

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

1/2003

НАУЧНО - ТЕХНИЧЕСКИЙ ЖУРНАЛ

**КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ
БАНКИ ЦИФРОВЫХ ФИЛЬТРОВ
СИГНАЛЬНЫЕ ПРОЦЕССОРЫ
БЫСТРАЯ РАЗРАБОТКА VoIP**

ISSN 1684-2634



9 771684 263005 >

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Научно-технический журнал
№1 (7)/2003

Издается с 1999 года.
Выходит четыре раза в год.

ГЛАВНЫЙ РЕДАКТОР
Ю.Б. ЗУБАРЕВ

ЗАМЕСТИТЕЛИ ГЛАВНОГО РЕДАКТОРА:
В.В. ВИТЯЗЕВ, В.П. ДВОРКОВИЧ

РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

Б. А. Бабаян, Ю. И. Борисов,
С. А. Грибачев, Г. В. Зайцев, Р. В. Зубарев,
А. П. Кирпичников, М. И. Кривошеев,
Н. А. Кузнецов, М. С. Куприянов, А. А. Ланнэ,
В. Г. Мистюков, С. Л. Мишенков,
А. А. Петровский, Ю. Н. Прохоров,
А. Н. Соловьев, Ю.Г. Сосулин,
Н. Г. Харатишвили, В. В. Шахильдян,
Ю. С. Шинаков

Адрес редакции:
103064 Москва, ул. Казакова, 16
Тел./факс: (095) 267-0662
E-mail: editor@dspa.ru
http://www.dspa.ru

Для писем:
129090 Москва, а/я 48.

Издатель:
ООО «КБВП»
Генеральный директор: С. А. Задворнов
Дизайн и верстка: И.А. Фацевская
Корректура: С. В. Витязев

Подписной индекс по каталогу
ОАО «Роспечать» – 82185

Подписано в печать 05.03.2003. Формат 60x90/8.
Гарнитура «Pragmatica C». Печать офсетная.
Бумага мелованная. Печ. л. 7,0. Тираж 1000 экз.
Заказ №128. Отпечатано в
ООО «Типография Принтхаус»
Москва, ул. Вавилова, вл. 9а, стр. 7,
тел.:132-7218, 105-07-98.

Издание зарегистрировано в Министерстве
Российской Федерации по делам печати,
телерадиовещания и средств массовых
коммуникаций.
Свидетельство о регистрации ПИ №77-1488
от 14.01.2000

© КБВП, 2003

В НОМЕРЕ:

- А.А. Зайцев
Методы построения банков цифровых фильтров: тематический обзор 2
A.A. Zaytsev
Theory and methods of filter bank design (Overview) 2
- И.В. Каюков, В.Б. Манелис
Квазикогерентный прием многолучевого сигнала с прерывистым пилот-сигналом в мобильных системах связи 11
I.V. Kayukov, V.B. Manelis
Quasicoherent reception of multipath signal with interrupted pilot signal in mobile communication systems 11
- Н.Г. Харатишвили, И.М. Чхеидзе, З.Дж. Гогилашвили
Применение морфологических преобразований для пирамидального кодирования изображений 17
N.G. Haratishvili, I.M. Cheidze, Z.Dj. Gogilashvili
The use of morphological transformations for pyramidal image encoding 17
- Б.А. Алпатов, П.В. Бабаян
Электронная юстировка изображений при мультиспектральном наблюдении 24
B.A. Alpatov, P.V. Babayan
Electronic registration of multispectral images 24
- А.В. Бондаренко, И.В. Докучаев, В.Б. Стешенко
Аппаратная реализация систем ввода, обработки, хранения и вывода изображений различных телевизионных форматов и стандартов 27
A. Bondarenko, I. Dokuchaev, V. Steshenko
Realization of multistandard TV I/O cards for digital image processing 27
- А.О. Валитов, А.В. Герус, И.Ф. Комлик, М.С. Куприянов
Декодирование файлов в стандарте MP3 33
A.O. Valitov, A.V. Gerus, I.F. Komlick, M.S. Kupriyanov
MP3-files decoding 33
- С.В. Иванов, В.В. Витязев, В.В. Езерский
Метод устранения фазовых искажений в устройствах радиометрии 36
S.V. Ivanov, V.V. Vityazev, V.V. Ezersky
The method of elimination of phase distortions in radiometry devices 36
- Н.Н. Абкаиров, А.А. Фрунзе
Быстрая разработка VoIP и других приложений телефонии 40
N. Abkairov, A. Frunze
Fast VoIP and other telephony applications development 40
- Новые разработки DSP компании Analog Devices (по материалам Internet)** 50
Analog Devices: DSP news overview (Internet materials) 50
- В.А. Куликов, В.Е. Чернов
Передовые платформы для отечественных систем цифровой обработки сигналов 55
V.A. Kulikov, V.E. Chernov
Advanced platforms for digital signal processing systems 55

УДК 621.372

Методы построения банков цифровых фильтров: тематический обзор

А.А. Зайцев

Введение

В последние десятилетия в области цифровой частотной селекции и многоскоростной обработки сигналов достигнуты значительные успехи [1–3]. Использование методов децимации и интерполяции позволяют с наименьшими затратами строить фильтры с высокими показателями частотной избирательности. Регистрация данных на повышенной частоте дискретизации и последующая децимация обеспечивают более высокое отношение сигнал/шум. Полосовая фильтрация с последующим понижением частоты дискретизации дает возможность строить эффективные системы частотной селекции сигналов. Многоскоростные системы находят применение в связи, при обработке речи, изображений и других цифровых данных различной природы возникновения, а также в спектральном анализе, в радиолокационных и антенных системах.

В данной статье рассматривается одно из основных направлений многоскоростной обработки сигналов – построение систем анализа/синтеза сигналов с использованием банка фильтров (БФ). Начиная с 80-х годов прошлого столетия это направление бурно развивается как в теоретическом, так и в прикладном плане. За рубежом по данной тематике опубликовано огромное количество работ и сделано множество научных докладов. К сожалению, русскоязычных публикаций, появившихся за последнее время по данной тематике, не так много. Среди наиболее значимых следует отметить [4, 5]. Цель этой работы – познакомить читателей с основными положениями и результатами, достигнутыми в области построения и расчета БФ.

Мы опустим основные теоретические сведения, связанные с многоскоростной фильтрацией, полифазным разбиением и блоками фильтров, так как они подробно изложены в превосходном методическом обзоре [4] и монографии [2].

Основную идею построения системы анализа/синтеза сигналов с использованием БФ раскрывает рис. 1.

Исходный сигнал разбивается при помощи фильтров анализа $H_i(z)$, $i=0...M-1$ на M субполос-

Представлен тематический обзор методов построения банков цифровых полосовых фильтров с полной децимацией, полным и квазиполным восстановлением, равномерным и неравномерным разделением частотных каналов. Рассмотрены постановка и решение задачи оптимизации параметров банков фильтров с использованием математического аппарата параунитарных матриц и систем «без потерь».

ных составляющих, которые в идеальном случае в частотной области не перекрываются. Подобранным соответствующим образом набор фильтров синтеза $F_i(z)$, $i=0...M-1$, можно восстановить исходный сигнал из его субполосных компонент. Вследствие ограничения ширины спектра сигналов на выходе БФ можно уменьшить частоту дискретизации субполосных сигналов пропорционально уменьшению ширины спектра. Для понижения частоты дискретизации на стадии анализа и последующего повышения на стадии синтеза используются соответственно компрессоры и экспандеры частоты дискретизации. В случае когда коэффициент прореживания в каждом канале равен отношению ширины спектра субполосного сигнала к ширине спектра исходного или

$$\sum_{i=0}^{M-1} \frac{1}{v_i} = 1, \quad (1)$$

говорят о системе анализа/синтеза с полной децимацией. Таким образом, получается M сигналов, отражающих поведение исходного сигнала в каждом частотном поддиапазоне, которые представлены в сумме тем же количеством отсчетов, что и исходный сигнал. Каждый субполосный сигнал в отдельности может быть эффективно обработан по некоторому алгоритму P_i , $i=0...M-1$.

Традиционно БФ разделяют на банки с равномерным и неравномерным разбиением каналов. Равномерное разбиение подразумевает одинаковый коэф-

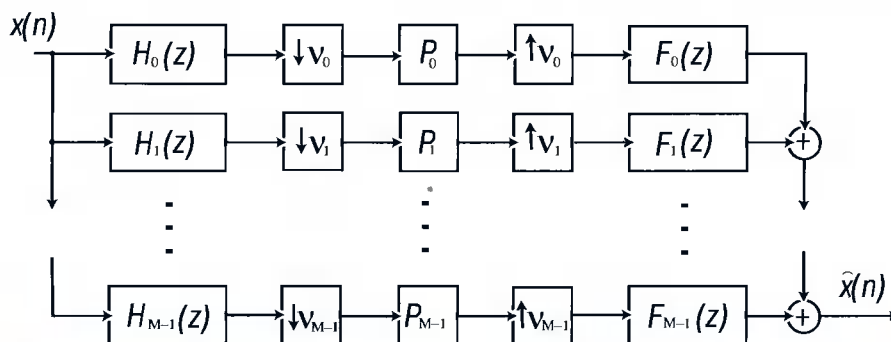


Рис. 1. Система анализа/синтеза сигналов на основе банка фильтров

коэффициент децимации и одинаковую суммарную ширину полосы пропускания каждого канала. В этот класс цифровых БФ входят также и банки с многокомпонентными фильтрами (имеющими более одной полосы пропускания). В случае неравномерного разбиения на каналы коэффициенты децимации различны и в общем случае могут быть выражены в виде рационального числа

$$v_i = \frac{q_i}{p_i}$$

В связи с тем что невозможно построить фильтр с идеальной прямоугольной АЧХ, приходится мириться с наличием у полосового фильтра переходной зоны и неравномерностью в полосе пропускания. Это обстоятельство в сочетании с эффектом «заворачивания» частот, возникающим при децимации, может привести к значительным искажениям при восстановлении сигнала. Однако существуют методики расчета БФ, которые позволяют полностью (банки фильтров с полным восстановлением – БФсПВ) или почти полностью, то есть с любой наперед заданной точностью, восстановить исходный сигнал. Теория построения БФсПВ была детально разработана П.П. Вайдьянатханом [3, 6–8].

Банки фильтров с равномерным разбиением на каналы

В случае равномерного разбиения исходного сигнала на субполосные составляющие: $v_i=M$ для всех $i=1...M$. При этом наиболее общее выражение для Z-образа $\hat{X}(z)$ восстановленного сигнала имеет следующую форму:

$$\hat{X}(z) = \frac{1}{M} \sum_{l=0}^{M-1} X(zW^l) \sum_{k=0}^{M-1} H_k(zW^l) F_k(z), \quad (2)$$

где $W=e^{-j2\pi/M}$. Компоненты $X(zW^l)$, $l \neq 0$, представляют собой так называемое «наложение» (*aliasing*). Они будут устранены только в том случае, если будет выполняться условие

$$\begin{bmatrix} H_0(z) & H_1(z) & \dots & H_{M-1}(z) \\ H_0(zW^1) & H_1(zW^1) & \dots & H_{M-1}(zW^1) \\ \dots & \dots & \dots & \dots \\ H_0(zW^{M+1}) & H_1(zW^{M+1}) & \dots & H_{M-1}(zW^{M+1}) \end{bmatrix} \cdot \begin{bmatrix} F_0(z) \\ F_1(z) \\ \dots \\ F_{M-1}(z) \end{bmatrix} = \begin{bmatrix} T(z) \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (3)$$

Матрица $[H_k(zW^l)]_{l,k}$, ($0 \leq k, l \leq M-1$) получила название матрицы компонент наложения (AC-матрица). Если наложение будет устранено, то структура, изображенная на рис. 1, будет инвариантной во времени (и естественно, линейной). $\hat{X}(z)$ связан с Z-образом $X(z)$ исходного сигнала передаточной функцией $T(z)$:

$$T(z) = \frac{\det \hat{X}(z)}{X(z)} = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z) H_k(z). \quad (4)$$

Таким образом, $T(z)$ представляет собой искажение, вызванное системой анализа/синтеза без наложения. Если $T(z)$ – всепропускающая (устойчивая) функция, то восстановленный сигнал не будет иметь амплитудных искажений. Если $T(z)$ – линейно-фазовая функция, то восстановленный сигнал не будет иметь фазовых искажений. В случае же, когда $T(z)$ вырождается в простую

задержку, система анализа/синтеза позволяет восстановить сигнал без искажений, т.е. является системой анализа/синтеза с полным восстановлением. В работах [6–8] было показано, что все три вида искажений, которым подвержен восстановленный сигнал $\hat{x}(n)$, могут быть полностью устранены с конечными затратами за счет соответствующего тщательного выбора фильтров $H_i(z)$ и $F_i(z)$.

Цифровые БФ с полным восстановлением

Теоретически можно подставить $T(z)=z^{-n_0}$ в (3) и, инвертировав матрицу AC, получить банк синтезирующих фильтров $F_k(z)$. Однако такой подход практически не применим, так как матрица AC в большинстве случаев плохо обусловлена. Кроме того, даже если $H_k(z)$ описывают КИХ-фильтры, обратная матрица в общем случае не будет относиться ни к КИХ-фильтрам, ни к устойчивым фильтрам.

Одним из наиболее общих подходов к решению задачи расчета БФсПВ является метод, предложенный американским ученым П.П. Вайдьянатханом. Данный метод базируется на использовании систем «без потерь» (*losslessness*) или параунитарных матриц. Рассмотрим основную идею данного метода, опираясь на результаты исследований, полученные в [3, 4, 6–9].

Систему, изображенную на рис. 1, используя полифазное представление и замечательные тождества [4], можно преобразовать к эквивалентной структуре, представленной на рис. 2.

Для устранения наложения необходимо найти решение для $F_k(z)$, фигурирующих в (3). Кроме того, для получения системы с полным восстановлением $T(z)$ в (3) должно соответствовать задержке. Для решения такой задачи необходимо выбрать $R(z)=E^{-1}(z)$. Чтобы избежать обращения матрицы $E(z)$, желательно выбрать параунитарную матрицу $\tilde{E}(z)=cE^{-1}(z)$. Тогда при выборе $R(z)$ в соответствии с (5) свойство полного восстановления будет гарантировано [6].

$$R(z) = z^{-k} \tilde{E}(z) = cz^{-k} E^{-1}(z) = cz^{-k} E^T(z^{-1}). \quad (5)$$

В работах [6–9] предлагается способ получения матрицы $E(z)$ на основе каскадного соединения простых

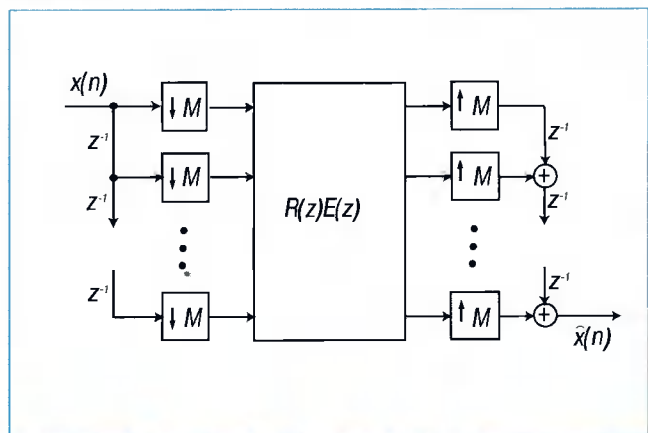


Рис. 2. Эквивалентная структура на основе полифазного разбиения

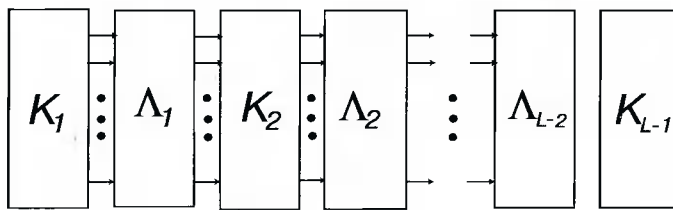


Рис. 3. Простой путь получения $M \times M$ матрицы без потерь

параунитарных матриц и матриц планарных вращений, в результате чего получается параунитарная матрица, зависящая от числа каскадов $(L-1)$ и набора параметров $\{\theta_j\}_{j=1, (L-1), (M-1)}$.

Матрицы K_i строятся на основе произведения матриц планарных вращений

$$K_i = \begin{bmatrix} \cos\theta_{1,i} & \sin\theta_{1,i} & 0 \\ \sin\theta_{1,i} & -\cos\theta_{1,i} & 0 \\ 0 & 0 & I_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_{2,i} & \sin\theta_{2,i} & 0 \\ 0 & \sin\theta_{2,i} & -\cos\theta_{2,i} & 0 \\ 0 & 0 & 0 & I_2 \end{bmatrix} \dots \dots \begin{bmatrix} I_{M-1} & 0 & 0 \\ 0 & \cos\theta_{M-1,i} & \sin\theta_{M-1,i} \\ 0 & \sin\theta_{M-1,i} & -\cos\theta_{M-1,i} \end{bmatrix} \quad (6)$$

Диагональные матрицы имеют форму (7)

$$\Lambda_i = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & z^M \end{bmatrix} \quad (7)$$

Оба типа матриц являются параунитарными матрицами. Следовательно, полученная матрица $E(z)$ также является параунитарной, а система, описываемая такой полифазной матрицей, является системой с полным восстановлением. Остается выбрать параметры $\theta_{i,j}$ таким образом, чтобы фильтры $H_k(z)$ имели «хорошие» частотные характеристики. В качестве характеристики каждого фильтра можно взять уровень подавления боковых лепестков его АЧХ. Тогда, учитывая, что банк фильтров удовлетворяет условию (8) (т.к. АС является матрицей без потерь)

$$\sum_{k=0}^{M-1} |H_k(e^{j\omega})|^2 = 1, \quad (8)$$

можно поставить задачу оптимизации параметров θ следующей целевой функции

$$\Phi(\theta) = \sum_{i=0}^{M-1} \left\{ \int_0^{\pi/M-\varepsilon} |H_i(e^{j\omega})|^2 d\omega + \int_{\pi(i+1)/M-\varepsilon}^{\pi} |H_i(e^{j\omega})|^2 d\omega \right\}. \quad (9)$$

Функция $\Phi(\theta)$ включает в себя интегрирование АЧХ только в зоне непрозрачности, так как при оптимизации (9), вследствие (8), АЧХ фильтров в полосе пропускания имеют соответствующие хорошие характеристики по неравномерности. Показатель ε определяет половину ширины переходной зоны АЧХ фильтров и обусловлен невозможностью построения идеального фильтра.

Таким образом, задача построения БФсПВ сведена к задаче многомерной нелинейной оптимизации. Следует отметить тот факт, что из-за большого количества параметров, по которым необходимо оптимизировать (9), и значительного числа необходимых перемножений матриц оптимизация

требует больших затрат по времени. Несмотря на то что существует способ [8], позволяющий уменьшить количество оптимизируемых переменных и степень сложности вычисления целевой функции, данная методика не годится для получения фильтров с высокими показателями частотной избирательности – чрезвычайно мала скорость сходимости.

Описанный выше способ получения полифазной матрицы системы анализа не подразумевает требования линейности фазовой характеристики каждого фильтра. Тем не менее это свойство во многих приложениях является желательным, так как линейно-фазовые БФ не вносят фазовые искажения в субполосные сигналы, что позволяет избежать серьезных искажений фазы восстанавливаемого сигнала при квантовании субполосных составляющих.

В работах [10–12] приводится способ получения линейно-фазовой параунитарной системы. Рассмотрим множество параунитарных передаточных функций, полифазные матрицы $E(z)$ которых удовлетворяют выражению

$$D \cdot z^{-N} E(z^{-1}) J_M = E(z), \quad (10)$$

где N – порядок параунитарной матрицы $E(z)$, D – диагональная матрица, элементы которой равны: $+1$ в тех строках, которые соответствуют симметричным фильтрам, и -1 в тех, которые соответствуют антисимметричным фильтрам. Такие полифазные матрицы соответствуют множеству БФ с линейной фазой.

Возможно, что существуют линейно-фазовые параунитарные системы, которые не удовлетворяют (10). Примером может служить «цепочка задержек», где фильтры анализа $H_i(z) = z^{-i}$, $i=0, \dots, M-1$. Однако, чтобы получить искомое разложение, необходимо наложить определенное ограничение на полифазную матрицу.

Перейдем непосредственно к разложению полифазной матрицы. В [10] показано, что для обеспечения линейности фазовых характеристик фильтров полифазная матрица должна быть построена следующим образом.

Для четных M :

$$E(z) = S P T_N \Lambda(z) T_{N-1} \Lambda(z) \dots \Lambda(z) T_0 P, \quad (11)$$

$$P = \begin{pmatrix} I_{M/2} & 0 \\ 0 & J_{M/2} \end{pmatrix}$$

$$T_i = \begin{pmatrix} I_{M/2} & I_{M/2} \\ I_{M/2} & -I_{M/2} \end{pmatrix} \cdot \begin{pmatrix} W_i & \mathbf{0} \\ \mathbf{0} & U_i \end{pmatrix} \cdot \begin{pmatrix} I_{M/2} & I_{M/2} \\ I_{M/2} & -I_{M/2} \end{pmatrix}$$

$$S = (1/\sqrt{2}) \cdot \begin{pmatrix} S_0 & \mathbf{0} \\ \mathbf{0} & S_1 \end{pmatrix} \cdot \begin{pmatrix} I_{M/2} & J_{M/2} \\ I_{M/2} & -J_{M/2} \end{pmatrix}$$

$$\Lambda(z) = \begin{pmatrix} I_{M/2} & \mathbf{0} \\ \mathbf{0} & z^{-1} I_{M/2} \end{pmatrix}$$

где W_i, U_i, S_0 и S_1 – ортогональные матрицы, которые могут быть полностью определены $C_2^{M/2}$ углами вращений каждая (подобно (6)); J – антидиагональная матрица. В случае когда требуется построить БФ симметричный относительно $\pi/2$, существует возможность сократить вдвое число параметров (по которым затем проводится оптимизация). Для этого необходимо построить полифазную матрицу следующим образом:

$$E(z) = SP\Lambda(z)PT_0\Lambda(z)\dots T_N P, \quad (12)$$

$$S = (1/\sqrt{2}) \cdot \begin{pmatrix} S_0 & \mathbf{0} \\ \mathbf{0} & J_{M/2} S_1 \end{pmatrix} \cdot \begin{pmatrix} I_{M/2} & J_{M/2} \\ I_{M/2} & -J_{M/2} \end{pmatrix} Q,$$

$$T_i = \begin{pmatrix} I_{M/2} & I_{M/2} \\ I_{M/2} & -I_{M/2} \end{pmatrix} \cdot \begin{pmatrix} V_{M/2} U_i V_{M/2} & \mathbf{0} \\ \mathbf{0} & U_i \end{pmatrix} \cdot \begin{pmatrix} I_{M/2} & I_{M/2} \\ I_{M/2} & -I_{M/2} \end{pmatrix}$$

где Q – симметричная матрица перестановок; V_M – диагональная матрица размером $M \times M$ со сменяющимися друг друга числами ± 1 , начиная с $+1$. Как видно из (12), число неизвестных параметров сократилось вдвое.

Для нечетных M :

$$E(z) = S\Lambda(z)PT_N\Lambda(z)T_{N-1}\Lambda(z)\dots T_0 P, \quad (13)$$

$$P = \begin{pmatrix} I_{(M+1)/2} & \mathbf{0} \\ \mathbf{0} & J_{(M-1)/2} \end{pmatrix}$$

$$T_i = \begin{pmatrix} A_i & \mathbf{0} & C_i \\ \mathbf{0} & 1 & \mathbf{0} \\ C_i & \mathbf{0} & A_i \end{pmatrix}$$

$$\Lambda(z) = \begin{pmatrix} I_{(M+1)/2} & \mathbf{0} \\ \mathbf{0} & z^{-1} I_{(M-1)/2} \end{pmatrix}$$

$$S = (1/\sqrt{2}) \begin{pmatrix} U_{(M+1)/2} & \mathbf{0} \\ \mathbf{0} & W_{(M-1)/2} \end{pmatrix} \begin{pmatrix} I_{(M-1)/2} & \mathbf{0} & J_{(M-1)/2} \\ \mathbf{0} & 1 & \mathbf{0} \\ J_{(M-1)/2} & \mathbf{0} & -I_{(M-1)/2} \end{pmatrix}$$

где T_i – ортогональные матрицы, $\Lambda(z) = \text{diag}(1 \ 1 \dots z^{-N/2} \dots 1 \ 1)$, $U_{(M+1)/2}$ и $W_{(M-1)/2}$ – произвольные ортогональные матрицы. Такое сложное построение матрицы позволяет придать ей структурные свойства линейности фазовых характеристик и матрицы «без потерь».

Цифровые БФ с квазиполным восстановлением

Достаточно часто построение системы анализа/синтеза с теоретически полным восстановлением

является не обязательным. В большинстве приложений требуется получить такие наборы фильтров анализа и синтеза, которые бы позволяли наилучшим в некотором смысле образом разбивать сигнал на субполосы и синтезировать оценку исходного сигнала с наперед заданной точностью. Для получения БФ с «почти полным» восстановлением разными авторами было предложено большое количество разнообразных методик. Многие из них имеют эвристический характер и предназначены для расчета ограниченного класса БФ (например, [13]). Другие подходы, претендующие на общность, можно разделить на класс БФ, получаемых при помощи так называемой « H_∞ -оптимизации» [14, 15], и на класс БФ, получаемых путем косинусной модуляции фильтра-прототипа [16–22].

Все фильтры анализа и синтеза могут быть получены путем модулирования значений коэффициентов фильтра-прототипа, а при их использовании может быть применена эффективная полифазная структура. Существует множество подходов к проектированию фильтра-прототипа. В основном эти подходы связаны с процедурой нелинейной оптимизации.

Можно показать, что если импульсная характеристика $h(n)$, $0 \leq n \leq N-1$ линейно-фазового КИХ фильтра-прототипа симметрична, то фильтры анализа и синтеза M -канального косинусно-модулированного банка фильтров могут быть получены как

$$h_k(n) = 2h(n) \cdot \cos\left\{ \frac{(2k+1)\pi}{2M} \left(n - \frac{N-1}{2} \right) + (-1)^k \frac{\pi}{4} \right\}, \quad (14a)$$

$$f_k(n) = 2h(n) \cdot \cos\left\{ \frac{(2k+1)\pi}{2M} \left(n - \frac{N-1}{2} \right) - (-1)^k \frac{\pi}{4} \right\}, \quad (14b)$$

где $h_k(n)$ и $f_k(n)$, $0 \leq k \leq M-1$, $0 \leq n \leq N-1$ представляют собой импульсные характеристики фильтров анализа и синтеза соответственно.

Восстановленный сигнал $\hat{X}(z)$ может быть записан в терминах этих фильтров, как:

$$M\hat{X}(z) = X(z)T_0(z) + \sum_{k=1}^{M-1} X(z)T_k(z), \quad (15)$$

где

$$T_k(z) = \sum_{n=0}^{M-1} F_k(z)H_k(ze^{-j2n\pi/M}), \quad 0 \leq k \leq M-1, \quad (16)$$

и представляет компоненты наложения, а $T_0(z)$ – общую передаточную функцию банка фильтров.

Расчет фильтра-прототипа может быть представлен как решение проблемы взвешенной минимаксной оптимизации, которая в [19] формулируется следующим образом.

Минимизировать верхнюю границу модуля взвешенной ошибки ε , удовлетворяющую

$$\frac{|E(\omega)|}{\chi(\omega)} \leq \varepsilon, \quad \forall \omega \in [0, \frac{\pi}{2M}] \cup [\omega_s, \pi], \quad (17)$$

где весовая функция $\chi(\omega) > 0$, $\forall \omega \in [0, \pi/2M] \cup [\omega_s, \pi]$. Функция ошибки $E(\omega)$ определяется как

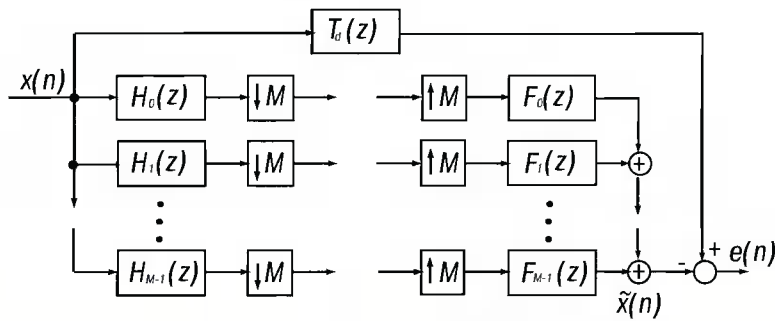


Рис. 4. Эквивалентная схема формирования ошибки восстановления

$$E(\omega) = \begin{cases} \sum_{k=0}^{M-1} \left| H\left[\omega + \frac{(2k+1)\pi}{2M}\right] + H\left[\frac{(2k+1)\pi}{2M} - \omega\right] \right|^2 - 1, & \omega \in \left[0, \frac{\pi}{2M}\right] \\ H(\omega), & \omega \in [\omega_s, \pi]. \end{cases} \quad (18)$$

Здесь $H(\omega)$ – Фурье-преобразование фильтра прототипа, а ω_s – граница зоны непрозрачности фильтра.

Целевая функция, которая позволит найти коэффициенты фильтра прототипа, минимизирующие функцию ошибки в смысле взвешенных наименьших квадратов, может быть записана как

$$f = \sum_{\omega \in [0, \pi/2M] \cup [\omega_s, \pi]} B(\omega) |E(\omega)|^2, \quad (19)$$

где $B(\omega)$ неотрицательная весовая функция.

Оптимальное решение, минимизирующее (19), в компактной форме достаточно трудно получить. Аналитическое выражение связи между $B(\omega)$ и $\gamma(\omega)$ также не известно. В [19] предложена итеративная процедура, которая позволяет минимизировать (19) во взвешенном минимаксном смысле. Существуют и другие подходы к задаче получения фильтра-прототипа.

Большинство из известных методов используются для одновременного расчета БФ анализа и синтеза. Несмотря на то что такой подход позволяет гораздо проще достигнуть полного восстановления, он не всегда подходит для решения задач построения БФ. Существует ряд приложений, для которых банк анализа проектируется с учетом заданной частотной избирательности (особенно это актуально при построении оптимальных субполосных кодеров.) Требование обеспечения заданных частотных характеристик не всегда может быть включено в процедуру одновременного синтеза банков квадратурно-зеркальных фильтров для достижения полного восстановления. Возможным путем решения данной проблемы может быть следующий: для рассчитанного БФ анализа, удовлетворяющего всем требованиям по частотной избирательности, найти БФ синтеза, который позволил бы достигнуть полного восстановления или минимизировал бы некоторую меру ошибки восстановления.

В [23–25] рассматривается проблема проектирования БФ синтеза, который сводит к нулю ошибку наложения (элайзинг) и минимизирует « H_∞ -норму» ошибки системы (рис. 4) для $T_d(z) = z^{-d}$ и $d > 0$. Такая « H_∞ -оптимизация» эквивалентна минимизации энергии ошибки восстановления сигнала $e(n) = \hat{x}(n) - x(n-d)$ для наилучшего на всем множестве сигналов с ограниченной энергией $x(n)$. Следует отметить, что

предлагаемые в этих работах алгоритмы позволяют строить БФ как с равномерным, так и с неравномерным разделением на каналы.

Как уже отмечалось выше, прямым путем получения фильтров синтеза является обращение матрицы компонент наложения:

$$F_{id}(z) = H_{AC}^{-1} \cdot [1 \ 0 \ \dots \ 0]^T = \frac{\text{adj}(H_{AC}) \cdot [1 \ 0 \ \dots \ 0]^T}{\det H_{AC}}. \quad (20)$$

Если ограничить $F(z)$ условием устойчивости и реализуемости, то задача построения БФ синтеза может быть представлена как поиск матрицы синтеза $F(z)$, которая бы минимизировала энергию сигнала ошибки для всех входных сигналов с ограниченной энергией и исключала ошибку наложения. Эта задача, в свою очередь, может быть сведена к эквивалентной « H_∞ -оптимизационной» задаче в частотной области, решение которой можно найти в выше указанных работах.

Можно показать, что если БФ анализа $H(z)$ устойчив и реализуем, то все устойчивые и реализуемые БФ синтеза $F(z)$, исключаящие элайзинг, могут быть представлены как

$$F(z) = G(z) (\det L)^{-1} \text{adj}(H_{AC}(z)) [1 \ 0 \ \dots \ 0]^T, \quad (21)$$

где $G(z)$ – некоторая устойчивая функция с действительными значениями для любого действительного z , L – постоянная матрица, определяемая как

$$L = [W^{k,l}]_{k=0..M-1, l=0..M-1}$$

Зададим $E(z) = Z(e(n))$, как Z -преобразование сигнала ошибки, тогда

$$\begin{aligned} E(z) &= T_E(z) X(z), \\ T_E(z) &= T_A(z) - T_{id}(z) = (1/M) \cdot (H(z)F(z) - cz^{-d}) = \\ &= (1/M) \cdot (G(z)z^{-(M-1)M/2} \det E(z^M) - cz^{-d}), \end{aligned} \quad (22)$$

где $E(z)$ – полифазная матрица БФ синтеза. Тот факт, что $T_E(z)$ – скалярная функция, ведет к более простой задаче оптимизации и более эффективному алгоритму расчета. Таким образом, необходимо найти устойчи-

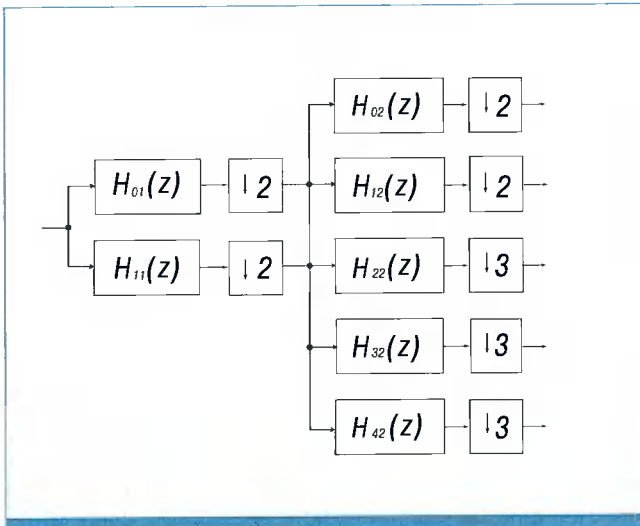


Рис. 5. Простой пример построения неравномерного БФ

вую и реализуемую передаточную функцию $G(z)$, которая бы минимизировала

$$\|T_E\|_{\infty} = \sup_{\omega} |T_E(e^{j\omega})| = \sup_{x \neq 0} \frac{\|e\|_2}{\|x\|_2}, \quad \|\cdot\|_2 = \sqrt{\sum_{n=-\infty}^{\infty} |n|^2}. \quad (23)$$

Таким образом, « H_{∞} -норма» $T_E(z)$ соответствует увеличению доли энергии, переходящей из входного сигнала $x(n)$ в сигнал ошибки $e(n)$ в наихудшем случае, а термин « H_{∞} -оптимизация» соответствует минимизации $\|T_E\|_{\infty}$ на всем множестве устойчивых и реализуемых передаточных функций $G(z)$. Решив эту задачу и найдя $G(z)$, как это предложено в вышеуказанных работах, при помощи (21) находим БФ синтеза $F(z)$.

