представить в виде двухэтапного процесса (рисунок 1.5).

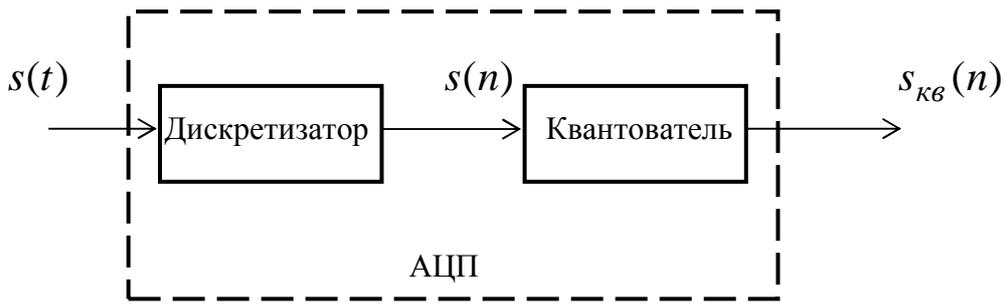


Рисунок 1.5 – Структурная схема аналого-цифрового преобразователя

На первом этапе формируется последовательность $s(n) = s(t) |_{t=nT}$, в которой отсчеты $s(n)$ представлены с неограниченной точностью. На втором этапе значение каждого отсчета $s(n)$ представляется числом, состоящим из конечного числа двоичных разрядов. В результате получается новая последовательность $s_{кв}(n)$. В реальных АЦП обе операции выполняются совместно, т. е. имеется единый блок, на вход которого поступает колебание $s(t)$, а на выходе формируется последовательность $s_{кв}(n)$. Разность $e(n) = s(n) - s_{кв}(n)$ называется шумом квантования или шумом аналого-цифрового преобразования.

Как было отмечено в предыдущем параграфе, полоса входного колебания должна быть ограничена, так как иначе последовательность $s_{кв}(n)$ не будет однозначно представлять $s(t)$. Поэтому аналого-цифровому преобразователю обычно предшествует аналоговый фильтр нижних частот. Желательно, чтобы в полосе сигнала характеристика фильтра имела минимальные пульсации, а подавление составляющих с частотой, превышающей половину частоты дискретизации, составляло не менее 40 дБ. Например, для обработки речевых сигналов достаточно использовать фильтр восьмого порядка (с крутизной спада характеристики 48 дБ на октаву) /6/.

В зависимости от особенностей метода квантования последовательности $s(n)$ шум квантования может иметь то или иное амплитудное распределение. Если наименьший шаг квантования равен Q , то взаимосвязь между $s_{кв}(n)$ и $s(n)$ для случая округления будет описываться характеристикой, представленной на рисунке 1.6. Поскольку число уровней квантования конечно, все отсчеты, выходящие за мак-

симальный (E_{\max}) или минимальный (E_{\min}) уровень, округляются до этих значений. Обычно такого ограничения сигнала сверху и снизу стараются избежать путем соответствующего выбора шага квантования Q уровня входного аналогового сигнала. Из рисунка 1.6 видно, что сигнал ошибки удовлетворяет (за исключением случаев превышения предельных уровней) соотношению

$$-\frac{Q}{2} \leq e(n) \leq \frac{Q}{2} \quad (1.15)$$

при любых n . Используя достаточно общие предположения, можно показать, что распределение сигнала ошибки является равномерным.

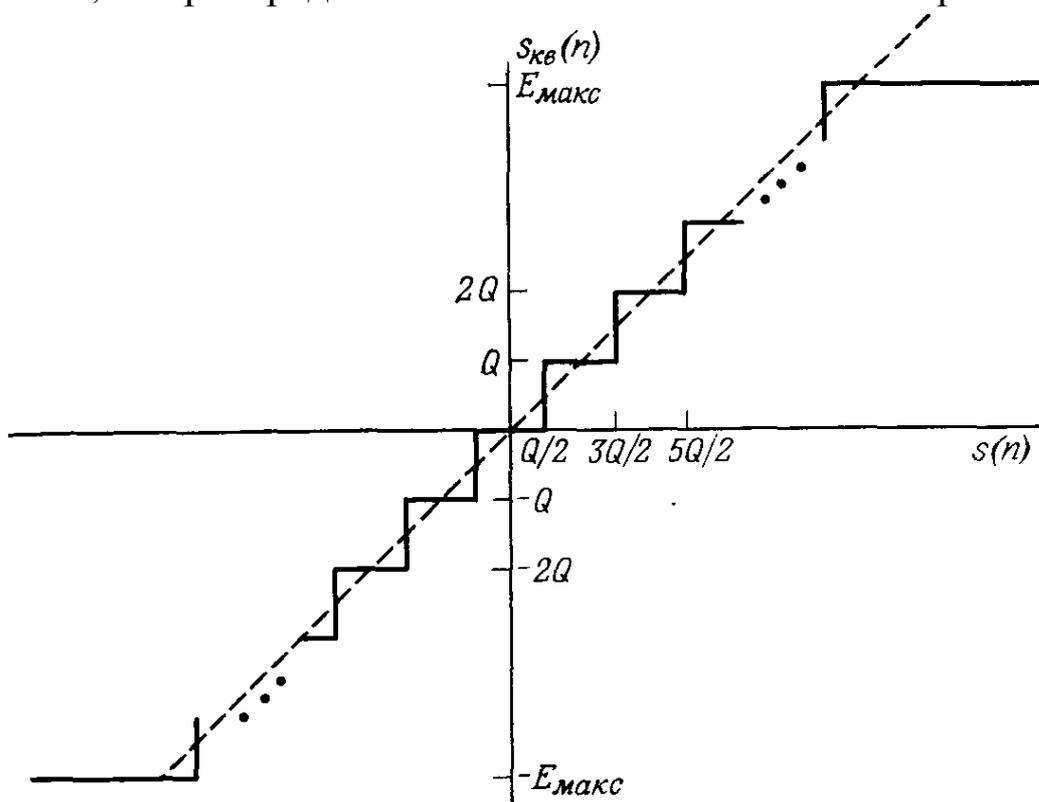


Рисунок 1.6 - Характеристика квантователя при использовании округления

На рисунке 1.7 представлен график плотности вероятности ошибки квантования при округлении $/8/$.

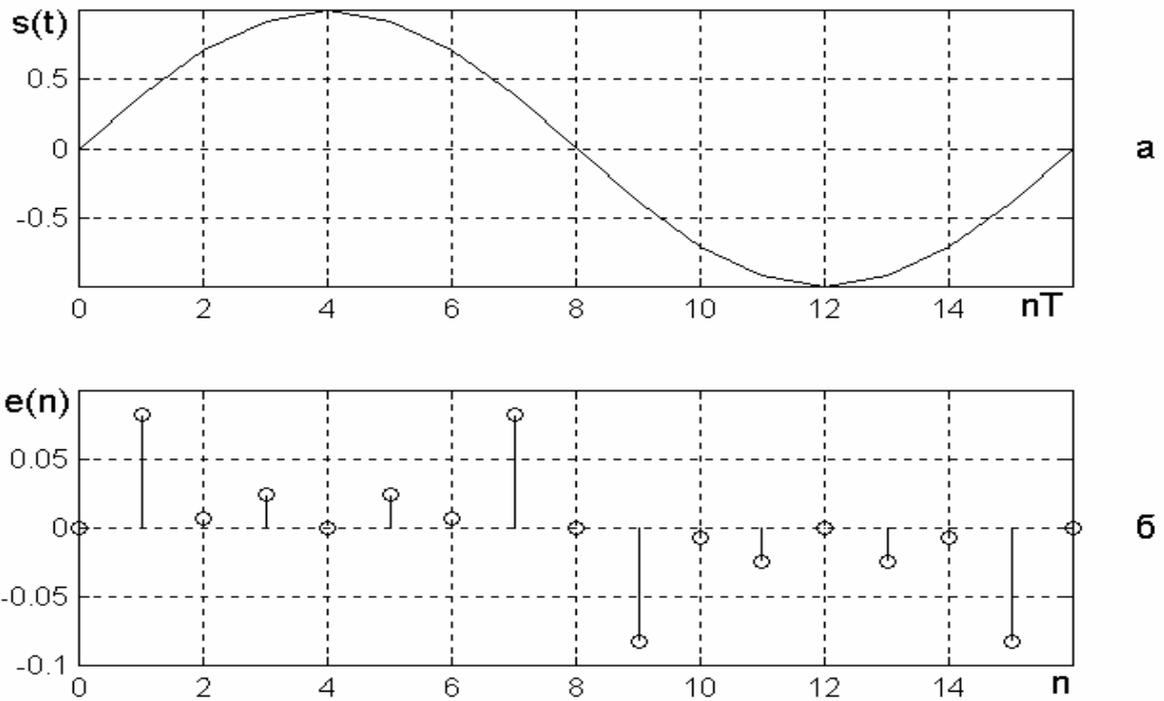


Рисунок 1.7- Плотность вероятности ошибки квантования при округлении

На рисунке 1.8 процесс округления в АЦП иллюстрируется на примере дискретизации аналогового синусоидального колебания $s(t)$. В нижней части рисунка 1.8 представлены ошибки квантования каждого из отсчетов

Как и ожидалось, последовательность, представляющая ошибку, имеет случайный характер.

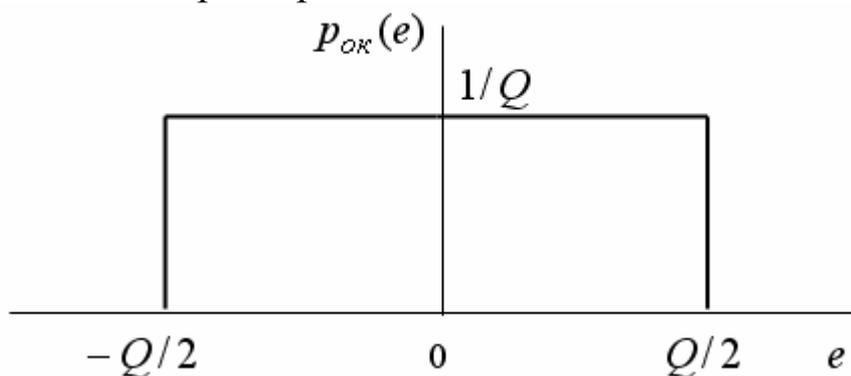


Рисунок 1.8 - Ошибка квантования синусоидального сигнала

При получении отсчетов $s_{кв}(n)$ могут быть использованы и другие способы квантования. Так, при *усечении* в качестве отсчета сигнала используется ближайший меньший уровень квантования. На ри-

сунке 1.9 графически представлено соотношение между $s_{кв}(n)$ и $s(n)$ при усечении. Поскольку результат усечения равен результату округления, уменьшенному на половину шага квантования, то график плотности вероятности сигнала ошибки будет таким, как показано на рисунке 1.10.

Из рисунков 1.7 и 1.10 видно, что ошибка квантования имеет среднее значение, равное нулю при округлении и $Q/2$ при усечении, а ее дисперсия в обоих случаях равна $Q^2/12$.

Любые аналоговые сигналы всегда сопровождаются шумом того или иного вида, т. е. обрабатываемая реализация имеет конечное отношение сигнал/шум. Например, при передаче речевого сигнала по обычной телефонной линии отношение сигнал/шум составляет около 36 дБ, так что никакое увеличение разрядности не позволит получить в дискретизированном сигнале отношение сигнал/шум большее, чем в исходном аналоговом сигнале.

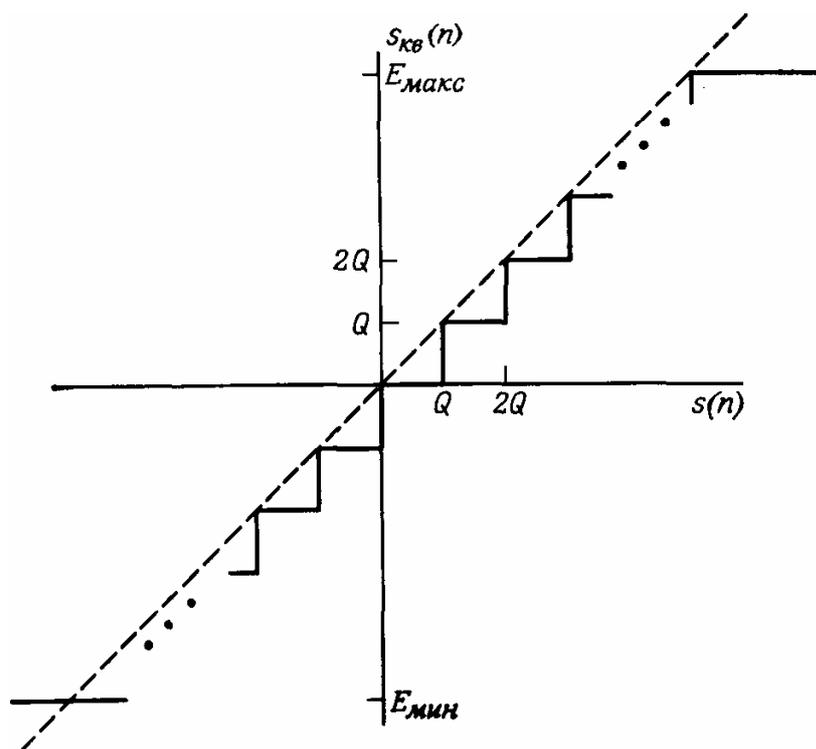


Рисунок 1.9 - Характеристика квантователя при использовании усечения

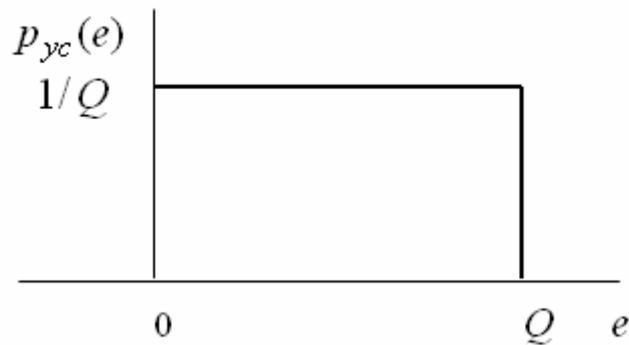


Рисунок 1.10 - Плотность вероятности ошибки усечения

Действительно, если шаг квантования Q значительно меньше амплитуды сигнала, младшие разряды отсчетов последовательности будут всего лишь более точно описывать шум, сопровождающий аналоговый сигнал. Отсюда следует, что увеличение числа разрядов АЦП сверх некоторой величины приводит лишь к увеличению точности представления входного аналогового шума. Итак, разрядность АЦП определяется характером преобразуемого сигнала.

Восстановление аналогового сигнала из цифрового осуществляется цифроаналоговым преобразователем (ЦАП).

Цифроаналоговый преобразователь представляет собой устройство, служащее для преобразования последовательности $y(n)$ в аналоговое колебание $y(t)$ вида

$$y(t) = \sum_n y(n)h(t - nT), \quad (1.16)$$

причем вид интерполирующей функции $h(t)$ определяется видом цифроаналогового преобразователя. Одной из наиболее распространенных интерполирующих функций $h(t)$ является прямоугольный импульс длительностью T секунд. Такие ЦАП обычно называют цифроаналоговыми преобразователями с интерполятором нулевого порядка. На рисунке 1.11 изображены типичная последовательность $y(n)$ и аналоговый сигнал $y(t)$ на выходе ЦАП с интерполятором нулевого порядка, рассчитанный по формуле (1.16).

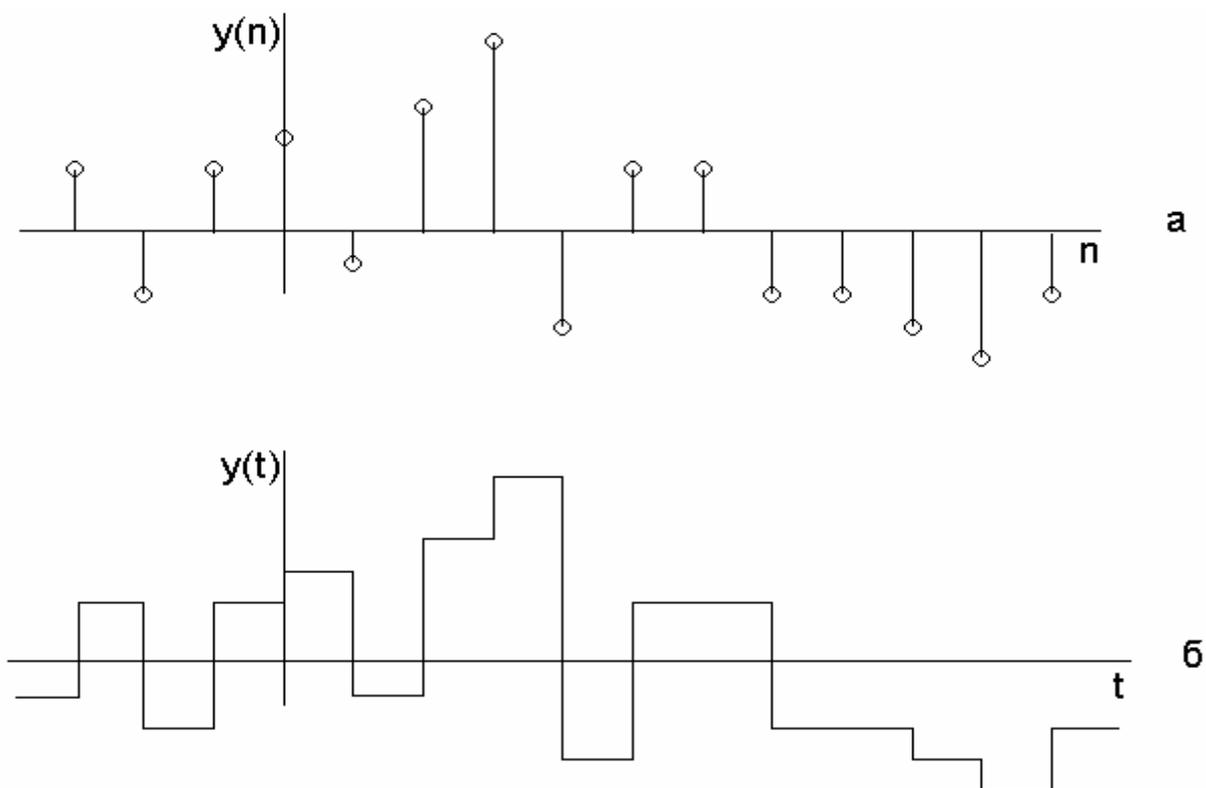


Рисунок 1.11 - Входной (а) и выходной (б) сигналы ЦАП с интерполятором нулевого порядка

Ясно, что выходное аналоговое колебание $y(t)$ содержит большое количество нежелательных высокочастотных составляющих. Для их подавления после ЦАП обычно включают аналоговый линейный фильтр нижних частот с постоянными параметрами и частотой среза не выше $1/(2T)$ Гц, где $1/T$ — частота дискретизации. Такой фильтр называют выходным, а комбинацию из ЦАП и выходного фильтра—восстанавливающим фильтром.

Цифроаналоговое преобразование в принципе позволяет без ошибки восстановить аналоговое колебание, эквивалентное (в некотором смысле) входной последовательности ЦАП. Из формулы (1.16), однако, следует, что $Y(\omega)$, спектр выходного колебания $y(t)$, не будет совпадать в полосе $0 \leq f \leq 1/(2NT)$ со спектром $Y_D(e^{j\omega})$ входной последовательности $y(n)$, так как, $Y(\omega)$ и $Y_D(e^{j\omega})$ связаны соотношением

$$Y(\omega) = Y_D(e^{j\omega})H(\omega), \quad (1.17)$$

где $H(\omega)$ — спектр $h(t)$.

Для интерполятора нулевого порядка

$$|H(\omega)| = \frac{2 \sin(\omega T / 2)}{\omega}. \quad (1.18)$$

Чтобы скомпенсировать искажение спектра, вносимое неравномерностью частотной характеристики ЦАП, последовательность $y(n)$ часто предварительно пропускают через цифровой фильтр, амплитудная характеристика которого аппроксимирует функцию

$$|G(e^{j\omega})| = \frac{\omega}{2 \sin(\omega T / 2)} = \frac{1}{|H(\omega)|}. \quad (1.19)$$

Таким образом, последовательное соединение устройств с характеристиками $Y_D(e^{j\omega})$ и $H(\omega)$ в целом обеспечивает равномерную частотную характеристику. На рисунке 1.12 показаны все операции, необходимые для перехода от последовательности $y(n)$ к аналоговому колебанию $y(t)$ со спектром, эквивалентным в полосе $-1/(2T) \leq f \leq 1/(2T)$ спектру исходной последовательности.

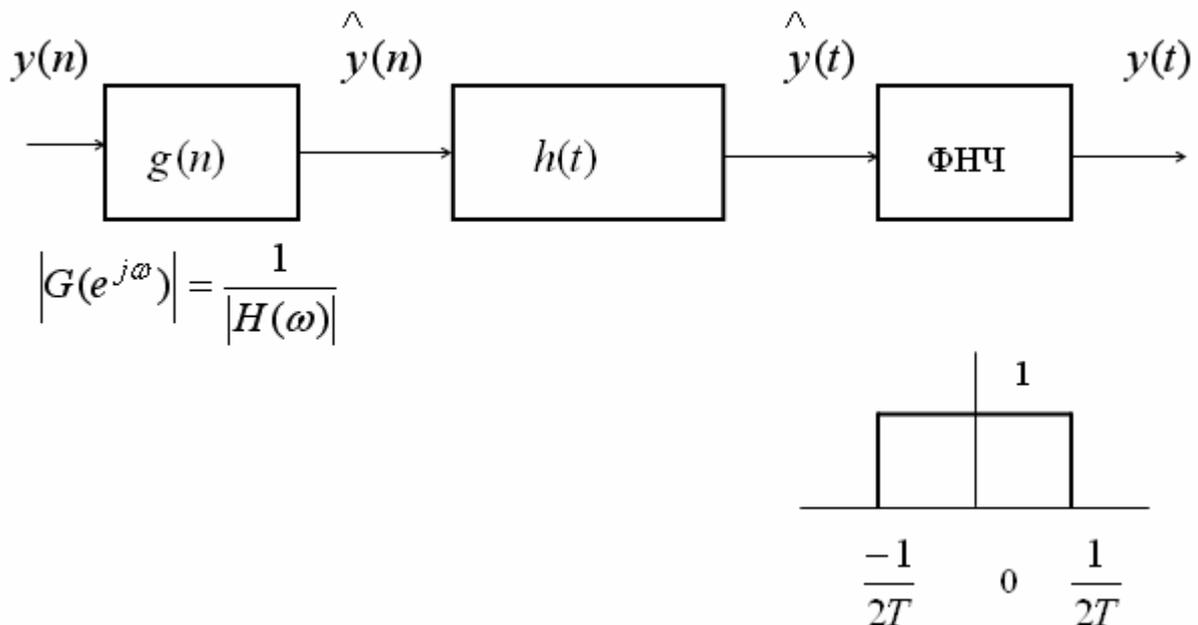


Рисунок 1.12 - Структурная схема цифроаналогового преобразователя с компенсацией частотной характеристики

Следует иметь в виду, что цифровой компенсирующий фильтр, показанный на рисунке 1.12, не обязательно выполнять в виде отдельного устройства, его можно включить в состав самой системы цифровой обработки.

1.2 Основы теории дискретных (цифровых) линейных систем

1.2.1 Линейные системы с постоянными параметрами

Цифровая система обработки сигналов по существу является алгоритмом преобразования одной последовательности (называемой *входной*) в другую (называемую *выходной*). Простое представление системы ЦОС дано на рисунке 1.13. Входная последовательность обозначена через $x(n)$, а выходная – через $y(n)$. Функционально они связаны соотношением

$$y(n) = \psi[x(n)],$$

где вид оператора $\psi(\bullet)$ зависит от свойств конкретной системы.

Линейная система определяется следующим образом /6/. Если $x_1(n)$ и $x_2(n)$ — некоторые входные последовательности, а $y_1(n)$ и $y_2(n)$ — соответствующие им отклики линейной системы, то при подаче на вход последовательности $ax_1(n) + bx_2(n)$ на выходе образуется последовательность $ay_1(n) + by_2(n)$ (a и b — произвольные постоянные).

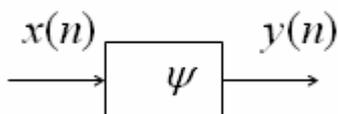


Рисунок 1.13 - Представление системы ЦОС

Система с *постоянными параметрами* (рисунок 1.14) характеризуется тем, что если входной последовательности $x(n)$ соответствует выходная последовательность $y(n)$, то входной последовательности $x(n - n_0)$ при любых n_0 соответствует на выходе последовательность $y(n - n_0)$.

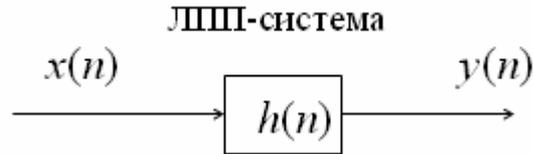


Рисунок 1.14. Представление линейной системы с постоянными параметрами

В линейной системе с постоянными параметрами (ЛПП-система) входная и выходная последовательности связаны соотношением типа свертки. Допустим, что $x(n)$ — входная, а $y(n)$ — выходная последовательности ЛПП-системы, и пусть $h(n)$ — отклик системы на единичный импульс (импульсная характеристика системы). Используя формулу (1.5), $x(n)$ можно записать в виде

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)u_0(n-m).$$

Поскольку $h(n)$ является откликом системы на последовательность $u_0(n)$, а параметры системы постоянны, $h(n-m)$ будет откликом на последовательность $u_0(n-m)$. Из свойства линейности следует, что откликом на последовательность $x(m) \cdot u_0(n-m)$ должна быть последовательность $x(m)h(n-m)$. Поэтому отклик на $x(n)$ будет равен

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m). \quad (1.20)$$

Он имеет вид свертки входного сигнала с отчетами импульсной характеристики системы. Простой заменой переменных равенство (1.20) может быть преобразовано к виду

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m). \quad (1.21)$$

Таким образом, последовательность $h(n)$ полностью описывает ЛПП-систему.

1.2.2 Физическая реализуемость. Устойчивость дискретных систем

ЛПП-систему называют *физически реализуемой*, если величина отклика при $n = n_0$ зависит только от отсчетов входной последовательности с номерами $n \leq n_0$. Для ЛПП-системы это означает, что импульсная характеристика $h(n)$ равна нулю при $n < 0$. Как мы увидим в дальнейшем, существует несколько систем, имеющих важное значение, но физически нереализуемых. К ним относятся идеальный фильтр нижних частот и идеальный дифференциатор. Поэтому значительная часть теории фильтров посвящена методам аппроксимации физически нереализуемых систем реализуемыми системами.

ЛПП-система называется *устойчивой*, если при любой ограниченной входной последовательности выходная последовательность также ограничена. Необходимым и достаточным условием устойчивости системы является следующее требование к импульсной характеристике:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty. \quad (1.22)$$

Необходимость и достаточность этого условия нетрудно показать. Предположим сначала, что условие (1.22) не удовлетворяется, т. е.

$$\sum_{n=-\infty}^{\infty} |h(n)| = \infty.$$

Рассмотрим ограниченную последовательность:

$$x(n) = \begin{cases} +1 & \text{при } h(-n) \geq 0, \\ -1 & \text{при } h(-n) < 0. \end{cases}$$

Согласно формулам (1.20 и 1.21), при $n=0$ отклик равен

$$y(0) = \sum_{m=-\infty}^{\infty} x(m)h(-m) = \sum_{m=-\infty}^{\infty} |h(-m)| = \sum_{m=-\infty}^{\infty} |h(m)| = \infty.$$

Таким образом, последовательность $y(0)$ не ограничена, так что неравенство (1.22) является необходимым условием устойчивости системы. Для доказательства достаточности предположим, что усло-

вие (1.22) выполняется, а на вход поступает ограниченная последовательность $x(n)$, т. е. $|x(n)| \leq M$.

Из формул (1.20, 1.21) получаем

$$|y(n)| = \left| \sum_{m=-\infty}^{\infty} x(m)h(n-m) \right| \leq \sum_{m=-\infty}^{\infty} |x(m)||h(n-m)| \leq M \sum_{m=-\infty}^{\infty} |h(n-m)| < \infty.$$

Последовательность $y(n)$ ограничена, поэтому система устойчива, что и требовалось доказать. На рисунке 1.15, а, б даны примеры импульсных характеристик устойчивой и неустойчивой систем.

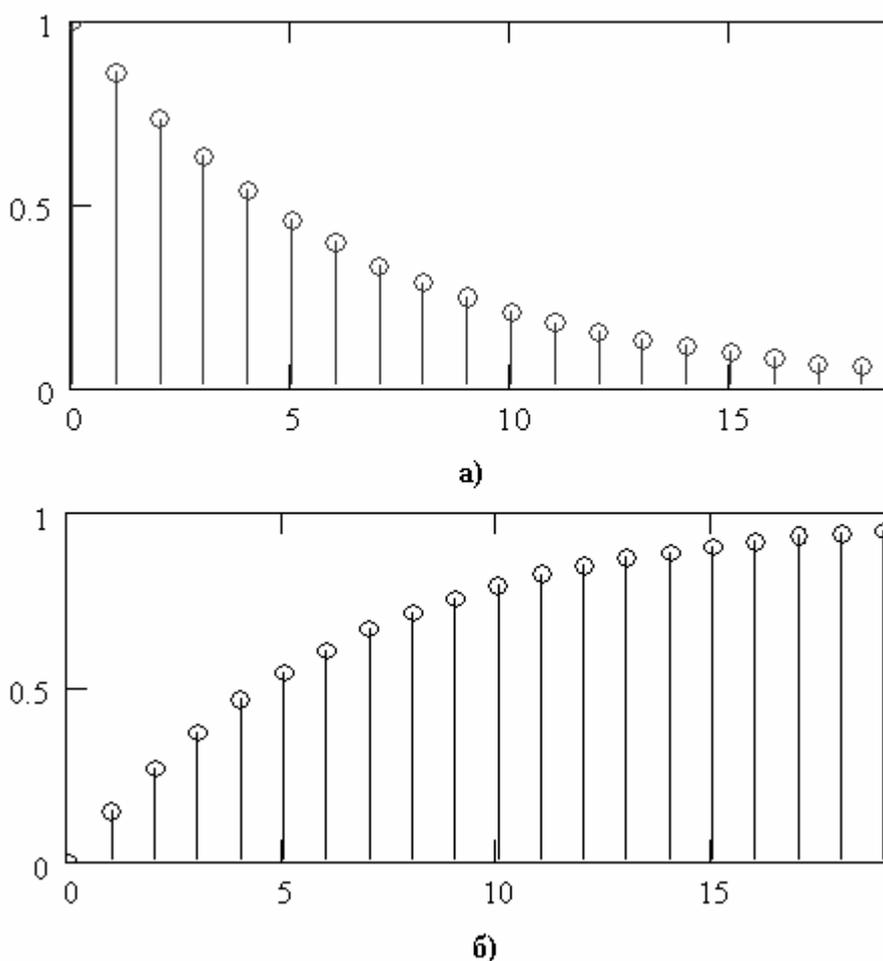


Рисунок 1.15- Импульсная характеристика устойчивой (а) и неустойчивой (б) систем

Импульсная характеристика, приведенная на рисунке 1.15, а, имеет вид $h(n) = \alpha^n u_{-1}(n)$, причем $0 < \alpha < 1$, поэтому условие (1.22) удовлетворяется и система устойчива. Выражение для импульсной

характеристики на рисунке 1.15, b имеет тот же вид, но $\alpha > 1$, поэтому условие (1.22) не выполняется и система неустойчива.

1.2.3 Разностные уравнения

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 и их кратность, порядок системы, частоты, соответствующие нулевому коэффициенту передачи, и т. д.

В самом общем случае линейное разностное уравнение M -го порядка с постоянными коэффициентами, относящееся к физически реализуемой системе, имеет вид /5,6/

$$y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{i=1}^M a_i y(n-i), \quad n \geq 0, \quad (1.23)$$

где коэффициенты $\{b_i\}$ и $\{a_i\}$ описывают конкретную систему, причем $a_M \neq 0$. Каким именно образом порядок системы M характеризует математические свойства разностного уравнения, будет показано ниже. Уравнение (1.23) записано в виде, удобном для решения методом прямой подстановки. Имея набор начальных условий (например, $x(i)$, $y(i)$ для $i = -1, -2, \dots, -M$) и входную последовательность $x(n)$, по формуле (1.23) можно непосредственно вычислить выходную последовательность $y(n)$ для $n \geq 0$. Например, разностное уравнение

$$y(n) = x(n) - 3y(n-1) \quad (1.24)$$

с начальным условием $y(-1) = 0$ и $x(n) = n^2 + n$ можно решить подстановкой, что дает

$$\begin{aligned}
y(0) &= x(0) - 3y(-1) = 0, \\
y(1) &= x(1) - 3y(0) = 2, \\
y(2) &= x(2) - 3y(1) = 0, \\
y(3) &= x(3) - 3y(2) = 12, \\
y(4) &= x(4) - 3y(3) = -16, \\
y(5) &= x(5) - 3y(4) = 78, \\
y(6) &= x(6) - 3y(5) = -192, \\
&\dots \quad \dots \quad \dots \quad \dots
\end{aligned}$$

Хотя решение разностных уравнений прямой подстановкой и целесообразно в некоторых случаях, значительно полезнее получать решение уравнения в явном виде. Методы нахождения таких решений подробно освещены в литературе по разностным уравнениям, и здесь будет дан лишь краткий обзор. 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 (1.24). Однородное уравнение имеет вид

$$y(n) + 3y(n-1) = 0. \quad (1.25)$$

Известно, что характеристическими решениями однородных уравнений, соответствующих линейным разностным уравнениям с постоянными коэффициентами, являются решения вида $A\alpha^n$. Поэтому, подставив в уравнение (1.25) $A\alpha^n$ вместо $y(n)$, получим

$$\begin{aligned}
A\alpha^n + 3A\alpha^{n-1} &= 0, \\
A\alpha^{n-1}(\alpha + 3) &= 0, \\
\alpha &= -3, \\
y_0(n) &= A(-3)^n.
\end{aligned}$$

Частное решение, соответствующее входной последовательности $x(n) = n^2 + n$, попробуем найти в виде

$$y_c(n) = Bn^2 + Cn + D.$$

Из уравнения (1.25) получаем

$$Bn^2 + Cn + D + 3B(n-1)^2 + 3C(n-1) + 3D = n^2 + n.$$

Поскольку коэффициенты при равных степенях n должны совпадать, B, C и D должны быть равны

$$B = \frac{1}{4}, \quad C = \frac{5}{8}, \quad D = \frac{9}{32}.$$

Таким образом, общее решение имеет вид

$$y(n) = \frac{n^2}{4} + \frac{5n}{8} + \frac{9}{32} + A(-3)^n.$$

Коэффициент A определяется из начального условия $y(-1) = 0$, откуда $A = -9/32$ и

$$y(n) = \frac{n^2}{4} + \frac{5n}{8} + \frac{9}{32} [1 - (-3)^n]$$

Выборочная проверка этого решения при $n \geq 0$ показывает полное его совпадение с приведенным выше прямым решением.

Очевидное преимущество решения 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 рисунке 1.16. Блок «задержка» осуществляет задержку на один отсчет. Рассмотренная форма построения системы, в которой для входной и выход-

ная форма построения системы, в которой для входной и выходной последовательностей используются отдельные элементы задержки, называется прямой формой. Ниже мы обсудим различные методы построения этой и других цифровых систем.

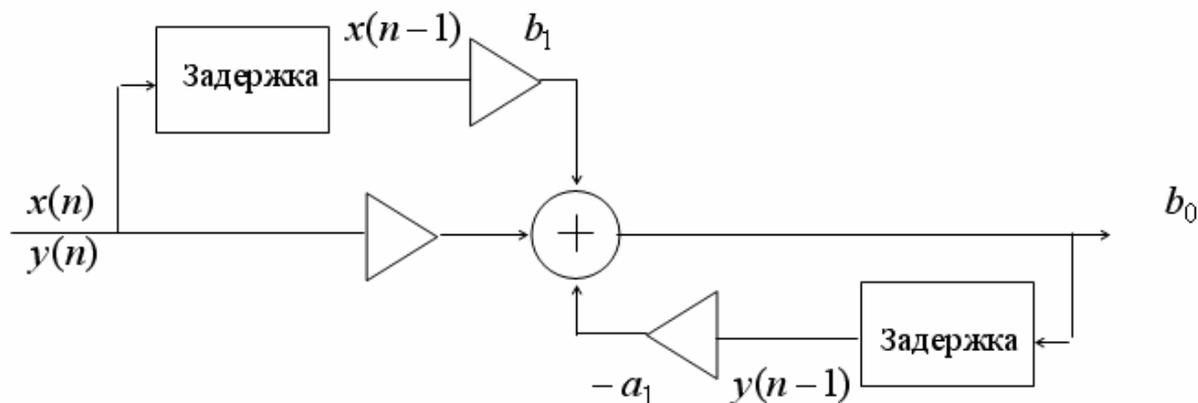


Рисунок 1.16 - Схема реализации простого разностного уравнения первого порядка

Разностное уравнение второго порядка самого общего вида $y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$ может быть реализовано с помощью схемы, приведенной на рисунке 1.17.

В этой схеме для входной и выходной последовательностей также используются отдельные элементы задержки.

Системы первого и второго порядков могут быть использованы при реализации систем более высокого порядка, так как последние могут быть представлены в виде последовательного или параллельного соединения систем первого и второго порядка.

1.3 Частотные характеристики цифровых устройств

Для определения характеристик ЛПП-систем используются специальные испытательные сигналы. В частности, для описания ЛПП – систем в частотной области используется специальный класс входных последовательностей, имеющих вид $x(n) = e^{j\omega n}$. Как будет показано, этот класс последовательностей является набором собственных функций ЛПП – систем дискретного времени, т.е. выходная последовательность совпадает с входной, умноженной на некоторый комплексный коэффициент, зависящий только от частоты ω .

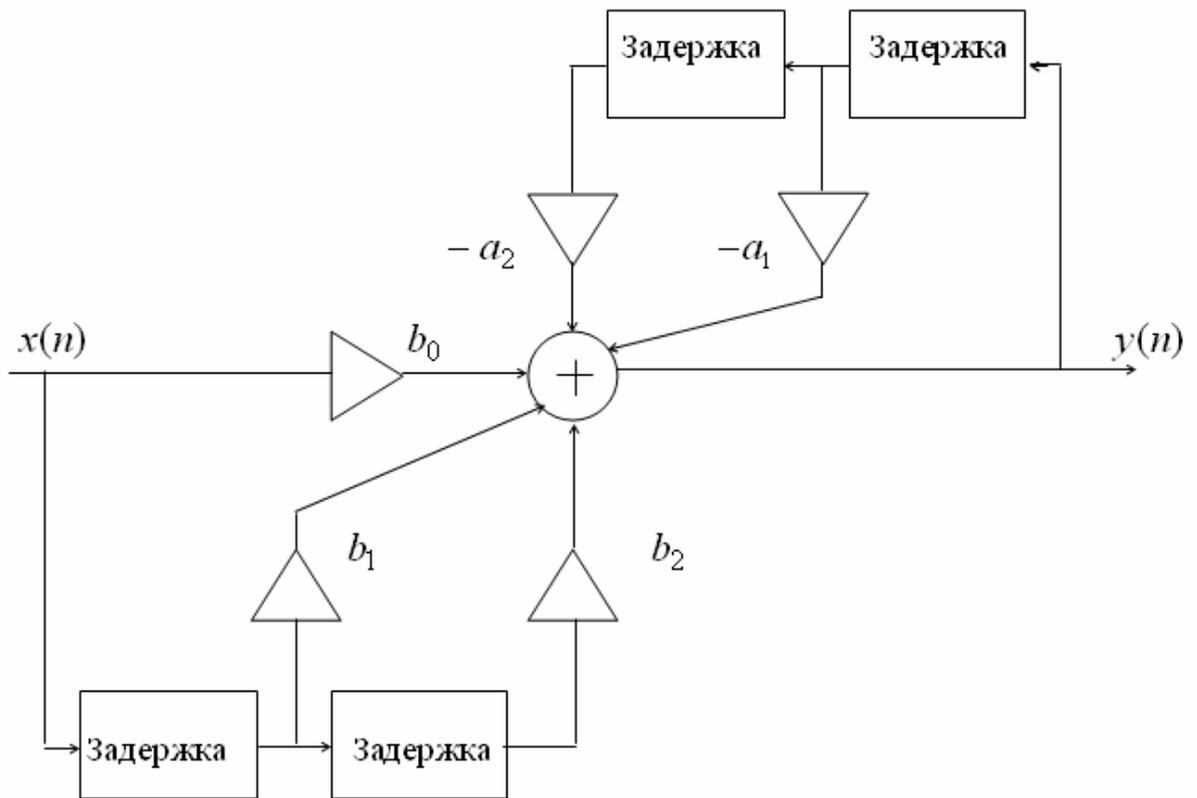


Рисунок 1.17 - Схема реализации разностного уравнения второго порядка

Рассмотрим класс входных последовательностей вида

$$x(n) = e^{j\omega n}, \quad -\infty < n < \infty.$$

Если такая последовательность поступает на вход ЛПП – системы с импульсной характеристикой $h(n)$, то на выходе в соответствии с выражением

$$y(n) = \sum_{m=-\infty}^{\infty} h(m) \cdot x(n-m),$$

появится последовательность

$$y(n) = \sum_{m=-\infty}^{\infty} h(m) \cdot e^{j\omega(n-m)} = e^{j\omega n} \sum_{m=-\infty}^{\infty} h(m) \cdot e^{-j\omega m} = x(n) \cdot H(e^{j\omega}).$$

Таким образом, для выбранного класса входных последовательностей отклик совпадает с входной последовательностью с точностью

до комплексного множителя $H(e^{j\omega})$, который выражается через импульсную характеристику системы следующим образом:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n) \cdot e^{-j\omega n}. \quad (1.26)$$

Поскольку последовательность вида $e^{j\omega n}$ функционально эквивалентна дискретизированной синусоиде с частотой ω , то множитель $H(e^{j\omega})$ называют частотной характеристикой системы, так как он представляет коэффициент передачи ЛПП – системы для каждого значения ω .

При всем многообразии ЛПП – систем они могут быть представлены каскадным соединением систем первого и второго порядков. Поэтому целесообразно рассмотреть частотные характеристики ЛПП – систем первого и второго порядков.

Разностное управление системы первого порядка имеет вид:

$$y(n) = x(n) + Ky(n-1),$$

с начальным условием $y(-1) = 0$. Легко установить, что ее импульсная характеристика равна

$$h(n) = \begin{cases} K^n, & n \geq 0, \\ 0, & n < 0. \end{cases}$$

Используя формулу (1.26), найдем частотную характеристику системы первого порядка

$$H(e^{j\omega}) = \sum_{n=0}^{\infty} h(n) \cdot e^{-j\omega n} = \sum_{n=0}^{\infty} K^n \cdot e^{-j\omega n} = \sum_{n=0}^{\infty} (K \cdot e^{-j\omega})^n. \quad (1.27)$$

При $|K| < 1$ сумма геометрической прогрессии (1.27) будет равна

$$H(e^{j\omega}) = \frac{1}{1 - K \cdot e^{-j\omega}}.$$

Представив $H(e^{j\omega})$ в виде

$$H(e^{j\omega}) = \frac{H_1(e^{j\omega})}{H_2(e^{j\omega})}$$

с учетом

$$\begin{aligned} H(e^{j\omega}) &= |H(e^{j\omega})| \cdot e^{j \arg H(e^{j\omega})} = \\ &= |H(e^{j\omega})| \cdot [\cos(\arg H(e^{j\omega})) + j \sin(\arg H(e^{j\omega}))] \end{aligned}$$

получим:

$$\begin{aligned} H(e^{j\omega}) &= \frac{|H_1(e^{j\omega})|}{|H_2(e^{j\omega})|} = \frac{1}{|1 - K \cdot (\cos \omega - j \sin \omega)|} = \\ &= \frac{1}{\sqrt{(1 - K \cos \omega)^2 + (K \sin \omega)^2}} = \\ &= \frac{1}{\sqrt{1 - 2K \cos \omega + K^2 \cos^2 \omega + K^2 \sin^2 \omega}} = \\ &= \frac{1}{\sqrt{1 - 2K \cos \omega + K^2}} \quad , \end{aligned}$$

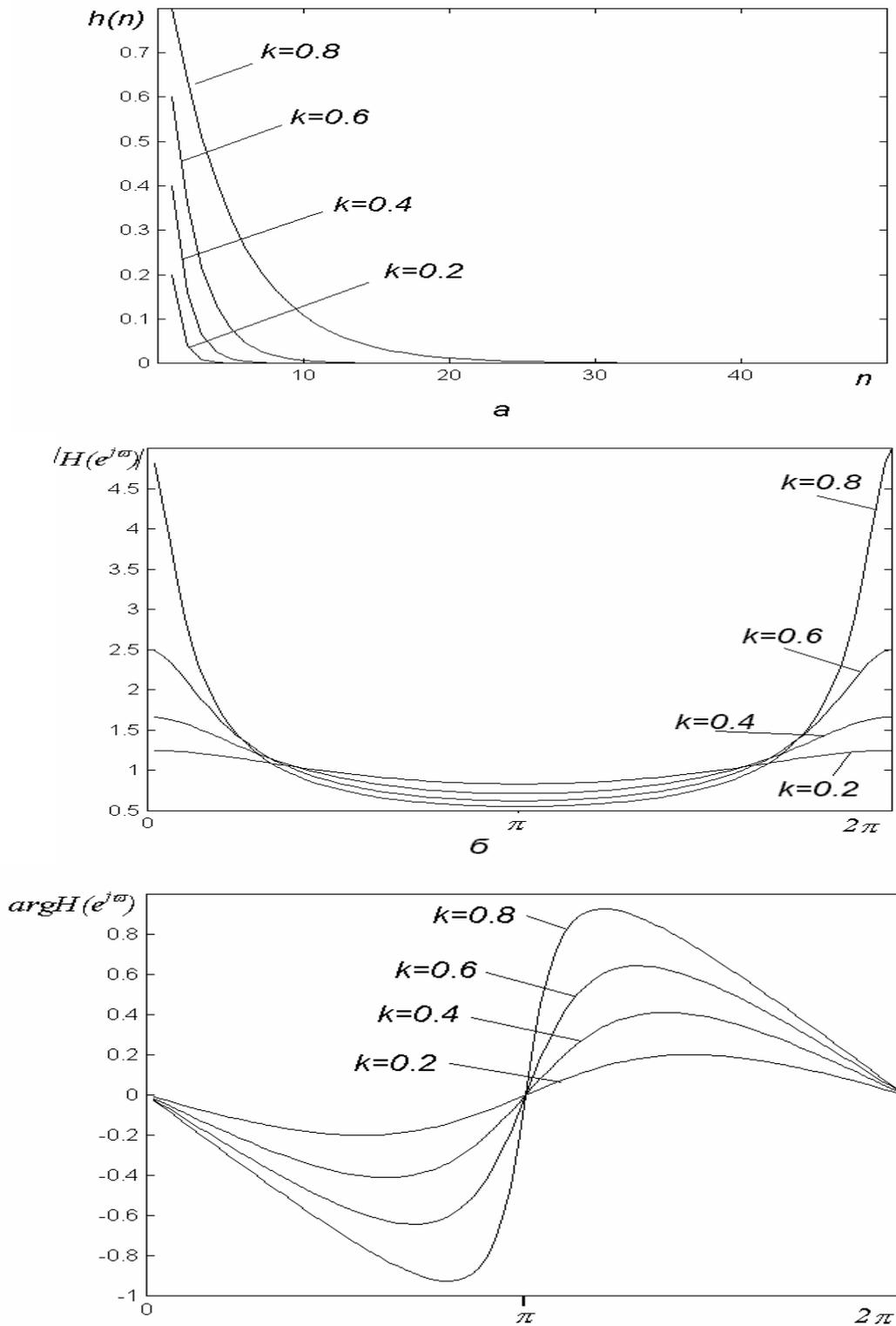
$$\begin{aligned} \arg H(e^{j\omega}) &= \arg(H_1(e^{j\omega})) - \arg(H_2(e^{j\omega})) = \\ &= 0 - \operatorname{argtg} \frac{K \sin \omega}{1 + K \cos \omega} = \operatorname{argtg} \frac{K \sin \omega}{1 + K \cos \omega} . \end{aligned}$$

Зависимость модуля комплексной частотной характеристики от частоты называется амплитудно-частотной характеристикой (АЧХ), а зависимость аргумента $H(e^{j\omega})$ от частоты - фазочастотной характеристикой (ФЧХ).

На рисунке 1.18 приведены графики импульсной характеристики, АЧХ и ФЧХ ЛПП – систем первого порядка для нескольких значений K как функции частоты ω в диапазоне $0 \leq \omega \leq 2\pi$.

Разностное уравнение системы второго порядка можно записать в виде:

$$y(n) = x(n) + a_1 y(n-1) + a_2 y(n-2) \quad . \quad (1.28)$$



б

Рисунок 1.18 – Импульсная характеристика (а), АЧХ (б) и ФЧХ (в) ЛПП-системы первого порядка

В общем случае уравнение второго порядка содержит также члены в виде $b_1x(n-1)$ и $b_2x(n-2)$, однако для простоты изложения

эти члены опущены. Если снова ввести начальные условия $y(-1) = 0$ и $y(-2) = 0$, то нетрудно показать, что импульсная характеристика системы может принять одну из двух форм:

$$h(n) = \alpha_1(p_1)^n + \alpha_2(p_2)^n, \quad (1.29)$$

где p_1 и p_2 - действительные числа, либо

$$h(n) = \alpha_1 \cdot r^n \cdot \sin(b_n + \varphi). \quad (1.30)$$

Импульсная характеристика вида (1.29) описывает две системы первого порядка и убывает как p_1^n и p_2^n . Выражение (1.30) описывает систему второго порядка, импульсная характеристика которой является затухающей синусоидой. Частотная характеристика, соответствующая выражению (1.30), может быть записана следующим образом:

$$H(e^{j\omega}) = \frac{1}{1 - 2r(\cos b) \cdot e^{-j\omega} + r^2 \cdot e^{-2j\omega}}.$$

Амплитудные и фазовые характеристики системы второго порядка, соответствующие фиксированному значению $b = \pi/4$ и различным r , приведены на рисунке 1.19.

Отметим некоторые свойства частотной характеристики. Нетрудно заметить, что частотная характеристика является периодической функцией ω , причем ее период равен 2π . Эта периодичность связана со спецификой дискретизированного колебания: входная последовательность с частотой $\omega + 2m\pi$ (где $m = \pm 1, \pm 2, \dots$) не отличается от входной последовательности с частотой ω , т.е.

$$\begin{aligned} \hat{x}(n) &= e^{j(\omega+2m\pi)n} = e^{j(\omega n+2m\pi n)} = e^{j\omega n} \cdot e^{j2m\pi n} = \\ &= e^{j\omega n} (\cos(2\pi mn) + j \sin(2\pi mn)) = \\ &= e^{j\omega n} (1 + j0) = e^{j\omega n} = x(n). \end{aligned}$$

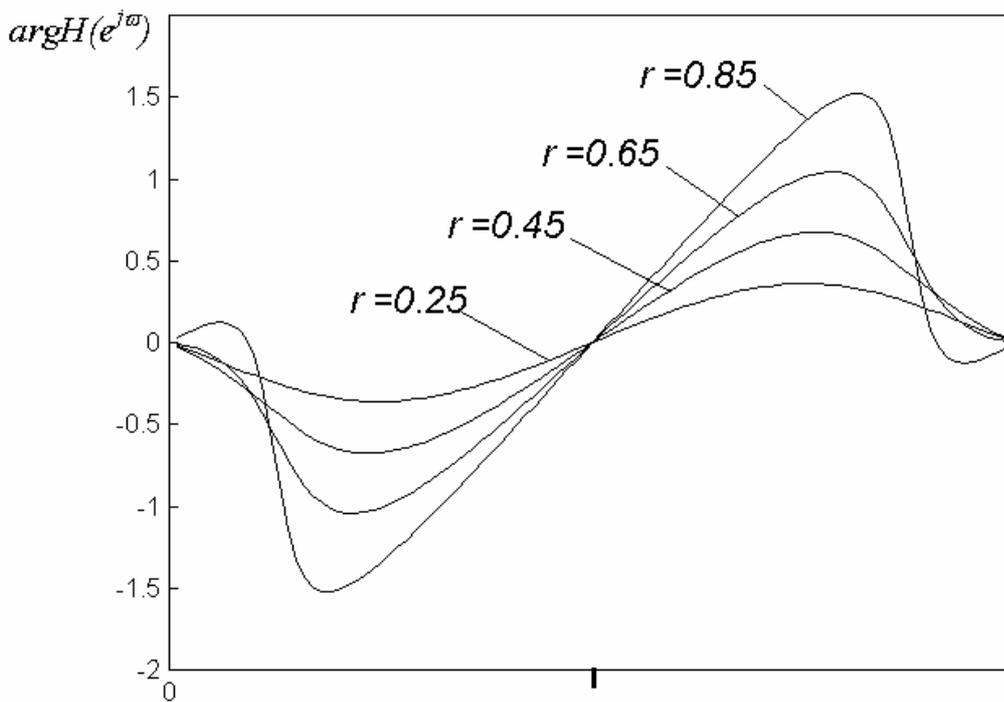
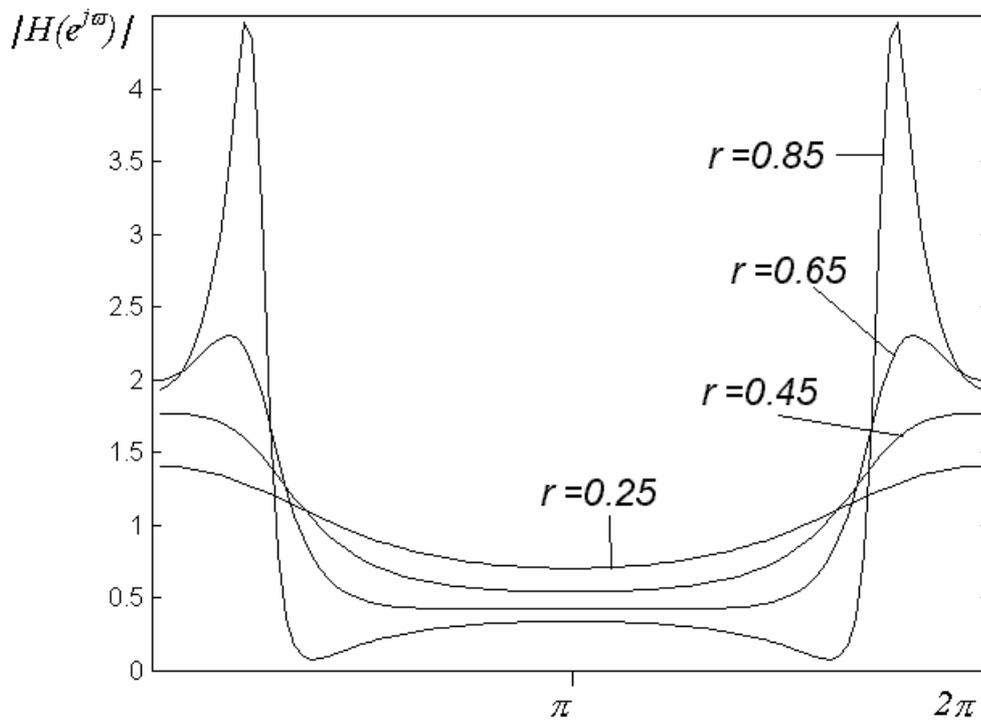


Рисунок 1.19 – АЧХ (а) и ФЧХ (б) ЛПП-системы второго порядка

Поскольку $H(e^{j\omega})$ - периодическая функция, то для полного описания достаточно задать ее на любом интервале длиной 2π . Обычно для этой цели используют интервал $0 \leq \omega \leq 2\pi$.

Другим важным свойством частотной характеристики является то, что для действительных $h(n)$ (как обычно и бывает на практике) модуль $H(e^{j\omega})$ симметричен, а фаза $H(e^{j\omega})$ - антисимметрична на интервале $0 \leq \omega \leq 2\pi$. Аналогично действительная часть $H(e^{j\omega})$ симметрична, а мнимая – антисимметрична на том же интервале. Поэтому при действительных импульсных характеристиках интервал частот, на котором задают частотную характеристику, обычно сокращают до $0 \leq \omega \leq 2\pi$.

Часто возникает необходимость представления импульсной и частотной характеристик с учетом интервалов дискретизации T . В этом случае выражение (1.26) примет вид:

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h(nT)e^{-j\omega nT}.$$

Функция $H(e^{j\omega T})$ периодична по частоте ω с периодом, равным $2\pi/T$. Частота в этом случае выражается в радианах в секунду. Заменой вида $\omega = 2\pi f$ можно выразить $H(e^{j\omega T})$ в герцах.

Как будет показано в одной из следующих глав, величина собственных шумов определяется порядком цифрового устройства обработки сигналов. Для уменьшения шумов рекомендуется каскадное соединение звеньев не выше второго порядка.

Наибольшее распространение нашли последовательное, параллельное соединения и соединение в цепи обратной связи.

Эквивалентная частотная характеристика последовательного соединения двух звеньев (рисунок 1.20, а) имеет вид:

$$H(j\omega) = H_1(j\omega) \cdot H_2(j\omega).$$

Для параллельного соединения (рисунок 1.20, б) используется следующее выражение:

$$H(j\omega) = H_1(j\omega) + H_2(j\omega).$$

При включении одного из звеньев в цепь обратной связи (рисунок 1.20, в) частотная характеристика определяется выражением:

$$H(j\omega) = \frac{H_1(\omega)}{1 - H_1(\omega) \cdot H_2(\omega)}.$$

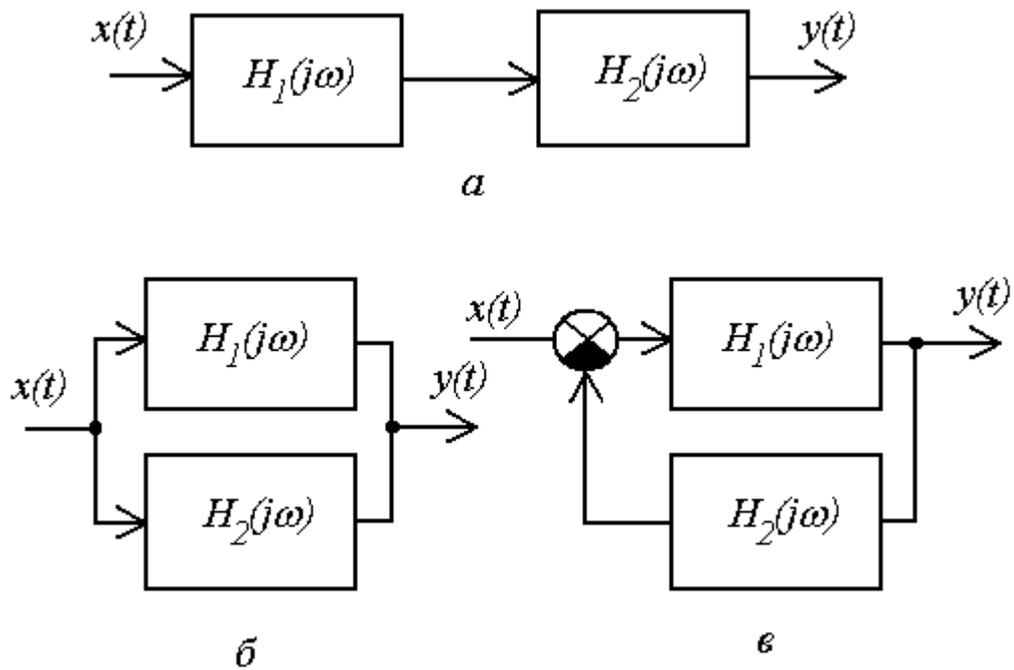


Рисунок 1.20 – Каскадное соединение звеньев

1. 4 Дискретные линейная и круговая свертки

Многие алгоритмы цифровой обработки сигналов представляют собой свертку этих сигналов с импульсной характеристикой устройства обработки. Для определения дискретной свертки используют несколько алгоритмов, называемых, соответственно, круговой, линейной и секционированной свертками [5..8].

Пусть $x(nT)$ и $h(nT)$ - две периодические последовательности с периодом по N отсчетов (T - период дискретизации). Круговой (периодической или циклической) сверткой таких последовательностей называется последовательность $y(nT)$, определяемая соотношением

$$y(nT) = \sum_{l=0}^{N-1} h(lT)x((n-l)T) \quad (1.33)$$

или эквивалентным соотношением

$$y(nT) = \sum_{l=0}^{N-1} x(lT)h((n-l)T) \quad (1.34)$$

Последовательность $y(nT)$ также является периодической с периодом в N отсчетов, поэтому достаточно вычислять её на одном пе-

риоде, например, при $n = 0, \dots, N-1$.

Соотношения (1.33) и (1.34) справедливы и для конечных непериодических последовательностей $x(nT)$ и $h(nT)$ ($n = 0, \dots, N-1$), если рассматривать их как один период соответствующих им периодических последовательностей.

Круговая свертка конечных последовательностей тоже будет конечной.

В матричной форме круговая свертка имеет вид:

$$\mathbf{y}=\mathbf{x}\mathbf{H}^T ; \mathbf{y}=\mathbf{h}\mathbf{X}^T, \\ \text{или } \mathbf{y}^T=\mathbf{H}\mathbf{x}^T ; \mathbf{y}^T=\mathbf{X}\mathbf{h}^T,$$

где \mathbf{y} , \mathbf{h} , \mathbf{x} - N -мерные векторы : $\mathbf{y} = [y(0), y(T), \dots , y((N-1)T)]$;

$\mathbf{x} = [x(0), x(T), \dots , x((N-1)T)]$; $\mathbf{h} = [h(0), h(T), \dots , h((N-1)T)]$,

\mathbf{H} и \mathbf{X} – циклические матрицы $N \times N$, а T - знак транспонирования матриц.

