

ЦИФРОВАЯ БРАБОТКА СИГНАЛОВ

1/2002

НАУЧНО - ТЕХНИЧЕСКИЙ ЖУРНАЛ

СЖАТИЕ И ПЕРЕДАЧА ИЗОБРАЖЕНИЙ
СИСТЕМА ОТОБРАЖЕНИЯ ГРАФИКИ
ЦОС В РАДИОЛОКАЦИИ
НОВЫЕ РАЗРАБОТКИ DSP

ISSN 1684-2634



9 771684 263005 >

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Научно-технический журнал
№1 (5)/2002

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:

П. А. Арутюнов, 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. Шахгильдян,
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 «ЛТИС»

Генеральный директор: С. А. Задворнов
Дизайн и верстка: Д. В. Желнов; А. А. Шумилин
Корректурa: С. В. Витязев

Изд. лиц. ИД № 05078 от 18.06.2001.

Подписано в печать 30.07.2002. Формат 60×90/8.
Гарнитура «PragmaticaLightC». Печать офсетная.

Бумага мелованная. Печ. л. 7,0+вкл. 0,25.

Тираж 1000 экз. Заказ № 2490. Отпечатано в ФГУП
«Производственно-издательский комбинат
ВИНИТИ», 140010, г. Люберцы Московской обл..
Октябрьский пр-т, 403. Тел. 554-21-86.

Издание зарегистрировано в Министерстве
Российской Федерации по делам печати,
телерадиовещания и средств массовых
коммуникаций.

Свидетельство о регистрации ПИ №77-1488
от 14.01.2000

© ООО «Лаборатория телекоммуникационных
измерительных систем», 2002

В НОМЕРЕ:

Ю.С. Радченко

**Алгоритмы сжатия изображений на
основе полиномиальных
преобразований (алгоритм GDCT) 2**

С.В. Сай

**Анализ четкости цветных статических
изображений в равноконтрастной
системе координат 6**

А. В. Дворкович

**Особенности построения программного
телевизионного кодера MPEG-2 10**

С. В. Мушкаев, С.В. Ландышев

**Применение процессора NM6403
(L1879VM1) для сжатия изображений 12**

А.В. Бумагин, Е.А. Климов Е.А., В.Б. Стешенко

**Системы отображения трехмерной
графики для авиационных индикаторов 19**

*Д. Ю. Бобров, А. П. Доброжанский, Г. В. Зайцев,
Ю. В. Маликов, И. Б. Цыбин*

**Цифровая обработка сигналов
в многофункциональных РЛС 28**

С. А. Дорохин

**Эффективность параллельных систем ЦОС,
построенных на процессорах ADSP-21160 40**

С. В. Витязев

Analog Devices: новые разработки DSP 45

С. В. Витязев

Texas Instruments: новые разработки DSP 52

Алгоритмы сжатия изображений на основе полиномиальных преобразований (алгоритм GDCT)

Ю.С. Радченко

Системы связи нового поколения ориентированы на передачу информационных потоков различного вида [1,2]. В связи с этим передача телевизионных и компьютерных изображений, видеоконференции и видеосотопная связь предполагают применение процедур сжатия изображений. В последнее время выполнен ряд работ, в которых предлагается применять разложения по базису классических ортогональных полиномов Эрмита, Якоби, Лежандра, Чебышева [3,4].

В данной работе приводятся результаты исследования одного из вариантов полиномиальных преобразований — разложение и синтез по базису полиномов Чебышева I и II рода. Вычисление спектральных коэффициентов производится с помощью квадратурной формулы Гаусса-Чебышева, обладающей наивысшей степенью алгебраической точности при заданном числе отсчетов.

1. Разложение сигналов по базису ортогональных полиномов

Пусть в подобласти $\{x,y\} \in \Omega$ наблюдается поле $s(x,y,\tau)$, представляющее собой фрагмент $u(x,y)|_{\Omega}(x,y)$ пространственного сигнала. Здесь $I_{\Omega}(x,y)$ — индикаторная функция подобласти, $\tau=(\tau_x, \tau_y)$ — параметры сдвига фрагмента в данном кадре. Базисные функции $\varphi_{mk}(x,y)$, используемые для дискретного представления сигнала $s(x,y,\tau)$, определяются аналитическими свойствами самого сигнала и геометрической формой подобласти Ω . Однако реализация процедуры сжатия, особенно для полиномиальных базисов, существенно упрощается, если имеет место факторизация функций $\varphi_{mk}(x,y) = \varphi_m(x) \cdot \varphi_k(y)$. Здесь $\varphi_m(x)$, $\varphi_k(y)$ — одномерные функции, определяемые ортогональными полиномами. Тогда для полезного сигнала $s(x,y,\tau)$ имеет место пара преобразований

$$s(x,y,\tau) = \sum_m \sum_k C_{mk}(\tau) \varphi_m(x) \varphi_k(y) \quad (1)$$

$$C_{mk}(\tau) = \iint s(x,y,\tau) \varphi_m(x) \varphi_k(y) dx dy.$$

Если обозначить a_x, a_y — характерные размеры подобласти Ω , $z_1 = x/a_x$, $z_2 = y/a_y$, то (1) можно переписать с использованием ортогональных полиномов в виде ;

$$s(x,y) = \sum_{m,k} C_{mk} p_m(x/a_x) p_k(y/a_y) \quad (2)$$

Предложен метод сжатия изображения, основанный на разложении по полиномам Чебышева и применении квадратурных формул Гаусса - Чебышева (алгоритм GDCT). Показано, что алгоритм GDCT является обобщением дискретного косинусного преобразования, используемого в системах JPEG и MPEG. Алгоритм имеет три ступени сжатия: прореживание исходного изображения, усечение спектра, сжатие без потерь спектральной матрицы. На примерах обработки искусственных текстур и натуральных изображений продемонстрированы возможности двух первых ступеней алгоритма GDCT.

$$C_{mk} = (d_m d_k)^{-1} \iint_{\Omega} s(a_x z_1, a_y z_2) p_m(z_1) p_k(z_2) dz_1 dz_2 = \\ = (d_m d_k)^{-1} \int p_m(z_1) dz_1 \int s(a_x z_1, a_y z_2) p_k(z_2) dz_2$$

Здесь d_m — норма ортогонального с весом $p(z)$ полинома $p_m(z)$. Поскольку для базисных функций $\varphi_{mk}(x,y) = \varphi_m(x) \cdot \varphi_k(y)$ вычисление спектральных коэффициентов производится последовательным интегрированием по координатам (x,y) , то для упрощения анализа рассмотрим сначала одномерные преобразования, а затем обобщим их на двумерный случай.

2. Алгоритм преобразований Чебышева

Пусть имеется процесс $s(z)$, квадратично интегрируемый с весом $p(z)$. Тогда можно записать квадратурную формулу гауссовского типа наивысшей алгебраической степени точности порядка $2N-1$ как

$$\int s(z) p(z) dz = \sum_{n=1}^N \lambda_n s(z_n)$$

Здесь z_n — нули полиномов $p_n(z)$, ортогональных с весом $p(z)$; λ_n — числа Кристоффеля. Узлы и веса $\{z_n\}$, $\{\lambda_n\}$ однозначно определяются видом ортогонального с весом $p(z)$ полинома $p_m(z)$.

Для полиномов Чебышева формула Меллера (Гаусса-Чебышева) имеет вид

$$\frac{1}{d_{m-1}} \int \frac{s(z) T_m(z)}{\sqrt{1-z^2}} dz = \frac{\pi}{d_m N} \sum_{n=0}^{N-1} s(z_n) T_m(z_n) \quad (3)$$

Здесь $z_n = \cos(\pi(2n+1)/2N)$ — нули полинома Чебышева I рода $T_N(z)$; $\lambda_n = \pi/N$ — то есть все весовые коэффициенты одинаковы, норма полинома Чебышева $d_m = \pi/2$, если $m \neq 0$, и $d_m = \pi$, если $m = 0$.

Учитывая, что $T_m(z_n) = \cos(m \cdot \arccos(z_n)) = \cos(\pi m(n+0.5)/N)$, приходим к выражению для прямого и обратного преобразований

$$C_m = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} s(z_n) \cos(\pi m \frac{n+0.5}{N}), \quad C_0 = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} s(z_n) \quad (4) \\ S_M(z) = g_m \sqrt{\frac{2}{N}} \sum_{m=0}^M C_m T_m(z) = g_m \sqrt{\frac{2}{N}} \sum_{m=0}^M C_m \cos(m \cdot \arccos(z))$$

Здесь

$$g_m = \begin{cases} \sqrt{0.5}, & m=0 \\ 1, & m>0 \end{cases};$$

$S_M(z)$ — сигнал, восстановленный по M спектральным коэффициентам.

В формуле (4) использована симметричная нормировка базисных функций, удобная для практической реализации преобразований в матричном виде. Выражение (4) весьма похоже на так называемое четное дискретное косинусное преобразование (DCT) [2]. Но имеет три важных отличия:

1) точки отсчета $z_n = \cos(\pi(n+0.5)/N)$ сигнала $s(z)$ берутся неравномерно;

2) синтез сигнала $S_M(z)$ выполняется в произвольной точке $z \in [-1, 1]$, а не в дискретном наборе точек отсчета, как в DCT;

3) точность формулы (4) при преобразовании достаточно гладких функций существенно выше, чем у DCT. Поэтому число отсчетов можно взять значительно меньше.

Еще одна особенность полиномиальных преобразований заключается в том, что число отсчетов N и число спектральных коэффициентов M , которые определяются преобразованием (4), вообще говоря, не связаны между собой. То есть матрица прямого и обратного преобразований в общем виде является прямоугольной.

Исследования показали, что целесообразно выбирать $M \leq N$, поскольку для индексов $m > N$ коэффициенты C_m вычисляются с достаточно большой относительной погрешностью.

Если при восстановлении взять неравномерную сетку отсчетов по закону $z_j = \cos(\pi(j+0.5)/L)$, $j=0, \dots, L-1$, то обратное преобразование принимает вид

$$S_M(z_j) = g_m \sqrt{\frac{2}{N}} \sum_{m=0}^M C_m \cos(\pi m \frac{j+0.5}{L}). \quad (5)$$

Преобразование (5) выглядит аналогично DCT, однако спектральные коэффициенты отличаются от спектральных коэффициентов DCT. В случае, если интересует равномерная сетка отсчетов $z_j = 2j/(L-1) - 1 + \delta$, где δ — некоторый сдвиг, $j=0, \dots, L-1$, то

$$S_M(z_j) = g_m \sqrt{\frac{2}{N}} \sum_{m=0}^M C_m \cos(m \cdot \arccos(2j/(L-1) - 1 + \delta)) \quad (6)$$

Алгоритм преобразования (4), (6) назовем обобщенным дискретным косинусным преобразованием (GDCT). В формулах (5), (6) L — число точек в блоке восстановленного изображения может быть произвольным. В таком случае возникает эффект масштабирования восстановленного изображения по сравнению с размерами исходного блока с изображением.

Если использовать разложение по полиномам Чебышева II рода, получим

$$C_m = \frac{2}{\pi} \int_{-1}^1 \sqrt{1-z^2} s(z) U_m(z) dz = \frac{2}{N+1} \sum_{k=0}^{N-1} s(z_k) \sin(\pi \frac{(k+1)(m+1)}{N+1}) \sin(\pi \frac{k+1}{N+1}); \quad (7)$$

$$S_M(z) = \sum_{m=0}^M C_m U_m(z).$$

Здесь учтено, что $z_k = \cos(\pi(k+1)/(N+1))$ и

$$U_m(z_k) = \frac{\sin((M+1) \arccos(z_k))}{\sqrt{1-z_k^2}} = \frac{\sin(\pi(k+1)(m+1)/(N+1))}{\sin(\pi(k+1)/(N+1))} \quad (8)$$

Алгоритм преобразований (7) сходен с дискретным синусным преобразованием.

3. Двумерный алгоритм преобразования GDCT

Перейдем теперь к двумерному варианту преобразований сигнала $s(x, y)$. Пусть в пределах блока из $N_1 \times N_1$ точек, берутся $N \times N$ отсчетов сигнала по закону

$$x_n = \text{ROUND}(0.5 \cdot (N_1 - 1) \cdot (1 + \cos(\pi(n+0.5)/N))) \quad (9)$$

$$y_k = \text{ROUND}(0.5 \cdot (N_1 - 1) \cdot (1 + \cos(\pi(k+0.5)/N)))$$

При равномерной дискретизации исходного изображения неравномерность отсчетов для вычисления Чебышевских преобразований достигается путем пропуска части пикселей.

Отбор отсчетов производится согласно (9). Параметры преобразования N_1 и N должны быть подобраны таким образом, чтобы операция округления $\text{ROUND}(\cdot)$ вносила минимальную погрешность в отклонение точек отсчетов (x_n, y_n) от положения узлов квадратурной формулы. Другой, более трудоемкий в вычислительном отношении, но более точный способ получения неравномерности отсчетов состоит в линейной интерполяции значений поля по пикселям, ближайшим к узлам квадратурной формулы

$$x_n = 0.5 \cdot (N_1 - 1) \cdot (1 + \cos(\pi(n+0.5)/N));$$

$$y_k = 0.5 \cdot (N_1 - 1) \cdot (1 + \cos(\pi(k+0.5)/N)).$$

Отсчеты сигнала образуют матрицу $\mathbf{S} = \|s_{nk}\| = \|s(x_n, y_k)\|$. Эта матрица преобразуется в матрицу спектральных коэффициентов \mathbf{C} размером $M \times M$. При обратном преобразовании может использоваться прямоугольная матрица размером $L \times M$. То есть восстановленный блок имеет размеры $L \times L$. Прямое и обратное преобразование Чебышева (GDCT) в матричном виде определяют операциями

$$\mathbf{C} = \|\mathbf{C}_m\| = \bar{\Phi} \mathbf{S} \Phi^T; \quad \mathbf{S} = \Psi^T \mathbf{C} \Psi, \quad (10)$$

Матрицы прямого и обратного преобразования GDCT имеют вид

$$\Phi = \left\| \Phi_m(n) \right\|_{NM} = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{0.5} \\ \cos\left(\pi m \frac{(n+0.5)}{N}\right) \end{bmatrix};$$

$$\Psi = \left\| \cos\left(m \cdot \arccos(z_j)\right) \right\|_{LM} =$$

$$= \left\| \cos\left(m \cdot \arccos\left(\frac{2j}{L-1} - 1 + \delta\right)\right) \right\|. \quad (11)$$

В частном случае можно полагать $\delta = 0$. Дискретизация аргументов $\{x, y\}$ по правилу (9) приводит к появлению погрешности квадратурной формулы и неточному вычислению спектральных коэффициентов. Для уточнения расчетов можно произвести интерполяцию значения поля $s(x, y)$ в точках расположения нулей полиномов Чебышева по ближайшим отсчетам.

4. Результаты исследования алгоритма GDCT

Итак, в блоке изображений размером $N1 \times N1$ элементов выбирается матрица $N \times N$ отсчетов ($N < N1$). Так происходит сжатие на ступени I процедуры.

Затем с помощью преобразования GDCT вычисляется матрица $M \times M$ из наиболее значимых спектральных коэффициентов, которые в дальнейшем прореживаются или при помощи матрицы квантования, или путем регулируемой низкочастотной фильтрации. Так происходит сжатие изображения на II ступени процедуры.

При восстановлении изображения формируется блок размером $L \times L$ элементов. Размер блока $L \times L$ может быть меньше, равен или больше размеров исходного блока $N1 \times N1$.

Для экспериментов был выбран набор искусственных изображений и натуральных фотографий. Цветные изображения были представлены в 24 битовом формате BMP. С этими изображениями осуществлялись следующие операции:

1) Перекодировка 8-битовых R, G, B-компонент цвета в Y, U, V по правилу:

$$Y = \text{int}(0.299R + 0.587G + 0.114B),$$

$$U = \text{int}(0.500R - 0.419G - 0.081B) + 128,$$

$$V = \text{int}(-0.169R - 0.331G + 0.500B) + 128.$$

2) Субдискретизация матриц Y, U, V по одному из законов: 4:4:4, 4:2:2, 4:2:0.

3) Спектральное преобразование матриц Y, U, V на основе GDCT. В зависимости от типа изображения на первой ступени сжатия проводилось прореживание в пропорции $N1/N = 16/6, 16/8, 8/6$. Затем проводилось преобразование спектральной матрицы **C** для реализации второй ступени сжатия изображения. При этом производилась «низкочастотная» фильтрация матрицы **C** путем обнуления коэффициентов $C_{m,j}$ при $m+1 > K$, где K регулировалось.

4) Для упрощения сопоставления исходного и восстановленного изображений проводилось восстановление блока размером $N1 \times N1$, то есть было выбрано $N1 = L$.

Затем производилась визуальная оценка качества изображения. На рис. 1-4 показаны результаты преобразований объекта типа «портрет» с помощью алгоритмов GDCT и JPEG. Степень сжатия за счет усечения спектров в алгоритмах JPEG и GDCT взята 5.6. Но в алгоритме GDCT дополнительно сжатие первой ступени в 1.8 (рис. 2) и 4 раза (рис. 3).

Исследования показали, что изображения с плавными градиентами яркости и цвета могут быть подвергнуты сильному сжатию уже на I ступени процедуры. Насыщенные мелкими деталями текстуры и натурные изображения испытывают искажения в GDCT, и рекомендуемая степень сжатия I ступени порядка 30-50%. Скорость убывания спектров в GDCT такого же порядка, как и в классическом дискретном косинусном преобразовании. Поэтому сжатие на II ступени такого же порядка, как и в алгоритме JPEG.

Так как положение узлов полиномов Чебышева образуют дробные числа, то были исследованы две реализации GDCT — с округлением положения узлов до ближайшего пикселя и с межпиксельной линейной интерполяцией. Эксперименты показали, что интерполяция улучшает передачу тонких деталей и наклонных краев изображений.

Выводы

Предложен метод сжатия изображения, основанный на разложении по полиномам Чебышева и применении квадратурных формул Гаусса-Чебышева (алгоритм GDCT). Показано, что этот алгоритм является обобщением дискретного косинусного преобразования, используемого в системах сжатия JPEG и MPEG. Алгоритм имеет дополнительную степень сжатия. Выполнено сравнение алгоритмов сжатия GDCT и классического JPEG для различных типов изображений. Показано, что в ряде случаев алгоритм GDCT позволяет увеличить степень сжатия до 3 раз по сравнению с алгоритмом JPEG. Поскольку предложенный алгоритм восстанавливает сигнал в любой заданной точке, его применение открывает возможности геометрического масштабирования при восстановлении сжатого изображения.

Сходство GDCT и DCT позволяет их объединять, дополняя друг друга, использовать существующие программно-аппаратные средства для обеспечения работы такого алгоритма. В частности, к алгоритму GDCT применима процедура быстрого преобразования Фурье.

Автор благодарен М.Ю. Радченко и к.т.н. А.Ю. Савинкову за помощь при разработке программного кода алгоритма GDCT на языке C++.

Литература

1. А.В. Дворкович, В.П. Дворкович, Б.Н. Мохин и др. Единые принципы сжатия цветных динамических изображений различного разрешения. Цифровая обработка сигналов, №1, 1999, с. 27-35.

2. Цифровая обработка телевизионных и компьютерных изображений./ Под ред. Ю.Б. Зубарева и В.П. Дворковича. М.: МЦНТИ, 1997, - 217 с.

3. Ю.С. Радченко, М.Ю. Радченко Оптимальные быстрые алгоритмы представления изображений в базисе ортогональных полиномов. Труды I международной конференции "Цифровая

обработка сигналов и ее применения" DSPA'98, 1998, Москва, т. III, с. 163-166.

4. Ю.С. Радченко, А.Ю. Кожин, М.Ю. Радченко Обнаружение и оценка параметра сдвига сжатых с помощью ортогональных полиномов сигналов. Радиотехника, №6, 1999, с. 17-19.

5. Ю.С. Радченко, А.Ю. Савинков Исследование алгоритмов сжатия изображений на основе полиномиальных преобразований (алгоритм GDKT). Труды III конференции "Цифровая обработка сигналов и ее применения" DSPA'2000, 2000, Москва, т. II, с. 89-91



Рис. 1. Исходный



Рис. 2. GDCT 8→8



Рис. 3. GDCT 12→6



Рис. 4. JPEG — 5.6

Анализ четкости цветных статических изображений в равноконтрастной системе координат

С.В. Сай

Введение

Системы цифровой обработки и передачи статических изображений применяются в различных областях науки и техники. В вещательном телевидении статические изображения широко используются в виде электронных испытательных таблиц с целью анализа искажений и настройки видеосистемы на требуемое качество. Развитие цифрового телевизионного вещания привело к разработке новых испытательных таблиц [1], которые содержат большое число динамических элементов, позволяющих выявить искажения цифрового кодирования, например, на основе стандарта MPEG. Таблицы содержат различные элементы для анализа искажений, обусловленные как внутрикадровым так и межкадровым кодированием. Качество изображения по таким таблицам обычно оценивается визуально с помощью субъективных экспертных оценок, что является трудоемким процессом и накладывает определенные ограничения при настройке и оптимизации параметров системы.

К одному из основных критериев качества изображения относится его четкость. Данный показатель численно оценивается разрешающей способностью системы, определяемой максимальным числом различимых линий ахроматической или цветной штриховой миры с различной пространственной ориентацией. Традиционным способом оценки разрешающей способности считается метод измерения максимальной пространственной частоты, при которой обеспечивается пороговый контраст воспроизводимого изображения предельно контрастной штриховой миры.

В аналоговых системах цветного телевидения измерения проводятся отдельно по каналам яркости и цветности. В этом случае характеристики изображения представлены двумя параметрами — разрешающей способностью по яркости и по цветности. Таким образом, разрешающая способность зависит от крутизны спада сквозных пространственно-частотных характеристик (ПЧХ) яркостного и цветоразностных каналов в области верхних частот, а также от значения выбираемого порога.

К недостаткам метода можно отнести следующее:

а) оценка четкости изображения по значениям пространственной частоты и амплитуды первой гармоники штриховой миры может привести к неверным результатам при исследовании искажений в цифровых видеосистемах, где в процессе обработки сигналов используются нелинейные преобразования;

б) зависимость порогового контраста от чувствительности зрения к восприятию цвета мелких деталей приводит к неоднозначности оценок, что затрудняет анализ;

в) испытательное изображение штриховой миры с предельным контрастом не позволяет оценить четкость изображения мелких малоконтрастных деталей.

В настоящей статье рассматривается метод анализа четкости в равноконтрастной системе координат с учетом зрительного восприятия мелких малоконтрастных деталей испытательного изображения.

Выбор цветового пространства

В общем случае зрительное восприятие и различимость мелких деталей статического изображения зависит от:

- углового размера (α) наблюдаемых объектов;
- исходного контраста (ΔL) объект-фон;
- цветовых координат (C_o) объекта;
- цветовых координат (C_ϕ) фона;
- общей визуальной зашумленности $\bar{\sigma}_\Sigma$ изображения.

При этом пороговый контраст деталей, при которых они еще различаются глазом, можно представить в виде функции:

$$\Delta L_{\text{пор}} = f(\alpha, \Delta, C_o, C_\phi, \bar{\sigma}_\Sigma)$$

$$\text{где } \Delta L_{\text{пор}} = \sqrt{(R_o - R_\phi)^2 + (G_o - G_\phi)^2 + (B_o - B_\phi)^2}$$

$C_o = \{R_o G_o B_o\}$ — цветовые координаты объекта, определяемые в системе RGB основных цветов приемника; $C_\phi = \{R_\phi G_\phi B_\phi\}$ — цветовые координаты фона;

$$\bar{\sigma}_\Sigma = \sqrt{(\alpha \bar{\sigma}_R^*)^2 + (\beta \bar{\sigma}_G^*)^2 + (\rho \bar{\sigma}_B^*)^2}; \quad \bar{\sigma}_R^*, \bar{\sigma}_G^*, \bar{\sigma}_B^* —$$

— взвешенные среднеквадратичные значения флуктуационных помех, полученные с учетом весовой функции видности шума; α, β, ρ — коэффициенты относительной видности помех.

Известно [2], что чувствительность зрения падает с уменьшением контраста и размера деталей, при этом человеческий глаз более чувствителен к изменениям яркости объекта и менее чувствителен к изменениям цветности. Флуктуационные помехи (ФП) ухудшают различение мелких малоконтрастных деталей. Также значение порогового контраста зависит от наблюдаемых цветов объекта и фона. Следовательно, важное значение приобретает выбор цветового пространства, в котором анализируются искажения. Использование в качестве цветового пространства RGB системы координат основных цве-

тов приемника приводит к тому, что пороговый контраст зависит от цветовых координат наблюдаемого объекта и, следовательно, не является постоянной величиной в пределах цветового пространства.

Более подходящим цветовым пространством является равноконтрастная система координат, что обусловлено примерно одинаковыми значениями порогов цветовых

различий в независимости от выбираемых цветов. Особенностью используемых на практике равноконтрастных систем координат является то, что они рекомендованы для анализа точности цветопередачи крупных деталей изображения. Однако в [3] рассмотрена возможность использования таких систем для анализа искажений мелких цветных деталей.

В настоящей работе использовано цветовое пространство МКО 1964 г. (Вышецки), рекомендованное в качестве весьма точной меры цветовых различий [4]. В данной системе координаты цвета $U^*V^*W^*$ определяются следующими соотношениями:

$$W^* = 25 Y^{1/3} - 17 \quad U^* = 13 W^* (u - u_{оп});$$

$$V^* = 13 W^* (v - v_{оп}).$$

где W^* — координата светлоты (индекс яркости); U^* и V^* — координаты цветности (индексы цветности); Y — координата яркости ($1 \leq Y \leq 100$); u и v — координаты цветности, полученные в диаграмме цветностей Мак-Адама; $u_{оп}$ и $v_{оп}$ — координаты цветности опорного белого цвета.

Переход от RGB-сигналов основных цветов приемника к равноконтрастным координатам осуществляется с помощью стандартных преобразований [4].

Цветовое различие определяется величиной радиус-вектора (r), значение которого оценивается числом минимальных цветовых порогов (МЦП):

$$r = 3 \sqrt{(W_1^* - W_2^*)^2 + (U_1^* - U_2^*)^2 + (V_1^* - V_2^*)^2} \quad (1)$$

где $U_1^* V_1^* W_1^*$ — цветовые координаты 1-го объекта; $U_2^* V_2^* W_2^*$ — цветовые координаты 2-го объекта. Цветовые различия между объектами считаются малозаметными [2] при $r \leq 2...3$ (МЦП).

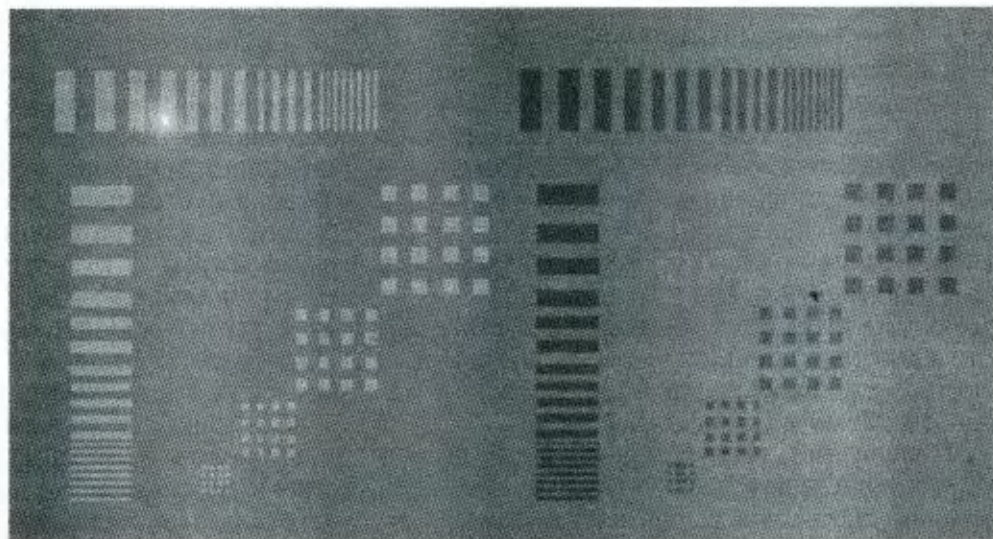


Рис. 1. Пример испытательного изображения

С другой стороны, выражение (1) представляет собой цветовой контраст между двумя объектами, и если, один из объектов является фоном, то контраст объект-фон находится как

$$\Delta L = 3 \sqrt{(W_o^* - W_\phi^*)^2 + (U_o^* - U_\phi^*)^2 + (V_o^* - V_\phi^*)^2} \quad (2)$$

В этом случае полагаем, что крупный объект изображения различается глазом на уровне фона при условии

$$\Delta L > \Delta L_{пор}, \quad (3)$$

где $\Delta L_{пор} = 2...3$ (МЦП) — пороговое значение.

С уменьшением размеров деталей контрастная чувствительность зрения падает, что приводит к увеличению порогового значения. Другими словами, уменьшение размеров деталей приводит к потере свойства равно контрастности системы ($U^*V^*W^*$) и, следовательно, критерий различения становится зависимым от α :

$$\Delta L > \Delta L_{пор}(\alpha) \quad (4)$$

Метод анализа

С целью решения задачи анализа четкости, с учетом зрительного восприятия мелких малоконтрастных деталей испытательного изображения, разработан метод, который можно разбить на следующие этапы.

На *первом этапе* формируется испытательное изображение, состоящее из следующих фрагментов:

- 1) горизонтальная и вертикальная штриховая мира с различной толщиной штрихов;
- 2) структурная текстура, состоящая из повторяющихся малоразмерных объектов.

Исходный цветовой контраст объект-фон задается в равноконтрастной системе координат.

На *втором этапе* выполняется преобразование

$$\{W^*(i,j), U^*(i,j), V^*(i,j)\} \rightarrow \{R(i,j), G(i,j), B(i,j)\}$$

необходимое для представления изображения в RGB системе сигналов, где (i, j) — пространственные координаты.

На рис. 1. показан пример испытательного изображения, где слева расположены объекты с исходным контрастом по индексу яркости: $dL_{(w^*)} = 12$ (МЦП) при условии, что $U_o^* = U_\phi^*$ и $V_o^* = V_\phi^*$. Справа — по индексу цветности: $dL_{(U^*)} = 160$ (МЦП) при условии, что $W_o^* = W_\phi^*$ и $V_o^* = V_\phi^*$. Цвет фона — серый ($W_\phi^* = 80$).

Ниже показано, что выбранные значения контраста в два раза превышают пороговые для пространственного размера объекта в группе равному одному элементу раstra.

На третьем этапе испытательное изображение кодируется, передается по каналу и декодируется на основе используемого алгоритма сжатия. На выходе видеосистемы изображение восстанавливается в цифровые значения RGB-сигналов и выполняется обратное преобразование

$$\{\tilde{R}(i,j), \tilde{G}(i,j), \tilde{B}(i,j)\} \rightarrow \{\tilde{W}^*(i,j), \tilde{U}^*(i,j), \tilde{V}^*(i,j)\}$$

где знак (\sim) указывает на то, что анализируются значения цветовых координат на выходе системы (входе дисплея).

На четвертом этапе анализируется четкость восстановленного изображения. Рассмотрим более подробно алгоритм анализа.

Четкость анализируется методом измерения цветового контраста для каждой группы элементов фрагмента изображения и сравнением с пороговым значением. Использование фрагментов изображения в виде вертикальной и горизонтальной штриховых мир, а также текстуры (см. рис. 1), позволяет оценить четкость по всем пространственным направлениям. Контраст анализируемых фрагментов изображения вычисляется следующим образом:

$$\Delta \tilde{L}(\delta) = \min_k \{M[\Delta \tilde{L}_n(\delta, k)] - \sigma[\Delta \tilde{L}_n(\delta, k)],\} \quad (5)$$

где $M[\Delta \tilde{L}_n(\delta, k)]$ и $\sigma[\Delta \tilde{L}_n(\delta, k)]$ — среднее значение и среднеквадратичное отклонение контраста по площади анализируемого фрагмента, состоящего из n объектов; δ — пространственный размер объекта в группе, определяемый числом элементов раstra; k — параметр, определяющий тип анализируемого фрагмента ($k=0$ — вертикальная штриховая мира, $k=1$ — горизонтальная штриховая мира, $k=2$ — текстура).

В качестве критерия оценки выбираем следующее условие — линии штриховой мира и элементы текстуры считаются различимыми, если мини-

мальное значение (5) цветового контраста объектов превышает пороговое значение.

Пороговые значения контраста получены экспериментальным путем с помощью субъективных оценок. При проведении эксперимента наблюдателю предлагалось испытательное изображение (рис. 1), синтезированное в среде C++Builder 5 с регулируемыми значениями цветового контраста по осям W^* и $U^*(V^*)$ с шагом 1 (МЦП). В процессе эксперимента наблюдатель изменял значения контраста, начиная с минимального, до тех пор, пока линии мира и элементы текстуры не становились различимыми в выбираемых фрагментах изображения. В эксперименте принимали участие студенты и сотрудники Хабаровского государственного технического университета. В табл. 1. приведены усредненные субъективные зависимости пороговых значений цветового контраста от размера объектов для испытательного изображения, показанного на рис. 1.

Как показали результаты эксперимента, значения порогов различения мелких деталей на сером фоне по осям U^* и V^* примерно одинаковы и превышают значения порогов по оси W^* в 12...13 раз.

На последнем, пятом этапе, производится оценка качества декодированного испытательного изображения путем сопоставления численных результатов анализа с пороговыми значениями (табл. 1). Для получения декодированного изображения с высокой четкостью необходимо выполнение условия [3]:

цветовой контраст линий штриховых мир и элементов текстуры должен быть не менее соответствующих пороговых значений $\Delta \tilde{L}_{w^*}(\delta)$ по индексу яркости для всех анализируемых фрагментов с размерами объектов $\delta \geq 1$, и не менее — по индексу цветности с размерами $\delta \geq 2$.