Цифровые БФ с неравномерным разбиением на каналы

БФ с неравномерным разбиением могут строиться с использованием различных методик [26–31]. Самым простым и надежным подходом является использование древовидных структур (рис. 5). В этом случае, используя БФ с равномерным разбиением и полным восстановлением, можно получить достаточно широкий, но неполный класс неравномерных БФ с ПВ, в котором коэффициент понижения частоты дискретизации в каждом канале будет выражаться целым числом.

Очевидно, что такой подход не всегда приемлем и существуют приложения, в которых требуется большая свобода в выборе коэффициента децимации в каждом канале или, что равнозначно, относительной ширины частотной полосы. Например, для субполосной обработки речевых и аудиосигналов идеальное разложение должно покрывать критические частотные полосы, которые расположены неравномерно, и к ним не могут быть подобраны древовидные структуры банков фильтров.

ров. В связи с этим возникает необходимость построения БФ с рациональными коэффициентами децимации.

Следует отметить, что в отличие от равномерных БФ теория построения БФ с неравномерным разбиением на каналы проработана значительно слабее. Кроме того, существуют ограничения на набор коэффициентов децимации и расположение соответствующих субполос в спектре сигнала, которые позволяют исключить эффект наложения.

Неравномерные системы могут быть различных форм и структур [27]. Рассмотрим три основные структуры с целочисленными и рациональными коэффициентами децимации. В случае различных коэффициентов децимации v_i рис. 1 может иллюстрировать пример простой неравномерной структуры в случае, когда все коэффициенты децимации выражаются целыми числами. Так как мы рассматриваем только системы с максимальной децимацией, то эти коэффициенты $v_i, i=0..M-1$ связаны соотношением (1).

Как показано в [28], набор коэффициентов децимации не может быть произвольным. Во-первых, он должен удовлетворять так называемому «тесту на совместимость», который состоит в следующем. Пусть коэффициенты децимации выражаются различными целыми числами n_i , и каждое такое значение n_i встречается N_i раз. Определим положительные целые $p_i = \text{НОК}\{n_i\}/n_i$ и $m_j = \min_{i \neq j} (\text{НОК}\{p_i, p_j\}/p_j)$ (НОК – наименьшее общее кратное). Тогда необходимым условием полного восстановления будет тот факт, что $m_j - 1 < N_j$ для любого j .

Другим необходимым условием является тот факт, что среди коэффициентов децимации не должно быть ни одной пары взаимно простых коэффициентов. Это условие возникает из эквивалентности свойств полного восстановления и биортогональности

$$\langle H_i(z), F_j(z) \rangle_{g_{ij}} = \delta(i-j),$$

где g_{ij} – наименьший общий делитель (n_i, n_j) . Если два коэффициента n_i и n_j взаимно простые, то $g_{ij} = 1$, и фильтры $H_i(z)$ и $F_j(z)$ не могут быть одновременно реализуемы.

Более общая структура БФ с неравномерным разделением каналов показана на рис. 6. В этой системе коэффициенты децимации в каждом канале выражаются рациональным числом. Для обеспечения полного восстановления при максимальной децимации необходимо потребовать выполнение условия:

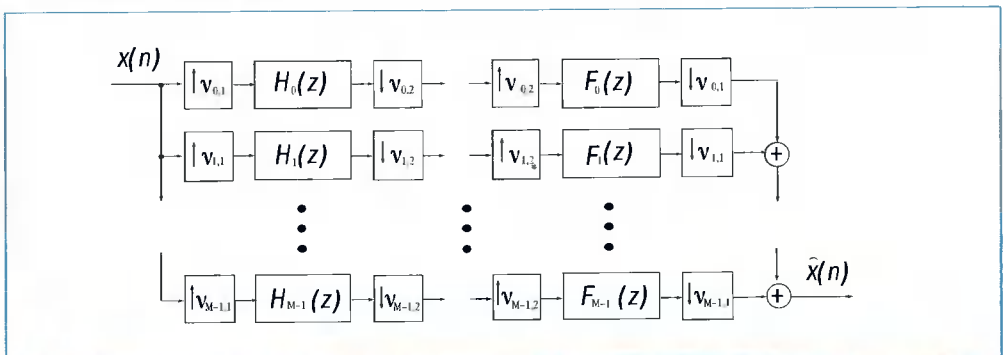


Рис. 6. Банк фильтров с неравномерным разделением на каналы и рациональными коэффициентами децимации

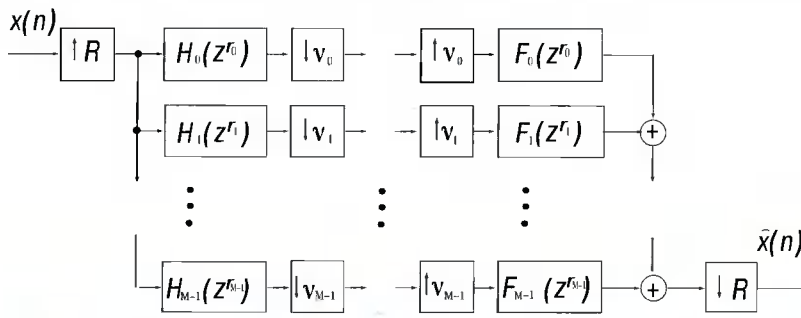


Рис. 7. Эквивалентная форма системы с рациональными коэффициентами децимации

$$\sum_{i=0}^{M-1} \frac{v_{i1}}{v_{i2}} = 1. \quad (24)$$

Следует отметить, что прямая реализация систем по данной структуре обычно не проводится. Существует альтернативная структура (рис. 7), связанная с нахождением наименьшего общего кратного R коэффициентов повышения частоты на этапе анализа $v_{i1}, i=0 \dots M-1$ и последующим вынесением экспандера частоты дискретизации за точку разделения на каналы. При этом новые коэффициенты децимации в каждом канале будут выражаться как $v_i = R(v_{i2}/v_{i1})$. Каждый фильтр анализа и синтеза подвергается преобразованию путем подстановки z^r вместо z , где $r_i = R/v_{i1}$. Подсистема синтеза дуальна системе анализа. Как видно из рис. 7, входной сигнал претерпевает увеличение частоты дискретизации в R раз и подается на систему анализа/синтеза с целыми коэффициентами децимации, после прохождения которой децимируется в R раз. Таким образом, для построения системы анализа/синтеза с рациональными коэффициентами децимации можно использовать теорию построения систем с целыми коэффициентами децимации. Очевидно, что при практической реализации данной системы увеличение вычислительных затрат из-за более высокой скорости работы банков фильтров не произойдет.

Другой вариант построения системы с рациональными коэффициентами децимации сводится к трансформации неравномерного БФ в L -канальный равно-

мерный БФ, где L – НОК коэффициентов децимации $L = \text{НОК}\{v_i\}$ (рис. 8). Каждый канал неравномерного БФ с дециматором v_k преобразуется в $p_k = L/v_k$ каналов равномерного БФ. Фильтры в каждой таком канале представляют собой задержанные версии исходного. При этом свойства полного восстановления и

максимальной децимации сохраняются.

Неравномерные системы также могут быть построены с использованием блочной децимации (рис. 9). В блочном дециматоре, определяемом как $v_2:v_1$, в каждом блоке размером v_2 отсчетов оставляют только первые v_1 отсчетов, а остальные удаляются. В этом случае также должно выполняться соотношение (24).

Как следует из вышеизложенного, неравномерное разбиение на каналы, хотя и дает большую свободу в выборе ширины каждой субполосной компоненты, требует соблюдения достаточно жестких ограничений. Однако применение обобщенных двухскоростных систем, основу которых составляет линейная периодически изменяющаяся во времени (ЛПИВ) система, а также компрессор и экспандер частоты дискретизации, может позволить обойти ряд таких ограничений [28, 29]. Методы расчета БФ с неравномерным разделением на каналы имеют примерно ту же классификацию, что и в случае равномерных БФ, за исключением того, что пока не найден способ построения БФ с ПВ и неравномерным разбиением на основе структурного синтеза полифазной матрицы системы. Известна работа по построению косинусно-модулированных БФ [30], а также работа по построению БФ с неравномерным разбиением на каналы на основе « H_∞ -оптимизации» [31].

Заключение

Отметим, что представленный тематический обзор не дает законченных рекомендаций и сравни-

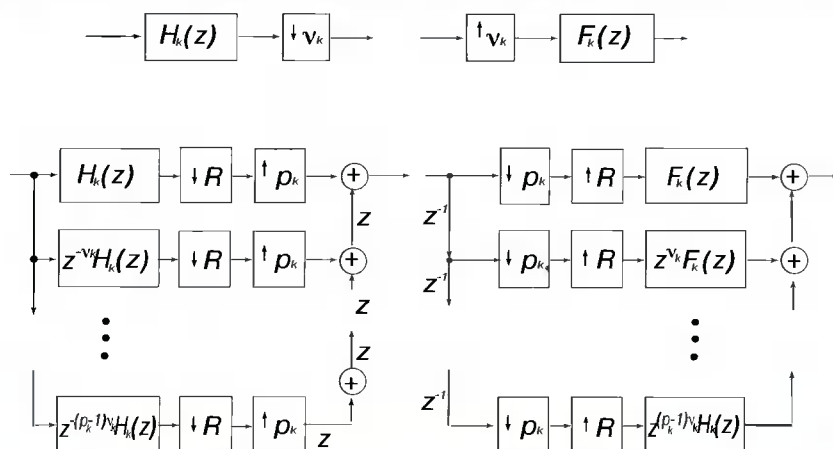


Рис. 8. Преобразование неравномерного БФ в эквивалентный равномерный

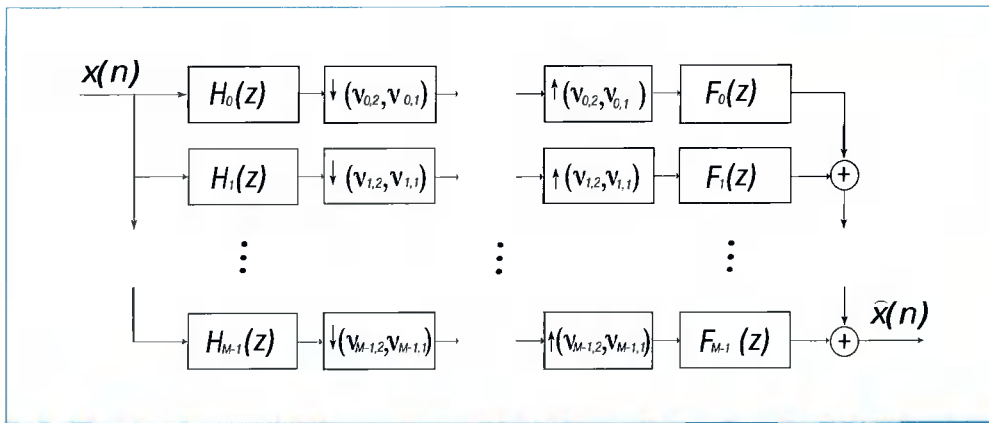


Рис. 9. Неравномерный банк фильтров с блочными дециматорами и интерполяторами

тельных характеристик. Обозначен лишь круг проблем, связанных с проектированием цифровых БФ, и намечены возможные подходы к их решению. Более подробные сведения о тех или иных методах построения БФ можно найти в многочисленных источниках,

использованных при подготовке настоящего обзора [1–31]. Кроме того, теория построения БФ не ограничивается только методами расчета коэффициентов фильтров. Большое число вопросов, связанных с построением оптимальных БФ в плане решения различных прикладных задач (таких, как субполосное кодирование, подавление шума, и др), осталось за рамками данной статьи. Ответы на эти вопросы дают бурно развивающиеся теория оптимального субполосного кодирования и теория построения банков фильтров основных компонент (Principal Component Filter Banks).

Литература

1. Crochiere R.E., Rabiner L. Multirate Digital Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1983.
2. Витязев В.В. Цифровая частотная селекция сигналов. М.: Радио и связь, 1993, 240 с.
3. Vaidyanathan P.P. Multirate Systems and Filter Banks, Prentice Hall, Englewood, NJ, 1993.
4. Вайдьянатхан П.П. Цифровые фильтры, блоки фильтров и полифазные цепи с многочастотной дискретизацией: Методический обзор // ТИИЭР, 1990. №3, с. 77–119.
5. Крот А.М., Кудрявцев В.О. Теория анализа и синтеза банк-фильтров и их применение // Зарубежная радиоэлектроника, 1999, №2, с. 3–17.
6. Vaidyanathan, P.P. Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property, IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-35, p. 476, April 1987.
7. Vaidyanathan, P.P. and Sanjit K. Mitra. Polyphase Networks, block Digital Filtering, LPTV Systems, and Alias-Free QMF Banks: A Unified Approach Based on Pseudocirculants. IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-36, p. 381, March 1988.
8. T. Q. Nguyen and P. P. Vaidyanathan, Maximally Decimated Perfect-Reconstruction FIR Filter Banks with Pairwise Mirror-Image Analysis (and Synthesis) Frequency Responses, IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-36, Number 5, p. 693, 1988.
9. Doganata, Z., Vaidyanathan, P.P., and Nguyen, T.Q. General synthesis procedures for FIR lossless transfer matrices, for perfect-reconstruction multirate filter bank applications, IEEE Trans. on Acoustics, Speech and Signal Proc., vol. ASSP-36, p. 1561, Oct. 1988.
10. Soman, A.K., and Vaidyanathan, P.P. Linear phase paraunitary filter banks: theory, factorizations, and applica-

tions, IEEE Trans. Signal Proc., vol. SP-41, p. 3480, December 1993.

11. L. Chen, K.-P. Chan, and T. Q. Nguyen A Generalized Algorithm for Linear-Phase Paraunitary Filter Banks IEEE Trans. Signal Proc., vol. 46, p. 1154, April 1998.

12. P. Saghizadeh and A. N. Willson, Jr. A New Approach to the Design of Critically Sampled M-Channel Uniform-Band Perfect-Reconstruction Linear-Phase FIR Filter Banks IEEE Trans. Signal Proc., vol 46, p. 1544, June 1998.

13. M. S. Spurbeck and C. T. Mullis Least Squares Approximation of Perfect Reconstruction Filter Banks IEEE Trans. Signal Proc., vol 46, p. 968, April 1998.

14. J. Huang and G. Gu A Direct Approach to the Design of QMF Banks via Frequency Domain Optimization, IEEE Trans. Signal Proc., vol 46, p. 2131, August 1998.

15. T. Chen and B.A. Francis, Design of multirate filter banks by Hinf optimization, IEEE Trans. Signal Proc., vol.SP-43, pp. 2822-2830, Dec. 1995.

16. R.D.Koilpillai, and P.P. Vaidyanathan, Cosine-modulated FIR filter banks satisfying perfect reconstruction, IEEE Trans. on Signal Processing, vol. SP-40, p. 770, April 1992.

17. Yuan-Pei Lin and P. P. Vaidyanathan, Linear phase cosine modulated maximally decimated filter banks with perfect reconstruction, IEEE Trans. Signal Proc., vol. SP-43, p. 2525, Nov. 1995.

18. T. Q. Nguyen & R. D. Koilpillai, The theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect reconstruction, IEEE Trans. Signal Proc., vol. SP-44, p. 473, March 1996.

19. H. Xu, W.S. Lu, and A. Antoniou, Efficient iterative design method for cosine-modulated QMF banks, IEEE Trans. Signal Proc., vol. SP-44, p. 1657, July 1996.

20. C.-K. Goh and Y.-C. Lim, An Efficient Algorithm for the Design of Weighted Minimax M-Channel Cosine-Modulated Filter Banks, IEEE Trans. Signal Proc., volume 46, p. 1426, May 1998.

21. A. Mertins, Subspace Approach for the Design of Cosine-Modulated Filter Banks with Linear-Phase Prototype Filter, *IEEE Trans. Signal Proc.*, vol 46, p. 2812, October 1998.

22. F. Argenti and E. Del Re, Design of Biorthogonal M-Channel Cosine-Modulated FIR/IIR Filter Banks, *IEEE Trans. Signal Proc.*, vol 48, p. 876, March 2000.

23. J. Huang and G. Gu, A Direct Approach to the Design of QMF Banks via Frequency Domain Optimization, *IEEE Trans. Signal Proc.*, vol 46, p. 2131, August 1998.

24. T. Chen and B.A. Francis, Design of multirate filter banks by Hinf optimization, *IEEE Trans. Signal Proc.*, vol. SP-43, p. 2822, Dec. 1995.

25. J. Huang, G. Gu, and Belle A. Shenoi, Design of multichannel QMF Bank via Frequency Domain Optimizations, *IEEE Trans. Circuits and Syst., Part II*, vol. 46, p.599, 1999.

26. J. Kovacevic, M. Vetterli, Perfect Reconstruction Filter Banks with Rational Sampling Factors, *IEEE Trans. Signal Proc.*, vol.41, p. 2047, June 1993.

27. K. Nayebi, T.P. Barnwell, and J.T. Spith, Nonuniform filter banks: A reconstruction and design theory, *IEEE Trans. Signal Proc.*, vol. SP-41, p. 1114, March 1993.

28. S. Akkarakaran and P.P. Vaidyanathan, New results and open problems on nonuniform filter-banks, New results and open problems on nonuniform filter-banks," *Proc. IEEE Int. Conf. Acoust. Speech, and Signal Proc.*, Phoenix, March 1999.

29. T.Chen, L. Qiu, E.Bai, General Multirate Building Structures with Application to Nonuniform Filter Banks, Special Issue on Multirate Systems, Filter Banks, Wavelets, and Applications, *IEEE Trans. on Circuits and Systems II:Analog and Digital Signal Processing*, vol.45, p.948, 1998.

30. J. Princen, The Design of nonuniform modulated filterbanks, *IEEE Trans. Signal Proc.*, vol. SP-43, p. 2550, Nov. 1995.

31. T. Chen, Nonuniform Multirate Filter Banks: Analysis and Design with an Hinf Performance Measure, *IEEE Trans. on Signal Processing*, vol. 45, p. 572, March, 1997.

ИНФОРМАЦИЯ

5-я Международная научно-техническая конференция и выставка «Цифровая обработка сигналов и ее применение – DSPA-2003»

Развитие современных информационных технологий характеризуется широким практическим использованием техники цифровой обработки сигналов (ЦОС) – одной из самых динамичных и быстроразвивающихся технологий в мире телекоммуникаций и информатизации общества. Методы и техника ЦОС вызывают повышенный интерес ученых и специалистов, работающих в таких областях, как связь и системы управления, радиотехника и электроника, акустика и сейсмология, радиовещание и телевидение, измерительная техника и приборостроение. Эффективное решение проблем информационных технологий реального времени дает мощный толчок к созданию и промышленному внедрению конкурентоспособных на мировом рынке средств телекоммуникаций, автоматического измерения и управления, приборостроения, бытовой радиоэлектронной аппаратуры, военных технологий.

Отвечая целям и задачам консолидации и поддержки отечественных ученых и специалистов, работающих в области передовых информационных технологий реального времени, расширения международных научных связей и содействия внедрению высоких наукоемких технологий XXI века, Российское НТОРЭС имени А.С. Попова совместно с российским отделением IEEE Signal Processing Society, Отделением информатики, вычислительной техники и автоматизации РАН, Институтом радиотехники и электроники РАН, Институтом проблем управления РАН, ФГУП НИИ радио, АО «АВТЭК» при поддержке Российского фонда фундаментальных исследований проводят в Москве 12–14 марта 2003 года 5-ю Международную научно-техническую конференцию и выставку «Цифровая обработка сигналов и ее применение – DSPA-2003».

Работа конференции будет проходить по десяти секциям, в частности:

- теория сигналов и систем;
- теория и методы цифровой обработки сигналов;
- обработка сигналов в системах телекоммуникаций;
- обработка сигналов в радиотехнических системах;
- обработка и передача изображений;
- цифровая обработка многомерных сигналов;
- обработка измерительной информации;
- нейрокompьютерная обработка сигналов и изображений;
- проектирование и техническая реализация систем ЦОС.

В обсуждении актуальных проблем цифровой обработки сигналов в рамках пленарного заседания и «круглого стола» примут участие: академик РАН Ю.В. Гуляев (директор ИРЭ РАН), академик РАН Н.А. Кузнецов (директор ИППИ РАН), профессор С. Митра (США), академик АН Грузии И.В. Прангишвили (директор ИПУ РАН), проф. Т. Сармаки (Финляндия), проф. И. Билинскис (Латвия), академик РАН А.С. Бугаев, профессор Т. Качорек (Польша), профессор А.А. Петровский (Белоруссия) и др.

Одновременно с конференцией пройдет выставка новых информационных технологий реального времени и инструментальных средств ЦОС. В работе выставки примут участие ведущие в этой области отечественные и зарубежные предприятия и фирмы.

В дни работы конференции и выставки будут организованы семинары-презентации по новым разработкам в области DSP компаний «Analog Devices», «Texas Instruments», «Xilinx», «РТСофт», «Скан Инжиниринг-телеком» и др.

Информационную поддержку Конференции и Выставки обеспечивают научно-технические журналы: «Цифровая обработка сигналов», «Chip News»; «Компоненты и технологии», «Радиотехника», «Электросвязь», «Схемотехника», «Электронные компоненты», «Датчики и системы».

УДК 621.391

Квазикогерентный прием многолучевого сигнала с прерывистым пилот-сигналом в мобильных системах связи

И.В. Каюков, В.Б. Манелис

Введение

В последние годы интенсивное развитие получили мобильные сотовые системы связи. Одним из требований к системам мобильной связи настоящего и следующих поколений является поддержание связи с абонентами, перемещающимися с высокой скоростью. Как следствие, мобильная система связи должна эффективно функционировать в условиях быстрых (высокочастотных) замираний.

Для достижения высокой помехоустойчивости и емкости мобильных сотовых систем связи 3-го и 4-го поколений предполагается использование M-PSK и M-QAM видов модуляции. Потенциальные преимущества этих видов модуляции могут быть реализованы только при высокой точности оценки комплексной огибающей принимаемого сигнала. Для этого полезный сигнал содержит пилот-компоненту. В последнее время широкое распространение получил прерывистый пилот-сигнал, когда пилот-символы мультиплексируются во времени с информационными символами. При этом передаваемый поток символов обычно сгруппирован по слотам, представляющим собой упорядоченную совокупность заданного числа информационных и пилот-символов. Алгоритмам обработки сигналов с прерывистым пилот-сигналом посвящено большое число публикаций [1–8 и др.].

В работах [1–4] оценка комплексной огибающей сигнала осуществляется только по пилот-символам. Идея описанных алгоритмов заключается в формировании оценки комплексной огибающей сигнала для каждой группы пилот-символов, а затем в их использовании для интерполяции комплексной огибающей каждого информационного символа. В [1] предлагается гауссовская интерполяция 0-го, 1-го и 2-го порядков. В [2–4] интерполяция комплексной огибающей производится посредством линейной фильтрации оценок комплексной огибающей пилот-символов. Импульсная характеристика фильтра может быть как фиксированной [2, 3], так и адаптивной к статистике канала связи [4].

Алгоритмы квазикогерентного приема, в которых оценка комплексной огибающей осуществляется только по пилот-символам, просты в реализации. Они позволяют получить качественную оценку комплексной огибающей сигнала в условиях низкочастотных замираний, однако становятся неработоспособными, когда интервал корреляции комплексной огибающей сигнала соизмерим или меньше расстояния между соседни-

Предложен адаптивный к частоте замираний алгоритм квазикогерентного приема многолучевого сигнала с прерывистым пилот-сигналом, который основан на фильтрации пилот-символов, адаптивном интерполировании комплексной огибающей сигналов лучей и итеративной процедуре с использованием информационных символов. Методом компьютерного моделирования выполнен анализ предложенного алгоритма. Показана высокая эффективность алгоритма по сравнению с альтернативными алгоритмами.

ми группами пилот-символов. Между тем информация о комплексной огибающей полезного сигнала содержится и на интервалах информационных символов. Использование этой информации в алгоритме оценки комплексной огибающей способно повысить его эффективность.

В [7, 8] представлены различные по сложности реализации алгоритмы квазикогерентного приема M-PSK сигналов, основанные на параметризации изменяющейся во времени комплексной огибающей сигнала путем разложения ее в ряд по некоторой системе функций. Оценка комплексной огибающей заключается в оценке параметров аппроксимирующего ряда. Для оценки комплексной огибающей используются информационные символы. Выполнены синтез и анализ алгоритма максимального правдоподобия (АМП), реализующего идею приема «в целом». Вычислительная сложность АМП резко возрастает с увеличением количества информационных символов на интервале анализа. Кроме того, представлено несколько алгоритмов, реализующих идею решающей обратной связи (РОС), когда для полиномиальной аппроксимации комплексной огибающей информационного символа используются оценки информационных параметров предыдущих символов. В случае быстрых замираний алгоритмы с РОС обладают низкой помехоустойчивостью. Это связано с тем, что оценка каждого информационного символа опирается на уже полученные оценки предыдущих символов. При этом ошибочно оцененные предыдущие символы увеличивают вероятность ошибки при оценке текущего символа. Описан итеративный алгоритм (ИА), основанный на последовательном взаимном улучшении оценки информационных параметров и оценки комплексной огибающей. ИА, так же, как и АМП, и алгоритмы восстановления несущей с РОС, предполагает полиномиальную аппроксимацию комплексной огибающей сигнала в пределах слота.

Недостаток алгоритмов, представленных в [7, 8], состоит в том, что не учитывается случайный характер изменения комплексной огибающей и ее статистические свойства. Между тем учет этих статистических

свойств должен улучшить качество квазикогерентного приема. Кроме того, в [7, 8] не решена проблема оптимизации длины аппроксимирующего ряда в зависимости от частоты замираний.

Из алгоритмов, учитывающих статистические свойства канала, отметим алгоритмы из [5, 6]. В [5] представлен алгоритм ESAE (Extended Symbol Aided Estimation), основанный на стохастической интерполяции комплексной огибающей. Для оценки комплексной огибающей информационных символов текущего слота используются результаты усреднения групп пилот-символов предыдущего, текущего и последующих слотов, а также оценки информационных символов предыдущего слота.

В [6] предложен усовершенствованный алгоритм ESAE. В соответствии с этим алгоритмом оценка комплексной огибающей осуществляется в три этапа. На первом этапе осуществляется предварительная оценка комплексной огибающей информационных символов текущего слота в соответствии с описанным выше алгоритмом ESAE. Полученная оценка комплексной огибающей используется для предварительной оценки информационных символов текущего слота. На втором этапе аналогичным образом производится оценка комплексной огибающей информационных символов последующего слота. Полученные при этом оценки комплексной огибающей используются для предварительной оценки информационных символов последующего слота. На третьем этапе находят окончательные оценки комплексной огибающей информационных символов текущего слота, для чего используются результаты усреднения групп пилот-символов предыдущего, текущего и последующих слотов, а также оценки информационных символов предыдущего слота и предварительные оценки информационных символов последующего слота. Полученные при этом окончательные оценки комплексной огибающей используются для окончательной оценки информационных символов текущего слота. Описанный алгоритм показал хорошее качество приема для частот замираний, не превышающих 108 Гц (что при частоте несущей 1,5 ГГц соответствует скорости движения абонента 75 км/ч) при частоте следования символов 4800 символ/с и расстоянии между соседними группами пилот-символов 3,3 мс. Ухудшение помехоустойчивости алгоритмов [5, 6] при более быстрых замираниях связано с тем, что используемые результаты усреднения групп пилот-символов соседних слотов становятся слабо коррелированными. Этот фактор снижает качество интерполяции комплексной огибающей.

В данной работе предложен алгоритм квазикогерентного приема многолучевого сигнала, эффективный как при низкочастотных, так и при высокочастотных замираниях, который основан на фильтрации пилот-символов, адаптивном интерполировании комплексной огибающей сигналов лучей и итеративной процедуре с использованием информационных символов.

Модель принимаемого сигнала

Пусть наблюдается реализация суммы полезного сигнала и шума

$$x(t) = s(t, A, \psi, b^{(d)}, a^{(d)}) + n(t), \quad (1)$$

где $n(t)$ – белый гауссовский шум с односторонней спектральной плотностью N_0 , $\langle n(t_1)n(t_2) \rangle = (N_0/2)\delta(t_1 - t_2)$, $\delta(\cdot)$ – дельта-функция,

$$s(t, A, \psi, b^{(d)}, a^{(d)}) = \sum_{j=1}^p \sum_{i=1}^N A_{j,i} [b_i^{(d)}(t - \tau_j) \cos(\omega_0 t - \psi_{j,i}) + a_i^{(d)}(t - \tau_j) \sin(\omega_0 t - \psi_{j,i})] \quad (2)$$

полезный многолучевой сигнал, содержащий p лучей, временные положения которых $\tau_j, j=1 \dots p$; ω_0 – несущая частота, $A = \{A_{j,i}\}$, $\psi = \{\psi_{j,i}\}, j=1 \dots p, i=1 \dots N$ – амплитуда и фаза сигналов лучей, приближенно постоянные на интервалах каждого элементарного символа $\Delta_{j,i}$, длина каждого интервала равна Δ ; N – количество символов, укладываемых на рассматриваемом интервале; $l_i(t)$, $Q_i(t)$ – псевдослучайные последовательности l и Q каналов i -го символа, причем $l_i(t - \tau_j) = 0$, $Q_i(t - \tau_j) = 0$, если $t \notin \Delta_{j,i}$; $b_i, a_i, i=1 \dots N$ – параметры символов, среди которых имеются неизвестные информационные параметры $b^{(d)} = \{b_i^{(d)}\}$, $a^{(d)} = \{a_i^{(d)}\}, i=1 \dots N_d$ (возможные значения величин $b_i^{(d)}, a_i^{(d)}, i=1 \dots N_d$ определяются видом модуляции), а также известные параметры $b_i^{(p)}, a_i^{(p)}, i=1 \dots N_p$, соответствующие пилот-символам; $N_d + N_p = N$.

Информационный и пилот-сигнал разделены во времени, например, как показано на рис. 1.

По наблюдаемым данным (1) необходимо вынести решение о векторах информационных параметров $b^{(d)}, a^{(d)}$.

Правило вынесения решения об информационных параметрах

Для оценки неизвестных информационных параметров сигнала (1) будем использовать метод максимального правдоподобия. В этом случае решение принимается в пользу таких значений параметров, при которых функционал отношения правдоподобия или его монотонная функция принимает наибольшее значение. При наличии оценок комплексных огибающих информационных символов сигналов лучей $\hat{A}_{c,j,i} = \hat{A}_{j,i} \cos \psi_{j,i}, \hat{A}_{s,j,i} = \hat{A}_{j,i} \sin \psi_{j,i}, j=1 \dots p, i=1 \dots N$ логарифм функционала отношения правдоподобия (ЛФОП) принимаемого сигнала с точностью до несущественных слагаемых имеет вид

$$\Omega(b^{(d)}, a^{(d)}) = \frac{2}{N_0} \int_0^T |x(t)s(t, b^{(d)}, a^{(d)})|^2 dt - \frac{1}{N_0} \int_0^T |s(t, b^{(d)}, a^{(d)})|^2 dt \quad (3)$$

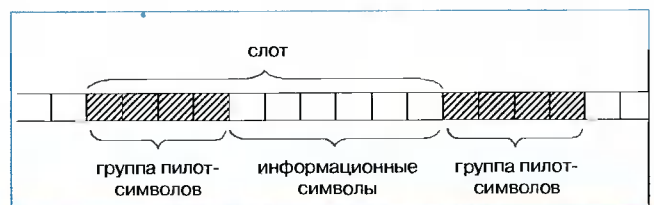


Рис. 1

Подставив (2) в (3), получим

$$\Omega(b^{(d)}, a^{(d)}) = \sum_{i=1}^{N_d} \Omega_i(b_i^{(d)}, a_i^{(d)}), \quad (4)$$

где

$$\Omega_i(b_i^{(d)}, a_i^{(d)}) = \frac{2}{N_0} [b_i^{(d)} \sum_{j=1}^p C_{j,i} + a_i^{(d)} \sum_{j=1}^p D_{j,i} - (b_i^{(d)2} + a_i^{(d)2}) \frac{\Delta}{4} \sum_{j=1}^p \bar{A}_{j,i}^2] \quad (5)$$

ЛФОП неизвестных параметров i -го информационного символа. В (5)

$$\begin{aligned} C_{j,i} &= \bar{A}_{c_{j,i}} X_{j,i} + \bar{A}_{s_{j,i}} Y_{j,i}, \\ D_{j,i} &= \bar{A}_{c_{j,i}} Y_{Q_{j,i}} - \bar{A}_{s_{j,i}} X_{Q_{j,i}}, \end{aligned} \quad (6)$$

мягкие решения информационных символов лучей, а величины $X_{j,i}, Y_{j,i}, X_{Q_{j,i}}, Y_{Q_{j,i}}$ определяются как

$$\begin{aligned} X_{j,i} &= \int_{\Delta_{j,i}} x(t) I_{\lambda}(t - \tau_j) \cos(\omega_0 t) dt, \\ Y_{j,i} &= \int_{\Delta_{j,i}} x(t) I_{\lambda}(t - \tau_j) \sin(\omega_0 t) dt, \\ X_{Q_{j,i}} &= \int_{\Delta_{j,i}} x(t) Q_{\lambda}(t - \tau_j) \cos(\omega_0 t) dt, \\ Y_{Q_{j,i}} &= \int_{\Delta_{j,i}} x(t) Q_{\lambda}(t - \tau_j) \sin(\omega_0 t) dt, \quad j=1 \dots p, i=1 \dots N_d \end{aligned} \quad (7)$$

Из (4) следует, что вероятностные меры для различных информационных символов не зависят друг от друга и решения по ним принимаются независимо. Для M-PSK видов модуляции $b^{(d)2} + a^{(d)2} = 1$, и последнее слагаемое в (5) является несущественным.

Для эффективной оценки информационных параметров требуется высокая точность оценки комплексной огибающей сигналов лучей $\bar{A}_{c_{j,i}}, \bar{A}_{s_{j,i}}, j=1 \dots p, i=1 \dots N_d$.

Оценка комплексной огибающей сигналов лучей

В предлагаемом алгоритме оценка комплексной огибающей сигналов лучей осуществляется для каждого слота последовательно в несколько этапов.

На **первом этапе** оценка комплексной огибающей для каждого луча определяется только по пилот-символам, для чего производятся следующие операции.

1. Определяются оценки комплексной огибающей пилот-символов для каждого луча

$$\begin{aligned} \tilde{A}_{c_i} &= \frac{2(b_i^{(p)} X_i + a_i^{(p)} Y_i)}{b_i^{(p)2} + a_i^{(p)2}}, \\ \tilde{A}_{s_i} &= \frac{2(b_i^{(p)} Y_{Q_i} - a_i^{(p)} X_{Q_i})}{b_i^{(p)2} + a_i^{(p)2}}, \quad i=1 \dots N_p. \end{aligned} \quad (8)$$

Оценки (8) являются оценками максимального правдоподобия комплексной огибающей пилот-символов сигналов лучей. Величины $X_{j,i}, Y_{j,i}, X_{Q_{j,i}}, Y_{Q_{j,i}}$ для пилот-символов в (8) определяются аналогично (7). Заметим, что в (8) и далее для упрощения выражений опущен индекс, соответствующий номеру луча.

2. Для подавления шумов полученные в соответствии с выражением (8) оценки комплексной огибающей пилот-символов для каждого луча фильтруются.