Матрицы \mathbf{H} и \mathbf{X} принимают следующие значения:

$$\mathbf{H} = \begin{bmatrix} h(0) & h(T) & \dots & h((N-1)T) \\ h(T) & h(2T) & \dots & h(0) \\ h(2T) & h(3T) & \dots & h(T) \\ \dots & \dots & \dots & \dots \\ h((N-1)T) & h((N-2)T) & \dots & h(T) \end{bmatrix} ;$$

$$\mathbf{X} = \begin{bmatrix} x(0) & x(T) & \dots & x((N-1)T) \\ x(T) & x(2T) & \dots & x(0) \\ x(2T) & x(3T) & \dots & x(T) \\ \dots & \dots & \dots & \dots \\ x((N-1)T) & x((N-2)T) & \dots & x(T) \end{bmatrix} .$$

Рассмотрим выполнение круговой свертки на следующем примере. Пусть $\mathbf{x} = [2, -2, 1, 0]$, а $\mathbf{h} = [1, 2, 0, 0]$ (рисунок 1.21).

Используя выражение (1.32), вычислим свертку $y(nT) = \sum_{l=0}^3 x(lT)h((n-l)T)$. Для компактности записи букву T (период дискретизации) в дальнейшем опустим:

$$y(0) = x(0)h(0-0) + x(1)h(0-1) + x(2)h(0-2) + x(3)h(0-3) = x(0)h(0) + x(1)h(-1) + x(2)h(-2) + x(3)h(-3) = 2*1 + (-2)*0 + 1*0 + 0*2 = 2;$$

$$y(1) = x(0)h(1) + x(1)h(0) + x(2)h(-1) + x(3)h(-2) = 2*2 + (-2)*1 + 1*0 + 0*0 = 2;$$

$$y(2) = x(0)h(2) + x(1)h(1) + x(2)h(0) + x(3)h(-1) = 2*0 + (-2)*2 + 1*1 + 0*0 = -3;$$

$$y(3) = x(0)h(3) + x(1)h(2) + x(2)h(1) + x(3)h(0) = 2*0 + (-2)*0 + 1*2 + 0*1 = 2.$$

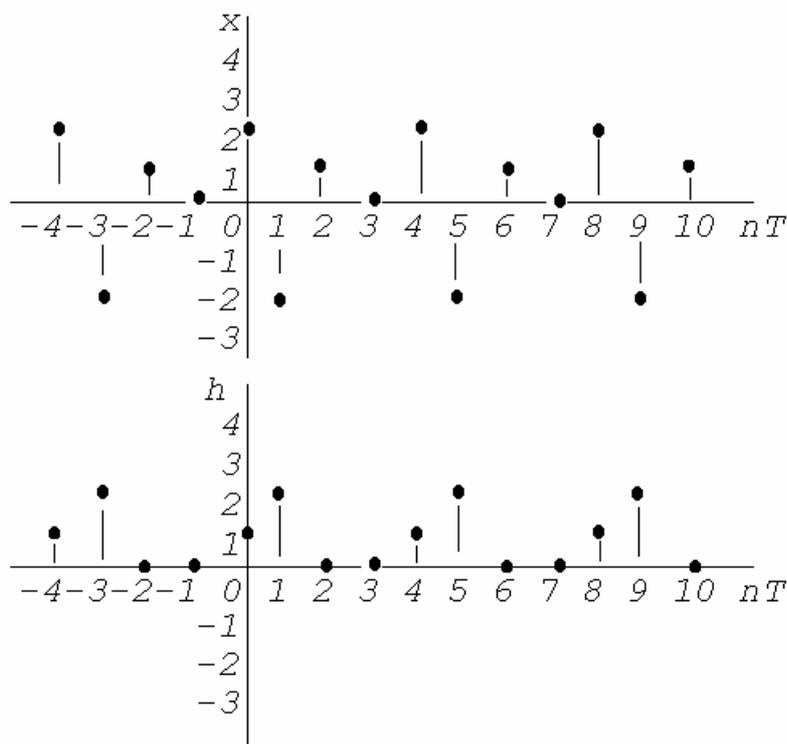


Рисунок 1.21

Таким образом, $\mathbf{y} = [2, 2, -3, 2]$.

В матричной форме

$$\mathbf{y} = [x(0) \ x(1) \ x(2) \ x(3)] \times \begin{bmatrix} h(0) & h(1) & h(2) & h(3) \\ h(-1) & h(0) & h(1) & h(2) \\ h(-2) & h(-1) & h(0) & h(1) \\ h(-3) & h(-2) & h(-1) & h(0) \end{bmatrix} = [y(0) \ y(1) \ y(2) \ y(3)].$$

Ввиду периодичности функции $h(nT)$ (рисунок 1.21)

$$\begin{aligned}
 h(-1) &= h(3) \text{ т.е. } h(N-1); \\
 h(-2) &= h(2) \text{ т.е. } h(N-2); \\
 h(-3) &= h(1) \text{ т.е. } h(N-3).
 \end{aligned}$$

Если изобразить значения $h(nT)$ на окружности в виде, представленном на рисунке 1.22, то видно, что значения $h(-1)$, $h(-2)$, $h(-3)$ могут быть получены движением по кругу против часовой стрелки от значения $h(0)$.

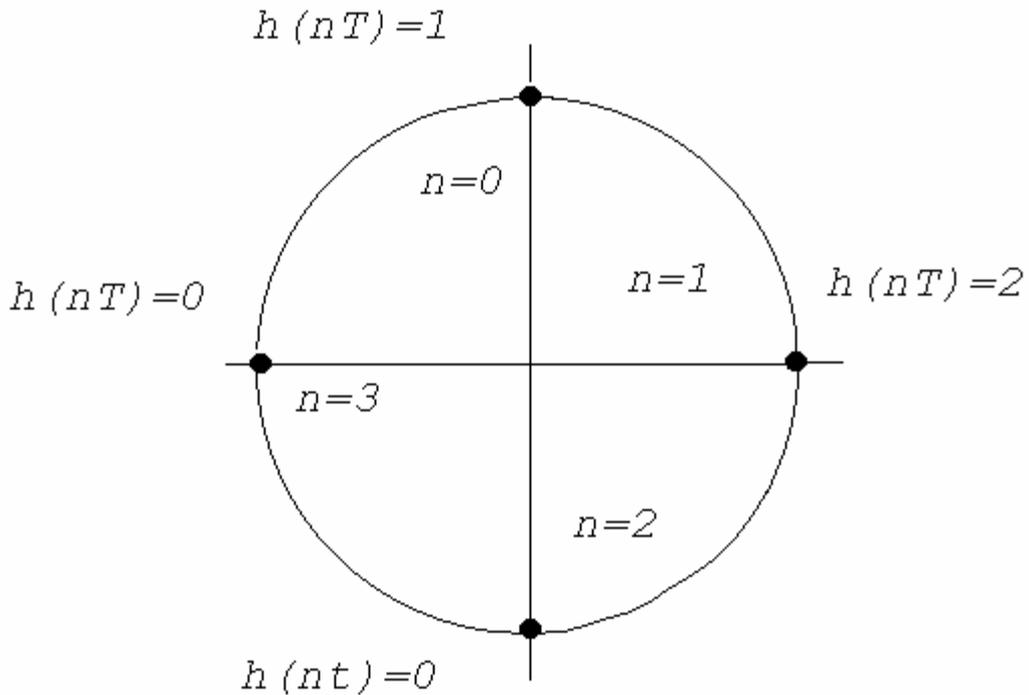


Рисунок 1.22

Поэтому свертка вида (1.31) или (1.32) называется круговой.

Техническая реализация процедуры круговой свертки осуществляется с использованием сдвиговых регистров, умножителей и сумматора (рисунок 1.23). Число разрядов регистров соответствует длине периодической последовательности.

В тех случаях, когда последовательности имеют разный период, для вычисления их свертки используется так называемая линейная свертка.

Линейной (апериодической) сверткой двух конечных последовательностей $x(nT)$ и $h(nT)$ по N_1 и N_2 отсчетов, соответственно, называется последовательность $y(nT)$, определяемая соотношением

$$y(nT) = \sum_{l=0}^n h(lT)x((n-l)T), \quad n=0, \dots, N_1+N_2-2, \quad (1.33)$$

или эквивалентным соотношением

$$y(nT) = \sum_{l=0}^n x(lT)h((n-l)T), \quad n=0, \dots, N_1+N_2-2, \quad (1.34)$$

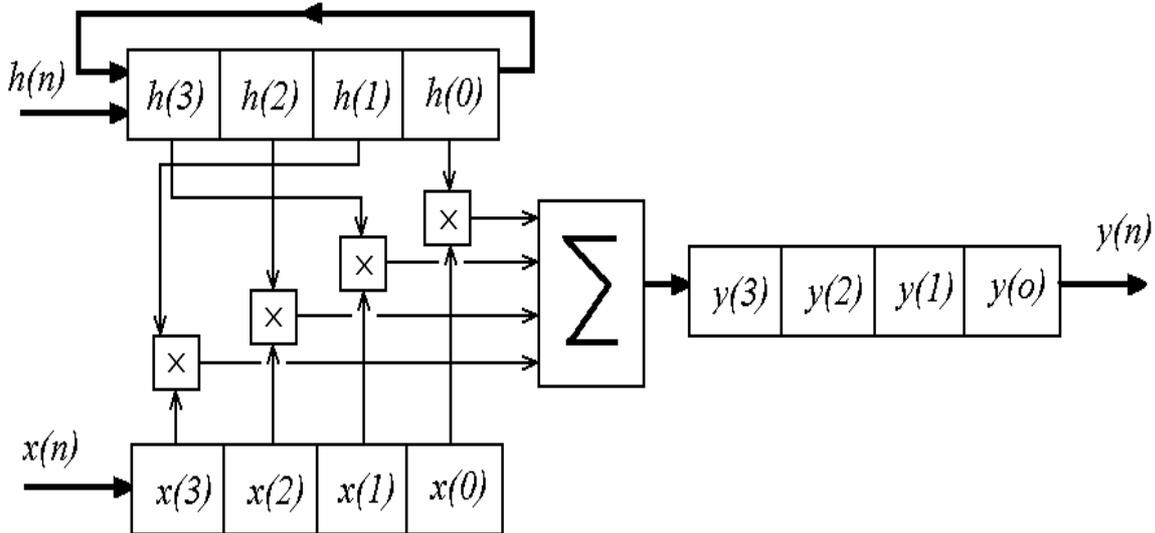


Рисунок 1.23

Сформируем последовательности $x_l(nT)$ и $h_l(nT)$ длиной по $N_1 + N_2 - 1$ отсчетов :

$$x_l(nT) = \begin{cases} x(nT) & \text{при } n = 0, \dots, N_1 - 1 \\ 0 & \text{при } n = N_1, \dots, N_1 + N_2 - 2; \end{cases}$$

$$h_l(nT) = \begin{cases} h(nT) & \text{при } n = 0, \dots, N_2 - 1 \\ 0 & \text{при } n = N_2, \dots, N_1 + N_2 - 2. \end{cases}$$

Тогда линейная свертка последовательностей $x(nT)$ и $h(nT)$ будет равна

$(N_1 + N_2 - 1)$ - точечной круговой свертке последовательностей $x_l(nT)$ и $h_l(nT)$:

$$y(nT) = \sum_{l=0}^{N_1+N_2-2} x(lT)h((n-l)T), \quad n=0, \dots, N_1+N_2-2, \quad .$$

В том случае, когда одна из последовательностей гораздо длиннее другой ($N_1 \gg N_2$ или $N_2 \gg N_1$), используют процедуры, основанные на разбиении длинной последовательности на короткие секции и вычислении частичных свертки, из которых формируется искомая

линейная свертка.

Существует два метода с секционированием свертки: метод перекрытия с суммированием и метод перекрытия с накоплением.

Метод перекрытия с суммированием. Графическая иллюстрация метода приведена на рисунке 1.24 .

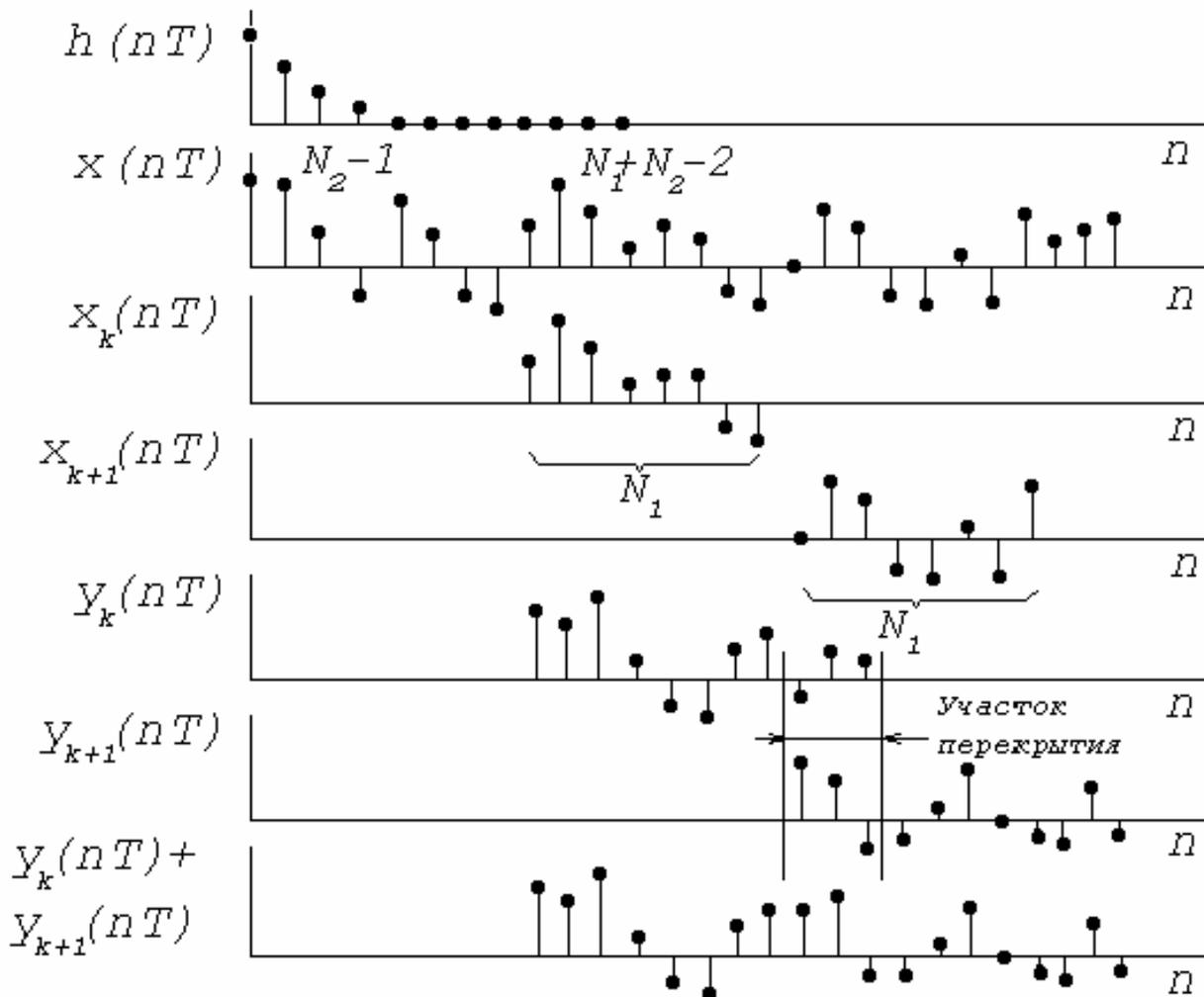


Рисунок 1.24

Пусть более длинной, а в общем случае неограниченной является последовательность $x(nT)$, а $h(nT)$ содержит N_2 отсчетов. Последовательность $x(nT)$ делится на смежные секции $x_k(nT)$ по N_1 отсчетов, так что

$$x(nT) = \sum_{k=0}^{\infty} x_k(nT),$$

$$\text{где } x_k(nT) = \begin{cases} x(nT) & \text{при } kN_1 \leq n \leq (k+1)N_1 - 1; \\ 0 & \text{при других значениях } n. \end{cases}$$

Вычисляем k -ю частичную линейную свертку $y_k(nT)$ последовательностей $x_k(nT)$ и $h(nT)$.

Каждая частичная свертка имеет длину N_1+N_2-1 и перекрывается с $(k+1)$ -частичной сверткой на участке длиной в N_2-1 отсчетов. Поэтому на участке перекрытия их нужно сложить.

Проделав указанные действия для всех k , получаем искомую свертку

$$y(nT) = \sum_{k=0}^{\infty} y_k(nT).$$

Метод перекрытия с накоплением. Графическая иллюстрация метода приведена на рисунке 1.25. В данном случае перекрываются не выходные, а входные секции. Пусть $h(nT)$ содержит N_2 отсчетов.

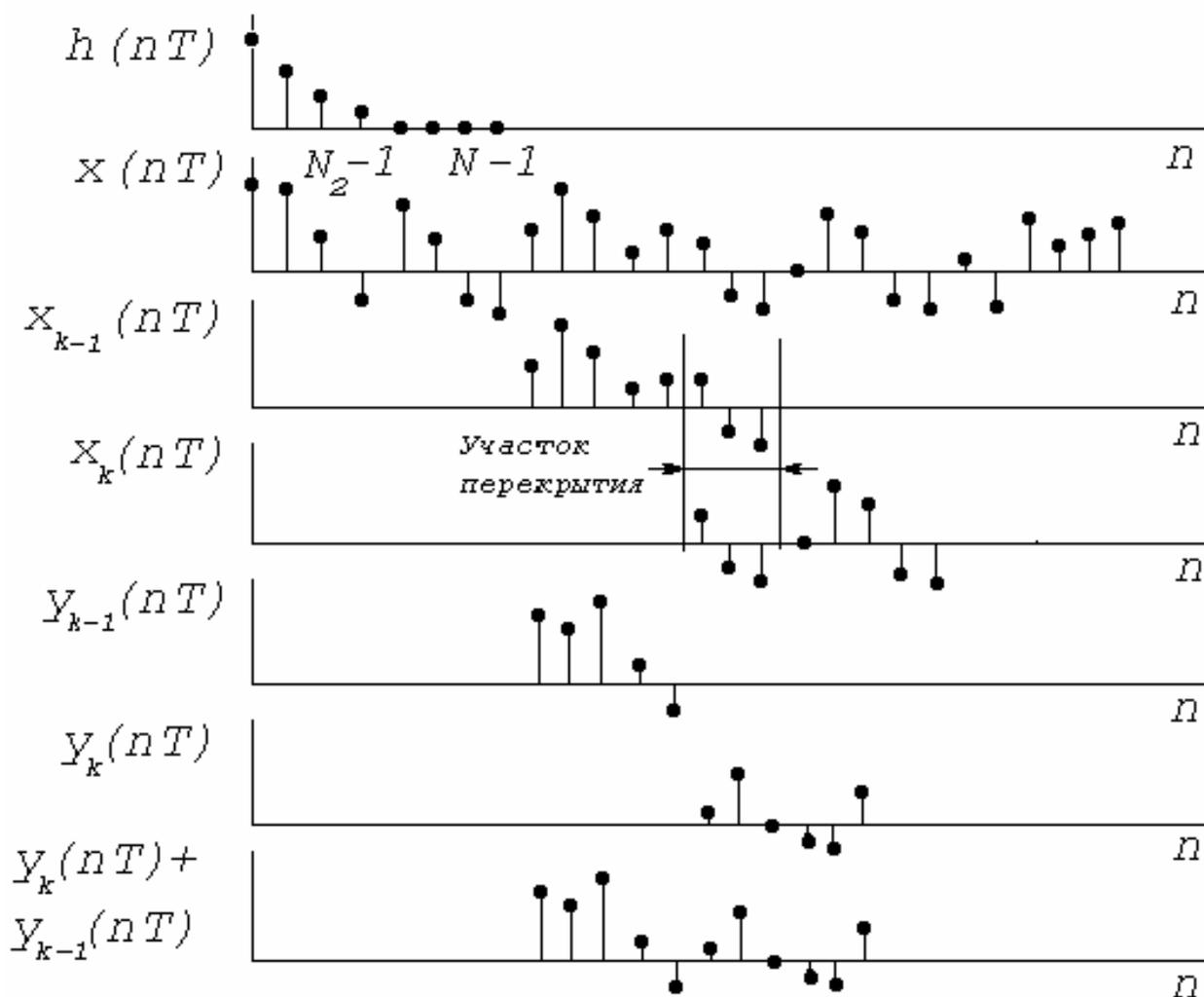


Рисунок 1.25

Длинная последовательность $x(nT)$ делится на секции по $N = N_1+N_2-1$ отсчетов, так что каждые две соседние перекрываются на участке длиной в N_2-1 отсчетов. Последовательность $h(nT)$ дополняется нулями до получения длины в N отсчетов:

$$h_1(nT) = \begin{cases} h(nT) & \text{при } n = 0, \dots, N_2 - 1; \\ 0 & \text{при } n = N_2, \dots, N - 1. \end{cases}$$

Находим k -ю частичную круговую свертку $y_k(nT)$ последовательностей $h_1(nT)$ и $x_k(nT)$. Последние (неверные из-за циклического характера круговой свертки) $N_2 - 1$ отсчетов каждой из последовательностей $y_k(nT)$ отбрасываются, а остальные присоединяются к верным отсчетам $(k-1)$ -й секции. Прделав описанную процедуру для всех k , получим искомую свертку.

2 ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ СИГНАЛОВ

2.1 Представление сигнала с помощью ортогональных функций

Ортогональные преобразования сигналов находят в последние годы широкое применение. Это обусловлено главным образом появлением быстродействующих цифровых вычислительных машин, достижениями в технологии цифровых схем и разработкой специализированных цифровых сигнальных процессоров. Теоретические исследования ортогональных преобразований проводятся в области обработки изображений и речевых сигналов, для их сжатия, в задачах распознавания образов, анализа и проектирования систем связи, фильтрации и т.п.

Существует следующее определение ортогональному сигналу /9/. Множество непрерывных функций действительного переменного $\{u_n(t)\} = \{u_0(t), u_1(t), \dots\}$ называется ортогональным на интервале $(t_0, t_0 + T)$, если

$$\int_T u_m(t) \cdot u_n(t) dt = \begin{cases} C, & \text{если } m = n \\ 0, & \text{если } m \neq n \end{cases} \quad (2.1)$$

Смысл выражения (2.1) поясним на примере скалярного произведения двух векторов $(\vec{A} \cdot \vec{B}) = |\vec{A}| \cdot |\vec{B}| \cdot \cos \psi$, где ψ - угол между векторами, характеризующий наличие или отсутствие проекций векторов друг на друга. При $\psi = \pi/2$ скалярное произведение векторов равно нулю, что характеризует максимальную «непохожесть» векторов друг на друга. Скалярное произведение двух сигналов $(u_m(t), u_n(t)) = \int_T u_m(t) \cdot u_n(t) dt$ равно нулю при их ортогональности.

При $C=1$ множество $\{u_n(t)\}$ называется ортонормированным.

Предположим, что $x(t)$ - действительный сигнал, заданный на интервале $(t_0, t_0 + T)$ и представленный в виде ряда

$$x(t) = \sum_{n=0}^{\infty} a_n u_n(t), \quad (2.2)$$

где a_n означает n -й коэффициент разложения.

Чтобы найти a_n , достаточно обе части (2.2) умножить на $u_m(t)$ и

проинтегрировать в пределах $(t_0, t_0 + T)$:

$$\int_T x(t) \cdot u_m(t) dt = \int_T \sum_{n=0}^{\infty} a_n u_n(t) \cdot u_m(t) dt. \quad (2.3)$$

С учетом (2.1) получаем

$$a_m = \frac{1}{C} \int_T x(t) u_m(t) dt, \quad m = 0, 1, \dots \quad (2.4)$$

Ортогональное множество $\{u_n(t)\}$, удовлетворяющее условию

$$\int_T u_n^2(t) dt < \infty, \text{ называется полным или замкнутым, если справедливо}$$

любое из следующих утверждений /6/:

1) не существует сигнала $x(t)$, удовлетворяющего условию

$$\int_T x_n^2(t) dt < \infty, \text{ такого что } \int_T x(t) u_n(t) dt = 0, n = 0, 1, \dots ;$$

2) для любого кусочно-непрерывного сигнала $x(t)$, удовлетво-

ряющего условию $\int_T x_n^2(t) dt < \infty$ при любом малом $\varepsilon > 0$ существует та-

кое N и конечное разложение

$$\hat{x}(t) = \sum_{n=0}^{N-1} a_n u_n(t),$$

при котором $\int_T |x(t) - \hat{x}(t)|^2 dt < \varepsilon$.

Таким образом, разложение по ортогональным функциям дает возможность представить любой сигнал $x(t)$ в виде бесконечного, но счетного множества чисел $\{a_0, a_1, a_2, \dots\}$. Кроме того, когда $\{u_n(t)\}$ является полным, такое представление возможно в виде конечного множества чисел $\{a_0, a_1, \dots, a_{N-1}\}$.

Бесконечная система попарно ортонормированных функций $\{u_n(t)\}$ представляет некоторый ортонормированный базис. Для любого сигнала всегда существует неединственный базис ортонормированных или ортогональных функций, что позволяет для каждого конкретного случая выбрать наилучший по некоторому критерию вариант разложения сигнала. Методы представления сигналов с помощью ортогональных функций можно разделить на две основные группы:

1) $\{u_n(t)\}$ состоит из гармонических (синусоидальных и (или) косинусоидальных) функций;

2) $\{u_n(t)\}$ состоит из негармонических функций.

К первой группе следует отнести широко распространенное преобразование Фурье, используемое для перевода сигнала из временной области представления в частотную. Ко второй группе относятся такие, как преобразование Уолша-Адамара, Хаара, вейвлет-преобразование и т.д.

Первоначально ортогональные преобразования были теоретически получены для непрерывного представления сигналов. Развитие средств цифровой вычислительной техники обусловило необходимость получения дискретной формы ортогональных преобразований непрерывных (аналоговых) сигналов и разработки алгоритмов преобразования применимых только для дискретных сигналов.

2.2 Дискретное преобразование Фурье

Дискретное преобразование Фурье (ДПФ) используется для разложения в ряд Фурье дискретных (цифровых) последовательностей.

Если $\{x(m)\}$ означает последовательность $x(m)$, $m = 0, 1, \dots, N - 1$ конечных действительных или комплексных чисел, то дискретное преобразование Фурье этой последовательности определяется как

$$X(k) = \sum_{m=0}^{N-1} x(m)W^{km}, \quad k = 0, \dots, N - 1, \quad (2.5)$$

где $W^{km} = e^{-j2\pi km/N}$; $j = \sqrt{-1}$.

Экспоненциальные функции W^{km} являются ортогональными, т.е. удовлетворяющими условию

$$\sum_{m=0}^{N-1} W^{km} \cdot W^{-lm} = \begin{cases} N, & \text{если } (k - l) \text{ равно нулю или целому,} \\ & \text{кратному } N, \\ 0 & \text{в других случаях.} \end{cases} \quad (2.6)$$

Согласно выражению (2.5) имеем

$$\sum_{k=0}^{N-1} X(k)W^{-km} = \sum_{k=0}^{N-1} W^{-km} [x(0) + x(1)W^k + \dots + x(m) \cdot W^{km} + \dots + x(N-1) \cdot W^{k(N-1)}]. \quad (2.7)$$

Используя соотношение (2.6) в уравнении (2.7), получаем обратное дискретное преобразование Фурье (ОДПФ), которое определяется следующим образом:

$$x(m) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W^{-km}, \quad m = 0, 1, \dots, N-1. \quad (2.8)$$

Так как выражения (2.5) и (2.8) составляют пару преобразований, то представление последовательности $\{x(m)\}$ через экспоненциальные функции W^{km} является единственным. Функции W^{km} являются N -периодическими, т.е.

$$W^{km} = W^{(k+N)m} = W^{k(m+N)}, \quad km = 0, \pm 1, \pm 2, \dots \quad (2.9)$$

Последовательности $\{X(k)\}$ и $\{x(m)\}$, определяемые выражениями (2.5) и (2.8), также являются N -периодическими. Иначе говоря, последовательности $\{X(k)\}$ и $\{x(m)\}$ удовлетворяют следующим условиям:

$$x(\pm v) = x(sN \pm m); \quad X(\pm k) = X(sN \pm k), \quad s = 0, \pm 1, \pm 2, \dots \quad (2.10)$$

Пользуясь выражениями (2.9) и (2.10), можно показать, что

$$\sum_{m=p}^q x(m)W^{km} = \sum_{m=0}^{N-1} x(m)W^{km}$$

и

$$\sum_{k=p}^q X(k)W^{-km} = \sum_{k=0}^{N-1} X(k)W^{-km},$$

когда p и q удовлетворяют условию $|p - q| = N - 1$. Пару преобразований, определяемых выражениями (2.5) и (2.8), удобно обозначать как $x(m) \leftrightarrow X(k)$.

Основные свойства ДПФ

Линейность. Дискретное преобразование Фурье является линейным преобразованием, т.е. если $x(m) \leftrightarrow X(k)$ и $z(m) = ax(m) + by(m)$, то

$$Z(k) = aX(k) + bY(k).$$

Свойство комплексной сопряженности. Если $\{x(m)\} = \{x(0) \times x(1) \dots x(N-1)\}$ - такая последовательность действительных чисел, что $N/2$ - целое число и $x(m) \leftrightarrow X(k)$, то

$$X(N/2 + l) = \overline{X(N/2 - l)}, \quad l = 0, 1, \dots, N/2,$$

где $\overline{X(k)}$ является величиной, комплексно-сопряженной.

Свойство сдвига отчетов. Если $x(m) \leftrightarrow X(k)$ и

$$z(m) = x(m + n), \quad n = 0, 1, 2, \dots, N - 1,$$

то

$$Z(k) = W^{-kn} X(k).$$

Свойство реализации круговой свертки. Если $\{x(m)\}$ и $\{y(m)\}$ - последовательности действительных чисел, при которых

$$x(m) \leftrightarrow X(k), \quad y(m) \leftrightarrow Y(k),$$

а свертки этих последовательностей определяются как

$$z(m) = \sum_{n=0}^{N-1} x(n)y(m-n), \quad m = 0, 1, \dots, N-1,$$

то

$$Z(k) = X(k)Y(k).$$

Свойство корреляции. Если $\{x(m)\}$ и $\{y(m)\}$ - последовательности действительных чисел, при которых $x(m) \leftrightarrow X(k)$, $y(m) \leftrightarrow Y(k)$, а их функция корреляции определяется соотношением

$$\hat{z}(m) = \sum_{n=0}^{N-1} x(n)y(m+n), \quad m = 0, 1, \dots, N-1,$$

то

$$\hat{Z}(k) = \bar{Z}(k)Y(k).$$

Сопоставляя Фурье-преобразование аналоговых сигналов и ДПФ, можно отметить следующее. Фурье-преобразование и понятие «спектр» относятся к бесконечной последовательности, а дискретное преобразование Фурье используется для периодических последовательностей с периодом N отсчетов, или для последовательностей конечной длины N .

Поскольку характеристики спектра последовательности, такие как спектральная плотность мощности, амплитуды и фазы отдельных частот, определяются всегда с использованием конечного числа отсчетов этой последовательности, ДПФ является одним из важнейших средств их определения.

На рисунке 2.1 приведен пример определения спектральной плотности амплитуды периодической последовательности $x(nT)$, $N = 16$.

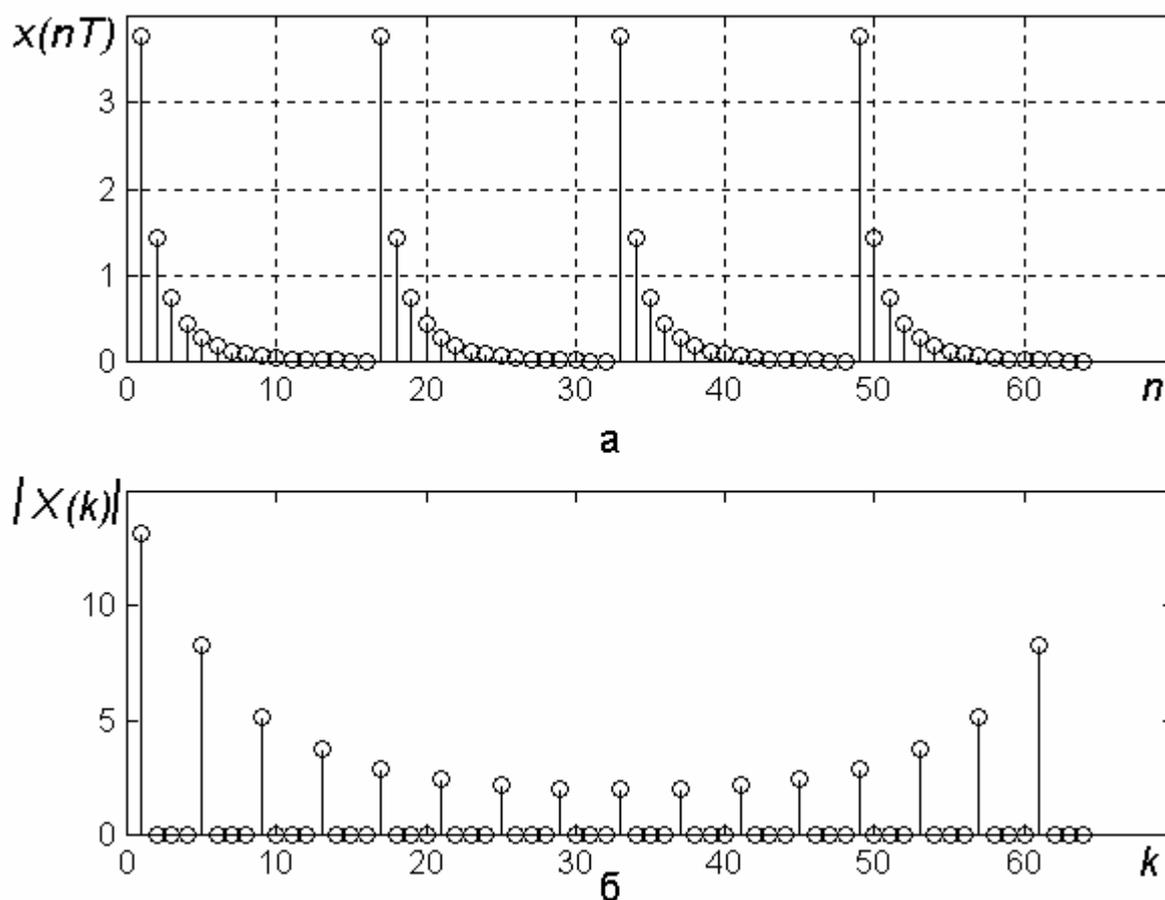


Рисунок 2.1 - Периодическая последовательность (а) и её спектральная плотность амплитуды (б)

Спектральная плотность амплитуды соответствует модулю комплексного спектра последовательности.

2.3 Z-преобразование

Одним из наиболее удобных методов представления последовательностей и работы с ними является Z-преобразование.

Для последовательности $x(n)$, заданной при всех n , оно определяется следующим образом:

$$X(Z) = \sum_{n=-\infty}^{\infty} x(n)Z^{-n}, \quad (2.11)$$

где Z - комплексная переменная.

Комплексная функция (2.11) определена лишь для тех значений Z , при которых степенной ряд сходится.

Z-преобразование последовательности можно рассматривать как способ ее однозначного представления в комплексной Z -плоскости. Из определения (2.11) видно, что Z-преобразование, вычисленное на единичной окружности, т.е. при $Z = e^{j\omega}$ дает

$$X(Z) \Big|_{Z=e^{j\omega}} = X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n},$$

что совпадает с преобразованием Фурье исходной последовательности.

Обычным способом графического изображения информации, содержащейся в Z-преобразовании, является задание особых точек (полюсов) и нулей функции $X(Z)$. Так, например, для систем первого и второго порядка расположение нулей (кружки) и полюсов (крестики) показано на рисунке 2.2.

Следует напомнить, что действительное число x_j называется нулем или корнем дробно-рациональной функции

$$f(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{k=0}^n a_{n-k}x^k}{\sum_{j=0}^m b_{m-j}x^j},$$

если $P(x_j) = 0$, а $Q(x_j) \neq 0$.

Действительное число x_i называется полюсом дробно-рациональной функции $f(x) = \frac{P(x)}{Q(x)}$, если $Q(x_i) = 0$, а $P(x_j) \neq 0$.

Зная расположение нулей и полюсов функции $X(Z)$, ее можно восстановить. Так, если известно, что функция $X(Z)$ имеет N полюсов в точках $Z = p_1, p_2, \dots, p_N$ и M нулей в точках $Z = z_1, z_2, \dots, z_M$, то она может быть записана в виде отношения произведений M

$$X(Z) = A \frac{\prod_{i=1}^M (1 - z_i \cdot Z^{-1})}{\prod_{i=1}^N (1 - p_i \cdot Z^{-1})},$$

где A - произвольная постоянная. Перемножив сомножители, получим, что наиболее общей формой $X(Z)$ является дробно-рациональная функция от Z^{-1} , т.е.

$$X(Z) = \frac{\sum_{i=0}^M a_i \cdot Z^{-i}}{1 + \sum_{i=1}^N b_i \cdot Z^{-i}}.$$

Полученное выражение будет нами использоваться при синтезе цифровых фильтров.

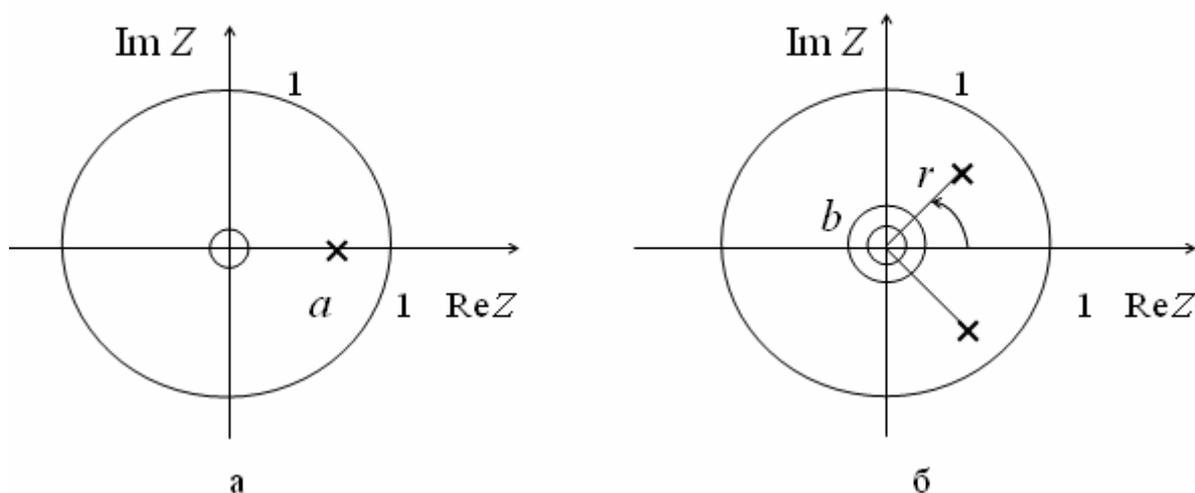


Рисунок 2.2 - Расположение нулей и полюсов для систем первого (а) и второго (б) порядков

Обратный переход от Z - преобразования к исходной последовательности формально определяется соотношением

$$x(n) = \frac{1}{2\pi j} \oint_{C_1} X(Z) \cdot Z^{n-1} \cdot dZ. \quad (2.12)$$

В правой части этого равенства стоит контурный интеграл в Z -плоскости по любому замкнутому контуру в области сходимости, охватывающему начало координат. Например, контуром интегрирования может быть окружность радиуса $C_1 > R_1$, где R_1 - радиус сходимости Z - преобразования (т.е. мы предполагаем, что последовательность $x(n)$ физически реализуема).

Z -преобразование широко используется при исследовании дискретных систем. Основными свойствами Z -преобразования являются линейность, задержка и свертка последовательностей.

Линейность. Z -преобразование линейно. Это означает, что если $X_1(Z)$ и $X_2(Z)$ Z -преобразованиями последовательностей $x_1(n)$ и $x_2(n)$, то при любых действительных a и b Z - преобразование последовательности $ax_1(n) + bx_2(n)$ равно $aX_1(Z) + bX_2(Z)$.

Задержка. Если последовательность $x_1(n)$ имеет Z - преобразование $X_1(Z)$, то Z -преобразование последовательности $x_1(n - n_0)$ при любых n_0 равно $Z^{-n_0} X_1(Z)$. Это свойство Z - преобразования особенно полезно при переходе от представления ЛПП-системы разностным уравнением к представлению ее Z - преобразованием и наоборот.

Например, разностное уравнение

$$y(n) = x(n) - b_1 y(n-1) - b_2 y(n-2) \quad (2.13)$$

можно представить Z - преобразованием

$$Y(Z) = X(Z) - b_1 Z^{-1} Y(Z) - b_2 Z^{-2} Y(Z)$$

или

$$Y(Z) = \frac{X(Z)}{1 + b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2}},$$

где

$$Y(Z) = \sum_{n=-\infty}^{\infty} y(n) Z^{-n}, \quad X(Z) = \sum_{n=-\infty}^{\infty} x(n) Z^{-n}.$$

Свертка последовательностей. Если $x(n)$ и $y(n)$ являются входной и выходной последовательностями дискретной ЛПП-системы с импульсной характеристикой $h(n)$, то

$$Y(Z) = X(Z) \cdot H(Z), \quad (2.14)$$

где $X(Z)$, $H(Z)$, $Y(Z)$ являются соответственно Z -преобразованиями последовательностей $x(n)$, $h(n)$ и $y(n)$. Таким образом, операция свертки последовательностей приводит к перемножению их Z -преобразований. Нетрудно заметить, что $H(Z)$ можно выразить из соотношения (2.14) в виде

$$H(Z) = \frac{Y(Z)}{X(Z)}.$$

Так, на примере уравнения (2.13) ясно, что $H(Z)$ и $h(n)$, может быть получена из разностного уравнения системы и наоборот. Для системы, описываемой уравнением (2.13), $H(Z)$ имеет вид

$$H(Z) = \frac{1}{1 + b_1 Z^{-1} + b_2 Z^{-2}}. \quad (2.15)$$

Выполнив замену вида $Z = e^{j\omega}$, можно перейти от выражения (2.15) к комплексной частотной характеристике

$$H(e^{j\omega}) = \frac{1}{1 + b_1 \cdot e^{-j\omega} + b_2 \cdot e^{-2j\omega}},$$

соответствующей разностному уравнению (2.14).

Пусть необходимо решить разностное уравнение

$$y(n) = x(n) - 3x(n-1) + 0.4y(n-1) - 0.1y(n-2). \quad (2.16)$$

Чтобы найти одностороннее Z -преобразование $Y(z)$, умножим обе части равенства (2.16) на z^{-n} и просуммируем от 0 до ∞ , в результате получим

$$Y(z) = X(z) - 3z^{-1}X(z) + 0.4z^{-1}Y(z) - 0.1z^{-2}Y(z),$$

откуда

$$Y(z) = \frac{X(z)(1 - 3z^{-1})}{1 - 0.4z^{-1} + 0.1z^{-2}}, \text{ так как } H(z) = \frac{Y(z)}{X(z)} \text{ получим}$$

$$H(z) = \frac{1 - 3z^{-1}}{1 - 0.4z^{-1} + 0.1z^{-2}}.$$

Заменяем параметр Z на $e^{j\omega}$:

$$\begin{aligned} H(j\omega) &= \frac{1 - 3e^{-j\omega}}{1 - 0.4e^{-j\omega} + 0.1e^{-j2\omega}} = \\ &= \frac{1 - 3\cos\omega + j3\sin\omega}{1 - 0.4\cos\omega + 0.1\cos 2\omega + j0.4\sin\omega - j0.1\sin 2\omega}. \end{aligned}$$

Отсюда

$$|H(j\omega)| = \frac{\sqrt{1 - 6\cos\omega + 9}}{\sqrt{(1 - 0.4\cos\omega + 0.1\cos 2\omega)^2 + (0.4\sin\omega - 0.1\sin 2\omega)^2}},$$

$$\arg(H(j\omega)) = \operatorname{arctg} \frac{3\sin\omega}{1 - 3\cos\omega} - \operatorname{arctg} \frac{0.4\sin\omega - 0.1\sin 2\omega}{1 - 0.4\cos\omega + 0.1\cos 2\omega}.$$

Таким образом, Z преобразование обеспечивает получение частотных характеристик цифровых устройств и упрощает процедуру их структурного синтеза. Коэффициенты ДПФ совпадают с результатом Z преобразования в N точках, равномерно распределенных по единичной окружности, то есть

$$X(k) = X(z) \Big|_{z = \exp(j2\pi k / N)}, k = 0, \dots, N - 1.$$

Свойства Z преобразования вытекают из свойств ДПФ.

2.4 Общая характеристика негармонических ортогональных преобразований

В последнее время в цифровой обработке сигналов кроме дискретного преобразования Фурье начали широко использовать другие ортогональные преобразования, такие как преобразование Хаара, преобразования Уолша-Адамара-Пэли, слэнт-преобразование (преобразование по пилообразному базису), вейвлет-преобразование и другие [9...12]. Общим для данных преобразований является то, что базисные функции разложения не являются гармоническими функциями. Негармонические ортогональные преобразования находят применение в оптической пространственной фильтрации, сжатии сигналов,

коррекции искажений изображений, кодировании, выделении признаков и т.д. Общим замечательным свойством этих преобразований, определяющим целесообразность их применения, является простота их вычислительной реализации. Для всех этих преобразований, как и для дискретного преобразования Фурье, существуют так называемые быстрые алгоритмы. Так как перечисленные преобразования не связаны с синусоидальным базисом разложения, для них введено такое понятие, как частота /9/.

Понятие частоты применимо к множеству синусоидальных (периодических) функций, точки пересечения нулевого уровня которых равномерно распределены по интервалу. Этот параметр обозначается f и позволяет различать отдельные функции, принадлежащие множествам $\{\cos(2\pi f t)\}$ и $\{\sin(2\pi f t)\}$, и интерпретируется как число полных периодов (или половина числа пересечения нулевого уровня) синусоидальной функции в секунду.

Обобщенная частота любой функции может быть определена как половина среднего числа пересечений нулевого уровня в секунду. При описании обобщенной частоты применяется термин «частота». Он используется для различения функций, точки пересечения нулевого уровня которых распределены неравномерно по интервалу и которые не обязательно являются периодическими. В случае синусоидальных функций понятие частоты совпадает с понятием частоты. Пользуясь приведенным выше определением периодических и непериодических функций, получим:

частота периодической функции равна половине числа пересечений нулевого уровня в секунду;

частота непериодической функции равна пределу половины числа пересечений нулевого уровня в секунду, если этот предел существует.

Для иллюстрации рассмотрим непрерывные функции $f_1(t)$ и $f_2(t)$, приведенные на рисунке 2.3, которые определены на полуоткрытом интервале $[-0,5; 0,5)$. Каждая функция имеет четыре пересечения нулевого уровня на интервале, и, следовательно, частота каждой из них равна двум.

Приведенное выше определение частоты можно с небольшими изменениями применять к соответствующей дискретной функции $f^*(t)$, получаемой из $f(t)$ с помощью равномерной дискретизации.

Если число перемен знака в секунду функции $f^*(t)$ равно η , то частота $f^*(t)$ определяется как $\eta/2$ или $(\eta+1)/2$ при η четном или нечетном соответственно.

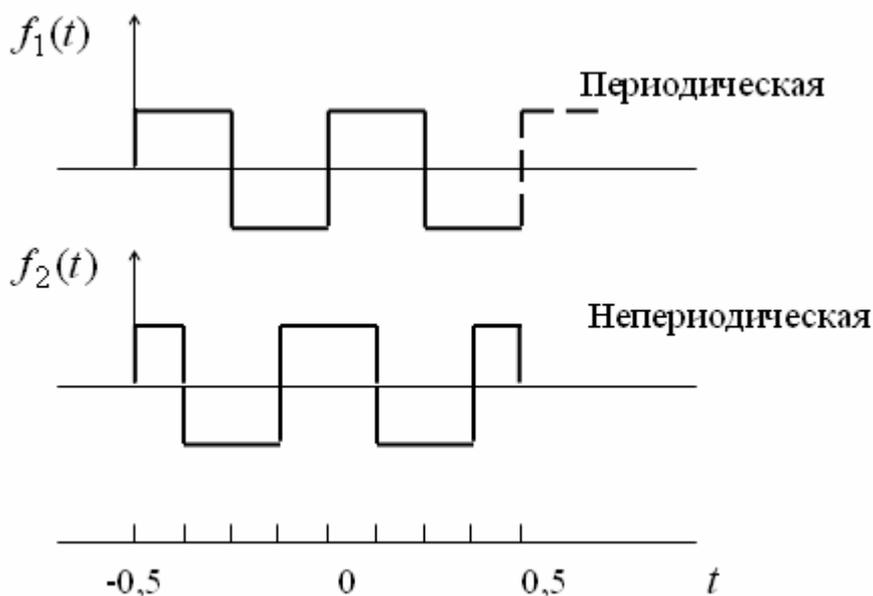


Рисунок 2.3 - Определение частоты непрерывной функции

Рассмотрим дискретные функции $f_1^*(t)$ и $f_2^*(t)$, полученные в результате дискретизации функций, приведенных на рисунке 2.3. При расположении отсчетов в восьми равноотстоящих точках (рисунок 2.4) видно, что $\eta_1 = 3$ и $\eta_2 = 4$. Таким образом, частота каждой из функций $f_1^*(t)$ и $f_2^*(t)$ равна 2, как и в случае $f_1(t)$ и $f_2(t)$.

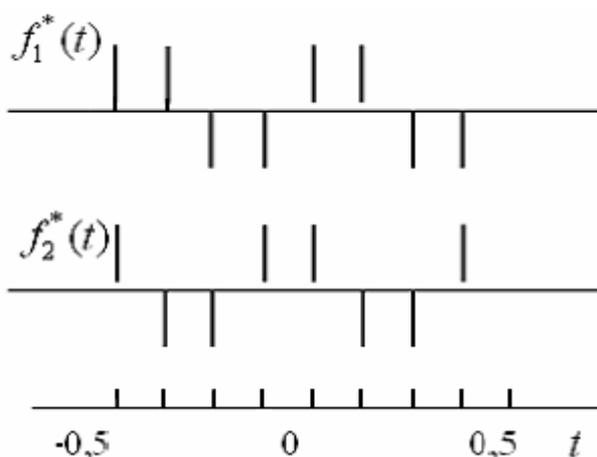


Рисунок 2.4 - Определение частоты дискретной функции

Подобно тому, как частота измеряется числом периодов в секунду (герцах), частота определяется числом пересечений нулевого

уровня в секунду. Для неё используется сокращение «zps» от zero-crossings per second – число пересечений нулевого уровня в секунду.

При разработке негармонических ортогональных преобразований, прежде всего, стремились к уменьшению числа нетривиальных операций. Тривиальными считаются умножение на единицу или ноль. Поэтому наибольшее распространение нашли ортогональные преобразования, у которых число ненулевых коэффициентов разложения при выполнении операции свертки меньше числа отсчетов, а также большинство ненулевых отсчетов равны единице и (или) минус единице. К числу таких преобразований относятся преобразования Хаара и Уолша-Адамара.

2.5 Функции Радемахера и Хаара. Преобразование Хаара

Множество функций Хаара $\{har(n, m, t)\}$, образующих периодическую, ортонормированную и полную систему функций, было предложено им в 1910 г. На рисунках 2.5, 2.6 изображены первые восемь функций Хаара. Рекуррентное соотношение, позволяющее получить $\{har(n, m, t)\}$, имеет вид

$$\begin{aligned}
 & har(0, 0, t) = 1, \quad t \in [0, 1); \\
 & \{har(r, m, t)\} = \begin{cases} 2^{r/2}, & \frac{m-1}{2^r} \leq t < \frac{m-1/2}{2^r}; \\ -2^{r/2}, & \frac{m-1/2}{2^r} \leq t < \frac{m}{2^r}; \\ 0, & \text{при остальных } t \in [0, 1), \end{cases} \quad (2.17)
 \end{aligned}$$

где $0 \leq r < \log_2 N$ и $1 \leq m < 2^r$.

Дискретизация системы функций Хаара, показанных на рисунке 2.5 приводит к матрице, изображенной на рисунке 2.6, каждая строка которой является дискретной функцией Хаара $Har(r, m, t)$. Полученные таким образом матрицы используются для преобразования Хаара и обозначаются $H^*(n)$, где $n = \log_2 N$.

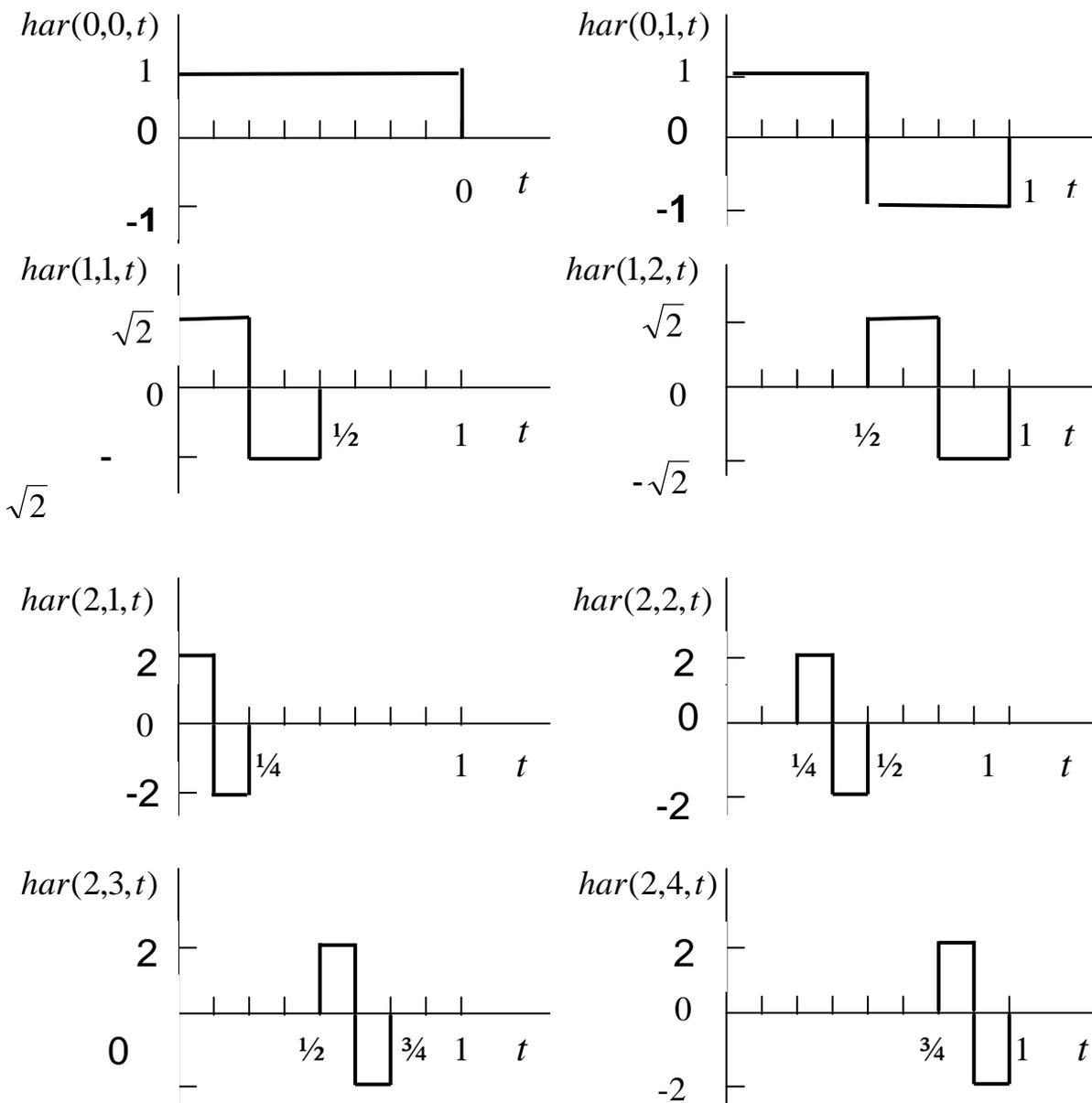


Рисунок 2.5 - Непрерывные функции Хаара

Коэффициенты преобразования Хаара $Y_x(k)$, $k = 0, 1, \dots, N - 1$, соответствующие входной последовательности

$$\{x(m)\} = \{x(0) \ x(1) \ \dots \ x(N - 1)\},$$

получаются в результате преобразования

$$\mathbf{Y}_x(n) = \frac{1}{N} \mathbf{H}^*(n) \mathbf{X}(n),$$

где $\mathbf{H}^*(n)$ - матрица Хаара размером $(N \times N)$.

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Рисунок 2.6 - Дискретные функции Хаара при $N = 8$
Преобразование Хаара

Как видно из рисунка 2.6, многие коэффициенты матрицы Хаара имеют нулевое значение, однако много коэффициентов имеют значения, отличные от нуля и единицы, следовательно, операции, выполняемые с этими коэффициентами, не являются тривиальными.

В 1922 году Радемахером была предпринята попытка на основе гармонических функций получить систему ортогональных импульсных функций с амплитудой ± 1 . Функции Радемахера представляют собой неполную систему ортонормированных функций. Функция Радемахера с индексом m , обозначаемая $rad(m, t)$, имеет вид последовательности прямоугольных импульсов и содержит 2^{m-1} периодов на полуоткрытом интервале $[0, 1)$, принимая значения $+1$ или -1 (рисунок 2.7). Исключение составляет функция $rad(0, t)$, которая имеет вид единичного импульса. Функции Радемахера – периодические с периодом 1, т.е. $rad(m, t) = rad(m, t + 1)$.

Кроме того, они обладают периодичностью и на более коротких интервалах: $rad(m, t + m2^{1-m}) = rad(m, t)$, $m = 1, 2, \dots$; $n = \pm 1, \pm 2, \dots$ Функции Радемахера можно получить с помощью рекуррентного соотношения $rad(m, t) = rad(1, 2^{m-1}t)$,

где

$$rad(1, t) = \begin{cases} 1, & t \in [0, 1/2), \\ -1, & t \in [1/2, 1). \end{cases}$$

Так как функции Радемахера не составляют полный базис функций разложения, то при выполнении обратного преобразования возникают некоторые ошибки.

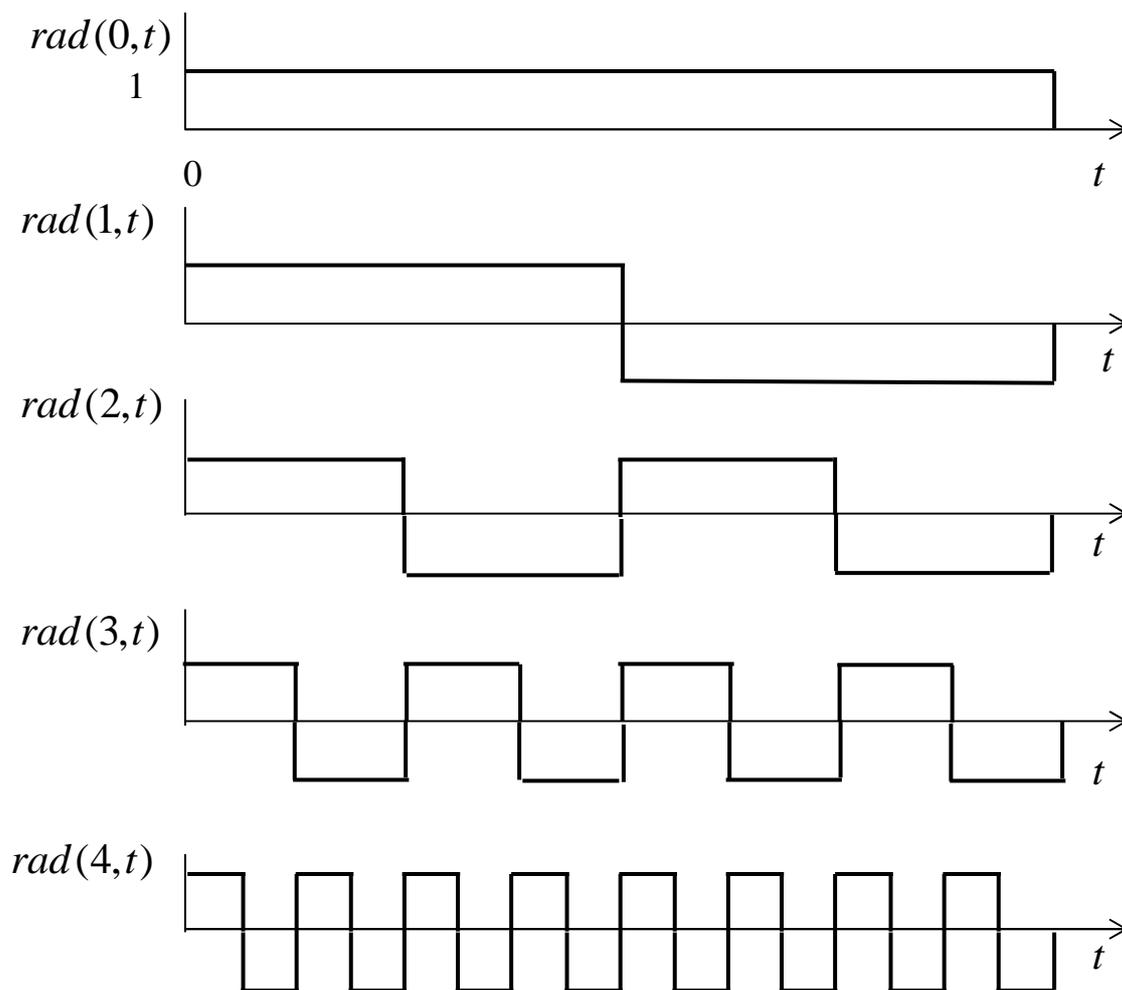


Рисунок 2.7 - Функции Радемахера

2.6 Функции Уолша. Дискретное преобразование Уолша

В 1923 г. Уолш получил полную систему ортонормированных прямоугольных функций, которая дополняет систему функций Радемахера и известна теперь как система функций Уолша. Множество функций Уолша обычно делится на три группы, отличающиеся порядком расположения (видом упорядоченья) отдельных функций в системе. Общеприняты следующие упорядочения:

- упорядочение по частоте (по Уолшу);
- диадическое упорядочение (по Пэли);
- естественное упорядочение (по Адамару).

Ниже каждое из этих упорядочений рассмотрено отдельно.