При выполнении такого условия качество изображения соответствует оценкам «хорошо» и «отлично» по субъективной пятибалльной шкале оценок.

Пример анализа четкости декодированного изображения

В качестве примера проанализируем четкость изображения в системе с цифровым сжатием на

Таблица 1. Зависимости пороговых значений контраста от размера объектов

δ	5	4	3	2	1
$\Delta \tilde{L}_{w^*}$	2	2	3	4	6
$\Delta \tilde{L}_{U^*}$	25	25	40	55	80

Таблица 2. Зависимости снижения контраста малоразмерных объектов

q	10	9	8	7	6	5	4	3	
$\Delta \tilde{L}_{w^*}$	1,7	1,4	1,3	1,0	0,5	0	0	0	$\delta = 1$
$\Delta \tilde{L}_{U^*}$	1,9	1,8	1,6	1,2	1,0	0,2	0	0	

основе стандарта JPEG, реализованного в графическом редакторе Adobe Photoshop 5.0.

В табл. 2. приведены зависимости снижения нормированных к пороговому значению цветового контраста линий штриховых мир и элементов текстуры испытательного изображения (см. рис. 1) от параметра q матрицы квантования коэффициентов трансформации для размеров объектов $\delta = 1$ по индексу яркости и по индексу цветности.

Как следует из полученных результатов, высокая четкость обеспечивается при выбранном параметре $q \geq 7$.

Визуальное сравнение качества исходного и декодированного изображений показало, что при выбранном q специфические искажения [1], обусловленные алгоритмом JPEG (блокинг-эффект, мозаичный эффект, окантовки на границах, искажения типа «ступеньки», искажения, имеющие вид базисных функций), практически не заметны для глаза и не снижают качество декодированного изображения.

Выводы

Рассмотренный метод анализа четкости по сравнению с известными обладает следующими особенностями и преимуществами.

1. Анализ четкости в равноконтрастной системе координат с учетом значений порогового контраста малоразмерных деталей, при которых они различаются глазом, позволяет объективно оце-

нить качество воспроизведения мелких малоконтрастных деталей изображения без дополнительных субъективных экспертиз.

2. Использование операций усреднения и вычисления среднего квадратичного отклонения в выражении (5) по площади анализируемого фрагмента позволяет учесть влияние помех и возможных нелинейных преобразований на четкость декодированного изображения.

3. Анализ четкости непосредственно по цифровым сигналам испытательного изображения на выходе системы (входе дисплея) учитывает влияние искажающих факторов всех звеньев системы в целом, что позволяет оперативно настраивать и оптимизировать параметры системы на требуемую четкость.

Литература

1. Дворкович А.В., Дворкович В.П., Макаров Д.Г., Новинский Н.Б., Соколов А.Ю. Испытательные таблицы для измерения качества цифрового и аналогового телевизионного вещания // М.: "625", № 8, 1999, С. 36-42.

2. Певзнер Б. М. Качество цветных телевизионных изображений. М.: Радио и связь, 1988, 244 с.

3. Сай С.В. Четкость цветного изображения в системах со сжатием визуальных данных. Хабаровск: Изд-во Хабар. гос. техн. ун-та, 1999, 143 с.

4. Новаковский С.В. Цвет в цветном телевидении. М.: Радио и связь, 1988, 288 с.

Уважаемые читатели!

Продолжается подписка на журнал «Цифровая обработка сигналов» на 2002 год. Периодичность подписки — 4 номера в год. Стоимость подписки на год — 440 руб., без учета НДС (10%).

Справки об условиях подписки и доставки журналов по телефонам:
(095) 105-3464, 105-3468, 267-0662

Желающих подписаться просим заполнить подписной лист и отправить его по факсу: (095) 105-34-64 или по электронной почте subscribe@dspa.ru

Подписной лист

Организация _____

Почтовый адрес _____

Контактное лицо _____

Телефон, факс, электронная почта _____

Банковские реквизиты:

Наименование банка: _____ Город: _____

Р/с: _____

К/с: _____ БИК: _____

Особенности построения программного телевизионного кодера MPEG-2

А. В. Дворкович

Цифровые телевизионные кодирующие устройства, работающие в соответствии со стандартом MPEG-2 [1], требуют использования вычислительной системы сравнительно высокой производительности. Поэтому до последнего времени они реализовывались только аппаратно с использованием специальных процессоров и микросхем. Однако в связи с появлением в последнее время недорогих высокопроизводительных мультипроцессорных персональных компьютеров появилась возможность создать программный телевизионный кодер MPEG-2 без специальной аппаратной вычислительной поддержки.

Эффективная программная реализация такого кодера связана с наиболее полным использованием возможностей архитектуры конкретной вычислительной системы. Описываемый кодер был оптимизирован для работы на многопроцессорном персональном компьютере (на процессорах Intel Pentium IV) под управлением операционных систем Linux и MS Windows 2000, которые обеспечивают полноценную поддержку мультипроцессорной обработки.

Оптимизация программы проводилась в нескольких направлениях: эффективное использование многопроцессорной обработки, использование специальных операций процессора, оптимизация работы программы с памятью.

Для того чтобы наиболее полно загрузить работой все имеющиеся в компьютере процессоры необходимо разбить программу на потоки, осуществляющие независимую параллельную обработку данных. Распределение программных потоков по процессорам осуществляется самой операционной системой. Обработка каждым потоком выделенной части кадра (половины, четверти) не является эффективным решением, так как одна из частей может быть обработана быстрее других (например, один участок кадра статичен, а остальные содержат движущиеся детали). В таком случае процессор, осуществлявший обработку статичной части изображения, будет простаивать до окончания обработки остальных частей кадра. Более эффективным решением является разбиение кадра на множество независимо обрабатываемых частей, которыми в случае использования стандарта MPEG-2 могут быть слайсы или макроблоки. Каждый поток по окончании обработки текущей части запрашивает у управляющего потока следующую необработанную часть. При этом важно, чтобы накладные расходы на синхронизацию работы потоков (запросы управляющему пото-

ку) были небольшими по сравнению с основной частью вычислений.

Процесс кодирования кадра может быть разбит на четыре последовательные стадии:

1 — получение данных и их предварительная обработка (в том числе префильтрация);

2 — поиск векторов движения, оценка мощности сигнала в макроблоке;

3 — компенсация движения, блочное преобразование, квантование, восстановление (обратное квантование, обратное косинусное преобразование и компенсация движения, процедура восстановления проводится только для I- и P-кадров) и кодирование;

4 — формирование выходного потока, подготовка к кодированию следующего кадра.

В пределах второй и третьей стадий обработка может вестись параллельно несколькими потоками. На первой и последней стадии работа программы может быть распараллелена частично, но эти стадии занимают не более 5% общего времени обработки кадра, поэтому неполное использование многопроцессорной обработки на этих стадиях кодирования мало сказывается на эффективности работы кодера в целом. Основное же время работы программы кодирования приходится на вторую и третью стадии.

Поиск векторов движения макроблоков может проводиться для каждого макроблока или слайса независимо от других при достаточной производительности вычислительной системы. Однако при недостатке этой производительности появляется необходимость уменьшать область поиска векторов движения с помощью предсказания наиболее вероятного значения вектора по величинам векторов соседних в пространстве и во времени макроблоков. Но и в случае поиска векторов движения с предсказанием, использующим значения векторов движения соседних макроблоков, можно использовать многопоточную обработку. В этом случае для формирования предсказания используются вектора макроблоков соседних кадров и тех макроблоков текущего кадра, которые уже найдены на момент начала обработки данного макроблока. Заметно повышает эффективность работы переход от обработки макроблоков в порядке сканирования (слева направо и сверху вниз) к обработке по спирали, начиная с центра кадра. В этом случае можно более качественно искать вектора движения в центре кадра (на котором, как правило, сосредоточено внимание наблюдателя) и менее качественно на периферии. Комбинирование методов ускоренного поиска векторов движения [2,3] с ме-

Таблица 1

Тип кадра	Доля времени обработки кадра по стадиям, %			
	1 стадия	2 стадия	3 стадия	4 стадия
I	2	2	86	10
P	1	54	40	5
B	1	69	30	-

тодами, использующими предсказание [4,5], позволяет проводить обработку данных в реальном масштабе времени.

Третья стадия кодирования (компенсация движения, блочное преобразование, восстановление и статистическое кодирование) в соответствии со стандартом MPEG-2 сможет проводиться независимо для слайсов кадра и поэтому эффективно выполняется несколькими потоками.

Показателем использования процессоров является доля их загруженности во время выполнения задачи. Эффективное распараллеливание работы программы позволяет загружать процессоры на 95 - 98 %.

Следующий этап оптимизации работы программы — использование специальных операций процессора. Процессор Pentium IV имеет множество специальных команд и регистров, предназначенных для быстрой обработки мультимедийных данных.

Наши эксперименты показали, что использование набора специальных операций процессора Pentium IV (SSE2, SSE, MMX) при выполнении основных ресурсоемких частей алгоритма кодирования (поиск движения, компенсация движения, прямое и обратное косинусные преобразования) дает сравнительно небольшой выигрыш по скорости выполнения (5 - 20 %). Это связано с малым количеством специальных регистров процессора. Небольшой объем кэш-памяти процессора так же не позволяет эффективно работать с большими участками данных телевизионного изображения, которые приходится обрабатывать при поиске движения.

Достаточно существенный выигрыш по скорости работы кодера дает использование ряда стандартных приемов сокращения времени выполнения программ: замена деления умножением и сдвигом, выполнение некоторых операций (умножение, взятие модуля, возведение в квадрат) по таблице и др.

Результаты статистического анализа времени выполнения различных стадий обработки кадров при работе программного телевизионного MPEG-2 кодера приведены в табл. 1.

Прочерк на месте 4 стадии обработки В-кадра означает, что эта стадия отсутствует.

Приведенные в табл. 1 результаты подтверждают, что основное время работы программы кодера приходится на поиск векторов движения, компенсацию движения, кодирование с использованием косинусного преобразования, квантование и восстановление кадров.

Проведенные оценки времени обработки различных типов кадров (I, P, B) показали, что использование ряда быстрых, но эффективных алгоритмов поиска движения (включая интеллектуальное предсказание для уменьшения области поиска) вместе с описанной выше оптимизацией позволяет в настоящий момент реализовать программный кодер MPEG-2 реального времени на двух-процессорном персональном компьютере.

Литература

1. ISO/IEC 13818-2 (ITU-T Recommendation H.262) *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video, 1995.*
2. Зубарев Ю.Б., Дворкович В.П., Неченаев В.В., Соколов А.Ю. *Анализ и компенсация движения в системе MPEG-2 - основа эффективного кодирования телевизионных изображений. 1-я Международная Конференция "Цифровая обработка сигналов и ее применение", Москва, МЦНТИ, 30 июня - 3 июля 1998 г. Доклады, т.3, стр. 134-139.*
3. Дворкович А.В., Дворкович В.П., Соколов А.Ю. *Способ поиска векторов движения деталей в динамических изображениях. Заявка на патент РФ №2000119205/09 (020355) от 20.07 2000.*
4. S. Kim, C.-C.J. Kuo. *Fast Motion Vector Estimation with a Markov Model for MPEG. Proc. Of Digital Video Compression: Algorithms and Technologies, 7-10 Feb. 1995, San Jose, California, SPIE Vol. 2419, p. 210.*
5. F. Kossentini, Y.-W. Lee, M.J.T. Smith, R.K. Ward *Predictive RD Optimized Motion Estimation for Very Low Bit-Rate Video Coding. IEEE J. on Selected Areas in Commun., Dec. 1997, Vol. 15, Num. 9, pp. 1752-1763.*

Применение процессора NM6403 (L1879BM1) для сжатия изображений

С.В. Мушкаев, С.В. Ландышев

Введение

В задачах с достаточно трудоемкими вычислениями всегда возникает вопрос о наиболее эффективном способе их реализации. Для различных архитектур он в каждом случае свой, но в целом принцип сводится к минимизации арифметических операций. Для процессора NM6403 это справедливо только отчасти, так как в ряде случаев полная минимизация вычислений приводит к неэффективности использования векторного узла из-за невозможности распараллелить алгоритм. Таким образом, основная задача заключается в выборе оптимального набора однотипных макроопераций, оперирующих блоками данных, таких как: умножение или суммирование отдельных частей матриц и векторов. К примеру, данный принцип позволяет эффективно организовывать на процессоре NM6403 вычисления БПФ по основанию 16 или 32 [3].

Алгоритм вычисления двумерного ДКП на процессоре NM6403

Прямое вычисление ДКП 8x8 можно рассматривать как предельный случай, когда дополнительная минимизация уже не требуется. Выделим эти макрооперации, исходя из формулы прямого ДКП.

Двумерное дискретное косинусное преобразование над блоком 8x8 определяется формулой

$$Y(u,v) = \frac{1}{4}w(u)w(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right],$$

где $u, v, x, y = 0, 1, 2, \dots, 7$

x, y — пространственные координаты пикселей в блоке 8x8,

u, v — координаты в частотной области,

$w(u) = 1/\sqrt{2}$ если $u = 0$; 1-в остальных случаях,

$w(v) = 1/\sqrt{2}$ если $v = 0$; 1-в остальных случаях.

В эквивалентном матричном виде формула прямого ДКП записывается в виде

$$[Y] = [C] \times [X] \times [C^T]$$

$[X]$ — исходная матрица 8x8;

$[Y]$ — результирующая матрица 8x8 дискретного косинусного преобразования;

$[C], [C^T]$ — матрицы 8x8 коэффициентов преобразования (косинусов) C_{ij} .

В данной статье анализируются возможности применения процессора NM6403 в области сжатия статических изображений. Рассматривается алгоритм реализации ДКП (дискретного косинусного преобразования), исследуются вопросы повышения точности вычислений. Приводится краткое описание реализации JPEG кодера на базе процессора NM6403, а также некоторые аспекты возможности применения векторного ядра в задачах кодирования

Таким образом, вычисление ДКП состоит в перемножении входной матрицы на две матрицы C и C^T . Данное умножение можно осуществить двумя способами:

1. $[Y] = ([C] \times [X]) \times [C^T]$;
2. $[Y] = [C] \times ([X] \times [C^T])$.

Так как вычисления строятся на арифметике с фиксированной точкой, то при реализации ДКП на векторном процессоре данные варианты имеют существенные различия. Выбор наиболее предпочтительного зависит от разрядностей элементов матриц X, C и C^T . Структура векторного умножителя такова, что накопление результата при перемножении двух матриц $[C]$ и $[X]$ происходит в той же разрядной сетке, которую имеет множитель — $[X]$. Поэтому первый вариант накладывает более жесткие ограничения на входные данные. Второй же вариант не привязан к разрядности входных данных и, следовательно, является более универсальным и гибким в выборе разрядности коэффициентов C и C^T , в силу этого ниже будет рассматриваться именно этот вариант.

Процесс умножения делится на два этапа:

1. Вычисление промежуточной матрицы $[T] = [X] \times [C^T]$

2. Вычисление конечной матрицы $[Y] = [C] \times [T]$

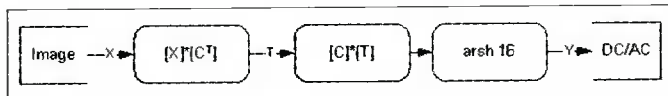
Проиллюстрируем одну из наиболее простых и быстрых реализаций алгоритма ДКП для обработки 8 разрядных входных данных. Разрядность матриц C^T и Y удобно взять равной 32 битам, это предотвратит возможность переполнения и упростит последующую за ДКП обработку данных. Точность преобразования зависит от количества значимых бит в элементах матриц C и C^T , как будет показано ниже использование 7 значимых младших бит и знака, обеспечивает приемлемую точность. Таким образом, разрядность матрицы C достаточно взять равной 8 битам.

Заметим, что все элементы матриц интерпретируются как знаковые, поэтому диапазон входных элементов X_{ij} лежит в пределах $-128 \dots 127$, что не исключает возможность обработки и обычных изображений с 256 градациями серого.

Коэффициенты преобразования с плавающей точкой (индекс i_p) $C_{fp}(i,j)$ лежат в диапазоне $(-0.491 \dots +0.491)$, перевод их в числа с фиксированной точкой (индекс i_k) осуществляется по

формуле $C_{fix}(i,j)=round(C_{fp}(i,j)*2^8)$, где 2^8 — масштабирующий множитель, $round()$ — функция округления до ближайшего целого. Использование одинакового масштабирующего множителя — 2^8 как для матрицы $[C]$, так и для $[C^T]$ позволяет накапливать результат в 32-х разрядной сетке матриц $[T]$ и $[Y]$ без переполнения и использовать единый нормирующий сдвиг на 16 бит вправо ($arsh 16$). То есть, вся процедура вычисления ДКП состоит из двух этапов умножения и нормирующего сдвига:

В развернутом виде последовательность умножения матриц и внутреннее разбиение упакованных данных показано на рис. 1 (жирными линиями выделены границы 64 разрядных слов,



тонкими — разбиение на составляющие элементы).

На первом этапе умножения вычисляется промежуточная матрица $[T]=[X]*[C^T]$. Коэффициенты $[C^T]$ загружаются в матрицу весовых коэффициентов по две колонки, а восемь векторов матрицы $[X]$ последовательно поступают на вход векторного умножителя (см. рис. 1). За каждый

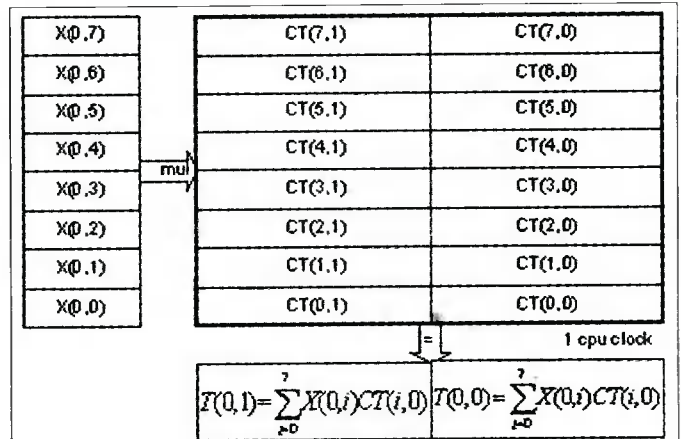


Рис. 2. Схема умножения нулевой строки матрицы $[X]$ на два левых столбца матрицы $[C^T]$. Этап 1

такт происходит умножение одного вектора матрицы $[X]$ на загруженную матрицу весовых коэффициентов, т.е. находится пара чисел матрицы $[T]$: $T(i,j)$ и $T(i,j+1)$ (рис. 2). Таким образом, полное перемножение двух матриц 8×8 требует 4 перезагрузки матрицы коэффициентов и происходит за 32 шага умножения. Учитывая, что коэффициенты матрицы $[C^T]$ постоянны и при работе со всем изображением не требуют постоянной

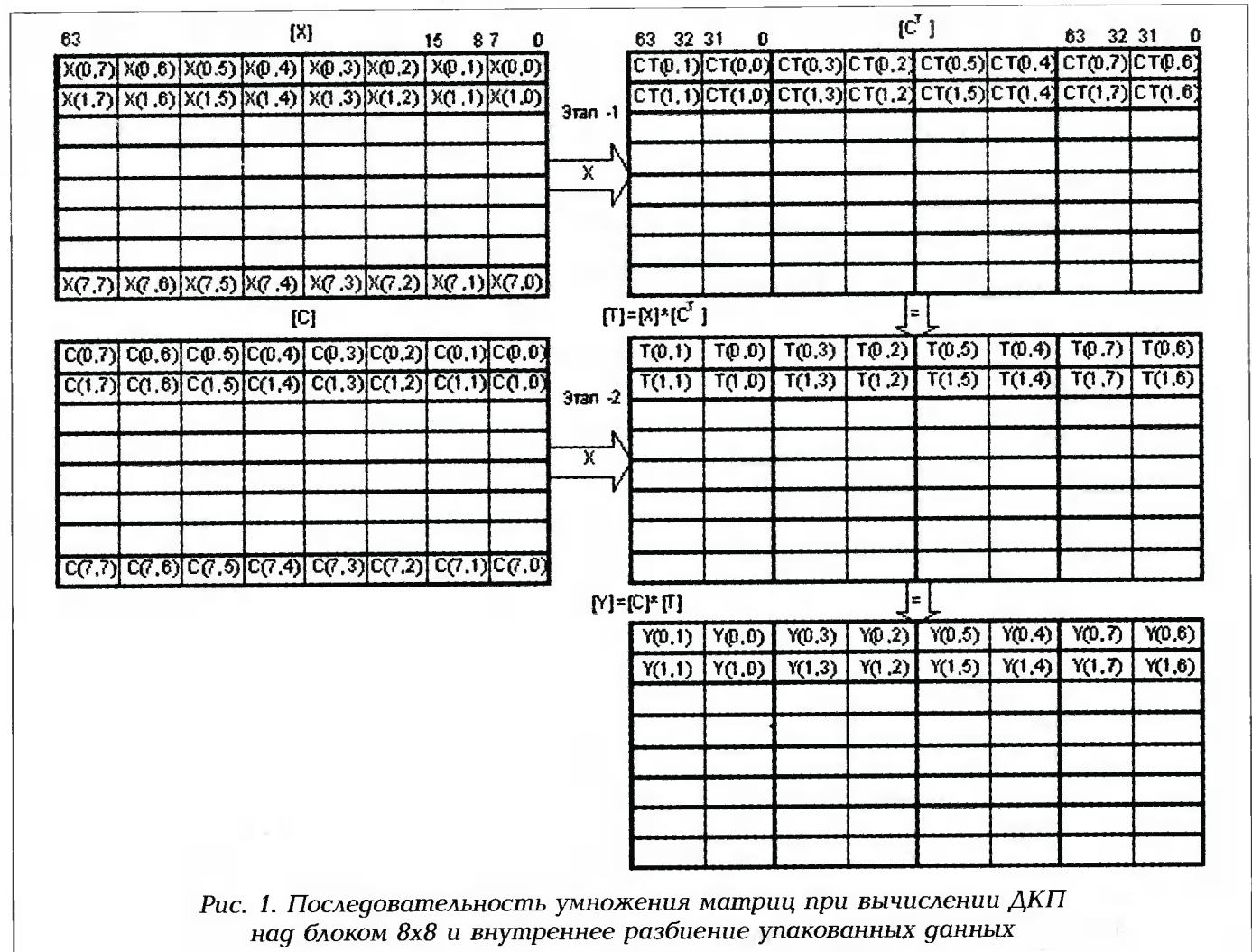


Рис. 1. Последовательность умножения матриц при вычислении ДКП над блоком 8×8 и внутреннее разбиение упакованных данных

перезагрузки, среднее время вычисления одного элемента матрицы [T] составляет примерно 0.5 такта. Эффективность умножения также достигается за счет использованием SIMD инструкции — per 32, выполняющей умножение блоками по 32 вектора.

На втором этапе находится матрица $[Y]=[C]*[T]$. Элементы промежуточной матрицы [T] загружаются в матрицу весовых коэффициентов также по две колонки, а вектора матрицы [C] последовательно поступают на вход векторного множителя (см. рис. 3). Процесс умножения такой же как и на первом этапе за тем исключением, что на данной стадии возможно оперировать блоками только по восемь 64-разрядных векторов (инструкция per 8), после чего матрицу весов необходимо перегружать. Так как перегрузка матрицы занимает 32 такта и на фоне 8 умножений не удастся полностью совместить эти операции, то каждый элемент матрицы $Y(i,j)$ в среднем находится в 4 раза дольше, т.е. примерно за 2 такта.

В итоге, суммарное расчетное время умножения составляет $64*0.5+64*2=160$ тактов. Реальная цифра составляет около 190 тактов. К общему времени следует также добавить 32 такта, необходимых для нормализующего сдвига, однако эту операцию можно выполнить параллельно с процедурами, следующими за ДКП, такими как: квантование или Z-упорядочение. В этом случае среднее время обработки одного пикселя изображения можно считать равное 3 тактам.

Точность алгоритмов ДКП преобразования

При выборе способа реализации того и или иного алгоритма приходится одновременно учитывать несколько факторов: входной и выходной диапазон данных, точность получаемых результатов и общая производительность. Все они взаимно исключают друг друга, но как показал анализ, правильный подбор коэффициентов и незначи-

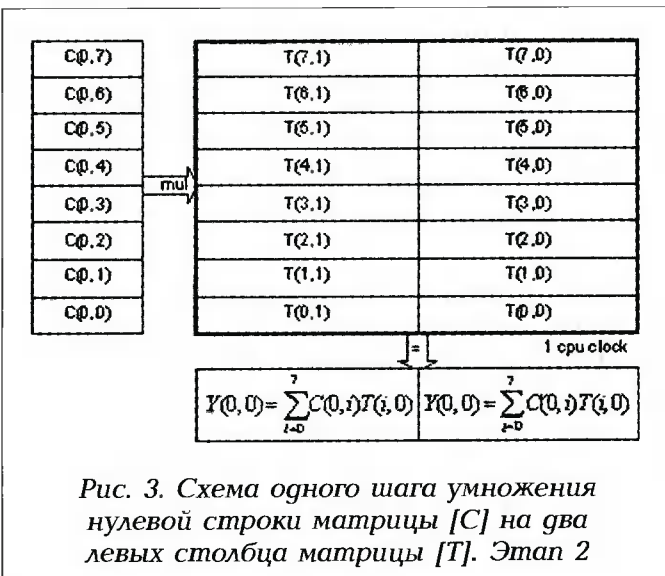
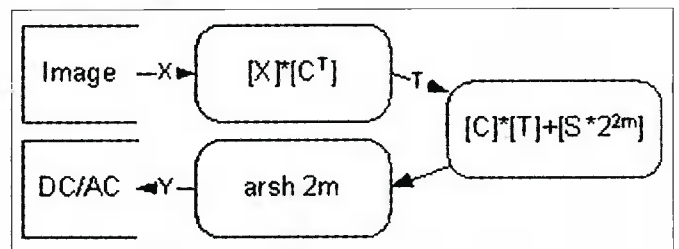


Рис. 3. Схема одного шага умножения нулевой строки матрицы [C] на два левых столбца матрицы [T]. Этап 2

тельная вариация вычислений может существенно повысить точность без потери общей производительности.

Точность вычисления ДКП зависит от формата представления коэффициентов преобразования C_{ij} в числах с фиксированной точкой, а также от способа конечной обработкой результатов. Формат представления коэффициентов определяется: разрядностью ячеек, масштабирующим коэффициентом-М и способом округления — $C_{fix} = \text{round}(C_{fp} * M)$. Конечная обработка состоит так же в округлении результатов умножения при отбрасывании дробной части. В случае ДКП коэффициенты преобразования C_{ij} лежат в диапазоне $(-0.491...+0.491)$, поэтому можно применять масштабирующий коэффициент — $M=2^m$, где m — эффективная разрядность ячеек (число значимых бит и знак). Тип округления в обоих случаях используется до ближайшего целого, так как эта схема показывает наилучшие результаты и на векторном ядре может быть реализована параллельно с операцией умножения путем прибавления к произведению $[C]*[T]$ константы $1/2*2^{2m}$ (число 0.5 с фиксированной точкой).

На основе данной схемы был проведен анализ зависимости точности ДКП преобразования от разрядности — m. Под разрядностью в данном контексте понимается количество значимых бит со знаком в n-битовом слове ($n \geq m$). Для просто-



ты коэффициенты обеих матриц [C] и $[C^T]$ вычислялись с одинаковым масштабирующим коэффициентом 2^m , а разрядная сетка для накопления результатов предполагалась достаточно широкой, чтобы исключить возможность переполнения.

Тестирование проводилось по всем блокам 8x8 изображения «Lena» размером 256x256.

Точность оценивалась по сходству результатов ДКП полученных на основе арифметики с фиксированной и плавающей точкой. Сравнение проводилось по нескольким критериям, ниже приводятся две наиболее показательные оценки:

$$RMSE(X) = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (x_{fix}(i) - x_{fp}(i))^2}$$
 — среднеква-

дратичная ошибка величины X по всему кадру. В качестве X подставлялись коэффициенты ДКП преобразования: DC (показатель средней яркости блока) и AC (значения пространственных частот). Раздельный анализ этих коэффициентов упрощает интерпретацию результатов.

$$PSNR(X) = 20 \cdot \log\left(\frac{255}{RMSE(X)}\right) \quad \text{— отношение}$$

сигнал/шум между исходным кадром и восстановленным изображением с помощью обратного ДКП по схеме:

Исх. кадр => ДКП_{fix} => ОДКП_{fp} => Восст. кадр

Как видно из рис. 4, при переходе от 8-разрядной точности коэффициентов преобразования к 9-разрядной наблюдается резкий перепад в точности DC коэффициентов.

Увеличение точности обусловлено особенностью записи коэффициентов C_{ij} , участвующих в вычислении DC, в формате с фиксированной точкой. Эти коэффициенты расположены в первом столбце матрицы $[C^T]$ и первой строке матрицы $[C]$, все они равны числу 0,35355339059327. В формате с фиксированной точкой это число записывается в двоичном виде как 0.0101 1010 1000 0010 0111... Видно, что после 9 знака после запятой следует 5 нулей, т.е. точность записи числа 0.35355 с помощью 9 бит (0101 1010 1b) будет такая же, как и с помощью 14 бит (0101 1010 1000 00b). В случае 8-бит ошибка связана с неопределенностью округления в большую или меньшую сторону, т.е. округление 8 бита до числа (0101 1011) практически также правомерно, как и до (0101 1011). Очевидно, что используя только один способ округления, полученные коэффициенты C_{ij} будут вносить некоторую систематическую ошибку в DC. Этот недостаток можно устранить, получая коэффициенты с помощью вероятностного округления, что равносильно поочередному округлению $C^T(i, 0)$ и $C(0, j)$ то в большую, то в меньшую сторону. Как видно из графиков (на рис. 5, рис. 8), в результате такой коррекции резко возрастает точность ДКП преобразования на разрядностях меньших 9.

В ряде задач часто требуется обрабатывать беззнаковые данные. В этих случаях дополнительное увеличение точности (см. рис. 7, рис. 8)

можно достичь, если предварительно перевести входные значения из диапазона $[0...2^m-1]$ в диапазон $[-2^{m-1}...+2^{m-1}-1]$, вычитая из входных данных число 2^{m-1} . Для обычных изображений это соответствует смещению диапазона $[0...255]$ в диапазон $[-128...127]$. В итоге результат ДКП

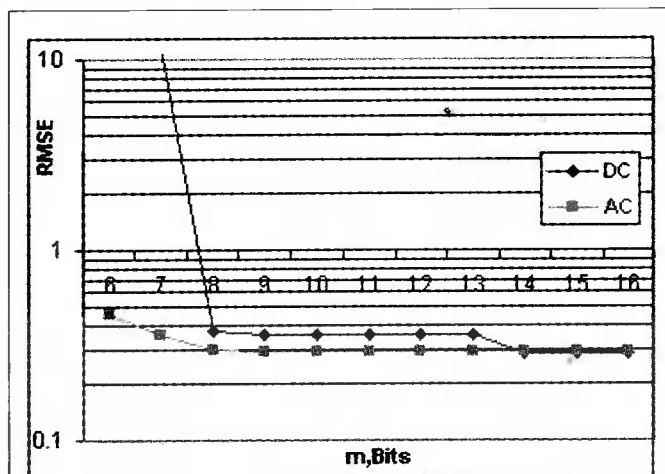


Рис. 5. Зависимость среднеквадратичной ошибки DC/AC коэффициентов от разрядности- m с коррекцией коэффициентов $C(i, j)$

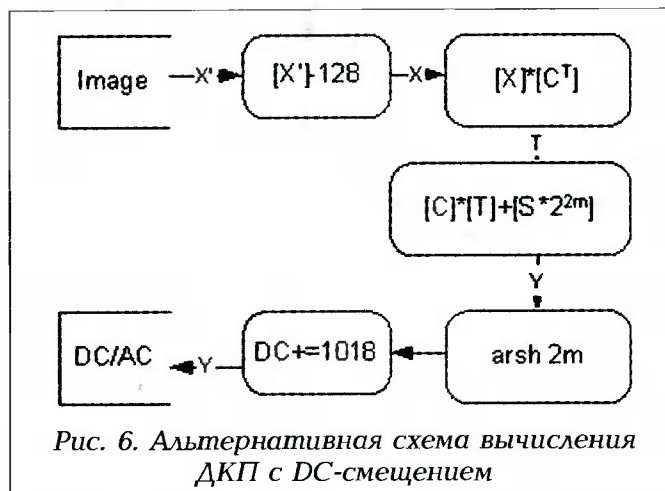


Рис. 6. Альтернативная схема вычисления ДКП с DC-смещением

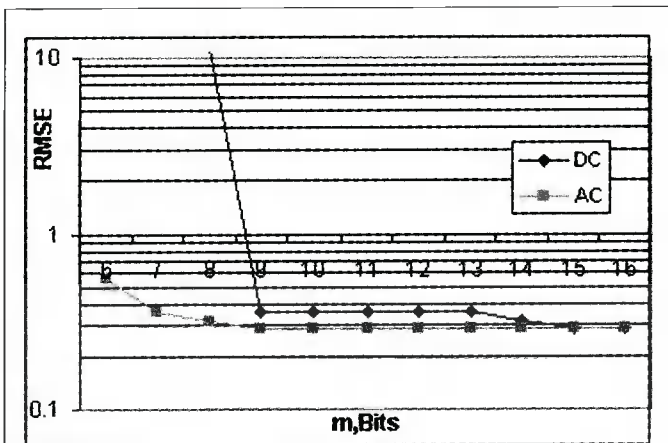


Рис. 4. Зависимость среднеквадратичной ошибки DC/AC коэффициентов от разрядности- m коэффициентов $C(i, j)$

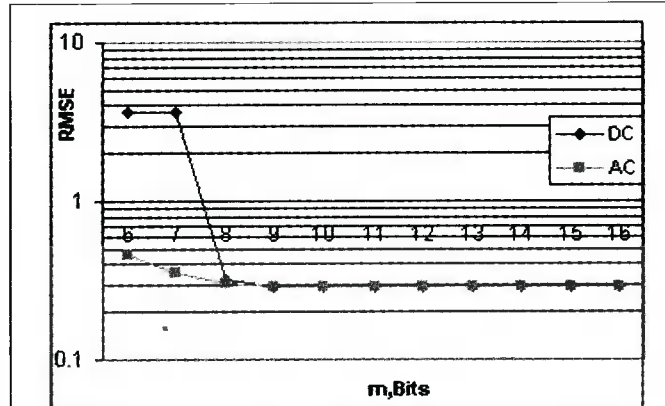


Рис. 7. Зависимость среднеквадратичной ошибки DC/AC коэффициентов от разрядности- m с коррекцией коэффициентов $C(i, j)$ и DC смещением

Краткое описание реализации JPEG кодера на процессоре NM6403

На рис. 9 приведена блок-схема JPEG-кодирования и относительное распределение процессорного времени между блоками (для NM6403). Для каждого блока указано усредненное время обработки (в процессорных тактах) одного пикселя изображения «Lena 256x256». Для последних трех блоков временные характеристики достаточно условны, так как они сильно зависят от таблиц квантования и самого изображения. Суммарное среднее время обработки одного пикселя изображения составляет около 15-16 тактов, что соответствует (при тактовой частоте NM6403 — 40 МГц) 25 кадрам в секунду для монохромных CIF изображений.