Профильтрованная оценка комплексной огибающей на интервале n -го пилот-символа каждого луча представляет собой результат весового суммирования в скользящем окне оценок комплексной огибающей пилот-символов

$$\bar{A}_{c_n} = \sum_{m=L}^{L-1} \tilde{A}_{c_{m+n}} R(\tau_m) / C_1, \quad \bar{A}_{s_n} = \sum_{m=L}^{L-1} \tilde{A}_{s_{m+n}} R(\tau_m) / C_1. \quad (9)$$

В (9) $2L$ – число символов в скользящем окне, $R(\tau)$ – корреляционная функция комплексной огибающей входного сигнала, τ_m – временное положение m -го пилот-символа относительно положения профильтрованного n -го пилот-символа,

$$C_1 = \sqrt{\sum_{m=L}^{L-1} \sum_{m_2=L}^{L-1} R(\tau_{m_1}) R(\tau_{m_2}) R(\tau_{m_1} - \tau_{m_2})} \quad (10)$$

нормировочный коэффициент, необходимый для выравнивая энергии дискретных процессов до и после фильтрации [9]. Выравнивание энергии необходимо, чтобы избежать ошибки в оценке амплитуды в случае использования амплитудных или амплитудно-фазовых видов модуляции, например M-QAM.

На рис.2 иллюстрируется нахождение весовых коэффициентов фильтрации $R(\tau_m), m=L \dots L-1$ для n -го пилот-символа.

3. Метод адаптивной стохастической интерполяции [6] (по методу наименьших средних квадратов) по профильтрованным оценкам комплексной огибающей пилот-символов $\bar{A}_c = \{\bar{A}_{c_n}\}, \bar{A}_s = \{\bar{A}_{s_n}\}, n=1 \dots J$ (J – количество используемых профильтрованных оценок комплексной огибающей пилот-символов при интерполяции слота) формируется предварительная оценка комплексной огибающей информационных символов слота

$$\bar{A}_{c_n} = \bar{A}_c^T K^{-1} r, \quad \bar{A}_{s_n} = \bar{A}_s^T K^{-1} r, \quad (11)$$

где $r = \{R(\tau_n - \tau_i)\}, i=1 \dots J, K = \{R(\tau_m - \tau_j)\}, m, j=1 \dots J$. Адаптивная интерполяция (11) предполагает измерение статистических свойств канала, а именно, автокорреляционной функции $R(\tau_m - \tau_j) = \langle A_d(\tau_m) A_d(\tau_j) \rangle = \langle A_s(\tau_m) A_s(\tau_j) \rangle$. Оценка автокорреляционной функции определяет значения K, r и может производиться одним из известных методов, например как в [10].

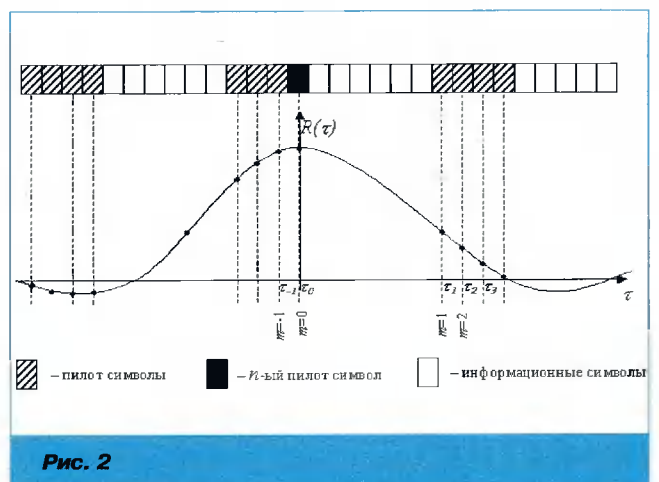
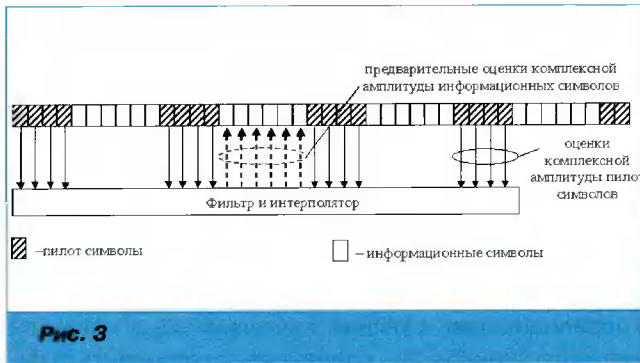


Рис. 2



Таким образом, на первом этапе оценка комплексной огибающей каждого информационного символа определяется с помощью интерполяции профильтрованных значений оценок комплексной огибающей пилот-символов, как показано на рис.3.

4. Полученные в результате интерполяции предварительные оценки комплексной огибающей \hat{A}_{c_n} , \hat{A}_{s_n} информационных символов подставляются в (6) для получения предварительных мягких решений информационных символов.

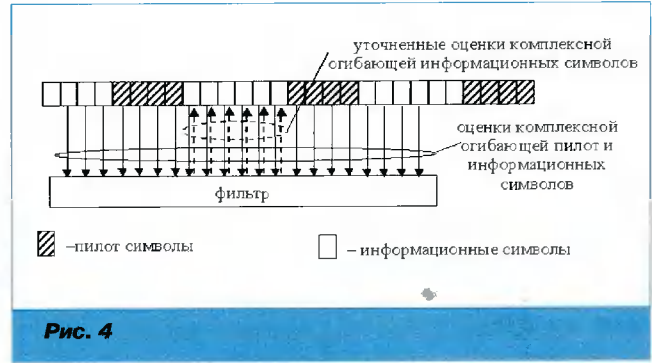
Второй этап является в общем случае итеративным. Оценка комплексной огибающей информационных символов сигналов лучей осуществляется с использованием как пилот-символов, так и информационных символов. При этом на каждой итерации производятся следующие операции.

1. Предварительные оценки комплексной огибающей информационных символов (11) и предварительные мягкие решения информационных символов (6) подставляются в (5), и оценка информационных параметров каждого символа слота определяется по положению максимума (5)

$$\hat{b}_n^{(d)}, \hat{a}_n^{(d)} = \operatorname{argmax} \Omega_n(b_n^{(d)}, a_n^{(d)}). \quad (12)$$

2. Формируются уточненные оценки комплексной огибающей информационных символов

$$\begin{aligned} \tilde{A}_{c_n} &= \frac{2(\hat{b}_n^{(d)} X_{i_n} + \hat{a}_n^{(d)} Y_{i_n})}{\hat{b}_n^{(d)2} + \hat{a}_n^{(d)2}}, \\ \tilde{A}_{s_n} &= \frac{2(\hat{b}_n^{(d)} Y_{Q_n} - \hat{a}_n^{(d)} X_{Q_n})}{\hat{b}_n^{(d)2} + \hat{a}_n^{(d)2}}. \end{aligned} \quad (13)$$



3. Для подавления шумов производится фильтрация уточненных оценок комплексной огибающей информационных символов (13) и оценок комплексной огибающей пилот-символов (8). В результате получим улучшенные оценки комплексной огибающей информационных символов. Для n -го информационного символа слота уточненная оценка комплексной огибающей имеет вид

$$\hat{A}_{c_n} = \sum_{m=-M}^M \tilde{A}_{c_{m+n}} R(\tau_m) / C_2, \quad \hat{A}_{s_n} = \sum_{m=-M}^M \tilde{A}_{s_{m+n}} R(\tau_m) / C_2. \quad (14)$$

В (14) $2M+1$ – число символов в скользящем окне фильтрации.

$$C_2 = \sqrt{\sum_{m1=-M}^M \sum_{m2=-M}^M R(\tau_{m1}) R(\tau_{m2}) R(\tau_{m1} - \tau_{m2})}$$

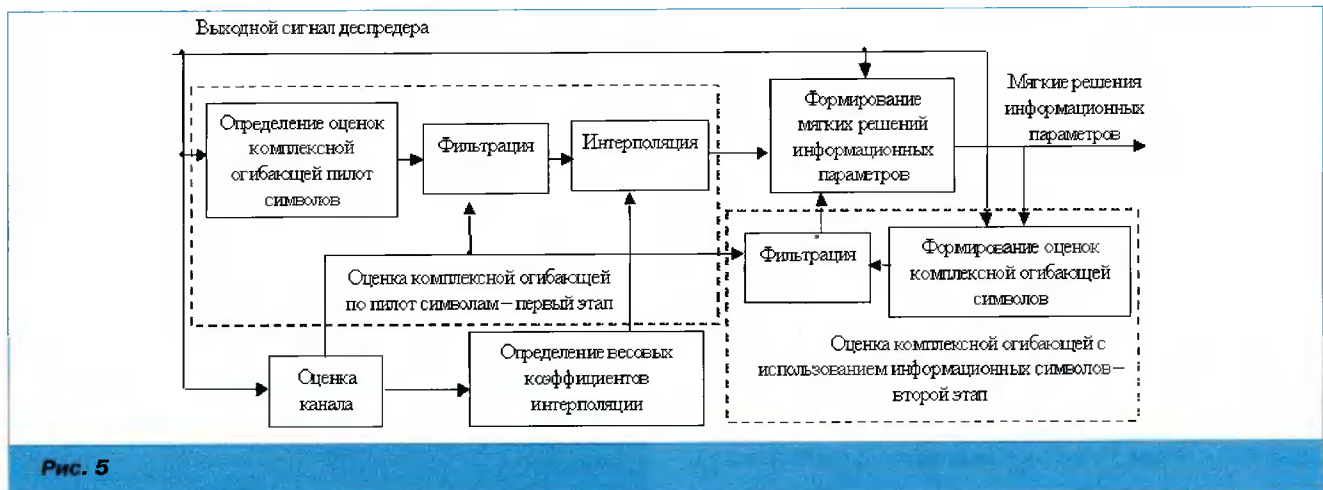
нормировочный коэффициент, необходимый для выравнивая энергии дискретных процессов до и после фильтрации (14).

Рис.4 иллюстрирует процесс получения уточненных оценок комплексной огибающей информационных символов слота на втором этапе.

4. Полученные в результате фильтрации улучшенные оценки комплексной огибающей \hat{A}_{c_n} , \hat{A}_{s_n} информационных символов подставляются в (6) для получения уточненных мягких решений информационных символов.

На каждой итерации второго этапа используются оценки комплексной огибающей и мягкие решения информационных символов, полученные на предыдущей итерации.

На рис.5 представлена упрощенная блок-схема устройства, реализующего предлагаемый алгоритм.



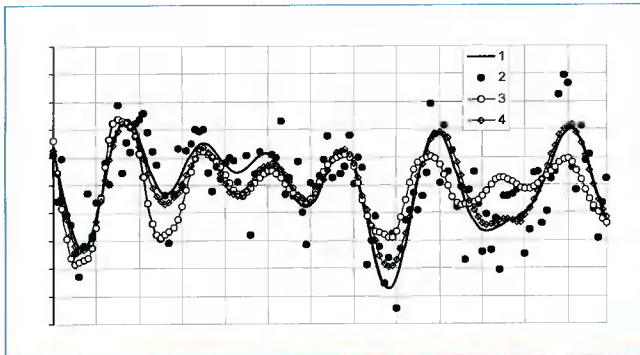


Рис. 6

Результаты моделирования

Анализ помехоустойчивости предлагаемого алгоритма квазикогерентного приема был выполнен методом компьютерного моделирования. Принимаемый сигнал представлял собой последовательность слотов (групп символов) длительностью 6,25 мс, каждый из которых содержал 6 информационных QPSK-символов и 4 пилот-символа. Замирания сигнала соответствовали модели Джейкса [11].

На рис.6 представлен пример оценки квадратуры комплексной огибающей принимаемого полезного сигнала на различных этапах работы алгоритма. Произведение частоты замираний F на длительность символа Δ выбиралось равным $FD=0,055$, что при частоте несущей 1,8 ГГц соответствует скорости движения абонента 500 км/ч. На рис.6 кривая 1 соответствует истинному значению квадратуры комплексной огибающей, кривая 2 – зашумленной квадратуре комплексной огибающей, кривая 3 – предварительной оценке квадратуры комплексной огибающей, полученной на пер-

вом этапе, кривая 4 – оценке квадратуры комплексной огибающей, полученной на втором этапе. Из рис.6 видно, что использование информационных символов позволяет резко повысить точность оценки комплексной огибающей.

На рис. 7, 8 приведены зависимости вероятности ошибочного приема двоичных символов предлагаемого алгоритма от отношения сигнал-шум. На рис. 7 приведены кривые, полученные при $FD=0,055$. Для сравнения приведены также характеристики квазиоптимального алгоритма с решающей обратной связью (КАРОС), итеративного алгоритма (ИА) и алгоритма максимального правдоподобия (АМП) [7,8]. Из рис. 7 видно, что помехоустойчивость предлагаемого алгоритма не уступает помехоустойчивости сложного для реализации алгоритма АМП, а выигрыш по сравнению с более простым алгоритмом КАРОС составляет 4–5 дБ. Предлагаемый алгоритм обладает быстрой сходимостью, так что при его реализации достаточно одной-двух итераций второго этапа. На рис. 7, 8 представлен вариант двух итераций второго этапа.

Кривые рис. 8 получены для различных значений частоты замираний F . Кривая 1 соответствует случаю $FD=0,055$, кривая 2 – $FD=0,037$, кривая 3 – $FD=0,019$. Видно, что при частоте несущей 1,8 ГГц и скорости движения абонента менее 300 км/ч характеристики предлагаемого алгоритма близки к характеристикам когерентного приема.

Заключение

В данной работе предложен адаптивный к частоте замираний алгоритм квазикогерентного приема многолучевого сигнала с прерывистым пилот-сигналом.

Предлагаемый алгоритм реализуется в два этапа. На первом этапе формируется предварительная оценка комплексной огибающей информационных символов посредством адаптивной интерполяции предварительно отфильтрованных значений оценок комплексной огибающей пилот-символов. Второй этап представляет собой в общем случае итеративную процедуру с использованием как пилот-символов, так и информационных символов, когда оценка информационных параметров и оценка комплексной огибающей последовательно уточняют друг друга. На каждой итерации второго этапа оценка комплексной огибающей информационных символов сигналов лучей осуществляется с помощью адаптивной фильтрации оценок комплексной огибающей информационных и пилот-символов, полученных на предыдущей итерации. Использование информационных символов для оценки комплексной огибающей позволяет в условиях быстрых замираний существенно увеличить точность оценки комплексной огибающей сигналов лучей и, как следствие, заметно уменьшить вероятность ошибки оценки информационных параметров.

Проведенный анализ показал высокую эффективность предлагаемого алгоритма по сравнению с альтернативными алгоритмами. Высокие точностные характеристики алгоритма в сочетании с относительной

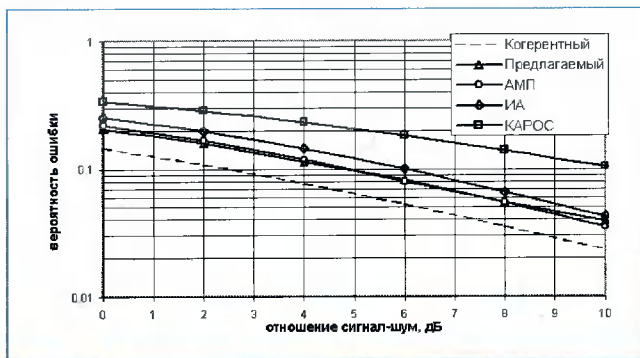


Рис. 7

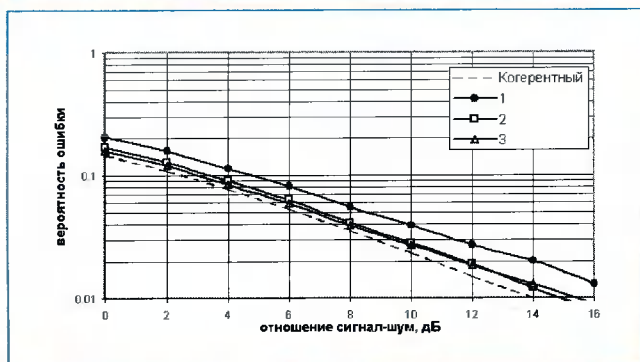


Рис. 8

простотой реализации делают его перспективным для использования в мобильных системах связи 3-го и 4-го поколений.

Литература

1. Sampei S., Sunaga T., Rayleigh fading compensation for QAM in land mobile radio communications, //IEEE Trans. Veh. Technol. – May 1993 – Vol. 42. – P.137-146,.
2. Andoh H., Sawahashi M., Adachi F., Channel Estimation Filter Using Time-Multiplexed Pilot Channel for Coherent RAKE Combining in DS-CDMA mobile radio //IEICE Trans. Commun. – July 1998 – Vol.E81-B. – №7. – P.1517-1526.
3. Abeta S., Sawahashi M., Adachi F. Performance comparison between time-multiplexed pilot channel and parallel pilot channel for coherent rake combining in DS-CDMA mobile radio //IEICE Trans. Commun. – July 1998 – Vol.E81-B. – №7. – P.1417-1425,.
4. Abeta S., Sawahashi M., Adachi F. Adaptive channel estimation for coherent DS-CDMA mobile radio using time-multiplexed pilot and parallel pilot structures //IEICE Trans. Commun. – Sept. 1999 – Vol.E82-B. – №9, – P.1505-1513,.
5. Nam L., Sakaniwa K. Extended symbol-aided estimation for non-selective Rayleigh fading channels //IEICE Trans. Fundam. – Nov. 1997 – Vol.E80-A. – №11 – P.2144-2154.
6. Genko K., Nam L., Sakaniwa K. Improvement of extended symbol-aided estimation for Rayleigh fading channels //IEICE Trans. Fundam. – Oct. 1998 – Vol.E81-A. – №10 – P.2073-2082,.
7. Алгоритмы квазикогерентного приема фазоманипулированных сигналов в канале с быстрым федингом / А. В. Гармонов, Ю.Е. Карпитский, В.Б. Манелис, А.Ю Савинков // Цифровая обработка сигналов, 2001, № 3, с. 2–8.
8. Квазикогерентный прием фазоманипулированных сигналов в канале с быстрым федингом/ А. В. Гармонов, Ю.Е. Карпитский, В.Б. Манелис, А.Ю Савинков // 7-я международная конференция и выставка «Цифровая обработка сигналов и ее применение», Москва, 2000, т.3, с. 123–128.
9. Левин Б.Р. Теоретические основы статистической радиотехники. М.: Радио и связь, 1989.
10. Каюков И. В., Манелис В. Б. Оценка канала распространения в мобильных системах связи // 7-я международная научно-техническая конференция «Радиолокация, навигация, связь», Воронеж, 2001, т.2, с. 958–965.
11. Связь с подвижными объектами в диапазоне СВЧ. /Под ред. У. К. Джейкса, М.: Связь, 1979.

Уважаемые читатели!

Со 2-го полугодия 2003 года вы можете подписаться на наш журнал в любом отделении связи. Подписной индекс издания по каталогу ОАО «Роспечать» – 82185.

Кроме того, сохраняется возможность оформить подписку непосредственно в редакции. Для этого необходимо заполнить подписной лист и отправить его по факсу (095) 105-3468 или по электронной почте info@dspa.ru

Подписной лист

Наименование организации _____

ИНН _____

Почтовый адрес _____

Контактное лицо _____

Телефон, факс, электронная почта _____

Банковские реквизиты

Наименование банка _____

Р/с _____

К/с _____

Город _____

БИК _____

Справки по телефонам: (095) 105-3464, 105-3468, 267-0662 или по электронной почте info@dspa.ru

УДК 621.397.2

Применение морфологических преобразований для пирамидального кодирования изображений

Н.Г. Харатишвили, И.М. Чхеидзе, З.Дж. Гогилашвили

Введение

Анализ методов пирамидального кодирования [1] показывает, что в них используются не только специфика, присущая многомерным сигналам с характерным наличием сильной пространственно-временной корреляционной связи между отсчетами, но и свойства зрительной системы человека, способной разделять анализируемое изображение на подполосы с различным частотным диапазоном и определенной пространственной ориентацией и проявлять меньшую чувствительность к восприятию высокочастотных составляющих [2].

В результате пирамидального преобразования исходное изображение представляется набором изображений, характеризующихся различным понижающимся «разрешением» (с повышением номера пирамиды) и сокращением числа элементов в 2 раза по каждой координатной оси. Каждый уровень пирамиды, кроме низкочастотной его вершины, представляет собой высокочастотное по своей природе изображение, с числом элементов, убывающим с повышением номера уровня. Ввиду слабой чувствительности глаза к высокочастотным составляющим эту информацию можно закодировать значительно меньшим числом битов на отсчет, чем исходное изображение. Низкочастотное изображение, несущее основную информацию об изображении, содержит значительно меньшее число элементов.

Пирамидальные построения могут быть ортогонального и неортогонального типа [2], отличаются между собой видом структурной схемы, по которой строится пирамида и видом фильтрации. Ортогональное построение осуществляется структурной схемой прямого преобразования разомкнутого типа, что и обуславливает полное частотное разделение.

Неортогональным построениям соответствуют структурные схемы замкнутого типа с отрицательной обратной связью, а сам процесс кодирования может быть рассмотрен адекватным процессу предсказания. Пирамида неортогонального типа строится на разности двух гауссоподобных двумерных весовых функций фильтра, что обуславливает хорошее приближение к импульсному отклику, представляющему собой явление латерального торможения в зрительной системе человека [2].

Пирамидальные построения являются обратимой операцией, допускающей полное восстановление, если отказаться от сжатия, ради которого осуществлялось это построение. Каждый уровень пирамиды строится по

В статье рассмотрены вопросы эффективности применения основных операций математической морфологии в пирамидальных построениях с нелинейной фильтрацией и приведены результаты экспериментальных исследований шумоподавляющих свойств морфологических фильтров, статистических свойств уровней морфологических пирамид и показана эффективность применения в пирамидальном кодировании пирамиды с модифицированной структурой построения ее уровней.

рекуррентному алгоритму, в котором используются операции децимации, интерполяции и фильтрации [1, 3].

Фильтрация, с применением которой осуществляется формирование каждого уровня пирамиды, может быть линейного или нелинейного типа. К нелинейной фильтрации можно отнести медианную, рангово-порядковую и морфологическую фильтрации [3]. Нелинейная фильтрация может оказаться в определенных условиях предпочтительней линейной фильтрации, поскольку при ее использовании сохраняются резкие перепады (контуры), линейная же фильтрация их смазывает [2].

Морфологическая фильтрация занимает особое место среди нелинейных преобразований, она сравнительно просто реализуется.

Принципы математической морфологии широко используются для анализа изображений [3] в биологических исследованиях, электронной микроскопии и приобретают все больший интерес в приложениях, связанных с машинным зрением в области автоматизированного контроля и т.д.

При исследовании кодирования с использованием любых методов сжатия эффективность кодирования источника информации оценивается коэффициентом сжатия. Величина коэффициента сжатия и качество восстановленного изображения (например, визуально воспринимаемые его характеристики) – два важнейших параметра, характеризующих эффективность обработки [4].

В реальных условиях обработки и передачи информации всегда присутствуют помехи: шумы различной природы, случайные ошибки. Под их воздействием в восстановленную информацию вносится определенная дополнительная погрешность, ухудшающая качество воспроизведенной информации, а следовательно, понижающая эффективность кодирования.

Влияние помех в системах сжатия данных обычно рассматривается отдельно на различных этапах преобразования сообщений [4]: как воздействие помех на изображения, подвергаемые обработке, так и воздействие помех на сжатые данные в процессе их последующей передачи и преобразования. Решение первой задачи в случае пирамидального кодирования связано с исследованием

шумоподавляющих свойств фильтров, используемых при построении уровней пирамиды, в то время как вторая задача рассматривает действие случайных ошибок на входе декодера и вытекающее из этого понижение точностных показателей восстановленных изображений.

Исследованию пирамидального кодирования всегда предшествует проведение анализа характеристик и свойств уровней и структуры сформированной пирамиды, шумоподавляющих свойств фильтров, на основании которых формируется пирамида, так как эффективность кодирования определяется тем, насколько результат пирамидального преобразования согласуется со свойствами зрительной системы человека и вносимые в кодируемую информацию искажения при значительных коэффициентах сжатия мало повлияли на качество восстановленных изображений.

1. Основные операции математической морфологии и принципы построения морфологической пирамиды

Вначале рассмотрим случай двоичных изображений. Обозначим через X множество всех точек объекта, которым условно припишем значения 1.

Для определения основных морфологических операций рассмотрим двумерное (2D) пространство E^2 , частично покрытое множеством $X (X \subseteq E^2)$.

Пусть $B \subseteq E^2$ – множество малого размера и простой формы, так называемый структурирующий элемент. Пусть $x \ominus b = \{x \pm b \mid x \in X\}$, где через x – обозначен элемент X , а через b – элемент множества B .

В [3] вводится понятие отраженного элемента \bar{B} , определяемого как $\bar{B} = \{-b \mid b \in B\}$ и означающего, что \bar{B} – симметричное относительно начала B множество, все элементы которого имеют значения элементов множества B , но с обратным знаком; X_{-b} – означает перемещение всех точек множества X на позицию $-b$, или – на вектор $(-b)$; B_x – означает перемещение всех элементов множества B на вектор x .

Фундаментальными морфологическими операциями являются «наращение» (dilation), обозначаемая \oplus , и «эрозия» (erosion), обозначаемая \blacktriangleleft , множества X с помощью B , которые определяются

$$X \oplus B = \bigcup_{b \in B} X + b = \{x + b \mid x \in X, b \in B\}, \quad (1.1)$$

$$X \blacktriangleleft B = \bigcap_{b \in B} X - b = \bigcap_{b \in B} X_{-b}. \quad (1.2)$$

Два других оператора могут быть определены как комбинации «эрозии» и «наращения». Так, операция «размыкание» (opening) \circ и «замыкание» (closing) \bullet X с помощью B определяются как

$$X \circ B = (X \blacktriangleleft B) \oplus B, \quad (1.3)$$

$$X \bullet B = (X \oplus B) \blacktriangleleft B. \quad (1.4)$$

Для иллюстрации поведения этих операторов можно привести пример.

Пусть дано множество X и множество B :

$$X = \{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (2,1), (3,1), (4,1), (5,1)\};$$

$B = \{(0,0), (0,1)\}$, где в скобках обозначены координаты в системе (строка, столбец).

Определим результат операции «наращение».

Множество B имеет два значения с координатами $(0,0)$ и $(0,1)$, и необходимо определить два множества $X_{(0,0)}$ и $X_{(0,1)}$ и их объединение, поскольку

$$\begin{aligned} X \oplus B &= X_{(0,0)} \cup X_{(0,1)}; \quad X_{(0,0)} \equiv X; \\ X_{(0,1)} &= \{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,2), \\ &\quad (3,2), (4,2), (5,2)\}, \\ X \oplus B &= \{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), \\ &\quad (2,1), (2,2), (3,1), (3,2), (4,1), (4,2), (5,1), (5,2)\}. \end{aligned}$$

Для нахождения множества, образованного операцией «эрозии» множества X структурирующим элементом B , воспользуемся выражением (1.2), из которого следует, что нужно найти $X_{-(0,0)}$ и $X_{-(0,1)}$, так как

$$\begin{aligned} X \blacktriangleleft B &= X_{-(0,0)} \cap X_{-(0,1)}; \quad X_{-(0,0)} \equiv X \\ X_{-(0,1)} &= \{(1,-1), (1,0), (1,1), (1,2), (1,3), (1,4), (2,0), \\ &\quad (3,0), (4,0), (5,0)\}, \\ X \blacktriangleleft B &= X_{-(0,0)} \cap X_{-(0,1)} = \{(1,0), (1,1), (1,2), (1,3), (1,4)\}, \end{aligned}$$

т.е. $X \blacktriangleleft B$ – соответствует элементам с координатами, общими как у X -множества, так и у $X_{-(0,1)}$.

На этом простом примере убеждаемся, что операция «эрозия» приводит к уменьшению числа элементов множества X , в то время как «наращение» расширяет его.

Рассмотрим теперь воздействие операции «размыкание» тем же структурирующим элементом B на то же множество X согласно (1.3)

$$X \circ B = (X \blacktriangleleft B) \oplus B = (X \blacktriangleleft B)_{(0,0)} \cup (X \blacktriangleleft B)_{(0,1)} = (X \blacktriangleleft B) \oplus B;$$

$$\begin{aligned} (X \blacktriangleleft B) \oplus B &= X \blacktriangleleft B. \\ \text{Поскольку } X \blacktriangleleft B &= \{(1,0), (1,1), (1,2), (1,3), (1,4)\} \text{ то} \\ (X \blacktriangleleft B)_{(0,1)} &= \{(1,1), (1,2), (1,3), (1,4), (1,5)\}, \\ (X \blacktriangleleft B)_{(0,0)} &= \{(1,0), (1,1), (1,2), (1,3), (1,4)\}, \\ (X \blacktriangleleft B)_{(0,0)} \cup (X \blacktriangleleft B)_{(0,1)} &= \{(1,0), (1,1), (1,2), (1,3), \\ &\quad (1,4)\} \cup \{(1,1), (1,2), (1,3), (1,4), (1,5)\} = \{(1,0), (1,1), \\ &\quad (1,2), (1,3), (1,4), (1,5)\}. \end{aligned}$$

Получим теперь выражение воздействия операции «замыкание» структурирующим элементом B на множество X согласно выражению (1.4)

$$\begin{aligned} X \bullet B &= \{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), \\ &\quad (2,1), (2,2), (3,1), (3,2), (4,1), (4,2), (5,1), (5,2)\}, \\ (X \oplus B)_{(0,0)} &= X \oplus B, \\ (X \oplus B)_{(0,1)} &= \{(1,-1), (1,0), (1,1), (1,2), (1,3), (1,4), \\ &\quad (1,5), (2,0), (2,1), (3,0), (3,1), (4,0), (4,1), (5,0), (5,1)\}, \\ (X \oplus B) \blacktriangleleft B &= (X \oplus B)_{(0,0)} \cap (X \oplus B)_{(0,1)} = \{(1,0), (1,1), \\ &\quad (1,2), (1,3), (1,4), (1,5), (2,1), (3,1), (4,1), (5,1)\}. \end{aligned}$$

Набор оптимальных операторов обобщается на многоуровневые сигналы (изображения) посредством пороговых или «теневых» функций [1, 3].

В многоуровневом изображении мы имеем функцию $f(x)$, которая описывает уровни интенсивностей отсче-

тов изображения, где под x понимается множество точек изображения области P , на которых определена функция f . В этом случае результат операции «наращение» функции f структурирующим элементом B определяется

$$(f \blacktriangleright B)(x) = \sup_{i \in B} \{f(x-i)\}. \quad (1.5)$$

Операция «эрозия» дает новую функцию, которая вычисляется так

$$(f \blacktriangleleft B)(x) = \inf_{i \in B} \{f(x+i)\}. \quad (1.6)$$

Последовательное применение этих операций определяет операции «размыкание» и «замыкание»:

$$f \circ B = (f \blacktriangleleft B) \oplus B, \quad (1.7)$$

$$f \bullet B = (f \oplus B) \blacktriangleleft B. \quad (1.8)$$

Операции математической морфологии можно рассматривать как нелинейные операции. Следовательно, использование их для построения уровней вспомогательной пирамиды – нелинейная фильтрация, «сглаживающая» подвергается фильтрации изображение и обеспечивающая уменьшение числа тех деталей и объектов в изображении, размер которых меньше размера структурирующего элемента.

Форма, размер и вид структурирующего элемента определяют природу и степень такого «сглаживания». При этом информация о потерянных деталях изображения содержится в разностном сигнале (изображении), определяющем уровни основной морфологической пирамиды.

Морфологическая пирамида формируется по аналогичному пирамиде Лапласа принципу [1], с той разницей, что уровни вспомогательной пирамиды формируются фильтром нелинейного типа, операции которого описываются математической морфологией.

Передаче по каналу связи подлежит вершина, а также уровни пирамиды, которые в значительной степени менее чувствительны к искажениям, вносимым квантованием, осуществляемым в целях сжатия объема информации.

Для примера рассмотрим операцию «размыкание».

Уровни вспомогательной пирамиды обозначим через: I_1, \dots, I_N, I_0 – соответствуют исходному изображению.

$I_{k+1} = \text{Downsample}[I_k \circ B]$, $k=0, 1, 2 \dots N-1$, N – число уровней пирамиды.

$$I_{k+1,l} = \text{Expand}[I_{k+1}]; L_k = I_k - I_{k+1,l}; I_n = L_{k+1}$$

Здесь через *Downsample* обозначена операция децимации отсчетов с коэффициентом 2 по каждой координатной оси. В результате децимации осуществляется «прореживание» отсчетов по частоте. $I_k \circ B$ – изображение, полученное в результате осуществления операции «размыкания» структурирующим элементом B на k -ый уровень вспомогательной пирамиды. $\text{Expand}[I_{k+1}]$ означает операцию интерполяции децимированного $(k+1)$ -го уровня, необходимую для создания разностного изображения, представляющего k -ый уровень основной пирамиды; $I_{k+1,l}$ – результат интерполяции.

Если допустить, что I_0 – исходное изображение содержит $2^8 \cdot 2^8$ отсчетов, то на k -ом уровне пирамиды будет $2^{8-k} \cdot 2^{8-k}$ отсчетов. При построении 3-уровневой пирамиды число отсчетов на вершине будет равно $2^5 \cdot 2^5$.

Обозначим теперь через $J(x, y)$ функцию, описывающую интенсивности рассматриваемого изображения, определенного на области P ,

$$P = \{(x, y) / J(x, y) \neq \infty\},$$

$$I_k = \{J(x, y) / (x, y) \in P\}.$$

В качестве структурирующего элемента возьмем площадь квадрата, например $[3 \times 3]$. Тогда результат операции «эрозии» структурирующим элементом B , заданным на плоскости, опишется:

$$\text{Eroded}J(x, y) = \inf_{(i,j) \in B} \{J(x+i, y+j)\},$$

и окончательно

$$\text{Opened}J(x, y) = \sup_{(i,j) \in B} \{\text{Eroded}J(x-i, y-j)\}.$$

Заметим, что для осуществления операции интерполяции при восстановлении пропущенных в результате децимации отсчетов использовался алгоритм двумерной линейной интерполяции, который дал лучший результат, чем использование операции «наращение».

2. Результаты экспериментальных исследований

Исследования проводились с использованием системы MathCad Professional Edition (PRO) 2001 [5].

Эксперименты были проведены на изображении портретного вида с наличием большого числа контуров и резких перепадов. Для возможности сравнения полученных результатов с известными в литературе экспериментами выбрано тестовое изображение «Лена», на котором проведено множество классических экспериментов [1, 2, 3].

2.1. Строились четыре вида трехуровневых пирамид с применением вышеназванных операций.

В табл. 1 представлены значения математического ожидания, дисперсии и энтропии (в битах) для двух уровней пирамиды, полученной для основных опера-

Табл ца 1

	Уровень	«Эрозия»	«Нарращение»	«Размыкание»	«Замыкание»	Исходное изображение
Математическое ожидание	I уровень	16,632	-16,169	-4,571	6,641	99,028
	II уровень	16,655	-18,472	-4,701	-16,411	
Дисперсия	I уровень	523,54	595,296	319,209	324,418	2792
	II уровень	605,364	745,034	1544	484,367	
Энтропия	I уровень	5,639	5,824	5,340	5,398	7,604
	II уровень	5,744	5,987	6,792	6,325	

ций математической морфологии: «эрозии», «наращение», «размыкание» и «замыкание».

Коэффициент корреляции между исходным и разностным изображениями, полученными в результате построения первого уровня разностной пирамиды с применением операции «размыкание», равнялся 0.333, а второго уровня того же построения – 0.192. Были вычислены также значения ковариационной функции предложенного пирамидального построения. Так, например, на первом уровне вышеописанной пирамиды значение ковариации между элементами на первой строке и втором столбце $cov(0,2)=97.529$, в то время как для исходного изображения это значение $cov(0,2)=1471$, ковариация $cov(0,0)$ для исходного изображения равна 1517.

На рис. 1а представлена гистограмма распределения вероятностей интенсивностей отсчетов первого уровня пирамиды, а на рис. 1б – гистограмма распределения вероятностей интенсивностей отсчетов исходного изображения.