Упорядочение по частоте или по Уолшу. Это упорядочение было предложено Уолшем. Будем обозначать множество функций Уолша, упорядоченных таким образом, через

$$S_w = \{wal_w(i, t), i = 0, 1, \dots, N - 1\},$$

где $N = 2^n$, $n = 1, 2, 3, \dots$; нижний индекс w обозначает упорядочение по Уолшу, а i соответствует i -му элементу S_w . Если через s_i обозначить частоту $wal_w(i, t)$, то s_i определяется как

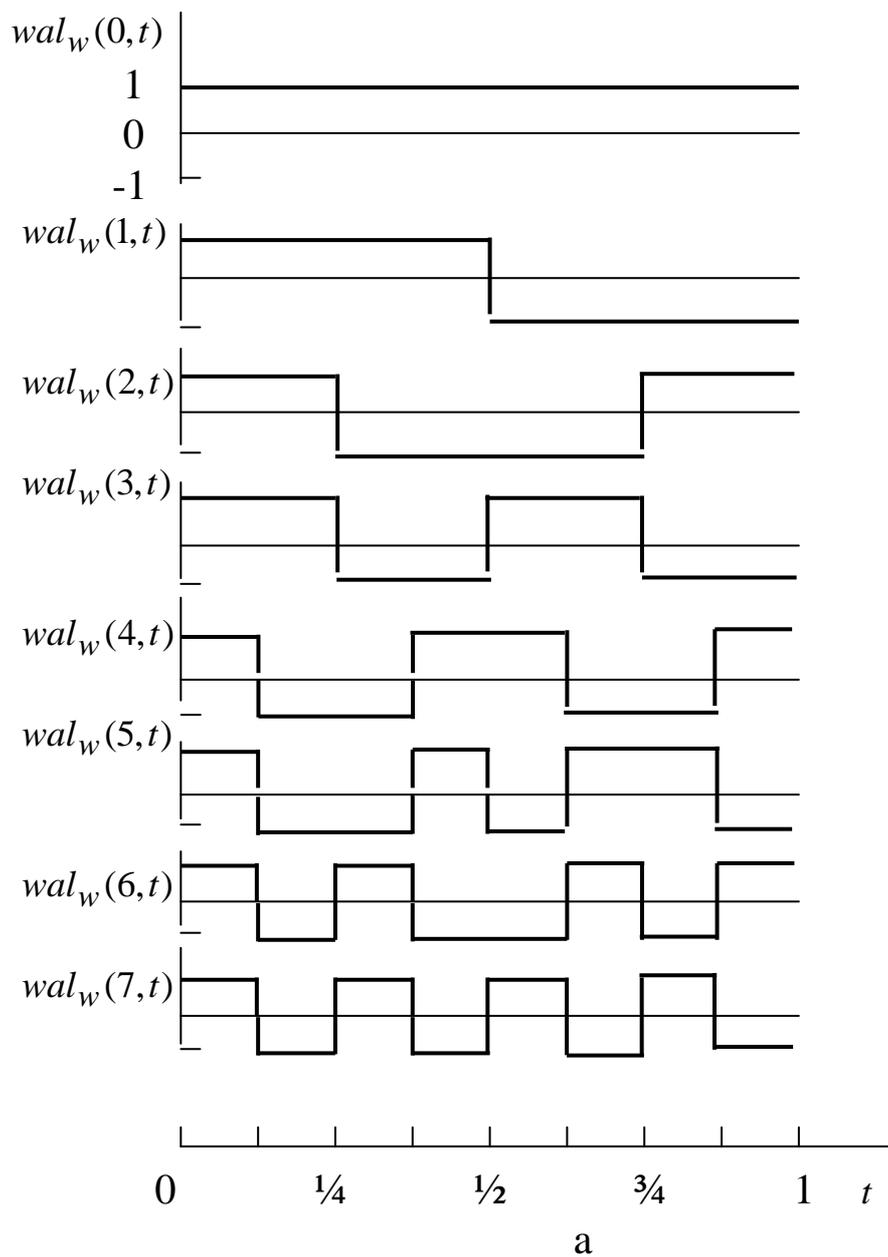
$$s_i = \begin{cases} 0, & i = 0; \\ i/2, & \text{четное}; \\ (i+1)/2, & i - \text{нечетное}. \end{cases}$$

Четные функции Уолша по аналогии с косинусными функциями обозначают cal , а нечетные (синусные) как sal . Функции cal и sal , соответствующие $wal_w(i, t)$, описываются следующим образом:

$$\begin{aligned} cal(s_i, t) &= wal_w(i, t), i - \text{четное}; \\ sal(s_i, t) &= wal_w(i, t), i - \text{нечетное}. \end{aligned}$$

Первые восемь функций Уолша в указанных выше обозначениях приведены на рисунке 2.8,а. Из данного рисунка видно, что частота следующей функции Уолша больше или равняется частоте предыдущей функции Уолша и имеет точно на одно пересечение нулевого уровня больше в открытом интервале $t \in (0, 1)$. Отсюда и следует название упорядочение по частоте. Элемент S_w можно получить из множества функций Радемахера при использовании кода Грея для записи номера функции и соответствующей перестановки функций в последовательности.

Дискретизация функций Уолша, изображенных на рисунке 2.8,а в восьми равноотстоящих точках приводит к матрице (8×8), показанной на рисунке 2.8,б. В общем случае получается матрица ($N \times N$). Такие матрицы будем обозначать $H_w(n)$, $n = \log_2 N$.



$$H_w(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{matrix} \leftarrow wal(0,t) \\ \leftarrow Sal(1,t) \\ \leftarrow Cal(1,t) \\ \leftarrow Sal(2,t) \\ \leftarrow Cal(2,t) \\ \leftarrow Sal(3,t) \\ \leftarrow Cal(3,t) \\ \leftarrow Sal(4,t) \end{matrix}$$

б

Рисунок 2.8 - Функции Уолша, упорядоченные по Уолшу, при $N = 8$: а – непрерывные; б - дискретные

Пусть u_i и v_i цифры i -го разряда в двоичном представлении целых чисел u и v соответственно, т.е.

$$u_{10} = (u_{n-1}u_{n-2}\dots u_1u_0)_2 \quad \text{и} \quad v_{10} = (v_{n-1}v_{n-2}\dots v_1v_0)_2.$$

Тогда элементы h_{uv}^w матрицы $H_w(n)$ имеют вид

$$h_{uv}^w = (-1)^{\sum_{i=0}^{n-1} r_i(u)v_i}; \quad u, v = 0, 1, \dots, N-1,$$

где

$$\begin{aligned} r_0(u) &= u_{n-1}; & r_1(u) &= u_{n-1} + u_{n-2}; \\ r_2(u) &= u_{n-2} + u_{n-3}, \dots, & r_{n-1}(u) &= u_1 + u_0. \end{aligned}$$

Диадическое упорядочение или упорядочение по Пэли. Диадическое упорядочение было введено Пэли. Функции Уолша являются элементами диадической группы и могут быть упорядочены с помощью перенумерации отсчетов с использованием кода Грея. Данное множество функций Уолша обозначается как

$$S_p = \{wal_p(i, t), \quad i = 0, 1, \dots, N-1\},$$

где индекс p обозначает упорядочение по Пэли, а i обозначает i -й элемент S_p . Множество S_p связано с множеством S_w , упорядоченным по Уолшу, соотношением

$$wal_p(i, t) = wal_w[b(i), t], \quad (2.18)$$

где $b(i)$ - переход от кода Грея к двоичному коду с индексом i^1 . Проиллюстрируем соотношение (2.18) на примере $N = 8$. Соответствующие результаты приведены в таблице 2.1

Применяя данные таблицы 2.1 к функциям $wal_w(i, t)$, $i = 0, 1, \dots, 7$, показанным на рисунке 2.8,а, получаем восемь функций Уолша $wal_p(i, t)$, изображенных на рисунке 2.9,а.

Таблица 2.1 - Соотношение между функциями Уолша, упорядоченными по Уолшу и упорядоченными по Пэли

i_{10}	i_2	$b(i)_2$	$b(i)_{10}$	Формула
0	000	000	0	$wal_p(0,t) = wal_w(0,t)$
1	001	001	1	$wal_p(1,t) = wal_w(1,t)$
2	010	011	3	$wal_p(2,t) = wal_w(3,t)$
3	011	010	2	$wal_p(3,t) = wal_w(2,t)$
4	100	111	7	$wal_p(4,t) = wal_w(7,t)$
5	101	110	6	$wal_p(5,t) = wal_w(6,t)$
6	110	100	4	$wal_p(6,t) = wal_w(4,t)$
7	111	101	5	$wal_p(7,t) = wal_w(5,t)$

Проводя дискретизацию функций Уолша (рисунок 2.9,а), получим матрицу (8×8), изображенную на рисунке 2.9,б. Эту матрицу также можно получить переупорядочением строк матрицы, представленной на рисунке 2.8,б. Матрицы, связанные с функциями Уолша, упорядоченными по Пэли, будем обозначать $H_p(n)$, $n = \log_2 N$. Элементы $h_{uv}^{(p)}$ матрицы $H_p(n)$ можно получить используя следующую формулу:

$$h_{uv}^{(p)} = (-1)^{\sum_{i=0}^{n-1} u_{n-1-i} v_i}, \quad u, \quad v = 0,1,\dots,N-1.$$

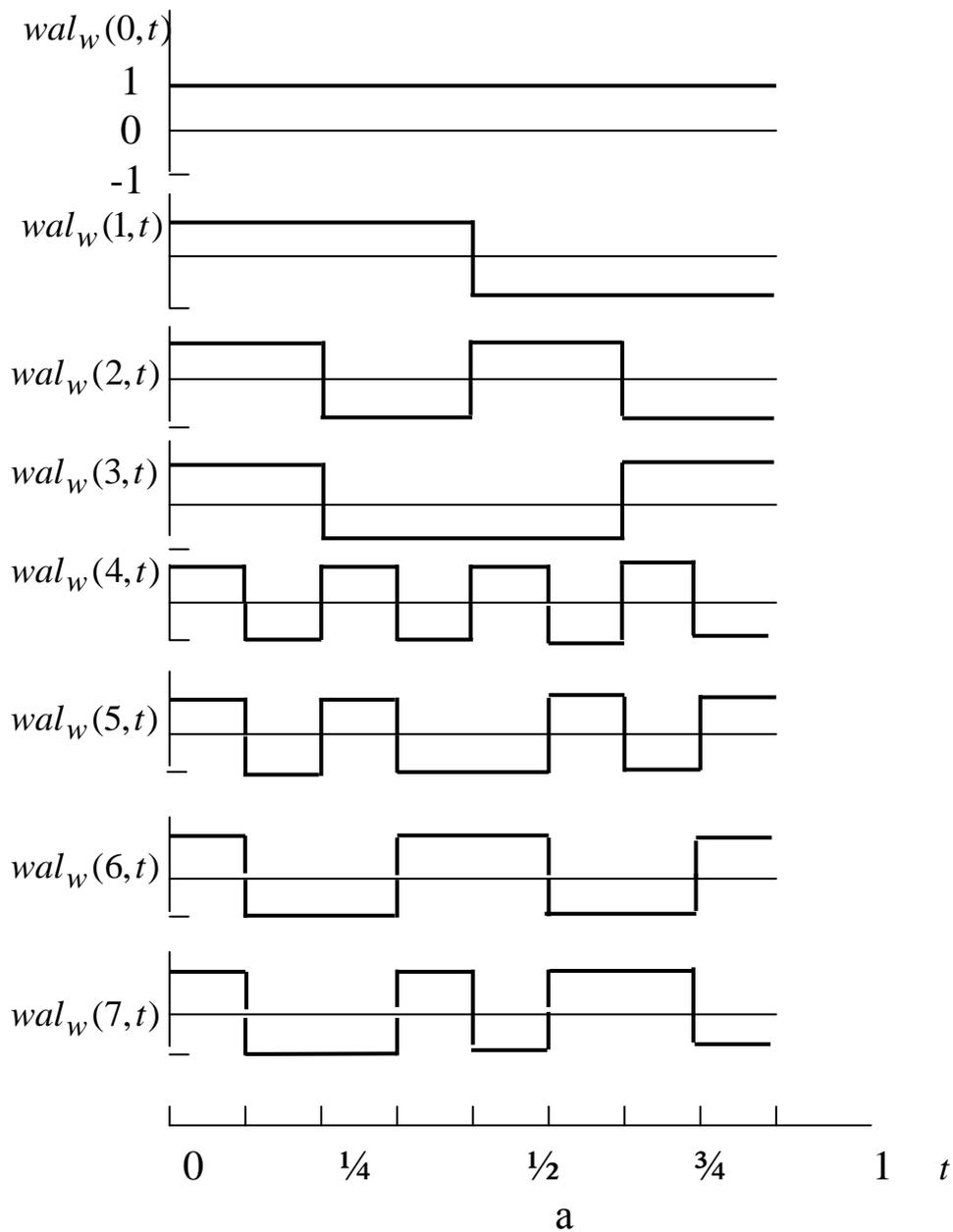
Естественное упорядочение или упорядочение по Адамару. Множество функций Уолша-Адамара обозначается следующим образом:

$$S_h = \{wal_h(i,t), \quad i = 0,1,\dots,N-1\},$$

где индекс h обозначает упорядочение по Адамару, i обозначает i -й элемент S_h . Функции, принадлежащие S_h , связаны с функциями, упорядоченными по Уолшу, соотношением

$$wal_h(i,t) = wal_w[b(<i>)t], \quad (2.19)$$

где $<i>$ - двоично-инвертированная запись i , а $b(<i>)$ -переход от кода Грея к двоичному коду $<i>$.



а

$$H_p(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

б

Рисунок 2.9 - Функции Уолша, упорядоченные по Пэли при $N = 8$: а – непрерывные; б – дискретные

Проиллюстрируем переход, описываемый соотношением (2.19) для $N = 8$; результаты расчета приведены в таблице 2.2.

Таблица 2.2 - Соотношение между функциями Уолша, упорядоченными по Уолшу и упорядоченными по Адамару

i	i_2	$\langle i \rangle_2$	$b(\langle i \rangle)_2$	$b(\langle i \rangle)_{10}$	Формула
0	000	000	000	0	$wal_h(0,t) = wal_w(0,t)$
1	001	100	111	7	$wal_h(1,t) = wal_w(7,t)$
2	010	010	011	3	$wal_h(2,t) = wal_w(3,t)$
3	011	110	100	4	$wal_h(3,t) = wal_w(4,t)$
4	100	001	001	1	$wal_h(4,t) = wal_w(1,t)$
5	101	101	110	6	$wal_h(5,t) = wal_w(6,t)$
6	110	011	010	2	$wal_h(6,t) = wal_w(2,t)$
7	111	111	101	5	$wal_h(7,t) = wal_w(5,t)$

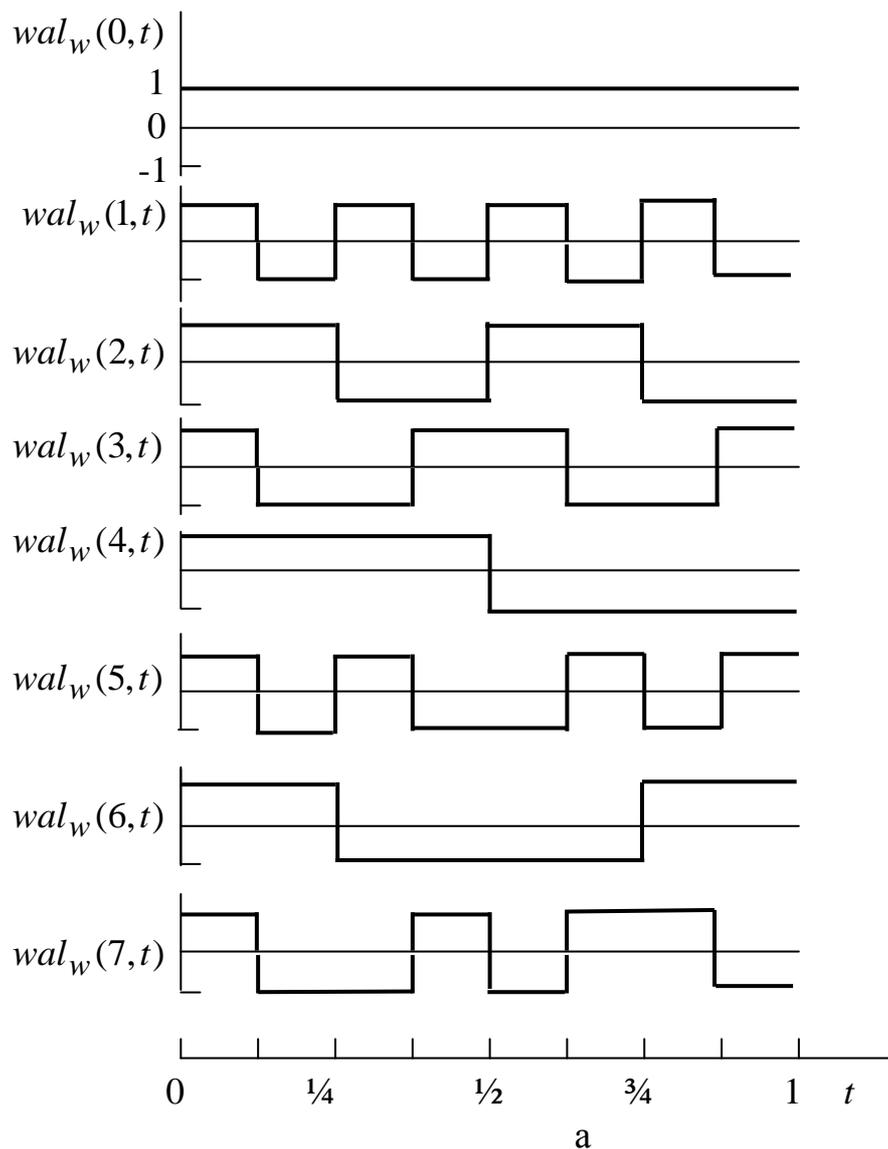
Используя данные таблицы 2.2 и функции $wal_w(i,t)$, $i = 0,1,\dots,7$ (см. рисунок 2.8,а), получаем первые восемь функций Уолша (рисунок 2.10 ,а).

Дискретизация функций Уолша (см. рисунок 2.10,а) приводит к матрице Адамара (8×8), изображенной на рисунке 2.10,б. В общем случае получается матрица $H_h(n)$ размером $(N \times N)$, где $n = \log_2 N$. Для этого класса матриц Адамара справедливо разбиение на подматрицы вида

$$H_h(n) = \begin{bmatrix} H_h(n-1) & H_h(n-1) \\ H_h(n-1) & -H_h(n-1) \end{bmatrix}. \quad (2.20)$$

Такие матрицы соответствуют естественному упорядочению. Элементы $h_{uv}^{(p)}$ матрицы $H_h(n)$ можно получить из следующей формулы:

$$h_{uv}^{(p)} = (-1)^{\sum_{i=0}^{n-1} u_i v_i}, \quad u, v = 0,1,\dots, N-1.$$



$$H_h(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

б

Рисунок 2.10 - Функции Уолша, упорядоченные по Адамару при $N = 8$:
а – непрерывные; б - дискретные

Прежде чем приступить к выводу различных алгоритмов реализации преобразования Уолша-Адамара (ПУА), полезно изучить некоторые аспекты, связанные с представлением непрерывного сигнала $x(t)$ в виде ряда Уолша. При этом предполагается, что $x(t)$ определен на полуоткрытом единичном интервале $t \in [0, 1)$.

Как известно, множество функций Уолша $\{wal_w(i, t)\}$ замкнуто. Это означает, что любой сигнал $x(t)$, который абсолютно интегрируем при $t \in [0, 1)$, можно представить в виде ряда Уолша

$$x(t) = \sum_{k=0}^{\infty} d_k wal_w(k, t). \quad (2.21)$$

Так как множество функций $\{wal_w(k, t)\}$ образует ортонормальную систему в замкнутом интервале $t \in [0, 1]$, то коэффициент d_k определяется как

$$d_k = \int_0^1 x(t) wal_w(k, t) dt, \quad k = 0, 1, 2, \dots \quad (2.22)$$

Напомним, что

$$wal_w(k, t) = cal(s_k, t), \quad k - \text{четное};$$

$$wal_w(k, t) = sal(s_k, t), \quad k - \text{нечетное},$$

где s_k - частота функции $wal_w(k, t)$, определяемая как

$$s_k = \begin{cases} 0, & k = 0; \\ k/2, & k - \text{четное}; \\ (k+1)/2, & k - \text{нечетное}. \end{cases}$$

Если $wal_w(k, t)$ выразить через составляющие sal и cal , то выражение (2.21) принимает вид

$$x(t) = a_0 wal_w(0, t) + \sum_{k=1}^{\infty} [a_k cal(k, t) + b_k sal(k, t)], \quad (2.23)$$

где $a_0 = d_0$; $a_k = d_{2k}$; $b_k = d_{2k-1}$.

Для получения конечного ряда, содержащего $N = 2^n$ слагаемых, приведенный выше ряд обрывается, в результате чего получаем

$$x(t) \approx a_0 wal_w(0, t) + \sum_{k=1}^{N/2-1} [a_k cal(k, t) + b_k sal(k, t)] + b_{N/2} sal(N/2, t). \quad (2.24)$$

Условия сходимости ряда (2.24) были приведены Уолшем, Пэли, Файном /6/:

1. Если $x(t)$ непрерывен при $t \in [0, 1)$, то ряд сходится равномерно к $x(t)$, т.е.

$$\lim_{k \rightarrow \infty} \{a_k, b_k\} = 0. \quad (2.25)$$

Таким образом, существует некоторое $k = N_0$, при котором все a_k и b_k при $k > N_0$ меньше, чем любое наперед заданное $\varepsilon > 0$. Если не учитывать эти коэффициенты при разложении $x(t)$ в ряд Уолша, то теряется незначительная информация.

2. В точках разрыва $x(t)$, лежащих в $t \in [0, 1)$, ряд сходится в среднем квадратическом.

Из приведенного рассмотрения видно, что представление сигналов в виде ряда Уолша аналогично представлению их в виде ряда Фурье. Этого и следовало ожидать из-за сильного сходства между синусоидами и функциями Уолша, что показано на рисунке 2.11 для случая $N = 8$.

Базисные функции ПУА представляют собой дискретные функции Уолша, которые можно выразить с помощью матриц Адамара $H_h(n)$. Эти матрицы можно получить по следующему рекуррентному правилу:

$$H_h(k) = \begin{bmatrix} H_h(k-1) & H_h(k-1) \\ H_h(k-1) & -H_h(k-1) \end{bmatrix}, \quad k = 1, 2, \dots, n, \quad (2.26)$$

где $H_h(0) = 1$ и $n = \log_2 N$. Например, при $k = 1$ и $k = 2$ выражение (2.26) дает

$$H_h(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \quad H_h(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

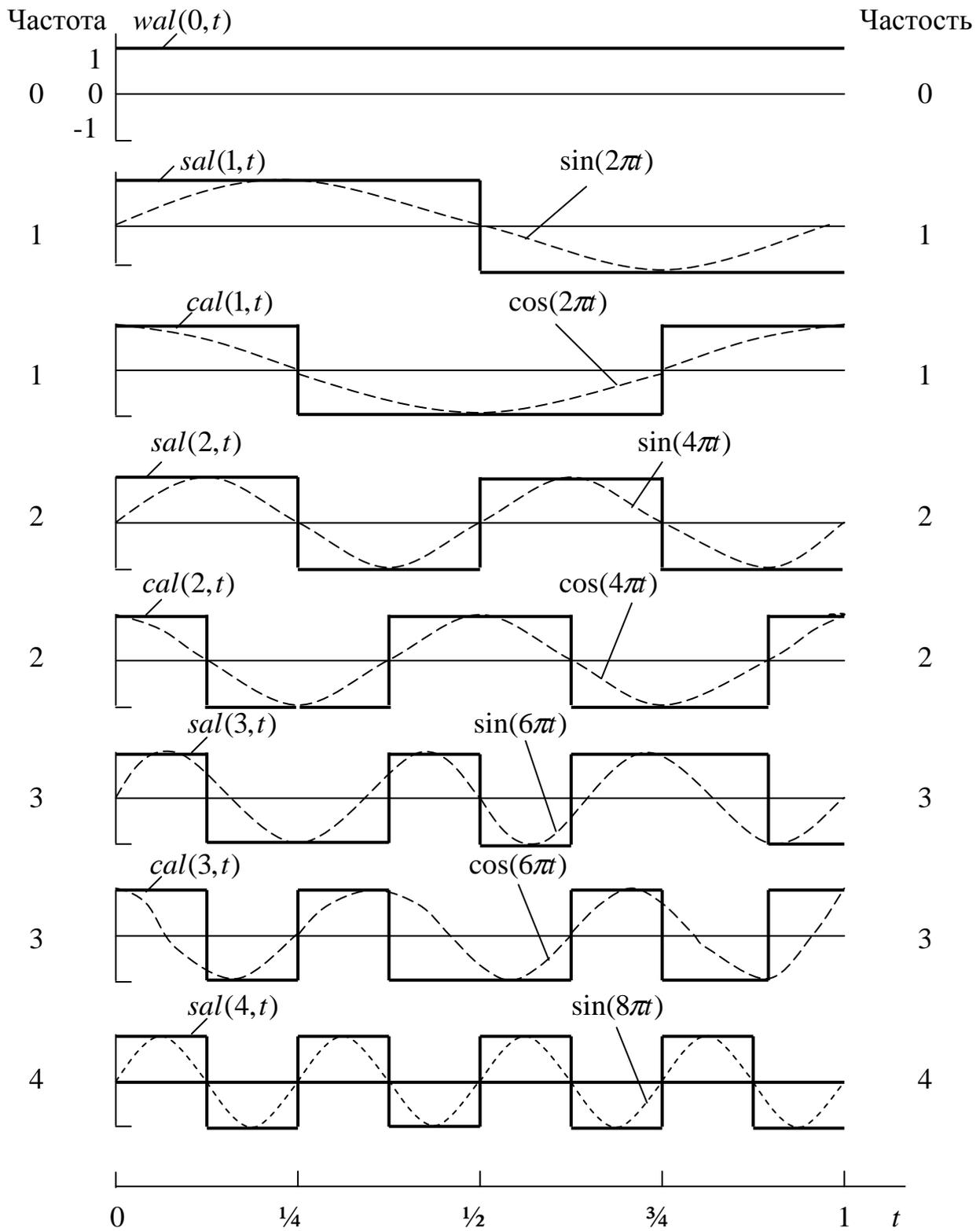


Рисунок 2.11 - Функции Уолша и гармоники Фурье

Можно показать, что матрицы $H_h(k)$ обладают следующими свойствами:

$H_h(k)$ - симметрическая матрица, т.е.

$$H_h(k)' = H_h(k), \quad (2.27)$$

где штрих обозначает операцию транспонирования;

$H_h(k)$ - ортогональная матрица, т.е.

$$H_h(k)' H_h(k) = 2^k I(k), \quad (2.28)$$

где $I(k)$ - единичная матрица размером $(2^k \times 2^k)$;

матрица, обратная $H_h(k)$, пропорциональна матрице

$$[H_h(k)]^{-1} = \frac{1}{2^k} H_h(k),$$

где $[H_h(k)]^{-1}$ - матрица, обратная $H_h(k)$.

Ввиду близости базисных функций разложения ПУА и преобразования Фурье это преобразование также иногда называется двоичным преобразованием Фурье (Binary Furies) BIFORE преобразованием.

Дискретное преобразование Уолша-Адамара $(WHT)_h$ можно записать в матричных или показательных выражениях.

Матричное определение. Пусть $\{X(m)\}$ - последовательность $X(m)$ с периодом N , $m = 0, 1, \dots, N-1$, состоящая из конечных действительных чисел, что записывается как

$$\{X(m)\} = \{X(0)X(1)\dots X(N-1)\}. \quad (2.29)$$

Последовательность $\{X(m)\}$ записывается в виде N - мерного вектора $X(n)$ следующим образом:

$$X'(n) = [X(0)X(1)\dots X(N-1)], \quad (2.30)$$

где $n = \log_2 N$; $X(n)'$ обозначает транспонированный вектор $X(n)$.

Преобразование Уолша-Адамара, упорядоченное по Адамару

$(\text{WHT})_h$, последовательности $\{X(m)\}$ определяется как

$$B_x(n) = \frac{1}{N} H_h(n) X(n), \quad (2.31)$$

где $B_x(k)$ обозначает k -й коэффициент $(\text{WHT})_h$, а $B_x(n)' = [B_x(0)B_x(1)\dots B_x(N-1)]$. Из формул (2.30) и (2.31) следует, что обратное преобразование Уолша-Адамара, упорядоченное по Адамару $(\text{IWHT})_h$, определяется следующим образом:

$$X(n) = H_h(n) B_x(n). \quad (2.32)$$

Так как формулы (2.31) и (2.32) образуют пару преобразований, то представление $\{X(m)\}$ с помощью $(\text{WHT})_h$ однозначно.

2.7 Ортогональное вейвлет-преобразование

Использование цифровых методов обработки изображений связано в значительной степени с вейвлетным (или дискретным волновым) преобразованием. Оно применяется, главным образом, для сжатия и анализа двумерных сигналов и для многих задач подобного рода оказывается более эффективным, чем преобразование Фурье. Основным отличием вейвлетного преобразования является разложение данных не по гармоническим (как для преобразования Фурье), а по другим функциям, называемым вейвлетобразующими [12].

Вейвлетобразующие функции, в противоположность бесконечно осциллирующим синусоидам, локализованы в некоторой ограниченной области своего аргумента, а вдали от нее равны нулю или ничтожно малы. Пример такой функции, называемой "мексиканской шляпой", показан на рисунке 2.12.

Для пояснения алгоритмов вейвлет-преобразования введем понятие транспонированного фильтра. Алгоритм цифровой фильтрации может быть представлен сверткой отчетов импульсной характеристики с отчетами входного сигнала вида

$$y(n) = \sum_{i=0}^{N-1} x(i) \cdot h(i-n) \text{ или } Y(\omega) = H(\omega) X(\omega), \quad (2.33)$$

где $H(\omega), X(\omega)$ соответственно, комплексная частотная

характеристика фильтра и спектр сигнала.

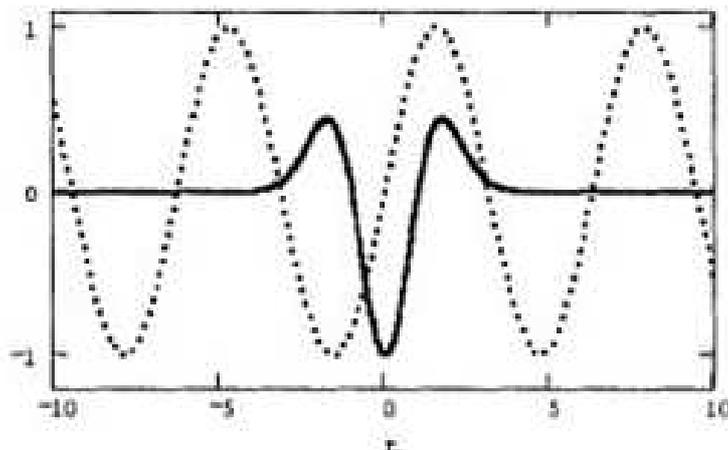


Рисунок 2.12 - Сравнение синусоиды и вейвлетобразующей функции «мексиканская шляпа»

В обозначениях z -преобразования свертка (2.33) примет вид

$$Y(z) = H(z)X(z). \quad (2.34)$$

Транспонированный фильтр имеет импульсную характеристику $h^*(n)$, состоящую из тех же коэффициентов, что и $h(n)$, переставленных в обратном порядке. В Фурье-области транспонированный фильтр имеет вид $\bar{H}(\omega)$.

Как известно, амплитудный спектр сигнала $|X(\omega)|$ характеризует распределение энергии сигнала по частотам $\omega \in [-\pi; \pi)$. При решении задач сжатия сигналов (уменьшения количества отчетов и (или) разрядов, используемых для их кодирования) иногда бывает полезно разложить сигнал на компоненты, энергия которых сосредоточена в различных частотных поддиапазонах (т.е. существенно отлична от нуля на различных подотрезках отрезка $\omega \in [-\pi; \pi)$, и кодировать их с разной степенью детальности (например, в зависимости от чувствительности человеческого уха к звукам различной частоты).

Решение данной задачи может быть выполнено следующим образом. Выберем два фильтра, с импульсной характеристикой $h(n)$ (подавляющий высокие частоты) и $g(n)$ (подавляющий низкие частоты), которые позволяли бы разложить сигнал на две компоненты $X_H(z), X_G(z)$ и вдвое их проредить (половина значений становится

лишней – ведь частотный диапазон сократился вдвое), а затем, с помощью транспонированных фильтров, точно восстановить по этим данным исходный сигнал (эту операцию можно применять рекурсивно). Условия на искомые фильтры удобно записать в терминах z -преобразования.

Пусть $Y(z)$ – z -преобразование одной из компонент. Перед кодированием выходная последовательность прореживается вдвое, а перед восстановлением исходного сигнала доводится до исходной длины вставкой нулей между соседними значениями. При этом z -преобразование из $Y(z)$ превращается в $(Y(z) + Y(-z))/2$. Подставим сюда выражение (2.34) для каждого из фильтров и получим z -преобразования компонент перед восстановлением

$$X_H(z) \rightarrow (H(z)X(z) + H(-z)X(-z))/2,$$

$$X_G(z) \rightarrow (G(z)X(z) + G(-z)X(-z))/2.$$

Z -преобразования транспонированных фильтров имеют вид $H(z^{-1})$ и $G(z^{-1})$. Сигнал восстановится с их помощью точно, если:

$$X(z) = [(H(z^{-1})H(z) + G(z^{-1})G(z))X(z)/2 + (H(z^{-1})H(-z) + G(z^{-1})G(-z))X(-z)/2].$$

Получаем условия точного восстановления :

$$H(z^{-1})H(z) + G(z^{-1})G(z) = 2,$$

$$H(z^{-1})H(-z) + G(z^{-1})G(-z) = 0.$$

Подставив $z = e^{j\omega}$, получим требования к частотным характеристикам искомых фильтров:

$$|H(\omega)|^2 + |G(\omega)|^2 \equiv 2, \tag{2.35}$$

$$\overline{H(\omega)}H(\omega + \pi) + \overline{G(\omega)}G(\omega + \pi) = 0 \tag{2.36}$$

Допустим, что найден фильтр с импульсной характеристикой $h(n)$, удовлетворяющей условию (2.35) тогда, при

$$G(\omega) = -e^{-j\omega} \overline{H(\omega + \pi)} \tag{2.37}$$

условие (2.36) выполняется. Таким образом, задача сводится к нахождению тригонометрического многочлена $H(\omega)$, удовлетворяющего условию (2.35). Фильтры, удовлетворяющие выражениям (2.35, 2.36),

называются квадратурными зеркальными фильтрами.

На рисунках 2.13, (а) и (б), показаны ДПФ такой пары фильтров $h(n)$ и $g(n)$, а также исходный сигнал до и после фильтрации (без прореживания) /12/.

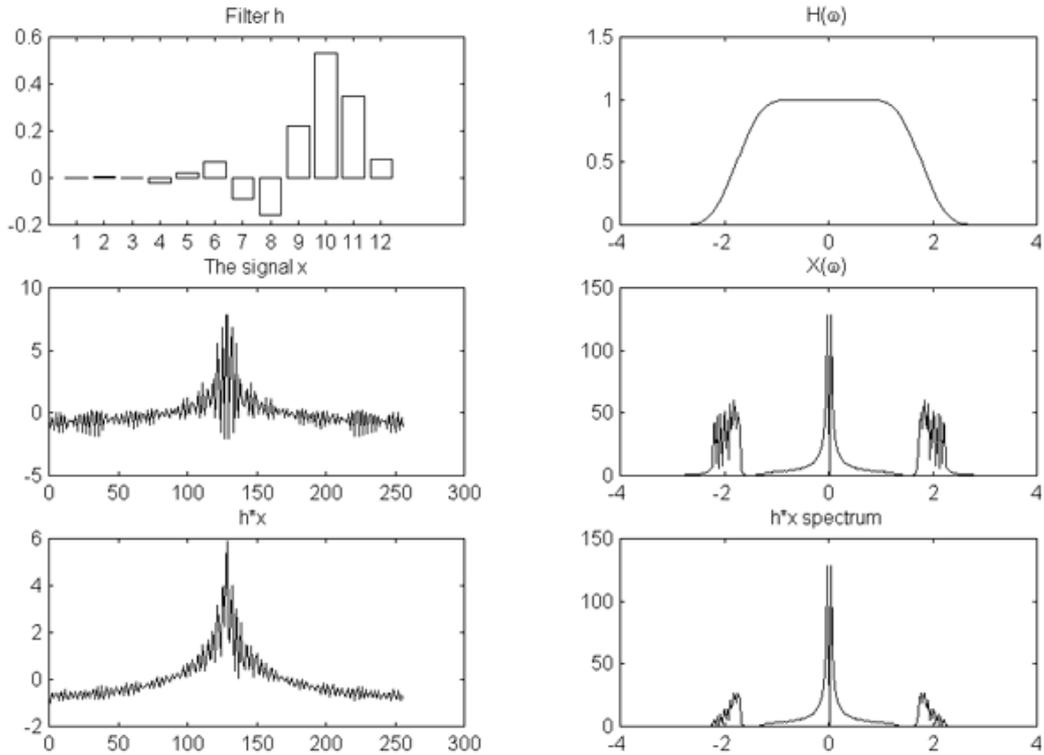


Рисунок 2.13, а

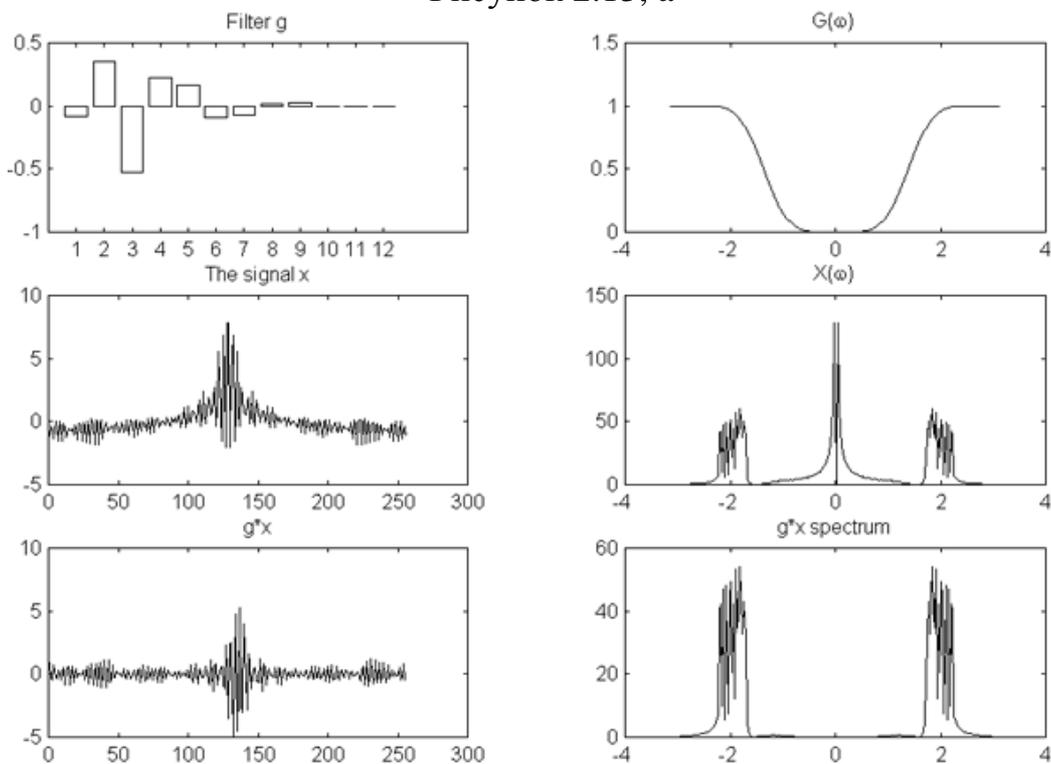


Рисунок 2.13, б

Точно такую же операцию можно применить к одной или обеим из полученных компонент, и т.д., добиваясь нужной локализации по частоте. Это позволяет адаптироваться к особенностям сигнала за счет выбора подходящего “дерева разложения”. Оно может выглядеть, например, так, как показано на рисунке 2.14.

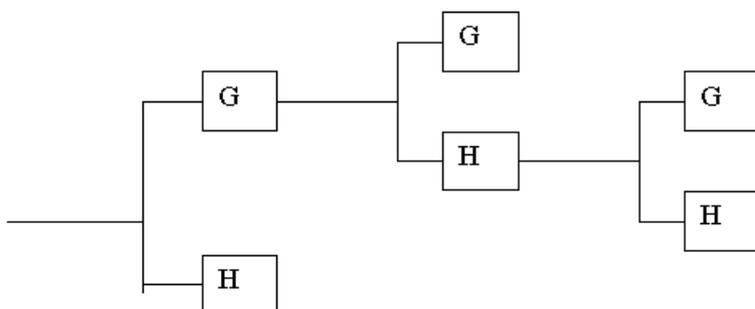


Рисунок 2.14

Другим вариантом разложения сигнала на частотные компоненты является пирамидное представление.

На рисунке 2.15 схематически изображено пирамидное представление одномерного сигнала. Сигналу ставятся в соответствие две пирамиды: *пирамида гауссианов (ПГ)* и *пирамида лапласианов (ПЛ)*. Эти названия отражают аналогию с популярными в графике операциями сглаживания (свертки с колоколообразным фильтром) и выделения перепадов (вычисления “дискретного оператора Лапласа”). Можно считать эту конструкцию упрощенным вариантом предыдущей.

В основании ПГ находится исходный сигнал. Следующий этаж ПГ – исходный сигнал, профильтрованный низкочастотным фильтром h и прореженный после этого вдвое – предполагается, что фильтр h подавляет верхнюю половину частотного диапазона, поэтому частоту выборки можно соответственно уменьшить. К этому этажу применяется та же операция, и так далее. В случае конечных сигналов каждый следующий этаж вдвое короче предыдущего.

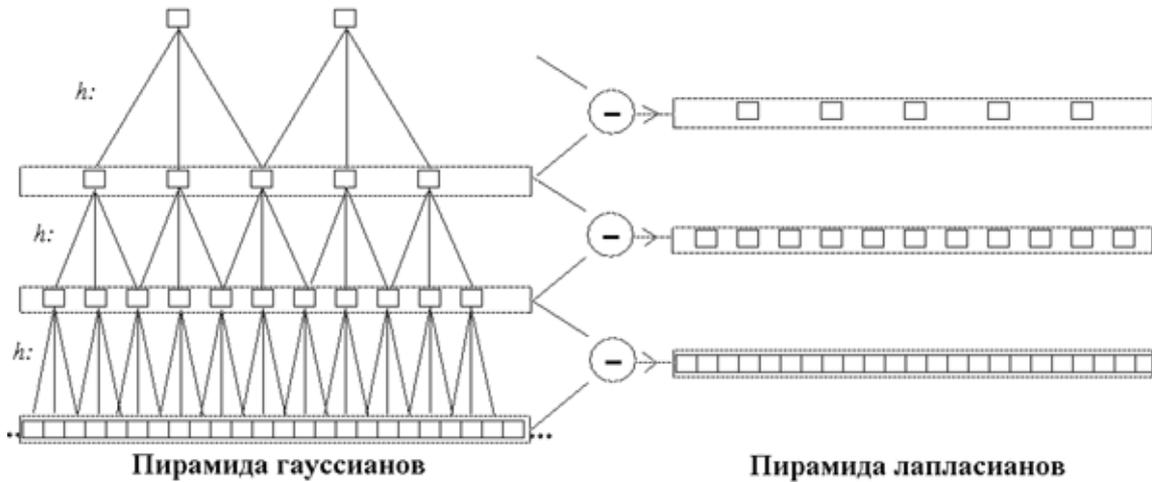


Рисунок 2.15 - Пирамидное представление

Этажи ПЛ – разности между последовательными этажами ПГ. Они вычисляются так. Пусть, например, g_1 и g_2 – первый и второй этажи ПГ, d_1 – первый этаж ПЛ, который мы хотим вычислить. Для этого сначала выравниваются длины этажей:

$$g_2 = \{\dots g_2(k-1)g_2(k)g_2(k-1)\dots\} \mapsto \{\dots g_2(k-1)0g_2(k)0g_2(k-1)\dots\},$$

а затем выполняется фильтрация транспонированным фильтром $h^*(n)$. В результате возникает вектор \tilde{g}_2 . По определению,

$$d_1 = g_1 - \tilde{g}_2.$$

Теперь вместо исходного сигнала (g_1) достаточно запомнить пару (g_2, d_1). Исходный сигнал можно точно восстановить по формуле:

$$g_1 = d_1 + \tilde{g}_2.$$

Сигнал g_2 вдвое короче исходного, а сигнал d_1 , как правило, почти целиком состоит из очень малых величин. Многие из этих величин можно без заметного ущерба для точности восстановления заменить нулями, а остальные закодировать более короткими словами, чем компоненты исходного сигнала. За счет этого общая длина записи (g_2, d_1) будет существенно меньше длины записи исходного сигнала. Это сокращение станет еще больше, если вычислить несколько

этажей ПЛ и запоминать вместо исходного сигнала несколько этажей ПЛ и последний этаж ПГ.

Степень сжатия информации этим методом зависит от выбора фильтра низкочастотного фильтра h . При экспериментах с пирамидными представлениями было сделано наблюдение: “качество” фильтра удобно выражать в терминах эквивалентной весовой функции. Эта функция возникает так. Нетрудно вычислить коэффициенты фильтров, свертка сигнала с которыми дает сразу второй этаж ПГ, третий этаж, и т.д. Оказывается, что при соответствующей нормировке векторы этих коэффициентов сходятся к некоей предельной форме – графику функции $\varphi(t)$, которая должна удовлетворять функциональному уравнению

$$\varphi\left(\frac{t}{2}\right) = \sqrt{2} \sum_k h_k \varphi(t-k) \quad (2.38)$$

Процесс получения изображен на рисунке 2.16.

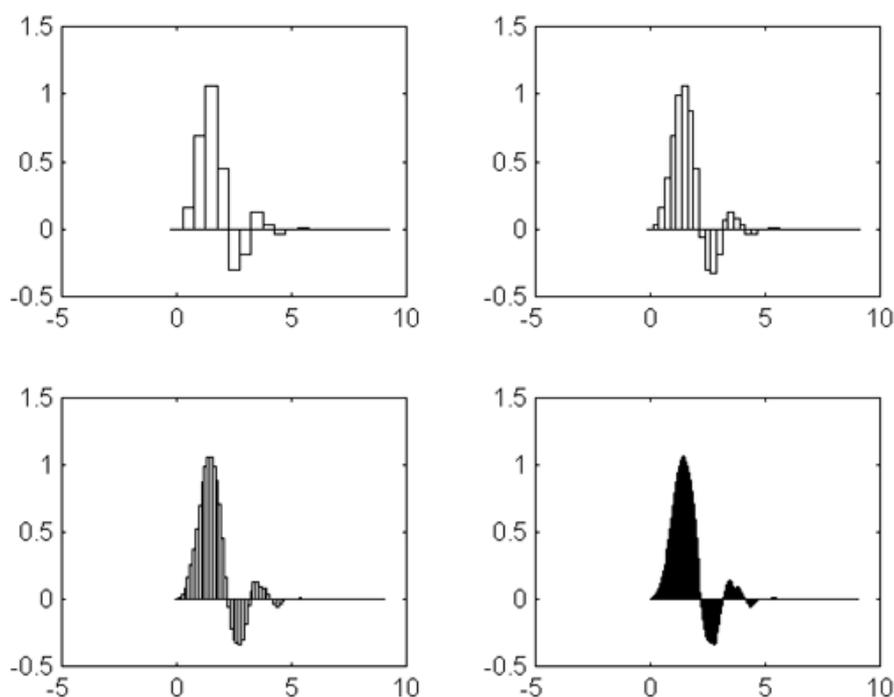


Рисунок 2.16

Концепция многомасштабного анализа (МА), к которому относится вейвлет-преобразование, дает схему представления сигналов, синтезирующую рассмотренные выше методы. Сначала опишем эту схему неформально.

Вклад пирамидного представления: пространство функций (сигналов) исчерпывается системой вложенных подпространств (аналог ПГ). Каждое из них порождено целочисленными сдвигами одной и той же функции $\varphi(t)$, растянутой в 2^n раз. Эта функция является решением уравнения (2.38). Для каждого подпространства n фиксировано и характеризует некоторый масштаб. Задача состоит в том, чтобы разложить сигнал на его “грубую” крупномасштабную версию и набор “деталей” (аналог ПЛ), отличающих версии промежуточных масштабов друг от друга.

Вклад техники разложения по поддиапазонам: коэффициенты импульсной характеристики $h(n)$ должны быть такими, чтобы фильтр $H(\omega)$ удовлетворял условиям (2.35). Оказывается, в этом случае процесс перехода от более тонкой к более грубой версии сигнала сводится к применению фильтра $h^*(n)$, а вычисление “деталей” – к применению фильтра $g^*(n)$.

Вклад самой схемы МА в эту картину таков: оказывается, что при выполнении предыдущих условий пространства “деталей” устроены аналогично пространствам разномасштабных версий. А именно, существует такая функция $\varphi(t)$, порождающая эти пространства своими сдвигами и растяжениями. Эта функция выражается через функцию $\varphi(t)$ по формуле, похожей на (2.38):

$$\psi\left(\frac{t}{2}\right) = \sqrt{2} \sum_k g_k \varphi(t - k). \quad (2.39)$$

Функция $\psi(t)$ называется *ортгональным вейвлетом*. Функция $\varphi(t)$ называется *скейлинг-функцией*.

Рассмотрим один из вариантов формирования ортогонального вейвлета на следующем примере. Пусть $\varphi(t) = 1$ на интервале $[0, 1)$ и $\varphi(t) = 0$ вне этого интервала. Тогда V_0 состоит из функций, постоянных на интервалах вида $[n, n + 1)$, V_1 - из функций, постоянных на ин-

тервалах вида $\left[\frac{n}{2}, \frac{n+1}{2}\right)$, V_{-1} – из функций, постоянных на интервалах вида $[2n, 2n+2)$, и т.д.

Будем считать, что пространство V_0 состоит из сигналов, заданных «с разрешением 1». Тогда пространство V_j – сигналы, заданные с разрешением 2^{-j} . Любое V_j отличается от V_0 только перемасштабированием. Поэтому пространство V_j порождено ортобазисом $\{2^{j/2}\varphi(2^j t - m)\}$. Например, V_{-1} порождено функциями вида $\left\{\frac{1}{\sqrt{2}}\varphi\left(\frac{t}{2} - m\right)\right\}$. Т.к. $V_{-1} \subset V_0$, функция $\frac{1}{\sqrt{2}}\varphi\left(\frac{t}{2}\right)$ обязана линейно выражаться через сдвиги $\varphi(t)$. Значит, существуют такие коэффициенты $\{h_k\}$, что $\varphi\left(\frac{t}{2}\right) = \sqrt{2} \sum_k h_k \varphi(t - k)$. Это в точности уравнение (2.38).

Скейлинг-функция – это эквивалентная весовая функция из пирамидного представления.

Можно предположить, что изначально заданный сигнал $x(t)$ известен с разрешением 1, другими словами, $x(t) \in V_0$, и нам даны коэффициенты x_k его разложения по сдвигам скейлинг-функции:

$$x(t) = \sum_n x_n \varphi(t - n).$$

Естественно считать версией масштаба 2 ортогональную проекцию $x(t)$ на подпространство V_{-1} . Она задается набором скалярных произведений $x(t)$ с функциями из ортобазиса V_{-1} , то есть величинами $c_r = \left\langle x(t), \frac{1}{\sqrt{2}}\varphi\left(\frac{t}{2} - r\right) \right\rangle$. Из уравнения (2.38) и условий ортогональности имеем:

$$c_r = \sum_s h_s x_{2r+s}. \quad (2.40)$$

Другими словами, проекция осуществляется путем свертки с фильтром h^* и прореживания вдвое. Заметим, что прореживание вдвое «встроено» в эту формулу (через индекс $2r + s$). Разумеется, это следствие выбора базиса в V_{-1} .

В качестве деталей сигнала $x(t)$, исчезающих при переходе к

масштабу 2, следует взять компоненту $x(t)$, ортогональную к сигналам масштаба 2, т.е. к пространству V_{-1} . Имеет место разложение $V_0 = V_{-1} \oplus W_{-1}$, где для любых функций $a(t) \in V_{-1}, b(t) \in W_{-1}$ выполнено $\langle a(t), b(t) \rangle = 0$. Замечательно, что ортобазисом W_{-1} будет набор функций $\left\{ \frac{1}{\sqrt{2}} \psi \left(\frac{t}{2} - m \right) \right\}$, где $\psi(t)$ задается формулой (2.40). Коэффициенты g_k имеют вид $g_k = (-1)^k h_{1-k}$, что эквивалентно формуле (2.36). Искомая проекция задается набором скалярных произведений $x(t)$ с функциями из ортобазиса W_{-1} , то есть величинами $d_r = \left\langle x(t), \frac{1}{\sqrt{2}} \psi \left(\frac{t}{2} - r \right) \right\rangle$. Совершенно аналогично формуле (2.40) получаем

$$d_r = \sum_b g_b x_{2r+s}, \quad (2.41)$$

что равносильно свертке с фильтром g^* и прореживанию вдвое. Та же схема действует на любом масштабе. При любом j $V_j = V_{j-1} \oplus W_{j-1}$, ортобазисом W_{j-1} будет $\left\{ 2^{(j-1)/2} \psi(2^{j-1}t - m) \right\}$ и разложение сигнала из V_j на сглаженную часть и детали (т.е. его проекции на V_{j-1} и W_{j-1}) находятся по формулам (2.40) и (2.41). Совокупность же функций $\left\{ 2^{j/2} \psi(2^j t - m) \right\}$, где j и m пробегают все целые значения, будет базисом всего пространства сигналов $L^2(R)$.

Тем самым, ортогональность базисных функций ОМА приводит к тому, что вычисление сглаженных версий сигнала и его деталей выполняется сверткой с парой квадратурных зеркальных фильтров.

В данном случае

$$h_0 = h_1 = \frac{1}{\sqrt{2}}, g_0 = \frac{1}{\sqrt{2}}, g_1 = \frac{-1}{\sqrt{2}}.$$

Другими словами, с точностью до нормировки коэффициенты проекции на V_{-1} являются суммами значений кусочно-постоянной функции на соседних единичных интервалах, а коэффициенты проекции на пространство деталей W_{-1} – разностями этих значений. Функ-

ция $\psi(t)$ равна 1 на интервале $\left[0, \frac{1}{2}\right)$, -1 на интервале $\left[\frac{1}{2}, 1\right)$, и нулю во всех остальных точках. Множество функций $\left\{2^{j/2}\psi\left(2^j t - m\right)\right\}$ образует ортонормированный базис всего пространства сигналов $L^2(R)$. Это классический базис Хаара (см. пункт 2.5).

Итак, ортогональный многомасштабный анализ на основе вейвлет-преобразования позволяет построить аналог ПГ и ПЛ, пользуясь квадратурными зеркальными фильтрами. Перепишем теперь формулы (2.40) и (2.41) в матричном виде. Введем матрицы H и G :

$$H = \begin{bmatrix} \dots & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & \dots & \dots \\ \dots & \dots \end{bmatrix}$$

$$G = \begin{bmatrix} \dots & g_{-2} & g_{-1} & g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & g_{-2} & g_{-1} & g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & g_{-2} & g_{-1} & g_0 & g_1 & g_2 & \dots & \dots \\ \dots & \dots \end{bmatrix}.$$

Тогда условия точного восстановления (2.35) дают условие на матрицы H и G :

$$H^* H + G^* G = 1. \tag{2.42}$$

Обозначив вектор исходных коэффициентов через x , можно записать его разложение в сумму огрубленной версии и серии векторов деталей так как показано на рисунке 2.17.

$$\left\{Gx, GHx, GH^2x, \dots, GH^{N-1}x, GH^N x\right\}$$

Рисунок 2.17

Эту процедуру иногда называют быстрым вейвлет-преобразованием (Fast Wavelet Transform), а иногда – алгоритмом Малла. Число итераций N может быть произвольным. Если вектор x

конечен, его надо продолжить «на бесконечность», проще всего это сделать периодическим образом. Каждое применение операторов H и G сокращает длину вектора вдвое, поэтому общее число операций линейно по длине входа.

Результатом преобразования является набор векторов $\{Gx, GHx, GH^2x, \dots, GH^{N-1}x, H^Nx\}$. Обратное преобразование делается по схеме, представленной на рисунке 2.18.

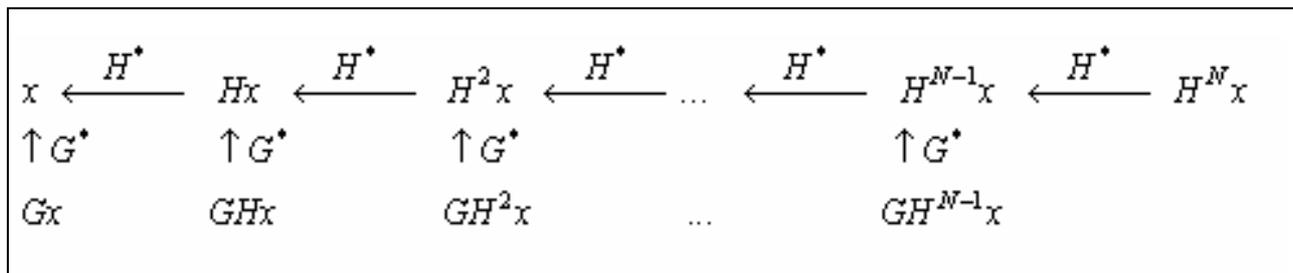


Рисунок 2.18

На первом шаге восстанавливается $H^{N-1}x$ по формуле $H^{N-1}x = H^*(H^Nx) + G^*(GH^{N-2}x)$, которая верна в силу (2.42). Затем вычисляется $H^{N-2}x = H^*(H^{N-1}x) + G^*(GH^{N-2}x)$, и т.д.

Пример применения ортогонального вейвлет-преобразования для сжатия и других вариантов обработки изображений приведены в главе 4.

3 БЫСТРЫЕ АЛГОРИТМЫ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЙ

3.1 Постановка задачи и обоснование обобщенного алгоритма быстрого преобразования Фурье (БПФ)

Рассмотренные в предыдущей главе алгоритмы дискретных ортогональных преобразований основаны на свертке двух последовательностей и в матричной форме имеют вид

$$\mathbf{y} = \mathbf{H}^T \mathbf{x}.$$

Количество арифметических операций для таких выражений существенно зависит от длины последовательностей. И при значительных N требуется высокое быстродействие вычислителя для реализации преобразования в реальном масштабе времени. Поэтому задача уменьшения числа операций всегда была и остается актуальной, даже при современном состоянии вычислительных средств.

Набор алгоритмов, называемых алгоритмами быстрого преобразования Фурье (БПФ), включает разнообразные методы уменьшения времени вычисления дискретного преобразования Фурье (ДПФ). Поскольку ДПФ является основной операцией в большинстве задач спектрального анализа, выполняемых в реальном и близком к реальному масштабах времени, очень важно уменьшать время вычисления ДПФ. Таким образом, алгоритмы БПФ, позволяющие ускорить вычисления ДПФ в 100 и более раз по сравнению с методом прямого вычисления ДПФ, имеют чрезвычайно важное значение.

Как было определено в предыдущей теме, ДПФ конечной последовательности $\{x(n)\}$, $0 \leq n \leq N - 1$ вычисляется по следующему выражению:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi nk/N)}, \quad k = 0, 1, \dots, N-1 \quad (3.1)$$

или в более удобной форме

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W^{nk}.$$

Функция W^{nk} является периодической последовательностью с периодом N , т.е.

$$W^{(n+mN)(k+lN)} = W^{nk}, \quad m, l = 0, \pm 1, \dots$$

Периодичность W^{nk} является одним из ключевых моментов алгоритмов БПФ.

Из соотношения (3.1) следует, что в случае, когда последовательность $x(n)$ является комплексной, при прямом вычислении N -точечного ДПФ нужно выполнить $(N-1)^2$ комплексных умножений и $N \cdot (N-1)^2$ комплексных сложений. Для достаточно больших N (порядка 1000 и более) прямое вычисление ДПФ требует выполнения чрезмерного количества вычислительных операций и не может быть реализовано в реальном масштабе времени даже быстродействующими вычислителями.

Основная идея БПФ состоит в том, чтобы разбить исходную N -точечную последовательность на две более короткие последовательности, ДПФ которых могут быть скомбинированы таким образом, чтобы получилось ДПФ исходной N -точечной последовательности. Так, например, если N четное, а исходная N -точечная последовательность разбита на две $(N/2)$ -точечные последовательности, то для вычисления искомого N -точечного ДПФ потребуется порядка $(N/2)^2 \cdot 2 = N^2/2$ комплексных умножений, т.е. вдвое меньше по сравнению с прямым вычислением. Здесь множитель $(N/2)^2$ дает число умножений, необходимое для прямого вычисления $(N/2)$ -точечного ДПФ, а множитель 2 соответствует двум ДПФ, которые должны быть вычислены. Эту операцию можно повторить, вычисляя вместо $(N/2)$ -точечного ДПФ два $(N/4)$ -точечных ДПФ (предполагая, что $N/2$ четное) и сокращая тем самым объем вычислений еще в два раза. Выигрыш в два раза является приближенным, поскольку не учитывается, каким образом из ДПФ меньшего размера образуется искомое N -точечное ДПФ.

Проиллюстрируем описанную методику для N -точечной последовательности $\{x(n)\}$, считая, что $N = 2^V$. Введем две $(N/2)$ -точечные последовательности $\{x_0(n)\}$ и $\{x_1(n)\}$ из четных и нечетных членов $x(n)$ соответственно, т.е.

$$x_0(n) = x(2n), \quad n = 0, 1, \dots, \frac{N}{2} - 1,$$

$$x_1(n) = x(2n + 1), \quad n = 0, 1, \dots, \frac{N}{2} - 1.$$

N -точечное ДПФ последовательности $\{x(n)\}$ можно записать как

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} + \sum_{n=0}^{N-1} x(n)W_N^{nk} = \\
 &\quad n - \text{четное} \qquad \qquad n - \text{нечетное} \\
 &= \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k}. \quad (3.2)
 \end{aligned}$$

С учетом того, что

$$W_N^2 = \left[e^{j(2\pi/N)} \right]^2 = e^{j(2 \cdot 2\pi/N)} = e^{j[2\pi/(N/2)]} = W_{N/2},$$

перепишем выражение (3.2) в виде

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N/2-1} x_0(n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x_1(n)W_{N/2}^{nk}, \quad (3.3) \\
 X(k) &= X_0(k) + W_N^k X_1(k),
 \end{aligned}$$

где $X_0(k)$ и $X_1(k)$ равны $(N/2)$ -точечным ДПФ последовательностей $x_0(n)$ и $x_1(n)$.