Вся предварительная обработка, состоящая из операций: DC-смещения, ДКП, квантования, Z-упорядочения и насыщения, достаточно хорошо векторизуется, так как эти процедуры не зависят от входных данных. Остальные операции связаны с кодированием (RLE) и упаковкой данных (VLC), в них присутствуют команды условного перехода и работа с кодами переменной длины, поэтому полностью векторная обработка данных становится невозможной. Однако в некоторых процедурах часть работы можно переложить на векторный узел с целью упрощения и ускорения последующей обработки на скалярном ядре.

В частности, в процедурах RLE и контроле кода с помощью векторной предобработки удается значительно сократить объем скалярных операций, а количество условных переходов свести к минимуму.

В обоих случаях суть векторной предобработки заключается в составлении битового вектора по некоторой анализируемой последовательности из 64-х элементов. Битовый вектор — это 64-разрядное слово, каждый бит которого определяется булевой функцией от соответствующего элемента последовательности: $b_j = f(x_j)$. В качестве простейших булевых функций на NM6403 возможно использовать следующие варианты:

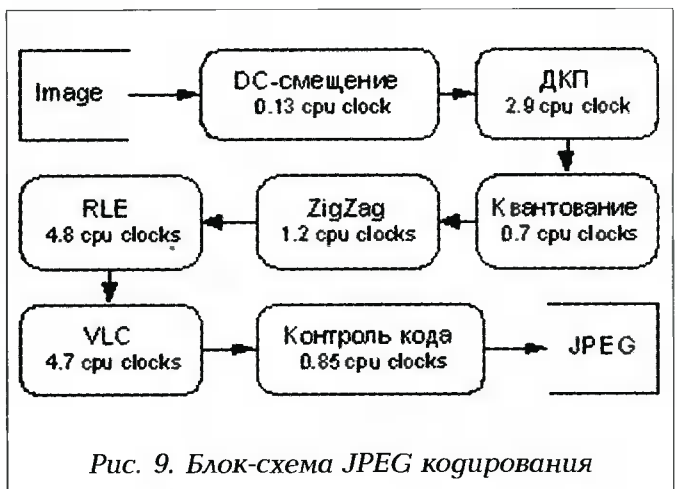


Рис. 9. Блок-схема JPEG кодирования

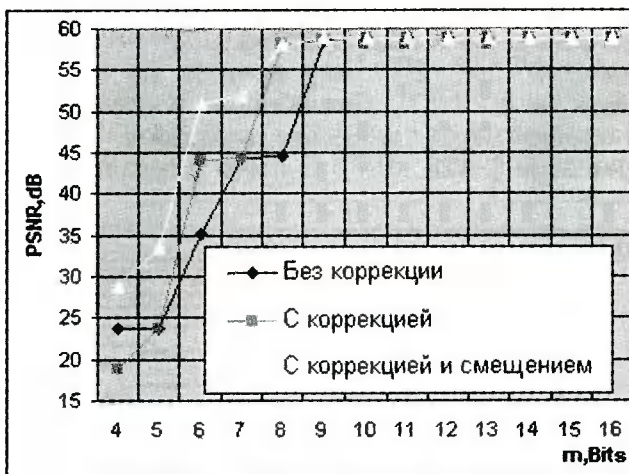


Рис. 8. Зависимость отношения сигнал/шум восстановленного изображения после ДКП от разрядности коэффициентов $C(i,j)$

Таблица 1. Зависимость отношения сигнал/шум восстановленного изображения от коэффициента квантования Q

Q	PSNR, dB		
	ДКП _{fix} -ОДКП _{fix}	ДКП _{fix} -ОДКП _{fp}	ДКП _{fp} -ОДКП _{fp}
1	52.84	58.65	58.9
2	48.41	49.63	49.65
4	45.74	46.38	46.39
8	42.13	42.37	42.37
16	38.03	38.12	38.12
32	33.78	33.82	37.82
64	29.80	29.84	29.84

преобразования будет отличаться от обычного варианта только DC коэффициентом на некоторую константу. После коррекции DC элемента результат полностью приводится к нормальному виду. Заметим, что механизм такого смещения заложен только в стандарте JPEG. Схема, приведенная на рис. 6, позволяет обрабатывать любые 8-разрядные изображения и в других стандартах: MPEG, H.263 и пр.

На практике схема, приведенная на рис. 6, с применением механизма DC-смещения позволяет без переполнения использовать более точное 12-битовое представление коэффициентов матрицы $[C]$. О точности такой схемы можно судить, сравнивая результаты прямого и обратного ДКП с плавающей и фиксированной точкой (см. табл. 1). В таблице приводится зависимость отношения сигнал/шум восстановленного изображения «Lena 256x256» от коэффициента квантования Q.

$$f(x_i) = \begin{cases} 0, x_i \geq 0 \\ 1, x_i < 0 \end{cases}, \quad f(x_i) = \begin{cases} 1, x_i \geq 0 \\ 0, x_i < 0 \end{cases}$$

$$f(x_i) = \begin{cases} 0, x_i = 0 \\ 1, x_i > 0 \end{cases}, \quad f(x_i) = \begin{cases} 0, x_i - \text{четный} \\ 1, x_i > 0 - \text{не четный} \end{cases}$$

Данные функции выполняются на векторном узле, поэтому за такт в битовом векторе можно накапливать по 2-4-8-16... бит в зависимости от разрядности элементов x_i . В сочетании с другими логическими и арифметическими функциями можно реализовать и более сложные варианты, например:

$$f(x_i) = \begin{cases} 0, x_i = 0 \\ 1, x_i \neq 0 \end{cases}$$

В частности, в процедуре контроля кода, следующей после упаковки кодов Хаффмана (процедура VLC) в бинарный поток, производится проверка на наличие в потоке символа FF. Число FF является зарезервированным служебным полем и в случае его вхождения должно быть заменено на FF00. Анализируя битовый вектор, полученный

по функции $f(x_i) = \begin{cases} 1, x_i = FF \\ 0, x_i \neq FF \end{cases}$ можно сравни-

тельно легко определить наличие или местоположение слова FF. Относительно быстрое получение битового вектора на процессоре NM6403 (8-64 такта в зависимости от разрядности x_i и сложности $f(x)$) позволяет эффективно применять данный механизм для задач поиска элемента в больших массивах.

Другой вариант использования битового вектора применяется в процедуре группового кодирования (RLE). Задача RLE состоит в поиске цепочек одинаковых элементов в некоторой последовательности и замене их на пары <счетчик повторений, значение>, что в результате уменьшает избыточность данных. При обработке изображений ищутся цепочки нулей в блоках по 64 элемента и заменяются на пары <RUN,LEVEL>, где RUN — длина цепочки (от 0 до 15), LEVEL — следующий за цепочкой ненулевой элемент или нуль если длина цепочки превышает 15.

Проиллюстрируем нахождение таких цепочек и их длину на примере.

Допустим, после операций ДКП, квантования и Z-упорядочения образовалась некоторая последовательность X, состоящая, к примеру, преимущественно из нулей, и пусть 11 ненулевых элементов (a,b,c,d,e,f,g,h,i,j,k) находятся на позициях 0,2,4,5,6,13,19,25,26,34,53 соответственно:

X= 0000000000k0000000000000000000j0000000ih00000g00000f000000edc0b0a
 $\times_{63} \dots \dots \times_1 \times_0$

Сформируем битовой вектор по формуле

$$b_i = f(x_i) = \begin{cases} 0, x_i = 0 \\ 1, x_i \neq 0 \end{cases}, i = 0..63$$

B= 00000000001000000000000000000010000000110000010000010000001110101
 $\times_{63} \dots \dots \times_1 \times_0$

Инвертируем битовой вектор: Y[00]= Not(B)

Y[00]= 111111111011111111111111111111101111111001111101111101111110001010

Выполним итеративный цикл от i=1 до 15

$$Y[i] = (Y[i-1] \gg 1) \text{ AND } (Y[i-1])$$

Y[01]= 01111111110111111111111111111110111111100111110111110111111000101
 Y[02]= 00111111110011111111111111111110011111100011110011110011111000000
 Y[03]= 0001111111000111111111111111110001111100001110001110001111000000
 Y[04]= 0000111111000011111111111111110000111100000110000110000111000000
 Y[05]= 0000011111000001111111111111110000011100000010000010000011000000
 Y[06]= 000000111100000011111111111111000000110000000000000000001000000
 Y[07]= 00000001110000000111111111111100000001000000000000000000000000
 Y[08]= 00000000110000000011111111111100000000000000000000000000000000
 Y[09]= 00000000010000000001111111111100000000000000000000000000000000
 Y[10]= 00000000001000000000011111111111000000000000000000000000000000
 Y[11]= 00000000000000000000000111111111110000000000000000000000000000
 Y[12]= 00000000000000000000000011111111110000000000000000000000000000
 Y[13]= 00000000000000000000000001111111110000000000000000000000000000
 Y[14]= 00000000000000000000000000111111110000000000000000000000000000
 Y[15]= 00000000000000000000000000011111110000000000000000000000000000

Просуммируем биты в каждой колонке и получим вектор S из 64 элементов:

$S = 0123456789A0123456789ABCDEFFFF0123456700123450123450123456000101$
 $X = 0000000000k000000000000000000j0000000ih00000g00000f000000edc0b0a$

Каждая полученная сумма s_i показывает количество подряд стоящих нулей слева от числа x_i , а значение суммы, увеличенное на единицу, указывает на относительное положение следующего элемента — LEVEL. Извлекая из вектора S смещения s_i+1 и совершая по ним переходы параллельно в векторе S и X, получим искомый набор <RUN,LEVEL>:

<a,?><b,1><c,1><d,0><e,0><f,6><g,5><h,5>
<i,0><j,7><0,15><a,2><?,10>

Выводы:

Анализ вычисления ДКП на арифметике с фиксированной точкой показал, что 8-разрядное представление коэффициентов C_{ij} по точности практически не уступает ДКП с плавающей точкой — соотношение сигнал/шум восстановленного изображения составляет около 58.2dB (рис. 8), а при разрядностях выше 14 точность достигает предела — PSNR=58.9dB. При той же схеме вычислений комбинирование разных разрядностей коэффициентов [C] и [C^T], например 8 и 12 соответственно, делает отличие от плавающей точки еще меньшим PSNR=58.6dB, а при уровне квантования более 4 оно исчезает полностью. Отличительная особенность прямого способа вычисления ДКП от быстрых алгоритмов с

фиксированной точкой заключается в минимальном количестве этапов умножения, что с одной стороны приводит к меньшему накоплению ошибок, а с другой — позволяет строить вычисления на векторном процессоре.

Продемонстрирована применимость векторного узла процессора NM6403 в задачах кодирования данных. Гибкая программируемая структура векторного сопроцессора и возможность оперировать данными произвольной разрядности позволяет эффективно реализовывать различные схемы работы с битовыми векторами, распараллеливая тем самым сложно векторизуемые алгоритмы.

Литература

1. ИТЦ Модуль «Процессор NM6403. Введение в архитектуру»
2. С. Кун. Матричные процессоры на СБИС. М.: Мир, 1991
3. «Организация параллельных вычислений в алгоритмах БПФ на процессоре NM6403» // Цифровая обработка сигналов, 2001 № 1)
4. «Организация параллельных вычислений преобразования Адамара на процессоре NM6403 (Λ1879BM1)» // Цифровая обработка сигналов, 2001 № 3).

Уважаемые авторы !

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

1) Требования к текстовым материалам и сопроводительным документам:

- Текст — текстовый редактор Microsoft Word на базе версии WINDOWS'95 или выше.
- Все таблицы и рисунки должны быть пронумерованы. На все рисунки, таблицы и библиографические данные должны быть ссылки в тексте статьи.
- Рукопись статьи сопровождается сведениями об авторах (Ф.И.О., организация, телефоны, электронная почта).

2) Требования к иллюстрациям:

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

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

Системы отображения трехмерной графики для авиационных индикаторов

А.В. Бумагин, Е.А. Климов Е.А., В.Б. Стешенко

Введение

В настоящее время актуальна проблема создания электронных систем синтеза и отображения

картин в реальном времени с высокой реалистичностью изображений. Такие системы, в частности, находят применение при построении различного рода индикаторов для кабин летательных аппаратов (ЛА) и тренажеров — в первую очередь, авиационных и космических. Решение этой задачи становится возможным благодаря достижениям микроэлектроники и вычислительной техники.

На бортовых индикаторах в авиационных и космических тренажерах должна воспроизводиться картина, наблюдаемая с борта ЛА во время полета (внешняя обстановка). Условия работы на тренажере должны быть адекватны условиям реального полета при плавном перемещении изображения на экране, без заметных скачков во всем заданном диапазоне скоростей имитируемого полета. Объекты внешней обстановки должны изображаться в перспективной проекции, а их изображения иметь цвет, яркость и текстуру поверхностей, соответствующие натуре. При синтезе изображений должна быть предусмотрена возможность выполнения таких специфических эффектов, как наложение полупрозрачных изображений на основное и сложение изображений (при имитации дымки, тумана и других эффектов).

Удовлетворительная плавность движения изображения достигается при частоте обновления кадров не ниже 25...30 Гц; для достижения реалистичности изображений разрешающая способность монитора должна быть порядка 10^6 пикселей. Снижение требуемого объема оборудования при одновременном повышении производительности системы синтеза изображений может быть достигнуто путем структурной специализации системы в целом (общий алгоритм синтеза) и основных устройств (алгоритмы конкретных задач). Следует отметить, что в специализированных устройствах за счет распараллеливания процесса и конвейерных структур можно за несколько тактов

Приводятся результаты исследований, посвященных созданию высокопроизводительной системы синтеза сложных трехмерных изображений в реальном времени. Проведена разработка идеологии функционирования блока рендеринга на базе графического конвейера; определение и детализация алгоритмов его функционирования с учетом специфики поставленной задачи и применения ПЛИС

выполнить вычисления, на которые компьютеру требуется целая подпрограмма

Графический процессор включает четыре блока (рис. 1): векторный процессор, блок рендеринга (растеризации), HOST-машина, дисплей. На первый блок возложена задача формирования примитивов для последующей прорисовки на основании информации, хранящейся в базе данных. Блок рендеринга производит прорисовку трехмерного изображения и подготавливает его для вывода на дисплей. Управление всей системой производится на базе HOST-машины.

Визуализация трехмерных объектов подразумевает реализацию следующих этапов:

- 1) геометрические преобразования, связанные с проекцией объектов на экран;
- 2) генерирование каркасов объектов;
- 3) вычисление характеристик освещения и отражения поверхностей объектов;
- 4) вычисление цвета и яркости каждой точки изображения с учетом условий освещения и текстуры поверхностей объектов (суммарное количество таких точек с учетом наложений может достигать нескольких миллионов);
- 5) заполнение граней изображений объектов соответствующими текстурами;
- 6) суммирование изображений с целью создания спецэффектов.

Внешняя обстановка включает в себя окружающую местность, искусственные сооружения и подвижные объекты.

Визуализация посредством плоских примитивов

3D-графика реализуется с помощью многоступенчатого механизма, называемого конвейером рендеринга. Ускорение расчетов обеспечивается за счет того, что вычисления для следующего объекта могут быть начаты до окончания вычисления предыдущего.

Конвейер рендеринга может быть разделен на две стадии: геометрическая обработка и растеризация. При геометрической обработке выполняется преобразование координат, отсечение невидимых частей объектов, расчет освещения, определение цвета каждой вершины и

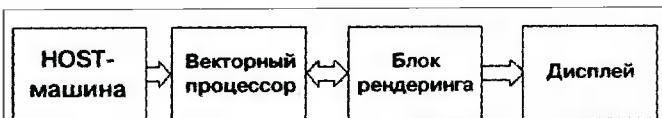


Рис. 1. Структурная схема графического процессора

деление изображения на более мелкие формы. Для описания характера поверхности объекта ее делят на многоугольники различного типа, чаще всего — на треугольники, учитывая простоту вычислений и легкость манипулирования. Для ускорения вычислений координаты объектов переводятся из вещественного в целочисленное представление.

На второй стадии применяется следующая последовательность операций: удаление скрытых поверхностей, наложение с учетом перспективы текстур (используя z-буфер), применение эффектов тумана и полупрозрачности. После этого очередная точка считается готовой к помещению в буфер со следующего кадра. В памяти, установленной на плате 3D-акселератора, хранятся текстуры, Z-буфер и буферы следующего кадра. При использовании шины PCI обычная оперативная память неприменима, так как быстродействие видеокарты существенно ограничено пропускной способностью шины.

Элементы графического конвейера представлены на рис. 2. Специфика задачи разработки блока формирования кадров изображения обусловлена достаточно жесткими требованиями к видеопроцессору (частота обновления изображения 75 Гц при экранном разрешении 1024x1024 пиксела). Реализуемые аппаратно алгоритмы должны обеспечивать быстрое формирование

изображения в буфере кадра при минимальных вычислительных затратах. Отсюда — большие объемы дорогостоящей видеопамати, отводимой для промежуточного и окончательного хранения изображения и пространственных параметров примитивов.

Входными данными для блока формирования кадров изображения являются трехмерные координаты вершин графических примитивов, предназначенных для растеризации. Под основным графическим примитивом будем подразумевать треугольник (полигон). Прорисовка полигонов организуется при помощи быстрых аппаратных алгоритмов, которые универсальны и легко распараллеливаются. Кроме координат вершин полигонов в блок кадрирования из геометрического вычислителя поступают данные о цвете каждой вершины (RGB), её глубине (расстоянии до плоскости наблюдения, Z-канал), текстуре и прозрачности (B-канал).

Метод растеризации, основанный на разбиении исходной сцены на треугольники, дает существенный выигрыш в быстродействии по сравнению с векторными методами по следующим причинам:

- треугольники легко «просчитываются»;
- растеризация одинаковых треугольников легко реализуется аппаратно;
- легко организовать быструю заливку;
- при усложнении сцены производительность графической системы растет линейно, пропорционально количеству примитивов, в то время как у векторных методов эта зависимость является квадратичной.

Следует также отметить, что при не очень сложной сцене (порядка нескольких десятков тысяч полигонов) векторные методы дают выигрыш в быстродействии, который однако существенно падает при усложнении сцены.

Геометрические представления и пересчет координат объектов

Пересчет координат — важный этап генерации трехмерных изображений. Преобразования координат точек объектов, хранящиеся в базе данных на host-машине, в плоскостные координаты и глубины вершин элементарных фэйсов (треугольников) производится геометрическим процессором на базе вычислительного модуля НМЦ4.01.

Для пересчета координат из системы X, Y, Z в систему необходимо вычислить матрицу поворота:

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$

где

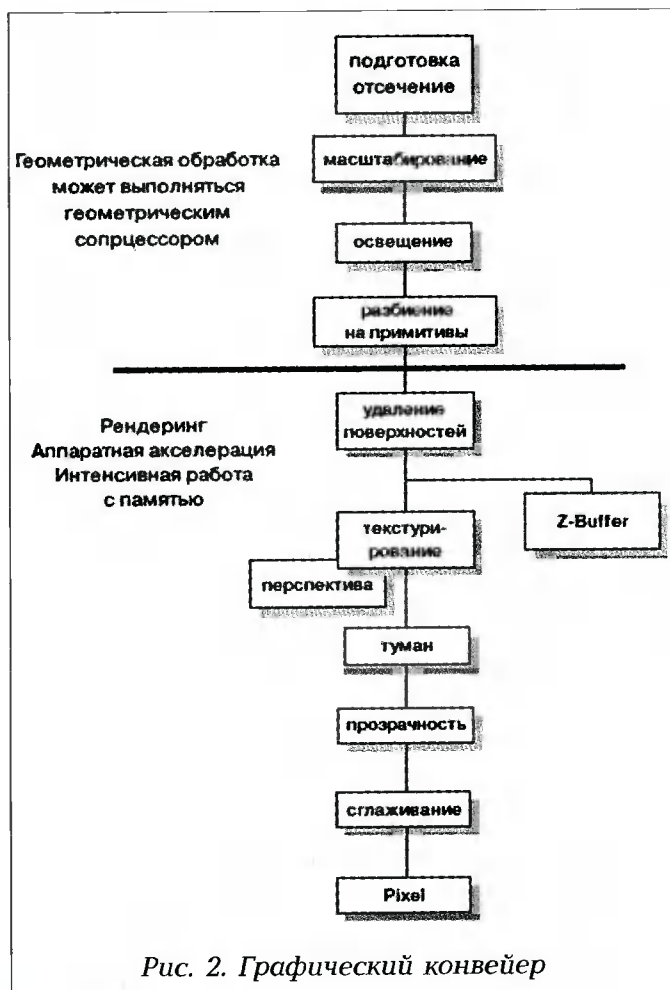


Рис. 2. Графический конвейер

$$\begin{aligned}
 a_{11} &= \cos\vartheta \cdot \cos\varphi; & a_{21} &= -\sin\vartheta \cdot \cos\varphi + \\
 &+ \cos\vartheta \cdot \sin\varphi \cdot \sin\gamma; & a_{31} &= \sin\vartheta \cdot \sin\gamma + \\
 &+ \cos\vartheta \cdot \sin\varphi \cdot \cos\gamma; & a_{12} &= \sin\vartheta \cdot \cos\varphi; \\
 a_{22} &= \cos\vartheta \cdot \cos\varphi + \sin\vartheta \cdot \sin\varphi \cdot \sin\gamma; \\
 a_{32} &= -\cos\vartheta \cdot \cos\varphi; & a_{13} &= -\sin\vartheta; \\
 a_{23} &= \cos\vartheta \cdot \sin\gamma; & a_{33} &= \cos\vartheta \cdot \cos\gamma.
 \end{aligned}$$

Координаты точки в повернутой системе вычисляются по формулам:

$$\begin{aligned}
 X_{1t} &= a_{11} \cdot X + a_{21} \cdot Y + a_{31} \cdot Z + X_{1t} \\
 Y_{1t} &= a_{12} \cdot X + a_{22} \cdot Y + a_{32} \cdot Z + Y_{1t} \\
 Z_{1t} &= a_{13} \cdot X + a_{23} \cdot Y + a_{33} \cdot Z + Z_{1t}
 \end{aligned}$$

где X_{1t} , Y_{1t} , Z_{1t} — повернутые координаты наблюдателя:

$$\begin{aligned}
 X_{1t} &= a_{11}X_t + a_{12}Y_t + a_{13}Z_t \\
 Y_{1t} &= a_{21}X_t + a_{22}Y_t + a_{23}Z_t \\
 Z_{1t} &= a_{31}X_t + a_{32}Y_t + a_{33}Z_t
 \end{aligned}$$

Индексом t отмечены текущие координаты транспортного средства, на котором находится наблюдатель. Координаты точки в экранной системе координат вычисляются по формулам:

$$X_0 = m \frac{Z_t}{X_t}; Y_0 = m \frac{Y_t}{X_t}, \text{ где } m = \text{ctg} \frac{\beta}{2}; \beta - \text{угловое}$$

поле зрения системы.

Для формирования изображения помимо информации, хранящейся в базе данных, необходимо в каждый момент времени знать положение наблюдателя относительно местности, а также расположение подвижных объектов. Эта информация поступает в систему синтеза из динамических моделей подвижных объектов и летательного аппарата.

Поскольку синтезированное изображение должно адекватно отображать картину, которую увидел бы наблюдатель со своего рабочего места через окно, равное экрану, то, очевидно, сначала необходимо определить какие объекты находятся в данный момент в поле зрения наблюдателя. Затем часть из этих объектов следует удалить, если их изображение окажется соизмеримым с разрешением экрана.

Прежде чем определить требования по производительности, объемам памяти и быстродействию, предъявляемым к векторному процессору, необходимо произвести оценку машинного представления элементов 3D-геометрии. После этого можно судить о возможностях предположительного быстродействия разрабатываемого графического акселератора.

Модель должна обеспечивать хорошее правдоподобие по отношению к реальности при требовании малополигональности геометрии. Повы-

шенные требования при разработке налагаемых текстур, которые восполняют дефекты геометрии и обеспечивают эффективное воздействие на наблюдателя при условии высокоскоростных перемещений движка по виртуальному пространству. Широко используются объемные и 2D-спецефффекты. Динамика графической системы и динамика системы управления должны соответствовать реальным уравнениям, описывающим движения объектов создаваемого мира. Накладываются жесткие требования на скорость прорисовки кадров при экранных разрешениях 1024×1024 и выше, поскольку предполагается использование системы при работе в реальном времени. Графическая система применима при разработке профессиональных симуляторов и тренажеров.

Рассмотрим кратко базовые элементы 3D-геометрии.

Точка (vertex) задается тремя числами x , y , z (типа float или double). Точки отдельно не существуют, а используются для представления других 3D-объектов. При выполнении векторно-матричных операций точка представляется в виде вектора $P = (x, y, z, 1)$. Суммарная длина описания одной точки составляет 12 байт.

Нормаль (normal), задается тремя числами lx , ly , lz (типа float или double). С помощью нормалей описываются вершины объектов типа 3D-элемент (face). Нормали необходимы при решении задач видимости, освещения и цвета. Суммарная длина описания одной нормали составляет 12 байт.

3D-элемент (face), задается тремя точками (vertex), тремя нормальями (normal) в точках типа (float), цветом (long) (рис. 3). Суммарная длина описания одного 3D-элемента составляет 76 байт. При задании точек при помощи указателей на существующие точки и нормали длина 3D-элемента значительно меньше и составляет 25 байт.

Геометрические объекты и их векторные представления находятся в устройстве хранения в виде, удобном для представления в соответствии со способом формирования на базе типовых 3D-элементов и 3D-групп. При запуске моделирующего комплекса происходит загрузка модельных массивов в оперативную память, и возникает

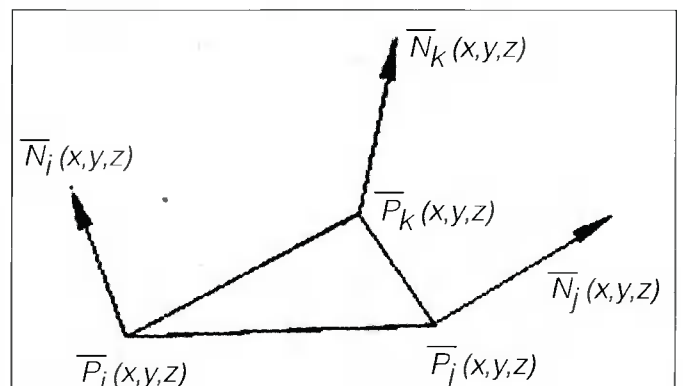


Рис. 3. Базовый примитив 3D-элемент (face)

проблема, как представлять эти массивы в оперативной памяти при выполнении симуляции.

Рендеринг

Рассмотрим алгоритмы, реализованные в блоке рендеринга.

Представление цвета и спецэффектов. В соответствии с трехкомпонентной теорией зрения цвет излучения определяется соотношением интенсивностей (яркостей) трех цветовых составляющих, а его яркость равна сумме яркостей этих составляющих $B = B_r + B_g + B_b$. Поскольку три цветовые составляющие одновременно определяют как яркость, так и цвет излучения, в технических устройствах для задания цвета используют приведенные значения этих составляющих:

$$r_0 = \frac{r}{r+g+b}; \quad g_0 = \frac{g}{r+g+b}; \quad b_0 = \frac{b}{r+g+b}.$$

Их физический смысл — в том, что они соответствуют конкретному цвету при значении общей яркости, равной 1. Для вычисления цветовых составляющих при произвольной освещенности достаточно значения составляющих умножить на полную яркость поверхности. Однако с целью сокращения вычислений, целесообразно значения приведенных составляющих уже при подготовке исходных данных умножать на полный коэффициент отражения поверхности:

$$r_s = r_0 \rho; \quad g_s = g_0 \rho; \quad b_s = b_0 \rho$$

Тогда будет сокращено количество операций на три умножения для каждой точки:

$$r = r_s EK; \quad g = g_s EK; \quad b = b_s EK$$

Если объект освещается несколькими источниками с одинаковым спектром, то при вычислениях используется та же формула, необходимо только просуммировать освещенности и коэффициенты яркости. При различных спектрах источников света все вычисления повторяются для каждого источника, а результаты суммируются. В этом случае для каждого источника необходимо предусмотреть индивидуальные исходные данные в базе данных.

Для получения различных спецэффектов (имитация полупрозрачности, тумана, тени и др.) возникает необходимость сложения (объединения) изображений. Возможны следующие варианты: замена (вытеснение) одного изображения другим и полное сложение. Общее изображение имеет суммарный цвет и суммарную яркость:

$$r_\sigma = r_1 + r_2; \quad g_\sigma = g_1 + g_2; \quad b_\sigma = b_1 + b_2$$

Эти два способа используются наиболее широко, так как не требуют сложных вычислений и могут быть применены одновременно при фор-

мировании различных частей одного общего изображения.

Интерполяция линий. Для интерполяции отрезков применен широко известный алгоритм Брезенхема, который легко реализуется аппаратно, не требует применения операций с плавающей точкой, не содержит операций деления. Операции умножения на 2, содержащиеся в алгоритме, сводятся к арифметическому сдвигу. В рамках архитектуры ПЛИС можно организовывать несколько параллельных задач прорисовки ребер треугольников, т.е. интерполировать три ребра одновременно.

Интерполяция цветов и глубин. Как отмечено выше, параллельно с прорисовкой ребер треугольников осуществляется интерполяция цветов и глубин их точек, которая с алгоритмической точки зрения полностью идентична, и для ее реализации применяются одинаковые устройства (интерполяторы). Поэтому эти задачи рассматриваются как одно целое. Затенение по Гуро — это метод линейной интерполяции освещенности в пределах одного полигона. В отличие от равномерного заполнения, в случае метода Гуро необходимо прежде всего определить и просчитать одним из известных способов приведенный вектор нормали для каждой вершины полигона. Получив нормаль, можно определить степень затенения или освещенность в вершинах, а далее — провести интерполяцию освещенности по полигону. При реализации затенения Гуро необходимо ограничиться использованием только треугольных полигонов.

Заливка примитивов. Процесс заливки примитивов — основной этап визуализации объектов; он наиболее трудоемок. Известно несколько методов заливки замкнутых областей, однако большинство из них при выводе очередного пиксела подразумевают сканирование соседних (смежных) точек с целью достижения границы и фиксированием информации о них (например, в стеке). К ним относится метод заливки с затравкой, являющийся универсальным. Если требуется заливать не произвольную замкнутую фигуру, а треугольник, то последний можно разбить на линии уровня с постоянными координатами Y и производить заливку вдоль этих линий (строчек). Таким образом, для заливки каждой строчки достаточно знать X-координаты ее границ, а промежуточные точки будут вычислены в одном цикле, число итераций которого соответствует числу промежуточных точек. В том же цикле вычисляются цвета и глубины.

Данный подход существенно убыстряет процесс заливки. Если для X-координат, цвета и глубины трех ребер треугольника сформировать независимые буферы памяти (буферы ребер), адресуемые по координате Y, то выбор данных очередной пары точек ребра для заливки сводится к простому извлечению последних из промежуточ-

ных буферов с одинаковым адресом (Y). Такой подход позволяет избежать сканирования памяти, повышая скорость заливки, и легко организуется аппаратно.

Интерполяция цветов и глубин промежуточных точек треугольников осуществляется параллельно и аналогично интерполяции цветов и глубин для ребер. Пары граничных точек из трех ребер формируются специализированными устройствами — блоками принятия решения и мультиплексирования.

В целом процесс заливки разбивается на следующие этапы:

- извлечение данных граничных точек ребер из промежуточных буферов;
- вычисление X -координаты очередной внутренней точки;
- интерполяция глубины и цвета;
- считывание информации о глубине и цвете точки с данными координатами, соответственно, из Z -буфера и буфера кадра;
- принятие решения о прорисовке данного пиксела;
- вычисление цвета пиксела с учетом фактора прозрачности;
- занесение новых данных в Z -буфер и буфер кадра.

В случае принятия решения о непрорисовке пиксела последние два пункта опускаются. Информация о координатах, глубинах и цветах заливщику передается по параллельным шинам. Для увеличения быстродействия предусмотрена возможность заглубления заливки, которое подразумевает извлечение данных о глубине не каждого пиксела, а, например, через один, сокращая число операций чтения-записи в память. Данный подход целесообразен при большом экранном разрешении.