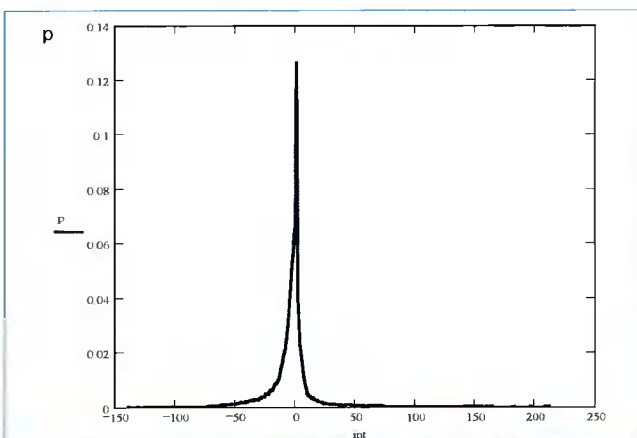


Рис. 1а

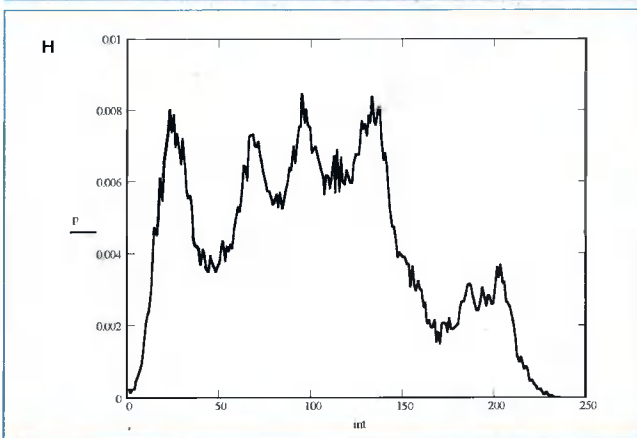


Рис. 1б

Анализ экспериментальных результатов показал:

1. Морфологическая пирамида, созданная на основании операций «замыкание» и «размыкание», характеризуется показателями с меньшей энтропией и дисперсией, чем полученные при использовании операций «наращение» и «эрозия». Самая низкая энтропия наблюдается на первом уровне пирамиды («размыкание»).

2. Построения морфологической пирамиды приводят к изменению статистических характеристик – смещению математического ожидания, уменьшению дисперсии, уменьшению энтропии (приблизительно на 2 бита).

Гистограмма распределения вероятности интенсивности отсчетов приобретает форму распределения Лапласа с более быстрым спадом к нулю, в то время как для исходного изображения гистограмма близка к равномерному распределению. Корреляционные связи между отсчетами разностного сигнала (автоковариационная функция) существенно ослабевают, хотя не наступает полная декорреляция. Между уровнями коэффициент корреляции также уменьшается.

Дальнейшие исследования касались зависимости статистических характеристик уровней пирамиды, полученных с применением операции «размыкание», от размера структурирующего элемента, который принимается равным: 2×2 , 3×3 , 4×4 , 5×5 , 6×6 отсчетов.

На рис. 2 представлены значения энтропии для разных размеров структурирующего элемента: кривая (1) соответствует энтропии первого уровня пирамиды, кривая (2) – второго уровня, кривая (3) – вершины пирамиды.

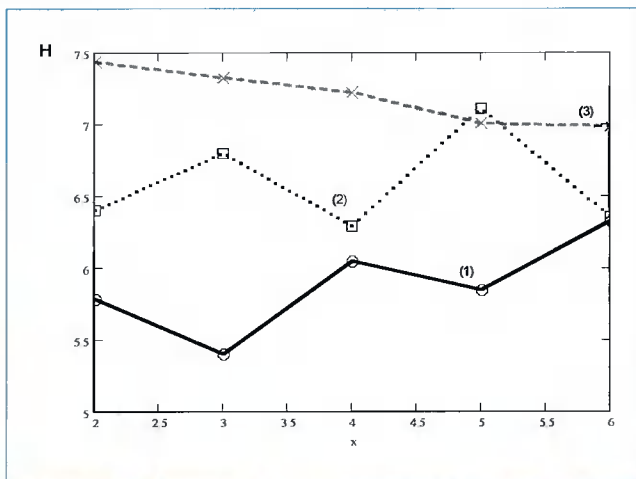


Рис. 2

Эксперименты свидетельствуют о том, что меньшие значения энтропии соответствуют структурирующему элементу с нечетным числом отсчетов, равным 3×3 . Данные приведены для первого уровня пирамиды. Все дальнейшие эксперименты проводились с таким структурирующим элементом данного типа.

2.2. Изучались шумоподавляющие свойства морфологических построений и проводилось сравнение с шумоподавляющими свойствами других фильтров, используемых в пирамидальных построениях.

Налагаемый шум моделировался генератором равномерно распределенных случайных чисел в интервале от 0 до 1, и каждый отсчет подвергался этой операции. Если выпавшее число оказывалось меньше заранее фиксированного значения, например 0,1, то интенсивность отсчета заменялась нулем (становилась черной точкой), если оказывалось больше 0,1, то интенсивность такого отсчета оставалась неизменной.

Зашумленное изображение отфильтровывалось фильтрами: морфологического, медианного и гауссового вида.

Результат фильтрации ошибок импульсного типа оценивался пиковым отношением сигнал-шум, которое определялось по выражению

$$ПОСШ = -10 \log \left\{ \frac{1}{256^2} \sum_{m=0}^{255} \sum_{n=0}^{255} [f(m,n) - \hat{f}(m,n)]^2 \right\}, \quad (2.1)$$

где $f(m,n)$ – интенсивность исходного изображения в точке (m,n) ; $m=0...255$, $n=0...255$; $\hat{f}(m,n)$ – интенсивность восстановленного изображения в той же точке; полное число строк и столбцов в кадре изображения – 256×256 .

На рис. 3 результаты сравнения шумоподавляющих свойств этих фильтров. На оси ординат отложены значения пикового отношения сигнал-шум (ПОСШ) зашумленного и отфильтрованных изображений. Кривая (1) отображает ПОСШ после фильтрации операцией «размыкание», кривая (2) – медианным фильтром, кривая (3) – Гауссовым, кривая (4) – отображает ПОСШ зашумленного изображения до фильтрации. На оси абсцисс отложены значения вероятности воздействующих шумов $p=0,1 - 0,4$.

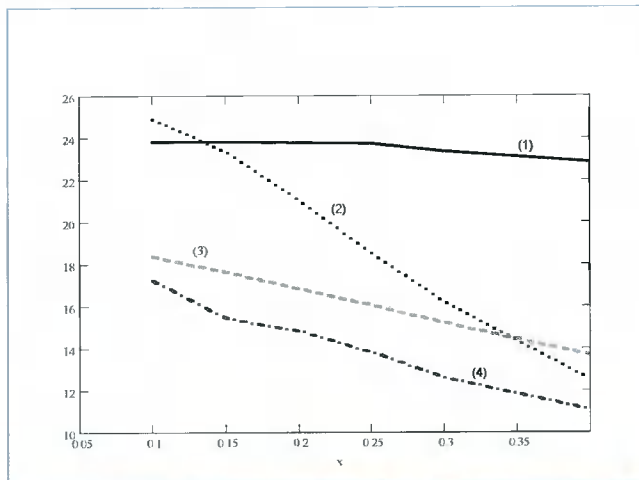


Рис. 3

Исследования показали, что фильтры с нелинейной фильтрацией в большей степени подавляют шумы импульсного типа (с равномерным распределением) в интервале вероятностей от $p=0,1$ до $p=0,35$, чем с линейной. При этом медианный фильтр с окном (3×3) для вероятностей $p=0,1, \dots, 0,125$ дает лучшие результаты по подавлению шумов, чем морфологический. Так например, $ПОСШ=24,8$ дБ при $p=0,1$ для медианы, для морфологического фильтра равно $23,8$ дБ, а для гауссового – $ПОСШ=18,3$ дБ; однако при увеличении вероятности морфологический фильтр дает значительно лучшие результаты. Так, например, при $p=0,2$ для медианного фильтра $ПОСШ=21$ дБ, для морфологического фильтра $ПОСШ=23,75$ дБ и для гауссового $ПОСШ=16,8$ дБ.

Кроме того, ПОСШ, полученное при фильтрации морфологическим фильтром, незначительно понижа-

ется при возрастании вероятности шума, в то время как в случае медианной фильтрации наблюдается резкое уменьшение ПОСШ (приблизительно от 24 до 15). Например, при $p=0,1$ $ПОСШ=23,8$ дБ для морфологического фильтра, $ПОСШ=24,8$ дБ для медианного фильтра, а при $p=0,4$ $ПОСШ=22,8$ дБ для морфологического фильтра и $ПОСШ=12,5$ дБ для медианного фильтра.

На рис. 4 представлено зашумленное и отфильтрованные изображения ($P=0,25$):

4а – исходное изображение; 4б – зашумленное изображение; 4в – отфильтрованное морфологическим фильтром; 4г – фильтром Гаусса; 4д – медианным фильтром.

Фильтр Гаусса является примером линейного фильтра, на основании которого формируются уровни пирамиды Лапласа.

Четкость и качество изображения визуально могут быть количественно оценены субъективной экспертизой, которая на данном этапе работы не производилась.

Были получены изображения ошибок между исходным (рис. 4а) и отфильтрованными изображениями: морфологическим фильтром с «размыканием» (рис. 4б), медианным фильтром (рис. 4д) и фильтром Гаусса (рис. 4г). Но для наглядности представления результатов выделялась определенная область изображения размером 40×40 элементов, содержащая контур для всех трех случаев; на рис. 5а представлены

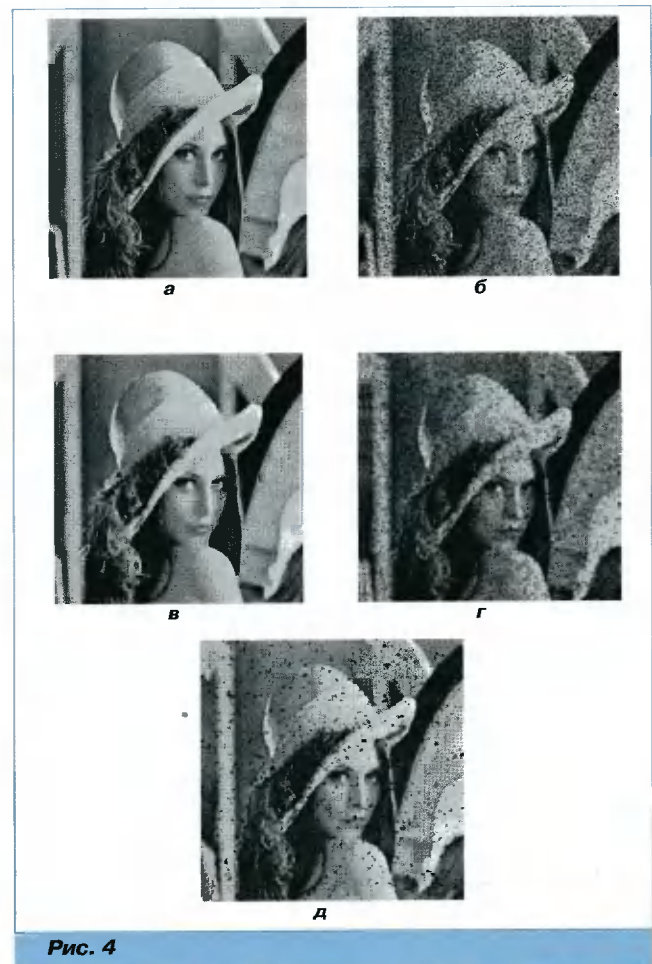


Рис. 4

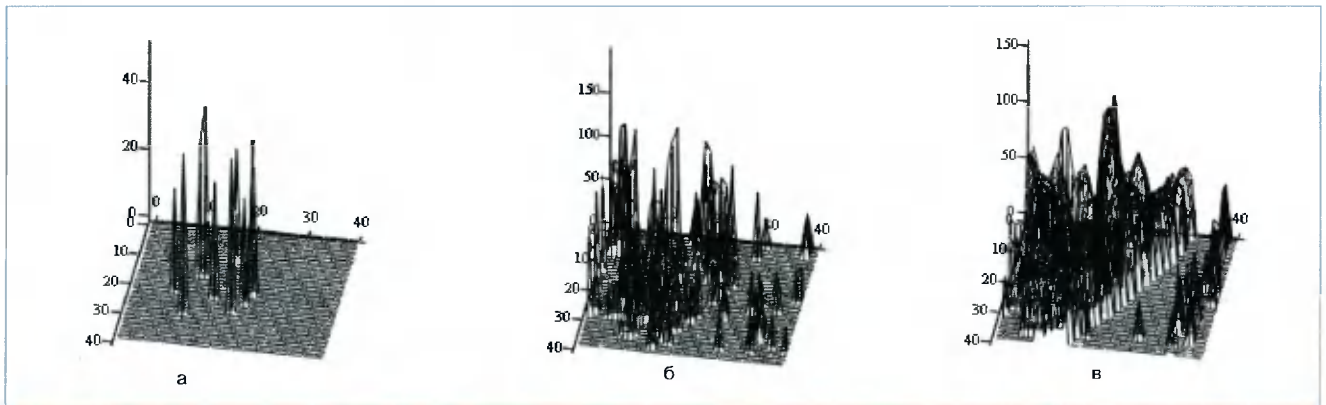


Рис. 5

ошибки, соответствующие использованию морфологического фильтра («размыкание»), рис. 5б – медианного и рис. 5в – фильтра Гаусса. Число ошибок, превышающих 0, для морфологического фильтра равно 10, медианного – 170, а для Гаусса – 617. Максимальное значение в первом случае – 51, во втором – 196, в третьем – 151,5. Следовательно, в данном случае контуры лучше сохраняются при применении морфологического фильтра.

2.3. Были проведены исследования эффективности применения в пирамидальном кодировании морфологической пирамиды с модифицированной структурой построения ее уровней.

Строилась морфологическая пирамида посредством операции «размыкание» структурирующим элементом 3×3 , с сокращением числа отсчетов в 9 раз в каждом последующем уровне (коэффициент децимации – 3). Снимались те же характеристики. Число отсчетов уже после первого построения составляло 85×85 . В табл. 2 приведены результаты эксперимента. Построение пирамиды ограничивалось первым этапом, так как создание следующего уровня пирамиды сопровождалось таким значительным возрастанием энтропии и динамического диапазона интенсивностей, что дальнейшее построение теряло смысл (табл. 2).

Для сравнения по эффективности кодирования изображений посредством предложенного построения (коэффициент децимации 3) с построением пирамиды (коэффициент децимации 2) применялась 2-битовая шкала оптимального квантования первого уровня пирамиды, рассчитанная по критерию Ллойда Макса [2]. Коэффициент сжатия оценивался отношением числа

бит, необходимых для передачи исходного изображения к числу бит, затрачиваемых после построения пирамиды и использования оптимального квантования. Качество восстановленного изображения характеризовалось точностными показателями: пиковым отношением сигнал-шум (ПОСШ, выражение (2,1)) и нормированной среднеквадратической ошибкой (НСКО, выражение (2,2)).

$$НСКО = \sqrt{\frac{\sum_{m=0}^{255} \sum_{n=0}^{255} [f(m,n) - \hat{f}(m,n)]^2}{\sum_{m=0}^{255} \sum_{n=0}^{255} [f(m,n)]^2}} \quad (2,2)$$

В табл. 3 отражены полученные результаты.

Анализ приведенных данных показал, что более эффективным с точки зрения сжатия является предложенное построение, поскольку увеличение коэффициента сжатия (более чем в 4 раза) сопровождается сравнительно незначительным понижением точностных показателей (примерно 10% для ПОСШ).

По всей вероятности, такой результат может быть объяснен тем, что коэффициент децимации как бы «согласуется» с размером структурирующего элемента.

Следует отметить, что в целях улучшения субъективного качества восстановленного изображения можно рекомендовать обработку восстановленного изображения исключением отрицательных значений интенсивностей отдельных отсчетов, которые высвечивались как белые точки, а также использованием медианной фильтрации.

Таблица 2

Уровни пирамиды	Математическое ожидание	Дисперсия	Энтропия	Размер изображения
Исходное изображение	99,028	2792	7,604	256x256
I уровень пирамиды	-3,252	329,8	3,786	85x85
Вершина	102,387	2776	7,485	85x85

Таблица 3

Коэффициент децимации	Коэффициент сжатия	Пиковое отношение сигнал-шум	Нормированная среднеквадратическая ошибка
2	1,88	25,89	0,116
3	9,071	23,25	0,157

Выводы

На основании проведенных исследований можно сделать следующие выводы.

1. Все четыре операции математической морфологии могут быть применены в пирамидальном кодировании изображений.

2. Пирамидальные построения морфологического типа обладают способностью преобразовывать статистику исходных изображений в вид, позволяющий результативно использовать различные варианты этих построений для эффективного кодирования изображений. При этом математическое ожидание интенсивности отсчетов смещается в сторону отрицательных значений, уменьшаются энтропия и дисперсия, происходит декорреляция, но не полная, как между уровнями, так и между отсчетами внутри уровня. Гистограмма распределения вероятности интенсивности отсчетов приобретает остроконечную узкую форму с более резким спадом к нулю, чем у пирамиды Лапласа, и со значительным увеличением процентного числа элементов с интенсивностью, равной нулю. При этом статистические свойства уровней пирамиды зависят как от вида используемой морфологической операции, так и от размера структурирующего элемента. Наименьшая энтропия получена для первого уровня пирамиды с операцией «размыкание» структурирующим элементом размером $[3 \times 3]$ отсчетов.

3. Фильтры с нелинейной фильтрацией в большей степени подавляют шумы импульсного типа (с равномерным распределением) в интервале вероятностей от $p=0,1$ до $p=0,35$, чем с линейной. При этом ПОСШ, полученное при фильтрации морфологическим фильтром с использованием операции «размыкание» структурирующим элементом размером $[3 \times 3]$, незначительно понижается при возрастании вероятности шума, в

то время как в случае медианной фильтрации наблюдается резкое уменьшение ПОСШ (приблизительно от 24 до 15). Кроме того, из результатов, полученных для фрагментов изображения ошибок, следует, что в отфильтрованном морфологическим фильтром изображении контуры и резкие перепады сохраняются в большей степени, чем при использовании фильтров медианного и гауссового типа.

4. Построение морфологической пирамиды с модифицированной структурой, в которой число отсчетов уменьшается в 9 раз (вместо 4) по сравнению с предыдущим уровнем, характеризуется лучшими статистическими характеристиками (уменьшением энтропии, дисперсии и остроконечностью гистограммы) и является эффективным с точки зрения сжатия, поскольку увеличение коэффициента сжатия более чем в 4 раза сопровождается сравнительно слабым понижением точностных показателей восстановленных изображений (примерно на 10% для ПОСШ).

Литература

1. Харатишвили Н.Г., Чхеидзе И.М., Ронсен Д., Инджия Ф.И. Пирамидальное кодирование изображений. М.: Радио и связь, 1996, 197 с.
2. Цифровое кодирование телевизионных изображений. / Под ред. И.И. Цукермана. М.: Связь, 1981, 240 с.
3. Марагос П., Шаффер Р.У. Морфологические системы для многомерной обработки сигналов. ТИИЭР, т. 78, №4, 1990, с. 109–131.
4. Орищенко В.И., Санников В.Г., Свириденко В.А. Сжатие данных в системах сбора и передачи информации. М.: Радио и связь, 1985, 185 с.
5. Кудрявцев Е.М. Mathcad 8. М.: ДМК, 2000, 314 с.

Электронная юстировка изображений при мультиспектральном наблюдении

Б.А. Алпатов, П.В. Бабаян

Введение

При решении ряда задач мультиспектральной обработки видеопоследовательностей возникает проблема совмещения кадров изображений, полученных датчиками с различной спектральной чувствительностью так, чтобы точки изображений, имеющие одинаковые координаты, являлись проекциями одной и той же точки пространства. Эту операцию можно выполнить не прибегая к электронным методам. Для этого необходимо иметь датчики с одинаковыми углами зрения, разместить их на малом расстоянии друг от друга и ориентировать одинаковым образом. Однако датчики, как правило, имеют отличающиеся углы зрения и точная ориентировка их связана со значительными трудностями. Избежать этих проблем можно, используя электронный алгоритм юстировки, разработке которого посвящена настоящая статья.

Постановка задачи

Рассмотрим поверхность S в 3-мерном пространстве, заданную уравнением $F(x, y, z) = 0$ в прямоугольной системе координат $Oxyz$. Поверхность наблюдается двумя видеодатчиками $D_k, k=1, 2$, которые проецируют точки поверхности S на плоскости изображений I_k (рис. 1).

Свяжем с каждым датчиком $D_k, k=1, 2$ прямоугольную систему координат $O^{(k)}x^{(k)}y^{(k)}z^{(k)}$ так, чтобы фокус датчика имел однородные координаты $(0, 0, -f_k, 1)$, где f_k – фокусное расстояние k -го датчика.

Условимся обозначать символом X как саму точку X , так и ее координаты в системе координат $Oxyz$, а символом $X^{(k)}$ – координаты точки X в системе координат $O^{(k)}x^{(k)}y^{(k)}z^{(k)}$. Для упрощения выкладок формулы приведены в однородных координатах [1]. Неоднородные координаты будем записывать с верхним индексом n . С учетом этих обозначений уравнение плоскости изображения I_k будет иметь вид $z^{(k)} = 0$.

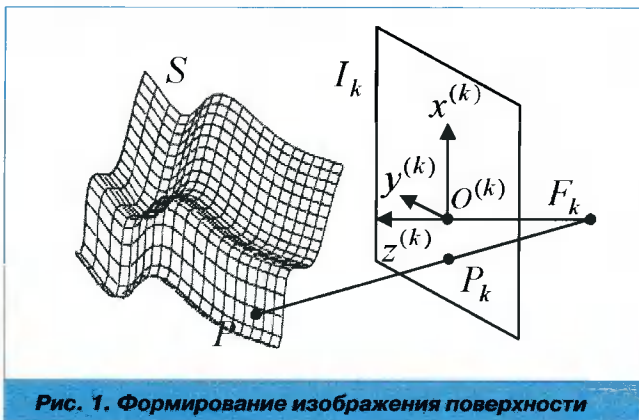


Рис. 1. Формирование изображения поверхности

Сформулирована задача юстировки двухспектральной видеопоследовательности. Предложено ее решение при наблюдении плоской сцены, которое сводится к решению системы линейных уравнений. Приведены результаты испытаний алгоритма юстировки на ЭВМ.

Системы координат $O^{(k)}x^{(k)}y^{(k)}z^{(k)}, k=1, 2$ связаны с системой координат $Oxyz$ преобразованиями Φ_k

$$X^{(k)} = \Phi_k(X). \quad (1)$$

Преобразование Φ_k включает в себя поворот вокруг всех трех осей, сдвиг и однородное изменение масштаба относительно осей x и y . Каждое Φ_k в однородных координатах представляется матрицей (4×4)

$$\Phi_k(X) = S_k M_{k,xy} R_{k,z} R_{k,y} R_{k,x} X. \quad (2)$$

Здесь $M_{k,xy} = \text{diag}(\lambda_k, \lambda_k, 1, 1)$ – матрица изменения масштаба, S_k – матрица сдвига (4×4) , $R_{k,z}, R_{k,y}, R_{k,x}$ – матрицы поворота вокруг соответствующих осей. К примеру,

$$R_{k,z} = \begin{bmatrix} \cos(\varphi_k) & -\sin(\varphi_k) & 0 & 0 \\ \sin(\varphi_k) & \cos(\varphi_k) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

где φ_k – угол поворота.

Каждая точка $P \in S$ проецируется на плоскости I_k . Обозначим проекции точки P через P_k . Допустим, что точка P имеет в системе координат $O^{(k)}x^{(k)}y^{(k)}z^{(k)}$ координаты $P^{(k)}$. Тогда согласно уравнениям перспективного [1] преобразования однородные координаты точки P_k в системе координат k

$$P_k^{(k)} = Q_k P^{(k)}, \text{ где}$$

$$Q_k = \begin{bmatrix} f_k & 0 & 0 & 0 \\ 0 & f_k & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & f_k \end{bmatrix}. \quad (4)$$

Связь между координатами проекции точки P и ее реальными (неоднородными) координатами задается уравнением -

$$\begin{bmatrix} X_{P_k^{(k)}}^n \\ Y_{P_k^{(k)}}^n \end{bmatrix} = \frac{f_k}{z_{P^{(k)}}^n + f_k} \begin{bmatrix} X_{P^{(k)}}^n \\ Y_{P^{(k)}}^n \end{bmatrix}. \quad (5)$$

Пусть нам каким-либо образом стали известны координаты N пар точек изображений $I^{(1)}$ и $I^{(2)}$, которые являются проекциями одной и той же точки поверхности S . Эти координаты мы обозначим соответственно

$$\begin{bmatrix} x_{n1} \\ y_{n1} \end{bmatrix} \text{ и } \begin{bmatrix} x_{n2} \\ y_{n2} \end{bmatrix}, \quad n=1,2,\dots,N.$$

Требуется найти такое преобразование Ψ , чтобы

$$\forall P \in S: \begin{bmatrix} x_{P_1^{(1)}}^H \\ y_{P_1^{(1)}}^H \end{bmatrix} = \Psi \left(\begin{bmatrix} x_{P_2^{(2)}}^H \\ y_{P_2^{(2)}}^H \end{bmatrix} \right). \quad (6)$$

Проще говоря, необходимо найти преобразование, которое устанавливает соответствие между точками изображений $I^{(k)}$, $k=1,2$ таким образом, чтобы соответствующие точки являлись проекциями одной и той же точки поверхности S .

Решение

Ограничимся случаем, когда поверхность S представляет собой плоскость. В основном это соответствует реальным условиям наблюдения. Так как выбор системы координат произволен, выберем ее так, чтобы уравнение поверхности S было $z=0$. Рассмотрим преобразование Ψ , которому подвергается точка P плоскости S при проецировании на плоскость изображения I_k . Из (2), (4) следует

$$P_k^{(k)} = \Psi_k(P) = D_k P, \quad \text{где } D_k = Q_k S_k M_{k,xy} R_{k,z} R_{k,y} R_{k,x}. \quad (7)$$

Заметим, что у матрицы D_k третья строка нулевая. Так как $z_P=0$, то третий столбец D_k может быть вычеркнут вместе с третьей координатой точки P . Поэтому (7) можно переписать в виде

$$P_k^{(k)'} = D_k' P'. \quad (8)$$

Здесь $P_k^{(k)'}$, P' – векторы, состоящие из первых, вторых и четвертых координат векторов $P_k^{(k)}$, P , а D_k' – это матрица D_k с удаленным третьим столбцом и третьей строкой.

Таким образом, координаты x, y , точек плоскости S связаны с координатами точек плоскости I_k проективным преобразованием (7). Тогда координаты точек плоскостей I_1 и I_2 связаны преобразованием

$$P_1^{(1)'} = D_1' D_2'^{-1} P_2^{(2)'}. \quad (9)$$

Поскольку проективные преобразования плоскости образуют группу [2], то обратное к проективному преобразованию будет проективным. Произведение проективных преобразований также является проективным преобразованием. Из (9) следует, что преобразование Ψ нужно искать в форме проективного преобразования, однозначно задаваемого матрицей вида

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}. \quad (10)$$

Функция Ψ описывается выражениями

$$x_{P_1^{(1)}}^H = \frac{a_1 x_{P_2^{(2)}}^H + a_2 y_{P_2^{(2)}}^H + a_3}{a_7 x_{P_2^{(2)}}^H + a_8 y_{P_2^{(2)}}^H + a_9}, \quad (11)$$

$$y_{P_1^{(1)}}^H = \frac{a_4 x_{P_2^{(2)}}^H + a_5 y_{P_2^{(2)}}^H + a_6}{a_7 x_{P_2^{(2)}}^H + a_8 y_{P_2^{(2)}}^H + a_9}.$$

Таким образом, задача сводится к определению параметров a_i , $i=1\dots 8$. Подставляя

$$\begin{bmatrix} x_{n1} \\ y_{n1} \end{bmatrix}, \begin{bmatrix} x_{n2} \\ y_{n2} \end{bmatrix}, \quad n=1,2,\dots,N$$

в (11) и выполнив элементарные преобразования, получим систему из $2N$ уравнений

$$\begin{aligned} x_{n2} a_1 + y_{n2} a_2 + a_3 - x_{n1} x_{n2} a_7 - x_{n1} y_{n2} a_8 &= x_{n1} a_9, \\ x_{n2} a_4 + y_{n2} a_5 + a_6 - y_{n1} x_{n2} a_7 - y_{n1} y_{n2} a_8 &= y_{n1} a_9, \end{aligned} \quad (12)$$

Система (12) будет определена, если $N \geq 4$.

Проблема выбора опорных точек

Чтобы решить задачу, поставленную выше, необходимо знать хотя бы четыре пары точек, которые являются проекциями одной и той же точки плоскости. Выявить такие точки автоматически при многоспектральном наблюдении трудно, потому что изображения сцены, полученные в разных спектральных диапазонах, могут очень сильно отличаться. Решить данную проблему можно, разместив на плоскости наблюдения несколько меток, которые имеют характерный внешний вид во всех спектральных диапазонах. Например, это могут быть прожекторы, лучи которых направлены в сторону датчиков.

Зрительный анализ двух разноспектральных кадров почти всегда позволяет найти несколько пар опорных точек.

Если видеосъемка производилась при полной неподвижности датчиков относительно друг друга, достаточно найти опорные точки в первых кадрах, вычислить параметры преобразования и использовать их на каждом кадре.

Результаты испытаний

Разработанный алгоритм лег в основу программного обеспечения для юстировки двухспектральных видеосюжетов. Входными данными программы являются два видеосюжета, полученные в разных спектральных диапазонах. Программа позволяет вручную выбирать опорные точки, вычисляет параметры проективного преобразования, преобразует каждый кадр одного из сюжетов и записывает преобразованный видеосюжет.

Эксперименты показали, что если выбор опорных точек достаточно точен, а их прообразы располагаются приблизительно в одной плоскости, то юстировка последовательностей выполняется с хорошим качеством по всей площади кадра. Для увеличения точности юстировки количество пар опорных точек бралось больше четырех, от 6 до 8, и система уравнений (12) решалась методом псевдообращения матрицы. Чтобы избежать плохой обусловленности системы (12), опорные точки нужно выбирать так, чтобы любые три из них по возможности не лежали на одной прямой.



Рис. 2. Исходный ТВ-кадр



Рис. 4. Преобразованный ТВ-кадр

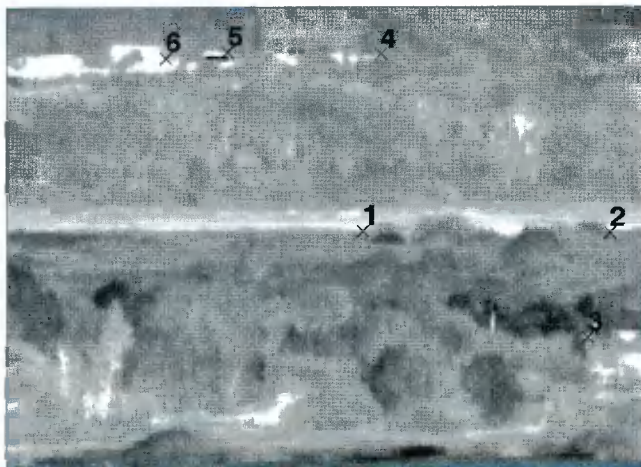


Рис. 3. Исходный ИК-кадр



Рис. 5. Преобразованный ИК-кадр

На рис. 2 приведено изображение, полученное с телевизионного датчика, а на рис. 3 – с инфракрасного. Это изображения одной и той же сцены в одно и то же время. Опорные точки отмечены крестиками и пронумерованы. Соответствующие пары точек имеют одинаковые номера. На рис. 4 показано телевизионное изображение, подвергнутое проективному преобразованию. Как видно, опорные точки переместились и находятся в тех же позициях, что и опорные точки изображения на рис. 3. На рис. 5 приведено инфракрасное изображение, преобразованное аналогичным образом.

При анализе матриц (10), вычисленных с использованием реальных сюжетов, выяснилось, что параметры a_7 и a_8 обычно близки к нулю. Сюжеты были получены недалеко расположенными друг от друга датчиками с достаточно большого расстояния от сцены. Таким об-

разом, проективное преобразование выродилось в более простое для вычислений аффинное. Например, матрицы преобразований из рис. 2 в рис. 4 и из рис. 3 в рис. 5. имеют вид, приведенный в табл. 1.

Трудной проблемой оказался ручной выбор опорных точек. Не всегда удается четко различить одинаковые точки на разном спектральных изображениях, а тем более, выбрать их лежащими не на одной прямой.

Выводы

Результаты экспериментальных исследований подтвердили принципиальную разрешимость проблемы юстировки двухспектральной видеопоследовательности в рамках предложенной методики.

Таблица 1

Кадр	Матрица проективного преобразования
ТВ	$\begin{bmatrix} 0,7374 & 0,0010 & 158,2661 \\ 0,0319 & 0,7702 & -5,3442 \\ -0,0000 & 0,0000 & 1,0000 \end{bmatrix}$
ИК	$\begin{bmatrix} 1,3539 & 0,0119 & -214,0708 \\ -0,0561 & 1,3073 & 15,9659 \\ 0,0001 & -0,0001 & 1,0000 \end{bmatrix}$

Литература

1. Иванов В.П., Батраков А.С. Трехмерная компьютерная графика. – М.: Радио и связь, 1995, 224 с.
2. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. – М.: Машиностроение, 1990, 320 с.

УДК 621.397.13

Аппаратная реализация систем ввода, обработки, хранения и вывода изображений различных телевизионных форматов и стандартов

А.В. Бондаренко, И.В. Докучаев, В.Б. Стешенко

Введение

Качественный скачок в степени интеграции современной элементной базы за последние годы позволяет реализовать схемотехнические решения, которые ранее не были достижимы. Это появление высокоскоростных АЦП/ЦАП с большим динамическим диапазоном, скоростных микросхем памяти SD/SGRAM большой емкости и ПЛИС [5], в которых можно разместить практически всю систему обработки информации. Растут требования и к системам «оцифровки» изображений для решения целого ряда задач обработки, в частности обработки медицинских изображений больших пространственных разрешений, а также в телевидении высокой четкости [2], [4]. Автоматизация медицинских и научных исследований, создание экспертных и измерительных систем, средств оперативной видеотелеконференц-связи заставляют искать пути получения и обработки высококачественных изображений. Отсутствие на российском рынке недорогих универсальных высоко-эффективных систем ввода-вывода и обработки изображений, работающих с различными видеоформатами, ставит задачу их создания.

В настоящее время научно-производственной фирмой «Растр Технолоджи» выпускается целый ряд видеопроцессоров, совмещенных с функцией захвата кадров (Frame Grabber), перекрывающих динамический и частотный диапазон АЦП от 8 бит 32 МГц (RT-822VP), 10 бит 60 МГц (RT-821VP, RT-823VP) и до 12 бит 105 МГц (RT-851VP) [1].