Из формулы (3.3) следует, что N -точечное ДПФ $X(k)$ может быть разложено на два $(N/2)$ -точечных ДПФ, результаты которых объединяются согласно выражения (3.2). Если бы $(N/2)$ -точечные ДПФ вычислялись обычным способом, то для вычисления N -точечного ДПФ потребовалось бы, очевидно, $(N^2/2 + N)$ комплексных умножений. При больших N (когда $N^2/2 \gg N$) это позволяет сократить время вычисления на 50%.

Поскольку $X(k)$ определено при $0 \leq k \leq N-1$, а $X_0(k)$ и $X_1(k)$ определены при $0 \leq k \leq N/2-1$, необходимо доопределить формулу (3.3) для $k \geq N/2$. Это определение может быть записано следующим образом:

$$X(k) = \begin{cases} X_0(k) + W_N^k X_1(k), & 0 \leq k \leq N/2-1, \\ X_0(k - N/2) + W_N^k X_1(k - N/2), & N/2 \leq k \leq N-1. \end{cases} \quad (3.4)$$

Соотношение (3.4) является прямым следствием периодичности ДПФ. Заметим также, что

$$W_N^{k+N/2} = -W_N^k, \text{ т.к.}$$

$$\begin{aligned} e^{-j\frac{2\pi(k+N/2)}{N}} &= e^{-j\frac{2\pi k}{N}} \cdot e^{-j\frac{2\pi N/2}{N}} = e^{-j\frac{2\pi k}{N}} \cdot e^{-j\pi} = \\ &= e^{-j\frac{2\pi k}{N}} \cdot (\cos \pi - j \sin \pi) = W_N^k \cdot (-1 + j0) = -W_N^k. \end{aligned}$$

С учетом этого формулу (3.4) можно переписать в следующем виде

$$\begin{aligned} X(k) &= X_0(k) + W_N^k X_1(k) \\ X(k + N/2) &= X_0(k) - W_N^k X_1(k), \end{aligned} \quad (3.5)$$

где $0 \leq k \leq N/2 - 1$.

Формула (3.5) называется базовой операцией БПФ и имеет графическое обозначение, приведенное на рисунке 3.1.

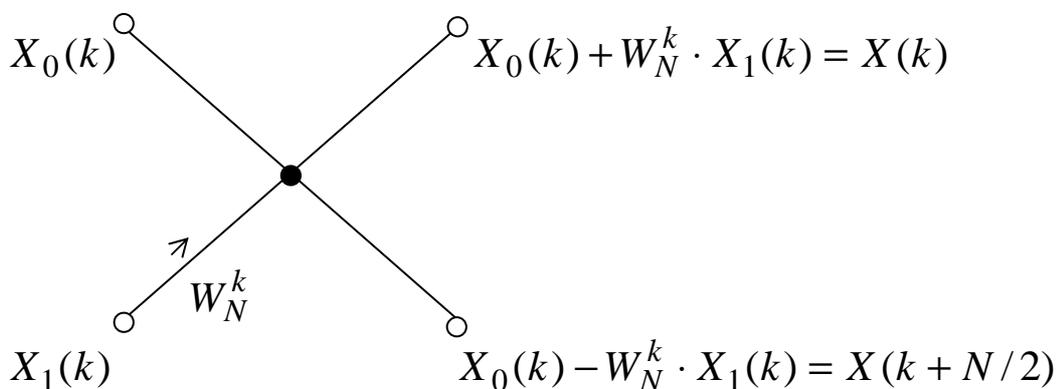


Рисунок 3.1- Базовая операция алгоритма БПФ

Процесс уменьшения размеров ДПФ продолжается до тех пор, пока не останутся только двухточечные ДПФ. Всего может быть выполнено $\nu = \log_2 N$ разбиений. Величина ν называется количеством этапов БПФ, а каждое разбиение, соответственно, этапом БПФ.

На рисунке 3.2. показано вычисление восьмиточечного ДПФ ($N = 8, \nu = 3$).

Значение поворачивающего коэффициента на каждом этапе БПФ определяется номером отчета k и числом точек ДПФ. На заключительном (третьем) этапе $k = \{0,1,2,3\}$, $N = 8$,
 на втором этапе $k = \{0,1\}$, $N = 4$,
 на первом этапе $k = 0$, $N = 2$.

Соответственно $W_{III} = \{W_8^0, W_8^1, W_8^2, W_8^3\}$; $W_{II} = \{W_4^0, W_4^1\}$, а так как $W_8^0 = W_4^0$ и $W_{N/2}^k = W_N^{2k}$, то $W_{II} = \{W_8^0, W_8^2\}$; $W_I = \{W_2^0\} = \{W_8^0\}$.

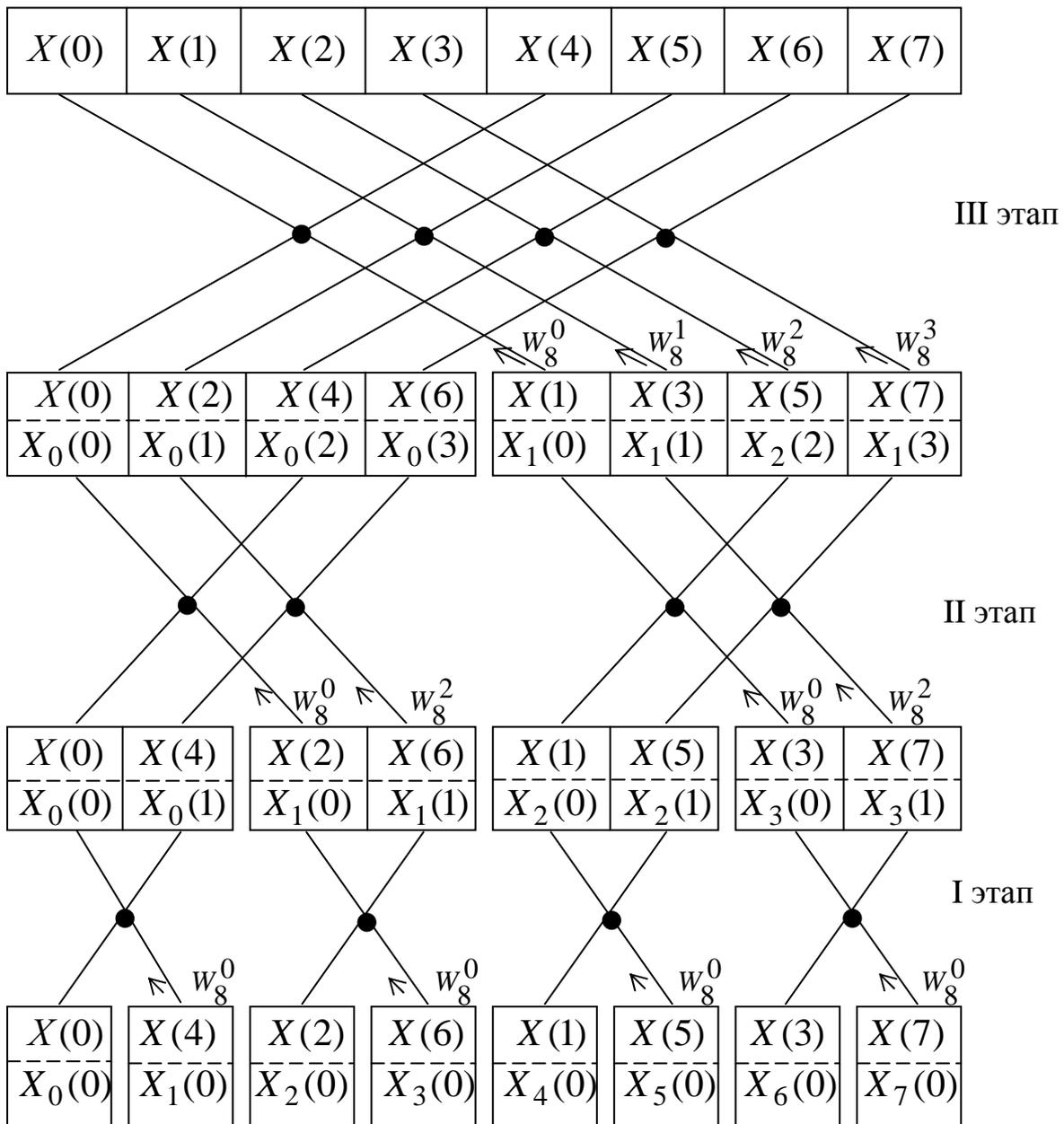


Рисунок 3.2

В соответствии с рисунком 3.2. на первом этапе входные отсчеты образуют пары не с соседними элементами, а с другими, т.е. для получения выходных отсчетов, расположенных в естественном порядке, входные отсчеты должны быть переставлены местами.

Существует правило изменения номеров отсчетов, называемое правилом двоичной инверсии.

Суть этого правила (таблица 3.1) заключается в зеркальном отражении единиц и нулей двоичного кода номера отсчета. Например, для цифры два двоичный код равен 010. Относительно центральной единицы правый разряд и левый равны, поэтому код не инвертируется, а вот для цифры три (код 011) единица младшего разряда должна меняться местами с нулем из старшего разряда.

Таблица 3.1

Номер	Двоичное представление	Двоичная инверсия	Двоично-инверсный номер
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Двоичная инверсия номера отсчета может выполняться и для четного числа разрядов. Например, $a = 1011 \Rightarrow a_{об.инв} = 1101$.

Рассмотренный выше алгоритм БПФ называется алгоритмом с прореживанием по времени, так как происходит перестановка отсчетов входного сигнала. Существует алгоритм и с прореживанием по частоте, т.е. алгоритм, у которого происходит перестановка выходных отсчетов (последовательности отсчетов в частотной области). В этом варианте алгоритма БПФ входная последовательность $\{x(n)\}$ разбивается на две последовательности, содержащие по $N/2$ отсчетов, каждая следующим образом: первая последовательность $\{x_0(n)\}$ состо-

ит из первых $(N/2)$ отсчетов $\{x(n)\}$, а вторая $\{x_1(n)\}$ - из остальных $(N/2)$ отсчетов $\{x(n)\}$, т.е.

$$\begin{aligned}x_0(n) &= x(n), \quad n = 0, 1, \dots, N/2 - 1, \\x_1(n) &= x(n + N/2), \quad n = 0, 1, \dots, N/2 - 1.\end{aligned}$$

При таком разбиении N -точечное ДПФ последовательности $x(n)$ можно записать в виде

$$\begin{aligned}X(k) &= \sum_{n=0}^{N/2-1} x(n)W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) \cdot W_N^{nk} = \\&= \sum_{n=0}^{N/2-1} x_0(n) \cdot W_N^{nk} + \sum_{n=0}^{N/2-1} x_1(n) \cdot W_N^{(n+N/2)k}.\end{aligned}$$

Учитывая, что

$$W_N^{(n+N/2)k} = e^{-j\frac{2\pi(n+N/2)k}{N}} = e^{-j\frac{2\pi nk}{N}} \cdot e^{-j\frac{2\pi N/2k}{N}} = W_N^{nk} \cdot e^{-j\pi k},$$

получим

$$X(k) = \sum_{n=0}^{N/2-1} [x_0(n) + e^{-j\pi k} \cdot x_1(n)] \cdot W_N^{nk}.$$

Запишем выражения отдельно для четных и нечетных отсчетов ДПФ:

$$\begin{aligned}X(k) &= \sum_{n=0}^{N/2-1} [x_0(n) + x_1(n) \cdot e^{-j\pi k}] \cdot W_N^{nk} = \\&= \sum_{n=0}^{N/2-1} [x_0(n) + x_1(n)] \cdot W_N^{nk},\end{aligned}$$

$$\begin{aligned}X(2k+1) &= \sum_{n=0}^{N/2-1} [x_0(n) + x_1(n) \cdot e^{-j\pi(2k+1)}] \cdot W_N^{n(2k+1)} = \\&= \sum_{n=0}^{N/2-1} [x_0(n) - x_1(n)] \cdot W_N^{n(2k+1)} = \\&= \sum_{n=0}^{N/2-1} \{ [x_0(n) - x_1(n)] \cdot W_N^n \} \cdot W_{N/2}^{nk}.\end{aligned}$$

С учетом этих выражений базовая операция БПФ с прореживанием по частоте имеет вид:

$$X(n) = x_0(n) + x_1(n),$$

$$X(n+1) = [x_0(n) - x_1(n)] \cdot W_N^n.$$

Ее графическое обозначение (так называемая «бабочка») приведено на рисунке 3.3.

Полный направленный граф восьмиточечного ДПФ с прореживанием по частоте изображен на рисунке 3.4.

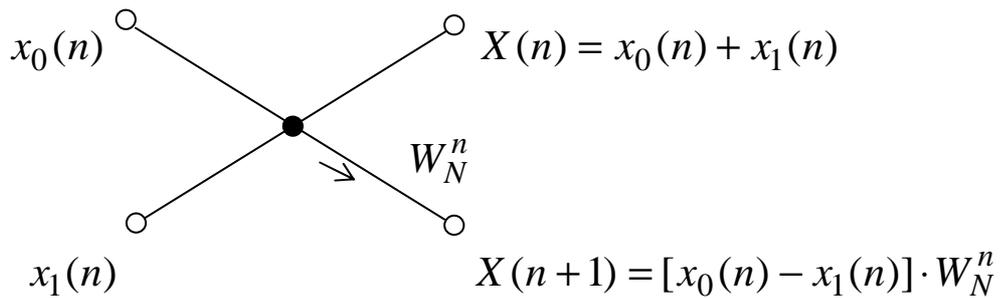


Рисунок 3.3

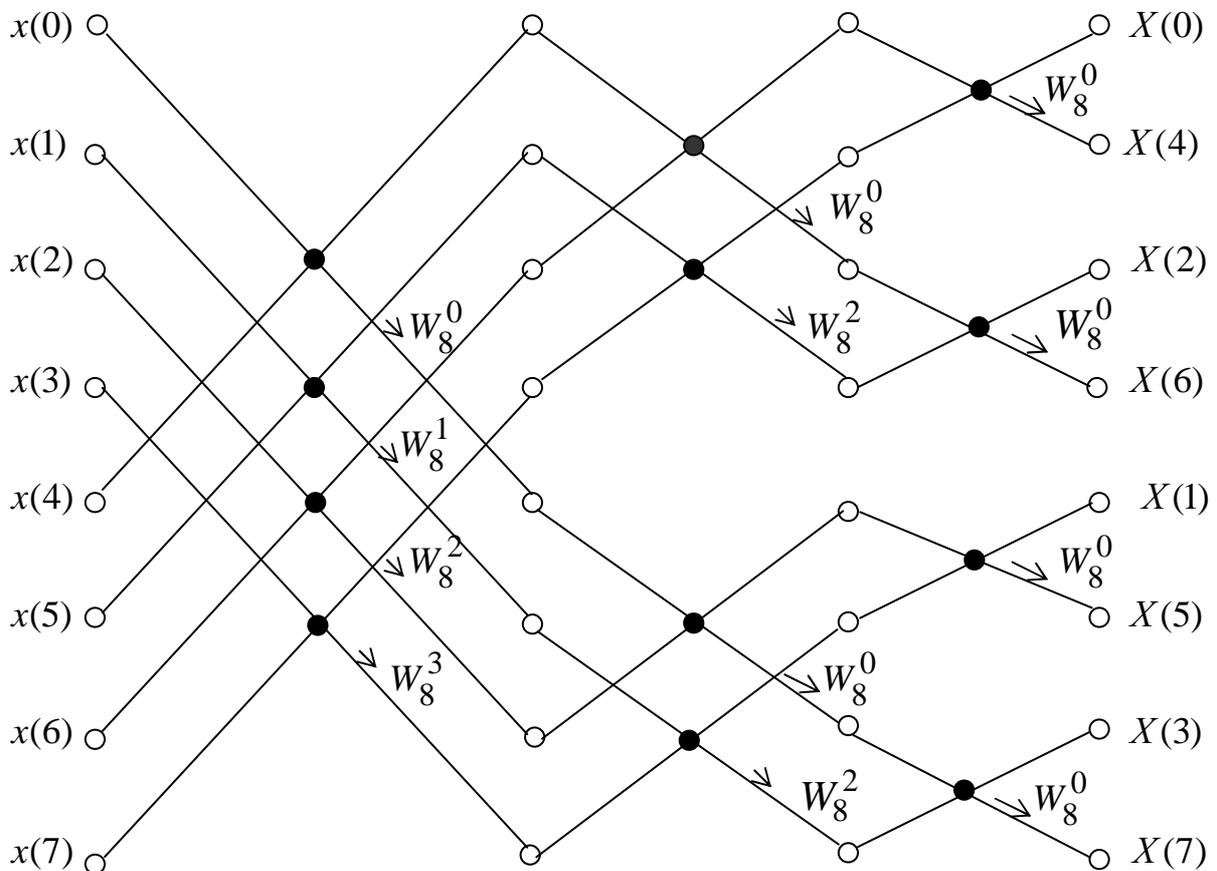


Рисунок 3.4

Сравнение алгоритмов БПФ с прореживанием по времени и частоте показывает, что основное их отличие заключается в месте проведения двоичной инверсии отсчетов (на входе или на выходе) и месте выполнения операции комплексного умножения на поворачивающий коэффициент W^{nk} (до операции сложения или после). В остальном алгоритмы очень похожи и являются как бы разнонаправленными реализациями одного процесса.

При использовании алгоритма БПФ требуется меньшее число операций, чем при прямом вычислении ДПФ. Напомним, что число комплексных умножений при выполнении прямого ДПФ равно $(N-1)^2$, а при выполнении БПФ - $(N/2) \cdot \nu$, так как на каждом этапе БПФ выполняется $N/2$ умножений, а число этапов $\nu = \log_2 N$. Необходимое условие $N = 2^\nu$ достигается доопределением исходной последовательности нулевыми отсчетами. При $N \geq 1024$ объем вычислений сокращается приблизительно на два порядка, что позволяет выполнять обработку сигналов, включающую вычисление ДПФ, в тех случаях, когда до появления БПФ она считалась неосуществимой.

Алгоритмы БПФ могут быть реализованы и для любого произвольного основания. Наиболее распространенными алгоритмами являются /12/ алгоритм с множителями поворота и алгоритмы с основаниями 4, 8 и 16, позволяющие значительно сократить число требуемых операций по сравнению с алгоритмами с основанием 2.

Отметим еще одну особенность алгоритма БПФ, заключающуюся в том, что на всех этапах преобразования используются коэффициенты W_N^k , $k = 0, 1, \dots, N-1$. Существует несколько способов получения этих коэффициентов. Простейший способ — составление таблицы, к которой можно обращаться в процессе счета. Единственный недостаток этого способа состоит в том, что для хранения этих коэффициентов необходима дополнительная память примерно из N ячеек, так что при больших значениях N имеющийся объем памяти ЦВМ может оказаться недостаточным. Второй способ заключается в непосредственном вычислении коэффициентов $W_N^k = \cos[(2\pi/N)k] - j \sin[(2\pi/N)k]$ с использованием каждый раз стандартных подпрограмм расчета синуса и косинуса. Этот способ связан с большими затратами времени, поскольку вычисление синуса и косинуса, как правило, достаточно продолжительно. Третий способ

основан на применении простой рекуррентной формулы

$$W_N^k = (W_N^{k-L})W_N^L \quad (3.6)$$

с начальным условием $W_N^0 = 1$, так как степени W на каждом этапе БПФ меняются с постоянным шагом. Так, в рассмотренном примере на первом этапе используются коэффициенты W^0 и W^4 , на втором — W^0, W^2, W^4 и W^8 , а на третьем $W^k, k = 0, 1, \dots, 7$. Поэтому, чтобы иметь возможность на каждом из этапов использовать формулу (3.6), достаточно запомнить или вычислить только множители W^8, W^4 и W^2 .

3.2 Применение алгоритмов БПФ

Вычисление амплитудного и фазового спектра. Пусть $x(t)$ — синусоидальный сигнал $x(t) = \sin(2\pi ft)$, $0 \leq t \leq 0,6$ с, $f = 50$ Гц и требуется с помощью ДПФ найти его амплитудный и фазовый спектры. Для применения алгоритма БПФ сигнал $x(t)$ следует продискретизировать и из полученной последовательности надо взять N таких дискретных значений сигнала, при которых $N=2^n$. Рассмотрим, например, случай $N = 512$ ($T = 0,001$)

$$X(m) = x(mT), \quad m = 0, 1, \dots, 511, \quad (3.7)$$

где T — интервал дискретизации и $X(0) = 1$. Амплитудный и фазовый спектры определяются как

$$C_x(k) = |X(k)|,$$

$$\psi_x(k) = \text{arctg} \{I_x(k)/R_x(k)\}, \quad k = 0, 1, \dots, 511,$$

где $R_x(k)$ и $I_x(k)$ — действительная и мнимая части $X(k)$. Использование алгоритма БПФ позволяет получить спектры, изображенные на рисунке 3.5 (амплитудный спектр нормирован).

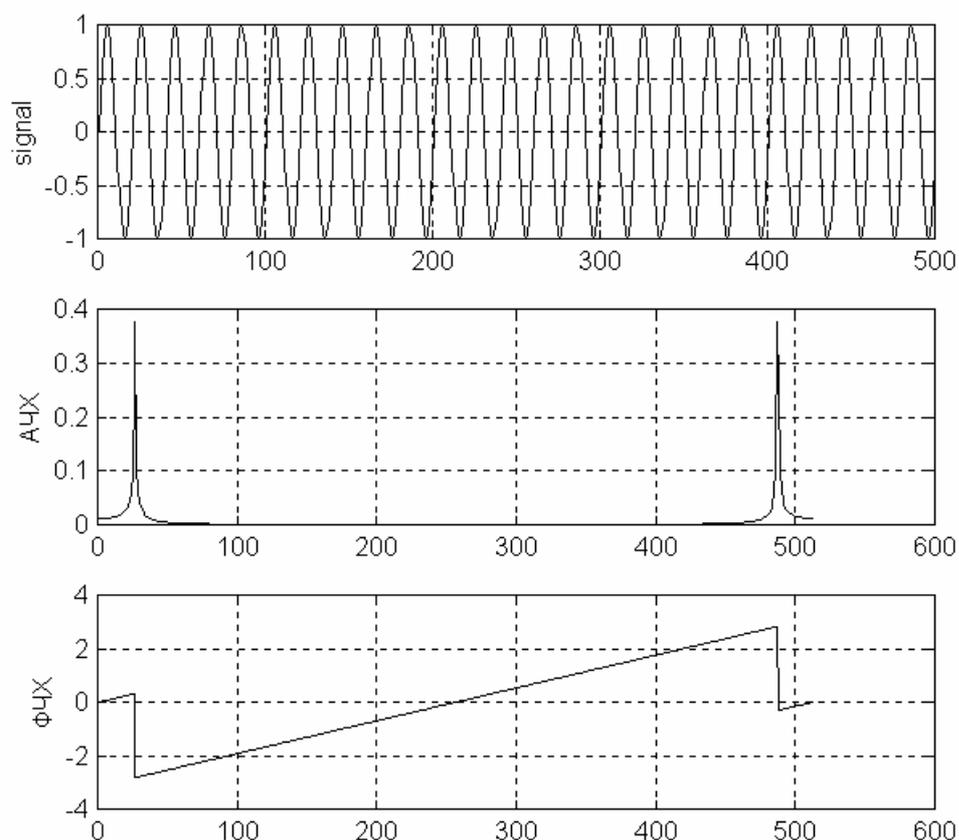


Рисунок 3.5

Из рисунка 3.5 следует, что в соответствии с теоремой о комплексной сопряженности амплитудный спектр имеет $N/2 + 1 = 257$ независимых спектральных составляющих:

$$|X(0)|, |X(1)|, \dots, |X(254)|, |X(255)|.$$

Нечетной функцией относительно точки $k = N/2$ при $N = 512$ является $\psi_x(k)$. Переменная k иногда называется «номером частоты».

Вычисление корреляционной последовательности. Если $\{X(m)\}$ и $\{Y(m)\}$ — две действительные последовательности с периодом N , то

последовательность $\{\hat{Z}(m)\}$, полученная в результате их корреляции, определяется по формуле

$$\hat{Z}(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)Y(m+h), \quad m = 0, 1, \dots, N-1.$$

Используя свойство корреляции, получим выражение

$$C_z(k) = \overline{C_x(k)} C_y(k), \quad k = 0, 1, \dots, N-1,$$

из которого следует, что $\{\hat{Z}(m)\}$ можно вычислить с помощью БПФ, как показано на рисунке 3.6.

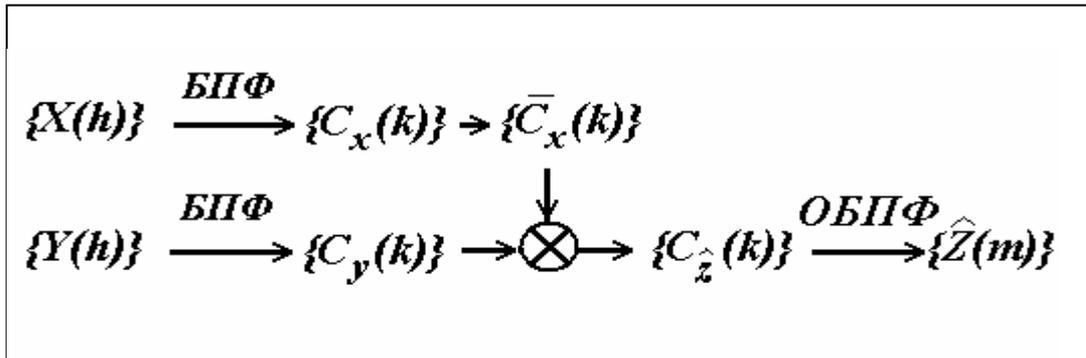


Рисунок 3.6

Вычисление свертки. Последовательность $\{Z(m)\}$, полученная в результате свертки двух действительных последовательностей $\{X(m)\}$ и $\{Y(m)\}$ с периодом N , определяется по формуле

$$Z(m) = \sum_{h=0}^{N-1} X(h)Y(m-h), \quad m = 0, 1, \dots, N-1.$$

Из свойства свертки ДПФ получаем

$$C_z(k) = C_x(k)C_y(k), \quad k = 0, 1, \dots, N-1,$$

Из формулы следует, что БПФ можно использовать для вычисления $\{Z(m)\}$, как показано на рисунке 3.7.

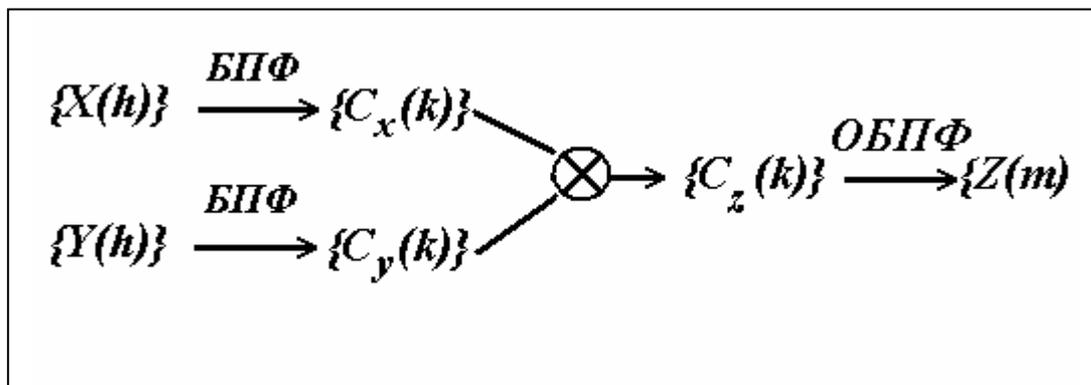


Рисунок 3.7

БПФ можно использовать подобным же образом для определения взаимной корреляции двух аperiodических последовательностей.

Синтез сигналов. Алгоритм БПФ можно использовать для синтеза L -периодического продолжения дискретного сигнала $x^*(t)$ (рисунок 3.8) по его заданному амплитудному и фазовому спектру.

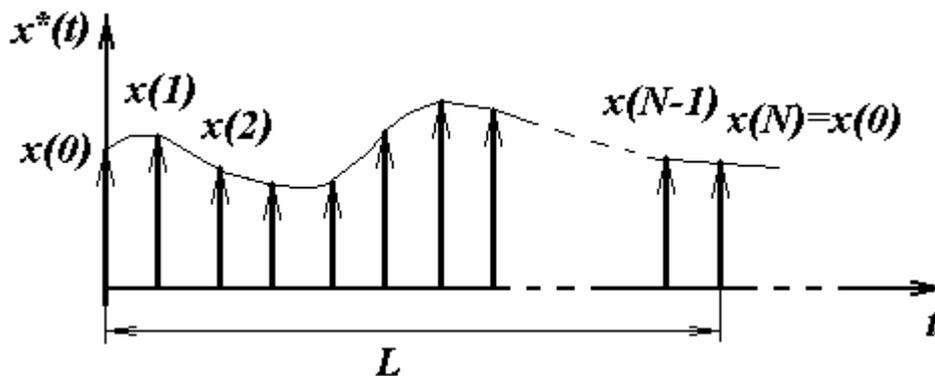


Рисунок 3.8

Пусть даны $|F_{x^*}(k\omega_0)|$ и $\psi_x(k\omega_0)$, $k = 0, 1, \dots, N/2$, где N — степень 2 и $\omega_0 = 2\pi/L$. Из $x^*(t)$ можно получить следующим образом:

$$1) F_{x^*}(k\omega_0) = |F_{x^*}(k\omega_0)| e^{i\psi_{x^*}(k\omega_0)}, \quad k = 0, 1, \dots, N/2;$$

$$2) \text{определим } C_x(k) = (1/L)F_{x^*}(k\omega_0); \quad k = 0, 1, \dots, N-1,$$

$$\text{где } C_x(N/2+l) = \overline{C_x(N/2-l)}, \quad l = 1, 2, \dots, N/2-1;$$

3) используем ОБПФ. для вычисления

$$X(m) = \sum_{k=0}^{N-1} C_x(k)W^{-km}, \quad m = 0, 1, \dots, N-1.$$

Рассмотрим пример для значений $F_{x^*}(k\omega_0)$, $k = 1, 2, \dots, 16$. При $N = 32$ в предположении $L = 32$ мс описанная выше процедура приводит к виду сигнала $x^*(t)$, изображенному на рисунке 3.9.

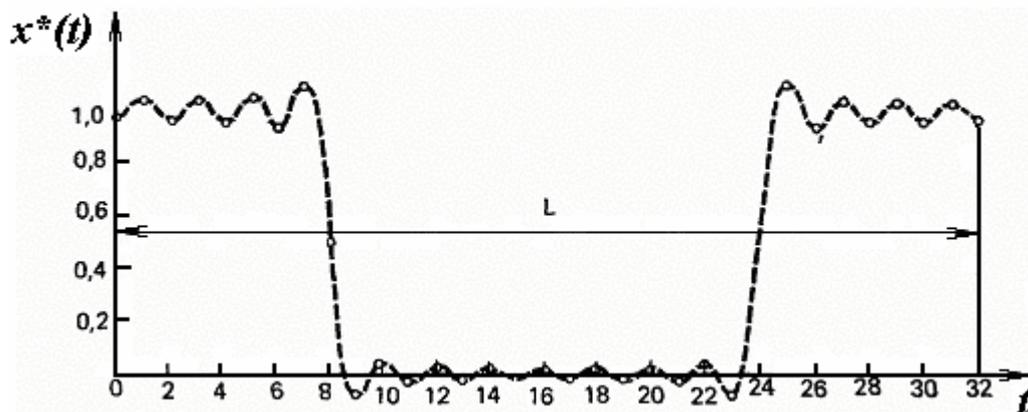


Рисунок 3.9

В данной главе рассматривался только одномерный алгоритм, когда исходные данные представляются в виде вектора. Из рассмотренного материала видно, что одномерный алгоритм БПФ можно применять $N_1 N_2$ раз для вычисления двумерного ДПФ, когда исходные данные представляются в виде матрицы размером $N_1 \times N_2$. Существует метод вычисления двумерного ДПФ с помощью одномерного БПФ, когда матрица исходных данных записывается в виде вектора размером $N_1 N_2 \times 1$. Такой подход применим и в более общем m -мерном случае, $m \geq 2$.

Вывод алгоритма БПФ выполнен на основе работ Кули и Тьюки /5,9/. На базе этого алгоритма осуществляется вывод быстрых алгоритмов и других преобразований.

3.3 Алгоритмы быстрого преобразования Уолша

3.3.1 Быстрое преобразование Уолша-Адамара, упорядоченное по Адамару

Алгоритм быстрого преобразования Уолша-Адамара (БПУА) является алгоритмом для эффективного вычисления преобразования Уолша-Адамара.

Рассмотрим вывод алгоритма с помощью разбиения матриц для случая $N = 8$. При $N = 8$ формула преобразования Уолша-Адамара

$$B_x(n) = \frac{1}{N} H_h(n) X(n) \quad (3.8)$$

записывается в виде:

$$B_x(3) = \frac{1}{8} H_h(3) X(3),$$

где $H_h(n)$ - матрица Адамара.

Пользуясь разложением

$$H_h(k) = \begin{bmatrix} H_h(k-1) & H_h(k-1) \\ H_h(k-1) & -H_h(k-1) \end{bmatrix}, \quad k = 1, 2, \dots, h, \quad (3.9)$$

где $H_h(0) = 1$ и $n = \log_2 N$, $H_h(3)$ можно выразить через $H_h(2)$

$$\begin{bmatrix} B_x(0) \\ B_x(1) \\ B_x(2) \\ B_x(3) \\ B_x(4) \\ B_x(5) \\ B_x(6) \\ B_x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} H_h(2) & H_h(2) \\ H_h(2) & -H_h(2) \end{bmatrix} \cdot \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix}. \quad (3.10)$$

Из данного преобразования следует, что

$$\begin{bmatrix} B_x(0) \\ B_x(1) \\ B_x(2) \\ B_x(3) \end{bmatrix} = \frac{1}{8} H_h(2) \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} \quad (3.11)$$

и

$$\begin{bmatrix} B_x(4) \\ B_x(5) \\ B_x(6) \\ B_x(7) \end{bmatrix} = \frac{1}{8} H_h(2) \begin{bmatrix} X_1(4) \\ X_1(5) \\ X_1(6) \\ X_1(7) \end{bmatrix}, \quad (3.12)$$

где

$$\begin{aligned} X_1(l) &= X(l) + X(4+l), \quad l = 0, 1, 2, 3; \\ X_1(l) &= X(l-4) - X(l), \quad l = 4, 5, 6, 7. \end{aligned}$$

Вновь применяя разложение (3.9) к формулам (3.11) и (3.12), получаем:

$$\begin{bmatrix} B_x(0) \\ B_x(1) \\ B_x(2) \\ B_x(3) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} \mathbf{H}_h(1) & \mathbf{H}_h(1) \\ \mathbf{H}_h(1) & -\mathbf{H}_h(1) \end{bmatrix} \cdot \begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} \quad (3.13)$$

$$\begin{bmatrix} B_x(4) \\ B_x(5) \\ B_x(6) \\ B_x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} \mathbf{H}_h(1) & \mathbf{H}_h(1) \\ \mathbf{H}_h(1) & -\mathbf{H}_h(1) \end{bmatrix} \cdot \begin{bmatrix} X_1(4) \\ X_1(5) \\ X_1(6) \\ X_1(7) \end{bmatrix}. \quad (3.14)$$

Из разбиения матрицы, приведенного в (3.13) и (3.14), получаем следующие выражения:

$$\begin{bmatrix} B_x(0) \\ B_x(1) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_1(0) + X_1(2) \\ X_1(1) + X_1(3) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_2(0) \\ X_2(1) \end{bmatrix};$$

$$\begin{bmatrix} B_x(2) \\ B_x(3) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_1(0) - X_1(2) \\ X_1(1) - X_1(3) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_2(2) \\ X_2(3) \end{bmatrix};$$

$$\begin{bmatrix} B_x(4) \\ B_x(5) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_1(4) + X_1(6) \\ X_1(5) + X_1(7) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_2(4) \\ X_2(5) \end{bmatrix};$$

$$\begin{bmatrix} B_x(6) \\ B_x(7) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_1(4) - X_1(6) \\ X_1(5) - X_1(7) \end{bmatrix} = \frac{1}{8} \mathbf{H}_h(1) \begin{bmatrix} X_2(6) \\ X_2(7) \end{bmatrix}.$$

Так как

$$\mathbf{H}_h(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

то эти формулы приводят к виду:

$$\begin{aligned}
8B_x(0) &= X_2(0) + X_2(1) = X_3(0); \\
8B_x(1) &= X_2(0) - X_2(1) = X_3(1); \\
8B_x(2) &= X_2(0) + X_2(3) = X_3(2); \\
8B_x(3) &= X_2(2) - X_2(3) = X_3(3); \\
8B_x(4) &= X_2(4) + X_2(5) = X_3(4); \\
8B_x(5) &= X_2(4) - X_2(5) = X_3(5); \\
8B_x(6) &= X_2(6) + X_2(7) = X_3(6); \\
8B_x(7) &= X_2(6) - X_2(7) = X_3(7).
\end{aligned}$$

Граф БПУА при $N = 8$ показан на рисунке 3.10.

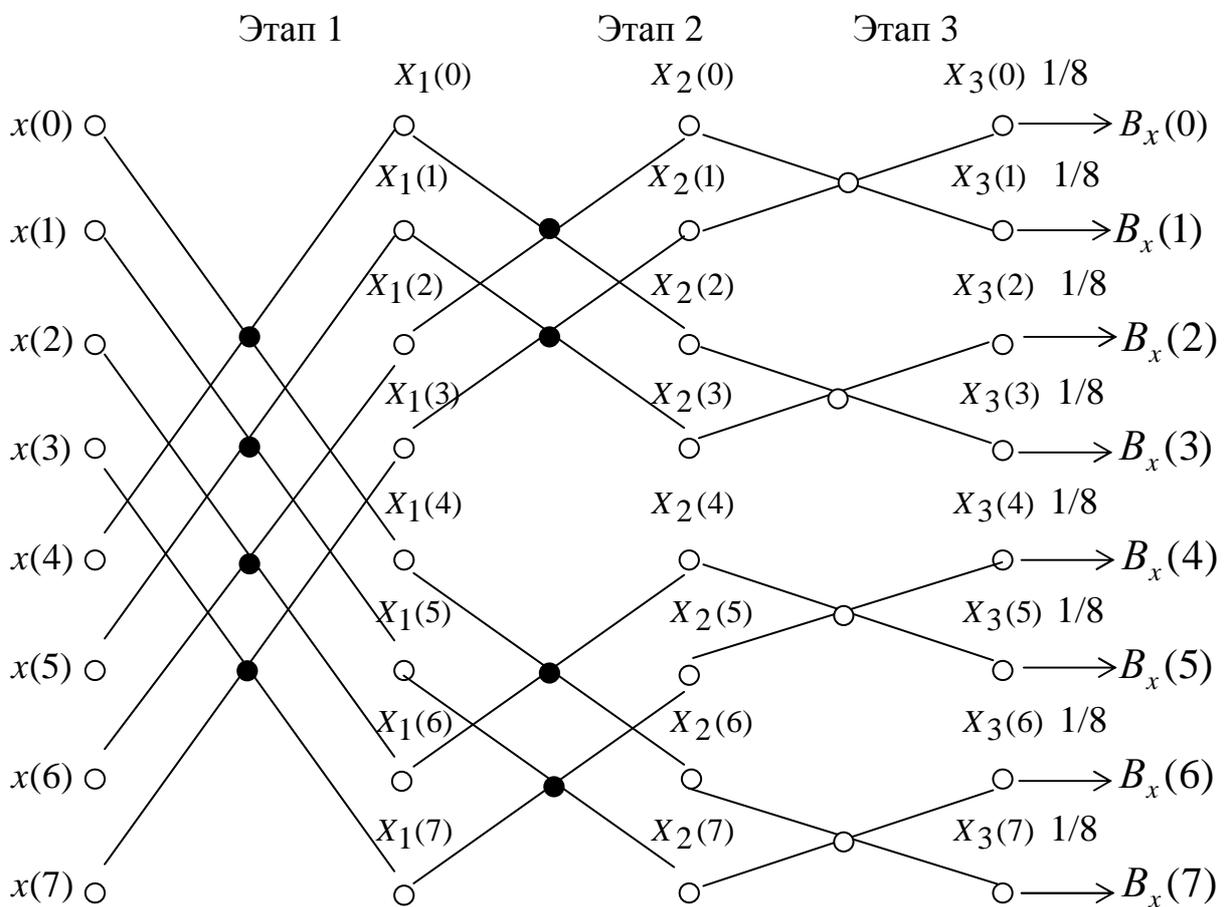


Рисунок 3.10

Из графа БПУА следует, что для его осуществления, за исключением нормировки с помощью множителя $1/8$, требуются только сложения и вычитания. Число операций сложения (вычитания), не-

обходимых для вычисления восьми коэффициентов $(WHT)_h$, равняется $N \cdot \log_2 N = 8 \cdot \log_2 8 = 24$.

Обобщения БПУА с упорядочением по Адамару могут быть получены непосредственно. Общая структура графа БПУА для любого $N = 2^n$ такая же, как у графа, изображенного на рисунке 3.10. Общее число итераций равно $n = \log_2 N$. Индекс итераций r принимает значения $r = 1, 2, \dots, n$. В r -ой итерации участвует 2^{r-1} групп по $N/2^{r-1}$ элементов в каждой группе. Половина элементов в каждой группе связана с операцией сложения, а другая – с операцией вычитания. Общее число арифметических операций, необходимое для вычисления всех коэффициентов преобразования, равняется приблизительно $N \log_2 N$ по сравнению с N^2 операциями дискретного преобразования Уолша-Адамара. Алгоритм БПУА можно применять для вычисления обратного преобразования Уолша-Адамара.

Пример. Пусть дана последовательность $\{X(m)\} = \{1 \ 2 \ 1 \ 1 \ 3 \ 2 \ 1 \ 2\}$. Пользуясь БПУА с упорядочением по Адамару, вычислить коэффициенты ПУА $B_x(k)$, $k = 0, 1, \dots, 7$.

Решение. Проводя вычисления в соответствии с графом, изображенным на рисунке 3.10, получаем:

$$X(0) = 1; \quad X_1(0) = 4; \quad X_2(0) = 6; \quad X_3(0) = 13 \xrightarrow{1/8} B_x(0);$$

$$X(1) = 2; \quad X_1(1) = 4; \quad X_2(1) = 7; \quad X_3(1) = -1 \xrightarrow{1/8} B_x(1);$$

$$X(2) = 1; \quad X_1(2) = 2; \quad X_2(2) = 2; \quad X_3(2) = 3 \xrightarrow{1/8} B_x(2);$$

$$X(3) = 1; \quad X_1(3) = 3; \quad X_2(3) = 1; \quad X_3(3) = 1 \xrightarrow{1/8} B_x(3);$$

$$X(4) = 3; \quad X_1(4) = -2; \quad X_2(4) = -2; \quad X_3(4) = -3 \xrightarrow{1/8} B_x(4);$$

$$X(5) = 2; \quad X_1(5) = 0; \quad X_2(5) = -1; \quad X_3(5) = -1 \xrightarrow{1/8} B_x(5);$$

$$X(6) = 1; \quad X_1(6) = 0; \quad X_2(6) = -2; \quad X_3(6) = -1 \xrightarrow{1/8} B_x(6);$$

$$X(7) = 2; \quad X_1(7) = -1; \quad X_2(7) = 1; \quad X_3(7) = -3 \xrightarrow{1/8} B_x(7);$$

$$\text{Следовательно, } \{B_x(k)\} = \left\{ \frac{13}{8} \ \frac{-1}{8} \ \frac{3}{8} \ \frac{1}{8} \ \frac{-3}{8} \ \frac{-1}{8} \ \frac{-1}{8} \ \frac{-3}{8} \right\}.$$

Операция нормировки с $N = 2^v$ выполняется посредством сдвига результата, представленного двоичным кодом, на v разрядов вправо.

3.3.2 Быстрое преобразование Уолша-Адамара, упорядоченное по Уолшу

Быстрое преобразование Уолша-Адамара с упорядочением по Уолшу представляет собой алгоритм для вычисления ДПУ с помощью сложения и вычитания без умножения на множитель $1/N$ в выражении ДПУ

$$\mathbf{W}_x(n) = \frac{1}{N} \mathbf{H}_w(n) \mathbf{X}(n).$$

Коэффициенты преобразования могут быть получены путем перестановки коэффициентов БПУА, упорядоченного по Адамару, но в этом случае требуется переход от кода Грея к двоичному коду, а также двоичная инверсия отсчетов, поэтому такой подход является недостаточно эффективным.

Рассмотрим алгоритм БПУА с упорядочением по Уолшу, предложенный Манцем [9]. Этот алгоритм по существу является модификацией рассмотренного алгоритма БПУА и наиболее просто может быть проиллюстрирован при $N = 8$.

Первый шаг алгоритма заключается в двоичной инверсии входной последовательности и расположении её в порядке возрастания двоично-инвертированных индексов.

Если $\{X(m)\} = \{X(0)X(1)\dots X(N-1)\}$ обозначает входную последовательность, то двоично-инвертированная последовательность с возрастающими индексами будет обозначаться как $\{\hat{X}(m)\} = \{\hat{X}(0)\hat{X}(1)\dots \hat{X}(N-1)\}$. Второй шаг заключается в определении «инверсии», которая наилучшим образом может быть проиллюстрирована с помощью примера, изображенного на рисунке 3.11.

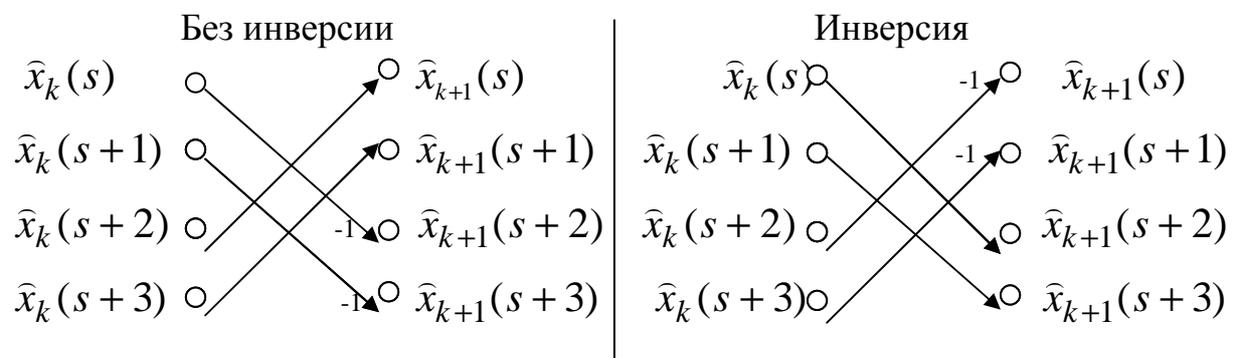


Рисунок 3.11

Обычно, т.е. без инверсии, граф преобразования строится следующим образом:

$$\left. \begin{aligned} \widehat{X}_{k+1}(s) &= \widehat{X}_k(s) + \widehat{X}_k(s+2); \\ \widehat{X}_{k+1}(s+1) &= \widehat{X}_k(s+1) + \widehat{X}_k(s+3); \\ \widehat{X}_{k+1}(s+2) &= \widehat{X}_k(s) - \widehat{X}_k(s+2); \\ \widehat{X}_{k+1}(s+3) &= \widehat{X}_k(s+1) - \widehat{X}_k(s+3); \end{aligned} \right\} \begin{array}{l} \text{сложения} \\ \text{вычитания.} \end{array}$$

С учетом инверсии граф преобразуется следующим образом:

$$\left. \begin{aligned} \widehat{X}_{k+1}(s) &= \widehat{X}_k(s) - \widehat{X}_k(s+2); \\ \widehat{X}_{k+1}(s+1) &= \widehat{X}_k(s+1) - \widehat{X}_k(s+3); \\ \widehat{X}_{k+1}(s+2) &= \widehat{X}_k(s) + \widehat{X}_k(s+2); \\ \widehat{X}_{k+1}(s+3) &= \widehat{X}_k(s+1) + \widehat{X}_k(s+3); \end{aligned} \right\} \begin{array}{l} \text{вычитания} \\ \text{сложения.} \end{array}$$

Как следует из этих формул, инверсия приводит к замене сложений в данной итерации графа на вычитания и наоборот.

Третий шаг заключается в определении «блока». Блок определяется как группа сложений и вычитаний, которая не связана с соседними группами, расположенными как выше, так и ниже. На рисунке 3.12 пунктирными линиями показано разделение графа, изображенного на рисунке 3.10, на блоки.

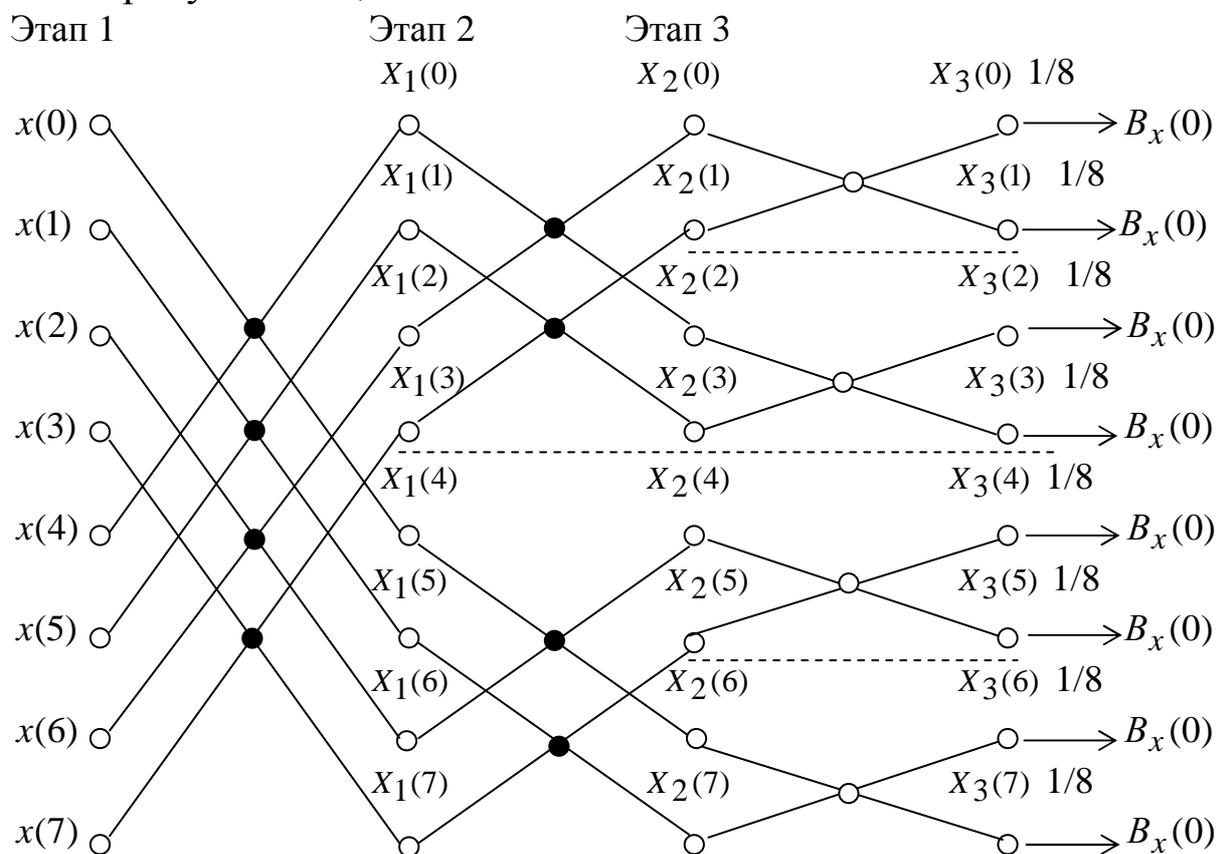


Рисунок 3.12

Наконец, приведем правила формирования блоков для алгоритма БПУА с упорядочением по Уолшу с учетом инверсий и вычитаний.

Правило 1. В итерации №1 инверсия не применяется.

Правило 2. Если обозначить через $b_m, m = 1, 2, \dots, 2^{k-1}$, блоки в k -ой итерации, то тогда блоки с инверсией располагаются, как показано ниже, где R обозначает инверсию,

$$\begin{array}{cccccc}
 b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \dots b_{2^{k-1}}, \\
 & \downarrow & & \downarrow & & \downarrow \quad \downarrow \\
 & R & & R & & R \quad R
 \end{array}$$

т.е. каждый второй блок, начиная с блока b_2 , претерпевает инверсию.

На рисунке 3.13 показан порядок инверсии при $N = 8$. В общем случае БПУА с упорядочением по Уолшу имеет граф с $n = \log_2 N$ итераций.

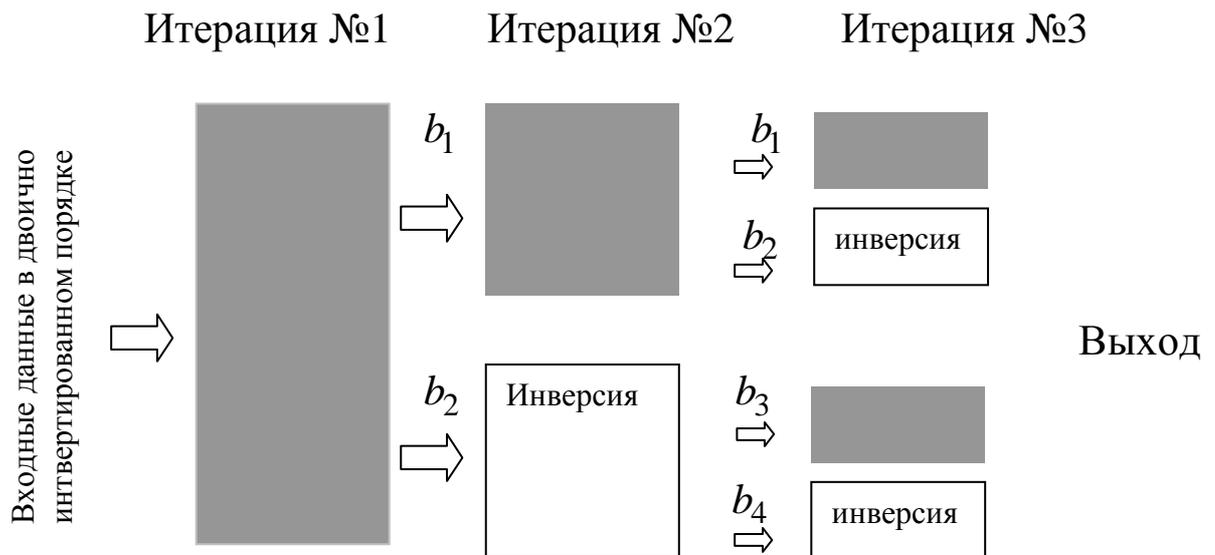


Рисунок 3.13

Таким образом, по отношению к ДПУ алгоритмы БПУА, упорядоченные по Уолшу и Адамару, имеют существенно меньшие вычислительные затраты, что позволяет использовать их на борту летательного аппарата в реальном масштабе времени.

3.4 Алгоритм быстрого преобразования Хаара

Коэффициенты преобразования Хаара $Y_x(k)$, $k = 0, 1, \dots, N - 1$ соответствующие входной последовательности $\{X(m)\} = \{X(0), X(1), \dots, X(N - 1)\}$, получаются в результате вычисления преобразования

$$Y_x(n) = \frac{1}{N} \mathbf{H}^*(n) \mathbf{X}(n),$$

где $\mathbf{H}^*(n)$ — матрица Хаара размером $N \times N$. Матрица $\mathbf{H}^*(n)$ получается в результате дискретизации множества функций Хаара $\{\text{har}(r, m, t)\}$, например, матрица Хаара 8×8 ($n = 3$) имеет вид

$$\mathbf{H}^*(3) = \begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & N/N \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & N/N \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 & N/4 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & N/4 \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & N/2 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & N/2 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & N/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & N/2 \end{array}$$

Рассматривая $\mathbf{H}^*(3)$, видим, что $N/2$ коэффициентов преобразования Хаара соответствуют корреляции двух соседних точек в пространстве входных последовательностей, $N/4$ коэффициентов соответствуют связям четырех соседних точек и т. д. до N/N коэффициентов, соответствующих всем N координатам пространства входных последовательностей. Это означает, что область преобразования в случае преобразования Хаара обладает свойством как локальной, так и глобальной чувствительности. При ДПФ и ПУА каждый коэффициент преобразования является функцией всех координат пространства входных последовательностей (свойство глобальной чувствительности), а в преобразовании Хаара это относится к первым двум коэффициентам.

Для осуществления преобразования Хаара требуется $2(N - 1)$ операций сложения/вычитания и N операций умножения, что показано на рисунке 3.14,а для $N = 8$. Этот алгоритм вычисления преобразо-

вания Хаара был предложен Эндрюсом /9/. Соответственно алгоритм для вычисления обратного преобразования Хаара изображен в виде графа на рисунке 3.14,б. Из рисунка 3.14 видно, что алгоритм Эндрюса не является алгоритмом типа Кули—Тьюки, по которому вычисляется БПФ. Ниже будет показано, что преобразование Хаара можно осуществить и с помощью алгоритма типа Кули — Тьюки.

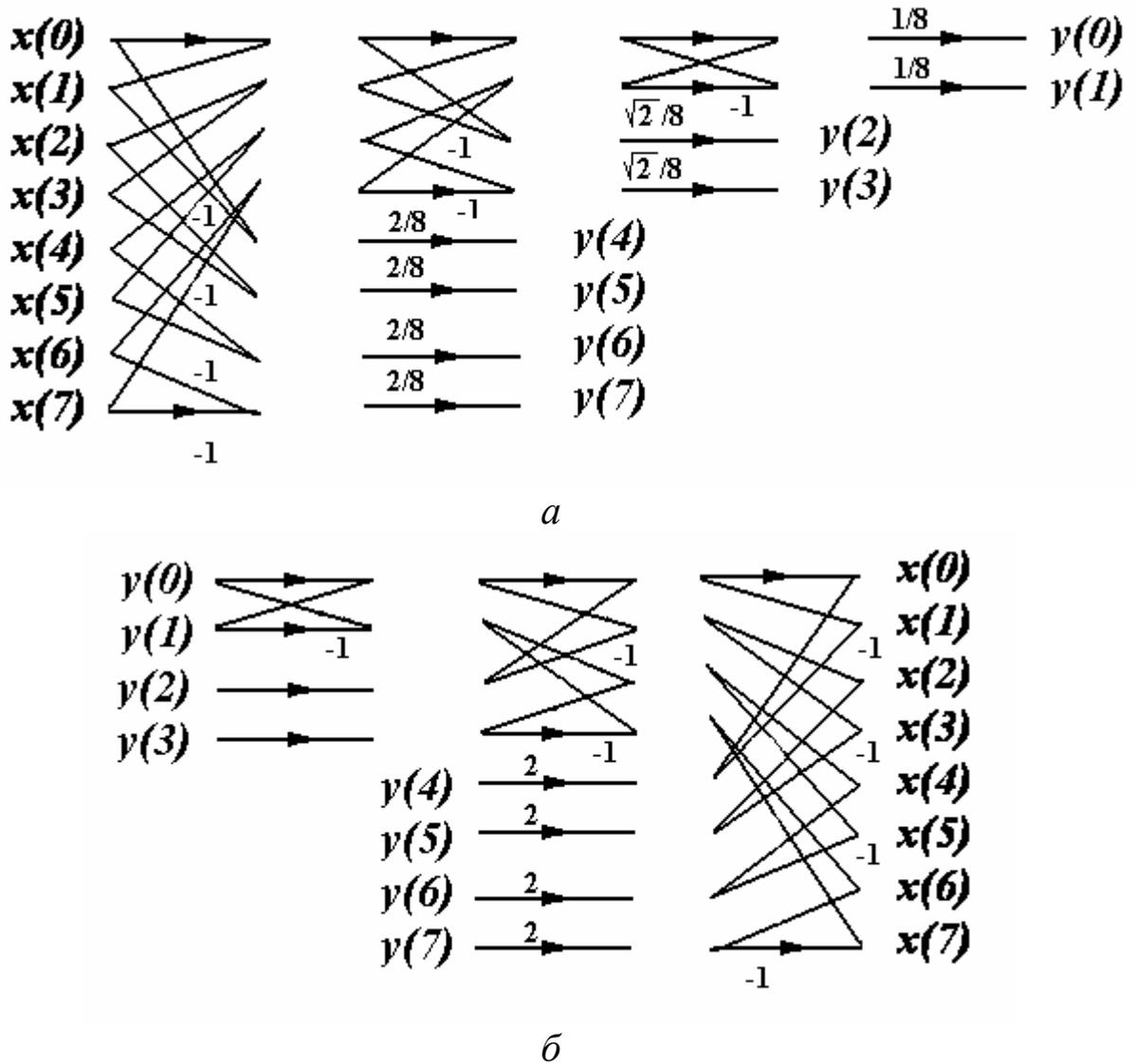


Рисунок 3.14 - Граф прямого и обратного преобразования Хаара, соответствующий алгоритму Эндрюса, $N=8$: а — прямое преобразование; б — обратное преобразование

Обоснование поиска такого алгоритма связано с тем, что процессор БПФ типа Кули — Тьюки можно использовать для вычисления ПУА с упорядочением по Адамару, ПУА с упорядочением по

Уолшу, обобщенного преобразования и преобразования Хаара дополнительно к вычислению коэффициентов ДПФ.

Этот алгоритм может быть наилучшим образом продемонстрирован при $N=8$. Запишем снова матрицу Хаара в виде:

$$\begin{array}{c}
 \text{Столбец \#} \\
 \begin{array}{cccccccc}
 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 \downarrow & \downarrow \\
 \end{array} \\
 \mathbf{H}^*(3) = \begin{array}{c} \left| \begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
 \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\
 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2
 \end{array} \right| \end{array}
 \end{array}$$

Переупорядочим столбцы $\mathbf{H}^*(3)$, пользуясь последовательно двоичной инверсией при $N=8$, $N=4$ и $N=2$, как показано ниже.