Удаление невидимых граней. Z -буферизация. Во время заливки необходимо иметь информацию и о взаимном расположении объектов. Для решения этой задачи применяется Z -буферизация, самый надежный метод удаления скрытых поверхностей. Z -буферизация заметно увеличивает производительность, однако Z -буфер занимает большие объемы памяти: например даже при разрешении 640×480 24-разрядный Z -буфер будет занимать около 900 Кб. Эта память должна быть также установлена на 3D-видеокарте. Разрешающая способность Z -буфера — главный его атрибут. Она критична для высококачественного отображения сцен с большой глубиной и чем она выше, тем точнее выполняется рендеринг удаленных объектов. Если при рендеринге разрешающей способности недостаточно, то два перекрывающихся объекта могут получить одну и ту же координату Z , что приводит к искажению изображения. Во избежание этих эффектов мощные графические процессоры имеют 32-разрядный Z -буфер и оборудуются

большими объемами памяти.

Вывод на экран. Видеоконтроллер (RAMDAC) предназначен для вывода изображения, сформированного в буфере кадра, на экран. На выходе микроконтроллера формируются RGB-сигналы для видеомонитора. Требования к видеоконтроллеру — обеспечение вывода на монитор изображения разрешением 1024×1024 пиксела с частотой обновления 75Гц. Цветовое разрешение — 32 бит. RAMDAC во всех других графических чипсетах имеет два режима работы. В первом режиме чипсет оперирует данными цветовой гаммы или палитры. В этом режиме 8-битные данные конвертируются в RGB цвета. Каждому из 256 возможных значений цвета соответствует положение в цветовой палитре, которая размещается в DAC (ЦАП = цифро-аналоговый преобразователь). Цветовая палитра формируется и хранится в RAM (память с произвольной выборкой) — отсюда и название RAMDAC — и может быть загружена с любой комбинацией цветов.

Реализация геометрического процессора на базе вычислительного модуля МЦ4.01

Геометрический процессор необходим для выполнения векторно-матричных операций, используемых при построении дисплейных изображений. Он строится на базе серийно выпускаемого двухпроцессорного модуля МЦ4.01.

На рис. 4 приведена структурная схема геометрического процессора в составе графической системы.

Плата геометрического процессора установлена в PCI-слоте персональной ЭВМ (HOST-машине), которая используется для формирования процессов симуляции виртуального мира.

Для вычислительных устройств на базе процессора Л1879ВМ1 (NM6403) используется механизм взаимодействия через разделяемую память. HOST-машина имеет доступ к разделяемой памяти вычислительного устройства. Через разделяемую память происходит загрузка программ, а также обмен массивами данных и синхронизация программ, работающих на HOST-машине и на процессорах вычислительного устройства. Из рис. 4 видно, что через PCI контроллер на плате HOST-машина может обращаться по чтению в разделяемую память модуля МЦ4.01, которая представлена двумя банками SRAM памяти по 512 Кбайт, расположенными на глобальной шине процессоров Л1879ВМ1. Другие блоки памяти вычислителя для HOST-машины недоступны. Модуль включает в себя два процессора Л1879ВМ1, работающих на частоте 40 МГц. Процессоры практически равноправны и окружены одинаковыми наборами периферийных устройств. Каждый из процессоров имеет следующее окружение:

- 32 Мб динамической памяти со временем

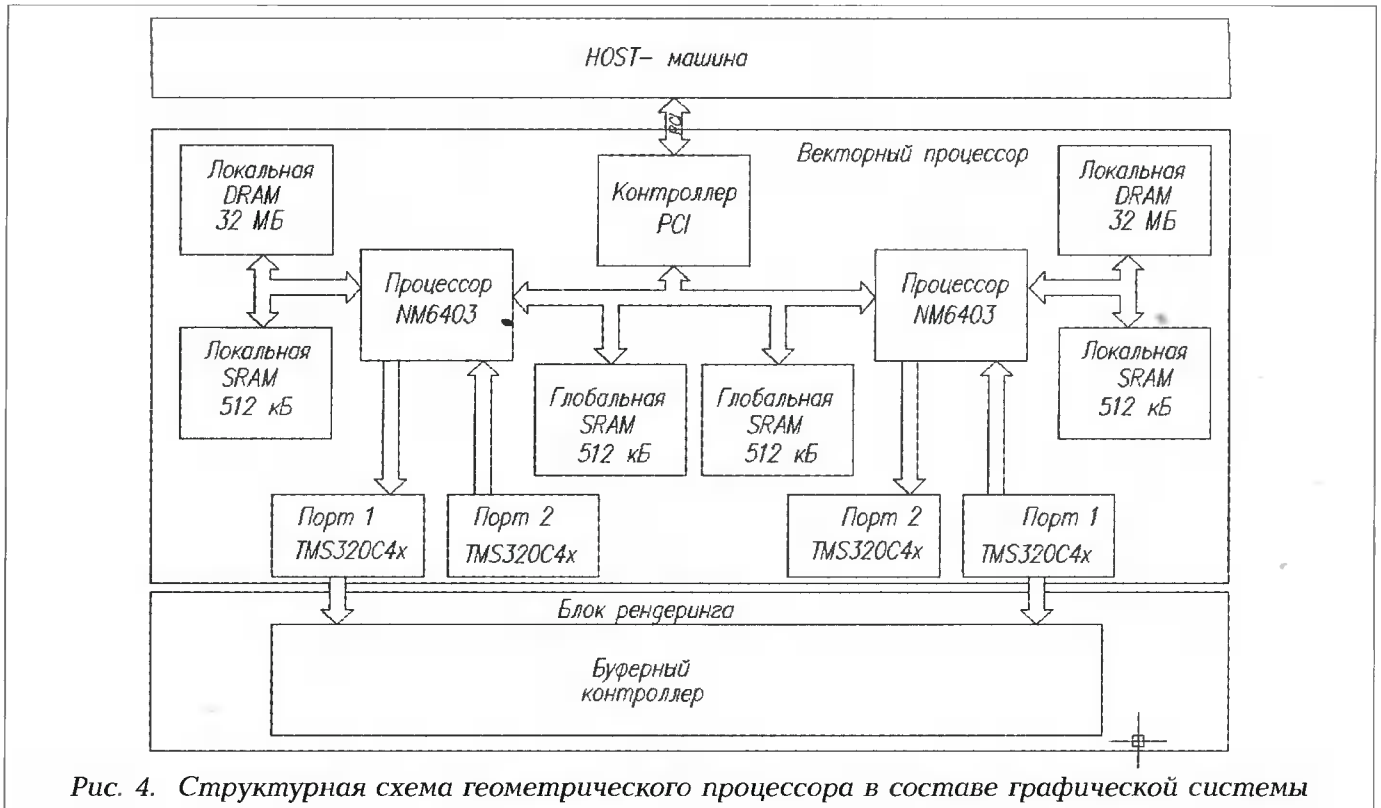


Рис. 4. Структурная схема геометрического процессора в составе графической системы

- доступа — 3 такта (75 нс);
- 512 Кбайт статической памяти со временем доступа — 1 такт (25 нс);
- 512 Кбайт статической памяти на глобальной шине с доступом — 1 такт (25 нс);
- 512 Кбайт статической памяти банк на глобальной шине с доступом — 2 такта (50 нс);
- 2 коммуникационных порта аппаратно совместимых с процессорами серии TMS320C4x.

Программа для выполнения операций преобразования над геометрическими объектами.

Для функционирования геометрического процессора было разработано программное обеспечение, реализующее базовые алгоритмы трехмерной геометрии.

Программа предназначена для выполнения операций преобразования над большим количеством геометрических объектов (несколько сот тысяч), представленных треугольниками; специфических операций для трехмерных ускорителей таких, как умножение координат на матрицу, расчет цвета в зависимости от направления нормали фэйса (обрабатываемого треугольника) по отношению к источнику освещения и материала объекта, характеризующего собственную светимость тела и изменение цвета тела под действием источника освещения.

Работа программы заключается в загрузке файла специально разработанного формата (SMF) в DRAM-память процессора L1879BM1, выполнении геометрических преобразований над объектом (представленном сеткой) таких, как повороты вокруг осей в зависимости от действий

пользователя. Управление объектом (вращение вокруг осей) осуществляется путем нажатия клавиш управления курсором (UP, DOWN, LEFT, RIGHT), а также путем перемещения манипулятора типа «мышь» с нажатой левой клавишей. Режим визуализации (объект представлен сеткой или закрасен) меняется нажатием клавиши «w».

Результаты работы программ представлены на рис. 5 и рис. 6.

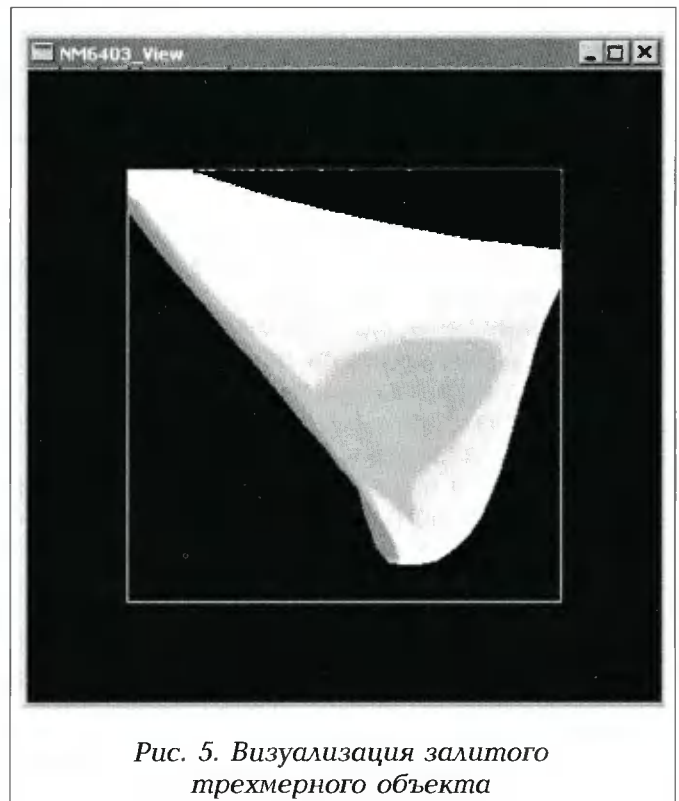


Рис. 5. Визуализация залитого трехмерного объекта

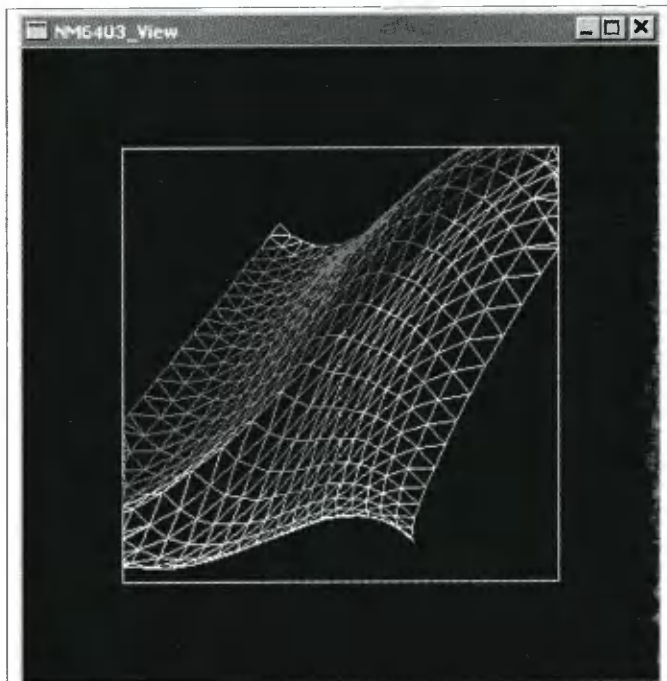


Рис. 6. Визуализация сетки трехмерного объекта

ника логически можно разделить на два этапа: прорисовка ребер и заполнение внутреннего пространства треугольника. Для совмещения во времени этих процессов был применен двухсторонний промежуточный буфер для хранения координат точек ребер, их глубин и цветов. Таким образом, пока прорисовываются границы одного треугольника, их цвет и глубины, производится заливка другого треугольника, границы которого прорисовывались ранее. Такой подход позволяет избежать простоя в прорисовке границ во время заливки.

Блок рендеринга включает в себя: буферный контроллер, блок прорисовки границ, интерполяторы глубин, интерполяторы цвета, блок заливки, RAM DAC, буферы ребер, цветов, глубин; Z-буфер, VIDEO RAM. Входными для блока рендеринга являются данные, получаемые с TMS портов геометрического процессора (8 разрядов — данные, 4 разряда — шина управления), а выходными — компонентный RGB видеосигнал. Для упрощения согласования работы этих устройств и повышения производительности обмен данными между устройствами производится с использованием двухсторонней буферной памяти. Структурная схема блока рендеринга приведена на рис. 7.

Рассматриваемая графическая система работает на основе стандартного графического контроллера, рассмотренного выше. Однако из-за ог-

Реализация блока рендеринга на базе ПЛИС

Для построения блока рендеринга с целью повышения производительности была выбрана следующая методика. Процесс прорисовки треуголь-

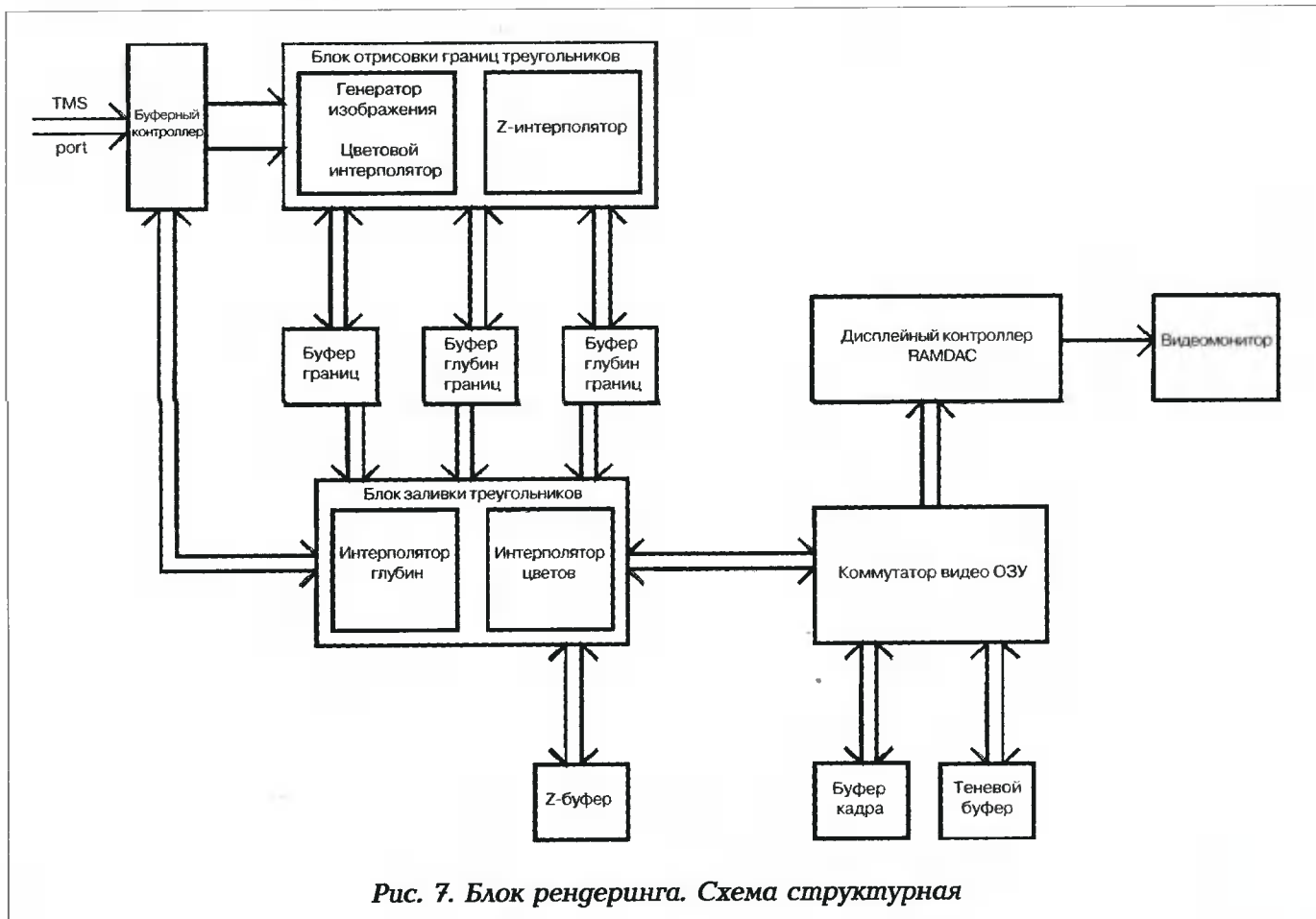


Рис. 7. Блок рендеринга. Схема структурная

раниченного быстродействия ПЛИС есть некоторая специфика реализации различных участков. Очевидно, что для получения максимального быстродействия графического процессора данные между устройствами блока рендеринга должны передаваться по независимым параллельным шинам высокой (до 96 бит) разрядности. Так, координаты X, Y вершин треугольников, Z -глубины и цвета передаются по параллельным шинам. Поэтому для большинства элементов блока рендеринга необходим выбор ПЛИС с достаточно большим числом выводов, но с учетом способа монтажа микросхем и их стоимости.

Поскольку увеличение разрядности (и числа) независимых магистралей ограничено числом выводов ПЛИС, было принято решение ввести мультиплексирование шин. Шины мультиплексированы таким образом, что за один такт работы шины по ней передается информация об одной точке треугольника.

Обычно при прорисовке треугольников используются последовательно следующие операции: прорисовка границ (ребер треугольников); заливка, с учетом информации о взаимном перекрытии примитивов и затенении (?-смещение и Z -буферизация); вывод изображения на экран.

Этот подход требует последовательного вычисления Z - и цветовой (Гуро) билинейной интерполяции для каждого пиксела. В случае последовательного выполнения перечисленных операций общее время вычислений будет равно сумме времен, затраченных на каждый этап. В связи с этим предлагается совместить во времени операции прорисовки границ, глубин и цветов точек границ треугольников с помещением их в промежуточные буферы в виде последовательных списков. При этом резко упрощается методика заливки: исключается промежуточная трассировка треугольников, а вычисление Z -составляющей и цвета каждого пикселя сводится к простым инкрементным методам.

Однако даже совмещение во времени этих операций не позволяет получить максимальную производительность, потенциально доступную при использовании микросхем программируемой логики и определяемую пропускной способностью памяти. Так как некоторые устройства не могут работать так быстро, вместо одного блока было использовано несколько идентичных со сдвинутыми по времени периодами обращения к памяти с применением FIFO-буферов.

Рассмотрим кратко составляющие блока рендеринга.

Буферный контроллер предназначен для сопряжения блока рендеринга с вычислительным модулем МЦ4.01. В его задачи входит: прием данных от модуля МЦ4.01 по протоколу коммуникационного порта Л1879ВМ; буферизирование полученных данных; подготовка данных к обработке; управление работой блока рендеринга. В

состав буферного контроллера входят следующие узлы: блок аппаратной эмуляции коммуникационного порта; блок выделения признаков; буфер с FIFO организацией; блок мультиплексирования; синхронизатор. Входными для блока рендеринга являются сигналы порта (8 разрядов — данные, 4 разряда — шина управления), а выходом — 96-битная мультиплексированная шина ребер. Шина ребер мультиплексирована таким образом, что за один цикл работы шины по ней передается информация об одном ребре треугольника. Цикл работы мультиплексированной шины разбит на два такта. За один такт цикла шины передается информация об одной вершине треугольника.

Блок прорисовки границ предназначен для формирования в буферах ребер, глубин границ и цветов границ данных, соответственно, о положении на экране, расстоянии до картинной плоскости и цвете ребер текущих обрабатываемых полигонов. Эта информация необходима для быстрой заливки треугольников естественной цветовой гаммой с учетом фактора прозрачности и сортировки их по дальности. Данные помещаются в виде списков. Для формирования границ применяется целочисленный алгоритм Брезенхемма. Вычисление глубин граничных точек осуществляется при помощи билинейной интерполяции, закраска — методом Гуро.

Блок заливки предназначен для быстрой заливки треугольников с интерполяцией цветов внутренних пикселов с учетом фактора прозрачности, а также для быстрого решения задачи удаления невидимых граней. Последняя осуществляется при помощи тотальной Z -буферизации примитивов. Треугольники обрабатываются последовательно по мере поступления информации из геометрического вычислителя. Не подлежащие заливке (меньше определенного порога по линейным размерам) треугольники аппроксимируются одним пикселем, расположенным в центре тяжести треугольника и имеющим цвет, равный среднему арифметическому цветов трех вершин треугольника. Производится аналогичная сортировка по глубине.

Z -буфер — быстродействующая память, предназначенная для хранения расстояния каждой точки видимого изображения до плоскости проецирования. Служит основным инструментом для сортировки объектов по дальности и фильтрации отдаленных объектов.

Буфер кадра — быстродействующее ОЗУ, содержащее цветовую информацию о всех пикселах, выводимых на дисплей в пределах одного кадра. В пределах этой памяти происходит прорисовка всех примитивов с учетом их сортировки по дальности. Дублирующий буфер кадра (теневой буфер) по функциям аналогичен основному кадровому буферу. Применяется для устранения эффекта мерцания экрана при переключении кадров.

Коммутатор видео ОЗУ производит переключение рабочего адресного пространства между основным буфером кадра и теневым.

Дисплейный контроллер содержит АЦП для каждого компонента цветовой палитры и производит вывод содержимого буфера кадра на экран дисплея в реальном времени.

Заключение

Основным результатом работы является создание идеологии методики функционирования блока формирования кадров изображения и векторного процессора визуализации объектов.

Подробно изложены методики визуализации сложных трехмерных объектов. Выбран и обоснован метод визуализации на базе плоских примитивов посредством графического конвейера. Рассмотрены также математические аспекты геометрических векторных преобразований.

Разработаны алгоритмы функционирования всех составляющих блока формирования кадров изображения с учетом аппаратной реализации и максимального быстродействия системы. Разработана функциональная схема блока, определена размерность шин и объемы элементов памяти.

Произведен подбор элементной базы для реализации блока формирования кадров изображения и векторного процессора с учетом максимально-го использования вычислительных мощностей.

Произведено моделирование элементов блока рендеринга для последующего программирования ПЛИС. Разработана методика и программное обеспечение для функционирования векторного вычислителя и блока отсечений на базе процессора L1879BM1 (NM6403).

Литература

1. Гилой В. *Интерактивная машинная графика: структуры данных, алгоритмы, языки: пер. с англ.* — М.: Мир, 1981. — 384 с.: ил.
2. Пухальский Г. И., Новосельцева Т. Я. *Проектирование дискретных устройств на интегральных микросхемах: Справ.* — М.: Радио и связь, 1990. — 304 с.: ил.
3. Роджерс Р. *Алгоритмические основы машинной графики: пер. с англ.* — М.: Мир, 1989. — 512 с.: ил.
4. Стешенко В. Б. *ПЛИС фирмы Altera: проектирование устройств обработки сигналов.* — М.: ДОДЭКА, 2000. — 128 с.: ил.

И Н Ф О Р М А Ц И Я

4-я Международная научно-техническая конференция и выставка «Цифровая обработка сигналов и её применение — DSPA'2002»

Отвечая целям и задачам консолидации и поддержки отечественных ученых и специалистов, работающих в области передовых информационных технологий реального времени, путём создания условий для объединения интеллектуального потенциала российских ученых, формирования отечественной школы специалистов по ЦОС, расширения международных научных связей и содействия внедрению высоких наукоёмких технологий XXI века, Российское НТО-РЭС 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 и Федеральной целевой программы «Интеграция» провели в г. Москве 27 февраля — 1 марта 2002 года 4-ю Международную научно-техническую конференцию и выставку «Цифровая обработка сигналов и её применение — DSPA'2002», собравших более тысячи участников из 10 стран мира и 25 городов России. Было заслушано 164 доклада, по материалам которых изданы Труды Конференции в 2-х томах на русском и английском языках общим объёмом — 478 стр.

Работа Конференции проходила по девяти секциям: теория сигналов и систем; теория и методы цифровой обработки сигналов; обработка и передача речевых и звуковых сигналов; обработка сигналов в радиотехнических системах; обработка и передача изображений; обработка и передача измерительной информации; цифровое радиовещание; цифровая обработка многомерных сигналов; нейрокомпьютерная обработка сигналов и изображений.

В обсуждении актуальных проблем цифровой обработки сигналов в рамках Пленарного заседания и круглого стола в первый день работы Конференции приняли участие: академик РАН Ю.В. Гуляев (директор ИРЭ РАН), академик РАН Н.А. Кузнецов (директор ИППИ РАН), академик АН Грузии И.В. Прангишвили (директор ИПУ РАН), академик РАН А.С. Бугаев, профессор В.В. Шахильдян (ректор МТУСИ) и др.

Одновременно с Конференцией была организована Выставка новых информационных технологий реального времени и инструментальных средств ЦОС. В работе Выставки, развернувшейся на 23 стендах, приняли участие ведущие в этой области отечественные предприятия и фирмы, в их числе: АО «АВТЭК», ИПУ РАН, АОЗТ «Инструментальные системы», ЗАО «СКАН Инжиниринг-телеком», ЗАО «СКАНТИ-Рус», ЗАО «РТСофт», «НИИ Автоматики» ЗАО «РАСТР», ООО «МАРАФОН», ООО НВФ «КРИПТОН» и др.

В дни работы Конференции и Выставки прошли семинары-презентации по новым разработкам в области DSP компаний «Analog Devices», (организатор АО «АВТЭК») и «Texas Instruments» (организаторы ЗАО «СКАНТИ-Рус» и «СКАН Инжиниринг-телеком»), а также фирмы ООО «Техноком-АТ» (Украина, Киев). С повышенным интересом участников семинара была встречена информация о новых модулях высокопроизводительной обработки сигналов на базе DSP компании «Texas Instruments» и ПЛИС фирмы «Xilinx».

Цифровая обработка сигналов в многофункциональных РЛС

Д. Ю. Бобров, А. П. Доброжанский, Г. В. Зайцев, Ю. В. Маликов, И. Б. Цыпин

ЧАСТЬ 2: АЛГОРИТМЫ ОБРАБОТКИ РАДИОЛОКАЦИОННЫХ СИГНАЛОВ

1. Введение

Настоящая статья является второй частью работы, посвященной методам обработки сигналов в современных МРЛС.

В первой части [1] рассмотрены общие принципы построения устройств обработки радиолокационных сигналов и методы преобразования сигналов в цифровую форму. В данной статье описываются основные алгоритмы обработки радиолокационных сигналов. Их реализация будет обсуждаться в третьей части.

Напомним, что рассматриваемое в работе устройство обработки сигналов (УОС) реализует все операции по частотно-временной обработке сигнала, которые можно реализовать в одном зондировании, а именно: оптимальную многоканальную частотно-временную фильтрацию сигналов, предварительное обнаружение полезных сигналов по результатам одного зондирования, измерение координат, соответствующих обнаруженным сигналам, вычисление сигналов ошибок по угловым координатам, дальности и скорости для сопровождаемых целей [1].

Алгоритмы обработки радиолокационных сигналов являются первой и наиболее важной составляющей триады: алгоритмы — архитектура процессора — элементная база, представляющей собой фундамент цифровой обработки. Другие две составляющие определяются с учетом особенностей класса выполняемых алгоритмов. Все алгоритмы обработки сигналов в МРЛС можно условно разбить на когерентную обработку (оптимальную фильтрацию) и некогерентную обработку (обнаружение и измерение координат).

2. Алгоритмы цифровой оптимальной фильтрации сигналов

В данном разделе без доказательства приводятся базовые алгоритмы цифровой многоканальной фильтрации. Подробности могут быть найдены в работе [2].

В типичной радиолокационной задаче требуется сформировать набор оптимальных фильтров, перекрывающих заданный диапазон задержек и доплеровских частот отраженных сигналов [3,4,5]. В данной статье каждый такой фильтр для фиксированных значений задержки и частоты будем называть каналом фильтрации, а двумерную матрицу выходных значений фильтров — матрицей дальность-скорость. Формат матрицы опре-

Рассматриваются алгоритмы цифровой обработки сигналов для современных многофункциональных радиолокационных станций (МРЛС). Приводятся основные алгоритмы цифровой обработки радиолокационных сигналов, сравнение их вычислительной эффективности и примеры обработки типовых сигналов

деляется требуемым диапазоном поиска цели по дальности и скорости.

Будем считать также, что на обработку поступают комплексные числа с частотой дискретизации F_D , сформированные на этапе преобразования сигнала в цифровую форму [1]. Обозначим $x(i)$ — i -ю выборку принятого сигнала, $i = 1, 2, \dots$.

Простейшим алгоритмом оптимальной фильтрации является алгоритм **корреляционной обработки**. Он реализуется в соответствии с формулой

$$y(i_0, F) = \sum_{i=i_0}^{n+i_0-1} x(i) \cdot a^*(i-i_0) \cdot \exp\left(-2\pi j \frac{(f_0 + F)}{F_D} \cdot (i-i_0)\right) \quad (1)$$

где $a(i)$ — комплексная огибающая зондирующего сигнала,

f_0 — несущая частота сигнала,

F — доплеровский частотный сдвиг принятого сигнала,

i_0 — задержка принятого сигнала,

n — число выборок на длительности зондирующего сигнала.

Формула (1) реализует один канал фильтрации для фиксированных значений задержки и доплеровской частоты. Блок-схема, поясняющая алгоритм, представлена на рис. 1.

Этапы корреляционной обработки имеют следующий физический смысл:

- перемножение с комплексно-сопряженной огибающей зондирующего сигнала, задержанной на время распространения входного сигнала (первый множитель на рис. 1);

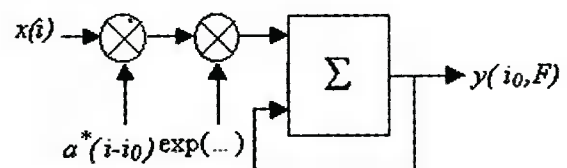


Рис. 1. Блок-схема корреляционной обработки

- «компенсация» несущей частоты и доплеровского сдвига сигнала (второй множитель на рис. 1);

- накопление выборок полученного сигнала.

Эффективность алгоритмов фильтрации удобно характеризовать количеством арифметических операций, необходимых для вычисления одного канала фильтрации. Этот параметр будем называть удельным объемом вычислений:

$$v = \frac{V}{M}, \quad (2)$$

где V — число арифметических операций алгоритма, а M — число фильтровых каналов, реализуемых с помощью этого алгоритма..

Из (1) можно подсчитать, что для корреляционной обработки удельный объем вычислений равен:

$$v_{КО} = 14n - 2 \quad (3)$$

Алгоритм **корреляционно-фильтровой обработки** является развитием алгоритма корреляционной обработки. Заметим, что последние два блока рис. 1 можно интерпретировать как вычисление спектральной составляющей сигнала на доплеровской частоте F . Поэтому для перекрытия всего доплеровского диапазона эти два блока можно заменить блоком вычисления спектра (рис. 2).

Алгоритм состоит в демодуляции сигнала, принятого с заданной дальности, и в спектральном анализе демодулированного сигнала в диапазоне доплеровских частот.

Более строго корреляционно-фильтровая обработка может быть описана следующими формулами:

$$y(k) = \sum_{s=0}^{N-1} z(s) \exp\left(-2\pi \cdot j \frac{ks}{N}\right), \text{ при } s \geq n \quad (4)$$

$$k = 0, 1, \dots, N-1, \quad z(s) = 0,$$

где

$$z(s) = a^*(s)x(s+i_0), \quad s = 0, 1, \dots, n-1 \quad (5)$$

а $N \geq n$ — число точек вычисления спектра.

Поскольку выражение (4) представляет собой дискретное преобразование Фурье, то его можно вычислить с помощью алгоритма быст-

рого преобразования Фурье (БПФ), дополняя при необходимости входной сигнал нулями. Алгоритм позволяет получить выходы оптимальных фильтров для всех значений доплеровской частоты при фиксированном значении задержки, т.е. вычислить один столбец в матрице дальность-скорость. Для вычисления других столбцов вычисления необходимо повторить, взяв из входного сигнала выборки для другой дальности (задержки).

Удельный объем вычислений корреляционно-фильтровой обработки при использовании алгоритма БПФ равен [2]:

$$v_{КОФ} = \frac{1}{\beta} (5 \log_2 N + 6), \quad (6)$$

где $\beta = \frac{M}{N}$ — отношение числа необходимых доплеровских каналов к числу вычисляемых доплеровских каналов.

Оптимальная фильтрация сигналов в частотной области выполняется путем вычисления спектра входного сигнала, умножения на частотную характеристику оптимального фильтра и обратного преобразования во временную область. При вычислении спектра входного сигнала и при обратном преобразовании во временную область целесообразно использовать алгоритмы БПФ и обратного БПФ (ОБПФ).

Блок-схема алгоритма представлена на рис. 3.