Общие характеристики видеопроцессоров

Рассмотрим как пример реализации специализированные платы серии RT-8xxVP, разработанные фирмой «Растр Технолоджи» [1]. Эти платы представляют собой мультиформатные видеопроцессоры, предназначенные для высококачественного ввода черно-белого телевизионного изображения в ПК и обработки его в реальном времени. В качестве источника сигнала могут использоваться телекамеры, ПЗС-линейки и матрицы, видеомагнитофоны, электронные микроскопы, рентгеновские установки, УЗИ-сканеры и другие устройства, выдающие видеосигнал в различных форматах (стандартах). Платы имеют 4-канальный входной мультиплексор и видеовыход, позволяющий контролировать на внешнем видеоконтрольном устройстве (ВКУ) ввод сигнала, выводить на ВКУ «замо-

Приведен обзор аппаратных средств цифровой обработки сигналов, обеспечивающих ввод, обработку, хранение и вывод изображений различных телевизионных форматов и стандартов. К таким устройствам относятся: видеопроцессоры реального времени, совмещенные с функцией захвата кадров (Frame Grabber); цифровые камеры высокого разрешения и динамического диапазона на ПЗС-матрицах и линейках; высокоскоростные интерфейсы систем ввода цифровых данных в компьютер.

роженный» кадр или последовательность кадров в реальном масштабе времени. Различные автоматические настройки дают возможность производить обработку видеосигнала без участия оператора. Видеопроцессоры имеют аналоговую ФАПЧ, представляющую собой фазовый детектор, фильтр, накопительный конденсатор, широкодиапазонный ГУН, делитель на 2 и выходной буфер. ФАПЧ позволяет стабилизировать дрожание пикселя до ± 4 нс. Используя счетчики с переменным коэффициентом деления, можно добиться большого диапазона перестройки строчных и пиксельных частот.

Для подключения внешних сигналов синхронизации или управляющих и исполнительных устройств в RT-82xVP имеется четырехразрядный, а в RT-851VP – девятиразрядный цифровой интерфейс (ТТЛ). Важным свойством плат является сочетание таких характеристик, как высокое пространственное разрешение и ввод сигналов без пропуска кадров с высокой частотой их следования (более 60 Гц). Это достигается благодаря специальной организации внутреннего буфера памяти и встроенному PCI-контроллеру, который работает в режиме DMA и обеспечивает среднюю скорость пересылки данных более 60 Мбайт/с (без участия центрального процессора) [3].

Типичные технические характеристики приборов приведены в табл. 1.

Аппаратная реализация

На рис. 1 представлена функциональная схема устройства RT-851VP.

Устройство состоит из следующих функциональных узлов [1], [3]:

- 4-канальный видеомultipлексор – осуществляет коммутацию аналоговых входов (время переключения – 8 нс);
- схема восстановления постоянной составляющей (DC) – осуществляет фиксацию уровня гашения;
- нормирующий усилитель – обеспечивает согласование уровня входного сигнала с динамическим диапазоном АЦП;

Таблица 1

Технические характеристики	RT-822VP	RT-823VP	RT-851VP
Разрядность данных при вводе, бит	8	от 8 до 10	от 8 до 12
Разрядность данных при выводе, бит	8	от 8 до 10	
Входное сопротивление, Ом	75		
Уровень входного сигнала, В	0,5 – 2		
Регулировка усиления, дБ	± 6 (256 градаций)		
Регулировка уровня черного, В	± 0,3 (256 градаций)		
Синхронизация	внутренняя и внешняя (ТТЛ)		
Частота строк, кГц	12 – 20	12 – 100	
Пиксельная частота, МГц	от 8 до 32	от 2,5 до 60	от 2,5 до 64
Объем внутренней памяти, МБ	4	8	16
Функции настройки и обработки изображений	RT-822VP	RT-823VP	RT-851VP
Автоматическая настройка оптимальных значений яркости и контрастности вводимого изображения	+	+	+
Усреднение полукадров	+	+	+
Накопление и усреднение 2, 4, 8, 16, 32, 64, 256 смежных кадров	до 256	до 64	до 64
Регулировка аспектного соотношения телевизионного изображения	+	+	+
Программная регулировка размеров и положения области ввода изображения в пределах телевизионного раstra	+	+	+
Программная регулировка размеров и положения области вывода изображения в пределах телевизионного раstra	-	+	+
Анализ и автоматическая настройка входного телевизионного сигнала	-	-	+
Рекурсивное накопление с различными весовыми коэффициентами непрерывной последовательности телевизионных кадров	-	-	+
Вычитание фона (маски) с различными весовыми коэффициентами	-	-	+
Сложение/вычитание вводимого телевизионного изображения с заданным кадром	-	-	+
Дополнительное повышение контрастности вводимого изображения	-	-	+
Режим обработки в окне	-	-	+
Построение гистограммы распределения яркости	-	-	+

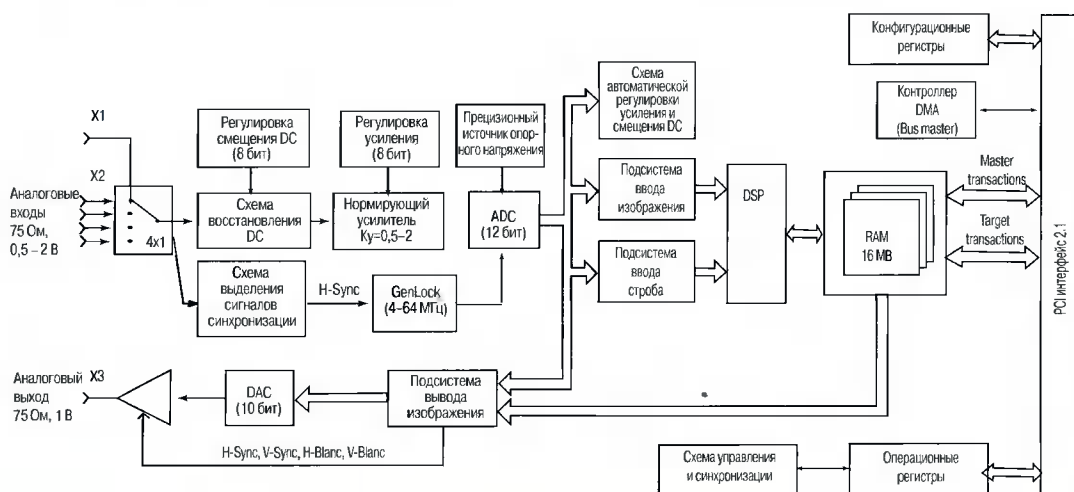


Рис. 1. Функциональная схема устройств RT-851VP

- схема регулировки смещения постоянной составляющей (DC) и усиления – позволяет программно автоматически или вручную настраивать яркость и контрастность входного телевизионного изображения (256 градаций);
- схема выделения сигналов синхронизации – выделяет из входного сигнала строчные и кадровые синхросимпульсы;
- ADC – 12-разрядный аналого-цифровой преобразователь (АЦП); для повышения стабильности и точности преобразования используется прецизионный источник опорного напряжения;
- GenLock – широкодиапазонный генератор частоты квантования входного сигнала (пиксельной частоты); имеет систему фазовой автоподстройки частоты (ФАПЧ) и обеспечивает дрожание пикселя (*jitter*) не более ± 4 нс;
- DAC – 10-разрядный выходной цифроаналоговый преобразователь (ЦАП); использует старшие 10 бит 12-разрядных данных;
- подсистема ввода изображения – основной канал ввода; формирует окно ввода кадров изображения, осуществляет управление фиксацией изображения во внутреннем буфере памяти;
- подсистема ввода строка – дополнительный канал ввода, работающий параллельно с основным; позволяет вводить изображение с другим форматом окна и другими функциями цифровой обработки, используется так же, как вспомогательный канал для DSP-обработки данных основного канала ввода;
- подсистема вывода изображения – формирует окно вывода выходного изображения, вырабатывает сигналы синхронизации, осуществляет переключение потоков данных (сквозной канал АЦП-ЦАП / вывод из внутреннего буфера памяти);
- выходной усилитель – формирует выходной телевизионный сигнал и обеспечивает согласование с 75-омной нагрузкой;

- схема управления и синхронизации – осуществляет общее управление всеми процессами в устройстве;
- RAM – 32-разрядный внутренний буфер памяти (SDRAM, 16 МБ); буфер функционально разделен на 8 банков по 2 Мбайта;
- DSP – блок цифровой обработки сигналов реального времени; выполняет целочисленные операции сложения, вычитания, умножения и деления элементов одного или нескольких изображений; позволяет реализовать различные виды накопления; блок DSP является динамически загружаемым в процессе работы устройства модулем (программа загружается из файла на диске); время обновления функции – не более 10 мс;
- PCI Interface V2.1, контроллер DMA, конфигурационные и операционные регистры – обеспечивают связь RT-851VP с оперативной памятью компьютера через шину PCI в режимах MASTER и TARGET.

Блок DSP, конвейер, конечный автомат работы с SDRAM и стеки FIFO реализованы на ПЛИС ALTERA семейства APEX EP20K100QC240-2; интерфейс-мастер PCI V2.1, интеллектуальный контроллер, входные/выходные ФАПЧ, управление электронными регулировками и настройками на семействе ACEX EP1K50FC256-2. Плата выполнена по технологии двухстороннего поверхностного монтажа с применением FBGA-256 корпуса с шагом 1 мм. Габаритные размеры – 120x102x18 мм [5], [6].

Устройство ввода и обработки изображений обеспечивает следующие основные режимы работы:

- **фиксацию** телевизионного *кадра* во внутреннем буфере памяти;
- **передачу** данных через интерфейс PCI;
- **мониторирование** (просмотр) входного телевизионного сигнала на внешнем мониторе (или ВКУ – видеоконтрольном устройстве);
- **визуализацию** (просмотр) на внешнем мониторе (ВКУ) телевизионных кадров, записанных во внутреннем буфере памяти.

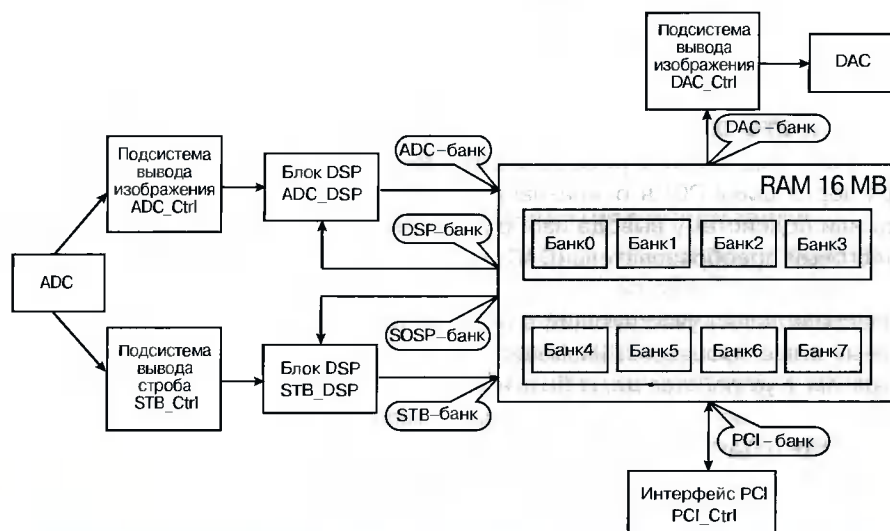


Рис. 2. Структура буфера видеопроцессора

Различные сочетания перечисленных режимов работы устройства предполагают его использование в следующих вариантах:

- непрерывный или одиночный ввод телевизионных кадров (с обработкой или без таковой) в компьютер с одновременным контролем вводимого исходного или обработанного изображения на внешнем мониторе;
- непрерывный или одиночный ввод телевизионных кадров (с обработкой или без) в компьютер с одновременным просмотром любого ранее введенного телевизионного кадра на внешнем мониторе; при этом формат вводимого и выводимого телевизионного изображения может быть различным;
- вывод серии ранее введенных кадров или полукадров на внешний монитор в реальном времени или с замедлением (без ввода изображения в компьютер).

Использование входного коммутатора на 4 аналоговых входа и 4 программируемых цифровых входов/выходов позволяет реализовать функциональные дополнительные возможности: подключение до 4 источников ТВ-сигнала (возможно RGB), использование внешних команд управления, подключение исполнительных устройств и т.д.

Отдельно хотелось бы рассмотреть особенности организации буфера памяти, поскольку, как правило, это наиболее критичный узел систем ввода и обработки видеоизображений. Структура буфера приведена на рис. 2 (видеопроцессор RT-851VP).

Внутренний буфер памяти имеет объем 16 Мбайт и состоит из 8 банков по 2 Мбайт. Каждый банк используется для записи и хранения одного телевизионного кадра.

Логика работы видеопроцессора предполагает одновременное обращение к внутренней памяти со стороны нескольких подсистем устройства. При захвате видеосигнала данные от аналого-цифрового преобразователя (ADC) сначала через подсистемы ввода изображения и строба заносятся в банки памяти, определяемые регистрами **ADC_Ctrl** и **STB_Ctrl**. Если при этом задействована какая-либо функция цифровой обработки, то в качестве второго операнда используются данные из банков, определяемых регистрами **ADC_DSP** и **STB_DSP** соответственно. После «оцифровки» всего кадра или строба данные могут передаваться через шину PCI в оперативную память компьютера или подсистему вывода изображения в цифроаналоговый преобразователь (DAC) и далее на ВКУ.

Банк памяти, непосредственно участвующий в одном из перечисленных выше процессов, называется активным. Таким образом, в устройстве могут быть несколько активных банков, которые условно назовем **PCI**, **ADC**, **DSP**, **STB**, **SDSP** и **DAC-активными банками**. Номера активных банков определяются соответственно битами D2-D0 регистров **PCI_Ctrl**, **ADC_Ctrl**, **ADC_DSP**, **STB_Ctrl**, **STB_DSP** и **DAC_Ctrl**.

Назначение активных банков при вводе и выводе одиночных кадров может быть любым. Достаточно, на-

пример, использовать нулевой банк в качестве активного ADC-банка для «оцифровки» кадра и в качестве активного PCI-банка для последующей передачи его через шину PCI.

При вводе или выводе непрерывного потока кадров манипулирование активными банками памяти должно подчиняться определенному правилу: «оцифровка» или вывод телевизионного изображения осуществляются в банке, отличном от банка, выбранного в данный момент для передачи данных по шине PCI. Это позволит избежать пропуска кадров при вводе и выводе изображения. Помимо этого следует учесть, что внутри буфера существует разделение банков памяти на две группы – это банки с номерами 0–3 и 4–7. Доступ к банкам из разных групп осуществляется параллельно, внутри одной группы – последовательно (в порядке поступления запросов от различных подсистем видеопроцессора). Поэтому при больших частотах «оцифровки» и использовании DSP-обработки, когда плотность потока данных значительно возрастает, для эффективного управления процессами ввода-вывода изображения следует пользоваться банками из разных групп. Правильная организация конвейера – довольно нетривиальная задача и будет рассмотрена отдельно.

Обмен данными на шине PCI может проходить в двух режимах: MASTER и TARGET. В режиме TARGET передача данных осуществляется с помощью простых языковых операторов типа **copy**. В режиме MASTER задействован механизм прямого доступа к памяти (DMA). Скорость передачи данных в режиме MASTER примерно в шесть раз больше и составляет более 60 Мбайт/с.

При обмене данными в режиме TARGET каждый банк памяти делится на 128 частей (окон). Одно окно является частью адресного пространства оперативной памяти компьютера размером 16 кбайт и начальным адресом, сдвинутым на 16 кбайт вверх относительно адреса, указанного в первом базовом адресном регистре конфигурационного заголовка платы. Например, если базовый адрес равен 000D0000h, то окно памяти, через которое будет вестись обмен данными, составит диапазон 000D4000h – 000D7FFFh. Номер окна памяти (от 0 до 127), подключенного к адресному пространству компьютера, определяется битами D9..D3 регистра **PCI_Ctrl**.

При обмене данными в режиме MASTER (или DMA) внутренний буфер памяти видеопроцессора на адресное пространство оперативной памяти компьютера не отображается. Банк памяти представляет собой единое целое и может быть записан или считан целиком за один раз. Обмен данными в режиме DMA будет рассмотрен ниже.

При «оцифровке» изображения строки ТВ-кадра во внутреннем буфере памяти располагаются последовательно друг за другом – сначала строки первого поля, затем второго. При передаче изображения в оперативную память компьютера для восстановления правильного чередования строк можно использовать чересст-

рочный обмен данными (то же относится и к выводу изображения на внешний монитор).

Таким образом, мы рассмотрели основы построения аппаратных средств цифровой обработки сигналов, обеспечивающих ввод, обработку, хранение и вывод изображений различных телевизионных форматов и стандартов, предназначенных для применения в научных исследованиях, медицине и промышленности.

Функции цифровой обработки

Используя регистры **ADC_DSP** и **STB_DSP**, осуществляется управление цифровой обработкой ТВ-сигнала, поступающего от подсистем ввода изображения и строба.

Биты	Описание
2..0	Выбор банка в качестве второго операнда для DSP-обработки (выбор активных DSP- и SDSP-банков)
3	Не используется
6..4	Выбор функции DSP-обработки: D2 D1 D0 0 0 0 – ввод без цифровой обработки 0 0 1 – усреднение первого и второго полей кадра 0 1 0 – исключение фона, записанного в DSP-банке 0 1 1 – вычитание DSP-банка 1 0 0 – сложение с DSP-банком 1 0 1 – операция AND с DSP-банком 1 1 0 – операция OR с DSP-банком 1 1 1 – операция XOR с DSP-банком
31..7	Не используются

Изображение, используемое в качестве второго операнда, должно быть записано в активный DSP-банк в 12-разрядном виде.

В видеопроцессоре реализованы следующие функции цифровой обработки:

- *усреднение первого и второго полей кадра* (второго операнда не требуется);
- *исключение фона* из вводимого изображения; функция работает по принципу сравнения значений одинаково расположенных пикселей в двух кадрах (оцифровываемым в данный момент и записанным в DSP банке) и заменой их на нулевое значение, если разность по модулю не превышает порога, заданного в регистре **DSP_Level** (8 бит);
- *вычитание двух кадров*; в качестве вычитаемого используется изображение из DSP-банка;
- *сложение двух кадров*; для исключения переполнения разрядной сетки операнды перед сложением делятся на 2;
- *наложение пространственной маски AND, OR, XOR.*

Регистры **ADC_Accum** и **STB_Accum** осуществляют управление режимом накопления кадров входного ТВ-сигнала, поступающего от подсистем ввода изображения и строба. Если задана какая-либо функция цифро-

вой обработки, то она выполняется в конечной стадии режима накопления при вводе последнего кадра.

В режиме накопления кадров выдача прерываний от подсистемы ввода осуществляется по каждому введенному кадру, а бит готовности (бит D0 регистра **Reg_Stat**) будет установлен по окончании ввода последнего из серии накапливаемых кадров. Процесс накопления можно принудительно остановить, сбросив бит D8 регистра **ADC_Ctrl**.

Биты	Описание
2..0	Выбор количества накапливаемых кадров: D2 D1 D0 0 0 0 – нет накопления 0 0 1 – 2 кадра 0 1 0 – 4 кадра 0 1 1 – 8 кадров 1 0 0 – 16 кадров 1 0 1 – 32 кадра 1 1 0 – 64 кадра 1 1 1 – не используется
31..3	Не используются

Еще одним свойством видеопроцессора является возможность проводить с «оцифрованным» сигналом линейное преобразование по формуле

$$y = a \cdot (x - b),$$

где x – сигнал на входе преобразователя;
 y – сигнал на выходе преобразователя;
 a – константа повышения контрастности;
 b – константа смещения (12 бит).

Суть этого преобразования – повышение контрастности вводимого изображения. Необходимость контрастирования может возникнуть в случае слабого ТВ-сигнала (если не хватает аналоговых регулировок) или сильно зашумленного сигнала (с применением накопления). Фактически знак умножения означает сдвиг старших значащих разрядов каждого пикселя влево, а знак вычитания – перемещения информационной части в область младших разрядов. Значение константы смещения (12 бит) определяется регистрами **ADC_Offset** для подсистемы ввода изображения и **STB_Offset** для подсистемы ввода строба. Значение константы повышения контрастности определяется регистрами **ADC_Contrast** и **STB_Contrast** соответственно.

Биты	Описание
1..0	Значение константы повышения контрастности: D1 D0 0 0 – 1 0 1 – 2 1 0 – 4 1 1 – 8
31..2	Не используются

Гистограмма яркости входного сигнала вычисляется по 8 старшим битам каждого пикселя всякий раз при вводе кадров и представляет собой массив из 256 эле-

ментов, где каждый элемент – это количество пикселей с одним значением яркости. Область вычисления – весь кадр или строб, определяется регистром **ADC_MinMax**.

Для построения гистограммы элементы массива последовательно друг за другом считываются из регистра **Reg_Hist**. Перед началом считывания необходимо записать в регистр **Reg_Hist** любое произвольное значение, тем самым установив счетчик элементов массива в 0. Построение гистограммы проводится после окончания фиксации кадра или строа.

Пример работы видеопроцессора RT-851VP

На рисунках ниже представлены результаты работы видеопроцессора RT-851VP.

Условия съемки – ТВ-камера WATEC-902H, калиброванная освещенность $10E^{-7}$ люкс, электронно-оптический преобразователь (ЭОП).

На *рис. 3а* показан входной кадр, представляющий собой шумовую составляющую ПЗС-матрицы и ЭОП.

На *рис. 3б* – результат обработки, время интеграции – около 4 с. Использовались следующие функции:



Рис. 3. Пример кадра на входе (а) и выходе (б) устройства обработки

- автоматическая настройка на максимальный динамический диапазон АЦП (12 бит);
- рекурсивное накопление по формуле $F_n = k \cdot F_n + (1-k) \cdot F_{n-1}$, при $k=1/32$; 18-битный аккумулятор;
- аппаратное контрастирование в 4 раза.

Использование функции рекурсивного накопления позволяет снизить требования к качеству ПЗС-матрицы, повышая соотношение сигнал/шум.

Литература

1. Техническая документация и руководство пользователя на видеопроцессоры серии RT-8xxVP. ООО «Растр Технолджи». М., 2002, www.rastr.net.
2. Бондаренко А.В., Докучаев И.В. Использование мультимедийного видеопроцессора RT-850 для визуализации и обработки ангиографических изображений в реальном времени. Труды I Евразийского конгресса, V национальной конференции по медицинской физике и инженерии «Медицинская физика-2001». М., 18–22 июня 2001.
3. Бондаренко А.В. RT-850 – мультимедийный видеопроцессор реального времени. Доклад на 3-й Международной конференции и выставке «Цифровая обработка сигналов и ее применение». М., 2000, 29 ноября – 1 декабря.
4. Бондаренко А.В. Профессиональный мультимедийный видеопроцессор реального времени. Труды международной конференции по биомедицинскому приборостроению «Биомедприбор-2000». 24–26 октября, М., 2000, т.1, с. 183.
5. Стешенко В.Б. Программируемые логические интегральные схемы: обзор архитектур и особенности применения в аппаратуре ЦОС. // Цифровая обработка сигналов, 2000, № 2, с. 39.
6. Стешенко В.Б. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов. М.: ДОДЕКА, 2000.

Статья подготовлена в ООО «Растр Технолджи» (Raster-msk@mtu-net.ru, www.rastr.net)

УДК 621.391.25

Декодирование файлов в стандарте MP3

А.О. Валитов, А.В. Герус, И.Ф. Комлик, М.С. Куприянов

Введение

В последнее время в мире информационных технологий наблюдается тенденция смены аналоговых методов передачи данных цифровыми методами. Многие западные специалисты предсказывают замещение в ближайшие двадцать лет сегодняшних FM-радиостанций узлами вещания DBA (Digital Audio Broadcast). В мире повсеместно внедряется телевидение в цифровом формате, например в стандарте High Definition TeleVision (HDTV).

Причина этих тенденций в преимуществах, которые дает цифровая передача данных по сравнению с аналоговыми методами: лучшая помехоустойчивость, потенциально большая производительность при узкой полосе пропускания, возможность уменьшения мощности передачи.

Однако при этом возникает ряд проблем. Одна из проблем состоит в том, что аудио- и видеоданные требуют для прямой (без предварительного кодирования) цифровой передачи большую полосу пропускания. Для решения этой проблемы используется дополнительная предобработка в целях уменьшения объема передаваемых данных.

Методы уменьшения объема данных

Рассмотрим различные группы методов сжатия данных и их применимость для предобработки аудио- и видеосигналов в целях передачи.

Первая группа методов основана на изменении уровней квантования сигнала. Самый простой способ состоит в увеличении размера кванта при квантовании, иными словами – в уменьшении разрядности представления данных (например, с 16 до 8 бит). Более эффективными являются методы нелинейного квантования.

Вторая группа методов снижения объема передаваемых данных основана на изменении частоты дискретизации соответственно в меньшую сторону (например, данные формата AUDIO CD можно преобразовать с частоты дискретизации 44,1 кГц в частоту 22,05 кГц после применения соответствующего цифрового фильтра).

Обе группы методов можно охарактеризовать как «методы уменьшения объемов данных с потерями», и обе эти группы приводят к очень значительным потерям, недопустимым для аудио- и видеосигналов высокого качества.

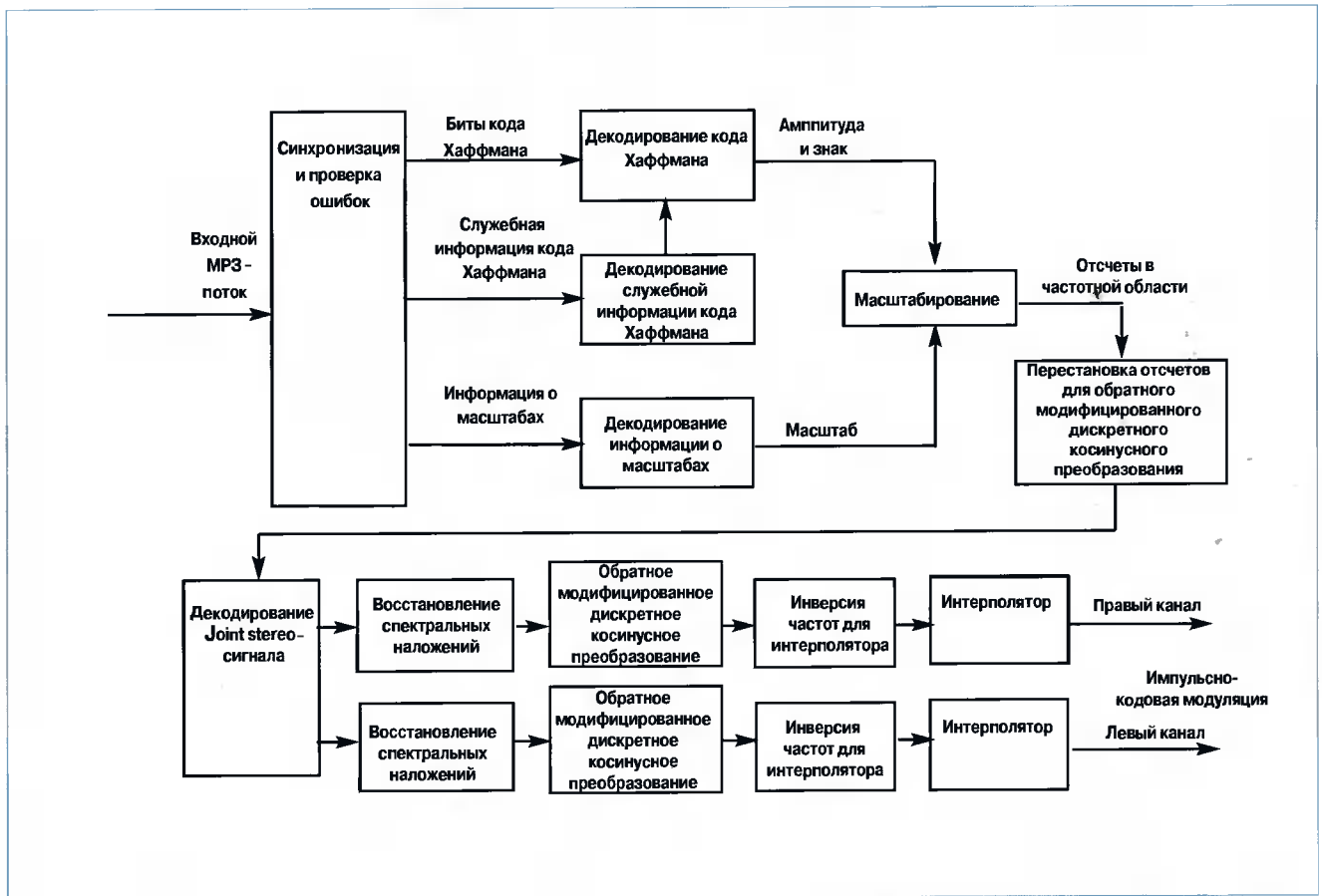
Наиболее приемлемое решение было найдено за счет сочетания рассмотренных выше методов с пси-

Описана методика проектирования и тестирования программного MP3-декодера на основе микропроцессора семейства DSP56800E. Результатом проектирования явился программный продукт с широкими возможностями его применения в различных встроенных аудиоприложениях и системах передачи данных.

хоакустической моделью восприятия человеком звуковых сигналов и моделью зрительного восприятия. Одними из первых в этой группе методов были разработаны Audio Spectral Perceptual Entropy Coding (ASPEC) и Masking pattern adapted Universal Subband Integrated Coding And Multiplexing (MUSICAM). Эта группа методов, так же, как и первые две рассмотренные, являются «методами уменьшения объемов данных с потерями». Но их результат воспринимается человеческим ухом и глазом почти так же, как и исходный сигнал, поскольку из сигнала удаляются лишь незначительные и слабовоспринимаемые человеком составляющие.

Общий стандарт

Позже рабочей группой Moving Pictures Expert Group (MPEG) в составе международной организации по стандартизации (ISO – International Organization of Standardization) был разработан общий стандарт для компрессии цифровых аудио- и видеосигналов, названный MPEG-1. Он состоял из трех частей: компрессия аудиосигналов, компрессия видеосигналов и системная часть, описывающая передачу аудио- и видеосигналов в едином потоке. Часть, описывающая компрессию аудиосигнала, состоит из трех алгоритмов кодирования, называемых: Layer I, Layer II, Layer III. Эти алгоритмы указаны с возрастанием сложности и степени сжатия данных. Layer I и Layer II нашли применение в закрытых системах, где присутствует широкая полоса пропускания и «стоимость» передачи дешева, т.е. объем трафика не критичен. Используются они достаточно редко. А Layer III является экономически привлекательным в системах, где «стоимость» передачи высока, узкая полоса пропускания или критичен объем информации, подлежащий хранению. Однако декодирование MP3 (MPEG 1 – Layer III) потока в реальном времени можно осуществлять уже с использованием процессора Pentium75. Поэтому благодаря относительно небольшим требованиям программного MP3-декодера к производительности процессора, наибольшей степени сжатия звуковых сигналов и малой потере качества этот формат получил широкое распространение и сейчас является наиболее часто используемым для хранения аудиоданных. Он используется как во мно-



гих портативных и стационарных устройствах широкого потребления, так и в профессиональной аудиотехнике. В этих условиях задача декодирования MP3-данных становится очень актуальной. Ниже приведена блок-схема MP3-декодера.

Методы решения задачи декодирования MP3

Наиболее простым решением задачи декодирования MP3-данных является использование специализированных интегральных микросхем, аппаратных MP3-декодеров, таких, как TL7232MD SAMSUNG, LC82310 SANYO, SPCA75xA SUNPLUS, и т.п. Таких микросхем на рынке множество, они характеризуются низкой ценой \$5–12 за 1 шт. в партии от 1000 шт., возможностью декодирования лишь с одинарной скоростью и очевидными ограничениями по функциональности. Хотя обычно функциональность данных микросхем не ограничивается аппаратным MP3-декодированием, а дополняется еще несколькими возможностями такими, как «антишок», эквалайзер, и т.п., все равно их набор всегда жестко ограничен.

Одним из наиболее быстрых решений является покупка готового программного решения по декодированию у сторонней компании. Так, в частности, имеется программная реализация декодера на микросхемах TMS320VC5410-100 фирмы Texas Instruments Inc., стоимость которых в партии от 1000 шт. составляет \$31. Декодеры требуют от 35 до 50 MIPS для одной скорости и примерно по 30–35 килобайт памяти программ и 28–32 килобайт памяти данных. Есть возможность

вносить также изменения в программу микропроцессора. Другим методом является использование дешевого DSP и микросхемы программируемой логики. При этом сигнальный процессор используется для выполнения дружественных DSP-операций, таких, которые могут эффективно использовать возможности данного типа процессоров, т.е. блоки алгоритма, где превалируют операции умножения, сложения, умножения с накоплением на аккумуляторе и циклы с фиксированным числом повторений. Программируемая логика используется для выполнения бит ориентированных операций и преобразований, таких, как выборка служебных данных из потока битов, декодирование кода Хаффмана согласно таблицам и выбранным служебным данным и перестановка отсчетов (reordering). Привлекательным в данном способе является то, что после составления VHDL-описания содержимого программируемой логики исчезает ориентация на конкретную микросхему. Для полной платформенной независимости главную программу сигнального процессора можно реализовать на языке Си. При переходе на другую платформу потребуются переделать лишь часть программы, касающейся взаимодействия процессора с программируемой логикой. Однако эффективность встроенного приложения на Си с большими объемами вычислений во многом определяется эффективностью используемого Си компилятора. Использование данного решения в изделии, которое не будет выпускаться большими партиями, также неперспективно, поскольку в этом слу-

чае исполнение логики в интегральном виде экономически не выгодно (слишком большой вклад в себестоимость), а использование программируемой логики дорого.

Наиболее предпочтительным из рассматриваемых вариантов является программная реализация алгоритмов декодирования MP3-данных. Основные достоинства данного варианта: возможность выбора наиболее подходящего семейства процессоров как по цене, так и по характеристикам, полный контроль над соотношением качество звучания/производительность декодера, выбор максимально удобного интерфейса взаимодействия декодера с остальной системой. Для профессиональной техники важны качество звучания и возможность его контроля. Эти возможности в полной мере реализуются при программной реализации декодера на основе процессора обработки сигналов.

Реализация MP3-декодера на DSP56800E

Для проектирования было выбрано семейство процессоров цифровой обработки сигналов DSP56800E фирмы Motorola. Данное семейство считает в себе функциональности процессора цифровой обработки сигнала и микроконтроллеров. Функциональность DSP использовалась для реализации вычислительных алгоритмов. Функциональность микроконтроллера полезна при решении интерфейсных задач и задач управления (например, CD-приводом), а также при работе с MP3-потокком битов (bit stream).

Программа декодирования MP3-потокка написана на языке Си согласно стандарту MPEG1 ISO/IEC 11172-3 с использованием арифметики с плавающей точкой и динамическим распределением памяти.

Программа была адаптирована для использования в процессорах семейства DSP56800E. Очевиден тот факт, что все арифметические операции в формате с плавающей точкой значительно более длительные, чем в формате с фиксированной точкой. Поэтому хранение данных и вычисления были переведены в дробно-рациональный формат с фиксированной точкой. Этот формат чисел был выбран потому, что диа-

пазон представления чисел, участвующих в MP3-декодировании, выходит за интервал $[-1, +1]$. Динамическое выделение памяти также было заменено статическим для ликвидации затрат на управление динамической памятью.

Все основные блоки алгоритма были переписаны на ассемблере семейства DSP56800E в целях оптимизации времени их выполнения.

В результате были получены следующие характеристики MP3-декодера:

Память программ:	34 килобайта
Память данных:	32 килобайта
Производительность:	46 MIPS – одинарная скорость 92 MIPS – двойная скорость

Согласно полученным метрикам MP3-декодер без внешней памяти можно реализовать на следующих процессорах семейства DSP56800E: DSP56855, DSP56857, DSP56858.