Шаг 1. Переставим столбцы $\mathbf{H}^*(3)$ в соответствии с двоичной инверсией номеров столбцов при $N=8$, т. е. $\{0, 1, 2, 3, 4, 5, 6, 7\} \rightarrow \{0, 4, 2, 6, 1, 5, 3, 7\}$, что приведет к матрице следующего вида:

$$\begin{array}{c}
 \mathbf{H}_1^*(3) = \begin{array}{c} \left| \begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
 \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\
 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} \\
 \hline
 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\
 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \\
 0 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\
 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2
 \end{array} \right| \end{array} \\
 \begin{array}{cccccccc}
 \uparrow & \uparrow \\
 \text{Столбец \#} & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3
 \end{array}
 \end{array}$$

Шаг 2. Переставим столбцы (4×4) матриц, заключенных в квадраты, в соответствии с двоичной инверсией номеров столбцов при $N=4$, т. е. $\{0, 1, 2, 3\} \rightarrow \{0, 2, 1, 3\}$. Это приводит к матрице

$$\mathbf{H}_2^*(3) = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} \\ 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 \end{vmatrix}.$$

Шаг 3. Переставим столбцы (2×2) матриц, заключенных в квадраты, в соответствии с двоичной инверсией номеров столбцов при $N=2$, т. е. $\{0,1\} \rightarrow \{1,0\}$, что приводит к матрице, совпадающей с $\mathbf{H}_2^*(3)$. Таким образом, окончательно получим:

$$\mathbf{H}_3^*(3) = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} \\ 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 \end{vmatrix}.$$

Матрица $\mathbf{H}_3^*(3)$ и матрица $\mathbf{H}(3)$ модифицированного ПУА идентичны. Отсюда следует, что преобразование Хаара при $N=8$ можно вычислить с помощью графа МПУА с незначительным изменением, показанным на рисунке 3.15.

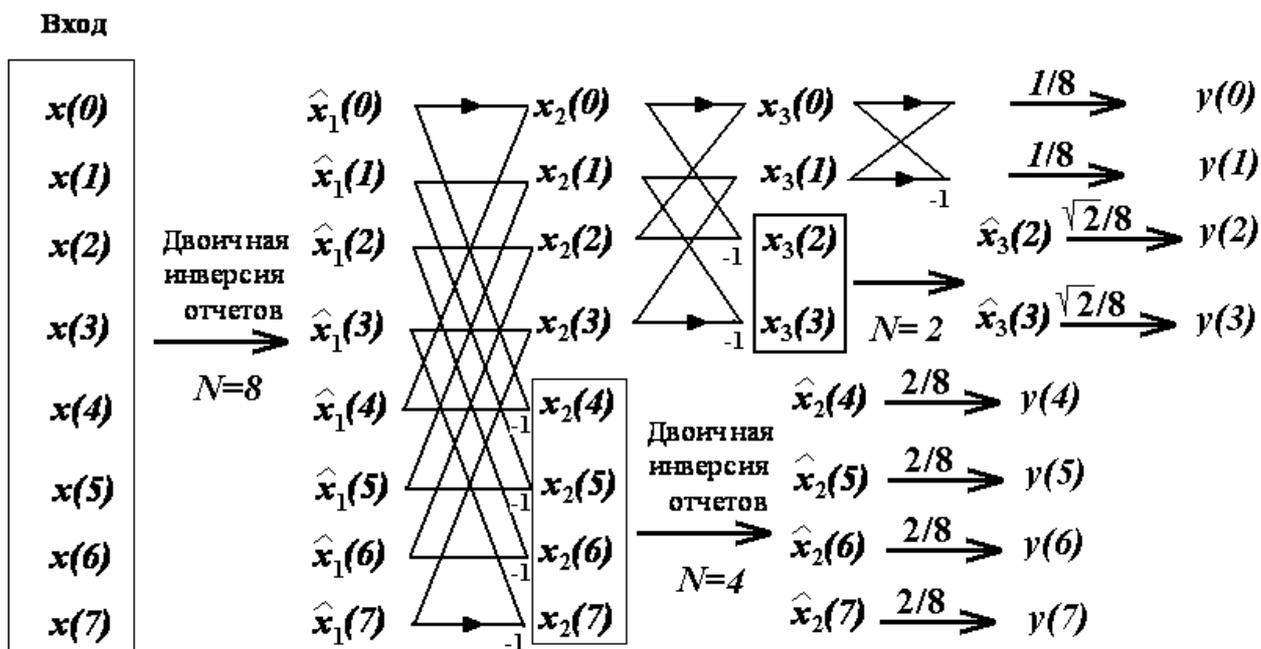


Рисунок 3.15 - Граф алгоритма Кули-Тьюки для вычисления преобразования Хаара, $N=8$

Этот граф фактически является упрощенным графом БПУА с упорядочением по Уолшу, приведенным в предыдущей главе. Из приведенного описания следует, что в общем случае для вычисления преобразования Хаара с помощью алгоритма типа Кули—Тьюки требуется $\log_2 N$ двоичных инверсий, $2(N-1)$ сложений/вычитаний и N умножений.

4 ПРИМЕНЕНИЕ АЛГОРИТМОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

4.1 Линейные алгоритмы цифровой обработки

Важнейший принцип классификации систем основан на том, что различные системы, вообще говоря, по-разному ведут себя при подаче на вход суммы нескольких сигналов. Если оператор системы таков, что справедливы равенства

$$\begin{aligned} K(\vec{U}_{ex1} + \vec{U}_{ex2}) &= K\vec{U}_{ex1} + K\vec{U}_{ex2}; \\ K(\alpha\vec{U}_{ex}) &= \alpha K\vec{U}_{ex}, \end{aligned} \quad (4.1)$$

где K - оператор системы; α - произвольное число, то данная система называется линейной.

Условие (4.1) выражает фундаментальный принцип суперпозиции.

Выполнение принципа суперпозиции позволило создать для анализа прохождения сигналов через линейные системы методы, основанные на определении реакции системы на типовые или стандартные воздействия. Так, воздействие $x(t)$ представляется в виде суммы (интеграла) типовых сигналов, т.е. $x(t) = \sum_i x_i(t)$. По каждому типо-

вому сигналу $x_i(t)$ и соответствующей ему характеристике цепи (системы) ($H(j\omega)$, $h(t)$ и т.д.) находят реакцию цепи

$$y_i = K\{x_i(t)\},$$

где K - линейный оператор цепи. Затем все найденные частные реакции суммируют (интегрируют)

$$y(t) = \sum_i y_i(t) = \sum_i K\{x_i(t)\}.$$

Суперпозиционные методы анализа, основанные на частотных характеристиках линейных систем и сигналов, называют спектральными или частотными методами анализа. Методы анализа, основанные на временных характеристиках линейных систем, называют временными методами анализа.

Основным методом линейной цифровой обработки сигналов является свертка отсчетов входной последовательности с отсчетами импульсной характеристики.

Одним из примеров дискретных линейных систем с постоянными параметрами, нашедших широкое применение в задачах спектрального анализа, являются цифровые фильтры (ЦФ).

4.1.1 Назначение, классификация, обобщенная схема цифрового фильтра

Фильтр в обобщенном смысле слова представляет собой устройство (или систему), которое преобразует заданным образом проходящий через него входной сигнал. По существу фильтр преобразует входные сигналы в выходные таким образом, что определенные полезные особенности входного сигнала сохраняются в выходном сигнале, а нежелательные свойства подавляются. Физическая природа фильтров очень разнообразна – это и воздушные фильтры, и световые, и топливные фильтры и т.д., но для нас наибольший интерес представляют электрические фильтры, предназначенные для выделения и пропускания требуемого электрического сигнала из смеси полезных и нежелательных сигналов. Часто основным признаком, по которому происходит выделение полезного сигнала, является его расположение на оси частот, поэтому близким к фильтрации понятием является частотная селекция сигнала.

Цифровые фильтры предназначены для осуществления спектрального анализа (частотной селекции) цифрового сигнала, т.е. подавления сигналов, частоты которых находятся в полосе подавления, и передачи на выход фильтра цифровых сигналов, частоты которых находятся в полосе прозрачности.

Цифровой фильтр может быть реализован как программа на универсальной цифровой ЭВМ, либо аппаратным способом в виде схемы, содержащей дискретные элементы цифровой техники, такие, как регистры, умножители и сумматоры. В настоящее время для решения задач цифровой обработки (в том числе и фильтрации) созданы специализированные однокристальные микропроцессоры, названные цифровыми сигнальными процессорами /5,6,13/.

В соответствии с частотно-избирательными свойствами различают следующие типы цифровых фильтров (рисунок 4.1):

1. Фильтр нижних частот – фильтр с полосой пропускания от 0 до некоторой частоты ω_p до бесконечности и полосой задерживания от некоторой частоты ω_s до бесконечности, где $\omega_p < \omega_s$ (рисунок 4.1,а).

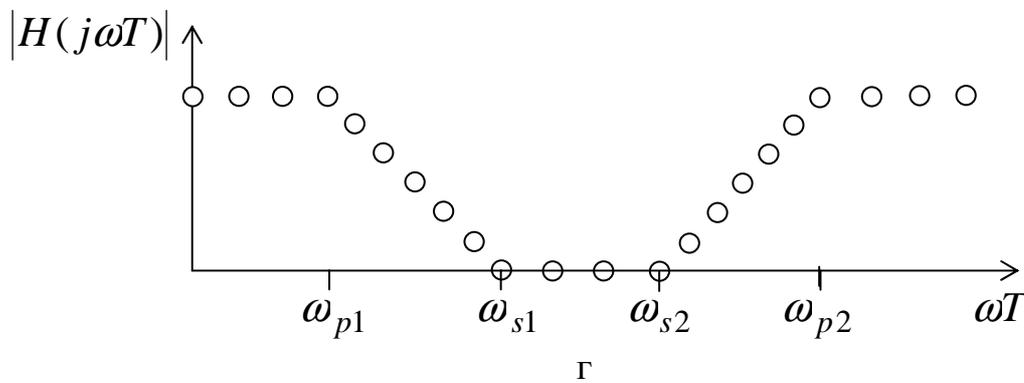
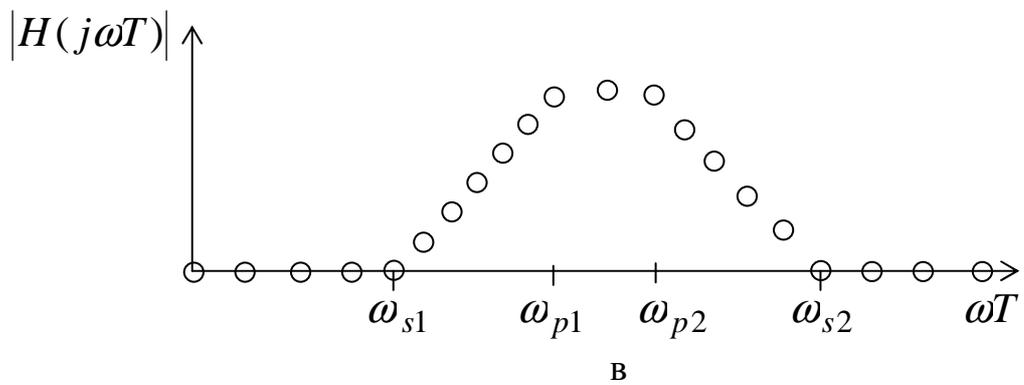
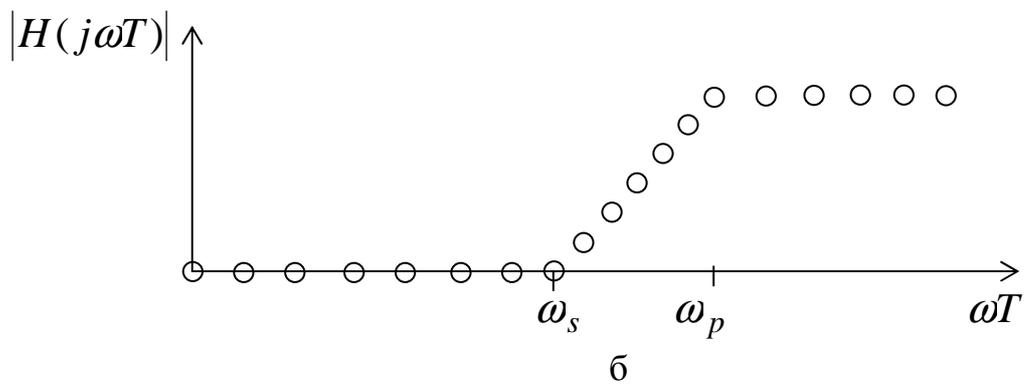
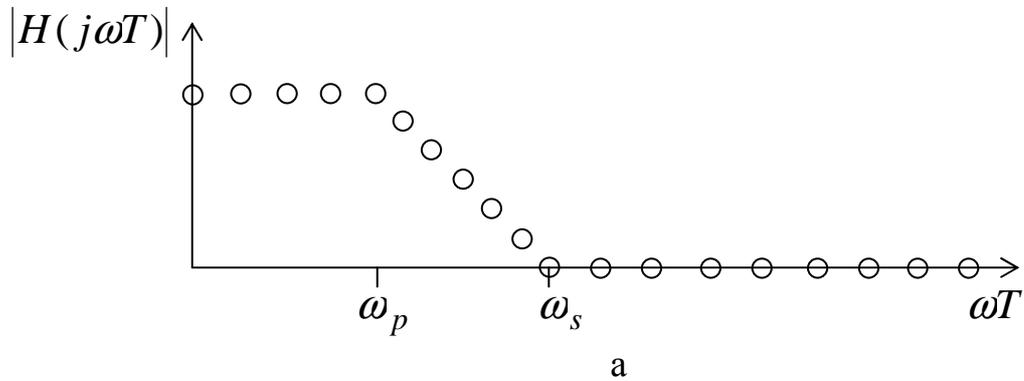


Рисунок 4.1 - Основные типы частотно-избирательных цифровых фильтров

2. Фильтр верхних частот – фильтр с полосой пропускания от некоторой частоты ω_p до бесконечности и полосой задерживания от 0 до ω_s , где $\omega_s < \omega_p$ (рисунок 4.1, б).

3. Полосовой фильтр с полосой пропускания от некоторой частоты ω_{p1} до другой частоты ω_{p2} и полосами задерживания от 0 до ω_{s1} и от ω_{s2} до ∞ , где $\omega_{s1} < \omega_{p1} < \omega_{p2} < \omega_{s2}$ (рисунок 4.1, в).

4. Заграждающий (режекторный) фильтр – фильтр с полосами пропускания от 0 до ω_{p1} и от ω_{p2} до ∞ и полосой задерживания от ω_{s1} до ω_{s2} , где $\omega_{p1} < \omega_{s1} < \omega_{s2} < \omega_{p2}$.

Полоса между частотами ω_p и ω_s называется переходной полосой.

Обобщенная структурная схема цифрового фильтра приведена на рисунке 4.2 /8/.

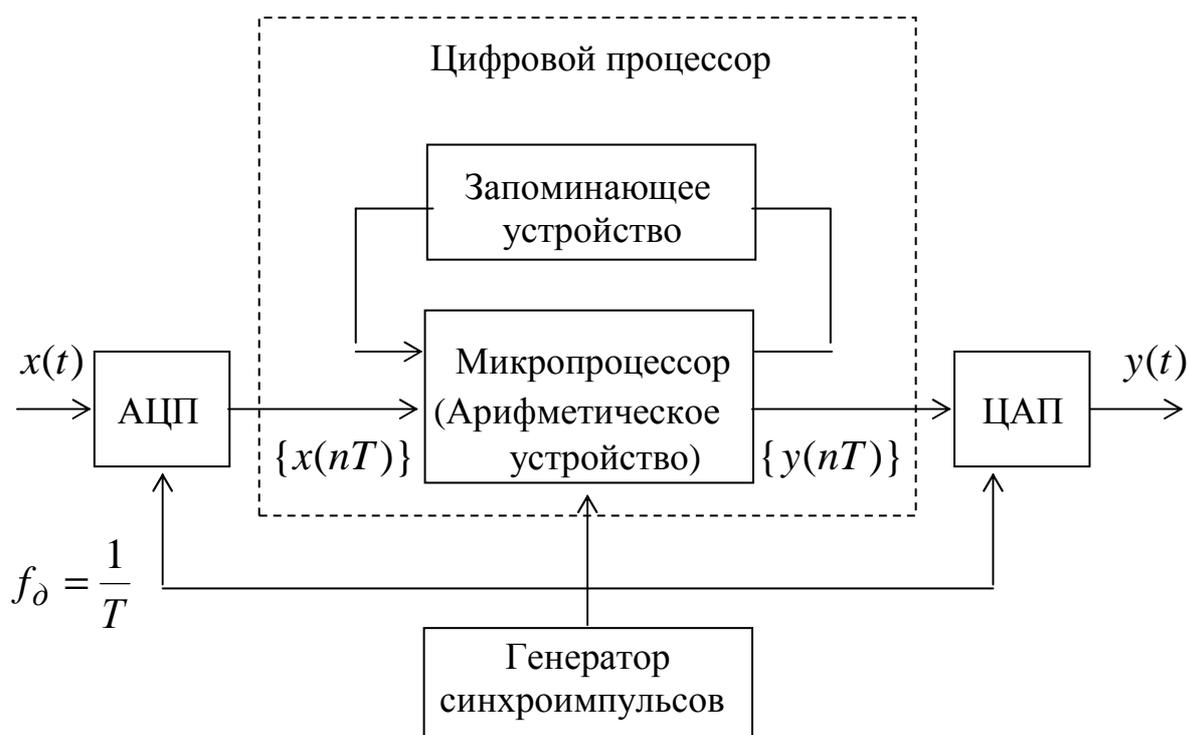


Рисунок 4.2

Принцип действия ЦФ, изображенного на рисунке 4.2, заключается в следующем. Непрерывный входной сигнал $x(t)$ поступает на первый вход аналого-цифрового преобразователя (АЦП), на второй вход которого от генератора синхроимпульсов поступает импульсная последовательность с частотой дискретизации (f_d). На выходе АЦП

в момент прихода синхроимпульса формируется цифровой код, являющийся результатом мгновенной оценки амплитуды входного сигнала $x(t)$ и имеющий фиксированное число разрядов. Цифровой код в виде последовательности двоично-квантованного сигнала $\{x(nT)\}$ поступает на вход цифрового процессора, состоящего из арифметического устройства (микропроцессора) и памяти (запоминающего устройства). Цифровой процессор преобразует поступающие на его вход двоичные числа в соответствии с заданным алгоритмом фильтрации в выходную последовательность кодов $\{y(nT)\}$, которая содержит информацию об оценке заданного параметра входного сигнала.

Если в дальнейшем необходимо иметь информацию в аналоговой форме, то к выходу цифрового процессора подключается цифро-аналоговый преобразователь (АЦП), восстанавливающий непрерывный сигнал $y(t)$ из последовательности кодов $\{y(nT)\}$.

Понятие фильтра было введено еще в 1915 г., с тех пор теория и практика аналоговой фильтрации находят все более широкое применение. Цифровая фильтрация имеет меньшую историю, но существенно лучшие перспективы, обусловленные ее достоинствами по отношению к аналоговой. Рассмотрим подробнее достоинства и недостатки цифровых фильтров.

Как известно [2,5], аналоговые фильтры физически реализуемы, если в их передаточных функциях степень полинома числителя не выше степени полинома знаменателя. Цифровые фильтры не имеют таких ограничений и могут иметь передаточные характеристики любого вида, даже такие, которые в аналоговых образцах принципиально получить невозможно.

Кроме того, главным недостатком аналоговых фильтров является неконтролируемость погрешности формирования выходного сигнала вследствие того, что параметры элементов аналоговых фильтров зависят от условий работы (температуры, давления, влажности и т.д.). Цифровые фильтры лишены этого недостатка. Они обладают стабильностью параметров, а погрешности фильтрации обусловлены лишь ошибками дискретизации и квантования.

Цифровой фильтр представляет собой цифровое устройство, реализующее заданный алгоритм. При этом входной и выходной сигналы являются цифровыми, так что в устройстве циркулируют только двоичные коды. Поскольку операция умножения отсчетов цифрового сигнала на коэффициенты фильтра иногда выполняется неточно за счет округлений или усечения результата произведений, то в общем

случае цифровой фильтр неточно реализует алгоритм фильтрации, и выходной сигнал, следовательно, отличается от точного решения. Однако в ЦФ погрешность выходного сигнала не зависит от условий, при которых работает фильтр. Причем, эта погрешность контролируема – ее можно уменьшить, увеличивая число разрядов, используемых для представления отсчетов цифровых сигналов.

Именно этим определяются основные достоинства ЦФ:

1. Высокая точность обработки сигналов и стабильность характеристик цифрового фильтра.

2. Возможность реализации ряда операций и алгоритмов, принципиально нереализуемых с помощью аналоговых элементов. Например: построение фильтра с идеально прямоугольной характеристикой; обработка сверхнизкочастотных сигналов, когда требуется неограниченное время хранения сигналов и т.д.

3. ЦФ легко реализуются в виде больших интегральных схем (БИС) специализированных микропроцессов, имеют малые габариты и вес.

С другой стороны, вследствие того, что ЦФ имеют дело с дискретным квантованным сигналом, им присущи специфические недостатки:

1. Относительно низкая скорость обработки сигналов, которая ограничивается временем задержки сигналов в цифровых элементах фильтра, то есть временем, затрачиваемым на операции умножения, суммирования, сдвига.

2. Относительно большая потребляемая электрическая мощность.

3. Необходимость использования на входе и выходе элементов АЦП и ЦАП. При этом потенциальная точность обработки сигналов ограничена характеристиками ЦАП и АЦП (разрядность, быстродействие и др.).

4. Необходимость предварительной аналоговой селекции входного сигнала и подавления помеховых колебаний, частота которых больше, чем $\omega_\delta/2$, т.к. АЧХ цифрового фильтра является периодической функцией с периодом равным ω_δ и соответствует заданным свойствам на интервале от $-\omega_\delta/2$ до $\omega_\delta/2$.

Математическая теория цифровых фильтров является частным случаем теории фильтрации применительно к дискретным сигналам. Любой аналоговый фильтр может быть представлен следующий математической моделью:

$$y(t) = \int_0^{\infty} x(\tau)h(t - \tau)d\tau, \quad (4.2)$$

где $x(\tau)$ - входной сигнал;

$h(t - \tau)$ - импульсная характеристика аналогового фильтра;

$t - \tau$ - интервал времени между моментом t появления выходного сигнала, соответствующего $x(\tau)$, и моментом прихода собственно $x(\tau)$.

Для линейных дискретных систем, каким являются цифровые фильтры, выражение (4.2) примет следующий вид:

$$y(n) = \sum_{m=0}^{\infty} x(m) \cdot h(n - m), \quad (4.3)$$

где $x(m)$ и $h(n - m)$ соответственно дискретные отсчеты входного сигнала и импульсной характеристики.

Для физически реализуемых фильтров все $h(n - m)$, для которых $n - m < 0$, равны нулю. Это означает, что выходной сигнал цифрового фильтра в n -й момент времени является взвешенной суммой всех предыдущих входных воздействий.

Импульсная характеристика ЦФ представляет собой результат дискретизации с периодом T_{Δ} импульсной характеристики некоторого аналогового фильтра прототипа, обладающего такой же амплитудно-частотной характеристикой (АЧХ). Чем меньше период дискретизации, тем ближе АЧХ цифрового фильтра и АЧХ аналогового фильтра-прототипа. Кроме того, АЧХ цифрового фильтра будет периодической функцией с периодом ω_{Δ} , поэтому о подобии АЧХ можно говорить лишь для интервала частот:

$$\frac{\pi}{T_{\Delta}} < \omega < \frac{\pi}{T_{\Delta}},$$

$$-\frac{\omega_{\Delta}}{2} < \omega < \frac{\omega_{\Delta}}{2}.$$

Выражение (4.3) может быть представлено в следующем виде:

$$y_n = h_0 x_n + h_1 x_{n-1} + h_2 x_{n-2} + \dots + h_m x_{n-m}. \quad (4.4)$$

В реальных устройствах число слагаемых ограничено, параметр m , характеризующий эти ограничения, называется порядком фильтра.

Фильтр, реализующий выражение (4.4), называется нерекурсивным или трансверсальным. Реализация такого фильтра на дискретных элементах приведена на рисунке 4.3.

В состав структурной схемы фильтра (рисунок 4.3) входят: элементы памяти (элементы задержки на период дискретизации); масштабные блоки, выполняющие в цифровой форме операции умножения на соответствующие коэффициенты; многоходовый сумматор, осуществляющий сложение полученных отсчетов. Структурная схема является графическим изображением алгоритма (4.4), который может быть легко записан в виде программы. Для этого в оперативной памяти ЦВМ образуются два массива длиной m : массив X (для хранения m - отсчетов входного сигнала) и H (для хранения масштабных коэффициентов h_i).

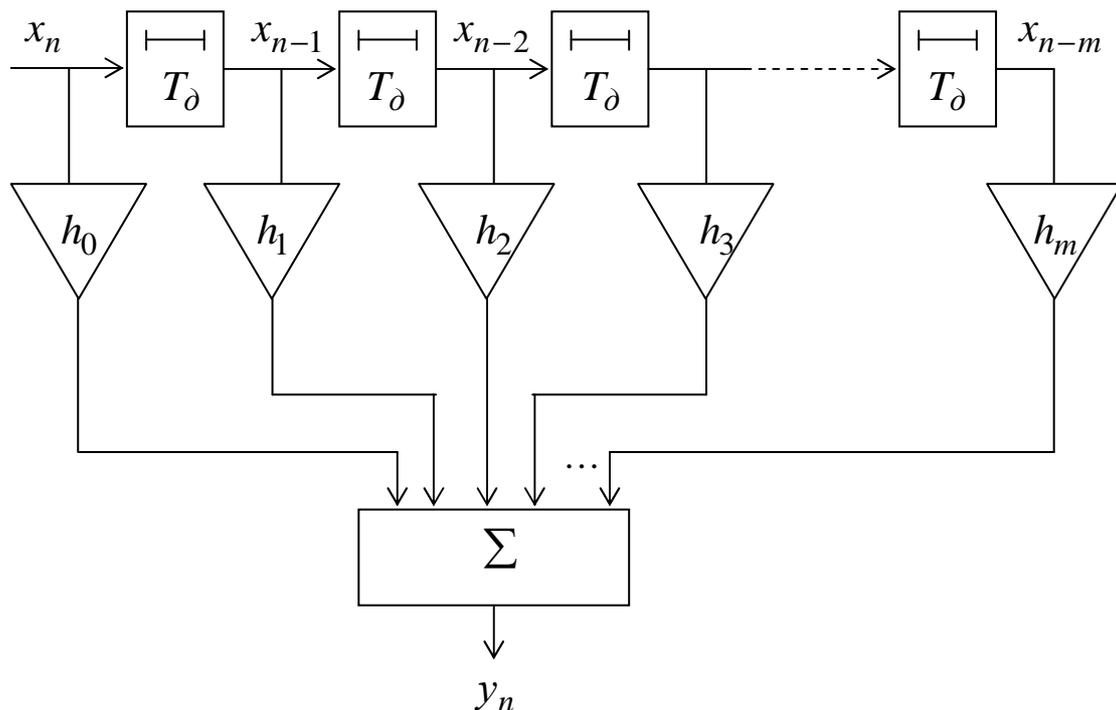


Рисунок 4.3 – Нерекурсивный цифровой фильтр на дискретных элементах

Содержание ячеек массива X меняется каждый раз с получением нового отсчета входного сигнала, то есть новый отсчет записывается в ячейку 1, а данные, которые были записаны в ней, должны быть переписаны в ячейку 2, данные ячейки 2 сдвигаются в ячейку 3

и т.д. Элементы массива X почленно умножаются на элементы массива H , суммируются, результат накапливается в массиве Y . Таким образом, на каждый такт дискретизации на выходе накопителя реализуется отсчет выходного сигнала y_n .

Кроме трансверсальных фильтров широко используются также рекурсивные фильтры. Для рекурсивных фильтров соотношение между входной последовательностью $\{x(n)\}$ и откликом фильтра $\{y(n)\}$ может быть записано следующим образом:

$$y(n) = F[y(n-1), y(n-2), \dots, x(n), x(n-1), \dots],$$

т.е. текущий отсчет отклика $y(n)$ определяется не только текущим и предшествующими значениями входной последовательности, но и предшествующими отсчетами отклика.

Как уже отмечалось, Z - преобразование любой линейной дискретной системы, в том числе и цифрового фильтра, можно выразить в виде дробно-рационального полинома по переменной Z^{-1} , т.е.

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\sum_{i=0}^N a_i Z^{-i}}{\sum_{i=0}^N b_i Z^{-i}}, \quad (4.5)$$

причем $b_0 = 1$.

Предположение о равенстве степени числителя и знаменателя обязательным не является, а служит лишь для упрощения дальнейших выводов.

Приведя равенство (4.5) к общему знаменателю, получим:

$$Y(Z) \sum_{i=0}^N b_i Z^{-i} = X(Z) \sum_{i=0}^N a_i Z^{-i},$$

или

$$\sum_{i=0}^N b_i Z^{-i} Y(Z) = \sum_{i=0}^N a_i Z^{-i} X(Z). \quad (4.6)$$

Если рассматривать члены вида $Z^{-k} Y(Z)$ как обратные Z преобразования последовательностей $y(n-k)$, то, взяв обратные Z преоб-

разования обеих частей равенства (4.6), можно получить искомое разностное уравнение:

$$\sum_{i=0}^N b_i y(n-i) = \sum_{i=0}^N a_i x(n-i). \quad (4.7)$$

Поскольку $b_0 = 1$, уравнение (4.7) можно решить относительно $y(n)$:

$$y(n) = \sum_{i=0}^N a_i x(n-i) - \sum_{i=1}^N b_i y(n-i). \quad (4.8)$$

По аналогии с реализацией выражения (4.4) структурная схема рекурсивного фильтра по выражению (4.8) примет вид, показанный на рисунке 4.4.

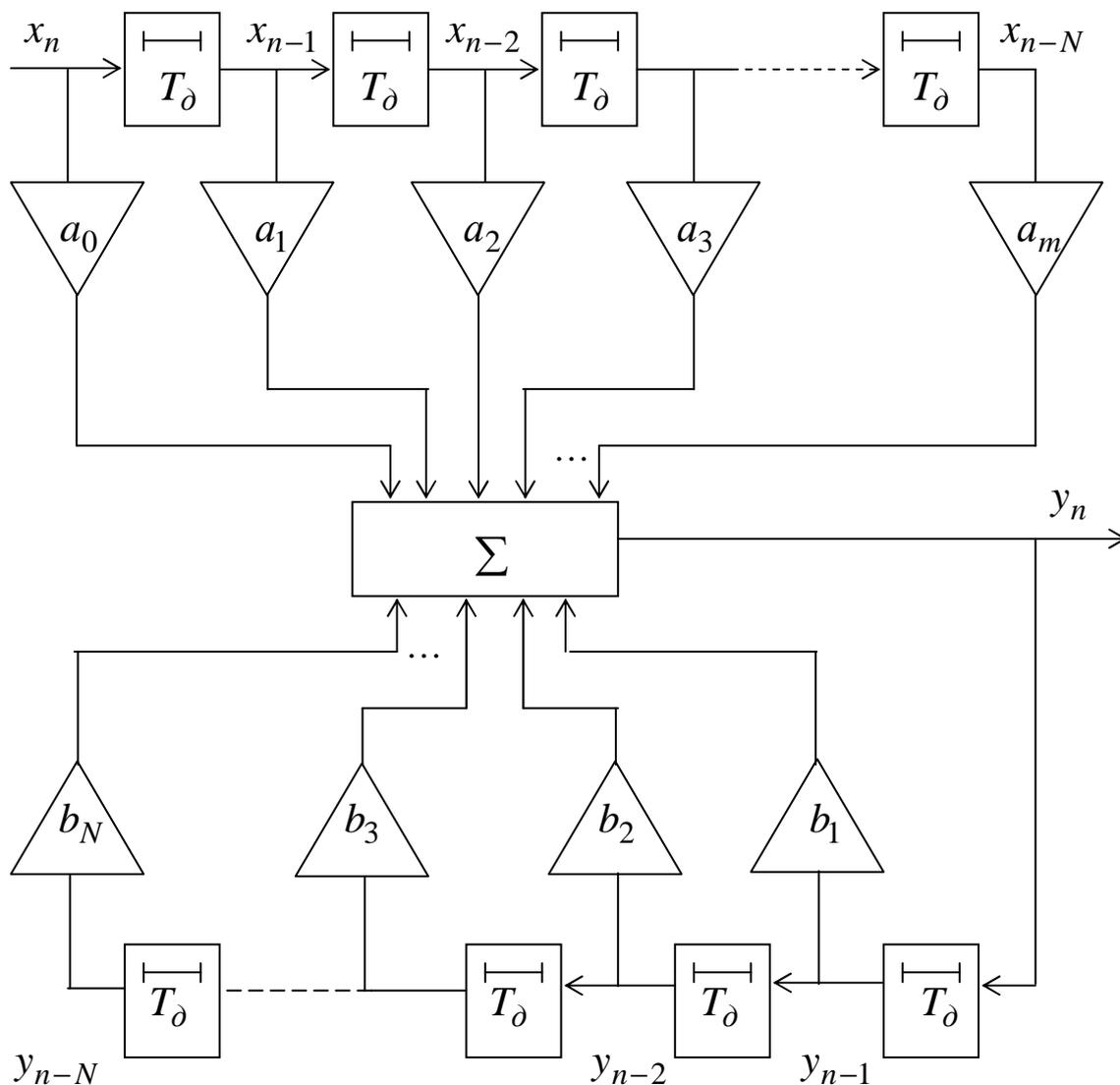


Рисунок 4.4 – Рекурсивный ЦФ

Нетрудно заметить, что верхняя часть схемы соответствует трансверсальному (нерекурсивному) фильтру. Алгоритм (4.8) также легко программируется и широко применяется как в виде вычислителя на дискретных сумматорах, умножителях и сдвиговых регистрах, так и в виде программы.

Таким образом, любой цифровой фильтр можно считать специализированным цифровым вычислителем, реализующим программу спектрального анализа, которая обеспечивает выполнение заданного алгоритма. На рисунке 4.5 показан входной сигнал, состоящий из суммы гармонических колебаний с частотами 50, 100 и 150 герц и гауссовского шума (а), спектр сигнала (б) и результат фильтрации (в) низкочастотным фильтром.

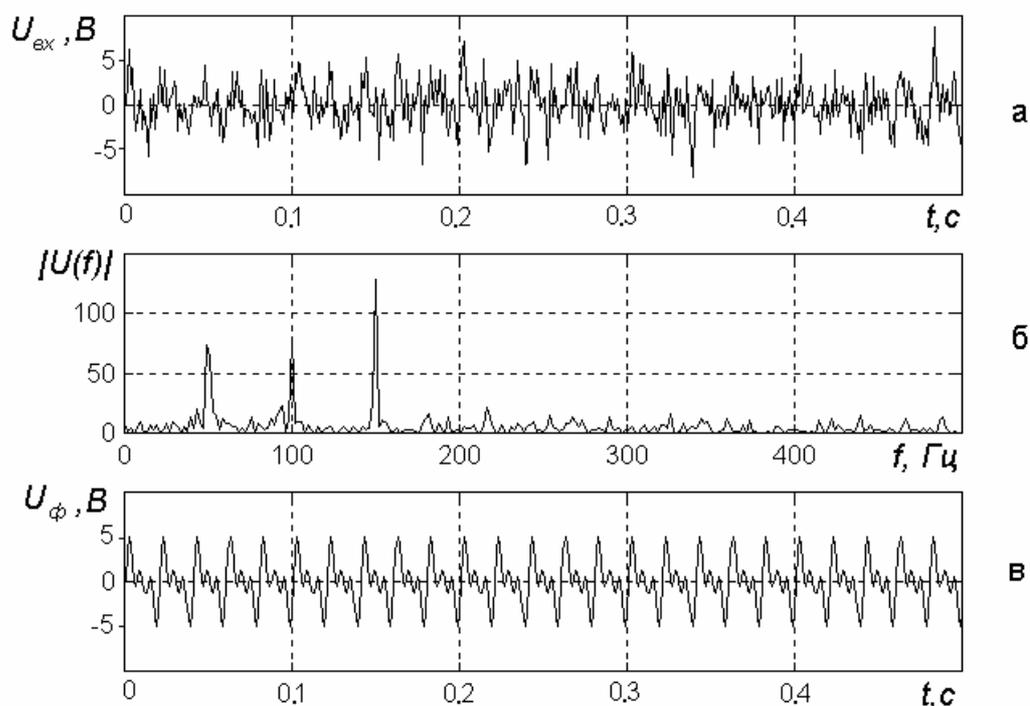


Рисунок 4.5 - Входной, зашумленный сигнал (а), его спектр (б) и результат фильтрации (в)

Импульсная характеристика $h(nT)$ представляет собой реакцию фильтра при нулевых начальных условиях на входное воздействие в виде единичного импульса

$$u_0 = \begin{cases} 1, & n = 0; \\ 0, & n \neq 0. \end{cases}$$

Из этого определения и определения передаточной функции следует, что

$$h(nT) = Z^{-1}\{H(z)\};$$

$$H(z) = Z\{h(nT)\} .$$

Например, пусть $H(z) = 1 + 0,3z^{-1} - 0,2z^{-2}$, при $n \geq 3$ тогда $h(0) = 1$; $h(T) = 0,3$; $h(2T) = -0,2$; $h(nT) = 0$, .

По виду импульсной характеристики принято делить ЦФ на фильтры с конечной импульсной характеристикой (КИХ) и бесконечной импульсной характеристикой (БИХ).

Избирательность фильтра частотной селекции определяется коэффициентом прямоугольности. Значение коэффициента прямоугольности равно отношению ширины полосы пропускания фильтра на уровне 0,9 к ширине полосы на уровне 0,1:

$$a = \frac{\Delta f_{0,9}}{\Delta f_{0,1}} .$$

Максимальное значение коэффициента прямоугольности равно единице для идеального фильтра. Аппроксимация идеальной прямоугольной АЧХ может осуществляться с использованием различных функциональных зависимостей, имеющих вид полиномов. Наибольшее распространение нашли аппроксимации полиномом Баттерворта, Чебышева и Золоторева-Кауэра (эллиптическим полиномом) /5,6,8,9/. Аппроксимация полиномом Баттерворта характеризуется максимально гладкой АЧХ, у которой отсутствуют пульсации как в полосе пропускания, так и в полосе режекции (рисунок 4.6, а). Аппроксимация полиномом Чебышева может быть с пульсациями в полосе пропускания (полином Чебышева первого рода, рисунок 4.6,б) или в полосе режекции (полином Чебышева второго рода, рисунок 4.6,в). Аппроксимация АЧХ эллиптическим полиномом допускает пульсации как в полосе прозрачности, так и в полосе режекции (рисунок 4.6,г). На рисунке 4.6 показана нормированная АЧХ низкочастотного фильтра с граничной частотой полосы пропускания $F_p = 1000$ Гц, полосы режекции $F_s = 1500$ Гц, коэффициентом затухания в полосе прозрачности не более $R_p = 3$ дБ и коэффициентом подавления в полосе режекции не менее $R_s = 15$ дБ. Порядок полинома для аппроксимации АЧХ

полиномом Баттерворта равен 5, для полиномов Чебышева 3, а для аппроксимации эллиптическим полиномом равно 2. Эта зависимость порядка полинома (и соответственно порядка фильтра) сохраняется и для других исходных данных.

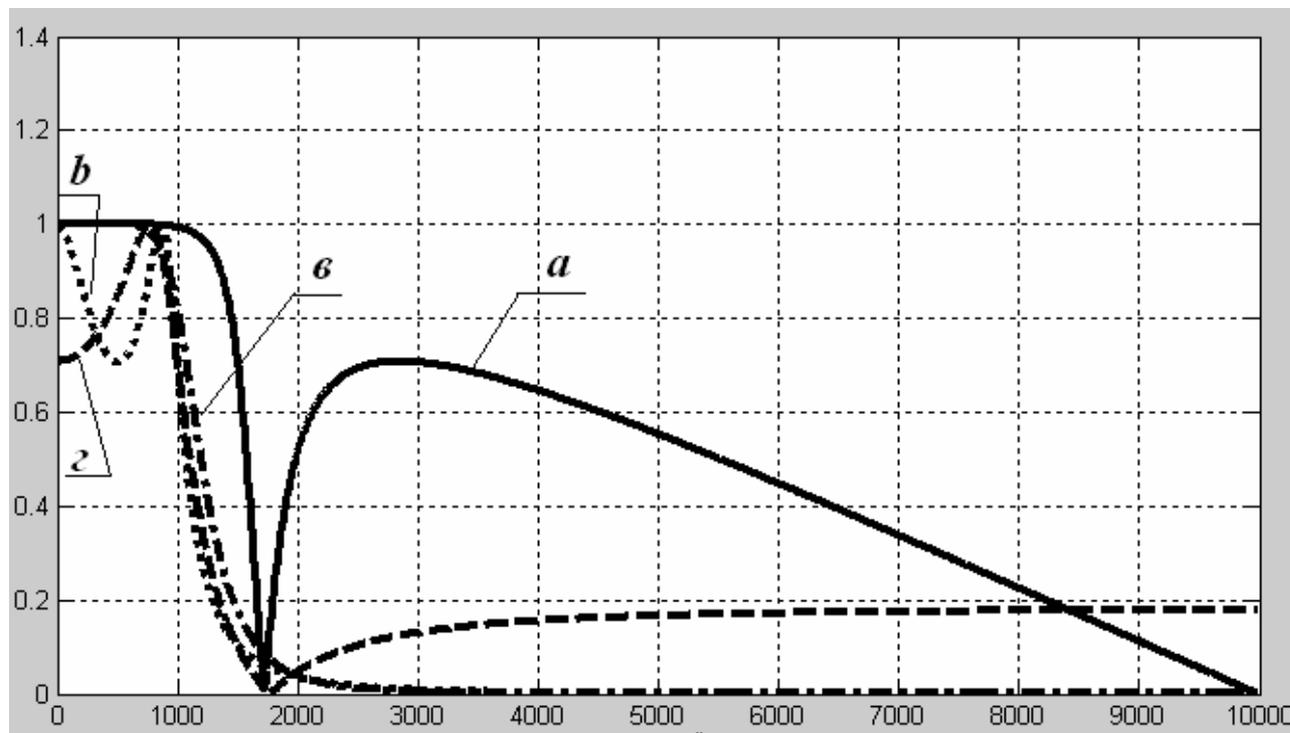


Рисунок 4.6 – Нормированная АЧХ низкочастотного фильтра. Для аппроксимации АЧХ использованы полиномы: а – Баттерворта; б – Чебышева первого рода; в – Чебышева второго рода; г – Золоторева-Кауэра.

4.1.2 Структурный синтез цифрового фильтра

Задача структурного синтеза цифрового фильтра включает: вычисление коэффициентов фильтра, определение способа соединения звеньев фильтра и построение структурной схемы фильтра.

Расчет коэффициентов фильтра можно вести прямым методом или по аналоговому прототипу. Наибольшее распространение получил метод расчета по аналоговому прототипу. Синтез по аналоговому прототипу основан на конформном отображении точек частотной характеристики аналогового фильтра из p - плоскости (операторной плоскости преобразования Лапласа) в z - плоскость. При этом может быть использовано билинейное z -преобразование, инвариантное преобразование импульсной характеристики, отображение дифферен-

циалов и т.д. /2,5/. Наибольшее распространение нашел метод билинейного z-преобразования, так как кроме основного достоинства связанного с высокой степенью соответствия результатов преобразования, для этого метода создано много программных продуктов, облегчающих задачу синтеза.

Одним из таких программных продуктов является пакет Signal processing toolbox интегрированной системы инженерных и научных расчетов MATLAB /14,15/.

Порядок выполнения структурного синтеза цифрового фильтра по аналоговому образцу методом билинейного преобразования при использовании аппроксимации полиномом Баттерворта показан на рисунке 4.7, а в таблице 4.1 приведены функции других видов аппроксимации.

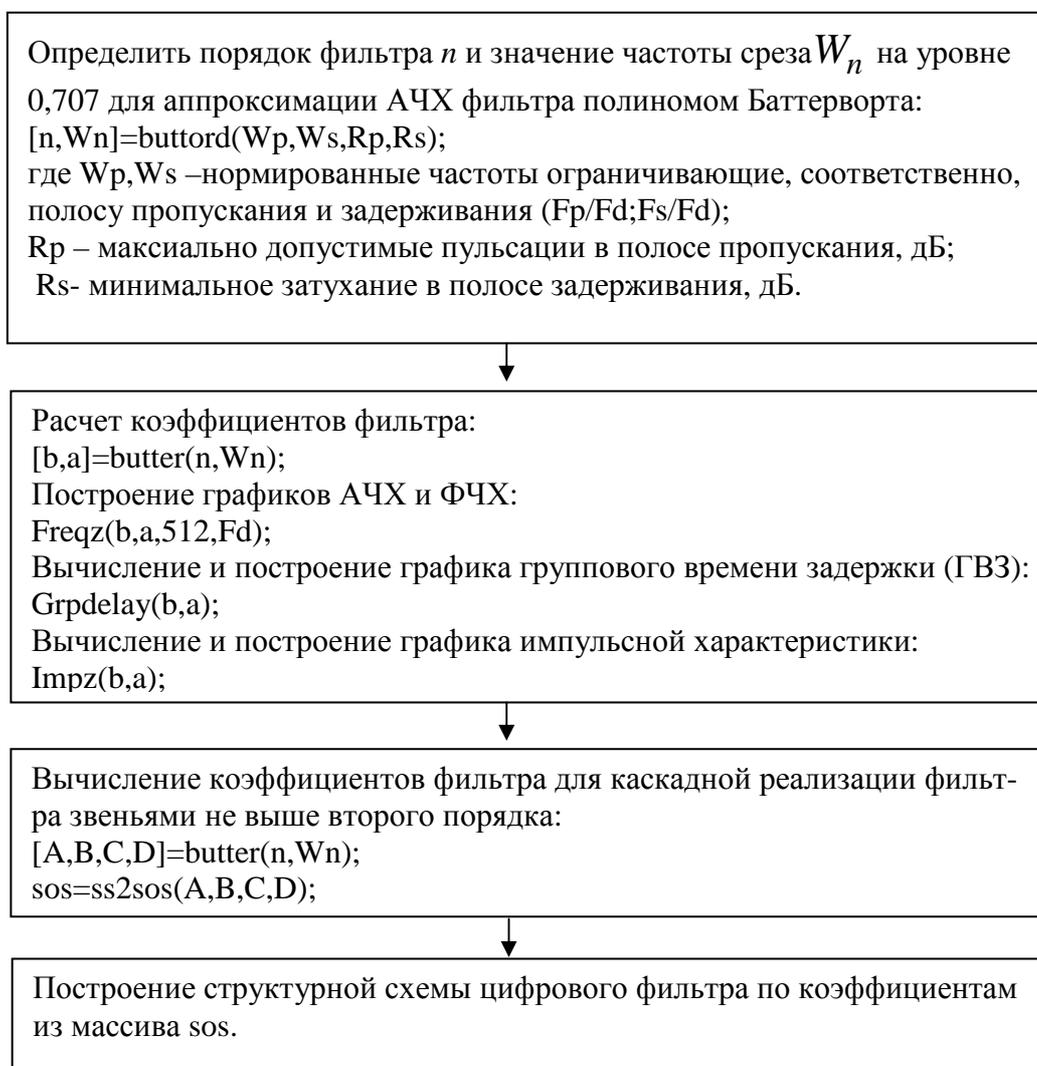


Рисунок 4.7

Таблица 4.1

Назначение функции	Полином Чебышева 1	Полином Чебышева 2	Эллиптический полином
Определение порядка ЦФ	$[n, Wn]=\text{cheb1ord}(Wp, Ws, Rp, Rs)$	$[n, Wn]=\text{cheb2ord}(Wp, Ws, Rp, Rs)$	$[n, Wn]=\text{ellipord}(Wp, Ws, Rp, Rs)$
Вычисление коэффициентов ЦФ	$[b, a]=\text{cheby1}(n, Rp, Wn)$	$[b, a]=\text{cheby2}(n, Rp, Wn)$	$[b, a]=\text{ellip}(n, Rp, Rs, Wn)$
Вычисление параметров каскадной формы	$[A, B, C, D]=\text{cheby1}(n, Rp, Wn);$	$[A, B, C, D]=\text{cheby2}(n, Rp, Wn);$	$[A, B, C, D]=\text{ellip}(n, Rp, Rs, Wn);$

4.2 Двумерные цифровые сигналы и их обработка

4.2.1 Двумерные цифровые сигналы

Цифровая обработка информации нашла широкое применение не только для обработки одномерных сигналов, чаще всего являющихся функцией времени, но и для обработки двумерных сигналов, называемых плоскостными изображениями, а также трехмерных, характеризующих, например, трехмерные пространственные объекты.

Изображения обычно задаются цифровой функцией $f(i, j)$ в виде двумерного массива (рисунок 4.8). В общем случае изображение может быть получено в любом диапазоне длин волн, хотя чаще всего используется диапазон видимого участка спектра, отсюда и название двумерного сигнала. Примером изображений являются различные фотографии, кадры видео и телефильмов, распределение яркости на экране осциллографа, а также распределение интенсивности отраженного от объекта оптического, ультразвукового или радиолокационного сигнала. Изображение есть не что иное, как функция цвета (или, в монохромном случае, функция интенсивности), определенная на участке плоскости, обычно прямоугольном или квадратном. «Реальное» изображение имеет непрерывную область определения, растровое изображение определено на дискретном наборе точек раstra - пикселях.

Таким образом, условимся называть изображением объекта двумерный сигнал, представляющий распределение в картинной

плоскости значений интенсивности (яркости) отраженного от объекта сигнала или интенсивности собственного свечения отдельных элементов объекта. Цифровые данные получаются путем выборки и квантования интенсивности точек из исходных изображений (изображений-оригиналов), например фотографий, с помощью устройства ввода изображений или путем оцифровки принятого за один кадр локационного сигнала. Элементы цифрового изображения $f(i, j)$ в i -м столбце и j -й строке могут быть заданы как бинарные черно-белые (две градации), многоградационные (например, 256 градаций) или в виде многоградационного вектора (например, с 256 градациями по каждой из составляющих — красной, зеленой и синей). В соответствии с этими представлениями изображение называется бинарным, полутоновым (монохромным) и многоспектральным (цветным).

Величина цифрового изображения $f(i, j)$ задается размерностью массива из m столбцов и n строк. На практике размер изображения, получаемый бортовыми устройствами, чаще всего колеблется от 256×256 (т. е. всего около 64000 элементов изображения до 512×512 (т. е. примерно до 256000 элементов) и более, а иногда достигает и нескольких десятков миллионов элементов. Ясно, что характерным отличием данных изображения является их громадный объем.

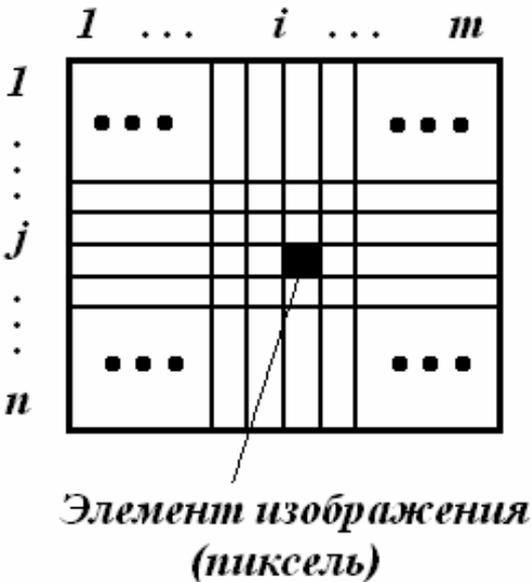


Рисунок 4.8 - Представление изображения в виде двумерного массива

Значение каждого отчета изображения в общем случае носит случайный характер. На знании статистических характеристик сигнала-

лов, в первую очередь функции распределения вероятностей, строятся многие алгоритмы обработки изображений.

Как известно, функция распределения $P(X)$ случайной величины x — это вероятность того, что случайная величина не превышает значение X [16]. Производная $P(X)$ по X

$$p(x) = dP(X)/dX |_{X=x}$$

называется плотностью распределения вероятностей случайной величины x .

Цифровые последовательности, отсчеты которых принимают конечное множество значений и количество отсчетов которых также конечно, характеризуются дискретными аналогами функции распределения и плотности распределения — относительной долей $P(m)$ отсчетов, чье значение не превышает заданную величину m , и относительной долей $his(m)$ отсчетов, имеющих заданное значение m . Последняя характеристика называется *гистограммой распределения значений*.

Гистограмма, описывающая частоту появления значений отдельных отсчетов сигнала независимо от значений других отсчетов, называется *одномерной*, или *гистограммой одномерного распределения*. Гистограмма, характеризующая частоту совместного появления значений нескольких отсчетов сигнала, называется *многомерной*, или *гистограммой многомерного распределения*.

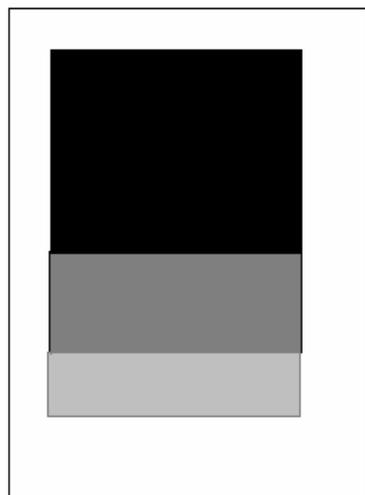
Гистограмма может использоваться как характеристика не только всего наблюдаемого изображения или всей реализации двумерного сигнала, но и отдельных их участков, или фрагментов. В этом случае она называется *локальной*.

Измерение локальных гистограмм применяется во многих адаптивных алгоритмах обработки изображений.

На рисунке 4.9 приведен черно-белый рисунок, состоящий из 57936 точек. Для представления градаций серого каждой точки (пикселя) предусмотрено 8 двоичных разрядов, что дает возможность кодирования 256 уровней. Уровню черного цвета соответствует код 0, а уровню белого — 255.

Гистограммы, измеренные по небольшим реализациям реального сигнала (небольшим фрагментам изображений), обычно довольно изрезаны. При увеличении количества измерений гистограмма сгла-

живается. Однако иногда необходимо получить сглаженную гистограмму при малом количестве измерений. Для этого применяются специальные методы, такие как ступенчатое сглаживание, сглаживание скользящим суммированием и др.



a



Медиана: 106 Число пикселей в рисунке : 57936

б

Рисунок 4.9 - Пример монохромного черно-белого рисунка (а) и его гистограмма (б); уровню черного соответствует 0

Наряду с гистограммами в качестве характеристик всего изображения или отдельных его фрагментов могут использоваться моменты распределения.

Основными моментами распределений, используемых в статистических измерениях, являются первый и второй, т.е. среднее значение

$$\bar{x} = \int_X xp(x)dx,$$

и дисперсия

$$D = \int_X x^2 p(x)dx,$$

а также среднеквадратическое отклонение

$$\sigma = \left(\int_X (x - \bar{x})^2 p(x)dx \right)^{1/2}.$$

Дискретными аналогами этих моментов являются:

$$\bar{m} = \sum_{m=0}^{M-1} m \text{ his}(m) \quad ;$$

$$D_m = \sum_{m=0}^{M-1} m^2 \text{ his}(m) \quad ; \quad (4.9)$$

$$\sigma_m = \left(\sum_{m=0}^{M-1} (m - \bar{m})^2 \text{ his}(m) \right)^{1/2} .$$

Как и гистограмма распределения, моменты могут использоваться в качестве характеристик всего изображения или отдельных его фрагментов. Такие моменты называются локальными. Использование моментов дает наибольший эффект для текстурных моделей изображения. Если изображения состоят из случайных деталей и фона, оценки, полученные по формулам (4.9), нельзя считать удовлетворительными, так как в них статистические характеристики деталей и фона смешиваются. Преодолеть это смешивание можно с помощью так называемых *робастных*, или *устойчивых оценок* параметров распределений. Термин «робастность» означает нечувствительность к малым отклонениям от предположений. По отношению к оценкам параметров распределений это нечувствительность к малым посторонним примесям в распределениях.

Наиболее известны и хорошо исследованы следующие устойчивые оценки среднего: *медиана выборки* и α - *усеченное среднее*.

Медиана выборки – это такое значение *MED* элемента выборки, для которого количество элементов выборки, имеющих большее *MED* и меньшее *MED* значений, одинаково. Медиану можно найти, расположив элементы выборки в ряд по возрастанию, или в *вариационный ряд*, и взяв средний элемент этого ряда. Медиану можно найти также из гистограммы распределения значений выборки, решив уравнение

$$\sum_{m=0}^{MED} \text{ his}(m) = 0,5 \quad .$$

Другая оценка - α - усеченное среднее – это среднее арифметическое значение элементов вариационного ряда, отстоящих не менее чем на α элементов от его концов.

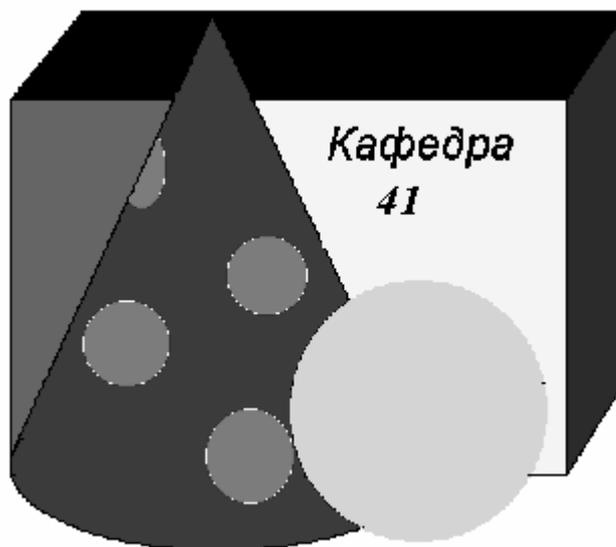
Медиана распределения и α - усеченное среднее устойчивы к примесям распределения, влияющим на его хвосты. Так, для изображений, содержащих небольшие детали на некотором фоне, они дают хорошую оценку среднего значения фона.

Ранг заданного элемента выборки, т.е. номер этого элемента в вариационном ряду, также является важной статистической характеристикой сигнала, связанной с гистограммой распределения его значений. Ранг показывает, сколько элементов выборки имеет значение, меньшее, чем значение данного элемента. Если выборка образована фрагментом сигнала, ранг (так же, как и медиана и другие рассмотренные выше характеристики) является локальным. Понятие ранга послужило причиной названия целой группы алгоритмов нелинейной обработки - ранговых алгоритмов, учитывающих статистические характеристики сигналов.

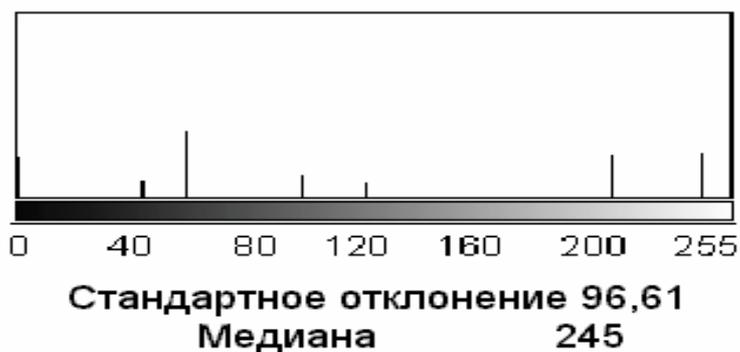
На рисунке 4.10,а приведен пример черно-белого изображения геометрических фигур. Для кодирования градаций серого (как и в случае изображения на рисунке 4.9) использовано восемь разрядов. Глобальная гистограмма этого изображения приведена на рисунке 4.10,б. Фрагмент этого же изображения показан на рисунке 4.11,а. Локальная гистограмма фрагмента приведена на рисунке 4.11,б.

Локальная гистограмма содержит меньшее число отсчетов. Все отсчеты имеют близкие значения градаций серого (величина стандартного отклонения меньше), поэтому гистограмма более компактна. Локальная гистограмма может быть использована для автоматизированного поиска (локализации) заданного объекта (фрагмента) на изображении.

Цифровая обработка изображений осуществляется с целью извлечения определенной информации, скрытой в этих изображениях. Характер этой обработки и извлекаемая информация зависят от конкретной области применения. Высокая информативность изображений и сложность обработки, вызванная большой размерностью, определили высокий интерес к вопросам цифровой обработки изображений.



а



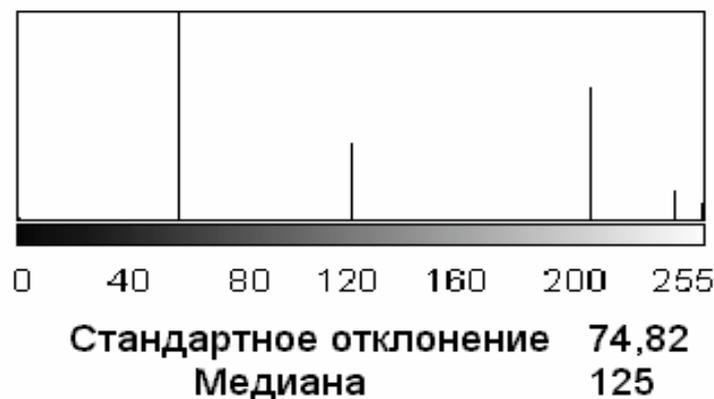
б

Рисунок 4.10 - Черно-белое изображение геометрических фигур (а)
и его глобальная гистограмма (б)

Благодаря развитию техники сверхбольших интегральных схем (СБИС) за последние годы цифровая обработка изображений находит все большее применение во многих областях, в том числе и военного назначения.



а



б

Рисунок 4.11 - Фрагмент изображения (а) и его локальная гистограмма (б)

В частности, в области воздушной и космической разведки осуществляется дистанционное зондирование земной поверхности, посредством которого производится извлечение информации об условиях ведения наземных операций, об объектах противника и расположении своих войск. В области распознавания производится классификация воздушных и наземных целей. В области автоматизации человеческой деятельности различные виды обработки изображений используются для реализации технического зрения, в том числе и в системах высокоточного оружия /17..27/.