В формульном выражении алгоритм описывается следующим образом. Спектр входного сигнала определяется формулой:

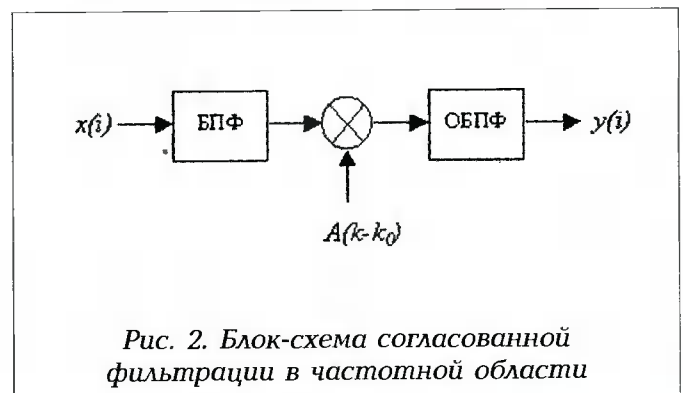
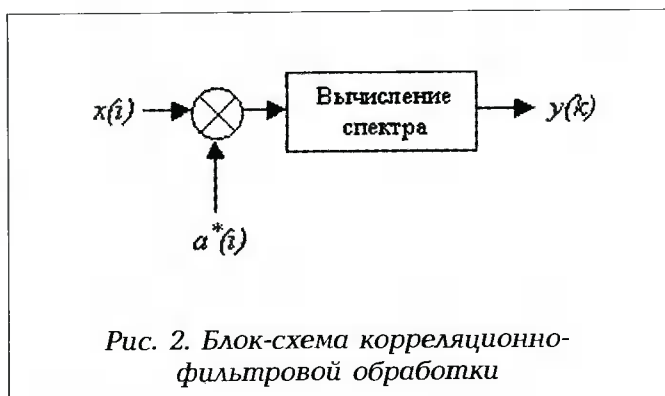
$$X(k) = \sum_{i=0}^{N-1} x(i) \exp\left(-2\pi \cdot j \frac{ik}{N}\right), \quad i = 0, 1, \dots, N-1 \quad (7)$$

Далее производится перемножение спектра сигнала и частотной характеристики оптимального фильтра $A(k)$ с учетом доплеровского сдвига k_0 :

$$Y(k) = X(k) \cdot A(k - k_0). \quad (8)$$

Для $k < k_0$ и $(k - k_0) \geq N$ принимается $A(k - k_0) = 0$.

Затем результат фильтрации переносится во временную область с помощью ОБПФ:



$$y(i) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) \exp\left(2\pi \cdot j \frac{ik}{N}\right) \quad (9)$$

При выполнении операции фильтрации возникает переходный процесс длительностью $n-1$ выборок. Выборки выходного сигнала, соответствующие переходному процессу, должны быть отброшены. Таким образом, число полезных выборок равно $N - n + 1$.

Выборки на выходе алгоритма соответствуют выходам оптимальных фильтров для $N - n + 1$ значений дальности и одной скорости (ее значение входит в частотную характеристику фильтра). Таким образом, алгоритм позволяет вычислять одну строку в матрице дальность—скорость. Для получения других строк вычисления необходимо повторить, изменив необходимым образом частотную характеристику фильтра.

Алгоритм согласованной фильтрации в частотной области реализует фильтрацию входного сигнала фильтром с заданной частотной характеристикой. Оптимальным фильтром для приема сигналов на фоне белого гауссовского шума является, как известно, согласованный фильтр [3-5]. Однако для получения приемлемого уровня боковых лепестков выходного сигнала частотная характеристика фильтра может корректироваться с помощью весовой функции (например, функции Хэмминга).

В радиолокационных применениях длительность обрабатываемого сигнала часто намного превышает длительность зондирующего импульса, поэтому для организации эффективных параллельных вычислений используется разбиение входного сигнала $x(i)$ на секции длины L . При этом удельный объем вычислений алгоритма фильтрации в частотной области составляет [2]:

$$v_{\text{од}} = \frac{2\alpha(5\log_2 \alpha n + 3)}{\alpha - 1 + \frac{1}{n}} \quad (10)$$

где $\alpha = \frac{L}{n}$

Длина секции L может быть выбрана произвольно. Минимум удельного объема вычислений достигается при выборе значения α в диапазоне 5...16 в зависимости от величины n . Эта зависимость является слабо выраженной. Так, при выборе $\alpha=2$ удельный объем возрастает всего на (30...40)%. Уменьшение длины секции позволяет увеличить число процессоров, работающих параллельно и независимо при реализации алгоритма.

Из сравнения формул для удельного объема вычислений перечисленных методов фильтрации можно сделать следующие качественные выводы. Если общее число каналов фильтрации невелико (порядка десятка), то наиболее эффективным

оказывается метод корреляционной обработки. При более значительном числе каналов, наоборот, корреляционная обработка неэффективна. Если при этом необходимое число каналов по скорости (по доплеровской частоте) мало (мала величина α в (6)), то более эффективным оказывается метод фильтрации в частотной области. Если же мало необходимое число каналов по дальности (мала величина α в (10)), то более эффективным является метод корреляционно-фильтровой обработки.

Рассмотрим некоторые особенности использования перечисленных алгоритмов для обработки типовых зондирующих сигналов, представленных в табл. 1 из [1]: импульсов с линейной частотной модуляцией (ЛЧМ-импульс), фазокодовой манипуляцией (ФКМ-импульс) и квазинепрерывного сигнала с высокой частотой повторения. Отметим при этом, что хотя все числовые параметры относятся к конкретным сигналам, многие из приводимых соображений справедливы для широких классов зондирующих сигналов.

3. Оптимальная фильтрация ЛЧМ-импульса

Параметры рассматриваемого сигнала ЛЧМ1 представлены в табл. 1 в первой части статьи: длительность импульса составляет $\tau_{\text{ИМГ1}} = 50$ мкс, ширина спектра $\Delta F = 2$ МГц, длительность периода зондирования $T_3 = 1$ мс.

ЛЧМ-импульс имеет «ножевидную» функцию неопределенности [3-5], поэтому величина отклика согласованного фильтра слабо падает при его рассогласовании по доплеровской частоте. Действительно, максимальное значение функции неопределенности ЛЧМ-импульса при частотной расстройке f можно оценить по формуле [3-5]:

$$\chi\left(-\frac{f}{\alpha}, f\right) = \text{rect}\left(\frac{f}{2\tau_{\text{ИМГ1}}\alpha}\right) \frac{\tau_{\text{ИМГ1}} - \frac{f}{\alpha}}{\tau_{\text{ИМГ1}}} \quad (11)$$

где $\alpha = \Delta F / \tau_{\text{ИМГ1}}$, ΔF — девиация частоты, а $\text{rect}(x)$ — функция, принимающая нулевое значение при $|x| > 0.5$ и единичное значение в противном случае.

Для рассматриваемого примера при максимальной доплеровской расстройке $|f| = 200$ кГц [1] потери составляют всего 0,92 дБ, что является вполне приемлемой величиной. Таким образом, в матрице дальность—скорость можно ограничиться единственным приемным каналом по частоте.

Число каналов по дальности для задачи поиска цели должно соответствовать максимально возможному диапазону дальностей. Определим этот параметр. Для рассматриваемого примера максимальная дальность определяется длительностью периода зондирования $T_3 = 1$ мс. Длитель-

ность анализируемого интервала времени приема сигнала вычисляется как разность периода зондирования и суммы длительностей бланка $T_{бл}$ и подготовки к зондированию T_r :

$$T_a = T_s - T_r - T_{бл} \quad (12)$$

В течение подготовки к зондированию производится установка луча антенны и автоматическая компенсация помех. В качестве примера время подготовки принимается равным $T_r = 300$ мкс. Длительность бланка равна $T_{бл} = 70$ мкс. Тогда $T_a = 630$ мкс. При частоте дискретизации $F_d = 10$ МГц [1] число принимаемых выборок входного сигнала равно $N = F_d T_a = 6300$, и число формируемых дальностных каналов имеет такую же величину. Временная диаграмма приема ЛЧМ-импульса показана на рис. 4.

Так как число каналов по дальности существенно больше, чем по скорости, то в соответствии с выводами предыдущего раздела явным преимуществом обладает алгоритм согласованной фильтрации в частотной области.

При реализации этого алгоритма для снижения уровня боковых лепестков откликов оптимальных фильтров может использоваться весовая функция в спектральной области, например, функция Хэмминга $W(i)$ [3-6]:

$$W(i) = 0,54 - 0,46 \cos\left(\frac{2\pi i}{L}\right), \quad i = 0..L-1 \quad (13)$$

При использовании весовой функции потери рассогласования по частоте уменьшаются, поэтому рассчитанное выше значение потерь можно считать верхней границей.

Частотная характеристика согласованного фильтра, хранящаяся в ПЗУ процессора сигналов, заранее умножается на весовую функцию:

$$A(i) = S^*(i) \cdot W(i), \quad (14)$$

где $S^*(i)$ — комплексно-сопряженный спектр зондирующего сигнала.

Использование весовой функции (13) приводит к потерям обработки 1,34 дБ [3-6]. Главный максимум корреляционной функции расширяется в 1,47 раза. Уровень боковых лепестков составит минус 42,8 дБ для идеального сигнала с бесконечной базой без доплеровского частотного сдвига. На практике уровень боковых лепестков возрастает как за счет неидеальности самого сигнала, так и за счет доплеровского сдвига и неидеальности предшествующего приемного тракта. По-видимому, наиболее достоверным способом получения конкретного значения этого уровня (за исключением натурального эксперимента) является моделирование в диапазоне доплеровских частот с учетом реальных характеристик предшествующих фильтров. Как показывает мо-

делирование, уровень боковых лепестков корреляционной функции удастся снизить до величины менее минус 30 дБ во всем диапазоне доплеровских частот.

Для реализации максимально параллельной обработки разобьем входной сигнал на секции длиной $L = 1024$ (рис. 4), что соответствует отношению $\alpha = 1024/500 = 2,048$. Число полезных отсчетов в одной секции составит $L - n + 1 = 1024 - 500 + 1 = 525$. Число требуемых секций определяется как отношение числа входных выборок установившегося сигнала на выходе оптимального фильтра к числу полезных выборок в одной секции:

$$N_c = \left\lceil \frac{N}{L-n+1} \right\rceil = \left\lceil \frac{6300}{525} \right\rceil = 12$$

где $\lceil x \rceil$ означает наименьшее целое число, превосходящее x . Шаг секций по оси времени равен числу полезных отсчетов секции — 525 отсчетов. Такое разбиение позволяет использовать до 12 независимо работающих процессорных элементов.

Не все сигналы, выборки которых наблюдаются, присутствуют полностью в интервале анализа. Сигналы от целей, находящихся на расстоянии менее 10,5 км, частично бланкируются. Сигналы от целей, расположенных не далее 3 км, в интервале анализа отсутствуют. Аналогично сигналы целей, расположенных на расстояниях от 97,5 км до 105 км, лишь частично присутствуют в интервале анализа.

Все вместе взятые секции позволяют получать 6300 выходных отсчетов, соответствующих различным каналам по дальности. Из них 5801 отсчет соответствуют сигналам, полностью присутствующим в интервале анализа. Сигналы, частично попадающие в интервал анализа, на выходе оптимального фильтра имеют меньшую амплитуду. Приблизительно можно принять, что их амплитуда пропорциональна длительности участка сигнала, попавшего в интервал анализа.

Расстановка секций на временной оси, показанная на рис. 4, выполнена таким образом, чтобы обеспечить работу с ближними целями, сигналы от которых частично попадают в бланк. При этом 470 первых выборок первой секции приходятся на бланк и принимаются равными нулю. Аналогично, последние 30 выборок последней секции принимаются равными нулю.

В табл. 1 представлены основные параметры описанного способа обработки ЛЧМ-импульса, а также аналогичные параметры для других сигналов, рассматриваемые в следующих разделах.

В таблице указаны также потери обработки, включающие «седловые» потери [4] по дальности, по скорости, потери из-за использования весовой функции и потери рассогласования,

4. Оптимальная фильтрация ФКМ-импульса

Параметры рассматриваемого сигнала ФКМ1 представлены в табл. 1 в первой части статьи: длительность импульса составляет $T_{имп} = 50$ мкс, длительность дискрета фазокодовой манипуляции — 0,5 мкс, ширина спектра $\Delta F = 2$ МГц, длительность периода зондирования $T_3 = 1$ мс. В качестве модулирующей последовательности можно использовать отрезок из 100 символов М-последовательности длины 127 (усеченную М-последовательность).

Свойства ФКМ-сигналов, использующих в качестве модулирующих кодов полные М-последовательности, хорошо изучены [3-5]. Судя по нашим исследованиям, хорошие корреляционные свойства сохраняются и для усеченных М-последовательностей. Для длины 100 полным компьютерным перебором М-последовательностей и их начальных фаз найдено, что наилучший отрезок М-последовательности дает уровень боковых лепестков на временной оси минус 20,9 дБ.

Функции неопределенности ФКМ-импульса имеет кнопочный вид [3-5]. Поэтому в данном случае требуется формирование значительного числа каналов как по дальности, так и по скорости. Формат матрицы дальность—скорость определяется многими факторами, поэтому желательно иметь возможность изменять его. Очевидно, что изменения формата допустимы в пределах вычислительных возможностей процессора обработки сигналов.

Максимальное число доплеровских каналов можно грубо оценить из очевидного соотношения:

$$N_v = F \cdot \tau_{имп} \quad (15)$$

где F — ширина перекрываемого диапазона доплеровских частот.

Для рассматриваемого случая $F < 400$ кГц. Легко видеть, что при этом требуется не более 20 доплеровских каналов. Число каналов по дальности обычно должно быть как минимум на порядок больше. Тогда в соответствии с выводами второго раздела явным преимуществом обладает алгоритм согласованной фильтрации в частотной области.

Ввиду равных длительностей сигналов ФКМ1 и ЛЧМ1, при обработке следует воспользоваться тем же методом секционирования, который используется для сигнала ЛЧМ1 (раздел 3). В таком случае при сохранении производительности процессора обработки сигналов можно получать матрицы дальность—скорость следующих форматов:

$$N_D = 525 \cdot i, \quad N_v = \left[\frac{12}{i} \right], \quad i = 1, \dots, 12, \quad N_v = \left[\frac{12}{i} \right] \quad (16)$$

где $[x]$ означает наибольшее целое число, не превосходящее x .

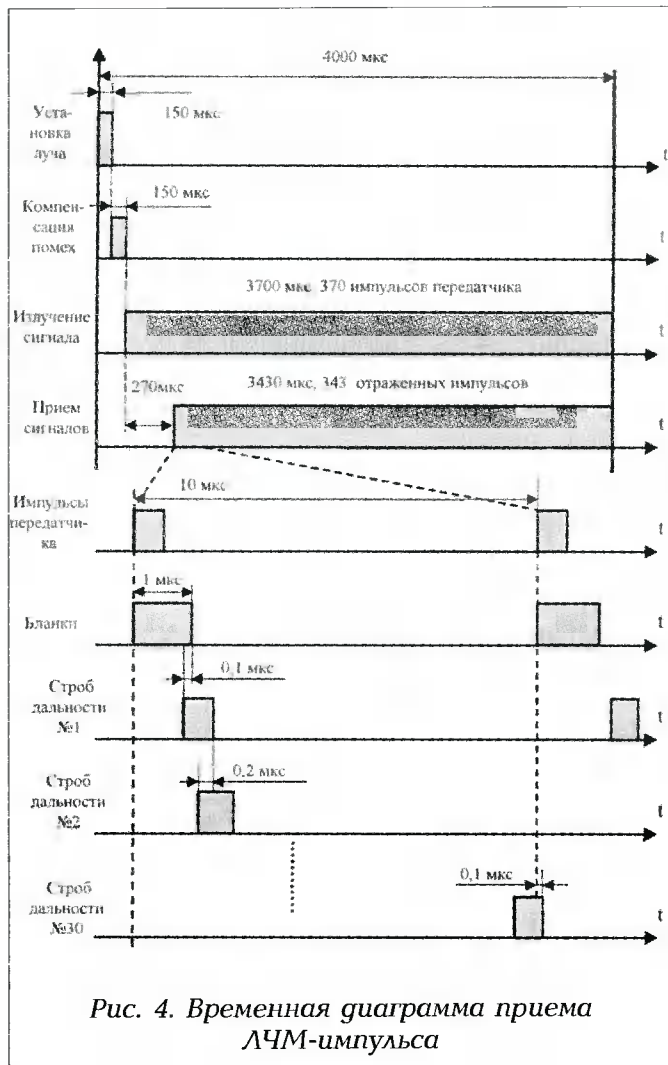


Рис. 4. Временная диаграмма приема ЛЧМ-импульса

которые возникают при искажении сложных сигналов в аналоговых трактах. Эмпирическое значение последних для ЛЧМ-сигнала принято равным 1 дБ, а для ФКМ-сигнала — 1,5 дБ.

Таблица 1 Основные параметры обработки типовых сигналов

Наименование параметра	Тип сигнала		
	ЛЧМ1	ФКМ1	КН1
Число каналов по дальности	6300	525 <i>i</i>	30
Число каналов по скорости	1	12/ <i>i</i>	512
Шаг каналов по дальности, мкс	0,1	0,1	0,3
Ширина канала по дальности, мкс	0,65	0,59	0,59
Шаг каналов по скорости, кГц	-	$9,77 \cdot p$ $p=1,2,\dots,41$	0,195
Ширина каналов по скорости, кГц	-	17,8	0,343
Потери обработки, дБ	2,9	3,6	4,4

После выбора модулирующего кода и метода обработки целесообразно провести моделирование в целях уточнения получаемых характеристик, в частности уровня боковых лепестков с учетом фильтра основной селекции, определяющего полосу пропускания, и заданного диапазона доплеровских частот. Для рассматриваемого примера уровень боковых лепестков в доплеровском диапазоне возрастает примерно до минус 14 дБ.

Основные параметры рассмотренного алгоритма приведены в табл. 1.

5. Оптимальная фильтрация квазинепрерывного сигнала

Рассмотрим теперь обработку квазинепрерывного сигнала КН1 с высокой частотой повторения $F_n = 100$ кГц, длительностью импульса $\tau_{\text{имп}} = 0,5$ мкс и периодом зондирования $T_3 = 4$ мс (табл. 1 в [1]).

Функция неопределенности такого сигнала имеет ряд повторяющихся пиков как по оси времени, так и по оси частоты [3-5]. В связи с небольшим однозначным диапазоном по дальности для данного режима (1,5 км) и высоким разрешением по доплеровской частоте (порядка $1/T_3$) задача обработки такого сигнала характеризуется большим числом каналов по скорости и малым числом каналов по дальности. В этом случае, как показано во втором разделе, наиболее эффективным является метод корреляционно-фильтровой обработки (4)-(5).

Особенностью этого алгоритма для квазинепрерывных сигналов является простейший вид операции демодуляции (4): для фиксированной дальности стробирование принятого сигнала пачкой импульсов, идентичной излученной, но сдвинутой на заданную дальность. После операции демодуляции необходимо провести спектральный анализ в однозначной доплеровской полосе $F_n = 100$ кГц.

Временная диаграмма приема сигнала показана на рис. 5. Период зондирования, как и для импульсных сигналов, начинается с установки луча антенной фазированной решетки в заданном направлении и компенсации помех. Оставшееся время 3700 мкс занято излучением сигнала. Начало приема сигнала задержано относительно начала излучения на 270 мкс, чтобы произошло установление мощных отражений от подстилающей поверхности, дальность до которых в типичном случае не превосходит 40 км. При этом время приема составит 3430 мкс и содержит $N_{\text{обр}} = 343$ отраженных импульса.

В нижней части рис. 5 в увеличенном масштабе показан один период повторения импульсов длительностью $T_n = 1/F_n = 10$ мкс. Длительность бланков выбрана равной 1 мкс; перекрытие τ_s соседних дальностных стробов — 0,2 мкс, что соответствует двум отсчетам АЦП [1]. Величина пере-

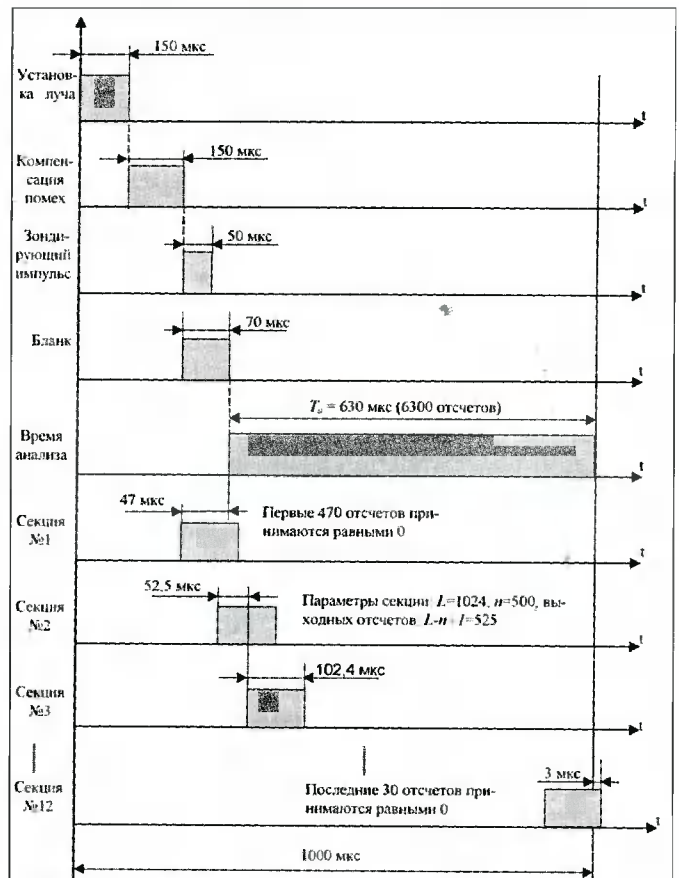


Рис. 5. Временная диаграмма приема квазинепрерывного сигнала

крытия определяет уровень «седловых» потерь по дальности:

$$L = -20 \lg \left(1 - \frac{\tau_{\text{имп}} - \tau_s}{4 \cdot \tau_{\text{имп}}} \right) = 1,4 \text{ dB}$$

При этом для перекрытия стробами всего интервала между соседними импульсами требуется $N_D = 30$ каналов дальности. Стробы дальности закрывают интервал в 92 выборки, при необходимом закрытии 90 выборок. Поэтому первый и последний стробы перекрываются с бланком одинаково, на 1 отсчет. Выборки, приходящиеся на бланки, при обработке принимаются равными нулю.

Для данного типа сигнала эффективность алгоритма вычисления спектра (4) можно увеличить, если, во-первых, принять во внимание, что большая часть выборок $a(s)$ в (5), и следовательно и выборок $z(s)$ в (4), равна нулю, а во-вторых, учесть, что при использовании БПФ спектр (4) вычисляется в диапазоне частот от 0 до F_n , тогда как требуется анализ только в однозначном диапазоне доплеровских частот шириной $F_n \ll F_d$.

Для описания этого более эффективного алгоритма введем необходимые обозначения. Пусть спектр $y(k)$ вычисляется в N равноотстоящих точках интервала $[f_H, f_H + F_n]$, где f_H кратно частоте повторения F_n :

$$f_k = f_n + \frac{kF_n}{N}, \quad k = 0, 1, \dots, N-1.$$

Обычно величина N не менее числа обрабатываемых импульсов $N_{обр}$. Пусть также n — число выборок на длительности импульса, а x_{ij} — i -я выборка сигнала в j -м периоде повторения. Введем для краткости обозначение $\expn(a) = (2\pi\sqrt{-1}a)$. Тогда рассматриваемый алгоритм описывается следующими формулами:

$$y(k) = \sum_{i=0}^{n-1} g_{ik} \expn \left[\frac{i}{F_n} \left(f_n + \frac{kF_n}{N} \right) \right] \quad (17)$$

где

$$g_{ik} = \sum_{j=0}^{N-1} x_{ij} \expn \left(\frac{jk}{N} \right), \quad (18)$$

$i = 0, 1, \dots, n-1; \quad k = 0, 1, \dots, N-1$

Формулу (18) целесообразно вычислять с помощью алгоритма БПФ. Для рассматриваемого сигнала удобно положить $N = 512$, дополнив $N_{обр} = 343$ обрабатываемых импульса необходимым количеством нулей.

Существо алгоритма поясняет рис. 6. Вначале выполняется анализ спектра для равноотстоящих ненулевых выборок с одинаковыми индексами i (формула (18)), затем производится суммирование частных спектров с учетом фаз (формула (17)).

Удельный объем вычислений алгоритма составляет:

$$v_{кфол} = n(5 \log_2 N + 8) - 2 \quad (19)$$

Для рассматриваемого сигнала формула (19) дает величину примерно в 20 раз меньшую, чем при прямом вычислении (4) с помощью БПФ.

При использовании квазинепрерывных сигналов с высокой частотой повторения прием приходится вести на фоне мощных отражений от подстилающей поверхности и гидрометеоров. Поэтому при вычислении частичных спектров (18) необходимо принять меры для подавления боковых лепестков спектра. Эта задача может быть решена либо путем введения весовой функции во

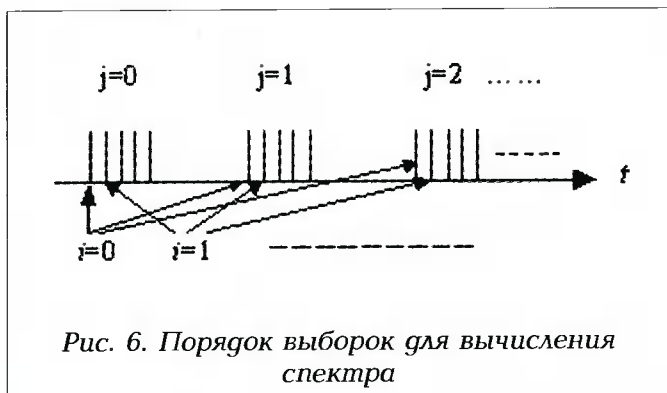


Рис. 6. Порядок выборок для вычисления спектра

временной области перед спектральным анализом [6], либо в частотной области путем эквивалентной фильтрации спектра [7]. Замечательно, что в последнем случае уровень боковых лепестков ниже минус 90 дБ может быть достигнут уже при порядке фильтра $m=7$ [7].

В данной работе рассмотрим подавление боковых лепестков с помощью весовой функции. Эта функция должна обеспечить снижение уровня лепестков ниже уровня шума вне области малых доплеровских частот. Необходимое подавление определяется величиной отражений от подстилающей поверхности и гидрометеоров и достигает величин 80...100 дБ [1]. Для получения такого высокого уровня можно использовать весовые функции Дольфа – Чебышева, обеспечивающие минимальное расширение главного лепестка при заданном уровне боковых.

В частотной области функции Дольфа – Чебышева описываются формулой

$$W(f) = \cos [m \cdot \arccos(z \cdot \cos \pi f)] \quad (20)$$

где m — степень полинома Чебышева, частота f нормализована к частоте выборок F_n ($f \in [0, 1]$), а величина z вычисляется как функция отношения ϵ максимального уровня боковых лепестков к амплитуде главного максимума:

$$z = \cosh \left\{ \frac{1}{m} \ln \left[\frac{1}{\epsilon} + \sqrt{\left(\frac{1}{\epsilon^2} - 1 \right)} \right] \right\} \quad (21)$$

Величина m в (20) должна быть четной и удовлетворять соотношению $m < N_{обр}$.

Выборки весовой функции во временной области могут быть получены с помощью обратного дискретного преобразования Фурье от функции (20).

Высокий уровень подавления приводит к довольно значительным потерям в отношении сигнал/шум, что является платой за возможность работы на фоне мощных сигналов с нулевой доплеровской частотой. Для рассматриваемого примера при уровне подавления 90 дБ потери составляют около 2,7 дБ, при этом происходит расширение главного максимума спектра сигнала в 1,76 раза. Расширение главного максимума приводит к снижению «седловых» потерь по частоте до 0,31 дБ.

Основные параметры рассмотренного алгоритма приведены в табл. 1 в конце раздела 3.

При использовании описанного режима работы в МРЛС с вращающейся антенной принимаемая «пачка» импульсов модулируется диаграммой направленности антенны, что искажает весовую функцию. Логично ожидать, что это снижает уровень подавления боковых лепестков спектра. Однако проведенные авторами исследования

показали, что этот эффект незначителен даже при высокой скорости вращения, при которой за период зондирования антенна поворачивается на 0,5 ширины диаграммы направленности. Возрастания уровня лепестков практически не наблюдается. Ширина спектра полезного сигнала и потери при его обработке при этом растут незначительно.

6. Обнаружение

Исходной информацией для алгоритмов обнаружения сигналов целей является матрица дальность-скорость, сформированная на предыдущем этапе. Обозначим эту матрицу $A(i, j)$ ($i = 0, 1, \dots, N - 1$, $j = 0, 1, \dots, M - 1$). Для удобства дальнейшей индексации примем, что индекс i для сигналов, обработка которых производится методом согласованной фильтрации в частотной области (импульсные сигналы), определяет номер канала по дальности, а для сигналов, обрабатываемых корреляционно-фильтровым методом (квазинепрерывные сигналы), — номер канала по скорости. Соответственно, индекс j при обработке методом согласованной фильтрации определяет номер канала по скорости, а при обработке корреляционно-фильтровым методом — номер канала по дальности. В случае, когда имеется всего один канал по скорости (для ЛЧМ-импульса), матрица $A(i, j)$ вырождается в вектор $A(i)$. Тем не менее, для унификации обозначений будем рассматривать этот вектор как матрицу $A(i, j)$ с одним столбцом ($M=1$).

Обнаружение, как правило, проводится стандартным методом амплитудного сравнения элементов матрицы $A(i, j)$ с порогом. Логика пороговой обработки может предусматривать использование двух типов порогов:

- адаптивного, обеспечивающего стабилизацию вероятности ложных тревог, вычисляемого пропорционально среднему уровню шума и применяемого в условиях априорной неопределенности помеховых условий работы;
- фиксированного, задаваемого по специальному алгоритму, учитывающему априорную информацию.

Пороговая обработка с фиксированным порогом заключается в сравнении элементов $A(i, j)$ с порогом, задаваемым от управляющего вычислительного комплекса. Характеристики этого способа могут быть найдены, например, в [3].

Рассмотрим алгоритм пороговой обработки со стабилизацией вероятности ложных тревог, являющийся модификацией известного алгоритма с вычислением порога в «скользящем окне» [4]. При этом порог $H(i, j)$ для элемента $A(i, j)$ вычисляется в соответствии со следующим выражением, представляющим собой оценку среднего уровня шума в некоторой области по координате i вокруг $A(i, j)$:

$$H(i, j) = \begin{cases} \frac{h}{n} \sum_{k=i-n/2}^{i+n/2} \min \left\{ A \left[\left(k + \frac{m}{2} \right), j \right], A \left[\left(k - \frac{m}{2} \right), j \right] \right\} \\ \text{при } \frac{n+m}{2} \leq i < N - \frac{n+m}{2}, \\ H \left(\frac{n+m}{2}, j \right) & \text{при } i < \frac{n+m}{2}, \\ H \left(N - \frac{n+m}{2}, j \right) & \text{при } i \geq N - \frac{n+m}{2}, \end{cases}$$

где n — величина окна накопления;

m — параметр для селекции шума от сигнала;

h — относительное пороговое значение или коэффициент пропорциональности, задаваемый управляющим вычислительным комплексом и зависящий от выбранной вероятности ложных тревог.

Вторая и третья строки в формуле определяют правила вычисления порога для случаев, при которых индекс в сумме в первой строке выходит за допустимые пределы $0 \leq i \leq N - 1$.

Увеличение окна накопления n , с одной стороны, приводит к повышению точности оценки среднего значения шума в окне, а, с другой стороны, ухудшает адаптацию в случае неравномерной шумовой картины в матрице дальность—скорость.

Основное отличие рассматриваемого алгоритма от известных [4] состоит в предварительной селекции шума от сигнала для устранения влияния сигнальных составляющих на уровень порога, что особенно важно при обнаружении малого сигнала на фоне большого в пределах окна накопления. Из этого следует, что величину параметра m целесообразно выбрать не менее ширины главного лепестка оптимального фильтра по координате j , измеренной в количестве выходных отсчетов. При этом полученное число целесообразно увеличить до ближайшего четного для соблюдения симметрии в формуле вычисления порогового значения.

Решение об обнаружении сигнала принимается в том случае, если выполняется условие:

$$A(i, j) > H(i, j)$$

Характеристики рассмотренного алгоритма пороговой обработки для случая гауссовского входного шума (зависимости вероятности ложной тревоги F от уровня порога h) и аддитивной смеси сигнала с гауссовским шумом (зависимости вероятности правильного обнаружения D от отношения сигнал/шум q), полученные с помощью математического моделирования, приведены на рис. 7 и 8 соответственно. При этом сначала проводилось моделирование алгоритмов когерентной обработки сигнала, описанных выше с учетом характеристик предварительной аналоговой обработки сигналов [2]. Отношение полосы

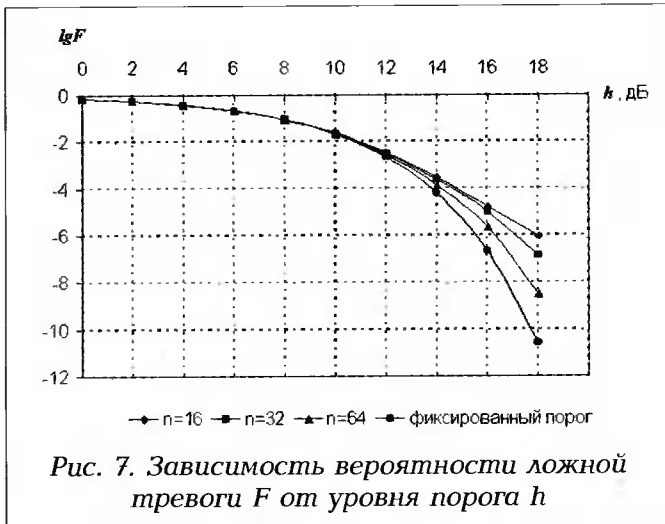


Рис. 7. Зависимость вероятности ложной тревоги F от уровня порога h

пропускания аналогового тракта к ширине спектра сигнала принималось равным 1, а отношение частоты дискретизации АЦП к ширине спектра сигнала — 4. Затем проводилось статистическое моделирование рассматриваемого метода пороговой обработки для различных величин окна накопления n при величине параметра $m > 6$.