Тестирование программы

Программа была проверена на предмет корректности декодирования с помощью MP3-файлов, закодированных четырьмя различными программами. Файлы с частотами дискретизации 44,1, 48, 32 кГц и с интенсивностью потока данных (bitrate) 32, 48, 64, 96, 112, 128, 160, 192, 256, 320 кбит/с и переменной интенсивностью (Variable Bite Rate-VRB), моно и стерео, с тремя различными способами кодирования стереосигнала: stereo, joint stereo, dual channel. Стереосигналы с частотой дискретизации 48 и 32 кГц представляют собой синусы с разными амплитудами и частотами по каждому каналу, а моносигналы просто сумму двух синусоидальных сигналов. Звуковые сигналы с частотой дискретизации 41,1 кГц представляли собой музыкальные композиции, взятые с дисков формата AUDIO CD. Тем самым были охвачены почти все возможные режимы работы MP3-декодера. В результате проверки не были получены искажения частот, фаз и амплитуд синусоидальных сигналов. Музыкальные композиции проверялись прослушиванием, результат подобного тестирования сугубо субъективный и не имеет объективных показателей.

Литература

1. Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part1: System. ISO/IEC 11172-1:1993.
2. Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part3: Audio. ISO/IEC 11172-3:1993/Cor.1:1996(E).
3. DSP56800E 16-Bit Digital Signal Processor Core Reference Manual. DSP56800ERM/D Rev. 2.0, 12/2001.

УДК 621.396.969

Метод устранения фазовых искажений в устройствах радиометрии

С.В. Иванов, В.В. Витязев, В.В. Езерский

Введение

Одним из перспективных направлений применения устройств радиометрии является измерение и контроль уровня химических веществ в различных емкостях. К числу наиболее известных примеров такого типа устройств относится радиодальномер с частотной модуляцией (ЧМ) [1, 2]. В ЧМ-дальномерах используется так называемый частотный метод измерения дальности (или уровня). Принцип функционирования ЧМ-дальномера основан на применении непрерывного зондирующего ЧМ-сигнала, частота которого в идеальном случае изменяется по линейному закону. При отражении от поверхности, находящейся на расстоянии R , в устройстве измерения возвращается сигнал, задержанный на время t_R . На выходе приемного устройства формируется так называемый разностный сигнал, мгновенная частота которого представляет собой разность между мгновенными частотами отраженного и зондирующего сигналов. Задержка отраженного сигнала приводит к пропорциональному смещению во времени его мгновенной частоты таким образом, что разность между излученной и принятой частотой постоянна в течение всего времени наблюдения за отраженным сигналом (при неизменном измеряемом расстоянии). По значению приращения частоты можно судить о дальности до облучаемой поверхности. Рассмотренный метод справедлив лишь при неограниченном линейном возрастании частоты излучаемого ЧМ-сигнала. По объективным причинам реализация такого закона изменения частоты сигнала физически невозможна. Чтобы практически реализовать метод измерения по частоте биений, вводится частотная модуляция сигнала по периодическому закону. Наиболее распространенным случаем является симметричный пилообразный закон изменения частоты сигнала. Рис. 1 поясняет принцип работы ЧМ-дальномера для рассматриваемого закона изменения частоты зондирующего сигнала.

На рис. 1а изображены законы изменения зондирующего и отраженного от поверхности ЧМ-сигналов ($f_d(t)$ и $f_n(t)$). Закон изменения частоты сигнала биений $F_0(t)$ показан на рис. 1б. Вид сигнала на выходе приемника $U_0(t)$ представлен на рис. 1в. В данном случае сигнал $U_0(t)$ представляет собой чистую синусоиду на интервале наблюдения, соответствующем половине периода закона модуляции ЧМ-сигнала T_m .

Разработан эффективный метод устранения скачков фазы в устройствах радиометрии, использующих преобразование измеряемого параметра в частоту гармонического сигнала. В основе метода лежит принцип распознавания и удаления участков сигнала с фазовыми скачками. Представлена структура устройства, реализующего данный метод, важным элементом которой является фильтр линейного предсказания. Показана эффективность применения интерполяции сигнала для повышения точности согласования участков сигнала до и после момента скачка фазы.

Постановка задачи

Математическая модель дискретного периодического сигнала, соответствующего сигналу, показанному на рис. 1в, представляет собой следующее выражение

$$U_0[nT] = A_0 \cos(2\pi \Phi[nT] + \varphi_0) + \eta[nT]$$
$$\Phi[nT] = |F_0 nT|;$$
$$-\frac{T_m}{2} \leq nT \leq \frac{T_m}{2},$$

где F_0 – разностная частота, непосредственно несущая информацию об измеряемом параметре, $\eta[nT]$ – дискретный гауссовский случайный процесс, $\Phi[nT]$ – периодическая функция с периодом T_m .

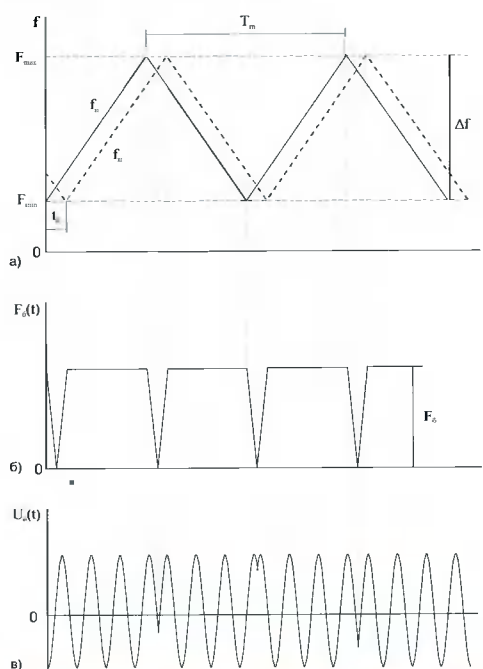


Рис. 1

Спектр сигнала, описываемого математической моделью (1), содержит в себе множество гармоник, кратных частоте $F_m=1/T_m$. Таким образом, присутствие скачков фазы (рис. 1в) при интервале наблюдения, превышающем $T_m/2$, приводит к «расползанию» спектра сигнала и смещению максимума спектральной мощности от основной частоты F_0 в сторону одной из кратных гармоник. Проведенные исследования показали, что наличие этих факторов не позволяет какими-либо известными методами фильтрации и спектрального анализа обеспечить необходимую высокую точность оценки разностной частоты F_0 .

Задача, которую необходимо решить для обеспечения высокой точности оценки разностной частоты, формулируется следующим образом: необходимо разработать метод коррекции сигнала в целях устранения в нем фазовых скачков. В качестве критерия, характеризующего точность работы устройства, выбрана эффективная ширина спектра сигнала на выходе устройства: $\delta F_{U,эфф}$. Скачки фазы будут считаться устраненными, если эффективная ширина спектра сигнала на выходе устройства коррекции удовлетворяет следующему условию

$$\delta F_{U,эфф} < F_m, \quad (2)$$

где $F_m=1/T_m$ – частота повторения периодического пилообразного закона изменения фазы сигнала (1). Если же значение критерия точности превышает заданную величину, то в спектре находится, как минимум, две достаточно мощные составляющие, что свидетельствует о неточной коррекции сигнала.

Метод устранения фазовых искажений

В основе работы предлагаемого цифрового метода – удаление участков сигнала, содержащих фазовые искажения, и непрерывная «сшивка» соседних неискаженных участков. Представленный метод коррекции используется не только в случае, когда моменты возникновения фазовых скачков известны, но и сам способен распознавать моменты возникновения этих скачков. Важным элементом подсистемы, реализующей этот метод, является адаптивное устройство линейного предсказания [3, 4]. Структура подсистемы с применением устройства адаптивного линейного предсказания приведена на рис. 2.

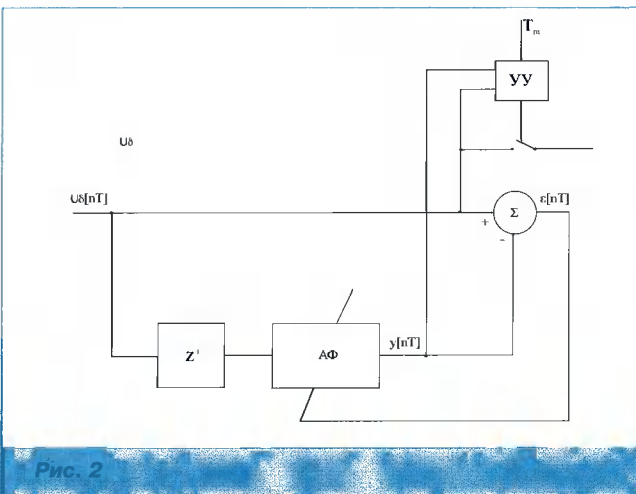


Рис. 2

Принцип работы данного метода следующий.

- Отсчеты входного сигнала записываются в выходной массив и одновременно поступают на вход устройства линейного предсказания.
- Если признак скачка фазы находится в неактивном состоянии, то отсчет входного сигнала записывается в массив.
- Если признак скачка фазы установился в активное состояние (момент возникновения скачка считаем известным), то устройством управления (УУ) выдается сигнал запрета записи последующих отсчетов входного сигнала. При этом запоминается последнее записанное в массив значение входного сигнала и разность между этим значением и предсказанным отсчетом сигнала в момент скачка фазы.
- В дальнейшем ожидается появление отсчета сигнала, имеющего близкое (с некоторым допуском) значение к предсказанному значению в момент скачка фазы (вычисляется разность между текущим и запомненным значениями сигнала и сравнивается с запомненным значением разности).
- При выполнении условий соответствия признак скачка фазы возвращается устройством управления (УУ) в неактивное состояние, а запись отсчетов входного сигнала в выходной массив разрешается.

Временная диаграмма, иллюстрирующая принцип работы данного алгоритма, приведена на рис. 3. На рис. 3а показан входной сигнал, на рис. 3б – выходной сигнал фильтра линейного предсказания. Выделенный жирным участок входного сигнала не записывается в выходной массив. Результаты моделирования работы алгоритма приведены на рис. 4, 5. На рис. 4а, 4б показаны входной сигнал и массив отсчетов выходного сигнала устройства коррекции, а на рис. 5а, 5б – их спектры. Входной сигнал описывается математической моделью (1). Участок входного сигнала, не записываемый в выходной массив, расположен между t_A и t_B (рис. 4). Из рис. 5 видно, что условие (2), характеризу-

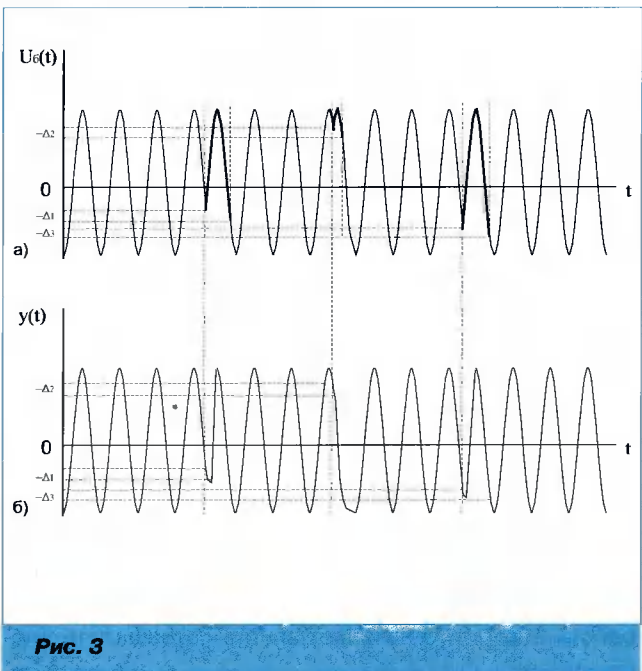


Рис. 3

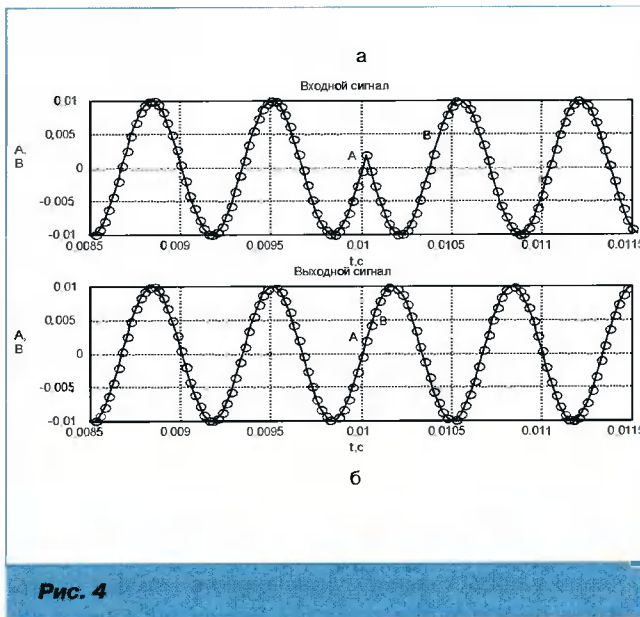


Рис. 4

общее качество работы устройства, выполняется ($F_m = 50 \text{ Гц}$), т.е. можно говорить об устранении влияния скачков фазы на точность оценки частоты полезного сигнала.

Исследования чувствительности устройства к уровню шумов показали, что адаптивный фильтр линейного предсказания малого порядка ($N=2$ или $N=4$), применяемый в данном устройстве, функционирует с достаточно низким уровнем ошибки предсказания, если значение отношения сигнал/шум удовлетворяет условию: $P_{S/\eta} \geq 40 \text{ дБ}$, что позволяет устройству практически безошибочно вырезать участки сигнала, содержащие фазовые скачки.

Устройство, реализующее данный алгоритм, может быть использовано также в случае, когда моменты возникновения фазовых искажений неизвестны. Функции подсистемы, позволяющей обнаружить искажение фазы, выполняет адаптивный фильтр линейного предсказания. Критерием, позволяющим утверждать о возникновении скачка фазы, может служить изменение знака производной в реальном сигнале по отношению к предсказанному отсчету сигнала. Рис. 6 иллюстрирует принцип определения момента возникновения скачка фазы в сигнале. На рис. 6а, 6б показаны входной и предсказанный адаптивным фильтром сигналы, а на рис. 6в, 6г – законы изменения знаков их производных.

По этим данным можно видеть, что в момент возникновения скачка фазы знак производной входного сигнала (рис. 6в) изменяется по отношению к знаку производной предсказанного значения сигнала (рис. 6г). Это обстоятельство и позволяет сделать вывод о наличии во входном сигнале скачка фазы.

Исследование алгоритма коррекции сигнала показало, что данный метод обеспечивает требуемую точность решения поставленной задачи при работе с сигналами, на один период которых приходится не менее 20 дискретных отсчетов. Если на период сигнала приходится меньшее количество отсчетов, то на точность работы алгоритма начинает влиять погрешность дис-

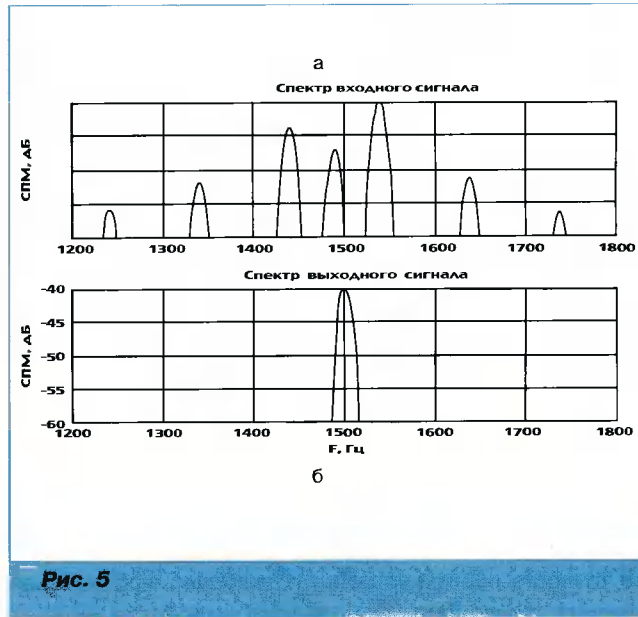


Рис. 5

кретизации сигнала, и она ухудшается. В то же время в периоде сигнала, частота которого находится в интервале $F \in [F_{\min} \dots F_{\max}]$, при заданной частоте дискретизации F_s в некоторых случаях может содержаться и меньшее число точек.

Для повышения точности функционирования методов коррекции сигнала при обработке высокочастот-

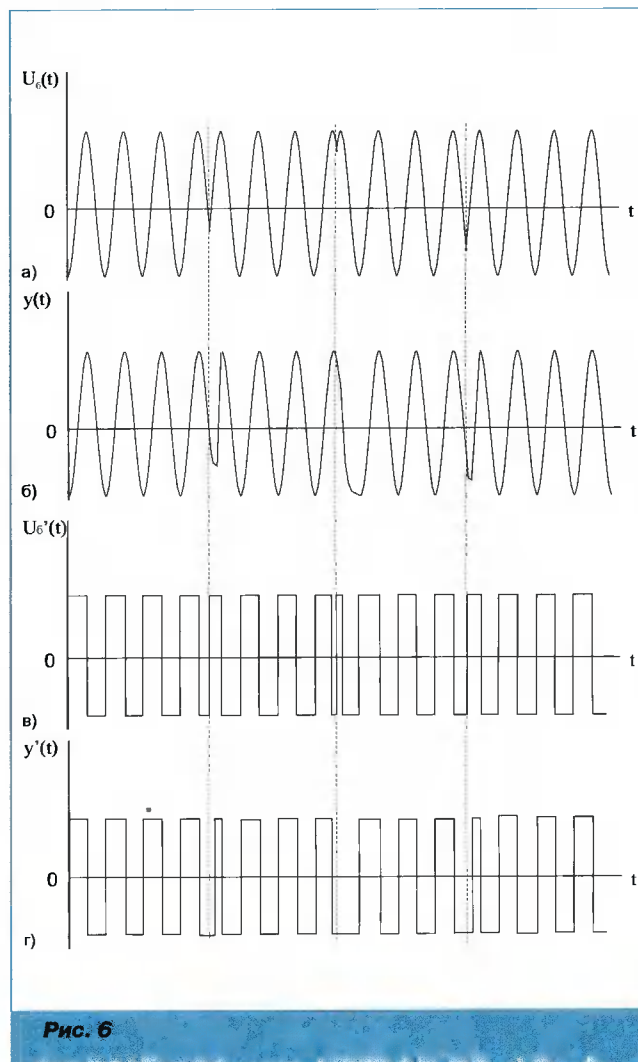


Рис. 6

ных сигналов необходимо увеличить частоту дискретизации входного сигнала в такое число раз (ν), чтобы количество дискретных отсчетов на период сигнала удовлетворяло условию $K_p \geq 20$. Это можно обеспечить посредством применения метода интерполяции сигнала [5].

На рис. 7 приведена функциональная схема устройства коррекции сигнала, основанная на линейном предсказании сигнала с использованием метода интерполяции. Обработка сигнала ведется на повышенной скорости, что обеспечивает увеличение точности функционирования алгоритма коррекции на высоких частотах. На выходе подсистемы устройство децимации осуществляет возврат к первоначальной частоте дискретизации обрабатываемого сигнала.

В данной схеме УУ – устройство управления, осуществляющее разрешение (0) или запрет (1) записи данных в выходной массив, T_m – момент возникновения скачка фазы (предполагается известным).

На рис. 8, 9 приведены результаты моделирования работы алгоритма коррекции сигнала, основанного на линейном предсказании. На рис. 8 приведен спектр выходного массива устройства коррекции, в котором содержатся отсчеты сигнала с частотой $F_0=3300$ Гц и частотой дискретизации $F_s=40000$ Гц, а на рис. 9 – спектр выходного массива, в котором содержится сигнал с той же частотой, обработанный на частоте дискретизации $F'_s=200000$ Гц.

При сравнении спектров на рис. 8 и рис. 9 видно, что в первом случае наблюдается присутствие боковых составляющих, уступающих по мощности основной составляющей на 20 дБ и более, а также небольшое смещение основной составляющей, что свидетельствует о недостаточно точном согласовании участков сигнала до и после моментов скачков фазы.

Выводы

Представленный цифровой метод на основе устройства линейного предсказания позволяет эффективно устранить фазовые искажения в разностном

сигнале устройства измерения уровня за счет вырезания искаженных участков сигнала. Применение интерполяции при недостаточно высокой частоте дискретизации приводит к повышению точности согласования участков сигнала до и после момента скачка фазы. Для обеспечения высокой точности функционирования необходимо обеспечить значение отношения мощностей сигнал/шум на входе устройства не ниже 40 дБ.

Литература

1. Финкельштейн М.И. Основы радиолокации. М.: Радио и связь, 1983.
2. Атаянц Б.А, Болонин В.А., Езерский В.В., Кагаленко Б.В. Адаптивный частотно-модулированный уровень // VI МНТК «Радиолокация, навигация, связь», Воронеж, 2000.
3. Уидроу Б., Стирнз С.. Адаптивная обработка сигналов. М.: Радио и связь, 1989.
4. Коуэн К.Ф.Н, Грант П.М. Адаптивные фильтры. М.: Мир, 1988.
5. Витязев В.В. Цифровая частотная селекция сигналов. М.: Радио и связь, 1993.

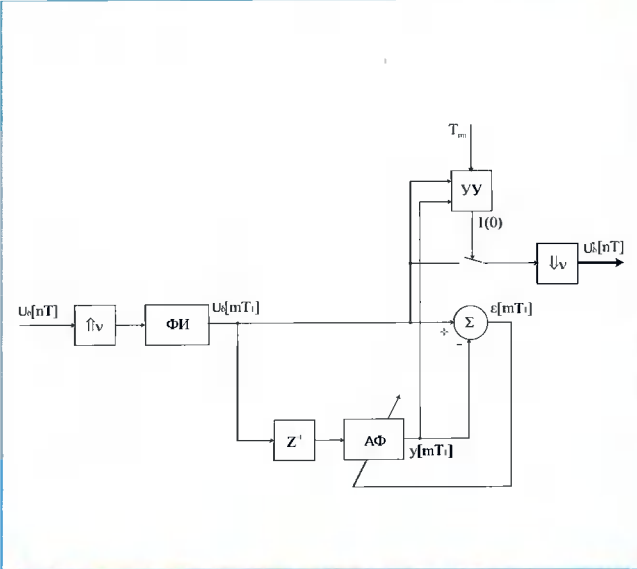


Рис. 7.

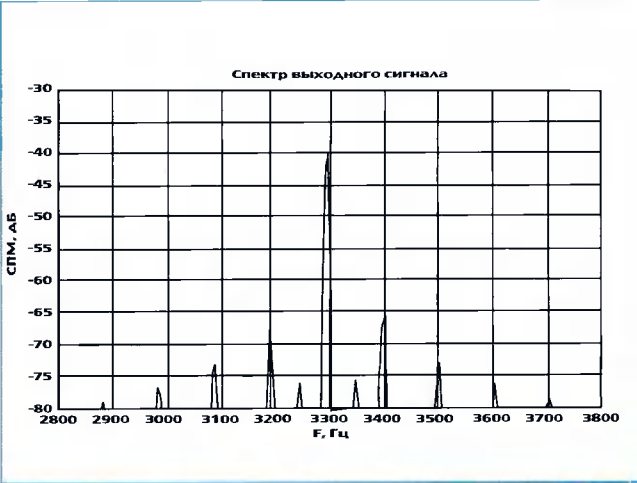


Рис. 8

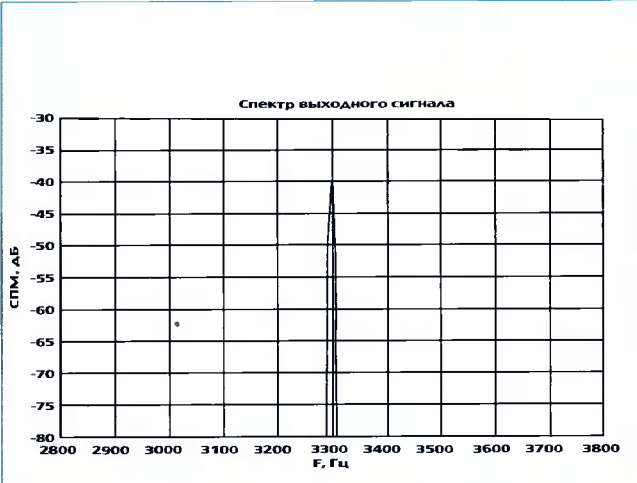


Рис. 9

УДК 621.395

Быстрая разработка VoIP и других приложений телефонии

Н.Н. Абкаиров, А.А. Фрунзе

Телефония сегодня

Современная телефония не ограничивается только передачей голосовых сообщений по телефонным и радиотелефонным линиям. Она также предоставляет возможность передавать данные через модем и подключаться к Internet из офиса или дома, дает возможность передавать различные по типу и содержанию данные. Многие компании снабжают свои продукты дополнительной функциональностью для передачи информации по телефонным каналам. Например, охранные системы могут автоматически делать звонки на пульт дежурного в охранной организации, когда злоумышленник пытается проникнуть в охраняемое помещение, или в случае возгорания вызвать пожарную бригаду и бригаду «скорой медицинской помощи». Новые счетчики в системах водо-, газо- и энергоснабжения обеспечивают удаленный контроль за расходом и использованием ресурсов путем накопления информации и передачи ее в центр. Службы поддержки покупателей и пользователей часто включают оборудование для эффективного перенаправления входящих звонков в нужный отдел или для ответа на стандартные вопросы на базе меню. Таких примеров можно привести множество.

Способы реализации функций телефонии

В целях реализации разнообразных телефонных функций инженерам-разработчикам обычно требуются соответствующие аппаратные средства для подключения к телефонным линиям. Помимо подключения к линиям существует еще и необходимость подавлять эхо, генерировать и детектировать различные телефонные сигналы, такие, как DTMF, сигнал ответа станции, сигнал «занято», и т.п. Все эти функции могут быть выполнены с помощью специализированных микросхем (чипсетов), реализующих фиксированную программу, но это не является единственным способом получить желаемую функциональность. Другой способ – реализовать перечисленные функции программным методом на цифровом сигнальном процессоре (DSP). Конечно, разработчик может воспользоваться готовым интегрированным решением на основе чипсета, но чипсет обычно ограни-

В статье сотрудников компании SPIRIT Corp. Н. Абкаирова и А. Фрунзе описывается программное обеспечение (ПО) Client Side Telephony (CST), отличительная особенность которого состоит в интеграции различных абонентских телефонных функций, используемых в телекоммуникационных задачах. ПО было разработано специально для популярного сигнального процессора TMS320C54 в тесной кооперации с фирмой Texas Instruments (США). В настоящее время чип C54CST, на котором стоит логотип SPIRIT, выпускается фирмой Texas Instruments большими тиражами. Поэтому данная статья, предназначенная для широкой аудитории потенциальных пользователей и инженеров, заинтересованных в эффективном и разнообразном использовании чипа C54CST, будет несомненно ей интересна. Компания SPIRIT Corp. разработала также более «продвинутое» версии этого ПО для DSP C54 и C55, которые призваны расширить круг пользователей и разработчиков телефонных приложений, и оболочку для них. Краткую информацию об этом читатель также найдет в данной статье. По меткому определению Уилла Страуса, президента фирмы Forward Concepts по исследованию рынка DSP, и Марка Метсона, менеджера самой популярной процессорной линии C5000 фирмы Texas Instruments, высказанного на презентации чипа C54CST в Далласе в ноябре 2001 г., чип C54CST, в котором интегрированы 14 популярных телефонных алгоритмов, представляет собой «армейский швейцарский нож» для разнообразных коммуникационных приложений. Читатель получит из статьи достаточное представление о самом ПО и его применении. Более подробная информация может быть получена из документации на ПО, объем которой составляет к настоящему времени сотни страниц.

*В.А. Свириденко, профессор, д.т.н.,
вице-президент по технологиям SPIRIT Corp.*

чивает возможности изменять, контролировать и расширять собственную функциональность. Использование же цифрового процессора с собственным ПО или ПО третьих лиц может сделать решение гораздо более гибким, настраиваемым, чем решение на основе чипсета. Данная статья концентрируется на этом (втором) способе реализации телефонных функций.

Решение	Преимущество	Недостаток
Чипсет	Может быть уже интегрирован	Недостаточно гибкости
DSP	Может быть очень гибким	Потребуется интеграция

Типичные шаги при реализации функций телефонии включают в себя выбор DSP-платформы и поиск поставщика телефонных алгоритмов (ПО, которое реализует функции телефонии). Выбор платформы и поиск поставщика ПО может потребовать некоторого времени. Сделав выбор и найдя ПО, можно приниматься за разработку собственной части ПО и начать интеграцию всего ПО и аппаратного обеспечения. Но вероятнее всего, эта стадия разработки потребует недель или даже месяцев, прежде чем конечный продукт будет готов, т.к. разработка, отладка, интеграция, тестирова-

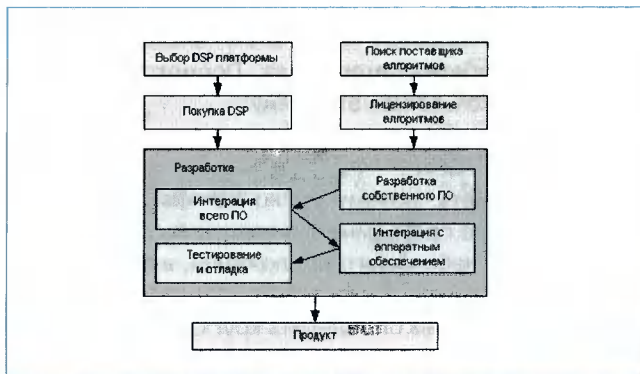


Рис. 1. Реализация функций телефонии на DSP

ние, повторная отладка и коррекция – все это займет много времени.

Знакомьтесь с C54CST

К счастью, процесс, изображенный выше, может быть сильно упрощен. В компании SPIRIT Corp. была разработана специальная телефонная среда – Client Side Telephony framework (или коротко CST), чтобы облегчить жизнь разработчикам продуктов с телефонными функциями. Среда CST включает в себя 14 телефонных алгоритмов, также разработанных в компании, таких, как эхоподавитель, алгоритмы для генерирования и детектирования телефонных сигналов, ADPCM-кодек, модемные алгоритмы и как дополнения (addons) факсовые алгоритмы и вокодеры. В настоящий момент среда CST доступна в виде процессора TMS320C54CST от компании Texas Instruments. Модификация C54CST – это фактически процессор TMS320C5406 с алгоритмами и телефонной средой, находящимися во внутреннем ПЗУ процессора. Телефонная среда связывает алгоритмы вместе, упрощая их использование и процесс интеграции. Среда CST, как целое, уменьшает время разработки и интеграции независимо от того, ведется ли разработка профессионалами в области цифровой обработки сигналов или новичками.

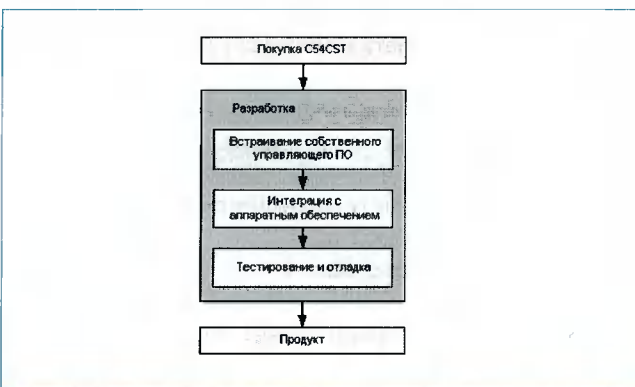


Рис. 2. Реализация функций телефонии на C54CST

Типичные приложения для среды CST:

- ◆ Устройства с необходимостью доступа к Internet
- ◆ Voice/Fax over IP-системы, мини-АТС
- ◆ Кассовые и торговые автоматы

- ◆ Банкоматы
- ◆ Системы проверки кредитных карт и чеков
- ◆ Охранные системы
- ◆ Счетчики расхода ресурсов
- ◆ Измерительные системы и системы удаленного сбора информации
- ◆ Таксофоны
- ◆ Цифровые автоответчики
- ◆ Устройства для настройки оборудования

Процессор C54CST содержит следующие алгоритмы:

- ◆ Модемные алгоритмы:
 - V.32bis/V.22bis (до 14,4 кбит/с)
 - V.42 – коррекция ошибок, V.42bis – сжатие данных
 - Modem Integrator – интегрирует остальные модемные алгоритмы и включает в себя V.14 синхронное/асинхронное преобразование
- ◆ Алгоритмы для телефонных сигналов:
 - UMTD/UMTG (Universal Multi-frequency Tone Detector/Generator – универсальный детектор/генератор многочастотных тонов)
 - DTMF-детектирование/генерирование
 - Call Progress Tone детектирование/генерирование
 - Caller ID типы 1 and 2
- ◆ Голосовые алгоритмы:
 - G.168 подавление эха линии (с длиной «хвоста» 16 или 32 мс)
 - G.726 ADPCM сжатие (16–40 кбит/с), включает в себя G.711 PCM
 - Automatic Gain Control (AGC) – АРУ
 - Voice Activity Detection (VAD) – детектор речи
 - Comfort Noise Generation (CNG) – генератор комфортного шума

Кроме этого, путем загрузки очень небольшого по размеру добавочного кода в C54CST можно дополнительно использовать следующие алгоритмы:

- ◆ Кассовый модем (V.29 и V.22 – быстрая связь)
- ◆ Факс G3
- ◆ Вокодеры:
 - G.723.1
 - G.729AB
 - Низкоскоростной вокодер, разработанный в SPIRIT Corp. (1200 бит/с)
- ◆ VoIP
- ◆ V.MoIP
- ◆ FoIP

Дополнения имеют небольшой размер, т.к. их код базируется на коде, находящемся в ПЗУ, и могут быть приобретены отдельно к среде CST. Дополнения хороши тем, что большая часть используемого кода находится внутри быстрого внутреннего ПЗУ и, следовательно, производительность не «деградирует» за счет циклов ожидания внешней памяти программ.

Помимо алгоритмов (среды CST) процессор C54CST содержит в себе цифровую часть бестрансформаторного интерфейса телефонной линии (Data Access Arrangement – DAA) и УАПП (асинхронный последовательный порт, UART) специально для недорогих приложений. DAA и UART могут использоваться, чтобы подключить сигнальный процессор к телефон-

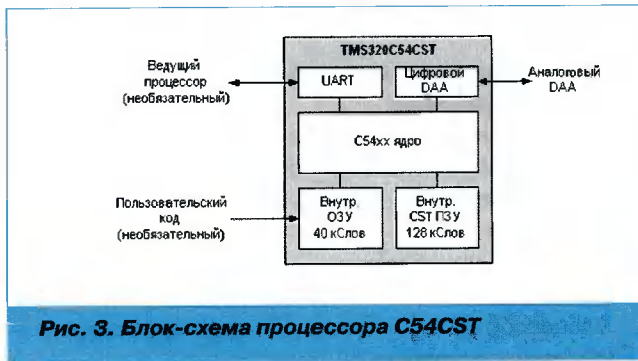


Рис. 3. Блок-схема процессора C54CST

ной линии и управляющему компьютеру (или микроконтроллеру) соответственно. Таким образом, для простого модемного приложения на основе микросхемы C54CST требуется небольшое количество внешних дискретных элементов – аналоговый DAA, чтобы подключиться к телефонной линии, и, возможно, преобразователь логических уровней в уровни RS-232 для подключения к управляющему процессору. Этого достаточно, чтобы подключить процессор C54CST к внешнему миру.

Итак, процессор C54CST может быть использован в качестве «начинки» для внешнего модема или процессора для встроенного (embedded) модема или может быть использован как DSP, содержащий в ПЗУ библиотеку вышеперечисленных телефонных алгоритмов и телефонную среду. Использование телефонной среды в принципе необязательно, т.к. с алгоритмами можно легко работать напрямую, поскольку они реализованы по стандарту xDAIS (об этом будет рассказано далее).