Среди задач, связанных с обработкой изображений, выделим следующие направления:

- улучшение качества изображения;
- сжатие изображений;
- поиск и распознавание заданного объекта на изображении.

Улучшение качества изображения направлено на такое изменение исходного изображения, при котором повышается эффективность следующих этапов извлечения информации оператором или автоматической системой ЦОС. Алгоритмы повышения качества изображения условно можно разделить на две группы: работающие на основе линейной фильтрации и нелинейные ранговые алгоритмы.

4.2.2 Улучшение качества изображения на основе линейной и нелинейной фильтрации

Реальные изображения наряду с полезной информацией содержат различные помехи. Источниками помех являются собственные шумы фотоприемных устройств, зернистость фотоматериалов, шумы каналов связи. Наконец, возможны геометрические искажения, вызванные расфокусировкой изображения. Пусть $f(x, y)$ – некоторое изображение, x, y – координаты. Реальное растровое изображение имеет конечные размеры: $A \leq x \leq B, C \leq y \leq D$ и состоит из отдельных пикселей, расположенных с некоторым шагом в узлах прямоугольной сетки (рисунок 4.8). Линейное преобразование изображения можно описать выражением

$$g(x, y) = \iint f(x, y)h(x, y, x', y')dx' dy'. \quad (4.10)$$

Выражение (4.10), где интегрирование ведется по всей области определения x и y , характеризует преобразование всего изображения целиком – *глобальную фильтрацию*. Ядро преобразования $h(x, y, x', y')$ – двумерную импульсную характеристику системы обработки, в оптике именуют функцией рассеяния точки (ФРТ). Это изображение точечного источника на выходе оптической системы, которое уже является не точкой, а некоторым пятном. В соответствии с выражением (4.10), все точки изображения $f(x, y)$ превращаются в пятна, происходит суммирование (интегрирование) всех пятен. Не следует думать, что эта процедура обязательно приводит к расфокусировке изображения, наоборот, можно подобрать такую ФРТ, которая позволит сфокусировать расфокусированное изображение [7].

На рисунке 4.12 представлена одна из возможных ФРТ. Вообще говоря, ФРТ определена на $(-\infty < x < \infty), (-\infty < y < \infty)$. ФРТ не должна изменяться при изменении начала отсчета по x и y , для этого она должна иметь вид: $h(x, y, x', y') = h_1(x-x', y-y')$. Кроме того, ФРТ должна

обладать осевой симметрией. В этом случае все точки изображения «расплываются» одинаковым образом, равномерно во все стороны (принцип *пространственной инвариантности*).

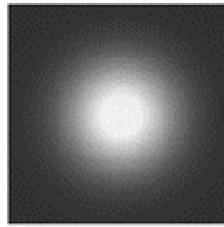


Рисунок 4.12 – Пример функции рассеяния точки

На практике глобальная фильтрация применяется редко. Чаще используют локальную фильтрацию, когда интегрирование и усреднение проводится не по всей области определения x и y , а по сравнительно небольшой окрестности каждой точки изображения. Функция рассеяния точки при этом имеет ограниченные размеры. Достоинством такого подхода является хорошее быстроедействие. Линейное преобразование принимает вид:

$$g_{ij} = \sum_D a_{kl} f_{1+k, j+l} \cdot \quad (4.11)$$

При обработке растровых изображений, которые состоят из отдельных пикселей, интегрирование заменено суммированием. Проще всего реализовать ФРТ конечных размеров в виде прямоугольной матрицы форматом $N \times N$. Число N может быть равным 3, 5, 7 и т.д. Например, при $N = 3$

$$H = \frac{1}{K} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}. \quad (4.12)$$

Суммирование ведется по окрестности D точки (i, j) ; a_{kl} – значения ФРТ в этой окрестности. Значение яркости пикселей в этой точке и в её окрестности умножаются на коэффициенты a_{kl} , преобразованная яркость (i, j) -го пикселя есть сумма этих произведений. Элементы матрицы удовлетворяют условию пространственной инвариантности, поэтому $a_{11} = a_{13} = a_{31} = a_{33}$, $a_{12} = a_{21} = a_{23} = a_{32}$. Только три элемента матрицы размером 3×3 независимы, в этом случае матрица инвариантна относительно поворотов, кратных 90° . Опыт обработки

изображений показывает, что отсутствие более строгой осевой симметрии ФРТ слабо сказывается на результатах. Иногда используют 8-угольные матрицы, инвариантные относительно поворотов на 45° .

Фильтрация согласно выражению (4.11) осуществляется перемещением слева направо (или сверху вниз) маски на один пиксель. При каждом положении апертуры производятся упомянутые выше операции, а именно перемножение весовых множителей a_{kl} с соответствующими значениями яркостей исходного изображения и суммированием произведений. Полученное значение присваивается центральному (i,j) -му пикселю. Обычно это значение делится на заранее заданное число K (нормирующий множитель). Маска содержит нечетное число строк и столбцов N , чтобы центральный элемент определялся однозначно.

Рассмотрим некоторые фильтры, сглаживающие шум. Пусть маска размером 3×3 имеет вид:

$$H_1 = 1/10 \begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix}.$$

Тогда яркость (i,j) -го пикселя после фильтрации определится как

$$g_{ij} = \frac{1}{10} (f_{i-1,j-1} + f_{i-1,j} + f_{i-1,j+1} + f_{i,j-1} + 2f_{i,j} + f_{i,j+1} + f_{i+1,j-1} + f_{i+1,j} + f_{i+1,j+1}).$$

Хотя коэффициенты a_{kl} можно выбрать из среднеквадратического или иного условия близости, не искаженного шумом s_{ij} и преобразованного g_{ij} изображений, обычно их задают эвристически. Приведем еще некоторые матрицы шумоподавляющих фильтров:

$$H_2 = 1/9 \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}, \quad H_3 = 1/16 \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}, \quad H_4 = 1/14 \begin{vmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{vmatrix}.$$

У фильтров $H_1 - H_4$ нормирующие множители K подобраны таким образом, чтобы не происходило изменения средней яркости обработанного изображения. Наряду с масками 3×3 используются маски большей размерности, например, 5×5 , 7×7 и т.п. В отличие от

фильтра H_2 , у фильтров H_1, H_3, H_4 весовые коэффициенты на пересечении главных диагоналей матрицы больше, чем коэффициенты, стоящие на периферии. Фильтры H_1, H_3, H_4 дают более плавное изменение яркости по изображению, чем H_2 .

Пусть отсчеты полезного изображения $f_{k,m}$ мало меняются в пределах маски. На изображение накладывается аддитивный шум: $f_{k,m} + n_{k,m}$, отсчеты шума $n_{k,m}$ случайны и независимы (или слабо зависимы) со статистической точки зрения. В этом случае механизм подавления шума с использованием приведенных фильтров состоит в том, при суммировании шумы компенсируют друг друга. Эта компенсация будет происходить тем успешнее, чем большее число членов в сумме, т.е. чем больше размер (апертура) маски. Пусть, например, используется маска $N \times N$, в пределах её полезное изображение имеет постоянную яркость f , шум с независимыми значениями отсчетов $n_{k,m}$, средним значением $\mu = 0$ и дисперсией σ^2 в пределах маски (такой шум называют *белым*). Отношение квадрата яркости (i,j) -го пикселя к дисперсии шума, т.е. отношение сигнал/шум, равно f^2 / σ^2 .

Рассмотрим, например, маску типа H_2 :

$$g_{ij} = 1/N^2 \sum_{k=1}^N \sum_{m=1}^N (f + n_{k,m}) = f + 1/N^2 \sum_{k=1}^N \sum_{m=1}^N n_{k,m}.$$

Средний квадрат яркости равен f^2 , средний квадрат интенсивности шума

$$\langle n_{i,j}^2 \rangle = 1/N^4 \sum_{k=1}^N \sum_{m=1}^N \langle n_{k,m}^2 \rangle = f + 1/N^4 \sum_{k=1}^N \sum_{m=1}^N \sum_{q=1}^N \sum_{p=1}^N \langle n_{k,m} n_{q,p} \rangle.$$

Двойная сумма отвечает $k = p, m = q$, эта сумма равна σ^2 / N^2 . Четырехкратная сумма равна нулю, так как отсчеты шума при $k \neq p, m \neq q$ независимы: $\langle n_{k,m} n_{q,p} \rangle = 0$. В результате фильтрации отношение сигнал/шум становится равным $N^2 f^2 / \sigma^2$, т.е. возрастает пропорционально площади маски. Отношение яркости (i,j) -го пикселя полезного изображения к среднеквадратическому отклонению шума возрастает пропорционально N . Применение маски 3×3 , в среднем, повышает отношение сигнала к шуму в 9 раз.

При импульсной помехе механизм подавления состоит в том, что импульс "расплывается" и становится мало заметным на общем фоне. Статистические свойства импульсных помех резко отличаются от статистических свойств изображений. Для изображений характерны плавные, небольшие изменения от элемента к элементу, а быстрые, скачкообразные изменения редки и образуют протяженные контуры. Импульсные же помехи представляют собой значительные по величине одиночные изолированные выбросы. Этим объясняется тот факт, что визуально очень легко отличить выбросы помехи от изображения, хотя такие помехи и оказывают сильное мешающее действие. Благодаря такому резкому отличию помех от изображений алгоритм фильтрации импульсных помех оказывается весьма простым. Однако часто в пределах апертуры значения полезного изображения все же изменяются заметным образом. Это бывает, в частности, когда в пределы маски попадают контуры. С физической точки зрения, все $H_1 - H_4$ являются *фильтрами нижних частот* (усредняющими фильтрами), подавляющими высокочастотные гармоники и шума, и полезного изображения. Это приводит не только к ослаблению шума, но и к размыванию контуров на изображении.

В частности, для подавления зернистого шума (на рисунке 4.13 его проявления отмечены кружками) применяется фильтр с усреднением значений или медианный фильтр. При использовании фильтра с усреднением значений, подобного приведенному на рисунке 4.14,а, для каждого элемента изображения $f_0(i, j)$ производится преобразование по формуле, аналогичной (4.12):

$$f_0(i, j) = \sum_{x=-1}^1 \sum_{y=-1}^1 (1/9) f_0(i+x, j+y).$$

Как видно из рисунка 4.14, а, при использовании такого фильтра зернистый шум сглаживается и становится незаметным. Вместе с тем применение фильтра с усреднением связано с таким нежелательным явлением, как размывание контуров изображения (рисунок 4.13) для части изображения, обведенной штриховой линией. Этот участок обведен штриховой линией и на рисунке 4.14, а.

Медианный фильтр позволяет заменить каждый элемент изображения центральным значением (медианой) группы из 3×3 элементов, в центре которой находится корректируемый элемент. Из рисунка 4.14, б видно, что контуры изображения почти сохранены, а зернистый шум сглаживается.

0	0	0	0	0	0
0	0	0	1	0	2
30	30	31	30	31	30
30	29	56	32	30	30
30	28	29	30	30	30
27	27	29	30	30	30

Рисунок 4.13 - Значение градаций серого отсчетов фрагмента изображения, содержащего импульсный шум

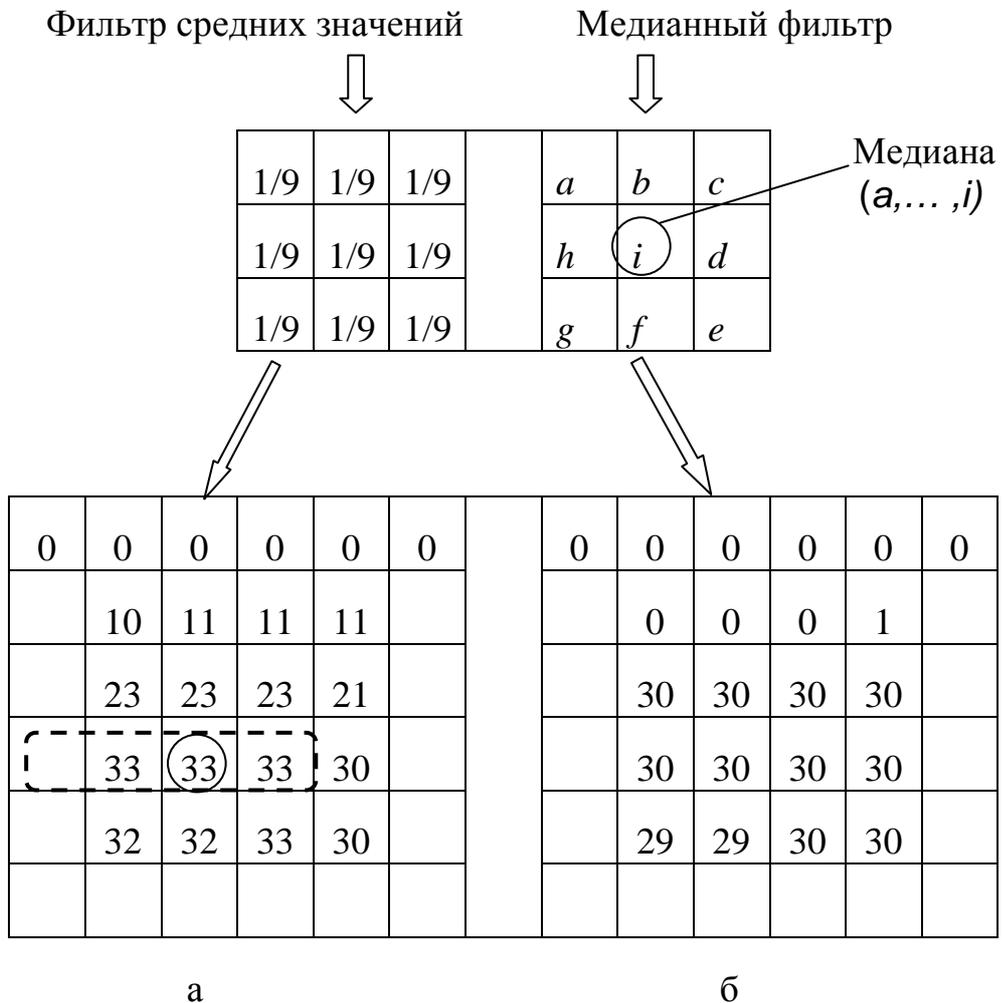


Рисунок 4.14 - Способы подавления зернистого шума; а — результат сглаживания с помощью фильтра усредненных значений; б — пример сглаживания с помощью медианного фильтра

Для устранения пятен на изображении эффективен фильтр подчеркивания контуров, подобный приведенному на рисунке 4.15. В этом фильтре преобразованные значения элементов изображения определяются по формуле

$$f_0(i, j) = 5f_0(i, j) - (f_0(i, j - 1) + f_0(i, j + 1) + f_0(i - 1, j) + f_0(i + 1, j)) .$$

При этом, как видно из примера, показанного на рисунке 4.15, изображение с пятном, размывающим контрастность изображения (а), преобразуется в изображение с селективно усиленным контрастом по контуру (б).

Контрастность слабая, изображение нечеткое

4	4	7	4	4	4
4	4	7	4	4	4
4	4	7	7	7	7
4	4	7	4	4	4
4	4	7	4	4	4
4	4	7	4	4	4

а

Фильтр подчеркивания контуров

0	-1	0
-1	5	-1
0	-1	0



Контрастность хорошая, изображение четкое

1	13	-2	1
1	10	13	13
1	13	-2	1
1	13	1	4

б

Рисунок 4.15 - Фильтрация с подчеркиванием контуров

4.2.3 Сжатие изображений

При цифровой обработке двумерных сигналов обычно имеют дело с большим количеством данных, которые в общем случае сильно коррелированы по строкам и столбцам. Каждый элемент изображения кодируется двоичным словом, содержащим от одного до 32 бит. Предположим, что данное изображение (или класс изображений) закодировано с помощью n бит на элемент. Тогда процедура сжатия может быть количественно оценена коэффициентом уменьшения бит m , который определяется следующим образом:

$$m = \frac{\text{число бит в коде исходного изображения}}{\text{число бит в коде преобразованного изображения}}.$$

Важнейшей мерой информационного содержания изображений является величина энтропии яркостных вариаций сигнала. Если изображение имеет K уровней яркости, а вероятность появления уровня k равна $P(k)$, то энтропия H_c (без учета корреляции уровней яркости) определяется как

$$H_c = -\sum_{k=0}^K P(k) \log_2(P(k)).$$

Информационная избыточность $r = b - H_c$, где b — наименьшее число бит, которыми можно представить уровни квантования сигнала. Величину информационной избыточности изображения непосредственно вычисляют в том случае, если имеется хорошая оценка энтропии, которой зачастую нет, так как соответствующие статистические свойства изображений неизвестны. Тем не менее, энтропию изображений можно оценить по гистограмме уровней яркости. Пусть $h(k)$ - частота появления уровня яркости k на исходном изображении, $0 < k < 2^b - 1$, и пусть размер изображения равен $M \times N$. Вероятность встречи на изображении уровня яркости k можно вычислить следующим образом:

$$\tilde{P}(k) = \frac{h(k)}{MN},$$

тогда оценка энтропии будет равна:

$$\tilde{H}_c = -\sum_{k=0}^{2^b-1} \tilde{P}(k) \log_2(\tilde{P}(k)).$$

Отсюда рассчитываем информационную избыточность $r = b - H_c$ и коэффициент сжатия $c : c = b / H_c$.

Отметим, что гистограмма дает неточную оценку энтропии вследствие корреляции уровней яркости. Более точная оценка может быть получена из гистограммы первых разностей уровней яркости.

Приведенные формулы могут быть использованы для получения теоретических пределов возможного сжатия изображений. Например, энтропия данных дистанционного зондирования со спутников находится в пределах от 4 до 5: $H_c \in [4, 5]$, здесь цифровой сигнал квантован на 256 уровней яркости и изображение представлено 8 бит на пиксель. Отсюда непосредственно вычисляются границы информационной избыточности: $r \in [3, 4]$ бит. Это, в свою очередь, приводит к тому, что соответствующие данные можно представлять, затрачивая в среднем от 4 до 5 бит на пиксель без потери информации, и коэффициент сжатия $c \in [1.6, 2]$.

Рассмотрим простейший способ реализации сжатия изображений, называемый зональным кодированием. Этот способ реализуется в три этапа.

1. Получают двумерное преобразование данного изображения путем его обработки блоками $(N \times N)$. Для преобразования могут быть использованы двумерные БПФ, преобразования Уолша-Адамара, Хартли, преобразования Корунена-Лоэва и т.д.
2. Из N^2 коэффициентов преобразования сохраняют N^2 / m коэффициентов, обладающих наибольшими двумерными дисперсиями. Все остальные коэффициенты приравниваются нулю.
3. Каждый из N^2 / m сохраненных коэффициентов кодируется с помощью b бит. Затем восстанавливается соответствующий $(N \times N)$ блока помощью обратного преобразования.

Так как кодируется только N^2 / m коэффициентов преобразования, а не N^2 исходных элементов изображения для каждого $(N \times N)$ блока, то среднее число битов на элемент восстановленного изображения равняется b / m .

В результате проведенных экспериментов [9] было определено, что сжатые с помощью дискретного косинусного преобразования изображения после восстановления в большей степени соответствуют

исходному изображению, чем сжатые с помощью преобразования Уолша-Адамара (ПУА). С другой стороны, ПУА вычисляется быстрее и проще при аппаратурной реализации. Следовательно, при заданном m выбор типа преобразования определяется объемом вычислений, объемом требуемой аппаратуры и требованиями, предъявляемыми к качеству изображения.

В настоящее время одним из наиболее эффективных методов сжатия изображений является вейвлет-преобразование (п. 2.7). Идея вейвлет-сжатия также напоминает идею сжатия, основанную на использовании преобразования Фурье. Дискретное преобразование Фурье ставит в соответствие набору из N значений функции, выражающей, например, зависимость некоторого параметра от пространственной координаты, набор из N коэффициентов Фурье - элементов частотного спектра этой функции. Если некоторые коэффициенты Фурье оказались равными 0, то оставшихся значений будет достаточно, чтобы полностью (без искажений) восстановить исходный набор. Таким образом, удастся представить исходную информацию меньшим числом значений. Дополнительно можно установить правила исключения ненулевых коэффициентов Фурье, например, исключать коэффициенты, модули которых окажутся меньше заданного числа. Разумеется, это приведет к искажению информации, но позволит увеличить степень сжатия. Понятно, что хорошую степень сжатия с малыми потерями информации можно получить лишь в случае, когда многие коэффициенты Фурье близки либо равны нулю. Реальные изображения имеют весьма сложную структуру, и их Фурье-образ может не удовлетворять подобным требованиям. Можно разбить изображения на области фиксированного размера и выполнять преобразования в каждой области отдельно. В каждой такой области изображение будет иметь меньше особенностей, чем все изображение в целом, и образы Фурье этих областей могут оказаться более подходящими для сжатия, чем образ всего изображения. Именно так и работает известный алгоритм сжатия JPEG. Недостатком такого подхода является то, что при высокой степени сжатия части единого изображения, обрабатываемые независимо друг от друга, могут плохо стыковаться - становится видно, что изображение собрано из разных кусков. Дискретное вейвлет-преобразование также переводит набор из N элементов в набор из N вейвлет-коэффициентов. Эти коэффициенты соответствуют не только амплитудам различных частот, но и различным пространственным участкам изображения. Таким образом, отпадает не-

обходимость в предварительном разбиении изображения на локальные области - эта возможность уже заложена в вейвлет-преобразовании, причем при этом не возникает никаких нежелательных побочных эффектов.

4.3 Алгоритмы нелинейной обработки речевых сигналов

Строго говоря, все физические системы, с которыми имеют дело в радиотехнике, в той или иной степени нелинейны. И в ряде случаев представление реальной системы упрощенной линейной моделью неоправданно, поэтому необходимо использование других методов и алгоритмов.

В этом случае используют обобщенный принцип суперпозиции, определяемый следующим образом. Если $x_1(n)$ и $x_2(n)$ - входные последовательности, а $y_1(n)$ и $y_2(n)$ - соответствующие выходные последовательности системы, т.е.

$$y_1(n) = \psi[x_1(n)],$$

$$y_2(n) = \psi[x_2(n)],$$

а c - произвольный скаляр, то система является гомоморфной, если справедливы соотношения:

$$\psi[x_1(n) \Delta x_2(n)] = \psi[x_1(n) * x_2(n)],$$

$$\psi[c \bullet x_1(n)] = c \circ y_1(n).$$

Здесь Δ , $*$, \bullet и \circ обозначены некоторые неконкретизированные математические операции типа умножения, сложения и свертки. Определение гомоморфизма звучит следующим образом. Отображение системы G в другую систему G' называется гомоморфизмом, если каждому элементу $a \in G$ соответствует определенный элемент $a' \in G'$, причем, если $c = a \Delta b$, то $c = a' * b'$ (Δ - операция, определенная в G , $*$ - операция, определенная в G'). Если такое отображение взаимно однозначно, оно называется изоморфизмом.

Важность рассматриваемого способа обработки определяется тем, что операция ψ , выполняемая гомоморфной системой, может быть реализована в виде последовательности операций (рисунок 4.16). Системы A_0 и A_0^{-1} взаимно обратны. Система L - является линейной

системой с постоянными параметрами, например, цифровым фильтром.

Таким образом, используя разложение, показанное на рисунке 4.16, можно обрабатывать последовательность на выходе системы A_0 стандартными методами. Вид систем A_0 и A_0^{-1} определяется из операции ψ .

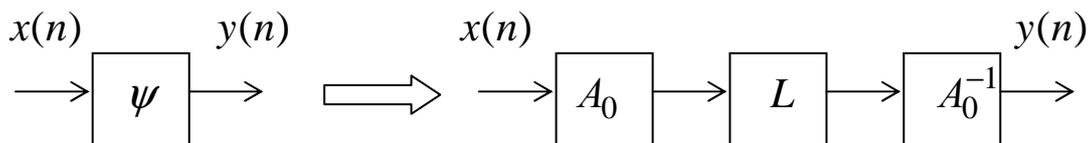


Рисунок 4.16 - Система гомоморфной обработки

Обоснование вида применяемой операции ψ можно осуществить на основе анализа схемы механизма образования речи. На рисунке 4.17 приведен один из наиболее распространенных вариантов схемы механизма образования речи по Фланагану /6/. При разговоре грудная клетка расширяется и сжимается, прокачивая воздух из легких по трахее через голосовую щель. Если голосовые связки напряжены, как при образовании звонких звуков, типа гласных, то они вибрируют и модулируют поток воздуха, превращая его в короткие импульсы (порции). Если голосовые связки расслаблены, воздух свободно проходит через голосовую щель, не подвергаясь модуляции. Воздушный поток проходит через глоточную полость мимо основания языка и, в зависимости от положения мягкого нёба, через ротовую или носовую полости. Поток воздуха выходит наружу через рот или нос (или обеими путями) и воспринимается как речь. В случае глухих звуков, таких как «с» в слове «снег» или «п» в слове «пик», голосовые связки расслаблены. При этом возможны два режима: либо образуется турбулентный поток, когда воздух проходит через сужения в голосовом тракте (как при образовании звука «с»), либо возникает короткий взрывной процесс, вызванный повышенным давлением воздуха за точкой перекрытия голосового тракта (звук «п»). При изменении положения артикуляторов (губ, языка, челюсти, мягкого нёба) во время произнесения непрерывной речи форма отдельных полостей (а следовательно, и резонансная частота) меняется.

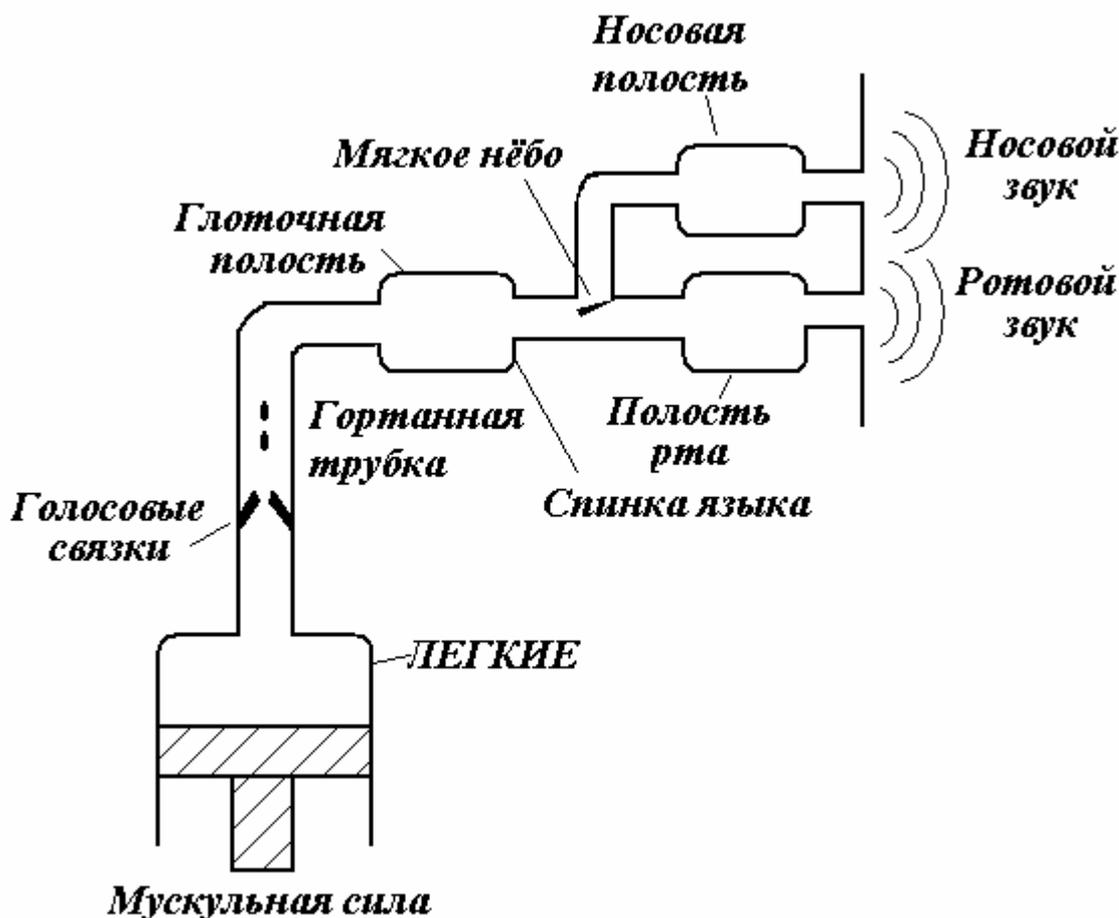


Рисунок 4.17 – Схема механизма образования речи

Голосовой тракт представляет собой акустическую трубу, простирающуюся от голосовой щели до губ. У взрослого мужчины она имеет длину около 17 см и частота первого (четвертьволнового) резонанса:

$$F_1 = \frac{1}{4} \frac{c}{l} = \frac{3 \cdot 10^8}{0,17} \approx 500 \text{ Гц.}$$

Площадь поперечного сечения акустической трубы не одинакова и в зависимости от положения артикуляторов составляет от 0 до 20см². Голосовой тракт имеет некоторые устойчивые виды резонансных колебаний, называемых формантами. На рисунке 4.18 приведен спектр некоторых звуков.

Таким образом, речевой сигнал для разных звуков является сочетанием для звонких звуков низкочастотных колебаний, для глухих звуков - турбулентных квазислучайных шумов или скачков давления.

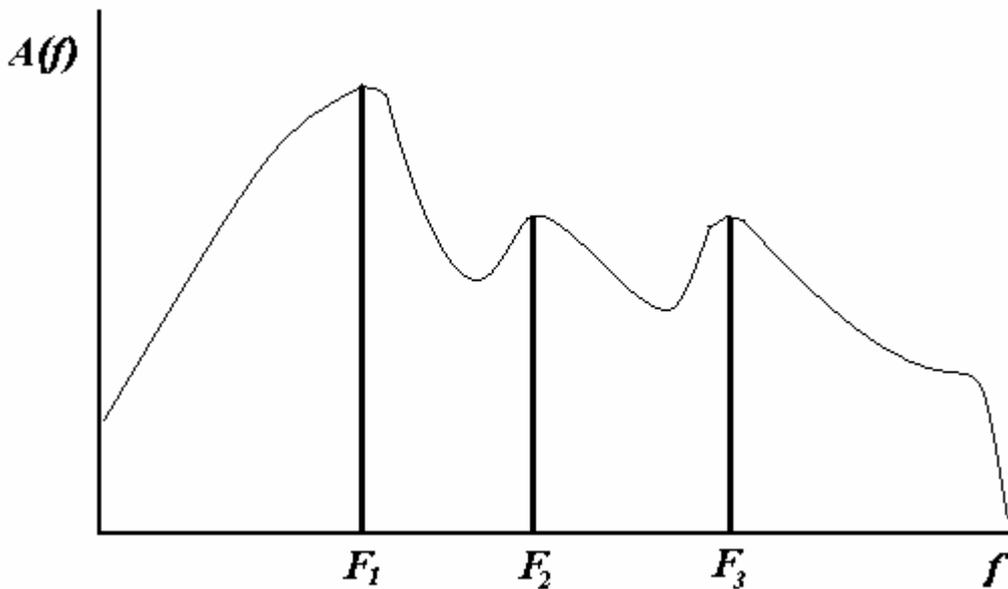


Рисунок 4.18 – Вариант спектра одного из звуков

Речевой сигнал обычно моделируется путем свертки трех компонент: последовательности импульсов, представляющих основной тон, импульса возбуждения глухих звуков и импульсной характеристики голосового тракта (рисунок 4.19).

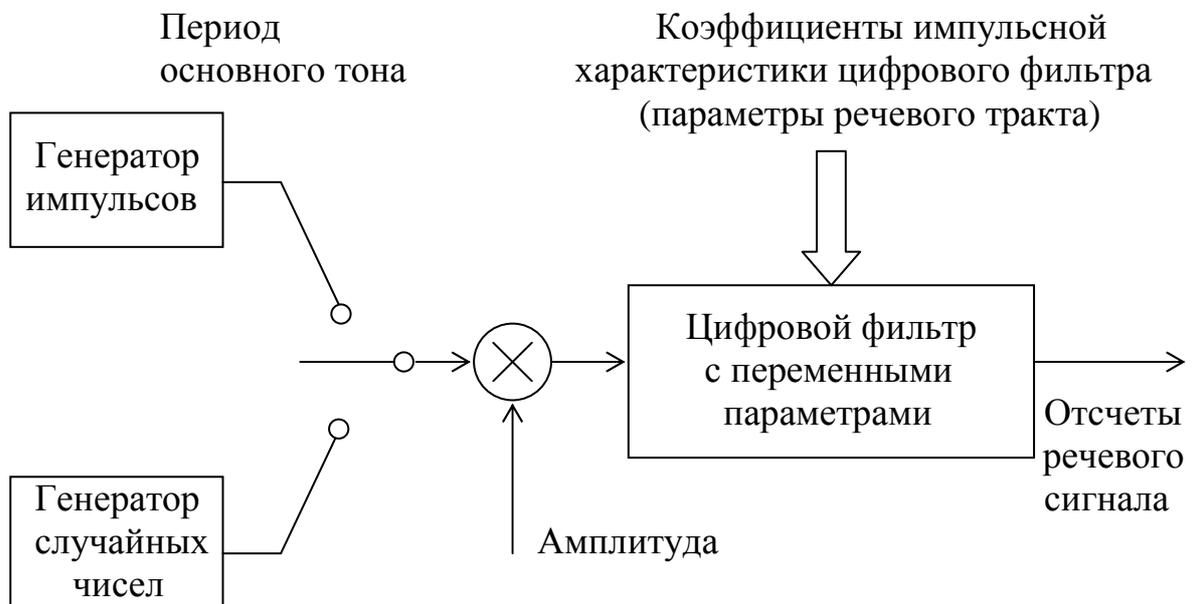


Рисунок 4.19 - Цифровая модель образования речи

Источниками возбуждения служат генератор импульсов с периодом основного тона, а также генератор случайных чисел. Генератор импульсов через каждые N_0 отсчетов вырабатывает импульс, соответствующий очередной порции воздуха. Интервал между импульсами называется периодом основного тона. Он равен величине, обратной частоте следования порций воздуха или частоте колебаний голосовых связок. Импульсы основного тона участвуют в формировании звонких звуков. Выходная последовательность генератора случайных чисел имитирует и квазислучайный турбулентный поток, и спад давления при образовании глухих звуков.

Каждый из источников (или оба) может быть соединен с входом линейного цифрового фильтра с переменными параметрами, моделирующими голосовой тракт. При этом коэффициенты фильтра (отсчеты импульсной характеристики) отражают свойства голосового тракта в зависимости от времени при непрерывной речи.

Регулировка усиления, введенная между источниками и фильтром, позволяет управлять громкостью выходного сигнала. Последовательность на выходе фильтра эквивалентна речевому сигналу, дискретизированному с соответствующей частотой.

Речевой сигнал является нестационарным, так как спектр речи изменяется во времени. Поэтому спектральный анализ речи с использованием ЛПП-систем затруднен. В этом случае целесообразно использовать гомоморфную фильтрацию.

Обозначим через $p(n)$ последовательность импульсов основного тона, через $e(n)$ - импульс возбуждения, через $u(n)$ - импульсную характеристику голосового тракта и, наконец, через $w(n)$ - весовую функцию («окно») конечной длительности, служащую для выделения участков речевого сигнала. Тогда

$$x(n) = [p(n) * e(n) * u(n)]w(n) \quad . \quad (4.13)$$

Поскольку $w(n)$ обычно меняется медленно, то соотношение (4.13) можно приближенно записать как

$$\begin{aligned} x(n) &\approx [p(n) * w(n) * e(n) * u(n)] \quad , \\ x(n) &\approx \hat{p}(n) * e(n) * u(n) \quad , \end{aligned} \quad (4.14)$$

т.е. выразить $x(n)$ в виде свертки трех последовательностей. Свертка легко приводится к суммированию, если воспользоваться преобразованием Фурье от (4.14) (что дает произведение) и прологарифмировать результат. Полученный сигнал можно пропустить через ЛПП-систему и обработать (фильтровать) каждую составляющую $x(n)$ некоторым желаемым образом. Для восстановления обработанного сигнала обратная система A_0^{-1} должна выполнять потенцирование (т.е. операцию, обратную логарифмированию) и обратное преобразование Фурье. Таким образом, структурная схема гомоморфной системы обработки речи будет иметь вид, представленный на рисунке 4.20.

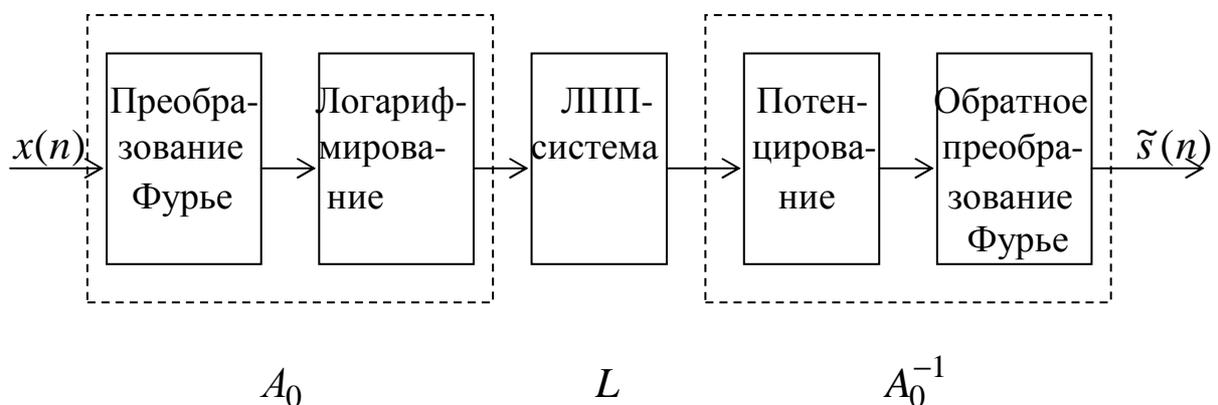


Рисунок 4.20 - Система гомоморфной обработки речи

Особенностью цифровых методов обработки является практически идентичная аппаратная часть устройства обработки для реализации задач различного предназначения. Функциональные особенности отражаются программным обеспечением. Однако существует не единственный возможный вариант технической реализации алгоритмов цифровой обработки, что определяет необходимость рассмотрения этого вопроса в следующей главе.

5 ТЕХНИЧЕСКИЕ СРЕДСТВА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

5.1 Общие принципы технической реализации алгоритмов цифровой обработки

Цифровая обработка сигналов широко используется в различных системах авиационного радиоэлектронного оборудования (РЭО) военного назначения. Но наибольшее развитие цифровые методы обработки нашли в радиолокационных системах ввиду сложности и важности решаемых ими задач.

Радиолокационные системы в виде бортовых РЛС (БРЛС) и РГС входят в состав РЭО практически всех самолетов, вертолетов и ракет. Они представляют собой основное средство для получения информации о наземных, надводных, подводных и воздушных объектах (целях).

Для современных летательных аппаратов военного назначения характерно включение бортовых РЛС в состав обзорно-прицельного или прицельно-навигационного комплексов, применяемых на различных этапах выполнения боевой задачи.

Объединение в единый комплекс информационных подсистем и датчиков, функционально связанных через систему цифровой обработки (БЦВМ), существенно расширяет боевые возможности комплекса.

Обобщенная структурная схема бортовой радиолокационной системы показана на рисунке 5.1 /28/. В состав типовой радиолокационной системы входят: антенное устройство с элементами управления диаграммой направленности и антенным переключателем; синхронизатор; задающий генератор и передатчик; аналоговый приемник; аналогово-цифровой преобразователь (АЦП); БЦВМ, включающая процессор обработки сигналов (ПОС) и процессор обработки данных (ПОД); система индикации и управления РЛС, как правило, сопряжена с инерциальной системой (ИНС) и системой управления вооружением (СУВ).

Антенное устройство обеспечивает формирование основной диаграммы направленности требуемой формы и ее перемещение в пространстве по заданному закону, а также диаграммы направленности компенсационного канала. Синхронизатор определяет (задает)

временной режим работы РЛС в целом и его отдельных составных частей (передатчика, приемника и др.). Антенный переключатель служит для подключения антенной системы к передатчику во время излучения зондирующего сигнала и к приемнику по окончании излучения. Аналоговый приемник осуществляет предварительное усиление на сверхвысокой частоте (СВЧ), преобразование частоты до промежуточной, нормировку сигналов, синхронное детектирование сигналов, формирование и усиление синфазной и квадратурной составляющих. АЦП выполняет преобразование сигналов в цифровую форму. Задающий генератор формирует непрерывные высокочастотные сигналы стабильной частоты малой мощности для последующего их усиления и импульсной модуляции в передатчике, а также сигналы линейной частотной и фазовой модуляции. Передатчик обеспечивает формирование мощных радиолокационных зондирующих сигналов с ВЧП, СЧП и НЧП импульсов, а также генерирования непрерывных (прерывистых) высокочастотных сигналов подсвета РГС УР класса «воздух-воздух».

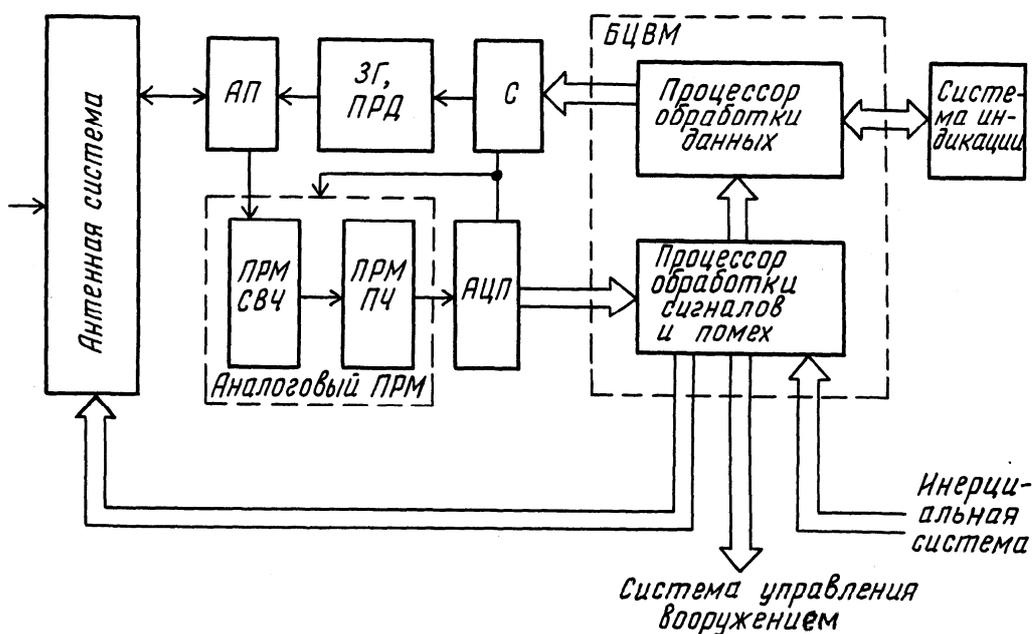


Рисунок 5.1

Процессор обработки радиолокационных сигналов (ПОС) обеспечивает решение задач первичной обработки сигналов, в частности, узкополосной доплеровской фильтрации и сжатия сигналов, пороговой их обработки и амплитудного детектирования, определения первичных отсчетов дальности до цели, скорости сближения с ней и др.

Процессор обработки данных (ПОД) осуществляет управление режимами РЛС, формой и положением диаграммы направленности антенной системы, реализует алгоритмы оценивания (фильтрации) координат и параметров движения целей в различных режимах работы, формирует данные для системы индикации и т.д. Принципиальных различий в технической реализации процессоров обработки данных и сигналов нет. Основное отличие заключается в их программном обеспечении.

При создании современной радиоэлектронной аппаратуры используются три основных подхода к технической реализации устройства цифровой обработки (УЦО): аппаратный, программный и программно-аппаратный. При аппаратном подходе получают УЦО с традиционной «жесткой» логикой, что обеспечивает наибольшее быстродействие устройств, но требует трудоемкой разработки индивидуальной структуры устройства (например, цифровой фильтр на дискретных элементах, см. п.4.2.1). При программном подходе УЦО реализуется в виде программы для универсальной цифровой ЭВМ. Программно-аппаратный подход предполагает разработку как программных, так и аппаратных средств и основан на применении встраиваемых микро-ЭВМ. Аппаратная часть такого УЦО содержит программируемое устройство обработки на основе микропроцессорной интегральной схемы, а также интегральные микросхемы, обеспечивающие организацию информационного обмена процессора и внешних устройств. Программная часть представлена реализованными на одном из языков программирования алгоритмами ЦОС. Этот вариант открывает широкие возможности для применения современных больших интегральных схем (БИС) микропроцессорных устройств и позволяет в наибольшей степени согласовать разрабатываемые программно-аппаратные средства с особенностями решаемых задач.

Современные микропроцессорные устройства развиваются по трем архитектурным направлениям: универсальные микропроцессоры, микроконтроллеры и цифровые сигнальные процессоры.

Микропроцессор (МП) представляет собой функционально законченное цифровое устройство, выполненное в виде одной или нескольких БИС и предназначенное для выполнения операций по обработке информации и управлению процессом обработки в соответствии с хранимой в памяти программой [29]. Универсальный микропроцессор, прежде всего, разрабатывается для выполнения разнообразных задач в составе цифровой ЭВМ или микропроцессорного вы-

числительного устройства (МПВУ). В узком смысле МП совпадает с центральным процессорным элементом (ЦПЭ) вычислительного устройства, выполненным на основе БИС. ЦПЭ обычно используется в качестве основного элемента микропроцессорного вычислительного устройства МПВУ, схема которого представлена на рисунке 5.2.

МПВУ минимальной конфигурации содержит ЦПЭ, блоки постоянного запоминающего устройства (ПЗУ) и оперативного запоминающего устройства (ОЗУ), генератор тактовых импульсов (ГТИ) и блок интерфейса (ИФ), через который осуществляется связь с элементами радиоэлектронного комплекса, являющимися для ЦПЭ внешними устройствами (ВУ). Будем считать, что МПВУ, представляющее собой специализированное вычислительное устройство, используется в аппаратуре для выполнения некоторого заданного алгоритма обработки информации (или совокупности алгоритмов). Поэтому основная программа работы МПВУ записывается в ПЗУ, которое служит также для хранения различных подпрограмм, констант, таблиц и других данных, известных уже на этапе проектирования устройства. ОЗУ используется для хранения данных, поступивших из ВУ или подготовленных для выдачи в ВУ, а также промежуточных результатов вычислений и некоторой адресной информации. Блок ГТИ, выполняемый, как правило, на основе кварцевого генератора, предназначен для выработки серий тактовых импульсов и некоторых вспомогательных сигналов, необходимых для работы ЦПЭ и синхронизации других блоков системы.

Интерфейс представляет собой совокупность шин для передачи информации, электронных схем, специальных сигналов и алгоритмов, управляющих обменом информацией. Блок интерфейса служит для сопряжения сигналов МПВУ и ВУ по временным и электрическим параметрам, а также в необходимых случаях для преобразования данных и управления обменом.

К основным узлам ЦПЭ относятся: управляющее устройство (УУ) с регистром команд (РК) и дешифратором команд (ДШК); арифметико-логическое устройство (АЛУ) с аккумулятором (А), который является основным рабочим регистром; блок регистров общего назначения (РОН) со счетчиком команд (СК).

Связь между блоками МПВУ осуществляется с помощью ряда шин: чаще всего это шины адреса (ША), шины данных (ШД), шины управления (ШУ). В некоторых случаях для увеличения быстродействия системы могут ввести отдельную шину команд (ШК).

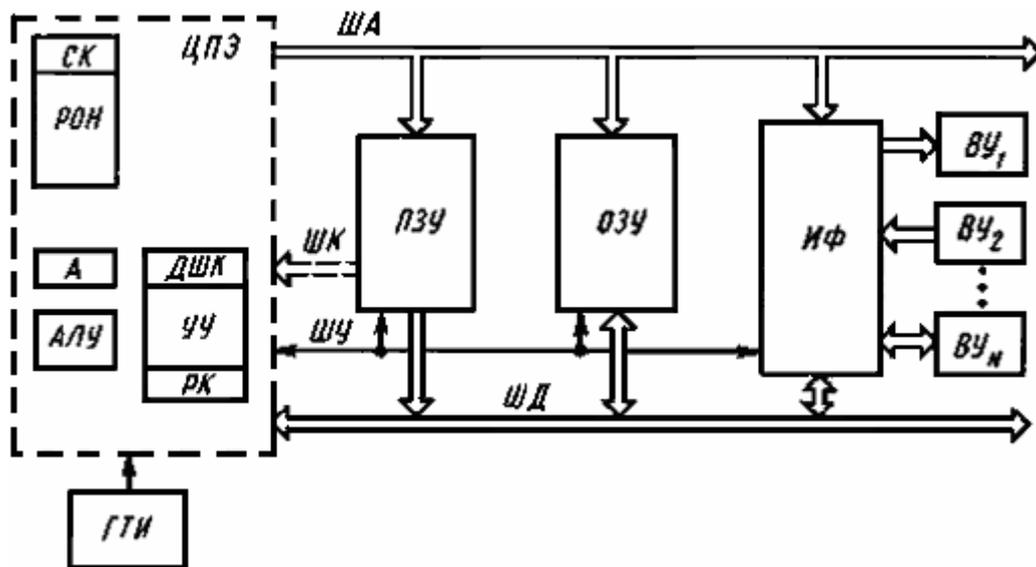


Рисунок 5.2 - Обобщенная структурная схема микропроцессорного вычислительного устройства

Выбор типа МП зависит от общей элементной базы аппаратуры, в которую входит разрабатываемое МПВУ, требуемого быстродействия, допустимого объема оборудования, личного опыта разработчика и т. д. При этом следует иметь в виду, что процесс проектирования МПВУ, как правило, является итеративным. Это означает, что при выполнении каждого шага проектирования возможен возврат назад для корректировки принятых решений, причем может потребоваться смена типа МП или даже внесение изменений в исходный алгоритм работы устройства.

При использовании универсального МП структура МПВУ практически не зависит от конкретных задач, решаемых устройством. Алгоритмы цифровой обработки являются частью программного обеспечения МПВУ и представляют собой прямую реализацию рассмотренных в предыдущих главах математических выражений.

Недостатком использования универсальных МП для реализации МПВУ является функциональная избыточность оборудования, относительно большие массогабаритные характеристики и наличие большого числа элементов, что, в свою очередь, снижает надежность системы в целом.

Частично данные недостатки устраняются при использовании в качестве ЦПЭ микроконтроллеров и цифровых сигнальных процессоров.

5.2 Применение однокристалльных микроконтроллеров для циф-

ровой обработки сигналов

Современный уровень развития микроэлектроники позволил создать БИС, на кристалле которой размещены не только АЛУ, УУ и РОН, как в однокристалльных МП, но и ОЗУ, ПЗУ, порты ввода-вывода и различные интерфейсные устройства вплоть до АЦП. По общей архитектуре такие БИС больше напоминают законченную ЭВМ. Однако незначительная емкость памяти, расположенная на кристалле, упрощенная и ориентированная на выдачу сигналов управления система команд предопределяет использование их как управляющих устройств (контроллеров) для автоматизации различного радиоэлектронного оборудования. Принято называть их однокристалльными микроконтроллерами (ОМК) /30,31/. Применение ОКМ позволяет существенно уменьшить число БИС, используемых в МПВУ, повысить надежность, снизить их стоимость и энергопотребление. Эти качества ОКМ позволяют легко их встраивать в различные бортовые радиоэлектронные устройства.

При разработке ОКМ в качестве основной задачи рассматривается задача управления некоторым объектом в реальном масштабе времени.

Отечественная микроэлектронная промышленность выпускает целый ряд ОКМ, не только отличающихся разрядностью и производительностью, но и ориентированных на конкретные типы технологического оборудования. В таблице 5.1 приведены основные типы восьмиразрядных ОКМ, выпускаемых отечественной промышленностью.

Таблица 5.1

Тип МК	ПЗУ команд (Кбайт)	EEPROM данных	ОЗУ данных	Тактовая частота (МГц)
КМ1816ВЕ48	1	-	64	6
КМ1816ВЕ49	2	-	128	11
КМ1816ВЕ51	4	-	128	12
КР1878ВЕ1	2	64	128	8

ОМК серии 1816 изготовлены по n-МОП-технологии и имеют похожую архитектуру. Они отличаются быстродействием, объемом ОЗУ, а также наличием или отсутствием на кристалле ПЗУ, ее типом и объемом. Наличие на кристалле портов ввода-вывода, тактового генератора, системы прерываний и таймера при высокой тактовой час-

тоте и восьмиразрядном АЛУ обеспечивает универсальность их использования для создания микропроцессорных систем различного назначения.

При сравнительной оценке характеристик ОМК обращают внимание на наличие следующих устройств и их параметры:

- Flash ROM - объем энергонезависимой памяти программ (в килобайтах);
- EEPROM - объем энергонезависимой памяти данных (в байтах);
- RAM - объем статической памяти данных (в байтах);
- External RAM - возможность подключения к микроконтроллеру дополнительной микросхемы внешней статической памяти данных (в килобайтах);
- ISP - возможность программирования микроконтроллера в системе (на целевой плате) при основном напряжении питания;
- SPM - функция самопрограммирования Flash ROM памяти микроконтроллера в системе без участия внешнего программатора;
- JTAG - встроенный JTAG - интерфейс;
- I/O (pins) - максимальное количество доступных линий ввода / вывода;
- Timer(s) 8/16 bit - количество и разрядность таймеров/счетчиков;
- USI - универсальный коммуникационный интерфейс;
- AC - аналоговый компаратор;
- ADC (channels) - количество каналов аналого-цифрового преобразования;
- Internal RC - наличие внутренней RC-цепочки для автономной работы микроконтроллера (без внешнего источника опорной частоты);
- WDT - сторожевой таймер;
- BDC - аппаратный программируемый блок защиты от сбоев при внезапном (в том числе и кратковременном) пропадании напряжения питания микроконтроллера;
- UART - асинхронный последовательный приемопередатчик;
- SPI - синхронный трехпроводной последовательный интерфейс;
- I²C - двухпроводной последовательный интерфейс;
- RTC - система реального времени;
- PWM (channels) - количество независимых каналов широтно-импульсной модуляции;
- Command Set - количество различных инструкций в системе команд микроконтроллера;

- Vcc - диапазон рабочих напряжений питания (в вольтах);
- Clock - диапазон рабочих частот (в мегагерцах);
- Packages - типы корпусов, в которые опрессовывается микроконтроллер, и общее количество выводов.

Наиболее производительным из ОМК 1816 является ОМК КМ1816ВЕ51, имеющий на кристалле значительную постоянную память объемом 4 Кб и возможностью расширения до 64 Кб. Его максимальная тактовая частота равна 12 МГц, а быстродействие составляет 1 млн. коротких операций в секунду. Существуют три модификации ОМК: ОМК 51 с масочным ПЗУ объемом 4 Кб рассчитан на крупносерийное оборудование; ОМК с ультрафиолетовым РПЗУ объемом 4 Кб ориентирован на системы, требующие периодической перенастройки; ОМК без встроенного ПЗУ имеет возможность подключения меньшего ПЗУ.

ОМК 51 является развитием ОМК 48, но отличается системой команд. Однако на уровне программ на языке Ассемблера оба ОМК программно совместимы «снизу вверх».

Наиболее современной отечественной разработкой является микроконтроллерное RISC-ядро ТЕСЕЙ, производимое ОАО «АНГСТРЕМ», на базе которого уже создано несколько микроконтроллеров, как универсальных, так и специализированного применения.

Характерной особенностью ядра ТЕСЕЙ являются:

- гарвардская RISC-архитектура, позволяющая выполнять любую из 52 команд 16-разрядного формата за два такта частоты процессора;
- единая система команд для всего семейства с возможностью адресации до двух операндов, находящихся в памяти;
- 4-ступенчатый конвейер выполнения команд;
- малое время отклика на прерывание и сохранение контекста;
- широкий диапазон конфигураций внутренней памяти команд, памяти данных и периферийных устройств.

Микроконтроллеры отличаются наличием энергонезависимой памяти данных, возможностью многократного перепрограммирования памяти программ, небольшим количеством внешних выводов и низким энергопотреблением.

Микроконтроллер КР1878ВЕ1 является представителем семейства ТЕСЕЙ и обладает следующими характеристиками:

- производительность до 4 MIPS на тактовой частоте 8 МГц;
- перепрограммируемая память программ 1К x 16;
- память данных RAM 128 x 8, EEPROM данных 64 x 8;
- периферия: сторожевой таймер, 16-бит таймер-счётчик с предделителем, 12 линий ввода/вывода и поддержка прерываний;
- ток потребления меньше 2 мА при напряжении питания 5В и тактовой частоте 5 МГц.

Микроконтроллер выпускается в 18-выводном исполнении.

Новые зарубежные разработки ОМК по своим функциональным возможностям и быстродействию ушли далеко вперед. Основные фирмы производители и номенклатура производимых микроконтроллеров представлены в таблице 5.2

Таблица 5.2 - Фирмы-производители 8-, 16- и 32-разрядных микроконтроллерных семейств

Фирма – производитель	Семейства МК					
	8-разрядные		16-разрядные		32-разрядные	
	Наименование	Число типов МК	Наименование	Число типов МК	Наименование	Число типов МК
Intel	MCS-48	10	MCS-96	10	i960xx	12
	MCS-51	60	MCS-196	30	i386xx	6
	MCS-151	2	MCS-296	1	i486xx	6
	MCS-251	4	i186xx	7		
Motorola	MC68HC05	200	MC68HC12	2	MC68EC0x0	8
	MC68HC08	22	MC68HC16	10	MC683xx	19
	MC68HC11	43			MCF5xxx	5
					MPC500	2
				MPC800	6	
Microchip	PIC12Cxxx	10				
	PIC16C5x	16				
	PIC16Cxxx	60				
	PIC17Cxxx	9				
	PIC18Cxxx	10				
	PIC14000	1				
Atmel	AT89	11			AT91	10
	AT90S (AVR)	28				
	AT94Kxx	3				

Продолжение таблицы 5.2

Philips	8051	100	80C51XA	9		
---------	------	-----	---------	---	--	--

Dallas Semiconductor	DS5000 DS8xCxxx	3 8				
Zilog	Z8/SAM8x Z8Plus	60 4				
Texas Instruments	TMS-370 TMS7000 TUSBxxxx	60 9 4	MSP430	20		
Hitachi	H8/300 H8/300L	17 47	H8/300H H8S/2xxx	20 45	SuperH	30
Mitsubishi	M740 M7450 M38000	600	MELPS770 0 M16C	100 200	M32Rx/D M32R/E M32000	1 2 2
NEC	78K0/0S	200	78K3/4	80		
Panasonic	MN10100	35	MN10200	40		
Samsung	KS86/88	40				
Holtek	HT48/49xxx	13				
AMD			Am186xx	7	ElanSCxxx	5
Analog Devices	AD μ C8xx	5				
Triscend	SoC-E5	5			SoC-A7	4
STMicroelectronics (бывшая SGS-Thomson)	ST6 ST7 ST9	25 50 10	ST10	10	ST100	
Fairchild Semiconductor	ACE	2				
Ubicom (Scenix)	SX	7				
Infineon (Siemens)	C500	17	C166	25	Tricore	3
Xemics	XE8000 CoolRISC	3				

Современные ОМК— это размещенная на одном кристалле сложная цифровая система, в состав которой входят 8-, 16- или 32-разрядный процессор, внутренняя память программ (до нескольких десятков килобайт) с возможностью расширения, перепрограммируемая электрическая энергонезависимая память (FLASH-память), широкий набор интерфейсных и периферийных устройств, порты ввода-вывода (параллельные и последовательные), таймеры, многоканальные аналого-

цифровые преобразователи, модули ШИМ и др. Обладая малым количеством коротких команд, выполняемых за один цикл, и используя конвейерный способ их обработки, ОМК обеспечивают производительность в несколько миллионов операций в секунду.

5.2.1 RISC-архитектура современных микроконтроллеров

Одно из основных условий при выборе микроконтроллера для управленческих задач является его способность работать в реальном масштабе времени. Под реальным масштабом понимается такая скорость обработки информации, при которой не происходит существенного старения информации за время между моментом её прихода и получением результата. Для реализации реального масштаба времени требуется максимальное быстродействие процессорного ядра. Достижение требуемого быстродействия требует новых архитектурных решений, одним из которых является RISC – архитектура, активно развиваемая фирмой Atmel. Это новое семейство высокопроизводительных 8-разрядных RISC (Reduced Instruction Set Computers) микроконтроллеров общего назначения, объединенных общей маркой **AVR**. Аббревиатура RISC означает усеченный перечень инструкций системы команд, причем в идеальном случае на выполнение каждой команды уходит время, не превышающее периода следования тактовых импульсов.

В основу концепции **AVR**-микроконтроллеров легли следующие принципы:

1. Использование новейшей, наиболее скоростной и экономичной КМОП технологии фирмы Atmel в сочетании с RISC архитектурой для разработки и производства быстрых 8-разрядных микроконтроллеров, сравнимых с 16-разрядными микропроцессорами и микроконтроллерами по производительности и превосходящих микросхемы стандартной КМОП логики по скорости. Ожидаемая производительность - до 20 MIPS на частоте 20 МГц. Время выполнения короткой команды на такой тактовой частоте составляет 50 нс;

2. Разработка архитектуры и системы команд **AVR** в теснейшем согласии с принципами языка Си так, чтобы аппаратная часть нового микроконтроллера и его система команд были неотъемлемыми частями одного целого и использовались с максимальным коэффициентом полезного действия.

3. Функциональное расширение микроконтроллера возможностью программирования в системе путем объединения Flash-технологии фирмы Atmel со стандартным скоростным последовательным интерфейсом (SPI). Это позволяет многократно модифицировать программу не только с помощью обычного программатора, но и непосредственно в системе, в конечном устройстве пользователя. При этом не требуется вводить никаких дополнительных аппаратных узлов и вспомогательных источников питания извне как через SPI интерфейс, так и с помощью обычного программатора. Число циклов перезаписи - не менее 100000.

4. Расширенный состав периферийных устройств, таких как:

- таймер/счетчик, разрядность 8 бит;
- таймер/счетчик, разрядность 16 бит с возможностью организации функций широтноимпульсной модуляции (ШИМ) и захвата/сравнения;
- аналоговый компаратор;
- скоростной последовательный интерфейс SPI;
- встроенная система сброса микроконтроллера;
- асинхронный дуплексный последовательный порт UART;
- контроллер прерываний;
- внутренний тактовый генератор;
- сторожевой (WATCHDOG) таймер;
- многоканальный аналого-цифровой преобразователь.