Для сравнительной оценки на рис. 7 и 8 приведены теоретические зависимости, соответствующие пороговой обработке с фиксированным порогом. Следует отметить, что при расчете теоретической зависимости учитывалось, что средний уровень шума вычисляется по формуле, аналогичной выражению для вычисления порога. В результате подобного вычисления среднего уровня шума получается значение, равное $0,89 \sigma$, где σ — среднеквадратическое значение исходного шума с релеевским законом распределения. Результаты проведенного математического моделирования показывают, что при рабочих значениях порога (14...16) дБ, соответствующих вероятности ложной тревоги примерно 10^{-5} , размер окна накопления целесообразно выбирать в диапазоне $n = 32...64$, что обеспечит потери при обработке не более 1 дБ. Эти потери можно интерпретировать как необходимое повышение уровня поро-

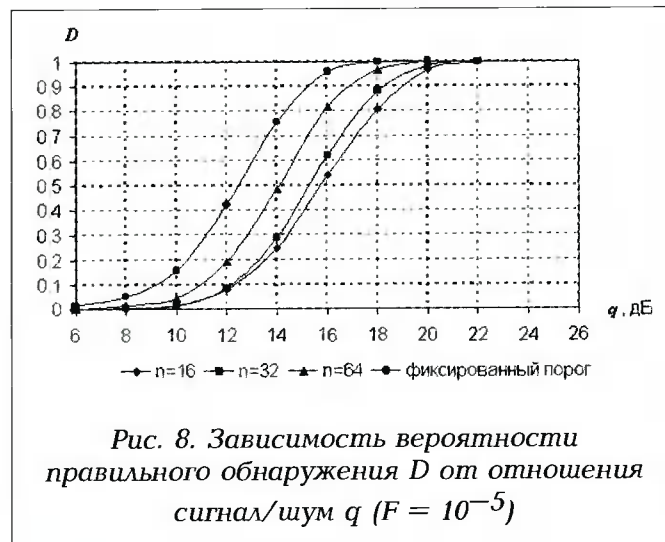


Рис. 8. Зависимость вероятности правильного обнаружения D от отношения сигнал/шум q ($F = 10^{-5}$)

га для достижения той же вероятности ложных тревог.

7. Измерение координат дальности и скорости

После выполнения пороговой обработки в матрице $A(i,j)$ известны все элементы, превысившие порог. Дальнейшая обработка проводится в два этапа.

Первый этап заключается в формировании в матрице $A(i,j)$ связанных областей элементов, каждая из которых принимается за целевую отметку. Второй этап состоит в измерении координат дальности и скорости этой отметки.

Для формирования в матрице $A(i,j)$ связанных областей элементов, превысивших порог, сначала производится анализ каждого столбца j_k , т.е. множества элементов $A(i,j_k), 1, 2, \dots, n$, и формирование в нем одномерных связанных областей. При этом множество элементов:

$$\Omega = \{A(i_{min}, j_k), A(i_{min} + 1, j_k), A(i_{min} + 2, j_k), \dots, A(i_{max}, j_k)\}$$

образует одномерную связную область, если все они больше порога, а элементы $A(i_{min} - 1, j_k)$ и $A(i_{max} + 1, j_k)$ меньше порога.

Одномерные области связываются в двумерные по следующему правилу. Производится просмотр связанных областей из всех пар рядом стоящих столбцов, т.е. столбцов с номерами j и $j + 1, j = 1, 2, \dots, m - 1$. Одномерная связная область Ω_1 с граничными координатами i_{min1}, i_{max1} , из j -го столбца и одномерная связная область Ω_2 с координатами i_{min2}, i_{max2} , из $j + 1$ -го столбца объединяется в двумерную связную область Ω при выполнении следующих условий:

$$i_{max1} \geq i_{min2}, \quad i_{max2} \geq i_{min1}$$

Иными словами объединение происходит, если найдется хотя бы одна строка, в которой обе области имеют по элементу. Каждая сформированная таким методом двумерная область Ω характеризуется максимальной амплитудой A_{max} с координатами i_0, j_0 и четырьмя граничными координатами.

На втором этапе для каждой связной области выполняется оценка ее параметров. Примем за основу метод максимума отношения правдоподобия. Тогда координаты дальности и скорости определяются координатами локальных максимумов в матрице дальность-скорость, удовлетворяющих условиям обнаружения. Поскольку простой поиск максимума дает оценку координат с дискретностью расстановки приемных каналов, необходимо выполнить уточнение координат для устранения ошибок из-за дискретности шкалы отсчета. Алгоритм уточнения координат основан на использовании измерений амплитуд в прием-

ных каналах, смежных с каналом, содержащим максимум. При этом расстановка смежных каналов должна быть достаточно плотной, чтобы все три канала находились в пределах главного максимума функции неопределенности. Это условие, как правило соблюдается, так как плотная расстановка необходима также для снижения «седловых» потерь. Значения амплитуд этих каналов используются для вычисления поправки к целочисленной координате максимума для уточнения оценки координаты. При этом сначала вычисляется сигнал ошибки e в соответствии с соотношением

$$e = \frac{A_+ - A_-}{A_{max}} \quad (22)$$

где A_+ — амплитуда канала с большим номером, A_- — амплитуда канала с меньшим номером, A_{max} — амплитуда максимума.

Помимо соотношения (22) для вычисления сигналов ошибки может использоваться соотношение

$$e = \frac{A_{max} - A_x}{A_{max} + A_x} \quad (23)$$

где $A_x = A_+$, если $A_+ > A_-$, в противном случае $A_x = A_-$.

Поправка Δ координаты к координате канала с максимальной амплитудой вычисляется по следующей формуле:

$$\Delta = \frac{e}{k} \quad (24)$$

где k — крутизна дискриминационной характеристики, зависящая от типа зондирующего сигнала и способа вычисления сигнала ошибки.

Зависимость $e(\Delta)$, называемая дискриминационной характеристикой, является нелинейной, однако вблизи точки $\Delta = 0$ приближается к линейной. Значение k в (24) представляет собой производную функции $e(\Delta)$:

$$k = \frac{\partial e}{\partial \Delta} \quad (25)$$

Вид дискриминационной характеристики зависит не только от выбора метода вычисления сигнала ошибок (22) или (23), но и от шага расстановки каналов. В качестве примера на рис. 9 представлен типичный вид дискриминационных характеристик по скорости для ФКМ-импульса. На рисунке использованы следующие обозначения: δ_f — шаг каналов по частоте, нормированный к ширине канала $1/\tau_{\text{ИМП}}$, по оси абсцисс отложена нормированная расстройка $f\tau_{\text{ИМП}}$ истинной координаты относительно канала с максимальной амплитудой.

Формула (24) соответствует линейному приближению дискриминационной характеристики.

Для получения наилучшей оценки целесообразно оптимизировать значение k по критерию минимума среднеквадратического отклонения аппроксимирующей прямой от истинной дискриминационной кривой:

$$k = \frac{\sum_{i=0}^{N-1} e_i \Delta_i}{\sum_{i=0}^{N-1} \Delta_i^2} \quad (26)$$

где e_i — значение дискриминационной характеристики для отклонения Δ_i ; суммирование производится для N точек, равномерно расставленных в пределах заданного рабочего диапазона отклонений.

Из двух способов (22), (23) вычисления сигнала ошибки целесообразно выбрать тот, который обеспечивает наименьшую среднеквадратическую ошибку линейризации в диапазоне отклонений координаты в пределах половины шага расстановки каналов. При этом, как правило, удается найти такое значение шага и способ вычисления сигнала ошибки, который обеспечивает пренебрежимо малую ошибку линейризации. Например, для рассмотренных в разделах 3-5 зондирующих

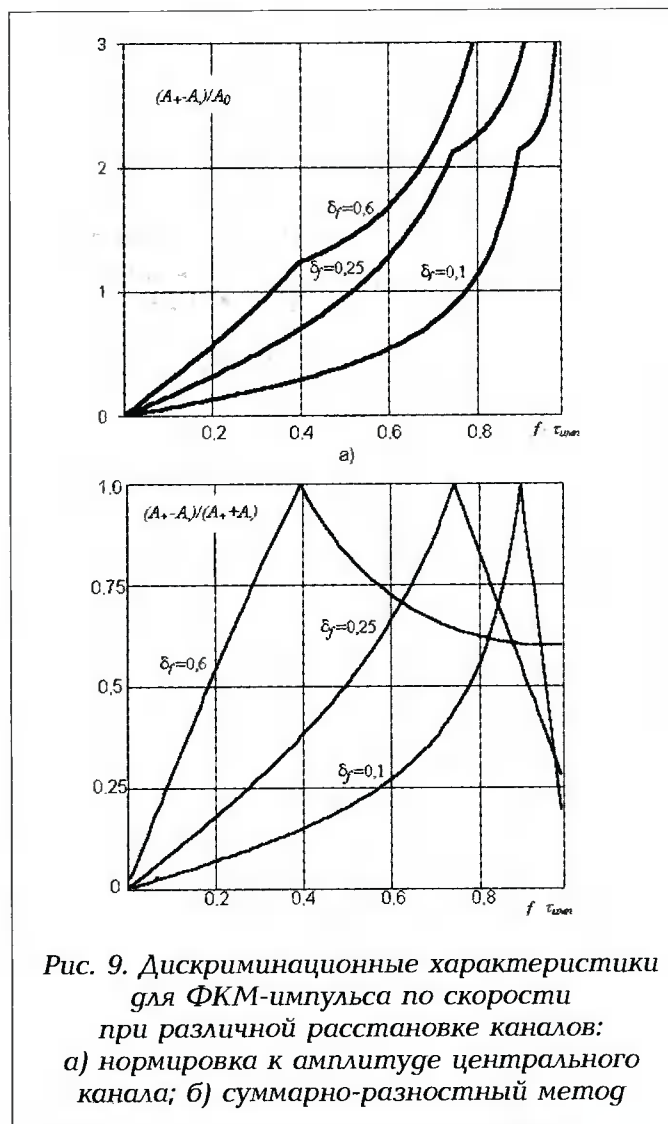


Рис. 9. Дискриминационные характеристики для ФКМ-импульса по скорости при различной расстановке каналов: а) нормировка к амплитуде центрального канала; б) суммарно-разностный метод

сигналов и способов обработки сигналы ошибки по дальности целесообразно вычислять в соответствии с (23), а сигналы ошибки по скорости — в соответствии с (22). При этом значения среднеквадратической ошибки линеаризации колеблются в диапазоне от 3×10^{-4} до 2×10^{-3} от ширины приемного канала по соответствующей координате.

В литературе описан также другой способ оценки координат, основанный на эвристическом алгоритме вычисления «центра тяжести» области вокруг максимума:

$$i = \frac{\sum_{j \in \Omega} j \cdot A_j}{\sum_{j \in \Omega} A_j}, \quad (27)$$

здесь суммирование производится по всем элементам связной области. Математическое моделирование показывает, что описанный выше дискриминационный метод дает в среднем более точную оценку, чем метод «центра тяжести».

Рис. 10 иллюстрирует это положение для случая оценки скорости при использовании ФКМ — сигнала. По горизонтальной оси отложено нормированное отклонение действительного значения доплеровской частоты от частоты настройки приемного канала с максимальной амплитудой. По вертикальной оси отложено нормированное среднеквадратическое значение ошибки оценки скорости. Нормировка по обеим осям производится умножением на $\tau_{\text{имп}}$.

Математическое моделирование показывает также, что для рассматриваемых в данной статье сигналов среднеквадратическое значение ошибки определения координат дискриминационным методом лишь на 10% - 20% больше нижней границы Крамера – Рао, а вклад ошибок линеаризации для отношения сигнал/шум 10...20 дБ пренебрежимо мал по сравнению с флуктуационной составляющей.

8. Обработка сигналов в режиме сопровождения

Режим сопровождения имеет два основных отличия от режима обнаружения. Во-первых, вы-

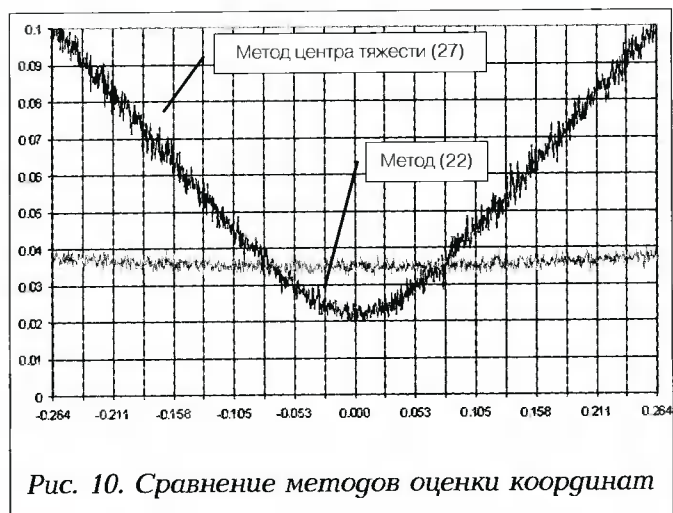


Рис. 10. Сравнение методов оценки координат

числяются сигналы ошибок по угловым координатам, для чего в моноимпульсной МРЛС обрабатываются разностные угловые каналы антенной системы. Эту обработку в принципе можно проводить в режиме обнаружения. Однако, она является неотъемлемым атрибутом именно для режима сопровождения.

Во-вторых, область априорной неопределенности положения цели гораздо меньше, чем для режима обнаружения. Поэтому формируемая матрица приемных каналов дальность—скорость в суммарном канале антенной системы имеет небольшие размеры и формируется вокруг ожидаемого положения цели, задаваемого следящей системой. Конкретные ее размеры зависят от частоты обращения к цели и ее динамики, и, как правило, не превышают нескольких единиц. Задачей устройства обработки является определение разности ожидаемых координат и оценок, полученных в текущем зондировании.

Рассмотрим обработку суммарного канала моноимпульсного радиолокатора. Малое число каналов фильтрации требует невысокой производительности процессора сигналов. Избыточная производительность может быть использована для выполнения описанной выше процедуры обнаружения с матрицей большой размерности с целью получения информации об обстановке вблизи сопровождаемой цели.

Алгоритм работы УОС в режиме сопровождения целей представлен на рис. 11. Управляющий вычислительный комплекс выдает в УОС пакет управляющей информации о предполагаемых координатах цели, размерности матрицы приемных каналов для алгоритма обнаружения, размерности и координатах подматрицы для алгоритма сопровождения.

На первом этапе обработка производится в соответствии с алгоритмом обнаружения, затем производится поиск максимума внутри матрицы сопровождения, которая является подматрицей матрицы обнаружения. Вычисление уточненных координат производится в соответствии с изложенными в предыдущих разделах алгоритмами. Важно отметить, что использование фиксированной дискретной расстановки фильтровых каналов не приводит к сколько-нибудь заметному проигрышу в точности измерений по сравнению с аналоговыми системами, где, как правило, используется непрерывная перестройка дискриминационных приемных каналов по дальности и скорости. Причина этого заключается в высокой линейности дискриминационных характеристик при реально используемых в цифровой обработке шагах расстановки каналов по дальности и скорости.

В режиме сопровождения полагается, что в подматрице сопровождения дальность—скорость имеется только одна цель. Конечное значение вероятности появления шумовых выбросов, превы-

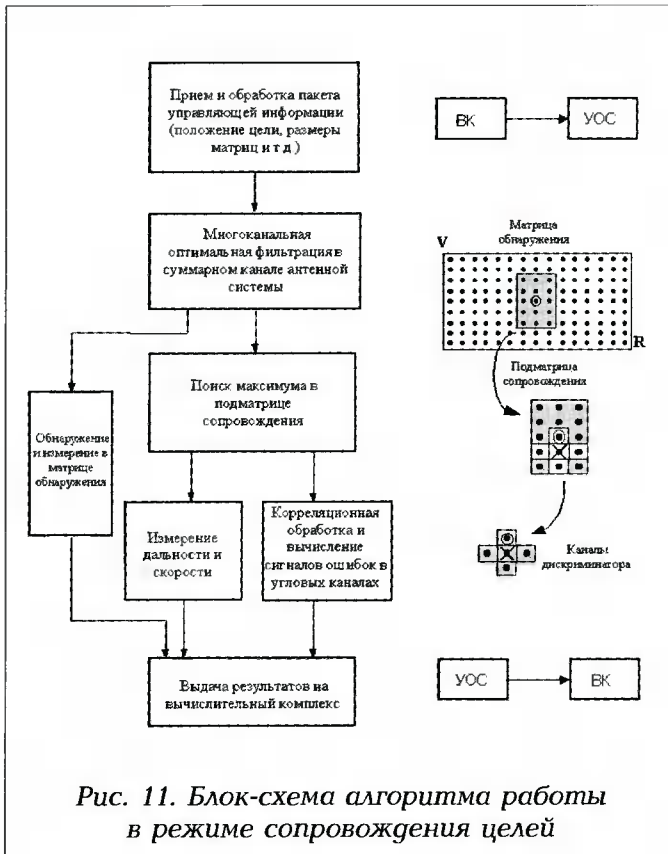


Рис. 11. Блок-схема алгоритма работы в режиме сопровождения целей

шающих амплитуду полезного сигнала, заставляет при оценке среднеквадратических ошибок измерений рассматривать аномальные измерения. Численный анализ, выполненный в соответствии с [8], показывает, что при отношении сигнал/шум более 10 дБ и размерности подматрицы, внутри которой находится цель, не более 1000 элементов разрешения, влияние аномальных измерений пренебрежимо мало.

В режиме сопровождения цели в моноимпульсных МРЛС, кроме выработки оценок дальности и скорости, необходимо вычислить значения сигналов ошибок по угловым координатам. Поскольку по результатам обработки сигнала суммарного канала известен номер канала по дальности и скорости, где обнаружена цель, то фильтрация сигналов в угловых каналах производится только для этого дальностно-скоростного канала. Как следует из выводов раздела 2, наиболее эффективным алгоритмом обработки в этом случае является корреляционная обработка.

Вычисление сигналов ошибок по угловым координатам выполняется в соответствии со следующим соотношением:

$$e_{\varphi} = Re \left(C_{\varphi} \cdot \frac{y_{\varphi}}{y_{\Sigma}} \right) \quad (28)$$

где y_{Σ} — комплексное значение на выходе оптимального фильтра в суммарном канале, дающего максимальный отклик в подматрице сопровождения дальность—скорость;

y_{φ} — комплексное значение на выходе того же

дальностно-скоростного фильтра в угловом канале;

C_{φ} — юстировочный множитель, обеспечивающий коррекцию усиления и фазовых сдвигов в аналоговых приемно-усилительных каналах.

9. Заключение

Теория цифровой обработки, развитая на сегодняшний день, позволяет эффективно решать задачи обработки радиолокационных сигналов в цифровом виде.

Рассмотренные в статье алгоритмы обработки радиолокационных сигналов условно делятся на две группы: алгоритмы многоканальной фильтрации и алгоритмы обнаружения — измерения. Первые характеризуются развитым параллелизмом и высокими вычислительными затратами, вторые значительно проще по числу операций, но зависят от содержания обрабатываемой информации и распараллеливаются значительно хуже. Эти особенности должны быть учтены при разработке архитектуры программируемого процессора сигналов.

Литература

1. Д.Ю.Бобров, А.П.Доброжанский, Г.В.Зайцев, Ю.В.Маликов, И.Б.Цыпин, *Обработка сигналов в многофункциональных РЛС// Цифровая обработка сигналов, 2001, №4*
2. Зайцев Г.В., *Оптимизация структуры и производительности цифрового устройства для согласованного приема сложных сигналов, Вопросы специальной радиоэлектроники, сер. «Радиолокационная техника», 1986, вып. 7, с. 61-77.*
3. Skolnic M. I., ed. *Radar Handbook, McGraw-Hill, 1990.*
4. Nathanson F.E., *Radar Design Principles, McGraw-Hill, 1991.*
5. Barton D. K., *Modern Radar System Analysis, Artech House, 1988.*
6. Хэррис Ф. Дж., *Использование окон при гармоническом анализе методом дискретного преобразования Фурье, ТИИЭР, том 66, 1978, №1, с. 60-96.*
7. Зайцев Г.В., *Синтез фильтров сглаживания боковых лепестков дискретного спектра, оптимальных по обобщенному минимаксному критерию, Вопросы специальной радиоэлектроники, Сер. «Радиолокационная техника», 1983, вып. 4, с.103-123.*
8. Куликов Е. И., Трифонов А. П., *Оценка параметров сигналов на фоне помех, Сов. Радио, Москва, 1978.*

Эффективность параллельных систем ЦОС, построенных на процессорах ADSP-21160

С. А. Дорохин

Последние поколения универсальных процессоров цифровой обработки сигналов (ЦОС) обладают впечатляющими характеристиками производительности. Но это лишь их потенциальные возможности, которые нужно уметь использовать в полной мере. Эффективность любой вычислительной системы можно определить как соотношение стоимости и производительности. Причем для систем ЦОС важна не просто производительность, а производительность для конкретного класса задач.

В данной статье рассматриваются варианты эффективного построения вычислительных систем ЦОС на примере использования процессоров ADSP-21160 фирмы Analog Devices.

Процессор ADSP-21160

Процессоры ADSP-21160 фирмы Analog Devices принадлежат ко второму поколению процессоров семейства SHARC (Super Harvard Architecture). Это универсальные высокопроизводительные процессоры, имеющие архитектуру и систему команд, оптимизированные для решения задач ЦОС. Основные области применения данных процессоров — это те задачи ЦОС, где требуется высокая производительность вычислительных устройств с плавающей точкой, таких, например, как обработка в реальном времени радиолокационных и гидроакустических сигналов.

Основные характеристики процессоров ADSP-21160 представлены в табл. 1. Для сравнения приведены также характеристики процессора предыдущего поколения ADSP-21060. В круглых

скобках указаны значения для варианта процессора с частотой 80 МГц.

Из таблицы видно, что пиковая производительность процессора ADSP-21160 в 5 раз превышает производительность процессора предыдущего поколения ADSP-21060. В целом, архитектура нового процессора повторяет архитектуру предшественника. Пятикратное увеличение производительности складывается из 2-х составляющих:

- повышение внутренней тактовой частоты в 2,5 раза;
- использование 2-х идентичных вычислительных узлов.

Основной особенностью процессора ADSP-21160 является архитектура SIMD (Single Instruction Multiply Data), которая позволила в одном процессорном такте выполнять до 6 вычислительных операций с плавающей точкой, что и дает пиковую производительность (при частоте 100 МГц) 600 МФЛОПС.

Блок схема архитектуры процессора ADSP-21160 приведена на рис. 1.

Процессор ADSP-21160 имеет два идентичных процессорных элемента (PE) по 3 вычислительных узла в каждом:

- 32/40 АЛУ с фиксированной и плавающей точкой;
- 32/40 умножитель с фиксированной и плавающей точкой, и 80 бит аккумулятор;
- сдвигатель.

Кроме того, каждый PE содержит адресный генератор и регистровый файл (32 40-битных регистра).

Таблица 1

Название параметра	ADSP-21160	ADSP-21060
Внутренняя тактовая частота	100 МГц (80 МГц)	40 МГц
Пиковая производительность	600 МФЛОПС (480 МФЛОПС)	120 МФЛОПС
Частота внешней шины данных	50 МГц (40 МГц)	40 МГц
Ширина внешней шины данных	64 бита	32 бита
Пропускная способность внешней шины данных	400 Мбайт/с (320 Мбайт/с)	160 Мбайт/с
Размер внутренней памяти	4 Мбита	4 Мбита
Двухпортовая внутренняя память	Да	Да
Число каналов прямого доступа к памяти (ПДП)	14	10
Встроенная многопроцессорная поддержка	Кластер до 6-ти процессоров и 6 коммуникационных портов до 100 Мбайт/с (80 Мбайт/с) каждый	Кластер до 6-ти процессоров и 6 коммуникационных портов до 40 Мбайт/с каждый
Время вычисления комплексного быстрого преобразования Фурье	90 мкс (112,5 мкс)	450 мкс

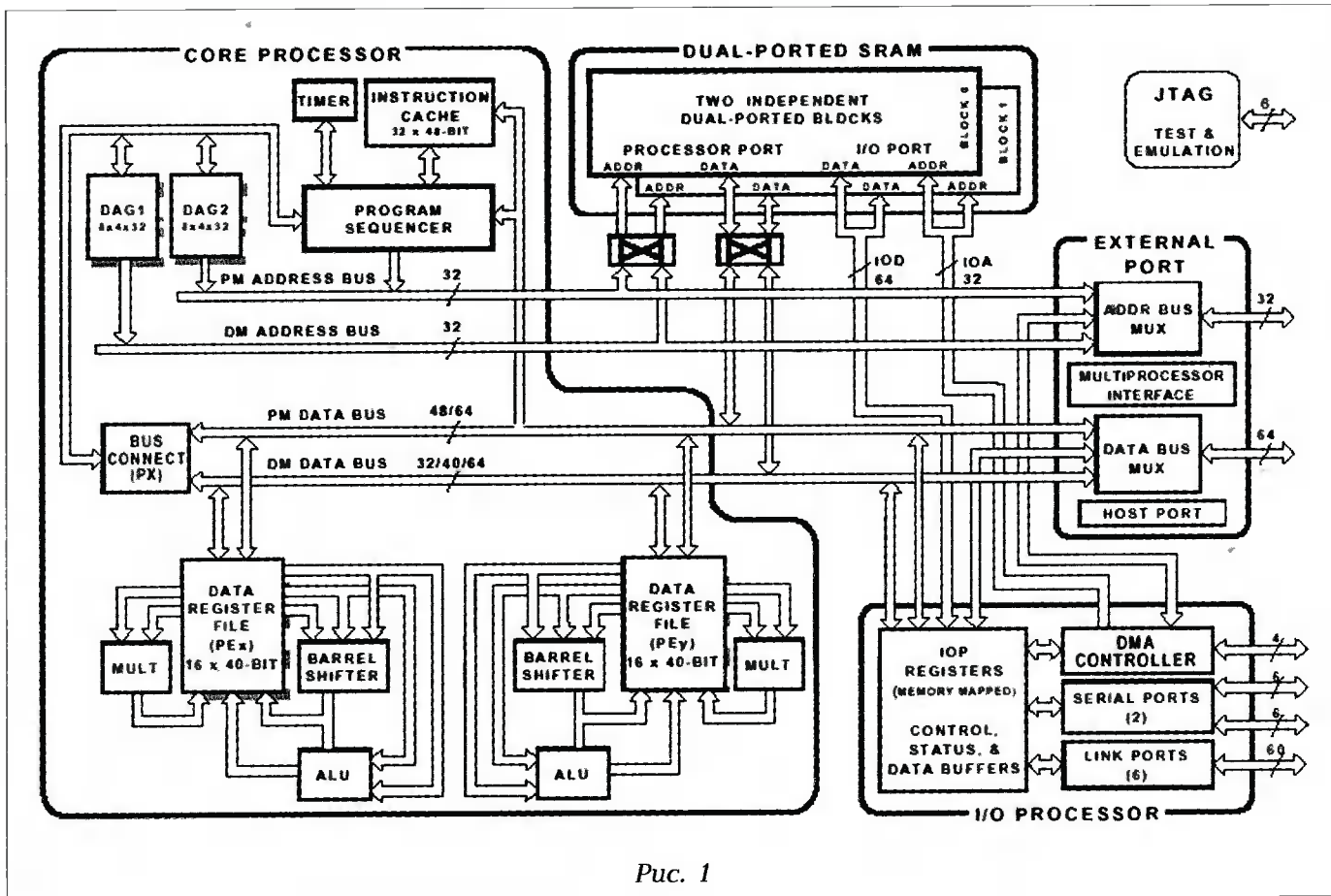


Рис. 1

Каждое командное слово имеет размер 48 бит. При использовании режима SIMD, это командное слово поступает на оба PE, которые начинают параллельные вычисления. Адресный генератор производит последовательную выборку данных через 64-битную шину (по 32 бита на каждый PE) до завершения всего цикла обработки массивов данных.

Процессор ADSP-21160 содержит конвейер на 3 команды. При программировании отложенных переходов необходимо указать 2 команды.

Процессор также использует специальную логику для исключения задержки выполнения циклов. В одном такте процессор может выполнить инструкцию, проверить условие завершения цикла, произвести декремент счетчика цикла и выполнить переход. Во время выполнения любого цикла может быть обслужено прерывание. Допускается вложенность до 6 циклов.

Любая команда процессора ADSP-21160 выполняется за 1 цикл.

Главным и определяющим фактором, влияющим на производительность вычислительной системы ЦОС, построенной на процессорах ADSP-21160, является оптимальное распределение памяти и использование режима SIMD. Так, если вычисления производятся над массивами данных, расположенными во внутренней памяти и распределенными таким образом, что операнды для команд находятся в различных банках этой памяти, то в режиме SIMD за один такт процессора одной командой можно выполнить следующие операции:

- 4 сложения (вычитания);
- 2 умножения;
- 2 чтения и 2 записи 32-битных слов памяти;
- 2 чтения или 2 записи 32-битных слов памяти с использованием ПДП.

Очень важно, что процессор ADSP-21160 позволяет в фоновом режиме, с использованием каналов ПДП, производить обмен массивами данных между внутренней памятью и внешними устройствами. Это позволяет доставлять исходные массивы во внутреннюю память и получать результаты вычислений, не требуя дополнительных тактов вычислительного устройства.

Если же аналогичные вычисления производить над массивами, расположенными во внешней памяти, то указанная команда выполняется не менее чем за 4 такта процессора:

- 2 такта чтения;
- 2 такта записи;
- 4 сложения и 2 умножения за счет конвейера можно производить на фоне чтения (записи) следующих операндов.

При этом, если на внешней шине данных производятся какие-либо операции, например, чтение данных из устройства ввода (АЦП), или обращение к внешней шине со стороны другого процессора в кластере, то производительность системы снижается на порядок и более.

Рассмотрим также особенности режима SIMD. В обычном режиме процессор ADSP-21160 производит вычисления аналогично процессору пре-

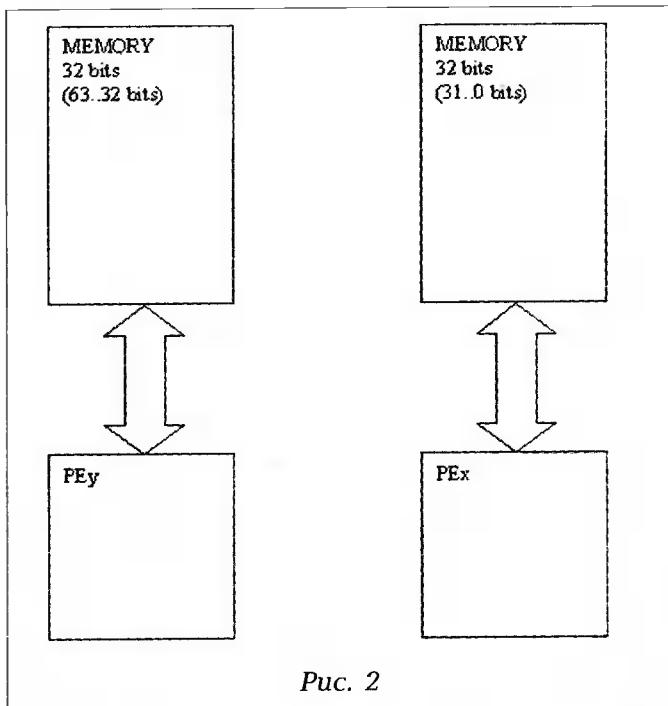


Рис. 2

дыдущего поколения ADSP-21060, то есть используется только один процессорный узел PEx, которому доступна вся память как 32-битная. В режиме SIMD в вычислениях задействуется второй процессорный узел PEy, но при этом появляются ограничения по доступу к памяти. Модель доступа к памяти в режиме SIMD условно представлена на рис. 2.

Из рис. 2 видно, что процессор в режиме SIMD представляет собой два изолированных процессора, каждый из которых имеет свою память, недоступную другому процессору. При этом старшие 32 бита 64-битной памяти доступны только процессору PEy, а младшие 32 бита — только процессору PEx. Отсюда следует, что максимальная эффективность вычислений достигается при одновременной обработке 2-х несвязанных наборов данных и использовании каналов ПДП для обмена массивами данных между внутренней памятью и внешними устройствами, в том числе с внешней памятью.

Варианты вычислительных архитектур на процессорах ADSP-21160

Одной из важнейших характеристик процессора ADSP-21160 являются его встроенные многопроцессорные возможности. Рассмотрим две базовые архитектуры — кластерную (с общей шиной) и распределенную.

Кластер процессоров ADSP-21160

Каждый процессор ADSP-21160 имеет встроенный арбитр шины, позволяющий без дополнительной аппаратуры объединять на общей шине до 6-ти процессоров. Причем на шине доступны внутренние ресурсы каждого процессора (внутренняя память и регистры ввода-вывода). Упрощенная блок-схема вычислительной системы, построенной на данной архитектуре, представлена на рис. 3.

Из рис. 3 видно, что вычислительная система состоит из 4-х основных компонентов:

- подсистема ввода (Input)
- вычислительная подсистема (CPU1..CPU6)
- подсистема памяти (Memory)
- подсистема вывода (Output)

Главное преимущество данной архитектуры при использовании процессоров ADSP-21160 — простота реализации.

Однако далеко не все задачи ЦОС могут эффективно решаться при использовании данной архитектуры.

Для получения максимальной эффективности многопроцессорной вычислительной системы необходима полная загрузка всех процессоров при параллельных вычислениях. Причем важно минимизировать непроизводительные потери времени, такие, например, как чтение-запись памяти, преобразование форматов данных, конкуренция за доступ к общему ресурсу и т.д.