Микросхема C54CST может быть подключена к последовательному порту компьютера с работающей терминальной программой. Такая конфигурация, где процессор C54CST выступает в роли внешнего ведомого устройства, называется режимом чипсета (chipset mode). В данном режиме процессор C54CST может управляться как обычный модем, посредством AT-команд. В режиме чипсета можно пользоваться как модемными, так и голосовыми алгоритмами.

Использование процессора C54CST для реализации встроенного модема (или иного встроенного устройства) очень похоже на работу в режиме чипсета, хотя есть несколько положительных отличий. Большинство AT-команд соответствует определенным функциям программного интерфейса (API) CST, что делает возможным прямую работу с функциями среды CST, если загрузить пользовательский код прямо во внутреннюю память процессора C54CST. В данном случае управляющий процессор (или компьютер) становится необязательной частью системы, т.к. загруженный в память процессора код может вполне выполнять функции управляющего процессора. В такой конфигурации, которая называется гибким режимом (flex mode), управляющая программа пользователя, находящаяся внутри процессора C54CST, получает полный доступ к его программным ресурсам, функциям и данным. В гибком режиме работы процессора пользовательская программа может воспользоваться уже существующей телефонной средой, что позволит избежать интегра-

ции с каждым из алгоритмов и значительно уменьшить размер разрабатываемого кода. Посмотрим, почему так важно использовать эту среду.

Телефонная среда? Зачем?

Чтобы увидеть, почему важно использовать среду при разработке программного продукта, проанализируем пример небольшого приложения, в котором выполняются некоторые стандартные телефонные операции, и попробуем определить круг общих задач, решение которых необходимо для многих программных продуктов с функциями телефонии и задач, которые могут решаться специализированной средой.

Рассмотрим модемное приложение и определим (в качестве примера) требования для удаленной измерительной системы. Представьте, что нам необходимо, чтобы регулярно некое измерительное устройство делало замеры (температуры, давления, уровня жидкости или какой-либо другой физической величины) и посылало результаты измерений в управляющий центр, выполняющий анализ собранных данных и принимающий решение относительно дальнейших действий.



Рис. 4. Схема удаленной измерительной системы

Чтобы добиться требуемой функциональности, устройство должно выполнить последовательность следующих действий:

1. Выполнить измерения
2. «Поднять трубку»
3. Дождаться сигнала ответа станции
4. Набрать номер управляющего центра
5. Инициализировать модемное ПО
6. Ждать ответного сигнала модема станции
7. Ждать установление модемного соединения
8. Передать данные через модем
9. «Опустить трубку»
10. Прекратить работу модемного ПО
11. Выдержать задержку
12. Повторить цикл с шага 1

Разберемся, что представляет из себя каждый из шагов.

Шаги 1, 11 и 12 не имеют отношения к телефонии, и мы их рассматривать не будем.

Шаг 2 означает, что устройство должно заставить DAA (или то, что используется вместо него для обеспечения соединения с телефонной линией) «поднять трубку» и подключиться к телефонной линии. Естественно, что эта операция зависит от типа DAA и может

потребовать от простой записи какой-то последовательности байт в регистры DAA до такой же записи с последующим анализом других регистров, чтобы определить, что операция была выполнена успешно. Здесь может потребоваться некоторая задержка. Если операция была успешной, можно переходить к шагу 3.

Шаг 3 заключается в чтении и анализе отсчетов, получаемых DAA при дискретизации сигнала от телефонной линии. Если в результате анализа обнаруживается тон (или последовательность тонов), соответствующий сигналу ответа телефонной станции, мы можем начать использовать телефонную линию для набора номера и передачи данных и переходим к шагу 4. Если же при анализе обнаруживается отсутствие ответа станции или обнаруживается сигнал занятости (опять же представляемый каким-то тоном или последовательностью тонов), мы не можем продолжать использовать линию. Кстати, важно заметить, что такие сигналы, как ответ станции и сигнал занятости, различны в разных странах, что, несомненно, должно быть принято во внимание при разработке систем, которые могут и будут использоваться за пределами страны-разработчика.

Шаг 4 – набор цифр телефонного номера. Опять же эта процедура может зависеть от DAA, т.к. импульсный набор цифр требует специального управления током линии. Тоновый набор цифр не зависит от DAA, но DTMF-сигналы, соответствующие цифрам номера, должны быть сгенерированы программным обеспечением и в виде отсчетов должны быть выведены через DAA в линию.

Шаг 5 (инициализация модема) зависит целиком от реализации модемного ПО. В общем случае это может потребовать резервирование памяти для переменных состояния и буферов модема, а также инициализацию некоторых переменных модема. Помимо самой инициализации нужно еще создать поток отсчетов между модемом и DAA для того, чтобы модем мог обмениваться информацией с линией.

Шаг 6 (специфичный для модемов и обычно выполняемый самим модемным ПО) заключается в ожидании ответного тона (обычно 2100 Гц) от модема управляющего центра, когда тот снимет трубку в ответ на входящий звонок и подключится к телефонной линии. Если ответный тон найден, можно начинать процедуру установки модемного соединения и затем перейти к шагу 7. В случае отсутствия ответного тона в течение некоторого времени или при детектировании сигнала занятости нужно повторить всю процедуру набора номера с шага 2.

Шаг 7 – это ожидание установления модемного соединения. Поток отсчетов между модемным ПО и DAA должны продолжаться до тех пор, пока не будет закончена работа с модемом. Во время процесса установления модемного соединения модемы могут «договариваться» о скорости соединения и используемых протоколах. Если модемам удастся «договориться», они устанавливают соединение и становится возможным обмениваться информацией через модемы. В этом случае мы переходим к следующему шагу.

Шаг 8 – это непосредственно обмен данными между модемами. Обмен зависит от реализации модемного ПО, в частности от размеров порций данных, которые модем может принять или вернуть за один вызов соответствующей модемной функции. Все вопросы о буферировании данных должны решаться той частью управляющего кода, которая непосредственно работает с модемом, и они должны быть согласованы с программным интерфейсом модема, иначе могут возникнуть потери принимаемых данных или данные будут посылаться неэффективно. По окончании обмена данными можно переходить к шагу 9.

Шаг 9, аналогично шагам 2 и 4, зависит от DAA. Здесь мы должны заставить модем и DAA «повесить трубку», что тоже может потребовать некоторого времени, чтобы корректно оборвать соединение.

Шаг 10 заключается в прекращении работы модема, освобождении памяти, ранее зарезервированной для него (на шаге 5), и прекращении потока отсчетов между модемом и DAA.

Очень важно после каждой неудачной попытки соединения восстанавливать все в нормальное положение так, чтобы можно было через некоторое время повторить рассмотренный цикл с самого начала. Так же, как и в случае успешного сеанса обмена данными между модемами, когда мы освобождаем память и прекращаем потоки отсчетов на шаге 10, мы должны выполнить то же самое в случае, если модемам не удалось установить соединение, или соединение было неожиданно прервано по техническим причинам на стороне станции, или качество линии было низким. То же самое касается и DAA – в случае неудачной попытки нужно отсоединиться от линии, «класть трубку».

Что же следует из выше описанного? Мы видим, что требуется выполнить множество задач, таких, как «поднятие» и «опускание трубки», детектирование сигнала ответа и сигнала занятости, генерация DTMF-тонов, инициализация модема, работа с модемом, прекращение работы модема, обеспечение потоков отсчетов между DAA и модемом и перечисленными детекторами и генераторами. Все эти операции выполняются в определенном порядке, и нам также нужно восстанавливать состояния ресурсов, если что-то нарушает обычный порядок выполнения программы. Вероятно, что эти и другие аналогичные операции будут использоваться в нескольких программных продуктах с телефонными функциями. «Поднятие» и «опускание трубки», детектирование входящих звонков, генерация и детектирование различных телефонных сигналов, набор номера, маршрутизация отсчетов, логика для надежной работы телефонных операций и т.д. – все это будет «мигрировать» из проекта в проект, т.к. это действительно наиболее общие и частые задачи для выполнения, и было бы хорошо многократно использовать код, который выполняет данную работу.

С тем, чтобы создать приложение, подобное рассмотренному, можно взять все подпрограммы, выполняющие описанные задачи, и собрать вызовы всех этих подпрограмм внутри одного большого цикла. В прин-

ципе это могло бы сработать и, возможно, для действительно малых приложений типа рассмотренного или даже еще меньших это было бы наиболее эффективно. Но вероятнее всего, это не «сработает», и потребуется немалое количество сил и времени, чтобы добиться правильной работы программы. Почему так? Потому что инженеру необходимо знать интерфейсы всех программных компонентов, требования и ограничения компонентов, а иногда даже и знания принципов цифровой обработки сигналов, и соответствующие навыки могут оказаться полезными, хотя не каждый инженер может обладать ими. А знание – это не единственно необходимый элемент. Нужно будет еще создать потоки данных между компонентами, преобразовывать данные из одного формата в другой, если того требуют интерфейсы компонентов. Требуется также понять поведение компонентов, организовать управление ими, проверять различные переменные состояния, делать решения во многих местах и т.д.

Какие еще задачи и проблемы могут ожидать разработчика?

Представьте, что алгоритмы и соответствующее ПО пришли от разных поставщиков. Что это может значить для инженера? Самая первая вещь, которую бы обнаружили, – очень разные интерфейсы у этих алгоритмов. Даже детекторы и генераторы различных тоновых сигналов могут иметь разные интерфейсы и обрабатывать отсчеты различными порциями. Хотя это не самая большая проблема, все же потребуются дополнительные код для интеграции подобных алгоритмов в единой системе.

Как инженер мог бы проверить работоспособность алгоритмов? Как мог бы он убедиться в том, что алгоритмы правильным образом интегрированы в его системе? Читать горы документации и писать тестовую программу для каждого алгоритма? Конечно, нельзя недооценивать документацию и пренебрежительно относиться к ней – ее все же следует изучить, но беда в том, что это изучение не всегда дает ответы на все имеющиеся вопросы. И конечно же, тестирование алгоритмов может потребовать рабочую подсистему ввода-вывода и, возможно, даже какое-то дополнительное оборудование, которое тоже уже работает, но этого часто не хватает. Часто бывает не лишним вместе с алгоритмом получить от поставщика еще и тестовую программу и/или какое-нибудь тестовое устройство, где бы эти алгоритмы действительно работали.

Вообразите теперь, что алгоритмы (все или только часть) не поддерживают многоканальности, тогда как ваш DSP все еще имеет неиспользованные ресурсы и мог бы обрабатывать больше данных и это то, что вы хотите? Что делать? Искать и лицензировать другие, лучшие, алгоритмы или попытаться создать их самостоятельно? Или, может быть, просто запаять на плату еще один дополнительный процессор рядом с первым? Хороший вопрос.

Представьте себе, что какие-то алгоритмы резервируют память для собственных нужд статически или в процессе работы, когда им заблагорассудится, динамически. Это реальная проблема, которая будет выра-

жаться в том, что алгоритм не будет работать в многоканальных системах (особенно в системах с вытесняющей многозадачностью) или будет приводить к фрагментации драгоценной памяти.

А что, если вы хотите продавать ваш продукт и за пределами вашей страны? Будут ли ваши алгоритмы и аппаратное обеспечение работать в других условиях? Можете ли вы просто перенастроить детекторы и генераторы телефонных сигналов на другие частоты и длительности, можете ли вы подстроить уровни?

Множество вопросов и проблем могут возникнуть в процессе разработки, интеграции и тестирования. Все из-за того, что компоненты плохо предназначены для работы друг с другом в рамках одной системы или они недостаточно гибки. Могут быть и иные причины проблем. Чтобы минимизировать возможные проблемы и ускорить разработку приложений с телефонными функциями, и был создан проект CST, в который входит и телефонная среда (CST framework) для организации основных телефонных операций, и соответствующие алгоритмы, которые не страдают от вышеперечисленных проблем.

Кратко опишем преимущества использования среды CST.

CST-алгоритмы

- ◆ Реализованы по стандарту xDAIS:
 - Поддерживают многоканальность
 - Не имеют статически зарезервированной памяти для изменяемых данных и сами не вызывают функции резервирования и освобождения памяти
 - Экземпляры алгоритмов могут быть созданы и уничтожены динамически, когда требуется
 - Поддерживают системы с вытесняющей многозадачностью, например DSP/BIOS (в таких системах конкурентная обработка данных может помочь снизить максимальное потребление времени процессора, MIPS)
 - ◆ Свойства могут изменяться во время создания экземпляров (например, алгоритм UMTD может быть настроен на детектирование различных сигналов, в т.ч. и телефонных сигналов для разных стран)
 - ◆ Доступ к алгоритмам унифицирован
 - ◆ Могут быть легко протестированы, когда CST работает в режиме чипсета или в гибком режиме (для гибкого режима существуют примеры программ, которые иллюстрируют использование тех или иных алгоритмов CST). Самое простое было бы попробовать и протестировать алгоритмы на существующем наборе средств разработки (CST Developer's Kit), в котором находится плата с процессором C54CST и всей необходимой периферией
 - ◆ Уже интегрированы в среду
 - ◆ Могут быть использованы без среды, если в этом есть необходимость
 - ◆ Доступны от одного поставщика в одном процессоре
 - ◆ Могут быть использованы в нескольких различных продуктах
- #### **Телефонная среда (CST framework)**
- ◆ Интегрирует в себя алгоритмы CST (предоставляет требуемые соединения и потоки данных; корректно

создает, вызывает и уничтожает экземпляры алгоритмов) и делает единообразным управление через несколько управляющих уровней (Уровень Service, Уровень Commander, Уровень Action и Уровень AT-команд)

- ◆ Алгоритмы могут быть добавлены (например, новые декодеры)
- ◆ Многоканальная
- ◆ Может работать с DSP/BIOS
- ◆ Предоставляет статистику (пиковое потребление MIPS, использование стека и динамической памяти)
- ◆ Имеет собственный менеджер памяти, поддерживающий добавление новой памяти
- ◆ Использует высокоуровневые драйверы для DAA и UART
- ◆ Существующие драйверы могут быть заменены целиком или частично
- ◆ Могут быть добавлены новые драйверы
- ◆ Может быть легко протестирована в режиме чипсета и в гибком режиме на наборе средств разработки (CST Developer's Kit), имеющем плату с процессором C54CST
- ◆ Может быть использована в нескольких различных продуктах

Драйверы для периферии

- ◆ Доступны через программный интерфейс драйверов LIO, обеспечивая простую интеграцию новых драйверов
 - ◆ Использует стандартную библиотеку работы с периферией процессора, CSL (Chip Support Library)
 - ◆ Драйвер UART делает внутреннее буферирование данных и поддерживает аппаратный контроль обмена данными, используя CTS- и RTS-сигналы RS-232
 - ◆ Драйвер DAA предоставляет управление подключением к телефонной линии, детектирование звонка, настройку регистров на специфичные для страны значения
- Рассмотрим чуть более подробно телефонную среду.

Среда RF3

Телефонная среда, к рассмотрению которой мы постепенно переходим, включает в себя еще одну специализированную среду – RF3 (Reference Framework level 3), разработанную в компании Texas Instruments. Среда RF3 является важным компонентом среды CST, и его описание нельзя опустить.

Использование xDAIS и ALGRF

Все алгоритмы в среде CST реализованы по стандарту xDAIS (eXpressDSPTM Algorithm Standard, известный, как xDAIS, является частью среды RF3). Это означает, что экземпляры алгоритмов могут динамически создаваться и можно создавать несколько экземпляров каждого алгоритма. Иными словами, несколько потоков данных могут независимо обрабатываться одним и тем же алгоритмом. Модуль ALGRF является ответственным за создание и уничтожение экземпляров алгоритмов, резервирование памяти для которых унифицировано. Каждый алгоритм экспортирует специальную функцию, которая возвращает таблицу с информацией,

о требованиях алгоритма к памяти. Таблица описывает необходимые для экземпляра блоки памяти, и каждый элемент таблицы содержит размер блока, границу выравнивания, тип блока и область памяти, в которой этот блок должен быть выделен. При создании экземпляра алгоритма производится вызов такой функции, после чего и происходит резервирование памяти согласно информации, находящейся в таблице.

Жизненный цикл xDAIS-объекта упрощенно представлен на рис. 5.

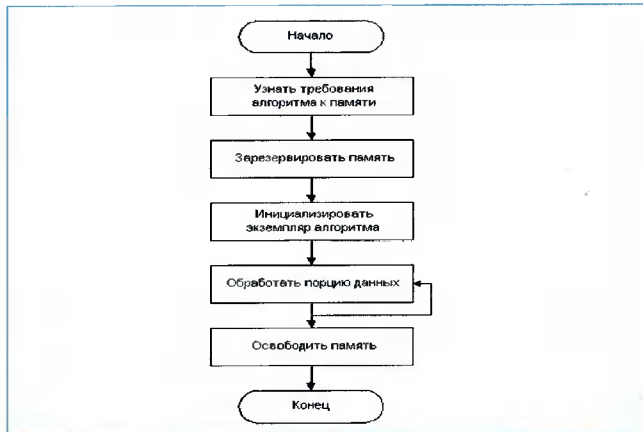


Рис. 5. Упрощенный цикл жизни экземпляра xDAIS-алгоритма

Существует также специальный тип памяти – рабочая память (scratch memory). Она предназначена для совместного использования несколькими экземплярами и хранения в них значений промежуточных вычислений во время работы алгоритмов. Как только один экземпляр заканчивает обработку порции данных и больше не нуждается в значениях своих промежуточных вычислений, другой экземпляр может использовать эту же самую память для своих аналогичных нужд. Такой подход позволяет сократить требования к памяти, хотя нужно отметить, что рабочей памятью нужно пользоваться осторожно, если в системе есть вытесняющая многозадачность. У алгоритмов может быть еще и специальная функция, предназначенная для перемещения зарезервированных блоков памяти, чтобы уменьшить фрагментацию памяти в системе.

В среде CST память для экземпляров алгоритмов резервируется динамически, экземпляры создаются только тогда, когда есть необходимость обрабатывать те или иные данные. Такие важные особенности, как возможность динамического создания экземпляров алгоритмов и возможность совместного использования памяти, дают максимум гибкости разработчику при создании собственных приложений в среде CST, будь то многоканальный детектор DTMF-сигналов или система с несколькими модемами.

Использование LIO и CSL

Периферийные драйверы в среде CST экспортируют свои функции через интерфейс LIO (Low-level I/O-интерфейс, известный, как LIO, является частью среды RF3). Это позволяет легко интегрировать в среду CST

драйверы для новых или других аппаратных устройств ввода-вывода. Интерфейс LIO позволяет переопределять методы драйверов и изменять функциональность последних путем изменения указателей на функции внутри LIO-структуры драйвера. Вся дополнительная функциональность, которая выходит за рамки функций ввода/вывода LIO или определяется конкретной реализацией, делается в методе ctrl()-драйвера. Этот метод принимает команды, специфичные для реализации, которые расширяют блок – ориентированный ввод-вывод метода submit(). Таким образом, LIO предоставляет общий интерфейс драйвера и возможность гибко переопределять драйверы или отдельные их функции, возможно, даже во время выполнения. Разработчик, создающий свое собственное приложение в среде CST или желающий работать с оригинальным CST, но с другой периферией, сможет легко интегрировать драйвер для кодека или другого устройства в среду CST, используя LIO.

Помимо интерфейса LIO, периферийные драйверы среды CST используют еще одну часть среды RF3 – CSL (Chip Support Library). CSL – это библиотека, содержащая подпрограммы для работы с устройствами, находящимися внутри процессора C54CST. В нашем случае это UART и DAA.

Во время тестирования и отладки, благодаря LIO и CSL, среда CST была перенесена на несколько различных плат с разными периферийными устройствами и разными процессорами серии C54xx.

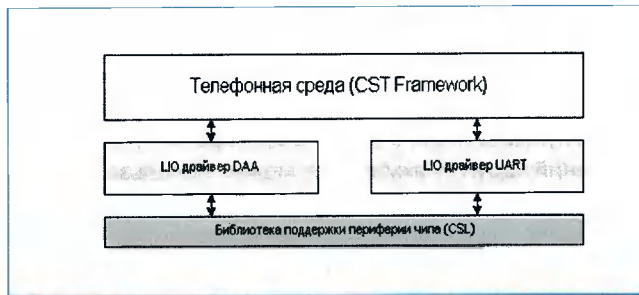


Рис. 6. Телефонная среда CST и подсистема драйверов

Использование DSP/BIOS

Кроме описанных компонентов, среда RF3 содержит еще и DSP/BIOS. DSP/BIOS – это небольшая система реального времени с вытесняющей многозадачностью. Среда CST была разработана таким образом, что она может работать и с DSP/BIOS. Преимущества использования DSP/BIOS очевидны. С DSP/BIOS можно реализовать конкурентную обработку данных и понизить пиковое значение потребления процессорного времени (MIPs). Например, можно запускать вокодеры, обрабатывающие данные большими порциями, или алгоритм V.42bis сжатия данных в модеме в низкоприоритетных нитях или задачах. Это позволит загрузить процессор более эффективно, «размазав» загрузку по времени. Мы уменьшили пиковую загрузку процессора алгоритмом V.42bis, которая возникает из-за поиска внутри словаря сжатия, почти до среднего значения загрузки всеми модемными алгоритмами.

Резюме

Таким образом, среда RF3 помогает разрабатывать эффективное ПО, в частности среду CST. Различные части RF3 помогают интегрировать драйверы в работающую систему и наделяют алгоритмы такими важными свойствами, как динамическое создание, эффективное совместное использование памяти и поддержка многоканальности. DSP/BIOS помогает улучшить производительность системы, распараллеливая процессы обработки данных.

Телефонная среда (CST framework)

Телефонная среда состоит из нескольких управляющих уровней.

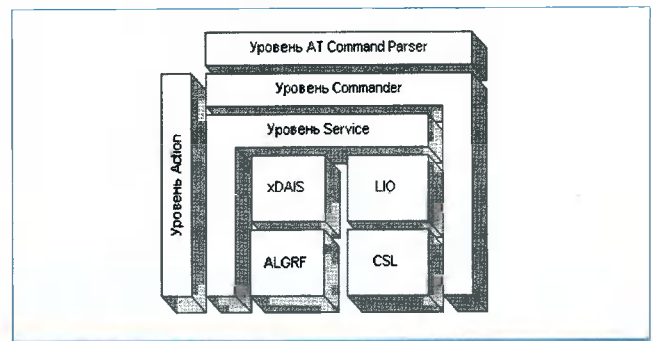


Рис. 7. Управляющие уровни в телефонной среде

Уровень Service

Сервисный уровень – это уровень, который выполняет всю непосредственную работу с алгоритмами (создание и уничтожение их экземпляров, вызов рабочих функций) и организует ввод-вывод. Данный уровень унифицирует доступ к алгоритмам (посредством специальных управляющих сообщений и событий) и создает потоки данных между драйверами устройств ввода-вывода и алгоритмами. Сервисный уровень – это уровень наиболее интенсивного использования RF3.

Уровень Commander

Двигаясь вверх от сервисного уровня, мы попадаем на командный уровень. На этом уровне среда CST может управляться последовательностями команд (скриптами) для выполнения стандартных телефонных операций, таких, как «поднятие» и «опускание» трубки, ожидание сигнала ответа станции, набор телефонного номера, ответ на входящие модемные и голосовые звонки, инициирование модемных и голосовых звонков, чтение/запись S-регистров (S-регистры – это регистры, значение которых отражается на функционировании модема и других частей CST) и т.д. Команды, из которых строятся высокоуровневые скрипты для выполнения телефонных операций, преобразуются в соответствующие сообщения для нижерасположенного сервисного уровня. Разработчик может создавать собственные команды и скрипты без необходимости модифицировать код CST. Командный уровень скрывает большинство аппаратно-зависимых деталей от разра-

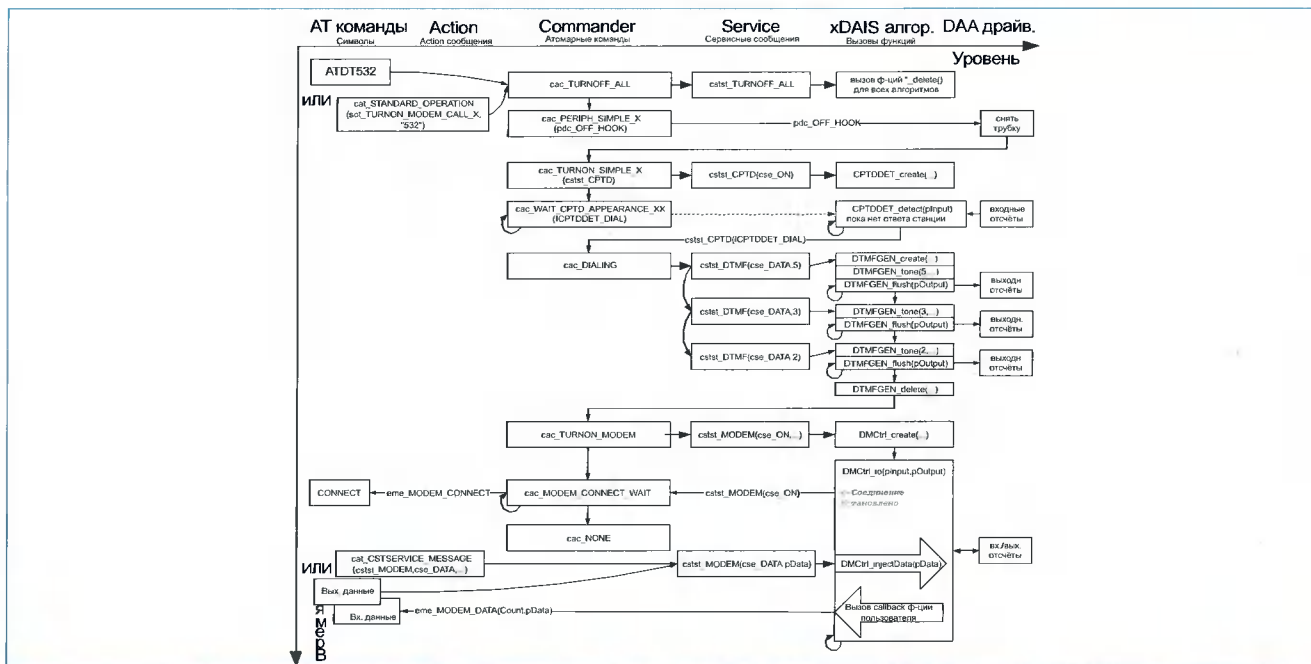


Рис. 8. Пример выполнения команд и их последовательностей на разных уровнях

ботчика и выступает в роли более абстрактного уровня взаимодействия с телефонными алгоритмами и соответствующими аппаратными устройствами.

Рис. 8 иллюстрирует, как простая задача набора номера и соединения с удаленным модемом для передачи данных преобразуется в более низкоуровневые задачи, выполняемые телефонной средой на различных управляющих уровнях.

AT-команда ATDT532 или соответствующая ей «акция» приводят к выполнению следующих операций:

- ◆ DAA снимает трубку
- ◆ Создается экземпляр алгоритма Call Progress Tone Detector для детектирования телефонных сигналов, в данном случае нас интересует сигнал ответа телефонной станции
- ◆ Сигнал ответа детектируется и ожидается. Как только он определяется, сервисный уровень посылает сообщение-событие командному уровню, чтобы начать выполнять следующую команду
- ◆ Создается экземпляр генератора DTMF-сигналов для набора номера в тональном режиме
- ◆ Последовательно генерируются DTMF-сигналы для цифр 5, 3 и 2
- ◆ Уничтожается экземпляр генератора DTMF-сигналов
- ◆ Создается экземпляр модемного интегратора (он создает экземпляры остальных модемных алгоритмов)
- ◆ Модем начинает работу
- ◆ Как только устанавливается модемное соединение, сервисный уровень посылает уведомительное сообщение командному уровню, чтобы можно было начинать посылать и принимать данные через модем
- ◆ Модем работает и забирает данные для отправки из сообщения сервисному уровню и возвращает принятые данные в callback-функцию

Все эти операции также включают в себя ввод-вывод отсчетов и логику для обеспечения надежной работы (которая не показана, чтобы не перегружать диаграм-

му). И все это делается самой телефонной средой! Приложение, разработанное для среды CST (вспомните про гибкий режим работы CST), не должно заниматься подобной «грязной» работой. Оно должно заботиться о своих собственных задачах (вспомните пример удаленной измерительной системы), среда же сделает все остальное.

Уровень AT Command Parser / Action

Сервисный и командный уровни являются самыми важными в среде, т.к. именно они выполняют телефонные операции. Но эти уровни не являются единственными. Существует еще один управляющий уровень, расположенный выше командного уровня. Этот уровень предназначен для организации взаимодействия пользователя CST и самой среды CST. Этим уровнем может быть или уровень действий, или уровень синтаксического анализатора AT-команд. Оба имеют одинаковое назначение, но выполнены по-разному и не могут использоваться одновременно.

Синтаксический анализатор AT-команд (AT Command Parser) взаимодействует с управляющим процессором (или компьютером) посредством AT-команд. Так же, как и обычный модем может управляться AT-командами и выдавать соответствующие ответные сообщения на команды, среда CST может делать все то же самое, если используется этот уровень. Фактически CST может работать как обычный модем с голосовыми и другими функциями, и все это доступно через AT-команды.

С другой стороны, уровень действий позволяет создавать автономные системы, не требующие вмешательства ведущего процессора для управления средой CST. В гибком режиме работы CST-управляющее приложение может быть загружено внутрь процессора C54CST и работать с CST через программный интерфейс CST, минуя генерацию AT-команд и обработку ответных сообщений. Приложение будет работать с уровнем действий, кото-

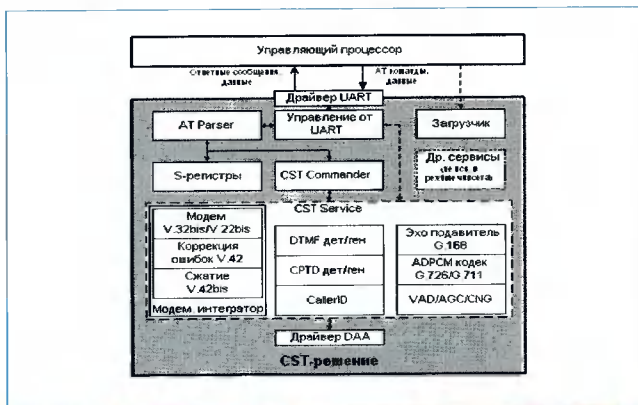


Рис. 9. Телефонная среда и управление ведущим процессором через AT-команды

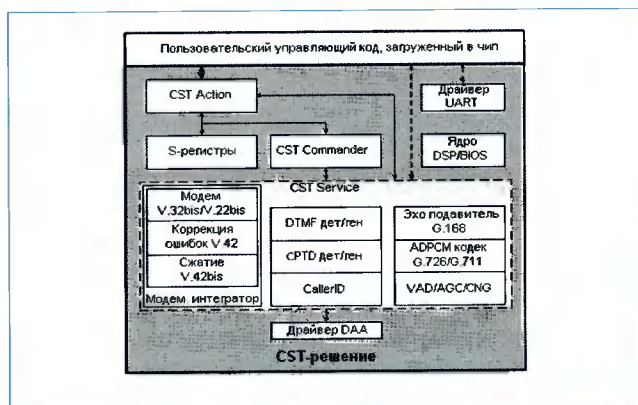


Рис. 10. Телефонная среда и управление внутренней программой через уровень действий

рый объединяет в своем интерфейсе доступ к нижележащим управляющим уровням телефонной среды.

Создание многих приложений для среды CST сводится к написанию программы, состоящей из нескольких основных функций, которые реализуют главную машину состояний программы. Первая функция будет посылать сообщения (запросы на выполнение той или иной телефонной операции) уровню действий в зависимости от текущего состояния программы. Эти сообщения могут нести в себе данные, например, которые должен послать модем. Вторая функция, callback-функция (или попросту функция, вызываемая средой), будет получать от среды уведомительные сообщения (например, о входящем звонке или сигнале «занято») или, например, данные, принятые модемом.

В CST SDK (набор средств разработки ПО для CST) существуют примеры таких программ, целью которых является показать, как можно создавать приложения для среды CST.

Резюме

Мы кратко ознакомились с представленной телефонной средой и выявили, что она помогает ускорить процесс разработки приложений с функциями телефонии, т.к. способна самостоятельно выполнять операции, типичные для большинства приложений телефонии, что уменьшает время, требуемое на интеграцию и разработку всей системы.

Посмотрим, как можно разрабатывать Voice Over IP-приложения с CST.

Разработка Voice Over IP-приложений

Следует сразу оговориться, что рассматривается только DSP-часть VoIP, ограниченная уровнем RTP/IFP-пакетов. Реализация более высокоуровневых протоколов обычно делается на управляющем процессоре или ПК и поэтому выходит за рамки этой статьи.

Типичное подключение CST в Voice Over IP (VoIP)-приложении изображено на рис. 11.

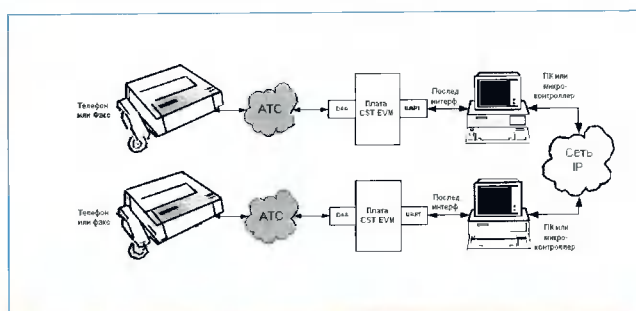


Рис. 11. Типичное подключение CST в VoIP-приложении

Для создания VoIP-приложений с CST мы можем воспользоваться следующими компонентами среды CST:

1. DAA и драйвер DAA для подключения к телефонной линии.
2. UART и драйвер UART для подключения к ПК или процессору, организующему доступ к IP-сети.
3. UMTD/UMTG для детектирования и генерирования DTMF-сигналов.
4. Эхоподавитель.
5. ADPCM-кодек или один из вокодеров (G.723.1 или G.729AB).
6. Телефонная среда для выполнения основных телефонных операций.
7. VoIP-дополнение, содержащее следующие компоненты:
 - а) управляющая логика (в т.ч. детектор тона 2100 и сигналов T.30 для определения звонков модемом или факсом);
 - б) тоновый режектор для вырезания DTMF-сигналов из речи и логика для их восстановления;
 - в) буфер для уменьшения влияния переменной задержки в сети (jitter буфер);
 - г) RTP (Real-Time Protocol)-мультиплексор/демультиплексор;
 - д) пакетирование.

Потоки данных и их обработка в VoIP-приложении, запускаемом на CST, изображены на рис. 12.

Хотелось бы особенно отметить преимущества, получаемые при использовании телефонной среды CST при создании этого приложения:

- ◆ Готовые драйверы периферии, находящейся на кристалле процессора.
- ◆ Использование высокоуровневого драйвера DAA, даже если внешний DAA не является совместимым с Si3016.