Внутренний тактовый генератор может запускаться от внешнего источника опорной частоты, от внешнего кварцевого резонатора или от внутренней RC цепи. Поскольку все AVR полностью статические, минимальная допустимая частота ничем не ограничена (вплоть до пошагового режима). Максимальная рабочая частота определяется конкретным типом микроконтроллера. Ограничения верхней границы частотного диапазона связаны с технологическими проблемами при производстве микросхем и будут постепенно изменяться в сторону более высоких значений.

Сторожевой таймер предназначен для защиты микроконтроллера от сбоев в процессе работы. Он имеет свой собственный RC-генератор, работающий на частоте 1 МГц. Как и для основного внутреннего RC-генератора, значение 1 МГц является приближенным и зависит, прежде всего, от величины напряжения питания микроконтроллера и от температуры.

Порты ввода/вывода **AVR** имеют число независимых линий "Вход/Выход" от 5 до 32. Каждый разряд любого порта может быть запрограммирован на ввод или на вывод информации. Мощные выходные драйверы обеспечивают типовую токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА. Все значения приведены для напряжения питания 5 В.

AVR работают в широком диапазоне питающих напряжений от 2,7 В до 6,0 В. Ток потребления в активном режиме зависит от величины напряжения питания и частоты, на которой работает микроконтроллер, и составляет менее 1 мА для 500 кГц, 5 ... 6 мА для 5 МГц и 8 ... 9 мА для частоты 12 МГц. **AVR** также могут быть переведены программным путем в один из двух режимов пониженного энергопотребления. Первый - режим холостого хода (IDLE), когда прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний и сторожевой таймер продолжают функционировать. Ток потребления здесь не превышает 2,5 мА на частоте 12 МГц. Второй - режим микропотребления (SLEEP), когда сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов. Выход из режима SLEEP возможен либо по сигналу сброса, либо от внешнего источника прерывания. При включенном сторожевом таймере ток потребления в этом режиме составляет около 80 мкА, а при выключенном - менее 1 мкА. (Все вышеприведенные значения справедливы для напряжения питания 5 В).

Температурные диапазоны работы микроконтроллеров **AVR** - коммерческий (0 ... 70 С) и промышленный (-40 ... +85 С).

С точки зрения программиста **AVR** представляет собой 8-разрядный RISC микроконтроллер, имеющий быстрый гарвардский процессор, память программ, память данных, порты ввода/вывода и интерфейсные схемы. Структурная схема микроконтроллера приведена на рисунке 5.3.

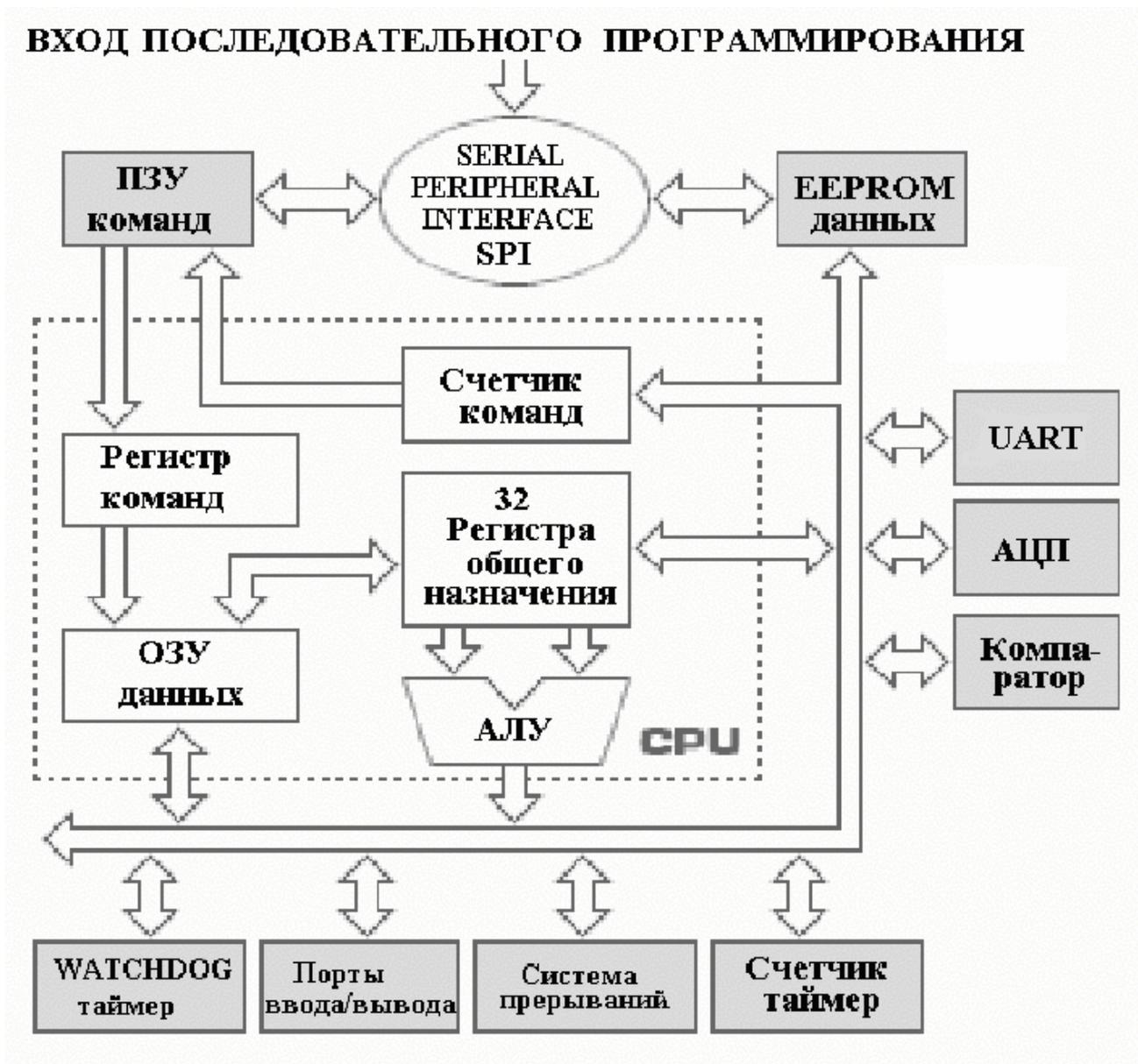


Рисунок 5.3 - Структурная схема AVR

Гарвардская архитектура *AVR* реализует полное логическое и физическое разделение не только адресных пространств, но и информационных шин для обращения к памяти программ и к памяти данных. Способы адресации и доступа к ним также различны. Такое построение уже ближе к структуре скоростных цифровых сигнальных процессоров и обеспечивает существенное повышение производительности за счет:

- одновременной работы центрального процессора, как с памятью программ, так и с памятью данных;

- расширения до 16 бит разрядной сетки шины данных памяти программ.

Каждый из 32-х регистров общего назначения длиной 1 байт непосредственно соединен с арифметико-логическим устройством (ALU) процессора. Это означает, что в *AVR* существует 32 регистра-аккумулятора. Это позволяет в сочетании с конвейерной обработкой выполнять одну операцию в ALU за один машинный цикл. Например, два операнда извлекаются из регистрового файла, выполняется команда и результат записывается обратно в регистровый файл в течение только одного машинного цикла.

Шесть из 32-х регистров файла могут использоваться как три 16-разрядных указателя адреса при косвенной адресации данных. Один из этих указателей применяется также для доступа к таблицам перекодировок, записанных в памяти программ микроконтроллера. Использование трех 16-битных указателей (X, Y и Z Pointers) существенно повышает скорость пересылки данных при работе прикладной программы.

Регистровый файл занимает младшие 32 байта в общем адресном пространстве SRAM *AVR*. Такое архитектурное решение позволяет получать доступ к быстрой "регистровой" оперативной памяти микроконтроллера двумя путями - непосредственной адресацией в коде команды к любой ячейке и обычными (пятью) способами адресации ячеек SRAM. В технической документации фирмы Atmel это свойство носит название "быстрое контекстное переключение" и является еще одной отличительной особенностью архитектуры *AVR*, повышающей эффективность работы микроконтроллера.

Во время переходов к выполнению процедур обработки прерываний или подпрограмм текущее состояние программного счетчика сохраняется в стеке. Обычно стек формируется программно и располагается в общем адресном пространстве оперативной памяти данных. Размер стека лимитируется только общим объемом SRAM и степенью ее использования прикладной программой. 16-разрядный указатель стека находится в общем адресном пространстве оперативной памяти и доступен для чтения и записи.

Система команд *AVR* весьма развита и насчитывает до 120 различных инструкций. Почти все команды имеют фиксированную длину в одно слово (16 бит), что позволяет в большинстве случаев объединять в одной команде и код операции, и операнд(ы). Различают пять групп команд *AVR*: условного ветвления, безусловного ветвления, арифметические и логические операции, команды пересылки

данных, команды работы с битами. По разнообразию и количеству реализованных инструкций *AVR* больше похожи на CISC (Complex Instruction Set Computer – компьютер со сложной системой команд), чем на RISC процессоры. Например, у PIC-контроллеров система команд насчитывает от 33 до 58 различных инструкций, а у MCS51 она составляет 111.

В целом архитектура *AVR* в сочетании с регистровым файлом и расширенной системой команд позволяет в короткие сроки создавать программы с очень эффективным кодом, как по скорости его выполнения, так и по компактности.

Еще одним достоинством семейства *AVR* является наличие средств поддержки разработок для *AVR*. Программные и аппаратные средства для новой платформы разрабатывались параллельно с самими микроконтроллерами и включают в себя компиляторы, внутрисхемные эмуляторы, отладчики, программаторы, простейшие отладочные платы-конструкторы практически на любой вкус. Более детальное знакомство с микроконтроллерами как отечественных, так и ведущих мировых производителей можно выполнить самостоятельно по имеющейся в большом количестве справочной литературе, например /30..32/.

5.3 Программируемые процессоры цифровой обработки сигналов

5.3.1 Основные операции обработки сигналов в цифровых системах и их влияние на структуру сигнального процессора

Среди микропроцессорных интегральных схем, используемых для задач цифровой обработки, наибольшее распространение нашли цифровые сигнальные процессоры (Digital Signal Processor - DSP) /30,33..36/.

Архитектура DSP определяется несколькими базовыми операциями, которые используются в алгоритмах ЦОС.

Для выделения таких операций проведем функциональный анализ основных направлений ЦОС, к которым относятся цифровая фильтрация и спектральный анализ.

Цифровая фильтрация. В области цифровой фильтрации разработчик систем ЦОС имеет дело с реализацией КИХ- и БИХ-фильтров (с конечной и бесконечной импульсными характеристиками

соответственно). Оба класса фильтров относятся к классу линейных систем с постоянными параметрами, в которых входная и выходная последовательности связаны отношениями типа свертки.

В КИХ-фильтре отсчет выходного сигнала определяется только значениями входного сигнала, а в БИХ-фильтре - значениями входного и выходного сигналов. Это хорошо видно из линейных разностных уравнений с постоянными коэффициентами, которыми описывается данный класс дискретных систем.

Таким образом, для построения систем цифровой фильтрации требуется эффективная реализация соотношения типа дискретной свертки, которая раскладывается на операции умножения и накапливающего суммирования, а также операции задержки.

Спектральный анализ. В области спектрального (или гармонического анализа) используются прямое и обратное дискретное преобразование Фурье, а также рациональный способ реализации дискретного преобразования Фурье - быстрое преобразование Фурье.

Спектральный анализ основан на известных методах представления данной функции при помощи других функций, которые называются базовыми и свойства которых считаются известными.

Для получения выходной последовательности в естественном порядке необходимо выполнить двоичную инверсию отчетов входной последовательности. Перестановка входных элементов состоит в образовании двоичных номеров выходной последовательности путем добавления единицы к старшему разряду с распространением переноса в сторону младших разрядов (вправо). Такая адресация получила название бит-реверсивной.

Вычисление коэффициента $W_N^k = \cos\left(\frac{2\pi k}{N}\right) - j \sin\left(\frac{2\pi k}{N}\right)$ можно

осуществлять следующим образом:

- используя подпрограммы или таблицы синуса и косинуса;
- прямым табличным способом (выборкой готовых значений из таблицы);
- используя рекуррентную формулу $W_N^k = (W_N^{k-1})W_N^1$ при $W_N^0 = 1$;
- таблично-алгоритмическим способом, так как на последующих этапах коэффициенты повторяются.

При использовании алгоритма БПФ с прореживанием по частоте требуется перестановка элементов выходной последовательности, а

базовая операция «бабочка» сводится к вычислению выражений:

$$x = A + BW_N^k,$$

$$y = A - BW_N^k.$$

Для получения амплитуд и фаз составляющих спектра (гармоник) необходимо также вычислить следующие выражения:

$$\frac{\sqrt{X_{\text{Re}}^2 + X_{\text{Im}}^2}}{N}; \quad \text{arctg} \frac{X_{\text{Im}}}{X_{\text{Re}}},$$

где $X_{\text{Re}}, X_{\text{Im}}$ - вещественная и мнимая части комплексных коэффициентов.

В гомоморфной обработке сигналов дополнительно требуется вычисление функций $\log_2 x$ и 2^x .

5.3.2 Обобщенная архитектура DSP

На рисунке 5.5 приведен операционный базис алгоритмов ЦОС. Поставив в соответствие каждой операции блок, можно получить обобщенную архитектуру DSP (рисунок 5.6). Очевидно, что реализация выделенных операций может быть различной. Предельное быстроедействие достигается при максимальной аппаратной поддержке операций, что приводит к значительным аппаратным затратам.

Это обстоятельство, а также желание обеспечить преемственность в решениях и программную совместимость приводят, в конечном счете, к аппаратной реализации не в целом той или иной операции, а отдельных ее составляющих. Причем в разных семействах (16-, 24-, 32-разрядных) архитектура отражает специфику задач ЦОС, на которые она ориентирована

Основными операциями являются: задержка, АЦП и ЦАП (функции кодека), умножение с накоплением - операция МАС (multiply and accumulate), умножение, сложение, вычисление значений функций $\sin, \cos, \text{arctg}, \log_2 x, 2^x$, табличное вычисление коэффициентов W_N^k . К особенностям следует также отнести необходимость использования бит-реверсивной адресации.

Обобщенная архитектура процессора DSP образуется тремя составляющими: процессорным ядром, подсистемами ввода/вывода и хранения. Структура процессорного ядра отражает выделенный опе-

рационный базис и учитывает особенности алгоритмов ЦОС. Блок генерации адреса обеспечивает необходимые способы адресации, в том числе бит-реверсивную. Синтезатор тактовой частоты (PLL) дает возможность регулировать производительность и потребляемую мощность. Блок выполнения операций реализует операции с фиксированной точкой и операции с плавающей точкой для приложений, где требуются высокая точность и производительность. Очевидно, что поддержка операций с плавающей точкой может и отсутствовать (используется обычно в 32-разрядных DSP).

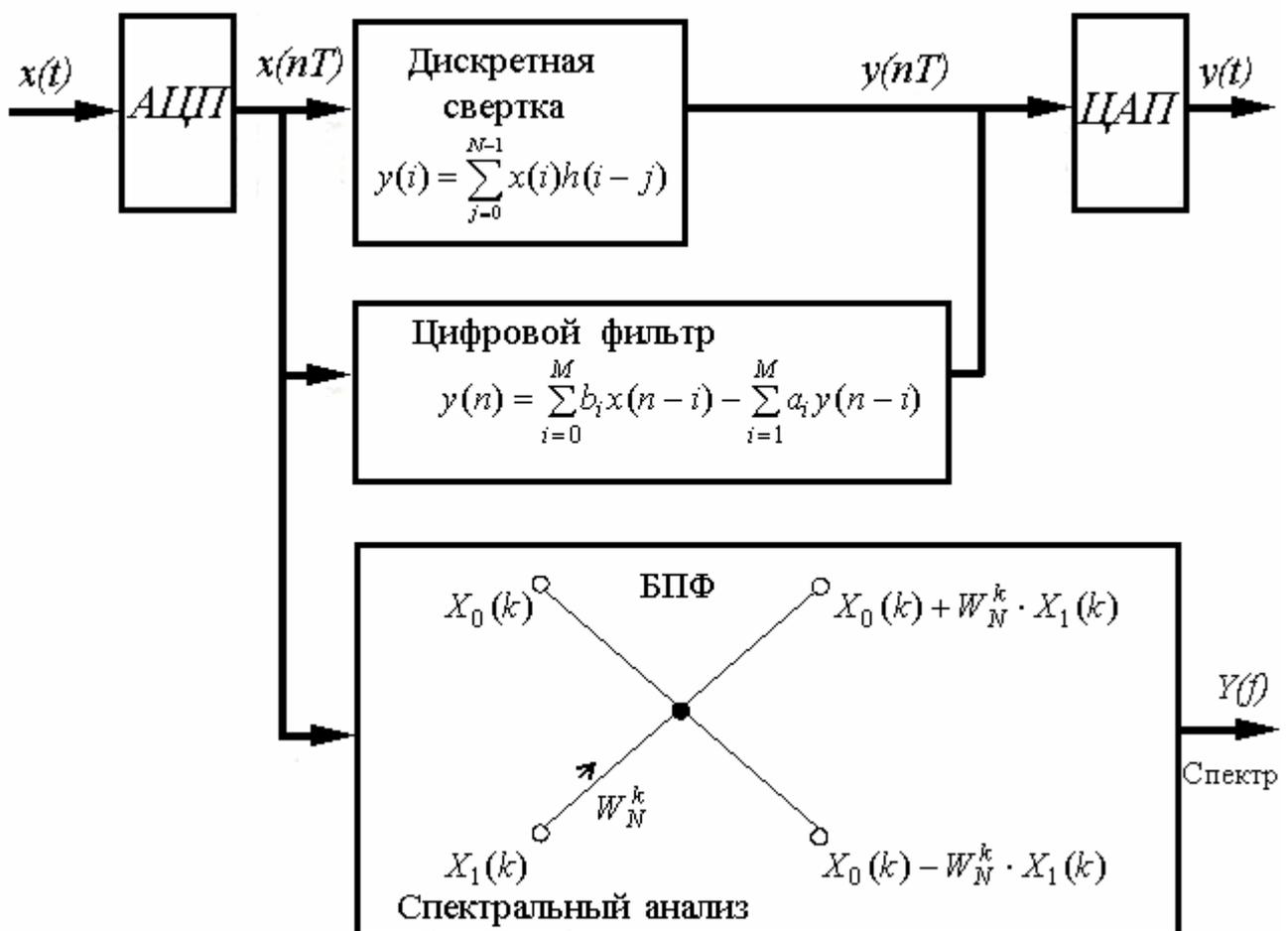


Рисунок 5.5 - Операционный базис процессоров DSP

В процессорное ядро введен блок отладки и контроля, который содержит средства внутрисхемной эмуляции (OnCE), средства тестирования и отладки (JTAG), сторожевой таймер (WDT). Данный набор

средств сегодня практически является стандартным.

Подсистема ввода/вывода включает кодек (АЦП и ЦАП), параллельные и последовательные порты ввода/вывода, host-интерфейс для связи с персональным компьютером или другой ЭВМ, аудио-интерфейс, широтно-импульсный модулятор, таймеры и другие устройства.

Организация подсистемы хранения также во многом отражает специфику алгоритмов ЦОС. Так, в частности, выделение двух блоков памяти X и Y позволяет одновременно считывать операнды для свертки, а выделение памяти программ позволяет распараллелить выполнение этапов командного цикла (чтение команды и формирование адреса - чтение операндов - выполнение - запись результата). Такая модель используется в архитектурах гарвардского типа.

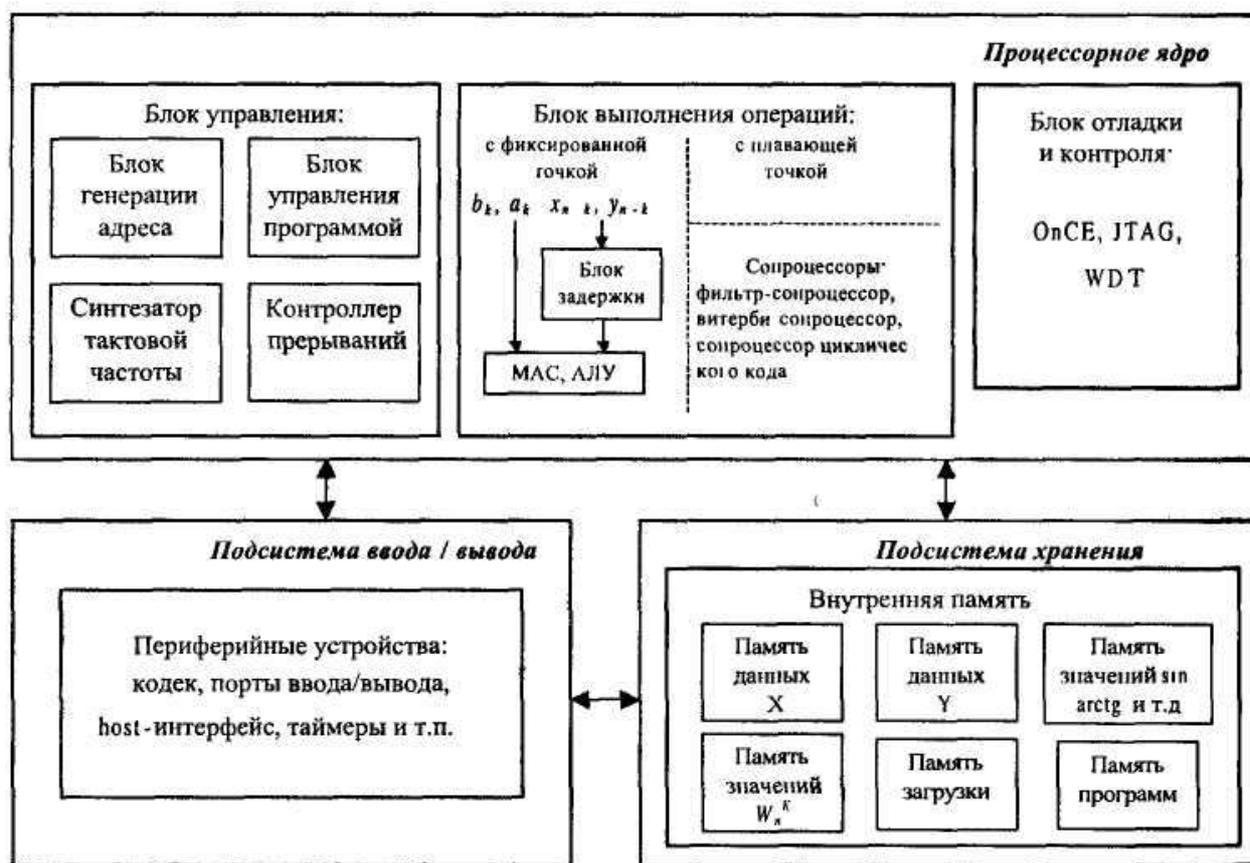


Рисунок 5.6 - Обобщенная архитектура DSP

Необходимые сложные операции (\sin , \arctg , $\sqrt{\quad}$ и др.) целесообразно выполнять табличным способом.

В случае использования A- и μ -нелинейных зависимостей, для представления отсчета требуется в данном случае 8 бит: 1 бит - знак, 3 бита - номер участка A- или μ -закона, 4 бита - номер шага на вы-

бранном участке. Данные нелинейные характеристики позволяют устранить избыточность данных за счет компандирования (COMpressor/ exPANDER), которое заключается в сужении динамического диапазона при аналого-цифровом преобразовании и расширение при цифроаналоговом преобразовании. Очевидно, что скорость выполнения алгоритма БПФ существенно возрастет, если хранить таблицы значений W_N^k . Наличие памяти загрузки позволяет упростить процесс занесения кода программы в процессор. Конкретную реализацию обобщенной архитектуры, рассмотрим на примере семейства DSP 56000, которое по организации процессорного ядра и составу периферии является базовым /29,30/.

5.4 Процессоры семейства DSP56000

5.4.1. Обзор архитектуры и шинной организации DSP56000

Высокая производительность процессоров DSP позволяет использовать их в области коммуникаций, высокоскоростного управления, обработки данных, компьютерных и аудио приложениях. Цифровая обработка сигналов эффективна в любой области электроники. А возможности DSP обеспечивают реализацию цифровой обработки для любой аналоговой электронной схемы.

Типичными задачами цифровой обработки сигналов, для решения которых успешно используется DSP, являются обработка одномерных и двумерных сигналов большой размерности, например, речь и изображение, в таких областях человеческой деятельности, как:

- теле- и радиокommunikации (голосовая почта, телеконференции, закрытая связь);
- компьютерное оборудование (матричные процессоры, рабочие станции, графические ускорители);
- медицинская электроника;
- цифровое видео;
- радары и сонары (навигация, океанография, поиск);
- сейсмография.

Операция умножения с накоплением (MAC) является базовой операцией алгоритмов типа свертки, широко используемых в задачах цифровой обработки. Структура DSP содержит необходимые блоки, обеспечивающие эффективную реализацию операции MAC. Два операнда непосредственно участвуют в операции умножения, и резуль-

тат операции затем суммируются с ранее полученными. Этот процесс происходит внутри DSP56000 с использованием двух отдельных модулей памяти за один цикл. Благодаря наличию двух модулей памяти и независимого суммирующего умножителя, можно объединить две пересылки, умножение и сложение в одну операцию.

Основу процессора составляют три параллельно работающих устройства арифметико-логическое устройство (ALU), устройство генерации адреса (AGU) и программируемый контроллер (PC).

DSP имеет периферийные устройства на кристалле в стиле микроконтроллеров: программируемую память, память данных и порты расширения памяти. Программная модель и система команд ориентированы на разработку эффективных и компактных программ.

Основные качественные и количественные характеристики семейства DSP56000:

- скорость - до 30 и более миллионов операций в секунду;
- точность представления данных - 24-разрядные данные, обеспечивающие обработку в динамическом диапазоне 144 децибел, промежуточные результаты хранятся в 56-разрядном аккумуляторе, что обеспечивает диапазон в 336 децибел;
- параллелизм - каждое из операционных устройств на кристалле, память, периферийные операции независимы и работают параллельно благодаря развитой системе шин;
- интеграция - в дополнение к трем независимым операционным устройствам DSP имеет шесть видов памяти на кристалле, три периферийных устройства (последовательный коммуникационный интерфейс (SCI), синхронный последовательный интерфейс (SSI), host-интерфейс, генератор тактовой частоты и семь шин (три адресных и четыре шины данных), при этом система является компактной и достаточно дешевой, а также обладает низким энергопотреблением;
- конвейер команд — трехступенчатый конвейер инструкций прозрачен для программиста;
- система команд - 62 инструкции, мнемоники которых совпадают с инструкциями микроконтроллеров и упрощают трансляцию программ для DSP. Дополнительные инструкции DSP служат для управления параллельными операционными устройствами;
- совместимость программного кода для всех представителей семейства;
- низкое энергопотребление - за счет использования КМОП-технологии, а также следующих дополнительных возможностей

управления энергопотреблением: инструкция WAIT; инструкция STOP, останавливающая внутренний генератор тактовой частоты; уменьшение потребления энергии за счет уменьшения частоты.

Приведенные характеристики отражают специализацию архитектуры DSP для приложений цифровой обработки сигналов.

Структура DSP56002, который является базовым для семейства, представлена на рисунке 5.7.

Основные компоненты DSP:

- шины данных;
- шины адреса;
- арифметико-логическое устройство данных;
- устройство генерации адреса;
- память данных X;
- память данных Y;
- контроллер программ;
- память программ;
- устройства ввода/вывода: расширение памяти (порт A); порты ввода/вывода (B и C); host-интерфейс; последовательные интерфейсы; таймеры.

Шины данных. DSP организован на регистрах центрального процессора, составляющих три независимых операционных устройства. Передача данных на кристалле осуществляется по четырем двунаправленным 24-разрядным шинам: шине данных X (XDB), шине данных Y (YDB), шине данных программ (PDB), глобальной шине данных (GDB). Шины данных X и Y могут объединяться в 48-разрядную шину для передачи инструкций. Шины XDB и YDB сделаны локальными для увеличения скорости и уменьшения потребления мощности. Все другие передачи данных происходят по шине GDB. Структура шин поддерживает передачу данных типов «регистр - регистр», «регистр - память», «память - регистр». Размер передаваемых данных изменяется от 24- до 56-битных слов в одном командном цикле. Переходы между шинами осуществляются с помощью специального переключателя.

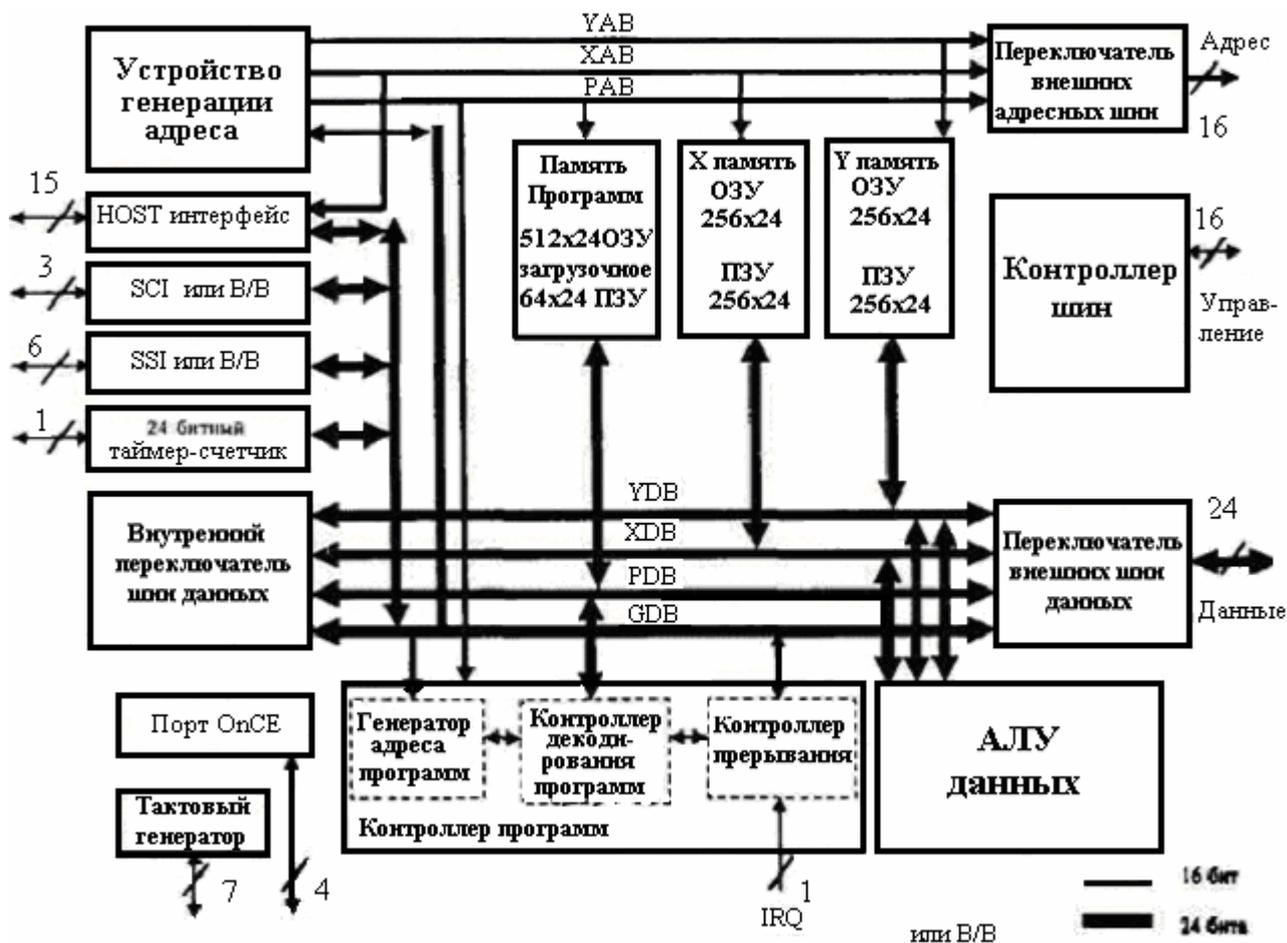


Рисунок 5.7 - Структура DSP56002

Шины адреса. Адреса внутренней памяти данных X и Y передаются по двум однонаправленным 16-битным шинам - адресной шине X (XAB) и адресной шине Y (YAB). Адреса памяти программ передаются по двунаправленной шине адресов программ (PAB).

Пространство внешней памяти адресуется с помощью однонаправленной 16-битной шины с тремя состояниями выходов, которую можно переключить на XAB, YAB или PAB. В одном командном цикле может быть произведен только один доступ к внешней памяти. Переход между шинами осуществляется с помощью переключателя внутренних шин. Он представляет собой программируемую матрицу ключей, позволяющую соединять две любые внутренние шины без задержек.

В блоке переключателя шин расположено устройство манипуляций с битами за счет которого переключатель обеспечивает доступ к любому адресному пространству. Устройство манипуляций с битами выполняет битовые операции (установку, сброс, сравнение и т.п.)

в памяти, адресных регистрах, регистрах управления и регистрах данных через шины XDB, YDB и GDB.

Арифметико-логическое устройство данных. АЛУ данных (рисунок 5.8) разработано для улучшения возможности обработки сигналов широкого динамического диапазона. Специальные схемы обеспечивают простой контроль ошибок округления и переполнения обрабатываемых данных. АЛУ содержит четыре 24-битных входных регистра, два 48-битных аккумулятора, два 8-битных регистра расширения аккумулятора, аккумулятор сдвига, две шины данных и параллельное неконвейеризованное устройство умножения с аккумулятированием (МАС). Операции АЛУ используют арифметику дополнения до двух, в которой для представления отрицательных чисел используется дополнительный машинный код. В регистры АЛУ могут быть записаны 24- или 48-разрядные операнды. АЛУ позволяет выполнять в одном командном цикле инструкции умножения, сложения, вычитания, итерационного деления, нормализации, сдвигов, логических операций. Регистры АЛУ данных могут быть прочитаны или записаны 24-битными или 48-битными операндами по шинам XDB, YDB. Операнды источника могут иметь длину 24, 48 и 56 бит и всегда находятся в регистрах АЛУ данных. Результат любой операции в АЛУ данных сохраняется в аккумуляторе. Формат чисел с фиксированной запятой в виде правильных дробей с представлением отрицательных чисел в дополнительном коде.

24-битные данные обеспечивают динамический диапазон в 144дБ. Такого диапазона достаточно для большинства приложений, в которых используются параметры разрядностью не больше 24 бит. 56-битный аккумулятор АЛУ данных обеспечивает 336 дБ внутреннего динамического диапазона.

АЛУ данных имеет следующие компоненты:

- четыре 24-битных входных регистра;
- параллельный умножитель - аккумулятор (МАС);
- два 48-битных регистра аккумулятора;
- два 8-битных регистра расширения аккумулятора;
- аккумулятор сдвига;
- два устройства сдвига/ограничения по шине данных.

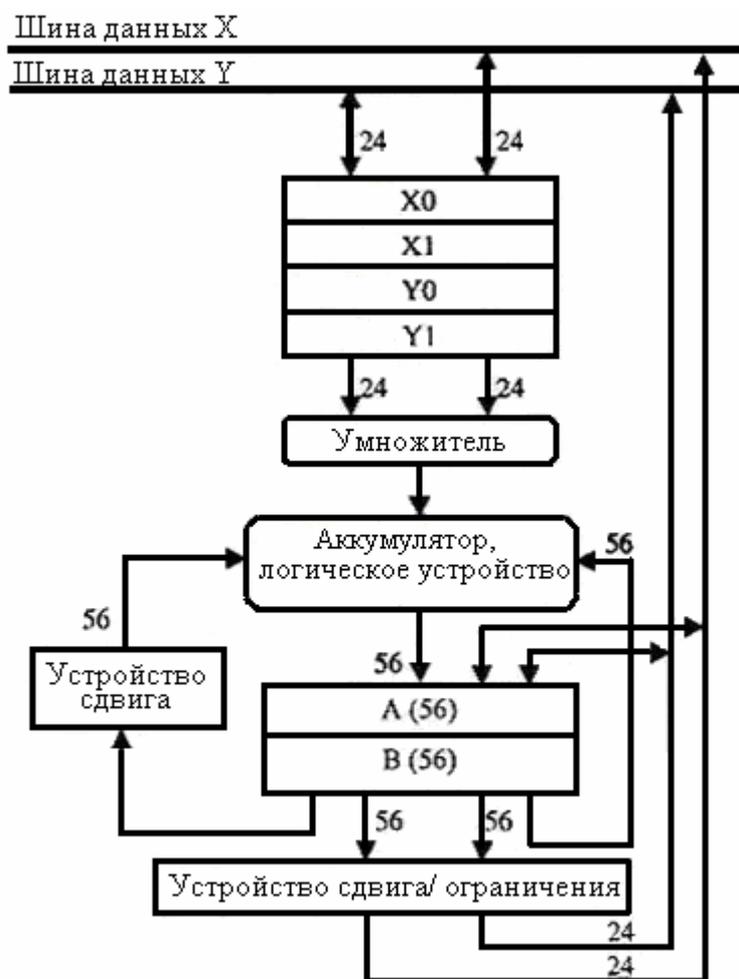


Рисунок 5.8 - АЛУ данных

Входные регистры АЛУ данных (X1, X0, Y1, Y0). Это 24-битные регистры данных общего назначения. Они могут использоваться как независимые 24-битные регистры или как два 48-битных регистра X и Y, образованные объединением X1 :X0 и Y1 :Y0 соответственно. X1 - старшее слово X, Y1 -старшее слово Y. Регистры служат входными буферами между XDB или YDB и устройством MAC. Они используются как операнды источника АЛУ данных. Содержимое регистров может быть выдано на соответствующую шину данных для процедур обработки прерываний.

Аккумуляторы АЛУ данных. Шесть регистров АЛУ данных (A2, A1, A0, B2, B1 и B0) формируют два 56-битных аккумулятора общего назначения. Каждый из аккумуляторов содержит по три объединенных регистра (A2:A1:A0 и B2:B1:B0). 24-битная MSP хранится в A1 или B1, 24-битная LSP - в A0 или B0. 8-битное расширение - в A2 или B2.

8-битные регистры расширения предназначены для защиты от переполнения. В DSP56000/DSP56001 диапазон значений операндов

составляет от -1 до +0,9999998. Если сумма двух чисел меньше -1 или больше +0,9999998, то возникает потеря значимого разряда или переполнение. 8-битный регистр расширения формирует результат в случае 8 переносов.

При использовании регистра расширения аккумулятора бит переполнения в регистре кодов условий всегда установлен в единицу. Расширение знака обеспечивается автоматически при записи в 56-битный аккумулятор А или В 48-битного или 24-битного операнда. Если записывается 24-битный операнд, ISP автоматически заполняется нулями для представления операнда в 56-битной форме,

Если аккумуляторы А и В прочитаны, можно с помощью операций масштабировать их содержимое на один бит влево или вправо для блока арифметики с плавающей точкой.

MAC и логическое устройство. MAC и логическое устройство входят в главное арифметическое устройство DSP и выполняют все вычисления с операндами данных. В случае арифметической команды устройство обрабатывает до трех операндов и выдает 56-битный результат в следующем формате: расширение: старшая значащая часть: младшая значащая часть (EXT: MSP: LSP).

Операция MAC выполняется независимо и параллельно с операциями на шинах данных, что облегчает буферизацию входных и выходных данных АЛУ. Арифметическое устройство MAC содержит умножитель и два аккумулятора. На вход умножителя могут поступать данные только из регистров X и Y. Умножитель выполняет умножение 24 x 24 с дополнением до двух. 48-битный результат выравнивается и суммируется с 56-битным содержимым аккумулятора А или В. 56-битная сумма сохраняется в аккумуляторе (рисунок 5.9).

8-битное расширение аккумулятора фиксирует переполнение до 256 и позволяет складывать или вычитать содержимое 56-битных аккумуляторов. Операция MAC (умножение/ сложение) - неконвейеризованная одноцикловая операция. Если инструкция предусматривает умножение без сложения, то устройство MAC очищает аккумулятор и складывает его содержимое с произведением. Таким образом, результат всех арифметических инструкций представляется в формате EXT:MSP:LSP (A2:A1:A0 или B2:B1.B0). Если результат сохраняется в 24-битном операнде, LSP просто отсекается или округляется до MSP.

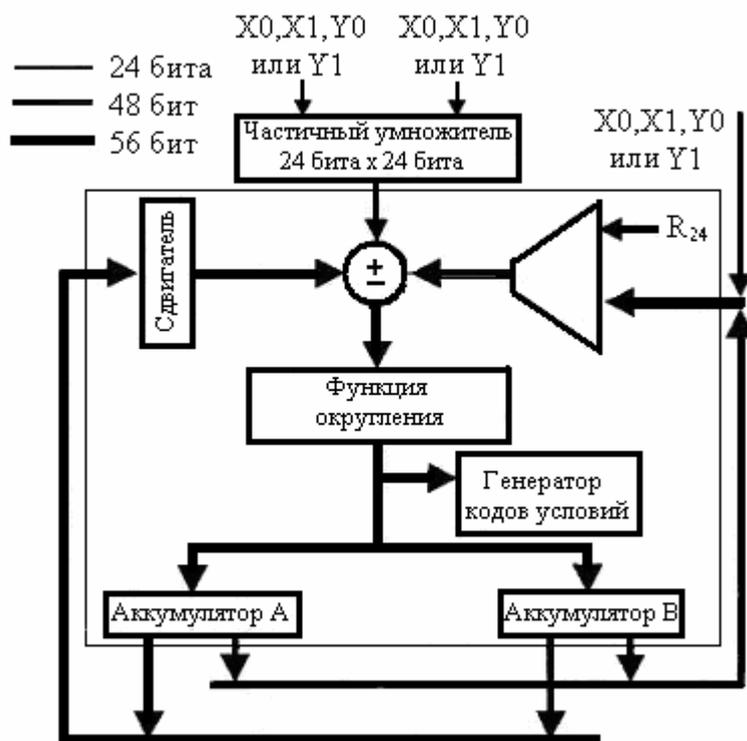


Рисунок 5.9 - Устройство MAC

Округление до ближайшего целого применяется при сложении произведения с содержимым аккумулятора в специальной команде DSP - MACR. Бит, до которого идет округление в аккумуляторе, определяется битом режима масштабирования в регистре статуса.

Логическое устройство выполняет операции логического умножения AND, логического сложения OR, исключающее ИЛИ EOR, инверсии NOT над регистрами АЛУ. Это устройство имеет разрядность 24 бита и оперирует старшей значащей частью операнда в аккумуляторе.

Устройство сдвига аккумулятора. Асинхронное параллельное устройство сдвига с 56-битным входом и 56-битным выходом подключено непосредственно к выходам MAC. Выполняет операции: нет сдвига; сдвиг на один бит влево: ASL, LSL, ROL; сдвиг на один бит вправо: ASR, ASL, ROR; обнуление.

Устройство сдвига/ограничения. Обеспечивает специальную обработку данных при выдаче из регистров аккумулятора на шины данных. Для каждой шины данных имеется отдельное устройство сдвига/ограничения. Операция ограничения в случае переполнения разрядной сетки выполняется посредством замены результата вычислений максимально возможной величиной. Знак данных устанавливается соответственно операнду источника: \$7FFFFFFF - для 24-битного

или \$7FFFFFF FFFFFFF - для 48-битного положительного числа, \$800000 - для 24-битного, \$800000 000000 - для 48-битного отрицательного числа.

МАС использует округление содержимого аккумулятора с одинарной точностью. Данный метод округления называется округлением до ближайшего целого или конвергентным округлением.

Пространство памяти DSP56002. Пространство памяти DSP56002 представлено на рисунке 5.10. Регистр OMR расширен битами YD (3 бит), MC (4 бит) SD (6 бит). Биты DE и YD задают режимы памяти (таблица 5.1).

Таблица 5.1 - Режимы работы памяти

DE	YD	Память данных
0	0	Внутренние ПЗУ запрещены и их адреса являются частью внешней памяти
0	1	Внутреннее ПЗУ данных X запрещено и является частью внешней памяти. Внутренние ОЗУ и ПЗУ данных Y запрещены и являются частью внешней памяти
1	0	ПЗУ данных X и Y разрешены
1	1	Внутренние ОЗУ и ПЗУ данных Y запрещены и являются частью внешней памяти. Внутреннее ПЗУ данных X разрешено

Однокристалльный режим. В однокристалльном режиме разрешены все внутренние памяти данных и программ. Программный сброс вызывает переход по адресу \$0000 и выполнение соответствующей процедуры.

Карты памяти для режима 0 и режима 2 идентичны, но в режиме 2 вектор сброса находится по адресу \$E000.

Режим загрузки из EEPROM. Загрузочные режимы предназначены для загрузки программ из байтового ПЗУ во внутреннюю память программ в течение сброса по питанию. После включения питания генератор состояний ожидания добавляет 15 состояний ожидания ко всем обращениям к внешней памяти, что позволяет использовать медленную память.

Программа загрузки использует байты в трех последовательно расположенных ячейках внешнего ПЗУ для образования слова внутренней памяти программ. В загрузочном режиме разрешены загрузочное ПЗУ на кристалле и выполнение программы загрузки.

Нормальный расширенный режим. В этом режиме разрешено внутреннее ОЗУ программ и вектор аппаратного сброса по адресу \$E000.

Режим разработки. В этом режиме внутреннее ОЗУ программ запрещено, а вектор аппаратного сброса находится по адресу \$0000. Все ссылки на пространство памяти программ переадресуются прямо во внешнюю память.

Режим загрузки по Host. В этом режиме разрешены загрузочное ПЗУ и выполнение программы загрузки. Режим аналогичен режиму 1, за исключением того, что программа загружается из внешней памяти программ по host-интерфейсу.

Режим загрузки по SCI. В этом режиме разрешены загрузочное ПЗУ и выполнение программы загрузки. Внутреннее и/или внешнее ОЗУ программ загружается по интерфейсу SCI. Количество слов программы и стартовый адрес должны быть определены.

Программа загрузки по SCI ожидает приема трех байтов, определяющих количество слов программы, трех байтов, определяющих адрес, с которого будут загружаться слова программы, и трех байтов для загрузки каждого слова программы. Режимы DSP56002 и управляющие сигналы приведены в таблице 5.2.

Данные принимаются в SCI, начиная с младших битов. После окончания приема слов программы начинается выполнение загруженной программы. SCI запрограммирован на работу в асинхронном режиме с длиной данных - 8 бит, 1 стоп-бит, проверка на четность отсутствует.

Таблица 5.2 -Операционные режимы работы DSP56002

Опер. режим	МС	МВ	МА	Описание
0	0	0	0	Однокристалльный режим - ОЗУ программ разрешено
1	0	0	1	Режим загрузки с EPROM, выход в режим 0
2	0	1	0	Нормальный расширенный режим - ОЗУ программ разрешено
3	0	1	1	Режим разработки - ОЗУ программ запрещено
4	1	0	0	Резерв
5	1	0	1	Загрузка по Host, выход в режим 0
6	1	1	0	Загрузка по SCI (внешняя синхронизация), выход в режим 0
7	1	1	1	Резерв

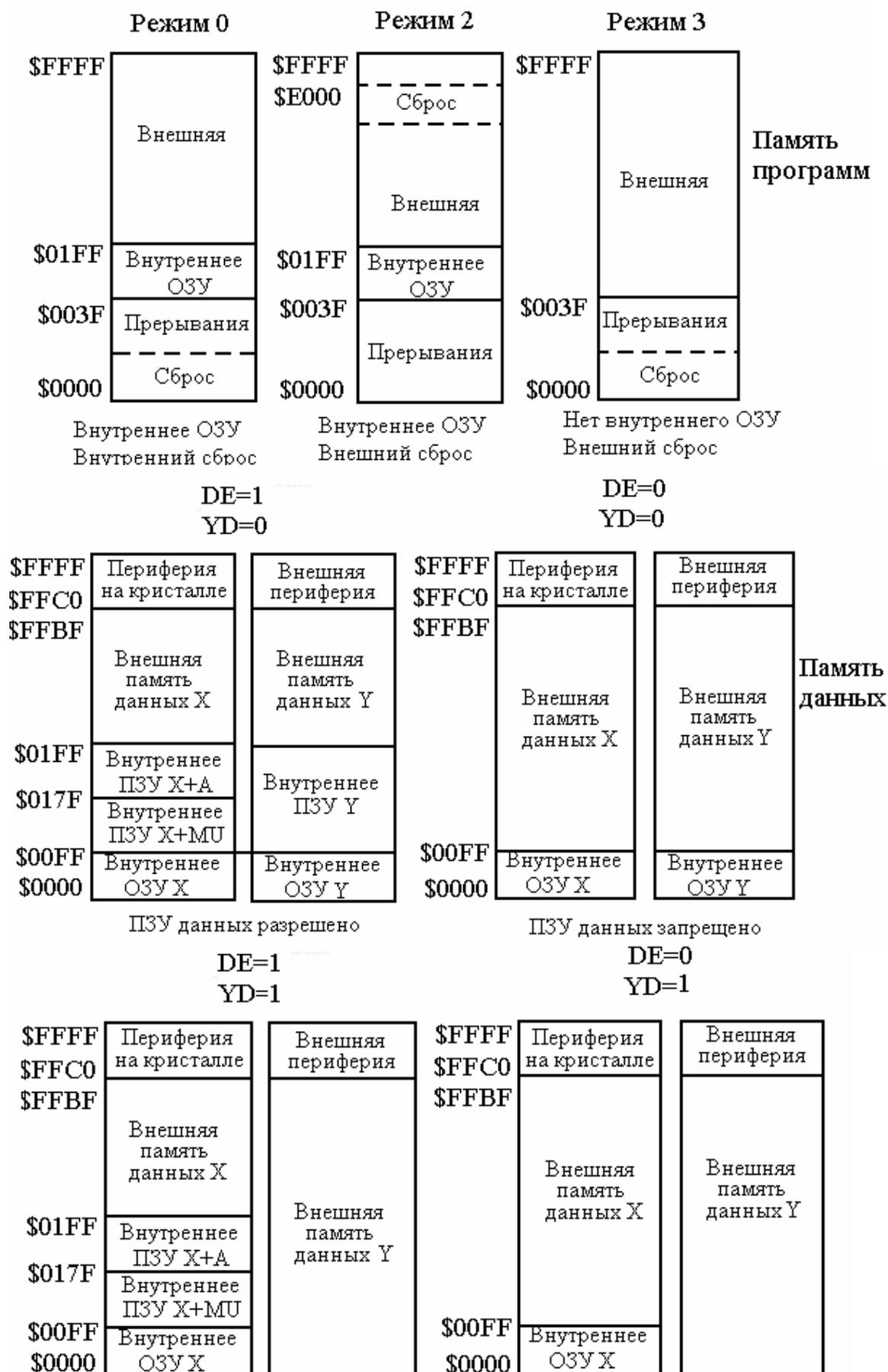


Рисунок 5.10 – Пространство памяти DSP56002

Устройство генерации адресов (address generate unit - AGU) и режимы адресации.

Структура AGU. AGU выполняет вычисление истинных (физических) адресов данных в памяти. Это устройство использует три типа адресной арифметики: линейную модификацию адреса, модульную модификацию адреса, адресацию с реверсивным переносом - и работает параллельно с остальными устройствами на кристалле, что сокращает время на генерацию адресов. Структура AGU показана на рисунке 5.11. Все регистры AGU являются 24-разрядными, в которых 16 младших разрядов являются значащими. Старшие 8 разрядов недоступны для записи и при чтении заполняются нулями.

Блок регистров адреса (R0-R3 и R4-R7). Каждый из двух подблоков регистров адреса содержит четыре 16-битных регистра, которые содержат адреса для обращения к памяти. Каждый регистр может быть прочитан или записан с использованием глобальной шины данных. При выдаче содержимого регистров на шину данных 16-битные регистры записываются в два младших значащих байта шины данных, а старший значащий байт заполняется нулями.

При записи в регистры старший значащий байт шины данных отсекается. Каждый адресный регистр может использоваться как вход АЛУ адресов для модификации регистров. Регистр из АЛУ нижних адресов и регистр из АЛУ верхних адресов доступны в одной инструкции.

Если параллельно осуществляется пересылка данных из памяти X и из памяти Y, адресные регистры разделяются на два блока: R0 - R3, R4 - R7. Содержимое регистров адреса может модифицироваться соответственно режиму адресации. Тип модификации определяется содержимым регистров модификации (Mn). Регистры смещения (Nn) используются для режима адресации «модификация со смещением».

Большинство режимов адресации модифицируют регистры адреса в цикле «чтение - модификация - запись».

Блок регистров смещения (N0-N3 и N4-N7). Каждый из двух подблоков регистров смещения содержит четыре 16-битных регистра, которые содержат величину смещения, используемую для модификации адресных указателей или данных. Каждый регистр может быть прочитан или записан с использованием глобальной шины данных. При выдаче содержимого регистров на шину данных 16-битные регистры записываются в два младших значащих байта шины данных, а старший значащий байт заполняется нулями. При записи в регистры старший значащий байт шины данных отсекается.

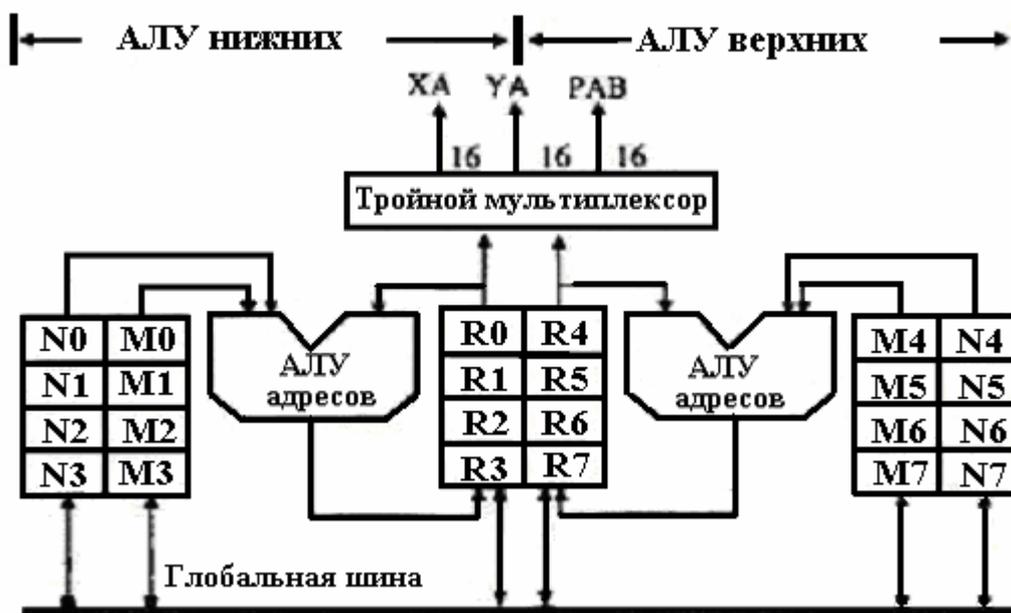


Рисунок 5.11 - Структура AGU

Блок регистров модификации (M0-M3 и M4-M7). Каждый из двух подблоков регистров модификации содержит четыре 16-битных регистра, которые определяют тип адресной арифметики для вычисления модификаций регистра адреса или данных. Каждый регистр может быть прочитан или записан с использованием глобальной шины данных. При выдаче содержимого регистров на шину данных 16-битные регистры записываются в два младших значащих байта шины данных, а старший значащий байт заполняется нулями. При записи в регистры старший значащий байт шины данных отсекается. Каждый регистр модификации устанавливается в \$FFFF после сброса, что определяет линейную арифметику при вычислении модификаций адреса.

АЛУ адресов. Два идентичных АЛУ адресов содержат 16-битный полный адрес, который может быть инкрементирован, декрементирован или к которому может быть добавлено содержимое регистра смещения. Второе полное слагаемое (называемое модулем) определяется результатом суммирования первого полного слагаемого с величиной модуля, хранящейся в регистре модификации. Третье полное слагаемое определяется инкрементированием, декрементированием содержимого адресного регистра или его суммированием с величиной смещения и переносом. Смещение и реверсивный перенос подаются параллельно на разные входы. Тестовая логика определяет,

какой из трех результатов подается на выход в качестве полного адреса

Каждое АЛУ адресов может модифицировать один регистр адреса в течение одного командного цикла. Содержимое регистра модификации определяет тип арифметики для модификации адреса. Величина модификатора декодируется в АЛУ адресов.

Адресация. DSP56000/DSP56001 обеспечивают три различных типа адресации: прямая регистровая, косвенная регистровая и специальная. При прямой регистровой адресации и специальных режимах использование AGU не является обязательным. При прямой регистровой адресации в команде указывается физический адрес регистра, в котором находится операнд. Если регистр адреса используется для указания ячейки памяти, адресация называется косвенной регистровой, т.е. содержимое регистра указывает не на сам операнд, а на адрес операнда. Специальная адресация определяет адрес с учетом его модификации. Модификация заключается в увеличении или уменьшении на единицу содержимого адресного регистра до или после выполнения команды.

Программный контроллер. Программный контроллер обеспечивает предварительную выборку инструкций, декодирование инструкций, управление аппаратными циклами и обработку исключений. Контроллер содержит 15-уровневый 32-разрядный системный стек и шесть непосредственно адресуемых регистров (рисунок 5.12), программный счетчик PC (Program Counter), регистр адреса цикла LA (Loop Address), счетчик петли LC (Loop Counter), регистр статуса SR (Status Register), регистр режима операций OMR (Operating Mode Register) и указатель стека SP (Stack Pointer).

Системный стек представляет собой отдельный блок внутренней памяти, используемый для хранения регистра статуса и программного счетчика при вызовах подпрограмм и длительных прерываниях. В стеке также хранятся LA и LC для организации программных циклов.

Каждая ячейка стека включает два 16-разрядных регистра: старшую часть (SSH) и младшую часть (SSL) стека.

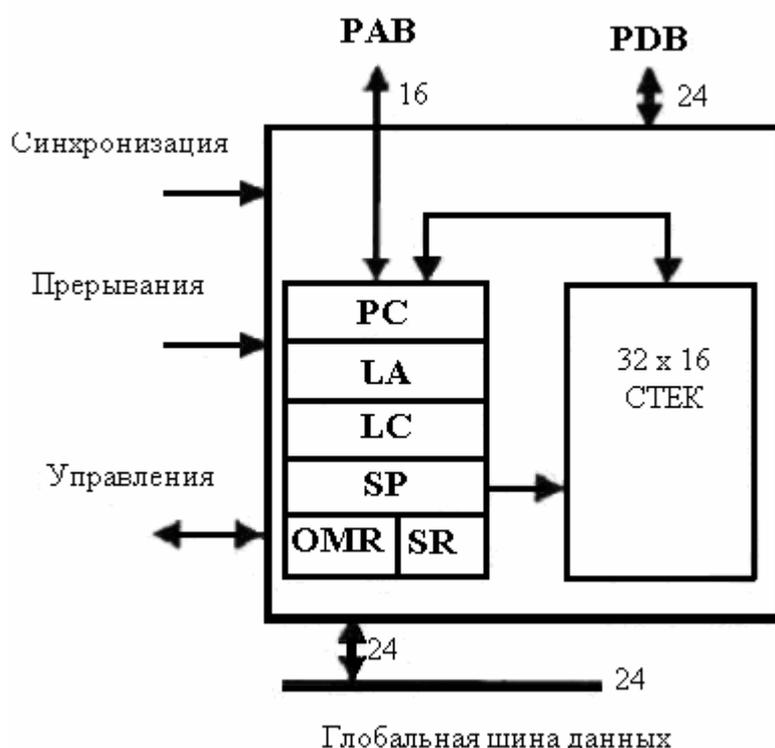


Рисунок 5.12 - Регистры контроллера

Все эти регистры могут быть прочитаны или записаны для упрощения отладки системы. Хотя ни один из регистров программного контроллера не является 24-битным, все они читаются и записываются через 24-битную шину PDB. При чтении регистров младшая часть битов (LSB) является значащей, а старшая значащая часть битов (MSB) заполняется нулями. При записи регистров младшая часть битов (LSB) является значащей, а старшая значащая часть битов просто отбрасывается, поскольку имеет смысл только младшая значащая часть битов (биты 15-0). Программный контроллер содержит конвейер с тремя ступенями и управляет пятью состояниями процессора: нормальным, обработкой исключений, сбросом, ожиданием и остановом.

Структура программного контроллера. Программный контроллер состоит из трех аппаратных блоков: контроллера декодирования программ (PDC), генератора адресов программ (PAG) и контроллера прерываний (PIC).

Контроллер декодирования программ. PDC включает в себя программную логику для декодирования, генератор адреса регистра, механизм выполнения циклов, механизм выполнения повторений, генератор кодов условий, механизм выполнения прерываний, регистра-защелку инструкций и его копию. PDC декодирует 24-битную инст-

рукцию, загруженную в регистр-защелку, и вырабатывает все сигналы, необходимые для управления конвейером. Копия регистра-защелки инструкций оптимизирует выполнение инструкций повторения и перехода.

Генератор адресов программ. PAG содержит: регистры PC, SP, SS (системный стек), OMR (регистр операционного режима), SR, LC и LA. Циклы, являющиеся основной конструкцией алгоритмов цифровой обработки сигналов, поддерживаются аппаратно.

При выполнении инструкции цикла DO в регистр счетчика цикла загружается количество повторений цикла, а в регистр адреса цикла - адрес последней инструкции цикла, и устанавливается флаг цикла в регистре статуса. Перед выполнением инструкции DO содержимое регистров LA, LC и SR сохраняется в стеке. Под управлением механизма выполнения циклов адрес первой инструкции цикла помещается в стек. Пока флаг цикла в регистре статуса не сброшен, механизм выполнения циклов сравнивает содержимое PC с содержимым LA для определения последней инструкции цикла. Когда последняя инструкция выбрана, содержимое LC сравнивается с единицей. Если равенство не выполняется, содержимое LC декрементируется и из SS читается адрес первой инструкции цикла. Если равенство выполняется, то значения LA, LC и флага цикла в SR восстанавливаются из стека, а выборка инструкций продолжается с адреса LA + 1.