Как было указано выше, для получения максимальной производительности процессора ADSP-21160 требуется оптимальное размещение программы и массивов данных для обработки обяза-

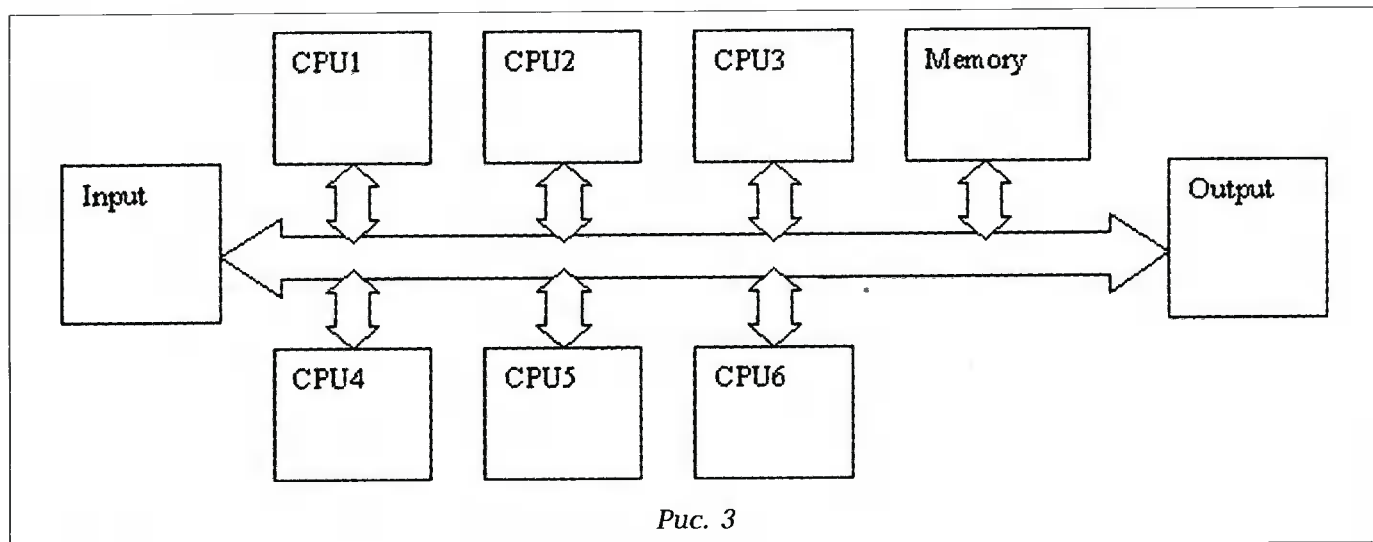


Рис. 3

тельно во внутренней памяти, и использование режима SIMD.

Например, пусть подсистема ввода принимает данные от 2-канального аналого-цифрового преобразователя (АЦП). Необходимо вычислить быстрое преобразование Фурье с наложением окна и отправить результат на Host-процессор через подсистему вывода. Если для заданного спектрального разрешения требуется менее 8К отсчетов входного сигнала, то все вычисления могут производиться во внутренней памяти процессора.

Для организации параллельных вычислений очередной блок входных отсчетов принимается во внутреннюю память очередного процессора в кластере. Спектральное окно заранее подгружается из внешней памяти во внутреннюю память. Обработка данных производится в режиме SIMD одновременно для 2-х каналов данных. Результаты обработки поочередно отправляются через подсистему вывода на Host-процессор. Здесь важно отметить, что для повышения эффективности вычислений целесообразно использование интеллектуальной подсистемы ввода данных, построенной, например, на современных программируемых логических интегральных схемах (ПЛИС) достаточного объема и быстродействия. Такая подсистема ввода может выполнять самые разнообразные типовые операции на проходе от АЦП к памяти процессора. Например, можно реализовать цифровую фильтрацию или в простейшем случае для процессора ADSP-21160 эффективно производить преобразование входных данных из формата с фиксированной точкой в формат с плавающей точкой.

В приведенном примере на эффективность вычислений не оказывает влияния главный недостаток кластерной архитектуры — конкурентный доступ к общим ресурсам. Здесь на шине производится только ввод данных и вывод результатов обработки.

Для подобных задач представленная архитектура имеет максимальное соотношение производительности и стоимости.

Распределенная система на процессорах ADSP-21160

Многие задачи ЦОС не укладываются в приведенную выше схему и требуют интенсивного обмена данными с внешней памятью. В этом случае кластерная архитектура может значительно снизить производительность вычислительной системы. Для таких задач наиболее эффективной является распределенная архитектура.

Примерная блок-схема распределенной системы на процессорах ADSP-21160 представлена на рис. 4.

Основные недостатки представленной архитектуры — сложность и высокая стоимость. Поэтому, для построения эффективной системы сле-

дует тщательно прорабатывать конкретную реализацию данной вычислительной архитектуры применительно к конкретной задаче или классу задач ЦОС. Так, большие трудности могут быть связаны с реализацией подсистемы ввода данных. Это должен быть достаточно сложный логический автомат, выполняющий распределение входного потока данных на каждый из процессоров в системе. Логика работы данного автомата привязана к требованиям конкретной задачи и целесообразна его реализация на ПЛИС. Следует отметить, что здесь может потребоваться достаточно большое количество таких ПЛИС в зависимости от характера входного потока данных. Так, если поток данных относительно небольшой, и он может быть принят через линки процессоров ADSP-21160, то логическая схема получается значительно проще, чем в случае приема высокоскоростного потока данных через параллельную 64-битную шину. Однако, в случае накопления во внешней памяти входных данных при использовании линков «отнимается» ресурс процессора. Наиболее целесообразным представляется использование подсистемы ввода данных по параллельной шине с использованием ПЛИС в подканалах каждого из процессоров, обеспечивающих гибкую перестройку системы на различные форматы входных потоков данных.

Не менее сложным является построение подсистемы вывода результатов обработки. Но, как правило, результирующий поток значительно менее интенсивный, чем входной. Это позволяет упростить данную подсистему. Так, наиболее целесообразным представляется использование в подсистеме вывода такого же процессора ADSP-

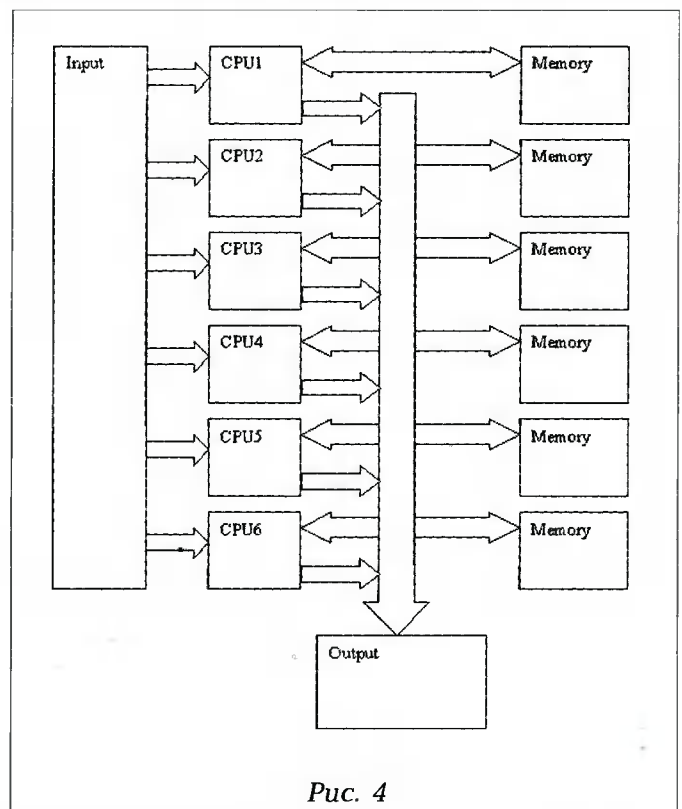


Рис. 4

21160, как и в вычислительной подсистеме. Он может принимать результаты обработки от других процессоров, используя линки.

Другим очень важным моментом является оптимальный выбор внешней памяти. Так, очевидно, что наибольшее быстродействие можно получить при использовании быстрой статической памяти, но в этом случае стоимость системы может оказаться настолько велика, что это перечеркнет все преимущества в быстродействии. Может оказаться гораздо эффективнее в плане соотношения производительности и стоимости использование синхронной динамической памяти. Главная особенность данного типа памяти — быстродействие, аналогичное статической памяти при доступе к массиву данных, расположенных по последовательным адресам. В данном случае требуется серьезная работа по адаптации существующих алгоритмов к использованию такой памяти, а именно, тщательная проработка размещения данных в памяти и эффективная организация обмена данными между внутренней и внешней памятью процессора ADSP-21160. Все это значительно увеличивает нагрузку на алгоритмиста и программиста. Что является эффективней: переработка ал-

горитма и затраты на программное обеспечение или значительное удорожание аппаратуры, решается отдельно в каждом конкретном случае. Нам представляется первое — лучше один раз затратить ресурсы на разработку соответствующих алгоритмов и далее тиражировать более дешевую вычислительную систему.

Заключение

Процессоры ADSP-21160 имеют все необходимые встроенные возможности для построения различных параллельных архитектур. Наиболее общими из таких архитектур являются кластерная и распределенная. Каждая из них имеет свои достоинства и недостатки. Эти архитектуры могут также дополнять друг друга при решении конкретной задачи. Любой разработчик прикладной задачи ЦОС, имея в своем распоряжении 2 типа аппаратуры указанных архитектур, может с максимальной эффективностью (в части соотношения стоимости и производительности) решить любую задачу. Важным моментом здесь является возможность гибкой настройки универсальной аппаратуры под конкретные алгоритмы и потоки данных.

5-я Международная Конференция и Выставка ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ЕЕ ПРИМЕНЕНИЕ

12 - 14 марта 2003 г., Москва, Россия

The 5th International Conference and Exhibition

DIGITAL SIGNAL PROCESSING AND ITS APPLICATIONS

ОРГАНИЗАТОРЫ:

- ◆ Российское научно-техническое общество радиотехники, электроники и связи им. А.С. Попова;
- ◆ IEEE Signal Processing Society;
- ◆ Российская секция IEEE;
- ◆ Отделение информатики, вычислительной техники и автоматизации РАН;

- ◆ Институт радиотехники и электроники РАН;
- ◆ Институт проблем управления РАН;
- ◆ Институт проблем передачи информации РАН;
- ◆ ФГУП НИИ Радио;
- ◆ АО «AUTEX Ltd.».

ОСНОВНЫЕ НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- ◆ Теория сигналов, методы кодирования, алгоритмы преобразований, цифровая фильтрация и спектральный анализ;
- ◆ Цифровая обработка многомерных сигналов;
- ◆ Цифровая обработка речевых и звуковых сигналов;
- ◆ Цифровая обработка изображений;
- ◆ ЦОС в системах телекоммуникаций, радиолокации и гидроакустики;
- ◆ ЦОС в системах управления;

- ◆ Цифровая обработка измерительной информации;
- ◆ Нейрокомпьютерная обработка сигналов;
- ◆ Цифровое телерадиовещание;
- ◆ Защита информации и ЦОС;
- ◆ Проектирование и техническая реализация систем ЦОС;
- ◆ Проблемы подготовки специалистов в области цифровых технологий обработки сигналов.

ЗАЯВКИ И ТЕКСТЫ (ТЕЗИСЫ) ДОКЛАДОВ должны быть представлены в одном экземпляре (на русском и английском языках) с указанием научного направления доклада и сведений об авторах (Ф.И.О., должность, место работы, почтовый адрес для переписки, телефон, факс, e-mail) на дискете (3.5 дюйма, Word 97, Word 2000), распечатке на принтере или переданы по электронной почте.

ДОКЛАДЫ, включенные в Программу конференции, будут опубликованы в Трудах конференции на русском и английском языках до начала ее работы.

ОБЪЕМЫ представленных докладов: на русском языке — до 3 стр., на английском языке — до 1 стр. (Шрифт 12).

СРОКИ ПРЕДСТАВЛЕНИЯ ДОКЛАДОВ

15 ноября 2002
Тексты докладов

15 декабря 2002
Извещение авторов о приеме докладов в программу

15 февраля 2003
Рассылка приглашений

Справки по телефонам: (095) 921-1639, 925-4155, 921-0610
E-mail: ntores@online.ru

Analog Devices: новые разработки DSP

С. В. Витязев

Процессоры ADSP-2195, ADSP-2196 — новые недорогие решения для телекоммуникационного оборудования

Компания Analog Devices подготовила к выпуску две новые модели класса ADSP-219x - процессоры обработки сигналов ADSP-2195 и ADSP-2196. Модели относятся к недорогим DSP. При их разработке была сохранена кодовая совместимость с другими процессорами семейства ADSP-21xx.

Отличительной особенностью новых DSP является наличие внутрикристалльного ПЗУ. ADSP-2195 сочетает на своём кристалле 32 Кбайта слов памяти SRAM и 16 Кбайт слов ПЗУ программ. ADSP-2196 имеет 16 Кбайт слов SRAM и 16 Кбайт слов ПЗУ программ. Наличие ПЗУ программ на кристалле позволяет хранить необходимые коды, ориентированные непосредственно на конкретное приложение, устраняя необходимость во внешних устройствах памяти. Это значительно снижает затраты на создание таких устройств, как модемы, IP-телефоны, устройства интегрированного доступа (IAD), оборудование для видеоконференций и другое пользовательское телекоммуникационное оборудование.

Следует отметить, что обе представляемые модели являются совместимыми по разъёмам с новейшим мощным процессором ADSP-2191. Это позволит создать широкий ряд продукции на базе одной аппаратной платформы и быстро наладить полномасштабное производство конечных изделий. Процессор ADSP-2191 имеет область оперативной памяти программ, в которой могут напрямую отображаться устройства ПЗУ ADSP-2195,



Рис. 1. Архитектура процессоров ADSP-2195, ADSP-2196

ADSP-2196. Эта важная особенность позволяет вести разработку программных кодов на ADSP-2191 с последующим переносом на ПЗУ кристалла ADSP-2195, ADSP-2196. При этом гарантируется адекватность программы в ПЗУ разработанному алгоритму и исчезает риск увеличения времени ожидания, что снижало производительность в других процессорах с ПЗУ на кристалле.

Архитектура процессоров ADSP-2195, ADSP-2196 изображена на рис. 1. В табл. 1 приведены их основные характеристики.

Таблица 1

Параметр	ADSP-2195	ADSP-2196
Разрядность	16 бит	
Формат	Фиксированная точка	
Рабочая частота	160 МГц	
Производительность	160 MIPS	
ОЗУ программ	48 Кбайт слов	24 Кбайт слов
ОЗУ данных	32 Кбайт слов	16 Кбайт слов
ПЗУ	48 Кбайт слов	48 Кбайт слов
DMA	11 каналов	
Таймеры	3 32-разрядных таймера	
Последовательные порты	3 многоканальных порта	
Параллельные порты	16 Мбайт слов адресного пространства	
Хост-порт	8 разрядов/16 разрядов	
Начальная загрузка	Внутрикристалльная	
SPI-порты	2	
UART	1	
Ввод-вывод общего назначения	16 контактов	
Напряжение питания ядра	2.5 В	
Напряжение питания УВВ	3.3 В	

Отметим, что модель ADSP-2196 можно назвать самой производительной на сегодняшний день по скорости вычислений из всех процессоров, цена которых не превышает \$10.

Сферы возможного применения процессоров:

- модемы;
- VoIP-телефония;
- громкоговорящая связь;
- контрольно-измерительная аппаратура;
- оптические сети;
- медицинское оборудование;
- устройства автоматического голосового управления;
- устройства доступа в Internet;
- устройства телеметрии.

Компания ADI предоставляет библиотеки программных продуктов к процессорам ADSP-2195, ADSP-2196. В частности, могут быть приобретены программные модули, реализующие алгоритмы кодирования/декодирования речи и эхо-компенсации, разработанные в полном соответствии с рекомендациями Международного союза электросвязи и обладающие высокими показателями качества функционирования. Их использование позволит значительно ускорить создание высокоэффективных, приемлемых по себестоимости и легко модернизируемых систем телекоммуникаций следующего поколения. Такие алгоритмы, как: G.165, G.168, G.711, G.722, G.723, G.726, G.728, G.729, DTMF — лишь немногие из тех, которые могут быть легко интегрированы в разрабатываемые системы и устройства. Кроме того, компании-партнёры ADI обеспечивают эффективную программную поддержку устройств ADSP-2195, ADSP-2196 и предлагают широкий спектр пакетов программ, библиотек функций, операционных систем реального времени, графических программных средств, программ фильтрации и других.

Analog Devices предоставляет полный набор инструментальных средств проектирования и отладки систем ЦОС на базе ADSP-2195, ADSP-2196, именуемый CROSS-CORE™. Этот набор включает программную среду VisualDSP++ 2.0, макетные платы EZ-KIT Lite и JTAG-эмуляторы.

Процессор ADSP-2195 имеет стоимость \$12.50, а процессор ADSP-2196 - \$9.95 при заказе от 10 000 изде-

лий. Оба устройства предлагаются в корпусе Mini-BGA и в стандартном LQFP-корпусе.

ADSP-21532 — новый DSP семейства Blackfin

Новый процессор компании Analog Devices - ADSP-21532, продолжает серию Blackfin — высокопроизводительных, малопотребляющих и эффективных в использовании устройств цифровой обработки сигналов, предназначенных для построения систем ЦОС следующего поколения. Процессор ориентирован на разработки, где критерий цены является существенным. Он оценивается в \$10 и значительно повышает доступность технологии Blackfin. Отличительной особенностью новой модели является усовершенствование системы динамического управления расходом мощности. Теперь изменение уровня подаваемого напряжения будет производиться внутрикристалльным регулятором. Это упростит конструкцию и понизит себестоимость продукции. Кроме того, процессор ADSP-21532 характеризуется наличием встроенных специализированных интерфейсов: I2S - аудио и CCIR-656 - видео, делая удобным его применение в системах потребительского/профессионального аудио и в носимых малогабаритных видеоустройствах.

Процессор ADSP-21532, общая структура которого приведена на рис. 2, имеет следующие характеристики:

- 16-разрядное ядро с фиксированной точкой;
- производительность/рабочая частота - 600 MMAC/300 МГц;
- SRAM/кэш-память программ уровня L1 - 48 Кбайт;

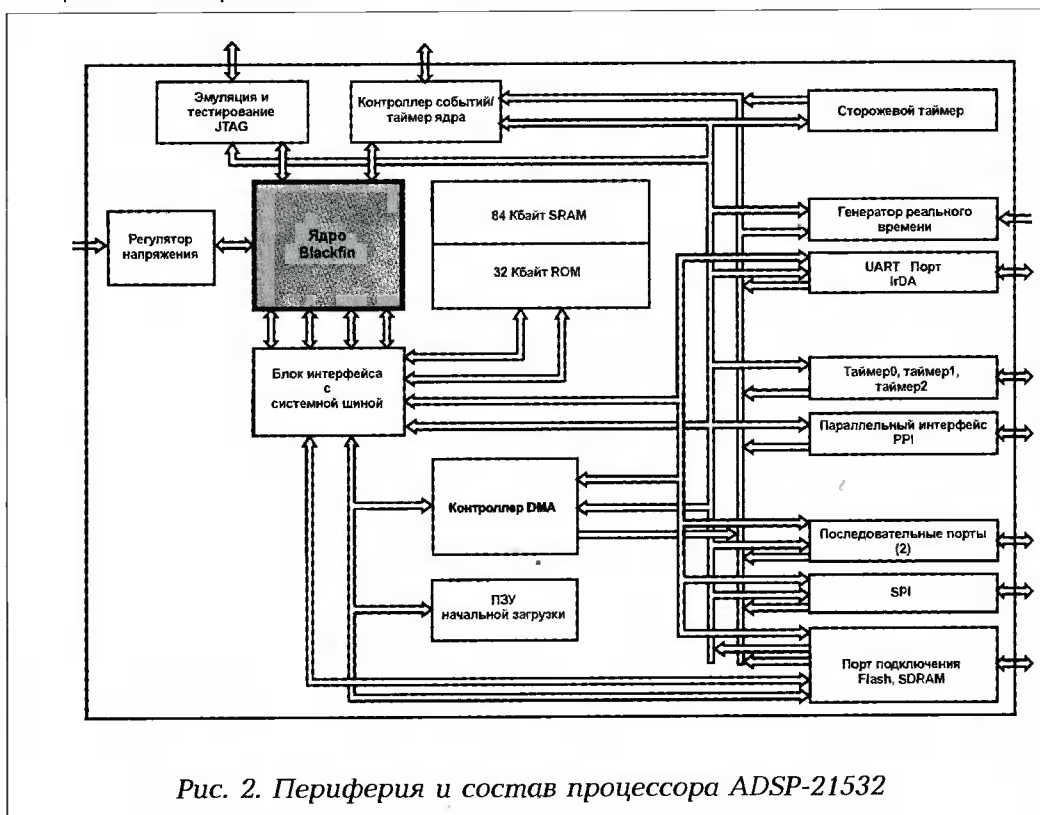


Рис. 2. Периферия и состав процессора ADSP-21532

- ПЗУ программ уровня L1 - 32 Кбайта;
- SRAM/кэш-память данных уровня L1 - 32 Кбайта;
- сверхоперативная SRAM на уровне L1 - 4 Кбайта;
- встроенная поддержка синхронных и асинхронных внешних устройств памяти;
- наличие блока управления ресурсами памяти, поддерживающего функции защиты;
- контроллер событий;
- параллельный интерфейс с периферией (PPI), поддерживающий формат видео данных CCIR-656;
- два двухканальных дуплексных синхронных порта последовательной передачи данных, способных поддерживать 8 стереоканалов I2S;
- SPI-порт;
- приёмопередатчик UART с поддержкой протокола IrDA;
- три таймера/счётчика общего назначения с возможностями ШИМ;
- таймер реального времени;
- сторожевой таймер;
- интерфейс отладки JTAG;
- схема PLL (повышение частоты тактирования в 1-31 раз);
- встроенный регулятор напряжения от источника 2.25-3.6 В;
- корпус MBGA (мини) со 160 выводами;
- диапазон рабочих температур: -25°C..+105°C.

Процессор ADSP-21532 имеет возможность функционировать в четырёх режимах потребления энергии, аналогичных режимам ADSP-21535 — «полная работоспособность», «активный режим», «сон», «глубокий сон». Это снижает общий расход энергии за счёт отключения некоторых блоков или функций в моменты, когда они не требуются. Дальнейшее снижение энергетических затрат осуществляется за счёт динамического регулирования напряжения питания ядра процессора и частоты тактирования всей периферии. В процессоре ADSP-21532 проведено разделение функциональных узлов по потребляемому напряжению, что повышает гибкость системы динамического регулирования питания. Внутренняя логика, за исключением генератора реального времени (RTC), питается от одного напряжения, RTC и УВВ кристалла — от другого и все остальные устройства ввода-вывода — от третьего. Независимость питания внутренней логики процессора максимизирует эффективность динамического регулирования его напряжения и позволяет использовать внутрикристальный регулятор, генерирующий внутреннее напряжение процессора на основе внешнего питания от источника 2.25 В-3.6 В. Управление напряжением осуществляется на основе содержимого программируемого регистра контроля регулятора напряжения.

Процессор содержит в своей структуре два двухканальных синхронных последовательных порта (SPORT), служащих для последовательного обмена данными с периферией и межпроцессорного взаимодействия. Эти порты характеризуются возможностью I2S-функционирования. Имея по два ряда независимых контактов приёма и передачи, каждый SPORT обеспечивает двунаправленную связь. Оба последовательных порта в совокупности предоставляют 8 стерео I2S-каналов для систем аудио.

Устройство ADSP-21532 также поддерживает параллельный интерфейс с периферией (PPI), благодаря чему может напрямую подсоединяться к параллельным АЦП и ЦАП, видео кодерам/декодерам и другой периферии. PPI способен функционировать в двух режимах: режиме общего назначения и в режиме CCIR-656. В режиме CCIR-656 PPI обеспечивает полудуплексную двунаправленную передачу данных по блокам до 10 бит.

Выпуск изделия ADSP-21532 на продажу планируется в 2002 году. Процессор будет поддерживаться полным набором инструментальных средств проектирования и отладки CROSS-CORE™. Ориентировочная стоимость изделия \$9.95 при заказе от 10 000 изделий.

Семейство процессоров TigerSHARC. Процессор ADSP-TS101S

Семейство сигнальных процессоров TigerSHARC компании Analog Devices ориентировано на применение в области телекоммуникаций. Это семейство процессоров характеризуется уникальной способностью обрабатывать 8-, 16- и 32-разрядные данные, как в формате с плавающей, так и с фиксированной точкой. Данное свойство является существенным, поскольку каждый из приведённых типов данных оптимален для тех или иных классов сигналов и алгоритмов. Например, широко применяемый декодер Витерби требует работы с 8-разрядными данными. В то же время, алгоритмы адаптивной коррекции и эккомпенсации нуждаются в повышенной точности и 32-разрядном представлении чисел. Таким образом, возможность обрабатывать данные различных форматов делает процессоры TigerSHARC достаточно гибкими и мощными, способными работать с различными по своей природе сигналами одновременно, эффективно реализуя протоколы следующего поколения (IMT-2000, xDSL и др.)

Процессоры TigerSHARC имеют суперскалярную архитектуру, способную поддерживать форматы RISC, VLIW и формат стандартного DSP. RISC-команды позволяют эффективно выполнять процедуры записи/хранения, повышать степень конвейеризации и скорость выполнения команд, осуществлять предсказание ветвлений и имеют другие достоинства. Формат VLIW (команды сверхбольшой длины) позволяет одной инструк-

цией организовать целый ряд действий, что часто делает использование программного кода более рациональным, особенно в случае команд управления.

Многие системы ЦОС, построенные на базе DSP, в частности системы телефонной связи, часто используют сложные структуры мультипроцессорных соединений, чтобы повысить работоспособность оборудования. Процессоры TigerSHARC имеют широкие возможности объединения в многопроцессорную сеть, благодаря наличию 64-разрядной шины внешнего интерфейса и специализированных связанных портов.

Отметим, наконец, высокую вычислительную производительность процессоров TigerSHARC. Рабочая частота ядра составляет 150 МГц, что позволяет ему осуществлять 1200 MMACs (миллионов операций умножения с накоплением) в секунду при работе с 16-разрядным целочисленным форматом; 300 MMACs при работе с 32-разрядными целочисленными данными и 900

MFLOPS (миллионов операций с плавающей точкой в секунду) при обработке 32-разрядных нецелочисленных данных.

Развитие данного семейства DSP компания Analog Devices планирует по трём направлениям:

- процессоры для сотовой 3G-связи, характеризующиеся частотой 250 МГц и возможностью одного кристалла полностью обеспечивать канал связи со скоростью 2 Мбит/с;
- процессоры для серверов VoIP, позволяющие повысить плотность передачи голоса;
- процессоры общего назначения, ориентированные на производительность 2.4 биллиона операций умножения с накоплением в секунду.

Архитектура процессоров TigerSHARC

Процессоры семейства TigerSHARC (рис. 3) содержат два блока вычислений, каждый из которых включает АЛУ, умножитель-накопитель, 64-разрядное устройство сдвига и файл регистров на тридцать два 32-разрядных слова. Арифмети-

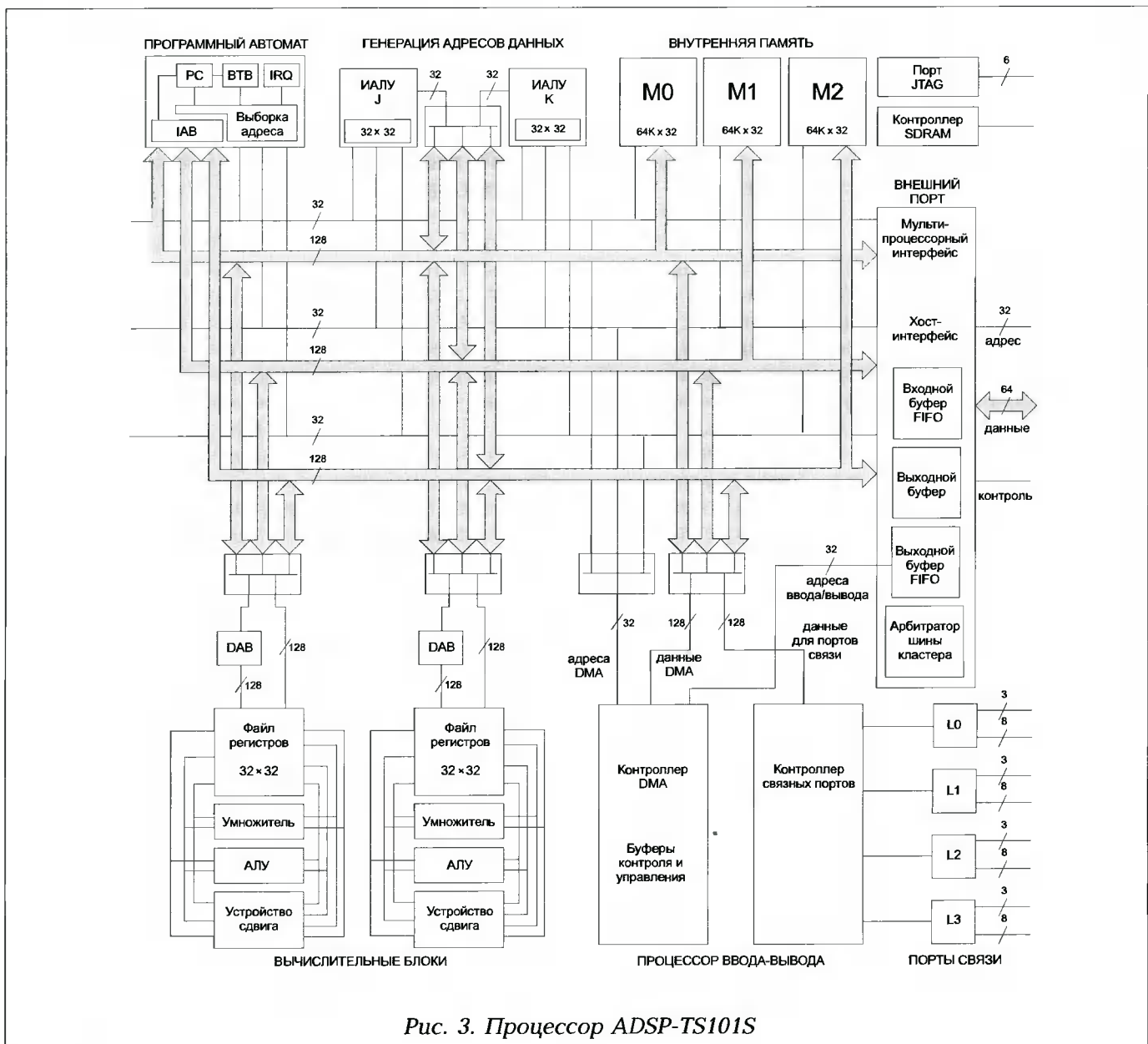


Рис. 3. Процессор ADSP-TS101S

ко-логическое устройство осуществляет стандартный набор операций и способно работать с числами в формате с фиксированной и плавающей точкой. Умножитель производит перемножение как целочисленных, так и нецелочисленных данных и умножение с накоплением для целочисленного типа. Устройство сдвига выполняет обычные функции и способно работать как с отдельными битами и словами, так и с потоком данных. Файлы регистров со множественным доступом используются для передачи данных между блоками вычислений и информационными шинами, а так же для хранения промежуточных результатов. Обращение команд к файлу регистров может осуществляться индивидуально или как к группам из двух или четырёх регистров (однословная, двухсловная и четырёхсловная выборка). Блоки вычислений содержат дополнительно 128-разрядные акселераторы, служащие для реализации декодирования решетчатого кода (например, в алгоритмах Витерби и Turbo-декодера) и операций комплексной корреляции, что часто требуется в телекоммуникационных алгоритмах.

Процессоры TigerSHARC включают в свой состав специализированные арифметико-логические устройства обработки целочисленных типов данных — ЦАЛУ. Эти блоки представляют собой мощный инструмент генерации адреса и целочисленных вычислений общего назначения. Блоки ЦАЛУ, как генераторы адреса, обеспечивают прямую и косвенную (с пред- и пост- модификацией) адресацию, а также адресацию по модулю и бит-реверсную адресацию. Адресация по модулю (циклическая буферизация) существенно повышает эффективность программирования линий задержки и других структур, часто требуемых алгоритмами ЦОС, в частности, при построении цифровых фильтров и реализации преобразования Фурье. ЦАЛУ предоставляют возможность эффективного использования до восьми циклических буферов с непрерывным считыванием и автоматическим циклическим возвратом. Блоки ЦАЛУ обеспечивают доступ к данным из памяти в четырёх- двух- и однословном формате.

Программный автомат процессоров TigerSHARC содержит буфер выравнивания команд (IAB). Этот буфер способен кэшировать до 5 выбираемых из памяти последовательностей команд, ожидающих выполнения. К другим особенностям программного автомата можно отнести прогнозирование ветвлений и наличие 128-разрядного целевого буфера ветвлений (TBT), позволяющих существенно сократить издержки выполнения условных и безусловных переходов.

Наличие буферов выравнивания данных (DAB) значительно повышает эффективность выполнения некоторых алгоритмов, в частности, КИХ-фильтрации. Буферы позволяют реализовать принцип SWP (sub-word parallelism) — парал-

лельную обработку данных малой длины в пределах каждого вычислительного блока — через них осуществляется загрузка регистров данными, подготовленными для параллельной обработки.

Процессор TigerSHARC имеет статическую суперскалярную архитектуру. Он способен в общем случае выполнять одновременно (за один цикл) от одной до четырёх 32-разрядных команд, выстроенных в единую последовательность. Параллелизм на уровне команд определяется перед выполнением и закодирован в программе.

TigerSHARC обладает возможностью работать по принципу SIMD (один поток команд / различные потоки данных), благодаря наличию двух вычислительных блоков. Также используется параллелизм на уровне данных для каждого блока вычислений в отдельности. При этом происходит параллельная обработка четырёх 16-разрядных или восьми 8-разрядных инструкций (каждым блоком).