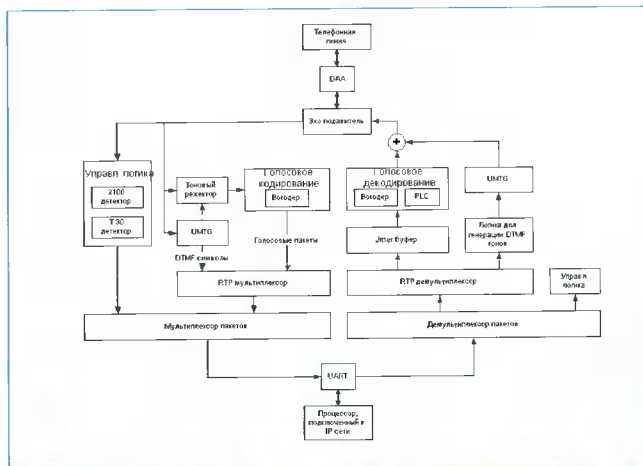


Рис. 12. C54CST в VoIP-приложении

- ♦ Готовые машины состояний для часто используемых операций с телефонной линией (таких, как набор номера).
- ♦ Легкая конфигурируемость и эффективное использование памяти процессора благодаря динамическому созданию всех алгоритмов, включая вокодеры.
- ♦ Встроенная поддержка многоканальности как на уровне алгоритмов, так и на уровне среды. Это свойство также позволяет использовать один и тот же алгоритм для разных целей даже внутри одного канала. Например, можно создать два экземпляра универсального детектора: один – для сигналов DTMF, другой – для сигналов MF-R1/R2.

Все это позволяет существенно сократить размер дополнительного кода, реализующего VoIP-функциональность и занимающего внутреннее ОЗУ, потому что большая часть кода находится уже в ПЗУ. Например, у одноканального варианта VoIP-приложения для процессора C54CST размер кода (память программы) составил 11 Кслов, а данных – 7 Кслов, что позволяет разместить еще и дополнительный код пользователя во внутреннем ОЗУ (которое всего 40 Кслов) и сделать все устройство без внешней памяти вообще.

Изображенная на этом рисунке функциональность может быть целиком реализована в программе, загружаемой в процессор C54CST в гибком режиме.

Сетевой ввод-вывод делается отдельным процессором (или компьютером), имеющим подключение к сети. В минимальном варианте, если не используются более высокоуровневые протоколы типа H.323, этот процессор не ведет никакой дополнительной обработки, и потоки данных в нем можно схематично изобразить, как показано на рис. 13.

Таким образом, используя процессор C54CST и внешний процессор (или компьютер) для сетевого ввода-вывода, можно создавать VoIP-приложения. Такое приложение было сделано на процессоре C54CST. При этом процессор был загружен менее чем на 50% даже при использовании самого ресурсоемкого вокодера (G.723.1), что говорит о возможности создания многоканального варианта этого приложения на одном процессоре. В многоканальных приложениях процессор C54CST может быть подключен к нескольким внешним

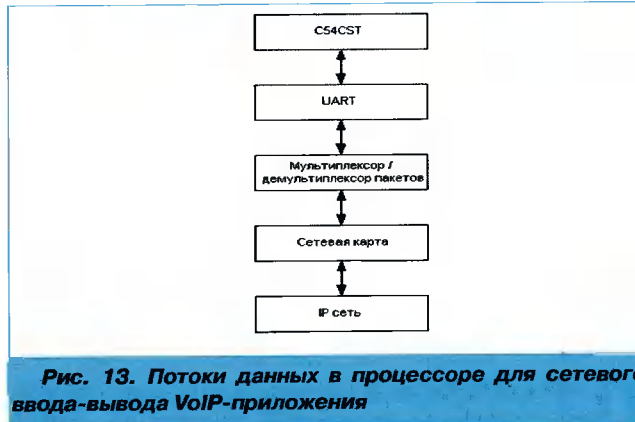


Рис. 13. Потоки данных в процессоре для сетевого ввода-вывода VoIP-приложения

DAA или SLIC или через фреймер к каналу T1/E1. При этом драйвер DAA может быть перегружен для использования других устройств ввода-вывода.

Заключение

Мы рассмотрели модемное приложение и VoIP-приложение. В обоих случаях, несмотря на различия между приложениями, телефонная среда и ее алгоритмы помогали выполнять необходимые телефонные операции. Следовательно, они действительно могут быть использованы для снабжения продуктов функциями телефонии.

Вкратце подытожим, что именно предлагает нам технология CST. Процессор C54CST является, с одной стороны, обычным DSP и имеет все преимущества программируемого устройства над чипсетом. С другой стороны, он предлагает готовый набор алгоритмов телефонии. Не нужно искать поставщиков требуемых алгоритмов, т.к. алгоритмы уже находятся во внутреннем ПЗУ. Покупая процессор C54CST, вы покупаете и алгоритмы. Процесс интеграции разнообразного ПО также отсутствует при разработке с использованием технологии CST, поскольку алгоритмы сделаны одним производителем, и телефонная среда, предлагаемая вместе с алгоритмами, правильным образом соединяет их, являясь также и примером работы с алгоритмами. Если потребуется работать с ними напрямую, минуя среду, код среды поможет разобраться в том, как это делается. Алгоритмы можно легко протестировать – для этого достаточно взять набор разработки (CST Developer's Kit) и запустить CST в режиме чипсета или попробовать запустить готовые приложения для гибкого режима. Поскольку телефонная среда выполняет многие операции самостоятельно и предоставляет обширные возможности для собственной настройки, разработка в среде CST практически сводится к написанию собственного управляющего ПО, интеграции с аппаратной частью, тестированию и отладке.

В заключение следует отметить, что программное решение CST, разработанное фирмой SPIRIT Corp. (www.spiritcorp.com), существует и для других сигнальных процессоров общего назначения серии C54xx и C55xx фирмы Texas Instruments. В этом случае весь программный код CST находится в ОЗУ (внутреннем или внешнем), и размер этого кода можно минимизировать в зависимости от используемой в среде CST-функциональности. Дополнительную информацию (включая развитие CST) можно получить на сайте SPIRIT Corp.

Новые разработки DSP компании Analog Devices

EDGE – аппаратно-программная платформа для построения систем связи третьего поколения

Проблема эффективного конструирования устройств телекоммуникаций 3G в настоящий момент является чрезвычайно актуальной. Analog Devices Inc. (ADI), как один из крупнейших поставщиков элементной базы для самых современных электронных систем, не могла обойти стороной этот вопрос. В своем развитии средств радиопередачи и приема, аналоговой и цифровой схмотехники, процессоров обработки сигналов компания достигла больших высот. Поэтому не является неожиданностью объявление компанией ADI о проведении разработок аппаратно-программной платформы, на базе которой возможно будет конструирование мобильных беспроводных устройств третьего поколения. Платформа будет именоваться EDGE (Enhanced Data rates for GSM Evolution) – платформа повышенных скоростей обработки и передачи данных стандарта GSM. Как видно из названия, разработчики ориентируются на стандарт GSM, достаточно широко распространенный в настоящее время. В частности, благодаря этому ожидается, что технология EDGE будет востребована и в Америке, и в Европе, и по всему миру. В США такие компании, как AT&T Wireless, и другие уже заявили о своем намерении использовать платформу EDGE.

Системы на базе платформы EDGE будут давать пользователю доступ к полной мультимедийной информации во всемирной сети. Пользуясь своими сотовыми телефонами, персональными секретарями (PDA) и другими средствами связи, можно будет кроме простых телефонных соединений осуществлять полноценную работу в Web с загрузкой и проигрыванием видео и аудио. Скорость обработки и передачи данных будет достигать 473,6 кбит/с, повышая втрое пропускную способность сегодняшних сетей беспроводной связи.

Что же будет представлять из себя новая платформа? Очевидно, для построения систем будущего нужны самые современные компоненты. Вычислительным ядром платформы EDGE будет служить цифровой сигнальный процессор Blackfin. Работая на тактовой частоте до 300 МГц и имея широкий набор специализированной периферии и возможности рационального взаимодействия с ней, а также включая эффективные средства по снижению потребляемой мощности, этот процессор был специально разработан для функционирования в носимых малогабаритных и самых современных телекоммуникационных средствах, обеспечивая обработку больших потоков мультимедиаданных и позволяя питаться от аккумуляторных батарей. Технология Blackfin, запатентованная компанией ADI, была отмечена рядом

независимых организаций как одно из самых важных открытий последнего времени в области аппаратного обеспечения систем цифровой обработки сигналов. Будучи интегрированным на платформу EDGE, процессор Blackfin выполняет задачи реализации алгоритмов модуляции, выравнивания канала связи, кодирования речи и высокоскоростных алгоритмов обработки данных.

Вторым базовым элементом платформы будет слушать приемопередатчик технологии Othello™, также запатентованной компанией ADI. Основной особенностью технологий Othello является прямое преобразование входного радиосигнала в область немодулированных частот, исключая одну или несколько стадий предобработки на промежуточных частотах. Это позволяет избавиться от дополнительных генераторов опорных (промежуточных) частот, фильтров и усилителей, а следовательно, понизить себестоимость, размеры и вес. Приемопередатчики Othello имеют ряд дополнительных достоинств, делающих эту технологию достаточно привлекательной для построения средств связи будущего.

Компания Analog Devices обеспечивает платформу EDGE и вспомогательными элементами аналоговой и цифровой схмотехники, а также системой управления расходом энергии, позволяющей существенно снизить потребляемую мощность.

Поддержка платформы EDGE программным обеспечением будет осуществлять компания TTPCom (Англия). Эта компания имеет ряд всемирно известных разработок и сотрудничает с такими ведущими производителями, как Hitachi, IBM, Intel, Toshiba, и другими. Компания TTPCom имеет большой опыт построения систем беспроводных телекоммуникаций, что в соединении с последними достижениями аппаратной базы компании Analog Devices должно дать действительно качественный результат.

Появление платформы ожидается в первой половине 2003 года.

ADSP-21992 – новый представитель семейства аналого-цифровых (mixed-signal) сигнальных процессоров

В феврале 2002 года компания Analog Devices анонсировала выпуск семейства ADSP-2199x – цифровых сигнальных процессоров (ЦСП) с аналоговыми компонентами. На одном кристалле были интегрированы традиционный ЦСП класса ADSP-219x, снабженный широким набором специализированной периферии, и элементы аналоговой техники, главным из которых является аналогово-цифровой преобразователь (АЦП). Семейство ориентировано, в первую очередь, на

встраиваемые ЦСП-применения, требующие одновременного эффективного решения задач управления и сигнальной обработки. Это, например, системы сбора данных, интеллектуальные датчики, робототехника, автоматизация производства, управление в оптических сетях и другие системы контроля и управления процессами, требующие высокой скорости и точности обработки больших потоков данных от нескольких источников в реальном времени. Появление семейства явилось следствием интеграции двух направлений исследований и разработок, проводимых компанией ADI, одно из которых – 16-разрядные ЦСП с фиксированной точкой семейства ADSP-21xx, а другое – элементы аналого-цифровой схемотехники и микроконтроллеры ADMCxxx.

Размещение АЦП на одном кристалле с вычислительным ядром позволяет повысить точность «оцифровки» входных сигналов и работы всей системы в целом. Передача данных с АЦП в память процессора осуществляется по специально выделенному каналу прямого доступа. Таким образом достигается высочайшая скорость передачи и становится возможной реализация сверхдискретизации и экстраполяции.

Первыми процессорами семейства ADSP-2199x являются модели ADSP-21990 и ADSP-21991. Их характеристики представлены в табл. 1.

Вычислительное ядро 16-разрядного сигнального процессора ADSP-219x, работающее в формате с фиксированной точкой, характеризуется временем командного цикла 6,25 нс и способно выполнять 160 миллионов команд в секунду (MIPS). Это мощное современное вычислительное ядро, отвечающее в полном объеме основным требованиям разработчиков систем ЦОС. Остановимся подробнее на компонентах периферии, интегрированных на кристалле процессора ADSP-2199x.

Блок АЦП, работая на тактовой частоте 20 МГц, обеспечивает типовое отношение сигнал/шум 70 дБ во всем температурном диапазоне от -40 до +105° С. При этом он способен выдавать двадцать миллионов «оцифрованных» отсчетов в секунду. Сигналы могут поступать от восьми различных источников. Аналоговые входы делятся на две группы по четыре входа и через мультиплексоры заводятся на схемы усиления, выборки и хранения, образуя каналы А и В. Блок АЦП обраба-

тывает одновременно два сигнала – один с канала А, другой с канала В. Цифровые данные с выхода АЦП представляются в 14-разрядном формате.

Трехфазный генератор ШИМ-сигналов, интегрированный на кристалле, предназначен для генерации управляющих сигналов требуемого вида, применяемых для управления приводами двигателей различных типов. Благодаря наличию ШИМ-генератора становится возможной выработка управляющих импульсов с минимальным участием вычислительного ядра процессора. Частота импульсов программируется 16-разрядным регистром. Ширина импульсов задается с разрешением 12,5 нс. Программируются также длительность «мертвого времени», симметричность или асимметричность формы импульсов и другие параметры сигналов. Кристалл ADSP-2199x снабжен дополнительным блоком генерации ШИМ-сигналов, включающим два канала, работающих независимо с различными параметрами ШИМ или в режиме фазового сдвига, когда несущая в обоих каналах одинакова, но сигналы отличаются по фазе.

Блок интерфейса с инкрементным шифратором, широко применяемым в системах управления двигателями для реализации обратной связи по положению, оперирует с входными сигналами частотой до 13,25 МГц. Схема интерфейса способна выполнять ряд вспомогательных функций, таких, как фильтрация входных сигналов от шума, выявление ошибок, функция контроля скорости и положения привода, индикация событий шифратора.

Процессор ADSP-2199x содержит три таймера общего назначения. Таймеры являются 32-разрядными и могут независимо использоваться в трех различных режимах: режиме генерации периодического импульсного сигнала; режиме измерения ширины входящих импульсов и режиме сигнализации внешнего события.

Сторожевой таймер процессора используется для защиты от непредвиденных ситуаций, возникающих в ходе выполнения программ. При нормальной работе таймер периодически заполняется, что предусматривается программистом, и никогда не достигает нуля. При нарушении работы значение счетчика постепенно уменьшается до нуля. Ноль сторожевого таймера обуславливает сброс (RESET) ядра DSP и периферии.

Таблица 1

	ADSP-21990	ADSP-21991	ADSP-21992
Частота	160 МГц		
Память программ	4К слов	32К слов	32К слов
Память данных	4К слов	8К слов	16К слов
Интерфейс с внешней памятью	до 1М слов		
АЦП	14 разрядов, 20 млн. отсчетов в сек.		
ШИМ, интерфейс кодера	имеется		
Посл. порты, SPI, таймеры	имеется		
Напряжение питания	2,5 В/3,3 В		
Цена	\$18,45	\$21,95	\$22,50
Корпус	мини-BGA-196; TQFP-176		
Дата выпуска	апрель 2003	апрель 2003	март 2003

16 выводов флагов, независимо программируемых на ввод или на вывод цифрового сигнала, широко используются для поддержки взаимодействия с внешними устройствами, а также в других целях.

Наличие порта SPI дает возможность процессору ADSP-2199x подключаться к линии последовательного интерфейса стандарта SPI и взаимодействовать с широким рядом внешних устройств, работающих с этим стандартом.

Синхронный последовательный порт расширяет возможности последовательного интерфейса и служит для объединения устройств в многопроцессорную систему.

Контроллер прерываний процессора способен в каждый момент времени откликнуться на одно из 17 прерываний, три из которых внутренние, два внешние, а 12 – прерывания от устройств периферии. Контроллер позволяет переходить к обработке прерываний с минимальными издержками. Это достигается за счет дублирования вычислительных регистров и регистров генераторов адреса. Исключается необходимость запоминать состояние в стеке перед обращением к подпрограмме обработки прерываний и восстанавливать состояние процессора при возвращении из нее. Требуется лишь переключиться с основного набора регистров на вспомогательный. Таким образом, достигается высокая скорость отклика на прерывание. Поскольку возможных прерываний периферии может быть до 32, программисту приходится определять, какие из них в данный момент времени включать в 12, одновременно обрабатываемых. Согласование прерываний периферии и прерываний ядра помогает осуществить специализированный контроллер прерываний периферии.

Периферия процессора поддерживается контроллером прямого доступа к памяти (DMA), позволяющим эффективно реализовывать перемещения данных и

команд между периферией и различными областями внутренней и внешней памяти без участия вычислительного ядра процессора.

К аналоговым элементам, размещенным на кристалле, помимо АЦП и его входных усилителей, относятся схема генерации опорного напряжения и схема генерации сигнала перезагрузки при включении питания (Power on Reset – POR).

Внутренний источник опорного напряжения обеспечивает генерацию точного и стабильного сигнала в 1 В, который используется АЦП. Этот сигнал может также быть выведен вовне через специальный выход микросхемы и использоваться внешними устройствами для реализации схем электрического смещения или сдвига уровня. При необходимости тот же вывод кристалла может использоваться для получения внешнего опорного сигнала и подачи его на АЦП вместо внутренне генерируемого.

Схема генерации сигнала перезагрузки при включении питания (POR) оказывается полезной, поскольку процессор после подачи питания находится в неопределенном состоянии. Схема POR при включении генерирует сигнал RESET, который может быть использован для полной перезагрузки ЦП и всей периферии, что переводит систему в известное состояние. Схема POR также генерирует сигнал перезагрузки при падении питающего напряжения ниже заданного уровня.

В августе 2002 года компания Analog Devices анонсировала появление новой модели семейства ЦП с аналоговыми компонентами – процессора ADSP-21992 (рис. 1). Эту модель отличает расширение внутренней кристалльной памяти RAM (табл. 1) и включение интерфейса CAN.

Сети протокола CAN (Controller Area Network) обладают рядом существенных достоинств, включая высокую надежность, низкую стоимость, гибкость, хорошую

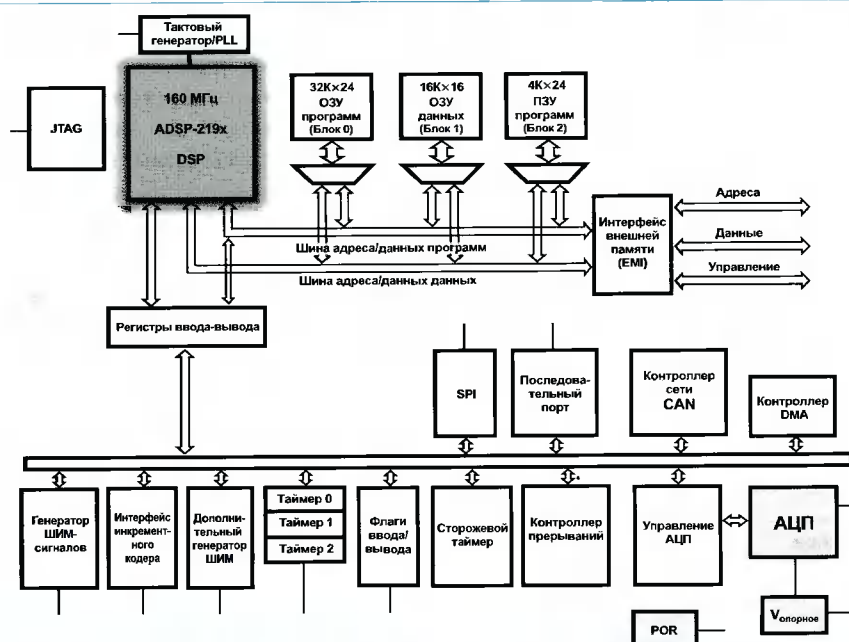


Рис. 1. Блок-схема аналого-цифрового процессора ADSP-21992

помехоустойчивость и все более широко применяются в транспортных системах, самых разнообразных машинах и механизмах, в системах управления роботами и в других сферах. Технология CAN-сетей становится одним из самых многообещающих решений для открытых распределенных систем автоматизации.

Сети, построенные в соответствии с протоколом CAN, включают в себя единственную линию передачи данных, обычно представляемую витой парой с подключенными к ней распределенными узлами. Узлы сети осуществляют непрерывный анализ состояния линии. После прохождения кадра синхронизации всем узлам одновременно разрешается передача сообщений. Таким образом поддерживается «мультимастерность». Любой узел сети может вести обмен данными с любым другим узлом. Осуществляя передачу кадра сообщения бит за битом, узел следит за состоянием шины. Если сигнал на шине совпадает с переданным им сигналом, он продолжает передачу. В противном случае он «понимает», что шина была захвачена другим узлом, передающим сообщение с большим приоритетом, и прекращает передачу, не разрушая передаваемого кадра. Таким образом гарантируется, что высокоприоритетное сообщение всегда передается первым и имеет известное максимальное время ожидания.

Назначение приоритета сообщения задается специально выделенным полем кадра сообщения – идентификатором. Возможно использование стандартного 11-разрядного идентификатора (2048 различных сообщений) и расширенного идентификатора – 29 разрядов (512 млн. сообщений). Другие выделенные поля протокола в пределах каждого сообщения обеспечивают высокоэффективный обмен между узлами и позволяют применять сложные механизмы обнаружения и контроля ошибок. Поле данных в каждом сообщении может достигать 8 байтов. Эта величина является достаточной почти для любого приложения. Относительно небольшая длина сообщений гарантирует малое время ожидания для сообщений с высоким приоритетом и повышает помехоустойчивость передачи.

В процессоре ADSP-21992 модуль CAN использует 16 блоков памяти RAM объемом восемь 16-разрядных слов каждый. Доступ к этим блокам осуществляется попеременно интерфейсом сети и вычислительным ядром ЦСП. Поскольку электрические характеристики цепей соединения строго индивидуальны для каждой сети CAN, интерфейс обычно разделяют на контроллер и приемопередатчик. Это позволяет одному устройству-контроллеру работать в различных сетях. CAN-модуль процессора ADSP-21992 является контроллером. При использовании в сети модуль подключается к сетевому приемопередатчику.

Указанные особенности процессора ADSP-21992 ориентируют его на широкое применение в автомобилестроении, промышленных системах управления двигателями, источниках бесперебойного питания и портативных устройствах измерений.

Процессор поддерживается полным набором программных и аппаратных средств проектирования и отладки CROSSCORE™. Программирование осуществляется с использованием пакета VisualDSP++ для 16-разрядных DSP Analog Devices. Для первичного ознакомления с характеристиками и особенностями процессора существует отладочный комплект (Starter Kit), включающий плату с процессором ADSP-21992, расширенными периферией и интерфейсами и комплект VisualDSP++, жестко привязанный к данной плате. Стоимость комплекта ADDS-21992-EZLITE составляет около \$650 (с НДС).

Серийный выпуск процессоров семейства ADSP-2199x должен начаться в марте 2003 года с модели ADSP-21992. Затем, в апреле 2003 года, ожидается появление ADSP-21990 и ADSP-21991. Стоимость процессоров при приобретении от 10 000 штук без учета НДС указана в табл. 1. Процессоры выпускаются в корпусах мини-BGA, 196 выводов, и TQFP, 176 выводов. Рабочий температурный диапазон: от -40 до +105° C (для модели ADSP-21992).

Новый высокоэффективный эмулятор на шину PCI

Набор инструментов разработки и отладки CROSSCORE™ компании Analog Devices в скором будущем будет пополнен новейшим PCI-эмулятором, обеспечивающим скорость загрузки данных на процессор до 2,2 Мбайт в секунду. Эмулятор рассчитан на процессоры ADI, поддерживающие интерфейс JTAG. Это семейства Blackfin™, SHARC®, TigerSHARC® и ADSP-219x. Новая технология, примененная при построении эмулятора, позволила повысить частоту тактирования интерфейса JTAG в 5 раз по сравнению с предыдущими поколениями эмулятора. Повышенное быстродействие устройства позволит более детально исследовать поведение ЦСП и обеспечить высокую надежность разрабатываемых систем ЦОС.

Другой особенностью нового отладочного средства является поддержка так называемого «Фоновый измерительный канал» (Background Telemetry Channel – BTC), осуществляющего практически незаметно для основного вычислительного процесса наблюдение за состоянием системы. Этот механизм использует принцип совместного доступа к ряду регистров ЦСП процессором и логикой эмуляции и характеризуется минимальным влиянием отладочных цепей на реальные рабочие характеристики системы. Канал BTC делает возможным сбор данных и наблюдение за состоянием процессора в реальном времени без необходимости прерывать работу системы, производить измерения и возобновлять работу. В настоящее время технология BTC поддерживается средой VisualDSP++ 3/0 для семейств DSP Blackfin и ADSP-219x.

Новый эмулятор доступен для приобретения с декабря 2002 года. Его стоимость составляет \$4000, не включая НДС и таможенные пошлины. При этом компания Analog Devices предлагает следующий вариант приобретения продукта: пользователи эмуляторов Summit-ICE PCI или Apex-ICE USB компании ADI могут

вернуть эти устройства производителю за кредит в \$2500 для дальнейшей полной оплаты и приобретения нового PCI-эмулятора. Такая программа будет действовать до 3 июня 2003 года.

В заключение отметим конструктивные особенности эмулятора. Длина кабеля, соединяющего персональный компьютер и приставку эмулятора увеличена до 6 футов (183 см), а длина кабеля от приставки до разъема подключения к отлаживаемой системе – до 1 фута (30,5 см.) Питающие напряжения: 1,8, 2,5, 3,3 и 5 В – определяются эмулятором автоматически, позволяя ему работать с процессорами различных семейств.

Поддержка DSP Analog Devices другими компаниями-производителями

Компания Hyperception Inc. анонсирует выпуск новой версии ставшего уже популярным программного средства проектирования и отладки систем ЦОС – интегрированной среды разработки систем реального времени – RIDE (Real-Time Integrated Development Environment). Среда будет поддерживать семейство сигнальных процессор Blackfin компании Analog Devices, начиная с модели ADSP-21535. Среда является компонентно-ориентированным программным средством проектирования. Разработчику предлагается широкий набор библиотечных компонентов, реализующих те или иные операции. Выбрав необходимые блоки, задав их параметры и указав связи между ними, разработчик создает на рабочем столе структуру, реализующую нужный ему алгоритм цифровой обработки сигналов. Среда RIDE сама автоматически генерирует

коды программы, соответствующей построенной структуре, на языке ассемблера DSP или языке Си. При наличии отладочного модуля Blackfin EZ-KIT Lite полученный код может быть реализован в реальном времени на «живом» процессоре. Среда RIDE предоставляет возможность контроля работы ЦСП, его отладки и профилирования. Компанией Hyperception Inc. предлагается несколько версий среды с поддержкой ADSP-21535 Blackfin EZ-KIT Lite: стандартная (\$3995), профессиональная (\$4995) и корпоративная (\$9995).

Компания Eonic начала выпуск вычислительных модулей 6U CompactPCI Atlas3-6U-HS, ориентированных на широкий ряд различных сфер применения и характеризующихся высокой вычислительной производительностью и одновременно эффективной реализацией каналов передачи данных. Модуль строится на основе двух блоков PMC, двух ПЛИС Altera и двух кластеров из трех сигнальных процессоров ADSP-21160 SHARC компании Analog Devices каждый. Эти компоненты объединяются в архитектуре обработки сигналов и коммуникаций – CSPA. Архитектура CSPA и совместное использование ПЛИС и сигнальных процессоров придает системе чрезвычайную гибкость и высокую вычислительную производительность, что позволяет разработчику конструировать оптимальную для каждого конкретного приложения схему передачи потоков данных и применять при этом пре- и постобработку сигналов. Модуль поддерживается рядом программных средств проектирования: VSPWorks, VisualDSP++ и Matlab/Simulink.

*Обзор подготовлен Витязевым С.В.
по материалам ИНТЕРНЕТ*

Уважаемые авторы!

Редакция научно-технического журнала «Цифровая обработка сигналов» просит вас соблюдать следующие требования к материалам, направляемым на публикацию.

Требования к текстовым материалам и сопроводительным документам:

- Текст – текстовый редактор Microsoft Word на базе версии WINDOWS'95 или выше.
- Таблицы и рисунки должны быть пронумерованы. На все рисунки, таблицы и библиографические данные указываются ссылки в тексте статьи.
- Объем статьи до 12 стр. (шрифт 12). Для заказных обзорных работ объем может быть увеличен до 20 стр.
- Название статьи на русском и английском языках.
- Рукопись статьи сопровождается:
 - краткой аннотацией на русском и английском языках;
 - номером УДК;
 - сведениями об авторах (Ф.И.О., организация, телефоны, электронная почта).

Требования к иллюстрациям:

- Векторные (схемы, графики) – желательно использование графических редакторов Adobe Illustrator или Corel DRAW.
- Растровые (фотографии, рисунки) – М 1:1, разрешение не менее 300 dpi, формат tiff.

Справки по телефонам: (095) 105-3464, 105-3468, 267-0662 или по электронной почте info@dspa.ru

Передовые платформы для отечественных систем цифровой обработки сигналов

В.А. Куликов, В.Е. Чернов

Одной из основ динамичного развития отечественного рынка систем цифровой обработки сигналов (ЦОС) и встроенных систем управления промышленной и бытовой техникой является доступность для российских разработчиков базовых аппаратных платформ и средств разработки и отладки устройств на основе современных ЦСП (DSP). Если проводить сравнение с CISC- или RISC-процессорами, то решение на базе ЦСП для обработки сигналов всегда лучше, дешевле и в большинстве случаев менее требовательно к ресурсу источника питания, что особенно важно для встроенных применений в мобильных системах.

Лучшим доказательством этого является включение ядра ЦСП корпорацией Intel в свой новейший процессор для мобильных приложений PXA800F.

Технологии ускорения обработки данных и их извлечения и занесения в память, периферия с большим числом каналов ввода/вывода и коммуникационных интерфейсов, специализированные ОСПВ обеспечивают работу ЦСП в режиме жесткого реального времени, безальтернативного для ответственных применений в промышленных встроенных системах контроля и обработки данных, конкурентоспособной бытовой технике.

ЗАО «СКАНТИ-Рус» специализируется в поставках самых современных ЦСП производства TEXAS INSTRUMENTS.

Став официальным представителем TEXAS INSTRUMENTS на рынке России и СНГ, ЗАО «СКАНТИ-Рус» обеспечивает прямой канал поставок самых современных ЦСП на отечественный рынок. О преимуществах цифровых сигнальных процессоров TEXAS INSTRUMENTS, основанных на гарвардской архитектуре, говорит тот факт, что сейчас они стоят почти в 70% всех цифровых телефонных аппаратов мобильной связи, сама же компания TEXAS INSTRUMENTS контролирует половину всего рынка ЦСП.

Поддерживая производственные программы своих заказчиков, ЗАО «СКАНТИ-Рус» предлагает разработчикам и производителям серийной продукции (OEM) все необходимое: ЦСП, микроконтроллеры, операционные усилители, схемы АЦП, ЦАП, источников питания и супервизоров, интерфейсные схемы и схемы стандартной логики, платы начальной отладки, оценочные модули, внутрисхемные эмуляторы, программную среду разработки Code Composer Studio, ОСПВ DSP BIOS – для разработки и производства законченных систем ЦОС, контроля и управле-

ния ЦОС для потребительского рынка и ответственных промышленных применений.

В спектре предложений ЗАО «СКАНТИ-Рус» несколько базовых платформ ЦСП TMS320.

Семейство C2000 рассчитано на применение в современных встроенных системах управления и контроля. Основная особенность процессоров данной платформы заключается в реализации управляющего микроконтроллера на основе ядра сигнального процессора. Процессоры данной платформы имеют большую встроенную флэш-память, многоканальные АЦП, генераторы ШИМ, многоразрядные таймеры, развитую систему коммуникационных портов, в том числе в стандарте CAN.

Краткие технические характеристики семейства C2000:

- TMS320Lx24xx: обработка данных с фиксированной запятой, разрядность – 16 бит, производительность до 40 MIPS;
- TMS320F28xx: обработка данных с фиксированной запятой, разрядность – 32 бит, производительность до 150 MIPS.

Архитектура ЦСП семейства C5000 оптимизирована по соотношению производительность/потребляемая мощность. Основная сфера применений – мобильные телекоммуникационные системы, мультимедиа, системы шифрования данных и т.д. Процессоры семейства C5000 обладают развитой периферией, включающей интерфейсы I2C, USB, набор универсальных последовательных портов с возможностью селекции каналов и буферизации, что позволяет настраивать их для работы с внешними потоками данных всех промышленных стандартов, в том числе с потоками T1/E1.

Краткие технические характеристики семейства C5000:

- TMS320VC54xx: обработка данных с фиксированной запятой, разрядность – 16 бит, производительность до 160 MIPS (532 – многоядерная структура);
- TMS320VC55xx: обработка данных с фиксированной запятой, разрядность – 16 бит, производительность 50–600 MIPS.

ЦСП семейства C6000 построены на базе оригинальной архитектуры VelocityTI, использующей преимущества процессорной технологии VLIW (Very Long Instruction Word или инструкция с длинным командным словом). Семейство C6000 обладает уникальными возможностями для обработки информационных пакетов, создания аппаратуры широкополосного доступа и обработки видеoinформации.

Краткие технические характеристики семейства C6000:

- TMS320C62xx: обработка данных с фиксированной запятой, разрядность – 16 бит, производительность 1200 – 2400 MIPS;
- TMS320C64xx: обработка данных с фиксированной запятой, разрядность – 16 бит, производительность 3200 – 4800 MIPS;
- TMS320C67xx: обработка данных с плавающей запятой, разрядность – 32 бита, производительность 600–1350 MFLOPS.

Процессоры имеют 8 параллельных вычислительных модулей и многоступенчатый конвейер, что позволяет достичь значительной производительности при сравнительно невысокой тактовой частоте (до 600 МГц).

Платформа OMAP (Open Multimedia Applications Platform, или открытая платформа для мультимедийных приложений) Texas Instruments является главным соперником Intel PCA (Intel Personal Internet Client Architecture, или архитектура персонального доступа в Интернет).

В ее основе комбинация на одном кристалле сигнального процессора с ультранизким потреблением энергии TMS320C55x и управляющего RISC-процессора с ядром ARM и схемы общей памяти, что позволяет достичь компромисса между требованием обеспечения высоких рабочих характеристик системы и ее низкого энергопотребления. Мост DSP/BIOS Bridge позволяет разработчику конечной системы оптимальным образом распределять задачу между сигнальным и RISC-процессором. При исполнении приложения на OMAP оптимизированные под ЦСП задачи исполняются асинхронно на TMS320C55x.

Предоставляемая Texas Instruments платформа разработчика позволяет вести разработку конечной системы так, как если бы на кристалле был один

RISC-процессор, избегая в ряде случаев проблем с более трудными (или для кого-то неизвестными) системами программирования ЦСП. Для OMAP обеспечена поддержка основных операционных систем рынка мобильных телекоммуникаций: Linux, Microsoft Windows CE, Palm OS, Symbian OS.

Включение в состав аппаратной части OMAP TMS320C55x позволяет в приложениях стандарта 3G успешно реализовывать алгоритмы распознавания речи с эхо- и шумоподавлением. Энергоэффективнее, чем на одном RISC-процессоре, исполняются на OMAP алгоритмы MP3 (MPEG Audio Decode, Layer 3) и AAC (Advanced Audio Coding).

Ядру архитектуры ARM отводится в OMAP в основной функции контроля и управления.

Будучи представленной на рынке уже несколько лет, OMAP стал стандартом де-факто, чему в немалой степени способствовали широкое распространение ядра ARM и программная совместимость его модификаций, а также тот факт, что около 350 компаний – официальных партнеров выпускают программные и аппаратные средства для разработки систем на базе TMS320.

На российском рынке заказчиками ЗАО «СКАНТИ-Рус» уже проявлен интерес к таким двухпроцессорным микросхемам на основе ядер ARM и DSP, как OMAP5910, TMS320VC5470 и TMS320VC5471, оснащенный Ethernet-контроллером.

Наряду с продукцией Texas Instruments для комплексной поддержки производственных программ своих заказчиков ЗАО «СКАНТИ-Рус» поставляет также компоненты компаний Motorola, National Semiconductor, Temic, Toshiba, Fujitsu, Intersil, Infineon, Xilinx.

Важным направлением работ ЗАО «СКАНТИ-Рус» является ведение университетской программы по развитию применений ЦСП, развитие лабораторной базы ВУЗов, поддержка конкурсов на лучшую разработку и статью по применению ЦСП.