Пересылка массива данных может быть выполнена с использованием механизма повторений. Инструкция REP загружает в LC количество повторений следующей за ней инструкции. Так как команда, которая будет повторяться, выбрана только один раз, это увеличивает производительность за счет уменьшения обращений к внешней шине. Однако инструкция REP не может быть прервана, поскольку выбирается только один раз.

Контроллер прерываний. PIC принимает все запросы прерываний, осуществляет арбитраж в каждом цикле и генерирует адрес вектора прерывания. Прерывания могут вызывать четыре внешних и 16 внутренних источников прерываний.

Используется структура гибкого приоритета прерываний. Каждое прерывание получает свой уровень приоритета (IPL) - от 0 до 3. Уровень 0 - самый низкий, а уровни 1 и 2 маскируются. Уровень 3 является высшим и не маскируется. Биты маски прерываний в регистре статуса (SR) показывают текущий уровень приоритета прерываний в процессоре. Прерывания, имеющие уровень приоритета меньше

текущего, не учитываются при арбитраже. Уровень приоритета 3 всегда вызывает прерывание процессора. Уровни приоритета прерываний для каждого периферийного устройства на кристалле (HI, SSI, SCI) и для каждого внешнего источника прерываний (\neg IRQA, \neg IRQB) могут задаваться программным путем от 0 до 2. Уровни приоритета устанавливаются при записи в регистр уровней приоритета, представленный на рисунке 5.13.

В DSP56002 регистр уровней приоритета прерываний расширен: добавлены биты 16 - TIL1 и 17 - TIL0, которые задают уровень приоритета прерываний от таймера. Источники прерываний и их уровни приоритета указаны в таблице 5.3. Каждый источник прерываний имеет свой вектор для вызова процедуры обработки прерывания, расположенной в младших 64 словах памяти программ. При переходе к обработке исключений текущая инструкция выполняется, если только выбранное слово не является первым словом двухсловной инструкции, иначе выполнение инструкции прерывается.

Далее выполняется две выборки адреса, во время которых PC не изменяется. PC генерирует выборку адреса инструкции прерывания, указывающего на первое слово двухсловной инструкции подпрограммы быстрой обработки прерывания. Затем производится выборка двух слов инструкции по вектору прерывания.

Если одно из этих слов - переход к подпрограмме, то быстрая обработка прерывания переводится в длительную. Процедура длительной обработки прерывания помещает текущий контекст процессора в стек. Подпрограммы и прерывания могут быть вложенными. Стек может быть расширен в память за счет использования программного доступа к регистрам SSH и SSL. Контакты внешних прерываний \neg IRQA и \neg IRQB могут быть настроены на прерывания по переднему или заднему фронту. Немаскируемое прерывание NMI осуществляется по уровню и выполняется по первому появлению на контакте \neg IRQB 10B после последнего обслуживания NMI или сброса. NMI имеет уровень приоритета 3 и не маскируется. NMI не используется как вход прерывания общего назначения, поскольку зарезервировано для аппаратных разработок.

Вектор аппаратного сброса может указывать на внутреннюю или внешнюю (P.\$E000) память программ. Немаскируемое прерывание, трассировка и программное прерывание используются при отладке.

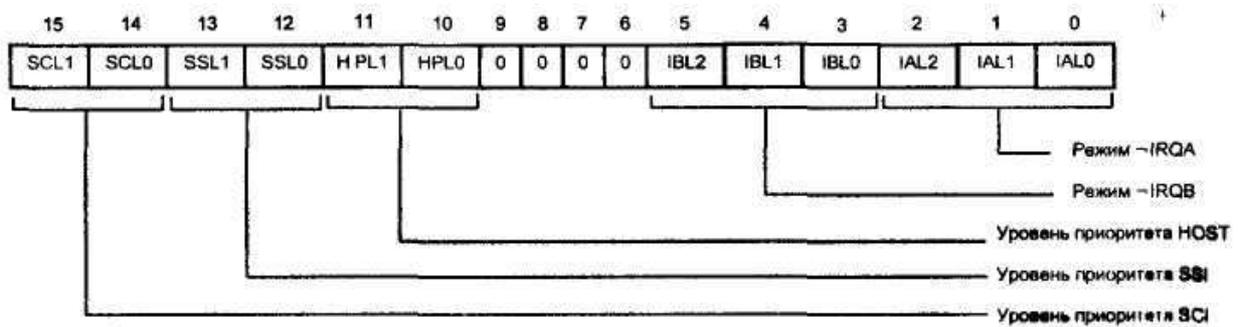


Рисунок 5.13- Формат регистра уровней приоритета

Программное прерывание используется для организации точек останова. При трассировке инструкции выполняются в пошаговом режиме.

Таблица 5.3 - Уровни приоритета

Стартовый адрес	IPL	Источник
P.\$0000/P.\$E000	3	Аппаратный внешний сброс
P.\$0002	3	Ошибка стека
P.\$0004	3	Трассировка
P.\$0006	3	Программное прерывание SWI
P.\$0008	0-2	Внешнее прерывание LLJIRQA
P.\$000A	0-2	Внешнее прерывание LUIRQB
P.\$000C	0-2	SSI - Приемник данных
P.\$000E	0-2	SSI - Приемник данных в состоянии исключения
P.\$0010	0-2	SSI - Передатчик данных
P.\$0012	0-2	SSI - Передатчик данных в состоянии исключения
P.\$0014	0-2	SCI - Приемник данных
P.\$0016	0-2	SCI - Приемник данных в состоянии исключения
P.\$0018	0-2	SCI - Передатчик данных
P.\$001A	0-2	SCI - Простой линии
P.\$001C	0-2	SCI - Таймер
P.\$001E	3	NMI - Зарезервировано для разработок
P.\$0020	0-2	Host - Приемник данных
P.\$0022	0-2	Host - Передатчик данных
P.\$0024	0-2	Host - Команда
P.\$0028	0-2	Доступно для команд Host
P.\$002A	0-2	Доступно для команд Host
P.\$002C	0-2	Доступно для команд Host
P.\$002E	0-2	Доступно для команд Host
P.\$0030	0-2	Доступно для команд Host
P.\$0032	0-2	Доступно для команд Host
P.\$0034	0-2	Доступно для команд Host
P.\$0036	0-2	Доступно для команд Host
P.\$0038	0-2	Доступно для команд Host
P.\$003A	0-2	Доступно для команд Host
P.\$003C	0-2	Доступно для команд Host
P.\$003E	0-2	Неправильная инструкция

Все периферийные устройства на кристалле используют одинаковый механизм запроса прерываний. Каждое устройство имеет одну линию запроса прерывания в PIC и две входные линии: линию чтения вектора и линию ответа.

Устройство, имеющее более одного источника прерывания, проводит собственный арбитраж прерывания в соответствии с его внутренним уровнем приоритета. PIC выбирает одно из периферийных устройств и посылает ему адрес вектора прерывания.

Конвейер инструкций. Программный контроллер содержит трехуровневый конвейер, выполняющий выборку, декодирование и исполнение инструкций. Устройства выборки, декодирования и исполнения расположены последовательно.

Синхронизация. DSP56000/DSP56001 использует четырехфазную синхронизацию выполнения инструкций. Синхронизация обеспечивается внутренним генератором, соединенным с внешним кристаллом контактами XTAL и EXTAL, или внешним генератором по линии EXTAL.

Программная модель. Программная модель контроллера представлена на рисунке 5.14. Она содержит наряду с программным счетчиком, регистром статуса и системным стеком дополнительные регистры счетчика цикла и адреса цикла, предназначенные для аппаратной поддержки инструкции DO.

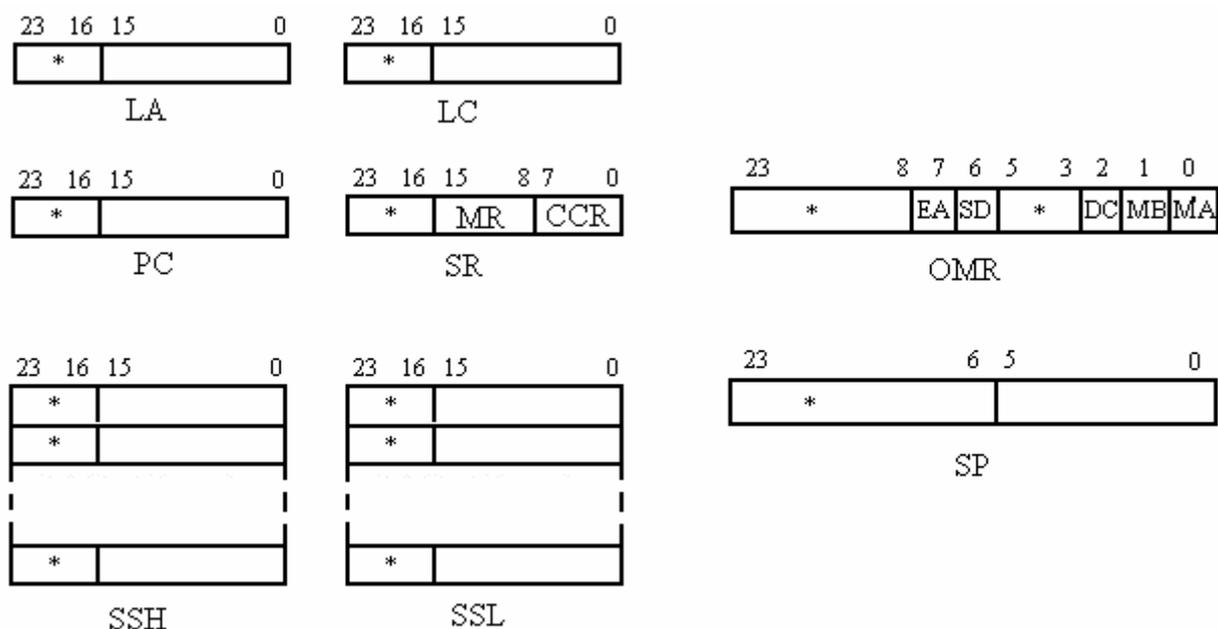


Рисунок 5.14 - Программная модель контроллера (* - запрещены для изменения, читаются как нуль)

Программный счетчик (PC), 16-битный регистр, содержащий адрес следующей ячейки в памяти программ. PC может указывать на инструкции, операнды данных или адреса операндов. Ссылки на этот регистр подразумеваются в большинстве инструкций. PC запоминается в стеке при выполнении программного цикла, подпрограммы или прерывания.

Регистр статуса (SR). 16-битный регистр статуса SR состоит из регистра режима MR, занимающего 8 старших битов и регистра кодов условий CCR, занимающего 8 младших битов. Регистр статуса сохраняется в стеке при выполнении программного цикла, подпрограммы или прерывания. Формат регистра показан на рисунке 5.15.

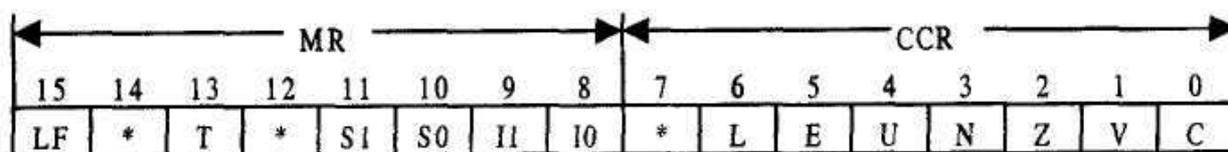


Рисунок 5.15 - Формат регистра статуса SR

Регистр управления (MR). MR - специальный управляющий регистр, определяющий текущее состояние процессора для супервизора. На биты регистра режима MR воздействуют сброс процессора, выполнение цикла, возврат из прерывания, программное прерывание и инструкции, оперирующие непосредственно с содержимым регистра. После аппаратного сброса в регистре устанавливается маска прерывания, биты масштабирования, флаг цикла, бит трассировки сбрасываются в «0».

Регистр кодов условий (CCR). Регистр кодов условий (флагов) определяет текущее состояние процессора для пользователя. На регистр кодов условий воздействуют арифметические и логические операции, операции пересылки и инструкции, оперирующие непосредственно с содержимым регистра. После сброса все биты CCR устанавливаются в «0».

Флаг переноса (C - бит 0) устанавливается в «1» при наличии переноса из старшей значащей части числа в результате сложения. Перенос или заем генерируется из бита 55 результата. Перенос также возникает при манипуляциях с битами, сдвигах и циклических сдвигах.

Флаг переполнения (V-бит 1) устанавливается при арифметическом переполнении 56-битного результата. Это означает, что результат не может быть представлен в аккумуляторе.

Флаг нуля (Z- бит 2) устанавливается в «1», если результат равен нулю.

Флаг отрицательного результата (N - бит 3) устанавливается в «1», если результат имеет отрицательный знак.

Флаг денормализации (U - бит 4) устанавливается, если два старших бита идентичны.

Флаг расширения (E- бит 5) сброшен в «0», если все биты целой части 56-битного результата - нули или единицы, иначе бит установлен. Целая часть определяется режимом масштабирования и битом E. Если бит расширения равен нулю, то младшая часть числа содержит все значащие биты, а старшая часть числа является расширением знака. В этом случае регистр расширения аккумулятора игнорируется. Если бит расширения равен единице, то регистр расширения аккумулятора используется.

Флаг ограничения (L - бит 6) установлен, если установлен бит переполнения. Также устанавливается при выполнении операций ограничения устройством сдвига/ограничения данных. Бит сбрасывается только при сбросе процессора или специальной командой.

Маска прерывания (I1,I0 - биты 8 и 9). Показывает текущий уровень приоритета прерываний в процессоре. Уровень приоритета изменяется программным путем. Биты маски прерываний устанавливаются в «1» при аппаратном сбросе процессора и не изменяются при программном сбросе.

Режим масштабирования (S1,S0 - биты 10 и 11). Биты определяют режим масштабирования для устройства сдвига/ограничения данных АЛУ и позиции округления для MAC. Действие режима масштабирования проявляется при чтении данных из аккумуляторных регистров А и В на шины данных и при округлении в MAC. При сбросе процессора и начале длительной обработки прерывания эти биты равны нулю.

Режим трассировки (T- бит 13). Бит устанавливается в начале выполнения каждой инструкции в пошаговом режиме. Если бит равен нулю, трассировка запрещена и выполнение инструкций идет нормальным образом. При сбросе процессора бит Т равен нулю.

Флаг цикла (LF- бит 15). Бит устанавливается при выполнении программного цикла и разрешает определение окончания цикла. По завершении цикла бит устанавливается в «0».

Регистр операционного режима (OMR). OMR -24-битный регистр (рисунок 5.16), в котором определены 5 битов для

Регистр адреса цикла и регистр счетчика цикла (LA и LC). Содержимое регистра адреса цикла LA указывает на местонахождение слова последней инструкции программного цикла LC -специальный 16-битный счетчик, определяющий число повторений цикла.

При выполнении инструкции DO содержимое регистра LA помещается в SSH, а регистра LC - в SSL, откуда оно извлекается только после окончания цикла по инструкции ENDDO

При выборке инструкции по адресу, содержащемуся в LA, проверяется содержимое LC. Если содержимое LC не равно единице, оно декрементируется и следующая инструкция выбирается по адресу, находящемуся в верхней ячейке стека В противном случае PC инкрементируется, флаг цикла сбрасывается, регистры LA и LC восстанавливаются из стека и продолжается нормальное выполнение программы.

Регистр LC программно доступен, и выполнение цикла может быть прекращено программным путем. Регистр LC также используется в инструкции REP.

Системный стек (SS). SS -отдельная внутренняя память размером 15 x 32 бита, разделенная на два блока SSL и SSH. SSH сохраняет содержимое программного счетчика, SSL -регистра статуса при вызовах подпрограмм и длительных прерываниях. В стеке сохраняются также значения LA и LC при выполнении программных циклов. Стек обеспечивает выполнение 15 длинных прерываний, семи циклов DO, 15 вызовов подпрограмм. Если стек переполняется, возникает немаскируемое прерывание - ошибка стека. Значение PC теряется, что приводит к ошибке программы, выполняющейся, когда возникло прерывание по переполнению стека.

Указатель стека (SP). 6-битный указатель стека определяет последнюю занятую ячейку стека и состояние стека (потеря содержимого, пустой, полный, переполнение). Формат регистра SP указан на рисунке 5.17.

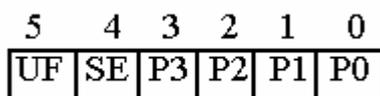


Рисунок 5.17- Формат регистра SP

Указатель стека (P0.P1.P2.P3—биты 0-3) показывает последнюю используемую ячейку стека. После аппаратного сброса все биты равны нулю, что означает - стек пуст.

Флаг ошибки стека (SE- бит 4). Показывает возникновение ошибки стека. Если стек заполнен, указатель стека равен 001111, и любая операция, вызывающая запись в стек, приведет к возникновению ошибки стека, т е SP будет равен 010000 Аналогично, если SP равен 0, любая операция чтения из стека приведет к возникновению ошибки стека. Флаг ошибки стека остается равным единице, пока пользователь не сбросит его в «0».

Флаг потери содержимого (UF- бит 5). Устанавливается при потере содержимого стека. Сбрасывается в «0» пользователем.

Остальные биты стека зарезервированы для будущих разработок

Состояния процессора. DSP всегда находится в одном из пяти состояний: нормальное, обработка исключения, сброс, ожидание и останов

Нормальное состояние процессора связано с выполнением инструкций. Инструкции (команды) выполняются с использованием трехуровневого конвейера.

Состояние обработки исключения связано с прерываниями, которые генерируются как внутренними устройствами DSP, так и поступают от внешних источников.

Состояние сброса возникает при появлении сигнала активного уровня на контакте RESET. В этом состоянии происходит сброс внутренних периферийных устройств, регистры модификации устанавливаются в \$FFFF, регистр уровней приоритета прерываний очищается, регистр BCR устанавливается в \$FFFF, что добавляет задержку к состоянию ожидания при всех обращениях к внешней памяти, указатель стека очищается, биты режима масштабирования, режима трассировки, флаг цикла и биты условий в SR сбрасываются, биты маски прерываний в SR устанавливаются в «1»; бит разрешения ПЗУ данных, бит задержки и бит стробирования памяти сбрасываются. DSP не выходит из состояния сброса, пока на контакте RESET сохраняется сигнал активного уровня. После выхода из состояния сброса биты операционного режима в OMR загружаются извне по линиям MODA, MODB и выполнение программы начинается с адреса \$E000 в памяти программ в нормальном расширенном режиме или с адреса \$0000 в остальных операционных режимах.

Состояние ожидания - это состояние пониженного энергопотребления, в которое процессор переходит при выполнении инструкции WAIT. В состоянии ожидания запрещена внутренняя синхронизация всех устройств на кристалле, за исключением внутренней периферии (генератор опорной частоты работает). Все внутренние процессы остановлены до тех пор, пока не поступит немаскируемое прерывание или не будет произведен сброс.

Состояние останова - это состояние самого низкого энергопотребления, в которое процессор переходит при выполнении инструкции STOP. В этом состоянии генератор опорной частоты отключен. При переходе в состояние останова кристалл сбрасывает все прерывания от периферийных устройств и внешние прерывания. Уровни приоритетов прерываний остаются такими же, как перед выполнением инструкции STOP. Процессор остановлен до тех пор, пока не появится сигнал низкого логического уровня на контакте \neg IRQA или на контакте \neg RESET. Эти события включают генератор, и после стабилизации частоты включается синхронизация всех устройств на кристалле. Период стабилизации частоты определяется битом SD в OMR.

5.4.2 Порты ввода/вывода

Порт А. Порт А может использоваться для расширения памяти или обычного ввода/вывода. Шина данных порта А имеет разрядность 24 бита, шина адреса разрядностью 16 бит обеспечивает доступ к памяти за один машинный цикл (рисунок 5.18).

Во время выполнения инструкции могут быть доступны следующие блоки памяти DSP: X-память данных, Y-память данных, память программ или несколько блоков сразу. Каждый из этих блоков памяти может быть внутренним или внешним. Шины данных и адреса порта А обеспечивают доступ к внешней памяти.

На рисунке 5.18 показаны сигналы порта А. Все сигналы разделены на три функциональные группы: адресные, данных и управления.

Сигналы шины управления также подразделяются на три группы: управление чтением и записью, выбор адресного пространства, управление доступом к шине. Сигналы управления чтением и записью могут быть декодированы как сигналы чтения и сигналы записи или сигнал записи может использоваться в качестве сигнала чтения/записи, сигнал чтения - в качестве выхода разрешения доступа к

памяти. Сигналы выбора памяти программ, памяти данных и типа памяти данных могут рассматриваться как дополнительные адресные линии, позволяющие расширять пространство адресуемой памяти от 64К до 192К слов.

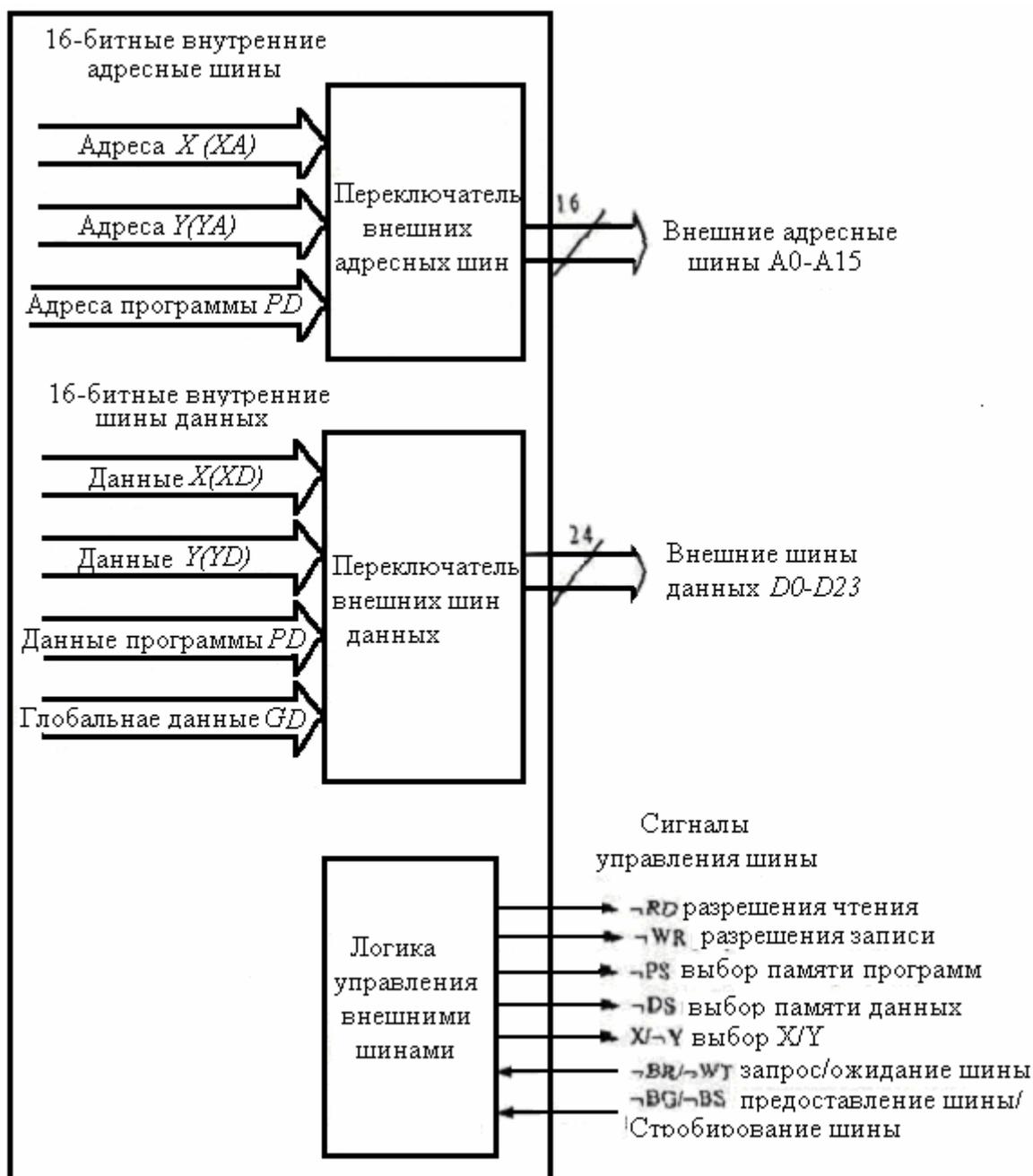


Рисунок 5.18 - Структура порта А

Порт В. Порт В может использоваться как 15 линий ввода/вывода общего назначения, каждая из которых конфигурируется индивидуально как вход или выход или как 8-битный двунаправленный host-интерфейс. Линии порта В представлены на рисунке 5.19. Сконфигурированный как порт ввода/вывода общего назначения, порт В может использоваться для управления внешними устройствами.

ми. В качестве host-интерфейса порт В обеспечивает соединение с другим процессором.

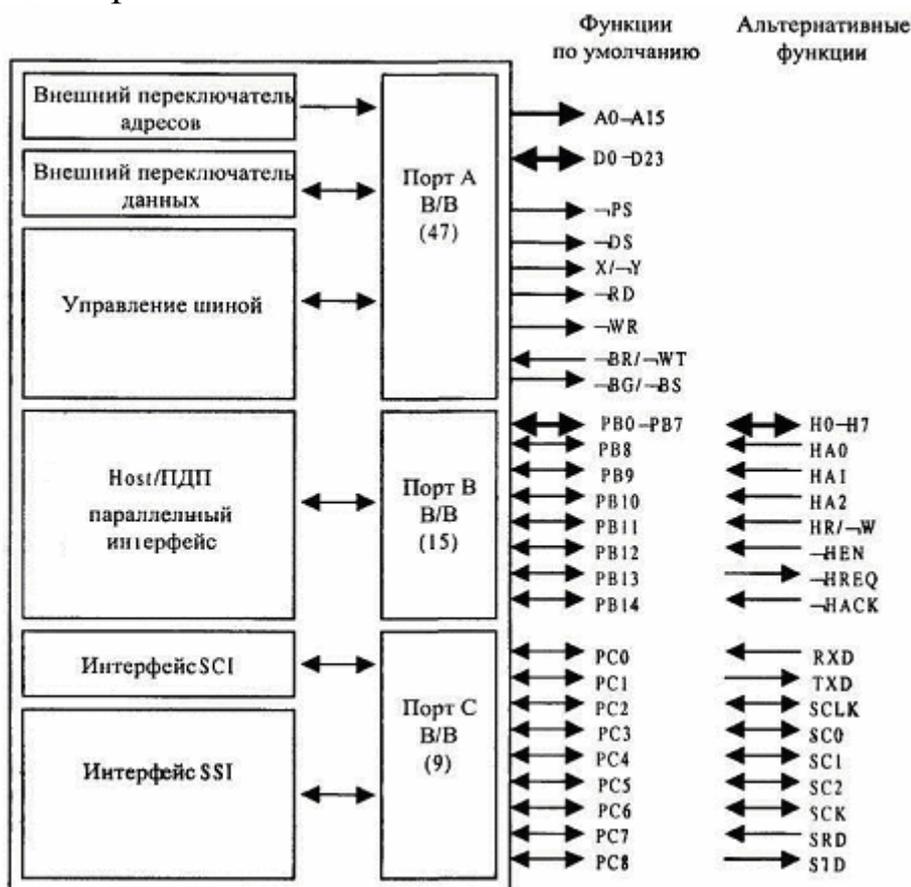


Рисунок 5.19 - Структура порта В

Порт С. Порт С имеет девять линий и три функциональных назначения (рисунок 5.20). Три из девяти линий могут использоваться для стандартного ввода/вывода или для последовательного коммуникационного интерфейса SCI, другие шесть линий могут использоваться для стандартного ввода/вывода или для синхронного последовательного интерфейса SSI.

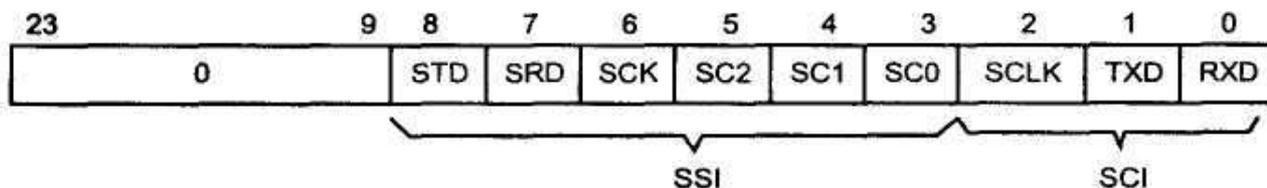


Рисунок 5.20 - Назначение линий порта С

Таким образом, порт С может использоваться для управления устройствами, если он сконфигурирован как порт стандартного ввода/вывода, и для соединения с другими DSP, процессорами, АЦП и ЦАП, если он сконфигурирован как последовательный интерфейс.

При использовании в качестве линий стандартного ввода/вывода порт С может быть представлен в виде трех регистров, которые управляют девятью контактами ввода/вывода. Это регистр управления порта С (PCC, адрес X.\$FFE1), регистр направления данных порта С (PCDDR, адрес X.\$FFE3) и регистр данных порта С (PCD, адрес X.FFE5).

Регистры PCDDR и PCD имеют разрядность 24 бита, из которых используются только младшие девять битов. Соответственно, установка в «1» бита в регистре PCDDR означает, что данная линия порта С используется для вывода, в противном случае - для ввода.

Выбор между стандартным вводом/выводом и последовательными интерфейсами осуществляется с помощью регистра PCC, в котором установка соответствующего бита в «1» означает использование последовательного интерфейса, а сброс в «0» - стандартный ввод/вывод.

Последовательный коммуникационный интерфейс SCI обеспечивает полнодуплексную последовательную связь с другими DSP, микропроцессорами или периферийными устройствами типа модемов. Сигналы интерфейса могут иметь ТТЛ-уровень или стандарт RS232C, RS422 и т. д. Этот интерфейс использует три линии: передачи данных (TXD), приема данных (RXD) и синхронизации (SCLK).

Прием данных (RXD - бит 0). Этот вход принимает байт последовательных данных и передает их в регистр сдвига. Асинхронный ввод данных осуществляется по положительному фронту сигнала синхронизации $1 \times SCLK$, если $SCKP = 0$.

Передача данных (TXD- бит 1). На этот выход передаются последовательные данные из регистра сдвига. Данные меняются по отрицательному фронту сигнала синхронизации SCLK, если $SCKP = 0$.

Синхронизация (SCLK— бит 2). Двухнаправленная линия обеспечивает синхронизацию при приеме или передаче данных в асинхронном режиме, а также при передаче данных в синхронном режиме.

Программная модель SCI. Программная модель SCI приведена на рисунке 5.21. Программная модель может быть представлена в виде регистров трех типов: 1) управления (SCR); 2) статуса (SSR); 3) управления синхронизацией (SCCR).

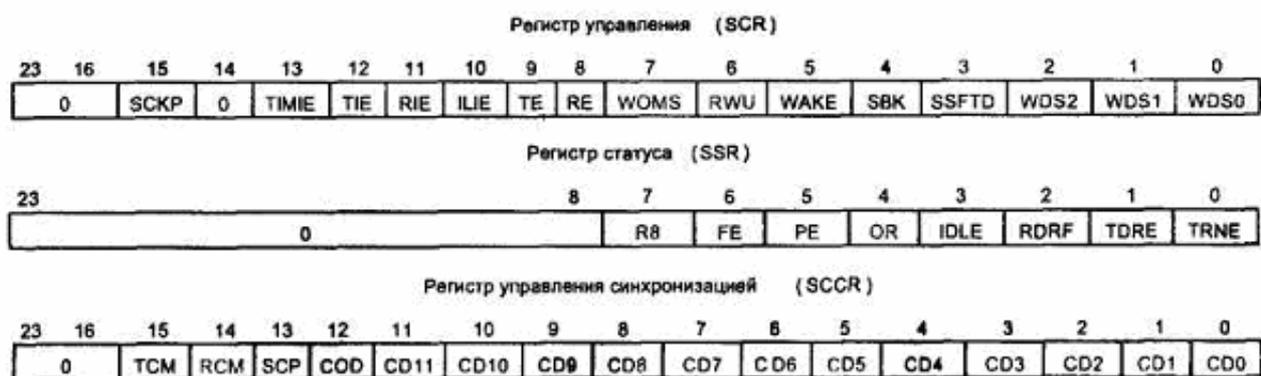


Рисунок 5.21 - Программная модель SCI

Регистр управления (SCR). 16-битный регистр чтения/записи, управляющий операциями последовательного интерфейса. Младшие пятнадцать из 16 битов имеют функциональное назначение (рисунок 5.21). Выбор слова осуществляется битами WDS0, WDS1, WDS2. Эти биты выбирают формат данных для передачи и приема в соответствии с таблицей 5.5.

Таблица 5.5 - Выбор формата данных

WDS2	WDS1	WDS0	Формат слова
0	0	0	8-битные синхронные данные (режим сдвигового регистра)
0	0	1	Резерв
0	1	0	10-битные асинхронные данные (старт-бит, 8 бит данных, стоп-бит)
0	1	1	Резерв
1	0	0	11-битные асинхронные данные (старт-бит, 8 бит данных, бит четности, стоп-бит)
1	0	1	11-битные асинхронные данные (старт-бит, 8 бит данных, бит нечетности, стоп-бит)
1	1	0	11-битные мультиточечные данные (старт-бит, 8 бит данных, бит типа данных, стоп-бит)
1	1	1	Резерв

Асинхронные режимы совместимы с большинством последовательных устройств типа DUART. Эти режимы поддерживают стандарт RS-232. Мультиточечный режим совместим с MC68681 DUART, интерфейсом SCI M68HC11 и последовательным интерфейсом Intel 8051. Синхронный режим, по существу, представляет собой сдвиговый регистр для расширения ввода/вывода. Синхронизация данных выполняется с использованием синхронизации приема и передачи, которая совместима с режимом 0 последовательного интерфейса Intel 8051. При аппаратном сбросе биты очищаются.

Направление сдвига определяется битом SSFTD. Регистр сдвига данных может быть запрограммирован на сдвиг битов, начиная с

младших ($SSFTD = 0$) или - со старших битов ($SSFTD = 1$). Местоположение битов четности и типа данных не изменяется - они находятся рядом со стоп-битом. Программный и аппаратный сбросы очищают этот бит.

Посылка маркера паузы (SBK- бит 4). Маркер паузы представляет собой нулевой фрейм данных. Если бит установлен и затем очищен, передатчик завершает передачу данных, посылает нулевой фрейм и возвращается к холостому режиму или посылке данных. Программный и аппаратный сбросы очищают этот бит.

Выбор режима пробуждения (WAKE- бит 5). Если бит равен нулю, выбирается режим пробуждения по холостой линии (свободной от выполнения каких-либо действий). В данном режиме приемнику повторно разрешен прием данных. Программное обеспечение передатчика должно обеспечить формирование пустой (холостой) строки между последовательными сообщениями. Холостая строка не должна устанавливаться между корректными сообщениями, поскольку каждый фрейм содержит старт-бит, равный нулю.

Если $WAKE = 1$, выбирается режим пробуждения по биту адреса. В этом режиме приемнику разрешено принимать данные, если последний бит принимаемых данных (8 или 9) равен единице.

Последовательный аудиointерфейс (SAI). DSP осуществляет объединение источников и приемников данных через последовательный аудиointерфейс (Serial Audio Interface -SAI). SAI является синхронным и специально предназначен для передачи аудиоданных. Он обеспечивает взаимодействие полнодуплексного последовательного порта с различными аудиоустройствами, в том числе такими как АЦП, ЦАП, устройствами компакт-дисков и т. д. SAI реализует широкий диапазон форматов последовательных данных, используемых в настоящее время в аудио-промышленности:

- формат I²S (Philips);
- формат CDP (Sony);
- формат MEC (Matsushita);
- промышленные А/Ц- и Ц/А-стандарты.

SAI состоит из независимых секций приемника, передатчика и разделяемого (совместно используемого) генератора скорости передачи данных. Каждая из секций приемника и передатчика может работать в режиме ведомого или ведущего. В режиме ведущего в соответствии с программой работы генератора скорости передачи данных производится внутреннее управление последовательным тактовым

сигналом и выбором линий данных. В режиме ведомого от внешнего источника передаются три вида сигналов. Передатчик состоит из трех регистров передаваемых данных, трех полностью синхронизируемых регистров сдвига на выходе и трех линий выхода последовательных данных, управляемых одним контроллером передатчика. Это обеспечивает возможность осуществления одновременной передачи для одного, двух или трех стерео-, аудиоустройств. Приемник состоит из двух регистров принимаемых данных, двух полностью синхронизируемых регистров сдвига на входе и двух линий входа последовательных данных, управляемых одним контроллером приемника. Это обеспечивает одновременный прием из одного или двух аудиоустройств.

5.5 Специализированная цифровая вычислительная машина «Багет-55»

Специализированная цифровая вычислительная машина (СЦВМ) «Багет-55» представляет собой комбинированное устройство, включающее высокопроизводительный программируемый процессор сигналов (ППС) с возможностью доукомплектования специализированным вычислительным устройством (СВУ) /37/.

СЦВМ предназначена для использования в качестве вычислителя для многоканальной цифровой обработки сигналов в реальном масштабе времени в составе авиационных бортовых систем различного назначения, а также для использования в составе систем управления различного класса.

СЦВМ предназначена для эксплуатации в следующих климатических условиях при охлаждении её воздухом с температурой от минус 10° до плюс 23° С, очищенным от капельной влаги, масла и механических частиц и расходе воздуха не менее 100 кг/ч:

- 1) пониженная рабочая температура - от минус 55° С;
- 2) повышенная рабочая температура - до плюс 60° С;
- 3) относительная влажность воздуха окружающей среды - не более 98% при температуре не более плюс 35° С;
- 4) пониженное атмосферное давление - до 5,47 КПа (41мм. рт. ст.).

Работоспособность изделия сохраняется после воздействия предельных температур: минус 60° , плюс 85° С, а также в условиях воздействия соляного (морского) тумана, атмосферных конденсиро-

ванных осадков - росы и внутреннего оледенения, статической пыли, плесневых грибов и акустического шума.

Основные технические параметры и технические характеристики СЦВМ Багет-55-04 /30/:

тип основного микропроцессора1B577 (аналог DSP96002);

тип дополнительного микропроцессора..... 1B578;

разрядность обрабатываемых данных:

- фиксированная запятая, бит 32;

- плавающая запятая, бит 32/64;

тактовая частота, МГц до 40;

внешняя тактовая частота, МГц до 10;

внешние интерфейсы ввода/вывода:

- последовательный интерфейс типа RS232 2 шт.;

- интерфейс дискретных сигналов: 1 шт.;

- количество входных линий: 4;

- количество выходных линий: 4;

- тип логики сигнала: ТТЛ;

- специализированный 32-разрядный параллельный интерфейс приема информации с выхода АЦП: 1 шт.;

код приема данных по специализированному

параллельному интерфейсу дополнительный;

максимальный темп приема данных по специализированному

параллельному интерфейсу, Мслов/с..... 20;

количество модулей, устанавливаемых в корпус СЦВМ..... 10;

тип входной сети питания..... 200/220 В, 50-400 Гц

потребляемая мощность, не более, Вт 60;

габаритные размеры, не более, мм :

- ширина 257;

- высота 197,5;

- длина (без ловителей) 376;

- длина (с ловителями) 405;

вес, не более, кг 15,2 .

СЦВМ Багет-55-04 включает в себя следующие составные час-

ти:

- программируемый процессор сигналов (DSP);

- корпус с установленными в нем модулем электропитания и кросс-платой;

- внутренние соединительные кабели.

В состав ППС изделия должны входить следующие функциональные модули:

- модуль обработки сигналов;
- модуль процессора данных;
- модуль приема и буферизации.

Модули, кабели, источник питания, объединительная панель (кросс-плата) размещены в герметизированном корпусе. Для потребителя доступны разъемы на передней панели корпуса, на которые выведены все интерфейсные сигналы и сигналы управления СЦВМ.

С целью расширения функциональных возможностей СЦВМ и ее исполнение могут быть доукомплектованы специализированным вычислительным устройством (СВУ), имеющим общий канал обмена данными с ППС и состоящим из функциональных модулей центрального процессора (ЦП).

СВУ СЦВМ имеет следующие технические характеристики:

- центральный процессор типа 1В578;
- системная шина VME;
- разрядность системной шины - 32 разряда;
- разрядность обрабатываемых данных:
 - целые числа - 8,16,32 разряда;
 - числа с плавающей запятой - 32 разряда;
- тактовая частота не менее 50 МГц;
- емкость ОЗУ не менее 16 Мбайт;
- электрически стираемая программируемая постоянная память EEPROM (flash type) емкостью не менее 32 Мбайт;
- количество таймеров не менее трех.

При необходимости использования дополнительных интерфейсов взаимодействия СЦВМ с внешними устройствами объекта может быть произведено доукомплектование платами мезонинными в части ППС и СВУ.

Все электронные устройства СЦВМ размещаются в корпусе, защищающем их в процессе эксплуатации от механических повреждений других внешних воздействующих факторов.

В нижней части корпуса установлена панель объединительная, при помощи которой организовано объединение внутренних магистралей модулей, входящих в состав СЦВМ, и преобразование входных сигналов СЦВМ, поступающих в виде парафазных сигналов, в сигналы вида ТТЛ.

В передней части объединительной платы установлены разье-

мы, через которые внешние интерфейсы СЦВМ с помощью внутренних кабелей выводятся на переднюю панель корпуса. На внутренней стороне задней стенки корпуса установлен источник электропитания, формирующий вторичные напряжения + 5В,+12В, и сигналы управления питанием. Связь источника электропитания с объединительной платой осуществляется через два 15-контактных разъема.

Доступ к объединительной плате осуществляется через съемную нижнюю крышку, доступ к модулям - через съемную верхнюю крышку корпуса.

С внешней стороны задней стенки корпуса имеются резьбовые отверстия для присоединения воздуховодов внешней системы вентиляции или для установки блока вентиляторов.

Структурная схема СЦВМ приведена на рисунке 5.22.

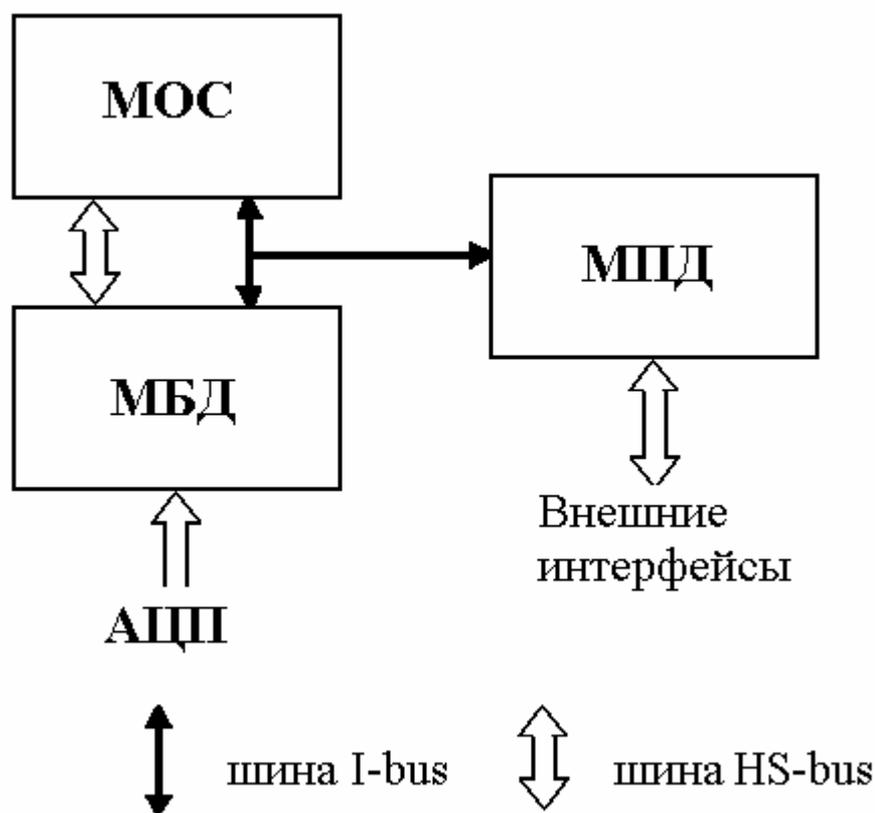


Рисунок 5.22 – Структурная схема СЦВМ

СЦВМ может функционировать в двух режимах - штатном и технологическом.

В штатном режиме работы обеспечивается:

- начальная инициализация СЦВМ по включению питания;
- определение режима начального тестирования по команде пользователя;

- начальное тестирование СЦВМ в соответствии с выбранным режимом;

- выдача результатов тестирования в память;

- передача управления программе пользователя, размещаемой в репрограммируемом постоянном запоминающем устройстве (РПЗУ).

В технологическом режиме работы обеспечивается:

- начальная инициализация СЦВМ по включению питания;

- определение режима начального тестирования по команде пользователя;

- начальное тестирование СЦВМ в соответствии с выбранным режимом;

- выдача результатов тестирования в память;

- дистанционная загрузка программ пользователя с инструментальной ПЭВМ;

- запуск на исполнение программы пользователя;

- запись отлаженного рабочего программного обеспечения в РПЗУ.

Модуль обработки сигналов (МОС). Модуль обработки сигналов представляет собой вычислительное устройство с симметричной внутренней архитектурой, содержащее четыре сигнальных микропроцессора, и является основной вычислительной единицей для программируемых процессоров сигналов.

МОС предназначен для обработки потоков информации, поступающих по I-шине от буфера данных или других МОС в реальном масштабе времени, под управлением модуля процессора данных.

МОС содержит 4 сигнальных процессора 1B577 (DSP-процессора), работающих на частотах до 40.0 МГц, и имеет пиковую производительность 160 MIPS или 240 MFLOPS. Структура модуля приведена на рисунке 5.23.

В состав МОС входят следующие функциональные блоки:

- вычислительные элементы ВЭ_А - ВЭ_Д, содержащие процессор 1B577, локальную статическую память (SRAM) объемом 2Мб, локальное запоминающее устройство для организации очередей типа «первым пришел – первым вышел» (first in – first out FIFO) емкостью 8К x 32-разрядных слов и логику управления;

- общая статическая память объемом 2 Мб;

- загрузочное ПЗУ объемом не менее 32 Кб (PROM);

- выходное FIFO, расположенное на общей шине (M-bus) процессоров;

- контроллер HS-шины (HS-CTRL);
- многофункциональный контроллер (MF-CTRL);
- энергонезависимое последовательное ЗУ объемом 2Кбит (EEPROM);
- контроллер I-шины (I-CTRL).

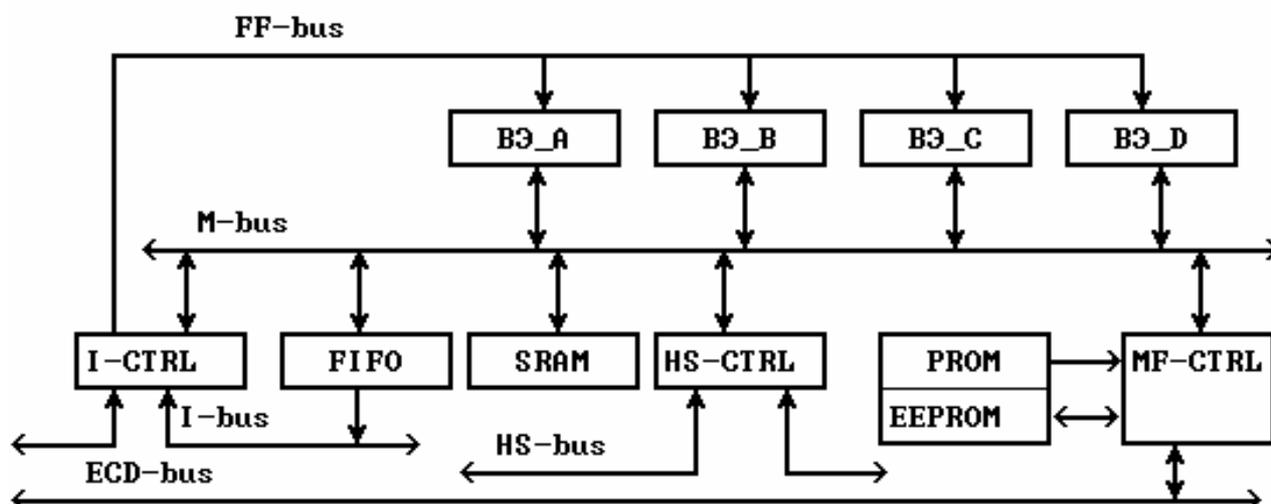


Рисунок 5.23 – Структура модуля МОС

К внутренним шинам модуля относятся:

- M-шина - общая шина сигнальных процессоров;
- Lx-шины - локальные шины ВЭ_А...ВЭ_Д;
- FF-шина - входная шина данных локальных FIFO.

К внешним шинам относятся:

- I-шина - шина приема/передачи данных от АЦП, может использоваться для локального межмодульного обмена данными;
- HS-шина - шина межмодульного обмена данными;
- ECD-шина - шина управления процессами в системе.

Модуль процессора данных (МПД) является комбинированным устройством и выполняет функции управляющего устройства в СЦВМ и обеспечивающего устройства при обмене данными СЦВМ с внешними системами.

Условно модуль можно разделить на две логические части:

субмодуль А и субмодуль Б. Субмодуль А реализован на базе сигнального микропроцессора 1В577 и предназначен для управления работой СЦВМ (управление событиями, процедурами смены режимов, отладка и т.п.). Субмодуль Б реализован на базе однокристалльного RISC-микропроцессора 1В578-ОМП и обеспечивает сопряжение СЦВМ с внешними системами. Связь между этими двумя субмодулями осуществляется через двухпортовую память.

В состав МПД входят следующие функциональные узлы:

субмодуль А:

- микропроцессор 1B577, работающий на частотах до 40 МГц;
- локальная статическая память (Local SRAM) объемом 1Мб;
- загрузочное последовательное ППЗУ объемом 256 Кбит;
- глобальная статическая память (Global SRAM) объемом 1Мб;
- контроллер HS-шины (HS-CTRL);
- многофункциональный контроллер (Multy-CTRL);

субмодуль Б:

- RISC-микропроцессор 1B578-ОМП, работающий на частоте 50МГц;
- статическая память микропроцессора (SRAM) объемом 8Мб;
- загрузочное ППЗУ объемом 4 Мбит;
- флэш-память объемом до 16 Мб;
- буферная статическая память (Buffer SRAM) объемом 1Мб, доступная как со стороны процессора, так и со стороны магистрали VME;
- контроллер последовательных каналов обмена данными типа RS232;
- универсальный контроллер (Uni-CTRL);
- контроллер шины PCI (PCI-CTRL);
- контроллер шины VME (VME-CTRL);
- двухпортовая память объемом до 64К 32-разрядных слов.

Модуль имеет набор интерфейсов, обеспечивающих его сопряжение как с модулями СЦВМ, так и с внешними устройствами. Обмен данными с модулями СЦВМ осуществляется по шинам HS и ESD. Для реализации сопряжения с внешними устройствами модуль имеет два слота с интерфейсом шины PCI, в которые могут быть установлены стандартные РМС-мезонины с необходимыми интерфейсами. Помимо перечисленного, модуль МПД содержит два последовательных канала обмена данными, имеющие электрические интерфейсы RS-232, а также набор дискретных сигналов, которые используются как команды управления, с уровнями ТТЛ (4 входных линии и 4 выходных линии).

Модуль приема и буферизации данных (МПД). Модуль приема и буферизации данных, или просто буфер данных (МБД), предназначен для приема данных, подлежащих обработке, от какого-либо внешнего устройства (АЦП),

МБД выполняет следующие функции:

- прием и буферизацию информации от внешних источников;
- присвоение идентификатора ("цвета" входному потоку данных);
- программируемую коммутацию потоков с разными идентификаторами;
- выдачу информации из данного МБД в последующий (мезонинная плата);
- выявление сбоев в работе входных каналов и формирование по ним события на ECD-шине;
- программно маскируемое формирование события на ECD-шине при поступлении сигналов "начало кадра" - ИНК.

Структурная схема МБД, приведена на рисунке 5.24.

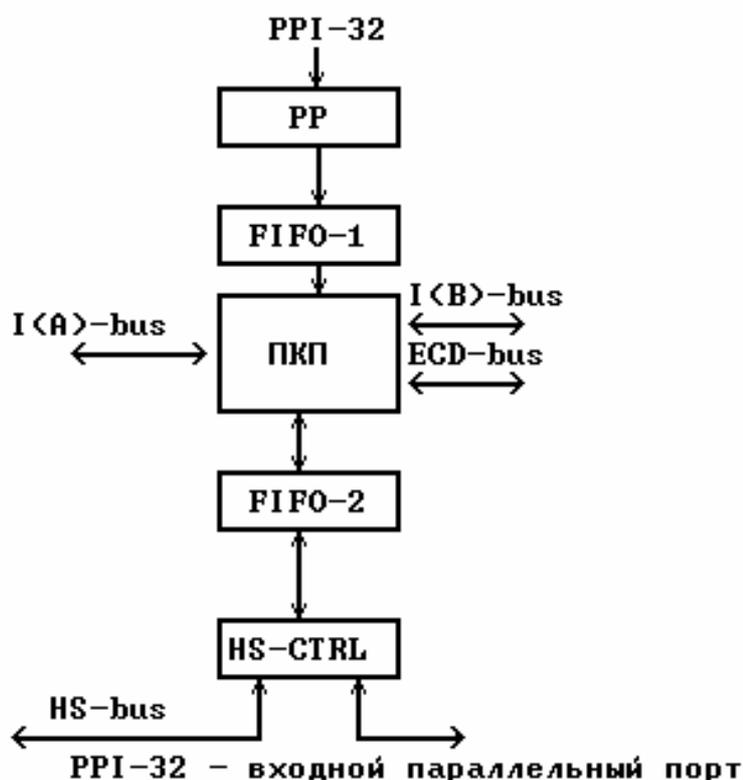


Рисунок 5.24 – Структурная схема модуля приема и буферизации данных

МБД содержит один 32-разрядный параллельный порт (PP-32), работающий только на прием данных от АЦП.

Модуль содержит следующие функциональные блоки:

- входной интерфейс параллельный порт;

- выходной интерфейс для мезонинной платы выходного порта;
- буферную память FIFO-1 глубиной 8К 32-разрядных слов на входе от АЦП;
- программируемый коммутатор потоков данных (ПКП) SWITCH;
- буферную память FIFO-2 глубиной 8К 32-разрядных слов на входе от HS-шины;
- контроллер HS-шины (HS-CTRL).

Модуль БД имеет многошинную (многоканальную) внутреннюю архитектуру, которая позволяет коммутировать шины между собой, обеспечивая тем самым разнообразное распределение потоков данных.

Максимальная пропускная способность внешнего выходного канала составляет 20 миллионов 32-разрядных слов / с (при системной частоте 40 МГц). Источниками данных для выходного канала могут выступать любые каналы, подключенные к программируемому коммутатору.

Таким образом, характеристики СЦВМ «Багет-55» обеспечивают эффективную цифровую обработку одно- и двумерных сигналов.

ЗАКЛЮЧЕНИЕ

Цифровая обработка сигналов является перспективным направлением развития науки, поэтому издано ранее и издается в последнее время большое количество литературы, так или иначе связанной с ЦОС.

Имеются книги по теоретическим основам ЦОС, в основном изданные в восьмидесятих годах прошлого столетия, и современные книги по техническим средствам ЦОС - микроконтроллерам, программируемым процессорам сигналов.

В монографии сделана попытка объединить материал и провести логическую связь от теоретических основ через практические методы и алгоритмы к техническим средствам реализации ЦОС.

Насколько это удалось автору, судить читателям. Отзывы и пожелания направляйте по адресу: г. Ставрополь, Беличенко д.6, кв.45, Баженову А.В.

СПИСОК ЛИТЕРАТУРЫ

1. Баскаков, С.И. Электродинамика и распространение радиоволн / С.И. Баскаков. -М.: Высшая шк., 2001.
2. Баскаков, С.И. Радиотехнические цепи и сигналы: Учеб. для вузов/С.И. Баскаков. - М.: Высш. шк., 2003.-462 с.
3. Радиотехнические системы: Учеб. для вузов по спец. «Радиотехника» / Ю.П. Гришин, В.П. Ипатов, Ю.М. Казаринов и др.; Под ред. Ю.М. Казаринова.- М.: Высш. шк., 1990.- 496 с.
4. Радиоэлектронные системы: основы построения и теория. Справочник / Ширман Я.Д., Лосев Ю.И., Минервин Н.Н., Москвитин С.В. Горшков С.А., Леховицкий Д.И., Левченко Л.С.// Под ред. Я.Д. Ширмана.- М.: ЗАО «МАКВИС», 1998.- 828 с.
5. Гольденберг, Л.М. Цифровая обработка сигналов: Справочник /Л.М.Гольденберг, Б.Д.Матюшкин, М.Н.Поляк. – М.: Радио и связь, 1985. – 312 с.
6. Рабинер, Л. Теория и применение цифровой обработки сигналов/ Рабинер Л., Гоулд Б. – М.: Мир, 1978. – 848 с.
7. Ярославский, Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику. – М.: Радио и связь, 1987. – 296 с.
8. Баженов, А.В. Вычислительные системы авиационных РЭК. Часть 4. Цифровая обработка сигналов в вычислительных системах АРЭК. Курс лекций./ А.В. Баженов, А.П. Курляндчик, В.А. Гимбицкий. Ставрополь: ФВАТУ,2000.-131с.
9. Ахмед, Н. Ортогональные преобразования при обработке цифровых сигналов: Пер. с англ. / Ахмед Н., Рао К.Р., Под ред. И.Б.Фоменко. – М.: Связь, 1980. – 248 с.
10. Бутаков, Е.А. Обработка изображений на ЭВМ /Бутаков, Е.А., Островский В.И., Фадеев И.Л. - М.: Радио и связь, 1987.
11. Хармут Х.Ф. Несинусоидальные волны в радиолокации и радиосвязи: Пер. с англ.- М.: Радио и связь, 1985.- 376 с.
12. Л. Левкович-Маслюк, А. Переберин. Вейвлет анализ и его приложения.Курс лекций. /http://inet.keldysh.ru/GraphiCon98/leo_lev/
13. Каппелини, В. Цифровые фильтры и их применение. Пер. с англ. /В.Каппелини, А.Дж.Константинидис, П.Эмилиани. – М.: Энергоатомиздат, 1983. – 360 с.

14. Потемкин, В.Г. Система инженерных и научных расчетов MATLAB 5.X/ В.Г. Потемкин. В 2-х томах. – М.: ДИАЛОГ-МИФИ, 1999.
15. Рудаков П.И., Сафонов И.В. Обработка сигналов и изображений. MATLAB 5.x / Под общ. ред. В.Г. Потемкина.- М.: ДИАЛОГ-МИФИ, 2000.- 416с.-(Пакеты прикладных программ;Кн.2).
16. Вентцель Е.С. Теория вероятностей. –М.: Гос. из-во физмат лит-ры, 1962.- 277 с.
17. Мартинес,Ф. Синтез изображений.-М.: Радио и связь, 1990.
18. Павлидис, Т.. Алгоритмы машинной графики и обработки изображений.- М.: Радио и связь, 1986.
19. Прэтт, У. Цифровая обработка изображений.-М.: Мир.- Т.1,2. 1982.
20. Сойфер, В.А. Компьютерная обработка изображений. Ч. 1,2 //Соровский образовательный журнал.- 1996.- № №2,3.
21. Хорн, Б.К. Зрение роботов.-М.: Мир, 1989.
22. Фомин Я.А. Тарловский Г.Р. Статистическая теория распознавания образов. - М.: Радио и связь, 1986.
23. Распознавание образов. Состояние и перспективы: Пер. с англ./ Под ред. И.Б. Гуревича. –М.: Радио и связь,1985.
24. Фор А. Восприятие и распознавание образов: Пер. с франц. –М.: Машиностроение,1989.
25. Кондратьев А.И. Теоретико-числовые распознающие алгоритмы. –М.: Наука, 1990.
26. Иванов В.Ю. Тимофеев А.В. Построение локально-оптимальных байесовых правил для классификации и идентификации объектов// Информационные технологии и интеллектуальные методы. Выпуск №2.- СПб.: СПИИРАН, 1997.-с.112-120.
27. Королев А.А. Особенности машинной обработки динамических изображений .// Информационные технологии и интеллектуальные методы. Выпуск №3.- СПб.: СПИИРАН, 1999, с.254-261.
28. Алексеев, Ю.Я. Помехозащищенность авиационных радиолокационных систем/ Ю.Я. Алексеев, В.Н. Антипов, В.А. Ефимов, В.И Меркулов, И.Н. Умнов / Под редакцией В.А. Ефимова.- М.: ВАТУ, 2001.-271 с.
29. Проектирование импульсных и цифровых устройств радиотехнических систем: Учеб. пособие./Под ред. Ю.М. Казаринова. - М.: Высш. шк. , 1985.- 319 с.

30. Микропроцессорные системы: Учебное пособие для вузов/ Е.К. Александров, Р.И. Грушвицкий, М.С. Куприянов и др. Под общ. Ред. Д.В. Пузанкова. – СПб.: Политехника, 2002.- 935 с.

31. Современные микроконтроллеры: Архитектура, средства проектирования, приемы применения, ресурсы сети Интернет. «Телесистемы». Под ред. Коршуна И.В.; Составление, пер. с англ. и литературная обработка Горбунова Б.Б. -М.: Издательство «Аким», 1998.-272 с.

32. Микроконтроллеры фирмы Analog Devices. <http://css.vstu.edu.ru/sprav.spr2/analog.html>

33. Марков, С. Цифровые сигнальные процессоры. Книга I.-М.: фирма МИКРОАРТ, 1996. – 114 с.

34. Куприянов, М. С. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования / М. С. Куприянов, Б. Д. Матюшкин. - 2-е изд. , перераб. и доп. - СПб.: Политехника, 2002. - 592 с.

35. Сикарев, А.А. Микроэлектронные устройства формирования и обработки сложных сигналов/ А.А. Сикарев, О.Н. Лебедев. – М.: Радио и связь, 1983.- 216 с.

36. Миронов,С. Цифровая обработка радиолокационных сигналов на основе процессора Л1879ВМ1/С.Миронов, В.Дударев, А.Богатов. Электронный журнал. Электроника. Раздел радиолокация. Выпуск №3- 2003.

37. СЦВМ «Багет-55-04». Техническое описание и инструкция по эксплуатации.