Память процессоров (внутренняя) представляет собой 6 Мбит SRAM, разделённых на три блока по 2 Мбита (64 Кслов×32 бита) каждый. Блоки памяти могут хранить и программы и данные в любой комбинации, требуемой конкретной разработкой. При этом следует учитывать, что размещение программ и данных в разных блоках позволяет процессору выбирать данные из памяти одновременно с выборкой команды. Каждый блок памяти имеет свою 128-разрядную внутреннюю шину передачи данных, что позволяет за один цикл делать одновременно три выборки. Такая архитектура шин разрешает ядру и УВВ процессора доступ за один цикл к восьми 32-разрядным данным и четырём 32-разрядным командам (общая пропускная способность 8.64 Гбайт/с).

Наличие связанных портов делает возможным объединение любого необходимого числа процессоров TigerSHARC в общую сеть. Порты реализуют взаимодействие пары процессоров по выделенной линии передачи. Такой вид объединения процессоров в мультипроцессорную систему является преимущественным, если требуется высокая пропускная способность межпроцессорного интерфейса. Альтернативой такому подходу является объединение процессоров через общую шину параллельного подключения. Для обеспечения возможности такого соединения процессоры TigerSHARC имеют внешний порт с пропускной способностью 600 Мбайт/с. Общая пропускная способность межпроцессорного интерфейса может достигать, таким образом, 1200 Мбайт/с (связные порты плюс внешняя шина) при частоте тактирования 150 МГц.

Процессор ADSP-TS101S

Сигнальный процессор ADSP-TS101S TigerSHARC ориентирован на применение в аппа-

ратуре базовых станций сетей сотовой связи и другом телекоммуникационном оборудовании обработки немодулированных сигналов. Процессор имеет ряд особенностей, позволяющих существенно снизить себестоимость систем беспроводной 3G-связи и наделить их высокой работоспособностью и гибкостью выполняемых функций. ADSP-TS101S способен устранить необходимость дополнительной внешней памяти и специализированных схем ASIC и FPGA, предоставляя внутреннюю SRAM большого объёма и имея возможность брать на себя специфические функции обработки сигналов, различных по своей природе. Высокая степень программируемости процессора позволяет легко адаптировать его под выполнение задач, требуемых конкретным применением, и строить системы любых основных 3G-стандартов, включая WCDMA/UMTS, cdma2000 и TDS-CDMA. Исполнение функциональных узлов программным, а не аппаратным, способом даёт также возможность последующего перехода систем на базе ADSP-TS101S на новую алгоритмическую платформу, что защищает оборудование от морального износа.

Характеристики процессора и его особенности:

Производительность:

- рабочая частота 180 МГц;
- командный цикл 5.56 нс;
- выполнение восьми 16-разрядных операций умножения с накоплением 40-разрядного результата или двух 32-разрядных умножений с накоплением 80-разрядного результата за один цикл;
- 1080 MFLOPS (миллионов операций с плавающей точкой в секунду) или 4.3 GOPS (миллиардов операций с фиксированной точкой в секунду);
- глубина конвейеризации — 8 циклов (3 цикла — выборка; 5 циклов - выполнение);
- параллельное выполнение за один цикл до четырёх 32-разрядных команд.

Интеграция на кристалле:

- три блока внутренней SRAM (6 Мбит);
- контроллер DMA, разрешающий работу 14 каналов прямого доступа к памяти с нулевыми издержками;
- расширение набора команд специализированными инструкциями для обработки немодулированных сигналов;
- 3 внутренних 128-разрядных шины с общей пропускной способностью 8.64 Гбайта/с.

Возможности построения мультипроцессорных структур:

- встроенная поддержка мультипроцессорного взаимодействия — возможность подключения к одной внешней шине до 8 процессоров;
- 4 связанных порта на 180 Мбайт каждый. Общая пропускная способность 1.44 Гбайта/с.
- внутрикристалльная схема поддержки межпроцессорного взаимодействия снижает требования к дополнительному внешнему устрой-

ству управления интерфейсом.

Процессор ADSP-TS101S TigerSHARC будет выпускаться в корпусе PBGA с габаритами 19x19 мм и 27x27 мм. Макетная плата изделия должна появиться в начале 2002 года.

DSM2180F3 — системная память для сигнальных процессоров ADSP-218x

В июле 2001 года компания STMicroelectronics (один из крупных производителей полупроводниковых устройств) объявила о выпуске микросхемы системной памяти для цифровых сигнальных процессоров (DSM) семейства ADSP-218x Analog Devices. DSM представляет собой флэш-память, интегрированную с программируемым логическим устройством (PLD), портами ввода-вывода и другими блоками, реализующими функции логики и интерфейса. Устройство DSM2180F3 — первое в семействе DSM. Оно позволяет подключать к системе ЦОС и легко (без дополнительных компонент) конфигурировать флэш-память большой ёмкости, упрощая и ускоряя разработку систем и снижая их себестоимость.

Флэш-память, предоставляемая DSM, имеет общий объём 128 Кбайт, что является достаточным для хранения большого числа итераций программных кодов и данных. Память сконфигурирована как 8 секторов по 16 Кбайт каждый. Внутренняя логика обращения к памяти, которая может быть легко перепрограммирована, позволяет сигнальному процессору обращаться к флэш двумя способами: посредством побайтовой DMA-адресации (BDMA) или как к внешней оверлейной памяти данных. Первый способ эффективен при чтении больших массивов информации, например, при загрузке во внутреннюю SRAM процессора кодов программ или данных на этапе включения системы. Второй способ преимущественен для записи во флэш-память, так как не требует конфигурирования каналов DMA. При этом разбиение флэш-памяти на секторы по 16 Кбайт обеспечивает эффективность обмена данными, поскольку процессоры ADSP-218x для адресации внешней оверлейной памяти данных имеют окно такой же размерности (16 Кбайт). Логика постраничной адресации даёт возможность DSP обращаться напрямую ко всему пространству памяти 128 Кбайт. Всё это делает удобной запись во флэш при изменении программных кодов, параметров и других данных.

На 16 макроячейках встроенного в DSM программируемого логического устройства (PLD) могут быть реализованы различные счётчики, делители частоты тактовых импульсов, мультиплексоры; реализуются функции выбора кристалла (для внешних устройств), конечного автомата и интерфейса.

Всё это позволяет обходиться без дополнительных внешних PLD и других логических устройств.

16 контактов «ввод-вывод» устройства DSM дополняют возможности процессора ADSP-218х по вводу-выводу, повышая вдвое и даже втрое эффективность обмена данными. Ввод-вывод в DSM осуществляется под управлением сигнального процессора или в соответствии с логикой PLD.

Существенным достоинством устройств DSM является то, что все их компоненты (флэш-память, PLD и УВВ) легко конфигурируются программированием через JTAG-интерфейс. При производстве на монтажную плату устанавливается чистое DSM, которое легко программируется уже после сборки. Это устраняет необходимость в дополнительных дорогостоящих сокетках, предварительно запрограммированной флэш-памяти, различных логических устройствах; ускоряет процесс разработки и производства конечной продукции и снижает затраты начального изготовления и последующей модернизации. Программной средой проектирования и конфигурации памяти DSM является оболочка PSDsoft Express™, которую бесплатно можно загрузить на свой ПК через Internet: <http://www.psdst.com/html/softcenter2.htm>. Эта среда, сопровождаемая различной справочной документацией, позволяет быстро и эффективно

создавать конфигурацию DSM-устройства в режиме «point-and-click» («подвёл указатель мыши — нажал кнопку»). На ПК создаётся объектный файл данных *.obj, который подлежит переносу на DSM. Данная операция выполняется посредством программатора (недорогого FlashLINK, PSDpro компании ST или программаторами других производителей), подключаемого к ПК или Laptop через параллельный порт.

DSM2180F3 идеально подходит для использования в устройствах, имеющих ограничения по габаритам, уровню электромагнитных излучений или по питанию. И сама флэш-память, и логические схемы DSM автоматически переходят в режим «нулевого питания» в промежутках между обращениями к памяти или выполнением логических функций. В данном режиме устройство DSM характеризуется малым уровнем протекающих токов — около 25 мкА. Это позволяет использовать DSM2180F3 в системах, питающихся от аккумуляторных батарей.

Более полную информацию по вопросам технологии и приобретения продукции компании Analog Devices можно получить через Internet: <http://www.analog.com/dsp> или <http://www.argus-soft.ru> и <http://www.autex.ru>.



ПАМЯТИ Петра Леонидовича Серебрянникова

Безвременно ушел из жизни наш коллега, друг, соратник, организатор и вдохновитель многих совместных проектов. Издание нового научно-технического журнала «Цифровая обработка сигналов» было одним из его последних проектов, которому он отдал немало жизненных сил и творческой энергии. С 2001 года при его непосредственном участии началось регулярное издание нашего журнала.

Петр Леонидович Серебрянников родился 22 ноября 1961 года в Москве. После окончания в 1984 году Московского института радиотехники, электроники и автоматики по специальности «Оптические и оптоэлектронные системы» работал старшим инженером в НИИ прикладной физики. Участвовал в разработке автоматизированных систем для определения параметров оптико-электронных приборов при сверхнизких температурах. В 1987 году возглавил творческую группу по разработке уникальных электронных устройств для сопряжения персональных ЭВМ с нестандартными устройствами ввода-вывода информации, а в 1995 году перешел на работу в АОЗТ «Полюс» в качестве руководителя информационно-аналитической службы. С 1998 года П. Л. Серебрянников становится научным и, вскоре главным редактором и издателем научно-технического

журнала «Метрология и измерительная техника в связи». С 2001 года П.Л.Серебрянников — издатель научно-технического журнала «Цифровая обработка сигналов».

На всех этапах трудовой деятельности П.Л.Серебрянникова отличали высокий профессионализм и ответственное отношение к делу. Он был разносторонним, талантливым человеком.

Светлая память о Петре Леонидовиче Серебрянникове будет жить в наших сердцах и в сердцах тех, с кем он работал и был тесно связан дружескими и деловыми контактами на протяжении его короткого жизненного пути.

Texas Instruments: новые разработки DSP

С. В. Витязев

Компания Texas Instruments представляет новый процессор и улучшает характеристики уже известных моделей семейства TMS320C67xx

Большая часть современного рынка ЦОС принадлежит мультимедиа системам, выполняющим обработку аудио и видео данных. Алгоритмы работы с такой информацией требуют огромной скорости вычислений, а также повышенной точности представления данных. Отвечая этим требованиям, компания Texas Instruments разработала новый 32-разрядный процессор с плавающей точкой, обладающий повышенной вычислительной производительностью и расширенным набором периферии, который будет служить хорошим решением при построении мультимедиа систем следующего поколения.

Сигнальный процессор TMS320C6713 (рис. 1) выполнен по технологии 0,13 мкм. Тактовая частота процессора составляет 225 МГц. Он способен выполнять до 1350 миллионов операций с плавающей точкой в секунду (MFLOPS) или 1800 миллионов команд в секунду (MIPS), а также, имея два

M-модуля, работающих как с целочисленными, так и с нецелочисленными данными, до 450 миллионов операций умножения с накоплением в секунду (MMACS).

На кристалле устройства интегрированы:

- 2 многоканальных последовательных аудиопорта (McASP) с возможностью организации до 16 стереоканалов формата IIS (Inter IC Sound), работой каналов на разной частоте тактовой и кадровой синхронизации и поддержкой формата S/PDIF;
- 2 многоканальных последовательных буферизированных порта (McBSP), поддерживающих высокоскоростное временное разделение каналов, интерфейс SPI и AC'97;
- 2 порта IIC (Inter IC Control), служащих для контроля устройств периферии, начальной загрузки из последовательной EEPROM или интерфейса с хост-процессором;

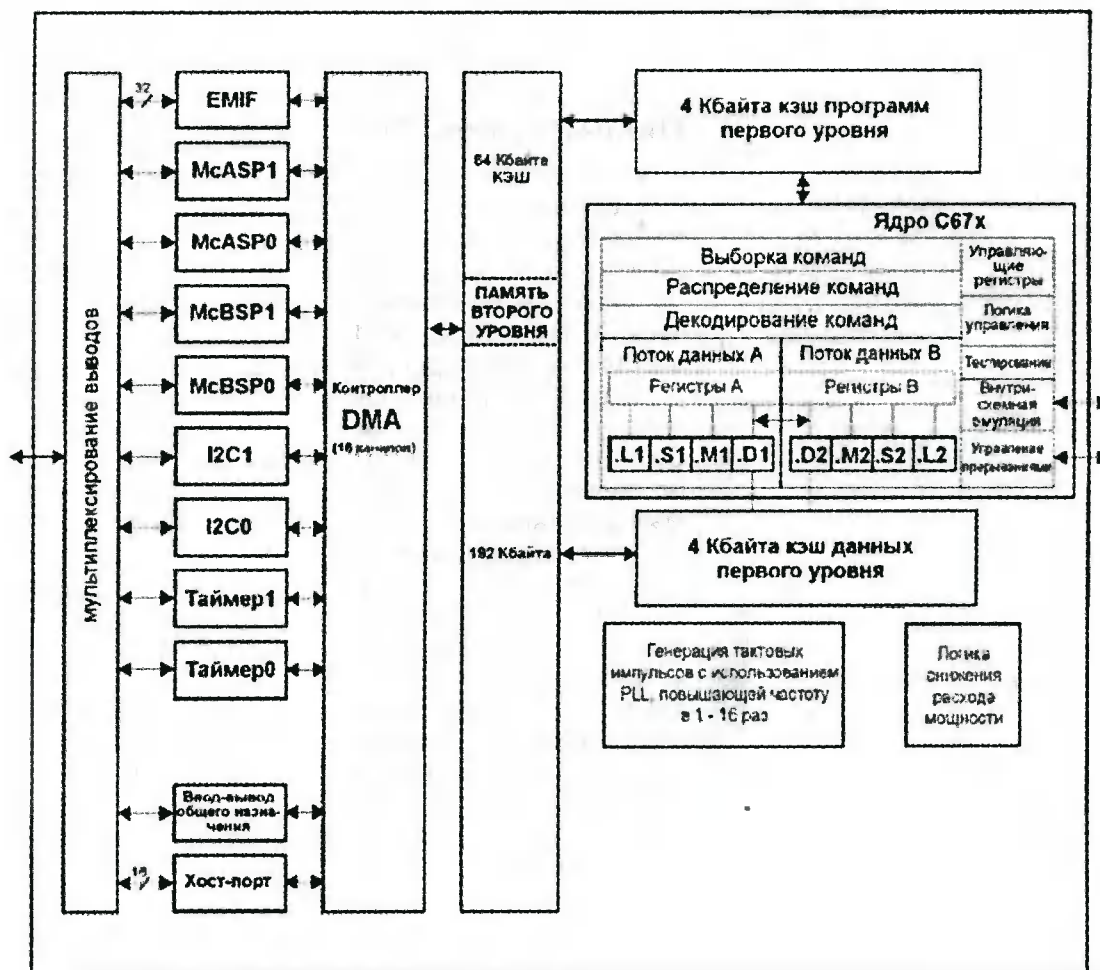


Рис. 1. Архитектура процессора TMS320C6713

- 2 таймера по 32 разряда общего назначения;
- процессор ввода-вывода общего назначения — 16 контактов;
- интерфейс с хост-машиной (HPI) — 16-разрядный порт;
- интерфейс с внешней памятью (EMIF), осуществляемый по 32-разрядной шине и обеспечивающий доступ к 512 Мбайтам адресного пространства;
- 16 каналов усовершенствованного контроллера прямого доступа к памяти (EDMA);
- память, включающая кэш программ и данных уровня 1 — по 4 Кбайта и 256 Кбайт памяти уровня 2, состоящей из 4 банков кэш (всего 64 Кбайта) и дополнительной RAM — 192 Кбайта.

Сферы применения нового процессора включают специализированные высококачественные аудиосистемы и аппаратуру, так называемого, «домашнего театра», где будет полезно использовать его исключительные возможности по выполнению как оригинальных, так и промышленных алгоритмов: Dolby Digital, MPEG-2, MP3, DTS и др., а также по реализации адаптивного анализа акустических свойств и их оптимизации. В медицинском оборудовании анализа изображений и системах работы с графикой процессор окажется эффективным в использовании при организации отображения, передачи, сжатия и повышения качества

данных в реальном времени. В устройствах робототехники он может обеспечивать высокую точность шага перемещения, выполняя до восьми 32-разрядных команд за цикл.

Процессор TMS320C6713 поддерживается инструментами проектирования eXpressDSP™, включая Code Composer Studio™ v2 и DSP/BIOS™ — ядро отладки в реальном времени. Компании-поставщики программных и аппаратных средств поддержки разработки также предлагают широкий ряд своих продуктов. В частности, компания MathWork, заявляет о возможности подключения к Code Composer Studio системы MATLAB и Simulink.

Процессор TMS320C6713 будет изготавливаться в корпусе BGA (256 выводов/27 мм) и TQFP (208 выводов). Начало поставок намечено на вторую четверть 2002 года. Однако уже сейчас можно приобрести стартовый набор разработчика по цене \$295. Ориентировочная стоимость самого процессора составляет \$26,85.

Компания Texas Instruments анонсирует выпуск популярных процессоров с плавающей точкой TMS320C6711 и TMS320C6712 со сниженной на 30% ценой и вместе с тем улучшенными показателями производительности. Процессор TMS320C6711 с частотой 200МГц (1200 MFLOPS)

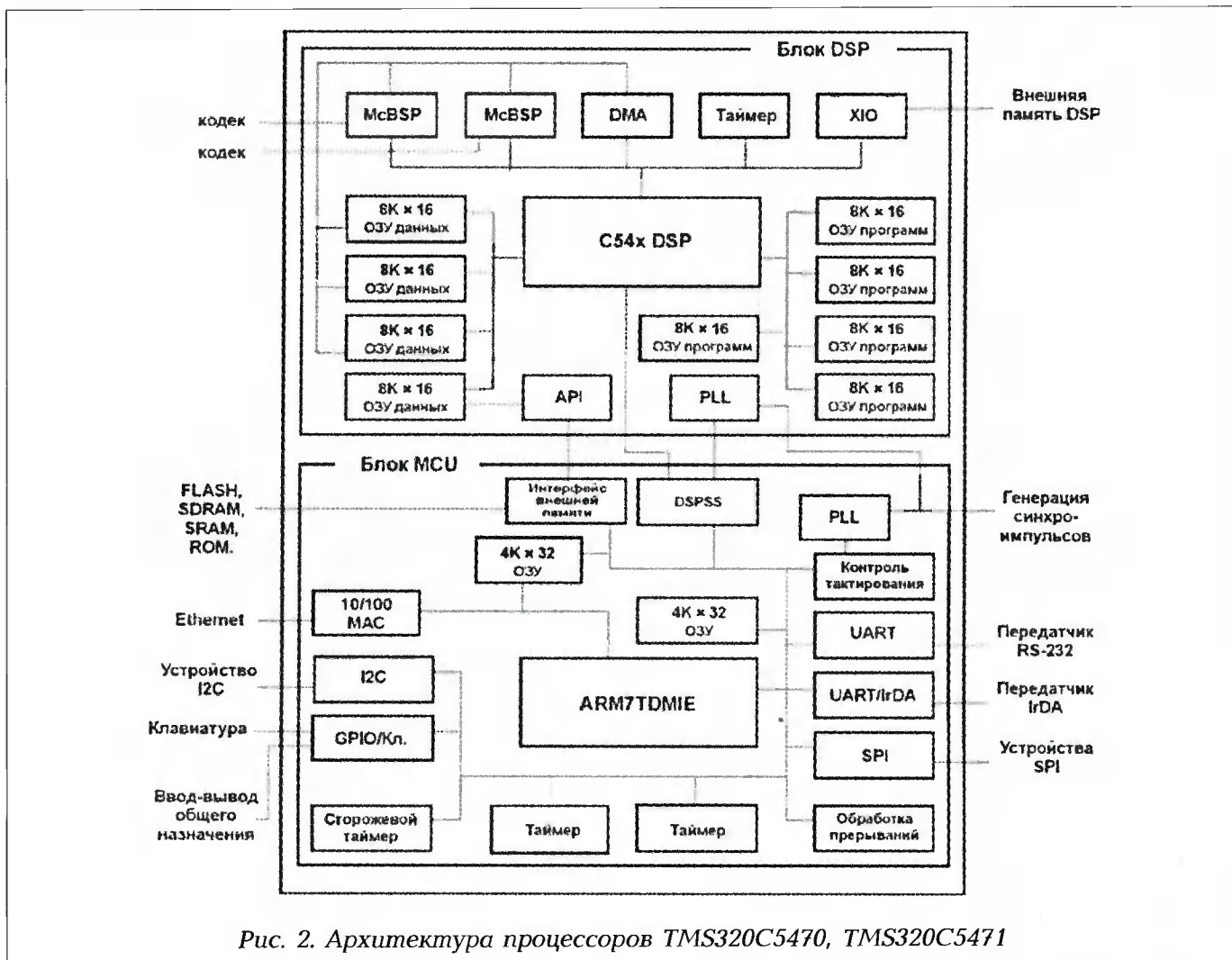


Рис. 2. Архитектура процессоров TMS320C5470, TMS320C5471

будет оцениваться в \$18,65, а процессор TMS320C6712 с частотой 150 МГц (900 MFLOPS) - в \$13,50.

Сигнальные процессоры TMS320C5470, TMS320C5471

К новейшим разработкам Texas Instruments в области DSP также следует отнести сигнальные процессоры TMS320C5470 и TMS320C5471. Процессоры предназначены для применения в таких сферах, как преобразование текст-речь, беспроводные системы передачи данных, распознавание голоса, безопасность в сетях, и, по мнению разработчиков, способны снизить себестоимость продукции и размеры устройств на 40% с уменьшением энергопотребления на 30%. Основная особенность процессоров заключается в том, что на одном кристалле интегрированы два различных по своим функциям вычислительных блока — традиционный DSP популярного семейства TMS320C54x и RISC-микроконтроллер ARM7 Thumb (рис. 2).

Интеграция двух блоков в пределах одной микросхемы даёт значительное уменьшение размеров конечной продукции, снижает потребление энергии, упрощает и ускоряет процесс производства и сокращает его себестоимость. При этом функциональная мощь устройства остаётся на высоком уровне.

Блок DSP имеет операционное ядро 16-разрядного процессора TMS320C54x с фиксированной точкой. Его рабочая частота составляет 100 МГц (100MIPS). Внутренняя оперативная память — 72К 16-разрядных слов. «Интеллектуальное» автоматическое управление расходом энергии и наличие режимов пониженного потребления дают существенную экономию по питанию. Периферия блока DSP включает:

- два высокоскоростных дуплексных многоканальных последовательных порта с буферизацией (McBSP), предназначенных для прямой связи DSP с кодеком;
- контроллер DMA на 6 каналов;
- порт интерфейса DSP с ARM (API), дающий доступ к их общему блоку памяти 8Кх16бит для эффективного обмена данными;
- интерфейс внешней памяти (XIO);
- программируемый генератор задержек (до 14 машинных тактов);
- программируемый аппаратный таймер для операций контроля и управления;
- программируемый тактовый генератор с PLL.

Блок микроконтроллера (MCU) включает ядро ARM7TDMI (enhanced ARM7TDMI) семейства ARM7 Thumb и набор периферии. Ядро микроконтроллера имеет улучшенные возможности эмуляции. Оно работает на частоте до 47,5 МГц. Внутренняя память составляет 16 Кбайт. К блоку MCU также применимы методы снижения энергозатрат. Периферия блока включает:

- контроллер доступа в сеть Ethernet 10/100 Мб/с IEEE 802.3 (на рисунке блок MAC) (**только у TMS320C5471 !**) — блок MAC;
- универсальный асинхронный приёмопередатчик (UART);
- интерфейс UART/IrDA;
- интерфейс SPI;
- сторожевой таймер;
- 36 контактов ввода-вывода общего назначения (GPIO), включающих 16 выводов поддержки специализированной клавиатуры (Кл.);
- интерфейс I²C (только в режиме «ведущий»);
- 2 таймера общего назначения;
- контроллер прерываний;
- интерфейс внешней памяти Flash, SRAM, SDRAM, ROM;
- гибкое управление тактированием периферии MCU;
- программируемый тактовый генератор с PLL.

Таким образом, оба блока представляют собой мощные системы обработки сигналов с богатым набором разнообразной периферии. При совместном решении задач каждый из блоков берёт на себя соответствующие своему предназначению функции, обеспечивая высокую производительность устройства в целом.

Характерными средствами разработки eXpressDSP для новых процессоров можно считать анонсируемую версию среды Code Composer Studio v2 OMAP, предоставляющую возможность работы с обоими блоками процессоров; технологию DSP/BIOS, выполняемую на блоке DSP; операционные системы реального времени DSPLinux, Nucleus PLUS и другие; средства анализа в реальном времени и библиотеки конфигурации периферии.

Стоимость процессоров составляет: 'C5470 - \$15.50; 'C5471 - \$17.57.

Набор разработчика потребительского оборудования телефонных систем связи

Компания Texas Instruments предлагает для уменьшения цены и сроков проектирования пользовательских систем телефонии воспользоваться специальным набором разработчика. Набор представляет собой макетную плату, строящуюся на базе сигнального процессора TMS320C54CST и включающую внешнюю память Flash и SRAM, последовательный интерфейс RS-232 и интерфейс JTAG. Плата подключается к телефонному каналу (КТФОП) через аналоговую схему интерфейса с телефонной линией (DAA), также входящую в набор. В память процессора зашиты 14 наиболее распространённых промышленных алгоритма обработки сигналов в пользовательских устройствах телефонии:

передача данных:

- модемы V.32 bis/V.32, V.22 bis/V.22;
- коррекция ошибок V.42;
- сжатие V.42 bis;

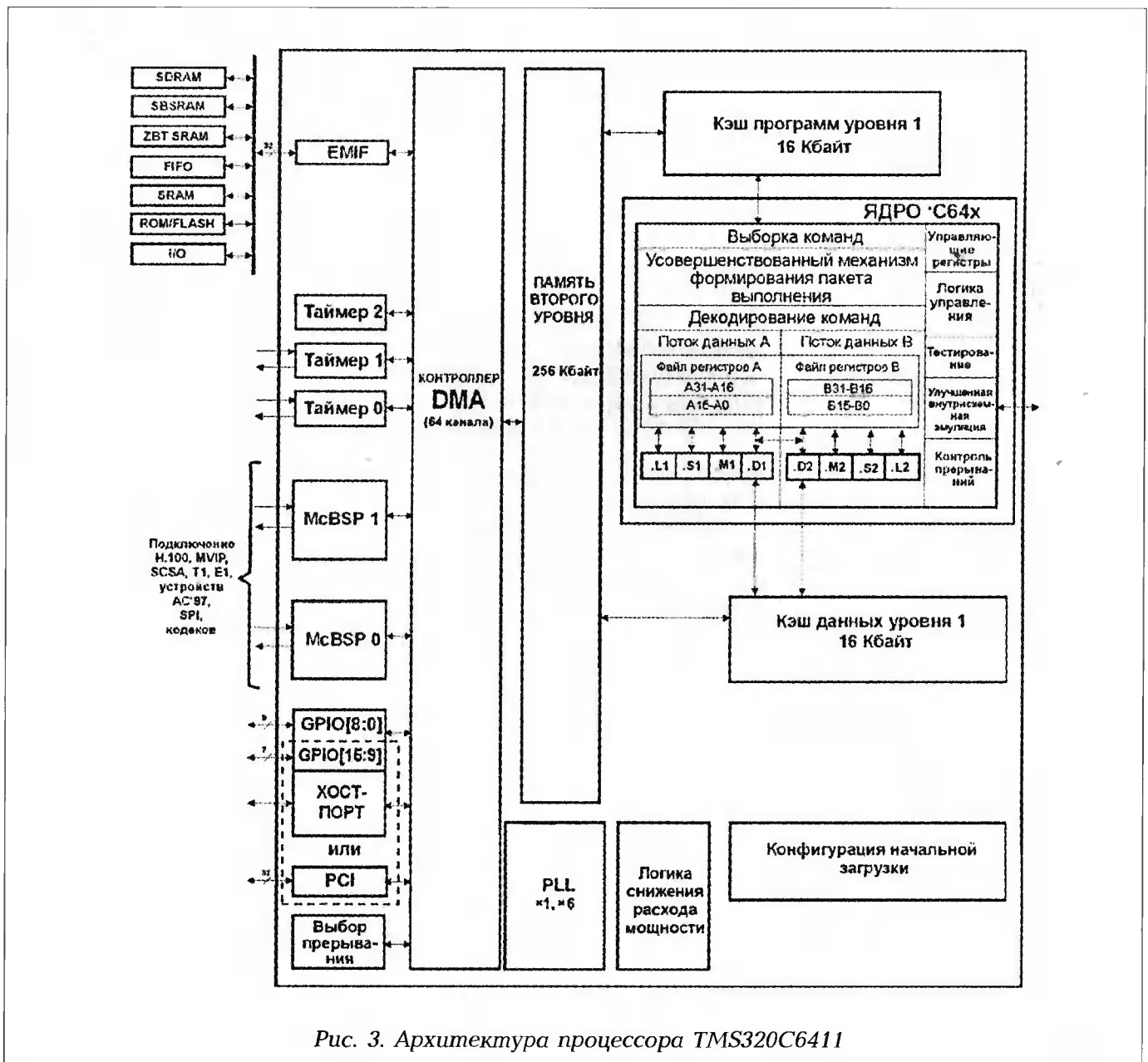


Рис. 3. Архитектура процессора TMS320C6411

общие алгоритмы телефонии:

- генерация DTMF;
- детектирование DTMF;
- функция Caller-ID типов 1 и 2;
- детектирование сигналов вызова и других тональных сигналов;

алгоритмы обработки голоса:

- эхо-компенсация G.165;
- сжатие G.726 ADPCM;
- PCM G.711;
- автоматическая регулировка усиления;
- функция определения голосовой активности;
- генерация комфортного шума.

Алгоритмы обработки сигналов были разработаны и реализованы компанией-поставщиком программного обеспечения SPIRIT Corporation (Россия).

Texas Instruments предоставляет всю документацию по схемотехнике, монтажу, программному обеспечению.

Процессор TMS320C6411 демонстрирует наивысшее отношение MMACS/\$ и MMACS/Вт среди всех промышленных процессоров с частотой 300МГц.

Компания Texas Instruments предлагает новый сигнальный процессор TMS320C6411 с наименьшим в своём классе энергопотреблением и высокой вычислительной производительностью. Сферы применения процессора относятся к многоканальной мультимедиа-обработке и включают: резидентные мультимедиа-серверы, системы наблюдения/безопасности, телекоммуникации, аппаратуру передачи данных и другие приложения.

Современные технологии построения систем ЦОС требуют от сигнальных процессоров не только высокой производительности, но и минимизации энергопотребления и стоимостного фактора. TMS320C6411 был разработан специально с учётом этих требований. Его операционное ядро по-

требляет 250 мВт при напряжении питания 1В. При этом оно работает на 300 МГц, обеспечивая 2400 MIPS/1200 MMACS. Вычислительную мощность процессора обеспечивают 8 независимых функциональных блоков — два умножителя и шесть АЛУ, работающих с файлом 64 регистров общего назначения. Построенный по технологии VelocityT1.2 процессор имеет ряд специальных команд, позволяющих более полно реализовать возможности ядра по скорости и параллельности выполнения операций. Память процессора строится по двухуровневой схеме, ориентированной на кэш. Память первого уровня включает 128 Кбит кэш программ с прямым доступом и 128 Кбит кэш данных со множественным доступом по 2 каналам. Память второго уровня образуют 2 Мбита программ/данных. Периферия процессора:

- 64 канала усовершенствованного ПДП (прямого доступа к памяти), обеспечивающих эффективность ввода-вывода;
- 32-разрядный интерфейс с внешней памятью, обеспечивающий высокую пропускную способность канала доступа к внешней синхронной и асинхронной памяти и периферии;
- два многоканальных последовательных буферизированных порта, служащих для аудио и телекоммуникационных интерфейсов;
- два 32-разрядных таймера с возможностями счёта внешних событий и один таймер, предназначенный для внутреннего использования;
- простое взаимодействие с хост-машиной — 16/32-разрядный порт, PCI, 33 МГц;
- ввод-вывод общего назначения - 16 контактов, программируемых на генерацию прерываний CPU или событий ПДП.

Процессор имеет корпус BGA 23 мм/532 вывода.

Перечисленные особенности процессора TMS320C6411 делают его идеальным решением для систем обеспечения безопасности, например, в устройствах сканирования глазного

яблока. Эта технология требует высочайшей вычислительной производительности, являясь в три раза сложнее алгоритма распознавания отпечатков пальцев. Однако с помощью TMS320C6411 такая задача решается достаточно легко. При этом высокоэффективный оптимизирующий компилятор Си, учитывая особенности архитектуры процессора, позволяет максимально реализовать его мощные возможности, что значительно ускоряет этап разработки систем.

Новый процессор совместим по своим кодам со всеми продуктами платформы C6000, гарантируя разработчикам надёжность технологий уже прошедших испытания в тысячах изделий. Процессор TMS320C6411 поддерживается средствами проектирования eXpressDSP, включающими Code Composer Studio v2 и ядро отладчика в реальном времени DSP/BIOS. Texas Instruments и компании-поставщики программных средств поддержки разработки предлагают широкий ряд библиотек алгоритмов обработки сигналов самых разных предназначений. Начать использовать новый процессор в своих разработках можно уже сейчас — процессор TMS320C6416 является полностью совместимым по своим кодам с TMS320C6411. Плата тестирования и оценки процессора TMS320C6416 выпускается теперь по сниженной цене — \$1995, а с полным набором программных средств Code Composer Studio и эмулятором XDS510 PP Plus — \$3995. Сам процессор должен стать доступным в инженерных образцах в третьем квартале 2002 года, а полномасштабный промышленный выпуск изделия планируется на четвёртый квартал 2002 года. Ориентировочная цена процессора TMS320C6411 — \$39. *За более подробной информацией по всем интересующим вас вопросам обращайтесь на сайт компании Texas Instruments: www.ti.com, а также на сайты ее официальных представителей в России: www.texas.ru.*